

MYSOHO PROJECT

박선교 김은진 서경빈
이준하 이지성 정희진

2023.05

순서

1. 마이소호 소개
2. 개발환경 및 일정
3. DB 구성
4. 프로젝트 구성
5. 화면 구성
6. 시연 영상

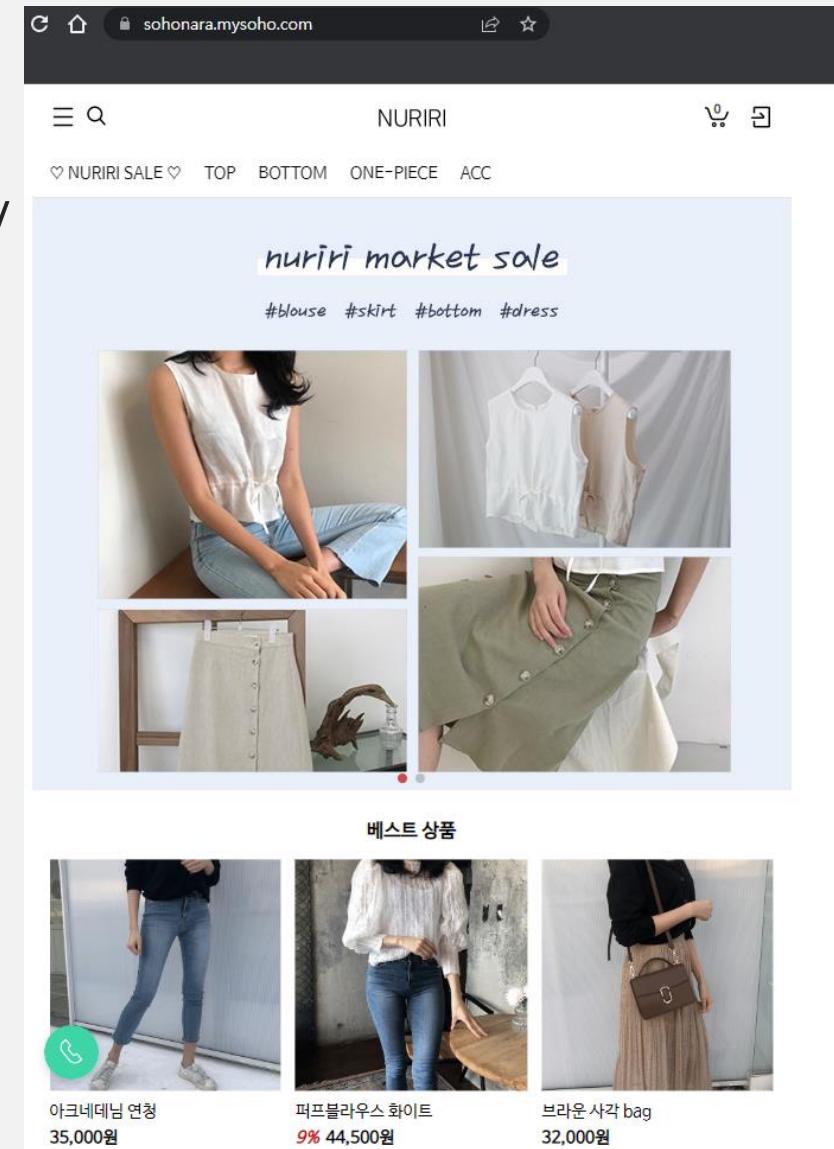
순 서

이름	마이소호
고객	10대~20대 여성
분류	캐쥬얼 여성의류
상품	<ul style="list-style-type: none">- TOP- BOTTOM- ONE-PIECE- ACC
도메인	미정

마이소호 샘플샵 소개



참고 사이트
<https://sohonara.mysoho.com/>



개발 환경

운영체제



Windows 10



개발 언어



서버



협업도구



GitHub

개발 도구



Visual Studio Code



GitHub Actions



배포



Amazon S3



Amazon
EC2

데이터베이스



MariaDB

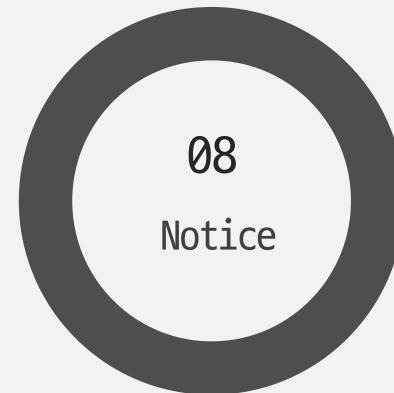


Amazon RDS

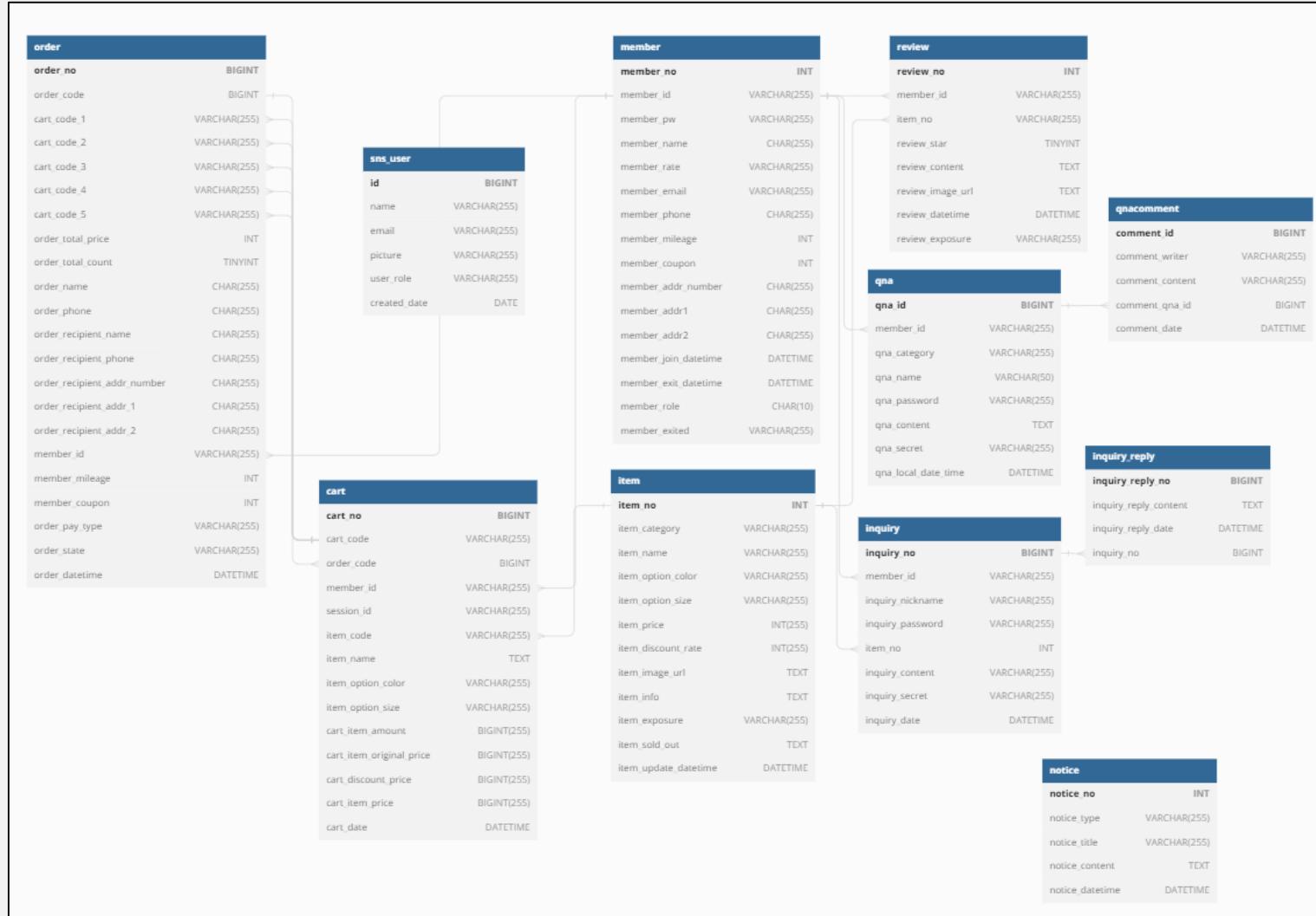
디자인 및 개발 진행

	1~2주	3~4주	5~6주	7~8주	9~10주
Front-end					
DB Table					
관리자페이지 (Back-end)					
사용자 페이지 (Back-end)					
Test/Debugging					
앱 프로그래밍 (Back-end)					
보고서/영상촬영/ 배포					

DB 설계



ERD



Member / SNS User



```
CREATE TABLE member (
    member_no INT PRIMARY KEY AUTO_INCREMENT, -- 고유키
    member_id VARCHAR(255) NOT NULL UNIQUE, -- 회원아이디
    member_pw VARCHAR(255) NOT NULL, -- 비밀번호
    member_name CHAR(255) NOT NULL, -- 회원이름
    member_rate VARCHAR(255) NOT NULL, -- 회원등급
    member_email VARCHAR(255) NOT NULL UNIQUE, -- 이메일
    member_phone CHAR(255) NOT NULL UNIQUE, -- 전화번호(중간에 -는 제거)
    member_mileage INT DEFAULT 0, -- 마일리지(적립금)
    member_coupon INT DEFAULT 0, -- 쿠폰
    member_addr_number CHAR(255) NOT NULL, -- 우편번호
    member_addr1 CHAR(255) NOT NULL, -- 기본주소
    member_addr2 CHAR(255) NOT NULL, -- 나머지 주소
    member_join_datetime DATETIME DEFAULT NOW(), -- 회원가입 날짜
    member_exit_datetime DATETIME DEFAULT NULL, -- 탈퇴 날짜
    member_role CHAR(10) DEFAULT 'ROLE_USER', -- 사용자/관리자
    member_exited VARCHAR(255) DEFAULT '회원' -- 탈퇴 여부(0: 회원, 1: 탈퇴 회원)
);
```



```
CREATE TABLE sns_user(
    id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL, -- 닉네임(별명)
    email VARCHAR(255) NOT NULL, -- 이메일(계정)
    picture VARCHAR(255) NULL, -- 프로필이미지 경로
    user_role VARCHAR(255) DEFAULT 'ROLE_USER',
    created_date DATE DEFAULT NOW()
);
```

Item



```
CREATE TABLE item(
    item_no INT PRIMARY KEY AUTO_INCREMENT , -- 고유item키
    item_category VARCHAR(255) NOT NULL, -- 카테고리 TOP, BOTTOM, ONE-PIECE, ACC
    item_name VARCHAR(255) NOT NULL, -- 상품이름
    item_option_color VARCHAR(255) NULL, -- 색상
    item_option_size VARCHAR(255) NULL, -- 사이즈
    item_price INT(255) NOT NULL, -- 가격
    item_discount_rate INT(255) NOT NULL, -- 할인율
    item_image_url TEXT , -- 이미지
    item_info TEXT DEFAULT '' , -- 상품설명
    item_exposure VARCHAR(255) DEFAULT '노출함' , -- 노출여부
    item_sold_out TEXT DEFAULT '판매중' , -- 품절여부 soldout
    item_update_datetime DATETIME DEFAULT NOW() -- 작성/수정 시간
);
```

Cart / Order



```
CREATE TABLE cart (
    cart_no BIGINT AUTO_INCREMENT NOT NULL PRIMARY KEY, -- 고유키
    cart_code VARCHAR(255) NOT NULL UNIQUE, -- 장바구니 코드(UUID포맷-32자리)
    order_code BIGINT, -- 주문정보 PK
    member_id VARCHAR(255) NULL, -- 아이디(회원)
    session_id VARCHAR(255) NULL, -- 세션아이디(비회원)
    -- 예 )32자리 - 3CB361E0BE1A9A7DE7DB926DF0772BAE
    item_code VARCHAR(255) NOT NULL, -- 상품 코드
    item_name TEXT NOT NULL, -- 상품이름
    item_option_color VARCHAR(255) NULL, -- 색상
    item_option_size VARCHAR(255) NULL, -- 사이즈
    cart_item_amount BIGINT(255) DEFAULT 1, -- 구매갯수
    cart_item_original_price BIGINT(255), -- 할인전가격(ex.49000원)
    cart_discount_price BIGINT(255), -- 할인율이 적용된 차감률 금액(ex.4900원)
    cart_item_price BIGINT(255), -- 상품가격(ex.44100원)
    cart_date DATETIME DEFAULT NOW() -- 장바구니에 담긴 시간/날짜
);
```



```
CREATE TABLE `order` (
    order_no BIGINT AUTO_INCREMENT PRIMARY KEY, -- 고유키
    order_code BIGINT NOT NULL UNIQUE, -- 주문코드
    -- 구매상품 정보
    cart_code_1 VARCHAR(255) NOT NULL UNIQUE, -- 장바구니 코드(UUID포맷-32자리)
    cart_code_2 VARCHAR(255) UNIQUE, -- 장바구니 코드(UUID포맷-32자리)
    cart_code_3 VARCHAR(255) UNIQUE, -- 장바구니 코드(UUID포맷-32자리)
    cart_code_4 VARCHAR(255) UNIQUE, -- 장바구니 코드(UUID포맷-32자리)
    cart_code_5 VARCHAR(255) UNIQUE, -- 장바구니 코드(UUID포맷-32자리)
    order_total_price INT NOT NULL, -- 주문 총금액
    order_total_count TINYINT NOT NULL, -- 주문 상품 개수
    -- 주문자/수령자 정보
    order_name CHAR(255) NOT NULL, -- 주문자 이름
    order_phone CHAR(255) NOT NULL, -- 주문자 연락처
    order_recipient_name CHAR(255) NOT NULL, -- 수령자 이름
    order_recipient_phone CHAR(255) NOT NULL, -- 수령자 연락처
    order_recipient_addr_number CHAR(255) NOT NULL, -- 수령자 우편번호
    order_recipient_addr_1 CHAR(255) NOT NULL, -- 수령자 기본주소
    order_recipient_addr_2 CHAR(255) NOT NULL, -- 수령자 나머지주소
    -- 회원 아이디
    member_id VARCHAR(255) NULL, -- 회원아이디
    -- 적립금 사용
    member_mileage INT DEFAULT 0, -- 사용 마일리지(적립금)
    -- 쿠폰 사용
    member_coupon INT DEFAULT 0, -- 사용 쿠폰 금액
    -- 결제방법
    order_pay_type VARCHAR(255) DEFAULT '휴대폰결제' NOT NULL, -- 휴대폰결제 or 무통장입금 선택
    -- 주문상태
    -- 미입금/주문완료 -> 배송대기 -> 배송중 -> 배송완료, 취소/반품/교환, 구매확정
    order_state VARCHAR(255) NOT NULL DEFAULT '미입금/주문완료', -- 주문상태
    order_datetime DATETIME DEFAULT NOW() -- 결제시간
);
```

QnA / QnA Comment

```
CREATE TABLE `qna`(  
    qna_id BIGINT PRIMARY KEY NOT NULL AUTO_INCREMENT, -- 고유키  
    member_id VARCHAR(255), -- 회원아이디  
    qna_category VARCHAR(255) NOT NULL, -- 카테고리  
    qna_name VARCHAR(50), -- 작성자 닉네임  
    qna_password VARCHAR(255) NOT NULL, -- 비밀번호  
    qna_content TEXT NOT NULL, -- 내용  
    qna_secret VARCHAR(255) NOT NULL, -- 비밀글 여부  
    qna_local_date_time DATETIME DEFAULT NOW() -- 작성시간  
);
```

```
CREATE TABLE qnacomment (  
    comment_id BIGINT PRIMARY KEY NOT NULL AUTO_INCREMENT, -- 고유키  
    comment_writer VARCHAR(255) NOT NULL, -- 답변자(관리자)  
    comment_content VARCHAR(255) NOT NULL, -- 내용  
    comment_qna_id BIGINT NOT NULL, -- qna 외래키  
    comment_date DATETIME DEFAULT NOW() -- 작성시간  
);
```

Review

```
CREATE TABLE review (
    review_no INT AUTO_INCREMENT PRIMARY KEY, -- 고유키
    member_id VARCHAR(255) NOT NULL, -- 아이디(회원) / 비회원은 후기를 달수없음
    item_no VARCHAR(255) NOT NULL, -- 상품 코드(UUID포맷-32자리)
    review_star TINYINT NOT NULL, -- 별점
    review_content TEXT NULL, -- 상품후기
    review_image_url TEXT NULL, -- 이미지
    review_datetime DATETIME DEFAULT NOW(), -- 작성시간
    review_exposure VARCHAR(255) DEFAULT '노출함' -- 노출여부
);
```

Inquiry / Inquiry Reply

```
● ● ●  
  
create table inquiry (  
    inquiry_no BIGINT PRIMARY KEY NOT NULL AUTO_INCREMENT, -- 고유키  
    member_id VARCHAR(255), -- 회원아이디  
    inquiry_nickname VARCHAR(255), -- 작성자 닉네임  
    inquiry_password VARCHAR(255), -- 비밀번호  
    item_no INT NOT NULL, -- item 외래키  
    inquiry_content VARCHAR(255) NOT NULL, -- 내용  
    inquiry_secret VARCHAR(255) NOT NULL, -- 비밀글 여부  
    inquiry_date DATETIME DEFAULT NOW() -- 작성날짜  
);
```

```
● ● ●  
  
create table inquiry_reply (  
    inquiry_reply_no BIGINT PRIMARY KEY NOT NULL AUTO_INCREMENT, -- 고유키  
    inquiry_reply_content TEXT NOT NULL, -- 내용  
    inquiry_reply_date DATETIME DEFAULT NOW(), -- 답변시간  
    inquiry_no BIGINT NOT NULL -- inquiry 외래키  
);
```

Notice

```
CREATE TABLE notice (
    notice_no INT AUTO_INCREMENT NOT NULL PRIMARY KEY, -- 게시글 번호
    notice_type VARCHAR(255) NOT NULL, -- 공지사항 종류(이벤트, 배송지연안내 등
    등) notice_title VARCHAR(255) NOT NULL, -- 공지사항 제목
    notice_content TEXT NULL, -- 공지사항 내용
    notice_datetime DATETIME DEFAULT NOW() -- 작성시간
);
```

프로젝트 구성

```
|  CaptchaUtil.java  
|  
+--comparator  
|    CookiesComparator.java  
|    HighReviewComparator.java  
|    ItemGradeComparator.java  
|    LowPriceComparator.java  
|    SellCountComparator.java  
  
+--config  
|    AwsS3Config.java  
|    SecurityConfig.java  
  
+--controller  
|    +--admin  
|        AdminInquiryController.java  
|        AdminMemberController.java  
|        AdminNoticeController.java  
|        AdminOrderController.java  
|        AdminProductController.java  
|        AdminQnaController.java  
|        AdminReviewController.java  
  
|    \--User  
|        CaptchaController.java  
|        UserInfoController.java  
|        UserInquiryController.java  
|        UserMainController.java  
|        UserNoticeController.java  
|        UserOrderController.java  
|        UserProductController.java  
|        UserQnaController.java  
|        UserReviewController.java  
  
+--dto  
...  
...
```

```
...  
+--dto  
|    +--cart  
|        CartResponseDto.java  
|        CartSaveRequestDto.java  
|  
|    +--inquiry  
|        InquiryResponseDto.java  
|        InquirySaveResponseDto.java  
|        InReplyResponseDto.java  
|        InReplySaveResponseDto.java  
|  
|    +--member  
|        MemberResponseDto.java  
|        MemberSaveRequestDto.java  
|  
|    +--notice  
|        NoticeResponseDto.java  
|        NoticeSaveRequestDto.java  
|        NoticeUpdateRequestDto.java  
|  
|    +--order  
|        OrderContentSaveRequestDto.java  
|        OrderResponseDto.java  
|        OrderSearchDto.java  
|  
|    +--product  
|        FileResponse.java  
|        ProductResponseDto.java  
|        ProductSaveRequestDto.java  
|        ProductSearchDto.java  
|  
|    +--qna  
|        QnaCommentResponseDto.java  
|        QnaCommentSaveDto.java  
|        QnaResponseDto.java  
|        QnaSaveDto.java  
|  
|    +--review  
|        ReviewResponseDto.java  
|        ReviewSaveResponseDto.java  
|  
|    \--security  
|        MemberJoinDto.java  
|        MemberLoginDto.java  
|        SessionUser.java  
  
+--entity  
...  
...
```

```
...  
+--entity  
|    |    AdminReplyEntity.java  
|    |    CartEntity.java  
|    |    InquiryEntity.java  
|    |    InReplyEntity.java  
|    |    MemberEntity.java  
|    |    NoticeEntity.java  
|    |    OrderEntity.java  
|    |    ProductEntity.java  
|    |    QnaCommentEntity.java  
|    |    QnaEntity.java  
|    |    QnaReplyEntity.java  
|    |    ReviewEntity.java  
|    |    SnsUser.java  
|  
|    \--repository  
|        AdminReplyRepository.java  
|        CartRepository.java  
|        InquiryRepository.java  
|        InReplyRepository.java  
|        MemberRepository.java  
|        NoticeRepository.java  
|        OrderRepository.java  
|        ProductRepository.java  
|        QnaCommentRepository.java  
|        QnaReplyRepository.java  
|        QnaRepository.java  
|        ReviewRepository.java  
|        SnsUserRepository.java  
  
+--enumeration  
...
```

```
...  
+--enumeration  
|    UserRole.java  
|  
+--object  
|    FileResponse.java  
|  
\--service  
    AwsS3Service.java  
    CartService.java  
    CustomOAuth2UserService.java  
    InquiryService.java  
    InReplyService.java  
    MemberService.java  
    NoticeService.java  
    OAuthAttributes.java  
    OrderService.java  
    ProductService.java  
    QnaCommentService.java  
    QnaService.java  
    ReviewService.java  
    SecurityService.java  
...
```

프로젝트 구성

```
application-oauth.yml  
application.properties  
application.yml  
db.sql  
  
+-- static  
|   +-- css  
|       admin.css  
|       category.css  
|       coupon.css  
|       footer.css  
|       header.css  
|       home.css  
|       jquery.bxslider.css  
|       jquery.bxslider.min.css  
|       login.css  
|       myorder-list-user.css  
|       myorder-review.css  
|       myorder.css  
|       myProductInquiries.css  
|       navbar.css  
|       notice-content.css  
|       Notice-of-product-information.css  
|       notice.css  
|       option-popup.css  
|       order-complete.css  
|       owl.carousel.css  
|       owl.carousel.min.css  
|       owl.theme.default.css  
|       owl.theme.default.min.css  
|       owl.theme.green.css  
|       pop-page-one-piece.css  
|       pop-page-top1.css  
|       pop-page-type1.css  
|       pop-page-type2.css  
|       product.css  
|       qna-user.css  
|       QnA-write.css  
|       QnA.css  
|       share-btn.css  
|       shopping-basket.css  
|       user-myInfo.css  
|       userjoin.css  
  
\-- js  
...  
...
```

```
\-- js  
    carousel.js  
    ckeditor.js  
    ckeditor.js.map  
    displayTypeChange.js  
    footer.js  
    header.js  
    home-displayTypeChange.js  
    home-sort-btn.js  
    html5shiv.js  
    myorder.js  
    option-popup.js  
    owl.autoplay.js  
    owl.carousel.js  
    owl.carousel.min.js  
    owl.navigation.js  
    popup.js  
    product-top1.js  
    shopping-basket.js  
  
\-- templates  
...
```

```
...  
\\-- templates  
    +-- admin  
        |   +-- inquiry  
        |       content.html  
        |       list.html  
    +-- member  
        |   content.html  
        |   list.html  
    +-- notice  
        |   content.html  
        |   list.html  
        |   write.html  
    +-- order  
        |   content.html  
        |   list.html  
    +-- product  
        |   list.html  
        |   modify.html  
        |   registration.html  
    +-- qna  
        |   content.html  
        |   list.html  
    \\-- review  
        |   content.html  
        |   list.html  
    +-- frgments  
        |   betweenbtn.html  
        |   betweenbtn2.html  
        |   footer.html  
        |   header.html  
        |   nav.html  
    \\-- user  
...
```

```
\-- user  
    +-- category  
        |   content.html  
        |   home.html  
        |   notice-content.html  
        |   notice.html  
        |   qna.html  
        |   search.html  
    +-- category-object  
        |   container-notice.html  
    +-- enlarge  
        |   enlargeProductImg.html  
        |   enlargeProductInfo.html  
    +-- order  
        |   order-complete.html  
        |   shopping-basket.html  
    +-- order-object  
        |   container-order-complete.html  
        |   container-shopping-basket-cart.html  
        |   container-shopping-basket-consent.html  
        |   container-shopping-basket-join.html  
        |   container-shopping-basket-mileage.html  
        |   container-shopping-basket-option.html  
        |   container-shopping-basket-orderer.html  
        |   container-shopping-basket-payment.html  
        |   container-shopping-basket-share.html  
        |   container-shopping-basket-summary.html  
    +-- popup  
        |   Inquiry-modify.html  
        |   Inquiry-write.html  
        |   pop-page1.html  
        |   pop-page2.html  
        |   pop-page3.html  
        |   pop-page4.html  
        |   qna-modify.html  
        |   qna-write.html  
    +-- popup-object  
...
```

```
...  
+-- popup-object  
    |   terms-of-use.html  
+-- product  
    |   content.html  
    |   contentTest.html  
+-- product-object  
    |   container-product-content-header.html  
    |   container-product-content-info.html  
    |   container-product-content-inquiry.html  
    |   container-product-content-nav.html  
    |   container-product-content-review.html  
    |   container-product-content-summary.html  
+-- user  
    |   coupons-mylist.html  
    |   myorder-list-user.html  
    |   myorder-list.html  
    |   myorder.html  
    |   myProductInquiries.html  
    |   qna-user.html  
    |   review-mylist.html  
    |   review-writeForm.html  
    |   user-mileage.html  
    |   user-myInfo-Pswd.html  
    |   user-myInfo.html  
    |   user-sns-join.html  
    |   userjoin.html  
    |   userlogin.html  
    |   userPassTest.html  
    \-- user-object  
        |   container-member-myorder-list1.html  
        |   container-member-myorder-list2.html  
        |   container-member-myorder-list3.html  
        |   container-member-myorder-list4.html  
        |   container-member-myorder-list5.html  
        |   container-member-myorder-list6.html  
        |   container-myorder-coupon-hidden-info.html  
        |   container-myorder-header.html  
        |   container-myorder-nav.html  
        |   container-nonMember-myorder-list1.html  
        |   container-nonMember-myorder-list2.html  
        |   container-nonMember-myorder-list3.html  
        |   container-nonMember-myorder-list4.html  
        |   container-nonMember-myorder-list5.html  
        |   container-nonMember-myorder-list6.html  
...
```

화면 구성



사용자 페이지

메인페이지



- 로그인 여부, 역할에 따라 장바구니, 로그인, 내정보, 관리자 페이지로 이동할 수 있음

<로그아웃> ☰

<로그인(유저)> ☰

<로그인(관리자)> ☰ ☰

```
<span sec:authorize="isAuthenticated()" id="adminLink">  
<span sec:authorize="hasRole('ADMIN')">
```

```
<span sec:authorize="isAnonymous()">
```

- 캐러셀

- 판매량에 따라 베스트 상품 6개 지정

메인페이지

베스트 상품

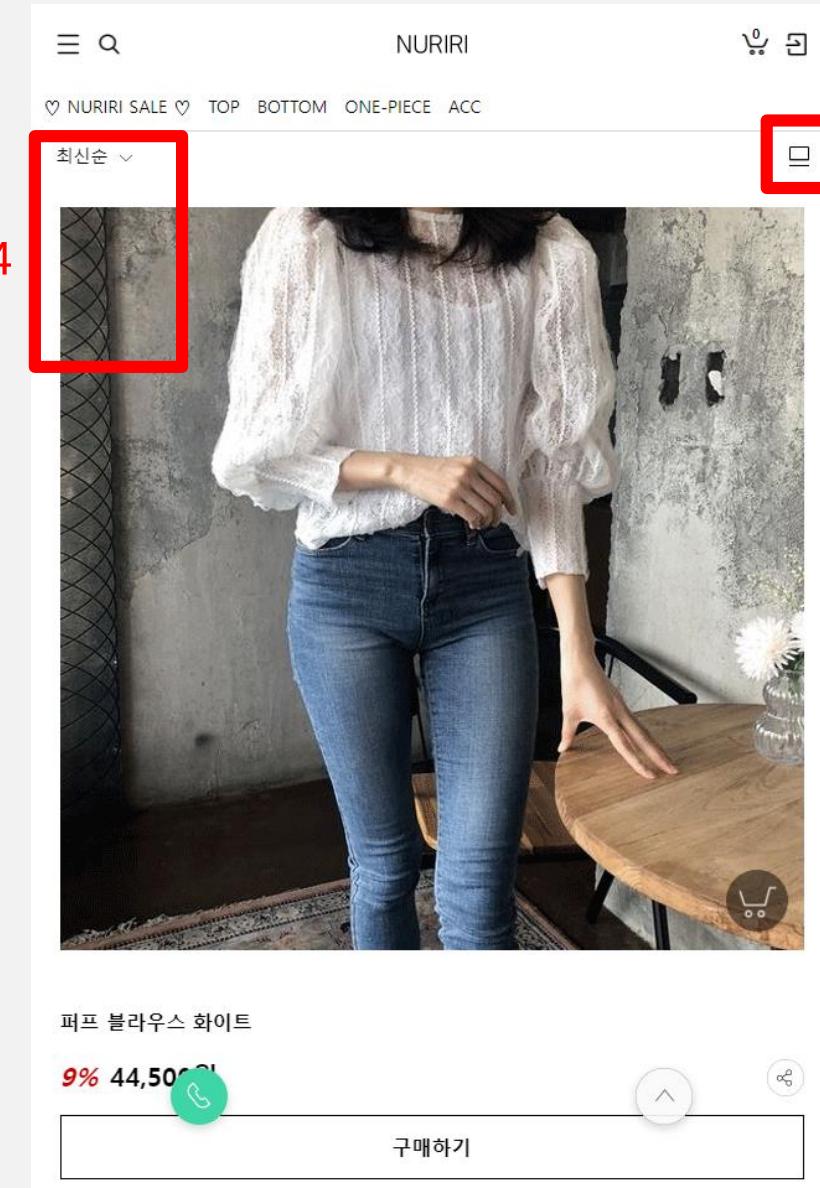


ProductService.java

```
@Transactional(readOnly = true)
public List<ProductResponseDto> findByItem(int num) {
    List<ProductEntity> entityList = new ArrayList();
    if (num == 6) { //BEST 6
        List<String> bestItemList = cartRepository.bestItemFindLimit6();
        for (int i = 0; i < 6; i++) {
            Optional<ProductEntity> entity = productRepository.findBestItem(bestItemList.get(i));
            if (!entity.isPresent()) {
                i--;
            } else {
                entityList.add(entity.get());
            }
        }
    } else { // 9
        entityList = productRepository.findLimit9();
    }

    List<ProductResponseDto> list = entityList.stream().map(ProductResponseDto::new).collect(Collectors.toList());
    return setItemDiscountPrice(list);
}
```

메인페이지



4. 상품 정렬

<최신순>
<판매순>
<낮은가격순>
<리뷰많은순>
<평점높은순>

5. 배치 디자인 바꾸기

메인페이지

상품 정렬

```
function sortSelected(name) {
    document.querySelector("#select").innerHTML =
        name +
        ' ';
    document.querySelector("#sort").style.display = "none";

    if (
        document.querySelector(".ico-list-1").style.display != "none" &&
        name == "최신순"
    ) {
        for (let i = 0; i < 4; i++) {
            document.querySelector("#recentListType" + i).style.display =
                "none";
            document.querySelector("#sellCountListType" + i).style.display =
                "none";
            document.querySelector("#LowPriceListType" + i).style.display =
                "none";
            document.querySelector("#HighReviewListType" + i).style.display =
                "none";
            document.querySelector("#HighGradeListType" + i).style.display =
                "none";
        }
        document.querySelector("#recentListType0").style.display = "block";
    } else if (
        name == "최신순" &&
        document.querySelector(".ico-list-2").style.display != "none"
    ) {
        ... // 생략
    }
    ...
}
```

home-sort-btn.js

배치 디자인 바꾸기

```
function listChangeBtn() {

    let nameText = document.querySelector("#select").innerHTML;
    let nameArr = nameText.split(' ');
    let name = nameArr[0].trim();

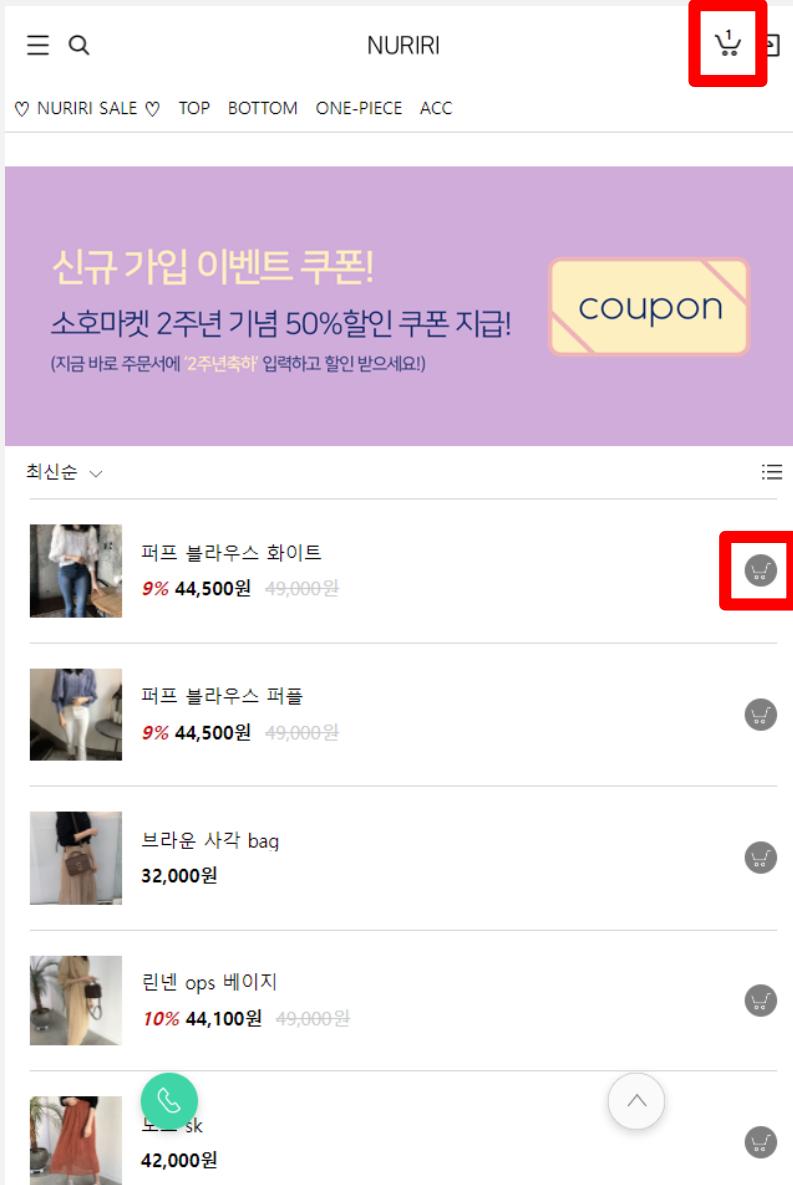
    if( document.querySelector(".ico-list-1").style.display != "none" && name == '최신순' ){

        for (let i =0; i<4; i++){
            document.querySelector("#recentListType"+i).style.display = "none";
            document.querySelector("#sellCountListType"+i).style.display = "none";
            document.querySelector("#LowPriceListType"+i).style.display = "none";
            document.querySelector("#HighReviewListType"+i).style.display = "none";
            document.querySelector("#HighGradeListType"+i).style.display = "none";
            document.querySelector(".ico-list-"+(i+1)).style.display = "none";
        }
        document.querySelector("#recentListType1").style.display = "flex";
        document.querySelector(".ico-list-2").style.display = "flex";
    } else if( name == '최신순' && document.querySelector(".ico-list-2").style.display != "none" ) {

        for (let i =0; i<4; i++){
            ...
        }
        document.querySelector("#recentListType2").style.display = "flex";
        document.querySelector(".ico-list-3").style.display = "flex";
    }
    ...
}
```

home-displayTypeChange.js

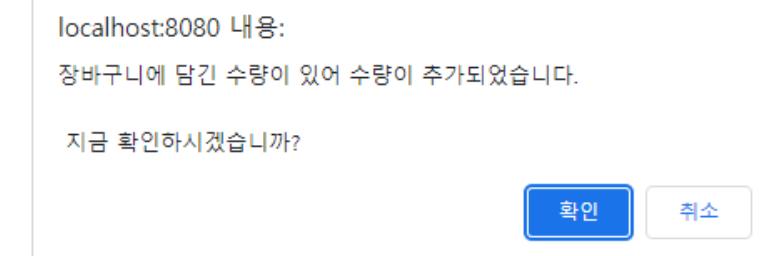
메인페이지



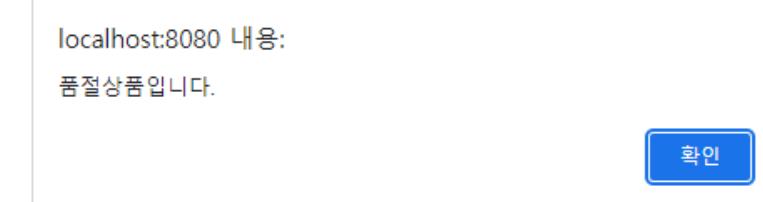
6. 장바구니 기능

장바구니 버튼을 누르면 쿠키에 정보를 저장하여 상품 수량 증가

이미 담긴 상품일 경우 confirm 창



품절 상품일 경우 alert 창



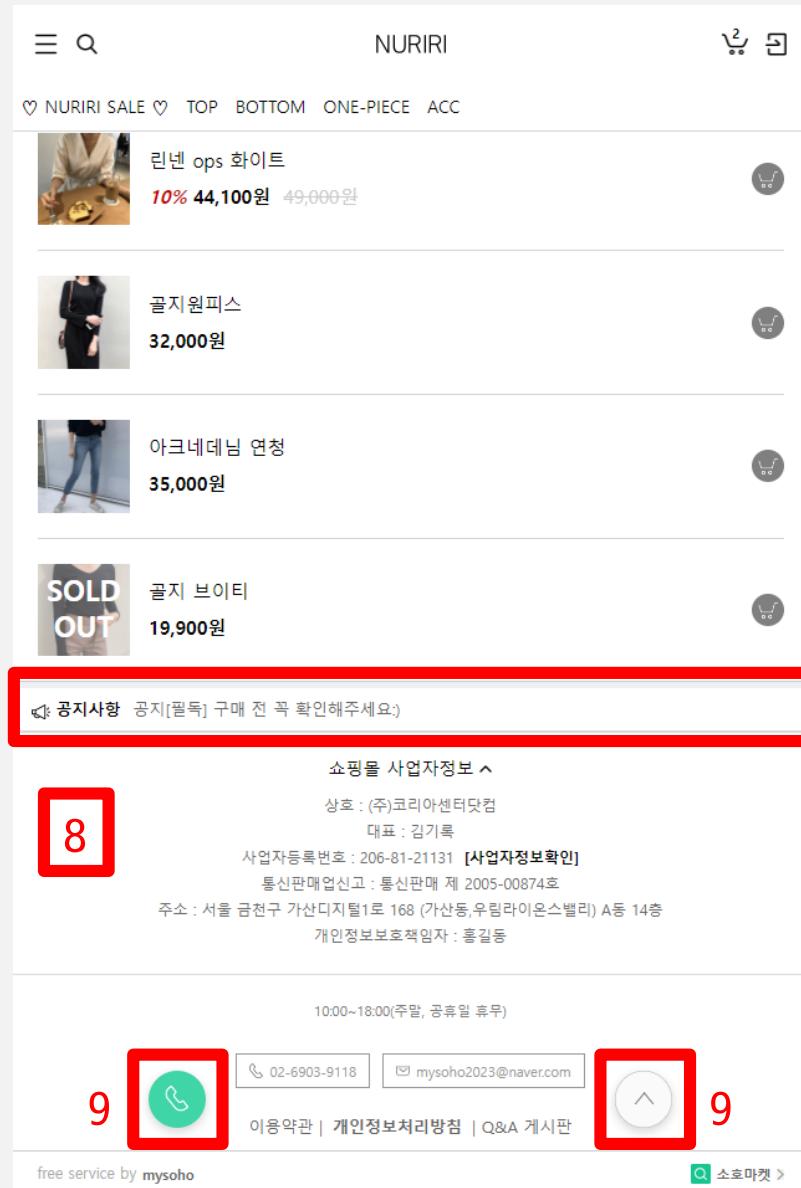
메인페이지

장바구니

```
● ● ●  
header.html  
  
var setCookie = function (cookieName, cookieValue) {  
    //쿠키 생성 함수 (매개변수로 "item_idx 1", 1)이 들어옴  
    var date = new Date(); //  
    date.setTime(date.getTime() + 24 * 60 * 60 * 1000); //date를 현재시간 + 3일 후로 설정한다.  
  
    let encodedCookieName = "";  
    let splitedCookieName = cookieName.split(".");  
  
    let encodedColor = encodeURIComponent(splitedCookieName[2]);  
    for (let idx in splitedCookieName) {  
        if (idx == 0) {  
            encodedCookieName += splitedCookieName[idx];  
        } else if (idx == 2) {  
            encodedCookieName += "." + encodedColor;  
        } else {  
            encodedCookieName += "." + splitedCookieName[idx];  
        }  
    }  
  
    document.cookie =  
        encodedCookieName + //쿠키이름 (key)  
        "=" +  
        cookieValue + // 쿠키(value)  
        ";expires=" +  
        date.toUTCString() + //쿠키 기한 // 설정한 데이터타임으로 기한을 준다.  
        ";path=/";  
};
```

```
● ● ●  
header.html  
  
function basketNumCount(a) {  
    //장바구니 아이콘 선택시 실행되는 함수 / 매개변수로 this가 들어온다.(itemNo)  
    basketNum++;  
    let cookieName = $(a).attr("value"); //attr("value") : value값을 가져온다. 만약 ""안에  
class면 class이름을 가져온다.  
    cookieName = cookieName.replace(/,/g, ".");  
    let cookieValue = 0; //아이템 수량 초기화  
    document.querySelector(".basket-num").innerHTML = basketNum; //오른 숫자 반영  
  
    if (getCookie("item_idx." + cookieName) == null) {  
        //해당 상품이 쿠키로 만들어진적이 없으면 새로 생성하라는 뜻  
  
        setCookie("item_idx." + cookieName, cookieValue + 1);  
    } else {  
        cookieValue = getCookie("item_idx." + cookieName); // getCookie("item_idx:" +  
        item_idx) ⇒ value이다 "2"  
        setCookie("item_idx." + cookieName, Number(cookieValue) + 1); //만약 같은 idx가 있으면  
        수량만 +1해서 기존 쿠키에 덮어씌운다.  
        if (  
            confirm(  
                "장바구니에 담긴 수량이 있어 수량이 추가되었습니다.\n \n 지금 확인하시겠습니까?"  
            ) == true  
        ) {  
            location.href = "/order";  
            //장바구니링크 추가 예정  
        } else {  
        }  
    }  
}
```

메인페이지



7. 최신 공지사항

8. Footer

사업자정보
전화번호, 이메일
이용약관
개인정보처리방침
Q&A

9. 전화걸기 버튼 위로 올리기 버튼

검색

♡ NURIRI SALE ♡ TOP BOTTOM ONE-PIECE ACC

인스타에서 정말 문의 많았던
퍼프 블라우스 :-)

< 베스트 상품

퍼프 블라우스 톤
9% 44,500원

퍼프 블라우스 퍼플
9% 44,500원

브라운 사각 bag
32,000원

검색

< 골지

골지 검색결과 : 2 건

소호마켓에서 검색하기

최신순 ▾

골지원피스
32,000원

골지 브이티
19,900원

쇼핑몰 사업자정보 ▾
10:00~18:00(주말, 공휴일 휴무)
02-6903-9118 | mysoho2023@naver.com

이용약관 | 개인정보처리방침 | Q&A 게시판

free service by mysoho

소호마켓 >

상품 정렬 기능 포함

최신순 ▾

- 최신순
- 판매순
- 낮은가격순
- 리뷰많은순
- 평점높은순

검색 결과가 없을 경우



골지요에 대한 검색결과가 없습니다.
철자나 띄어쓰기를 확인해보세요.

검색

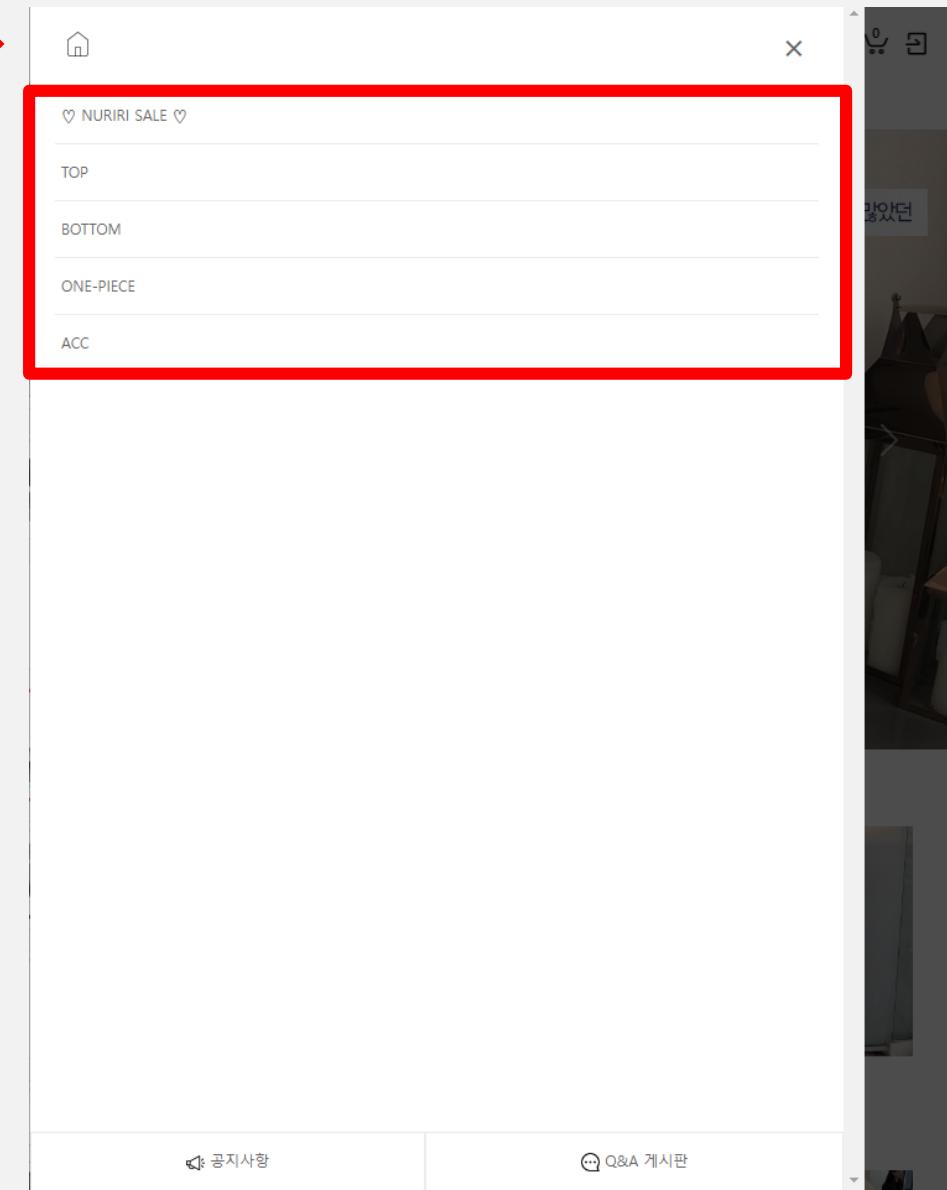
```
● ● ●  
@Transactional(readOnly = true)  
public List<ProductResponseDto> SortItem(List<ProductResponseDto> dtoList, String type) {  
    List<ProductResponseDto> list = new ArrayList<>();  
    for (ProductResponseDto dto : dtoList) {  
        list.add(dto.clone());  
    }  
  
    if (type.equals("판매량")) {  
        for (ProductResponseDto dto : list) {  
            dto.setSalesCount(cartRepository.findByItemSortSale(dto.getItemNo()));  
        }  
        list.sort(new SellCountComparator());  
    } else if (type.equals("낮은가격")) {  
        list.sort(new LowPriceComparator());  
    } else if (type.equals("리뷰")) {  
        for (ProductResponseDto dto : list) {  
            dto.setReviewCount(reviewRepository.findByItemReview(dto.getItemNo()));  
        }  
        list.sort(new HighReviewComparator());  
    } else if (type.equals("평점")) {  
        for (ProductResponseDto dto : list) {  
            Integer reviewStarAVG = reviewRepository.findByItemReviewStarAVG(dto.getItemNo());  
            if (reviewStarAVG == null) {  
                dto.setReviewStar(0);  
            } else {  
                dto.setReviewStar(reviewStarAVG);  
            }  
        }  
        list.sort(new ItemGradeComparator());  
    }  
    return list;  
}
```

ProductService.java

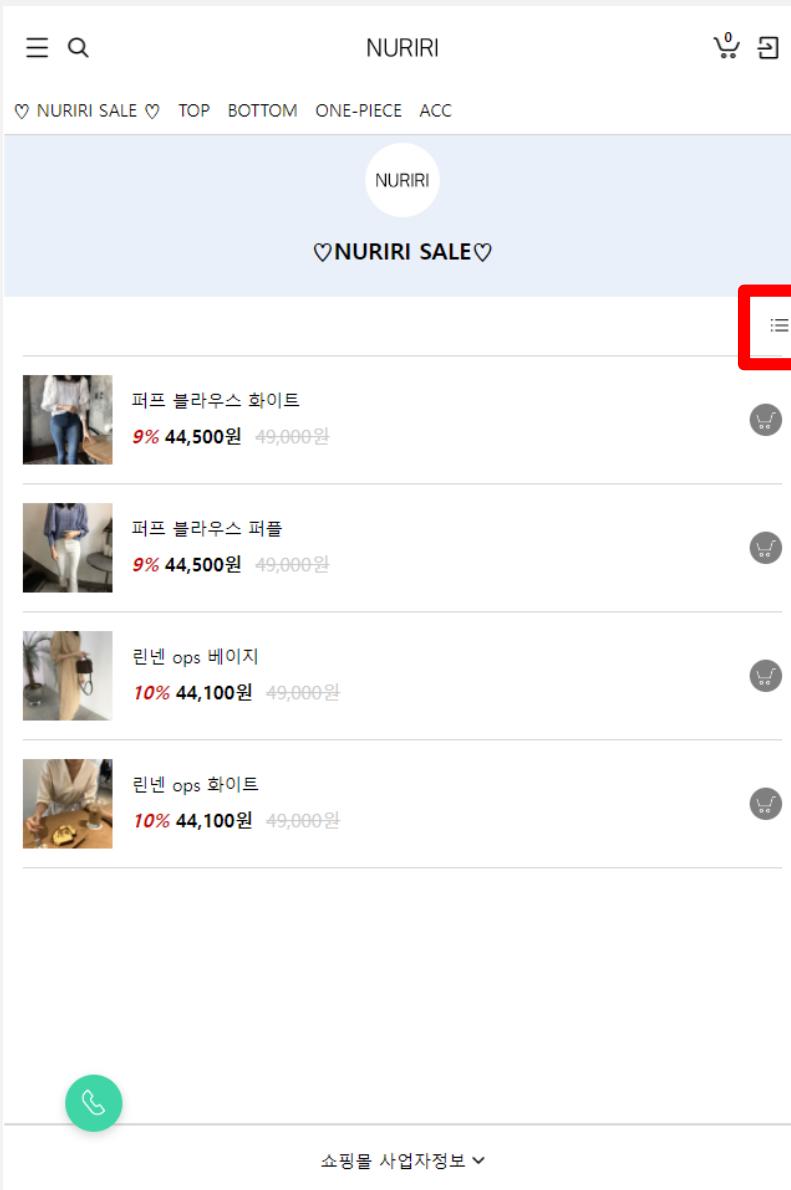
```
● ● ●  
function checkTxt(e) {  
  
    let val = document.querySelector("#keyword").value;  
    // 입력값이 공백일 때  
    if (val === "") {  
        alert("검색어를 입력해주세요.");  
        return false;  
    }  
    return true;  
}  
  
search.html
```

```
● ● ●  
function sortSelectedSearch(name) {  
    document.querySelector("#select").innerHTML =  
        name + " <img src=\"https://image.makeshop.co.kr/mysoho/assets/shop/img/common/arr_bottom.png\" alt=\">\""  
    document.querySelector("#sort").style.display = "none";  
  
    if( name == '최신순' ){  
  
        for (let i =0; i<5; i++){  
            document.querySelector(".listType"+i).style.display = "none";  
        }  
        document.querySelector(".listType0").style.display = "block";  
  
    } else if( name == '판매순' ) {  
  
        for (let i =0; i<5; i++){  
            document.querySelector(".listType"+i).style.display = "none";  
        }  
        document.querySelector(".listType1").style.display = "block";  
    }  
    ... // 생략  
}  
  
search.html
```

카테고리



카테고리



배치 디자인 바꾸기 가능

```
ProductService.java
```

```
@Transactional(readOnly = true)
public List<ProductResponseDto> findByCategory(String keyword) {
    List<ProductEntity> entityList;
    if (keyword.equals("SALE")) {
        entityList = productRepository.findByCategorySale(keyword);
    } else {
        entityList = productRepository.findByCategory(keyword);
    }
    List<ProductResponseDto> list =
        entityList.stream().map(ProductResponseDto::new).collect(Collectors.toList());
    return setItemDiscountPrice(list);
}
```

```
function listChangeBtn()
```

home-displayTypeChange.js

상세페이지



옵션이 한 개인 경우

대표사진/상품명/할인율/가격 확인

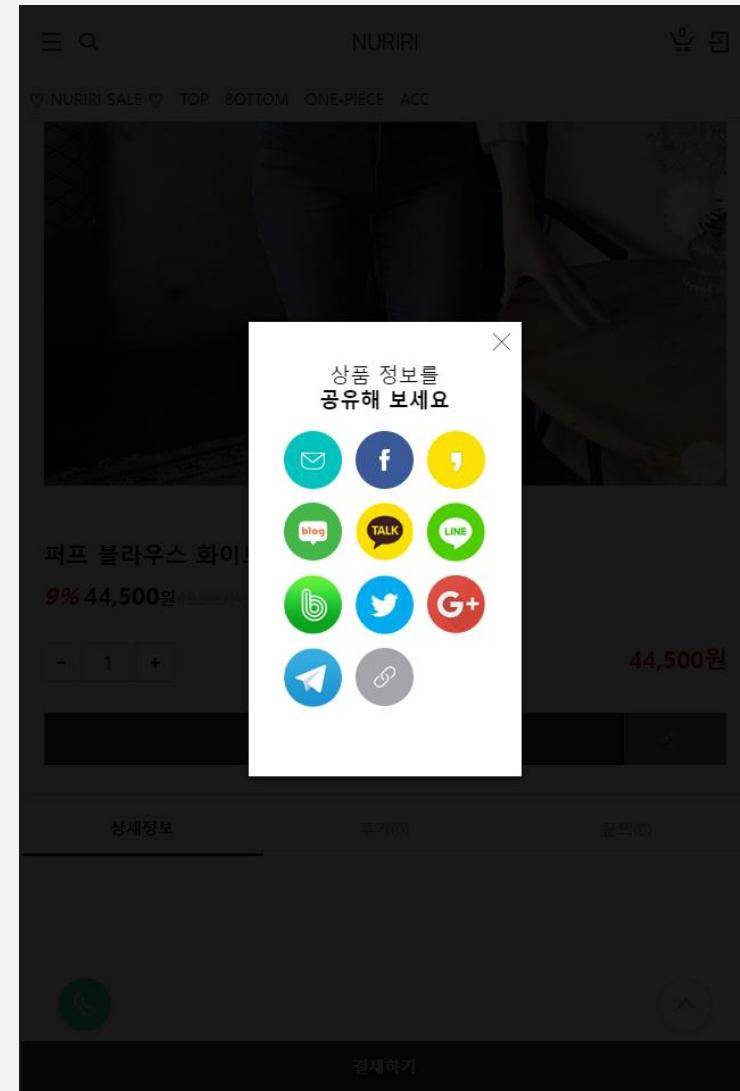
1. 확대 보기 가능
2. 수량 선택
3. 상품 가격 확인
4. 장바구니 담기
공유 링크

상세페이지

확대보기



공유링크

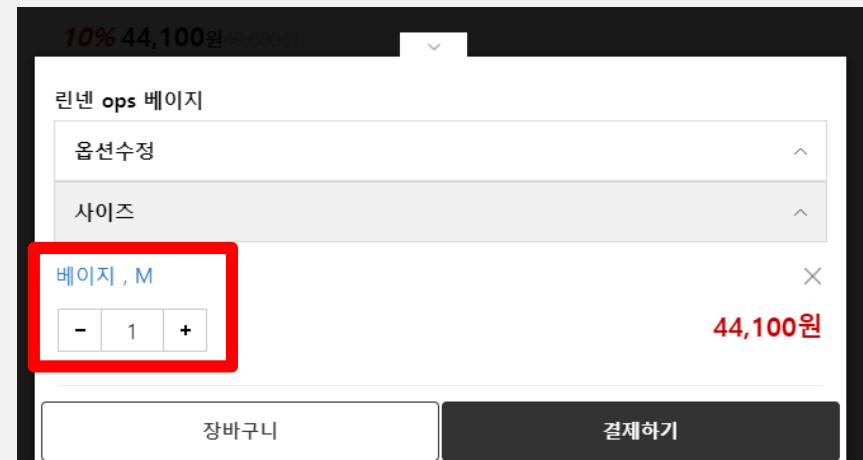


상세페이지



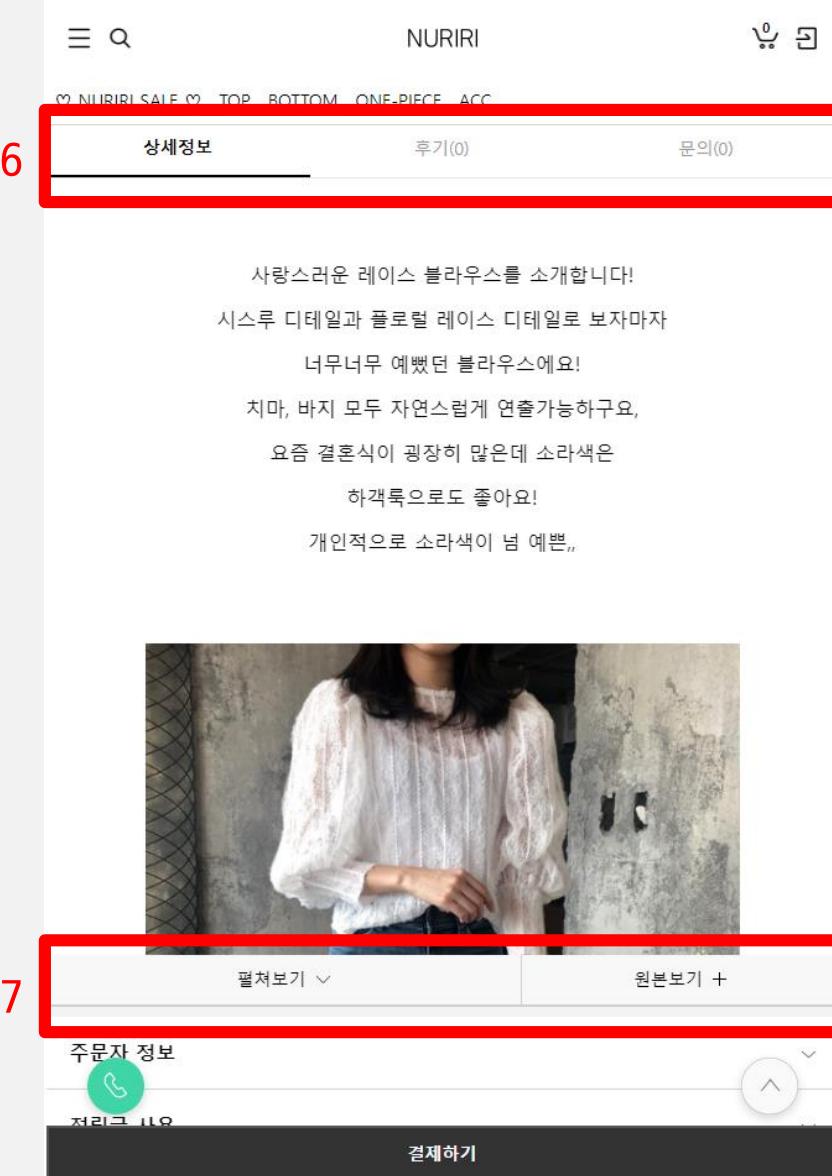
5

5. 옵션이 여러 개인 경우
옵션 수정 버튼이 추가



32

상세페이지



6. 상세정보/후기/문의 탭에서
각 정보 확인 가능

7. 상세정보 내용 확인을 위한
펼쳐보기/접어보기 기능
원본보기 기능

상세페이지

펼쳐보기/접어보기

상세정보

후기(0) 문의(0)

이상이 없을시 교환해드립니다.
+교환 반품이 불가한 경우
-고객의 실수로 주문서 제출이 잘못된 경우
-제품 적용이나 고객의 부주의로 인한 제품의 변형, 퀘온, 오염이나 파손 등으로 인해 상품의 가치가 떨어진 경우
(ex: 텁스틱, 좌장품, 음식 등 이를질, 향수 및 섬유유연제, 담배냄새등의 향이 배인경우)
-인위적인 수선이나 세탁으로 인한 상품 훼손
(불량, 오배송 이더라도 새상품이 아닌 세탁 등 인위적인 수선 기대진 상품은 교환/반품 불가)
+린넨(이) 소재, 체크소재 등 재질 특성상의 경사는 불량으로 간주하지 않아
한물이 불가합니다.
+실밥, 단추위의 초크자국, 옷먼지 등의 사유

위 컬러가 모두 뚜렷하게 구분이 되는지 확인하세요.

제품의 정확한 정보 표현을 위하여
모니터의 밝기 및 명암 감마 조정을 권장합니다.

접어보기 ^ 원본보기 +

주문자 정보

적립금

결제하기

원본보기

상품 상세 정보

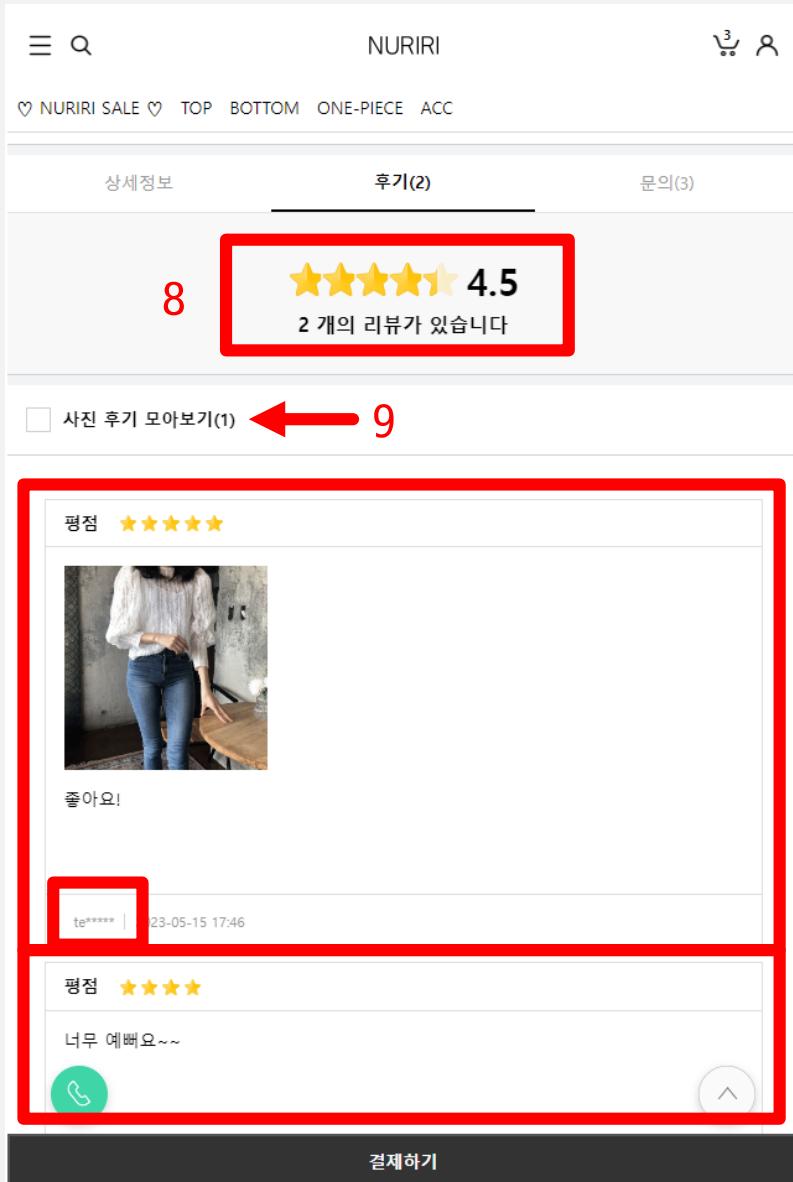
상품 정보 고시

퍼프 블라우스 (바지는 개인 소장입니다)

아이보리,소라 / 44,500원
상세 사이즈 free
[어깨 37 소매 57.5 가슴 47 총기장 57]

사랑스러운 레이스 블라우스를 소개합니다!
시스루 디테일과 플로럴 레이스 디테일로 보자마다
너무너무 예뻤던 블라우스에요!
치마, 바지 모두 자연스럽게 연출 가능하구요,
요즘 결혼식이 굉장히 많은데 소라색은
하객룩으로도 좋아요!
개인적으로 소라색이 넘 예쁜..

상세페이지



8. 리뷰 전체 평점/개수

9. 사진 후기 모아보기 체크박스

10. 사진이 있는 후기

11. 사진이 없는 후기

상세페이지

리뷰 평균 평점



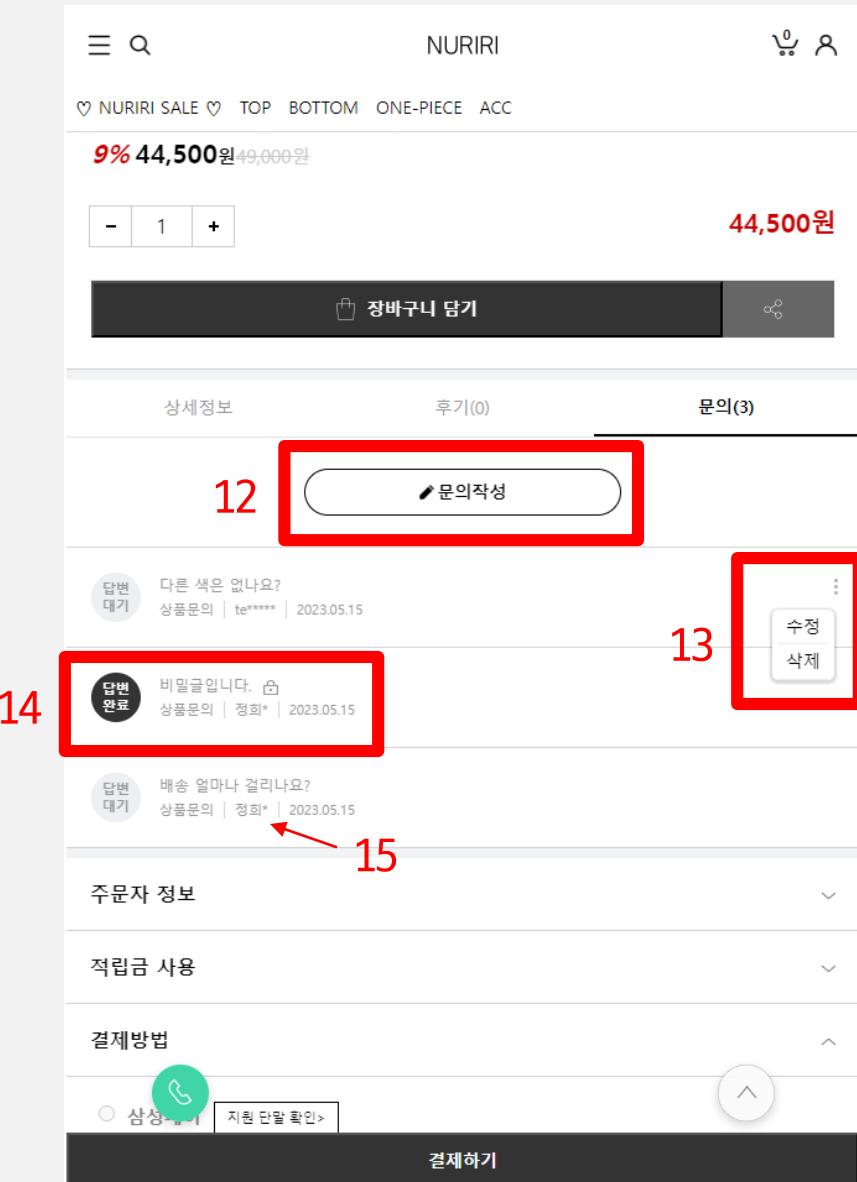
UserOrderController.java

```
public Double avgStar(List<ReviewResponseDto> reviewList) {
    int size = reviewList.size(); // 3. 상품 리뷰 갯수

    byte sum = 0; // 4. 상품별점 평균
    for(int i=0; i<size; i++){
        byte reviewStar = reviewList.get(i).getReviewStar();
        sum += reviewStar;
    }
    double avg1 = sum / Double.valueOf(size);
    double avg2 = Math.round(avg1*10);
    double avgStar = avg2 / 10;
    return avgStar;
}
```

ReviewService.java

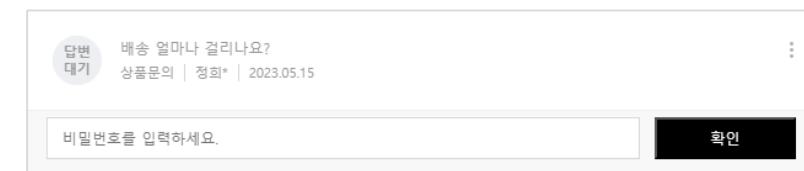
상세페이지



12. 문의 작성 페이지로 이동

13. 수정 페이지로 이동
해당 문의 삭제

비회원일 경우 비밀번호 입력 후
수정/삭제 가능



14. 관리자가 답변할 경우



비밀글일 경우 비밀글입니다.

15. 닉네임/아이디 마스킹

상세페이지

문의 닉네임 마스킹

```
public List<String> inquiryMaskingId(List<InquiryResponseDto> list) {  
    List<String> nameList = new ArrayList<>();  
    for(int i=0 ; i < list.size();i++){  
  
        String qnaName = list.get(i).getMemberId();  
        if(qnaName == null){  
            qnaName = list.get(i).getInquiryNickname();  
        }  
        String qnaHiddenName;  
        if (qnaName.length() == 2){  
            qnaHiddenName = qnaName.replace(qnaName.charAt(1), '*');  
        }else if(qnaName.length() == 1){  
            qnaHiddenName = qnaName;  
        }  
        else{  
            qnaHiddenName = qnaName.substring(0,2);  
            for (int j=0; j<qnaName.length()-2; j++){  
                qnaHiddenName += "*";  
            }  
        }  
        nameList.add(qnaHiddenName);  
    }  
    return nameList;  
}
```

InquiryService.java

상세페이지

The screenshot shows the delivery information section of the NURIRI shopping cart. The page includes a header with a search icon, the brand name 'NURIRI', and a language switch. Below the header, there are tabs for '상세정보' (Delivery Information), '후기(0)' (Reviews), and '문의(3)' (Inquiries). The '상세정보' tab is selected.

주문자 정보 (Delivery Person Information):

- 16.** **주문자정보 입력** (Enter Delivery Person Information)
Name: 이름 (Input field) and 010 (Phone number input field). A note says "- 없이 숫자만 입력" (Only numbers).
- 17.** **수령자정보 입력** (Enter Receiver Information)
Name: 이름 (Input field) and 010 (Phone number input field). A note says "- 없이 숫자만 입력" (Only numbers).
- 18.** **수령자 주소 입력** (Enter Receiver Address)
Address fields: 우편번호 (Postcode), 기본주소 (Basic Address), and 나머지주소 (Remaining Address). A note at the bottom says "배송 전 연락 바랍니다." (Please contact before delivery).

적립금 사용 (Use Points): Includes a green phone icon and a dropdown menu.

결제방법 (Payment Method): Includes a dropdown menu.

결제하기 (Proceed to Payment): A large black button at the bottom.

16. 주문자정보 입력
이름/전화번호

17. 수령자정보 입력
이름/전화번호

18. 수령자 주소 입력
우편번호칸을 누르면 주소
검색 창으로 이동(Daum API)

상세페이지



Daum API를 이용하여 주소 검색 후
Input에 주소 반영

```
function sample6_execDaumPostcode() {
    new daum.Postcode({
        onComplete: function (data) {
            // 방법에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.

            // 각 주소의 노출 규칙에 따라 주소를 조합한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var addr = ""; // 주소 변수
            var extraAddr = "" // 참고항목 변수

            // 사용자가 선택한 주소 타입에 따라 해당 주소 값을 가져온다.
            if (data.userSelectedType === "R") {
                // 사용자가 도로명 주소를 선택했을 경우
                addr = data.roadAddress;
            } else {
                // 사용자가 지번 주소를 선택했을 경우()
                addr = data.jibunAddress;
            }

            // 사용자가 선택한 주소가 도로명 타일일때 참고항목을 조합한다.
            if (data.userSelectedType === "R") {
                // 법정동명이 있을 경우 추가한다. (법정리는 제외)
                // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
                if (data.bname !== "" && /[동|로|가]/g.test(data.bname)) {
                    extraAddr += data.bname;
                }
                // 건물명이 있고, 공동주택일 경우 추가한다.
                if (data.buildingName !== "" && data.apartment === "Y") {
                    extraAddr +=
                        extraAddr !== ""
                            ? ", " + data.buildingName
                            : data.buildingName;
                }
                // 표시할 참고항목이 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
                if (extraAddr !== "") {
                    extraAddr = "(" + extraAddr + ")";
                }
                // 조합된 참고항목을 해당 필드에 넣는다.
                document.getElementById("sample6_extraAddress").value = extraAddr;
            } else {
                document.getElementById("sample6_extraAddress").value = "";
            }

            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById("sample6_postcode").value = data.zonecode;
            document.getElementById("sample6_address").value = addr;
            // 커서를 상세주소 필드로 이동한다.
            document.getElementById("sample6_detailAddress").focus();
        },
    }).open();
}
```

shopping-basket.js

상세페이지

☰ Q NURIRI Ⓜ ⓘ

♡ NURIRI SALE ♡ TOP BOTTOM ONE-PIECE ACC

상세정보 후기(0) 문의(3)

주문자 정보

주문자정보 ^

회원가입

이름

010 - 없이 숫자만 입력

회원가입 정보

아이디(5자~20자 이내)

이메일(비밀번호 찾기 시 이메일로 발송됩니다)
- 비밀번호를 잊어버린 경우 해당 이메일로 비밀번호 관련 정보가 제공 됩니다. 바르게 입력해주세요.

비밀번호 (영문,숫자,특수문자 조합 8자리 이상)

비밀번호 재입력

수령자정보 주문자 정보와 같습니다.

이름

010 - 없이 숫자만 입력

수령자 주 우편번호

결제하기

19

19. 비회원일 경우
회원가입 체크박스가 나타남

체크할 경우
회원가입 정보 입력폼이 나타남

아이디/이메일/
비밀번호/비밀번호 재입력
입력 가능

결제하기 버튼을 누르면
회원가입 동시 진행

상세페이지

The screenshot shows the payment method selection step of the NURIRI website. At the top, there are navigation icons and a search bar. Below that, a breadcrumb path shows 'NURIRI SALE' and categories like 'TOP', 'BOTTOM', 'ONE-PIECE', and 'ACC'. A red box labeled '20' highlights the '적립금 사용' (Point Use) section. It contains a field for '보유적립금 2,000원' (Available Points 2,000 won) with a dropdown menu and a checkbox for '모두 사용' (Use All). Below this, a red box labeled '21' highlights the '결제방법' (Payment Method) section. It includes radio buttons for '삼성페이' (Samsung Pay), '휴대폰결제' (Mobile Payment), and '무통장입금' (Bank Transfer). The '삼성페이' option is selected. At the bottom, a red box labeled '22' highlights the '약관/결제 동의' (Agreement/Payment Consent) section. It lists several checkboxes: '이용약관 동의(필수)', '개인정보 수집 이용 동의 (필수)', '개인정보 수집 이용 동의 (선택)', and '위 구매조건 확인 및 결제진행 동의 (필수)'. There are three '보기' (View) buttons next to the checkboxes. At the very bottom, there are buttons for '총 상품' (Total Product) and '총 배송비' (Total Shipping Cost), both with green phone icons. The total cost is shown as '100원 + 0원'. A large '결제하기' (Pay) button is at the bottom right.

20. 적립금 사용

비회원일 경우

보유적립금 0원

로그인 후, 사용 가능합니다.

21. 결제 방법 선택 휴대폰 결제/무통장입금

무통장입금일 경우

● 무통장입금 (반드시 주문자 성함으로 입금)

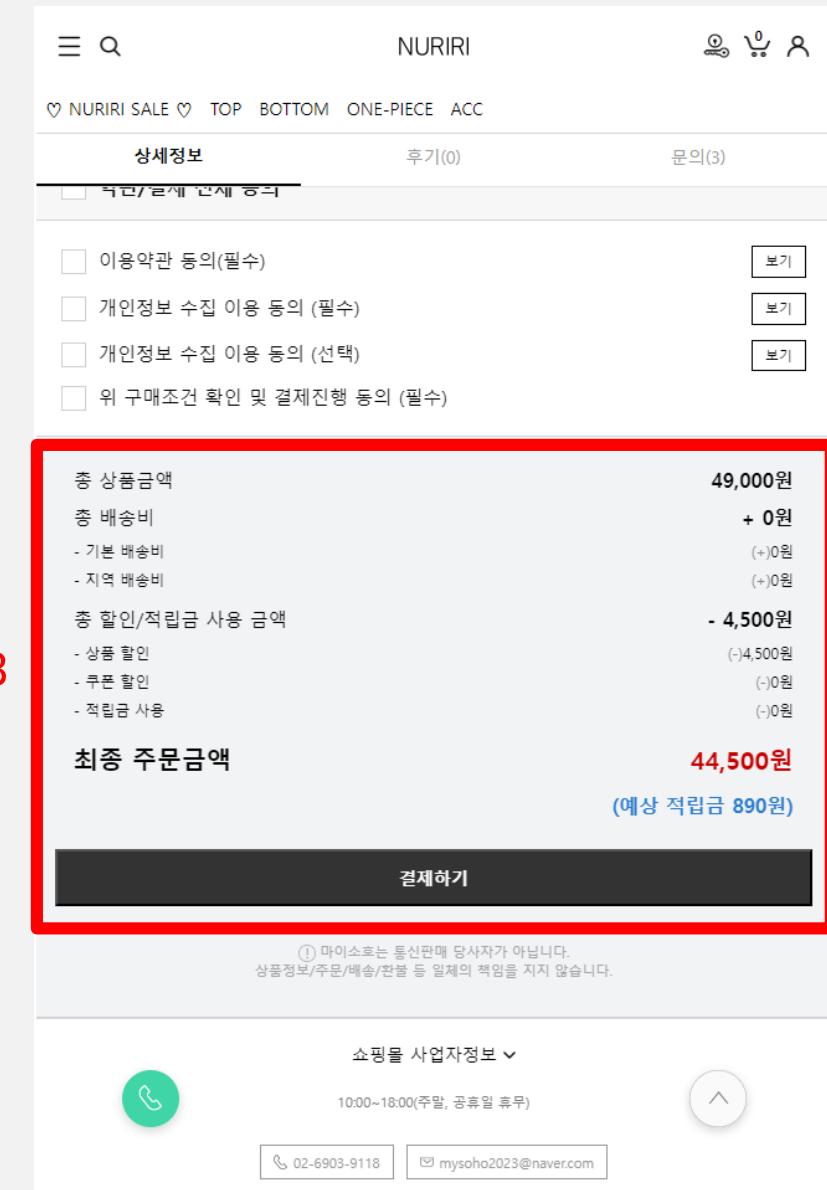
기업은행 0106-859-3030119 (예금주 :소호나라)

입금 계좌번호 선택

기업은행 0106-859-3030119 (예금주 :소호나라)

22. 약관/결제 동의 <보기> 버튼을 누를 시 약관/결제 페이지로 이동

상세페이지



23

23. 상품/할인 금액, 적립금 확인 결제하기 누를 시 유효성 체크

이름: 한글 2~5자

휴대폰번호: 휴대폰번호 형식

아이디: 영문 또는 영문, 숫자
포함 5자~20자 이내

이메일: 이메일 형식

비밀번호: 영문, 숫자, 특수문자
조합 8자리 이상

비밀번호 재입력: 비밀번호와
일치 여부

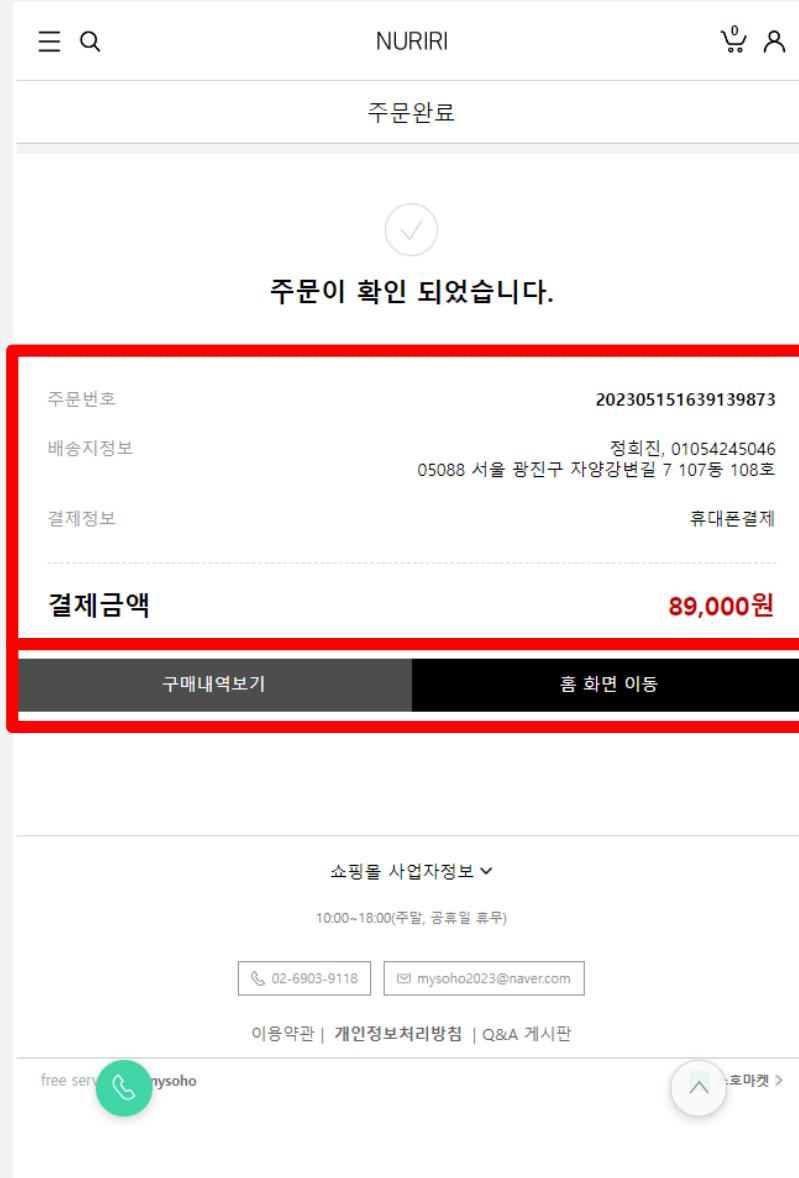
주소: 빈칸이 아닌지 체크 여부

동의: 필수 동의 체크 여부

회원가입을 같이 하는 경우
이미 가입한 회원인지도 확인

43

상세페이지



유효하지 않은 입력을 넣거나
주문 실패했을 시 alert

localhost:8080 내용:

수령자 이름을 입력해주세요.

확인

그외 주문완료 페이지로 이동

1. 주문 정보 확인

2. 구매 내역 보기:
마이페이지로 이동

홈 화면 이동:
메인페이지로 이동

상세페이지

구매 처리

```
    @PostMapping("/order/payAction")
    @ResponseBody
    public String orderPayAction(OrderContentSaveRequestDto orderSaveRequestDto, HttpServletRequest request,
        @AuthenticationPrincipal User user, HttpServletResponse response,
        @RequestParam String[] colorList, @RequestParam String[] sizeList,
        @RequestParam String[] amountList, @RequestParam String[] itemCodeList,
        @Valid MemberJoinDto memberJoinDto, BindingResult bindingResult,
        @RequestParam(value = "mileage", required = false, defaultValue = "0") Long mileage
    ) {

        //////////////////////////////////////////////////////////////////// member DB에 넣기 (회원가입) ///////////////////////////////////////////////////////////////////
        ...
        // 회원가입 페이지와 동일

        //////////////////////////////////////////////////////////////////// cart DB에 넣기 ///////////////////////////////////////////////////////////////////
        String[] cartCodeList = {null, null, null, null, null};
        Long orderCode = orderService.generateOrderNo();

        for (int i=0; i<itemCodeList.length; i++) {

            Long cartItemAmount = Long.parseLong(amountList[i]); // cartItemAmount
            String itemOptionColor = colorList[i]; // itemOptionColor
            String itemOptionSize = sizeList[i]; // itemOptionSize
            String itemCode = itemCodeList[i]; // itemCode
            String cartCode = UUID.randomUUID().toString(); // cartCode
            cartCodeList[i] = cartCode;
            CartSaveRequestDto cartSaveRequestDto = cartService.saveCartDto(cartCode, orderCode,
                itemCode, cartItemAmount, itemOptionColor, itemOptionSize, memberId);
        }
    }
```

```
    // DB에 넣기
    boolean success = cartService.saveCart(cartSaveRequestDto);
    if (success) {
        try {
            // 기존에 있던 쿠키 삭제하기
            String encodedColor = URLEncoder.encode(itemOptionColor, "UTF-8");
            Cookie cookie =
                new Cookie("item_idx." + itemCode + "." + encodedColor + "." + itemOptionSize, String.valueOf(cartItemAmount));
            cookie.setPath("/");
            cookie.setMaxAge(0);
            response.addCookie(cookie);
        }catch (Exception e){
            e.printStackTrace();
            return "<script>alert('결제 중 오류가 발생했습니다.\\n다시 결제해주세요.');//location.href='/order';</script>";
        }
    } else {
        return "<script>alert('결제 중 오류가 발생했습니다.\\n다시 결제해주세요.');//location.href='/order';</script>";
    }
}

//////////////////////////////////////////////////////////////////// order DB에 넣기 ///////////////////////////////////////////////////////////////////

// orderDTO 설정하기
OrderContentSaveRequestDto orderDto =
    orderService.saveOrderDtoComplete(orderSaveRequestDto, orderCode, cartCodeList, memberId);
boolean success = orderService.saveOrderDto(orderDto);
if (!success) {
    return "<script>alert('결제 중 오류가 발생했습니다.\\n다시 결제해주세요.');//location.href='/order';</script>";
}

return "<script>location.href='/order/complete?orderCode=" + orderCode + "'</script>";
}
```

UserOrderController.java

상세페이지

유효성 검사

```
<script>
function checkValid(){

    let checkId = /^[a-zA-Z0-9]{4,12}$/; //아이디 유효성 검사
    let checkPW = /(?=.*[0-9])(?=.*[a-z])(?=.*\W)(?=\\S+$).{8,20}/; //비밀번호 유효성 검사
    let checkName = /^[가-힣]{2,5}$/; //이름 유효성 검사
    let checkEmail = /^[A-Za-z0-9_.-]+@[A-Za-z0-9_.-]+\.[A-Za-z0-9_.-]*$/; //이메일 유효성 검사
    let checkPhone = /^[0-9]{10,11}$/; //이메일 유효성 검사

    if(document.querySelector("#final-size-result").value === '' || document.querySelector("#final-color-result").value === ''){
        return "옵션을 선택해 주세요."
    }

    // 주문자 정보 이름
    let orderName = document.getElementById("order-first-input").value;
    if (orderName == "" || !checkName.test(orderName)){
        return "주문자 이름을 2~5자의 한글로 입력해주세요."
    }

    // 주문자 정보 번호
    let orderPhoneBack = document.getElementById("order-sec-input").value;
    let orderPhone = document.getElementById("orderPhone").value;
    if (orderPhoneBack == "" || !checkPhone.test(orderPhone)){
        return "주문자 휴대폰 번호를 정확히 입력해주세요."
    }

    ... // 생략

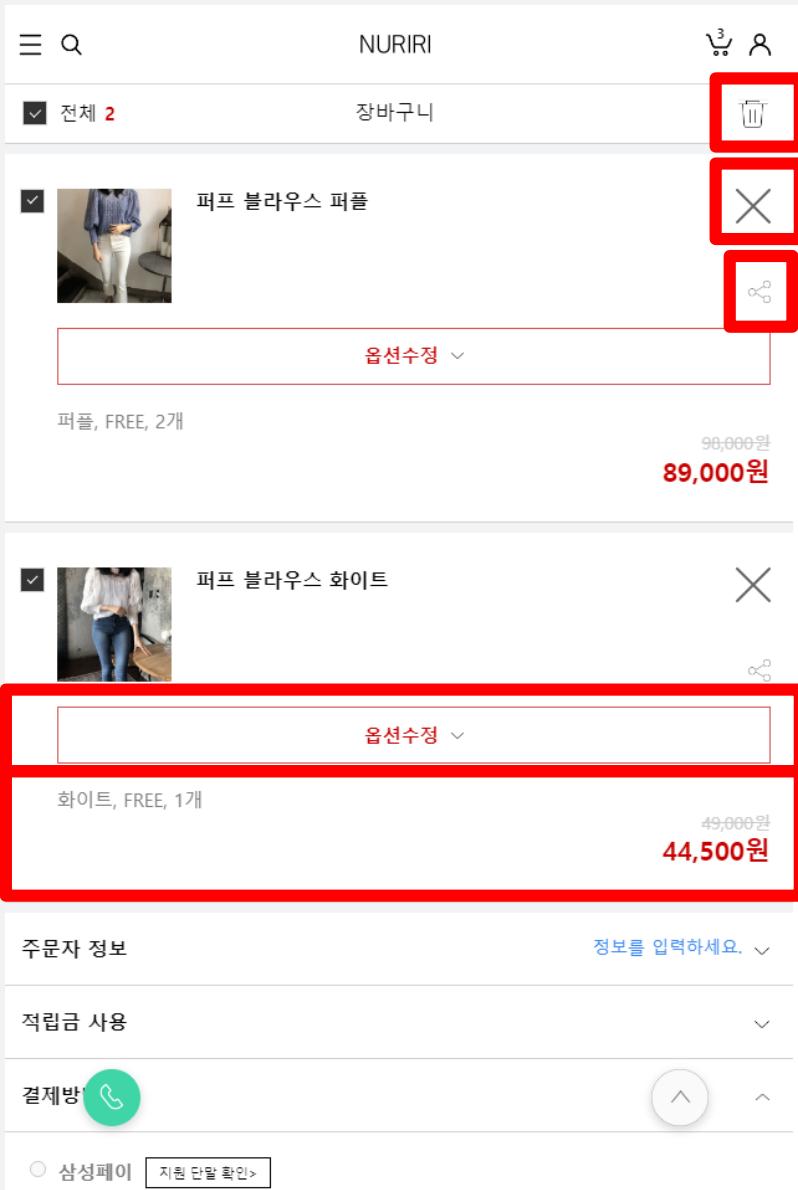
    // 회원가입 할 경우
    if ([[${member}]] == null && document.getElementById("joinCheckBox").checked){
        // 아이디
        let username = document.getElementById("username").value;
        if (username == "" || !checkId.test(username)){
            return "아이디를 정확히 입력해주세요."
        }

        ... // 생략
    }

    return "success";
}
</script>
```

user/product/content.html

장바구니



쿠키에 저장된 상품 정보 받아오기

- 체크 표시된 상품
장바구니에서 삭제 (선택 삭제)
- 해당 상품 장바구니에서 삭제
- 공유 링크
- 옵션 수정



5. 상품 정보

장바구니

주문자 정보 정보를 입력하세요. ▾

적립금 사용 ▼

결제방법 휴대폰결제 ▾

약관/결제 전체 동의

이용약관 동의(필수) 보기

개인정보 수집 이용 동의 (필수) 보기

개인정보 수집 이용 동의 (선택) 보기

위 구매조건 확인 및 결제진행 동의 (필수)

총 상품금액	147,000원
총 배송비	+ 0원
- 기본 배송비	(+) 0원
- 지역 배송비	(+) 0원
총 할인/적립금 사용 금액	- 13,500원
- 상품 할인	(-) 13,500원
- 쿠폰 할인	(-) 0원
- 적립금 사용	(-) 0원
최종 주문금액	133,500원
(예상 적립금 4,005원)	

결제하기

① 마이소호는 통신판매 당사자가 아닙니다.
상품정보/주문/배송/판별 등 일체의 책임을 지지 않습니다.

◀ ◁

쇼핑몰 사연자정보 ▾

상품 상세 페이지의
구매 입력 폼과 동일

선택된 상품의 상품 금액 계산

결제하기 버튼을 누르면 유효성 검사
진행 후 결제 완료 페이지로 이동

장바구니

장바구니 정보 가져오기



UserOrderController.java

```
Cookie[] cookies = request.getCookies(); // 모든 쿠키 가져오기
Arrays.sort(cookies, new CookiesComparator()); // 쿠키 정렬하기

List<ProductResponseDto> itemList = new ArrayList();
List<CartResponseDto> cartList = new ArrayList();
if(cookies!=null){
    itemList = productService.itemListByCookies(cookies);
    cartList = productService.cartListByCookies(cookies);
}
```



CookiesComparator.java

```
public class CookiesComparator implements Comparator<Cookie> {
    @Override
    public int compare(Cookie c1, Cookie c2) {
        String c1Name = c1.getName();
        String c2Name = c2.getName();

        return c2Name.compareTo(c1Name);
    }
}
```

장바구니

장바구니 정보 가져오기

ProductService.java

```
public List<CartResponseDto> cartListByCookies(Cookie[] cookies) {
    List<CartResponseDto> cartList = new ArrayList();
    for (Cookie c : cookies) {
        String name = c.getName(); // 쿠키 이름 가져오기
        String value = c.getValue(); // 쿠키 값 가져오기

        if (name.startsWith("item_idx.")) {

            // 변수 선언
            Long itemNo = Long.parseLong(name.split("\\.")[1]);
            Long cartItemAmount = Long.parseLong(value);
            String itemOptionColor = "";
            try {
                itemOptionColor = URLDecoder.decode(name.split("\\.")[2], "UTF-8");
            } catch (Exception e) {
                e.printStackTrace();
            }
            String itemOptionSize = name.split("\\.")[3];

            // itemList
            ProductResponseDto productResponseDto = findById(itemNo);

            // cartList
            String cartCode = UUID.randomUUID().toString();
            Long cartDiscountPrice = productResponseDto.getItemPrice() * productResponseDto.getItemDiscountRate() / 100;
            Long cartItemPrice = (productResponseDto.getItemPrice() - cartDiscountPrice) / 100 * 100;
            cartDiscountPrice = productResponseDto.getItemPrice() - cartItemPrice;

            CartResponseDto cartResponseDto = CartResponseDto.builder()
                .cartCode(cartCode)
                .itemCode(String.valueOf(itemNo))
                .itemName(productResponseDto.getItemName())
                .itemOptionColor(itemOptionColor)
                .itemOptionSize(itemOptionSize)
                .cartItemAmount(cartItemAmount)
                .cartItemOriginalPrice(productResponseDto.getItemPrice())
                .cartDiscountPrice(cartDiscountPrice)
                .cartItemPrice(cartItemPrice)
                .build();
            cartList.add(cartResponseDto);
        }
    }
    return cartList;
}
```

ProductService.java

```
public List<ProductResponseDto> itemListByCookies(Cookie[] cookies) {
    List<ProductResponseDto> itemList = new ArrayList();
    for (Cookie c : cookies) {
        String name = c.getName(); // 쿠키 이름 가져오기

        if (name.startsWith("item_idx.")) {
            // 변수 선언
            Long itemNo = Long.parseLong(name.split("\\.")[1]);
            // itemList
            ProductResponseDto productResponseDto = findById(itemNo);
            itemList.add(productResponseDto);
        }
    }
    return itemList;
}
```

장바구니

장바구니 상품 수정/삭제

```
UserOrderController.java

@PostMapping("/order/deleteAction")
@ResponseBody
public String orderDeleteAction(@RequestParam String itemOptionColor, @RequestParam String itemOptionSize,
                                @RequestParam String itemNo, HttpServletRequest request, HttpServletResponse response) {

    Cookie[] cookies = request.getCookies();
    String encodedItemOptionColor = itemOptionColor;
    int cartNum = 0;

    try {
        encodedItemOptionColor = URLEncoder.encode(itemOptionColor, "UTF-8");
    } catch (Exception e) {
        e.printStackTrace();
        return "<script>alert('삭제 중 오류가 발생했습니다.\\n다시 입력해주세요.');//location.href='/order';</script>";
    }

    if(cookies!=null){
        for(Cookie c : cookies){
            String name = c.getName();
            String value = c.getValue();
            if (name.startsWith("item_idx.")){
                cartNum++;
            }
            if (name.equals("item_idx."+ itemNo + "." + encodedItemOptionColor + "." + itemOptionSize)) {
                Cookie cookie = new Cookie(name, value);
                cookie.setPath("/");
                cookie.setMaxAge(0);
                response.addCookie(cookie);
            }
        }
    }

    // 장바구니 비었을 때
    if (cartNum == 1){
        return "<script>alert('장바구니가 비었습니다.\\n관심 있는 상품을 담아보세요.');//location.href='//';</script>";
    }

    return "<script>location.href='/order';</script>";
}
```

```
UserOrderController.java

@PostMapping("/order/modifyAction")
@ResponseBody
public String orderModifyAction(@RequestParam String changedSize, @RequestParam String changedColor,
                                @RequestParam String changedAmount, @RequestParam String originalColor,
                                @RequestParam String originalSize, HttpServletResponse response,
                                @RequestParam String itemNo, HttpServletRequest request) {

    // 기존에 있던 쿠키 삭제
    Cookie[] cookies = request.getCookies();
    String encodedOriginalColor = originalColor;

    try {
        encodedOriginalColor = URLEncoder.encode(originalColor, "UTF-8");
    } catch (Exception e) {
        e.printStackTrace();
        return "<script>alert('수정 중에 오류가 발생했습니다.\\n다시 수정해주세요.');//location.href='/order';</script>";
    }

    if(cookies!=null){
        for(Cookie c : cookies){
            String name = c.getName();
            String value = c.getValue();

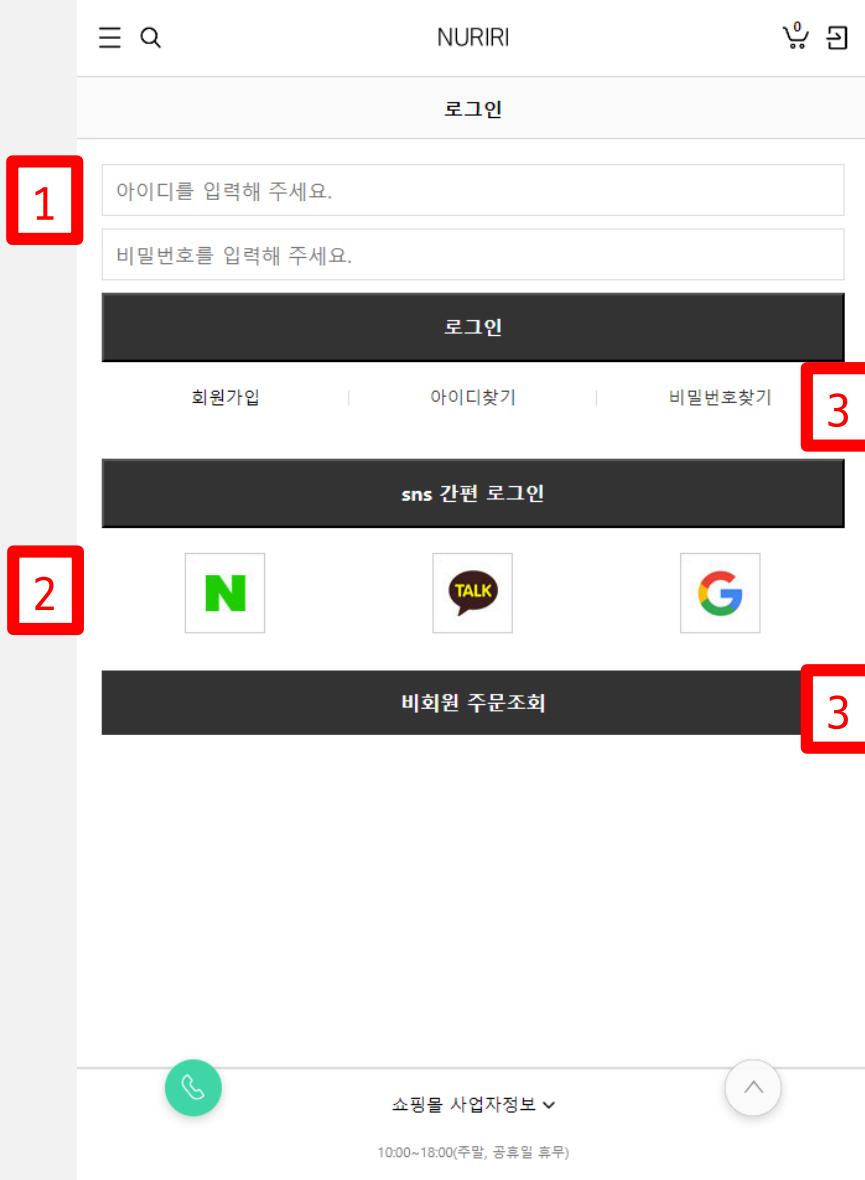
            if (name.equals("item_idx."+ itemNo + "." + encodedOriginalColor + "." + originalSize)) {
                Cookie cookie = new Cookie(name, value);
                cookie.setPath("/");
                cookie.setMaxAge(0);
                response.addCookie(cookie);
            }
        }
    }

    // 쿠키 생성
    Cookie cookie;
    String encodedChangedColor;

    try {
        encodedChangedColor = URLEncoder.encode(changedColor, "UTF-8");
        cookie = new Cookie("item_idx."+ itemNo + "." + encodedChangedColor + "." + changedSize, changedAmount);
        cookie.setPath("/");
        cookie.setMaxAge(60*60*24*3);
        response.addCookie(cookie);
    } catch (Exception e){
        e.printStackTrace();
        return "<script>alert('수정 중에 오류가 발생했습니다.\\n다시 수정해주세요.');//location.href='/order';</script>";
    }

    return "<script>location.href='/order';</script>";
}
```

로그인



1. 로그인

Spring security 적용

비밀번호는 encoding하여
DB에 들어감

아이디/비밀번호가 일치하지
않을 경우 alert창

localhost:8080 내용:
아이디/비밀번호를 확인해주세요.

확인

2. SNS 로그인

3. 회원가입
아이디찾기/비밀번호찾기
비회원 주문조회

로그인



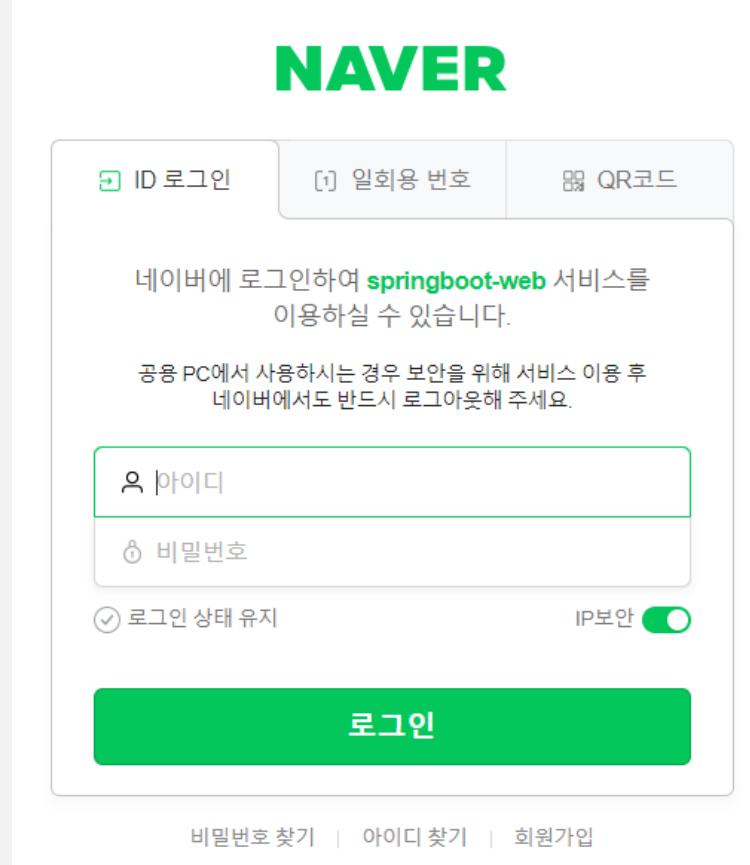
```
@Configuration  
@RequiredArgsConstructor  
@EnableWebSecurity //웹보안 활성화를 위한 annotation  
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
  
    final private SecurityService securityService;  
    final private CustomOAuth2UserService customOAuth2UserService;  
    @Override  
    public void configure(WebSecurity web) {  
        web  
            .ignoring()  
            .antMatchers("/find/**");  
    }  
    @Override  
    protected void configure(HttpSecurity http) throws Exception {  
        http  
            .authorizeRequests() // 요청에 대한 보안설정을 시작  
            .antMatchers("/", "/order/**", "/plan/**", "/product/**", "/qna/**", "/notice/**", "/inquiry/**", "/myorder",  
                "/search", "/terms/**", "/enlarge/**", "/css/**", "/js/**", "/img/**", "/captchaImg.do").permitAll()  
            .antMatchers("/user/join").permitAll()  
            .antMatchers("/user/joinAction").permitAll()  
            .antMatchers("/myorder/**").hasAnyRole("USER", "ADMIN")  
            .antMatchers("/admin/**").hasRole("ADMIN")  
            .anyRequest().authenticated()  
        .and()  
            .formLogin() //로그인 인증에 대한 설정을 시작  
            .loginPage("/user/login") //  
            .loginProcessingUrl("/user/loginAction") //로그인 액션 URI를 지정한다.  
            .successHandler( (request,response,authentication) -> {  
                request.getSession().setAttribute("username", request.getParameter("name"));  
                response.sendRedirect("/");  
            })  
            .failureUrl("/user/login?error")  
            .permitAll()  
        .and()  
            .logout()  
            .logoutRequestMatcher(new AntPathRequestMatcher("/user/logoutAction"))  
            .deleteCookies("JSESSIONID", "XSRF-TOKEN")  
            .invalidateHttpSession(true)  
            .logoutSuccessUrl("/")  
        .and()  
            .oauth2Login()  
            .successHandler(successHandler())  
            .failureHandler(failureHandler())  
            .userInfoEndpoint().userService(customOAuth2UserService);  
    }  
}
```

SecurityConfig.java

```
@Bean  
public PasswordEncoder passwordEncoder() {  
    return new BCryptPasswordEncoder();  
}  
  
@Override  
protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
    auth.userDetailsService(securityService).passwordEncoder(passwordEncoder());  
}  
@Bean  
SimpleUrlAuthenticationSuccessHandler successHandler() { //로그인 성공시 부가 작업  
    return new SimpleUrlAuthenticationSuccessHandler("/snsLoginSuccess");  
}  
@Bean  
SimpleUrlAuthenticationFailureHandler failureHandler() {  
    return new SimpleUrlAuthenticationFailureHandler("/snsLoginFailure");  
}
```

SecurityConfig.java

로그인



NAVER

ID 로그인 일회용 번호 QR코드

네이버에 로그인하여 **springboot-web** 서비스를 이용하실 수 있습니다.

공용 PC에서 사용하시는 경우 보안을 위해 서비스 이용 후 네이버에서도 반드시 로그아웃해 주세요.

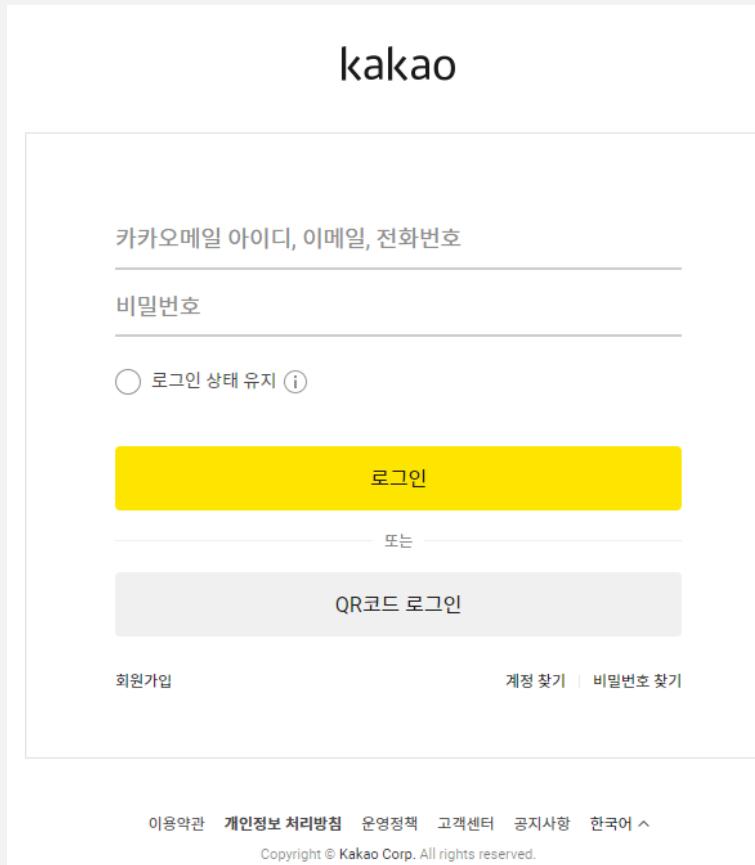
아이디

비밀번호

로그인 상태 유지 IP보안

로그인

[비밀번호 찾기](#) | [아이디 찾기](#) | [회원가입](#)



kakao

카카오메일 아이디, 이메일, 전화번호

비밀번호

로그인 상태 유지 [?](#)

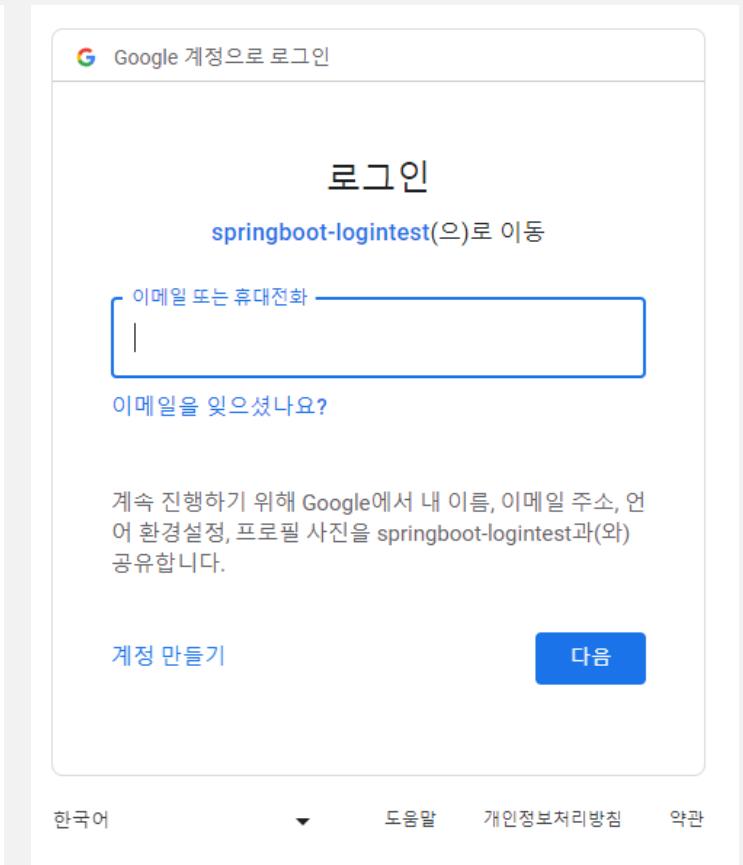
로그인

또는

QR코드 로그인

회원가입 [계정 찾기](#) | [비밀번호 찾기](#)

이용약관 [개인정보 처리방침](#) 운영정책 고객센터 공지사항 한국어 ^
Copyright © Kakao Corp. All rights reserved.



Google 계정으로 로그인

로그인

[springboot-logintest\(으\)로 이동](#)

이메일 또는 휴대전화

이메일을 잊으셨나요?

계속 진행하기 위해 Google에서 내 이름, 이메일 주소, 언어 환경설정, 프로필 사진을 **springboot-logintest**과(와) 공유합니다.

[계정 만들기](#) **다음**

한국어 ▾ 도움말 개인정보처리방침 약관

OAuth2로 SNS 로그인 구현

로그인

```
CustomOAuth2UserService.java
```

```
@RequiredArgsConstructor
@Service
public class CustomOAuth2UserService implements OAuth2UserService<OAuth2UserRequest, OAuth2User> {
    private final SnsUserRepository userRepository;
    private final HttpSession httpSession;

    @Override
    public OAuth2User loadUser(OAuth2UserRequest userRequest) throws OAuth2AuthenticationException {
        OAuth2UserService delegate = new DefaultOAuth2UserService();
        OAuth2User oAuth2User = delegate.loadUser(userRequest);

        String registrationId = userRequest.getClientRegistration().getRegistrationId();
        String userNameAttributeName = userRequest.getClientRegistration().getProviderDetails()
            .getUserInfoEndpoint().getUserNameAttributeName();

        OAuthAttributes attributes = OAuthAttributes.of(registrationId, userNameAttributeName, oAuth2User.getAttributes());
        SnsUser user = saveOrUpdate(attributes);
        httpSession.setAttribute("user", new SessionUser(user));

        return new DefaultOAuth2User(
            Collections.singleton(new SimpleGrantedAuthority(user.getRoleKey())),
            attributes.getAttributes(),
            attributes.getNameAttributeKey());
    }

    private SnsUser saveOrUpdate(OAuthAttributes attributes) {
        SnsUser user = userRepository.findByEmail(attributes.getEmail())
            .map(entity → entity.update(attributes.getName(), attributes.getPicture()))
            .orElse(attributes.toEntity());

        return userRepository.save(user);
    }
}
```

```
application-oauth.java
```

```
spring:
  security:
    oauth2:
      client:
        registration:
          google:
            client-id: // 입력
            client-secret: // 입력
            scope: profile,email
          kakao:
            client-id: // 입력
            client-name: Kakao
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
# http://localhost:8080/ login/ oauth2/code/ kakao
            scope: profile_nickname, profile_image, account_email
            authorization-grant-type: authorization_code
            client-authentication-method: POST
          naver:
            client-id: // 입력
            client-secret: // 입력
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            authorization-grant-type: authorization_code
            scope: name,email,profile_image
            client-name: Naver
        provider:
          kakao:
            authorization_uri: https://kauth.kakao.com/oauth/authorize
            token_uri: https://kauth.kakao.com/oauth/token
            user-info-uri: https://kapi.kakao.com/v2/user/me
            user_name_attribute: id
          naver:
            authorization-uri: https://nid.naver.com/oauth2.0/authorize
            token-uri: https://nid.naver.com/oauth2.0/token
            user-info-uri: https://openapi.naver.com/v1/nid/me
            user-name-attribute: response
```

로그인

```
@Getter
public class OAuthAttributes {
    private Map<String, Object> attributes;
    private String nameAttributeKey;
    private String name;
    private String email;
    private String picture;

    @Builder
    public OAuthAttributes(Map<String, Object> attributes, String nameAttributeKey, String name, String email, String picture) {
        this.attributes = attributes;
        this.nameAttributeKey = nameAttributeKey;
        this.name = name;
        this.email = email;
        this.picture = picture;
    }

    public static OAuthAttributes of(String registrationId, String userNameAttributeName, Map<String, Object> attributes) {
        if("kakao".equals(registrationId)){
            return ofKakao("id", attributes);
        }
        if("naver".equals(registrationId)) {
            return ofNaver("id", attributes);
        }
        return ofGoogle(userNameAttributeName, attributes);
    }

    public static OAuthAttributes ofKakao(String userNameAttributeName, Map<String, Object> attributes) {
        // kakao, naver, id, google, userNameAttributeName
        // Attributes > properties
        // kakao_account > profile
        Map<String, Object> response = (Map<String, Object>) attributes.get("kakao_account");
        Map<String, Object> profile = (Map<String, Object>) response.get("profile");
        return OAuthAttributes.builder()
            .name((String) profile.get("nickname"))
            .email((String) response.get("email"))
            .picture((String) profile.get("profile_image_url"))
            .attributes(attributes)
            .nameAttributeKey(userNameAttributeName)
            .build();
    }
}
```

OAuthAttribute.java

```
private static OAuthAttributes ofGoogle(String userNameAttributeName, Map<String, Object> attributes) {
    return OAuthAttributes.builder()
        .name((String) attributes.get("name"))
        .email((String) attributes.get("email"))
        .picture((String) attributes.get("picture"))
        .attributes(attributes)
        .nameAttributeKey(userNameAttributeName)
        .build();
}

private static OAuthAttributes ofNaver(String userNameAttributeName, Map<String, Object> attributes) {
    Map<String, Object> response = (Map<String, Object>) attributes.get("response");

    return OAuthAttributes.builder()
        .name((String) response.get("name"))
        .email((String) response.get("email"))
        .picture((String) response.get("profile_image"))
        .attributes(response)
        .nameAttributeKey(userNameAttributeName)
        .build();
}

public SnsUser toEntity() {
    return SnsUser.builder()
        .name(name)
        .email(email)
        .picture(picture)
        .role(UserRole.USER)
        .build();
}
```

OAuthAttribute.java

회원가입

The screenshot shows a sign-up form for 'NURIRI'. At the top, there are navigation icons: three horizontal lines, a magnifying glass, and a search bar placeholder 'NURIRI'. Below the search bar is the title '회원가입' (Membership Sign-up). A red box highlights the first two input fields: '아이디' (ID) containing 'grgr' and '비밀번호' (Password) containing '.....'. To the right of the password field, a note says '사용가능한 비밀번호입니다.' (A valid password). Below these fields are input fields for '이름' (Name), '이메일' (Email), and '휴대폰번호' (Phone Number). The next section is '주소' (Address) with fields for '우편번호' (Postal Code), '기본주소' (Basic Address), and '나머지주소' (Remaining Address). At the bottom left, there are several checkboxes for terms and conditions, with the first one checked. On the right side of the bottom row are three '보기' (View) buttons. At the very bottom is a large black button labeled '가입하기' (Sign-up).

회원가입 유효성 체크

아이디: 영문 또는 영문, 숫자 포함 5자~20자 이내

비밀번호: 영문, 숫자, 특수문자 조합 8자리 이상

비밀번호 재입력: 비밀번호와 일치 여부

이름: 한글 2~5자

이메일: 이메일 형식

휴대폰번호: 휴대폰번호 형식

주소: 빈칸이 아닌지 체크 여부

동의: 필수 동의 체크 여부

키를 누를 때마다 유효성 체크 표시

회원가입

회원가입

grgr
아이디는 영문 또는 영문, 숫자를 포함하여 5자~20자 내로 입력 가능합니다. (-_를 제외한 특수 문자 입력 불가)

.....
사용가능한 비밀번호입니다.

.....
비밀번호가 일치합니다.

이름을 입력해주세요

이메일(비밀번호 찾기 시 이메일로 발송됩니다)

휴대폰번호

주소

1 우편번호

기본주소

나머지주소

전체 동의

이용약관 동의 (필수)

개인정보 수집 · 이용 동의 (필수)

개인정보 수집 · 이용 동의 (선택)

2

1. 우편번호칸을 누르면 주소 검색 창으로 이동(Daum API)

2. 가입하기 버튼을 누르면
입력 유효성 체크하고
이미 가입한 사람인지 체크

회원가입이 안될 경우:
입력이 공백 또는 유효하지 않음
이미 가입한 회원
(동일한 이메일, 전화번호
아이디가 이미 등록)

localhost:8080 내용:
비밀번호와 비밀번호 확인창이 일치하지 않습니다.

localhost:8080 내용:
회원가입을 축하합니다.

회원가입

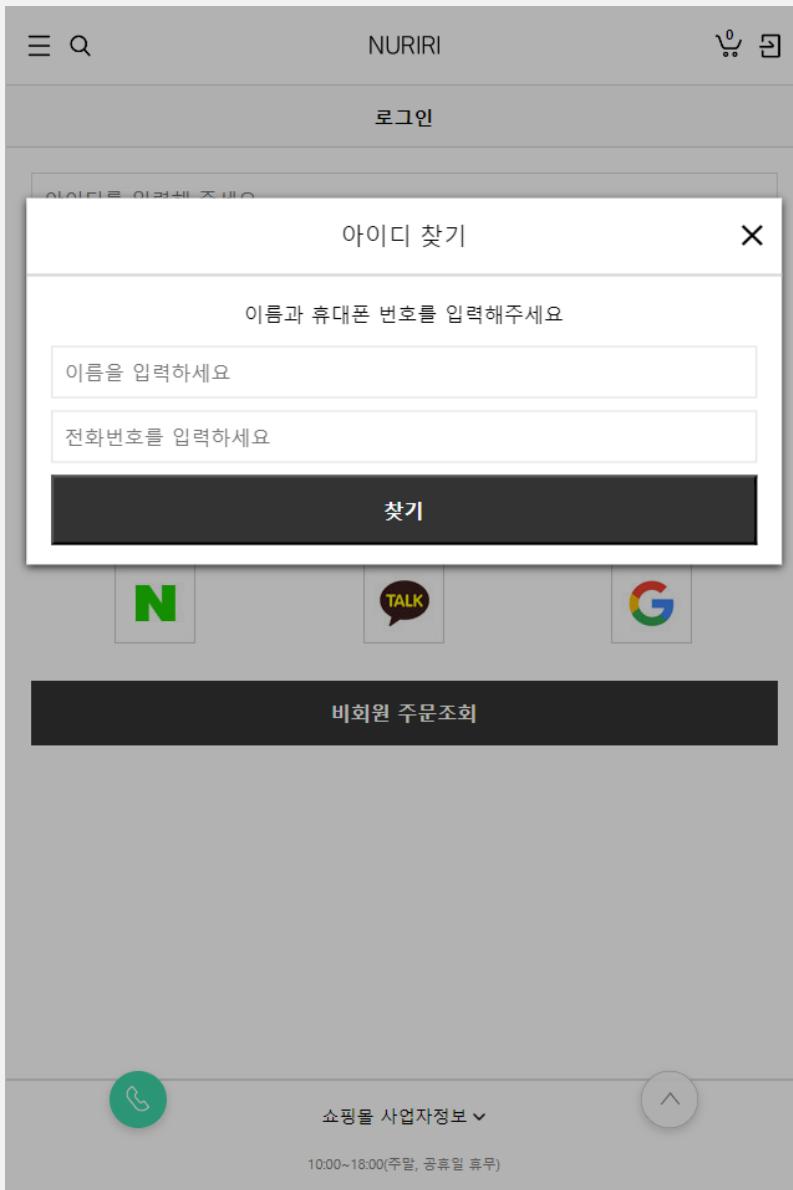
```
@Getter  
@Setter  
@Builder  
public class MemberJoinDto {  
    @Nullable  
    private Long memberNo;  
    @NotBlank  
    @Pattern( regexp = "^[a-zA-Z0-9_-]{5,20}$", message = "아이디를 양식에 맞게 기입해 주세요")  
    private String username;  
    @NotBlank  
    @Pattern( regexp = "(?=.*[0-9])(?=.*[a-z])(?=.*[\W])(?=\\S+).{8,20}" , message = "패스워드를 양식에 맞게 기입해 주세요" )  
    private String password;  
    @Pattern( regexp = "[가-힣]{2,5}$", message = "이름을 양식에 맞게 작성해 주세요" )  
    private String memberName;  
    @NotBlank  
    @Pattern( regexp = "^[A-Za-z0-9_\\.\\-]+@[A-Za-z0-9\\-]+\\.[A-Za-z0-9\\-]+$", message = "이메일을 양식에 맞게 기입해 주세요" )  
    private String memberEmail;  
  
    @NotBlank  
    private String memberRate;  
    @NotBlank  
    @Pattern( regexp = "[0-9]{10,11}$", message = "전화번호를 양식에 맞게 기입해 주세요" )  
    private String memberPhone;  
  
    private Long memberMileage;//마일리지  
  
    private Long memberCoupon;  
    @NotBlank(message = "우편 번호를 기입해 주세요")  
    private String memberAddrNumber;  
    //NotBlank  
    private String memberAddr1;  
    @NotBlank(message = "나머지 주소를 기입해 주세요")  
    private String memberAddr2;  
  
    ...  
}
```

MemberJoinDto.java

```
@PostMapping("/user/joinAction")  
@ResponseBody  
public String joinAction(@Valid MemberJoinDto dto, BindingResult bindingResult) {  
    LocalDate today = LocalDate.now();  
    dto.setMemberJoinDatetime(today);  
  
    if (bindingResult.hasErrors()) {  
        String detail = bindingResult.getFieldError().getDefaultMessage();  
        return "<script>alert('" + detail + "'); history.back();</script>";  
    }  
    String encodedPassword = passwordEncoder.encode(dto.getPassword());  
    dto.setPassword(encodedPassword);  
    try {  
        MemberEntity entity = dto.toSaveEntity();  
        memberRepository.save(entity);  
    } catch (DataIntegrityViolationException e) {  
        e.printStackTrace();  
        bindingResult.reject("signupFailed", "이미 등록된 사용자입니다.");  
        if(e.getRootCause().toString().contains("member_id")){  
            return "<script>alert('이미 등록된 아이디입니다. 다른 아이디를 사용해 주세요.');//history.back();</script>";  
        } else if(e.getRootCause().toString().contains("member_email")){  
            return "<script>alert('이미 등록된 이메일 주소입니다. 다른 이메일 주소를 사용해 주세요.');//history.back();</script>";  
        } else if(e.getRootCause().toString().contains("member_phone")){  
            return "<script>alert('이미 등록된 전화번호입니다. 다른 전화번호를 사용해 주세요.');//history.back();</script>";  
        }  
    } catch (IllegalArgumentException e) {  
        e.printStackTrace();  
        return "<script>alert('회원가입 실패했습니다.');//history.back();</script>";  
    }  
  
    HttpStatus status = HttpStatus.OK;  
    if (status == HttpStatus.OK) {  
        return "<script>alert('회원가입을 축하합니다.');// location.href='/user/login';</script>";  
    } else {  
        return "<script>alert('회원가입 실패했습니다.');// history.back();</script>";  
    }  
}
```

UserInfoController.java

아이디 찾기



로그인 페이지에서
아이디 찾기를 눌러 확인 가능

정확한 입력을 했을 때
alert창이 뜨며 아이디 확인 가능

localhost:8080 내용:
회원님의 아이디는 testId 입니다.

확인

비밀번호 찾기

The image shows a composite screenshot of a web application's password recovery feature across three main sections:

- Left Section (Login Page):** Shows a form titled "비밀번호 찾기" (Password Recovery) with a placeholder "이메일을 입력해주세요" (Please enter your email) and a "발송하기" (Send) button. Below the form are social sharing icons for Naver, KakaoTalk, and Google. A callout box with a red arrow points to the "비밀번호 찾기" button, containing the text "로그인 페이지에서 비밀번호 찾기 누르기" (Press the password recovery button on the login page).
- Middle Section (Email Confirmation):** Displays an incoming email from "mysoho.sample@naver.com" with the subject "マイソホ サンプル店 パスワード変更案内メールです". The email body contains a link to "mysoho.sample.com/password-change". A callout box with a red arrow points to the email body, containing the text "입력된 메일로 비밀번호 변경 메일이 보내짐" (A password change email will be sent to the entered email address).
- Right Section (Password Change Page):** Shows a page titled "마이소호 サンプル店 パスワード変更" (MySoHo Sample Shop Password Change). It includes a "비밀번호 변경하기" (Change Password) button and a note about password change requests. A callout box with a red arrow points to the "비밀번호 변경하기" button, containing the text "비밀번호 변경하기 버튼을 누르면 변경 페이지로 이동해 변경 가능" (When you press the change password button, it will move to the change page and allow you to change it).

At the bottom right of the image, there is a small ".java" file icon.

비밀번호 찾기

```
public boolean sendEmail(final String getEmail){
    try {
        String content =
            " <div style=\"height:100%;background:#fff;color:#000;font-size:15px;line-height:1.5;letter-spacing:-0.5px\"> +
            "<table style=\"text-align:left;table-layout:fixed;width:100%;border:0;border-spacing:0;border-collapse:collapse\">"+
            "<tbody>"+
            "<tr>"+
            "<td>"+
            "<div style=\"max-width:740px;border:1px solid #d7d7d7;letter-spacing:-1px\">+
            "<div style=\"border-bottom:1px solid #d7d7d7;background:#f6f6f6\">"+
            "<h1 style=\"padding:20px 6%;margin:0\"><span style=\"display:block;font-size:20px;font-weight:normal;color:#000;word-break:break-word\">마이소호
샘플샵 비밀번호 변경 링크 안내 이메일입니다</span></h1>+
            "</div>+
            "<div style=\"padding:4% 6%\">+
            "<h2 style=\"font-size:32px;color:#00b0ff;word-break:break-word!important\">마이소호 샘플샵 비밀번호 변경 </h2>+
            "<div>+
            "<p style=\"margin-bottom:25px;font-size:17px;color:#454545;line-height:25px;word-break:break-word!important\">+
            "안녕하세요. 마이소호 샘플샵입니다. <br>+
            "비밀번호 변경을 원하실 경우<br>+
            "하단 비밀번호 변경하기를 눌러주세요."+
            "<br><br>+
            "<a href=\"http://localhost:8080/find/password2?getEmail="+getEmail+"\" rel=\"noopener\" style=\"display:inline-block;width:300px;height:50px;line-height:52px;background:#333;box-sizing:border-box;color:#fff;font-size:16px;font-weight:bold;text-align:center;text-decoration:none\" target=\"_blank\">비밀번호 변경 하기</a>"+
            "<br><br>+
            "</p>+
            "</div>+
            "</div>+
            "<div style=\"overflow:hidden;height:auto;padding:10px 0;border-top:1px solid #d7d7d7;background:#f6f6f6\">+
            "<div style=\"width:100%;margin-right:15px;font-size:13px;color:#000;line-height:26px;letter-spacing:0;padding:0 6%\">+
            "<p style=\"margin:0;padding:0\">서울 금천구 가산디지털1로 168 (가산동, 우컴라이온스밸리) A동 14층 | 사업자번호 2068121131</p>+
            "<p style=\"margin:0;padding:0\">02-6903-9118 | <a href=\"mailto:help@mysoho.com\" style=\"color:#0000ff!important;text-decoration:none\" target=\"_blank\">help@mysoho.com</a></p>+
            "</div>+
            "</div>+
            "</td>+
            "</tr>+
            "</tbody>+
            "</table><div class=\"yj6qo\"></div><div class=\"adL\">+
            "</div></div>;

MimeMessage mimeMessage = javaMailSender.createMimeMessage();
MimeMessageHelper mimeMessageHelper = new MimeMessageHelper(mimeMessage,true,"UTF-8");
mimeMessageHelper.setTo(getEmail);
mimeMessageHelper.setFrom("mysoho2023@naver.com");
mimeMessageHelper.setSubject("마이소호 샘플샵 비밀번호 변경 안내 이메일입니다");
mimeMessageHelper.setText(content,true);
javaMailSender.send(mimeMessage);
return true;
}catch (MessagingException e){
    e.printStackTrace();
    return false;
}
}
```

MemberService.java

비회원 주문조회

The screenshot shows a web page titled '비회원 주문조회' (Non-member Order Inquiry). At the top, there are navigation icons for search, refresh, and language selection. Below the title, a sub-header reads '비회원 주문조회'. A magnifying glass icon with a list icon is displayed above a message: '나의 구매 내역을 확인할 수 있습니다.' (You can check your purchase history). The main form area has a red border around the input field for '주문자명' (Orderer Name), which contains the placeholder '010' and the instruction '- 없이 숫자만 입력' (Enter only numbers). Below this is a large black button labeled '주문조회' (Order Inquiry). At the bottom of the page, there is a section for '쇼핑몰 사업자정보' (Shop Operator Information) with a dropdown menu showing '쇼핑몰 사업자정보 ▼', a note about operating hours '10:00~18:00(주말, 공휴일 휴무)', and contact details: phone number '02-6903-9118' and email 'mysoho2023@naver.com'. The footer includes links for '이용약관' (Terms of Service), '개인정보처리방침' (Privacy Policy), 'Q&A 게시판' (FAQ Board), and a 'free service by mysoho' logo. On the right side, there is a small circular icon with an upward arrow and a link to '소모바켓'.

주문자명/주문자 핸드폰번호 입력 후
주문 조회를 누르면
비회원 주문 조회 페이지로 이동

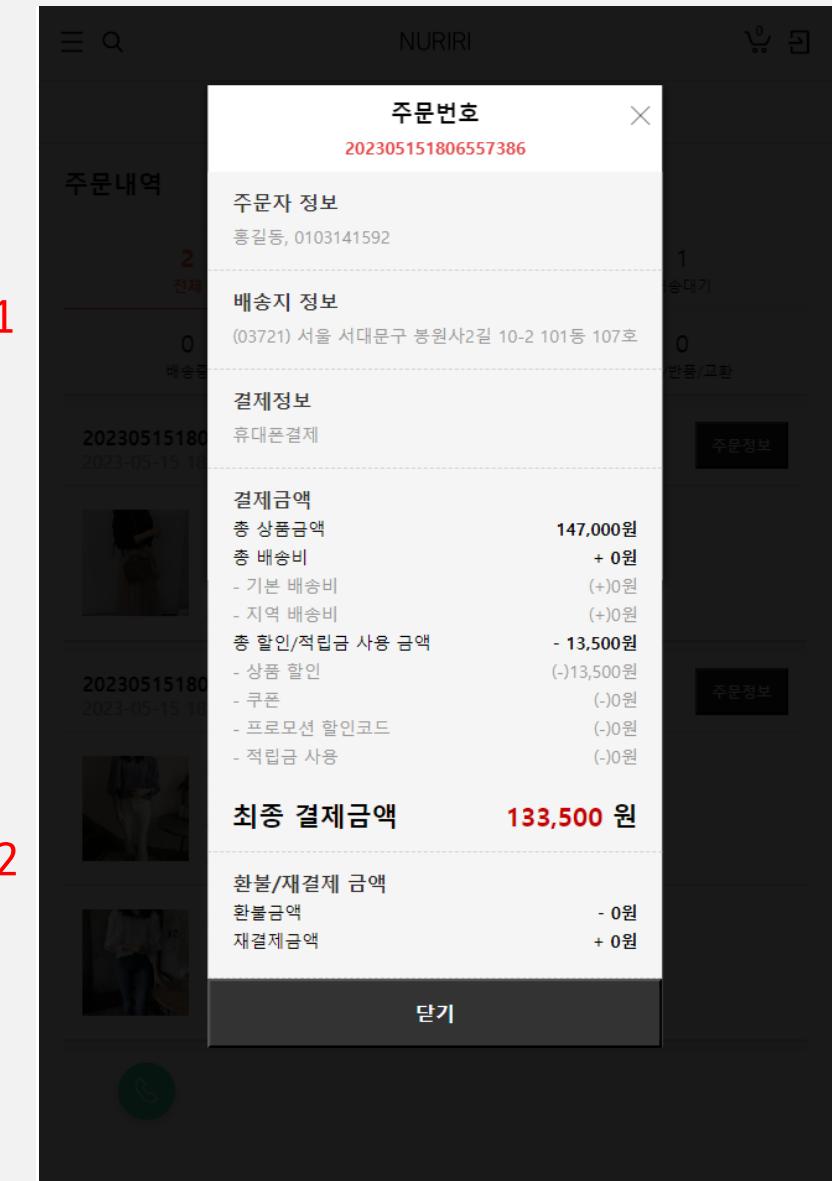
주문 정보가 없으면 alert 창

localhost:8080 내용:

주문 정보가 없습니다.

확인

비회원 주문조회



1. 주문 수량 및 상태 확인 가능
2. 주문 번호/날짜/주문 상태/상품 정보/가격/수량 확인 가능
3. 주문정보 버튼을 누르면 결제 정보 확인 가능

비회원 주문조회



```
@Transactional(readOnly = true)
public List<CartResponseDto> getCartList(OrderResponseDto dto) {
    List<CartResponseDto> cartList = new ArrayList();
    if (dto.getCartCode1() != null) {
        //주문정보의 장바구니 코드로 장바구니 정보 dto 생성
        CartResponseDto responseDto = findByCart(dto.getCartCode1());
        //장바구니 리스트에 셋팅된 dto 담기
        cartList.add(responseDto);
    }
    if (dto.getCartCode2() != null) {
        //반복
        cartList.add(findByCart(dto.getCartCode2()));
    }
    ... // 생략
    return cartList;
}
```

CartService.java

```
@Transactional(readOnly = true)
public List<CartResponseDto> getCartList(OrderResponseDto dto, String type) {
    List<CartResponseDto> cartList = new ArrayList();

    if (dto.getCartCode1() != null) {
        //주문정보의 장바구니 코드로 장바구니 정보 dto 생성
        CartResponseDto cartDto;
        //회원일 때
        if (type.equals("member")) {
            cartDto = findByCartMember(dto.getCartCode1());
        }
        //비회원일 때
        else {
            cartDto = findByCartNonMember(dto.getCartCode1());
        }
        //장바구니에 상품 이미지 url 셋팅
        cartDto.setItemImageUrl(productRepository.findByUrl(cartDto.getItemCode()));
        //장바구니 리스트에 셋팅된 dto 담기
        cartList.add(cartDto);
    }
    if (dto.getCartCode2() != null) {
        //2~5 카트 반복
        CartResponseDto cartDto;
        if (type.equals("member")) {
            cartDto = findByCartMember(dto.getCartCode2());
        }
        else {
            cartDto = findByCartNonMember(dto.getCartCode2());
        }
        cartDto.setItemImageUrl(productRepository.findByUrl(cartDto.getItemCode()));
        cartList.add(cartDto);
    }
    ... // 생략

    return cartList;
}
```

CartService.java

비회원 주문조회



```
public int[] orderStateType(List<OrderResponseDto> orderList) {
    int stateType1 = 0;
    int stateType2 = 0;
    int stateType3 = 0;
    int stateType4 = 0;
    int stateType5 = 0;

    for (OrderResponseDto orderDto : orderList) {
        switch (orderDto.getOrderState()) {
            case "결제대기" -> stateType1++;
            case "배송대기" -> stateType2++;
            case "배송중" -> stateType3++;
            case "배송완료" -> stateType4++;
            default -> stateType5++;//취소/반품
        }
    }
    int[] stateType = {stateType1, stateType2, stateType3, stateType4, stateType5};

    return stateType;
}
```

UserInfoController.java



```
function changeRed(n) {
    growArr = document.querySelectorAll(".grow ");
    for (let i = 0; i < 6; i++) {
        growArr[i].classList.remove("red");
        document.querySelector(".list-type" + i).style.display = "none";
    }
    growArr[n].classList.add("red");
    document.querySelector(".list-type" + n).style.display = "block";

    if (document.querySelector("#value" + n).innerHTML === "0") {
        alert("주문정보가 없습니다.");
    }
}
```

myorder.js

마이페이지

☰ Q NURIRI ☰

마이페이지

정희진 님

1 내 정보 로그아웃

2 ① 나의 적립금
1,780원

② 나의 쿠폰
0원

3 나의 후기 내역 >
나의 문의 내역 >

주문내역

1 전체	0 미입금/주문완료	0 배송대기
0 배송중	1 배송/처리완료	0 취소/반품/교환

4 202305151639139873
2023-05-15 16:39

주문정보

후기작성

배송완료 퍼프 블라우스 화이트 44,500원 / 2개

1. 내정보 페이지로 이동
로그아웃
2. 적립금/쿠폰 확인
3. 나의 후기 페이지로 이동
나의 문의 페이지로 이동
4. 주문 내역 확인
비회원 주문조회와 거의 동일
배송완료되면 후기 작성 가능

마이페이지

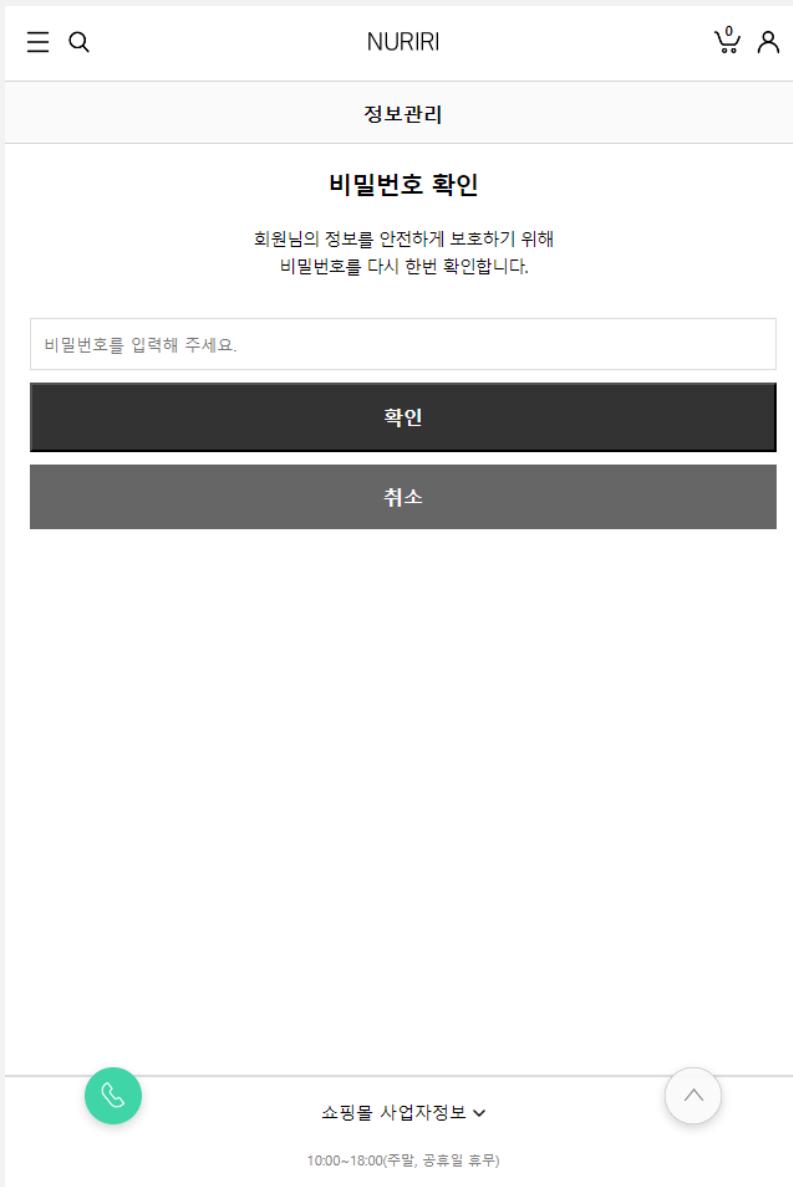
```
①②③  
@Transactional(readOnly = true)  
public List<OrderResponseDto> findByOrderList(String id) {  
    //회원 아이디로 카트테이블에서 정보 가져오기  
    List<CartEntity> cartEntityList = cartRepository.findByCartMemberId(id);  
    //entity → dto로 변환  
    List<CartResponseDto> cartDtoList = cartEntityList.stream().map(CartResponseDto::new).toList();  
    //카트 테이블에서 주문번호만 추출해서 담을 orderCodeList 생성  
    List<Long> orderCodeList = new ArrayList<>();  
    //반복해서 주문번호만 추출 후 리스트에 담기  
    for (CartResponseDto cartDto : cartDtoList) {  
        orderCodeList.add(cartDto.getOrderCode());  
    }  
    //주문번호리스트에서 중복 값 제거  
    List<Long> newList = orderCodeList.stream().distinct().toList();  
    //주문번호들로 주문테이블에서 정보 가져와서 담을 리스트 생성  
    List<OrderResponseDto> orderList = new ArrayList<>();  
    //반복해서 주문테이블 객체 가져와서 dto로 변환 후 리스트에 담기  
    for (Long orderCode : newList) {  
        OrderEntity orderEntity = orderRepository.findByOrderCode(orderCode).get();  
        OrderResponseDto dto = new OrderResponseDto(orderEntity);  
        orderList.add(dto);  
    }  
    return orderList;  
}
```

CartService.java

myorder-list-user.html

```
①②③  
function changeRed(n)  
public List<CartResponseDto> getCartList(OrderResponseDto dto)  
public List<CartResponseDto> getCartList(OrderResponseDto dto, String type)  
public int[] orderStateType(List<OrderResponseDto> orderList)
```

내 정보



내 정보 버튼을 누르면
내 정보 페이지에 가기 전에
비밀번호를 확인

확인 완료되면 alert창과 함께
내정보 페이지로 이동

localhost:8080 내용:
비밀번호 확인완료

확인

확인 실패 시 alert창

localhost:8080 내용:
비밀번호 확인실패

확인

내 정보

The screenshot shows a mobile application interface for managing personal information. At the top, there are navigation icons: three horizontal lines, a magnifying glass, the text 'NURIRI', and a language switch. Below this is a header '내정보' (My Information). The main content area is divided into sections:

- 기본정보**: Contains fields for '기존 비밀번호' (Existing Password), '새 비밀번호' (New Password), '비밀번호 확인' (Password Confirmation), '정회진' (Return Address), an email field with 'testId@gmail.com', and a phone number field with '01054245046'. A red box labeled '1' highlights the '비밀번호 확인' field.
- 주소**: Contains fields for '05088' (Postcode), '서울 광진구 자양강변길 7' (Address), and '107동 108호' (Apartment). A red box labeled '2' highlights the '107동 108호' field.
- At the bottom, there is a checkbox for '개인정보 수집 이용동의(선택)' (Personal Information Collection Use Agreement (Optional)) with the subtext '(쇼핑/할인소식 SMS수신)' (Shopping/Discount News SMS Reception) and a '보기' (View) button. Below this is a red button bar containing two buttons: '탈퇴하기' (Logout) and '변경하기' (Change).

1. 내 정보 수정

비밀번호/닉네임/이메일/
전화번호/주소 변경 후
변경하기 버튼을 누르면
유효성 체크 및 수정

2. 회원탈퇴

리뷰

마이페이지

☰ Q NURIRI ☰ ☰

마이페이지

정회자 님

내 정보 로그아웃

① 나의 적립금
1,780원

② 나의 쿠폰
0원

나의 후기 내역 >

나의 문의 내역 >

주문내역

1 전체	0 미입금/주문완료	0 배송대기
0 배송중	1 배송/처리완료	0 취소/반품/교환

202305151639139873
2023-05-15 16:39

주문정보

후기작성

배송완료
퍼프 블라우스 화이트
44,500원
/ 2개

주문이 배송완료 상태가 되면
후기 작성 버튼이 나타나고
누르면 리뷰 작성 페이지로 이동

리뷰 작성 페이지

☰ Q NURIRI ☰ ☰

리뷰 작성

퍼프 블라우스 화이트
49000원

별점을 선택해주세요.

어떤점이 좋으셨나요?

파일 선택 선택된 파일 없음

등록

별점/후기/사진 입력 후
등록버튼을 누르면
유효성 체크 후 등록

쇼핑몰 사업자정
10:00~18:00(주말, 공휴일 휴무)

02-6903-9118 myoho2023@naver.com

이용약관 | 개인정보처리방침 | Q&A 게시판

free service by myoho

온마켓 >

리뷰

내가 작성한 리뷰 페이지

NURIRI

내가 작성한 리뷰

사진후기
퍼프 블라우스 화이트
좋아요!
★★★★★ / 2023-05-15 17:46

일반후기
퍼프 블라우스 화이트
너무 예뻐요~~
★★★★ / 2023-05-15 17:44

내가 작성한 리뷰 페이지에서
목록확인/수정/삭제 가능

쇼핑몰 사업자정보 ▼
10:00~18:00(주말, 공휴일 휴무)
02-6903-9118 mysoho2023@naver.com

이용약관 | 개인정보처리방침 | Q&A 게시판

free service by mysoho

소호마켓 >

리뷰 수정 페이지

리뷰 수정

퍼프 블라우스 화이트
49000원

★★★★ 별점을 선택해주세요

좋아요!

파일 선택 선택된 파일 없음

수정

쇼핑몰 사업자정보 ▼
10:00~18:00(주말, 공휴일 휴무)

리뷰

상품 상세 페이지

☰ 🔎 NURIRI ☰ ☰

♡ NURIRI SALE ♡ TOP BOTTOM ONE-PIECE ACC

상세정보 후기(2) 문의(3)

★★★★★ 4.5
2 개의 리뷰가 있습니다

사진 후기 모아보기(1)

평점 ★★★★★



좋아요!

te***** | 2023-05-15 17:46

평점 ★★★★★

너무 예뻐요~~

te***** | 2023-05-15 17:44

결제하기

구매한 상품의 상세 페이지에서
작성된 모든 후기를 볼 수 있음

QnA

QnA 목록 페이지

The screenshot shows a mobile application interface. At the top, there is a navigation bar with icons for home, search, and account. Below the navigation bar, there is a banner for "NURIRI SALE". The main content area is titled "QnA" and displays a list of questions and answers. Each item in the list includes the question text, the user's name (e.g., "상품문의"), the nickname ("닉네*"), and the date ("2023.05.12"). There are also "답변 대기" (Pending Response) and "답변 완료" (Response Completed) status indicators. On the right side of each list item, there are three vertical dots, and next to them are buttons for "수정" (Edit) and "삭제" (Delete). A large red arrow points from the bottom left towards the "QnA 게시판" button in the navigation bar.

상단 ⚙ 을 누르고
Q&A 게시판으로 들어가기

▶ 공지사항

QnA 게시판

☰ QnA 목록 페이지

NURIRI

☰ QnA 게시판

검색어를 입력해 주세요

×

☰ 문의작성

답변 대기 비밀글입니다. 상품문의 | 닉네* | 2023.05.12

답변 대기 안녕하세요. 상품문의 | 닉네* | 2023.05.12

답변 대기 비밀글입니다. 상품문의 | je*** | 2023.05.12

답변 완료 안녕하세요. 상품문의 | je*** | 2023.05.12

수정

삭제

74

QnA

QnA 목록 페이지

☰ Q NURIRI ⌂ ⌂

Q&A 게시판

1 < 검색어를 입력해 주세요

2 문의작성

답변 대기 비밀글입니다. 상품문의 | 낙네* | 2023.05.12

3

답변 대기 안녕하세요. 상품문의 | 낙네* | 2023.05.12

4

답변 대기 비밀글입니다. 상품문의 | je*** | 2023.05.12

답변 완료 안녕하세요. 상품문의 | je*** | 2023.05.12

📞 ⌂ ⌂

1. 검색어로 문의 찾기

2. 문의 작성 페이지로 이동

3. 문의 수정 및 삭제

비회원일 경우 비밀번호 입력 후
수정/삭제 가능

답변 대기 반품 원합니다.
상품문의 | 홍길* | 2023.05.16

비밀번호를 입력하세요.

확인

4. 관리자가 답변할 경우

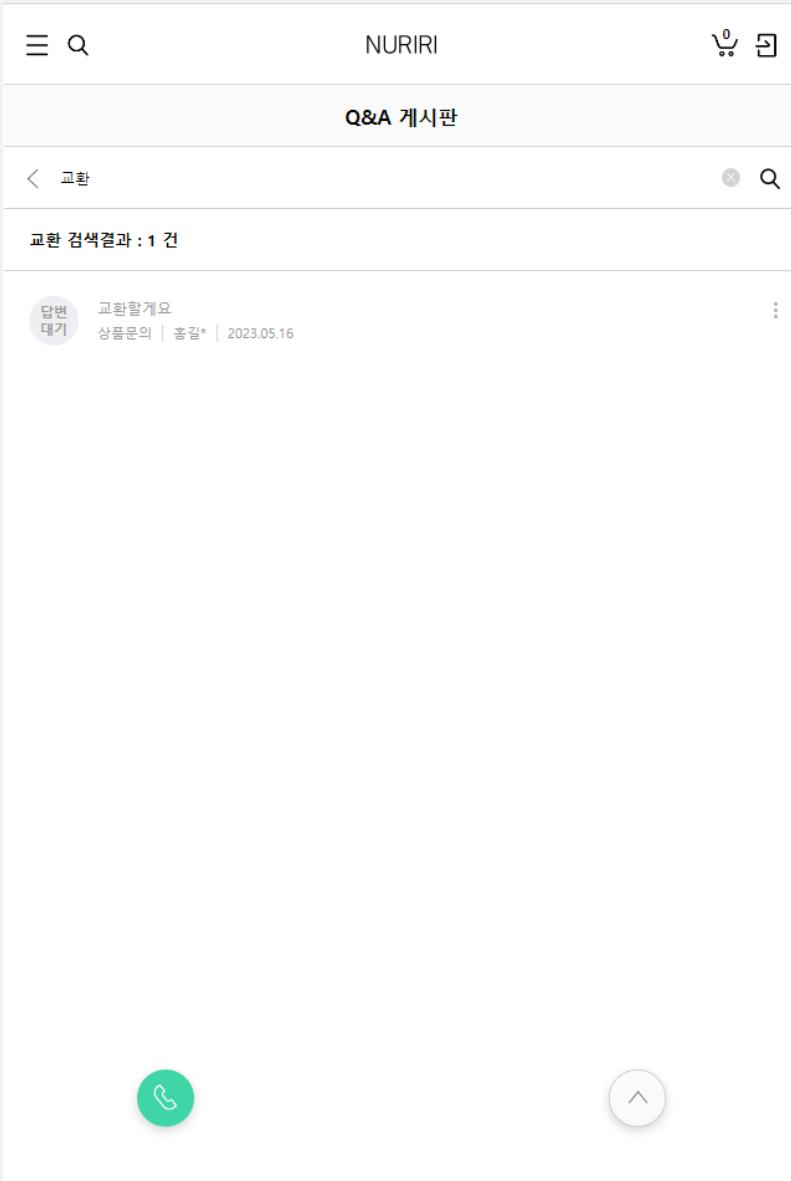


비밀글일 경우 비밀글입니다. 🔒

닉네임/아이디 마스킹 je***

QnA

QnA 목록 페이지



검색창에 검색어를 입력하여 문의 검색

검색 결과가 없을 경우



QnA

QnA 작성 페이지

NURIRI

문의작성

상품문의

1

문의내용

2

비밀글

3

닉네임을 입력해주세요. (20자 이내)

문의 내용을 입력해주세요.(1,000자 이내)

비밀번호를 입력하세요.

* 입력한 비밀번호는 게시글 확인 시 필요합니다.

35386937

새로고침

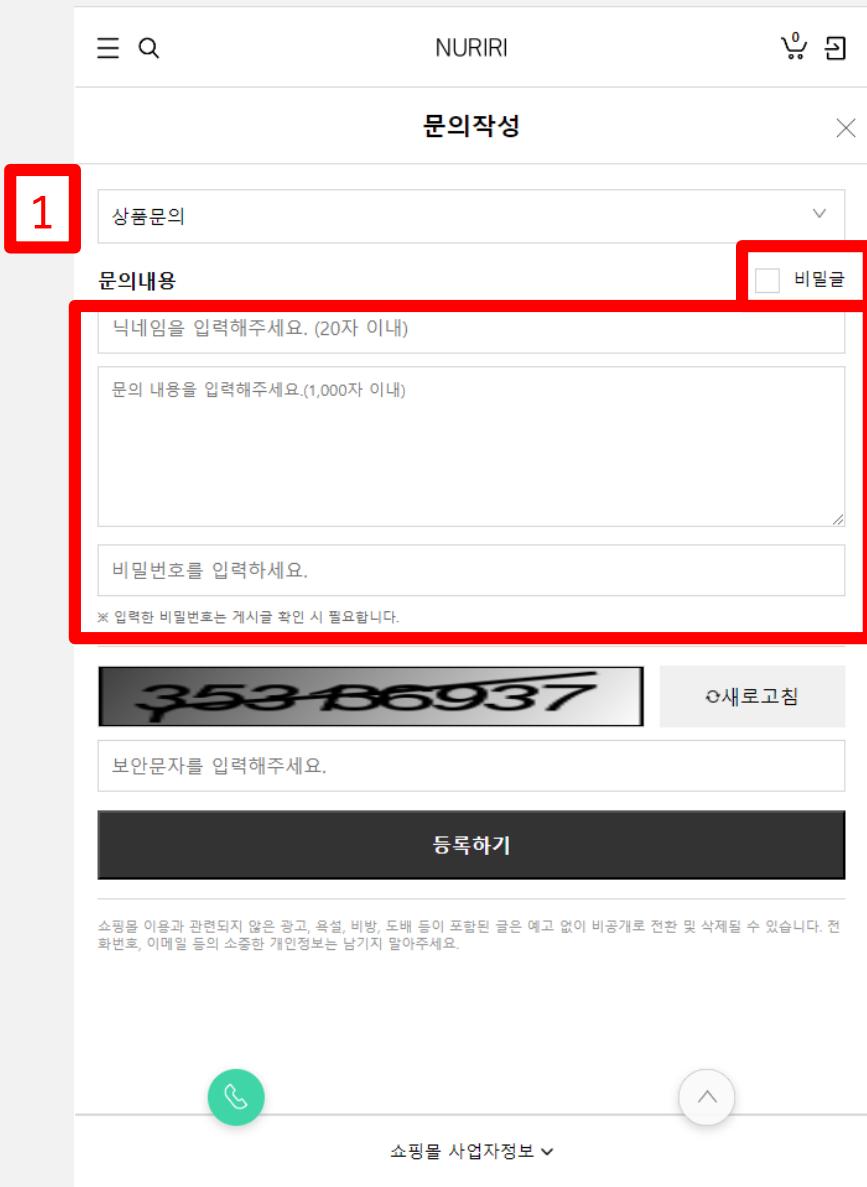
보안문자를 입력해주세요.

등록하기

소핑몰 이용과 관련되지 않은 광고, 육설, 비방, 도배 들이 포함된 글은 예고 없이 비공개로 전환 및 삭제될 수 있습니다. 전화번호, 이메일 등의 소중한 개인정보는 남기지 말아주세요.

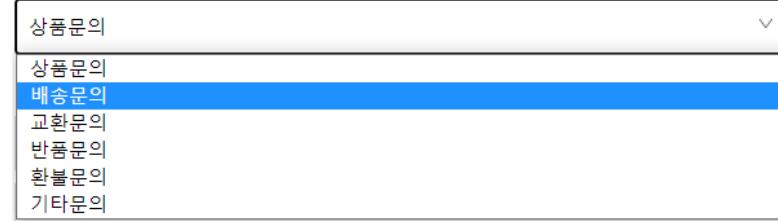
77

쇼핑몰 사업자정보 ▼



Q&A 목록 페이지에서 문의작성
을 누르고 작성 페이지로 이동

1. 문의 유형 선택



상품/배송/교환/환불/기타 문의

2. 비밀글 여부 선택

3. 닉네임/문의 내용/비밀번호 입력

회원일 경우 닉네임과 비밀번호
는 불필요하므로 나타나지 않음

QnA

QnA 작성 페이지

☰ Q NURIRI ☰ ☰

문의작성 ×

상품문의

문의내용 비밀글
닉네임을 입력해주세요. (20자 이내)

문의 내용을 입력해주세요.(1,000자 이내)

비밀번호를 입력하세요.

* 입력한 비밀번호는 게시글 확인 시 필요합니다.

353186937

새로고침

보안문자를 입력해주세요.

등록하기

소핑몰 이용과 관련되지 않은 광고, 유통, 비방, 도배 등이 포함된 글은 예고 없이 비공개로 전환 및 삭제될 수 있습니다. 전화번호, 이메일 등의 소중한 개인정보는 남기지 말아주세요.

쇼핑몰 사업자정보 ▼

4

5

4. 보안문자 기능

5. 등록하기 버튼을 누르면
유효성 검사 후 문의 등록

localhost:8080 내용:

닉네임을 입력해주세요

확인

localhost:8080 내용:

등록되었습니다

확인

QnA

보안 문자



```
public class CaptchaUtil {  
    private static int width = 150;      /*보안문자 이미지 가로크기*/  
    private static int height = 50;     /*보안문자 이미지 세로크기*/  
  
    /*CaptCha Image 생성*/  
    public void getImgCaptCha(HttpServletRequest req, HttpServletResponse res) {  
        /*폰트 및 컬러 설정*/  
        List<Font> fontList = new ArrayList<Font>();  
        fontList.add(new Font("", Font.HANGING_BASELINE, 30));  
        fontList.add(new Font("Courier", Font.ITALIC, 30));  
        fontList.add(new Font("", Font.PLAIN, 30));  
        List<Color> colorList = new ArrayList<Color>();  
        colorList.add(Color.BLACK);  
  
        Captcha captcha = new Captcha.Builder(width, height)  
            // .addText() 또는 아래와 같이 정의 : 6자리 숫자와 폰트 및 컬러 설정  
            .addText(new NumbersAnswerProducer(9), new DefaultWordRenderer(colorList, fontList))  
            .addNoise().addBorder()  
            .addBackground(new GradiatedBackgroundProducer())  
            .build();  
  
        /*JSP에서 Captcha 객체에 접근할 수 있도록 session에 저장*/  
        req.getSession().setAttribute(Captcha.NAME, captcha);  
        CaptchaServletUtil.writeImage(res, captcha.getImage());  
    }  
  
}
```

CaptchaUtil.java



```
@Controller  
@Slf4j  
public class CaptchaController {  
  
    // captcha 이미지 가져오는 메서드  
    @GetMapping("/captchaImg.do")  
    @ResponseBody  
    public void captchaImg(HttpServletRequest req, HttpServletResponse res) throws Exception{  
        new CaptchaUtil().getImgCaptCha(req, res);  
    }  
}
```

CaptchaController.java



```
<script>  
function getImage(){  
    var rand = Math.random();  
    var url = '/captchaImg.do?rand=' + rand;  
    document.querySelector('#captchaImg').setAttribute('src', url);  
}  
</script>
```

qna-write.html

QnA

QnA 수정 페이지(답변 없을 때)

☰ Q NURIRI ☰

문의작성 ×

상품문의

문의내용

1 흥길동

비밀글

교환할게요

비밀번호를 입력하세요.

※ 입력한 비밀번호는 게시글 확인 시 필요합니다.

02 203052

새로고침

보안문자를 입력해주세요.

2 수정하기

소핑몰 이용과 관련되지 않은 광고, 육설, 비방, 도배 등이 포함된 글은 예고 없이 비공개로 전환 및 삭제될 수 있습니다. 전화번호, 이메일 등의 소중한 개인정보는 남기지 말아주세요.

쇼핑몰 사업자정보 ▾

80

Q&A 목록 페이지에서 수정버튼을 누르고 수정 페이지로 이동

1. 문의 등록 페이지와 마찬가지로 회원일 경우 닉네임/비밀번호는 나타나지 않음
2. 수정하기 버튼을 제외하고 문의 등록 페이지와 동일

관리자가 답변을 등록하지 않았다면 수정 가능

QnA

QnA 수정 페이지(답변 있을 때)

☰ Q NURIRI ☰

문의작성 ×

상품문의 ▾

1 문의내용

1 흥길동

2 교환할게요

2 답변내용

3 네

소평을 이용과 관련되지 않은 광고, 육설, 비방, 도배 등이 포함된 글은 예고 없이 비공개로 전환 및 삭제될 수 있습니다. 전화번호, 이메일 등의 소중한 개인정보는 남기지 말아주세요.

쇼핑몰 사업자정보 ▾

10:00~18:00(주말, 공휴일 휴무)

📞 02-6903-9118 📩 mysoho2023@naver.com

1. 회원일 경우 닉네임 부분은 나타나지 않음
2. 관리자가 답변을 등록했다면 수정 불가능
3. 관리자의 답변 내용이 보여짐

QnA

The screenshot shows the 'My Page' section of a mobile application. At the top, there are navigation icons: three horizontal lines, a magnifying glass, the text 'NURIRI', and a user profile icon. Below the navigation bar, there are two tabs: '나의 상품문의' (My Product Inquiry) and '나의 Q&A'. The '나의 Q&A' tab is selected. A red box highlights the '문의작성' (Inquiry Creation) button, which has a pen icon. Another red box highlights a question card with the number '2' and the text '교환 원합니다~~~' (Want to exchange~~~). To the right of the card are '수정' (Edit) and '삭제' (Delete) buttons. At the bottom of the screen, there is a footer with links: '소핑몰 사업자정보' (Business Information), operating hours '10:00~18:00(주말, 공휴일 휴무)', contact numbers '02-6903-9118' and email 'mysoho2023@naver.com', and legal links '이용약관', '개인정보처리방침', and 'Q&A 게시판'. The bottom right corner features a green circular icon with a white phone receiver.

마이페이지에서 나의 문의 내역으로 들어가면 해당 유저가 작성한 QnA를 볼 수 있음

1. 문의 작성 페이지로 이동
2. 유저의 문의내역 확인/수정/삭제

등록된 QnA가 없을 경우



상품문의

상품 상세 페이지

The screenshot shows a product detail page for a NURIRI item. At the top, there are navigation icons for search, category, and account. Below that, a promotional banner for 'NURIRI SALE' is displayed, followed by the price information: '9% 44,500원' (Original price 49,000원). A quantity selector with buttons for minus, one, and plus is shown next to the price. Below this is a large black button labeled '장바구니 담기' (Add to Cart) with a shopping bag icon. Underneath the price, there are three tabs: '상세정보' (Product Details), '후기(0)' (Reviews), and '문의(3)' (Inquiries). A red box highlights the '문의(3)' tab, which contains a list of three inquiries. Each inquiry has a '답변 대기' (Pending Response) status, a message content, and a timestamp ('2023.05.15'). To the right of each message is a vertical ellipsis menu with options for '수정' (Edit) and '삭제' (Delete). Below the inquiry list, there are sections for '주문자 정보' (Buyer Information), '적립금 사용' (Use Points), and '결제방법' (Payment Method). The payment method section includes a radio button for '삼성Pay' and a link to '지갑 단말 확인' (Check Wallet Device). At the bottom is a large black '결제하기' (Proceed to Payment) button.

상품 상세페이지의 문의 탭에서
상품문의 작성/수정/삭제 가능

상품 문의 목록 확인 가능

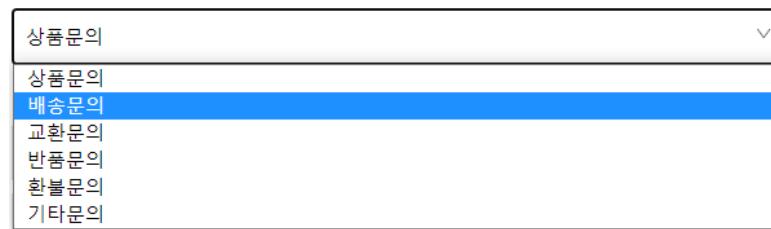
상품문의

상품문의 작성 페이지

The screenshot shows a mobile application interface for NURIRI. At the top, there is a search bar with a magnifying glass icon and the brand name 'NURIRI'. Below the search bar, the title '문의작성' (Inquiry Writing) is displayed, with a red rectangular box highlighting the '상품문의' (Product Inquiry) tab. The main form area contains fields for '문의내용' (Inquiry Content), which includes a nickname input field ('닉네임을 입력해주세요. (20자 이내)'), a main inquiry text area ('문의 내용을 입력해주세요. (1,000자 이내)'), and a password input field ('비밀번호를 입력하세요.'). A note at the bottom of this section states: '※ 입력한 비밀번호는 게시글 확인 시 필요합니다.' (A password entered here is required for viewing posts). Below these fields is a large button containing the number '90-1365454'. To the right of this button is a '새로고침' (Refresh) button. Further down is a '등록하기' (Register) button. At the very bottom of the form, a small note reads: '쇼핑몰 이용과 관련되지 않은 광고, 육설, 비방, 도배 등이 포함된 글은 예고 없이 비공개로 전환 및 삭제될 수 있습니다. 전화번호, 이메일 등의 소중한 개인정보는 남기지 말아주세요.' (Comments related to shopping mall use, such as advertising, libel, or spam, will be converted to private and deleted without notice. Please do not leave sensitive personal information like phone numbers or emails.).

문의 유형이 1가지(상품문의)라는 점을 제외하고 QnA 작성 페이지와 동일

QnA은 문의 유형 선택 가능



닉네임/문의내용/비밀번호/
비밀글 여부/보안 문자 입력 후
문의 등록

회원일 경우 닉네임/비밀번호는
나타나지 않음

상품문의

상품문의 수정 페이지(답변 없을 때)

☰ Q NURIRI ☰ 🔍

문의작성 ×

1 상품문의

2 정회진

비밀번호를 입력하세요.

※ 입력한 비밀번호는 게시글 확인 시 필요합니다.

23328537

새로고침

보안문자를 입력해주세요.

3 수정하기

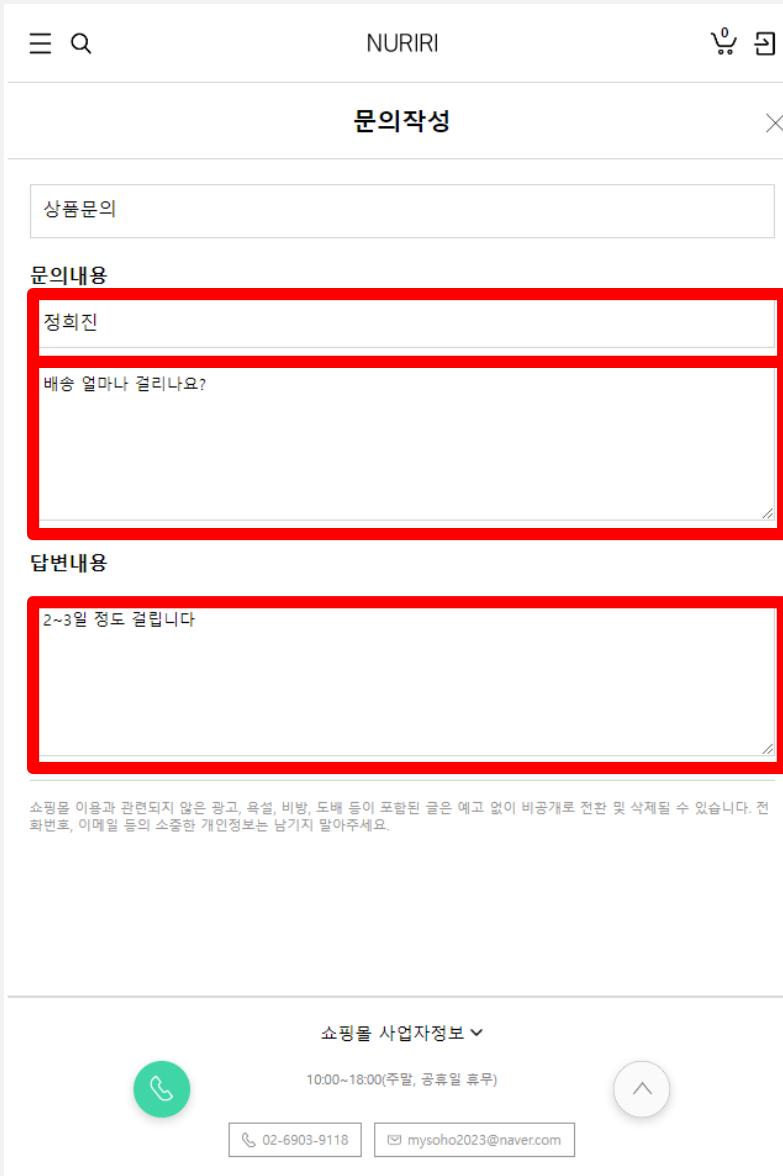
쇼핑몰 사업자정보 ▾

1. 문의 유형이 1가지(상품문의)
2. 문의 등록 페이지와 마찬가지로 회원일 경우 닉네임/비밀번호는 나타나지 않음
3. 수정하기 버튼을 제외하고 문의 등록 페이지와 동일

관리자가 답변을 등록하지 않았다면 수정 가능

상품문의

상품문의 수정 페이지(답변 있을 때)



1. 회원일 경우 닉네임 부분은 나타나지 않음
2. 관리자가 답변을 등록했다면 수정 불가능
3. 관리자의 답변 내용이 보여짐

상품문의

나의 문의 페이지

The screenshot shows the 'My Inquiry' section of the NURIRI website. At the top, there are navigation icons for search, account, and language. Below that, tabs for 'My Product Inquiry' and 'My Q&A' are visible, with 'My Product Inquiry' being active. A red box highlights a specific inquiry entry:

**퍼프 블라우스 화이트
49000원**

답변 대기
다른 색은 없나요?
상품문의 | te***** | 2023.05.15

수정
삭제

At the bottom of the page, there are sections for 'Business Information' (including address, operating hours, phone number, and email), 'User Agreement', 'Privacy Policy', and 'FAQ'. There are also links for 'free service by mysoho' and 'SoHo Market'.

마이페이지에서 나의 문의 내역으로 들어가면 해당 유저가 작성한 상품 문의를 볼 수 있음

해당 유저의 문의를 확인/수정/삭제 할 수 있음

등록된 상품문의가 없을 경우

The screenshot shows the 'My Product Inquiry' section of the NURIRI website. The page header includes 'My Product Inquiry' and 'My Q&A' tabs. The main content area displays a message: '작성한 상품문의가 없습니다.' (No product inquiries have been written).

공지사항

The image shows a mobile application interface. On the left, there is a navigation bar with icons for home, search, and account. Below the icons are category links: NURIRI SALE, TOP, BOTTOM, ONE-PIECE, and ACC. A large red arrow points from the bottom-left towards a red-bordered button at the bottom of the screen. This button contains the text "▶ 공지사항". To the right of the arrow is a red box containing the Korean text: "상단 ⚙을 누르고 공지사항 게시판으로 들어가기". The main content area displays a search results page for "공지사항". The results include a search bar with placeholder text "검색어를 입력해 주세요", a search icon, and a message: "공지[필독] 구매 전 꼭 확인해주세요." followed by "공지사항 | 2023.05.12".

상단 ⚙을 누르고 공지사항
게시판으로 들어가기

▶ 공지사항

NURIRI

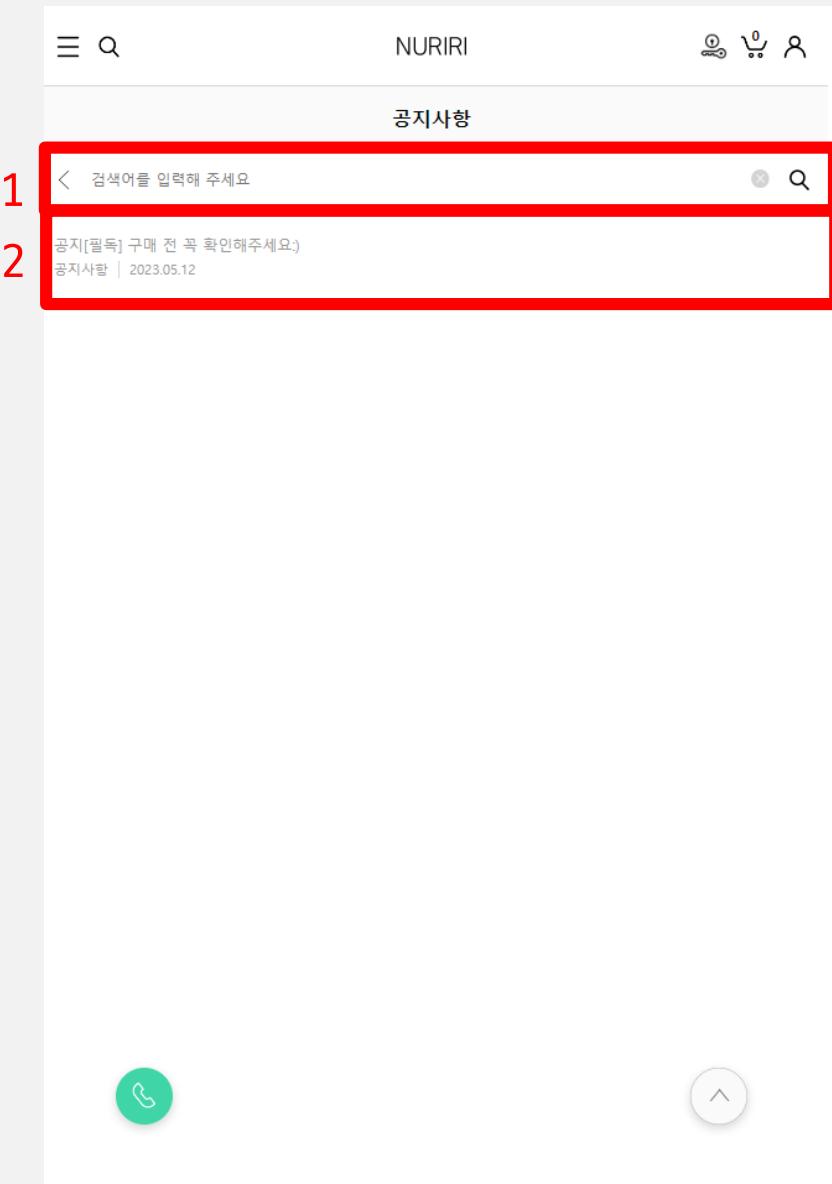
공지사항

< 검색어를 입력해 주세요

공지[필독] 구매 전 꼭 확인해주세요.)
공지사항 | 2023.05.12

88

공지사항



1. 검색창에 검색어를 입력해
공지사항 찾기
2. 공지사항 확인
공지사항 상세 페이지로 이동

검색 결과가 없을 경우



공지사항

The screenshot shows a mobile application interface. At the top, there is a navigation bar with icons for menu, search, and user profile. The main title is "공지사항". Below the title, there is a message in Korean: "공지[필독] 구매 전 꼭 확인해주세요.")". Underneath the message, there is a section titled "공지사항 내용" which contains the text "공지사항 내용" and a large green logo for "spring boot" featuring a power button icon inside a hexagon. At the bottom of the screen, there is a red button labeled "목록".

1

2

90

10:00~18:00(주말, 공휴일 휴무)

쇼핑몰 사업자정보 ▾

1. 공지사항 제목/카테고리/날짜/내용 확인
2. 공지사항 목록 페이지로 이동

관 리 자 페 이 지

관리자 페이지

1

The screenshot shows the 'User Page' interface. On the left, there is a sidebar menu with several items: 회원관리 (highlighted with a red box), 상품관리, 주문관리, 리뷰관리, 통합문의관리, 상품문의관리, and 공지사항관리. At the top right, there is a header with the text 'NURIRI' and a '로그아웃' (Logout) button. A red box highlights the 'NURIRI' button.

회원관리

회원등급 전체

검색조건 전체 검색

총 회원수 : 2

<input type="checkbox"/>	회원등급	이름	아이디	연락처	이메일	가입일자	상세정보
<input type="checkbox"/>	일반회원	정희진	testId2	01054245046	testId@gmail.com	2023-05-12	<button>수정</button>
<input type="checkbox"/>	일반회원	정희진	testId	01012341234	naahah79@naver.com	2023-05-12	<button>수정</button>

선택삭제

이전 1 다음

2

사용자 페이지에서 관리자가 Header에 있는 ☰를 눌렀을 때
관리자 페이지로 이동

1. 7개 카테고리

회원관리

주문관리

통합문의관리

공지사항관리

상품관리

리뷰관리

상품문의관리

nav에서 각 페이지로 이동 가능

2. NURIRI를 누르면 사용자 페이지로 이동

로그아웃

회원관리-목록

The screenshot shows the '회원관리' (Member Management) page. On the left, there is a sidebar with various management options: 회원관리 (1), 상품관리, 주문관리, 리뷰관리, 통합문의관리, 상품문의관리, and 공지사항관리.

The main area has a header '회원관리' and a sub-header '회원등급'. It includes two dropdown menus: '회원등급' (Grade) set to '전체' (All) and '검색조건' (Search Condition) also set to '전체' (All). A red box highlights this search section.

The table below shows member details:

번호	회원등급	이름	아이디	연락처	이메일	가입일자	상세정보
2	일반회원	정희진	testId2	01054245046	testId@gmail.com	2023-05-12	<button>수정</button> (3)
3	일반회원	정희진	testId	01012341234	naahah79@naver.com	2023-05-12	<button>수정</button>

A red box highlights the '수정' button in the third row. Below the table is a '선택삭제' (Select Delete) button.

At the bottom, there is a navigation bar with buttons for '이전' (Previous), '다음' (Next), and the page number '1'.

A large red box labeled '4' highlights the page number '1'.

1. 회원 검색

회원등급: 전체/VIP/일반회원
키워드 검색조건:
전체/이름/아이디

2. 회원 목록

3. 수정 페이지로 이동

4. 페이지 nav

회원관리-목록

검색



```
@Transactional(readOnly = true)
public Page<MemberResponseDto> findByKeyword(String findByType1, String findByType2, String keyword, int page) {

    Page<MemberEntity> list;
    Pageable pageable = PageRequest.of(page, 10);

    if (findByType1.equals("all") && findByType2.equals("all")) { //키워드만 입력했을 때
        list = memberRepository.findByKeyword(keyword, pageable);
    } else if (!findByType1.equals("all") && findByType2.equals("all")) { //회원등급 & 키워드가 입력했을 때
        list = memberRepository.findByType1(findByType1, keyword, pageable);
    } else if (findByType1.equals("all") && !findByType2.equals("all")) { //이름,아이디 & 키워드가 입력했을 때

        if (findByType2.equals("아이디")) {
            list = memberRepository.findById(keyword, pageable);
        } else { //((findByType2.equals("이름"))
            list = memberRepository.findByName(keyword, pageable);
        }
    } else{ //(!findByType1.equals("all") && !findByType2.equals("all"))
        if (findByType2.equals("이름")) {
            list = memberRepository.findByName(findByType1, keyword, pageable);
        } else{ //((findByType2.equals("아이디"))
            list = memberRepository.findById(findByType1, keyword, pageable);
        }
    }

    return list.map(MemberResponseDto::new);
}
```

MemberService.java

```
@GetMapping("/admin/member/list")
public String adminMemberList(@RequestParam(value = "findByType1", required = false) String findByType1,
                             @RequestParam(value = "findByType2", required = false) String findByType2,
                             @RequestParam(value = "keyword", required = false) String keyword,
                             @RequestParam(value = "page", defaultValue = "0") int page,
                             Model model) {

    Page<MemberResponseDto> list;
    int totalPage;
    List<Integer> pageList;

    if ((findByType1 == null) && (findByType2 == null) && (keyword == null)
        || (findByType1.equals("null")) && (findByType2.equals("null")) && (keyword.equals("null"))
        || (findByType1.equals("all")) && (findByType2.equals("all")) && (keyword.equals(""))){
        list = memberService.findAll(page);
    } else {
        list = memberService.findByKeyword(findByType1, findByType2, keyword, page);
    }
    totalPage = list.getTotalPages();
    pageList = noticeService.getPageList(totalPage, page);
    model.addAttribute("list", list);
    model.addAttribute("findByType1", findByType1);
    model.addAttribute("findByType2", findByType2);
    model.addAttribute("keyword", keyword);
    model.addAttribute("pageList", pageList);
    model.addAttribute("totalPage", totalPage);

    //검색 회원 명수
    long listCount = memberRepository.count();
    model.addAttribute("listCount", listCount);

    return "admin/member/list"; //listForm.html로 응답
}
```

AdminMemberController.java

회원관리-목록

목록 찾기

```
MemberService.java

@Transactional(readOnly = true)
public Page<MemberResponseDto> findAll(int page) {

    List<Sort.Order> sorts = new ArrayList<>();
    sorts.add(Sort.Order.desc("memberNo"));
    Pageable pageable = PageRequest.of(page, 10, Sort.by(sorts));
    Page<MemberEntity> list = memberRepository.findAll(pageable);

    return list.map(MemberResponseDto::new);
}
```

ID로 찾기

```
MemberService.java

public MemberResponseDto findById(long id) {
    MemberEntity entity = memberRepository.findById(id).orElseThrow(() →
        new IllegalArgumentException("해당 사용자가 없습니다."));
    MemberResponseDto dto = new MemberResponseDto(entity);
    return dto;
}
```

회원관리-목록

페이지

```
// 페이지 리스트 반환 (리스트 안 요소 개수는 항상 5)
public List<Integer> getPageList(final int totalPage, final int page) {

    List<Integer> pageList = new ArrayList<>();

    if (totalPage <= 5){ // 전체 페이지 개수가 5이하일 때,
        // ex. 전체 페이지가 3일 때, [0, 1, 2] 출력
        for (Integer i=0; i<=totalPage-1; i++){
            pageList.add(i);
        }
    } else if(page >= 0 && page <= 2){ // 전체 페이지 개수가 5초과이고, 현재 페이지가 0~2일 때,
        // ex. 현재 페이지가 1일 때, [0, 1, 2, 3, 4] 출력
        for (Integer i=0; i<=4; i++){
            pageList.add(i);
        }
    } else if (page >= totalPage-3 && page <= totalPage-1){ // 전체 페이지 개수가 5초과이고, 현재 페이지가 끝에서 0,1,2번째 일 때,
        // ex. 전체 페이지 개수가 7이고, 현재 페이지가 5일 때, [2, 3, 4, 5, 6] 출력
        for (Integer i=5; i>=1; i--){
            pageList.add(totalPage - i);
        }
    } else{ // 그외일 때, 앞,뒤로(-2 ~ +2) 계산한 값 출력
        for (Integer i=-2; i<=2; i++){
            pageList.add(page + i);
        }
    }

    return pageList;
}
```

NoticeService.java

회원관리-수정

관리자페이지

회원정보

NURIRI 로그아웃

회원번호	9
회원등급	일반회원
아이디	testId2
이름	정희진
연락처	01054245046
이메일	testId@gmail.com
적립금	0
권한	일반
가입일자	2023-05-12
우편번호	05088
주소	기본주소 서울 광진구 자양강변길 7
나머지주소	107동 108호

1

2

- 회원번호/회원등급/아이디/이름/연락처/이메일/적립금/권한/가입일자/주소 확인 가능
- 수정 버튼
- 목록 버튼: 목록 페이지로 이동

회원관리-수정

```
@PostMapping("/admin/member/content/modify")
@ResponseBody
public String adminMemberContentModify(MemberSaveRequestDto dto) {

    boolean result = memberService.modify(dto);

    if (result) {
        return "<script>alert('회원정보수정 완료'); location.href='/admin/member/list';</script>";
    } else {
        return "<script>alert('회원정보수정 실패'); history.back();</script>";
    }
}
```

AdminMemberController.java

수정

```
@Transactional
public boolean modify(MemberSaveRequestDto dto){

    MemberEntity entity = memberRepository.findById(dto.getMemberNo()).get();
    dto.setMemberJoinDatetime(entity.getMemberJoinDatetime());
    try {
        memberRepository.save(dto.toUpdateEntity());
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
    return true;
}
```

MemberService.java

상품관리-목록

The screenshot shows the '상품목록' (Product List) page. On the left is a sidebar with navigation links: 회원관리, 상품관리 (highlighted with a red box), 주문관리, 리뷰관리, 통합문의관리, 상품문의관리, and 공지사항관리. The main area has a header '상품목록' and 'NURIRI 로그아웃'. The search bar at the top has '상품분류' (Category) set to '대분류' and '검색조건' (Search Condition) set to '전체'. Below the search bar is a table with 5 products:

상품코드	이미지	상품명	색상	사이즈	가격	노출상태	판매상태	상세정보
20000		퍼프 플라우스 화이트	화이트	FREE	49000원	노출함	판매중	저장 수정 삭제
20001		퍼프 플라우스 퍼플	퍼플	FREE	49000원	노출함	판매중	저장 수정 삭제
20002		브라운 사각 bag	브라운	FREE	32000원	노출함	판매중	저장 수정 삭제
20003		린넨 ops 베이지	베이지	S,M,L	49000원	노출함	판매중	저장 수정 삭제
20004		도트 sk	베이지,오렌지	FREE	42000원	노출함	판매중	저장 수정 삭제

At the bottom of the table are buttons for '선택삭제' (Select Delete) and '선택수정' (Select Modify). The page footer includes '이전 1 다음' (Previous 1 Next) and a large red box labeled '선택삭제'.

1. 상품 검색

상품분류: 대분류/TOP/BOTTOM/
ONE-PIECE/ACC

키워드 검색조건:
전체/상품명/
상품코드/상품가격

2. 등록 페이지로 이동

3. 노출상태/판매상태 저장
수정 페이지로 이동
상품 삭제

4. 선택 삭제

5. 페이지 nav

상품관리-목록

검색 기능

```
AdminProductController.java
@.GetMapping("/admin/product/list")
public String productList(Model model,
    @RequestParam(value = "findByType1", required = false) String findByType1,
    @RequestParam(value = "findByType2", required = false) String findByType2,
    @RequestParam(value = "keyword", required = false) String keyword,
    @RequestParam(value = "page", defaultValue = "0") int page) {

    Page<ProductResponseDto> list;
    int totalPage;
    List<Integer> pageList;

    if ((findByType1 == null) && (findByType2 == null) && (keyword == null)
        || (findByType1.equals("null")) && (findByType2.equals("null")) && (keyword.equals("null"))
        || (findByType1.equals("all")) && (findByType2.equals("all")) && (keyword.equals("")))) {
        list = productService.findAll(page);
    } else {
        list = productService.findByKeyword(findByType1, findByType2, keyword, page);
    }
    totalPage = list.getTotalPages();
    pageList = noticeService.getPageList(totalPage, page);
    model.addAttribute("list", list);
    model.addAttribute("findByType1", findByType1);
    model.addAttribute("findByType2", findByType2);
    model.addAttribute("keyword", keyword);
    model.addAttribute("totalPage", totalPage);
    model.addAttribute("pageList", pageList);
    //상품 개수
    long listCount = productRepository.count();
    model.addAttribute("listCount", listCount);

    return "admin/product/list";
}
```

```
ProductService.java
@Transactional(readOnly = true)
public Page<ProductResponseDto> findByKeyword(String findByType1, String findByType2, String keyword, int page) {

    Page<ProductEntity> list;
    Pageable pageable = PageRequest.of(page, 10);

    if (findByType1.equals("all") && findByType2.equals("all")) {
        list = productRepository.findByKeyword(keyword, pageable);
    } else if (!findByType1.equals("all") && findByType2.equals("all")) {
        list = productRepository.findByType1(findByType1, keyword, pageable);
    } else if (findByType1.equals("all") && !findByType2.equals("all")) {

        if (findByType2.equals("상품명")) {
            list = productRepository.findByName(keyword, pageable);
        } else if (findByType2.equals("상품번호")) {
            list = productRepository.findByItemNo(keyword, pageable);
        } else { //상품가격
            list = productRepository.findByItemPrice(keyword, pageable);
        }

    } else{ //(!findByType1.equals("all") && !findByType2.equals("all"))
        if (findByType2.equals("상품명")) {
            list = productRepository.findByName(findByType1, keyword, pageable);
        } else if (findByType2.equals("상품코드")) {
            list = productRepository.findByItemNo(findByType1, keyword, pageable);
        } else { //상품가격
            list = productRepository.findByItemPrice(findByType1, keyword, pageable);
        }
    }

    return list.map(ProductResponseDto::new);
}
```

상품관리-목록

선택 삭제



admin/product/list.html

```
function deleteValue(){

    var itemNo = "";
    $("input[name='RowCheck']:checked").each(function (){
        itemNo = itemNo + $(this).val() + ",";
    });
    if(itemNo == ""){
        alert("삭제할 대상을 선택하세요.");
        return false;
    }
    $.ajax({
        url : '/admin/product/delete/check',
        type : 'POST',
        traditional : true,
        data : {
            itemNo : itemNo,
            [[${_csrf.parameterName}]] : [[${_csrf.token}]]
        },
        success: function(jdata){
            if(jdata = 1){
                alert("삭제 완료");
                location.replace("/admin/product")
            }
            else{
                alert("삭제 실패")
            }
        }
    });
}
```



ProductService.java

```
@Transactional
public boolean productDeleteCheck(String arrStr) {
    try {
        String[] arrIdx = arrStr.split(",");
        for (int i=0; i<arrIdx.length; i++) {
            ProductEntity productEntity = productRepository.findById((long)
Integer.parseInt(arrIdx[i])).get();
            productRepository.delete(productEntity);
        }
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
    return true;
}
```

상품관리-등록

관리자페이지 상품정보 등록 NURIRI 로그아웃

회원관리
상품관리 1
주문관리
리뷰관리
통합문의관리
상품문의관리 2
공지사항관리

상품정보
판매가 10000
색상 color
사이즈 S,M,L
노출상태 노출함
판매상태 판매중

 할인정보
할인율 10
 10% 퍼센트

1. 상품분류/상품명 설정
2. 대표 이미지 설정
AWS S3에 이미지 저장
3. 판매가/색상/사이즈
노출상태/판매상태 설정

상품관리-등록

4. 할인율 설정

5. 상품 설명
Ckeditor5로 구현

6. Input 값 넣은 후 누르면 등록

등록하기

판매가	10000
색상	color
사이즈	S,M,L
노출상태	노출함
판매상태	판매중

할인율	10 ●% 퍼센트
-----	--------------

상품 설명	<p>Paragraph <input type="button" value="▼"/> <input checked="" type="button" value="B"/> <input type="button" value="I"/> <input type="button" value="O"/> <input type="button" value="H1"/> <input type="button" value="H2"/> <input type="button" value="H3"/> <input type="button" value="H4"/> <input type="button" value="H5"/> <input type="button" value="H6"/> <input type="button" value="H7"/> <input type="button" value="H8"/> <input type="button" value="H9"/> <input type="button" value="H10"/> <input type="button" value="H11"/> <input type="button" value="H12"/> <input type="button" value="H13"/> <input type="button" value="H14"/> <input type="button" value="H15"/> <input type="button" value="H16"/> <input type="button" value="H17"/> <input type="button" value="H18"/> <input type="button" value="H19"/> <input type="button" value="H20"/> <input type="button" value="H21"/> <input type="button" value="H22"/> <input type="button" value="H23"/> <input type="button" value="H24"/> <input type="button" value="H25"/> <input type="button" value="H26"/> <input type="button" value="H27"/> <input type="button" value="H28"/> <input type="button" value="H29"/> <input type="button" value="H30"/> <input type="button" value="H31"/> <input type="button" value="H32"/> <input type="button" value="H33"/> <input type="button" value="H34"/> <input type="button" value="H35"/> <input type="button" value="H36"/> <input type="button" value="H37"/> <input type="button" value="H38"/> <input type="button" value="H39"/> <input type="button" value="H40"/> <input type="button" value="H41"/> <input type="button" value="H42"/> <input type="button" value="H43"/> <input type="button" value="H44"/> <input type="button" value="H45"/> <input type="button" value="H46"/> <input type="button" value="H47"/> <input type="button" value="H48"/> <input type="button" value="H49"/> <input type="button" value="H50"/> <input type="button" value="H51"/> <input type="button" value="H52"/> <input type="button" value="H53"/> <input type="button" value="H54"/> <input type="button" value="H55"/> <input type="button" value="H56"/> <input type="button" value="H57"/> <input type="button" value="H58"/> <input type="button" value="H59"/> <input type="button" value="H60"/> <input type="button" value="H61"/> <input type="button" value="H62"/> <input type="button" value="H63"/> <input type="button" value="H64"/> <input type="button" value="H65"/> <input type="button" value="H66"/> <input type="button" value="H67"/> <input type="button" value="H68"/> <input type="button" value="H69"/> <input type="button" value="H70"/> <input type="button" value="H71"/> <input type="button" value="H72"/> <input type="button" value="H73"/> <input type="button" value="H74"/> <input type="button" value="H75"/> <input type="button" value="H76"/> <input type="button" value="H77"/> <input type="button" value="H78"/> <input type="button" value="H79"/> <input type="button" value="H80"/> <input type="button" value="H81"/> <input type="button" value="H82"/> <input type="button" value="H83"/> <input type="button" value="H84"/> <input type="button" value="H85"/> <input type="button" value="H86"/> <input type="button" value="H87"/> <input type="button" value="H88"/> <input type="button" value="H89"/> <input type="button" value="H90"/> <input type="button" value="H91"/> <input type="button" value="H92"/> <input type="button" value="H93"/> <input type="button" value="H94"/> <input type="button" value="H95"/> <input type="button" value="H96"/> <input type="button" value="H97"/> <input type="button" value="H98"/> <input type="button" value="H99"/> <input type="button" value="H100"/></p>
-------	---

6

상품관리-등록

등록

```
@Transactional  
public boolean productRegistration(ProductSaveRequestDto dto){  
    try {  
        dto.setItemUpdateDatetime(LocalDateTime.now());  
        ProductEntity entity = dto.toSaveEntity();  
        productRepository.save(entity);  
    } catch (Exception e) {  
        e.printStackTrace();  
        return false;  
    }  
    return true;  
}
```

ProductService.java

이미지 업로드

```
@ResponseBody  
@RequestMapping("/find/admin/product/registration/action")  
public String productRegistrationAction(@RequestParam MultipartFile uploadfile, ProductSaveRequestDto dto)  
throws IllegalStateException, IOException {  
  
    String url = awsS3Service.upload(uploadfile);  
  
    new ResponseEntity<>(FileResponse.builder()  
        .uploaded(true).  
        url(url).  
        build(), HttpStatus.OK);  
  
    dto.setItemImageUrl(url);  
  
    boolean result = productService.productRegistration(dto);  
    if (!result) {  
        return "<script>alert('등록 실패');location.href='/admin/product/list/';</script>";  
    }  
    return "<script>alert('등록 완료');location.href='/admin/product/list/';</script>";  
}
```

AdminProductController.java

대표 이미지

파일 선택 선택된 파일 없음

상품관리-등록

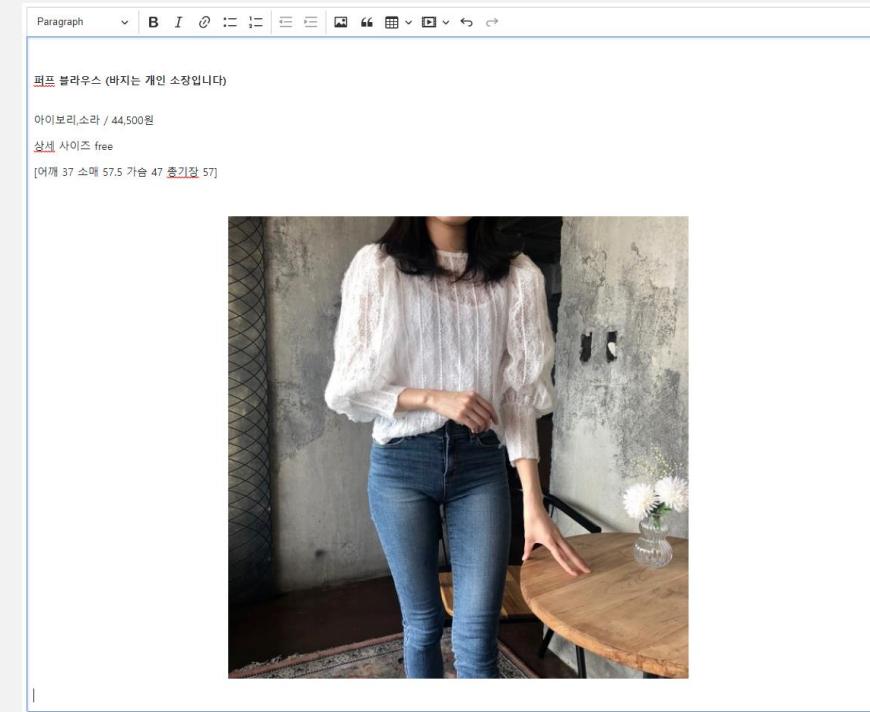
ckeditor5

```
● ● ● admin/product/registration.html

<script src="/js/ckeditor.js"> </script>
<script>
    let editor;

    ClassicEditor
        .create(document.querySelector('#editor'),{
            language: 'ko',
            ckfinder: {
                uploadUrl: '/find/admin/imgUpload'
            }
        })
        .then( newEditor => {
            editor = newEditor;
        })
        .catch( error => {
            console.error(error);
        });
    
```

```
function function1(){
    const editorData = editor.getData();
    document.querySelector('#editorData').value = editorData;
}
</script>
```



상품관리-등록

AwsS3Service.java

```

public String upload(MultipartFile file) {
    String fileName = createFileName(file.getOriginalFilename());
    ObjectMetadata objectMetadata = new ObjectMetadata();
    objectMetadata.setContentLength(file.getSize());
    objectMetadata.setContentType(file.getContentType());

    try(InputStream inputStream = file.getInputStream()) {
        amazonS3.putObject(new PutObjectRequest(bucket, fileName, inputStream, objectMetadata)
            .withCannedAcl(CannedAccessControlList.PublicRead));
    } catch(IOException e) {
        throw new ResponseStatusException(HttpStatus.INTERNAL_SERVER_ERROR, "파일 업로드에 실패했습니다.");
    }

    return defaultUrl + fileName;
}

```

AwsS3Config.java

```

@Configuration
public class AwsS3Config {

    @Value("${cloud.aws.credentials.access-key}")
    private String accessKey;

    @Value("${cloud.aws.credentials.secret-key}")
    private String secretKey;

    @Value("${cloud.aws.region.static}")
    private String region;

    @Bean
    public AmazonS3Client amazonS3Client() {
        BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);
        return (AmazonS3Client) AmazonS3ClientBuilder.standard()
            .withRegion(region)
            .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
            .build();
    }
}

```

AwsS3Service.java

```

private String createFileName(String fileName) {
    return UUID.randomUUID().toString().concat(getFileExtension(fileName));
}

private String getFileExtension(String fileName) {
    try {
        return fileName.substring(fileName.lastIndexOf("."));
    } catch (StringIndexOutOfBoundsException e) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "잘못된 형식의 파일(" + fileName + ") 입니다.");
    }
}

```

application.yml

```

cloud:
  aws:
    credentials:
      access-key: // 입력
      secret-key: // 입력
    region:
      static: ap-northeast-2
    s3:
      bucket:
        url: https://s3-doreen-bucket.s3.ap-northeast-2.amazonaws.com/
        name: s3-doreen-bucket
    stack:
      auto: false

```

상품관리-수정

관리자페이지

상품정보 수정

NURIRI 로그아웃

이미지

대표 이미지 변경 [파일 선택] 선택된 파일 없음

1

상품정보

상품분류	TOP
상품명	퍼프 블라우스 화이트
판매가	49000
색상	화이트
사이즈	FREE
노출상태	노출함
판매상태	판매 중

2

할인정보

할인율	9 ● 0% 퍼센트
-----	---------------

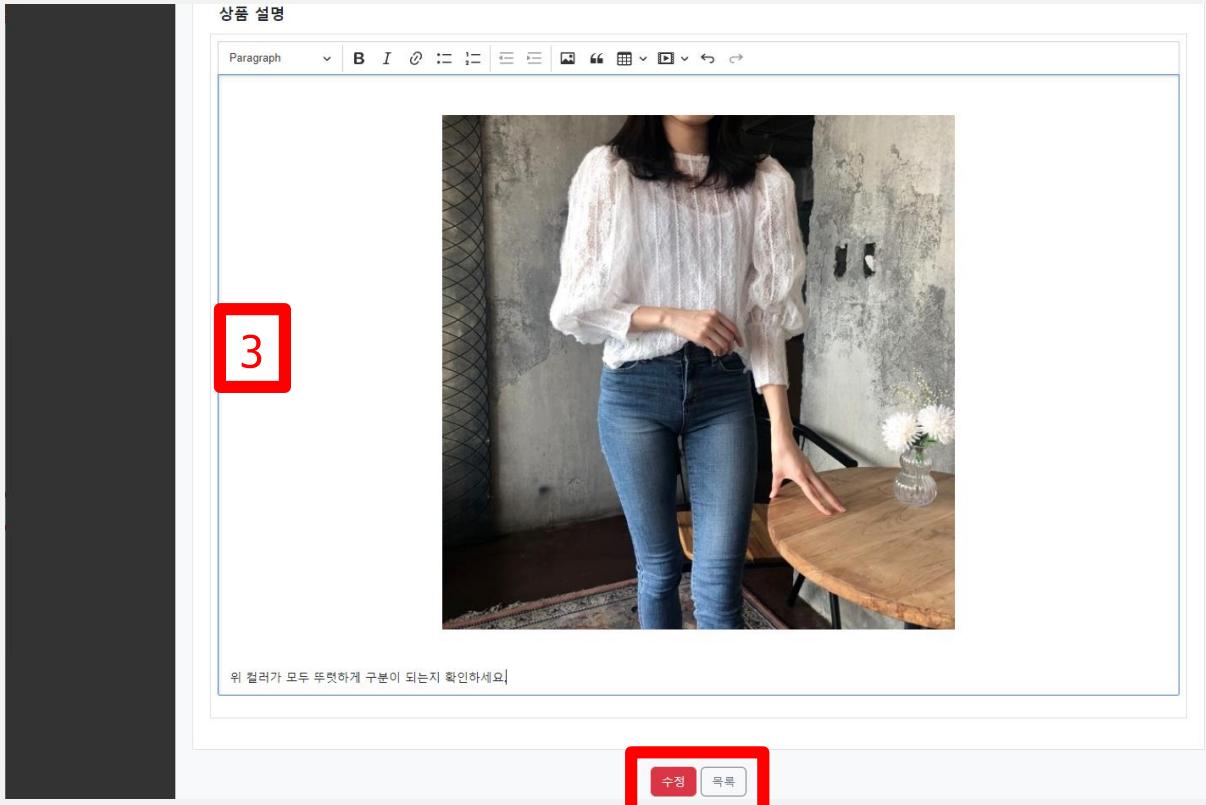
1. 대표이미지 변경

AWS S3에 이미지 저장

만약 이미지변경을 하지 않을 경우
기존 이미지로 유지

2. 상품분류/상품명/판매가/ 색상/사이즈/노출상태/ 판매상태/할인율 확인 가능

상품관리-수정



4

3. CKeditor5로 게시판 구현
4. 수정 버튼: input 수정 후 누르면 수정
목록 버튼: 목록 페이지로 이동

상품관리-수정

AdminProductController.java

```

@RequestBody
@RequestMapping("/find/admin/product/modify/action")
public String productModify(@RequestParam MultipartFile uploadfile, ProductSaveRequestDto dto) {
    String url;
    if (uploadfile.getOriginalFilename().equals("")) {
        url = productService.findByUrl(dto.getItemNo());
    } else {
        url = awss3Service.upload(uploadfile);
        new ResponseEntity<>(FileResponse.builder()
            .uploaded(true).
            .url(url).
            .build(), HttpStatus.OK);
    }
    dto.setItemImageUrl(url);

    boolean result = productService.productModify(dto);
    if (!result) {
        return "<script>alert('수정 실패');location.href='/admin/product/list/';</script>";
    }
    return "<script>alert('수정 완료');location.href='/admin/product/list/';</script>";
}

```

AwsS3Config.java

```

@Configuration
public class AwsS3Config {

    @Value("${cloud.aws.credentials.access-key}")
    private String accessKey;

    @Value("${cloud.aws.credentials.secret-key}")
    private String secretKey;

    @Value("${cloud.aws.region.static}")
    private String region;

    @Bean
    public AmazonS3Client amazonS3Client() {
        BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);
        return (AmazonS3Client) AmazonS3ClientBuilder.standard()
            .withRegion(region)
            .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
            .build();
    }
}

```

AwsS3Service.java

```

private String createFileName(String fileName) {
    return UUID.randomUUID().toString().concat(getFileExtension(fileName));
}

private String getFileExtension(String fileName) {
    try {
        return fileName.substring(fileName.lastIndexOf("."));
    } catch (StringIndexOutOfBoundsException e) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "잘못된 형식의 파일(" + fileName + ") 입니다.");
    }
}

```

application.yml

```

cloud:
  aws:
    credentials:
      access-key: // 입력
      secret-key: // 입력
    region:
      static: ap-northeast-2
    s3:
      bucket:
        url: https://s3-doreen-bucket.s3.ap-northeast-2.amazonaws.com/
        name: s3-doreen-bucket
      stack:
        auto: false

```

주문관리-목록

주문일	주문번호	주문자명	결제방법	결제금액	주문상태	상세정보
2023-05-08 15:47	202305081547325578	박선교	휴대폰결제	44,500원	배송대기	<button>주문상태 변경</button> <button>수정</button>
2023-04-04 18:25	10000	종길동	무통장입금	35,000원	배송완료	<button>주문상태 변경</button> <button>수정</button>
2023-04-04 18:25	10001	둘리	휴대폰결제	35,000원	배송완료	<button>주문상태 변경</button> 수정
2023-04-04 18:25	10002	박선교	무통장입금	35,000원	배송완료	<button>주문상태 변경</button> <button>수정</button>
2023-04-04 18:25	10003	박선교	무통장입금	35,000원	배송완료	<button>주문상태 변경</button> <button>수정</button>
2023-04-04 18:25	10004	박선교	무통장입금	35,000원	배송완료	<button>주문상태 변경</button> <button>수정</button>
2023-04-04 18:25	10005	박선교	무통장입금	35,000원	배송완료	<button>주문상태 변경</button> <button>수정</button>

1. 상품 검색

주문상태: 전체/결제대기/배송대기/배송중/배송완료/취소/반품/교환
기간: 날짜 선택/오늘/어제/1주일/1개월

키워드 검색조건:

전체/주문자명/주문번호/전화번호

2. 주문상태 변경

3. 수정 페이지로 이동

4. 페이지 nav

주문관리-목록

검색 기능



AdminOrderController.java

```
@GetMapping("/admin/order/list")
public String orderList(Model model,
    @RequestParam(value = "findBy", required = false) String findBy,
    @RequestParam(value = "keyword", required = false) String keyword,
    @RequestParam(value = "dateStart", required = false) String dateStart,
    @RequestParam(value = "dateEnd", required = false) String dateEnd,
    @RequestParam(value = "page", defaultValue = "0") int page) throws ParseException {
    Page<OrderResponseDto> list;
    int totalPage;
    List<Integer> pageList;
    if ((findBy == null) && (keyword == null) && (dateStart == null) && (dateEnd == null)
        || (dateStart.equals("null")) && (dateEnd.equals("null")) && (keyword.equals("null"))
        || (dateStart.equals("")) && (dateEnd.equals("")) && (keyword.equals(""))){
        list = orderService.findAll(page);
    } else {
        //오늘, 어제, 1주일, 1개월 검색
        if ((!dateStart.equals("")) && (dateEnd.equals(""))){
            list = orderService.findByDate(dateStart, page);
        }
        //기간 검색
    } else if ((!dateStart.equals("")) && (!dateEnd.equals(""))){
        list = orderService.findByDate(dateStart, dateEnd, page);
    } else {
        //검색 키워드가 있을 때
        list = orderService.findByKeyword(findBy, keyword, page);
    }
    totalPage = list.getTotalPages();
    pageList = noticeService.getPageList(totalPage, page);
    model.addAttribute("pagelist", pageList);
    model.addAttribute("list", list);
    model.addAttribute("findBy", findBy);
    model.addAttribute("keyword", keyword);
    model.addAttribute("dateStart", dateStart);
    model.addAttribute("dateEnd", dateEnd);
    //검색 상품 개수
    long listCount = orderRepository.count();
    model.addAttribute("listCount", listCount);
    return "admin/order/list";
}
```



OrderService.java

```
@Transactional(readOnly = true)
public Page<OrderResponseDto> findByDate(String start, String end, int page) throws
ParseException {
    Page<OrderEntity> list;
    Pageable pageable = PageRequest.of(page, 10);

    String[] date = dateSetting(start, end);
    String dateStartStr = date[0];
    String dateEndStr = date[1];

    list = orderRepository.findByOrderNoContaining(dateStartStr, dateEndStr, pageable);
    return list.map(OrderResponseDto :: new);
}
```



OrderService.java

```
@Transactional(readOnly = true)
public String[] dateSetting(String start, String end) throws ParseException {
    //문자열을 날짜형식으로 변환
    DateFormat sdFormatStart = new SimpleDateFormat("yyyy-MM-dd");
    DateFormat sdFormatEnd = new SimpleDateFormat("yyyy-MM-dd");
    Date tempDateStart = sdFormatStart.parse(start);
    Date tempDateEnd = sdFormatEnd.parse(end);
    Calendar cal1 = Calendar.getInstance();
    Calendar cal2 = Calendar.getInstance();
    cal1.setTime(tempDateStart);
    cal2.setTime(tempDateEnd);
    DateFormat df1 = new SimpleDateFormat("yyyy-MM-dd");
    DateFormat df2 = new SimpleDateFormat("yyyy-MM-dd");
    cal2.add(Calendar.DATE, +1);

    //날짜형식을 문자열로 변환
    String dateStartStr = df1.format(cal1.getTime())+" 00:00:00";
    String dateEndStr = df2.format(cal2.getTime())+" 00:00:00";

    String[] date = {dateStartStr, dateEndStr};
    return date;
}
```

주문관리-목록

주문상태/키워드 검색

```
●●●  
@Transactional(readOnly = true)  
public Page<OrderResponseDto> findByKeyword(String findBy, String keyword, int page) {  
  
    Page<OrderEntity> list;  
    Pageable pageable = PageRequest.of(page, 10);  
  
    if (findBy.equals("all")) {  
        list = orderRepository.findAllContaining(keyword, pageable);  
    } else if (findBy.equals("orderName")){  
        list = orderRepository.findByOrderNameContaining(keyword, pageable);  
    } else if (findBy.equals("orderPhone")) {  
        list = orderRepository.findByOrderPhoneContaining(keyword, pageable);  
    }else if (findBy.equals("orderNo")) {  
        list = orderRepository.findByOrderNoContaining(keyword, pageable);  
    } else {  
        //주문상태 검색  
        list = orderRepository.findByOrderState(keyword, pageable);  
    }  
  
    return list.map(OrderResponseDto::new);  
}
```

OrderService.java

날짜 검색

```
●●●  
public Page<OrderResponseDto> findByDate(String mode, int page) throws ParseException {  
    String[] date = dateSetting(mode);  
    String dateStartStr = date[0];  
    String dateEndStr = date[1];  
    return findByDate(dateStartStr, dateEndStr, page);  
}
```

OrderService.java

주문상태 변경

```
●●●  
@ResponseBody  
 @RequestMapping("/admin/order/status/modify")  
public String orderStatusModify(OrderContentSaveRequestDto dto) {  
    boolean result = orderService.statusModify(dto.getOrderNo(),dto.getOrderState());  
    if (!result) {  
        return "<script>alert('주문상태 변경 실패');location.href='/admin/order/list/'</script>";  
    }  
    return "<script>alert('주문상태 변경 완료');location.href='/admin/order/list/'</script>";  
}
```

AdminController.java

```
●●●  
@Transactional  
public boolean statusModify(Long id, String orderState) {  
    try {  
        OrderResponseDto dto = findById(id);  
        dto.setOrderState(orderState);  
        orderRepository.save(dto.toEntity());  
    } catch (Exception e) {  
        e.printStackTrace();  
        return false;  
    }  
    return true;  
}
```

OrderService.java

주문관리-상세

관리자페이지 주문관리 NURIRI 로그아웃

1 주문자 / 수령자 정보

주문번호	202305081547325578
주문자명	박선교
주문자 연락처	01040246575
수령자명	박선교
수령자 연락처	01040246575
수령자 우편번호	03672
수령자 기본주소	서울 서대문구 명지대5길 5
수령자 나머지주소	102

2 구매 상품 정보

상품명	퍼프 블라우스 퍼플
색상	퍼플
사이즈	FREE
수량	1
상품금액	44,500원

1. 주문자/수령자 정보
고객요청이 왔을 때 수령자 정보 및 배송지 수정만 가능하도록 설계
2. 구매 상품 정보: 상품명/색상/사이즈 /수량/상품금액

주문관리-상세

상품명	퍼프 블라우스 퍼플
색상	퍼플
사이즈	FREE
수량	1
상품금액	44,500원

결제 정보	
적립금	0 원
쿠폰	0 원
결제방식	휴대폰결제
주문 총수량	1
결제금액	44,500원
주문일시	2023-05-08 15:47

주문상태	
주문상태	배송대기

수정 목록

3. 결제 정보: 적립금/쿠폰/결제방식/
주문 총수량/결제금액
/주문일시

4. 주문 상태: 결제대기/배송대기/
배송중/배송완료/취소
/반품/교환

5. 수정

주문관리-상세

주문정보 수정

```
AdminOrderController.java
```

```
@ResponseBody  
@RequestMapping("/admin/order/content/action")  
public String orderContentAction(OrderContentSaveRequestDto dto) {  
  
    boolean result = orderService.updateOrderContent(dto);  
    if (!result) {  
        return "<script>alert('수정 실패');location.href='/admin/order/list/'</script>";  
    }  
    return "<script>alert('수정 완료');location.href='/admin/order/list/'</script>";  
}
```

```
OrderService.java
```

```
@Transactional  
public boolean updateOrderContent(OrderContentSaveRequestDto dto) {  
    try {  
        OrderEntity entity = orderRepository.findById(dto.getOrderNo()).get();  
        OrderResponseDto responseDto = new OrderResponseDto(entity);  
        responseDto.setOrderRecipientName(dto.getOrderRecipientName());  
        responseDto.setOrderRecipientPhone(dto.getOrderRecipientPhone());  
        responseDto.setOrderRecipientAddrNumber(dto.getOrderRecipientAddrNumber());  
        responseDto.setOrderRecipientAddr1(dto.getOrderRecipientAddr1());  
        responseDto.setOrderRecipientAddr2(dto.getOrderRecipientAddr2());  
        responseDto.setOrderState(dto.getOrderState());  
        orderRepository.save(responseDto.toEntity());  
    } catch (Exception e) {  
        e.printStackTrace();  
        return false;  
    }  
    return true;  
}
```

리뷰관리-목록

1

2

3

4

상품번호	상품명	후기종류	아이디	내용	별점	작성일	노출여부	관리	
20005	린넨 ops 화이트	일반후기	right	사진이 없어요3	3	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20005	린넨 ops 화이트	일반후기	right	사진이 없어요2	3	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20005	린넨 ops 화이트	일반후기	right	사진이 없어요	3	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20005	린넨 ops 화이트	사진후기	right	코딩개어려워요	4	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20005	린넨 ops 화이트	사진후기	hong	가성비 좋아요2	5	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20007	아크네디님 연청	사진후기	right	옷이 예뻐요	4	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20006	골지원피스	사진후기	lee	배송이 빨리요	4	2023-04-04 18:25	노출함	노출상태 변경	상세정보
20005	린넨 ops 화이트	사진후기	hong	가성비 좋아요	5	2023-04-04 18:25	노출함	노출상태 변경	상세정보

1. 상품 검색

기간: 날짜 선택/오늘/어제/1주일
/1개월

키워드 검색조건:
아이디/ 상품코드

2. 노출상태 변경

3. 상세 페이지로 이동

4. 페이지 nav

리뷰관리-목록

검색 기능

```
●●● AdminReviewController.java

@RequestMapping("/admin/review/list")
public String listForm(Model model,
    @RequestParam(value = "dateStart", required = false) String dateStart,
    @RequestParam(value = "dateEnd", required = false) String dateEnd,
    @RequestParam(value = "page", defaultValue = "0") int page,
    @RequestParam(value = "keyword", required = false) String keyword,
    @RequestParam(value = "findBy", required = false) String findBy, throws ParseException {

    Page<ReviewResponseDto> list;
    List<Integer> pageList;

    if ((findBy == null) && (keyword == null) && (dateStart == null) && (dateEnd == null)
        || (dateStart.equals("null")) && (dateEnd.equals("null")) && (keyword.equals("null"))
        || (dateStart.equals("")) && (dateEnd.equals("")) && (keyword.equals(""))){
        //페이지된 리스트 가져오기
        list = reviewService.getPageList(page);
    } else {
        if (!dateStart.equals("")) && (dateEnd.equals("")) { //dateStart 값만 있을때
            list = reviewService.findByDate(dateStart, page);
        } else if (!dateStart.equals("")) && (!dateEnd.equals("")) { //
            list = reviewService.findByDate(dateStart, dateEnd, page);
        } else {
            list = reviewService.findByKeyword(findBy, keyword, page);
        }
    }

    pageList = reviewService.getPageList(list.getTotalPages(), page);

    List<String> itemList = new ArrayList<>();
    for (ReviewResponseDto dto : list) {
        ProductResponseDto itemDto = productService.findById(Long.parseLong(dto.getItemNo()));
        itemList.add(itemDto.getItemName()); //item이름을 itemList에 넣어줌
    }

    model.addAttribute("itemList", itemList);
    model.addAttribute("list", list);
    model.addAttribute("findBy", findBy);
    model.addAttribute("keyword", keyword);
    model.addAttribute("pageList", pageList);
    model.addAttribute("dateStart", dateStart);
    model.addAttribute("dateEnd", dateEnd);
    model.addAttribute("listCount", reviewRepository.count());

    return "admin/review/list";
}
```

ReviewService.java

```
●●● ReviewService.java

public Page<ReviewResponseDto> findByDate(String mode, int page) throws ParseException {
    String[] date = orderService.dateSetting(mode);
    String dateStartStr = date[0];
    String dateEndStr = date[1];
    return findByDate(dateStartStr, dateEndStr, page);
}
```

ReviewService.java

```
●●● ReviewService.java

@Transactional(readOnly = true)
public Page<ReviewResponseDto> findByDate(String start, String end, int page) throws ParseException {
    //기간 검색
    Page<ReviewEntity> list;
    Pageable pageable = PageRequest.of(page, 10);

    String[] date = orderService.dateSetting(start, end);
    String dateStartStr = date[0];
    String dateEndStr = date[1];

    list = reviewRepository.findByReviewNoContaining(dateStartStr, dateEndStr, pageable);
    return list.map(ReviewResponseDto::new);
}
```

ReviewService.java

```
●●● ReviewService.java

@Transactional(readOnly = true)
public Page<ReviewResponseDto> findByKeyword(String findBy, String keyword, int page){
    Page<ReviewEntity> list;
    List<Sort.Order> sorts = new ArrayList<>();
    sorts.add(Sort.Order.desc("reviewNo"));
    Pageable pageable = PageRequest.of(page, 10, Sort.by(sorts));
    if(findBy.contains("memberId")){
        list = reviewRepository.findByMemberIdContaining(keyword, pageable);
    } else {list = reviewRepository.findByItemNoContaining(keyword, pageable);}
    return list.map(ReviewResponseDto::new);
}
```

리뷰관리-목록

노출상태 변경

```
AdminReviewController.java

@ResponseBody
@RequestMapping("admin/review/status/modify")
public String reviewStatusModify(ReviewSaveResponseDto dto) {
    boolean result = reviewService.statusModify(dto.getReviewNo(), dto.getReviewExpo());
    if (!result) {
        return "<script>alert('노출상태 변경 실패');location.href='/admin/review/list/'</script>";
    }
    return "<script>alert('노출상태 변경 완료');location.href='/admin/review/list/'</script>";
}
```

```
ReviewService.java

@Transactional
public boolean statusModify(Long id, String reviewState) {
    try {
        ReviewEntity entity = findById(id);
        ReviewResponseDto dto = new ReviewResponseDto(entity);
        dto.setReviewExpo(reviewState);
        reviewRepository.save(dto.toEntity());
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
    return true;
}
```

리뷰관리-상세

관리자페이지

회원관리 상품관리 주문관리 리뷰관리 통합문의관리 상품문의관리 공지사항관리

리뷰관리

작성자 right
상품번호 20007
평점 4
작성일자 2023-04-04 18:25

리뷰내용
웃이 예뻐요

리뷰 이미지



1 2

1. 리뷰 내용
작성자/상품번호/평점/작성일자/
리뷰내용
2. 리뷰 이미지 여부에 따라
일반후기와 사진후기로 분류됨

일반후기

리뷰내용

작성자	right
상품번호	20005
평점	3
작성일자	2023-04-04 18:25
리뷰내용	사진이 없어요3
리뷰 이미지	

목록

통합문의관리-목록

The screenshot shows the '통합문의관리' (Unified Document Management) page. On the left, there is a sidebar with various management links. The main area has a header '통합문의관리' and a search bar with filters for '문의유형' (Document Type), '기간' (Period), and '검색조건' (Search Condition). Below the search bar is a table titled '전체 문의 내역 : 4' (Total Document Record : 4) showing four entries. Each entry includes columns for '번호' (Number), '문의유형' (Document Type), '글내용' (Content), '작성자' (Writer), '작성일' (Date), and a '상세정보' (Detailed Information) button. At the bottom of the table are navigation buttons '이전' (Previous), '1' (Current page), and '다음' (Next). A red box labeled '1' highlights the search filters, another labeled '2' highlights the '상세정보' button for the second document, and a third labeled '3' highlights the page navigation buttons.

번호	문의유형	글내용	작성자	작성일	상세정보
4	상품문의	안녕하세요.	닉네임	2023-04-04 18:25	[상세정보]
3	상품문의	안녕하세요.	닉네임	2023-04-04 18:25	[상세정보]
2	상품문의	안녕하세요.	정회진	2023-04-04 18:25	[상세정보]
1	상품문의	안녕하세요.	정회진	2023-04-04 18:25	[상세정보]

1. 통합 문의 검색

문의유형: 전체/상품문의/배송문
의/교환문의/반품문의/환불
문의/기타문의

기간: 날짜 선택/오늘/어제/1주일
/1개월

키워드 검색조건:
내용/작성자

2. 통합문의 상세 페이지로 이동

3. 페이지 nav

통합문의관리-목록

검색 기능

```
AdminQnaController.java

@GetMapping("admin/qna/list")
public String adminQnaList(@RequestParam(value = "keywordType", required = false) String keywordType,
    @RequestParam(value = "keyword", required = false) String keyword,
    @RequestParam(value = "dateStart", required = false) String dateStart,
    @RequestParam(value = "dateEnd", required = false) String dateEnd,
    @RequestParam(value = "page", defaultValue = "0") int page,
    Model model) throws ParseException {

    Page<QnaResponseDto> list;
    List<Integer> pageList;

    if ((keywordType == null) && (keyword == null) && (dateStart == null) && (dateEnd == null)
        || (dateStart.equals("null")) && (dateEnd.equals("null")) && (keyword.equals("null"))
        || (dateStart.equals("")) && (dateEnd.equals("")) && (keyword.equals("")))) { // 초기기능을 쓰지 않을 때
        list = qnaService.findAll(page);
    } else {
        // 오늘, 어제, 1주일, 1개월 검색
        if ((!dateStart.equals("")) && (dateEnd.equals(""))) {
            list = qnaService.findByDate(dateStart, page);
            // 기간 검색
        } else if ((!dateStart.equals("")) && (!dateEnd.equals(""))) {
            list = qnaService.findByDate(dateStart, dateEnd, page);
        } else {
            // 검색 키워드가 있을 때
            list = qnaService.findByKeyword(keywordType, keyword, page);
        }
    }

    pageList = noticeService.getPageList(list.getTotalPages(), page);
    model.addAttribute("keywordType", keywordType);
    model.addAttribute("keyword", keyword);
    model.addAttribute("dateStart", dateStart);
    model.addAttribute("dateEnd", dateEnd);
    model.addAttribute("list", list);
    model.addAttribute("pageList", pageList);
    model.addAttribute("listCount", qnaRepository.count());
    return "admin/qna/list";
}
```

```
QnaService.java

@Transactional(readOnly = true)
public Page<QnaResponseDto> findByDate(String mode, int page) throws ParseException {
    String[] date = orderService.dateSetting(mode);
    String dateStartStr = date[0];
    String dateEndStr = date[1];

    return findByDate(dateStartStr, dateEndStr, page);
}
```

```
QnaService.java

@Transactional(readOnly = true)
public Page<QnaResponseDto> findByDate(String start, String end, int page) throws ParseException {
    Page<QnaEntity> list;
    Pageable pageable = PageRequest.of(page, 10);

    String[] date = orderService.dateSetting(start, end);
    String dateStartStr = date[0];
    String dateEndStr = date[1];

    list = qnaRepository.findByQnaLocalDateTimeContaining(dateStartStr, dateEndStr, pageable);
    return list.map(QnaResponseDto::new);
}
```

```
QnaService.java

@Transactional(readOnly = true)
public Page<QnaResponseDto> findByKeyword(String keywordType, String keyword, int page) {
    Page<QnaEntity> list = null;
    List<Sort.Order> sorts = new ArrayList<>();
    sorts.add(Sort.Order.desc("qnaId"));
    Pageable pageable = PageRequest.of(page, 10, Sort.by(sorts));
    if (keywordType.equals("title")) {
        list = qnaRepository.findByQnaContentContaining(keyword, pageable);
    } else if (keywordType.equals("writer")) {
        list = qnaRepository.findByQnaNameContaining(keyword, pageable);
    } else {
        list = qnaRepository.findByQnaCategoryContaining(keyword, pageable);
    }
    return list.map(QnaResponseDto::new);
}
```

통합문의관리-답변등록

The screenshot shows a web-based administration interface for managing inquiries. On the left, a sidebar lists various management categories: 회원관리 (User Management), 상품관리 (Product Management), 주문관리 (Order Management), 리뷰관리 (Review Management), 통합문의관리 (Integrated Inquiry Management), 상품문의관리 (Product Inquiry Management), and 공지사항관리 (Notice Management). The '통합문의관리' item is currently selected and highlighted.

The main content area displays two forms:

- 문의 (Inquiry Detail)**: A form showing a query from a user named '니네임' (Nirename) regarding a product inquiry. The details include:
 - 작성자: 니네임
 - 문의유형: 상품문의 (Product Inquiry)
 - 작성일자: 2023-04-04 18:25
 - 문의내용: 안녕하세요.A red box labeled '1' highlights this entire section.
- 답변 등록 (Response Registration)**: A form for registering a response to the above inquiry. It includes:
 - 작성자: 관리자 (Administrator)
 - 답변내용: (Empty text area)A red box labeled '2' highlights this section.

1. 문의 상세
작성자/문의유형/작성일자/문의내용

2. 답변 등록 시 얼럿창

localhost:8080 내용:

답변등록 완료

확인

통합문의관리-답변수정/삭제

관리자페이지

통합문의관리

NURIRI 로그아웃

문의

작성자	닉네임
문의유형	상품문의
작성일자	2023-04-04 18:25
문의내용	안녕하세요.

답변

작성자	관리자
답변내용	네 안녕하세요
작성일자	2023-05-15 15:42

수정 삭제 목록

1

2

1. 문의 답변
작성자/답변내용/작성일자

2. 답변 수정/삭제 시 얼럿창

localhost:8080 내용:

답변수정 완료

확인

통합문의관리-답변 등록/수정/삭제

등록/수정/삭제

```
●●●  
@PostMapping("admin/qna/comment/write")  
@ResponseBody  
public String commentWrite(QnaCommentSaveDto dto){  
  
    Long commentQnaId = dto.getCommentQnaId();  
    boolean result = qnaCommentService.save(dto.toEntity());  
    if(!result){  
        return "<script>alert('답변등록 실패');history.back();</script>";  
    }  
    return "<script>alert('답변등록 완료'); location.href='/admin/qna/content/" + commentQnaId + "'"; </script>";  
}
```

AdminQnaController.java

```
●●●  
@PostMapping("admin/qna/comment/modify")  
@ResponseBody  
public String commentModify(QnaCommentSaveDto dto){  
    Long commentQnaId = dto.getCommentQnaId();  
    boolean result = qnaCommentService.save(dto.toModifyEntity());  
    if(!result){  
        return "<script>alert('답변수정 실패');history.back();</script>";  
    }  
    return "<script>alert('답변수정 완료');location.href='/admin/qna/content/" + commentQnaId + "'";</script>";  
}
```

AdminQnaController.java

```
●●●  
@GetMapping("admin/qna/comment/delete/{id}")  
@ResponseBody  
public String commentDelete(@PathVariable("id") long id, @RequestParam Long qnaId){  
    boolean result = qnaCommentService.delete(id);  
    if(!result){  
        return "<script>alert('답변삭제 실패');history.back();</script>";  
    }  
    return "<script>alert('답변삭제 완료');location.href='/admin/qna/content/" + qnaId + "'";</script>";  
}
```

AdminQnaController.java

```
●●●  
@Transactional  
public boolean save(QnaCommentEntity qnaCommentEntity) {  
    try{  
        qnaCommentRepository.save(qnaCommentEntity);  
    }catch (Exception e){  
        e.printStackTrace();  
        return false;  
    }  
    return true;  
}
```

QnaCommentService.java

```
●●●  
@Transactional  
public boolean delete(Long id){  
    try {  
        qnaCommentRepository.deleteById(id);  
    }catch (Exception e){  
        e.printStackTrace();  
        return false;  
    }  
    return true;  
}
```

QnaCommentService.java

상품문의관리-목록

관리자페이지

상품문의관리

NURIRI 로그아웃

기간: 연도-월-일 ~ 연도-월-일 오늘 어제 1주일 1개월

검색조건: 회원아이디 검색

전체 문의 내역 : 6

번호	상품명	내용	작성자	작성일	관리
6	도트 sk	스크린이랑 같나요	비회원	2023-04-04 18:25	상세정보 2
5	린넨 ops 베이지	청사이즈인가요	비회원	2023-04-04 18:25	상세정보
4	브라운 사각 bag	배송빨리빨리	비회원	2023-04-04 18:25	상세정보
3	도트 sk	스크린이랑 같나요	right	2023-04-04 18:25	상세정보
2	린넨 ops 베이지	청사이즈인가요	lee	2023-04-04 18:25	상세정보
1	브라운 사각 bag	배송빨리빨리	hong	2023-04-04 18:25	상세정보

이전 1 다음

1. 상품 문의 검색

기간: 날짜 선택/오늘/어제/1주일
/1개월

키워드 검색조건:
회원아이디/내용

2. 상품문의 상세 페이지로 이동

3. 페이지 nav

상품문의관리-목록

검색 기능

AdminInquiryController.java

```
•••
@GetMapping("/admin/inquiry/list")
public String list(Model model,
    @RequestParam(value = "dateStart", required = false) String dateStart,
    @RequestParam(value = "dateEnd", required = false) String dateEnd,
    @RequestParam(value = "keyword", required = false) String keyword,
    @RequestParam(value = "findBy", required = false) String findBy,
    @RequestParam(value = "page", defaultValue = "0") int page) throws ParseException {

    Page<InquiryResponseDto> list;
    if(keyword == null && findBy == null && dateStart == null && dateEnd == null
        || (dateStart.equals("null")) && (dateEnd.equals("null")) && (keyword.equals("null"))
        || (dateStart.equals("")) && (dateEnd.equals("")) && (keyword.equals(""))){
        list = inquiryService.getPage(page);
    }else {
        if(!dateStart.equals("") && dateEnd.equals ""){
            list = inquiryService.findByDate(dateStart, page);
        } else if ((!dateStart.equals("")) && (!dateEnd.equals(""))){
            list = inquiryService.findByDate(dateStart, dateEnd, page);
        } else {
            list = inquiryService.findByKeyword(findBy, keyword, page);
        }
    }
    int totalPage = list.getTotalPages();
    List<Integer> pageList = noticeService.getPageList(totalPage, page);

    List<String> itemList = new ArrayList<>();
    for (InquiryResponseDto dto: list){
        Long itemNo = dto.getItemNo();
        ProductResponseDto itemDto = productService.findById(itemNo);
        itemList.add(itemDto.getItemName());
    }
    model.addAttribute("itemList", itemList);
    model.addAttribute("list", list);
    model.addAttribute("findBy", findBy);
    model.addAttribute("keyword", keyword);
    model.addAttribute("pageList", pageList);
    model.addAttribute("dateStart", dateStart);
    model.addAttribute("dateEnd", dateEnd);

    long itemCount = inquiryRepository.count();
    model.addAttribute("listCount", itemCount );

    return "admin/inquiry/list";
}
```

InquiryService.java

```
@Transactional(readOnly = true)
public Page<InquiryResponseDto> findByKeyword(String findBy, String keyword, int page){
    Page<InquiryEntity> list;
    List<Sort.Order> sorts = new ArrayList<>();
    sorts.add(Sort.Order.desc("inquiryNo"));
    Pageable pageable = PageRequest.of(page, 10, Sort.by(sorts));

    if(findBy.contains("memberId")){
        list = inquiryRepository.findByMemberIdContaining(keyword, pageable);
    } else {
        list = inquiryRepository.findByInquiryContentContaining(keyword, pageable);
    }
    return list.map(InquiryResponseDto::new);
}
```

InquiryService.java

```
public Page<InquiryResponseDto> findByDate(String mode, int page) throws ParseException {
    String[] date = orderService.dateSetting(mode);
    String dateStartStr = date[0];
    String dateEndStr = date[1];
    return findByDate(dateStartStr, dateEndStr, page);
}
```

InquiryService.java

```
@Transactional(readOnly = true)
public Page<InquiryResponseDto> findByDate(String start, String end, int page) throws ParseException {
    Page<InquiryEntity> list;
    Pageable pageable = PageRequest.of(page, 10);

    String[] date = orderService.dateSetting(start, end);
    String dateStartStr = date[0];
    String dateEndStr = date[1];

    list = inquiryRepository.findByReviewNoContaining(dateStartStr, dateEndStr, pageable);

    return list.map(InquiryResponseDto::new);
}
```

상품문의관리-답변등록

관리자페이지

상품문의관리

NURIRI 로그아웃

1

문의

작성자	비회원
상품명	도트 sk
작성일자	2023-04-04 18:25
문의내용	스크린이랑 같나요

2

답변 등록

작성자	관리자
답변내용	

등록

1. 문의 상세
작성자/상품명/작성일자/문의내용

2. 답변 등록 시 얼럿창

localhost:8080 내용:
답변등록 완료

확인

상품문의관리-답변수정/삭제

관리자페이지

상품문의관리

NURIRI 로그아웃

문의

작성자	비회원
상품명	도트 sk
작성일자	2023-04-04 18:25
문의내용	스크린이랑 같나요

답변

작성자	관리자
답변내용	안녕하세요 고객님 네 똑같습니다^^ 모니터에 따라 색상차이가 있을 수 있으니 참고해주세요~
작성일자	2023-05-15 15:06

수정 삭제 목록

1

2

1. 문의 답변
작성자/답변내용/작성일자

2. 답변 수정/삭제 시 얼럿창

localhost:8080 내용:

답변수정 완료

확인

상품문의관리-답변 등록/수정/삭제

등록/수정/삭제

```
●●●  
AdminInquiryController.java  
  
@RequestMapping("/admin/inquiry/writeAction")  
@ResponseBody  
public String replyWrite(InReplySaveResponseDto dto, @RequestParam("replyInquiryNo") Long replyInquiryNo){  
    boolean result = inReplyService.save(dto);  
    if(result) {  
        return "<script>alert('답변등록 완료'); location.href='/admin/inquiry/content?inquiryNo=" + replyInquiryNo + "'"; </script>";  
    }else{  
        return "<script>alert('답변등록 실패'); history.back();</script>";  
    }  
}
```

```
●●●  
AdminQnaController.java  
  
@RequestMapping("/admin/inquiry/modifyAction")  
@ResponseBody  
public String modify(InReplySaveResponseDto dto,  
                     @RequestParam("replyNo") Long replyNo,  
                     @RequestParam("replyInquiryNo") Long replyInquiryNo){  
  
    boolean result = inReplyService.modify( dto,replyNo );  
    if(result) {  
        return "<script>alert('답변수정 완료'); location.href='/admin/inquiry/content?inquiryNo=" + replyInquiryNo + "'"; </script>";  
    }else{  
        return "<script>alert('답변수정 실패'); history.back();</script>";  
    }  
}
```

```
●●●  
AdminQnaController.java  
  
@RequestMapping("/admin/inquiry/deleteAction")  
@ResponseBody  
public String delete(@RequestParam("replyNo") Long replyNo,  
                     @RequestParam("replyInquiryNo") Long replyInquiryNo){  
  
    boolean result = inReplyService.delete(replyNo);  
    if(result){  
        return "<script>alert('답변삭제 완료'); location.href='content?inquiryNo=" + replyInquiryNo + "'</script>";  
    }else {  
        return "<script>alert('답변삭제 실패'); history.back();</script>";  
    }  
}
```

```
●●●  
InReplyService.java  
  
@Transactional  
public boolean save(final InReplySaveResponseDto dto){  
    try{  
        inReplyRepository.save( dto.toEntity() );  
    }catch (Exception e){  
        e.printStackTrace();  
        return false ;  
    }  
    return true;  
}
```

```
●●●  
InReplyService.java  
  
@Transactional  
public boolean modify(final InReplySaveResponseDto dto, final Long replyNo){  
    try {  
        Optional<InReplyEntity> optional = inReplyRepository.findById(replyNo);  
        InReplyEntity entity = optional.get();  
        entity.update( dto.getReplyContent(), dto.getReplyInquiryNo() );  
        inReplyRepository.save( entity );  
        return true;  
    }catch (Exception e){  
        e.printStackTrace();  
        return false;  
    }  
}
```

```
●●●  
InReplyService.java  
  
@Transactional  
public boolean delete(Long inquiryReplyNo){  
    try {  
        Optional<InReplyEntity> optional = inReplyRepository.findById(inquiryReplyNo);  
        inReplyRepository.delete(optional.get());  
        return true;  
    }  
    catch (Exception e){  
        e.printStackTrace();  
        return false;  
    }  
}
```

공지관리-목록

The screenshot shows the 'Notice Management - List' page. On the left is a sidebar with various management links. The main area has a header '공지사항관리' and a search bar with dropdowns for '공지유형' (Type) and '검색조건' (Search Condition). A red box labeled '1' highlights the search bar area. Below it is a table for notice entries. The first row of the table is highlighted with a red box labeled '2'. It contains columns for '번호' (Number), '제목' (Title), '작성일' (Created Date), and '관리' (Management). The title column shows '공지[필독] 구매 전 꼭 확인해주...'. The date is '2023-05-12 12:16'. The management column has two buttons: '상세정보' (Details) and '삭제' (Delete), both highlighted with a red box labeled '3'. At the bottom of the table is a navigation bar with buttons '이전', '1', and '다음', with '1' highlighted with a red box labeled '4'.

1. 상품 검색

공지유형: 전체/공지사항/이벤트
키워드 검색조건: 제목/내용

2. 글쓰기 페이지로 이동

3. 상세정보 페이지로 이동 공지사항 삭제

4. 페이지 nav

공지관리-목록

검색 기능



AdminNoticeController.java

```
@GetMapping("/admin/notice/list")
public String noticeList(@RequestParam(value = "findBy", required = false) String findBy,
                        @RequestParam(value = "keyword", required = false) String keyword,
                        @RequestParam(value = "page", defaultValue = "0") int page,
                        Model model) {
    Page<NoticeResponseDto> list;
    if (findBy == null) {
        list = noticeService.findAll(page);
    } else{
        list = noticeService.findByKeyword(findBy, keyword, page);
    }

    int totalPage = list.getTotalPages();
    List<Integer> pageList = noticeService.getPageList(totalPage, page);

    model.addAttribute("list", list);
    model.addAttribute("findBy", findBy);
    model.addAttribute("keyword", keyword);
    model.addAttribute("pageList", pageList);

    long listCount = noticeRepository.count();
    model.addAttribute("listCount", listCount);

    return "admin/notice/list";
}
```



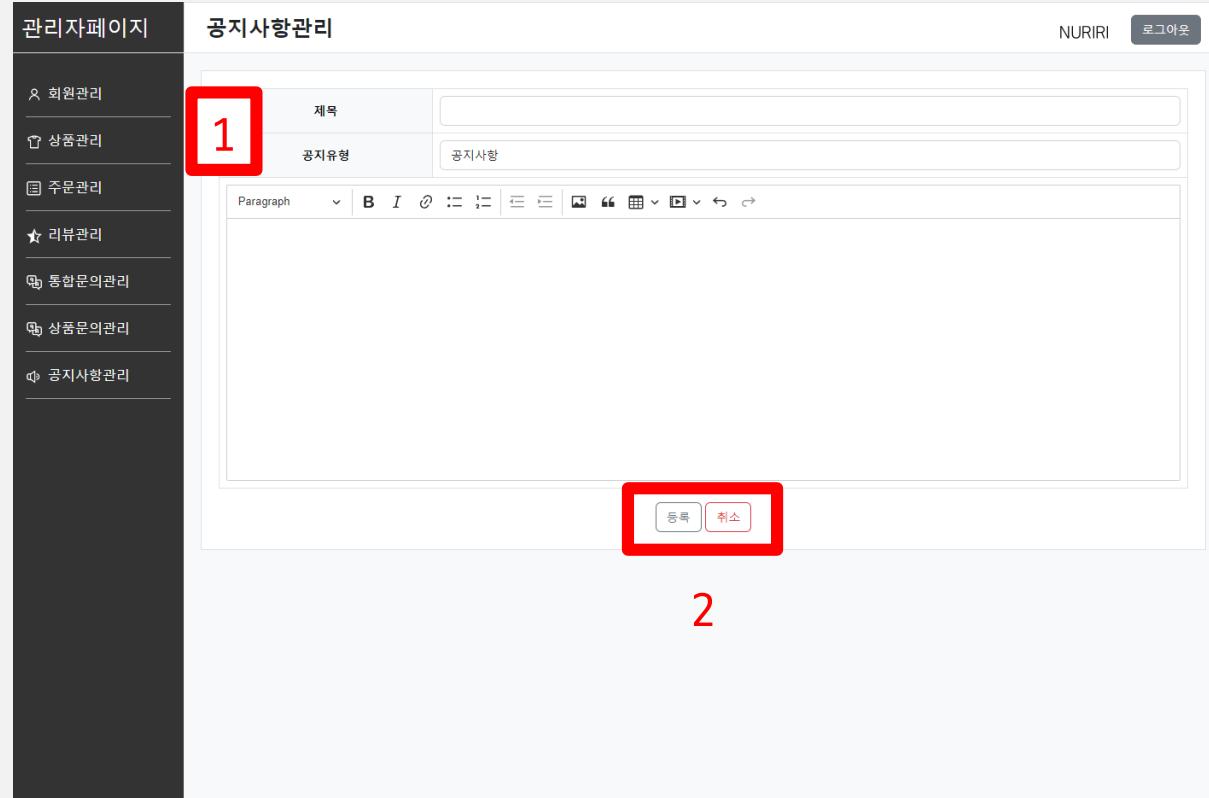
NoticeService.java

```
@Transactional(readOnly = true)
public Page<NoticeResponseDto> findByKeyword(String findBy, String keyword, int page)
{
    Page<NoticeEntity> list;
    List<Sort.Order> sorts = new ArrayList<>();
    sorts.add(Sort.Order.desc("noticeNo"));
    Pageable pageable = PageRequest.of(page, 10, Sort.by(sorts));

    if (findBy.equals("title")){ // 제목으로 찾을 때
        list = noticeRepository.findByNoticeTitleContaining(keyword, pageable);
    }else if (findBy.equals("content")){ // 내용으로 찾을 때
        pageable = PageRequest.of(page, 10);
        list = noticeRepository.findByNoticeContentContaining(keyword, pageable);
    }else{ // 공지사항인지, 이벤트인지 구분할 때
        list = noticeRepository.findByNoticeType(keyword, pageable);
    }

    return list.map(NoticeResponseDto::new);
}
```

공지관리-글쓰기



1. 게시판 제목

공지유형 : 공지사항, 이벤트
CKeditor5

2. 등록/취소

공지관리-상세

관리자페이지

공지사항관리

NURIRI 로그아웃

회원관리

상품관리

주문관리

리뷰관리

통합문의관리

상품문의관리

1

공지사항 관리

번호	1
제목	공지[필독] 구매 전 꼭 확인해주세요!
카테고리	공지사항
작성일	2023-05-12 12:16

공지사항 내용

spring boot

2

목록 수정 삭제

1. 게시판 번호/제목/카테고리/작성일/내용 확인 가능
2. 목록 버튼: 목록 페이지로 이동
수정 버튼: 수정 페이지로 이동
삭제 버튼: 해당 게시판 삭제

공지관리-수정

관리자페이지

공지사항관리

NURIRI 로그아웃

회원관리

상품관리

주문관리

리뷰관리

통합문의관리

상품문의관리

공지사항관리

번호: 1

제목: 공지[필독] 구매 전 꼭 확인해주세요~

공지유형: 공지사항

Paragraph 툴바

공지사항 내용

Spring Boot

수정 취소

수정 버튼: 수정된 input 내용으로 수정
취소 버튼: 상세 페이지로 이동

공지관리-목록

공지사항 등록/수정/삭제

```
AdminNoticeController.java
@PostMapping("/admin/notice/writeAction")
@ResponseBody
public String noticeWriteAction(NoticeSaveRequestDto dto) {

    if(dto.getNoticeTitle().length() > 250){
        return "<script>alert('제목은 250자 이내로 써주세요.');" + history.back();</script>";
    }

    Boolean success = noticeService.save(dto);
    if(success) {
        return "<script>alert('게시글 등록 완료'); location.href='/admin/notice/list';</script>";
    }else{
        return "<script>alert('게시글 등록 실패'); history.back();</script>";
    }
}
```

```
AdminNoticeController.java
@PostMapping("/admin/notice/modifyAction")
@ResponseBody
public String noticeModifyAction(NoticeUpdateRequestDto dto) {

    if(dto.getNoticeTitle().length() > 250){
        return "<script>alert('제목은 250자 이내로 써주세요.');" + history.back();</script>";
    }

    Boolean success = noticeService.update(dto);
    if(success) {
        return "<script>alert('게시글 수정 완료'); location.href='/admin/notice/content/" + dto.getNoticeNo() + "'</script>";
    }else{
        return "<script>alert('게시글 수정 실패'); history.back();</script>";
    }
}
```

```
AdminNoticeController.java
@GetMapping("/admin/notice/deleteAction")
@ResponseBody
public String noticeDeleteAction(@RequestParam("noticeNo") Long noticeNo) {

    Boolean success = noticeService.delete(noticeNo);
    if(success) {
        return "<script>alert('게시글 삭제 완료'); location.href='/admin/notice/list';</script>";
    }else{
        return "<script>alert('게시글 삭제 실패'); history.back();</script>";
    }
}
```

```
NoticeService.java
@Transactional
public Boolean save(final NoticeSaveRequestDto dto) {

    try{
        NoticeEntity entity = dto.toEntity();
        noticeRepository.save(entity);
    }
    catch(Exception e){
        e.printStackTrace();
        return false;
    }
    return true;
}
```

```
NoticeService.java
@Transactional
public Boolean update(final NoticeUpdateRequestDto dto) {

    Optional<NoticeEntity> entity = noticeRepository.findById(dto.getNoticeNo());
    if (!entity.isPresent()){
        return false;
    }

    entity.get().update(dto.getNoticeType(), dto.getNoticeTitle(), dto.getNoticeContent());
    return true;
}
```

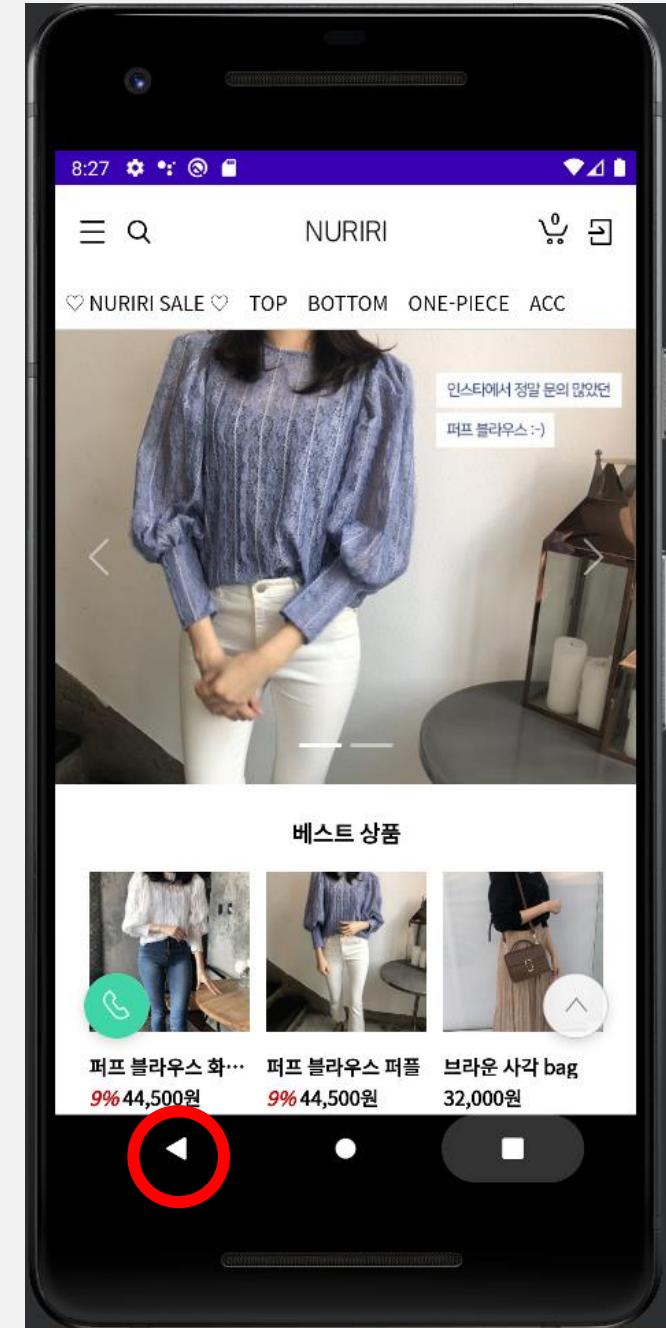
```
NoticeService.java
@Transactional
public Boolean delete(final Long noticeNo) {
    Optional<NoticeEntity> entity = noticeRepository.findById(noticeNo);
    if (!entity.isPresent()){
        return false;
    }
    try{
        noticeRepository.delete(entity.get());
    }
    catch(Exception e){
        e.printStackTrace();
        return false;
    }
    return true;
}
```

안드로이드 앱

뒤로가기 버튼

```
override fun onBackPressed() { //백버튼 기능 누를 시 수행할 로직 구현

    if (webView == webPopupView){ // 팝업창이 아닐때, 백버튼 구현
        if(webView.canGoBack()){ // 웹사이트에서 뒤로 갈 페이지가 존재한다면
            webView.goBack() //웹사이트 뒤로가기
        }else{
            super.onBackPressed() //본래의 백버튼 기능 수정(안드로이드)
        }
    }else{ // 팝업창일때
        // 백버튼을 누르면 창 닫게 만들기
        containerView.removeView(webPopupView)
        webPopupView.destroy()
        webPopupView = webView
    }
}
```



alert(), confirm()

alert()

```
// JS로 알럿창을 띄울때 호출됨.  
override fun onJsAlert(  
    view: WebView?,  
    url: String?,  
    message: String?,  
    result: JsResult  
) : Boolean {  
    AlertDialog.Builder(view!!.context)  
        .setMessage(message)  
        .setPositiveButton(  
            android.R.string.ok  
        ) { dialog, which -> result.confirm() }  
        .setNegativeButton(  
            android.R.string.cancel  
        ) { dialog, which -> result.cancel() }  
        .create()  
        .show()  
  
    return true  
}
```

confirm()

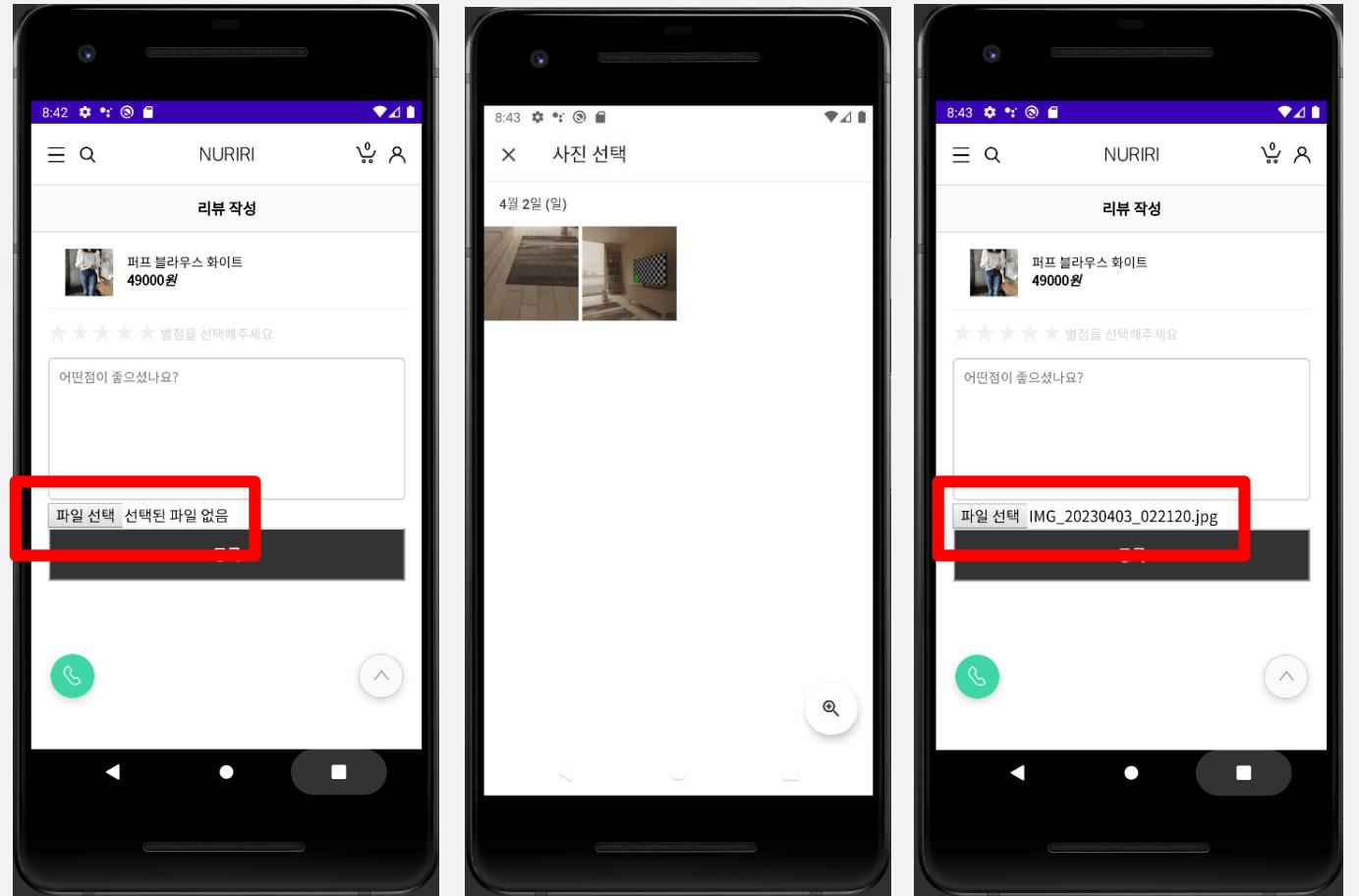
```
// JS로 confirm창을 띄울때 호출됨.  
override fun onJsConfirm(  
    view: WebView?,  
    url: String?,  
    message: String?,  
    result: JsResult  
) : Boolean {  
    AlertDialog.Builder(view!!.context)  
        .setTitle("title")  
        .setMessage(message)  
        .setPositiveButton(  
            android.R.string.ok  
        ) { dialog, which -> result.confirm() }  
        .setNegativeButton(  
            android.R.string.cancel  
        ) { dialog, which -> result.cancel() }  
        .setCancelable(false)  
        .create()  
        .show()  
  
    return true  
}
```



input file

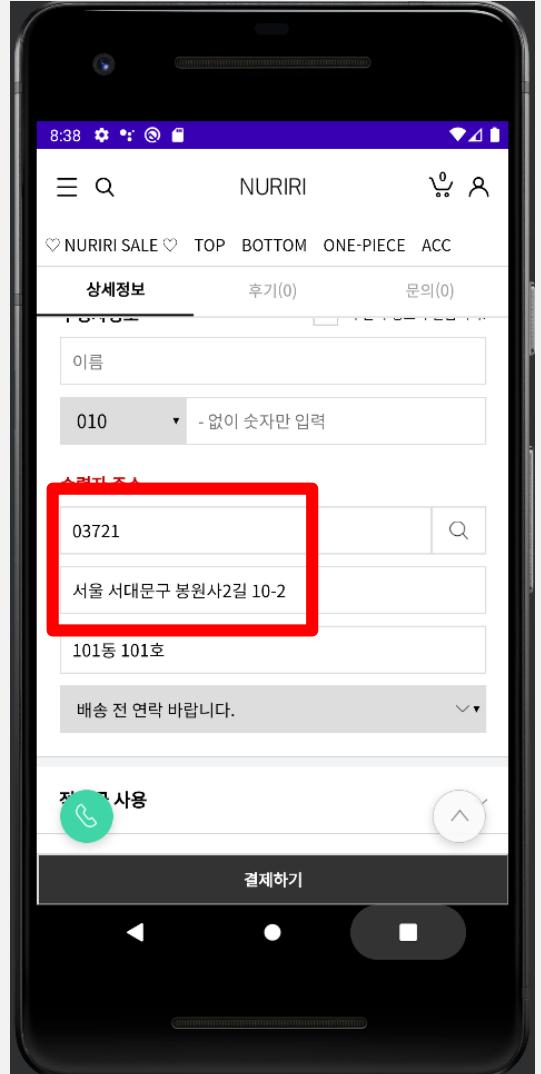
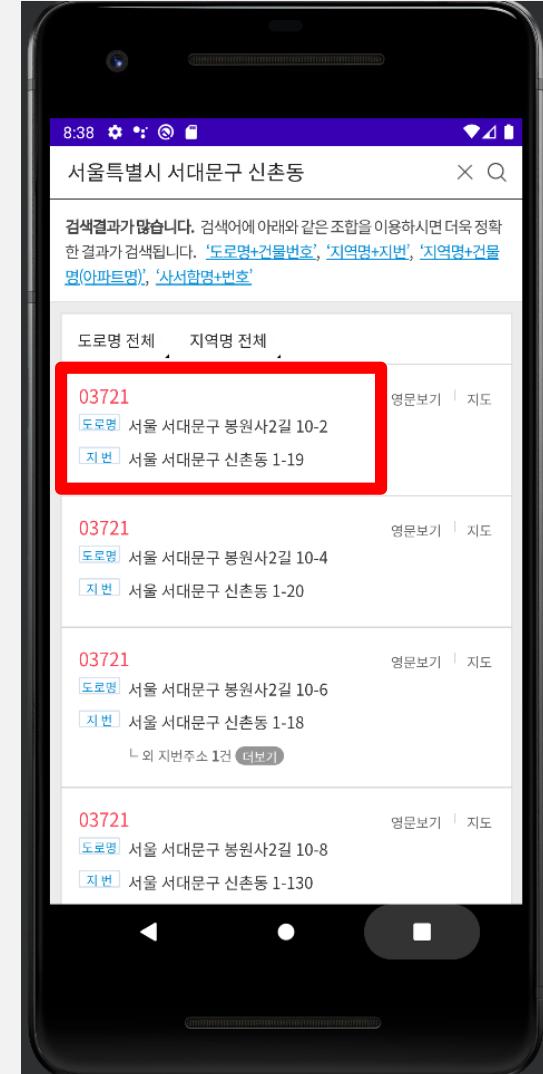
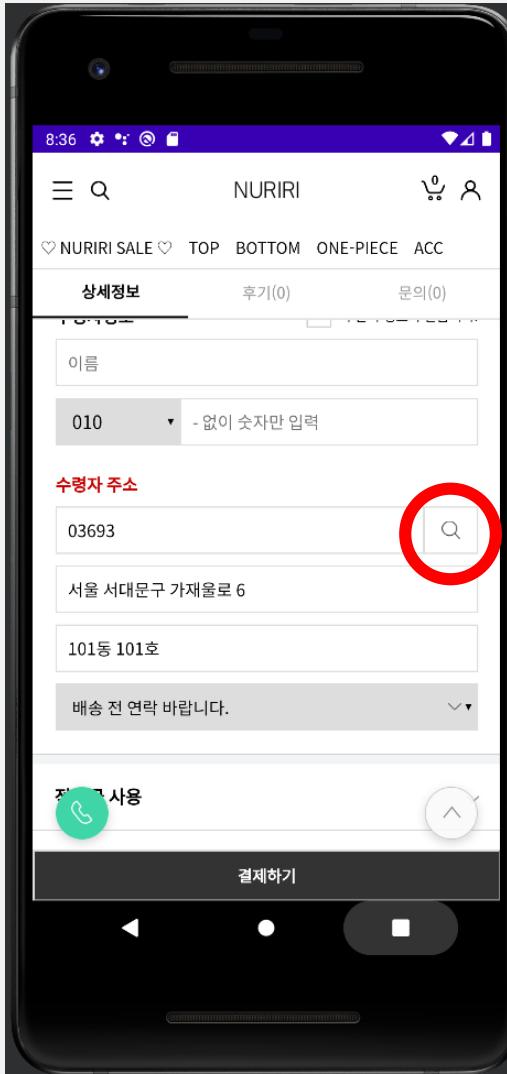
```
// input file(파일 선택 버튼)을 눌렀을 때 호출됨.  
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)  
override fun onShowFileChooser(  
    webView: WebView?,  
    filePathCallback: ValueCallback<Array<Uri?>?>,  
    fileChooserParams: FileChooserParams?  
>: Boolean {  
  
    if (mFilePathCallback != null) {  
        mFilePathCallback!!.onReceiveValue(null)  
        mFilePathCallback = null  
    }  
    mFilePathCallback = filePathCallback  
  
    val intent = Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI)  
    val REQUEST_CODE = 2000  
    startActivityForResult(intent, REQUEST_CODE)  
  
    return true  
}
```

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
    when (requestCode) {  
        2000 -> //fileChooser로 파일 선택 후 onActivityResult에서 결과를 받아 처리함  
            if (resultCode == RESULT_OK) {  
                //선택한 파일 경로  
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {  
                    mFilePathCallback!!.onReceiveValue(WebChromeClient.FileChooserParams.parseResult(resultCode, data))  
                } else {  
                    mFilePathCallback!!.onReceiveValue(arrayOf(data?.data))  
                }  
                mFilePathCallback = null  
            } else {  
                //cancel 했을 경우  
                if (mFilePathCallback != null) {  
                    mFilePathCallback!!.onReceiveValue(null)  
                    mFilePathCallback = null  
                }  
            }  
        }  
    }  
    super.onActivityResult(requestCode, resultCode, data)  
}
```

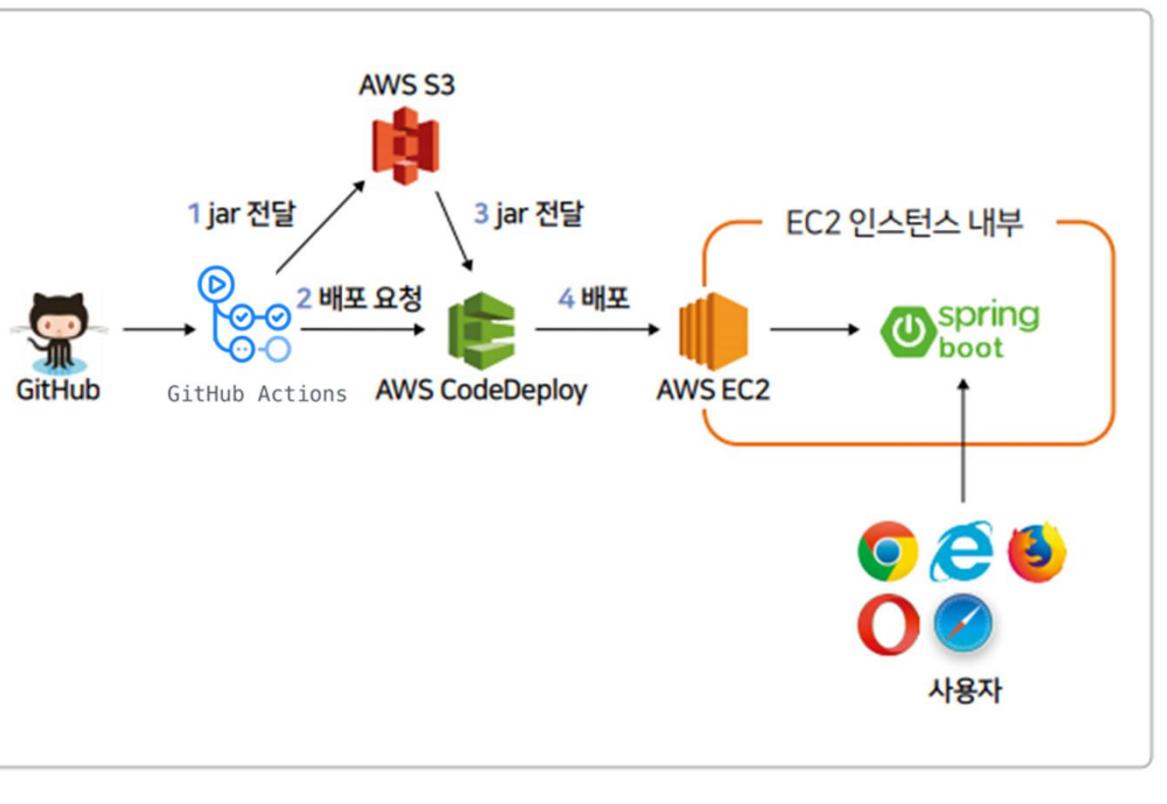


window.open(), window.close()

```
// JS로 window.open() 함수가 실행할 때 호출됨.  
@SuppressLint("SetJavaScriptEnabled")  
override fun onCreateWindow(  
    view: WebView?, isDialog: Boolean,  
    isUserGesture: Boolean, resultMsg: Message?): Boolean {  
  
    webPopupView = WebView(this@MainActivity).apply {  
        webViewClient = WebViewClient()  
        // 팝업창의 세팅 설정  
        settings.javaScriptEnabled = true  
        ...  
  
        // 팝업창 크기 설정  
        webPopupView.setLayoutParams(  
            ConstraintLayout.LayoutParams(  
                ConstraintLayout.LayoutParams.MATCH_PARENT,  
                ConstraintLayout.LayoutParams.MATCH_PARENT  
            )  
        )  
  
        containerView.addView(webPopupView) // 팝업창 containerView에 add  
  
        webPopupView.setWebChromeClient(object : WebChromeClient() {  
            // JS로 window.close() 함수가 실행할 때 호출됨.  
            override fun onCloseWindow(window: WebView) {  
                containerView.removeView(window)  
                window.destroy()  
                webPopupView = webView  
            }  
  
            webPopupView.setWebViewClient(object : WebViewClient() {  
  
                // 여러 event 함수 override  
            })  
  
            (resultMsg?.obj as WebView.WebViewTransport).webView = webPopupView  
            resultMsg.sendToTarget()  
            return true  
        })  
    }  
}
```



배 포



Github Actions, AWS CodeDeploy,
AWS S3, AWS EC2, AWS RDS를 이용하여
배포 진행

DB 식별자					
	▲ 역할	엔진	리전 및 AZ	크기	▼
○	database-ec2-loginjoindb	인스턴스	MariaDB	ap-northeast-1d	db.t2.micro
배포 ID	상태	배포 유형	컴퓨팅 플랫폼	애플리케이션	배포 그룹
○	d-EOJRCVNHN	성공	현재 위치	EC2/온프레미스	ec2-loginjoindb-codedeploy-app
					s3://action...
Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태
○	-	i-0c211a969cbb5f63b	대기 중	t2.micro	-
					경보 없음
		가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...	단력적 IP
		ap-northeast-1c	ec2-35-77-138-20.ap-n...	35.77.138.20	35.77.138.20

AWS RDS

AWS CodeDeploy

142

AWS EC2

시연 영상

T h a n k y o u