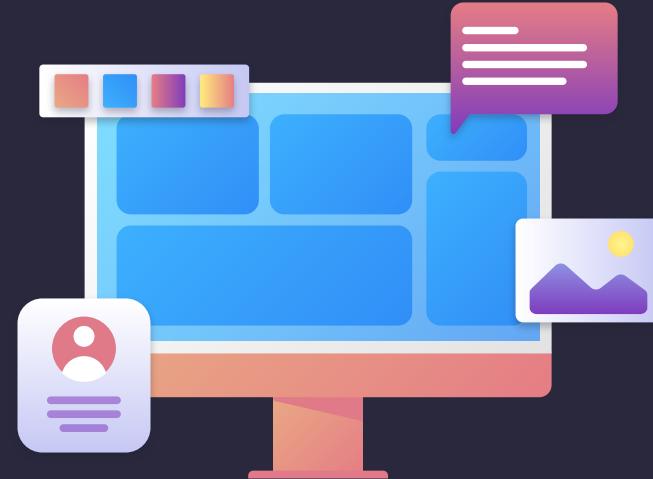


POLAROID

Team . 왕 꿈 트 리

SNS & Shopping



/목차



/01 /팀원 소개 및 역할

/02 /사이트 소개 & 개발 목적

/03 /개발 일정

/04 /DB & Mind Map 설계

/05 /개발도구 소개 &
주요기능

/06 /기능 시연 사진



/01 팀원 소개

Developer : 허성욱

Member & Spring Security & Oauth & SMTP

Developer : 오지석

SNS & Pay & Admin

Developer : 심형철

Live Chatting (STOMP & Sock JS & WebSocket)

Developer : 심은지

Like & Comment script & Web Design

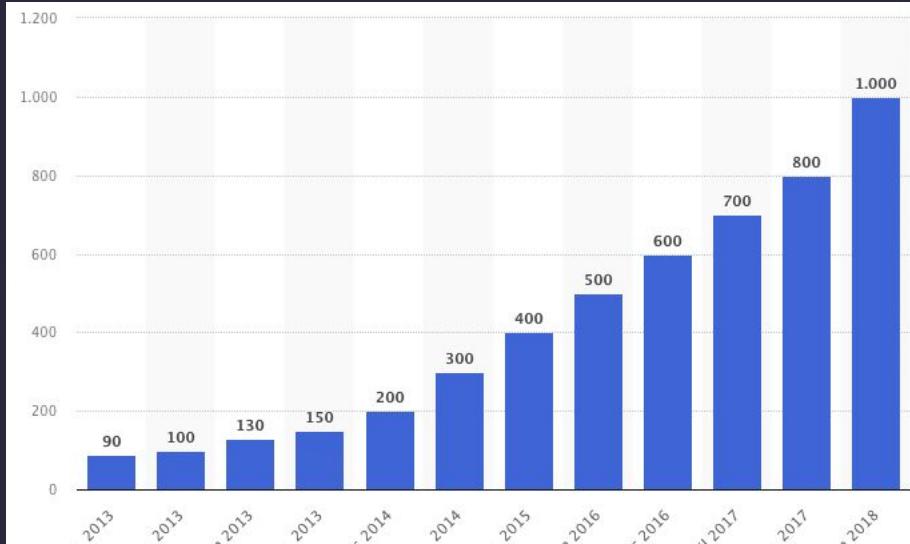
Developer : 고윤성

Shopping mall & Pick & Review



/02 사이트 소개 및 개발목적

SNS 사용자 증가 추이



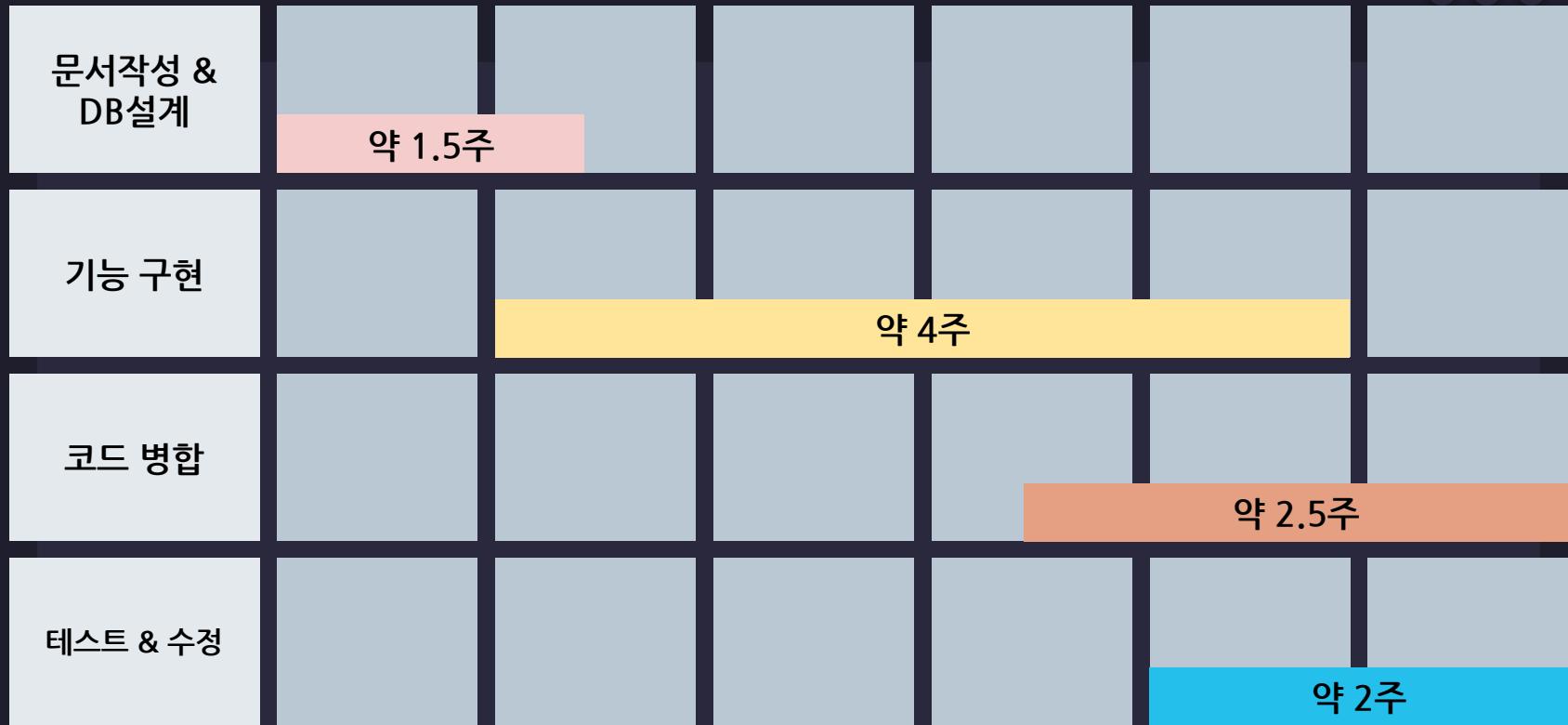
◎

최근 Instagram과 같은 SNS 시장이 지속적으로 커짐에 따라 SNS와 Shopping Mall을 결합한 사이트가 있으면 좋겠다고 생각해 해당 프로젝트를 진행함.

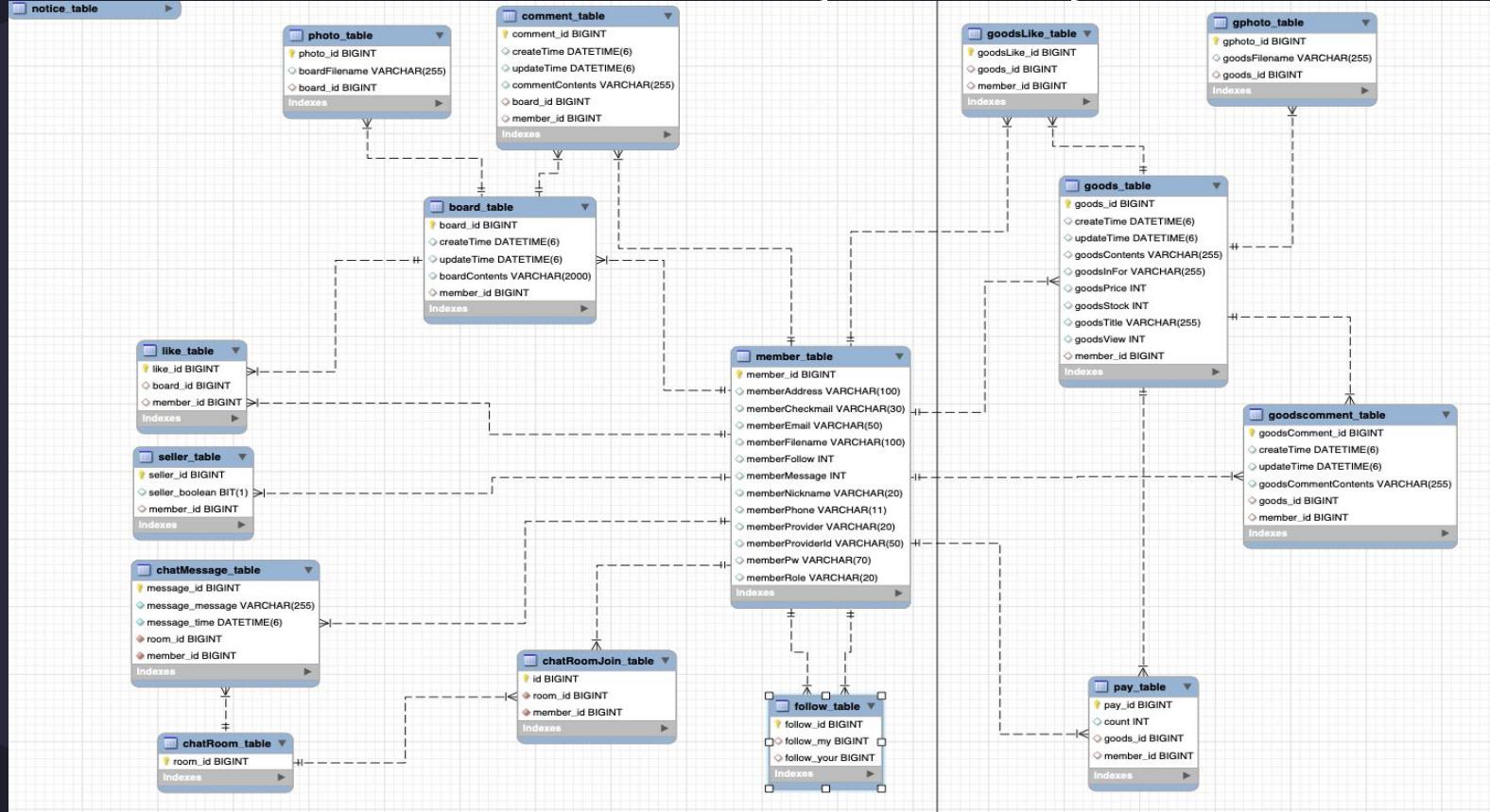
☰

☆

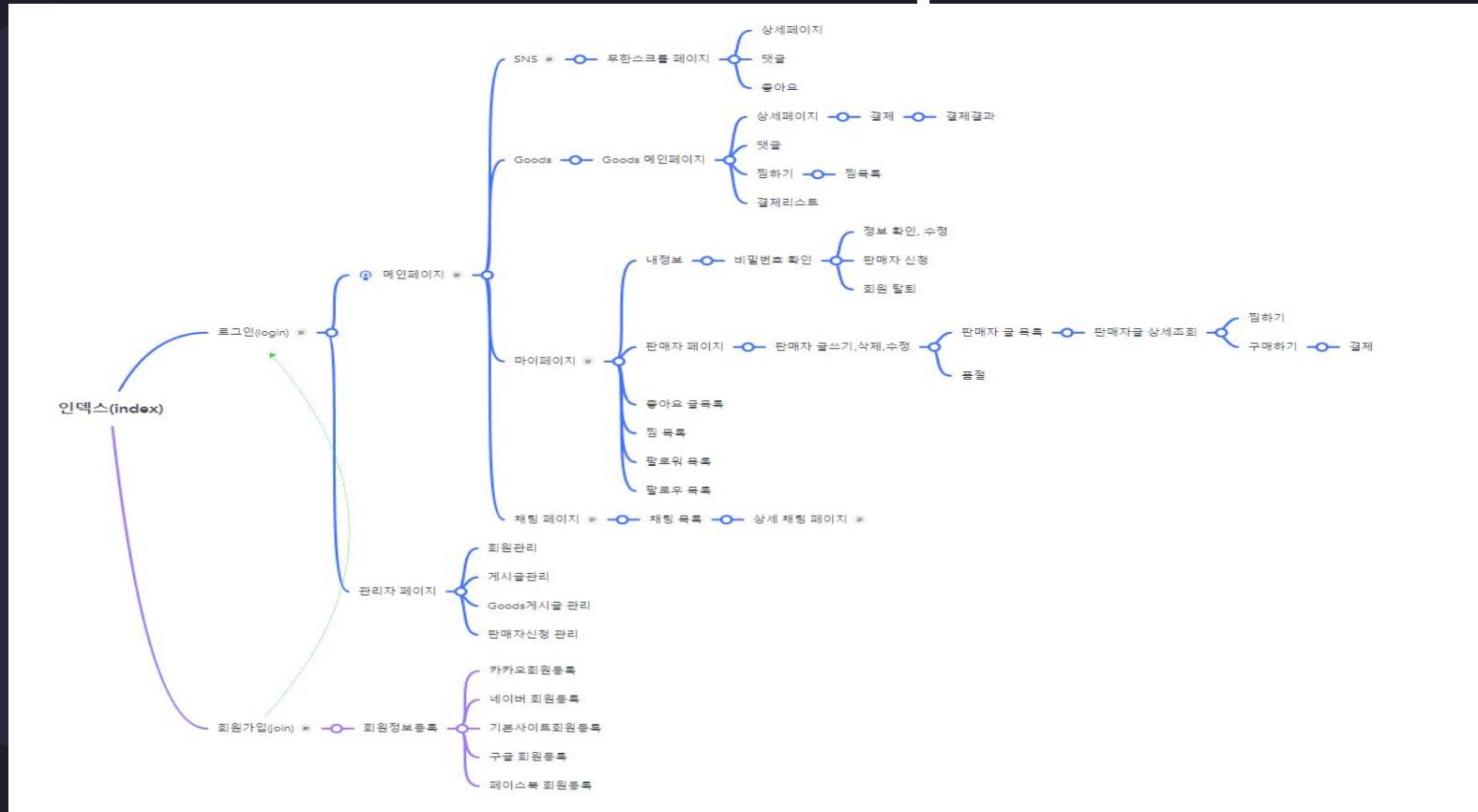
/03 개발 일정 (총 6주)



/04 DB 설계 (총 15개)

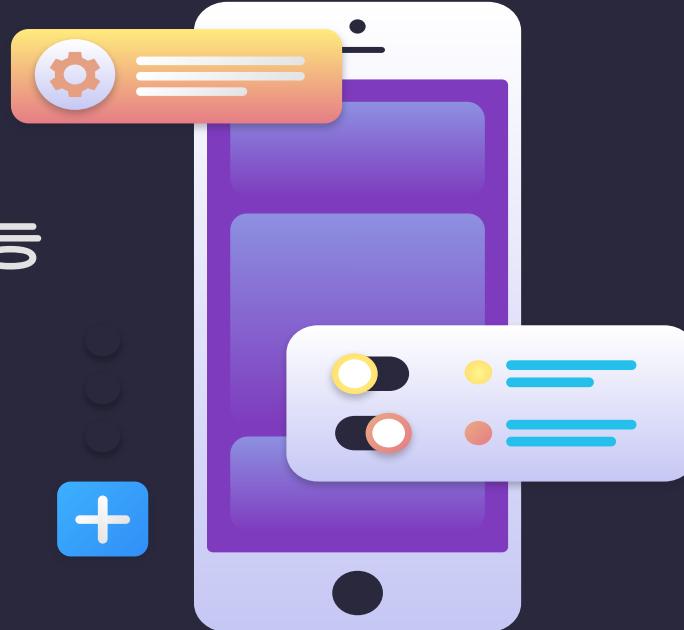


/04 Mind Map



/05

/개발도구 소개 & 주요 기능



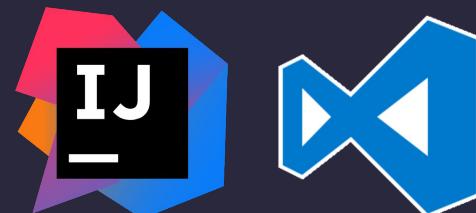
/05 개발 도구 및 환경



/ LANGUAGE



/ IDE



/ OTHER



/05 주요 기능

Member

- OAuth 2.0을 사용한 Login Service
 - ↳ Google, Kakao, Naver, Facebook
- Spring Security를 이용한 User 권한 관리
- E-Mail 인증을 통한 보안 관리

Admin

◎

- 회원 & SNS 게시물 & Goods 게시글 관리
- 회원의 판매자 신청시 판매자 권한부여

목

☆

/05 주요 기능

Goods (Shopping Mall)

- Pagination을 사용한 Paging 처리
 - 다중 파일 업로드 및 Review & Pick 기능구현
 - I'm port & 카카오결제 API를 이용한 결제기능
 - Ajax를 이용한 재고관리
- ◎ • 결제 리스트 건별 관리 및 결제 영수증 출력



/05 주요 기능

Board (SNS)

- 무한 스크롤을 사용해 SNS 게시판 구현
- 다중 파일 업로드 및 출력
- 좋아요 & 댓글

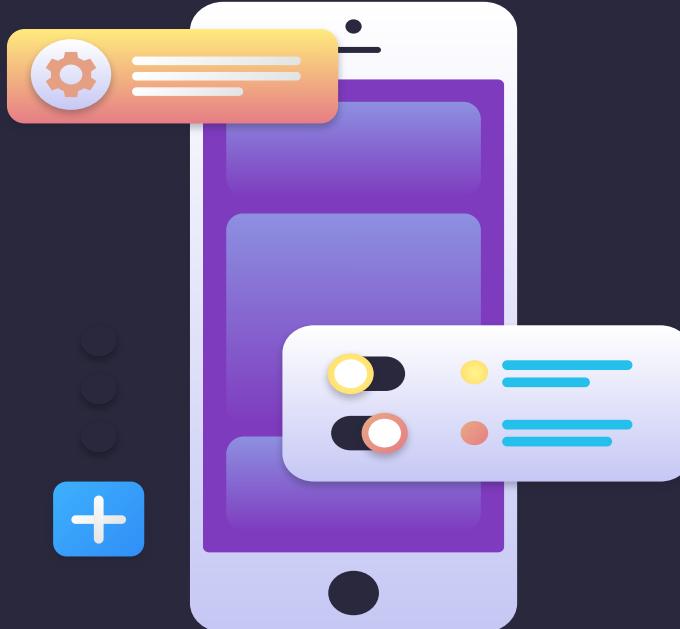
Chat

- WebSocket을 통해 양방향 통신기능 구현
- STOMP를 이용해 메시지의 유형, 형식, 내용 정의
- SockJS를 이용해 다양한 브라우저 지원

/06

/기능 시연 & 코드 사진

SNS 및 프로필 사진은 초상권 보호를 위해
이모티콘으로 가린 점 양해 바랍니다.





POLAROID

○

style

story

share

shopping

○

LOGIN

Signup

board

goods



index.html 화면

타임리프를 사용하여

관리자만 보이는 관리자페이지가 따로 출력되고 일반 유저들은 관리자페이지를 제외하고 출력이 된다.

```
<div class="vertical_box text_center">
    <div class="vertical_mid">
        <div class="index_logo">
            
        </div>
        <div class="index_text_box">
            <p class="index_text fs_8">○</p>
            <p class="index_text fs_14">style</p>
            <p class="index_text fs_14">story</p>
            <p class="index_text fs_14">share</p>
            <p class="index_text fs_14">shopping</p>
            <p class="index_text fs_8">○</p>
        </div>
        <div>
            <input class="index_btn index_login_btn fs_20 fw_b" th:if="${session['LoginEmail']} == null" type="button" value="LOGIN" onclick="location.href='/login'" />
            <input class="index_btn index_login_btn fs_20 fw_b" th:if="${session['LoginEmail']} != null" type="button" value="LOGOUT" onclick="location.href='/logout'" />
        </div>
        <div>
            <input class="index_btn index_signup_btn fs_16" th:if="${session['LoginEmail']} == null" type="button" value="Signup" onclick="location.href='/join'" />
            <input class="index_btn index_signup_btn fs_16" th:if="${session['LoginEmail']} != null" type="button" value="MYPAGE" onclick="location.href='/member/mypage'" />
        </div>
        <!-- 임시 버튼 -->
        <div>
            <input class="index_btn index_signup_btn fs_16" type="button" value="SNS" onclick="location.href='/board/'" />
            <!--
                <input class="index_btn index_signup_btn fs_16" type="button" value="board_detail" onclick="location.href='/board_detail'" />-->
        </div>
        <div>
            <input class="index_btn index_signup_btn fs_16" type="button" value="goods" onclick="location.href='/goods/'" />
            <!--
                <input class="index_btn index_signup_btn fs_16" type="button" value="board_detail" onclick="location.href='/board_detail'" />-->
        </div>
        <!-- 관리자 -->
        <a class="btn btn-outline-dark" href="/admin/memberList" th:if="${session['LoginRole']} == 'ROLE_ADMIN'">회원관리</a>
        <a class="btn btn-outline-dark" href="/admin/boardList" th:if="${session['LoginRole']} == 'ROLE_ADMIN'">게시글관리</a>
        <a class="btn btn-outline-dark" href="/admin/goodsList" th:if="${session['LoginRole']} == 'ROLE_ADMIN'">판매글관리</a>
        <a class="btn btn-outline-dark" href="/admin/sellerList" th:if="${session['LoginRole']} == 'ROLE_ADMIN'">판매원장 관리</a>
    </div>
```



join.html 화면

회원가입

아이디 5~20 사이의 영문 소문자와 숫자를 사용하여 입력하세요

인증 이메일 인증번호를 받을 이메일을 입력하세요 인증번호 발송

인증번호 인증번호

비밀번호 8~16자 영문 대 소문자, 숫자를 사용하여 입력하세요

비밀번호 확인 비밀번호를 다시 입력하세요

닉네임 영문 대 소문자, 숫자, 특수문자('_')중에서 입력하세요

전화번호 '-' 없이 숫자만으로 핸드폰 번호를 입력하세요

주소 우편번호 우편번호
 주소
 상세주소

가입하기



join.html 화면

Join 페이지에서 중요한 인증메일체크와 인증번호를 처리하는 script입니다

Ajax를 이용하여 인증코드발송메일을 확인후 확인이되면 인증요청한 이메일로 인증코드가 발송되며 코드를 비교하여 회원가입이 가능합니다.

```
// 인증메일체크
function mailCheck() {
    const mail = $("#memberCheckmail").val();
    if (mail.length==0){
        alert("인증번호를 받을 이메일을 입력하세요")
    }
    else {
        const checkResult=document.getElementById( elementId: "mailCheckStatus");
        $.ajax({
            type: 'post',
            url: "sendMail",
            data: {"mail": mail},
            success: function (result) {
                checkResult.style.color = 'green';
                checkResult.innerText='인증코드 발송 메일을 확인해주세요';
            },
            error: function () {
                checkResult.style.color = 'red';
                checkResult.innerText='인증코드 발송 실패 이메일을 확인해주세요';
            }
        });
    }
}

// 인증코드 체크
function codeCheck(){
    const code = $("#code").val();
    const checkResult=document.getElementById( elementId: "codeCheckStatus");
    if (code.length==0){
        checkResult.style.color = 'red';
        checkResult.innerText='인증코드를 입력하세요';
    }
    else {
        $.ajax({
            type: 'post',
            url: "codeCheck",
            data: {"code": code},
            success: function (result) {
                if (result == "ok") {
                    checkResult.style.color = 'green';
                    $("#memberCheckmail").attr("readonly",true);
                    checkResult.innerText = 'GOOD';
                } else {
                    checkResult.style.color = 'red';
                    checkResult.innerText = '인증코드 불일치';
                }
            },
            error: function () {
                console.log("코드확인 오류발생")
            }
        });
    }
}
```





회원가입 DB 화면

회원가입 시 DB에 저장되는 데이터 사진입니다.

Spring Security로 아이디 및 비밀번호를 암호화 처리된 정보로 저장했고, 역할을 지정해 접근 가능 역할이 아닐 시 **Interceptor** 처리되게 했습니다.





접근 불가 화면

localhost:8081 내용:

해당 페이지에 대한 권한이 없습니다.

확인



접근 불가 화면

Spring Security로 권한을 부여해 주소값을 변경해서 접근하는법을 막았습니다.

```
public class AccessDenied implements AccessDeniedHandler {  
  
    @Override  
    public void handle(HttpServletRequest request, HttpServletResponse response,  
                       AccessDeniedException accessDeniedException) throws IOException, ServletException {  
        response.setContentType("text/html; charset=UTF-8");  
  
        PrintWriter out = response.getWriter();  
  
        out.println("<script>alert('해당 페이지에 대한 권한이 없습니다.'); location.href='/'</script>");  
  
        out.flush();  
    }  
}
```



login.html 화면

로그인

일반 로그인

아이디

비밀번호

[로그인](#)

[비밀번호를 잊어버렸나요?](#)

[회원가입](#)

간편 로그인

N NAVER

KAKAOTALK

G GOOGLE

f FACEBOOK



login.html 화면

Oauth를 이용하여 페이스북, 카카오, 구글, 네이버 간편로그인도 가능하게 제작하였습니다.

```
public FacebookUserInfo(Map<String, Object> attributes) { this.attributes=attributes; }

@Override
public String getProviderId() { return (String) attributes.get("id"); }

@Override
public String getProvider() { return "facebook"; }

@Override
public String getEmail() { return (String) attributes.get("email"); }

@Override
public String getName() { return (String) attributes.get("name"); }

public class GoogleUserInfo implements OAuth2UserInfo{
    private Map<String, Object> attributes; // oauth2User.getAttributes

    public GoogleUserInfo(Map<String, Object> attributes) { this.attributes=attributes; }

    @Override
    public String getProviderId() { return (String) attributes.get("sub"); }

    @Override
    public String getProvider() { return "google"; }

    @Override
    public String getEmail() { return (String) attributes.get("email"); }

    @Override
    public String getName() { return (String) attributes.get("name"); }
```

```
public class KakaoUserInfo implements OAuth2UserInfo{
    private Map<String, Object> attributes; // oauth2User.getAttributes

    public KakaoUserInfo(Map<String, Object> attributes){ this.attributes=attributes; }

    @Override
    public String getProviderId() {
        int a=(Integer) attributes.get("id");
        return Integer.toString(a);
    }

    @Override
    public String getProvider() { return "kakao"; }

    @Override
    public String getEmail() {
        Map<String, Object> map= (Map<String, Object>) attributes.get("kakao_account");
        System.out.println(map);
        return (String) map.get("email");
    }

    @Override
    public String getName() {
        Map<String, Object> map= (Map<String, Object>) attributes.get("kakao_account");
        System.out.println(map);
        return (String) map.get("nickname");
    }
}
```

```
public interface OAuth2UserInfo {
    String getProviderId();
    String getProvider();
    String getEmail();
    String getName();
}
```



addInfo.html 화면

Oauth를 이용한 간편로그인시에 DB에 저장하기 위해 간단한 추가정보를 입력하여 가입이 가능합니다.

추가정보

닉네임 영문 대 소문자, 숫자, 특수문자('_')중에서 입력하세요

전화번호 '-' 없이 숫자만으로 핸드폰 번호를 입력하세요

주소 우편번호

주소
 상세주소





lostPassword.html 화면

비밀번호 변경

아이디

[다음](#)

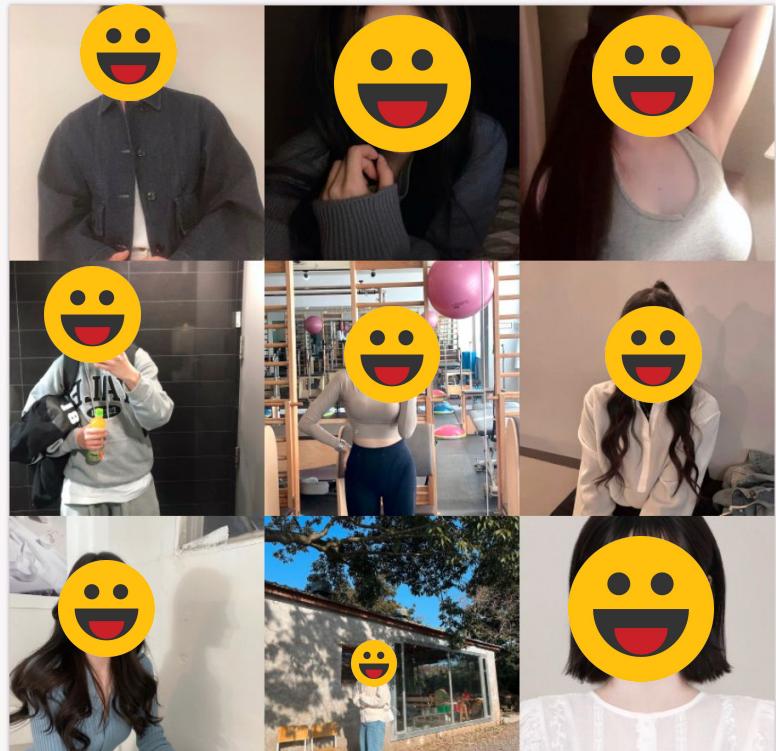




board/main.html 화면

POLAROID

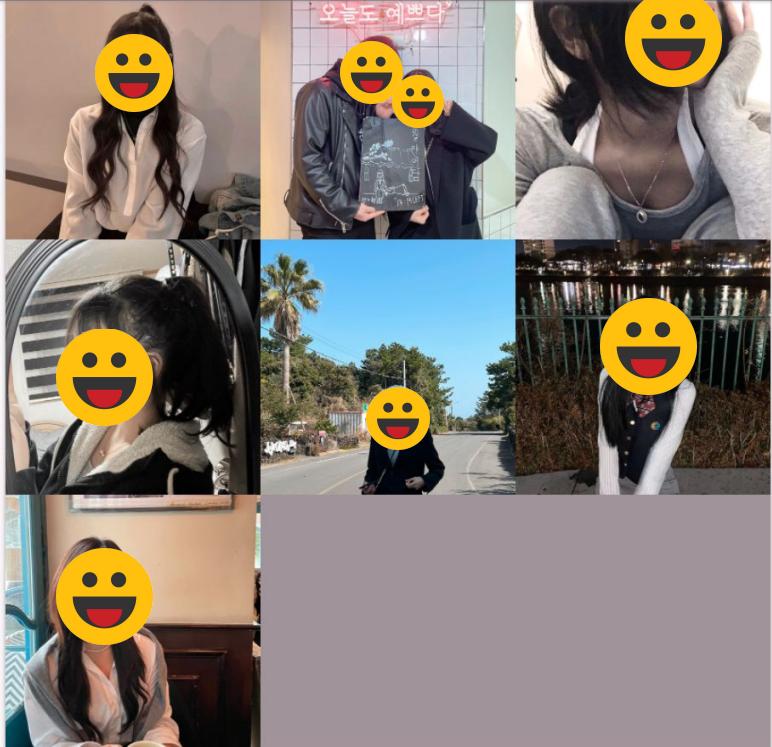
Search





board/main.html 화면

POLAROID

Search 

board/main.html 화면

SNS 메인페이지에서 무한스크롤을 사용하기 위해서 페이징과 스크립트를 이용하여 무한스크롤을 재현했습니다.

```
@GetMapping("main")
public @ResponseBody
Page<BoardPagingDTO> paging(Pageable pageable, Model model) {
    Page<BoardPagingDTO> boardList = bs.paging(pageable);

    System.out.println("boardList.getContent() = " + boardList.getContent()); // 요청 페이지에 들어있는 데이터, toString이 없기 때문에 주소값이 출력
    System.out.println("boardList.getTotalElements() = " + boardList.getTotalElements()); // 전체 글 갯수
    System.out.println("boardList.getNumber() = " + boardList.getNumber()); // JPA 기준 요청 페이지
    System.out.println("boardList.getTotalPages() = " + boardList.getTotalPages()); // 전체 페이지 갯수
    System.out.println("boardList.getSize() = " + boardList.getSize()); // 한 페이지에 보여지는 글 갯수
    System.out.println("boardList.hasPrevious() = " + boardList.hasPrevious()); // 이전 페이지 존재 여부
    System.out.println("boardList.isFirst() = " + boardList.isFirst()); // 첫 페이지인지 여부
    System.out.println("boardList.isLast() = " + boardList.isLast()); // 마지막 페이지인지 여부

    Long boardId = 1000000000000L;

    for(BoardPagingDTO b: boardList) {
        if(b.getBoardId() <= boardId) {
            boardId = b.getBoardId();
        }
    }

    System.out.println("boardId = " + boardId);

    model.addAttribute(attributeName: "boardId", boardId);

    return boardList;
}
```

board/main.html 화면

SNS 메인페이지에서 무한스크롤을 사용하기 위해서 페이지과 스크립트를 이용하여 무한스크롤을 재현했습니다.

```
<script th:inline="javascript">
/**/
let boardList = /*[${{boardList}}]*/
/**/
let
plus = 1
let boardId = 0
const getScrollTop = function () {
    return (window.pageYOffset !== undefined) ? window.pageYOffset : (document.documentElement || document.body.parentNode || document.body).scrollTop;
};
const getDocumentHeight = function () {
    const body = document.body;
    const html = document.documentElement;
    return Math.max(
        body.scrollHeight, body.offsetHeight, html.clientHeight, html.scrollHeight, html.offsetHeight
    );
}
window.onscroll = function (e) {
    if (plus < boardList.totalPages) {
        if (getScrollTop() === getDocumentHeight() - window.innerHeight) {

            setTimeout(function () {
                let addScroll = document.getElementById("addScroll")
                if (!boardList.last) {
                    // console.log("boardList : ",boardList)
                    plus += boardList.number + 1
                    // console.log(boardId)
                    const reqUrl = "/scroll/?page=" + plus
                    // console.log(reqUrl)
                    $.ajax({
                        type: "get",
                        url: reqUrl,
                        success: function (result) {

```



board/save.html 화면



SNS 게시글 작성할 때 #을 이용하여 해쉬태그 사용이 가능하며 사진은 5장까지 올릴 수 있도록 제한했습니다.

The screenshot shows a web application interface for creating a post. At the top, there is a dark header bar with the 'POLAROID' logo on the left, a search bar in the center, and user profile icons on the right. Below the header is a light gray content area with a central white box for post creation. The box has a title '게시글 작성' at the top, followed by a large red button with a white plus sign. Below the button is a text input field containing placeholder text: '내용을 입력하세요! ex) 즐거운 친구들과 #찐친 #우정사진 #2022'. At the bottom of the white box is a red '저장하기' (Save) button. On the far left of the main content area, there are three small circular icons: a circle with a dot, a circle with a horizontal line, and a star.

board/findById.html 화면

게시글을 클릭 시 나오는 화면으로 글쓴이의 프로필사진과 닉네임이 출력되고 사진 하단에 좋아요, 댓글 작성시간이 출력되며 화면 우측하단에는 글쓴이와 관리자만 보이는 수정,삭제 모든 유저가 볼 수 있는 글쓰기,최상단으로가기 이미지를 출력하여 편의성을 추가 했습니다.

POLAROID

Search

Auto

◎

☰

☆

EX.HTML



board/update.html 화면

게시글 수정페이지에서 내용수정이 가능합니다.

The screenshot shows a web browser window with the title 'board/update.html 화면'. The header includes the 'POLAROID' logo, a search bar, and user icons for messaging, likes, and profile. The main content area has a light gray background and features a centered form titled '게시글 수정'. It contains a text input field with placeholder text '내용을 입력하세요! ex) 즐거운 친구들과 #찐친 #우정사진 #2022' and a large red '수정하기' button at the bottom. On the far left, there are navigation icons: a circle with a dot, a square with a line, and a star.

POLAROID

Search

Message icon

Heart icon

User icon

게시글 수정

내용을 입력하세요! ex) 즐거운 친구들과 #찐친 #우정사진 #2022

수정하기

◎

☰

☆

EX.HTML



board/update.html 화면

The screenshot shows a web application interface for updating a post. At the top, there's a navigation bar with the 'POLAROID' logo on the left and three icons on the right: a speech bubble, a heart, and a smiley face. A modal window is open in the center, displaying a success message: 'localhost:8081 내용: 수정이 완료되었습니다.' (Content: Modification completed successfully.) with a blue '확인' (Confirm) button. Below the modal, the main content area has a title '게시글 수정' (Post Update) and a preview box containing the text '증명사진 #맞팔 #환경 #2022 #좋아요'. At the bottom of this area is a dark grey button labeled '수정하기' (Edit).





board/findById.html 화면

내용을 수정하고 좋아요기능 작동과 해시태그를 추가한 게시글입니다.

POLAROID

Auto

2022-03-26 13:40
증명사진! #맞팔 #환영 #2022 #좋아요
 1

board/findById.html 화면

게시글 작성시 #을붙인 글을 a태그로 변환해 해시태그 느낌을 주었고 해시태그 클릭시 서칭이 작동되게 제작 하였습니다.

```
<script>
    let content = document.getElementById("boardContents").value
    let splitedArray = content.split(' ');
    let linkedContent = '';
    for (let word in splitedArray) {
        word = splitedArray[word];
        if (word.indexOf('#') == 0) {
            let reqUrl = "/board/search/" + word.replace('#', "");
            word = '<a href="' + reqUrl + '">' + word + '</a>';
        }
        linkedContent += word + ' ';
    }
}
```

```
function tagSearch() {
    let keyword = document.getElementById("keyword").value
    location.href = "/board/search/" + keyword.replace("#", "")
}
```



board/search.html 화면



POLAROID

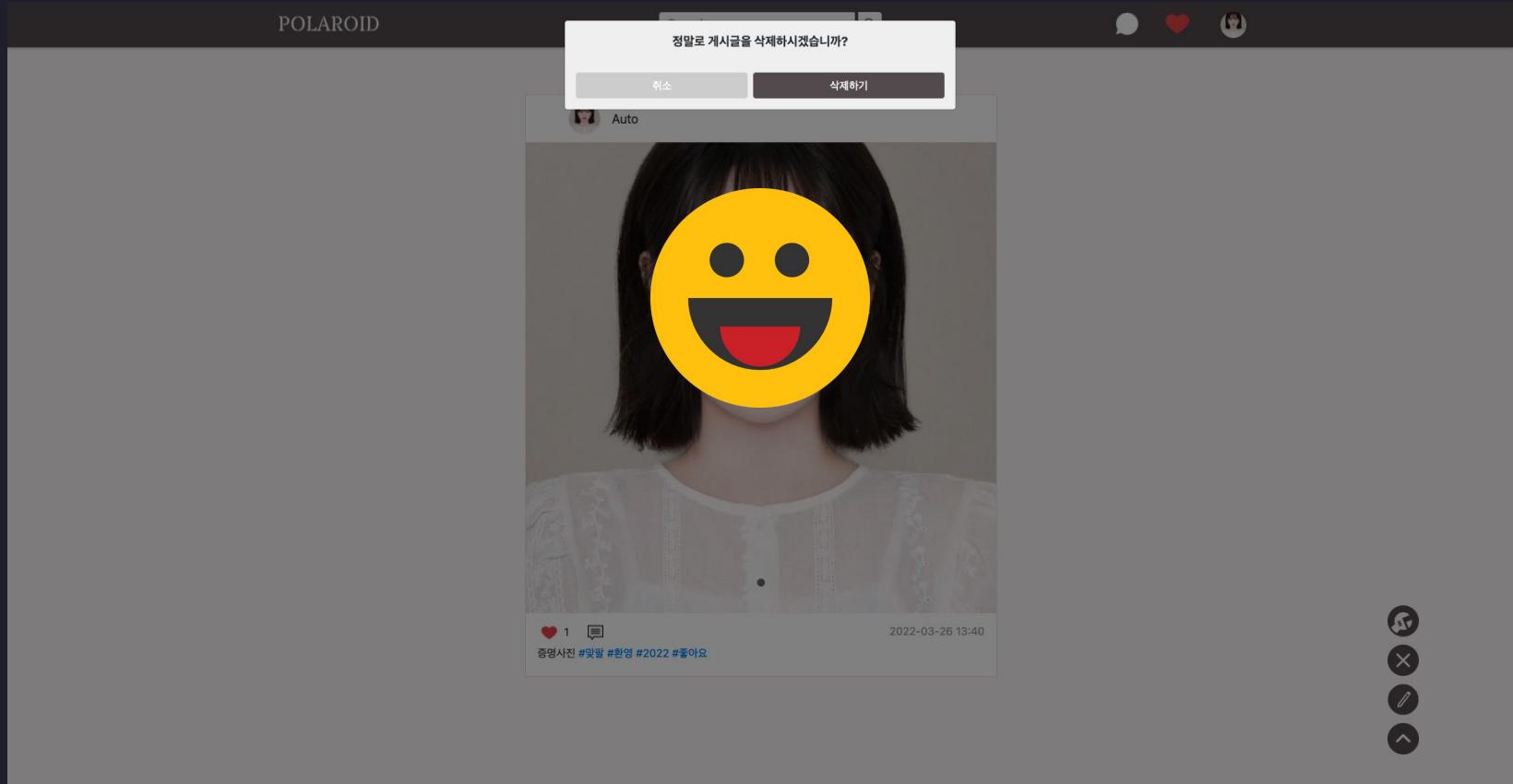
#2022 검색결과 : 2개



INDEX.HTML

게시글 삭제 작동 사진

게시글 삭제시 모달창을 통하여 한번더 확인후 삭제가 가능합니다.





board/update.html 화면

게시글 수정시 alert창 출력 후 수정이 완료됩니다.

The screenshot shows a web application interface for modifying a post. At the top, there's a navigation bar with the 'POLAROID' logo and social media icons (comment, heart, profile). A modal window is open, displaying an alert message: 'localhost:8081 내용: 수정이 완료되었습니다.' (Content: Modification completed.) with a blue '확인' (Confirm) button. Below the modal, the main content area has a title '게시글 수정' and a text input field containing the text '증명사진 #맞팔 #한정 #2022 #좋아요'. A dark grey '수정하기' (Update) button is at the bottom of this section. The overall design is clean with a white background and a dark header/footer.





POLAROID

다중 사진 출력 화면



POLAROID

Search

Auto

0
#냥이 2022-03-26 13:45



INDEX.HTML



POLAROID

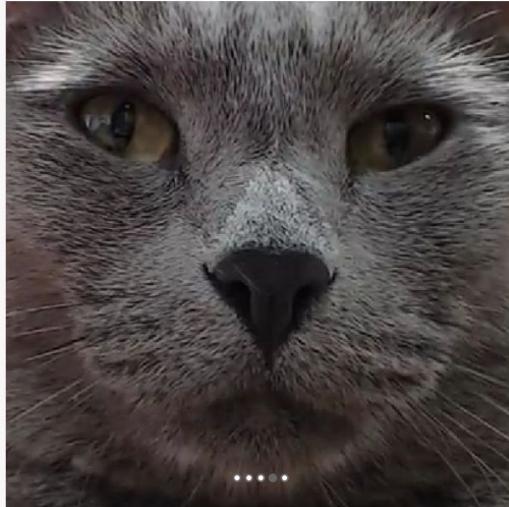
다중 사진 출력 화면



POLAROID

Search

Auto



...

0 2022-03-26 13:45
#냥이



INDEX.HTML



member/myPage.html 화면

Mypage 본인게시글도 무한스크롤로 확인이 가능합니다.





팔로우 시스템 구현 화면

타 유저의 SNS로 들어가서 팔로우가 가능하여 팔로우 수가 늘어납니다.

The screenshot shows a web application interface for a user named 'SemiAuto'. At the top, there is a navigation bar with the 'POLAROID' logo on the left and three icons on the right: a speech bubble, a heart, and a profile picture. A central modal window displays a success message: 'localhost:8081 내용: 팔로우 되었습니다.' with a blue '확인' button. Below the modal, the user's profile card features a yellow smiley face icon, the name 'SemiAuto', and social metrics: 0 follows, 1 following, and 0 posts. Two buttons, '팔로우' and '채팅', are visible at the bottom of the profile card. The background of the page is white.





팔로우 시스템 구현 화면

본인을 팔로우한 유저의 리스트를 볼 수 있습니다.

POLAROID

Search

팔로워 팔로잉 게시물
1 0 32

팔로우 취소 채팅

팔로워

SemiAuto

◎

☰

☆

EX.HTML



Mypage 기능 구현 코드

마이페이지에서 각종 기능들을 구현하는 코드입니다. 팔로우, 팔로잉, 게시글수, 내게시물, 좋아요목록, 상품찜목록 등이 있습니다.

```
@GetMapping("{{mypage}}")
public String mypageForm(@AuthenticationPrincipal PrincipalDetails principalDetails,
                        Model model){
    model.addAttribute( attributeName: "status", attributeValue: false);
    // 알림 처리
    notice(principalDetails.getMember().getId());
    MemberEntity member=memberService.findById(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "member", member);
    // 팔로우 팔로잉 수
    ArrayList<Integer> followCount=followService.followCount(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "follower", followCount.get(0));
    model.addAttribute( attributeName: "following", followCount.get(1));
    // 팔로잉 목록
    List<FollowEntity> following = followService.followingList(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "followingList", following);
    // 팔로워 목록
    List<FollowEntity> follower = followService.followerList(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "followerList", follower);
    // 게시글 수
    List<BoardEntity> boardCount=boardService.boardCount(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "boardCount", boardCount.size());
    // 내 게시글
    List<BoardDetailDTO> boardList=boardService.myPage(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "boardList", boardList);
    // hsq 3.13 추가 좋아요 목록
    List<BoardDetailDTO> likeList=boardService.likeList(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "likeList", likeList);
    // hsw 3.13 추가 찜 목록
    List<GoodsDetailDTO> pickList=goodsService.pickList(principalDetails.getMember().getId());
    model.addAttribute( attributeName: "pickList", pickList);

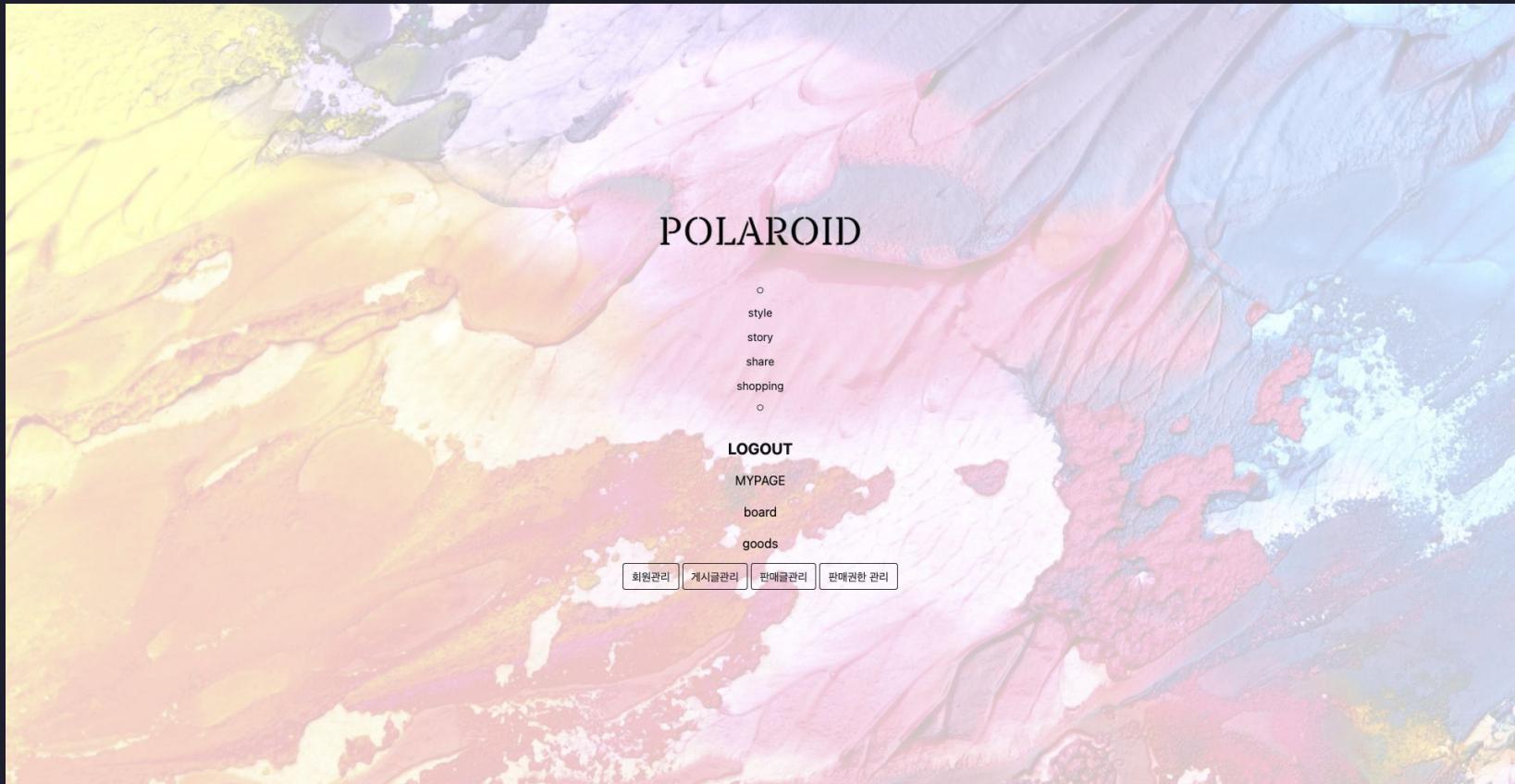
    System.out.println("MemberController.mypageForm");
    return "member/mypage";
```





Index.html (admin) 화면

관리자 index화면 입니다. 하단에 관리자전용 페이지로 이동할 수 있으며 회원, 게시글, 판매글, 판매권한부여를 관리할 수 있습니다.



admin/sellerList.html 화면

판매자 요청을 한 유저의 리스트가 출력됩니다. 권한부여를 통하여 판매자로 지정 하실 수 있습니다.

판매자 권한 신청 번호	프로필	닉네임	권한 부여
1		SemiAuto	<button>권한 부여</button>



판매자신청 코드

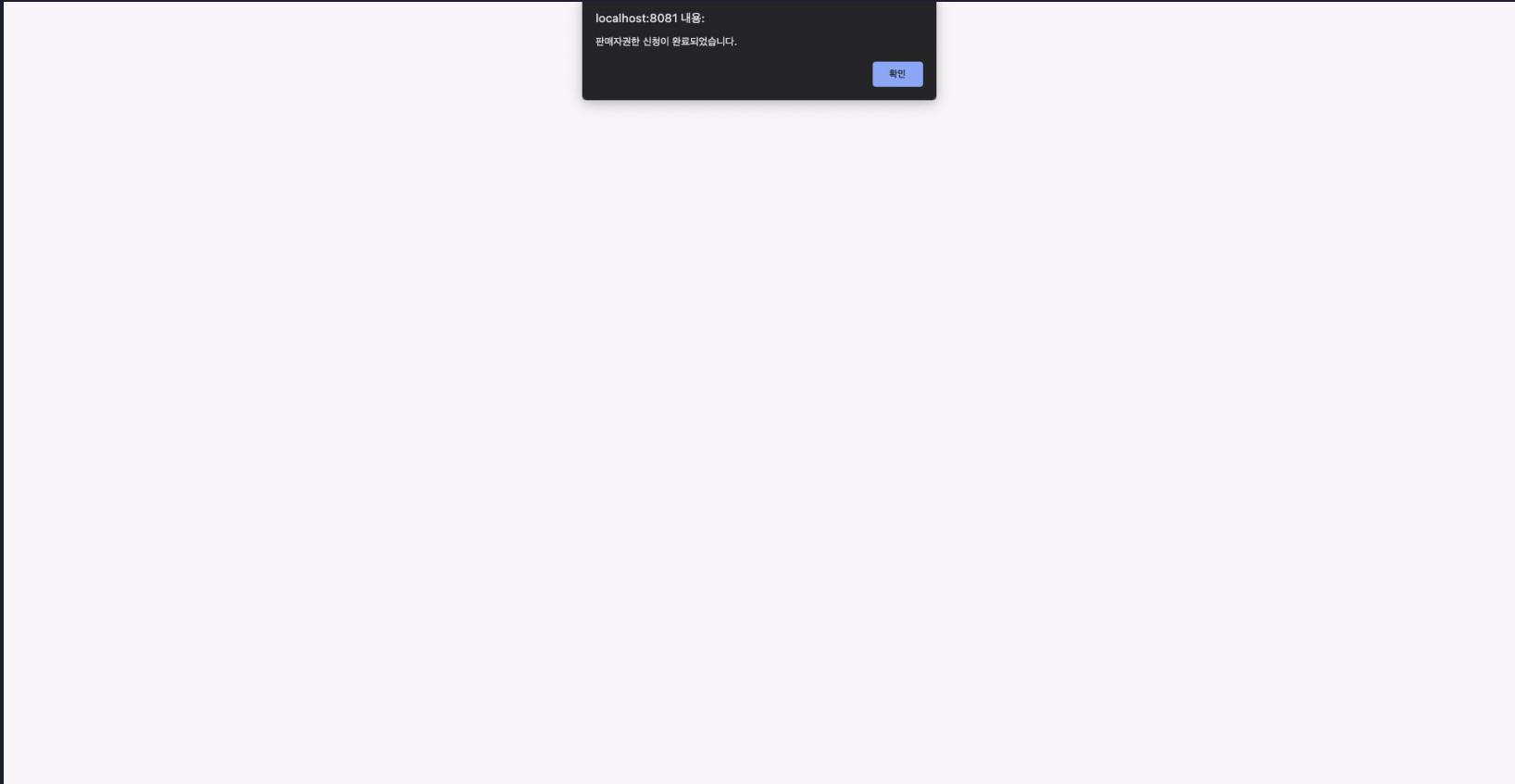
판매자 권한신청을 통해서 관리자페이지로 넘어가고 관리자는 판매자 권한부여를 할 수 있습니다.

```
// 판매자 권한신청
@GetMapping("/sellerRole")
public @ResponseBody String sellerRole(@AuthenticationPrincipal PrincipalDetails principalDetails){
    System.out.println("MemberController.sellerRole");
    String duplicate=sellerRoleService.save(principalDetails.getMember().getId());
    String result=null;
    if(duplicate=="ok"){
        result=<script>alert('판매자권한 신청이 완료되었습니다.');//location.href='mypage'</script>;
    }
    else {
        result=<script>alert('이미 판매자권한을 신청 하셨습니다.');//location.href='mypage'</script>;
    }
    return result;
}
```

```
//판매자 권한부여
@GetMapping("/admin/giveRole/{memberId}")
public String giveRole(@PathVariable Long memberId){
    sellerRoleService.giveRole(memberId);
    return "redirect:/admin/sellerList";
}
```



판매자 권한 부여완료 화면





POLAROID

판매자 권한이 있는 사용자 화면



판매 권한이 생긴 유저는 닉네임 옆에 리본이 생깁니다.

The screenshot shows a user profile for 'SemiAuto' (nickname). The profile picture is a yellow smiley face. The user has 1 follower, 1 following, and 0 posts. There are three buttons at the bottom: '팔로우 취소' (Unfollow), '채팅' (Chat), and '상점' (Shop). The top navigation bar includes the 'POLAROID' logo, a search bar, and social media icons for messaging, heart, and profile.

POLAROID

SemiAuto

팔로워 1 팔로잉 1 게시물 0

팔로우 취소 채팅 상점

EX.HTML





admin/memberList.html 화면

회원 번호	프로필 사진	회원 닉네임	회원 이메일	삭제
3		Autowried	[REDACTED]com_kakao	<button>회원 삭제</button>
2		SemiAuto	[REDACTED]com_naver	<button>회원 삭제</button>
1		Auto	[REDACTED]com_google	<button>회원 삭제</button>



admin/boardList.html 화면

게시글 번호	대표 사진	작성자	작성자 이메일	상세 조회
32		Auto	[REDACTED]com_google	<button>상세 조회</button>
31		Auto	[REDACTED]com_google	<button>상세 조회</button>
30		Auto	[REDACTED]com_google	<button>상세 조회</button>
29		Auto	[REDACTED]com_google	<button>상세 조회</button>
28		Auto	[REDACTED]com_google	<button>상세 조회</button>
27		Auto	[REDACTED]com_google	<button>상세 조회</button>
26		Auto	[REDACTED]com_google	<button>상세 조회</button>
25		Auto	[REDACTED]com_google	<button>상세 조회</button>
24		Auto	[REDACTED]com_google	<button>상세 조회</button>
23		Auto	[REDACTED]com_google	<button>상세 조회</button>





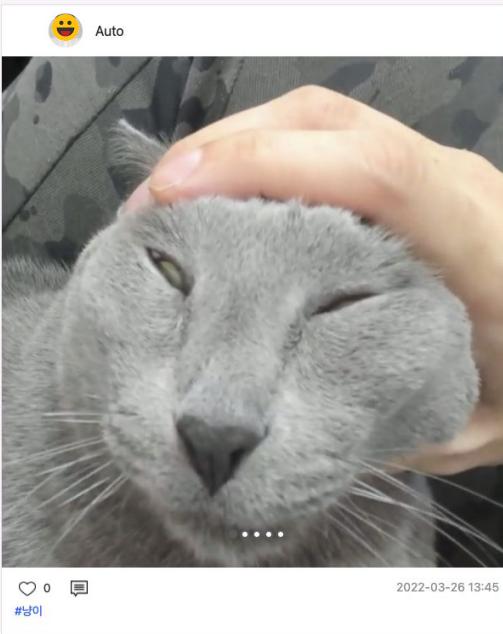
POLAROID

Admin 게시글 삭제 (좌측하단) 화면



POLAROID

Search



INDEX.HTML

admin/goodsList.html 화면

판매글 번호	대표 사진	판매자	판매자 이메일	상품명	가격	상세 조회
17		SemiAuto	ohjs813@naver.com_naver	코트	120000	<button>상세 조회</button>
16		SemiAuto	ohjs813@naver.com_naver	장미꽃 티셔츠	33000	<button>상세 조회</button>
15		SemiAuto	ohjs813@naver.com_naver	마우이 티셔츠	25000	<button>상세 조회</button>
14		SemiAuto	ohjs813@naver.com_naver	요가곰 티셔츠	20000	<button>상세 조회</button>
13		SemiAuto	ohjs813@naver.com_naver	네서널 지오그래픽 티셔츠	66000	<button>상세 조회</button>
12		SemiAuto	ohjs813@naver.com_naver	캘빈클라인 티셔츠	75000	<button>상세 조회</button>
11		SemiAuto	ohjs813@naver.com_naver	체크 셔츠	55000	<button>상세 조회</button>
10		SemiAuto	ohjs813@naver.com_naver	화이트 셔츠	65000	<button>상세 조회</button>
9		SemiAuto	ohjs813@naver.com_naver	헤링본 셔츠	70000	<button>상세 조회</button>
8		SemiAuto	ohjs813@naver.com_naver	에일 셔츠	80000	<button>상세 조회</button>



POLAROID

Admin 판매글 삭제 (좌측하단) 화면



POLAROID

제목

Search



상품 내용



수량

- 1 +

구매하기 ⭐



INDEX.HTML

goods/paging.html 화면

POLAROID

제목 Search

CHAT HEART PROFILE

Goods' List

코트

★ 0 ⚠ 0

남은재고 5개

SemiAuto 120000원

장미꽃 티셔츠

★ 0 ⚠ 0

남은재고 8개

SemiAuto 33000원

마우이 티셔츠

★ 0 ⚠ 0

남은재고 9개

SemiAuto 25000원

요가곰 티셔츠

★ 0 ⚠ 0

남은재고 15개

SemiAuto 20000원

INDEX.HTML

goods/paging.html 화면

POLAROID

제목 Search

言论  

 SemiAuto 70000원  크로스백 ★ 0 ⚡ 0 남은재고 15개	 SemiAuto 80000원  선글라스 ★ 0 ⚡ 0 남은재고 17개	 SemiAuto 43000원  니트 ★ 0 ⚡ 0 남은재고 30개	 SemiAuto 34000원  청자켓 ★ 0 ⚡ 0 남은재고 40개
 SemiAuto 30000원	 SemiAuto 125000원	 SemiAuto 45000원	 SemiAuto 60000원

First < 1 2 > Last

goods/paging.html 화면

쇼핑몰페이지의 기본화면을 구성해주는 페이징코드입니다. 모델을 통해 기본적인 정보들을 모두 가져옵니다.

```
// 굿즈보드 페이징
@GetMapping("goods/paging")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String paging(@PageableDefault(page = 1) Pageable pageable, Model model, HttpSession session) {

    // 헤더
    Long memberId = (Long)session.getAttribute( name: "LoginNumber");
    mc.notice(memberId);
    model.addAttribute( attributeName: "member", ms.findById(memberId));

    Page<GoodsPagingDTO> goodsList = gs.paging(pageable);
    model.addAttribute( attributeName: "goodsList", goodsList);
    model.addAttribute( attributeName: "memberId", memberId);
    // 삼항연산자로 바꿈
    int startPage = (((int) (Math.ceil((double) pageable.getPageNumber() / PagingConstGoods.BLOCK_LIMIT))) - 1) * PagingConstGoods.BLOCK_LIMIT + 1;
    int endPage = ((startPage + PagingConstGoods.BLOCK_LIMIT - 1)
        < goodsList.getTotalPages() ? startPage + PagingConstGoods.BLOCK_LIMIT - 1 : goodsList.getTotalPages());
    model.addAttribute( attributeName: "startPage", startPage);
    model.addAttribute( attributeName: "endPage", endPage);
    return "goods/paging";
}
```





goods/save.html 화면

POLAROID

제목

Search



상품 등록

+

작성자

SemiAuto

제목

가격

내용

제고

상품 정보

상품정보를 입력하세요

저장하기



goods/save.html 화면

게시글 작성은 담당하는 코드입니다. Spring Security를 사용하여 판매자 권한이 부여된 유저와 관리자만 작성이 가능합니다.
판매자의 정보값을 받아와서 판매글쓰기가 가능합니다.

```
// 굿즈보드 작성페이지
@GetMapping("goods/save")
@PreAuthorize("hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String saveForm(@AuthenticationPrincipal PrincipalDetails principalDetails, Model model, HttpSession session) {
    model.addAttribute(attributeName: "member", principalDetails.getMember());
    Long memberId = (Long) session.getAttribute(name: "LoginNumber");
    mc.notice(memberId);
    model.addAttribute(attributeName: "memberId", memberId);
    model.addAttribute(attributeName: "member", ms.findById(memberId));
    return "goods/save";
}
```

```
// 굿즈보드 작성
@PostMapping("goods/save")
@PreAuthorize("hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String save(@ModelAttribute GoodsSaveDTO goodsSaveDTO, HttpSession session, Model model) throws IOException {
    Long goodsId = gs.save(goodsSaveDTO);
    Long memberId = (Long) session.getAttribute(name: "LoginNumber");
    model.addAttribute(attributeName: "member", ms.findById(memberId));
    for (MultipartFile g: goodsSaveDTO.getGoodsFile()) {
        gs.sendFile(goodsId, g);
    }
    return "redirect:/goods/";
}
```





goods/myList.html 화면

POLAROID

제목

Search



SemiAuto `s ShoppingMall List

글쓰기

상품 사진	상품 이름	가격	조회수	게시일시	세부내용보기
코트	120000원	0	2022-03-27 13:23	☰	
장미꽃 티셔츠	33000원	1	2022-03-27 13:21	☰	
마우이 티셔츠	25000원	0	2022-03-27 13:21	☰	
요가곰 티셔츠	20000원	0	2022-03-27 13:20	☰	
네서널 지오그래픽 티셔츠	66000원	0	2022-03-27 13:17	☰	
캠비클라인 티셔츠	75000원	0	2022-03-27 13:15	☰	



goods/myList.html 화면

본인게시글 리스트를 출력해주는 코드입니다. 상품사진, 상품이름, 가격, 조회수 등이 출력이 됩니다.

```
// 굿즈보드 본인게시글 확인
@GetMapping("list/{memberId}")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String myList (@PathVariable Long memberId,
                     @ModelAttribute GoodsDetailDTO goodsDetailDTO, Model model,
                     @PageableDefault(page = 1) Pageable pageable, HttpSession session) {
    Page<GoodsPagingDTO> goodsDetailDTOList = gs.list(memberId, pageable);
    Long memberIdH = (Long)session.getAttribute("LoginNumber");
    mc.notice(memberIdH);

    String memberNickname = mr.findById(memberId).get().getMemberNickname();
    model.addAttribute("member", ms.findById(memberId));
    model.addAttribute("goodsList", goodsDetailDTOList);
    model.addAttribute("memberNickname", memberNickname);

    int startPage = (((int) (Math.ceil((double) pageable.getPageNumber() / PagingConstGoods.BLOCK_LIMIT))) - 1) * PagingConstGoods.BLOCK_LIMIT + 1;
    int endPage = ((startPage + PagingConstGoods.BLOCK_LIMIT - 1) <
                  goodsDetailDTOList.getTotalPages()) ? startPage + PagingConstGoods.BLOCK_LIMIT + 1 : goodsDetailDTOList.getTotalPages();

    model.addAttribute("startPage", startPage);
    model.addAttribute("endPage", endPage);

    model.addAttribute("memberId", memberId);
    return "/goods/myList";
}
```



goods/findById.html 화면

POLAROID

제목

Search

상품 내용

수량

- 1 +

구매하기



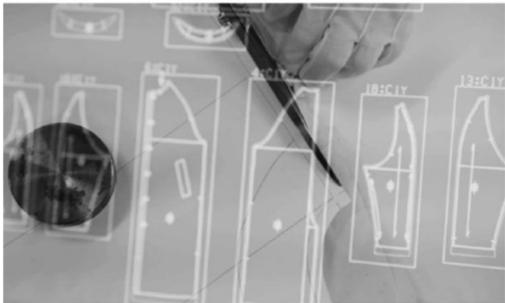


goods/findById.html 화면

POLAROID

제목 Search

서비스는 드립워크에서 제공 됩니다. 디자인에서 꼽고자기 디자이너가 꼼꼼하게 점검합니다.



수량
- 1 +

구매하기 ☆

상품정보 상품후기

상품 정보

상품 이름	화이트 세츠
상세 설명	화이트 세츠
가격	65000원
수량	10





goods/findById.html 화면

굿즈 상세조회페이지에서는 Spring Security로 기본유저,관리자,판매자가 조회가 가능합니다.
그리고 게시글에 들어 왔을때 조회수가 증가합니다.

```
// 굿즈보드 상세조회
@GetMapping("/{goodsId}")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String findById(@PathVariable("goodsId") Long goodsId, Model model, HttpSession session) {
    GoodsDetailDTO goods = gs.findById(goodsId);

    // 헤더
    Long memberId = (Long) session.getAttribute("LoginNumber");
    mc.notice(memberId);
    model.addAttribute("member", ms.findById(memberId));

    model.addAttribute("memberId", memberId);
    model.addAttribute("member", ms.findById(memberId));

    model.addAttribute("goods", goods);
    model.addAttribute("goodsCommentList", goods.getGoodsCommentList());

    model.addAttribute("imageSize", goods.getGoodsPhoto().size());

    int likeStatus = gs.findLike(goodsId, memberId);
    model.addAttribute("like", likeStatus);
    System.out.println("likeStatus = " + likeStatus);

    gs.viewUp(goodsId);

    return "goods/findById";
}
```





goods/findById.html 화면

POLAROID

제목 Search

시리즈코드는 스마트폰에서 직접 활용합니다. 디자인에서 끌고와서
디자이너가 편하게 참조합니다.

pay

QR결제 카톡결제

휴대폰으로 스캔하면
결제 화면으로 이동합니다.
스마트폰 카메라 및 모든 QR스캐너로 가능

상품정보 상품후기

상품 이름 상세 설명

가격 65000원

수량 10

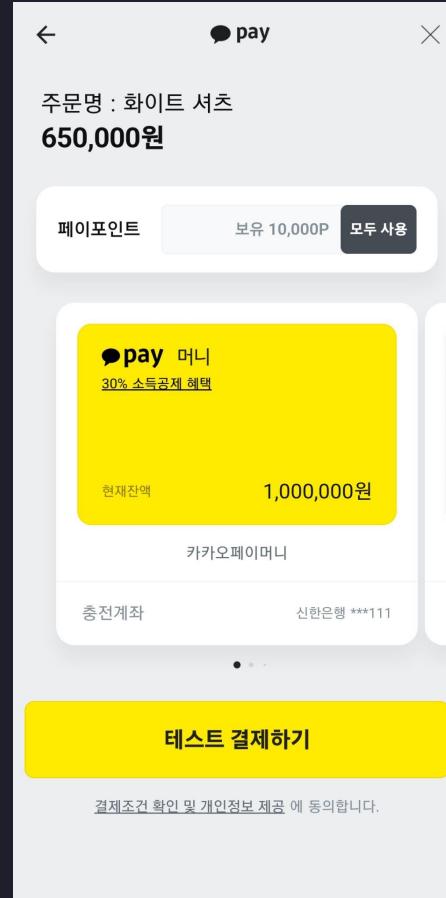
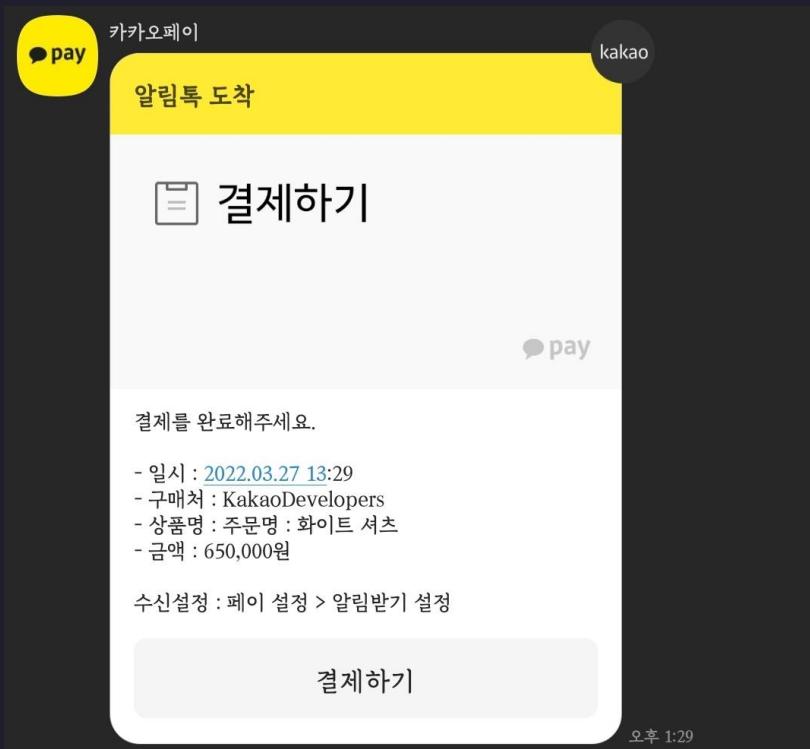
구매하기 ☆

The image shows a screenshot of a web browser displaying a product detail page for a Polaroid item. A yellow modal window titled 'pay' is overlaid on the page, containing a QR code for QR payment. The background page includes a search bar, a message about using the series code, and a product information section with tabs for '상품정보' and '상품후기'. Below this are fields for '상품 이름' and '상세 설명', and at the bottom are sections for '가격' (65000원) and '수량' (10). To the right of the modal, there is a button labeled '구매하기' (Buy) and a star icon. The overall theme is a mobile-friendly e-commerce interface.





Kakaopay 결제 화면





결제 성공 화면

POLAROID

localhost:8081 내용:
결제가 완료되었습니다. 결제 금액 : 650000

확인



수량
- 10 +

구매하기 ★

상품정보 상품후기

상품 정보

상품 이름	화이트 세츠
상세 설명	화이트 세츠
가격	65000원
수량	10





goods/paySuccess.html 화면

POLAROID

제목



Search

**Autowried**님의 영수증

구매하신 물품 : 화이트 셔츠

구매하신 물품 가격: 65000원

구매하신 물품 수량: 10개

구매하신 물품 총액: 650000원

[돌아가기](#)

goods/paySuccess.html 화면

결제가 완료되면 구매한 제품, 가격, 수량, 총액이 출력이 되어 본인이 결제한 영수증을 확인 할 수 있습니다.

```
// 결제 완료 창
@GetMapping("paySuccess/{goodsId}")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String paySuccess(@PathVariable Long goodsId, HttpSession session, Model model) {
    Long memberId = (Long) session.getAttribute("LoginNumber");
    mc.notice(memberId);
    model.addAttribute("member", ms.findById(memberId));
    PayDetailDTO payDetailDTO = gs.payFind(goodsId, memberId);
    model.addAttribute("pay", payDetailDTO);
    return "goods/paySuccess";
}
```



결제는 i'm Port 와 카카오페이지 API를 사용하였습니다.

```
// 결제
@PostMapping("pay")
public @ResponseBody String pay(@RequestParam ("goodsId") Long goodsId, @RequestParam("memberId") Long memberId,
                                @RequestParam ("count") int count, Model model, HttpSession session) {
    Long memberIdH = (Long)session.getAttribute( name: "LoginNumber");
    mc.notice(memberIdH);
    model.addAttribute( attributeName: "goodsId", goodsId);
    model.addAttribute( attributeName: "memberId", memberId);
    model.addAttribute( attributeName: "member", ms.findById(memberId));
    gs.pay(goodsId, count);
    gs.paySuccess(goodsId, memberId, count);
    return "ok";
}

// 결제 완료 창
@GetMapping("paySuccess/{goodsId}")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String paySuccess(@PathVariable Long goodsId, HttpSession session, Model model) {
    Long memberId = (Long) session.getAttribute( name: "LoginNumber");
    mc.notice(memberId);
    model.addAttribute( attributeName: "member", ms.findById(memberId));
    PayDetailDTO payDetailDTO = gs.payFind(goodsId, memberId);
    model.addAttribute( attributeName: "pay", payDetailDTO);
    return "goods/paySuccess";
}
```

결제에 성공하면 DB에 저장되는 Service입니다. 본인이 결제한 리스트도 볼 수 있습니다.

```
// 결제 성공 정보 저장
@Override
public void paySuccess(Long goodsId, Long memberId, int count) {
    MemberEntity memberEntity = mr.findById(memberId).get();
    GoodsEntity goodsEntity = gr.findById(goodsId).get();
    PayEntity payEntity = PayEntity.toPayEntity(memberEntity, goodsEntity, count);
    pr.save(payEntity);
}

// 결제정보
@Override
public PayDetailDTO payFind(Long goodsId, Long memberId) {
    GoodsEntity goodsEntity = gr.findById(goodsId).get();
    MemberEntity memberEntity = mr.findById(memberId).get();
    PayEntity payEntity = pr.findTop1ByGoodsEntityAndMemberEntityOrderByIdDesc(goodsEntity, memberEntity);
    PayDetailDTO payDetailDTO = PayDetailDTO.toPayDetailDTO(payEntity);
    return payDetailDTO;
}

// 결제 리스트
@Override
public List<PayDetailDTO> payList(Long memberId) {
    MemberEntity memberEntity = mr.findById(memberId).get();
    List<PayEntity> payEntityList = pr.findAllByMemberEntityOrderByIdDesc(memberEntity);
    List<PayDetailDTO> payDetailDTOList = PayDetailDTO.toPayDetailDTOList(payEntityList);
    return payDetailDTOList;
}
```



goods/findById.html 화면

POLAROID

localhost:8081 내용:
제고가 부족합니다.

확인



수량
- 11 +

구매하기 ★

상품정보 상품후기

상품 정보

상품 이름	화이트 서초
상세 설명	화이트 서초
가격	65000원
수량	10



주문수량과 가격등 체크후 결제가 진행되며 재고가 없을시 재고가 부족하다는 알림창이 뜹니다.

```
if(countStock>=0) {
    const IMP = window.IMP; // 생략 가능
    IMP.init('imp59250378');
    // import 관리자 페이지 -> 내정보 -> 가맹점식별코드
    // ''안에 띄어쓰기 없이 가맹점 식별코드를 붙여넣어주세요. 안그러면 결제창이 안뜹니다.
    IMP.request_pay({
        pg: 'kakao',
        pay_method: 'card',
        merchant_uid: 'merchant_' + new Date().getTime(),
        /*
        * merchant_uid에 경우
        * https://docs.iamport.kr/implementation/payment
        * 위에 url에 따라가시면 넣을 수 있는 방법이 있습니다.
        */
    })
}
```

```
name: '주문명 : ' + goodsInfo.goodsTitle,
// 결제창에서 보여질 이름
// name: '주문명 : ${auction.a_title}',
// 위와같이 model에 담은 정보를 넣어 쓸수도 있습니다.
amount: goodsInfo.goodsPrice * count,
// amount: ${bid.b_bid},
// 가격
buyer_name: 'memberNickname',
// 구매자 이름, 구매자 정보도 model값으로 바꿀 수 있습니다.
// 구매자 정보에 여러가지로 있으므로, 자세한 내용은 맨 위 링크를 참고해주세요.
buyer_postcode: '123-456',
```

```
function (rsp) {
    console.log(rsp);
    if (rsp.success) {
        let msg = '결제가 완료되었습니다.';
        msg += '결제 금액 : ' + rsp.paid_amount;
        alert(msg);
        // success.submit();
        // 결제 성공 시 정보를 넘겨줘야한다면 body에 form을 만든 뒤 위의 코드를 사용하는 방법이 있습니다.
        // 자세한 설명은 구글링으로 보시는게 좋습니다.
        $.ajax({
            type: 'post',
            url: '/goods/pay',
            data: {
                "count": count,
                "goodsId": goodsId,
                "memberId": memberId
            },
            success: function () {
                location.href="/goods/paySuccess/" + goodsId;
            },
            error: function () {
                console.log("ajax 오류")
            }
        });
    } else {
        let msg = '결제에 실패하였습니다.';
        msg += '에러내용 : ' + rsp.error_msg;
        alert(msg)
    }
}
else {
    alert("재고가 부족합니다.")
}
```



goods/payList.html 화면

POLAROID

제목 Search

言论 爱心 微笑

Autowried's Purchase List

제품	상품명	결제수량	결제금액	총결제금액
	화이트 셔츠	10개	65000원	650000원



goods/payList.html 화면

결제한 리스트를 확인하는 코드입니다. 제품사진과 제품명 수량 가격등이 출력이 됩니다.

```
// 본인 결제리스트
@GetMapping("payList")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String payList(HttpSession session, Model model) {
    Long memberId = (Long) session.getAttribute("LoginNumber");
    mc.notice(memberId);
    MemberEntity memberEntity = ms.findById(memberId);
    model.addAttribute("member", ms.findById(memberId));
    List<PayDetailDTO> payDetailDTOList = gs.payList(memberId);
    model.addAttribute("payList", payDetailDTOList);
    model.addAttribute("member", memberEntity);
    return "goods/payList";
}
```





재고 없을시 품절 화면

POLAROID

제목 Search



남은재고 5개

SemiAuto 120000원



네셔널 지오그래픽 티셔츠

남은재고 10개

SemiAuto 66000원

남은재고 8개

SemiAuto 33000원



켈빈클라인 티셔츠

남은재고 7개

SemiAuto 75000원

남은재고 9개

SemiAuto 25000원



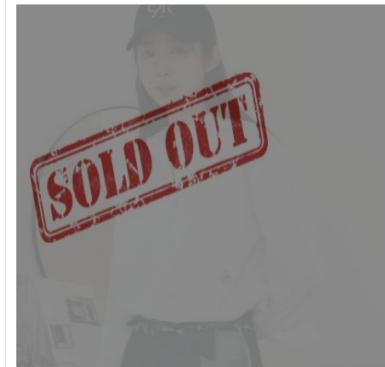
체크 셔츠

남은재고 10개

SemiAuto 55000원

남은재고 15개

SemiAuto 20000원



화이트 셔츠

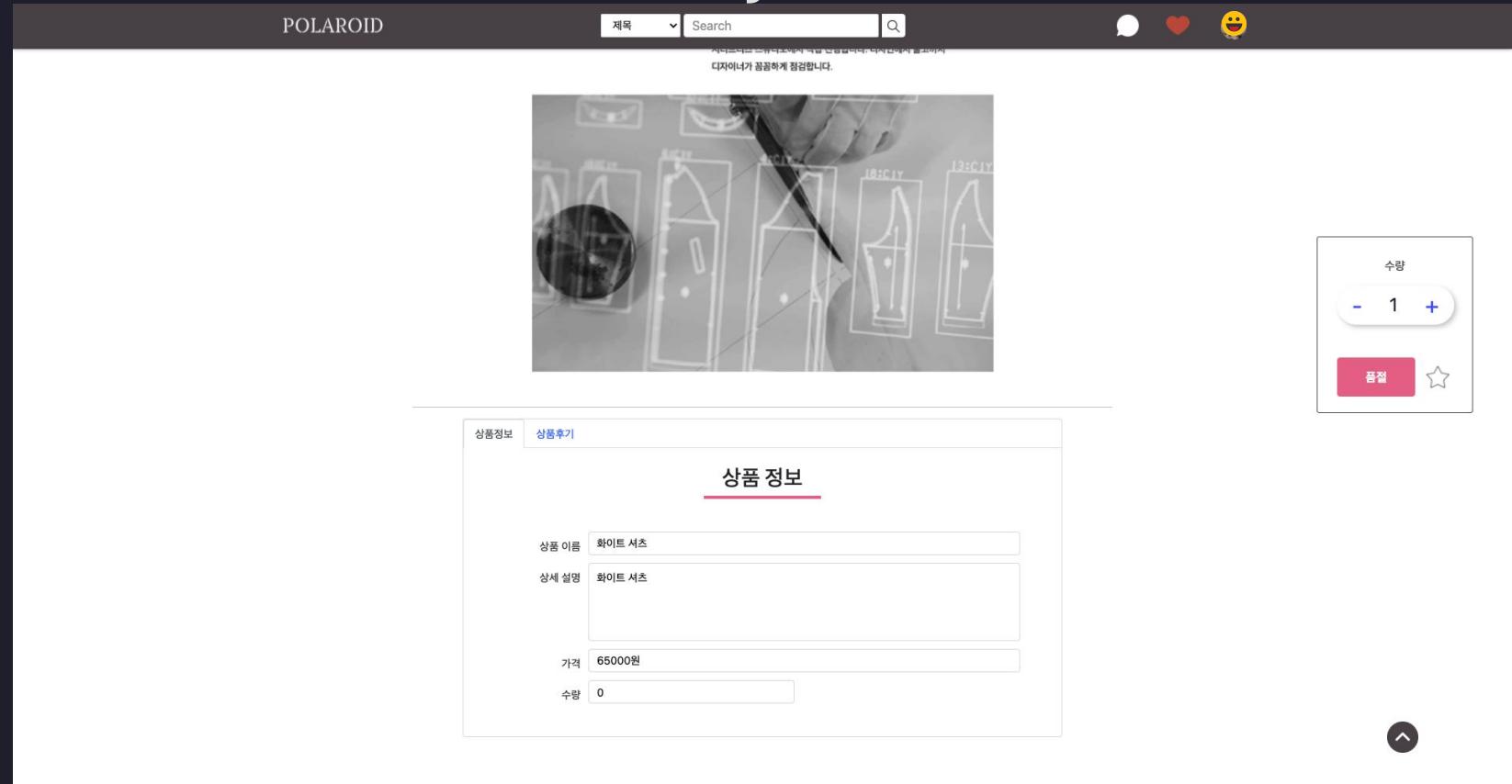
남은재고 0개

SemiAuto 65000원



PetCare







goods/update.html 화면

POLAROID

localhost:8081 내용:
수정이 완료되었습니다.

상품 수정

작성자: SemiAuto

글번호: 10

제목: 화이트 서초

가격: 65000

내용: 화이트 서초

재고: 30

상세 설명: 화이트 서초

수정하기



goods/update.html 화면

수정을 통해서 재고와 내용 등을 수정이 가능합니다.

```
// 업데이트폼
@GetMapping("update/{goodsId}")
@PreAuthorize("hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String updateForm(@PathVariable Long goodsId, Model model, HttpSession session) {
    GoodsDetailDTO goodsDetailDTO = gs.findById(goodsId);
    Long memberId = (Long) session.getAttribute("LoginNumber");
    mc.notice(memberId);
    model.addAttribute("member", ms.findById(memberId));
    System.out.println("goodsDetailDTO.getMemberId() = " + goodsDetailDTO.getMemberId());
    model.addAttribute("goods", goodsDetailDTO);
    return "goods/update";
}

// 글 업데이트
◎ @PutMapping("{goodsId}")
public ResponseEntity update(@RequestBody GoodsUpdateDTO goodsUpdateDTO) {
    System.out.println("goodsUpdateDTO = " + goodsUpdateDTO);
    gs.update(goodsUpdateDTO);
    return new ResponseEntity(HttpStatus.OK);
}
```



goods/search.html 화면

POLAROID

제목 후드

검색 결과



검정후드

 koyoun 50000원



후드

 koyoun 58000원

First < 1 > Last



goods/search.html 화면

POLAROID

작성자

검색 결과



검정후드

 koyoun 50000원



자켓

 koyoun 60000원



후드

 koyoun 58000원

First < 1 > Last

goods/search.html 화면

서칭코드로 글쓴이, 게시글제목으로 검색이 가능합니다. 검색한 내용은 페이지화되어 출력이 됩니다.

```
@GetMapping("search")
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
public String search(@PageableDefault(page =1) Pageable pageable,
                     @ModelAttribute GoodsSearchDTO goodsSearchDTO,
                     Model model, HttpSession session) {
    // 헤더
    Long memberId = (Long)session.getAttribute( name: "LoginNumber");
    if(memberId != null) {
        mc.notice(memberId);
    }

    model.addAttribute( attributeName: "memberId", memberId);
    model.addAttribute( attributeName: "member", ms.findById(memberId));
    Page<GoodsPagingDTO> goodsDetailDTOList = gs.search(goodsSearchDTO, pageable);
    model.addAttribute( attributeName: "goodsList", goodsDetailDTOList);
    int startPage = (((int) (Math.ceil((double) pageable.getPageNumber() / PagingConstGoods.BLOCK_LIMIT))) - 1) * PagingConstGoods.BLOCK_LIMIT + 1;
    int endPage = ((startPage + PagingConstGoods.BLOCK_LIMIT - 1) <
                  goodsDetailDTOList.getTotalPages()) ? startPage + PagingConstGoods.BLOCK_LIMIT - 1 : goodsDetailDTOList.getTotalPages();
    model.addAttribute( attributeName: "startPage", startPage);
    model.addAttribute( attributeName: "endPage", endPage);
    model.addAttribute( attributeName: "select", goodsSearchDTO.getSelect());
    model.addAttribute( attributeName: "search", goodsSearchDTO.getSearch());
    return "goods/search";
}
```



실시간 채팅 화면 화면

POLAROID

Search

POLAROID

localhost:8081/personalChat/2

SemiAuto

Autowried

너무 이뻐요!

방금

메시지를 입력하세요

전송

팔로우 채팅 상점

SemiAuto

팔로워 1 팔로잉 1 게시물 0





POLAROID

안읽은 채팅이 있을 시 헤더 화면



The screenshot shows a user profile header for 'SemiAuto'. The profile picture is a yellow smiley face. The username 'SemiAuto' is displayed next to it, with a small red notification badge. Below the profile are four gray buttons labeled '정보수정', '좋아요', '찜', and '글작성'. To the right of the profile are three icons: a speech bubble with a red '1', a heart, and a smiley face.

POLAROID

Search

SemiAuto 1

팔로워 1
팔로잉 1
게시물 0

정보수정 좋아요 찜 글작성 상점

◎

☰

☆

^

INDEX.HTML



POLAROID

- □ ×

채팅방 화면

POLAROID

Search



Autowried

너무 이뻐요!

1

03-27 13:33

입장



INDEX.HTML



실시간 채팅 테스트 화면

POLAROID

Search

言论

SemiAuto

Autowried

너무 이뻐요!

김사합니다~!

너무 이뻐요!

2022-03-27 13:33:03

SemiAuto

김사합니다~!

방금



채팅코드

채팅방 목록과 채팅하는 사람의 닉네임체크 채팅방 생성코드입니다. WebSocket기반 Sock JS와 STOMP를 사용한 Live 채팅입니다

```
// 채팅방 목록
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
@GetMapping("/chat")
public String chatHome(@AuthenticationPrincipal PrincipalDetails principalDetails, Model model){
    // 헤더
    memberController.notice(principalDetails.getMember().getId());
    model.addAttribute(attributeName: "member", memberService.findById(principalDetails.getMember().getId()));

    model.addAttribute(attributeName: "nickname", principalDetails.getMember().getMemberNickname());

    // 채팅방 목록
    MemberEntity member = memberService.findByNickname(principalDetails.getMember().getMemberNickname());
    List<ChatRoomJoinEntity> chatRoomJoins = chatRoomJoinService.findByUser(member);
    List<ChatRoomFormDTO> chatRooms = chatRoomService.setting(chatRoomJoins, member);

    // 채팅방 노티스
    for(ChatRoomFormDTO c: chatRooms){
        int count = noticeRepository.countRoom(principalDetails.getMember().getId(), c.getId());
        c.setMessageNotice(count);
    }

    model.addAttribute(attributeName: "chatRooms", chatRooms);

    model.addAttribute(attributeName: "nickname", principalDetails.getMember().getMemberNickname());
    for(ChatRoomFormDTO c: chatRooms){
        MemberEntity byId=memberService.findById(c.getId());
        c.setMember(byId);
    }
    if (member == null){
        model.addAttribute(attributeName: "memberName", attributeValue: "");
        model.addAttribute(attributeName: "memberId", attributeValue: 0);
    } else {
        model.addAttribute(attributeName: "memberName", member.getMemberNickname());
        model.addAttribute(attributeName: "memberId", member.getId());
    }
}

return "chat/main";
```

```
// 닉네임 체크
@GetMapping("/users/nameChk/{name}")
public @ResponseBody String nameChk(@PathVariable("name") String name){
    if(memberService.findByNickname(name)!= null){
        return "success";
    }
    return "false";
}
```

```
// 채팅방 생성
@PreAuthorize("hasRole('ROLE_MEMBER') or hasRole('ROLE_ADMIN') or hasRole('ROLE_SELLER')")
@PostMapping("/chat/newChat")
public Long newChat(@RequestParam("receiver") String user1, @RequestParam("sender") String user2){
    Long chatRoomId = chatRoomJoinService.newRoom(user1,user2);
    return chatRoomId;
}
```



채팅코드

```
@RequestMapping("/*{/personalChat/{chatRoomId}}")
public String goChat(@PathVariable("chatRoomId")Long chatRoomId, Model model,
                     @AuthenticationPrincipal PrincipalDetails principalDetails){
    Optional<ChatRoomEntity> opt = chatRoomService.findById(chatRoomId);
    ChatRoomEntity chatRoom = opt.get();
    List<ChatMessageEntity> messages = chatRoom.getMessages();
    Collections.sort(messages, (t1, t2) -> {
        if (t1.getId() > t2.getId()) return 1;
        else return -1;
    });
    if (principalDetails.getMember() == null){
        model.addAttribute("attributeName", "memberName", attributeValue: "");
        model.addAttribute("attributeName", "memberId", attributeValue: 0);
    }else {
        model.addAttribute("attributeName", "MemberName", principalDetails.getMember().getMemberNickname());
        model.addAttribute("attributeName", "memberId", principalDetails.getMember().getId());
    }
    List<ChatRoomJoinEntity> list = chatRoomJoinService.findByChatRoom(chatRoom);
    model.addAttribute("messages", messages);
    model.addAttribute("nickname", principalDetails.getMember().getMemberNickname());
    model.addAttribute("profile", principalDetails.getMember().getMemberFilename());
    model.addAttribute("chatRoomId", chatRoomId);

    chatMessageService.deleteCount(chatRoomId, principalDetails.getMember().getId());

    int cnt = 0;
    for (ChatRoomJoinEntity join : list){
        if (join.getMember().getMemberNickname().equals(principalDetails.getMember().getMemberNickname()) == false){
            model.addAttribute("receiver", join.getMember().getMemberNickname());
            ++cnt;
        }
    }
    if (cnt >= 2){
        return "redirect:/chat";
    }
    if (cnt == 0){
        model.addAttribute("receiver", attributeValue: "");
    }
    return "chat/chatRoom";
}
```



채팅코드

```
@Transactional(readOnly = true)
public List<ChatRoomJoinEntity> findByUser(MemberEntity member) {
    return chatRoomJoinRepository.findByMember(member);
}

@Transactional(readOnly = true)
public Long check(String user1, String user2){
    MemberEntity userFirst = memberService.findByNickname(user1);
    List<ChatRoomJoinEntity> listFirst = chatRoomJoinRepository.findByMember(userFirst);
    Set<ChatRoomEntity> setFirst = new HashSet<>();
    for(ChatRoomJoinEntity chatRoomJoin : listFirst){
        setFirst.add(chatRoomJoin.getChatRoom());
    }
    MemberEntity userSecond = memberService.findByNickname(user2);
    List<ChatRoomJoinEntity> listSecond = chatRoomJoinRepository.findByMember(userSecond);
    for(ChatRoomJoinEntity chatRoomJoin : listSecond){
        if(setFirst.contains(chatRoomJoin.getChatRoom())){
            return chatRoomJoin.getChatRoom().getId();
        }
    }
    return 0L;
}
```

```
@Transactional
public Long newRoom(String user1, String user2) {
    Long ret = check(user1, user2);
    if(ret != 0){
        //이미 존재하는 방이면 해당 방 번호 리턴
        return ret;
    }
    ChatRoomEntity chatRoom = new ChatRoomEntity();
    ChatRoomEntity newChatRoom = chatRoomRepository.save(chatRoom);
    if(user1.equals(user2)){
        //나 자신과의 채팅은 한명만 존재
        createRoom(user1, newChatRoom);
    } else{
        //두명 다 입장
        createRoom(user1, newChatRoom);
        createRoom(user2, newChatRoom);
    }
    return newChatRoom.getId();
}
```



POLAROID

- □ ×

Thank You.



INDEX.HTML