

신한DS 금융 SW아카데미

희희낙찰

(중고 경매 쇼핑몰)

TEAM 5조 희희낙찰

신현상, 신정인, 이두리, 손동희, 황수영, 최재명

[멘토] 정진 강사님

목 차

- 01 프로젝트 개요
- 02 프로젝트 팀 구성 및 역할
- 03 프로젝트 수행 절차 및 방법
- 04 프로젝트 수행 경과
- 05 자체 평가 의견

01

K-Digital Training

프로젝트 개요

1 2 5

프로젝트 주제
및 선정 배경,
기획의도, 내용

※ 선정 배경

1. 디지털 기술의 발전
2. 경제적 이유
3. 사회적 연결성
4. 거래 효율의 극대화

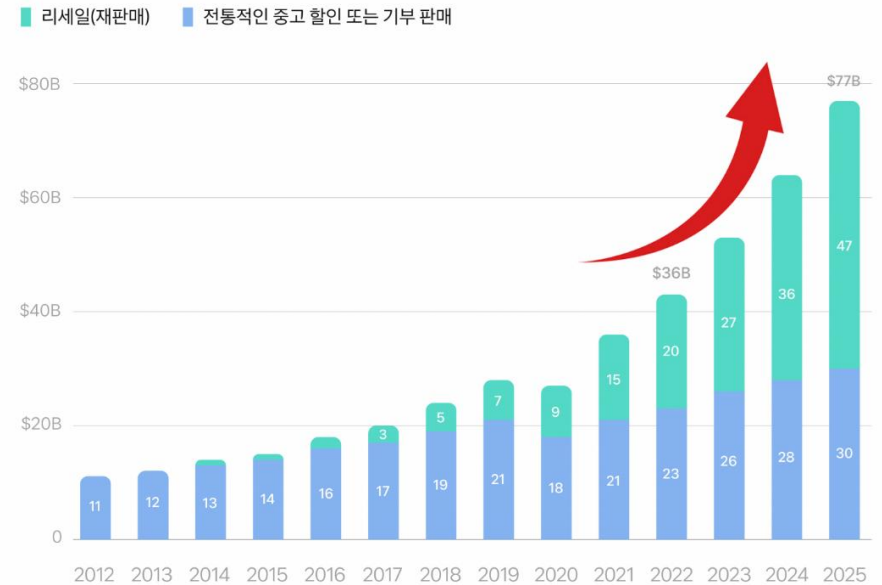
구현 목표

1. 사용자 가입 및 관리 시스템
2. 중고 상품 판매 및 구입 시스템
3. 중고 상품 실시간 경매 시스템
4. 포인트 결제 및 내역 확인 시스템
5. 사용자 간의 소통을 위한 채팅 시스템

기대 효과

1. 사용자 기반 맞춤형 시스템으로 거래의 편리함 증대
2. 실시간 경매 시스템의 활용으로 다양한 사용자 참여 유도
3. 판매자 평점 관리를 통한 사용자 신뢰성 확보와 장기적인 거래 증가 기대

전 세계 중고 거래 시장 규모 (예측)



출처: thredUP, GlobalData 2021

01

K-Digital Training

프로젝트 개요

3

활용 장비 및
자료

개발환경



FRONT

BACK

DB/Cloud

협업툴



01

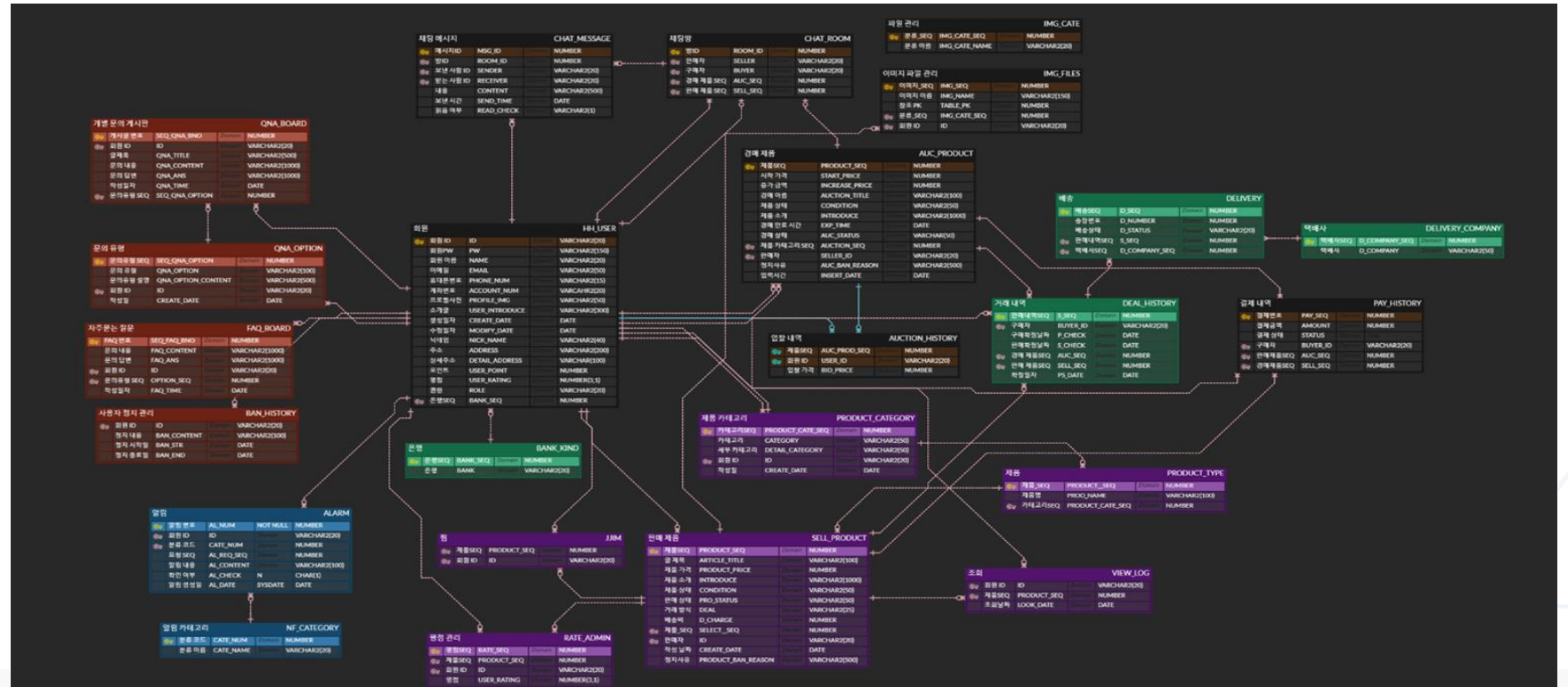
K-Digital Training

프로젝트 개요

4

프로젝트
구조

ER-Diagram



01

K-Digital Training

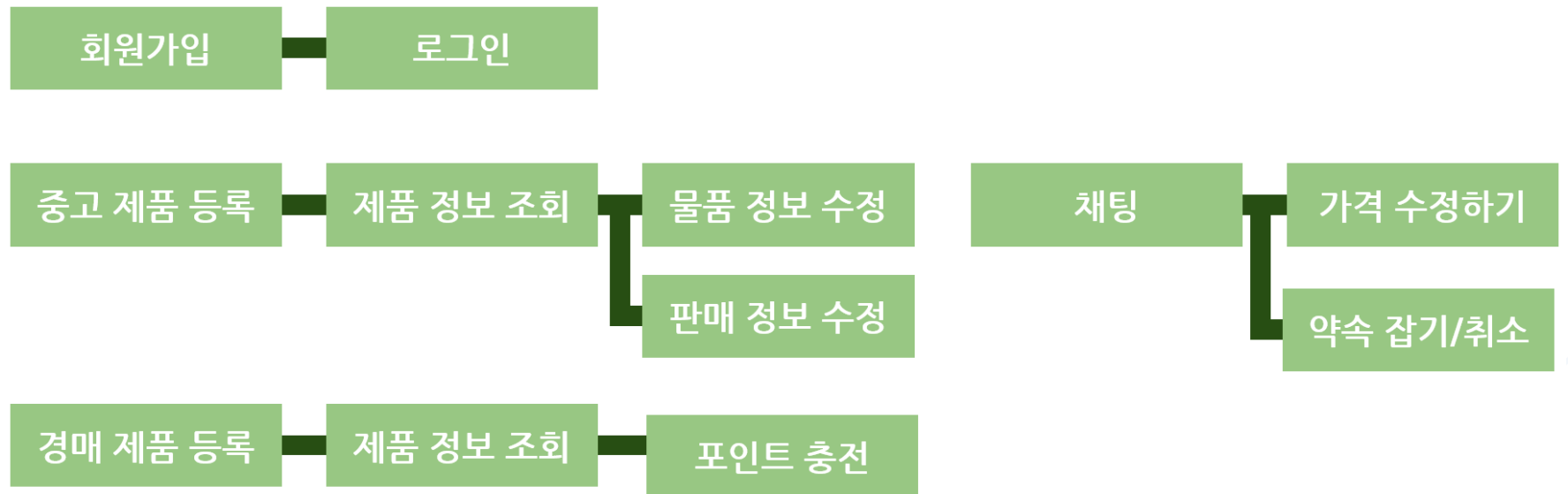
프로젝트 개요

4

프로젝트
구조

서비스 흐름도

1. 판매자



01

K-Digital Training

프로젝트 개요

4

프로젝트
구조

서비스 흐름도

2. 구매자

회원가입

로그인

중고 제품 조회

판매자와 채팅

즉시 결제

경매 제품 조회

입찰하기

포인트 충전

채팅

결제하기

01

K-Digital Training

프로젝트 개요

4

프로젝트
구조

서비스 흐름도

3. 공통

전체/미확인 알
림 확인

메인

실시간 인기 상
품 조회

추천 상품 조회

방금 등록된 상
품 조회

카테고리별
상품 조회

상품 검색

중고 제품 조회

추천 제품 조회

최근 본 상품 조
회

마이페이지

판매/구매/찜내역
확인

FAQ게시판

QNA게시판

개인정보 조회/
변경

포인트 내역
조회/충전

01

K-Digital Training

프로젝트 개요

4

프로젝트
구조

서비스 흐름도

4. 관리자

로그인

관리자 홈

회원정보 관리

회원정보 조회

이용상태 관리

상품 관리

일반상품 관리

경매상품 관리

카테고리 관리

고객 지원

1:1 문의 관리

FAQ 내용 관리

문의 유형 관리

02

K-Digital Training

프로젝트 팀 구성 및 역할

▶ 해당 프로젝트를 진행하면서 **훈련생** 별로 **주도적으로 참여한 부분을 중심으로 작성한다.**

* 프로젝트 운영 중 **멘토**의 지원내역도 간략하게 작성

훈련생	역할	담당 업무
신현상	팀장	 피그마 작성  DB 설계  웹 소켓을 이용한 실시간 경매 구현
		 스프링 시큐리티 구축  JWT 구현  Oauth2.0을 통한 소셜 로그인 구현
		 로그인, 회원가입 기능 구현  Elasticsearch 기능 구현  결제 라이브러리 연동
		 Amazon S3 구현, IAM 관리  아키텍처 설계  공동 JS 및 모듈 개발, GIT 협업관리
신정인	팀원	 피그마 작성  DB 설계  웹 소켓을 이용한 실시간 채팅 구현
		 약속잡기/취소 기능 구현  채팅방 내 결제기능 구현  판매제품 가격 수정 구현
이두리	팀원	 피그마 작성  DB 설계  마이페이지 구현
		 QNA 게시판 구현  FAQ 게시판 구현














02

K-Digital Training

프로젝트 팀 구성 및 역할

▶ 해당 프로젝트를 진행하면서 **훈련생** 별로 **주도적으로 참여한 부분을 중심으로 작성한다.**

* 프로젝트 운영 중 **멘토**의 지원내역도 간략하게 작성

훈련생	역할	담당 업무
손동희	팀원	 피그마 작성  판매 상품 상세페이지 구현  물품 등록 및 수정 페이지 구현  DB 설계  메인화면 페이지 구현
황수영	팀원	 피그마 작성  웹 소켓을 이용한 실시간 알림  이용 중지 유저 페이지 구현  DB 설계  BxSlider 라이브러리 활용
최재명	팀원	 피그마 작성  DB 설계  관리자 페이지 구현
정진 강사님	멘토	<ul style="list-style-type: none"> • 구매확정관련 트리거 문장 피드백 • faq 검색 기능을 프론트에서 처리하는 방법 조언 • 개발환경 오류 해결 • DB 설계 방향성 제시 • 스프링 시큐리티+JWT 자문 • 트랜잭션 어노테이션 개념 설명 및 올바른 설계 방법 제안

03

K-Digital Training

프로젝트 수행 절차 및 방법

▶ 프로젝트의 **사전 기획**과 **프로젝트 수행 및 완료** 과정으로 나누어서 작성한다.

- 프로젝트 수행 절차를 도식화하여 제시하거나, 더 효과적으로 전달하는 방법 등이 있다면 수정하여 작성 가능
- 기획 단계에서 도출된 주제와 아이디어를 기반으로 실제 프로젝트를 수행한 세부적인 기간과 활동 내용 작성

구분	기간	활동	비고
사전 기획	5/28(화) ~ 6/5(수)	 프로젝트 기획 및 주제 선정  요구사항 명세서 작성	아이디어 선정
자료 수집	6/10(월) ~ 6/14(금)	 유사 서비스 기반 자료 수집  유사 프로젝트 ERD자료 수집	구현 참고 데이터 수집
데이터 전처리	6/17(월) ~ 6/21(금)	 Elasticsearch 기술 적용을 위한 텍스트 데이터의 직렬화 진행	기능구현을 위한 DB상 데이터 전처리 진행
기능개발 진행	6/17(월) ~ 7/9(화)	 프로토 디자인 및 ERD 설계 기반 담당페이지 구현	개인별 담당 페이지 개발 진행
서비스 구축	7/1(월) ~ 7/9(화)	 데이터 통합 후 구동테스트  배포환경 디버깅 진행	최적화, 오류 수정
총 개발기간	5/28(화) ~ 7/9(화)(총 6주)		

X



04 K-Digital Training 프로젝트 수행 경과



기능 설명 채팅(WebSocket)

```
<!-- websocket -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-websocket</artifactId>
  <version>5.3.10</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-messaging</artifactId>
  <version>5.3.10</version>
</dependency>
```

pom.xml
spring-websocket
spring-messaging 추가

- >  WebSocketConfig.java
- >  WebSocketEventListener.java



웹소켓 설정 클래스

웹소켓 연결 및 구독 이벤트 처리
각 토픽의 구독자 수 확인



기능 설명 채팅(WebSocket)

ChattingController.java 웹소켓 설정

```
// 소켓: 메시지 (+이미지) insert
// @SendTo 대신 convertAndSend 사용
//convertAndSend(destination, payload)
@MessageMapping("/chat")
public void sendMessage(ChatMessageDTO message) throws IOException {
    int subscribeCount = websocketEventListener.getSubscribersCount("/topic/chatroom/" + message.getRoomId());

    if(subscribeCount>=2) {
        message.setReadCheck("Y");
    } else {
        message.setReadCheck("N");
    }

    cService.saveMessage(message);

    messagingTemplate.convertAndSend("/topic/chatroom/" + message.getRoomId(), message);
}

@MessageMapping("/joinRoom")
public void joinRoom(Map<String, Object> map) throws IOException {
    messagingTemplate.convertAndSend("/topic/chatroom/" + map.get("roomId"), map);
}
```

채팅 메시지를 보낼 때

채팅방에 들어왔을 때



기능 설명 채팅(WebSocket)

chatting.js 웹소켓 설정

```

var ChatStompClient = null; // stomp 소켓 클라이언트 전역 설정

// 채팅룸 커넥트
//send(destination(보낼 목적지), header, body)
//subscribe(destination(구독할 목적지), callback(목적지에서 메시지 왔을 때 호출할 함수), header)
function chatConnect(chatRoomId) {
    disconnect();
    var ChatSocket = new SockJS('/heehee/chatws');
    ChatStompClient = Stomp.over(ChatSocket);
    ChatStompClient.connect({}, function (frame) {
        ChatStompClient.send("/app/joinRoom", {}, JSON.stringify({"userId": loginMemberNo, "roomId": chatRoomId}));
        ChatStompClient.subscribe('/topic/chatroom/' + chatRoomId, function (chatMessage) {
            var json = JSON.parse(chatMessage.body);
            if(json.userId != null) { // 사용자가 채팅방에 참가할 때
                selectChattingFn(); // 채팅방 메시지 목록 전체 재출력(안 읽음 -> 읽음)
            } else { // 사용자가 채팅방에 존재하지 않을 때
                showMessage(json); // 보낸 채팅 메시지만 화면에 출력
            }
        });
    });
}

// 연결 끊기
function disconnect() {
    if (ChatStompClient !== null) {
        ChatStompClient.disconnect();
    }
    console.log("Disconnected");
}

```

채팅방에 들어왔을 때
소켓 연결

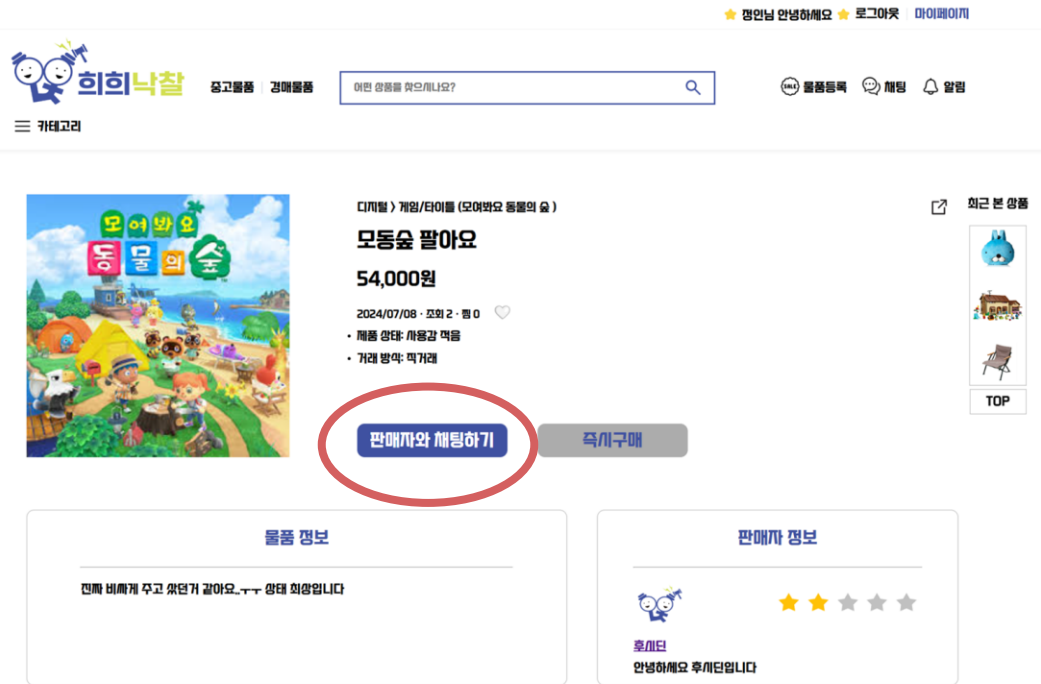
구독한 채팅방에서
메세지가 왔을 때 함수 호
출

소켓 연결 해제

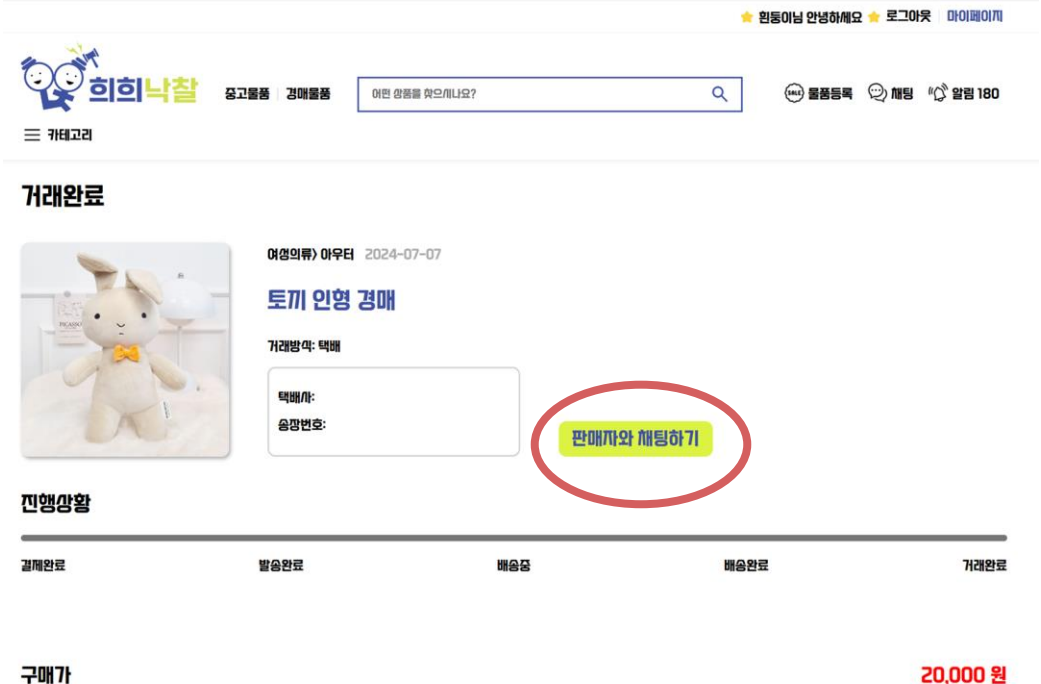


기능 설명 채팅(WebSocket)

중고제품 상세페이지에서
상품 등록자에게 실시간 채팅 구
현



경매제품 낙찰 시 마이페이지에서
상품 등록자에게 실시간 채팅 구
현



ChattingController.java

```
//채팅방 생성: 판매자와 채팅 클릭 시
@PostMapping("/seller")
@ResponseBody
public int insertChatRoom(@RequestBody Map<String, Object> map) {
    return cService.insertChatRoom(map);
}
```

```
//채팅방 생성: 경매 낙찰 후 채팅 클릭 시
@PostMapping("/auction")
@ResponseBody
public int insertAuctionChat(@RequestBody Map<String, Object> map) {
    return cService.insertAuctionChat(map);
}
```



기능 설명 채팅(WebSocket)

ChattingController.java 채팅 페이지 및 채팅방 목록 반환

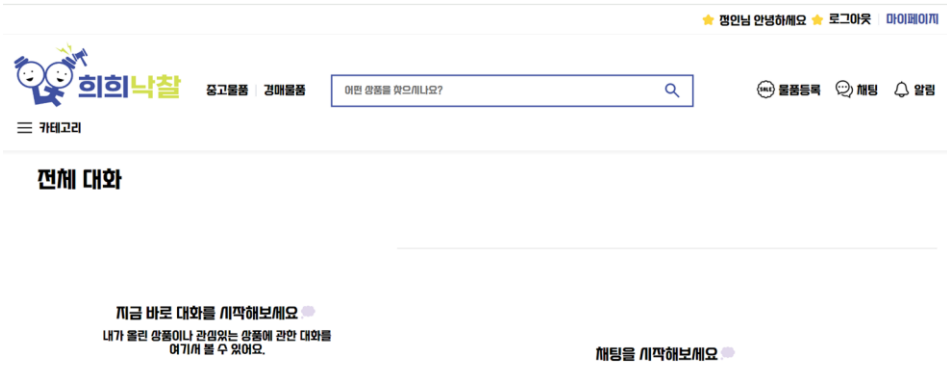
```
// 채팅 페이지
@GetMapping
public String chatting(Model model, Principal principal) {
    List<CategoryDTO> mainCateList = mainservice.mainCateList(); // 카테고리 서비스 호출

    String userId = "";
    if (principal != null)
        userId = principal.getName();

    int alarmCount = alarmService.alarmCount(userId);

    // model에 담을 것: 유저별 채팅방 목록
    model.addAttribute("roomList", cService.getRoomList(userId));
    model.addAttribute("userId", userId);
    model.addAttribute("mainCateList", mainCateList);
    model.addAttribute("alarmCount", alarmCount); // 알림 개수
    return "chatting/chatting";
}
```

```
// 채팅방 목록 조회
@GetMapping(value = "/roomList", produces = "application/json; charset=UTF-8")
@ResponseBody
public List<ChatRoomDTO> getRoomList(Principal principal) {
    String userId = "";
    if (principal != null)
        userId = principal.getName();
    //System.out.println(cService.getRoomList(userId));
    return cService.getRoomList(userId); // 로그인 유저 Id 전달
}
```



채팅방이 없는 경우



채팅방을 선택하지 않은 경우



기능 설명 채팅(WebSocket)

chatting.js
1.5초마다 채팅방 목록 업데이트

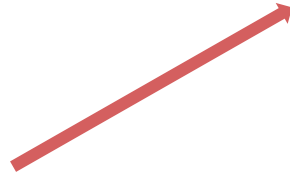
```
document.addEventListener("DOMContentLoaded", ()=>{
  // 채팅방 목록에 클릭 이벤트 추가
  roomListAddEvent();

  // 채팅방 리스트 뿌리기
  fetchChatRoomList();
  setInterval(function() {
    fetchChatRoomList();
    //selectChattingFn();
  }, 1500);
});
```

```
//채팅방 목록 1.5초마다 불러오기
function fetchChatRoomList() {
  fetch("/heehee/chatting/roomList")
  .then(response => response.json())
  .then(data => {
    console.log(data);
    if(data.length>0) $(".noneRoom").hide();
    updateChatRoomList(data);
  })
  .catch(err => console.log(err));
}
```

```
// 채팅방 목록 조회
@GetMapping(value = "/roomList", produces = "application/json; charset=UTF-8")
@ResponseBody
public List<ChatRoomDTO> getRoomList(Principal principal) {
  String userId = "";
  if (principal != null)
    userId = principal.getName();
  //System.out.println(cService.getRoomList(userId));
  return cService.getRoomList(userId); // 로그인 유저 Id 전달
}
```

ChattingController.java
채팅방 목록 조회





기능 설명 채팅(WebSocket)

```

//채팅방 목록 업데이트 함수
function updateChatRoomList(data) {
  // 채팅방 목록 출력 영역 선택
  const chattingList = document.querySelector(".chatting-list");

  // 조회한 채팅방 목록을 순회
  for (let room of data) {

    let existingChatRoom = null;
    let li = null;

    // 채팅방 아이템을 찾기
    existingChatRoom = chattingList.querySelector(`li[room-id="${room.id}"]`);
    //console.log(chattingList.querySelector(`li[room-id="${room.id}"]`));

    // 이미 있는 채팅방이면 unreadCount, sendTime, lastContent만 업데이트
    if (existingChatRoom) {
      // receiverid가 null이면 "(알 수 없음)"으로 처리
      if (room.receiverid === null) {
        existingChatRoom.querySelector(".receiver-image").setAttribute("src", "https://sh-heehee-bucket.s3.ap-northeast-2.amazonaws.com/images/mypage/logo_pi");
        existingChatRoom.querySelector(".receiver-nickname").innerText = "(알 수 없음)";
      }

      const content = existingChatRoom.querySelector(".recent-message").innerHTML;
      //안 읽은 메시지가 있거나 메시지가 새로 와서 업데이트해야하는 경우
      if(room.unreadcount>0 || content!=room.lastcontent){
        //기존 unread-count div가 있는 경우
        if(existingChatRoom.querySelector(".unread-count")){
          existingChatRoom.querySelector(".unread-count").innerText = room.unreadcount;
          existingChatRoom.querySelector(".send-time").innerText = room.sendtime;
          existingChatRoom.querySelector(".recent-message").innerHTML = room.lastcontent.startsWith("[img_asdfzv]") ? "사진" : room.lastcontent;
        }
        //기존 unread-count div가 없는 경우
        else{
          if(room.unreadcount>0){
            const unreadCount = document.createElement("p");
            unreadCount.classList.add("unread-count");
            unreadCount.innerText = room.unreadcount;

            const nameCount = existingChatRoom.querySelector(".name-count");
            nameCount.append(unreadCount);
          }

          existingChatRoom.querySelector(".send-time").innerText = room.sendtime;

```

chatting.js
채팅방 목록 업데이트 함수

기존에 있었던 채팅방인지 / 새로운 채팅방인지,
안 읽은 알림이 있었는지 / 없었는지 구분해 업데이트



기능 설명 채팅(WebSocket)

chatting.js
채팅방 클릭 시
채팅 메시지 출력 및 메시지 읽음 처리

```
// 채팅방 목록에서 채팅방 선택 시 추가되는 이벤트
function roomListAddEvent(){

const chattingItemList = document.getElementsByClassName("chatting-item");

for(let item of chattingItemList){
  item.addEventListener("click", e => {

    // 전역변수에 채팅방 번호, 상대 id, 상대 닉네임 저장
    selectRoomId = item.getAttribute("room-id");
    selectReceiverId = item.getAttribute("receiver-id");

    selectReceiverName = item.children[1].children[0].children[0].innerText;

    const unreadCountElem = item.querySelector(".unread-count");
    if (unreadCountElem) {
      unreadCountElem.remove();
    }

    // 모든 채팅방에서 select 클래스를 제거
    for(let it of chattingItemList) it.classList.remove("select")

    // 현재 클릭한 채팅방에 select 클래스 추가
    item.classList.add("select");

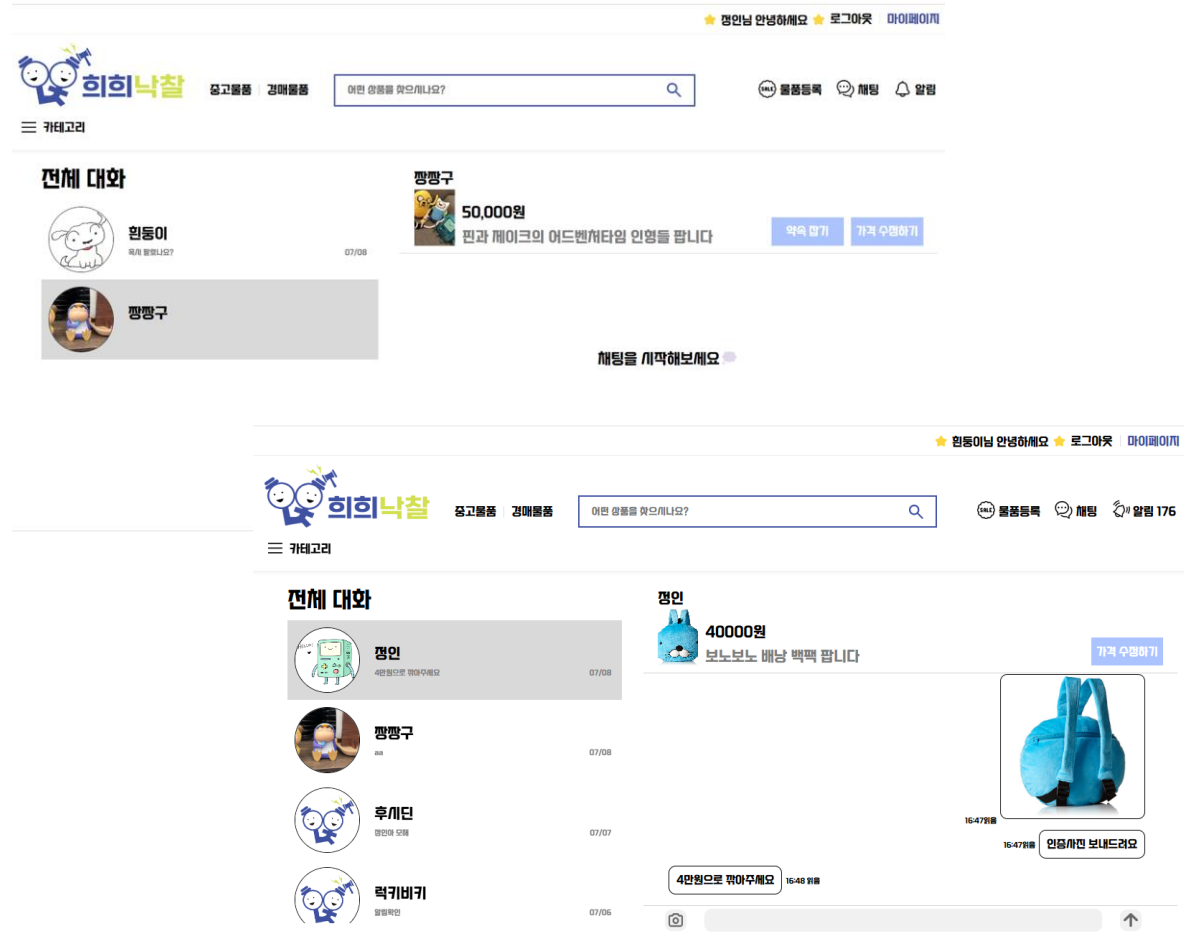
    // 채팅 메시지가 있는 경우
    // 비동기로 메시지 목록을 조회하는 함수 호출
    selectChattingFn();

    fetch("/heehee/chatting/read",{
      method: "PUT",
      headers: { "content-type": "application/json"},
      body: JSON.stringify({"chatRoomId": selectRoomId, "loginUserId": loginMemberNo})
    })
    .then(resp => resp.text())
    .then(result => console.log(result))
    .catch(err => console.log(err));

    // 선택한 채팅방에서 상대 메시지 읽음 체크
    // 업데이트 성공 시 1 반환
    @PutMapping("/read")
    @ResponseBody
    public int updateReadCheck(@RequestBody Map<String, Object> map) {
      return cService.updateReadCheck(map);
    }

    //소켓 연결
    chatConnect(selectRoomId);
  });
}
```

채팅 내역이 없는 경우 / 있는 경우 화면



ChattingController.java



기능 설명 채팅(WebSocket)

chatting.js 채팅방 메시지 표시 함수

```

// 비동기로 메시지 목록 조회하는 함수
function selectChattingFn() {
  fetch('/heehee/chatting/${selectRoomId}')
  .then(response => response.json())
  .then(roomDetail => {
    const chattingContent = document.querySelector(".chatting-content");
    chattingContent.innerHTML="";

    //채팅 메시지 위 영역: 상대방 닉네임, 판매 물품 정보(이미지, 가격, 제품명)
    const contentHeader = document.createElement("div");
    contentHeader.classList.add("content-header");

    const receiverNickname = document.createElement("p");
    receiverNickname.classList.add("receiver-nickname");
    receiverNickname.innerHTML = selectReceiverName ? selectReceiverName : "(알 수 없음)";

    const sellingInfo = document.createElement("div");
    sellingInfo.classList.add("selling-info");

    const sellingImage = document.createElement("img");
    sellingImage.classList.add("selling-image");

    var imgName = "mypage/logo_profile.jpg";

    if(roomDetail.roomProductDTO.productSeq){
      imgName = roomDetail.roomProductDTO.productImg ? `${roomDetail.roomProductDTO.productType}/${roomDetail.roomProductDTO.productImg}` : "mypage/logo_profile.jpg";
    }

    const imgUrl = `https://sh-heehee-bucket.s3.ap-northeast-2.amazonaws.com/images/${imgName}`;
    sellingImage.setAttribute("src", imgUrl);

    const priceName = document.createElement("div");
    priceName.classList.add("price-name");

    const sellingPrice = document.createElement("p");
    sellingPrice.classList.add("selling-price");
    const price = roomDetail.roomProductDTO.productPrice ? roomDetail.roomProductDTO.productPrice : 0;
    const formatPrice = price.toLocaleString('ko-KR');
    //console.log(formatPrice);
    sellingPrice.innerText = formatPrice + '원';

    const sellingName = document.createElement("p");
    sellingName.classList.add("selling-name");
    sellingName.innerHTML = roomDetail.roomProductDTO.productName ? roomDetail.roomProductDTO.productName : "(삭제된 게시물)";

    priceName.append(sellingPrice, sellingName);
  });
}

```

ChattingController.java

```

// 채팅방 목록에서 채팅방 클릭 시 해당 채팅방의 판매 물품 정보, 유저 계좌 및 포인트 정보, 메시지 목록 조회
@GetMapping(value = "/{id}", produces = "application/json; charset=UTF-8")
@ResponseBody
public RoomDetailDTO getRoomDetail(@PathVariable("id") int chatRoomId, Principal principal) {
  String userId = "";
  if (principal != null)
    userId = principal.getName();
  Map<String, Object> map = new HashMap<String, Object>();
  map.put("chatRoomId", chatRoomId);
  map.put("loginUserId", userId);
  return cService.getRoomDetail(map); // 채팅방 번호와 로그인 유저 Id 전달
}

```

로그인 유저가 구매자인지 / 판매자인지,
보낸 메시지의 수신자인지 / 발신자인지,
보낸 메시지가 이미지인지 텍스트인지,
읽음 여부가 Y인지 N인지
구분해 버튼 및 메시지 표시



기능 설명 채팅(WebSocket)

```

if(receiverNickname.innerHTML != "(알 수 없음)" && sellingName.innerHTML != "(삭제된 게시물)"){
  // 전송 버튼 클릭 시 이벤트 추가
  inputSend.addEventListener('click', ()=>{
    sendMessage(inputElement.value);
    //console.log(inputElement.value);
    inputElement.value="";
  });

  // 엔터 키 입력 시 이벤트 추가
  inputElement.addEventListener('keydown', (event) => {
    if (event.key == 'Enter') {
      sendMessage(inputElement.value);
      inputElement.value = "";
    }
  });
}

```

chatting.js

```

//이미지 업로드: js에서 사용
@PostMapping("/upload/image")
@ResponseBody
public List<String> uploadImages(@RequestPart List<MultipartFile> imgs) throws IOException {

  ArrayList<String> imgNames = new ArrayList<String>();

  if (imgs != null && !imgs.isEmpty()) {
    for (MultipartFile img : imgs) {
      try {
        String imgName = s3Service.uploadOneObject(img, "images/chat/");
        imgNames.add(imgName);
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
  }
  return imgNames;
}

```

ChattingController.java
이미지 업로드 함수

```

function sendMessage(inputValue){
  if(inputValue!=" " || selectedFiles.length > 0){
    const offset = new Date().getTimezoneOffset() * 60000;
    const today = new Date(Date.now() - offset);
    const sendTime = today.toISOString().replace('T', ' ').replace('Z', "").substring(0,19);

    if (selectedFiles.length > 0) {
      const formData = new FormData();

      selectedFiles.forEach(file => {
        formData.append('imgs', file);
      });

      fetch("/heehee/chatting/upload/image" {
        method: 'POST',
        body: formData
      })
      .then(response => response.json())
      .then(data => {
        console.log(data);
        const message = {
          'roomId': selectRoomId,
          'sender': loginMemberNo,
          'receiver': selectReceiverId,
          'content': inputValue,
          'sendTime': sendTime,
          'imgs': data
        };
        stompClient.send("/app/chat", {}, JSON.stringify(message));
        selectedFiles = []; // 이미지 전송 후 파일 초기화
        imageInput.value = ""; // 파일 입력 초기화
      })
      .catch(err => console.log(err));
    } else {
      const message = {
        'roomId': selectRoomId,
        'sender': loginMemberNo,
        'receiver': selectReceiverId,
        'content': inputValue,
        'sendTime': sendTime,
        'imgs': []
      };
      stompClient.send("/app/chat", {}, JSON.stringify(message));
    }
  }
}

```

chatting.js
채팅 메시지 보내기 함수



기능 설명 채팅(WebSocket)

chatting.js
채팅 보여주기 함수

로그인 유저가 구매자인지 / 판매자인지,
보낸 메시지의 수신자인지 / 발신자인지,
보낸 메시지가 이미지인지 텍스트인지,
읽음 여부가 Y인지 N인지
구분해 채팅 메시지 표시

```
function showMessage(chatMessage){
  if(document.querySelector(".noneRoomChat")){document.querySelector(".noneRoomChat").remove();}
  const messageList = document.querySelector(".message-list");

  const contentBody = document.querySelector(".content-body");

  if(chatMessage.sender == loginMemberNo){

    if(chatMessage.imgs.length == 0){

      const myChat = document.createElement("div");
      myChat.classList.add("my-chat");

      const chatDate = document.createElement("span");
      chatDate.classList.add("chatDate");
      let read = '안 읽음';

      if(chatMessage.readCheck == 'Y'){
        read = '읽음';
      }

      chatDate.innerHTML = chatMessage.sendTime.substr(11,5) + read;

      const chat = document.createElement("p");
      chat.classList.add("chat");
      chat.innerHTML = chatMessage.content;

      myChat.append(chatDate, chat);
      messageList.append(myChat);
      contentBody.append(messageList);

      contentBody.scrollTop = contentBody.scrollHeight;
    }
    else{
      chatMessage.imgs.forEach(img => {

        const myChat = document.createElement("div");
        myChat.classList.add("my-chat");

        const chatDate = document.createElement("span");
        chatDate.classList.add("chatDate");
        let read = '안 읽음';

        if(chatMessage.readCheck == 'Y'){
          read = '읽음';
        }
      });
    }
  }
}
```



기능 설명 채팅(WebSocket)

구매자가 채팅 보냄



판매자에게 알림 도착



중고물품 | 경매물품

어떤 상품을 찾으시나요?

카테고리

전체 대화



평인
욕새 팔렸나요?



07/08



짱짱구

07/08



후기민
평인야 도해

07/07



럭키비키
알림확인

07/06

채팅을 시작해보세요

★ 평인님 안녕하세요 ★ 로그인 마이페이지



중고물품 | 경매물품

어떤 상품을 찾으시나요?



물품등록



채팅



알림

카테고리

전체 대화



원동이
욕새 팔렸나요?

07/08

원동이



50,000원

보노보노 배낭 백팩 팝니다

결제하기

16:42 안 읽음

욕새 팔렸나요?

안 읽음 상태



기능 설명 채팅(WebSocket)

알림 확인 시 알림 아이콘 사라짐

★ 원동이님 안녕하세요 ★ 로그아웃 마이페이지



중고물품 경매물품

어떤 상품을 찾으시나요?

물품등록 채팅 알림 176

카테고리

전체 대화



정인

혹시 팔렸나요?

07/08

정인



50,000원

보노보노 배낭 백팩 팝니다

가격 수정하기

혹시 팔렸나요?

16:42 읽음



팽팽구

08

07/08



후시딘

명인아 오빠

07/07



럭키비키

알뜰학년

07/06



★ 정인님 안녕하세요 ★ 로그아웃 마이페이지



중고물품 경매물품

어떤 상품을 찾으시나요?

물품등록 채팅 알림

카테고리

전체 대화



원동이

혹시 팔렸나요?

07/08



팽팽구

원동이



50,000원

보노보노 배낭 백팩 팝니다

구매하기

16:42 읽음

혹시 팔렸나요?

구매자 채팅방에서도
읽음으로 상태 변화





기능 설명 채팅(WebSocket)

chatting.js

판매중이고 택배 거래인 중고제품의 경우
구매자에게 결제하기 버튼 표시

```
const buttonDiv = document.createElement("div");
buttonDiv.classList.add("button-div");
```

```
//중고 제품인 경우에만 버튼 추가
if(roomDetail.roomProductDTO.productType=='sell'){
  if (status == '구매자') {
    if(roomDetail.roomProductDTO.dealStatus == '택배' && roomDetail.roomProductDTO.sellProStatus=='판매중'){
      const payButton = document.createElement("button");
      payButton.classList.add("pay");
```

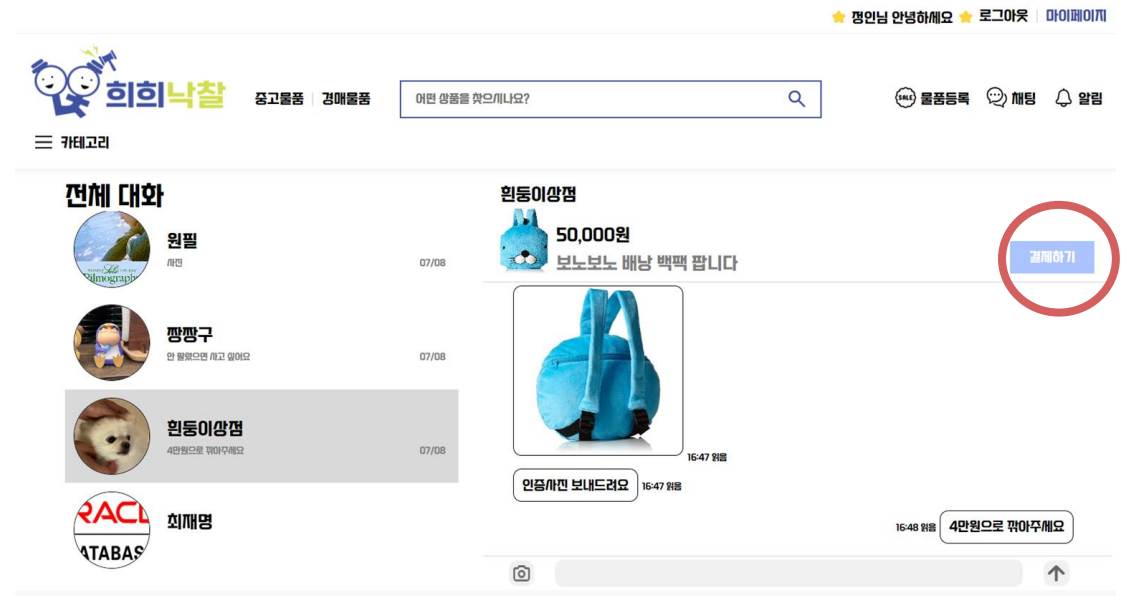
```
payButton.innerText = "결제하기";
```

```
//결제 내역 테이블에 동일 구매자id&제품 seq 데이터의 결제상태가 '완료'면 결제하기 버튼 비활성화
if(payStatus == '완료'){
  payButton.disabled = true;
}
```

```
payButton.addEventListener("click", () => {
  // 구매자 버튼 클릭 이벤트 처리
  const payName = roomDetail.roomProductDTO.productName;
  const amount = roomDetail.roomProductDTO.productPrice;
  const sellSeq = roomDetail.roomProductDTO.productSeq;
  payForSell(payName, amount, sellSeq, payButton);
});
```

```
buttonDiv.append(payButton);
}
```

결제하기 버튼 활성화



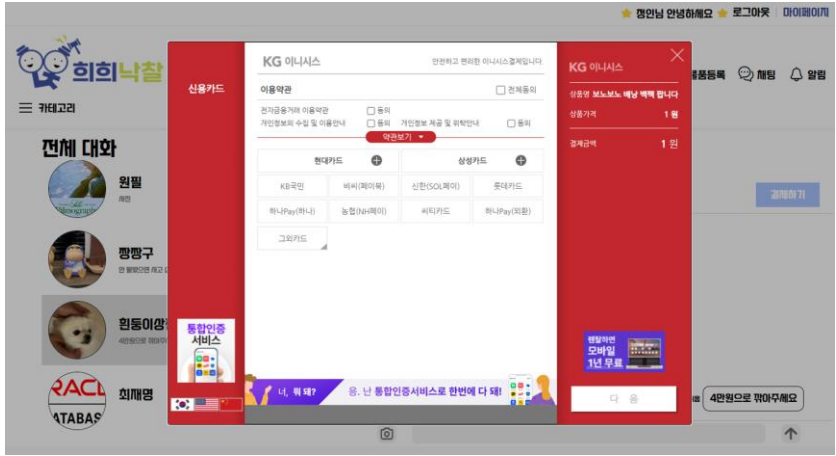


기능 설명 채팅(WebSocket)

```
function payForSell(payName, amount, sellSeq, payButton) {
  $.ajax({
    url: '/heehee/pay/before',
    method: 'POST',
    data: {
      "payName" : payName,
      "amount" : amount,
      "sellSeq" : sellSeq
    },
    success: function (data, status, xhr) {
      payment(payName,data, payButton);
    },
    error: function (data, status, err) {
      console.log(err);
    }
  });
}
```

chatting.js
결제 함수

```
function payment(payName, payInfo, payButton) {
  IMP.init("imp22447463");
  IMP.request_pay(
    {
      pg: "html5_inicis.INIpayTest", //테스트 시 html5_inicis.INIpayTest 기재
      pay_method: "card",
      merchant_uid: payInfo.paySeq, //상점에서 생성한 고유 주문번호
      name: payName,
      amount: 1,
      buyer_email: payInfo.buyerEmail,
      buyer_name: payInfo.buyerId,
      buyer_tel: payInfo.buyerTel, //필수 파라미터 입니다.
      buyer_addr: payInfo.buyerTel,
      buyer_postcode: "123-456",
      m_redirect_url: "{모바일에서 결제 완료 후 리디렉션 될 URL}",
      escrow: true, //에스크로 결제인 경우 설정
      vbank_due: "20240725",
      bypass: {
        acceptmethod: "noeasypay", // 간편결제 버튼을 통합결제창에서 제외(PC)
        P_RESERVED: "noeasypay=Y", // 간편결제 버튼을 통합결제창에서 제외(모바일)
      },
      period: {
        from: "20240101", //YYYYMMDD
        to: "20241231", //YYYYMMDD
      },
    }, function (rsp) {
      if(rsp.success) {
        completePayment(payInfo, payButton);
        return true;
      }
    }
  );
}
```





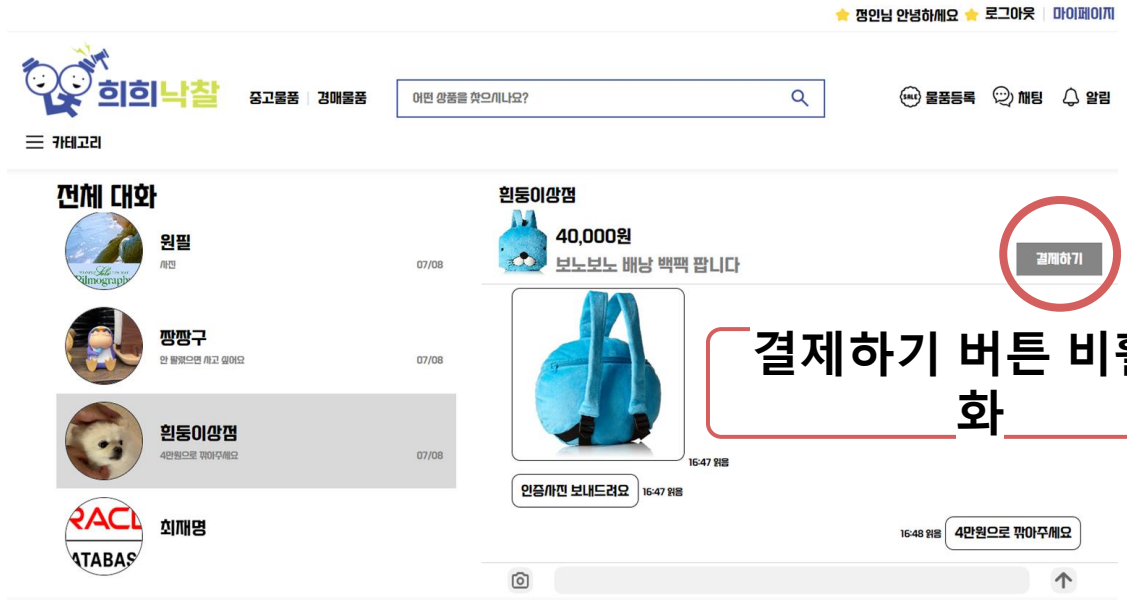
기능 설명 채팅(WebSocket)

```
function completePayment(payInfo, payButton) {
  $.ajax({
    url: '/heehee/pay/complete',
    method: 'PUT',
    contentType: 'application/json',
    data: JSON.stringify({ "paySeq": payInfo.paySeq }),
    success: function (data, status, xhr) {
      console.log(data);
      console.log(status);
      console.log(xhr);
      fetch("/heehee/chatting/reserve", {
        method: "POST",
        headers: {"Content-Type": "application/json"},
        body: JSON.stringify({
          "buyerId" : loginMemberNo,
          "productSeq" : payInfo.sellSeq
        })
      })
    }
  })
  .then(resp => resp.text())
  .then(result => console.log(result))
  .catch(err => console.log(err));
  payButton.disabled = true;
  showToast("결제에 성공하였습니다!");
  return true;
},
error: function (data, status, err) {
  console.log(err);
}
});
}
```

chatting.js
결제 함수

ChattingController.java
결제 함수

```
//약속 잡기 / 결제
@PostMapping("/reserve")
@ResponseBody
public void reserve(@RequestBody Map<String, Object> map) {
  cService.reserve(map);
}
```

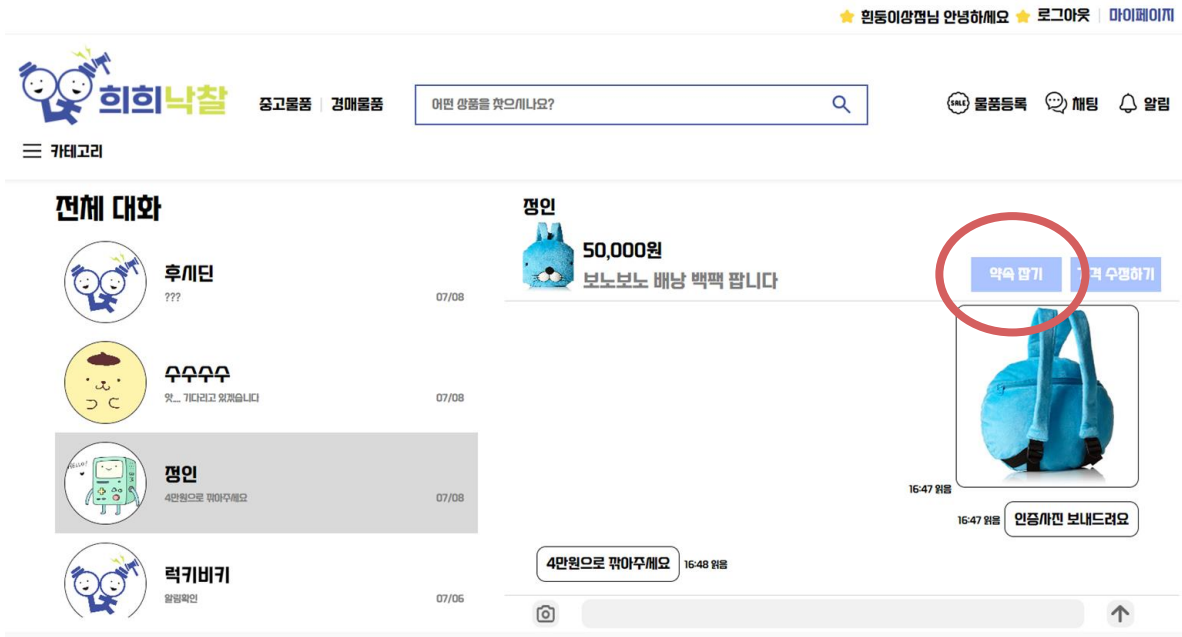




기능 설명 채팅(WebSocket)

chatting.js

판매중이고 직거래인 중고제품의 경우
구매자에게 약속잡기/약속취소 버튼 표시



```

} else if(status == '판매자'){
  //약속잡기 버튼
  if(roomDetail.roomProductDTO.dealStatus == '직거래'){
    if(roomDetail.roomProductDTO.sellProStatus=='판매중'){
      const rsvButton = document.createElement("button");
      rsvButton.classList.add("reserve");
      rsvButton.innerText = "약속 잡기";
      // 약속 잡기 버튼 클릭 이벤트 처리
      rsvButton.addEventListener("click", () => {
        if(rsvButton.innerText == "약속 잡기"){
          reserve(roomDetail.roomProductDTO.productSeq);
          rsvButton.innerText = "약속 취소";
        }
        else if(rsvButton.innerText == "약속 취소"){
          cancelReserve(roomDetail.roomProductDTO.productSeq);
          rsvButton.innerText = "약속 잡기";
        }
      });
      buttonDiv.append(rsvButton);
    }
  }
}
//판매 제품의 판매 상태가 예약중이면 약속 잡기 버튼 비활성화
else if(roomDetail.roomProductDTO.sellProStatus=='예약중'){
  const rsvButton = document.createElement("button");
  rsvButton.classList.add("reserve");
  rsvButton.innerText = "약속 취소";
  // 약속 취소 버튼 클릭 이벤트 처리
  rsvButton.addEventListener("click", () => {
    if(rsvButton.innerText == "약속 잡기"){
      reserve(roomDetail.roomProductDTO.productSeq);
      rsvButton.innerText = "약속 취소";
    }
    else if(rsvButton.innerText == "약속 취소"){
      cancelReserve(roomDetail.roomProductDTO.productSeq);
      rsvButton.innerText = "약속 잡기";
    }
  });
  buttonDiv.append(rsvButton);
}
}
}

```



기능 설명 채팅(WebSocket)

ChattingController.java
약속잡기 함수

```
//약속 잡기 / 결제
@PostMapping("/reserve")
@ResponseBody
public void reserve(@RequestBody Map<String, Object> map) {
    cService.reserve(map);
}
```

```
// 약속잡기 함수
function reserve(productSeq){
    fetch("/heehee/chatting/reserve", {
        method: "POST",
        headers: {"Content-Type": "application/json"},
        body: JSON.stringify({
            "buyerId": selectReceiverId,
            "productSeq": productSeq
        })
    })
    .then(resp => resp.text())
    .then(result => console.log(result))
    .catch(err => console.log(err));
}
```

chatting.js
약속잡기 함수

```
const button = document.querySelector(".reserve");

chattingModal.style.display='block';
chattingModal.innerHTML="";

const modalContent2 = document.createElement("div");
modalContent2.classList.add("modal-content2");

const Info2 = document.createElement("div");
Info2.innerText='약속 잡기에 성공하였습니다!';

modalContent2.append(Info2);

chattingModal.append(modalContent2);

setTimeout(()=>{
    chattingModal.style.display='none';
}, 3000);
}
```





기능 설명 채팅(WebSocket)

```
//약속 취소
@PostMapping("/reserve/cancel")
@ResponseBody
public void cancelReserve(@RequestBody Map<String, Object> map) {
    cService.cancelReserve(map);
}
```

ChattingController.java
약속 취소 함수

```
//약속 취소 함수
function cancelReserve(productSeq){
    fetch("/heehee/chatting/reserve/cancel",{
        method: "POST",
        headers: {"Content-type": "application/json"},
        body: JSON.stringify({
            "buyerId": selectReceiverId,
            "productSeq": productSeq
        })
    })
    .then(resp => resp.text())
    .then(result => console.log(result))
    .catch(err => console.log(err));
}
```

chatting.js
약속취소 함수

```
const button = document.querySelector(".reserve");

chattingModal.style.display='block';
chattingModal.innerHTML="";

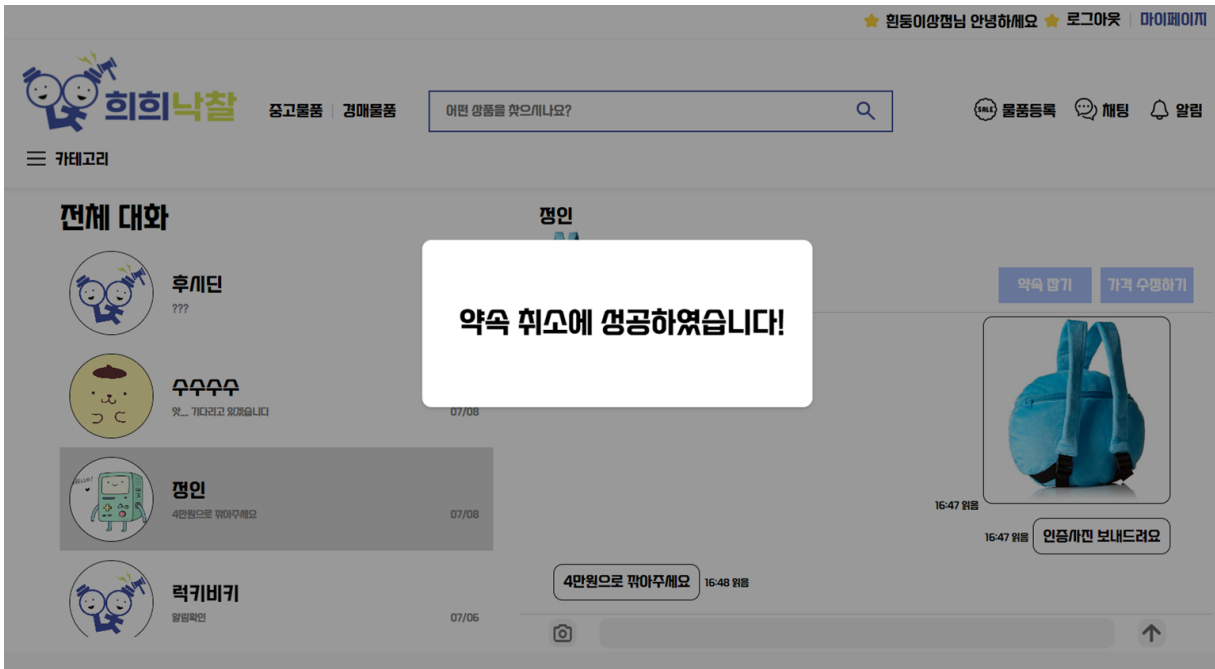
const modalContent2 = document.createElement("div");
modalContent2.classList.add("modal-content2");

const Info2 = document.createElement("div");
Info2.innerHTML='약속 취소에 성공하였습니다!';

modalContent2.append(Info2);

chattingModal.append(modalContent2);

setTimeout(()=>{
    chattingModal.style.display='none';
}, 3000);
}
```





기능 설명 채팅(WebSocket)

```
function editPrice(productPrice, productSeq){
  chattingModal.style.display='block';
  chattingModal.innerHTML="";

  const modalContent = document.createElement("div");
  modalContent.classList.add("modal-content");

  const currentPrice = document.createElement("div");
  currentPrice.classList.add("current-price");
  currentPrice.innerText='기존 가격 : ' + productPrice + '원';

  const currentInput = document.createElement("div");
  currentInput.classList.add("current-input");

  const ciElement = document.createElement('input');
  ciElement.className = 'new-price';
  ciElement.type = 'number';
  ciElement.placeholder = '수정할 가격을 입력해주세요. (원)';

  currentInput.append(ciElement);

  const chModalBtn = document.createElement("div");
  chModalBtn.classList.add("chModalBtn");

  const submitPrice = document.createElement("button");
  submitPrice.classList.add("submit-price");
  submitPrice.innerText='수정하기';

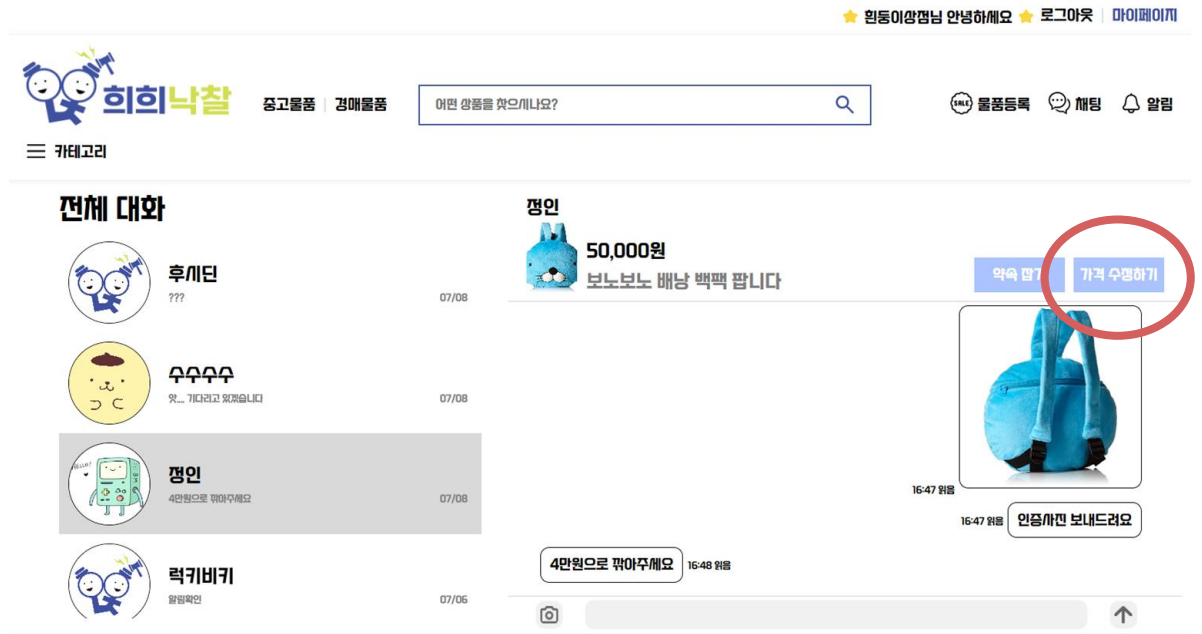
  submitPrice.addEventListener('click', ()=>{
    chattingModal.style.display='none';
    const newPrice=document.querySelector('.new-price').value;
    document.querySelector('.selling-price').innerText = newPrice + '원';
    // console.log(newPrice, productSeq);
    fetch("/heehee/chatting/price",{
      method : "PUT",
      headers : {"Content-Type": "application/json"},
      body : JSON.stringify({"newPrice" : newPrice, "productSeq" : productSeq})
    })
    .then(resp => resp.text())
    .then(result => {
      console.log(result);
      showToast("가격 수정에 성공하였습니다!");
    })
    .catch(err => console.log(err));
  });
}
```

chatting.js
판매중인 중고제품의 경우
구매자에게 가격 수정하기 버튼 표시

```
if(roomDetail.roomProductDTO.sellProStatus=='판매중'){
  const payButton = document.createElement("button");
  payButton.classList.add("edit");

  payButton.innerText = "가격 수정하기";

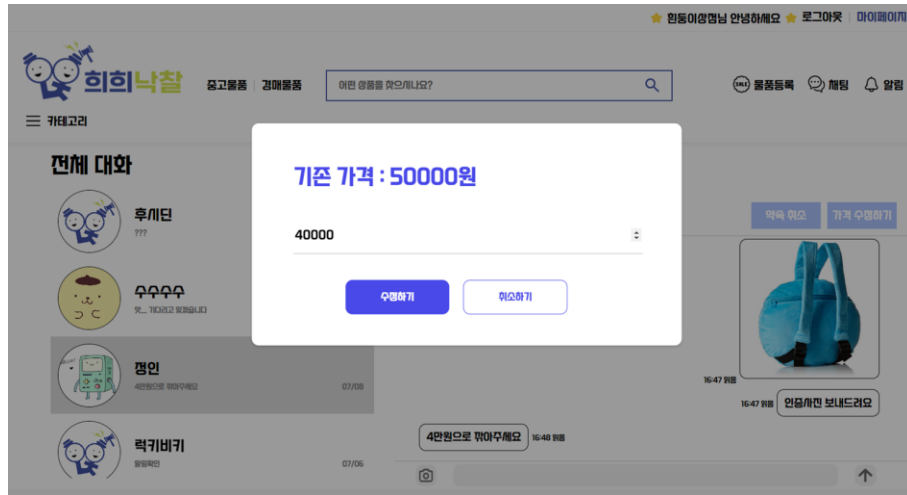
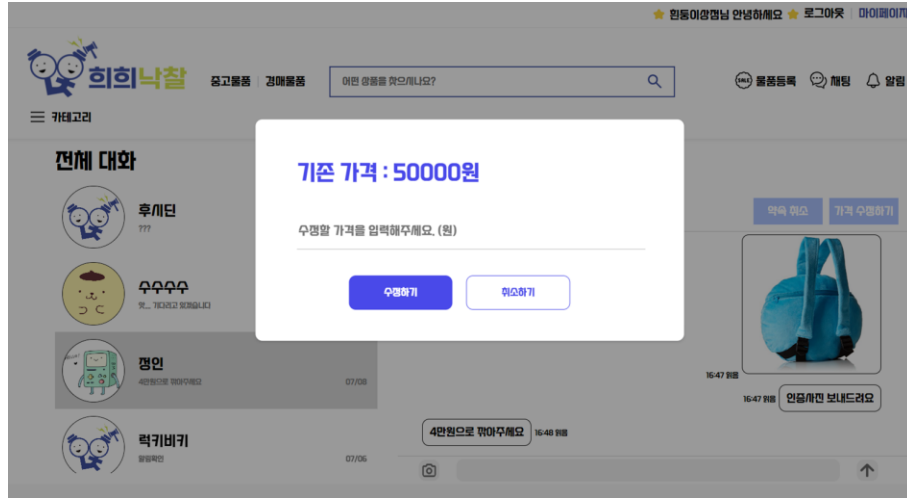
  payButton.addEventListener("click", () => {
    // 판매자 버튼 클릭 이벤트 처리
    editPrice(roomDetail.roomProductDTO.productPrice, roomDetail.roomProductDTO.productSeq);
  });
  buttonDiv.append(payButton);
}
```



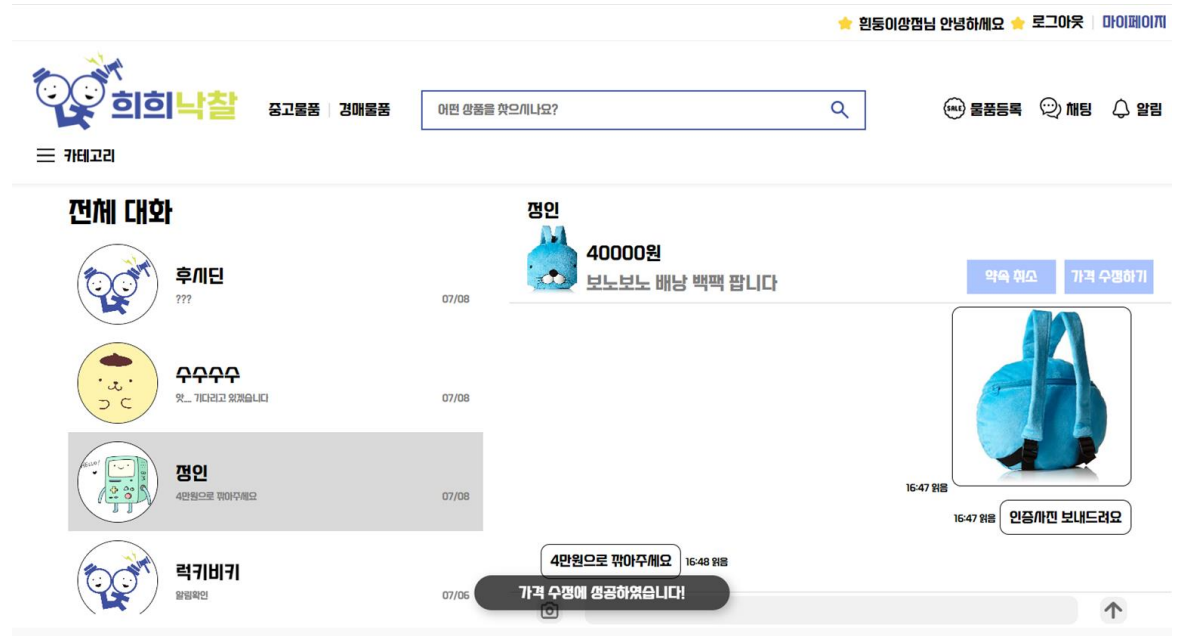


기능 설명 채팅(WebSocket)

ChattingController.java 가격 수정하기 함수



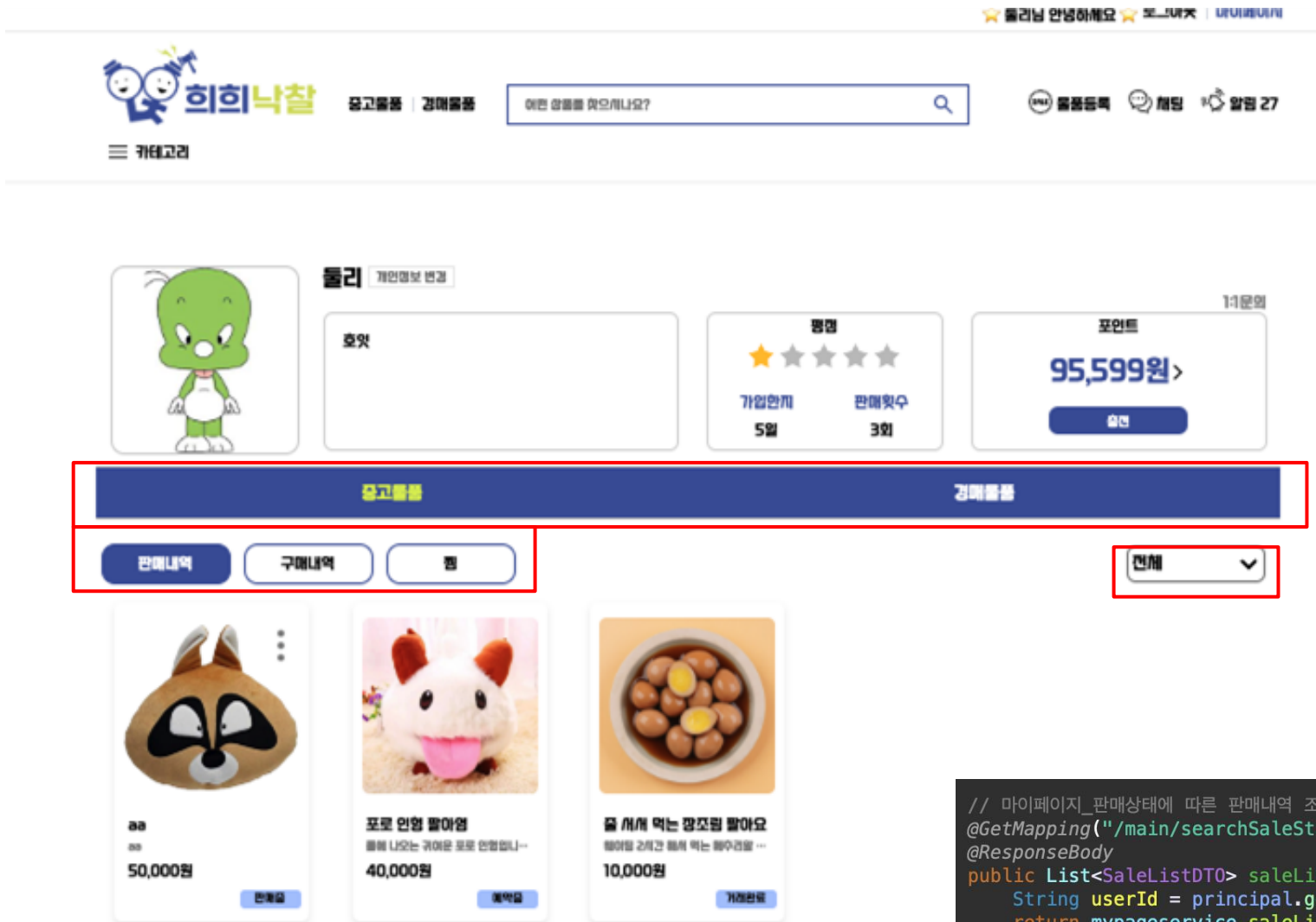
```
// 선택한 채팅방 -> 가격 수정 모달 -> 수정하기 눌렀을때 가격 수정
// 업데이트 성공 시 1 반환
@PutMapping("/price")
@ResponseBody
public int updatePrice(@RequestBody Map<String, Object> map) {
    return cService.updatePrice(map);
}
```





기능 설명 마이페이지

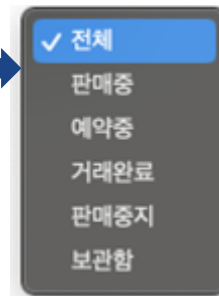
메인화면



```
<select id="saleList" parameterType="map" resultType="salevo">
  SELECT *
  FROM SELL_PRODUCT SELL, IMG_FILES IMG
  WHERE SELL.PRODUCT_SEQ =
  IMG.TABLE_PK
  AND SELL.ID = #{userId}
  AND IMG_CATE_SEQ = 2

  <if test="status!='all'">
    AND SELL.PRO_STATUS = #{status}
  </if>
  <if test="status!='판매보류'">
    AND SELL.PRO_STATUS != '판매보류'
  </if>

  AND IMG.IMG_SEQ IN (SELECT MIN(IMG_SEQ)
  FROM IMG_FILES
  WHERE
  IMG_CATE_SEQ =
  2
  GROUP BY TABLE_PK)
</select>
```



판매 상태에 따른 판매내역 조회

```
// 마이페이지_판매상태에 따른 판매내역 조회 (중고)
@GetMapping("/main/searchSaleStatus")
@ResponseBody
public List<SaleListDTO> saleList(Principal principal, @RequestParam("status") String status) {
  String userId = principal.getName();
  return mypageservice.saleList(status, userId);
}
```



ajax 활용

```
function changeStatus(status) {
    $(".list").empty();
    $(".sub_menu").show();
    $("#deleteJjim").hide();
    $.ajax({
        url: '/heehee/mypage/main/searchSaleStatus',
        method: 'GET',
        data: { 'status': status },
        success: function(data) {
            var salelist = $('#salelist');
            console.log(data);
            var output = '';
            if (data.length === 0) {
                output = "<p class='message'>해당 판매 내역이 없습니다.</p>";
            } else {
                data.forEach(function(sale, index) {
                    var detailUrl = '';
                    if (sale.proStatus === '판매중' || sale.proStatus === '판매보류' || sale.proStatus === '판매중지') {
                        detailUrl = "/heehee/sell/productdetail/" + sale.productSeq;
                    } else {
                        detailUrl = "/heehee/mypage/saledetail/" + sale.productSeq;
                    }
                    var dropdownId = 'dropdown_' + index;
                    var subDropdownId = 'sub_dropdown_' + index;
                    var statusChangeMenu = '';
                    var productBanReason = '';
                    if (sale.proStatus === '판매중') {
                        statusChangeMenu = `
                            
                            <ul id="${dropdownId}" class="dropdown dropdown-menu" style="display: none;">
                                <li onclick="location.href='/heehee/sell/productmodify/${sale.productSeq}'">상품수정</li>
                                <li onclick="sub_dropdown('${subDropdownId}')">상태변경</li>
                                <li onclick="deleteSale(${sale.productSeq})">상품삭제</li>
                            </ul>
                        `;
                    } else if (sale.proStatus === '판매보류') {
                        statusChangeMenu = `
                            
                            <ul id="${dropdownId}" class="dropdown dropdown-menu" style="display: none;">
                                <li onclick="hideSubDropdown('${subDropdownId}')">취소</li>
                                <li onclick="updateStatus(${sale.productSeq}, '판매중')">판매보류</li>
                            </ul>
                        `;
                    } else if (sale.proStatus === '판매중지') {
                        productBanReason = `<div class="productBanReason"><p>판매중지</p></div>`;
                    }

                    // "판매중지" 상태일 때의 배경색 지정
                    var productStyle = sale.proStatus === '판매중지' ? 'background-color: orange;' : '';
                    var imgClass = sale.proStatus === '판매중지' ? 'gray-scale' : '';

                    var formattedPrice = new Intl.NumberFormat('ko-KR').format(sale.productPrice) + '원';

                    output += `
                        <div class="product">
                            <div>${statusChangeMenu}</div>
                            <div class="product_img_div" onclick="location.href='${detailUrl}'">
                                <div class="product_img ${imgClass}">
                                    
                                </div>
                                <div>${productBanReason}</div>
                            </div>
                            <p>${sale.articleTitle}</p>
                            <p class="rankProdIntro">${sale.introduce}</p>
                            <p id="productPrice">${formattedPrice}</p>
                            <span id="s_statusVal_div"><p id="s_statusVal" style="${productStyle}">${sale.proStatus}</p></span>
                        </div>
                    `;
                });
            }
            saleList.html(output);
        },
        error: function(xhr, status, error) {
            alert('데이터를 가져오는 중 오류가 발생했습니다. ');
        }
    });
}

```



기능 설명 마이페이지

메인화면



후렌치파이 팔아요!!!
아주 맛있고 달디단 감귤 후렌치파...
10,000원

판매중



후렌치파이 팔아요!!!
아주 맛있고 달디단 감귤 후렌치파...
10,000원

판매중



후렌치파이 팔아요!!!
아주 맛있고 달디단 감귤 후렌치파...
10,000원

판매중



후렌치파이 팔아요!!!
아주 맛있고 달디단 감귤 후렌치파...
10,000원

판매보류



보노보노 미니 일체형 보조
배터리
엄청 똑신하고 귀여워요!!!
70,000원

판매중지

판매 상품 상태 변경 메뉴바

```
function dropdown(dropdownId, imgElement) {
    var dropdownMenu = $("#" + dropdownId);
    var isVisible = dropdownMenu.is(":visible");

    $(".dropdown-menu").hide();
    $(".sub-dropdown-menu").hide();
    $(".photo_status").attr("src", "https://sh-heehee-bucket.s3.ap-northeast-2.amazonaws.com/images/mypage/photo-status.png");

    if (!isVisible) {
        dropdownMenu.show();
        $(imgElement).attr("src", "https://sh-heehee-bucket.s3.ap-northeast-2.amazonaws.com/images/mypage/photo-close.png");
    }
}

function sub_dropdown(subDropdownId) {
    $(".sub-dropdown-menu").hide();
    $("#" + subDropdownId).toggle();
}

function hideSubDropdown(subDropdownId) {
    $("#" + subDropdownId).hide();
}
```



기능 설명 마이페이지

구매&판매 제품 상세 페이지

예약중



뷰티/미용) 향수/아로마 (그랑핸드 무나키 향수)

그랑핸드 무나키 향수

거래방식: 직거래

진행상황

거래완료

예약중

거래완료

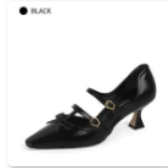
구매가 4,000 원

배송비 0 원

총 결제금액 4,000 원

직거래

예약중



신발) 여성화 (레이첼록스 메리제인 펌프스)

레이첼록스 메리제인 펌프스 6cm 225사이즈

거래방식: 택배

구매가:
배송비:

진행상황

결제완료

발송완료

배송중

배송완료

거래완료

구매가 89,000 원

배송비 4,000 원

총 결제금액 93,000 원

택배

거래 방식에 따라
화면 다르게 구성

```

/* 거래방식(deal)에 따라 progress_ing가 다르게 보임*/
if (dealText === '택배') {

    // 진행 상황에 따라 progress value 값 조절
    // '결제완료 (0%)': 거래내역 있을 때

    // '거래완료 (100%)': proStatus가 '거래완료'일 때
    if (proStatus === '거래완료') {
        $("#graph").attr('value', 100);
    }

    // '배송완료 (75%)': 택배 배송이 완료되었을 때 + 판매 확정날짜가 있을 때
    else if (sCheck != '' || dStatus === '배송완료') {
        $("#graph").attr('value', 75);
    }

    // '배송완료 (50%)': dStatus가 '배송중'일때
    else if (dStatus === '배송중') {
        $("#graph").attr('value', 50);
    }

    // '발송완료 (25%)': 송장을 입력했을 때(송장 내역이 있을 때)
    else if (dNumberText != '') {
        $("#graph").attr('value', 25);
    }

} else {

    // 진행 상황에 따라 progress value 값 조절
    // '거래완료 (100%)': proStatus가 '거래완료'일 때
    if (proStatus === '거래완료') {
        $("#graph").attr('value', 100);
    }

}
}

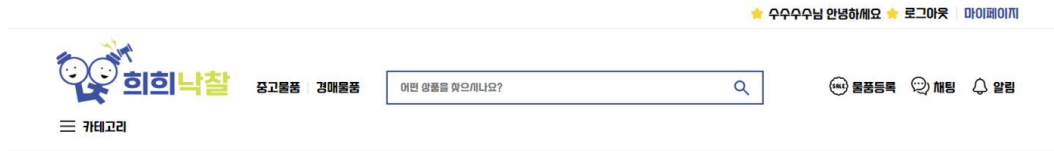
```



기능 설명 마이페이지

<판매 상품>

배송 관리



예약중



송장 입력하기

×

택배사 선택

입력

송장 입력 모달

```
// 마이페이지 - 판매 상품 상세페이지 송장입력 모달
@PostMapping("/saledetail/{productSeq}/insertDelivery")
public String insertDelivery(@PathVariable("productSeq") int productSeq, InsertDeliveryDTO delivery, String buyerId,
    RedirectAttributes redirectAttr) {
    delivery.setBuyerId(buyerId); // 알림 insert 하려면 구매자 아이디 필요
    int result = mypageservice.insertDelivery(delivery, productSeq); // 알림 insert 하려면 상품 시퀀스 필요
    return "redirect:/mypage/saledetail/{productSeq}";
}
```

```
<div class="profileModal" id="deliveryModal">
  <div class="profileModalBody">
    
    <form class="modal_form"
      action="{path}/mypage/saledetail/{saleDetail.productSeq}/insertDelivery"
      method="post">
      <p id="dModal_p">송장 입력하기</p>
      <div class="input" id="input_d">
        <div class="Modal_delivery">
          <select name="dCompanySeq">
            <option>택배사 선택</option>
            <c:forEach var="dc" items="{dcOption}">
              <option value="{dc.DCompanySeq}">{dc.DCompany}</option>
            </c:forEach>
          </select>
        </div>
        <div class="Modal_delivery">
          <input type="number" placeholder="운송장 번호 -없이 입력" name="dNumber"maxlength="9" oninput="maxLengthCheck(this)">
        </div>
        <input type="hidden" name="buyerId" value="{saleDetail.buyerId}">
        <input type="hidden" name="sSeq" value="{saleDetail.SSeq}">
      </div>
      <div class="btn_modal" id="d_btn">
        <button class="btn_submit" id="btn_delivery">입력</button>
      </div>
    </form>
  </div>
</div>
```

거래방식: 택배



기능 설명 마이페이지

<판매 상품>


배송 관리

★ 우수수님 안녕하세요 ★ 로그아웃 마이페이지

히히낙찰 중고물품 경매물품

☰ 카테고리

예약중



식품) 간편조리식품 (열치볶음)

집밥 시리즈 열치볶음

거래방식: 택배

택배가: 우체국택배
송장번호: 110429698

click

거래완료

송장 내용 업데이트

결제완료	발송완료	배송중	배송완료	거래완료
진행상황				
판매가				5,000원
배송비				2,500원
총 판매금액				7,500원

히히낙찰
(03939) 서울 마포구 월드컵북로4길 77
heehee@nakchal.co.kr
사업자 등록번호: 260-87-01655
통신번호: 2021-서울마포-0387호

고객센터
02-6392-0044 (유료)
평일 오전 9시 - 18시 (주말 / 공휴일 휴무, 점심시간 13시 - 14시)

히히낙찰은 통신판매용 계약이며, 통신판매의 당/해가 아닙니다.
전자상거래 피해계의 소비자보호에 관한 법률 등 관련 법령 및 히히낙찰의 약관에 따라 상품, 상품정보, 거래에 관한 책임은 개별 판매자에게 귀속하고, 히히낙찰은 원칙적으로 회원과 거래에 대하여 책임을 지지 않습니다.
다만, 히히낙찰이 직접 판매하는 상품에 대한 책임은 히히낙찰에게 귀속합니다.

© HeeHeenakchal, Inc All rights reserved.




★ 우수수님 안녕하세요 ★ 로그아웃 마이페이지

히히낙찰 중고물품 경매물품

☰ 카테고리

예약중



식품) 간편조리식품 (열치볶음)

집밥 시리즈 열치볶음

거래방식: 택배

택배가: 우체국택배
송장번호: 110429698

진행상황
구매자가 아직 거래 완료 버튼을 누르지 않았습니다.

결제완료	발송완료	배송중	배송완료	거래완료
진행상황				
판매가				5,000원
배송비				2,500원
총 판매금액				7,500원

히히낙찰
(03939) 서울 마포구 월드컵북로4길 77
heehee@nakchal.co.kr
사업자 등록번호: 260-87-01655
통신번호: 2021-서울마포-0387호

고객센터
02-6392-0044 (유료)
평일 오전 9시 - 18시 (주말 / 공휴일 휴무, 점심시간 13시 - 14시)

히히낙찰은 통신판매용 계약이며, 통신판매의 당/해가 아닙니다.
전자상거래 피해계의 소비자보호에 관한 법률 등 관련 법령 및 히히낙찰의 약관에 따라 상품, 상품정보, 거래에 관한 책임은 개별 판매자에게 귀속하고, 히히낙찰은 원칙적으로 회원과 거래에 대하여 책임을 지지 않습니다.
다만, 히히낙찰이 직접 판매하는 상품에 대한 책임은 히히낙찰에게 귀속합니다.

© HeeHeenakchal, Inc All rights reserved.

거래방식:
택배



기능 설명 마이페이지

구매 & 판매 확정

판매자, 구매자
모두 거래 완료 버튼 클릭하
면

```
// 마이페이지 - 판매 상품 상세페이지:거래완료 버튼
@PostMapping("/saledetail/updateSCheck")
@ResponseBody
public ResponseEntity<?> updateSCheck(int proSeq, HttpServletResponse response) {
    response.setContentType("text/plain;charset=UTF-8");
    return mypageservice.updateSCheck(proSeq);
}
```

```
// 마이페이지 - 구매 상품 상세페이지:거래완료 버튼
@PostMapping("/purchasedetail/updatePCheck")
@ResponseBody
public ResponseEntity<?> updatePCheck(int proSeq, HttpServletResponse response) {
    response.setContentType("text/plain;charset=UTF-8");
    return mypageservice.updatePCheck(proSeq);
}
```

```
create or replace TRIGGER update_ps_date_trigger
BEFORE UPDATE OF P_CHECK, S_CHECK ON DEAL_HISTORY
FOR EACH ROW
BEGIN
    -- Update PS_DATE if both P_CHECK and S_CHECK are not null
    IF :NEW.P_CHECK IS NOT NULL AND :NEW.S_CHECK IS NOT NULL THEN
        :NEW.PS_DATE := SYSDATE;
    END IF;

    -- Update AUC_PRODUCT if AUC_SEQ is not null and PS_DATE is updated
    IF :NEW.PS_DATE IS NOT NULL AND :OLD.AUC_SEQ IS NOT NULL THEN
        UPDATE AUC_PRODUCT
        SET AUC_STATUS = '거래완료'
        WHERE PRODUCT_SEQ = :OLD.AUC_SEQ;
    END IF;

    -- Update SELL_PRODUCT if SELL_SEQ is not null and PS_DATE is updated
    IF :NEW.PS_DATE IS NOT NULL AND :OLD.SELL_SEQ IS NOT NULL THEN
        UPDATE SELL_PRODUCT
        SET PRO_STATUS = '거래완료'
        WHERE PRODUCT_SEQ = :OLD.SELL_SEQ;
    END IF;
END;
```

트리거 이용해 거래 확정일자 업데이트

우우우님 안녕하세요 ✨ 로그아웃 마이페이지

중고물품 경매물품

👤 상품등록 🗨 채팅 🔔 알림

☰ 카테고리

거래완료

익종) 간편조리식품 (멸치볶음) 2024-07-08

집밥 시리즈 멸치볶음

거래량: 1백

택배사: 우체국택배
송장번호: 110429698

발송완료	배송중	배송완료	거래완료
			5,000원
			2,500원
			7,500원

거래방식:
택배



기능 설명 마이페이지

평점

거래완료



식품 > 간편조리식품 (함박 스테이크) 2024-07-08

진짜 맛있는 함박 스테이크

거래방식: 택배

택배사: 우체국택배
송장번호: 123456789

진행상황

결제완료	발송완료	배송중	배송완료	평가
구매가				1,000원
배송비				500원
총 결제금액				15,500원

평가하기

```
document.addEventListener('DOMContentLoaded', function () {
  var userRating = ${info.userRating}; // EL문법때문에 js파일로 따로 못뺌
  var stars = document.querySelectorAll('#seller_score .star');
  alert(userRating);
  for (var i = 0; i < userRating; i++) {
    stars[i].src = 'https://sh-heehee-bucket.s3.ap-northeast-2.amazonaws.com/images/sell/star1.png';
  }
});
```

판매자 평점 주기



등록하기

판매자 평점 주기



등록하기



기능 설명 마이페이지

평점

```
// 평점 매기기
@PostMapping("/rating")
@ResponseBody
@Transactional
public Map<String, Object> userRating(RateAdminDTO rateDTO, Model model, Principal principal) throws IOException {
    Map<String, Object> response = new HashMap<String, Object>();

    String userId = principal.getName();
    rateDTO.setUserid(userId);
    int result = mypageservice.rating(rateDTO);
    mypageservice.updateRate(rateDTO);

    // response.put("success", true);
    // return response;

    // 성공 실패 분류 로직 구현

    if (result == 1) {
        response.put("success", true);
        return response;
    } else {
        return response;
    }
}
```

```
<insert id="insertRating"
    parameterType="com.shinhan.heehee.dto.request.RateAdminDTO">
    <selectKey keyProperty="selectedRating" order="BEFORE"
        resultType="int">
        SELECT SEQ_RATE.NEXTVAL FROM DUAL
    </selectKey>
    INSERT INTO RATE_ADMIN
    VALUES(#{selectedRating}, #{productSeq},
    #{sellerId}, #{userRating})
</insert>

<update id="updateRating"
    parameterType="com.shinhan.heehee.dto.request.RateAdminDTO">
    UPDATE HH_USER
    SET USER_RATING = ROUND(#{selectedRating},
    0)
    WHERE ID = #{sellerId}
</update>
```

평점 삽입 후, 해당 유저의
평균 평점을 계산하여 갱신함



기능 설명 마이페이지

개인정보 수정

```
// 마이페이지 프로필 수정 페이지: 프로필 수정
@PostMapping("/profile/updateProfile")
public String editProfile(@RequestParam("profileImg") MultipartFile profileImages,
    @RequestParam("originalProfileImg") String originalProfileImg, @RequestParam("nickName") String nickName,
    @RequestParam("userIntroduce") String userIntroduce, Principal principal, RedirectAttributes redirectAttr) {
    String userId = principal.getName();
    String image = "";
    try {
        if (profileImages != null && !profileImages.isEmpty()) {
            // AWS S3에 이미지 업로드
            image = s3Service.uploadOneObject(profileImages, "images/mypage/");
        } else if (originalProfileImg != null) {
            image = originalProfileImg;
        }

        // 데이터베이스에 프로필 정보 저장
        mypageservice.editProfile(nickName, userIntroduce, image, userId);
    } catch (IOException e) {
        // 업로드 실패 시 처리할 로직
        e.printStackTrace();
    }
    return "redirect:/mypage/profile";
}
```

```
<form id="profileForm" action="${path}/mypage/profile/updateProfile"
method="post" enctype="multipart/form-data">
<div class="profileImg">

</div>
<input type="hidden" id="original" name="originalProfileImg"
value="${profile.profileImg}" required="required"> <input
type="file" id="fileInput" name="profileImg" accept="image/*"
style="display: none;">
<div class="btn_profile">
<input type="button" class="btn_image" id="btn_modify"
value="사진변경" onclick="chooseFile();">
<input type="button" class="btn_image" id="btn_delete" value="삭제"> -->
</div>
<table class="input_left">
<tbody>
<tr>
<th>닉네임</th>
<td><input type="text" id="nickName" name="nickName"
value="${profile.nickName}" required="required"
oninput="changeBtn('${profile.nickName}')" maxlength="16">
</td>
</tr>
<tr>
<th>소개글</th>
<td><textarea id="userIntroduce"
name="userIntroduce" maxlength="100">${profile.userIntroduce}</textarea></td>
</tr>
</tbody>
</table>
<div class="btn_profile">
<input type="submit" id="btn_submit" value="적용">
<p id="btn_cancel"
onclick="location.href='${path}/mypage/profile'">취소</p>
</div>
</form>
```

★ 둘러님 안녕하세요 ★ 로그아웃 | 마이페이지



광고물품 경매물품

어떤 상품을 찾으시나요?

회원등록 채팅 알림 27

☰ 카테고리

프로필 수정



사진변경

닉네임	<input type="text" value="둘리"/>
소개글	<input type="text" value="호맛"/>

적용 취소

내 프로필



둘리

호맛

이미지 클릭



기본 정보	
아이디	dlenf0924
이름	탈퇴 테스트
이메일	aa@aa

전화번호	010-2235-0970 >
내 계좌	국민은행 111-111111-1111 >
주소	서울 마포구 가암대로 1701호 >

비밀번호 수정

회원 탈퇴



기능 설명 마이페이지

개인정보 수정

★ 둘러님 안녕하세요 ★ 로그아웃 | 마이페이지



종교물품 경매물품

어떤 상품을 찾으시나요?



물품등록



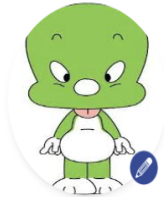
채팅



알림 27

카테고리

내 프로필



둘리

호빗

기본 정보

아이디 dlienfi0924

이름 탈퇴 테스트

이메일 aa@aa

전화번호 010-2235-0970 >

내 계좌 국민은행 111-1111-1111 >

주소 서울 마포구 가양대로1 701호 >

비밀번호 수정

회원 탈퇴



전화번호 변경

010-4842-3750 인원번호 발송

인원번호를 입력해주세요. 인원하기

저장

현재 계좌: 신한은행 12345678

신한은행

저장

비밀번호 변경

수정하기

취소하기

주소 변경

서울 마포구 가양대로 8 검색

201호

저장

정말 탈퇴하시겠습니까?

계정의 모든 정보는 삭제되며 복구되지 않습니다.

아니오

네



기능 설명 마이페이지

포인트 충전 내역 조회

```
function searchPointList(){
    var month = $("#month_option").val();

    $.ajax({
        url: '/heehee/mypage/pointlist/searchPoint',
        method: 'GET',
        data: { 'month': month },
        success: function(data) {
            var pointHistory = $('#point_history');
            var output = '';
            if (data.length === 0) {
                output = "<p class='message'>충전 내역이 없습니다.</p>";
            } else {

                data.forEach(function(p) {
                    let dt = new Date(p.payDate).toLocaleDateString();
                    const formattedPoint = new Intl.NumberFormat().format(p.amount);
                    output += `
                    <div class="detail">
                        <p class="detail_date">${dt}</p>
                        <p id="detail_point">${formattedPoint}원</p>
                    </div>`;
                });
            }

            pointHistory.html(output);
        },
        error: function(xhr, status, error) {
            alert('데이터를 가져오는 중 오류가 발생했습니다.');
```

```
// 포인트 내역 페이지: 월별조회
@GetMapping(value = "/pointlist/searchPoint", produces = "application/json")
public @ResponseBody List<PointListDTO> searchPoint(Principal principal, @RequestParam String month) {
    String userId = principal.getName();
    String[] parts = month.split("-");
    String year = parts[0];
    String monthPart = parts[1];
    List<PointListDTO> result = mypageservice.searchPoint(userId, year, monthPart);
    return result;
}
```

현재 포인트
412,000원

충전하기

2024년 7월

월별 조회

2024. 7. 8.	10,000원
2024. 7. 8.	2,000원
2024. 7. 8.	400,000원



기능 설명 마이페이지

포인트 충전 내역

현재 포인트
0원
충전하기

2024년 7월

충전 내역이 없습니다.

현재 포인트: 0원

10000

충전하기

신용카드

KG 이니시스

이용약관 전체동의

전자금융거래 이용약관 동의

개인정보의 수집 및 이용안내 동의

개인정보 제공 및 위탁안내 동의

약관보기

현대카드 +	삼성카드 +	
KB국민	비씨(페이북)	신한(SOL페이)
롯데카드	하나Pay(하나)	농협(NH페이)
씨티카드	하나Pay(외환)	
그외카드		

통합인증 서비스

이니백과

카드연동 | 동주PKD | 원금영수증 | 계좌당첨 | 카드연동 | 카드연동 | 카드연동 | 카드연동

점말하면 모바일 1년 무료

다음

```

function chargePoint(point) {
  var amount = $("#charge").val();
  payForPoint("포인트 충전", amount, point);
}

function payForPoint(payName, amount, point) {
$.ajax({
  url: '/heehee/pay/before',
  method: 'POST',
  data: {
    "payName": payName,
    "amount": amount
  },
  success: function(data, status, xhr) {
    payment(payName, amount, data, point);
    return true;
  },
  error: function(data, status, err) {
    console.log(err);
  }
});
}

function payment(payName, newPoint, payInfo, point) {
  IMP.init("imp22447463");
  IMP.request_pay({
    pg: "html5_inicis.INIpayTest", //최소 시 html5_inicis.INIpayTest 거래
    pay_method: "card",
    merchant_uid: payInfo.paySeq, //상점에서 생성한 고유 주문번호
    name: payName,
    amount: 1,
    buyer_email: payInfo.buyerEmail,
    buyer_name: payInfo.buyerId,
    buyer_tel: payInfo.buyerTel, //필수 피라미터입니다.
    buyer_addr: payInfo.buyerTel,
    buyer_postcode: "123-456",
    m_redirect_url: "{모바일에서 결제 완료 후 리디렉션 될 URL}",
    escrow: true, //에스크로 결제인 경우 설정
    vbank_due: "20240725",
    bypass: {
      acceptmethod: "noeasypay", // 간편결제 버튼을 통합결제형에서 제외(PC)
      P_RESERVED: "noeasypay=Y", // 간편결제 버튼을 통합결제형에서 제외(모바일)
    },
    period: {
      from: "20240101", //YYYYMMDD
      to: "20241231", //YYYYMMDD
    },
  }, function(rsp) {
    if (rsp.success) {
      completePayment(payInfo.paySeq, newPoint, point);
      return true;
    }
  });
}

function completePayment(paySeq, newPoint, point) {
$.ajax({
  url: '/heehee/pay/complete',
  method: 'PUT',
  contentType: 'application/json',
  data: JSON.stringify({ "paySeq": paySeq }),
  success: function (data, status, xhr) {
    console.log(data);
    console.log(status);
    console.log(xhr);
    fetch("/heehee/mypage/chargePoint",{
      method: "PUT",
      headers: { 'Content-Type': "application/json" },
      body: JSON.stringify({
        "point": newPoint,
        "userPoint": point
      })
    })
    .then(resp => resp.text())
    .then(result => console.log(result))
    .catch(err => console.log(err));
    return true;
  },
  error: function (data, status, err) {
    console.log(err);
  }
});
setTimeout(() => window.location.reload(), 1000);
}

```



1:1 Q&A

- 제품 사용, 오염, 전용 박스 손상, 라벨 제거, 사은품 및 부속 사용/분실 시, 교환/환불이 불가능 합니다.
- 고객님의 주문내역을 선택, 질문이 필요한 상품을 선택하시면 1:1상담이 가능합니다.
- 주문취소/교환/환불은 마이페이지>주문내역에서 신청하실 수 있습니다.
- 1:1문의 처리 내역은 마이페이지>1:1문의를 통해 확인하실수 있습니다.
- 상품 정보(사이즈, 실측, 예상 배송일 등) 관련 문의는 해당 상품 문의에 남겨셔야 빠른 답변이 가능합니다.

1:1문의하기

문의제목

문의유형

-
-
-
-
-
-

문의내용

사진첨부 최대 (3개)

첨부 이미지 미리보기

```
function previewImages(event) {
  var files = event.target.files;

  // 이미 선택된 파일 수 체크
  var selectedFiles = $('#preview_container').find('.preview_item').length;

  // 최대 3개
  if (selectedFiles + files.length > 3) {
    alert("최대 3개까지만 첨부할 수 있습니다.");
    return;
  }

  // 미리보기 컨테이너 초기화
  $('#preview_container').empty();

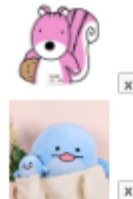
  for (var i = 0; i < files.length; i++) {
    var file = files[i];
    var reader = new FileReader();

    reader.onload = function(e) {
      var img = $('<img>').attr('src', e.target.result);
      var removeBtn = $('<button type="button" class="remove_img">x</button>')
        .click(function() {
          $(this).parent().remove();
        });

      var preview = $('<div class="preview_item">').append(img).append(removeBtn);
      $('#preview_container').append(preview);
    };

    reader.readAsDataURL(file);
  }
}
```

사진첨부 최대 (3개)





나의 문의

○ [태움] 배송에 대한 문의

답변대기

○ [태움] 배송이 안옵니다

답변완료

○ [태움] 배송이 안왔어요!

답변완료



```

$( ".header" ).on( "click", show );
$("#myQna .header").each(function() {
  var $ansStatus = $(this).find(".ansStatus");
  var $qnaAns = $(this).next(".content").find(".qnaAns");

  if ($qnaAns.text().trim() !== "") {
    $ansStatus.text("답변완료");
  } else {
    $ansStatus.text("답변대기");
    $ansStatus.css("background-color", "#ABC3FF");
  }
});
$("#btn_submit").on("click", insertQna);

```

답변 유무 표시



기능 설명 마이페이지

FAQ 게시판

FAQ 자주 묻는 질문

- 전체
- 배송
- 주문/결제
- 회원정보
- 상품확인
- 취소/교환/환불
- 서비스

무엇을 도와드릴까요?

구분	제목
배송	직거래는 어떻게 진행되나요?
배송	배송조회가 정상적으로 되지 않는다면?
배송	결제방식을 바꾸고 싶어요.
배송	택배 박스가 파손되어 배송됐어요.
배송	배송 조회는 어떻게 하나요?
배송	배송완료된 상품을 수령받지 못했어요.
주문/결제	계좌번호를 삭제할 수는 없나요?
주문/결제	다른 결제 수단으로 선택이 되지 않아요.
주문/결제	회회낙찰에서 결제안적이 없는데 결제 문자를 받았어요.
주문/결제	등록된 계좌번호를 변경하고 싶어요.
회원정보	가입 시 본인인증 문자가 오지 않아요.
회원정보	비회원으로 구매할 수는 없나요?
회원정보	회원 탈퇴는 어떻게 해야하나요?
회원정보	소셜 로그인은 어떻게 이용하나요?



```
// 마이페이지-FAQ_문의유형에 따른 FAQ조회
@GetMapping("/faqBoard/faqOption")
@ResponseBody
public List<FaQDTO> faqOption(int option) {
    return mypageservice.faqOption(option);
}
```

```
function faqOption(option) {
    $("#searchInput").val("");
    $.ajax({
        url: '/heehee/mypage/faqBoard/faqOption',
        method: 'GET',
        data: { 'option': option },
        success: function(data) {
            var faqList = $('#faqList');
            var output = '';
            if (data.length === 0) {
                $('#message').html("<p class='message'>해당 유형의 FAQ가 없습니다.</p>");
            } else {
                data.forEach(function(fa) {
                    $('#message').empty();
                    output += `
                    <tr class="question">
                        <td class="type">${fa.qnaOption}</td>
                        <td class="title">${fa.faqContent}</td>
                    </tr>
                    <tr class="answer">
                        <td class="type">답변</td>
                        <td class="title">${fa.faqAns}</td>
                    </tr>`;
                });
            }
            faqList.html(output);
            $(".answer").hide();
            $(".question").on("click", show);
        },
        error: function(xhr, status, error) {
            alert('데이터를 가져오는 중 오류가 발생했습니다.');
```



기능 설명 마이페이지

FAQ 게시판

FAQ 자주 묻는 질문

- 전체
- 배송
- 주문/결제
- 회원정보
- 상품확인
- 취소/교환/환불
- 서비스



구분	제목
배송	픽거래는 어떻게 진행되나요?
배송	배송조회가 정상적으로 되지 않는다면?
배송	결제방식을 바꾸고 싶어요.
배송	택배 박스가 파손되어 배송됐어요.
배송	배송 조회는 어떻게 하나요?
배송	배송완료된 상품을 수령받지 못했어요.
주문/결제	계좌번호를 삭제할 수는 없나요?
주문/결제	다른 결제 수단으로 선택이 되지 않아요.
주문/결제	회회낙할때 결제안적이 있는데 결제 문자를 받았어요.
주문/결제	등록된 계좌번호를 변경하고 싶어요.
회원정보	가입 시 본인인증 문자가 오지 않아요.
회원정보	비회원으로 구매할 수는 없나요?
회원정보	회원 탈퇴는 어떻게 해야하나요?
회원정보	소셜 로그인은 어떻게 이용하나요?

faq 제목 검색 기능

```

function search() {
  var keyword = $('#searchInput').val();
  $.each($(".title"), function(index, item) {
    var text = $(item).text();
    if (!text.includes(keyword))
      $(this).parent().addClass("searchStyle");
    else
      $(this).parent().removeClass("searchStyle");

    $("#faqTable").removeClass("searchStyle");
  });
}

```



기능 설명 관리자 페이지

메인페이지 구성

★ 손님님 안녕하세요 ★ 로그아웃 | 관리자 페이지



관리자 홈

회원정보 관리

상품 관리

고객 지원

전체 주문통계

전체 주문현황(일반/경매)				일반상품 주문현황				경매상품 주문현황			
일반 주문건수	일반 주문액	경매 주문건수	경매 주문액	판매중	예약중	거래완료	판매중지	입찰	낙찰	거래완료	판매중지
51	5,585,737	65	2,254,685	16	22	8	1	12	29	9	3

최근 등록내역

상세정보 바로가기

등록번호	구매자ID	판매자ID	품목명	거래상태	거래금액	등록일
204	sinsang		보노보노 세트 팔아여~~	판매중	50,000	2024-7-9
203	b		후렌치파이 팔아요!!!	판매보류	10,000	2024-7-9
202	hsyaa11		레인부츠 새상품 팝니다	판매중	30,000	2024-7-8
201	navertest		שלגי 인형 팝니다.	판매보류	2,000,000	2024-7-8
200	a		비모 그림 팔아요	판매보류	2,000	2024-7-8

최근 문의내역

1:1문의 바로가기

작성자ID	처리현황	제목	작성일
hsyaa11	답변완료	그냥 남기고 싶어/꺼 남겨봐요	2024-7-9
a	답변완료	배송이 안와요!	2024-7-8
swimmming	처리대기	식물 팔고 싶어요	2024-7-8
wonpil	답변완료	문의합니다.	2024-7-8
dldl	처리대기	배송이 안올니다	2024-7-8

최근 회원가입

회원정보 관리 바로가기

회원ID	회원명	이메일	전화번호	가입일
hsyaa11	황수영	test123@test.com	010-9273-5460	2024-7-8
sinsang321	신연상	sinsang@gmail.com	010-9290-2752	2024-7-8
sinsang0452	신연상	skqlgogol2@hanmail.net	010-9290-2752	2024-7-8
chlwoaud	최재영	chlwoaud@gmail.com	010-8904-4778	2024-7-8
thsdhfl123	손동희	ghost1229@naver.com	010-2235-0970	2024-7-8

```
$(document).ready(function() {
  // 페이지 로드 시 테이블을 초기화
  loadTotalOrder();
  loadProStatus();
  loadAucStatus();
  loadRecentProductTable();
  loadRecentQuestionTable();
  loadRecentJoinTable();

  //전체 주문통계 테이블들의 데이터 Ajax로 가져오기
  //전체 주문현황 (일반/경매)
  function loadTotalOrder(){
    $.ajax({
      url: '/heehee/admin/searchTotalOrder',
      method: 'GET',
      dataType: 'json',
      success: function(data) {
        var dbData = data[0];
        $('#sellProCtn').text(dbData.sellProCtn.toLocaleString());
        $('#sellProPriceSum').text(dbData.sellProPriceSum.toLocaleString());
        $('#aucProCtn').text(dbData.aucProCtn.toLocaleString());
        $('#aucProPriceSum').text(dbData.aucProPriceSum.toLocaleString());
      },
      error: function(xhr, status, error) {
        console.error("데이터 가져오기 실패:", error);
        alert('데이터 로드에 실패하였습니다.');
      }
    });
  }

  /* 최근 문의내역 조회 (기능: 최근 등록 문의관련 모니터링용 대시보드 항목 조회) */
  SELECT ID , QNA_ANS , QNA_TITLE , QNA_TIME
  FROM (SELECT ID , QNA_ANS , QNA_TITLE , QNA_TIME
        FROM QNA_BOARD
        ORDER BY QNA_TIME DESC
       )
  WHERE ROWNUM <= 5;

  /* 최근 회원가입 조회 (기능: 최근 가입회원 모니터링용 대시보드 항목 조회) */
  SELECT ID , NAME , EMAIL , PHONE_NUM , CREATE_DATE
  FROM (SELECT ID , NAME , EMAIL , PHONE_NUM , CREATE_DATE
        FROM HH_USER
        ORDER BY CREATE_DATE DESC
       )
  WHERE ROWNUM <= 5;
});
```

최근 5개 항목 호출 SQL 처리

Ajax활용 테이블내용 동적 업데이트

요약 데이터의 본 페이지로 이동

```
<div id="recentProductContainer">
  <div id="mainTitleContainer"><p class="mainTitle">최근 등록내역</p><button class="mainMenuBtn" onclick="location.href='${path}/admin/product'">상세정보 바로가기</button></div>
  <div class="recentProductDiv">
    <table class="recentProductTable">
      <thead>
        <tr>
          <th>등록번호</th> <!-- AdminProductDTO -->
          <th>구매자ID</th>
          <th>판매자ID</th>
          <th>품목명</th>
          <th>거래상태</th>
          <th>거래금액</th>
          <th>등록일</th>
        </tr>
      </thead>
      <tbody id="recentProductTableBody">
        <!-- Ajax로 동적 업데이트 -->
      </tbody>
    </table>
  </div>
  <!-- recentOrder -->
</div>

<div id="recentQuestionContainer">
  <div id="mainTitleContainer"><p class="mainTitle">최근 문의내역</p><button class="mainMenuBtn" onclick="location.href='${path}/admin/qnaManager'">1:1문의 바로가기</button></div>
  <div id="recentQuestionDiv">
    <table class="recentQuestionTable">
      <thead>
        <tr>
          <th>작성자ID</th> <!-- AdminQnaManagerDTO -->
          <th>처리현황</th>
          <th>제목</th>
          <th>작성일</th>
        </tr>
      </thead>
      <tbody id="recentQuestionTableBody">
        <!-- Ajax로 동적 업데이트 -->
      </tbody>
    </table>
  </div>
  <!-- recentQuestion -->
</div>
```



기능 설명 관리자 페이지

구매자가 없는 상태는 구매자ID 공백 처리

최근 등록내역

[상세정보 바로가기](#)

등록번호	구매자ID	판매자ID	품목명	거래상태	거래금액	등록일
204		sinsang	보노보노 세트 팔아어~~	판매중	50,000	2024-7-9
203		b	후렌시파이 팔아요!!!	판매보류	10,000	2024-7-9
202		hsyaa11	레인부스 새상품 팝니다	판매중	30,000	2024-7-8
201		navertest	쏟기 인형 팝니다.	판매보류	2,000,000	2024-7-8
200		a	비모 그림 팔아요	판매보류	2,000	2024-7-8

null 값 출력 방지

Javascript 코드

```
//최근 상품 등록내역 테이블 데이터를 Ajax로 가져오기
function loadRecentProductTable() {
  $.ajax({
    url: "/heehee/admin/searchRecentProduct",
    method: 'GET',
    success: function(data) {
      var tableBody = $('#recentProductTableBody');
      tableBody.empty();

      data.forEach(function(item) {
        var buyerId = (!item.buyerId || item.buyerId.trim() === '') ? '' : item.buyerId ;
        var createDatePas = new Date(item.createDate).toLocaleDateString();
        var createDateRev = createDatePas.substring(0, createDatePas.length -1).replaceAll(".", "-");

        var row =
          "<tr>" +
            "<td>" + item.productSeq + "</td>" +
            "<td>" + buyerId + "</td>" +
            "<td>" + item.id + "</td>" +
            "<td>" + item.articleTitle + "</td>" +
            "<td>" + item.proStatus + "</td>" +
            "<td>" + item.productPrice.toLocaleString() + "</td>" +
            "<td>" + createDateRev + "</td>" +
          "</tr>";
        tableBody.append(row);
      });
    },
    error: function(xhr, status, error) {
      console.log('AJAX request failed:', status, error);
      alert('데이터를 가져오는 중 오류가 발생했습니다.');
```



기능 설명 관리자 페이지

JavaScript를 통한 Dropdown 메뉴 및 항목별 hover 구현

```
document.addEventListener('DOMContentLoaded', function() {
  var dropdowns = document.querySelectorAll('.dropdown > a');

  dropdowns.forEach(function(dropdown) {
    dropdown.addEventListener('click', function() {
      var submenu = this.nextElementSibling;
      var isVisible = submenu.style.display === 'block';

      // Hide all submenus
      document.querySelectorAll('.submenu').forEach(function(sub) {
        sub.style.display = 'none';
      });

      // Toggle the clicked submenu
      if (!isVisible) {
        submenu.style.display = 'block';
      } else {
        submenu.style.display = 'none';
      }
    });

    // Hover 효과 구현
    dropdown.addEventListener('mouseenter', function() {
      dropdown.classList.add('hovered');
    });

    dropdown.addEventListener('mouseleave', function() {
      dropdown.classList.remove('hovered');
    });
  });

  // 클릭된 메뉴의 background 변경
  const clickableLinks = document.querySelectorAll(".movePage, .dropdown > a");

  function handleClick(event) {
    // div에서 모든 "click" 클래스 제거
    clickableLinks.forEach((link) => {
      link.classList.remove("click");
    });
  }

  // 클릭한 div만 "click" 클래스 추가
  event.target.classList.add("click");
}

clickableLinks.forEach((link) => {
  link.addEventListener("click", handleClick);
});

// Hide all submenus on page load
document.querySelectorAll('.submenu').forEach(function(sub) {
  sub.style.display = 'none';
});
});
```

관리자 홈

회원정보 관리

상품 관리

고객 지원

사이드메뉴 클릭

관리자 홈

회원정보 관리

상품 관리

일반상품 관리

상세조회

경매상품 관리

상세조회

카테고리 관리

고객 지원

사이드메뉴 Dropdown menu

Dropdown 메뉴 구성을 위한 HTML 구조

```
<head>
<meta charset="UTF-8">
<title>sideMenu</title>
<link rel="stylesheet" href="{path}/resources/css/admin/sideMenu.css?after">
<script type="text/javascript" src="{path}/resources/js/admin/sideMenu.js"></script>
</head>
<body>
  <div id="sideMenuContainer">
    <ul>
      <li><a class="movePage nonClick" href="{path}/admin/main">관리자 홈</a></li>
      <li class="dropdown">
        <a class="dropdownMenu nonClick" href="javascript:void(0)">회원정보 관리</a>
        <ul class="submenu">
          <li><a class="movePage" href="{path}/admin/user">회원정보 관리</a></li>
          <li><a class="movePage" href="{path}/admin/ban">이용상태 관리</a></li>
        </ul>
      </li>
      <li class="dropdown">
        <a class="dropdownMenu nonClick" href="javascript:void(0)">상품 관리</a>
        <ul class="submenu">
          <li><a class="middleMenu">일반상품 관리</a></li>
          <li><a class="movePage" href="{path}/admin/product">상세조회</a></li>
          <li><a class="middleMenu">경매상품 관리</a></li>
          <li><a class="movePage" href="{path}/admin/auction">상세조회</a></li>
          <li><a class="movePage" href="{path}/admin/category">카테고리 관리</a></li>
        </ul>
      </li>
      <li class="dropdown">
        <a class="dropdownMenu nonClick" href="javascript:void(0)">고객 지원</a>
        <ul class="submenu">
          <li><a class="middleMenu">문의사항 관리</a></li>
          <li><a class="movePage" href="{path}/admin/qnaManager">1:1 상담문의</a></li>
          <li><a class="middleMenu">FAQ 관리</a></li>
          <li><a class="movePage" href="{path}/admin/faqManager">FAQ 내용관리</a></li>
          <li><a class="middleMenu">문의 유형 관리</a></li>
          <li><a class="movePage" href="{path}/admin/questionManager">문의 유형 관리</a></li>
        </ul>
      </li>
    </ul>
  </div>
</body>
```

Dropdown 메뉴 구성을 위한 CSS

```
li a.movePage:hover {
  background-color: #6381a1;
  color: white;
}

li a.dropdownMenu:hover {
  background-color: #6381a1;
  color: white;
}

/* 클릭된 메뉴에 대한 컬러 변경 */
.click {
  background-color: #6381a1;
  color: white;
}
```



기능 설명 관리자 페이지

필터 검색

키워드 검색 시

회원정보 관리

회원 검색

검색어 테스트

기간검색 연도. 월. 일. ~ 연도. 월. 일.

상세 정보

회원명	아이디	이메일	전화번호	주소	가입일
테스트	a	aaaa@aaaa	010-8904-4778	서울 마포구 월드컵북로4길 77	2024-6-21
테스트	b	sinsang@naver.com	010-4842-3750	서울특별시 마포구 월드컵북로 4길 77	2024-6-21
테스트	abc	skqlgogo12@naver.com	010-9290-2752	부산 남구 무민사로 5	2024-7-2
탈퇴 테스트	dlenf0924	aa@aa	010-2235-0970	경기 화성시 동탄반역로 16	2024-7-3

Mapper의 MyBatis 조건문을 통한 필터검색 구현

```

<!-- 회원정보 관리 - 회원정보 관리 - 조회 기능 (기능 : 키워드, 날짜로 필터검색 가능) -->
<select id="searchAllUser" resultType="adminUserVO">
  SELECT NAME, ID, EMAIL, PHONE_NUM, ADDRESS, CREATE_DATE
  FROM HH_USER
  WHERE 1=1
  <if test="category != null and keyword != null and !category.isEmpty() and !keyword.isEmpty()">
    <choose>
      <when test="category == 'memberName'">
        AND NAME LIKE '%' || #{keyword} || '%'
      </when>
      <when test="category == 'userID'">
        AND ID LIKE '%' || #{keyword} || '%'
      </when>
      <when test="category == 'email'">
        AND EMAIL LIKE '%' || #{keyword} || '%'
      </when>
    </choose>
  </if>
  <if test="startDate != null and !startDate.isEmpty()">
    AND CREATE_DATE >= TO_DATE(#{startDate}, 'YYYY-MM-DD')
  </if>
  <if test="endDate != null and !endDate.isEmpty()">
    AND CREATE_DATE <= TO_DATE(#{endDate}, 'YYYY-MM-DD')
  </if>
  ORDER BY CREATE_DATE ASC
</select>

```

필터검색을 위한 HTML 기본구조

```

<p class="searchTitle">회원 검색</p>
<div id="searchContainer">
  <div id="midContainer">
    <p class="searchContextUser">검색어</p>
    <select id="searchCategory">
      <option value="memberName">회원명</option>
      <option value="userID">아이디</option>
      <option value="email">이메일</option>
    </select>
    <input type="text" id="searchInput" placeholder="입력란 (제목 혹은 내용 입력)">
    <button type="submit" class="commonSmallBtn" id="searchButton">검색</button>
    <button type="submit" class="commonSmallBtn" id="resetButton">초기화</button>
  </div>
  <div id="htmContainer">
    <p class="searchPeriodContext">기간검색</p>
    <select id="dateCategory">
      <option value="joinDate">가입일</option>
    </select>
    <input type="date" id="startDate" placeholder="입력란 (시작일)">
    <p> ~ </p>
    <input type="date" id="endDate" placeholder="입력란 (종료일)">
    <div class="radioContainer">
      <input type="radio" id="today" name="dateSelect">
      <label for="today" class="radioLabel">오늘</label>
    </div>
    <div class="radioContainer">
      <input type="radio" id="week" name="dateSelect">
      <label for="week" class="radioLabel">일주일</label>
    </div>
    <div class="radioContainer">
      <input type="radio" id="lastMonth" name="dateSelect">
      <label for="lastMonth" class="radioLabel">지난달</label>
    </div>
    <div class="radioContainer">
      <input type="radio" id="oneMonth" name="dateSelect">
      <label for="oneMonth" class="radioLabel">1개월</label>
    </div>
    <div class="radioContainer">
      <input type="radio" id="threeMonth" name="dateSelect">
      <label for="threeMonth" class="radioLabel">3개월</label>
    </div>
  </div>
</div>

```



기능 설명 관리자 페이지

필터 검색

기간 검색 시

회원정보 관리

회원 검색

검색어

기간검색 2024. 07. 01. ~ 2024. 07. 08.

상세 정보

회원명	아이디	이메일	전화번호	주소	가입일
sinsang22	민현상	sinsang2752@gmail.com	010-9290-2752	경기 고양시 덕양구 동산동 368-2	2024-7-2
테스트	abc	skqlgogo12@naver.com	010-9290-2752	부산 남구 우민사로 5	2024-7-2
탈퇴 테스트	dlenf0924	aa@aa	010-2235-0970	경기 화성시 동탄반역로 16	2024-7-3
민현상	navertest	sinsang2752@naver.com	010-9290-2752	경기 성남시 분당구 불정로 6	2024-7-4

Mapper의 MyBatis 조건문을 통한 필터검색 구현

```

<!-- 회원정보 관리 - 회원정보 관리 - 조회 기능 (기능 : 키워드, 날짜로 필터검색 가능) -->
<select id="searchAllUser" resultType="adminUserVO">
  SELECT NAME, ID, EMAIL, PHONE_NUM, ADDRESS, CREATE_DATE
  FROM HH_USER
  WHERE 1=1
  <if test="category != null and keyword != null and !category.isEmpty() and !keyword.isEmpty()">
    <choose>
      <when test="category == 'memberName'">
        AND NAME LIKE '%' || #{keyword} || '%'
      </when>
      <when test="category == 'userID'">
        AND ID LIKE '%' || #{keyword} || '%'
      </when>
      <when test="category == 'email'">
        AND EMAIL LIKE '%' || #{keyword} || '%'
      </when>
    </choose>
  </if>
  <if test="startDate != null and !startDate.isEmpty()">
    AND CREATE_DATE >= TO_DATE(#{startDate}, 'YYYY-MM-DD')
  </if>
  <if test="endDate != null and !endDate.isEmpty()">
    AND CREATE_DATE <= TO_DATE(#{endDate}, 'YYYY-MM-DD')
  </if>
  ORDER BY CREATE_DATE ASC
</select>

```

Jquery를 활용하여 Radio버튼 클릭을 통한 기간 프리셋 기능 구현

```

// 라디오 버튼 클릭 시 날짜 필터 설정
$('#input[name="dateSelect"]').click(function() {
  var today = new Date();
  var startDate, endDate;

  // label.radioLabel 인라인 스타일 제거
  $('.radioContainer label').each(function() {
    $(this).css('color', ''); // 인라인 스타일 제거
  });

  if (this.id === 'today') {
    startDate = endDate = today.toISOString().split('T')[0];
  } else if (this.id === 'week') {
    startDate = new Date(today.setDate(today.getDate() - 7)).toISOString().split('T')[0];
    endDate = new Date().toISOString().split('T')[0];
  } else if (this.id === 'lastMonth') {
    var firstDayLastMonth = new Date(today.getFullYear(), today.getMonth() - 1, 1);
    var lastDayLastMonth = new Date(today.getFullYear(), today.getMonth(), 1);
    startDate = firstDayLastMonth.toISOString().split('T')[0];
    endDate = lastDayLastMonth.toISOString().split('T')[0];
  } else if (this.id === 'oneMonth') {
    startDate = new Date(today.setMonth(today.getMonth() - 1)).toISOString().split('T')[0];
    endDate = new Date().toISOString().split('T')[0];
  } else if (this.id === 'threeMonth') {
    startDate = new Date(today.setMonth(today.getMonth() - 3)).toISOString().split('T')[0];
    endDate = new Date().toISOString().split('T')[0];
  }

  $('#startDate').val(startDate);
  $('#endDate').val(endDate);
  loadTable();

  .radioContainer input[type="radio"] {
    display: none;
  }

  .radioContainer label {
    display: block;
    cursor: pointer;
  }

  .radioContainer input[type="radio"]:checked + label {
    color: #210, 223, 81;
  }
});

```

Radio버튼 클릭 시 선택된 기간의 글자색 변경



기능 설명 관리자 페이지

테이블 스크롤링

데이터가 많아질 경우 원활한 조회를 위한 테이블 스크롤링 기능 추가

카테고리 관리

카테고리 편집

검색어 번호 입력란 (제목 혹은 내용 입력) 검색 초기화

상세 내용

선택	번호	카테고리	세부 카테고리	작성자ID	작성일
<input type="checkbox"/>	124	가구/인테리어	수공예/수선	admin	2024-6-1
<input type="checkbox"/>	125	가구/인테리어	오피스 인테리어 용품	admin	2024-6-1
<input type="checkbox"/>	126	가구/인테리어	인테리어 소품	admin	2024-6-1
<input type="checkbox"/>	127	가구/인테리어	꽃/원예	admin	2024-6-1
<input type="checkbox"/>	128	가구/인테리어	조명	admin	2024-6-1
<input type="checkbox"/>	129	가구/인테리어	카페트/러그/매트	admin	2024-6-1
<input type="checkbox"/>	130	가구/인테리어	커튼/블라인드	admin	2024-6-1
<input type="checkbox"/>	131	생활/주방용품	주방용품	admin	2024-6-1
<input type="checkbox"/>	132	생활/주방용품	욕실용품	admin	2024-6-1
<input type="checkbox"/>	133	생활/주방용품	생활용품	admin	2024-6-1
<input type="checkbox"/>	134	공구/산업용품	드릴/전동공구	admin	2024-6-1
<input type="checkbox"/>	135	공구/산업용품	수공구/가정용 공구	admin	2024-6-1
<input type="checkbox"/>	136	공구/산업용품	공구함	admin	2024-6-1
<input type="checkbox"/>	137	공구/산업용품	산업용품/기계	admin	2024-6-1
<input type="checkbox"/>	138	공구/산업용품	측정/계측/레벨	admin	2024-6-1
<input type="checkbox"/>	139	공구/산업용품	공정기계/용접/가스	admin	2024-6-1
<input type="checkbox"/>	140	공구/산업용품	에어/유압공구	admin	2024-6-1
<input type="checkbox"/>	141	공구/산업용품	기타 산업용품	admin	2024-6-1
<input type="checkbox"/>	142	식품	건강식품	admin	2024-6-1
<input type="checkbox"/>	143	식품	건강기능식품	admin	2024-6-1

테이블 레이아웃 고정 후 최대 높이로 영역 부여 및 overflow 설정

```

table {
  table-layout: fixed;
  width: 100%;
  border-collapse: collapse;
}

table tbody {
  display: block;
  max-height: 690px;
  overflow-y: scroll;
  width: 100%;
  border-bottom: 1px solid rgba(0, 0, 0, 0.05);
}

```




기능 설명 관리자 페이지

JavaScript를 통하여 등록된 답변 체크하여 텍스트 출력 구현

```
$(document).ready(function() {
  // 페이지 로드 시 테이블을 초기화
  loadTable();

  // 검색 버튼 클릭 시
  $('#searchButton').click(function() {
    loadTable();
  });

  // 초기화 버튼 클릭 시
  $('#resetButton').click(function() {
    $('#searchInput').val('');
    loadTable();
  });

  // 테이블 데이터를 Ajax로 가져오기
  function loadTable() {
    var category = $('#searchCategory').val();
    var keyword = $('#searchInput').val();

    $.ajax({
      url: '/heehee/admin/searchQnaAll',
      method: 'GET',
      data: { 'category': category,
              'keyword': keyword
            },
      success: function(data) {
        var tableBody = $('#tableBody');
        tableBody.empty();

        data.forEach(function(item) {
          var ansStatus = (!item.qnaAns || item.qnaAns.trim() === '' ) ? '처리대기' : '답변완료';
          var qnaTimePas = new Date(item.qnaTime).toLocaleDateString();
          var qnaTimeRev = qnaTimePas.substring(0, qnaTimePas.length -1).replaceAll(".", "-");

          var row =
            "<tr>" +
            "<td><input type='checkbox' class='rowCheckbox' data-id='" + item.seqQnaBno + "'></td>" +
            "<td>" + item.seqQnaBno + "</td>" +
            "<td>" + ansStatus + "</td>" +
            "<td>" + item.qnaOption + "</td>" +
            "<td>" + item.qnaTitle + "</td>" +
            "<td>" + item.id + "</td>" +
            "<td>" + qnaTimeRev + "</td>" +
            "</tr>";
          tableBody.append(row);
        });
      },
      error: function(xhr, status, error) {
        alert('데이터를 가져오는 중 오류가 발생했습니다. ');
      }
    });
  }
});
```

문의사항 답변여부 확인

```
//최근 회원가입 테이블 데이터를 Ajax로 가져오기
function loadRecentQuestionTable() {
  $.ajax({
    url: "/heehee/admin/searchRecentQuestion",
    method: 'GET',
    success: function(data) {
      var tableBody = $('#recentQuestionTableBody');
      tableBody.empty();

      data.forEach(function(item) {
        var ansStatus = (!item.qnaAns || item.qnaAns.trim() === '' ) ? '처리대기' : '답변완료';
        var qnaDatePas = new Date(item.qnaTime).toLocaleDateString();
        var qnaDateRev = qnaDatePas.substring(0, qnaDatePas.length -1).replaceAll(".", "-");

        var row =
          "<tr>" +
          "<td>" + item.id + "</td>" +
          "<td>" + ansStatus + "</td>" +
          "<td>" + item.qnaTitle + "</td>" +
          "<td>" + qnaDateRev + "</td>" +
          "</tr>";
        tableBody.append(row);
      });
    },
    error: function(xhr, status, error) {
      console.log('AJAX request failed:', status, error);
      alert('데이터를 가져오는 중 오류가 발생했습니다. ');
    }
  });
};
```



기능 설명 관리자 페이지

회원 이용정지

부적절한 사용자에게 대한 이용 정지처리 가능 (Ajax 활용하여 기능 구현)

이용상태 관리

회원 검색

검색어 회원명 입력란 (제목 혹은 내용 입력) 검색 초기화

상세 정보


선택	회원명	아이디	정지내용	시작일	종료일	정지 등록	삭제
<input type="checkbox"/>	신현상	sinsang27521	정지기간 재연장 테스트	2024-6-30	2024-7-13		
사용자 이용정지		chlwoaud	사이트 이용규칙 위반	2024. 07. 08.	2024. 07. 12.	정지 등록	



상세 정보

선택	회원명	아이디	정지내용	시작일	종료일	정지 등록	삭제
<input type="checkbox"/>	최재명	chlwoaud	사이트 이용규칙 위반	2024-7-8	2024-7-12		
<input type="checkbox"/>	신현상	sinsang27521	정지기간 재연장 테스트	2024-6-30	2024-7-13		

정지된 아이디로 로그인 시도 시



서비스 이용제한 안내

안녕하세요. 회희낙할 입니다.

아래와 같은 이유로 회희낙할을 이용할 수 없게 되었습니다.

제재이유 : 사이트 이용규칙 위반
제재기간 : 24/07/08 ~ 24/07/12

[확인](#)

JavaScript Ajax 코드

```

//정지 등록 버튼 클릭 시 (정지 신규추가)
$('#addButton').click(function(){
    var id = 'new';

    //기존의 수정 행 제거
    if ($('#editRow').length > 0){
        $('#editRow').remove();
    }

    if ($('#tableBody').children('.newRow').length > 0){
        $('#newRow').remove();
    } else {
        var newRow =
            "<tr class='newRow'>" +
            "<td colspan='6'>" +
            "<div class='updateContainer'>" +
            "<p class='editTitle'>사용자<br>이용정지</p>" +
            "<input type='text' id='newBanId' class='userBanIdInput' placeholder='정지할 사용자 ID 입력'>" +
            "<input type='text' id='newBanContent' class='userBanContentInput' placeholder='이용정지 사유를 입력해주세요'>" +
            "<input type='date' id='startDateSelect' class='userBanStr2'>" +
            "<input type='date' id='endDateSelect' class='userBanEnd2'>" +
            "<button id='saveNewButton' class='saveNewButtonBan' data-id='"+ id + "'>정지 등록</button>" +
            "</div>" +
            "</td>" +
            "</tr>";
        $('#tableBody').append(newRow);
    }
});

// 저장 버튼 클릭 시 (신규등록)
$(document).on('click', '#saveNewButton', function(){
    var id = $('#newBanId').val();
    var banContent = $('#newBanContent').val();
    var banStr = $('#startDateSelect').val();
    var banEnd = $('#endDateSelect').val();

    console.log("신규등록 id : " + id);
    console.log("신규등록 banContent : " + banContent);
    console.log("신규등록 banStr : " + banStr);
    console.log("신규등록 banEnd : " + banEnd);

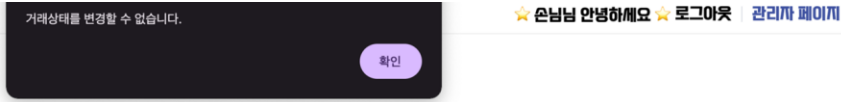
    $.ajax({
        url: '/heehee/admin/insertBanUser',
        method: 'POST',
        data: { 'id': id,
            'banContent': banContent,
            'banStr': banStr,
            'banEnd': banEnd
        },
        success: function(){
            loadTable();
        },
        error: function(){
            alert('정지상태 신규등록 중 오류가 발생했습니다. ');
        }
    });
});

```



기능 설명 관리자 페이지

일반상품의 상태 변경에 대한 제약조건 구현 (예약중, 거래완료 상품 상태변경 불가)



일반상품 상세조회

상품 검색

검색어 등록번호 입력란 (제목 혹은 내용 입력)

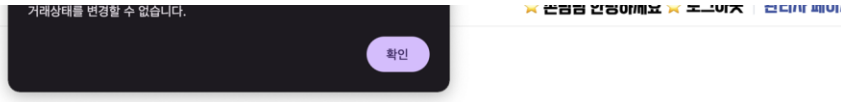
상세 내용

선택	등록번호	카테고리	세부 카테고리	판매자ID	제목	게시일	상태
<input checked="" type="checkbox"/>	6	키보드	피규어/인형	null	보노보노 인스탁스 미니9	2024-6-17	예약중

판매중지 사유
null

판매상태 수정

경매상품의 상태 변경에 대한 제약조건 구현 (낙찰, 유찰, 거래완료 상품 상태변경 불가)



경매상품 상세조회

상품 검색

검색어 등록번호 입력란 (제목 혹은 내용 입력)

상세 내용

선택	등록번호	카테고리	세부 카테고리	판매자ID	제목	게시일	상태
<input type="checkbox"/>	3	여성의류	상의	a	상품부린	2024-6-23	유찰
<input type="checkbox"/>	4	여성의류	아우터	a	상품부린6	2024-6-18	판매중지
<input type="checkbox"/>	5	여성의류	상의	a	상품부린	2024-6-18	유찰
<input type="checkbox"/>	6	여성의류	상의	a	상품부린	2024-6-18	유찰
<input type="checkbox"/>	21	여성의류	아우터	a	상품부린	2024-6-19	거래완료
<input checked="" type="checkbox"/>	41	여성의류	상의	a	상품부린4	2024-6-21	낙찰

판매중지 사유
null

판매상태 수정

관리자 권한 제약조건 부여

상태에 대한 수정 불가 제약조건을 JavaScript로 구현

```
// 저장 버튼 클릭 시
$(document).on('click', '#saveEditButton', function() {
    var productSeq = $(this).attr('data-id');
    var proStatus = $("#editStatus" + productSeq).val();
    var productBanReason = $("#editInput" + productSeq).val();
    var currProStatus = getSelectedRow().closest('tr').find('td').eq(7).text();
    var userid = getSelectedRow().closest('tr').find('td').eq(4).text();

    if(currProStatus == '예약중' || currProStatus == '거래완료' || currProStatus == '판매보류'){
        alert('거래상태를 변경할 수 없습니다. ');
    } else {
        $.ajax({
            url: '/heehee/admin/updateProductStatus',
            method: 'POST',
            data: { 'productSeq': productSeq,
                'proStatus': proStatus,
                'productBanReason': productBanReason,
                'id': userid
            },
            success: function() {
                loadTable();
            },
            error: function() {
                alert('등록 중 오류가 발생했습니다. ');
            }
        });
    }
});
```

```
// 저장 버튼 클릭 시
$(document).on('click', '#saveEditButton', function() {
    var productSeq = $(this).attr('data-id');
    var aucStatus = $("#editStatus" + productSeq).val();
    var aucBanReason = $("#editInput" + productSeq).val();
    var currAucStatus = getSelectedRow().closest('tr').find('td').eq(7).text();
    var userid = getSelectedRow().closest('tr').find('td').eq(4).text();

    if (currAucStatus == '낙찰' || currAucStatus == '유찰' || currAucStatus == '거래완료'){
        alert('거래상태를 변경할 수 없습니다. ');
    } else {
        $.ajax({
            url: '/heehee/admin/updateAucStatus',
            method: 'POST',
            data: { 'productSeq': productSeq,
                'aucStatus': aucStatus,
                'aucBanReason': aucBanReason,
                'id': userid
            },
            success: function() {
                loadTable();
            },
            error: function() {
                alert('등록 중 오류가 발생했습니다. ');
            }
        });
    }
});
```



기능 설명 관리자 페이지

부적절한 일반/경매 상품에 대한 판매 정지처리 가능 (Ajax 활용하여 기능 구현)

일반상품 상세조회

상품 검색

검색어

상세 내용

선택	등록번호	카테고리	세부 카테고리	판매자ID	제목	게시일	상태
<input checked="" type="checkbox"/>	181	키덜트	피규어/인형	aaa12	핀과 케이크의 어드벤처타임 인형들입니다	2024-7-8	판매중

판매중지 사유
null

판매중지 사유 수평



일반상품 상세조회

상품 검색

검색어

상세 내용

선택	등록번호	카테고리	세부 카테고리	판매자ID	제목	게시일	상태
<input checked="" type="checkbox"/>	181	키덜트	피규어/인형	aaa12	핀과 케이크의 어드벤처타임 인형들입니다	2024-7-8	판매중지

판매중지 사유
위급 금지품목으로 판단되어 판매 중지합니다

판매중지 사유 수평

상품 판매중지



기능 설명 관리자 페이지

상품 판매중지

JavaScript Ajax 코드 - 판매상태 수정 진행

```

// 수정 버튼 클릭 시
$('#editButton').click(function() {
    var selected = getSelectedRow();

    if (selected.length === 1) {
        var row = selected.closest('tr');
        var id = selected.data('id');

        // 수정할 내용 입력란을 추가
        if (row.next().hasClass('banReason')) {
            row.next().next().remove();
            row.next().remove();
        } else {
            $.ajax({
                url: '/heehee/admin/getProductBanReason',
                method: 'GET',
                data: { 'productSeq': id, },
                success: function(contentData) {
                    var editRow =
                        "<tr class='editRow'>" +
                        "<td colspan='8'>" +
                        "<div class='updateContainer'>" +
                        "<p class='editTitle'>판매상태<br>수정</p>" +
                        "<select id='editStatus' + id + "' class='selectStatus'>" +
                        "<option value='판매중' " + (row.find('td').eq(7).text() === '판매중' ? 'selected' : '') + ">판매중</option>" +
                        "<option value='판매중지' " + (row.find('td').eq(7).text() === '판매중지' ? 'selected' : '') + ">판매중지</option>" +
                        "</select>" +
                        "<input type='text' id='editInput' + id + "' class='singleInputPro' placeholder='판매중지 사유를 입력해주세요.'>" +
                        "<button id='saveEditButton' class='saveEditButton' data-id='" + id + "'>수정 등록</button>" +
                        "</div>" +
                        "</td>" +
                        "</tr>";

                    var banReason =
                        "<tr class='banReason'>" +
                        "<td colspan='8'>" +
                        "<div class='productBanReason'>" +
                        "<p class='productUpdate'>판매중지 사유</p>" +
                        "<p>" + contentData.productBanReason + "</p>" +
                        "</div>" +
                        "</td>" +
                        "</tr>";
                    row.after(editRow);
                    row.after(banReason);
                },
                error: function(xhr, status, error){
                    alert('정지사유를 가져오는 중 오류가 발생했습니다. ');
                }
            });
        }
    } else if (selected.length === 0) {
        alert('수정할 항목을 선택해주세요. ');
    } else {
        alert('수정할 항목을 하나만 선택해주세요. ');
    }
});

```

JavaScript Ajax 코드 - 판매상태 수정 후 등록 시 제약조건 설정

```

// 저장 버튼 클릭 시
$(document).on('click', '#saveEditButton', function() {
    var productSeq = $(this).attr('data-id');
    var proStatus = $("#editStatus" + productSeq).val();
    var productBanReason = $("#editInput" + productSeq).val();
    var currProStatus = getSelectedRow().closest('tr').find('td').eq(7).text();
    var userid = getSelectedRow().closest('tr').find('td').eq(4).text();

    if(currProStatus == '예약중' || currProStatus == '거래완료' || currProStatus == '판매보류'){
        alert('거래상태를 변경할 수 없습니다. ');
    } else {
        $.ajax({
            url: '/heehee/admin/updateProductStatus',
            method: 'POST',
            data: { 'productSeq': productSeq,
                'proStatus': proStatus,
                'productBanReason': productBanReason,
                'id': userid
            },
            success: function() {
                loadTable();
            },
            error: function() {
                alert('등록 중 오류가 발생했습니다. ');
            }
        });
    }
});

// 삭제 버튼 클릭 시
$('#deleteButton').click(function() {
    var selected = getSelectedRow();

    if (selected.length > 0) {
        if (confirm('선택된 항목을 삭제하시겠습니까?')) {
            selected.each(function() {
                var productSeq = $(this).attr('data-id');

                $.ajax({
                    url: '/heehee/admin/deleteProduct',
                    method: 'POST',
                    data: { 'productSeq': productSeq
                },
                    success: function() {
                        loadTable();
                    },
                    error: function() {
                        alert('삭제 중 오류가 발생했습니다. ');
                    }
                });
            }
        } else {
            alert('삭제할 항목을 선택해주세요. ');
        }
    }
});

```



기능 설명 알림 (WebSocket)

화면 로드 시 알림 목록은 숨겨놓기

SALE 물품등록 채팅 알림 4

```

$(function () {

    // 알림 숨겨놓기
    $(".alarm_container").hide();
    $("#alarmAll").removeClass("add");
    $("#alarmUnck").removeClass("add");

    $("#alarmDiv").on("click", alarmList); // 클릭하면 알림 보여주거나 숨기기
    $(".alarm_type").on("click", alarmClick); // 알림 종류 클릭 시 css 추가

    $("#alarmAll").on("click", alarmAll); // 전체 알림 보기
    $("#alarmUnck").on("click", alarmUnck); // 미확인 알림 보기

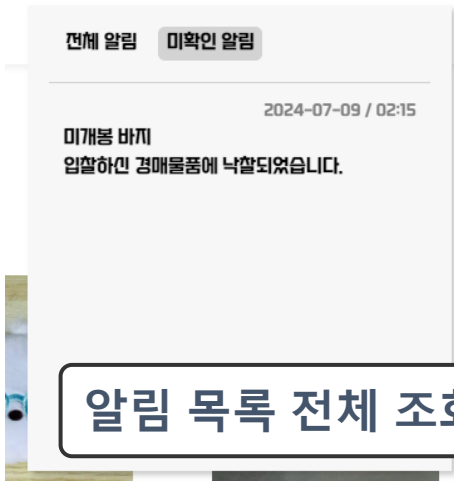
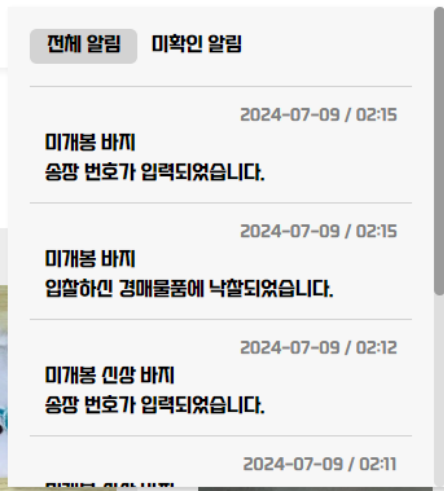
    $("#alarmImg").on("click", alarmCheck); // 알림 아이콘 클릭 시 애니메이션 효과 제거
});

```

알림 아이콘에 on click 이벤트를 부여하여 상태값에 따라 보이게 하거나 숨김

SALE 물품등록 채팅 알림 4

SALE 물품등록 채팅 알림 4



```

// 클릭하면 알림 보여주거나 숨기기
function alarmList() {
    // $(".alarm_container").toggle();

    if ($("#alarm_container").css("display") == "none") {
        $("#alarmUnck").addClass("add"); // default 미확인 알림
        $(".alarm_container").show();

        alarmUnck();
    } else {
        $(".alarm_container").scrollTop(0);
        $("#alarmAll").removeClass("add"); // default 미확인 알림
        $(".alarm_container").hide();
    }
}

```

알림 목록 전체 조회, 미확인 조회로 구분



기능 설명 알림 (WebSocket)

알림 종류 별 버튼을 눌렀을 때 선택된 버튼
에
css 효과를 부여하며, 이벤트 버블링 막기



```

// 알림 종류 클릭 시 css 추가
function alarmClick(e) {
  if (e.target.id == "alarmAll") {
    $("#alarmAll").addClass("add");
    $("#alarmUnck").removeClass("add");
    e.stopPropagation(); // 부모 요소는 실행되지 않고 자식 요소만 실행
  } else {
    $("#alarmUnck").addClass("add");
    $("#alarmAll").removeClass("add");
    e.stopPropagation(); // 부모 요소는 실행되지 않고 자식 요소만 실행
  }
}

```



기능 설명 알림 (WebSocket)

output 변수 선언 후 responseData에 담겨있는 리스트를 화면에 그려주기 위해 반복문 실행

컨트롤러

```

// 알림 전체 조회
@ResponseBody
@GetMapping("/alarmAll")
public List<AlarmChatDTO> alarmList(Principal principal) {
    String userId = "";

    // 로그인 유저 아이디 받기
    if (principal != null) userId = principal.getName();

    List<AlarmChatDTO> alarmList = alarmService.alarmList(userId);
    return alarmList;
}

// 미확인 알림 조회
@ResponseBody
@GetMapping("/alarmUnck")
public List<AlarmChatDTO> alarmUnck(Principal principal) {
    String userId = "";

    // 로그인 유저 아이디 받기
    if (principal != null) userId = principal.getName();

    List<AlarmChatDTO> alarmUnck = alarmService.alarmUnck(userId);
    return alarmUnck;
}

```

알림 목록 조회 시 비동기 통신 사용

```

function alarmUnck() {
    $.ajax({
        url : "/heehee/alarm/alarmUnck",
        type : "GET",
        success : function(responseData) {
            console.log(responseData);

            // 알림창 길이 원래대로
            $(".alarm_container").removeClass("none");

            // 알림별로 경로 다르게 걸어줘야 함
            var output = "<div id='allAlarm' class='alarmClick'>";

            // 알림 리스트 있는지 체크하고 html 다르게 찍어주기
            if (responseData.length === 0 || responseData[0].alDate === null) {
                $(".alarm_container").addClass("none");

                output += "<p>" + "☑ 최근 알림이 없습니다" + "</p>";
            } else {
                // 알림 확인 시 상태값 N => Y, null 되므로 모두 확인하면 html 다르게 찍어주기 위해 모든 요소로 확인
                var check = responseData.every(data => data.alCheck === null);
                // console.log(check);

                if (check === true) {
                    $(".alarm_container").addClass("none");

                    output += "<p>" + "☑ 최근 알림이 없습니다" + "</p>";
                } else {
                    // 알림 전체 리스트 반복문
                    $.each(responseData, function(index, item) {

```



기능 설명 알림 (WebSocket)

```

AND A.CATE_NUM = 1 -- 채팅
AND A.AL_REQ_SEQ = CM.MSC_ID)
WHERE ADD_MONTHS(TO_DATE(alDate, 'YYYY/MM/DD / HH24:MI'), 1) <![CDATA[>]]> SYSDATE
ORDER BY alDate DESC NULLS LAST, alCheck NULLS LAST
/select>

```

2024-07-09 / 02:17
swimmming
새로운 메시지가 있습니다.

쿼리문에서 한달 전 기록들은 조회되지 않도록 처리

알림을 클릭했을 때 해당하는 경로로 페이지를 이동하기 위해 on click 이벤트를 부여 후 redirect 하기 위해 각각의 컨트롤러에서 매핑한 url 작성

```

// 알림 전체 리스트 반복문
$.each(responseData, function(index, item) {

if (item.alDate != null && item.cateNum == 1 && item.alCheck == "N") {
// 채팅
// output += "<ul onclick='urlClick(\"/heehee/chatting\")'>";
// output += "<ul onclick='urlClick(\"/heehee/chatting\" + item.reqSeq + \"\")' + \" alNum=\" + item.alNum + \">";
// output += "<ul onclick='urlClick(\"/heehee/chatting\" + item.reqSeq + \"\")' + \" alNum=\" + item.alNum + \" alCheck=\" + item.alCheck + \">";

output += "<ul onclick='urlClick(\"/heehee/chatting\")' + \" alNum=\" + item.alNum + \" alCheck=\" + item.alCheck + \">";
output += "<li class='alarm_date'>\" + item.alDate + \"</li>\",
output += "<li>\" + item.sender + \"</li>\";
output += "<li>\" + item.alContent + \"</li>\";
output += "</ul>\";

} else if (item.alDate != null && item.cateNum == 2 && item.alCheck == "N") {
// 판매
// output += "<ul onclick='urlClick(\"/heehee/saledetail/\" + item.reqSeq + \"\")'>";
// output += "<ul onclick='urlClick(\"/heehee/saledetail\" + item.reqSeq + \"\")' + \" alNum=\" + item.alNum + \">\";

output += "<ul onclick='urlClick(\"/heehee/mypage/saledetail\" + item.reqSeq + \"\")' + \" alNum=\" + item.alNum + \" alCheck=\" + item.alCheck + \">\";
output += "<li class='alarm_date'>\" + item.alDate + \"</li>\",
output += "<li>\" + item.sender + \"</li>\";
output += "<li>\" + item.alContent + \"</li>\";
output += "</ul>\";

```

```

// 알림 별 페이지 이동
function urlClick(url) {
location.href = url;
}

```



기능 설명 알림 (WebSocket)

쿼리문 (카테고리 별 알림 조회)

<!-- 미확인 알림 조회 -->

```

<select id="alarmUnck" parameterType="string" resultType="AlarmUnck"
SELECT *
FROM(SELECT NVL(MAX(AP.AUCTION_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN AUC_PRODUCT AP ON (A.AL_REQ_SEQ = AP.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 8 -- 경매물품 판매 재개
AND A.AL_REQ_SEQ = AP.PRODUCT_SEQ
AND AUC_STATUS = '합일'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(AP.AUCTION_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN AUC_PRODUCT AP ON (A.AL_REQ_SEQ = AP.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 8 -- 경매물품 판매 중지
AND A.AL_REQ_SEQ = AP.PRODUCT_SEQ
AND AUC_STATUS = '판매중지'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(S.ARTICLE_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 7 -- 중고물품 판매 재개
AND A.AL_REQ_SEQ = S.PRODUCT_SEQ
AND S.PRO_STATUS = '판매중'

```

```

UNION ALL
SELECT NVL(MAX(S.ARTICLE_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 7 -- 중고물품 판매 중지
AND A.AL_REQ_SEQ = S.PRODUCT_SEQ
AND S.PRO_STATUS = '판매중지'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(B.BAN_CONTENT), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN BAN_HISTORY B ON (A.ID = B.ID)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 6 -- 계정 정지
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(Q.QNA_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN QNA_BOARD Q ON (A.ID = Q.ID)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 4 -- 문의
AND A.AL_REQ_SEQ = Q.SEQ_QNA_BNO
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(S.ARTICLE_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 5 -- 상품 입력 (중고물품)
AND A.AL_REQ_SEQ = S.PRODUCT_SEQ
AND S.PRO_STATUS = '발송완료'

```

```

UNION ALL
SELECT NVL(MAX(S.ARTICLE_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 2 -- 판매
AND A.AL_REQ_SEQ = S.PRODUCT_SEQ
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(AP.AUCTION_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN AUC_PRODUCT AP ON (A.AL_REQ_SEQ = AP.PRODUCT_SEQ)
JOIN DEAL_HISTORY DH ON (A.ID = DH.BUYER_ID)
JOIN DELIVERY D ON (DH.S_SEQ = D.S_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 5 -- 상품 입력 (경매물품)
AND A.AL_REQ_SEQ = DH.AUC_SEQ
AND D.D_STATUS = '발송완료'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(S.ARTICLE_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
JOIN DEAL_HISTORY DH ON (A.ID = DH.BUYER_ID)
JOIN DELIVERY D ON (DH.S_SEQ = D.S_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 5 -- 상품 입력 (중고물품)
AND A.AL_REQ_SEQ = DH.SELL_SEQ
AND D.D_STATUS = '발송완료'

```

```

UNION ALL
SELECT NVL(MAX(S.ARTICLE_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 5 -- 상품 입력 (중고물품)
AND A.AL_REQ_SEQ = S.PRODUCT_SEQ
AND D.D_STATUS = '발송완료'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(AP.AUCTION_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN AUC_PRODUCT AP ON (A.AL_REQ_SEQ = AP.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 3 -- 경매
AND A.AL_REQ_SEQ = AP.PRODUCT_SEQ
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(CM.SENDER, 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(CM.SEND_TIME, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN CHAT_MESSAGE CM ON (A.ID = CM.RECEIVER)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 1 -- 채팅
AND A.AL_REQ_SEQ = CM.MSG_ID
AND CM.READ_CHECK = 'N'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
GROUP BY CM.SENDER)
GROUP BY SENDER, ALCONTENT, ALDATE, CATENUM, REQSEQ, ALNUM, ALCHECK
ORDER BY alDate DESC NULLS LAST, alCheck NULLS LAST

```

```

MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN SELL_PRODUCT S ON (A.AL_REQ_SEQ = S.PRODUCT_SEQ)
JOIN DEAL_HISTORY DH ON (A.ID = DH.BUYER_ID)
JOIN DELIVERY D ON (DH.S_SEQ = D.S_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 5 -- 상품 입력 (중고물품)
AND A.AL_REQ_SEQ = DH.SELL_SEQ
AND D.D_STATUS = '발송완료'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(MAX(AP.AUCTION_TITLE), 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(A.AL_DATE, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN AUC_PRODUCT AP ON (A.AL_REQ_SEQ = AP.PRODUCT_SEQ)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 3 -- 경매
AND A.AL_REQ_SEQ = AP.PRODUCT_SEQ
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
UNION ALL
SELECT NVL(CM.SENDER, 'From. 회회낙찰') sender,
MAX(A.AL_CONTENT) alContent,
MAX(TO_CHAR(CM.SEND_TIME, 'YYYY-MM-DD / HH24:MI')) alDate,
MAX(NF.CATE_NUM) cateNum,
MAX(A.AL_REQ_SEQ) reqSeq,
MAX(A.AL_NUM) alNum,
MAX(A.AL_CHECK) alCheck
FROM ALARM A JOIN NF_CATEGORY NF ON (A.CATE_NUM = NF.CATE_NUM)
JOIN CHAT_MESSAGE CM ON (A.ID = CM.RECEIVER)
WHERE A.ID = #{userId}
AND A.CATE_NUM = 1 -- 채팅
AND A.AL_REQ_SEQ = CM.MSG_ID
AND CM.READ_CHECK = 'N'
AND AL_CHECK = 'N'
AND A.AL_NUM = (SELECT MAX(IA.AL_NUM) FROM ALARM IA
AND IA.ID = A.ID AND IA.AL_CHECK = 'N')
GROUP BY CM.SENDER)
GROUP BY SENDER, ALCONTENT, ALDATE, CATENUM, REQSEQ, ALNUM, ALCHECK
ORDER BY alDate DESC NULLS LAST, alCheck NULLS LAST

```



기능 설명 알림 (WebSocket)

WebSocketConfig에서 설정한 경로로 연결

```

알림 커넥트 이벤트
Opening Web Socket...
▶ Array(2)
Web Socket Opened...
>>> CONNECT
accept-version:1.1,1.0
heart-beat:10000,10000

<<< CONNECTED
version:1.1
heart-beat:0,0
user-name:swimmming

connected to server undefined

Connected: CONNECTED
user-name:swimmming
heart-beat:0,0
version:1.1

>>> SUBSCRIBE
id:sub-0
destination:/topic/alarm/swimmming

```

화면 로드 시 바로 로그인 여부를 체크하고 웹소켓을 연결하는 함수 호출

```

@Override
public void registerStompEndpoints(StompEndpointRegistry registry) {
    // stomp 접속 url -> /ws-stomp
    registry.addEndpoint("/ws"/"chatws"/"auctionws") //연결될 엔드포인트
        .withSockJS(); //SocketJS를 연결한다는 설정
}

```

홈페이지 어디에서나 알림을 받기 위해 header에서 연결 및 구독

```

<script src="/heehee/resources/js/common.js" </script>
<script>
var socket = new SockJS('/heehee/ws'); // WebSocketConfig 설정에서 sockJS 연결 주소
stompClient = Stomp.over(socket);

$(document).ready(function() {
    // 로그인 여부 확인
    beforeConnectCheck();
}

```



기능 설명 알림 (WebSocket)

알림 생성 및 생성된
알림 개수 같이 보내주기

```
// 알림 웹소켓
@MessageMapping("/alarm/{fromUser}")
public void handleBid(AlarmDTO alarmDto, @DestinationVariable("fromUser") String fromUser) {
    alarmDto.setId(fromUser); // 알림 받을 유저
    alarmService.alarmInsert(alarmDto); // 알림 생성
    int alarmCnt = alarmService.alarmCount(fromUser); // 받은 알림 수

    messagingTemplate.convertAndSend("/topic/alarm/" + fromUser, alarmCnt);
}
```

컨트롤러

홈페이지 어디에서나 알림을 받기 위
해 header에서 연결 및 구독

웹소켓이 연결되고 알림을 띄워줄 때
실행 될 동작 작성

```
<!-- 미확인 알림 개수 -->
<select id="alarmCount" parameterType="String" resultType="int">
    SELECT COUNT(AL_NUM) AS CNT
    FROM ALARM
    WHERE ID = #{userId}
           AND AL_CHECK = 'N'
</select>
/manner>
```

```
+
5 function beforeConnectCheck() {
6     // 로그인 하면 소켓 연결
7     if ("${userId}" != "") alarmConnect();
8 }
9
10 function alarmConnect() {
11     console.log("알람 커넥트 이벤트");
12     stompClient.connect({}, function(frame) {
13         // setConnected(true);
14         console.log('Connected: ' + frame);
15
16         // topic/alarm/{userId} 구독
17         stompClient.subscribe('/topic/alarm/' + "${userId}", function(response) {
18             showResponse(JSON.parse(response.body));
19         });
20         //alarmUnck();
21     });
22 }
23
24 // 소켓 연결 후 실행
25 function showResponse(res) {
26     console.log(res);
27
28     // 알림 올 경우 아이콘 변경, 애니메이션 효과 추가
29     $("#alarmImg").attr("src", "https://sh-heehee-bucket.s3.ap-northeast-2.amazonaws.com/images/header/icon_alarm_O.png");
30     $("#alarmImg").addClass("alarmImg");
31     $("#alarmCnt").text(res);
32     showToast('🔔 새로운 알림이 있습니다 💎');
33 }
34
```



기능 설명 알림 (WebSocket)

알림을 보내야 하는 기능들이 insert 될 때 알림이 같이 insert 되어야 하므로 각각의 서비스에서 처리

```

if(message.getReadCheck()=="N") {
    AlarmDTO alarmDTO = new AlarmDTO();

    String receiverId = message.getReceiver();

    alarmDTO.setId(receiverId);
    alarmDTO.setCateNum(1); // 알림 분류 코드 (채팅)
    alarmDTO.setReqSeq(message.getMsgId());
    alarmDTO.setAlContent("새로운 메시지가 있습니다.");

    alarmDao.alarmInsert(alarmDTO);

    int alarmCnt = alarmDao.alarmCount(receiverId);
    messagingTemplate.convertAndSend("/topic/alarm/" + receiverId, alarmCnt);
}

@Transactional
public int insertDelivery(InsertDeliveryDTO delivery, int productSeq) {
    int result = mypageDao.insertDelivery(delivery);

    if (result == 1) {
        // 알림 insert
        AlarmDTO alarmDTO = new AlarmDTO();
        String buyerId = delivery.getBuyerId();

        alarmDTO.setId(buyerId); // 구매자 아이디
        alarmDTO.setCateNum(5); // 알림 분류 코드 (배송)
        alarmDTO.setReqSeq(productSeq); // 상품 시퀀스
        alarmDTO.setAlContent("배송 번호가 입력되었습니다.");

        alarmDao.alarmInsert(alarmDTO);

        int alarmCnt = alarmDao.alarmCount(buyerId);
        messagingTemplate.convertAndSend("/topic/alarm/" + buyerId, alarmCnt);
    }
    return result;
}

```

```

// 고객 지원 - 1:1 상담문의 - 열람/답변 중 답변 기능 (기능 : 선택된 항목에 대한 답변내용 작성 및 Update 가능)
@Transactional
public int updateQnaAns(int seqQnaBno, String newValue, String id) {
    int result = adminDAO.updateQnaAns(seqQnaBno, newValue);
    if(result == 1) {
        AlarmDTO alarmDTO = new AlarmDTO();

        alarmDTO.setId(id);
        alarmDTO.setCateNum(4); // 알림 분류 코드 (문의)
        alarmDTO.setReqSeq(seqQnaBno);
        alarmDTO.setAlContent("문의 답변이 등록되었습니다.");

        alarmDAO.alarmInsert(alarmDTO);

        int alarmCnt = alarmDAO.alarmCount(id);
        messagingTemplate.convertAndSend("/topic/alarm/" + id, alarmCnt);
    }
    return result;
    //adminDAO.updateQnaAns(seqQnaBno, newValue);
}

// 상품 관리 - 경매상품 상세조회 - 수정 기능 (기능 : 선택된 상품의 판매상태(AUC_STATUS)를 수정)
@Transactional
public int updateAucStatus(int productSeq, String aucStatus, String aucBanReason, String id) {
    int result = adminDAO.updateAucStatus(productSeq, aucStatus, aucBanReason);
    if(result == 1) {
        int alarmCnt = alarmDAO.alarmCount(id);
        messagingTemplate.convertAndSend("/topic/alarm/" + id, alarmCnt);
    }
    return result;
}

```

```

<!-- 알림 생성 -->
<insert id="alarmInsert" parameterType="alarmDTO">
    INSERT INTO ALARM VALUES (ALARM_SEQ.NEXTVAL,
                                #{id},
                                #{cateNum},
                                #{reqSeq},
                                #{alContent},
                                'N',
                                SYSDATE)
</insert>

```



기능 설명 알림 (WebSocket)

알림이 오면 아이콘에 애니메이션 효과 추가 및
확인하지 않은 알림의 개수 추가

c:if 태그로 미확인 알림이 있는지 체크 후 있을 경우 알림 아이콘 변경 및 애니메이션 효과를 추가하여 사용자에게 알려주기

알림이 오면 토스트 메시지를 함께 띄워줌

```

:div>
  <c:if test="\${alarmCount > 0}">
    \${alarmCount}</span>
  </c:if>
  <c:if test="\${alarmCount == null || alarmCount == 0}">
    </span>
  </c:if>
:/div>
:div class="alarm_container">
  <div>
    <div id="alarmAll" class="alarm_type add">전체 알림</div>
    <div id="alarmUnck" class="alarm_type add">미확인 알림</div>
  </div>
  <%-- 알림 찍어주는 위치 --%>
  <div id="here" class="alarm_list"></div>
:/div>

```



기능 설명 알림 (WebSocket)

```

<!-- 알림 확인 -->
<update id="alarmUpdate" parameterType="int">
  UPDATE ALARM
  SET AL_CHECK = 'Y'
  WHERE AL_NUM = #{alNum}
  AND AL_CHECK = 'N'
</update>

```

```

// 알림 확인
function alarmRead() {
  // var li = event.target.parentElement;

  var alNum = $(this).attr("alNum");
  console.log(alNum);

  $.ajax({
    url : "/heehee/alarm/alarmUpdate/" + alNum,
    type : "POST",
    success : function(responseData) {
      if (responseData == "1") {
        console.log("알림 확인");
      } else {
        console.log("이미 확인");
      }
    },
    error : function(data) {
      console.log(data);
      alert("알림 확인 오류 입니다.");
    }
  });
}

```

2

```

}
output += "</div>";
$("#here").html(output);

// 알림 확인
$("#allAlarm ul").on("click", alarmRead);

// 확인한 알림 색상 변경
alarmVisited();
}

```

1

알림을 클릭할 경우 해당 알림의 경로로 이동하며, 이 때 알림 읽음 여부를 업데이트 하고 구분하기 위해 색상 변경

```

// 확인한 알림 색상 변경
function alarmVisited() {
  var beforeCheck = $("#allAlarm").children("ul");

  for (let i = 0; i < beforeCheck.length; i++) {
    // console.log("몇번타? : " + i);

    if (beforeCheck.eq(i).attr("alCheck") == "Y") {
      // console.log(i);
      beforeCheck.eq(i).addClass("visited");
    }
  }
}
}

```

3



기능 설명

실시간 경매 (경매 목록 페이지)

★ 후기단님 안녕하세요 ★ 로그아웃 마이페이지



중고물품 | 경매물품

어떤 상품을 찾으시나요?



경매등록



채팅



알림 51

카테고리

마감 임박 상품

<p>에어팟 맥스</p> <p>23:36:12</p> <p>입찰가 400,000원</p>	<p>뉴진스 해린 티셔츠입니다.</p> <p>24:44:12</p> <p>입찰가 25,500원</p>	<p>백설기</p> <p>25:18:12</p> <p>입찰가 125,000원</p>	<p>DO 포카 경매 합니다!!!</p> <p>25:43:12</p> <p>입찰가 20,500원</p>	<p>메이플스토리 프라이밍 헤리 MX 역속 키보드 (새제품)</p> <p>26:52:12</p> <p>입찰가 52,000원</p>	<p>산리오 포차코 포차코 포토모코 마스코트 인형</p> <p>28:50:12</p> <p>입찰가 10,000원</p>	<p>테스트</p> <p>47:16:12</p> <p>입찰가 1,600원</p>
<p>Sony WH-1000XM4 무선 노이즈 캔슬링 헤드폰</p> <p>48:20:12</p>	<p>통기타</p> <p>86:20:12</p>	<p>아이패드 프로 12.9 6세대</p> <p>94:20:12</p>	<p>에어팟</p> <p>114:00:12</p>	<p>보테가베네타 인트레치아 토 루튼 반지갑(블랙그린)</p> <p>118:20:12</p>		

컨트롤러

```
@GetMapping
public String auction(Model model, Principal principal) {
    List<AuctionProdDTO> aucList =auctionService.aucProdList();
    model.addAttribute("aucList", aucList);
    model.addAttribute("lastURL", "auc");
    if (principal != null) {
        model.addAttribute("userId", principal.getName());
        model.addAttribute("userInfo", myPageService.sellerInfo(principal.getName()));
        int alarmCount = alarmService.alarmCount(principal.getName());
        model.addAttribute("alarmCount", alarmCount);
    }
    List<CategoryDTO> mainCateList = mainService.mainCateList();
    model.addAttribute("mainCateList", mainCateList); // header 카테고리
    return "/main/auction";
}
```

진행중인 경매 리스트 조회

쿼리문

```
<select id="aucProdList" resultType="com.shinhan.heehee.dto.response.auction.AuctionProdDTO">
SELECT *
FROM(
SELECT PRODUCT_SEQ, NVL((SELECT MAX(BID_PRICE) FROM AUCTION_HISTORY HISTORY WHERE HISTORY.AUC_PROD_SEQ = AUC.PRODUCT_SEQ), START_PRICE) AS AUC_PRICE
, AUCTION_TITLE, TO_CHAR(EXP_TIME, 'yyyy/mm/dd') AS EXP_DATE
, TO_CHAR(EXP_TIME, 'hh24:mi:ss') AS EXP_TIME, IMG.IMG_NAME
FROM AUC_PRODUCT AUC, IMG_FILES IMG
WHERE IMG.TABLE_PK = AUC.PRODUCT_SEQ
AND IMG.IMG_CATE_SEQ = 3
AND AUC.STATUS = '입찰'
AND AUC.EXP_TIME > SYSDATE
AND IMG.IMG_SEQ = (SELECT MIN(IMG_SEQ) FROM IMG_FILES WHERE TABLE_PK = AUC.PRODUCT_SEQ AND IMG_CATE_SEQ = 3)
AND ROWNUM <=[CDATA[<]]> 15)
GROUP BY PRODUCT_SEQ, AUC_PRICE, AUCTION_TITLE, EXP_DATE ,EXP_TIME,IMG_NAME
ORDER BY EXP_DATE, EXP_TIME
</select>
```



기능 설명

실시간 경매 (입찰가 갱신)

마감 시간 카운트 다운

```
function fGet2char(num) {
    return num < 10 ? "0" + num : String(num);
}

function remaindTime(seq, expDate, expTime) {
    var expD = expDate;
    var dSplit = expD.split("/");

    var expT = expTime;
    var tSplit = expT.split(":");

    var now = new Date();
    var exp = new Date(dSplit[0], dSplit[1] - 1, dSplit[2], tSplit[0], tSplit[1], tSplit[2]);

    let gap = Math.max(exp.getTime() - now.getTime(), 0);
    let day = Math.floor((gap / 3600000) / 24);
    let HH = fGet2char(Math.floor((gap / 3600000) % 24 + (day * 24)));
    let MM = fGet2char(Math.floor((gap % 3600000) / 60000));
    let SS = fGet2char(Math.floor((gap % 60000) / 1000));

    if((gap/60000) < 30) {
        $(".c_" + seq).removeClass("blue");
        $(".c_" + seq).addClass("red");
    }

    $(".h_" + seq).html(HH);
    $(".m_" + seq).html(MM);
    $(".s_" + seq).html(SS);
}
}
```

실시간 입찰가 갱신

```
@GetMapping("/prices")
@ResponseBody
public List<AuctionProdDTO> prices(@RequestParam(value = "seqArr[]", required = false) List<Integer> seqArr) {
    if(seqArr == null) throw new AucListZeroException();
    List<AuctionProdDTO> aucPriceList = auctionService.aucPriceList(seqArr);
    return aucPriceList;
}

function getAucPrice() {
    var cards = $(".card");
    var seqArr = [];
    for(let i = 0; i < cards.length; i++) {
        var seq = $(".card").eq(i).children(".seq").val();
        seqArr.push(seq);
    }

    $.ajax({
        url: '/heehee/auc/prices',
        method: 'GET',
        data : {"seqArr" : seqArr},
        dataType : 'json',
        success: function (data, status, xhr) {
            for(let i = 0; i < data.length; i++) {
                $("#pr_" + data[i].productSeq).text(formatNumberWithCommas(data[i].aucPrice) + "원")
            }
        },
        error: function (data, status, err) {
            console.log(err);
        }
    });
}
```

비동기 요청 후 ResponseData를 반영
(Ajax Polling 기술 사용)



기능 설명

실시간 경매 (입찰 페이지)

컨트롤러

```
@GetMapping("/detail/{aucSeq}")
public String detail(@PathVariable("aucSeq") int aucSeq, Model model, Principal principal) {
    AuctionProdInfoDTO aucProdInfo = auctionService.aucProdInfo(aucSeq);
    if (aucProdInfo == null)
        return "redirect:/auc";

    aucProdInfo.setIntroduce(aucProdInfo.getIntroduce().replace(" ", "&nbsp;").replace("\n", "<br>"));

    if (principal != null) {
        model.addAttribute("userId", principal.getName());
        model.addAttribute("userInfo", myPageService.sellerInfo(principal.getName()));
        int alarmCount = alarmService.alarmCount(principal.getName());
        model.addAttribute("alarmCount", alarmCount);
    }

    SellerInfoResponseDTO sellerInfo = auctionService.sellerInfo(aucProdInfo.getSellerId());

    model.addAttribute("lastURL", "auc");
    model.addAttribute("aucProdInfo", aucProdInfo);
    model.addAttribute("aucImgs", auctionService.aucProdImgList(aucSeq));
    model.addAttribute("sellerInfo", sellerInfo);

    List<CategoryDTO> mainCateList = mainService.mainCateList();
    model.addAttribute("mainCateList", mainCateList); // header 카테고리
    return "/auction/detail";
}
```

서비스

```
public AuctionProdInfoDTO aucProdInfo(int aucSeq) {
    return auctionDAO.aucProdInfo(aucSeq);
}

public List<AuctionImgsDTO> aucProdImgList(int aucSeq) {
    return auctionDAO.aucProdImgList(aucSeq);
}

public List<CategoryDTO> mainCateList() {
    return auctionDAO.mainCateList();
}

public SellerInfoResponseDTO sellerInfo(String userId) {
    return auctionDAO.sellerInfo(userId);
}

public SellerInfoResponseDTO sellerInfo(int aucSeq) {
    return auctionDAO.sellerInfo(aucProdInfo(aucSeq).getSellerId());
}
```

```
<select id="aucProdInfo" parameterType="int"
resultType="com.shinhan.heehee.dto.response.auction.AuctionProdInfoDTO">
SELECT A.*, U.NICK_NAME AS CURRENT_NICK FROM
(SELECT * FROM
SELECT PRODUCT_SEQ
, NVL((SELECT MAX(BID_PRICE) FROM AUCTION_HISTORY HISTORY WHERE HISTORY.AUC_PROD_SEQ = AUC.PRODUCT_SEQ), START_PRICE) AS AUC_PRICE
, INCREASE_PRICE
, AUCTION_TITLE
, INTRODUCE
, CONDITION
, TO_CHAR(EXP_TIME, 'yyyy/mm/dd') AS EXP_DATE
, TO_CHAR(EXP_TIME, 'hh24:mi:ss') AS EXP_TIME
, AUC_STATUS
, AUCTION_SEQ
, CATE.CATEGORY
, CATE.DETAIL_CATEGORY
, SELLER_ID
, (SELECT COUNT(DISTINCT(USER_ID)) FROM AUCTION_HISTORY HISTORY WHERE HISTORY.AUC_PROD_SEQ = AUC.PRODUCT_SEQ) AS JOIN_COUNT
FROM AUC_PRODUCT AUC , PRODUCT_CATEGORY CATE , AUCTION_HISTORY HIS
WHERE AUC.AUCTION_SEQ = CATE.PRODUCT_CATE_SEQ
AND HIS.AUC_PROD_SEQ(+) = AUC.PRODUCT_SEQ
AND PRODUCT_SEQ = #{aucSeq}
ORDER BY BID_PRICE DESC)
WHERE ROWNUM = 1) A,
HH_USER U
WHERE A.CURRENT_ID = U.ID(+)
</select>
```

```
<select id="aucProdImgList" parameterType="int" resultType="com.shinhan.heehee.dto.response.auction.AuctionImgsDTO">
SELECT *
FROM IMG_FILES
WHERE TABLE_PK = #{aucSeq}
AND IMG_CATE_SEQ = 3
</select>
```

```
<select id="sellerInfo" parameterType="String" resultType="com.shinhan.heehee.dto.response.auction.SellerInfoResponseDTO">
SELECT ID, NICK_NAME, PROFILE_IMG, USER_INTRODUCE, USER_RATING FROM HH_USER WHERE ID = #{userId}
</select>
```

쿼리문



중고물품 | 경매물품

어떤 상품을 찾으시나요?

경매등록 채팅 알림

네이버 테스트님 안녕하세요 | 로그아웃 | 마이페이지

카테고리



21:06:52

차량/오토바이 > 오토바이/스쿠터

메이플스토리 프라임 체리 MX 짝죽 키보드 (새제품)

53000원

입찰자수: 2명

제품 상태: 신품

입찰 시 증가 금액: 1000원

예상 낙찰자: 네이버 테스트

나의 보유 포인트

47,000 P

포인트 증편

입찰하기

물품 정보

메이플 스토리 한정판 짝죽 키보드
단순 개봉 새제품 주원버섯
* 운할 수가 없고 가능합니다!
체리 MX 짝죽 스위치 탑재

택배 거래: +5000원 (에어캡 안전포장)
+@ 추가 운할 의뢰 가능합니다*

진심키보드 EVENT
계좌이체 = 당일 택배 접수 및 무료 배송

판매자 정보



★★★★★

물리

호빗



기능 설명

실시간 경매 (실시간 입찰)

소켓 연결, 구독

```
function connect() {
  var socket = new SockJS('/heehee/auctionws');
  AucStompClient = Stomp.over(socket);
  AucStompClient.connect({}, function (frame) {
    console.log('Connected: ' + frame);
    AucStompClient.subscribe('/topic/auction/' + aucSeq, function (response) {
      console.log(response);
      aucResponse(JSON.parse(response.body));
    });
  });
}
```

입찰 보내기

```
function sendBid() {
  if("${userId}" == "") {openLogin(); return false;}
  var now = new Date();
  var currentPrice = $("#auc_price").val();
  var bidPrice = parseInt(currentPrice) + ${aucProdInfo.increasePrice};
  let gap = Math.max(exp.getTime() - now.getTime(), 0);
  if(gap == 0) {
    showToast("종료된 경매입니다.");
    return false;
  }
  if(bidPrice > point) {showToast("포인트가 부족합니다. 충전 후 이용 가능합니다."); return false;}
  var userId = "${userId}";
  AucStompClient.send("/app/bid/"+aucSeq, {}, JSON.stringify({aucProdSeq: aucSeq, "userId": userId, "bidPrice": bidPrice, "userNickName": "${userNickName}"}));
}
```

경매 제품 PK값을 줘서
각 제품마다 소켓 구독

입찰 API

```
@MessageMapping("/bid/{aucSeq}")
public void handleBid(AuctionHistoryDTO aucHistory, @DestinationVariable("aucSeq") int aucSeq, Principal principal) {
  // 현재 스레드가 락을 획득할 때까지 대기, 락을 획득한 후에는 다른 스레드가 이 락을 획득할 수 없게함
  ReentrantLock lock = auctionLocks.computeIfAbsent(aucSeq, k -> new ReentrantLock());
  lock.lock();
  try {
    aucHistory = auctionService.insertAucHistory(aucHistory, principal.getName());
    aucHistory = auctionService.joinCount(aucHistory);
    messagingTemplate.convertAndSend("/topic/auction/" + aucSeq, aucHistory);
  } finally {
    lock.unlock();
    auctionLocks.remove(aucSeq, lock);
  }
}
```

반응 시 함수

```
function aucResponse(message) {
  $("#auc_price").val(message.bidPrice);
  if(message.bidPrice != "undefined") $(".price").text(message.bidPrice + "원");
  $("#joinCount").text(message.joinCount);
  $("#current_user_nickname").text(message.userNickName);
  getRemainingPoint();
}
```



기능 설명

낙찰, 유찰 스케줄러

서비스

```
@Scheduled(fixedRate = 30000)
@Transactional
public void updateBiddingStatus() {
    List<SchedulerBidDTO> successArr = auctionDAO.bidSuccessList();
    List<SchedulerBidDTO> failArr = auctionDAO.bidFailList();

    List<Integer> successIntList = new ArrayList<>();
    List<Integer> failIntList = new ArrayList<>();

    AlarmDTO buyerAlarmDto = new AlarmDTO();
    AlarmDTO sellerAlarmDto = new AlarmDTO();

    InsertDealHistoryDTO dealHistoryDto = new InsertDealHistoryDTO();

    // 낙찰 리스트 PROD_SEQ 담기
    for (SchedulerBidDTO bid : successArr) {
        successIntList.add(bid.getProductSeq());
    }

    // 유찰 리스트 PROD_SEQ 담기
    for (SchedulerBidDTO bid : failArr) {
        failIntList.add(bid.getProductSeq());
    }

    if (successIntList.size() > 0) {
        auctionDAO.updateBidSuccess(successIntList);

        for (SchedulerBidDTO bid : successArr) {
            sellerAlarmDto.setSellerId(bid.getSellerId());
            sellerAlarmDto.setCateNum(3);
            sellerAlarmDto.setReqSeq(bid.getProductSeq());
            sellerAlarmDto.setAlContent("등록하신 경매물품이 낙찰되었습니다.");

            alarmDAO.alarmInsert(sellerAlarmDto);

            int sellerAlarmCnt = alarmDAO.alarmCount(bid.getSellerId());
            messagingTemplate.convertAndSend("/topic/alarm/" + bid.getSellerId(), sellerAlarmCnt);

            buyerAlarmDto.setBuyerId(bid.getBuyerId());
            buyerAlarmDto.setCateNum(3);
            buyerAlarmDto.setReqSeq(bid.getProductSeq());
            buyerAlarmDto.setAlContent("입찰하신 경매물품에 낙찰되었습니다.");

            dealHistoryDto.setAucSeq(bid.getProductSeq());
            dealHistoryDto.setBuyerId(bid.getBuyerId());

            auctionDAO.insertDealHistory(dealHistoryDto);

            auctionDAO.updateWinningBid(bid);

            alarmDAO.alarmInsert(buyerAlarmDto);

            int buyerAlarmCnt = alarmDAO.alarmCount(bid.getBuyerId());
            messagingTemplate.convertAndSend("/topic/alarm/" + bid.getBuyerId(), buyerAlarmCnt);
        }
    }
}
```

서비스

```
if (failIntList.size() > 0) {
    auctionDAO.updateBidFail(failIntList);

    for (SchedulerBidDTO bid : failArr) {
        sellerAlarmDto.setSellerId(bid.getSellerId());
        sellerAlarmDto.setCateNum(3);
        sellerAlarmDto.setReqSeq(bid.getProductSeq());
        sellerAlarmDto.setAlContent("등록하신 경매물품이 유찰되었습니다.");

        alarmDAO.alarmInsert(sellerAlarmDto);

        int sellerAlarmCnt = alarmDAO.alarmCount(bid.getSellerId());
        messagingTemplate.convertAndSend("/topic/alarm/" + bid.getSellerId(), sellerAlarmCnt);
    }

    mypageDAO.updateSecondDstatus();
    mypageDAO.updateFirstDstatus();

    logger.info("----- 낙찰, 유찰, 택배 스케줄러 -----");
}
```

쿼리문

```
<select id="bidSuccessList" resultType="com.shinhan.heehee.dto.response.auction.SchedulerBidDTO">
    SELECT AUC.PRODUCT_SEQ
    ,AUC.EXP_TIME
    ,AUC.AUC_STATUS
    ,AUC.SELLER_ID
    ,HIS.USER_ID AS BUYER_ID
    ,HIS.BID_PRICE
    FROM AUC_PRODUCT AUC, AUCTION_HISTORY HIS
    WHERE AUC.PRODUCT_SEQ = HIS.AUC_PROD_SEQ
    AND EXP_TIME <![CDATA[<]]> SYSDATE
    AND AUC_STATUS = '입찰'
    AND HIS.BID_PRICE = (SELECT MAX(INH.BID_PRICE) FROM AUCTION_HISTORY INH WHERE INH.AUC_PROD_SEQ = AUC.PRODUCT_SEQ
    AND PRODUCT_SEQ IN (SELECT AUC_PROD_SEQ FROM AUCTION_HISTORY GROUP BY AUC_PROD_SEQ)
</select>

<select id="bidFailList" resultType="com.shinhan.heehee.dto.response.auction.SchedulerBidDTO">
    SELECT *
    FROM AUC_PRODUCT
    WHERE EXP_TIME <![CDATA[<]]> SYSDATE
    AND AUC_STATUS = '입찰'
    AND PRODUCT_SEQ NOT IN (SELECT AUC_PROD_SEQ FROM AUCTION_HISTORY GROUP BY AUC_PROD_SEQ)
</select>
```



기능 설명

경매 물품 등록

★ 후시딘님 안녕하세요 ★ 로그인 | 마이페이지



중고물품 경매물품

어떤 상품을 찾으시나요?



≡ 카테고리

경매 등록

상품 이미지

사진 추가

카테고리

대분류	여성의류
여성의류	아우터
남성의류	상의
신발	바지
가방/가방	키링
시계	원피스
주얼리	캠프수트

제품명

제품명을 입력해주세요.

상품 상태

인품(미사용) 사용감 없음 사용감 있음 고장/파손

상품 설명

설명을 입력해주세요.

경매 마감일(시간)

연도, 월, 일, -- --:--

경매 시액가

가격을 입력해주세요.

경매 참가 금액

참가금액을 입력해주세요.

등록하기

컨트롤러

```
@GetMapping("/regi")
public String regi(Model model, Principal principal) {
    String userId = principal.getName();
    model.addAttribute("lastURL", "auc");
    model.addAttribute("userId", userId);

    int alarmCount = alarmService.alarmCount(userId);
    model.addAttribute("alarmCount", alarmCount);

    model.addAttribute("categoryList", productmodifyService.category());

    List<CategoryDTO> mainCateList = mainService.mainCateList();
    model.addAttribute("mainCateList", mainCateList); // header 카테고리

    return "/auction/aucProductRegi";
}
```

```
@PostMapping("/regi")
public String aucProdRegist(@RequestParam("uploadImgs") List<MultipartFile> uploadImgs
    ,AuctionInsertDTO aucInsertDto, Principal principal) throws IOException {
    aucInsertDto.setExpTime(aucInsertDto.getExpTime().replace("T", " "));
    aucInsertDto.setUploadFiles(uploadImgs);
    aucInsertDto.setSellerId(principal.getName());

    auctionService.insertAucProd(aucInsertDto, principal.getName());

    return "redirect:/auc/detail/" + aucInsertDto.getProductSeq();
}
```

쿼리문

```
<insert id="insertAucProd" parameterType="com.shinhan.heehee.dto.request.auction.AuctionInsertDTO">
    <selectKey keyProperty="productSeq" order="BEFORE" resultType="int">
        SELECT SEQ_AUC_PROD.NEXTVAL FROM DUAL
    </selectKey>
    INSERT INTO AUC_PRODUCT (PRODUCT_SEQ, START_PRICE, INCREASE_PRICE, AUCTION_TITLE, CONDITION, INTRODUCE, EXP_TIME, AUC_STATUS, AUCTION_SEQ, SELLER_ID, INSERT_DATE)
    VALUES ({productSeq}, {startPrice}, {increasePrice}, {auctionTitle}, {condition}, {introduce}, TO_DATE({expTime}, 'YYYY-MM-DD hh24:mi'), '입찰', {auctionSeq}, {sellerId}, SYSDATE)
</insert>

<insert id="insertImgFile" parameterType="com.shinhan.heehee.dto.request.ImageFileDTO">
    INSERT INTO IMG_FILES VALUES(SEQ_IMG_FILES.nextval, {imgName}, {tablePk},3,{userId})
</insert>
```

모든 필드에 공백 값 체크



기능 설명

스프링 시큐리티, OAuth 2.0

Config

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .csrf().disable()
        .httpBasic().disable()
        .sessionManagement()
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS)
        .and()
        .exceptionHandling()
        .authenticationEntryPoint(customAuthenticationEntryPoint())
        .accessDeniedHandler(customAccessDeniedHandler())
        .and()
        .authorizeRequests()
        .antMatchers("/main").permitAll()
        .antMatchers("/main/**").permitAll()
        .antMatchers("/user/**").permitAll()
        .antMatchers("/auc").permitAll()
        .antMatchers("/auc/detail/**").permitAll()
        .antMatchers("/mypage/**").authenticated()
        .antMatchers("/sell/productregi").authenticated()
        .antMatchers("/sell/productmodify/**").authenticated()
        .antMatchers("/auc/regi").authenticated()
        .antMatchers("/admin").hasAnyRole("ADMIN")
        .antMatchers("/admin/**").hasAnyRole("ADMIN")
        .and()
        .logout()
        .logoutUri("/user/logout")
        .invalidateHttpSession(true)
        .deleteCookies("JSESSIONID")
        .logoutSuccessHandler(jwtLogoutSuccessHandler())
        .and()
        .oauth2Login()
        .authorizationEndpoint()
        .baseUri("/oauth2/authorize")
        .and()
        .redirectEndpoint()
        .baseUri("/login/oauth2/code/**")
        .and()
        .successHandler(oAuth2AuthenticationSuccessHandler())
        .failureHandler(oAuth2AuthenticationFailureHandler())
        .clientRegistrationRepository(clientRegistrationRepository())
        .and()
        .addFilterBefore(jwtFilter(), UsernamePasswordAuthenticationFilter.class);
}

```

JWT 사용으로 Session 비활성화

```

public class CustomAuthenticationEntryPoint implements AuthenticationEntryPoint {
    @Override
    public void commence(HttpServletRequest request, HttpServletResponse response,
        AuthenticationException authException) throws IOException, ServletException {
        // JSON 응답을 보낼 때
        response.setContentType("application/json");
        response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);
        response.getWriter().write("{\"error\": \"Unauthorized access\"}");

        // 또는 특정 페이지로 리디렉션할 때
        response.sendRedirect("/heehee/main");
    }
}

public class CustomAccessDeniedHandler implements AccessDeniedHandler {
    @Override
    public void handle(HttpServletRequest request, HttpServletResponse response,
        AccessDeniedException accessDeniedException) throws IOException, ServletException {
        response.sendRedirect("/heehee/main");
    }
}

@Component
public class JwtLogoutSuccessHandler implements LogoutSuccessHandler {
    @Override
    public void onLogoutSuccess(HttpServletRequest request, HttpServletResponse response, Authentication authentication)
        throws IOException, ServletException {
        Cookie tokenCookie = new Cookie("Authorization", null);
        tokenCookie.setMaxAge(0);
        tokenCookie.setPath("/");
        response.addCookie(tokenCookie);
    }
}

```

OAuth SuccessHandler

```

public class OAuth2AuthenticationSuccessHandler extends SimpleUrlAuthenticationSuccessHandler {
    @Autowired
    UserService userService;

    @Autowired
    private AuthenticationManager authenticationManager;

    @Autowired
    AuthenticationSuccess success;

    @Autowired
    AuthenticationFailure failure;

    @Autowired
    JwtUtil jwtUtil;

    @Override
    public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response,
        Authentication authentication) throws IOException, ServletException {
        // Authentication 객체에서 OAuth2AuthenticationToken 추출
        OAuth2AuthenticationToken oAuth2Token = (OAuth2AuthenticationToken) authentication;

        // DefaultOAuth2User 객체를 통해 사용자 정보 추출
        DefaultOAuth2User oAuth2User = (DefaultOAuth2User) oAuth2Token.getPrincipal();

        // 사용자 상세 정보 출력
        oAuth2User.getAttributes().forEach((key, value) -> {
            logger.info(key + ": " + value);
        });

        // 사용자 이름 출력
        String username = oAuth2User.getUserName();
        logger.info("Username: " + username);

        // clientName 추출
        String clientName = oAuth2Token.getAuthorizedClientRegistrationId();
        logger.info("clientName: " + clientName);

        switch (clientName) {
            case "google":
                handleGoogleLogin(oAuth2User, request, response);
                break;
            case "naver":
                handleNaverLogin(oAuth2User, request, response);
                break;
            case "kakao":
                handleKakaoLogin(oAuth2User, request, response);
                break;
        }
    }

    private void handleGoogleLogin(DefaultOAuth2User oAuth2User, HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
        String name = oAuth2User.getUserName();
        String email = oAuth2User.getAttribute("email");

        UserDTO userDto = userService.findByEmail(email);

        if(userDto != null) {
            authenticationHandle(request, response, userDto.getUserName(), userDto.getPassword());
        } else {
            String encodedName = URLEncoder.encode(name, "UTF-8");
            String encodedEmail = URLEncoder.encode(email, "UTF-8");
            response.sendRedirect("/heehee/main?status=signup&name="+ encodedName + "&email="+ encodedEmail);
        }

        System.out.println("Logged in with Google");
    }

    private void handleNaverLogin(DefaultOAuth2User oAuth2User, HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
        Map<String, Object> naverRes = oAuth2User.getAttribute("response");
        String email = (String) naverRes.get("email");
        String name = (String) naverRes.get("name");

        UserDTO userDto = userService.findByEmail(email);

        if(userDto != null) {
            authenticationHandle(request, response, userDto.getUserName(), userDto.getPassword());
        } else {
            String encodedName = URLEncoder.encode(name, "UTF-8");
            String encodedEmail = URLEncoder.encode(email, "UTF-8");
            response.sendRedirect("/heehee/main?status=signup&name="+ encodedName + "&email="+ encodedEmail);
        }

        System.out.println("Logged in with Naver");
    }

    private void handleKakaoLogin(DefaultOAuth2User oAuth2User, HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
        Map<String, Object> kakao_account = oAuth2User.getAttribute("kakao_account");
        Map<String, Object> profile = (Map<String, Object>) kakao_account.get("profile");

        String email = (String) kakao_account.get("email");
        String nickName = (String) profile.get("nickname");

        UserDTO userDto = userService.findByEmail(email);

        if(userDto != null) {
            authenticationHandle(request, response, userDto.getUserName(), userDto.getPassword());
        } else {
            String encodedNickName = URLEncoder.encode(nickName, "UTF-8");
            String encodedEmail = URLEncoder.encode(email, "UTF-8");
            response.sendRedirect("/heehee/main?status=signup&nickName="+ encodedNickName + "&email="+ encodedEmail);
        }

        System.out.println("Logged in with Kakao");
    }
}

```



기능 설명

스프링 시큐리티, OAuth 2.0, JWT

ClientRegistrationRepository

```

@Configuration
public class CustomClientRegistrationRepository implements ClientRegistrationRepository {

    private final Map<String, ClientRegistration> registrations = new HashMap<>();

    @Override
    public ClientRegistration findByRegistrationId(String registrationId) {
        return registrations.get(registrationId);
    }

    public CustomClientRegistrationRepository() {
        // 여기에 클라이언트 등록 정보를 추가합니다.
        registrations.put("google", googleClientRegistration());
        registrations.put("kakao", kakaoClientRegistration());
        registrations.put("naver", naverClientRegistration());
    }

    private ClientRegistration googleClientRegistration() {
        return ClientRegistration.withRegistrationId("google")
            .clientId("965074068088-1a323p3et6ropa92pdccrvs41u7opq1m.apps.googleusercontent.com")
            .clientSecret("GOCSXP-FRxsSmTVEgkjQJiCXze_6OmM3ftaH")
            .scope("openid", "profile", "email")
            .authorizationUri("https://accounts.google.com/o/oauth2/auth")
            .tokenUri("https://oauth2.googleapis.com/token")
            .userInfoUri("https://www.googleapis.com/oauth2/v3/userinfo")
            .jwkSetUri("https://www.googleapis.com/oauth2/v3/certs")
            .userNameAttributeName("sub")
            .clientName("HeeHee")
            .redirectUriTemplate("{baseUrl}/login/oauth2/code/{registrationId}")
            .authorizationGrantType(AuthorizationGrantType.AUTHORIZATION_CODE)
            .build();
    }

    private ClientRegistration kakaoClientRegistration() {
        return ClientRegistration.withRegistrationId("kakao")
            .clientId("856ecd87a235038c7924d9f15ca3512d")
            .clientSecret("ee7w410gopzbryeHsgeOOjNjGVgG1EH")
            .clientAuthenticationMethod(ClientAuthenticationMethod.POST)
            .authorizationGrantType(AuthorizationGrantType.AUTHORIZATION_CODE)
            .scope("profile_nickname", "account_email")
            .authorizationUri("https://kauth.kakao.com/oauth/authorize")
            .tokenUri("https://kauth.kakao.com/oauth/token")
            .userInfoUri("https://kapi.kakao.com/v2/user/me")
            .userNameAttributeName("id")
            .clientName("Kakao")
            .redirectUriTemplate("{baseUrl}/login/oauth2/code/{registrationId}")
            .build();
    }
}

```

JwtFilter

```

@AllArgsConstructor
@NoArgsConstructor
public class JwtFilter extends OncePerRequestFilter {

    JwtUtil jwtUtil;
    CustomUserDetailsService customUserDetailsService;

    @Override
    protected void doFilterInternal(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, FilterChain filterChain)
        throws ServletException, IOException {
        String authorizationHeader = null;
        final Cookie[] cookies = httpServletRequest.getCookies();

        if (cookies != null && authorizationHeader == null) {
            for (Cookie cookie : cookies) {
                if (cookie.getName().equals("Authorization")) {
                    authorizationHeader = cookie.getValue().replaceAll("%20", " ");
                }
            }
        }
        String token = null;
        String userName = null;
        if (authorizationHeader != null && authorizationHeader.startsWith("Bearer ")) {
            token = authorizationHeader.substring(7);
            // 토큰 만료 시 로직
            try {
                userName = jwtUtil.extractUsername(token);
            } catch (ExpiredJwtException ex) {
                Cookie tokenCookie = new Cookie("Authorization", null);

                tokenCookie.setMaxAge(0);
                tokenCookie.setPath("/");
                httpServletResponse.addCookie(tokenCookie);

                ex.printStackTrace();
            }
        }
        if (userName != null && SecurityContextHolder.getContext().getAuthentication() == null) {
            UserDetails userDetails = customUserDetailsService.loadUserByUsername(userName);

            if (jwtUtil.validateToken(token, userDetails)) {
                UsernamePasswordAuthenticationToken usernamePasswordAuthenticationToken = new UsernamePasswordAuthenticationToken(
                    userDetails, null, userDetails.getAuthorities());
                usernamePasswordAuthenticationToken
                    .setDetails(new WebAuthenticationDetailsSource().buildDetails(httpServletRequest));
                SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken);
                httpServletRequest.setAttribute("UserNickName", customUserDetailsService.findNickName(userName).get("NICKNAME"));
            } else {
                Cookie tokenCookie = new Cookie("Authorization", null);

                tokenCookie.setMaxAge(0);
                tokenCookie.setPath("/");
                httpServletResponse.addCookie(tokenCookie);
            }
        }
        filterChain.doFilter(httpServletRequest, httpServletResponse);
    }
}

```



기능 설명

JWT, S3

JwtUtil

```

@Component
public class JwtUtil {

    private String secret = "fasdjsdkjsdafhfsdklhjfsadkljsardkljkbsskfkjsarklsda";

    public String extractUsername(String token) {
        return extractClaim(token, Claims::getSubject);
    }

    public Date extractExpiration(String token) {
        return extractClaim(token, Claims::getExpiration);
    }

    public <T> T extractClaim(String token, Function<Claims, T> claimsResolver) {
        final Claims claims = extractAllClaims(token);
        return claimsResolver.apply(claims);
    }

    private Claims extractAllClaims(String token) {
        return Jwts.parser().setSigningKey(secret).parseClaimsJws(token).getBody();
    }

    private Boolean isTokenExpired(String token) {
        return extractExpiration(token).before(new Date());
    }

    // userdetail에 uuid를 가져온다.
    public String generateToken(String username) {
        Map<String, Object> claims = new HashMap<>();
        return createToken(claims, username);
    }

    private String createToken(Map<String, Object> claims, String subject) {
        return Jwts.builder().setClaims(claims).setSubject(subject).setIssuedAt(new Date(System.currentTimeMillis()))
            .setExpiration(new Date(System.currentTimeMillis() + 1000 * 60 * 60 * 10))
            .signWith(SignatureAlgorithm.HS256, secret).compact();
    }

    public Boolean validateToken(String token, UserDetails userDetails) {
        final String username = extractUsername(token);
        if (isTokenExpired(token)) {
            throw new JwtTokenExpiredException("만료된 토큰 입니다.");
        }
        return (username.equals(userDetails.getUsername()) && !isTokenExpired(token));
    }
}

```

S3Config

```

@Configuration
@EnableWebMvc
public class AWSS3Configuration implements WebMvcConfigurer{

    private String accessKey = "AKIAZAABOJHT2CWSCSEQ";
    private String secretKey = "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";

    @Bean
    public AmazonS3 awsS3Client() {
        BasicAWSCredentials credentials = new BasicAWSCredentials(accessKey, secretKey);
        AmazonS3 s3Client = AmazonS3ClientBuilder
            .standard()
            .withRegion(Regions.AP_NORTHEAST_2)
            .withCredentials(new AWSSStaticCredentialsProvider(credentials))
            .build();
        return s3Client;
    }

    @Bean(name = "multipartResolver")
    public CommonsMultipartResolver multipartResolver() {
        CommonsMultipartResolver resolver = new CommonsMultipartResolver();
        resolver.setMaxUploadSizePerFile(5242880); // 파일당 최대 업로드 크기(5MB)
        return resolver;
    }
}

```



기능 설명

S3 ,ElasticSearch

S3Service

```

@Service
public class AWSS3Service {

    @Autowired
    private AmazonS3 s3Client;

    private String bucketName = "sh-heehee-bucket";

    public ArrayList<String> uploadObject(List<MultipartFile> multipartFiles, String filePath) throws IOException {

        ArrayList<String> saveUrls = new ArrayList<String>();

        for (MultipartFile file : multipartFiles) {
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentLength(file.getSize());
            metadata.setContentType(file.getContentType());
            metadata.setContentDisposition("inline");
            s3Client.putObject(new PutObjectRequest(bucketName, filePath + file.getOriginalFilename(),
                file.getInputStream(), metadata));
            saveUrls.add(s3Client.getUrl(bucketName, filePath + file.getOriginalFilename()).toString());
        }

        return saveUrls;
    }

    public String uploadOneObject(MultipartFile multipartFile, String filepath) throws IOException {

        ObjectMetadata metadata = new ObjectMetadata();
        metadata.setContentLength(multipartFile.getSize());
        metadata.setContentType(multipartFile.getContentType());
        metadata.setContentDisposition("inline");

        s3Client.putObject(new PutObjectRequest(bucketName, filepath + multipartFile.getOriginalFilename(),
            multipartFile.getInputStream(), metadata));
        String[] urlSplit = s3Client.getUrl(bucketName, filepath + multipartFile.getOriginalFilename()).getFile()
            .toString().split("/");
        return urlSplit[urlSplit.length - 1];
    }
}

```

검색 인덱스 생성

```

// 검색 인덱스 생성 메서드
public void createIndexWithNGramAnalyzer() throws IOException {
    CreateIndexRequest request = new CreateIndexRequest("prod_index");

    // 인덱스 설정
    request.settings(Settings.builder()
        .put("index.analysis.analyzer.ngram_analyzer.tokenizer", "standard")
        .putList("index.analysis.analyzer.ngram_analyzer.filter", "lowercase", "ngram_filter")
        .put("index.analysis.filter.ngram_filter.type", "ngram")
        .put("index.analysis.filter.ngram_filter.min_gram", 1) // 최소 글자
        .put("index.analysis.filter.ngram_filter.max_gram", 25) // 최대 글자
        .put("index.max_ngram_diff", 24) // 최대 최소 글자 차이
    );

    // 매핑 설정
    String mapping = "{"
        + "\"properties\": {"
        + "  \"gubun\": {\"type\": \"text\"},"
        + "  \"productSeq\": {\"type\": \"integer\"},"
        + "  \"title\": {"
        + "    \"type\": \"text\",
        + "    \"analyzer\": \"ngram_analyzer\",
        + "    \"search_analyzer\": \"standard\",
        + "    \"fields\": {"
        + "      \"keyword\": {"
        + "        \"type\": \"keyword\"
        + "      }
        + "    }
        + "  },"
        + "  \"idate\": {\"type\": \"text\"},"
        + "  \"imgName\": {\"type\": \"text\"}"
        + "}"
        + "}";
    request.mapping(mapping, XContentType.JSON);

    // 인덱스 생성 요청
    CreateIndexResponse createIndexResponse = client.indices().create(request, RequestOptions.DEFAULT);
    boolean acknowledged = createIndexResponse.isAcknowledged();
    if (acknowledged) {
        System.out.println("인덱스 생성 완료");
    }
}

```



기능 설명

ElasticSearch

인덱스 재생성 및 엘라스틱 서버에 데이터 동기화 메서드

```
// 인덱스 재생성 메소드
public void recreateIndex() throws IOException {
    try {
        client.indices().delete(new DeleteIndexRequest("prod_index"), RequestOptions.DEFAULT);
    } catch (Exception e) {
        // 인덱스가 존재하지 않는 경우 무시
    }
    createIndexWithNGramAnalyzer();
    syncDataToElasticsearch();
}

// 주기적으로 Elasticsearch 서버에 데이터를 동기화하는 메소드
// 1분마다 실행
@Scheduled(fixedRate = 60000)
public void syncDataToElasticsearch() throws IOException {
    List<ElasticSyncDTO> dataList = auctionDAO.aucProdAll();
    for (ElasticSyncDTO data : dataList) {
        Map<String, Object> jsonMap = new HashMap<>();
        jsonMap.put("gubun", data.getGubun());
        jsonMap.put("productSeq", data.getProductSeq());
        jsonMap.put("title", data.getTitle());
        jsonMap.put("price", data.getPrice());
        jsonMap.put("idate", data.getIdate());
        jsonMap.put("imgName", data.getImgName());

        IndexRequest request = new IndexRequest("prod_index")
            .id(String.valueOf(data.getProductSeq()))
            .source(jsonMap, XContentType.JSON);
        client.index(request, RequestOptions.DEFAULT);
    }
    logger.info("Elasticsearch 서버에 데이터 동기화 메소드 실행");
}
```

실시간 검색 기능 메서드

```
// 비슷한 title 그룹화
public List<Map<String, Object>> groupingSearch(String keyword) throws IOException {
    SearchRequest searchRequest = new SearchRequest("prod_index");
    SearchSourceBuilder sourceBuilder = new SearchSourceBuilder();

    // 검색 쿼리 및 그룹핑 설정
    sourceBuilder.query(QueryBuilders.matchQuery("title", keyword).analyzer("standard"))
        .aggregation(AggregationBuilders.terms("group_by_title").field("title.keyword"));

    searchRequest.source(sourceBuilder);

    SearchResponse searchResponse = client.search(searchRequest, RequestOptions.DEFAULT);

    List<Map<String, Object>> results = new ArrayList<>();
    Aggregations aggregations = searchResponse.getAggregations();
    Terms terms = aggregations.get("group_by_title");

    int putCnt = 0;

    for (Terms.Bucket bucket : terms.getBuckets()) {
        Map<String, Object> result = new HashMap<>();
        result.put("title", bucket.getKeyAsString());
        result.put("count", bucket.getDocCount());
        results.add(result);
        if (putCnt == 14) break;
        putCnt++;
    }

    return results;
}
```



기능 설명 메인 화면

★ 고객님의 안녕하세요 ★ 로그아웃 마이페이지



장고물품 | 판매물품

어떤 상품을 찾으시나요?

물품등록 채팅 알림 8

☰ 카테고리

실시간 인기 상품

1위 모동숲 팔아요 전체 베스트가 주고 싶던거 같아요... → 양태 희망합니다 54,000원	2위 튀동숲 팔아요 미개봉 세장품입니다. 45,000원	3위 헬로키티 키링 팔아요 미개봉 세장품 앞줄가능 예뻐볼거 교환불가 4,500원	4위 <배터리100%>애플워치 se 44mm 새고나서 안 쓰고 거의 보라만 해서 배터리 효율 100%입니다 220,000원	5위 레이헬록스 메리제인 펌프 스 6cm 225사이즈 225사이즈(정사이즈) 6cm 펌프 날형 굽면 x(7)본으로 구매 해드 89,000원	6위 그랑헨드 무나키 향수 총해 5월에 구매하였고 3~4회 부 렸습니다 각인 굽어오리 굽쳐라게 -- 4,000원
7위 포차코 동전지갑 팔아요 포차코 명품 동전지갑 판매합니다 마년 12월에 구매하고 개봉만하 18,000원	8위 포켓몬 빠오 반기 안리오 포차코 가자 물 다 개봉전개만 했습니다 개당 0.4 일괄 0.7 택이 별도 7,000원	9위 aa aa 50,000원	10위 핀과 제이크의 어드벤처타 임 인형을 팝니다 교환 환불 안됩니다 굽거래기 택포 입니다 50,000원	11위 반다이 건담mg 반다이 건담mg, 누건담, 도색 완성 품입니다. 도색이 멋진 제품입니다. 500,000원	12위 맛도리 팥조림 팔아요 완전 맛있어요 진짜! 8,000원

조회수 상위 12개 제품 노출

```
@GetMapping("/main")
public String main(Model model, Principal principal) {
    List<CategoryDTO> mainCateList = mainservice.mainCateList();
    model.addAttribute("mainCateList", mainCateList); // 카테고리 서비스 호출
    String loginId = (principal != null) ? principal.getName() : "admin";
    model.addAttribute("userId", loginId);
```

```
int alarmCount = alarmService.alarmCount(loginId);
model.addAttribute("alarmCount", alarmCount);
```

```
model.addAttribute("rankProdList", mainservice.rankProdList());
model.addAttribute("recommandList", mainservice.recommandList(loginId));
model.addAttribute("recentProdList", mainservice.recentProdList());
return "/main/main";
```

```
<select id="prodRank" resultType="com.shinhan.heehee.dto.response.MainProdRankDTO">
    SELECT MIN(IMG.IMG_SEQ) AS IMG_SEQ, IMG_NAME, SELL.ARTICLE_TITLE,
    SELL.INTRODUCE, SELL.PRODUCT_PRICE, SELL.PRODUCT_SEQ, VL.VIEW_COUNT
    FROM SELL_PRODUCT SELL
    INNER JOIN IMG_FILES IMG ON SELL.PRODUCT_SEQ = IMG.TABLE_PK
    LEFT JOIN (
        SELECT PRODUCT_SEQ, COUNT(*) AS VIEW_COUNT
        FROM VIEW_LOG
        GROUP BY PRODUCT_SEQ
    ) VL ON SELL.PRODUCT_SEQ = VL.PRODUCT_SEQ
    WHERE IMG.IMG_SEQ IN (
        SELECT MIN(IMG_SEQ)
        FROM IMG_FILES
        WHERE IMG_CATE_SEQ = 2
        GROUP BY TABLE_PK
    )
    AND ROWNUM <![CDATA[<]]> 13
    AND SELL.PRO_STATUS NOT IN ('판매중지', '판매보류', '거래완료', '예약중')
    AND SELL.ID IS NOT NULL
    GROUP BY IMG.IMG_NAME, SELL.ARTICLE_TITLE, SELL.INTRODUCE, IMG_NAME, SELL.PRODUCT_PRICE,
    SELL.PRODUCT_SEQ, VL.VIEW_COUNT
    ORDER BY VL.VIEW_COUNT DESC, IMG_SEQ
</select>
```



기능 설명 메인 화면

```
model.addAttribute("recommandList", mainservice.recommandList(loginId));
```

당신을 위한 추천 상품

					
25,000원 4시간전	18,000원 17시간전	220,000원 18시간전	4,500원 18시간전	265,000원 4시간전	15,000원 3시간전

모든 상품을 랜덤으로 30개 조회

```

<select id="prodreco" parameterType="String" resultType="com.shinhan.heehee.dto.response.MainProdRecoDTO">
WITH MIN_IMG_SEQ AS (
SELECT
SELL.ARTICLE_TITLE,
MIN(IMG.IMG_SEQ) AS MIN_IMG_SEQ
FROM
SELL_PRODUCT SELL
INNER JOIN
IMG_FILES IMG
ON
SELL.PRODUCT_SEQ = IMG.TABLE_PK
WHERE
IMG.IMG_CATE_SEQ = 2
AND SELL.PRO_STATUS NOT IN ('판매중지', '판매보류', '거래완료', '예약중')
AND SELL.ID IS NOT NULL
AND SELL.ID != #{userId}
GROUP BY
SELL.ARTICLE_TITLE
),
SELECTED_ROWS AS (
SELECT
IMG.IMG_SEQ,
IMG.IMG_NAME,
SELL.ARTICLE_TITLE,
SELL.PRODUCT_PRICE,
SELL.CREATE_DATE,
SELL.PRODUCT_SEQ,
CASE
WHEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) <![CDATA[<]]> 60 THEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) || '초전'
WHEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) <![CDATA[<]]> 3600 THEN ROUND((SYSDATE - CREATE_DATE)*24*60) || '분전'
WHEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) <![CDATA[<]]> 86400 THEN ROUND((SYSDATE - CREATE_DATE)*24) || '시간전'
ELSE TRUNC(SYSDATE - CREATE_DATE) || '일전'
END AGO TIME,
ROWNUM AS RNUM
FROM
MIN_IMG_SEQ
INNER JOIN
SELL_PRODUCT SELL
ON
MIN_IMG_SEQ.ARTICLE_TITLE = SELL.ARTICLE_TITLE
INNER JOIN
IMG_FILES IMG
ON
MIN_IMG_SEQ.MIN_IMG_SEQ = IMG.IMG_SEQ
WHERE
IMG.IMG_CATE_SEQ = 2
AND SELL.PRO_STATUS NOT IN ('판매중지', '판매보류', '거래완료', '예약중')
AND SELL.ID IS NOT NULL
AND SELL.ID != #{userId}
ORDER BY
DBMS_RANDOM.RANDOM
)
SELECT *
FROM SELECTED_ROWS
WHERE RNUM <![CDATA[<]]> 31

</select>

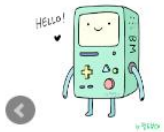





```



기능 설명 메인 화면

```
model.addAttribute("recentprodList", mainservice.recentprodList());
```

방금 등록된 상품

					
비모 그림 팔아요	밥도둑 새우장 팔아요	맛도러 광조림 팔아요	반다이 건담mg	레고 미니피규어6100812 김 은2 미개봉 팝니다	7/12 한화이글스 LG 야구 1루 내야지정역 4면역 팔아요
2,000원	15,000원	8,000원	500,000원	400,000원	25,000원
36분전	34분전	34분전	4시간전	4시간전	4시간전

● ● ● ●

최근 등록된 상품 30개 조회

```
<select id="recently" resultType="com.shinhan.heehee.dto.response.MainProdRecentlyDTO">
  WITH MIN_IMG_SEQ AS (
  SELECT
    SELL.ARTICLE_TITLE,
    MIN(IMG.IMG_SEQ) AS MIN_IMG_SEQ
  FROM
    SELL_PRODUCT SELL
  INNER JOIN
    IMG_FILES IMG
  ON
    SELL.PRODUCT_SEQ = IMG.TABLE_PK
  WHERE
    IMG.IMG_CATE_SEQ = 2
    AND SELL.PRO_STATUS NOT IN ('판매중지', '판매보류', '거래완료', '예약중')
    AND SELL.ID IS NOT NULL
  GROUP BY
    SELL.ARTICLE_TITLE
  )
  SELECT * FROM (
  SELECT
    IMG.IMG_SEQ,
    IMG.IMG_NAME,
    SELL.ARTICLE_TITLE,
    SELL.PRODUCT_PRICE,
    SELL.CREATE_DATE,
    SELL.PRODUCT_SEQ,
    CASE
      WHEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) <![CDATA[<]]> 60 THEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) || '초전'
      WHEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) <![CDATA[<]]> 3600 THEN ROUND((SYSDATE - CREATE_DATE)*24*60) || '분전'
      WHEN ROUND((SYSDATE - CREATE_DATE)*24*60*60) <![CDATA[<]]> 86400 THEN ROUND((SYSDATE - CREATE_DATE)*24) || '시간전'
      ELSE TRUNC(SYSDATE - CREATE_DATE) || '일전'
    END AGO_TIME,
    ROWNUM AS RNUM
  FROM
    MIN_IMG_SEQ
  INNER JOIN
    SELL_PRODUCT SELL
  ON
    MIN_IMG_SEQ.ARTICLE_TITLE = SELL.ARTICLE_TITLE
  INNER JOIN
    IMG_FILES IMG
  ON
    MIN_IMG_SEQ.MIN_IMG_SEQ = IMG.IMG_SEQ
  WHERE
    IMG.IMG_CATE_SEQ = 2
    AND SELL.PRO_STATUS NOT IN ('판매중지', '판매보류', '거래완료', '예약중')
    AND SELL.ID IS NOT NULL
  ORDER BY
    (SYSDATE - CREATE_DATE)
  )
  WHERE RNUM <![CDATA[<]]> 31
</select>
```



기능 설명 중고제품 상세 페이지(구매자 입장)



디지털 웨어러블(워치/밴드) (애플워치 SE 44mm) → **대분류 > 소분류 (상품명)**

<배터리100%>애플워치 se 44mm → **글 제목**

220,000원 → **제품 가격**

2024/07/08 · 조회 5 · 찜 0 → **찜 기능**

- 제품 상태: 사용감 없음
- 거래 방식: 택배
- 배송비: 5,000원

판매자와 채팅하기

즉시구매

해당 물품 판매자 정보

물품 정보

사고나개 안 쓰고 거의 보관만 해개 배터리 효율 100%입니다! 필름 안 깨놔개 기스는 조금 있어요(사진 여러장 올려줬으니 참고 해주세요) 네이비색 정품 스트랩 하나랑 충전기 드릴게요 예닐 불가하구 쿨거기 택포입니다:) 번개페이도 받을게요

판매자 정보



후시딘
안녕하세요 후시딘입니

```

<select id="productInfo" parameterType="com.shinhan.heehee.dto.request.ProductDetailRequestDTO" resultType="com.shinhan.heehee.dto.i
SELECT
SELL.*,
CATE.*,
TYPE.PROD_NAME,
JJIM_CNT.COUNT AS JJIM_CNT,
USER_INFO.PROFILE_IMG,
USER_INFO.NICK_NAME,
USER_INFO.USER_RATING AS USER_RATING,
USER_INFO.USER_INTRODUCE,
(SELECT COUNT(*) FROM VIEW_LOG WHERE PRODUCT_SEQ = #{PROD_SEQ}) AS VIEW_CNT,
(SELECT COUNT(*) FROM JJIM WHERE PRODUCT_SEQ = #{PROD_SEQ}) AS JJIM_CNT,
(SELECT COUNT(*) FROM JJIM WHERE PRODUCT_SEQ = #{PROD_SEQ} AND ID = #{userId}) AS specifiedJjimCnt
FROM
SELL_PRODUCT SELL
INNER JOIN
PRODUCT_TYPE TYPE ON SELL.SELECT_SEQ = TYPE.PRODUCT_SEQ
INNER JOIN
PRODUCT_CATEGORY CATE ON TYPE.PRODUCT_CATE_SEQ = CATE.PRODUCT_CATE_SEQ
LEFT JOIN
(SELECT PRODUCT_SEQ, COUNT(*) AS COUNT FROM JJIM GROUP BY PRODUCT_SEQ) JJIM_CNT ON SELL.PRODUCT_SEQ = JJIM_CNT.PRODUCT_SEQ
LEFT JOIN
(SELECT ID, PROFILE_IMG, NICK_NAME, USER_RATING, USER_INTRODUCE FROM HH_USER) USER_INFO ON SELL.ID = USER_INFO.ID
WHERE
SELL.PRODUCT_SEQ = #{PROD_SEQ}
AND
SELL.PRO_STATUS NOT IN('거래완료')
AND
SELL.ID IS NOT NULL
</select>

```

물품의 정보들과 조회수, 찜, 글 작성 날짜 조회

📌 기능 설명 중고제품 상세 페이지(구매자 입장)

2024/07/08 · 조회 19 · 찜 3



- 제품 상태: 신품
- 거래 방식: 택배
- 배송비: 2,500원



2024/07/08 · 조회 20 · 찜 4



- 제품 상태: 신품
- 거래 방식: 택배
- 배송비: 2,500원

찜 하트 변경 스크립트

```
<select id="selectJjim" parameterType="int" resultType="int">
    SELECT COUNT(*) AS JJIM_CNT
    FROM JJIM
    WHERE PRODUCT_SEQ = #{productSeq}
</select>
```

```
<insert id="insertJjim" parameterType="com.shinhan.heehee.dto.request.JjimDTO">
    INSERT INTO JJIM VALUES (#{productSeq}, #{id})
</insert>
```

찜 조회 및 삽입 쿼리

```
function addJjim() {
    $('#emptyHeart').hide();
    $('#fullHeart').show();

    var productSeq = ${info.productSeq};
    var id = "${userId}";
    $.ajax({
        url: '/heehee/sell/insertJjim',
        method: 'POST',
        contentType: 'application/json',
        data: JSON.stringify({ "productSeq": productSeq, "id": id }),
        success: function (data, status, xhr) {
            console.log(data);
            if(data.success == true) {
                showToast(data.message);
                $('#product_etc').val('${info.jjimCnt}');
                updateProduct();
            } else {
                showToast(data.message);
            }
        },
        error: function (data, status, err) {
            console.log(err);
        }
    });
};
```



기능 설명 중고제품 상세 페이지(구매자 입장)

2024/07/08 · 조회 20 · 찜 4



- 제품 상태: 신품
- 거래 방식: 택배
- 배송비: 2,500원



2024/07/08 · 조회 20 · 찜 3



- 제품 상태: 신품
- 거래 방식: 택배
- 배송비: 2,500원

찜 하트 변경 스크립트

```
<delete id="deleteJjim" parameterType="com.shinhan.heehee.dto.request.JjimDTO">
  DELETE FROM JJIM
  WHERE product_seq = #{productSeq}
  AND ID = #{id}
</delete>
```

찜 삭제 쿼리

```
function deleteJjim() {
  $('#fullHeart').hide();
  $('#emptyHeart').show();

  var productSeq = ${info.productSeq};
  var id = "${userId}";
  $.ajax({
    url: '/heehee/sell/deleteJjim',
    method: 'DELETE',
    contentType: 'application/json',
    data: JSON.stringify({ "productSeq": productSeq, "id": id }),
    success: function (data, status, xhr) {
      console.log(data);
      if(data.success == true) {
        showToast(data.message);
        $('#product_etc').val('${info.jjimCnt}');
        updateProduct();
      } else {
        showToast(data.message);
      }
    },
    error: function (data, status, err) {
      console.log(err);
    }
  });
}
```

📌 기능 설명 중고제품 상세 페이지(구매자 입장)

추천 제품



```
<select id="prodrecoDetail" parameterType="com.shinhan.heehee.dto.request..
  SELECT *
  FROM (SELECT IMG.IMG_NAME, SELL.PRODUCT_SEQ
        FROM IMG_FILES IMG
        JOIN SELL_PRODUCT SELL ON IMG.TABLE_PK = SELL.PRODUCT_SEQ
        WHERE IMG.IMG_CATE_SEQ = 2
              AND CREATE_DATE >= SYSDATE - 30
              AND SELL.PRO_STATUS NOT IN('판매중지', '판매보류', '거래완료', '예약중')
              AND SELL.ID IS NOT NULL
              AND SELL.ID != #{userId}
              ORDER BY DBMS_RANDOM.RANDOM)
  WHERE ROWNUM <![CDATA[<]]> 6
        AND PRODUCT_SEQ != #{PROD_SEQ}
</select>
```

본인을 제외한 다른 사용자가
30일 이전까지 올린 제품들 추천

📌 기능 설명 중고제품 상세 페이지(구매자 입장)

최근 본 상품



TOP



최근 본 상품



TOP

```
<select id="selectRecently" parameterType="String" resultType="com.shinhan.heehee.dto.response.ProdDetailImgDTO">
  SELECT *
FROM (
  SELECT *
FROM (
  SELECT SELL.PRODUCT_SEQ, IMG.IMG_SEQ, IMG.IMG_NAME, MAX(VLOG.CREATE_DATE) CRE_DATE
  FROM VIEW_LOG VLOG
  JOIN IMG_FILES IMG ON VLOG.PRODUCT_SEQ = IMG.TABLE_PK
  JOIN SELL_PRODUCT SELL ON VLOG.PRODUCT_SEQ = SELL.PRODUCT_SEQ
  WHERE VLOG.ID = #{userId}
  AND IMG.IMG_CATE_SEQ = 2
  AND IMG.IMG_SEQ = (
    SELECT MIN(IMG_SEQ)
    FROM IMG_FILES
    WHERE TABLE_PK = VLOG.PRODUCT_SEQ
    AND IMG_CATE_SEQ = 2
  )
  AND SELL.ID IS NOT NULL
  GROUP BY SELL.PRODUCT_SEQ, IMG.IMG_SEQ, IMG.IMG_NAME
  )
  ORDER BY CRE_DATE DESC
)
WHERE ROWNUM <![CDATA[<]]> 5
```

최근 본 상품이
최대 4개까지 저장됨



기능 설명 중고제품 상세 페이지(구매자 입장)

상태가 판매중일 경우



디지털 > 게임/타이틀 (모여봐요 동물의 숲)

모동숲 팔아요

54,000원

2024/07/08 · 조회 4 · 찜 0

- 제품 상태: 사용감 없음
- 거래 방식: 픽거래

직거래일 경우 즉시구매 불가능

판매자와 채팅하기

즉시구매



디지털 > 웨어러블(워치/밴드) (애플워치 SE 44mm)

<배터리100%>애플워치 se 44mm

220,000원

2024/07/08 · 조회 5 · 찜 0

- 제품 상태: 사용감 없음
- 거래 방식: 택배
- 배송비: 5,000원

택배인 경우에만 즉시구매 가능

판매자와 채팅하기

즉시구매

물품 정보

진짜 비싸게 주고 샀던거 같아요... 상태 최상입니다

판매자 정보

호시딘 ★★★★★
안녕하세요 후시딘입니다

물품 정보

사고나져 안 쓰고 거의 보관만 해겨 배터리 효율 100%입니다! 필름 안 꺼내서 기스는 조금 있어요(사진 여러장 올려왔으니 참고 하주세요) 네이버백 평종 스트랩 하나랑 충전기 드릴게요 예닐 불가하구 콜거시 택포입니다:) 번개레이도 받을게요

판매자 정보

호시딘 ★★★★★
안녕하세요 후시딘입니다



```
<c:if test="${userId != info.id && info.proStatus == '판매중' && info.deal == '택배'}">
  <div class="button-container">
    <button loginUserId="${userId}" sellerId = "${info.id}" sellSeq = "${info.prod
    <button id="gobuy" style="cursor: pointer" prodSeq = "${info.productSeq}" al
  </div>
</c:if>
```

```
<c:if test="${userId != info.id && info.proStatus == '판매중' && info.deal == '직거래'}">
  <div class="button-container">
    <button loginUserId="${userId}" sellerId = "${info.id}" sellSeq = "${info.prod
    <button id="disabled_btn" disabled>즉시구매</button>
  </div>
</c:if>
```



기능 설명 중고제품 상세 페이지(판매자 입장)

상태가 판매중일 경우



디지털 > 웨어러블(워치/밴드) (애플워치 SE 44mm)

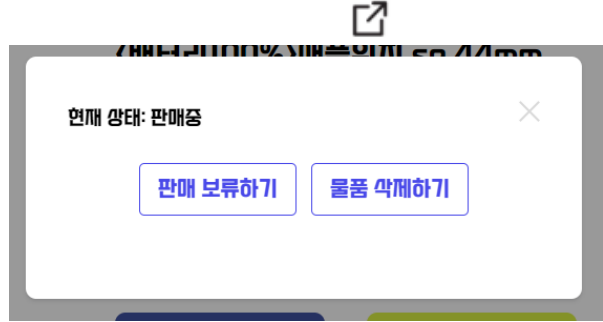
<배터리100%>애플워치 se 44mm

220,000원

2024/07/08 · 조회 10 · 찜 0

- 제품 상태: 사용감 적음
- 거래 방식: 택배
- 배송비: 5,000원

판매자는 본인 물품이므로
찜 기능이 없음



해당 물품의 정보 수정 가능

물품정보 수정

판매상태 수정

판매보류 또는 글 삭제 가능

물품 정보

사고나개 안 쓰고 거의 보관만 해서 배터리 효율 100%입니다! 필름 안 꺼내서 기스는 조금 있어요(사진 여러장 올려줬으니 참고해주세요) 네이비색 정품 스트랩 하나랑 충전기 드릴게요 예닐 불가하구 쿨거시 택배입니다:) 번개페이도 받을게요

판매자 정보



후시딘

안녕하세요 후시딘입니다



기능 설명 중고 제품 상세 페이지(구매자 입장)

상태가 판매중이 아닐 경우



키덜트 > 피규어/인형 (보노보노)



보노보노 미니 일체형 보조배터리

70,000원

2024/06/18 · 조회 97 · 찜 1

- 제품 상태: 신품
- 거래 방식: 택배
- 배송비: 5,000원

※ 판매 중지된 물품은 구매할 수 없습니다. →

판매 상태에 대한 설명

```

<c:if test="${userId != info.id && info.proStatus == '판매중지'}">
  <div class="button-container">
    <p id="reserve_explanation">※ 판매 중지된 물품은 구매할 수 없습니다.</p>
  </div>
</c:if>

```



기능 설명 중고제품 상세 페이지(판매자 입장)

상태가 판매중이 아닐 경우



키덜트 > 피규어/인형 (보노보노)



보노보노 미니 일체형 보조배터리

70,000원

2024/06/18 · 조회 92 · 찜 1

- 제품 상태: 신품
- 거래 방식: 택배
- 배송비: 5,000원

※ 판매 중지된 물품은 수정할 수 없습니다.

판매중지 사유

안전성 기준에 부합하지 않으므로 판매중지됨 →

판매자에겐 사유가 표기됨

```

<c:if test="${userId == info.id && info.proStatus == '판매중지'}">
  <div class="text-container">
    <p id="reserve_explanation">※ 판매 중지된 물품은 수정할 수 없습니다.</p>
    <p id="banReason">판매중지 사유</p>
    <p id="paused_explanation">${info.productBanReason}</p>
  </div>
</c:if>

```



기능 설명 물품 정보 수정(판매자 입장)

물품정보 수정

판매상태 수정

정보 수정

상품 이미지

사진 추가



카테고리

대분류	시계
신발	남성시계
가방/지갑	여성시계
시계	시계용품
주얼리	
패션 액세서리	
디지털	

글 제목

(배터리100%)애플워치 se 44mm

상품 이름

애플워치 SE 44mm

상품 상태

인품(미사용) 사용중 없음 사용중 있음 고장/파손

상품 설명

사과나서 안 쓰고 거의 보관만 해서 배터리 요즘 100%입니다!
필름 안 깨난게 기스는 조금 있어요(사진 여러장 올려왔으니 참고 해주세요)
내이전에 명품 스트랩 하나랑 충전기 드릴게요
애는 불거하구 글거게 부탁드립니다!
번개메이도 받을게요

상품 가격

220000

거래 유형

택배 픽거래

배송비

5000

```
// 수정하기 페이지
@GetMapping("/productmodify/{prod_seq}")
public String modify(@PathVariable("prod_seq") Integer prodSeq, Model model, Principal principal) {
    String userId = (principal != null) ? principal.getName() : "admin";

    ProductDetailRequestDTO sampleDTO = new ProductDetailRequestDTO(prodSeq, userId);

    int alarmCount = alarmService.alarmCount(userId);
    model.addAttribute("alarmCount", alarmCount); // 알람 개수

    List<CategoryDTO> mainCateList = mainservice.mainCateList(); // 카테고리 서비스 호출
    model.addAttribute("mainCateList", mainCateList);

    /* if(prodInfo == null) return "/"; */
    ProdDetailDTO prodDetailDTO = productmodifyservice.prodInfo(sampleDTO);
    if(!prodDetailDTO.getId().equals(userId)) {
        model.addAttribute("rankProdList", mainservice.rankProdList());
        model.addAttribute("recommandList", mainservice.recommandList(userId));
        model.addAttribute("recentprodList", mainservice.recentprodList());
        return "redirect:/main";
    }

    model.addAttribute("userId", userId);

    model.addAttribute("info", prodDetailDTO);

    model.addAttribute("prodImgList", productmodifyservice.prodImg(prodSeq));

    model.addAttribute("categoryList", productmodifyservice.category());
    return "/used/productmodify";
}
```

로그인한 유저의 아이디를 받아와 물품을 등록한 아이디와 일치할 경우에만 해당 페이지 접근 가능, 기존에 입력한 데이터들이 이미 입력되어 있음



기능 설명

물품 정보 수정(판매자 입장)

물품정보 수정

판매상태 수정

```
<delete id="deleteImgFiles" parameterType="com.shinhan.heehee.dto.request.ImageFileDTO">
  DELETE FROM IMG_FILES
  WHERE IMG_CATE_SEQ = 2
  AND TABLE_PK = #{tablePk}
  AND IMG_SEQ = #{imgSeq}
</delete>
```

```
<insert id="insertImgFile" parameterType="com.shinhan.heehee.dto.request.ImageFileDTO">
  INSERT INTO IMG_FILES VALUES(SEQ_IMG_FILES.nextval, #{imgName},#{tablePk},2,#{userId})
</insert>
```

```
<select id="productModify" parameterType="int" resultType="com.shi
  SELECT SELL.PRODUCT_SEQ, MIN(IMG.IMG_NAME) AS IMG_NAME,
  SELL.INTRODUCE, SELL.PRODUCT_PRICE
  FROM SELL_PRODUCT SELL
  INNER JOIN IMG_FILES IMG ON SELL.PRODUCT_SEQ = IMG.TABLE_PK
  WHERE IMG.IMG_CATE_SEQ = 2
  GROUP BY SELL.PRODUCT_SEQ, SELL.INTRODUCE, SELL.PRODUCT_PRICE
</select>
```

이미지는 기존에 있던 파일을 삭제할 수도 있으므로 갱신이 아닌 삭제 후, 삽입하는 방식을 사용함

정보 수정

상품 이미지

사진 추가



카테고리

대분류	시계
신발	남성시계
가방/지갑	여성시계
시계	시계용품
주얼리	
패션 액세서리	
디지털	

글 제목

<배터리100%>애플워치 se 44mm

상품 이름

애플워치 SE 44mm

상품 상태

인품(미사용) 사용감 적음 사용감 많음 고장/파손

상품 설명

새로나와 안 쓰고 거의 보관만 해줘 배터리 효율 100%입니다!
필름 안 깨놔서 기스는 조금 있어요(사진 여러장 올려줬으니 참고해주세요)
네이버액 명품 스트랩 하나랑 충전기 드릴게요
예능 볼거리를 골거기 택포입니다:)
번개메이도 받을게요

상품 가격

220000

거래 유형

택배 짝거래

배송비

5000



기능 설명

물품 정보 수정(판매자 입장)

물품정보 수정

판매상태 수정

```
<update id="proStatusSelling" parameterType="int">
  UPDATE SELL_PRODUCT
  SET PRO_STATUS = '판매중'
  WHERE PRODUCT_SEQ = #{productSeq}
</update>
```

```
<update id="proStatusPutoff" parameterType="com.shinhan.heehee.dto.request.ProdDetailDTO">
  UPDATE SELL_PRODUCT
  SET PRO_STATUS = '판매보류'
  WHERE PRODUCT_SEQ = #{productSeq}
</update>
```

판매보류 또는 물품 삭제가 가능함



📌 기능 설명 판매자 프로필 페이지(구매자 입장)



후시딘

안녕하세요 후시딘입니다

평점
★★★★☆
가입인기 40일 판매횟수 0회

판매중인 제품목록

- 포켓몬 바오 반끼 안리오 포차 코 가차
- 포차코 동면제갈 팔아요
- (배터리100%)애플워치 se 44mm
- 헬로키티 키링 팔아요
- 안리오 미니 피규어 팔아요

```
@GetMapping("/sellerProfile/{id}")
public String home(@PathVariable("id") String id, Model model, Principal principal) {
    if(principal != null) {
        int alarmCount = alarmService.alarmCount(principal.getName());
        model.addAttribute("alarmCount", alarmCount); // 알림 개수
    }

    List<CategoryDTO> mainCateList = mainservice.mainCateList(); // 카테고리 서비스 호출
    model.addAttribute("mainCateList", mainCateList);

    model.addAttribute("sellerinfo", sellerprofileservice.sellerinfo(id));
    model.addAttribute("sellerprodList", sellerprofileservice.sellerprod(id));
    model.addAttribute("dealComplete", sellerprofileservice.dealComplete(id));
    return "/used/sellerProfile";
}
```

판매자의 간략한 정보 호출을 위함

📌 기능 설명 판매자 프로필 페이지(구매자 입장)



후시딘

안녕하세요 후시딘입니다

평점

★★★★☆

가입한지 40일 판매횟수 0회

판매중인 제품목록



포켓몬 삐오 반지 안리오 포차 코 가챠



포차코 동전지갑 팔아요



<배터리100%>애플워치 se 44mm



헬로키티 키링 팔아요



안리오 미니 피규어 팔아요

```
<select id="sellerInfo" parameterType="String" resultType="com.shinhan.heehee.dto.response.SellerProfileDTO">
    SELECT HH_USER.*, ROUND(SYSDATE - CREATE_DATE) AS CREATE_DIFF
    FROM HH_USER
    WHERE ID = #{id}
</select>
```

```
<select id="sellerprod" parameterType="String" resultType="com.shinhan.heehee.dto.response.SellerProdDTO">
    SELECT *
    FROM SELL_PRODUCT SELL, IMG_FILES IMG
    WHERE SELL.PRODUCT_SEQ = IMG.TABLE_PK
    AND SELL.ID = #{id}
    AND IMG_CATE_SEQ = 2
    AND IMG.IMG_SEQ IN (SELECT MIN(IMG_SEQ)
        FROM IMG_FILES
        WHERE IMG_CATE_SEQ = 2
        AND SELL.PRO_STATUS NOT IN('판매중지','판매보류','거래완료','예약중')
        GROUP BY TABLE_PK)
    <!-- AND ROWNUM <![CDATA[<]]> 6 -->
    ORDER BY DBMS_RANDOM.RANDOM
</select>
```

해당 물품을 등록한 판매자의 정보와 판매자가 판매중인 물품 리스트 조회



기능 설명 중고 물품 등록(판매자 입장)

SALE 물품등록

물품 등록

상품 이미지

재판 수가

카테고리

대분류	남영의류
여성류	아우터
남영의류	상의
신발	바지
가방/지갑	캠프수트
시계	켓업/켓트
슈얼러	인더웨어/올웨어

글 제목

글 제목을 입력해주세요.

상품 이름

상품명을 입력해주세요.

상품 상태

신중(미사용) 사용감 없음 사용감 많음 고장/파손

상품 설명

설명을 입력해주세요.

상품 가격

가격을 입력해주세요.

거래 유형

택배 직접거래

배송비

배송비를 입력해주세요.

등록하기

// 등록하기 페이지

@GetMapping("/productregi")

```
public String registry(Model model, Principal principal) {
```

```
String userId = principal.getName();
```

```
int alarmCount = alarmService.alarmCount(userId);
```

```
model.addAttribute("alarmCount", alarmCount); // 알람 개수
```

```
List<CategoryDTO> mainCateList = mainservice.mainCateList(); // 카테고리 서비스 호출
```

```
model.addAttribute("mainCateList", mainCateList);
```

```
/* if(prodInfo == null) return "/"; */
```

```
model.addAttribute("userId", userId);
```

```
model.addAttribute("categoryList", productmodifyService.category());
```

```
/* return "/used/productregi"; */
```

```
return "/used/productregi";
```

```
}
```

비 로그인 사용자가 진입 시 메인화면으로 이동
됨



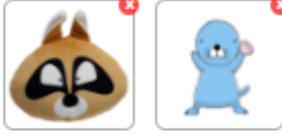
기능 설명 중고 물품 등록(판매자 입장)

SALE 물품등록

물품 등록

상품 이미지

사진 추가



카테고리

대분류	카테고리
디지털	패규어/인형
가전제품	레고/블럭
스포츠/레저	프리모빌
차량/오토바이	RC/드론
스타트업	보드게임
키보드	새마을은행

글 제목

보노보노 세트 팔아여~~

상품 이름

보노보노 세트

상품 상태

신중(미사용) 사용중 먹음 사용중 많음 고장/파손

상품 설명

보노보노 귀여운 두마리입니다
사용중 별로 없어요!!!

상품 가격

50000

거래 유형

택배 픽거래

배송비

2500

등록하기

```
<insert id="insertImgFile" parameterType="com.shinhan.heehee.dto.request.ImageFileDTO">
  INSERT INTO IMG_FILES VALUES(SEQ_IMG_FILES.nextval, #{imgName},#{tablePk},2,#{userId})
</insert>
```

```
<insert id="insertProduct" parameterType="com.shinhan.heehee.dto.request.ProductModifyRequestDTO">
  <selectKey keyProperty="prodSeq" order="BEFORE" resultType="int">
    SELECT SEQ_SELL_PROD.NEXTVAL FROM DUAL
  </selectKey>
  INSERT INTO SELL_PRODUCT
  VALUES (#{prodSeq}, #{articleTitle}, #{productPrice}, #{introduce}, #{state}, '판매중', #{deal}, #{dCharge}, #{selectSeq}, #{userId}, SYSDA
</insert>
```

```
<insert id="insertProductCategory" parameterType="com.shinhan.heehee.dto.request.ProductModifyRequestDTO">
  <selectKey keyProperty="selectSeq" order="BEFORE" resultType="int">
    SELECT SEQ_PROD_TYPE.NEXTVAL FROM DUAL
  </selectKey>
  INSERT INTO PRODUCT_TYPE
  VALUES (#{selectSeq}, #{cateSeq}, #{prodName})
</insert>
```

이미지, 물품 정보, 물품 카테고리

- ▶ 사전 기획의 관점에서 프로젝트 결과물에 대한 완성도 평가 (8점)

기존에 운영중인 중고 사이트에 실시간 경매라는 기능을 추가하여 재미와 신선함을 느낄 수 있다고 생각하고 가장 어려웠던 경매 기능을 포함해 주요 기능을 모두 구현한 점, 2순위로 이야기했던 찜 기능, 엘라스틱 서치, 사용자 평점 관리 기능, 최근 본 상품 기능까지 구현했다는 점에서 높은 점수를 주었다.

2점을 깎은 이유는 2순위 기능 중 시세조회 기능과 사기조회 기능을 구현하지 못했고 구현해보고 싶었던 스웨거 및 junit 테스트 자동화를 하지 못했기 때문이다.

05

K-Digital Training

자체 평가 의견

▶ 개인 또는 우리 팀이 잘한 부분과 아쉬운 점



손동희

저희 팀은 단합이 잘 되어 의견 충돌 없이 각자 맡은 역할을 성실히 수행하였습니다. 다만, 제가 개인적으로 실력이 부족해 팀에 많이 의존하게 된 것이 아쉽습니다.



이두리

저희 팀의 장점은 서로 모르는 부분에 대해 질문하고 도움을 주며 어느 팀보다 단합이 잘 되었던 것입니다. 하지만 제 개인적으로는 당당한 페이지를 구현하는 데 필요한 시간을 잘못 분배하여 프로젝트의 마무리 단계에서 시간이 촉박해진 점이 아쉽습니다.



최재명

다양한 전공 및 경험을 가진 사람들의 장점이 잘 결합되어 훌륭한 시너지를 경험할 수 있었습니다. 개인적으로 아쉬웠던 점은 말았던 파트의 필요 기술 상 보다 최신 및 고급의 기술을 경험하지 못했던 것 같습니다.



신현상

각자 맡은 역할을 성실히 수행하여서 책임감이 강한 팀원들을 만나 보람했습니다. 아쉬운 점은 너무 책임감이 강한 나머지 다들 건강에 해가 될 정도로 열심히 했던 부분에서는 아쉬움이 남습니다.



신정인

서로 도움을 주고 받으며 각자 맡은 기능을 끝까지 구현해낸 점, 팀원간 소통이 원활해 프로젝트를 진행하며 생기는 수정사항도 서로 오해 없이 공유된 점이 좋았습니다. 아쉬운 점은 역할을 분배할 때 뷰 페이지 수를 고려하지 못해 몇명이 너무 많은 페이지를 맡게 된 점입니다.



황수영

팀 분위기 자체가 다 같이 열심히 하는 분위기여서 더 열심히 노력하게 되고 잘 하고 싶은 욕심이 생기는 것 같다. 이번 프로젝트에서 웹소켓이라는 개념을 처음 알게 되었는데 팀원들이 많이 도와준 덕분에 무사히 마무리한 것 같다.

05

K-Digital Training

자체 평가 의견

▶ 프로젝트 결과물의 추후 개선점 또는 보완할 점

- 1 반응형 웹, 채팅방 목록 순서 등 사용자의 편의성을 위한 디테일 부족
- 2 경매 페이지 CSS 디테일 부족
- 3 UI 통일성 부족
- 4 시세조회 등 후순위 기능 미구현

▶ 프로젝트를 진행하며 느낀 점이나 경험한 성과



손동희

프로젝트를 진행할수록 보완할 점이 지속적으로 늘어났던 경험을 하였습니다. 따라서, 프로젝트의 전체적인 흐름을 이해하고, 설계를 완벽하게 짜놓은 상태에서 진행하는 것이 매우 중요한 일이라고 생각하게 되었습니다.



이두리

이번 팀프로젝트는 웹개발의 전반적인 과정을 직접 참여하고 경험할 수 있는 뜻깊은 시간이었습니다. 프론트엔드와 백엔드, 데이터베이스를 연동하며 웹 애플리케이션을 완성하는 과정에서 많은 것을 배울 수 있었습니다.



최재명

첫 팀 프로젝트를 해보면서 하나의 웹사이트가 만들어지는 흐름에 대해 잘 이해할 수 있었습니다. 또한 생각보다 사소한 이유로도 발생하는 오류들이 많았으며, 이를 해결하는 과정들을 겪으면서 책이나 기술문서로 배운 것들을 최대한 많이 사용해보야겠다고 생각했습니다.



신현상

모든 팀원분들이 열심히 참여해주시고, 즐거운 분위기 속에서 프로젝트를 진행할 수 있도록 도와주셔서 진심으로 감사드립니다. 이번 프로젝트를 통해 엘라스틱 서치와 소켓 등 다양한 기술을 사용해보며 정말 흥미로운 경험을 할 수 있어서 좋았습니다.



신정인

이전까지는 백엔드만 맡아 프로젝트를 해왔었는데 처음으로 풀스택 프로젝트를 진행하게 되었습니다. 프론트와 백이 서로 어떻게 데이터를 요청하고 응답받는지 이해하고 프로그램의 전반적인 흐름을 이해할 수 있는 좋은 계기가 되었습니다.



황수영

조원분들 모두 본인이 맡은 기능이 아님에도 같이 고민해주고 작은 부분을 정할 때도 소통을 잘 해주어서 원활하게 진행할 수 있었습니다. 프로젝트를 시작했을때 이해했다고 생각한 부분들이 미흡한 점이 많아서 더 열심히 해야겠다고 다짐하는 계기가 되었고 다들 실력자셔서 다방면으로 많이 배울 수 있는 기회가 되어 너무 좋았습니다.