

포트폴리오

발표

GALGODA 호텔 예약 사이트

이다예

목차

프로젝트 개요

세부기능 소개

담당기능 시연

코드 리뷰

Q&A

프로젝트 개요

GALGODA : 태그로 원하는 호텔을 찾아 예약하는 웹사이트 개발 프로젝트

기획의도

사용자가 원하는 호텔을 찾는데 소요하는 시간을 최소화하고자
태그를 등록해 호텔을 빠르게 검색하고 예약 가능하게 함

수행기간

2024.02.14 - 2024.03.29

기획 및 화면/DB 설계(3주), 프로젝트 구현 (3주)

개발환경

운영체제 : Windows OS

개발환경 : Eclipse, Visual Studio Code

DBMS : Oracle DB - sqldeveloper

Server : Apache Tomcat 9.0

개발언어 : Java SE 11, HTML5, CSS, JavaScript, jQuery

라이브러리 및 기술 : Servlet, JSP, JSTL, Bootstrap

인기 호텔



GALGODA서울



GALGODA인천



GALGODA대구



GALGODA제주

[GALGODA 프로젝트 Github Link](#)

세부기능 소개



사용자(고객)

- 로그인 : 회원가입, 로그인, ID/PW 찾기
- 호텔 : 태그/호텔명 검색, 찜하기, 리뷰 조회
- 예약/결제 : 호텔 선택, **이용금액 결제**
- 마이페이지 : 예약 조회/변경/취소,
리뷰 등록/삭제, 정보수정,
회원탈퇴, 위시리스트 관리



사용자(호텔)

- 로그인 : 회원가입, 로그인, ID/PW 찾기
- 호텔 : 호텔 정보 수정, 리뷰관리,
사용자 신고, 객실 등록/수정
- 예약 : 예약 현황 조회, 예약 승인/취소
- 마이페이지 : 정보수정/회원탈퇴
- 고객센터 : 공지사항, 문의사항 확인/답변



관리자

- 홈페이지 : 이용약관 관리
- 호텔 관리 : 태그 관리, 등록 현황 조회,
호텔 계정 관리
- 사용자 관리 : 개인정보 조회,
사용자 계정 관리, 회원 설정
- 고객센터 : 문의사항 관리,
공지사항 관리, 신고 관리

담당기능 시연 - 회원가입

The screenshot shows a registration process divided into two main sections:

Step 1: 회원가입 (User Registration)

- Header:** Includes a search bar ("호텔검색"), login ("로그인"), and registration ("회원가입") buttons.
- Section 1:** Contains a checked checkbox "[필수] GALGODA 이용약관" (Required) and a scrollable text area for terms and conditions.
- Section 2:** Contains a checked checkbox "[필수] 개인정보 수집 및 이용" (Required) and a scrollable text area for privacy policy.
- Section 3:** Contains an unchecked checkbox "[선택] 개인정보 마케팅 활용 동의" (Optional) and a scrollable text area for marketing consent.
- Footer:** A "본인인증 하기" (Proceed with Verification) button.

Step 2: 본인인증 (Identity Verification)

- Header:** Shows a progress bar indicating the user has moved from Step 1 to Step 2.
- Text Fields:** Two input fields for "휴대폰 번호" (Mobile Number) and "인증번호" (Verification Number), each with a "받기" (Receive) and "확인" (Check) button.
- Buttons:** A blue "회원가입 화면으로 이동" (Move to Registration Screen) button.

1 : JavaScript를 활용해
필수 항목 모두 체크 시에만
본인인증 버튼 활성화

2 : CoolSMS API를 활용해
본인인증 처리

담당기능 시연 - 회원가입

회원가입

1 일반 회원 **2** 호텔 회원

회원가입

기본정보 입력

성명(국문) *			
성명(영문)			
생년월일 *	연도-월-일	□	
이메일 *	<input type="button" value="중복확인"/>		
연락처 *			
주소 *	우편번호	기본주소1	<input type="button" value="우편번호 찾기"/>
기본주소2		세부주소	

웹사이트 정보 입력

아이디 *	5~12자 이내 영문 또는 영문/숫자 조합	<input type="button" value="중복 확인"/>
비밀번호 *	8~20자 이내 영문/숫자 조합(특수문자 가능)	
비밀번호 확인 *	<input type="button" value="일치 확인"/>	

회원가입 완료

1 : 회원가입 시 일반/호텔 회원으로 DB 구분해서 Insert 처리

2 : Ajax를 활용해 서버로 이메일 중복확인 요청을 전송

3 : 카카오 주소찾기 API를 활용해 사용자의 주소 입력

4 : JavaScript로 아이디 유효성 검사 및 Ajax로 중복확인 요청

담당기능 시연 - 로그인

The screenshot shows the login page of the GALGODA system. At the top, there is a search bar labeled "호텔검색" and two links: "로그인" and "회원가입". Below the search bar, the word "로그인" is centered above a horizontal line. A message "GALGODA에 오신 것을 환영합니다!" is displayed. To the left of the input fields, a blue circle contains the number "1". Below the message, there are three tabs: "고객" (Customer), "호텔" (Hotel), and "관리자" (Manager). The "고객" tab is highlighted with a blue border. Below the tabs are two input fields: "아이디" (ID) and "비밀번호" (Password), each preceded by an icon (user and key). A large brown "로그인" button is positioned below the password field. At the bottom of the form, there are links for "아이디 찾기" (Find ID), "비밀번호 찾기" (Find Password), and "회원가입" (Sign Up).

1 : 사용자의 유저타입 클릭에 따라
숨겨진 입력 필드에 유저타입을
저장해 고객/호텔/관리자 DB별
ID/PW 일치여부 확인 후 로그인

담당기능 시연 - ID찾기/PW재설정

아이디 찾기

1

2

3

비밀번호 재설정

아이디, 휴대폰 번호 입력 및 인증번호 확인 절차를 모두 진행해주세요.

아이디를 입력하세요

휴대폰 번호를 입력하세요

인증번호를 입력하세요

인증번호 받기

인증번호 확인

비밀번호 재설정

1 : CoolSMS API를 활용해 등록된 핸드폰번호로 본인인증

2 : 유저 타입에 따라 각 DB에서 이름과 휴대폰번호로 ID 찾기

3 : 유저타입별 DB에서 ID와 휴대폰번호 조회 확인 시 PW 재설정 화면으로 이동

담당기능 시연 - 마이페이지

The screenshot shows the 'My Page' section of the Galgoda app. At the top, there is a search bar with a magnifying glass icon and the text '호텔검색'. To the right of the search bar are the user's name '홍길동 님' and a profile icon, followed by a 'Logout' button.

The main title '홍길동님의 마이페이지' is displayed above a horizontal line. Below the line, a message reads 'Galgoda에서 여행을 시작해보세요!'.

Three circular icons represent user statistics:

- 예약현황**: 3 (highlighted with a blue circle)
- 위시리스트**: 5
- 작성 리뷰**: 1

Below these stats are three dark brown rectangular buttons labeled:

1. 예약 관리
2. 위시리스트
3. 개인정보 수정

1 : Oracle Count 함수를 이용
사용자별 예약 현황/위시리스트/
리뷰 수를 화면에 표시

2 : 각 메뉴바에 해당하는
페이지 링크 연결 처리

담당기능 시연 - 마이페이지

Q 호텔검색

홍길동 님 로그아웃

마이페이지

예약관리

위시리스트

리뷰관리

개인정보 수정

예약관리

예약관리

GALGODA 제주

2024년 05월 27일 - 2024년 05월 29일

2024년 05월 27일 - 2024년 05월 29일

스탠다드더블 - 총 1개
예약인원 - 총 1명
추가옵션 - 조식, 베드추가

1

2

3

4

Previous 1 2 3 Next

1 : 고객별 예약 내역 조회 시
페이지 처리

2 : DB에서 예약 대기/승인/취소
상태 조회 후 표시

3 : 상세조회 페이지로 이동

4 : 예약 변경/취소 페이지로 이동

담당기능 시연 - 마이페이지

The screenshot shows a web-based hotel booking cancellation system. The top navigation bar includes a search icon, a user profile (홍길동 님), and a log-out button. The main menu on the left has items: 마이페이지 (highlighted in blue), 예약관리, 위시리스트, 리뷰관리, and 개인정보 수정.

The main content area is titled "예약 변경 및 취소" and displays information for a booking at "GALGODA 제주". The "호텔 정보" section shows the hotel name as "GALGODA제주", check-in date as "2024년 05월 27일", and check-out date as "2024년 05월 29일". It also lists 1 guest, 1 standard double room, and 1 guest. Below these are checkboxes for meal type (조식, 레이트체크인, 베드추가, 스파패키지, 무료취소), all of which are checked except for 레이트체크인.

The "예약자 정보" section contains fields for the booking contact: 예약자 (홍길동), 연락처 (010-2015-3458), 이메일 (test@gmail.com), 결제금액 (원) (200000), and 결제수단 (카카오페이지).

At the bottom, there are three buttons: "예약변경" (highlighted in blue), "예약변경 후 결제하기", and "예약취소".

Four numbered circles (1, 2, 3, 4) highlight specific features:

- 호텔명은 readonly 처리 (Hotel name is read-only)
- 체크인/체크아웃날짜, 객실타입, 객실 수를 변경할 때 예약금액이 변경되도록 JavaScript 활용 (Check-in/check-out dates, room type, and number of guests change when the deposit amount changes due to JavaScript usage)
- 클릭 시 변경 내용 DB에 반영 (Changes reflected in the DB upon click)
- 포트원 API를 활용하여 카카오페이지 테스트결제로 구현 (Implemented using the Portain API with KakaoPay test payment integration)

- 1 : 호텔명은 `readonly` 처리
- 2 : 체크인/체크아웃날짜, 객실타입, 객실 수를 변경할 때 예약금액이 변경되도록 JavaScript 활용
- 3 : 클릭 시 변경 내용 DB에 반영
- 4 : 포트원 API를 활용하여 카카오페이지 테스트결제로 구현

담당기능 시연 - 마이페이지

Q 호텔검색

홍길동 님 로그아웃

마이페이지

- [예약관리](#)
- 위시리스트**
- [리뷰관리](#)
- [개인정보 수정](#)

위시리스트



GALGODA부산

[예약하기](#) 2



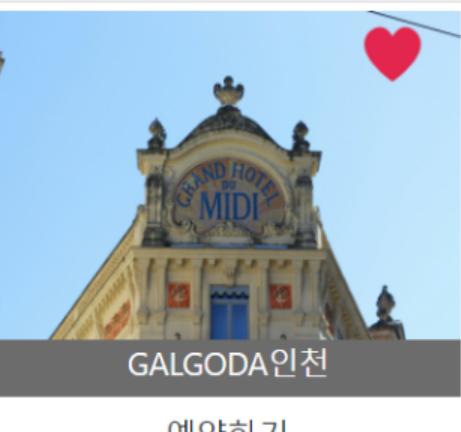
GALGODA서울

[예약하기](#)



GALGODA제주

[예약하기](#)



GALGODA인천

[예약하기](#)



GALGODA대구

[예약하기](#)

1 : 클릭 시 해당 하트에 연결된 체크박스 ID를 Ajax로 전달해 위시리스트 해제 처리

2 : 해당 호텔 예약 페이지로 이동

담당기능 시연 - 마이페이지

호텔검색

홍길동 님 로그아웃

마이페이지

예약관리

위시리스트

리뷰관리

개인정보 수정

1

2

홍길동 님 (user01)

홍길동

user01

....

비밀번호 변경

연락처

010-2015-3458

이메일

hong@gmail.com

증복확인

주소

08505 서울 금천구

우편번호 찾기

(가산동, km타워)

2층, 3층

3

정보수정

4

회원탈퇴

- 1 : PW를 입력해 화면에 진입
아이디는 readonly로 변경 불가
- 2 : 모달창에서 비밀번호 변경 가능
- 3 : 이메일 중복확인을 진행해야
정보 수정 완료 버튼 활성화
- 4 : 클릭 시 회원탈퇴 페이지로 이동

담당기능 시연 - 마이페이지

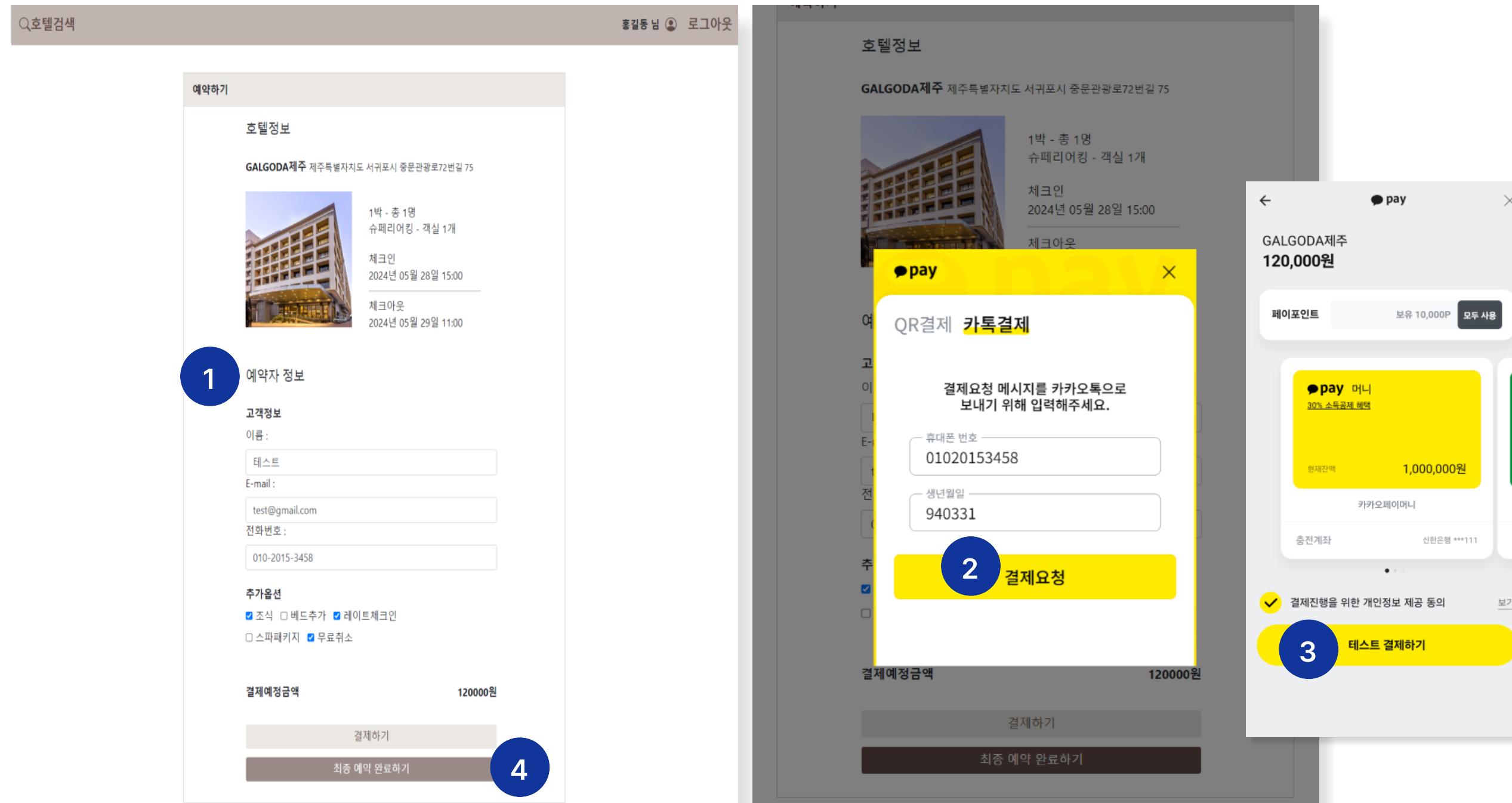
The screenshot shows a web interface for managing a user account. At the top, there is a navigation bar with a search icon, the text '호텔검색', and a user profile section labeled '홍길동 님' with a log-out button. On the left, a sidebar menu lists '마이페이지' (selected), '예약관리', '위시리스트', '리뷰관리', and '개인정보 수정'. The main content area is titled '회원 탈퇴' and contains a warning message: '회원 탈퇴를 신청하기 전에 안내 사항을 꼭 확인해주세요.' Below this, five checkboxes are listed, each with a detailed description of what happens upon logout:

- 사용하고 계신 아이디는 탈퇴할 경우 재사용 및 복구가 불가능합니다
탈퇴한 아이디는 본인과 타인 모두 재사용 및 복구가 불가하오니 신중하게 선택하시기 바랍니다.
- 탈퇴 후 회원정보 및 사이트 이용기록은 모두 삭제됩니다.
회원정보 및 예약 내역 등 사이트 이용기록은 모두 삭제되며, 삭제된 데이터는 복구되지 않습니다.
- 탈퇴 후에도 게시판형 서비스에 등록한 게시물은 그대로 남아 있습니다.
본 사이트에 올린 리뷰 및 문의글은 탈퇴 시 자동 삭제되지 않고 그대로 남아 있습니다.
삭제를 원하는 게시글이 있다면 반드시 탈퇴 전 비공개 처리하거나 삭제하시기 바랍니다.
탈퇴 후에는 회원정보가 삭제되어 본인 여부를 확인할 수 있는 방법이 없어, 게시글을 임의로 삭제해드릴 수 없습니다.
- 탈퇴 후에는 기존 아이디로 다시 가입할 수 없으며 아이디와 데이터는 복구할 수 없습니다.
게시판형 서비스에 남아 있는 게시글은 탈퇴 후 삭제할 수 없습니다.
- 안내 사항을 모두 확인하였으며, 이에 동의합니다.

In the bottom right corner of the content area, there is a blue circular button with the number '1' and a dark brown rectangular button labeled '회원탈퇴'.

1 : JavaScript로 각 체크박스 상태를 모니터링해 모든 체크박스 선택 시 회원탈퇴 버튼 활성화

담당기능 시연 - 예약 상세 및 결제



1 : 로그인한 사용자의 정보와 다른 예약정보도 기재할 수 있게 함

2 : 결제하기 버튼 클릭 시 포트원 API를 활용해 카카오페이지 테스트 결제 진행

3 : 입력한 정보에 따라 카카오페이지 테스트 결제 실행

4 : 결제하기 완료가 되어야만 최종 예약 완료 버튼 활성화되며 예약 완료 시 DB에 예약 대기 상태로 반영됨

코드 리뷰 - 로그인

LoginController.java

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String userType = request.getParameter("userType");
    String userId = request.getParameter("userId");
    String userPwd = request.getParameter("userPwd");

    if ("customer".equals(userType)) {
        Customer customer = new CommonService().loginCustomer(userId, userPwd);
        HttpSession session = request.getSession();
        String redirectURL = request.getContextPath();

        // 로그인 결과에 따른 처리
        if (customer != null) {
            // 성공적으로 로그인된 경우
            session.setAttribute("loginCustomer", customer);
        } else {
            // 로그인 실패한 경우
            // 로그인 실패 메시지 처리 등
            session.setAttribute("alertMsg", "고객 로그인에 실패했습니다.");
            redirectURL += "/views/common/loginMain.jsp";
        }
    }

    response.sendRedirect(redirectURL);
}

```

1 : 로그인 화면에서 유저타입 선택 시
input type=hidden 영역에서
userType 값이 전달됨

해당 값을
request.getParameter로
가져와서 String 변수에 설정

userType이 "customer",
"hotel", "admin"과 일치할 시
세션에 로그인 정보를 저장함

고객은 로그인 시 메인 페이지로,
호텔/관리자는 각 마이페이지로
이동하도록 처리

코드 리뷰 - 본인인증

authenti.jsp

```

<script>

var verificationCode = "";

function sendVerificationCode() {
    var phoneNumber = document.getElementById("inputPhoneNumber").value;

    if (phoneNumber != "") {
        $.ajax({
            url: "<%=contextPath%>/authentiSms.co",
            type: "GET",
            data: { phone: phoneNumber },
            dataType: "json",
            success: function(response) {
                if (response.message === "인증번호가 전송되었습니다.") {
                    verificationCode = response.verificationCode; // 전역 변수에 저장
                    alert("인증번호 전송에 성공했습니다.");
                } else {
                    alert("인증번호 전송에 실패했습니다.");
                }
            },
            error: function() {
                alert("서버 오류로 인증번호를 발송할 수 없습니다.");
            }
        });
    } else {
        alert("휴대폰 번호를 입력해주세요.");
    }
}

```

1 : 회원가입 본인인증 화면에서
인증번호 발송 버튼 클릭 시 코드

핸드폰번호가 공백이 아닐 경우
/authentiSms.co 컨트롤러로
핸드폰번호를 보내면서 요청

서버측 요청 처리 여부에 따라
그에 맞는 alert창 표시

코드 리뷰 - 본인인증

SignUpAuthentiController.java

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String phone = request.getParameter("phone");
    HttpSession session = request.getSession(true);
    String verificationCode = CoolSmsSender.sendVerificationCode(phone, session);

    response.setContentType("application/json");
    response.setCharacterEncoding("UTF-8");

    PrintWriter out = response.getWriter();
    Gson gson = new Gson();

    // HashMap을 사용하여 JSON 응답 생성
    Map<String, String> jsonResponse = new HashMap<>();

    if (verificationCode != null) {
        jsonResponse.put("message", "인증번호가 전송되었습니다.");
        jsonResponse.put("verificationCode", verificationCode); // verificationCode를 JSON 응답에 포함
    } else {
        jsonResponse.put("message", "인증번호 전송에 실패했습니다.");
    }

    // Gson을 사용하여 HashMap을 JSON 문자열로 변환
    String json = gson.toJson(jsonResponse);
    out.println(json);
    out.close();
}

```

1 : jsp단에서 전달 받은 phone값을
CoolSmsSender 클래스로 전달

CoolSmsSender 클래스에서
인증번호를 반환 받음

HashMap<String, String>을
사용하여 JSON 응답을 생성하고,
Gson 라이브러리를 사용하여 이를
JSON 문자열로 변환

코드 리뷰 - 본인인증

CoolSmsSender.java

```

package com.galgoda.common.model.service;

import java.util.HashMap;

public class CoolSmsSender {
    public static String generateVerificationCode() {
        // 랜덤한 4자리 숫자 생성
        Random random = new Random();
        int verificationCode = random.nextInt(10000); // 0부터 9999 사이의 랜덤한 숫자 생성
        return String.format("%04d", verificationCode); // 4자리로 맞추기 위해 포맷팅
    }

    public static String sendVerificationCode(String phone, HttpSession session) {
        String apiKey = "I[REDACTED]"; // 쿨에스엠에스 API 키
        String apiSecret = "P[REDACTED]"; // 쿨에스엠에스 API 시크릿 키

        Message coolsms = new Message(apiKey, apiSecret);

        // 발송할 메시지 내용 및 수신자 번호 설정
        String verificationCode = generateVerificationCode(); // 랜덤한 인증번호 생성
        String message = "인증번호는 [" + verificationCode + "]입니다."; // 발송할 메시지 내용
        HashMap<String, String> params = new HashMap<>();
        params.put("to", phone); // 수신자 번호
        // 발신자 번호는 서비스 제공업체에서 승인된 번호여야 함
        params.put("from", "01020153458"); // 발신자 번호 (본인의 발신자 번호로 설정해야 함)
        params.put("text", message); // 발송할 메시지 내용

        try {
            // 인증번호 발송 요청
            coolsms.send(params);
            session.setAttribute("sentCode", verificationCode);
            return verificationCode; // 생성된 인증번호 반환
        } catch (CoolSmsException e) {
            // 예외 발생 시 로그 출력
            e.printStackTrace();
            return null; // 실패 시 null 반환
        }
    }
}

```

1 : CoolSms API를 활용해
인증번호를 생성해서 발송

코드 리뷰 - 결제

ReservationHotelDetailForm.jsp

```
//버튼 클릭하면 실행
function payment() {
    var hotelName = document.getElementById("hotelName").value;
    var roomName = document.getElementById("roomName").value;

    var currentDateString = new Date().toLocaleString();
    var combinedString = hotelName + " - " + roomName + " - " + currentDateString;
    //가맹점 주문번호가 계속 중복되지 않게 바뀌어야 해서 이렇게 설정

    var amount = document.getElementById("totalPrice").value;
    var buyerEmail = document.getElementById("email").value;
    var buyerName = document.getElementById("userName").value;
    var buyerTel = document.getElementById("phone").value;

    IMP.init(''); // 아임포트 관리자 콘솔에서 확인한 '가맹점 식별코드' 입력
    IMP.request_pay({
        // param
        pg: "kakaopay.TC0ONETIME", // pg사명 or pg사명.CID (잘못 입력할 경우, 기본 PG사가 띄워짐)
        pay_method: "card", // 지불 방법
        merchant_uid: combinedString, // 가맹점 주문번호 (아임포트를 사용하는 가맹점에서 중복되지 않은 임의의 문자열을 입력) - 중복되지 않게 루마입으로 입력함
        name: hotelName, // 결제창에 노출될 상품명
        amount: amount,
        buyer_email: buyerEmail,
        buyer_name: buyerName,
        buyer_tel: buyerTel
    }, function(rsp) { // callback
        if (rsp.success) {
            alert("결제가 완료되었습니다. 최종 예약을 완료해주세요.");
            $('#confirm_res').prop("disabled", false); // 최종 예약 버튼 활성화
        } else {
            alert("결제에 실패했습니다.");
        }
    })
};
```

1 : 포트원 API를 활용해 결제 처리

가맹점 주문번호가 계속 중복되지
않도록 호텔명과 방 이름, 날짜를
조합해 가맹점 주문번호로 생성

Q&A