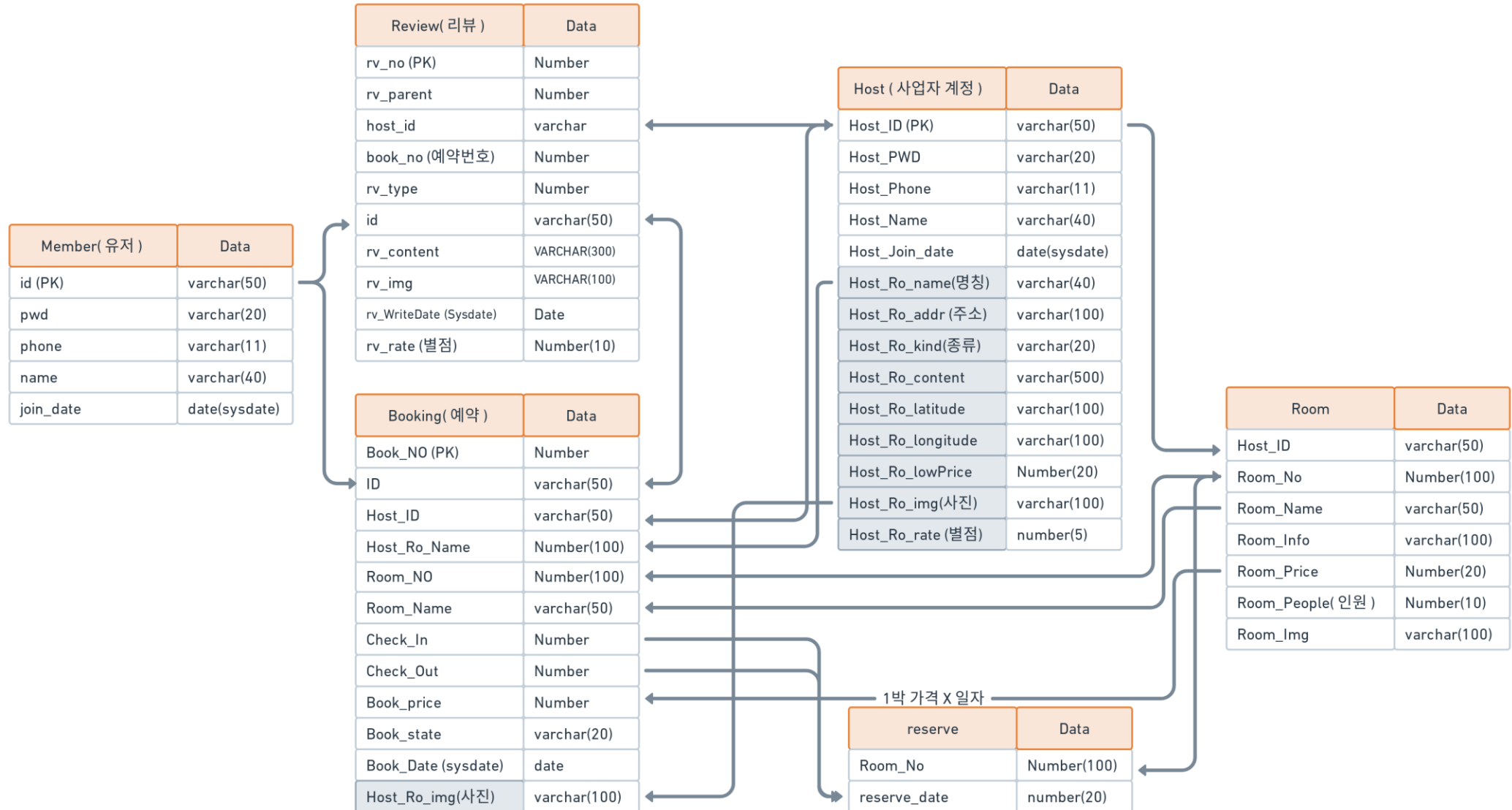


STAYING 숙박 예약 플랫폼

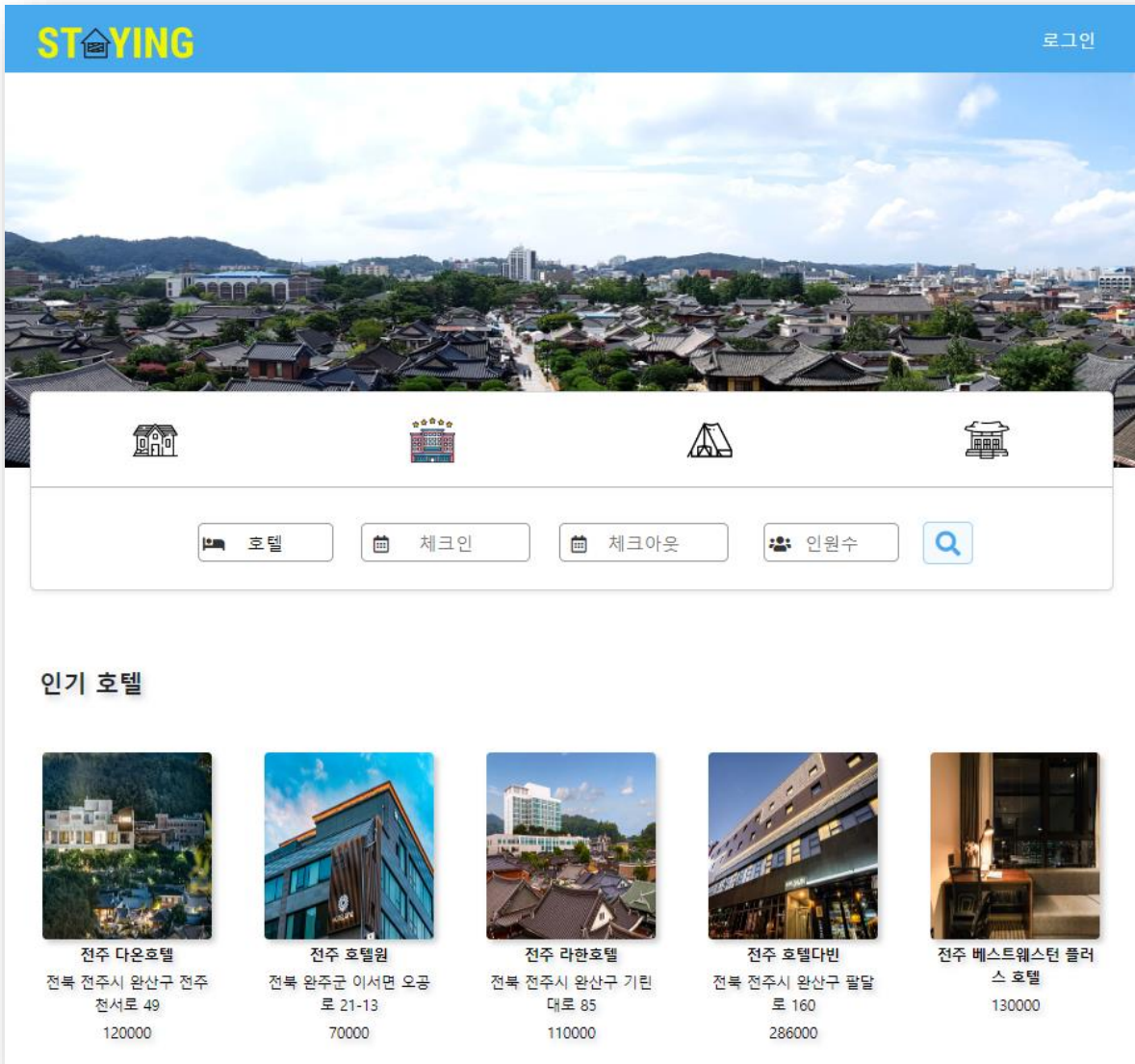


프론트 엔드 : 김상윤, 이예림
백 엔드 : 백종은

DB 구조



프로젝트 개요



구분

웹 개발

명칭

Staying 전주 숙박 예약 플랫폼

개발 인원

프론트 엔드 2명, 백 엔드 1명

개발 환경

VS CODE, Eclipse, SQLdeveloper

개발 기간

22. 10. 25 ~ 22. 11. 25 (1 개월)

개발 언어

프론트 엔드

▶ HTML, javascript, css, jQuery

백엔드

▶ JAVA

웹 서버

Apache Tomcat

데이터베이스

Oracle DB

소개

전주 지역의 숙박시설 등록, 조회 및 예약이 가능한 웹 사이트

서비스 개요 -- 메인페이지



index.jsp 헤더 – 로그인한 유저에 따라 다른 내용을 출력

```
<nav role="navigation">
  <c:choose>
    <c:when test="${user_type eq 'user'}"> <!-- 개인 유저 로그인인 경우 -->
      <a href="${contextPath}/member/member_info.do">마이페이지</a>
      <a class="btn_login" href="${contextPath}/member/logout.do">로그아웃</a>
    </c:when>
    <c:when test="${user_type eq 'host'}"> <!-- 사업자 로그인인 경우 -->
      <a href="${contextPath}/room/my_room_info.do">내 사업장</a>
      <a class="btn_login" href="${contextPath}/member/logout.do">로그아웃</a>
    </c:when>
    <c:otherwise> <!-- 로그인 하지않은 경우 -->
      <a class="btn_login" href="${contextPath}/login.jsp">로그인</a>
    </c:otherwise>
  </c:choose>
</nav>
```

숙박 검색 영역 – 검색 조건을 서블릿으로 넘겨 DB에서 검색

```
// 조건으로 사업장을 검색하는 메서드 !!!!!!!!!!!!!!!
public List<HostVO> searchHost(int ckiDate, int ckoDate, String room_kind, String lowPrice, String highPrice, int room_people, String rate){
    List<HostVO> searchedHost = new ArrayList<HostVO>(); // 리턴할 리스트

    try {
        conn=dataFactory.getConnection();

        String query = "SELECT * FROM hosttbl WHERE "
            + "host_id IN(SELECT DISTINCT host_id FROM roomtbl " // 조건 검색한 방 목록이 포함된 업체를 가져옴
            + "WHERE room_no NOT IN"
            + "(SELECT DISTINCT room_no FROM reservetbl WHERE reserve_date BETWEEN ? and ?)" // 지정한 체크인 체크아웃 일자 사이 예약이 없는방
            + " AND room_people >= ?"; // 숙박 가능 인원수를 지정

        if(lowPrice != null && !(lowPrice.equals(""))){ // 가격대 지정이 있는 경우 (지정이 없는 경우 적용 X)
            query += " AND room_price BETWEEN "+lowPrice+" AND "+highPrice; // 가격대를 지정
        }
        query += " "; // 중간 끝표
        if(rate != null && !(rate.equals(""))){ // 별점 조건이 있는 경우 ( 없는 경우 적용 X)
            query += " AND host_ro_rate > " + rate; // 해당 별점보다 높은 업체리스트를 구분
        }
        query += " AND host_ro_kind = ? ORDER BY host_ro_lowprice ASC"; // 숙박유형 추가 및 가격순 정렬

        System.out.println(query); // 쿼리문 확인

        pstmt=conn.prepareStatement(query);

        pstmt.setInt(1, ckiDate); // 체크인 날짜
        pstmt.setInt(2, ckoDate); // 체크아웃 날짜
        pstmt.setInt(3, room_people); // 숙박 인원
        pstmt.setString(4, room_kind); // 방 종류

        ResultSet rs = pstmt.executeQuery();
```

인기 호텔



전주 다운호텔
전북 전주시 완산구 전주
전서로 49
120000



전주 호텔W
전북 완주군 이서면 오공
로 21-13
70000



전주 라하호텔
전북 전주시 완산구 기린
대로 85
110000

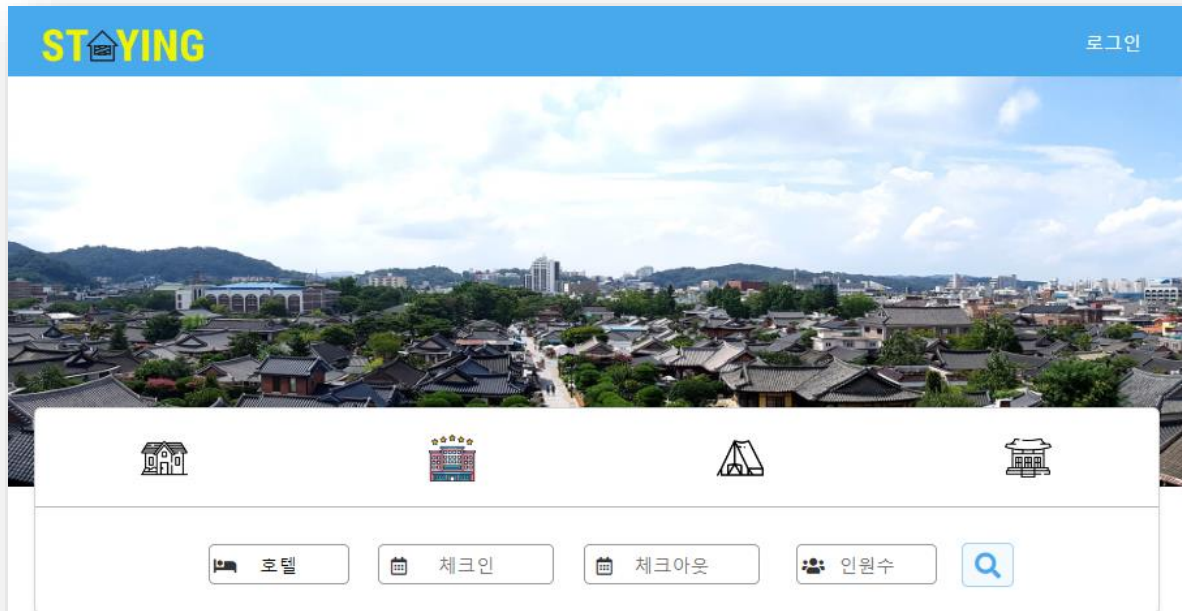


전주 호텔다빈
전북 전주시 완산구 팔달
로 160
286000



전주 베스트웨스턴 플러
스 호텔
130000

서비스 개요 -- 메인페이지



인기 호텔



전주 다온호텔
전북 전주시 완산구 전주
전서로 49
120000



전주 호텔원
전북 완주군 이서면 오공
로 21-13
70000



전주 라한호텔
전북 전주시 완산구 기린
대로 85
110000



전주 호텔다빈
전북 전주시 완산구 팔달
로 160
286000



전주 베스트웨스턴 플러
스 호텔
130000

인기 업체 리스트 - 숙박 유형별로 별점 상위 5개를 가져옴

컨트롤러에서 숙박 유형별 별점 상위5위를 담아 index로 포워딩
(DAO에서 상위 리스트를 요청)

```
}else if(action.equals("/main_page.do")) {  
    // 메인페이지를 요청했을 경우  
  
    HttpSession session = request.getSession(); // 세션 생성 또는 가져옴  
    String user_type = (String) session.getAttribute("user_kind"); // 로그인된 유저 타입을 확인  
  
    // 각 숙박업소간 상위 5위 리스트를 생성  
    List<HostVO> bestPension = hostService.bestRate("pension");  
    List<HostVO> bestHotel = hostService.bestRate("hotel");  
    List<HostVO> bestCamping = hostService.bestRate("camping");  
    List<HostVO> bestHanok = hostService.bestRate("hanok");  
  
    request.setAttribute("best_pension", bestPension); // 각 유형별 리스트를 포워딩  
    request.setAttribute("best_hotel", bestHotel);  
    request.setAttribute("best_camping", bestCamping);  
    request.setAttribute("best_hanok", bestHanok);  
    |  
    request.setAttribute("user_type", user_type); // 로그인된 유저 타입을 전달  
  
    nextPage = "/index.jsp";  
    DAO에서 별점상위권 리스트를 DB에서 받아와 전달  
    // 별점순으로 상위 숙박업소 5개를 가져오는 메서드  
    public List<HostVO> bestRate(String host_ro_kind){  
        List<HostVO> bestList = new ArrayList<HostVO>(); // 리턴할 리스트  
        try {  
            conn = dataFactory.getConnection();  
  
            String query = "SELECT * FROM"  
                + " (SELECT * FROM hosttbl WHERE host_ro_kind=? ORDER BY host_ro_rate DESC)"  
                + " WHERE ROWNUM <= 5"; // 별점순으로 정렬 후 상위 5개만 추출  
  
            pstmt=conn.prepareStatement(query);  
            System.out.println(query); // 쿼리문 확인  
  
            pstmt.setString(1, host_ro_kind); // 받아온 숙소 유형을 지정  
  
            ResultSet rs = pstmt.executeQuery();  
            while(rs.next()) {  
                String host_id = rs.getString("host_id");  
                String host_ro_name = rs.getString("host_ro_name");  
                String host_ro_addr = rs.getString("host_ro_addr");  
                int host_ro_lowprice = rs.getInt("host_ro_lowprice");  
                String host_ro_img = rs.getString("host_ro_img");  
                int host_ro_rate = rs.getInt("host_ro_rate");  
  
                HostVO hostVO = new HostVO();  
  
                hostVO.setHost_id(host_id);  
                hostVO.setHost_ro_name(host_ro_name);  
                hostVO.setHost_ro_addr(host_ro_addr);  
                hostVO.setHost_ro_lowprice(host_ro_lowprice);  
                hostVO.setHost_ro_img(host_ro_img);  
                hostVO.setHost_ro_rate(host_ro_rate);  
  
                bestList.add(hostVO); // 각 업소 객체를 리스트에 담음  
            }  
            rs.close();  
            conn.close();  
            pstmt.close();  
        }  
    }  
}
```

서비스 개요 - 숙박검색 리스트

STAYING

로그인

호텔

2022-12-07

2022-12-09

2

Q

지도에 표시

검색하기

가격

11000

150000

평점

☐ ★★★★★ +

☐ ★★★★ +

☐ ★★★ +

숙소 유형

☐ 펜션

☒ 호텔

☐ 한옥

☐ 캠핑장

전주 호텔원

다양한 부대시설이 마련되어 있어 편리한 휴식을 누릴 수...

★★★★★ 8.0

1박 / 70000원

전주 라한호텔

루프탑 수영장 그리고 특별한 살롱 한옥마을의 정취를 품...

★★★★★ 9.0

1박 / 110000원

웨스턴 플러스 호텔

전주의 주요 관광지와의 접근성이 뛰어나 편안하고 즐거운...

업체 정보

메인페이지에서 검색 조건으로 검색 한 리스트를 stay_list.jsp로 포워딩
(좌측 검색영역 가격대, 평점 지정시 해당 조건을 추가해 검색)

```

} else if (action.equals("/search.do")) {
    // 검색 요청시 넘겨줄 항목들

    // 체크인 아웃 날짜를 가져와서 '-'를 제거후 int로 저장
    int checkInDate = Integer.parseInt(repString(request.getParameter("check_in")));
    // 체크인 아웃 날짜를 가져와서 '-'를 제거후 int로 저장
    int checkOutDate = Integer.parseInt(repString(request.getParameter("check_out")));

    HttpSession session = request.getSession(); // 세션 생성 또는 가져옴
    session.setAttribute("cki_date", checkInDate); // 체크인 날짜 (yyyyMMdd)
    session.setAttribute("cko_date", checkOutDate); // 체크아웃 날짜 (yyyyMMdd)

    String room_kind = request.getParameter("room_kind"); // 숙소 유형
    String room_lowprice = request.getParameter("min_price"); // 최저가격
    String room_highprice = request.getParameter("max_price"); // 최고가격
    int room_people = Integer.parseInt(request.getParameter("stay_person")); // 숙박인원
    String rate = request.getParameter("grade_value"); // 별점 조건

    List<HostVO> searchedHost = hostService.searchHost(checkInDate, checkOutDate, room_kind, room_lowprice, room_highprice, room_people, rate);

    String user_type = (String) session.getAttribute("user_kind"); // 로그인된 유저 타입을 확인
    request.setAttribute("user_type", user_type); // 로그인된 유저 타입을 전달

    // 검색했던 조건을 다음페이지로 유지하기 위해 검색 조건을 포워딩
    request.setAttribute("check_in", request.getParameter("check_in"));
    request.setAttribute("check_out", request.getParameter("check_out"));
    request.setAttribute("person", room_people);
    request.setAttribute("room_kind", room_kind);

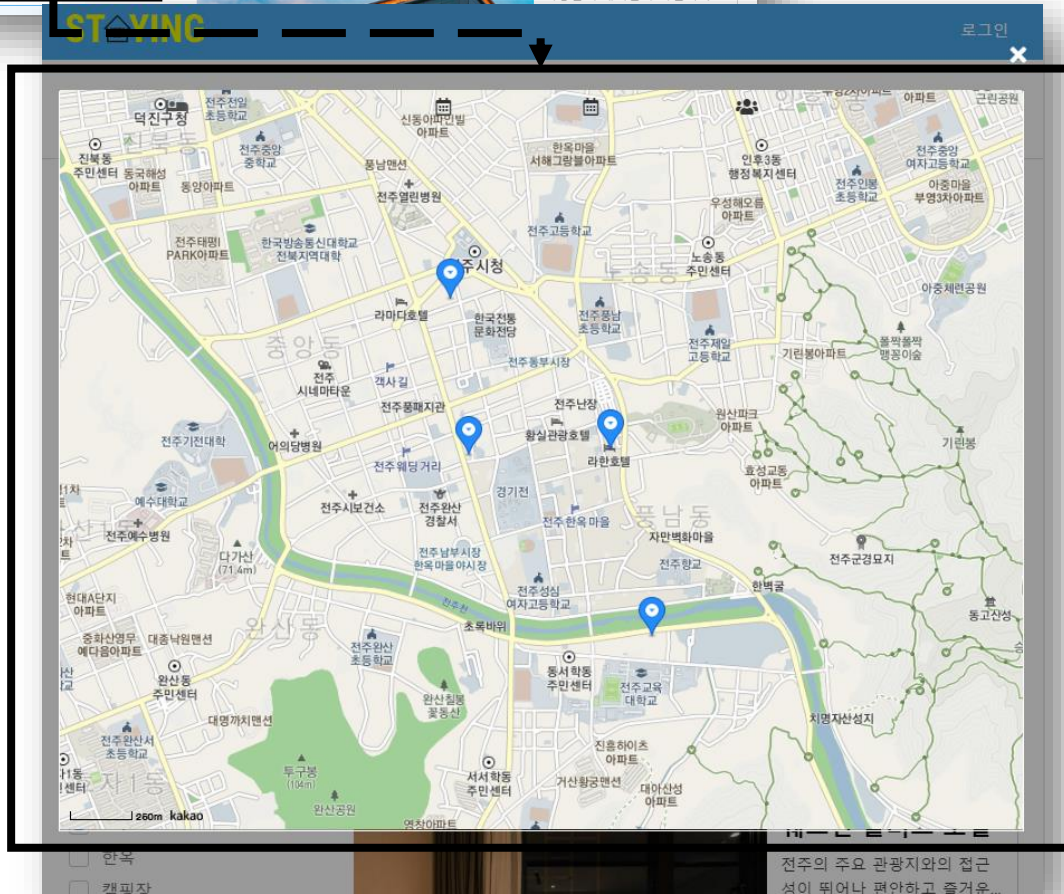
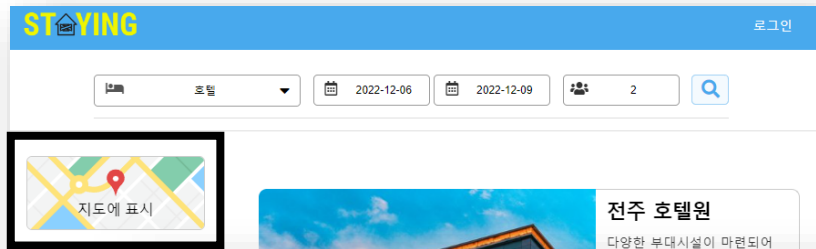
    request.setAttribute("hostList", searchedHost); // 생성한 리스트를 포워딩
    nextPage = "/stay_list.jsp";
}
    
```

포워딩한 숙소 목록을 stay_list.jsp에 출력
해당 숙소 클릭시 host컨트롤러의 hostInfo에 해당 업소의 id를 담아 전달

```

<%-- 숙소 목록 영역 --%>
<section class="stay_list_area">
    <c:forEach var="host" items="${hostList}">
        <a class="stay_content" href="${contextPath}/room/roomInfo.do?host_id=${host.host_id}&host=${host}">
            <div class="stay_image">
                
            </div>
            <div class="stay_info">
                <div class="stay_info_l">
                    <h2>${host.host_ro_name}</h2>
                    <p>${host.host_ro_content}</p>
                    <p class="view_star">${host.host_ro_rate/10}</p>
                    <p>1박 / ${host.host_ro_lowprice }원</p>
                </div>
            </div>
        </a>
    </c:forEach>
</section>
    
```

서비스 개요 - 숙박검색 리스트



업체정보 지도에 표시

컨트롤러에서 포워딩한 숙박업체 리스트에서
업체명과 위도, 경도 정보를 가져와 배열에 담음.

```
<script>
    let map_name = new Array();
    let map_lat = new Array();
    let map_long = new Array();

    <!-- 업체 이름과 업체 위도, 경도를 배열에 담음-->
    <c:forEach var="host" items="${hostList}">
        map_name.push("${host.host_ro_name}");
        map_lat.push("${host.host_ro_latitude}");
        map_long.push("${host.host_ro_longitude}");
    </c:forEach>
</script>
```

담아온 업체명, 위도, 경도 리스트를 지도 모달창을 여는 함수에 담아 실행

```
<div class="map_area">
    <!-- 지도버튼 클릭시 지도를 여는 함수에 업체명, 위도, 경도 리스트를 담아 실행 -->
    <button type="button" class="map" id="map" onclick='openModal(map_name, map_lat, map_long)''>
        
    </div>
    <div>
        
        <p>지도에 표시</p>
    </div>
</button>
</div>
```


서비스 개요 - 숙박업체 정보

STAYING

로그인



웨스턴 플러스 호텔

전주의 주요 관광지와와의 접근성이 뛰어나 편안하고 즐거운 휴식을 누릴 수 있는 호텔입니다.



객실 상세 위치 후기



스탠다드 트윈

투숙 가능인원 2명

편의시설 >

가격

111900원 / 1박

예약하기



디럭스 패밀리 프리미어

업체정보 표시

받은 사업자 id로 해당 사업장의 정보를 조회 후 포워딩

```
// 사업자id로 사업장의 공개 정보를 확인하는 메서드
public HostVO info_ro_host(String host_id) {
    HostVO hostInfo = null;
    try {
        conn=dataFactory.getConnection();

        String query = "SELECT * FROM hosttbl WHERE host_id=?";
        System.out.println(query); // 쿼리문 확인

        pstmt=conn.prepareStatement(query);

        pstmt.setString(1, host_id); // 해당 id의 호스트 정보를 가져옴
        ResultSet rs = pstmt.executeQuery();

        rs.next();

        String host_ro_name = rs.getString("host_ro_name");
        String host_ro_addr = rs.getString("host_ro_addr");
        String host_ro_kind = rs.getString("host_ro_kind");
        String host_ro_content = rs.getString("host_ro_content");
        String host_ro_latitude = rs.getString("host_ro_latitude");
        String host_ro_longitude = rs.getString("host_ro_longitude");
        String host_ro_img = rs.getString("host_ro_img");
        int host_ro_rate = rs.getInt("host_ro_rate");

        hostInfo = new HostVO(); // HostVO 생성 후 정보를 담음

        hostInfo.setHost_id(host_id);
        hostInfo.setHost_ro_name(host_ro_name);
        hostInfo.setHost_ro_addr(host_ro_addr);
        hostInfo.setHost_ro_kind(host_ro_kind);
        hostInfo.setHost_ro_content(host_ro_content);
        hostInfo.setHost_ro_latitude(host_ro_latitude);
        hostInfo.setHost_ro_longitude(host_ro_longitude);
        hostInfo.setHost_ro_img(host_ro_img);
        hostInfo.setHost_ro_rate(host_ro_rate);

        rs.close();
        pstmt.close();
        conn.close();
    } catch (Exception e) {
        System.out.println("업체 정보 확인중 에러 : " + e.getMessage());
    }
    return hostInfo; // hostInfo VO를 반환
}
```

```
// 해당 host_id의 방 목록 조회
public List<RoomVO> hostRoomList(String host_id){
    List<RoomVO> roomList = new ArrayList<RoomVO>();
    try {
        conn = dataFactory.getConnection();

        String query = "SELECT * FROM roomtbl"
            + " WHERE host_id=?"
            + " ORDER BY room_price ASC"; // 낮은 가격부터 정렬
        pstmt=conn.prepareStatement(query);

        pstmt.setString(1, host_id);

        ResultSet rs = pstmt.executeQuery();

        while(rs.next()) {
            String room_name = rs.getString("room_name");
            String room_info = rs.getString("room_info");
            int room_price = rs.getInt("room_price");
            int room_people = rs.getInt("room_people");

            String room_img = rs.getString("room_img");
            int room_no = rs.getInt("room_no");

            RoomVO roomVO = new RoomVO();

            roomVO.setHost_id(host_id);

            roomVO.setRoom_name(room_name);
            roomVO.setRoom_info(room_info);
            roomVO.setRoom_price(room_price);
            roomVO.setRoom_people(room_people);
            roomVO.setRoom_img(room_img);
            roomVO.setRoom_no(room_no);

            roomList.add(roomVO);
        }
        rs.close();
        conn.close();
        pstmt.close();
    } catch (Exception e) {
        System.out.println("사업자 방목록 조회중 에러 : " + e.getMessage());
    }
    return roomList;
}
```

받은 사업자 id로 해당 사업자의 방 목록을 조회 후 포워딩

해당 업체의 방 목록 표시

서비스 개요 - 숙박업체 정보

STAYING

로그인



웨스턴 플러스 호텔

전주의 주요 관광지와의 접근성이 뛰어나 편안하고 즐거운 휴식을 누릴 수 있는 호텔입니다

갈



객실 상세 위치 후기



해당 업체의 지도 위치 표시

메뉴의 '위치' 클릭시 업체이름, 위도, 경도를 show_map 함수에 담아 실행

```
function place_click(stay_name, lat, lng) {  
    $('#click_place').show().siblings('div').hide();  
    $('#menu > div').css("font-weight", "normal");  
    $('#place').css("font-weight", "bold");  
    show_map(stay_name, lat, lng);  
}
```

받아온 위도 경로와 업체 이름을 카카오맵 API를 이용해 출력

```
function show_map(stay_name, lat, lng) {  
    var mapContainer = document.getElementById('map');  
    mapOption = {  
        center: new kakao.maps.LatLng(lat, lng),  
        level: 3  
    };  
  
    var map = new kakao.maps.Map(mapContainer, mapOption);  
  
    var position = new kakao.maps.LatLng(lat, lng);  
  
    var marker = new kakao.maps.Marker({  
        map: map,  
        position: position  
    });  
  
    marker.setMap(map);  
  
    var geocoder = new kakao.maps.services.Geocoder();  
  
    var overlay = new kakao.maps.CustomOverlay({  
        content: "",  
        map: map,  
        position: marker.getPosition()  
    });  
}
```

서비스 개요 - 숙박업체 정보



루프탑 수영장 그리고 특별한 살롱 한옥마을의 정취를 품은 전주의 호텔입니다 모던한 인테리어를 입힌 아늑한 객실과 이색적인 한옥마을 전망을 자랑합니다



객실 상세 위치 후기

babb1234

★★★★★ 9.0

2022. 12. 5.

객실도 깨끗하고 시설도 좋았습니다.



사장님 답글 : 이용 감사합니다.

해당 업체의 리뷰와 리뷰 답글 출력

해당 업체와 동일한 업체id인 리뷰를 작성일자 순으로 불러옴

```
// 해당 host의 리뷰 목록을 가져오는 메서드
public List<ReviewVO> review_list_host(String host_id) {
    // 리스트 객체생성
    List<ReviewVO> reviewList = new ArrayList();

    try {
        // DB에 연결
        conn=dataFactory.getConnection();
        // 쿼리문 작성 (계정명 SQL문)
        String query="select *"+
            " from reviewtbl WHERE host_id=? START WITH rv_type=0 CONNECT BY PRIOR"+
            " rv_no=rv_parent ORDER SIBLINGS BY rv_no DESC";
        // articleNo와 parentNo가 같은것을 article번호대로 정렬 (리뷰글 뒤에 답글이 들어감)
        System.out.println(query);
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, host_id);
        ResultSet rs=pstmt.executeQuery();
        // 데이터 목록의 첫번째 부터 가져옴
        while(rs.next()){
            int rv_no =rs.getInt("rv_no");
            int rv_parent =rs.getInt("rv_parent");

            int book_no =rs.getInt("book_no");
            int rv_type =rs.getInt("rv_type");

            String id = rs.getString("id");
            String rv_content = rs.getString("rv_content");
            String rv_img = rs.getString("rv_img");
            Date rv_writeDate = rs.getDate("rv_writeDate");
            int rv_rate = rs.getInt("rv_rate");

            ReviewVO review = new ReviewVO(); // 리뷰 정보를

            review.setRv_no(rv_no);
            review.setRv_parent(rv_parent);
            review.setHost_id(host_id);
            review.setBook_no(book_no);
            review.setRv_type(rv_type);
            review.setId(id);
            review.setRv_content(rv_content);
            review.setRv_img(rv_img);
            review.setRv_writeDate(rv_writeDate);

            // 위에서 생성한 ArticleVO 객체를 리스트에 담음
            reviewList.add(review);
        }
        rs.close();
        pstmt.close();
        conn.close();
    }catch(Exception e) {
        System.out.println("글 목록 조회중 에러 : "+e.getMessage());
    }
    return reviewList;
}
```

rv_type 이 0인 경우 리뷰 글
rv_type 이 1인 경우 리뷰 답글
리뷰 번호 = 부모글 번호 인 글(리뷰 글)
뒤에 해당 글과 동일한 부모 번호를 가진 글(답글)을
정렬해 리뷰 글 뒤에 답글이 오게 한다.

서비스 개요 – 숙박업체 정보

해당 업체의 리뷰와 리뷰 답글 출력 pt.2

컨트롤러에서 포워딩한 리뷰 정보를
Stay_list.jsp에서 rv_type으로 리뷰 글, 게시글 을 구분
각각 다른 형태로 출력해준다. (rv_type이 0일 경우 리뷰 글, 1일 경우 답글)

```
<%-- 리뷰 --%>
<div id="click_review">
  <c:if test="${empty review_list}">
    <div class="reviews">
      <p class="review_info"><span></span></p>
      <p>
        <div class="review_date">
          <span></span>
          <div class="review_content">
            작성된 리뷰가 없습니다.
          </div>
        </div>
      </p>
    </div>
  </c:if>
  <c:forEach var="review" items="${review_list}">
    <c:choose>
      <c:when test="${review.rv_type eq 0}">
        <div class="reviews">
          <p class="review_info"><span>${review.id}</span></p>
          <p>
            <p class="view_star">${review.rv_rate/10}</p>
            <div class="review_date">
              <span><fmt:formatDate value="${review.rv_writeDate}" /></span>
            <div class="review_content">
              ${review.rv_content}
              <c:if test="${!empty review.rv_img}">
                
              </c:if>
            </div>
          </p>
        </div>
      </c:when>
      <c:when test="${review.rv_type eq 1}">
        <div class="reviews">
          <div class="host_comments">사장님 답글 : </span><span>${review.rv_content}</span></div>
          <hr>
        </div>
      </c:when>
    </c:choose>
  </c:forEach>
</div>
<%-- // 리뷰 --%>
```



루프탑 수영장 그리고 특별한 살롱 한옥마을의 정취를 품은 전주의 호텔입니다 모던한 인테리어를 입힌 아늑한 객실과 이색적인 한옥마을 전망을 자랑합니다



객실 상세 위치 후기

babb1234 ★★★★★ 9.0

2022. 12. 5. 객실도 깨끗하고 시설도 좋았습니다.



사장님 답글 : 이용 감사합니다.

서비스 개요 - 숙박업체 정보

STAYING

로그인



웨스턴 플러스 호텔

전주의 주요 관광지와의 접근성이 뛰어나 편안하고 즐거운 휴식을 누릴 수 있는 호텔입니다.



객실 상세 위치 후기



스탠다드 트윈

투숙 가능 인원 2명

편의시설 >

가격

111900원 / 1박

예약하기



디럭스 패밀리 프리미어

예약 페이지 링크

jsp에서 클릭시
업체의 업체 id와 해당 방의 방 번호를 담아서 예약 컨트롤러로 전달한다.

```
<p id="pay_btn2" class="card-text">
  <button type="button" class="btn btn-outline-primary btn-sm"
    onclick="Location.href='${contextPath}/booking/booking_page.do?host_id=${hostInfo.host_id}
    &room_no=${room.room_no}' ">예약하기</button>
  <!-- 해당 업체의 업체id와 방 번호를 담아 예약 컨트롤러로 전달 한다. -->
</p>
```

세션에서 로그인 정보를 확인.
사업자 계정이거나 비 로그인시 로그인페이지로 보낸다.

```
}else if(action.equals("/booking_page.do")) {
  // 예약 확인페이지 요청

  HttpSession session = request.getSession(); // 세션 생성 또는 가져옴
  String logged_in = (String) session.getAttribute("isLogon");
  String user_type = (String) session.getAttribute("user_kind");

  if(logged_in != null) { // 로그인 되어있는 경우
    if(user_type.equals("user")) { // 유저로 로그인되어있는 경우

      //..... 페이지로 넘길 내용
      //.....

    }else {
      // 사업자일 경우 로그아웃 시키고 유저로 로그인 시킴
      session.removeAttribute("log_id");
      session.removeAttribute("isLogon");
      session.removeAttribute("user_kind");

      request.setAttribute("msg", "유저 전용 서비스입니다.");

      nextPage = "/login.jsp";
    }
  } else { // 로그인이 필요한 경우
    request.setAttribute("msg", "로그인이 필요한 서비스입니다.");
    nextPage = "/login.jsp";
  }
}
```


서비스 개요 - 예약 페이지

STAYING

마이페이지로그인

예약자 정보

이름

김 백종은

전화번호


01012341234

결제 정보

☒ 카카오페이 ☐ 신용카드

☐ 약관동의

결제하기



웨스턴 플러스 호텔
스탠다드 트윈

체크인
12월 07일 15시

체크아웃
12월 09일 12시

2박 223800원

STAYING

©2022 사이트이름. All Rights Reserved.
주소: 서울특별시 종로구 어딘가. 사업자등록번호: 777-88-99999.
대표이사: 누군가

고객센터
Tel. 000-111-2222
E-mail staying@staying.co.kr

예약 페이지 링크

세션에서 로그인된 유저의 id를 가져와 정보를 조회 후 포워딩

```
String id = (String) session.getAttribute("log_id");  
MemberService memberService = new MemberService();  
MemberVO memberVO = memberService.infoMember(id);
```

총 숙박 요금

체크인 날짜와 체크아웃 날짜 사이의 숙박일수를 구해
총 숙박일의 숙박비를 계산후 포워딩

```
int cki_date = Integer.parseInt(String.valueOf(session.getAttribute("cki_date"))); //체크인 날짜  
int cko_date = Integer.parseInt(String.valueOf(session.getAttribute("cko_date"))); // 체크아웃 날짜  
  
int room_no = Integer.parseInt(request.getParameter("room_no")); // 방 번호를 가져옴  
RoomService roomService = new RoomService();  
RoomVO roomVO = roomService.room_no_info(room_no); // 방 번호로 방 정보를 불러옴  
  
// 두 날짜간 간격을 통해 숙박일수를 구함  
int day=0;  
for (int i = cki_date; i < cko_date; i++) {  
    day++;  
}  
// 1박 가격에 숙박일수를 곱해 총 숙박비를 계산  
int full_price = (roomVO.getRoom_price()*day);
```

서비스 개요 - 예약 페이지

STAYING

마이페이지로그인

예약자 정보

이름

오 백종은

전화번호


01012341234

결제 정보

☒ 카카오페이 ☐ 신용카드

☐ 약관동의

결제하기



웨스턴 플러스 호텔

스탠다드 트윈

체크인

12월 07일 15시

체크아웃

12월 09일 12시

2박 223800원

STAYING

©2022 사이트이름. All Rights Reserved.

주소: 서울특별시 종로구 어딘가. 사업자등록번호: 777-88-99999.

대표이사: 누군가

고객센터

Tel. 000-111-2222

E-mail staying@staying.co.kr

예약, 결제

아임포트 API를 사용, 결제 화면 출력
결제 실패시 에러코드와 에러메세지를 전달

```
// 결제 수행 함수
function payment(data) {
    let uid = new Date().getTime() + Math.random() + 'a';
    IMP.init('.....'); //아임포트 가맹점 식별코드
    IMP.request_pay({ // param
        pg: "kakaopay.TC00NETIME", //pg사명
        pay_method: "card", //지불 방법
        merchant_uid: uid,
        name: host_ro_name, // 상품명, 업체이름을 가져옴
        amount: full_price, // 결제 금액, 총 숙박금액을 가져옴
        buyer_name: user_name,
        buyer_tel: user_phone
    }, function (rsp) { // callback
        if (rsp.success) {
            alert("결제 성공");
            document.book_hidden.submit();
        } else {
            alert("실패 : 코드(" + rsp.error_code + ") / 메세지(" + rsp.error_msg + ")");
        }
    });
}
```

결제 성공시 book_hidden form을 submit
포워딩 받은 예약관련 정보를 컨트롤러로 전달

```
<!-- 예약 정보를 보내기 위한 hidden form -->
<form name="book_hidden" method="post" action="{contextPath}/booking/booking_room.do">
    <input type="hidden" value="{user_info.id}" name="log_id">
    <input type="hidden" value="{host_info.host_ro_img}" name="host_ro_img">
    <input type="hidden" value="{host_info.host_ro_name}" name="host_ro_name">

    <input type="hidden" value="{room_info.room_name}" name="room_name">
    <input type="hidden" value="{room_info.room_no}" name="room_no">
    <input type="hidden" value="{cki_date_ori}" name="check_in">
    <input type="hidden" value="{cko_date_ori}" name="check_out">
    <input type="submit" value="테스트 버튼">
</form>
```

서비스 개요 - 예약 페이지

마이페이지 로그인



웨스턴 플러스 호텔
스탠다드 트윈

체크인
12월 07일 15시

체크아웃
12월 09일 12시

2박 223800원

고객센터
Tel. 000-111-2222
E-mail staying@staying.co.kr

예약 등록

결제 성공시 book_hidden form에서 보낸 예약관련 정보를 컨트롤러에서 받음

```
}else if(action.equals("/booking_room.do")) {  
    // 예약 생성  
  
    HttpSession session = request.getSession(); // 세션 생성 또는 가져옴  
  
    String id = (String)session.getAttribute("log_id"); // 세션에서 id를 가져옴  
  
    int room_no = Integer.parseInt(request.getParameter("room_no")); // 방번호  
    String room_name = request.getParameter("room_name"); // 방 이름  
    String host_ro_img = request.getParameter("host_ro_img"); // 업체 이미지파일명  
    String host_ro_name = request.getParameter("host_ro_name"); // 업체 이름  
  
    // 체크인 아웃 날짜를 yyyyMMdd형태 int로 저장  
    int checkInDate = Integer.parseInt(request.getParameter("check_in"));  
    int checkOutDate = Integer.parseInt(request.getParameter("check_out"));  
  
    bookingService.add_book(checkInDate, checkOutDate, id, room_no, room_name, host_ro_img, host_ro_name);  
  
    request.setAttribute("msg", "예약이 완료되었습니다.");  
    nextPage="/booking/member_book_list.do";  
}
```

컨트롤러에서 받은 정보와 예약일자과 등록중 계산한 숙박일 수를 bookingTBL에 저장

```
try {  
    conn= dataFactory.getConnection();  
  
    String query = "INSERT INTO bookingtbl"  
        + "(book_no, id, host_id, room_no, check_in, check_out, book_price, room_name, host_ro_img, host_ro_name, book_state)"  
        + " VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, 'before')";  
  
    pstmt=conn.prepareStatement(query);  
  
    System.out.println(query);  
  
    pstmt.setInt(1, book_no);  
    pstmt.setString(2, id);  
    pstmt.setString(3, host_id);  
    pstmt.setInt(4, room_no);  
    pstmt.setInt(5, cki_date);  
    pstmt.setInt(6, cko_date);  
    pstmt.setInt(7, (room_price*(days-1))); // 1박 가격 * 숙박일을 총 결제금액에 담음  
  
    pstmt.setString(8, room_name);  
    pstmt.setString(9, host_ro_img);  
    pstmt.setString(10, host_ro_name);  
  
    pstmt.executeUpdate();  
  
    pstmt.close();  
    conn.close();  
}catch(Exception e) {  
    System.out.println("예약등록중 에러 : " + e.getMessage());  
}
```

해당 방의 예약된 일자 등록

예약된 숙박 일 수 만큼 반복문으로 방 번호와 예약 날짜(yyyyMMdd)를 같이 reserveTBL에 저장
숙박업체 검색시 해당 일자에 예약이 되어있는 방을 구분 하는데 사용

```
// 예약을 등록하는 메서드  
public void add_book(int cki_date, int cko_date, int room_price, int room_no, String host_id, String id, String room_name, String host_ro_img, String host_ro_name) {  
    int days = 0; // 숙박일자  
    int book_no = getBookNo(); // 예약번호 생성  
    try {  
        conn= dataFactory.getConnection();  
  
        String query = "INSERT INTO reservetbl(room_no, reserve_date) VALUES(?,?)";  
        pstmt = conn.prepareStatement(query);  
        pstmt.setInt(1, room_no);  
  
        // 숙박 날짜 만큼 반복 (1 값에 해당 일자를 올려서 DB에 올림);  
        for (int i = cki_date; i <= cko_date; i++) {  
            days++;  
            pstmt.setInt(2, i);  
            pstmt.executeUpdate();  
        }  
  
        pstmt.close();  
        conn.close();  
    }  
    catch(Exception e) {  
        System.out.println("예약날짜 테이블 등록중 에러 : " + e.getMessage());  
    }  
}
```

reserve	Data
Room_No	Number(100)
reserve_date	number(20)

서비스 개요 - 로그인 페이지

로그인, 회원가입 유형을 선택

일반회원의 경우 /member/memberLogin.do 로 전달
사업자의 경우 / host/hostLogin.do 로 전달

네이버 로그인 API를 사용 - naver_callback.jsp를 호출

네이버 API로 부터 받아온 이메일과 이름을 form에 담은 후
회원가입 추가 정보 입력창으로 전달

```
<body>
<form action="${contextPath}/member/naver_login_req" id="naver_login">
<input type="text" id="email" name="email">
<input type="text" id="name" name="name">
<input type="text" id="phone" name="phone">
</form>

<script type="text/javascript">
var naver_id_login = new naver_id_login("", "http://192.168.4.150:8090/staying/naver_callback.jsp");
// 접근 토큰 값 출력
// alert(naver_id_login.oauthParams.access_token);

// 네이버 사용자 프로필 조회
naver_id_login.get_naver_userprofile("naverSignInCallback()");
// 네이버 사용자 프로필 조회 이후 프로필 정보를 처리할 callback function
function naverSignInCallback() { // 클릭된 이후 사용될 내용
    let email = naver_id_login.getProfileData('email');
    let name = naver_id_login.getProfileData('name');

    $('#email').val(email);
    $('#name').val(name);
    $('#naver_login').submit();
}
</script>

</body>
```

STAYING

로그인

일반 회원

사업자

아이디

ID

비밀번호

Password

로그인

회원가입

네이버로 로그인

서비스 개요 - 로그인 페이지

로그인

컨트롤러로 전달한 id, pwd가 일치하는 계정이 존재하는지 확인

```
// 계정 존재 여부 확인 메서드
public boolean isExisted(MemberVO memberVO) {
    boolean result = false;
    String id = memberVO.getId();
    String pwd = memberVO.getPwd();
    try {
        conn = dataFactory.getConnection();

        String query = "SELECT COUNT(*) FROM member2tbl WHERE 1=1 AND id = ? AND pwd = ?";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, id);
        pstmt.setString(2, pwd);

        ResultSet rs = pstmt.executeQuery();

        rs.next();
        int acc_exist = rs.getInt("count(*)"); // 해당 계정이 존재할 경우 1을 반환
        if(acc_exist == 1) {
            result = true;
        }

        pstmt.close();
        conn.close();
        rs.close();
    } catch (Exception e) {
        System.out.println("로그인 중 예외 : " + e.getMessage());
    }
    return result;
}
```

해당 계정이 존재할 경우 세션에 로그인 여부, 아이디, 유저 유형을 저장
▷ 계정이 존재하지 않을 경우 다시 로그인 페이지로 포워딩

```
if(loggedIn) {
    // id, pwd가 일치할 경우

    HttpSession session = request.getSession(); // 세션 생성 또는 가져옴

    session.setAttribute("isLogon", "true"); // logon 확인 -> true
    session.setAttribute("user_kind", "user");
    session.setAttribute("log_id", id); // 세션에 로그인 저장

    System.out.println((String)session.getAttribute("log_id") + "로 로그인됨");
    nextPage="/host/main_page.do";
} else {
    // id나 pwd가 일치하지 않을 경우
    request.setAttribute("msg", "ID나 패스워드를 다시 확인해주세요.");
    nextPage = "/login.jsp";
}
```

The image shows a web application interface for 'STAYING'. At the top, there is a blue header with the 'STAYING' logo on the left and the word '로그인' (Login) on the right. Below the header, there are two tabs: '일반 회원' (General Member) and '사업자' (Business). The '일반 회원' tab is active. Under this tab, there are two input fields: '아이디' (ID) with a placeholder 'ID' and '비밀번호' (Password) with a placeholder 'Password'. Below these fields is a blue button labeled '로그인' (Login), which is highlighted with a dashed blue box. Below the login button are two other buttons: a grey one labeled '회원가입' (Sign Up) and a green one labeled '네이버로 로그인' (Login with Naver).

서비스 개요 - 회원가입 페이지

STAYING로그인

회원가입

중복확인

 인증번호 발송
 인증번호 확인

☐ 약관동의 (필수)

서비스 이용약관
제 1조 (목적)
본 약관은 서비스(이하 "회사"라 한다)는 홈페이지
에서 제공하는 서비스(이하 "서비스"라 한다)를 제

회원가입

아이디 중복확인

입력한 아이디를 Ajax로 컨트롤러로 전달 후
중복 아이디 존재 여부를 Ajax로 받아온다.

```
//ID 중복 확인 함수
function fn_idCheck() {
    let id=$('#id').val(); // input에 입력한 id값을 가져옴
    if(id==''){
        $('#id_check_result').html("ID를 입력해주세요");
        $('#id_check_result').css('color','red');
        return;
    }
    $.ajax({
        type:"post",
        async:true,
        dataType:"text",
        url:(ctx +"/member/id_check"),
        data:{id:id},
        success: function(data, textStatus){
            if(data=="usable"){
                $('#id_check_result').html("사용 가능한 ID입니다.");
                $('#id_check_result').css('color','black');
            }else{
                $('#id_check_result').html("이미 사용중인 ID입니다.");
                $('#id_check_result').css('color','red');
            }
        }
    })
}
```

받아온 입력 ID가 현재 DB에 가입된 ID에 존재하는지 확인후 결과를 리턴

```
// id존재여부를 확인하는 메서드
public boolean idExist(String email) {
    boolean result = false;
    try {
        conn = dataFactory.getConnection();

        String query = "SELECT COUNT(*) FROM member2tbl WHERE 1=1 AND id = ?";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, email);

        ResultSet rs = pstmt.executeQuery();

        rs.next();
        int acc_exist = rs.getInt("count(*)");
        if(acc_exist == 1) {
            result = true;
        }
    }catch (Exception e) {
        System.out.println("ID 존재여부 확인중 에러 : " + e.getMessage());
    }
    return result;
}
```

서비스 개요 - 회원가입 페이지

STAYING로그인

회원가입

중복확인

인증번호 발송
 인증번호 확인

☐ 약관동의 (필수)
서비스 이용약관
제 1조 (목적)
본 약관은 서비스(이하 "회사"라 한다)는 홈페이지
에서 제공하는 서비스(이하 "서비스"라 한다)를 제

회원가입

인증번호 발송

입력한 전화번호를 Ajax로 컨트롤러로 전달

```
function ck_ph(){  
    let phoneNum = $('#phone').val();  
    $.ajax({  
        type:'post',  
        async:false,  
        url:(ctx+"member/phoneCK"),  
        data:{"ph_num":phoneNum},  
        success:function(){  
            Swal.fire('인증번호가 발송되었습니다.');        }  
    })  
}
```

서비스에서 난수 생성 후 해당 난수를 네이버 SENS API로 문자메세지 발송

```
// 랜덤 난수 생성 + 문자 발송 서비스  
public String sendRandomMessage(String tel) {  
    // 문자 발송을 위한 Sms_Ck 객체 생성  
    Sms_Ck message = new Sms_Ck();  
    Random random = new Random();  
  
    String ran_Num = ""; // 난수를 담을 변수 선언 (String)  
    for (int i = 0; i < 5; i++) { // 5자리 생성  
        String ran = Integer.toString(random.nextInt(10)); // 0~9사이의 랜덤난수  
        ran_Num += ran;  
    }  
    System.out.println("랜덤 난수 " + ran_Num);  
  
    message.send_msg(tel, ran_Num); // 해당 번호와 난수로 메시지 전달  
  
    return ran_Num; // 생성한 난수를 컨트롤러로 반환  
}  
  
서비스에서 리턴한 난수를 컨트롤러에서 세션에 저장  
  
}else if(action.equals("/phoneCK")) {  
    // 인증번호 요청 발송시  
    String ph_Num = request.getParameter("ph_num"); // 전화번호를 받아옴  
    MemberService = new MemberService();  
    String ran_Num = memberService.sendRandomMessage(ph_Num); // 문자 발송 + 난수 생성 요청  
  
    HttpSession session = request.getSession(); // 세션 생성  
  
    session.setAttribute("ran_Num", ran_Num); // 난수를 세션에 저장  
    return;  
}
```

서비스 개요 - 회원가입 페이지

STAYING

로그인

회원가입

이름

아이디

영문/숫자 8~20 비밀번호

비밀번호 확인

전화번호

인증번호

중복확인

인증번호 발송

인증번호 확인

☐ 약관동의 (필수)

서비스 이용약관
제 1조 (목적)
본 약관은 서비스(이하 "회사"라 한다)는 홈페이지
에서 제공하는 서비스(이하 "서비스"라 한다)를 제

회원가입

인증번호 확인

입력한 인증번호를 Ajax로 컨트롤러로 전달

```
// 인증번호 확인 함수
function ck_num(){
    let ck_num = $("#input_check_code").val();
    console.log(ck_num);
    $.ajax({
        type: 'post',
        async: false,
        url: (ctx + "member/ck_num"),
        dataType: 'text',
        data: {"ck_num": ck_num},
        success: function(data, textStatus){
            console.log(data);

            if(data === 'correct'){
                Swal.fire('인증번호가 일치합니다.');
```

인증번호 일치시

- ▶ 회원가입 버튼 활성화
- ▶ 인증번호 버튼 비활성화

세션에 저장된 인증번호와 입력한 인증번호 일치 여부를 확인후 결과를 반환

```
}else if(action.equals("/ck_num")) {
    // 인증번호 확인을 (ajax로 인증번호를 보냄)
    String ck_num = request.getParameter("ck_num");

    HttpSession session = request.getSession(); // 세션에서 난수를 가져옴
    String ran_Num = (String) session.getAttribute("ran_Num"); // 입력한 인증번호와 난수의 일치 여부를 확인

    PrintWriter out = response.getWriter();

    if(ran_Num.equals(ck_num)) { // 인증번호가 일치할 경우
        out.print("correct");
    }else { // 인증번호가 일치하지 않을 경우
        out.print("non_correct");
    }
    return;
}
```


서비스 개요 - 회원가입 페이지

STAYING로그인

회원가입

중복확인

 인증번호 발송
 인증번호 확인
☐ 약관동의 (필수)

서비스 이용약관
제 1조 (목적)
본 약관은 서비스(이하 "회사"라 한다)는 홈페이지
에서 제공하는 서비스(이하 "서비스"라 한다)를 제

회원가입

회원가입

입력한 form의 정보를 컨트롤러로 전달
(사업자 회원의 경우 해당 정보를 담아 사업체 정보 입력창으로 전달)

```
}else if (action.equals("/addMember.do")){ // 신규 회원가입 요청시
    String id = request.getParameter("id");
    String pwd = request.getParameter("pwd");
    String name = request.getParameter("name");
    String phone = request.getParameter("phone");

    MemberVO memberVO = new MemberVO(id,pwd,phone, name);

    memberService= new MemberService();
    memberService.addMember(memberVO); // memberService에 유저 등록 요청

    request.setAttribute("msg", "회원가입이 완료되었습니다.");
    nextPage="/index.jsp";
}
```

해당 유저의 정보로 DB에 새 유저정보 추가

```
// 유저 등록 메서드
public void addMember(MemberVO memberVO) {
    try {
        conn=dataFactory.getConnection();

        String id = memberVO.getId();
        String pwd = memberVO.getPwd();
        String phone = memberVO.getPhone();
        String name = memberVO.getName();

        // 쿼리문 작성
        String query = "INSERT INTO Member2tbl (id, pwd, phone, name) VALUES(?,?,?,?)";

        System.out.println(query); // 쿼리문 확인

        pstmt=conn.prepareStatement(query);

        pstmt.setString(1, id);
        pstmt.setString(2, pwd);
        pstmt.setString(3, phone);
        pstmt.setString(4, name);

        pstmt.executeUpdate(); // 쿼리문 실행

        pstmt.close();
        conn.close();

    }catch(Exception e) {
        System.out.println("유저 등록중 예외 : "+e.getMessage());
    }
}
```

Member(유저)	Data
id (PK)	varchar(50)
pwd	varchar(20)
phone	varchar(11)
name	varchar(40)
join_date	date(sysdate)

서비스 개요 - 사업자 가입 추가정보 입력창

STAYING

로그인

주소 검색

카카오 주소 API를 활용 주소를 검색, 검색한 주소를 form에 입력
▷Hidden form 으로 위도와 경도도 함께 입력

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

            // 도로명 주소의 노출 규칙에 따라 주소를 표시한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var roadAddr = data.roadAddress; // 도로명 주소 변수
            var extraRoadAddr = ''; // 참고 항목 변수

            // 법정동명이 있을 경우 추가한다. (법정리는 제외)
            // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
            if(data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                extraRoadAddr += data.bname;
            }
            // 건물명이 있고, 공동주택일 경우 추가한다.
            if(data.buildingName !== '' && data.apartment === 'Y'){
                extraRoadAddr += (extraRoadAddr !== '' ? ', ' + data.buildingName : data.buildingName);
            }
            // 표시할 참고항목이 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
            if(extraRoadAddr !== ''){
                extraRoadAddr = ' (' + extraRoadAddr + ')';
            }

            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById('postcode').value = data.zonecode;
            document.getElementById('roadAddress').value = roadAddr;

            var geocoder = new kakao.maps.services.Geocoder();

            // 주소로 좌표를 검색합니다
            geocoder.addressSearch(roadAddr, function(result, status) {

                // 정상적으로 검색이 완료됐으면
                if (status === kakao.maps.services.Status.OK) {

                    document.getElementById("latitude").value = result[0].y;
                    document.getElementById("longitude").value = result[0].x;

                }
            });
        }
    }).open();
}
```

사업장 등록

시설 명

시설 주소

우편번호

우편번호 찾기

도로명주소

상세주소

시설 유형 ☐ 펜션 ☐ 호텔 ☐ 캠핑장 ☐ 한옥

시설 설명

대표 이미지

파일 선택

선택된 파일 없음

등록하기

서비스 개요 – 사업자 가입 추가정보 입력창

사업자 등록

Form의 내용과 업로드한 이미지를 컨트롤러로 전달
이미지 파일 업로드 메서드로 업로드한 파일을 temp폴더에 임시 저장,
나머지 파라미터 값들도 Map에 저장

```
// 이미지 파일 업로드 메서드
private Map<String, String> upload(HttpServletRequest request, HttpServletResponse response) {

    Map<String, String> articleMap= new HashMap<String, String>();
    String encoding="utf-8"; // 파일명 인코딩 지정
    File currentDirPath=new File(ART_IMAGE_REPO); //이미지 경로 지정 (미리 지정한 경로 변수)
    DiskFileItemFactory factory = new DiskFileItemFactory();
    factory.setRepository(currentDirPath); // 업로드할 c:\temp를 설정
    factory.setSizeThreshold(1024*1024);
    ServletFileUpload upload = new ServletFileUpload(factory); // factory에 있는 파일을 업로드
    try {
        List items=upload.parseRequest(request);
        for(int i=0; i<items.size(); i++) {
            FileItem fileItem=(FileItem)items.get(i);
            if(fileItem.isFormField()) { // Form에서 온 데이터인지 확인
                // 파일 이름을 지정했던 인코딩 방식으로 출력해줌
                // 파일 업로드와 같이 전송된 새 글 관련 (제목, 내용) 매개변수를 저장한 후 반환
                System.out.println((fileItem.getString()+ " = "+fileItem.getString(encoding)));

                // 이미지 이름을 DB 필드에 저장하기 위해 철자에 저장 (articleMap에 담음)
                articleMap.put(fileItem.getFieldName(), fileItem.getString(encoding));
            }else {
                System.out.println("파라미터 이름 : "+ fileItem.getFieldName());
                System.out.println("파일 이름 : "+ fileItem.getName());
                System.out.println("파일 크기 : "+ fileItem.getSize()+"bytes");
                if(fileItem.getSize()>0) { // 파일 사이즈가 0이 아닐경우
                    // 마지막 \ 뒤의 내용 (이미지의 이름)의 위치 [없을경우 -1을 반환]
                    int idx = fileItem.getName().lastIndexOf("\\");
                    if(idx == -1) { // 해당 내용을 찾지 못했을 경우
                        // 혹시 \이 아닌 /로 경로 구분을 했는지 확인
                        idx=fileItem.getName().lastIndexOf("/");
                    }
                    // \가 끊긴 다음부터 String에 파일이름
                    String fileName = fileItem.getName().substring(idx+1);
                    articleMap.put(fileItem.getFieldName(), fileName);

                    // 업로드한 이미지를 일단 temp에 저장
                    File uploadFile= new File(currentDirPath + "\\temp\\" + fileName);
                    fileItem.write(uploadFile);
                }
            }
        }
    } catch (Exception e) {
        System.out.println("파일 업로드중 에러 : " + e.getMessage());
    }
    return articleMap;
}
```

임시 저장된 이미지를 사업자 id 폴더로 이동

```
if(ro_img != null && ro_img.length() != 0) {
    // temp폴더에 임시로 업로드된 파일객체를 생성
    File srcFile = new File(ART_IMAGE_REPO + "\\temp\\" + ro_img);
    // 글 번호로 폴더를 생성
    File destDir = new File(ART_IMAGE_REPO+"\"+host_id);
    destDir.mkdir(); // 해당 경로의 폴더생성
    // temp폴더의 파일을 글번호 폴더로 이동
    FileUtils.moveFileToDirectory(srcFile, destDir, true);
    srcFile.delete(); // temp의 임시 파일을 삭제함
}
```

전달받은 값들과 이미지 파일명을 테이블에 저장

```
// 사업자 회원 추가 메서드
public void addHost(HostVO hostVO) {
    try {
        conn=dataFactory.getConnection();

        String host_id = hostVO.getHost_id();
        String host_pwd = hostVO.getHost_pwd();
        String host_phone = hostVO.getHost_phone();
        String host_name = hostVO.getHost_name();

        String host_ro_name = hostVO.getHost_ro_name();
        String host_ro_addr = hostVO.getHost_ro_addr();
        String host_ro_kind = hostVO.getHost_ro_kind();
        String host_ro_latitude = hostVO.getHost_ro_latitude();
        String host_ro_longitude = hostVO.getHost_ro_longitude();
        String host_ro_content = hostVO.getHost_ro_content();
        String host_ro_img = hostVO.getHost_ro_img();

        String query = "INSERT INTO hosttbl(host_id, host_pwd, host_phone, host_name,"
            + " host_ro_name, host_ro_addr, host_ro_kind, host_ro_latitude,"
            + " host_ro_longitude, host_ro_content, host_ro_img, host_ro_rate)"
            + " VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        System.out.println(query); // 쿼리문 확인

        pstmt=conn.prepareStatement(query);

        pstmt.setString(1, host_id);
        pstmt.setString(2, host_pwd);
        pstmt.setString(3, host_phone);
        pstmt.setString(4, host_name);

        pstmt.setString(5, host_ro_name);
        pstmt.setString(6, host_ro_addr);
        pstmt.setString(7, host_ro_kind);
        pstmt.setString(8, host_ro_latitude);
        pstmt.setString(9, host_ro_longitude);
        pstmt.setString(10, host_ro_content);
        pstmt.setString(11, host_ro_img);

        pstmt.executeUpdate(); // 쿼리문 실행

        pstmt.close();
        conn.close();
    } catch (Exception e) {
        System.out.println("사업자 회원 추가중 에러 : " + e.getMessage());
    }
}
```

로그인

사업장 등록

우편번호

우편번호 찾기

도로명주소

상세주소

☐ 펜션 ☐ 호텔 ☐ 캠핑장 ☐ 한옥

파일 선택 선택된 파일 없음

등록하기

서비스 개요 - 마이페이지

STAYING

마이페이지로그아웃

마이 페이지

내 정보 관리

예약 관리

내 정보 관리

이름

백종은

아이디

babb1234

비밀번호

영문/숫자 8~20 비밀번호

비밀번호 확인

비밀번호 재입력

전화번호

01012341234

비밀번호 변경

유저 정보

세션에서 로그인된 id를 확인, 해당id로 유저정보를 가져옴

```
}else if(action.equals("/member_info.do")) {  
    // 유저 정보 요청 (내 정보 보기)  
  
    HttpSession session = request.getSession(); // 세션 생성 또는 가져옴  
    String id = (String)session.getAttribute("log_id"); // 로그인 된 id를 가져옴  
    System.out.println(id);  
  
    if(id==null || id.equals("")) { // 로그인 되어있지 않을 경우  
        request.setAttribute("msg", "로그인이 필요한 서비스입니다.");  
        nextPage = "/login.jsp"; // 로그인 페이지로 이동  
    }else {  
  
        MemberVO memberVO = new MemberVO();  
  
        memberService = new MemberService();  
        //해당 id의 유저정보를 가져옴  
        memberVO = memberService.infoMember(id);  
        System.out.println(memberVO.getId());  
        System.out.println(memberVO.getPwd());  
  
        request.setAttribute("memberInfo", memberVO);  
        nextPage="/member_info.jsp";  
    }  
}
```

비밀번호 변경

DB에서 해당 유저의 pwd를 수정

```
public void update_pwd(MemberVO memberVO) {  
    try {  
        String id = memberVO.getId();  
        String pwd = memberVO.getPwd();  
  
        conn=dataFactory.getConnection();  
        String query = "update member2tbl set pwd=? where id = ?";  
        System.out.println(query); // 쿼리문 확인  
        pstmt=conn.prepareStatement(query);  
  
        // 쿼리문의 ?에 위에서 받은 값을 담음  
  
        // where 조건에 담을 id값  
        pstmt.setString(1, pwd);  
        pstmt.setString(2, id);  
  
        pstmt.executeUpdate(); // 업데이트문을 수행해줌  
        pstmt.close();  
        conn.close();  
        // 수행만 해주는 경우에는 close 필요 X  
    }catch(Exception e) {  
        System.out.println("비밀번호 변경중 에러"+e.getMessage());  
    }  
}
```


서비스 개요 - 내 예약 관리



유저 예약목록

세션에서 로그인된 id를 확인, 해당id의 예약정보를 가져옴

```
}else if(action.equals("/member_book_list.do")) {  
    // 마이페이지 예약한 내역 불러오기  
    HttpSession session = request.getSession(); // 세션 생성 또는 가져옴  
    String log_id = (String) session.getAttribute("log_id"); // 세션에서 로그인한 아이디를 가져옴  
  
    List<BookingVO> book_list = bookingService.get_member_book(log_id); // 해당 id의 예약정보를 불러옴  
  
    request.setAttribute("book_list", book_list); // 예약 리스트를 포워딩  
    nextPage= "/member_book.jsp";  
}
```

리뷰 작성

리뷰 작성이 이미 된 글인경우경고문을 출력

```
<c:if test="${book.book_state eq 'before'}"> <!-- 리뷰작성 전의 글인 경우 -->  
<div class="review_cancel_area">  
    <a type="button" onclick="fn_review_active(${book.host_id}, ${book.book_no}, ${status.index})">리뷰작성</a>  
    <a href="${contextPath}/booking/del_book.do?book_no=${book.book_no}&cki=${book.check_in}&cko=${book.check_o}</div>  
</c:if>  
<c:if test="${book.book_state eq 'after'}"> <!-- 리뷰 작성이 이미 된 글인 경우 -->  
<div class="review_cancel_area">  
    <a type="button" onclick="fn_alert_review()">리뷰작성</a>  
    <a href="${contextPath}/booking/del_book.do?book_no=${book.book_no}&cki=${book.check_in}&cko=${book.check_o}</div>  
</c:if>
```

리뷰 작성 이전인 글의 경우 해당 예약의 체크아웃 일자를 확인, 현재 날짜 이전이면 리뷰작성창을 출력, 현재 날짜 이후면 경고문을 출력


```
function fn_review_active(host_id,book_no,stat) {  
  
    document.getElementById("host_id_hidden").value = host_id;  
    document.getElementById("book_no_hidden").value = book_no;  
  
    let set_id = "check_out_hidden"+stat  
  
    let check_out = document.getElementById(set_id).value; // 체크아웃 일자를 가져옴 (yyyyMMdd 형태 number값)  
    let year = check_out.substring(0, 4);  
    let month = check_out.substring(4, 6);  
    let day = check_out.substring(6, 8); // 년, 월, 일로 구분  
  
    let check_out_date = new Date(year, month-1, day); // 위에서 구분한 값을 Date값으로 변환  
    let now = new Date();  
    // 현재 일자를 저장  
    if(check_out_date > now) {  
        // 체크아웃 전 일 경우  
        $('#exampleModal').modal("hide");  
        alert("아직 체크아웃 전입니다");  
    }else {  
        // 체크아웃 후 일 경우  
        $('#exampleModal').modal("show");  
    }  
}
```

서비스 개요 - 내 예약 관리


STAYING마이페이지로그아웃

마이 페이지
내 정보 관리
예약 관리

예약 내역 조회



전주 호텔원
스탠다드 더블
체크인 11월 17일
체크아웃 11월 18일
리뷰작성 취소하기



전주 라한호텔
디럭스 트윈
체크인 11월 23일
체크아웃 11월 25일
리뷰작성 취소하기

예약 취소

컨트롤러로 해당 예약 번호를 전달, 해당 예약번호의 예약 정보를 삭제

```
// 예약 삭제 메서드
public void del_book(int book_no) {
    try {
        conn=dataFactory.getConnection();

        // 쿼리문 (해당 예약번호의 예약을 삭제)
        String query = "DELETE FROM bookingtbl WHERE book_no=?";
        System.out.println(query); // 쿼리문 확인
        pstmt=conn.prepareStatement(query);

        pstmt.setInt(1, book_no); // ?에 받아온 id를 담음
        pstmt.executeUpdate(); // 쿼리문 실행

        pstmt.close();
        conn.close();

    }catch(Exception e) {
        System.out.println("유저 삭제중 에러 : " + e.getMessage());
    }
}
```

체크인 체크아웃 날짜 사이의
Reserve 테이블에 저장된 예약된 날짜 정보도 전부 삭제

```
// reserve 테이블 삭제 메서드
public void del_reserve(int room_no, int check_in, int check_out) {
    try {
        conn=dataFactory.getConnection();

        // 쿼리문 (해당 기간내의 해당 방 예약을 삭제)
        String query = "DELETE FROM reservetbl WHERE room_no=? AND reserve_date BETWEEN ? and ?";
        System.out.println(query); // 쿼리문 확인
        pstmt=conn.prepareStatement(query);

        pstmt.setInt(1, room_no); // ?에 받아온 id를 담음
        pstmt.setInt(2, check_in);
        pstmt.setInt(3, check_out);
        pstmt.executeUpdate(); // 쿼리문 실행

        pstmt.close();
        conn.close();

    }catch(Exception e) {
        System.out.println("유저 삭제중 에러 : " + e.getMessage());
    }
}
```

서비스 개요 - 내 예약 관리(리뷰 작성 모달)

리뷰 작성

★★★★★

10자 이상 작성해주세요

파일 선택 2.jpg

취소 리뷰등록

리뷰 등록

업로드 한 이미지를 임시폴더(temp)에 저장, 받아온 파라미터 값은 Map에 저장

```
for(int i=0; i<items.size(); i++) {
    FileItem fileItem=(FileItem)items.get(i);
    if(fileItem.isFormField()) { // Form에서 온 데이터인지 확인
        // 파일 이름을 지정했던 인코딩 방식으로 출력해줌
        // 파일 업로드와 같이 전송된 새 글 관련 (제목, 내용) 매개변수를 저장한 후 반환
        System.out.println((fileItem.getString()+" "+fileItem.getString(encoding)));

        // 이미지 이름을 DB 필드에 저장하기 위해 컬럼에 저장 (articleMap에 담음)
        articleMap.put(fileItem.getFieldName(), fileItem.getString(encoding));
    }else {
        System.out.println("파라미터 이름 : "+ fileItem.getFieldName());
        System.out.println("파일 이름 : "+ fileItem.getName());
        System.out.println("파일 크기 : "+ fileItem.getSize()+"bytes");
        if(fileItem.getSize()>0) { // 파일 사이즈가 0이 아닐경우
            // 마지막 \\\ 뒤의 내용 (이미지의 이름)의 위치 [없을경우 -1을 반환]
            int idx = fileItem.getName().lastIndexOf("\\");
            if(idx == -1) { // 해당 내용을 찾지 못했을 경우
                // 혹시 \\이 아닌 /로 경로 구분을 했는지 확인
                idx=fileItem.getName().lastIndexOf("/");
            }
            // \\\가 붙긴 다음부터 String이 파일이름
            String fileName = fileItem.getName().substring(idx+1);
            articleMap.put(fileItem.getFieldName(), fileName);

            // 업로드한 이미지를 일단 temp에 저장
            File uploadFile= new File(currentDirPath + "\\temp\\" + fileName);
            fileItem.write(uploadFile);
        }
    }
}
```

받아온 파라미터 값으로 테이블에 리뷰글을 저장, 리뷰글의 rv_parent값은 0, 답글은 1

```
// 쿼리문 (데이터 삽입)
String query = "insert into reviewTBL(rv_no, rv_parent, host_id, book_no, rv_type, id, rv_content, rv_img, rv_rate) "
+ "values(?,?,?,?,?,?,?,?)";
```

bookingTBL에서 해당 예약번호의 리뷰 작성여부를 업데이트

```
conn=dataFactory.getConnection();
String query = "update bookingtbl set book_state='after' where book_no = ?";
System.out.println(query); // 쿼리문 확인
pstmt=conn.prepareStatement(query);
```

// 쿼리문의 ?에 위에서 받은 값을 담음

```
// where 조건에 담을 book_no값
pstmt.setInt(1, book_no);
```

```
pstmt.executeUpdate(); // 업데이트문을 수행해줌
pstmt.close();
conn.close();
```

서비스 개요 - 내 예약 관리(리뷰 작성 모달)

리뷰 별점

리뷰 등록 후 해당 업체의 리뷰 별점의 평균을 구한다.
(별점이 없는 경우는 제외 (리뷰 답글))

```
// 해당 host의 리뷰 수 별점의 총합을 가져오는 메서드
private int getSumRate(String host_id) {
    int sum = 0;
    try {
        conn=dataFactory.getConnection();
        // articleNo가 가장 큰 값을 가져오는 쿼리문
        String query="SELECT rv_rate FROM reviewTBL WHERE host_id=? AND rv_rate > 1";
        pstmt=conn.prepareStatement(query);
        pstmt.setString(1, host_id);
        ResultSet rs = pstmt.executeQuery();

        while(rs.next()) {
            sum +=rs.getInt("rv_rate"); // 글의 별점 합을 더함
        }

        conn.close();
        pstmt.close();
        rs.close();
    }catch (Exception e) {
        System.out.println("리뷰 카운트 를 가져오는 중 에러 : "+e.getMessage());
    }
    return sum;
}
```

해당 업체의 별점 총합을 구한다

```
int count = getReviewCount(host_id); // 해당 업체의 리뷰 수를 가져온다.
int sum = getSumRate(host_id); // 해당 업체의 리뷰별점 총 합을 구한다
int avgRate = sum/count;
```

계산한 평균 별점으로 해당 숙박 업체의 별점 정보를 업데이트.

```
// host의 평점을 등록하는 메서드
private void updateRate(String host_id,int rate) {
    try {
        conn=dataFactory.getConnection();
        // 해당 사업자의 별점 정보를 업데이트
        String query = "update hosttbl set host_ro_rate=? WHERE host_id=?";
        pstmt=conn.prepareStatement(query);
        pstmt.setInt(1,rate);
        pstmt.setString(2, host_id);
        pstmt.executeUpdate();


        pstmt.close();
        conn.close();
    }catch (Exception e) {
        System.out.println("별점 업데이트중 에러 : "+e.getMessage());
    }
}
```


서비스 개요 - 사업자 업체 관리

STAYING

내 사업장 로그아웃




완주 라르고 펜션



객실관리 리뷰

객실 추가

객실 삭제

객실 사진	객실 타입	최대 투숙 가능 인원	가격
	라벤더	4 명	150000 원
	자스민	4 명	180000 원
	로즈마리	8 명	380000 원

객실 추가 페이지로 이동

```
<button type="button" class="btn btn-outline-primary btn-sm" id="add_click"
onclick="location.href='{contextPath}/room/add_room_form.do' ">객실 추가</button>
```

내 객실 목록

로그인한 사업자의 id를 확인

```
HttpSession session = request.getSession(); // 세션 생성
String host_id = (String)session.getAttribute("log_id"); // 로그인한 사업자의 아이디를 가져옴
```

```
List<RoomVO> roomList = roomService.hostRoomList(host_id); // 로그인된 사업자id의 방 리스트를 가져옴
```

해당 사업자id에 등록된 방 목록을 리스트로 가져옴

```
conn = dataFactory.getConnection();

String query = "SELECT * FROM roomtbl1"
+ " WHERE host_id=?"
+ " ORDER BY room_price ASC"; // 낮은 가격부터 정렬
pstmt=conn.prepareStatement(query);

pstmt.setString(1, host_id);

ResultSet rs = pstmt.executeQuery();

while(rs.next()) {
    String room_name = rs.getString("room_name"); // 방 이름
    String room_info = rs.getString("room_info"); // 방 설명
    int room_price = rs.getInt("room_price"); // 1박 가격
    int room_people = rs.getInt("room_people"); // 수용 인원
    String room_img = rs.getString("room_img"); // 이미지파일명
    int room_no = rs.getInt("room_no"); // 방 번호

    RoomVO roomVO = new RoomVO();

    roomVO.setHost_id(host_id);

    roomVO.setRoom_name(room_name);
    roomVO.setRoom_info(room_info);
    roomVO.setRoom_price(room_price);
    roomVO.setRoom_people(room_people);
    roomVO.setRoom_img(room_img);
    roomVO.setRoom_no(room_no);




    roomList.add(roomVO); // 리스트에 해당 roomVO를 담음
}
```


서비스 개요 - 사업자 업체 관리(객실 삭제)

STAYING내 사업장로그아웃

완주 라르고 펜션

객실관리리뷰

객실 사진	객실 타입	최대 투숙 가능 인원	가격	객실 삭제
	라벤더	4 명	150000 원	<input checked="" type="checkbox"/>
	자스민	4 명	180000 원	<input type="checkbox"/>
	로즈마리	8 명	380000 원	<input type="checkbox"/>

객실 삭제

객실 삭제 클릭시 체크 박스를 출력

체크한 객실의 방 번호를 컨트롤러에서 배열로 받아옴

```
String[] room_no_st = request.getParameterValues("room_no");
```

받아온 리스트의 방 번호와 일치하는 방들을

DAO를 통해 테이블에서 삭제

삭제 후 해당 방의 이미지 폴더도 삭제

```
for(String room_no_str : room_no_st) {  
    int room_no = Integer.parseInt(room_no_str);  
  
    // 글을 삭제하면서 관련 정보 객체들 가져옴  
    roomVO = roomService.delRoom(room_no);  
  
    File imgDir = new File(ART_IMAGE_REPO+"\\\\"+roomVO.getRoom_no());  
    if(imgDir.exists()) { // 해당 이미지 폴더가 존재할 경우  
        FileUtils.deleteDirectory(imgDir); // 해당 폴더를 삭제  
    }  
}  
  
// 방을 지움  
try {  
    conn=dataFactory.getConnection();  
  
    String query="DELETE FROM roomtbl WHERE room_no=?";  
    pstmt = conn.prepareStatement(query);  
    pstmt.setInt(1, room_no);  
  
    pstmt.executeUpdate();  
  
    pstmt.close();  
    conn.close();  
}  
catch (Exception e) {  
    System.out.println("방 삭제중 에러 : " + e.getMessage());  
}
```

서비스 개요 - 사업자 업체 관리(업체 리뷰 확인)

STAYING

내 사업장 로그아웃

전주 라한호텔



객실관리 리뷰

babb1234

★★★★★ 0.0

2022. 12. 5.

객실도 깨끗하고 시설도 좋았습니다.



답글 작성

답글 작성 버튼 클릭시
답글 작성 창 표시

사장님 답글

답글 저장

내 객실 리뷰 확인

내 id로 내 업체에 등록된 리뷰 목록을 불러옴
(업체 정보페이지의 리뷰글 정보와 같은 메서드를 사용)

```
conn=dataFactory.getConnection();
// 쿼리문 작성 (계중형 SQL문)
String query="select *"+
    " from reviewtbl WHERE host_id=? START WITH rv_type=0 CONNECT BY PRIOR"+
    " rv_no=rv_parent ORDER SIBLINGS BY rv_no DESC";
// articleNo와 parentNo가 같은것을 article번호대로 정렬 (리뷰글 뒤에 답글이 들어감)
System.out.println(query); // 쿼리문 확인
pstmt=conn.prepareStatement(query);
pstmt.setString(1, host_id);
ResultSet rs=pstmt.executeQuery();
```

리뷰 답글 저장

해당 리뷰 글의 글번호를 받아서 parent_no로 지정

```
session = request.getSession();
String id = (String) session.getAttribute("log_id");

int parent_no = Integer.parseInt(request.getParameter("rv_no"));
```

rv_type을 1(리뷰글)에 parent_no를 원글 글 번호로
리뷰테이블에 글 추가

```
// 쿼리문 (데이터 삽입)
String query = "insert into reviewTBL(rv_no, rv_parent, host_id, book_no, rv_type, id, rv_content, rv_rate) "
    + "values(?,?,?,?,?,?,?,?)";
System.out.println(query); // 쿼리문 확인

pstmt=conn.prepareStatement(query);

// 쿼리문의 ?에 위에서 달아온 내용을 담음
pstmt.setInt(1, rv_no);
pstmt.setInt(2, rv_parent);
pstmt.setString(3, host_id);
pstmt.setInt(4, book_no);
pstmt.setInt(5, 1); // rv_type은 1 (리뷰글)
pstmt.setString(6, id);
pstmt.setString(7, rv_content);
pstmt.setInt(9, 0); // 별점은 0으로 지정해 평균 별점 등록시 인식X

pstmt.executeUpdate(); // 쿼리문 실행
```

서비스 개요 - 사업자 객실 추가

STAYING

마이페이지 로그인

전주 라한호텔

객실사진

파일선택 선택된 파일 없음

객실명

가격

인원

편의시설

저장하기

객실 추가

업로드 한 이미지를 임시폴더(temp)에 저장, 받아온 파라미터 값은 Map에 저장

받은 파라미터 값으로 roomTBL 테이블에 방 정보를 등록

```

conn = dataFactory.getConnection();

String room_name = roomVO.getRoom_name(); // 방 이름
String room_info = roomVO.getRoom_info(); // 방 편의시설

int room_people = roomVO.getRoom_people(); // 수용인원
String room_img = roomVO.getRoom_img(); // 방 이미지 파일명

String query = "INSERT INTO roomtbl(host_id, room_no, room_name, room_info, room_price, "
+ "room_people, room_img) VALUES(?,?,?,?,?,?,?)";

pstmt=conn.prepareStatement(query);
pstmt.setString(1, host_id);
pstmt.setInt(2, room_no); // 등록할 방 번호 (개별 메서드에서 생성)
pstmt.setString(3, room_name);
pstmt.setString(4, room_info);
pstmt.setInt(5, room_price);
pstmt.setInt(6, room_people);
pstmt.setString(7, room_img);

pstmt.executeUpdate(); // 쿼리문 실행

해당 방의 가격이 현재 업체에 등록된 최저가격보다 저렴한 경우
해당 방의 가격으로 업체의 최저가격을 업데이트 (검색 시에 사용)
if(min_price>room_price) {
    try {
        conn = dataFactory.getConnection();
        // 해당 방의 최저가격을 업데이트
        String query = "update hosttbl set host_ro_lowprice=? where host_id=?";

        pstmt=conn.prepareStatement(query);
        pstmt.setInt(1, room_price);
        pstmt.setString(2, host_id);

        pstmt.executeUpdate();

        pstmt.close();
        conn.close();
    }catch (Exception e) {
        System.out.println("최저가격 등록중 에러 : " + e.getMessage());
    }
}
    
```

Host_Ro_name(명칭)	varchar(40)
Host_Ro_addr(주소)	varchar(100)
Host_Ro_kind(종류)	varchar(20)
Host_Ro_content	varchar(500)
Host_Ro_latitude	varchar(100)
Host_Ro_longitude	varchar(100)
Host_Ro_lowPrice	Number(20)
Host_Ro_img(사진)	varchar(100)
Host_Ro_rate(별점)	number(5)