




**온라인 쇼핑몰**  
**DAILY**



기본정보

-  김 현 수
-  010 6630 8025
-  hyunsoo3466@naver.com

학력

- 학점은행제 시스템제어 전문학사 취득

보유기술

- Java
- JavaScript
- jQuery
- SQL

이력

- ㈜한국플랜트서비스 전기팀  
담당업무 : Exciter 및 자동화 PLC 로직 설비 및 기기 운영  
Tool : Siemens, Ls산전, 미쯔비시  
기간 : 2017.03 ~ 2022.03
- ㈜코암테크 자동화사업팀  
담당업무 : 공조 자동화 PLC 로직 설비 운영 및 국내외 시운전  
HMI 개발 및 유지보수  
Tool : Siemens  
기간 : 2015.12 ~ 2017.02

수료

- 이젠아카데미 자바 안드로이드 웹 & 앱 개발자 과정 수료  
(Java, JS, Spring, MySQL, ReactNative)  
기간 : 2022.05 ~ 2022.12

자격증

- 정보처리기사 (필)
- 설비보전기사
- 발전정비사(전기)3급
- 6축 자동 로봇프로그래밍
- 생산자동화산업기사
- 기계정비산업기사
- 전기기능사
- 공유압기능사
- 특수용접기사
- 운전면허1종 보통

못하는게 아니라 하지 않는 것이다

## 프로젝트명

---

DAILY 웹 쇼핑몰

## 수행기간

---

2022.10 ~ 2022.12

## 개발목표

---

- 화면구성 단계부터 (HTML) Controller, Service, DAO, DTO, Mapper(Query), SQL(DB) 까지 전체 공정의 이해를 바탕으로 사용자가 원하는 데이터의 가공 및 출력을 완성한다.

## 구현 기능

---

1. 회원가입 기능 구현  
(가입 및 변경 시 중복 아이디 검열 기능)
2. 로그인 기능과 로그아웃기능 구현
3. 가입한 회원의 정보 조회 기능
4. 회원 정보 수정에 관한 알람창 및 수정 기능
5. 구매물품 리뷰와 사진첨부 및 별점을 남기는 기능
6. 구매물품 아이템의 목록과 정보확인

## 개발환경

---

Tool : Spring4  
Server : Tomcat 9.0  
DB : MySQL

## 사용언어

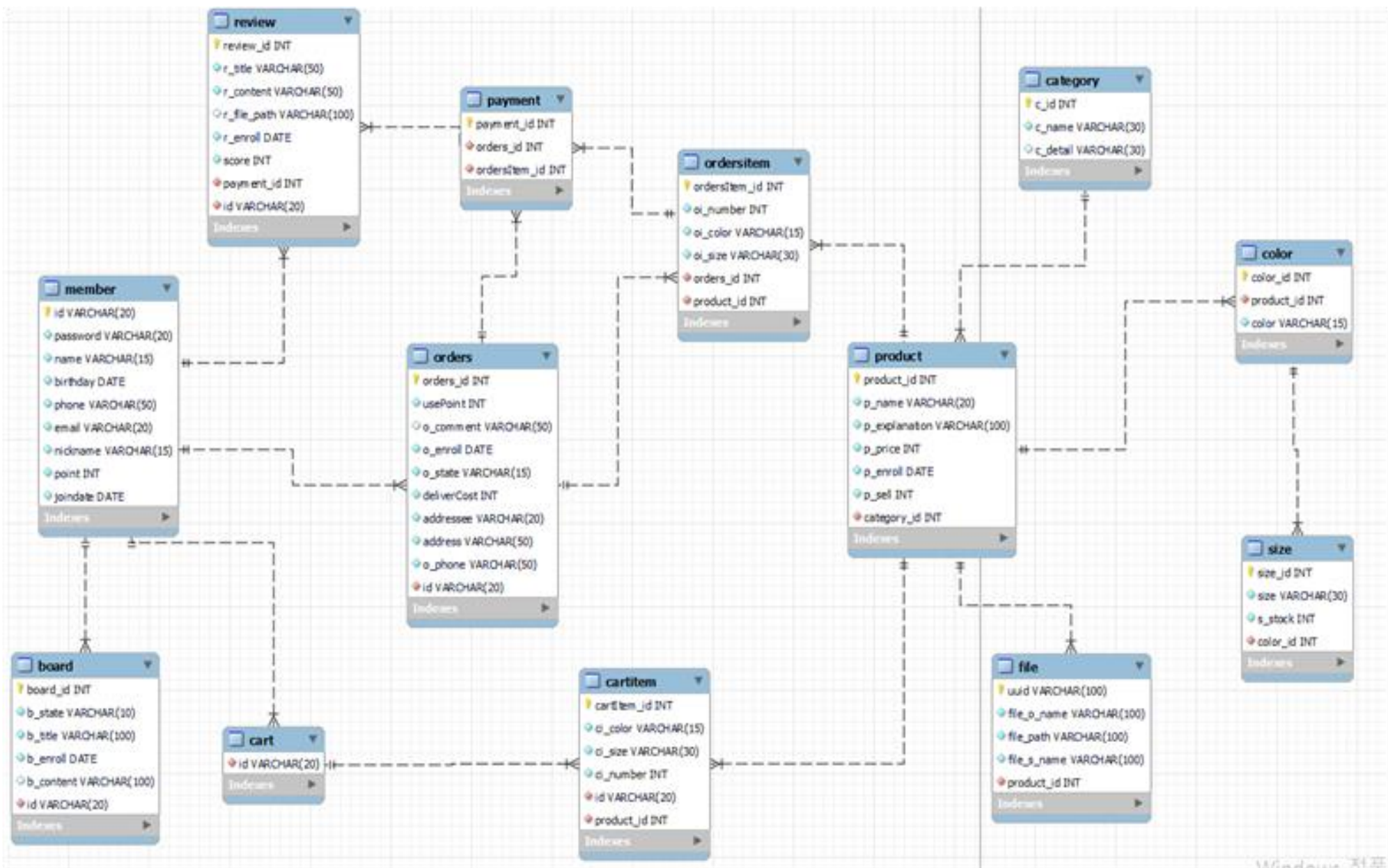
---



담당 기능 및 일정 회의

| 타이틀    | 진행         | 10월 | 11월 |     |     |     | 12월 |
|--------|------------|-----|-----|-----|-----|-----|-----|
|        |            | 4주차 | 1주차 | 2주차 | 3주차 | 4주차 | 1주차 |
| 개념설계   | 기획 선정      |     |     |     |     |     |     |
|        | 시나리오 구성    |     |     |     |     |     |     |
|        | 엔티티 타입 정의  |     |     |     |     |     |     |
|        | 속성 정의      |     |     |     |     |     |     |
| 논리설계   | 정규화 작업     |     |     |     |     |     |     |
|        | 테이블정의      |     |     |     |     |     |     |
| 프로젝트준비 | 개발 환경 설정   |     |     |     |     |     |     |
|        | 공통 HTML 셋팅 |     |     |     |     |     |     |
| 사이트 구성 | HTML 작업    |     |     |     |     |     |     |
|        | 기능 구현      |     |     |     |     |     |     |

# Project ERD 설계



메인 페이지

DAILY

Q & A

New

남성

여성

남녀공용

Q&A



ALPHA

로그인 페이지

DAILY

test1

.....

로그인

아이디 찾기 | 비밀번호 찾기 | 회원가입

Get connected with us on social networks



COMPANY NAME

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

PRODUCTS

Angular  
React

USEFUL LINKS

Pricing  
Settings

CONTACT

New York, NY 10011, US  
info@example.com

## 회원가입

1. 회원가입 클릭 시 다음 화면이 출력
2. DB에 존재하는 아이디와 중복될 경우 중복 알림
3. '비밀번호'와 '비밀번호확인' 이 불일치 시 알림 발생

## 회원가입 page DAILY

test1

중복확인

.....

비밀번호 재확인

이름

생년월일

'-'를 제외하고 입력해 주세요.

이메일

닉네임

취소

회원가입

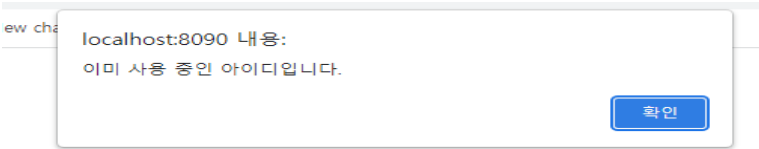
## 회원가입 주요기능1

- 1. 아이디가 중복될 경우 Ajax를 통해 비동기식으로 DB안에 있는 Data들과 즉각적인 피드백을 할 수 있으며 메시지와 함께 안내 문구를 발생시킨다

### 아이디 중복체크 jQuery

```
2  * 회원가입 관련 함수
3  */
4
5  //-----
6  // 아이디 중복체크
7  //-----
8
9  function fn_idCheck() {
10     // alert($("#id").val());
11     $.ajax({
12         url: "/member/idCheck",
13         type: "post",
14         dataType: "json",
15         data: { id: $("#id").val() },
16         success: function (data) {
17             if (data == 1) {
18                 alert("이미 사용 중인 아이디입니다.");
19             } else if (data == 0) {
20                 // 버튼 id="idCheck"의 속성인 value의 값을 "Y"로 변경한다.
21                 $("#idCheck").attr("value", "Y");
22                 alert("사용 가능한 아이디입니다.");
23             }
24         },
25     });
26 }
```

### 중복 알람창



test1

중복확인

.....

비밀번호 재확인

이름

생년월일

'-'를 제외하고 입력해 주세요.

이메일

닉네임

취소

회원가입



## 회원가입 주요기능2

1. 아이디 중복검사 진행 시 DB에 저장 되어있는 값 중에 ID요소를 선택하고 비교한다. 비교값으로는 0과 1을 결국 반환하게 되는데 이것을 jQuery단에서 0과1이라면 각각 그것에 해당하는 결과값이 출력되어 사용자에게 보여지게 된다

### 중복검사 DAO

```
// 아이디 중복 검사 (AJAX)
@Override
public int idCheck(MemberVO memberVO) throws DataAccessException {
    return sqlSession.selectOne(NAMESPACE + ".idCheck", memberVO);
}
```

### 중복검사 Controller

```
// 아이디 중복 검사 (AJAX)
@ResponseBody
@RequestMapping(value = "/idCheck", method = RequestMethod.POST)
public int idCheck(MemberVO memberVO) throws Exception {

    System.out.println("MemberController 아이디 중복 검사 (AJAX) id ==> " + memberVO.getId());

    int result = memberService.idCheck(memberVO);
    System.out.println("result : " + result);
    return result;
}
```

### 중복검사 Service

```
// 아이디 중복 검사 (AJAX)
@Override
public int idCheck(MemberVO memberVO) throws Exception {
    int result = memberDAO.idCheck(memberVO);
    return result;
}
```

## 회원가입 주요기능3

1. 아이디 로그아웃 진행 시  
Get방식으로 세션의 값을 제거하는 요소를 사용하여 세션에 저장된 회원의 정보를 삭제하며 리디렉트를 활용하여 페이지를 새로고침 시킨다. 이 기능에서는 Session에 저장된 정보만 제거하면 되기 때문에 별도의 Service나 DAO, 기타 View는 없고 다시 초기 화면의 메인 페이지로 돌린다.

## 로그아웃 Controller

```
106  
107 // 로그아웃 처리  
108 @RequestMapping(value = "/logout", method = RequestMethod.GET)  
109 public String logout(HttpSession session) {  
110  
111     session.removeAttribute("member");  
112  
113     return "redirect:/";  
114 }
```

## 로그인된 메인 페이지



## 로그아웃된 메인 페이지



로그인

- 1. 로그인 시도시 DB에 저장되어 있는 ID와 Password가 일치 하지 않는다면 알람창이 발생되면서 로그인이 되지 않는다.

로그인 시도

로그인 실패

localhost:8090 내용:

아이디 또는 비밀번호가 일치하지 않습니다.

확인

DAILY

DAILY

test1

.....

로그인

아이디 찾기 | 비밀번호 찾기 | 회원가입

아이디

비밀번호

로그인

아이디 찾기 | 비밀번호 찾기 | 회원가입

## 로그인 주요기능

1. 로그인 시도가 발생되면 DB에 저장되어 있는 유저 정보를 조회하게 되고 ID와 Password를 Service 단에서 비교하여 값이 같지 않다면 null값을 반환하게 된다  
이 값을 가지고 Controller단에서 설정한 알람 문구가 발생하게 되며 다시 로그인 화면으로 돌아가게 된다.

## 로그인처리 Mapper

```
<select id="getMember" resultType="member">
    <![CDATA[
        select *
        from member
        where id = #{id}
    ]]>
</select>
```

## 로그인처리 Controller

```
// 로그인 처리
@RequestMapping(value = "/login", method = RequestMethod.POST)
public String login(MemberVO vo, Model model, HttpSession session) {
    System.out.println("로그인");

    MemberVO member = memberService.getLoginById(vo);
    if(member == null) {
        model.addAttribute("message", "아이디 또는 비밀번호가 일치하지 않습니다.");
        return "member/loginForm";
    }
}
```

## 로그인처리 Service

```
@Override
public MemberVO getLoginById(MemberVO member) {

    MemberVO result = memberDAO.getLoginById(member);
    if(result != null) { // 아이디가 존재하면
        if(result.getPassword().equals(member.getPassword())) {
            return result;
        }
    }
    return null;
}
```

## 로그인처리 DAO

```
// 로그인 처리
@Override
public MemberVO getLoginById(MemberVO member) {
    return sqlSession.selectOne(NAMESPACE + ".getMember", member);
}
```

회원 정보 조회

- 1. 회원 정보 조회시 다음 화면이 출력된다
- 2. 로그인 되어있는 사용자의 DB 정보를 읽어와 화면에 출력한다
- 3. 비밀번호와 같은 개인정보에 민감한 사항은 별표(\*)처리한다
- 4. "구매 리스트 이동" 버튼 및 "정보수정" 버튼을 누를시 Page 이동 및 별도의 팝업 알림창이 출력되며 사용자 정보 수정이 가능하다

DAILY

Q

sssss님

로그아웃

New

남성

여성

남녀공용

Q&A

내 정보

주문 내역 조회

구매후기

회원 정보 조회

이름

teyyyyy

이름 변경

비밀번호

\*\*\*\*\*

비밀번호 변경

생년월일

2022-12-14

생년월일 변경

닉네임

sssss

닉네임 변경

이메일

test111@daily.com

이메일 변경

포인트 조회

100

구매 리스트로 이동

## 회원정보조회 상세화면1 (정보수정)

DAILY

sssss님 로그아웃

New남성남녀공용Q&A

내 정보  
주문 내역 조회  
구매후기

### 회원 정보 조회

|        |                   |
|--------|-------------------|
| 이름     | tessss1           |
| 비밀번호   | .....             |
| 생년월일   | 2022-12-14        |
| 닉네임    | sssss             |
| 이메일    | test111@daily.com |
| 포인트 조회 | 100               |

구매 리스트로 이동

이름 변경

비밀번호 변경

생년월일 변경

닉네임 변경

이메일 변경

회원 정보 변경

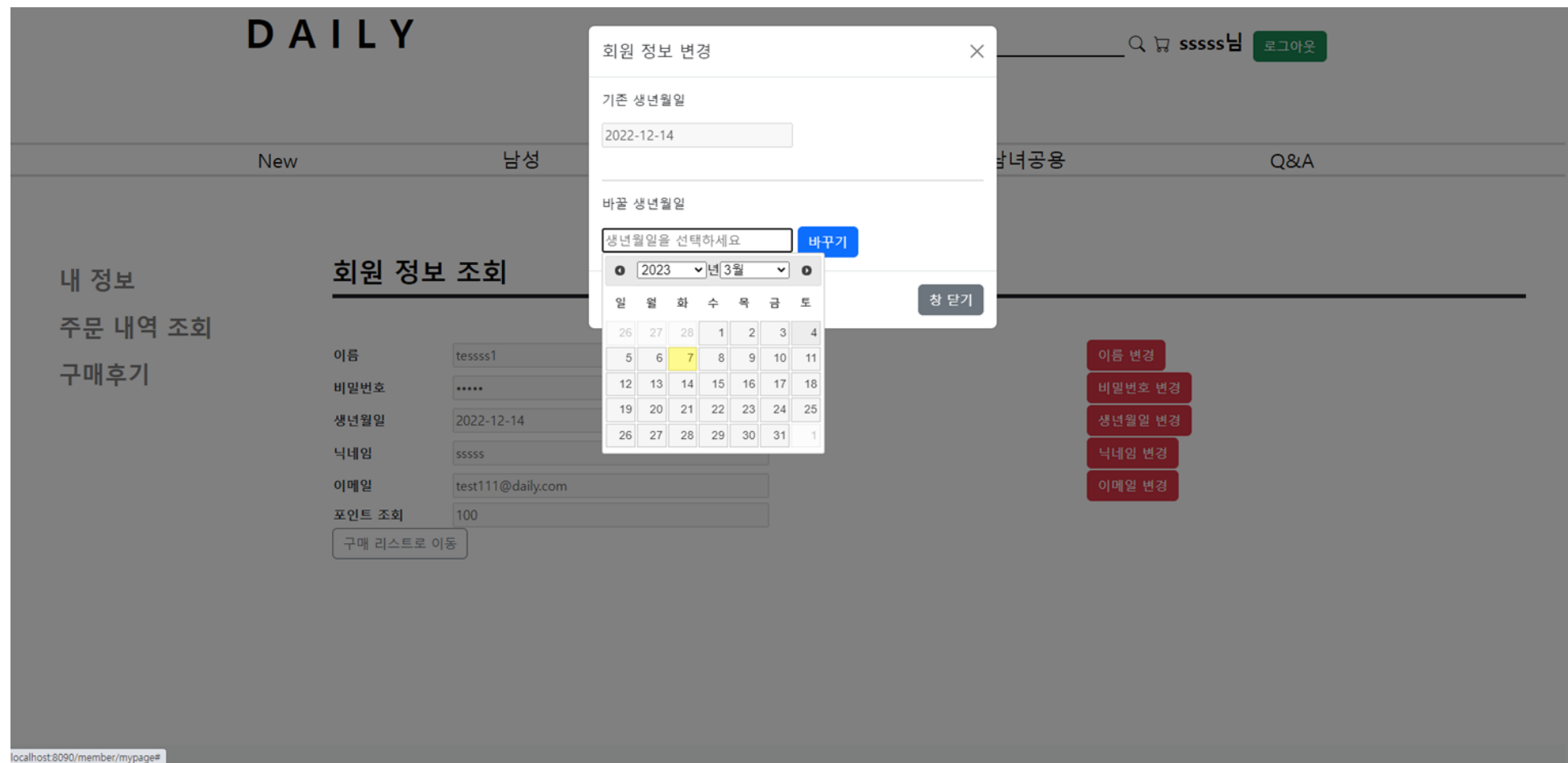
기존 이름  
tessss1

바꿀 이름  
입력란

바꾸기

창 닫기

## 회원정보조회 상세화면2 (생년월일)



## 회원정보조회 주요기능1

1. 로그인 할 때 Session에 사용자의 정보를 member 객체로 저장하여 사용 할 수 있게 한다
2. 사용자가 볼 수 있는 View Page에 Controller에서 값을 전달할 수 있는 model이라는 요소를 사용. Session에 저장된 member의 값을 가지고서 model에 주입하여 사용자에게 보여지게 한다

### 회원로그인처리 Controller

```
@RequestMapping(value = "/login", method = RequestMethod.POST)
public String login(MemberVO vo, Model model, HttpSession session) {
    System.out.println("로그인");

    MemberVO member = memberService.getLoginById(vo);
    if(member == null) {
        model.addAttribute("message", "아이디 또는 비밀번호가 일치하지 않습니다.");
        return "member/loginForm";
    }

    session.setAttribute("member", member);
}
```

### 회원정보 View 관련Controller

```
@RequestMapping(value = "/mypage")
public String mypageForm(HttpSession session, Model model) {

    model.addAttribute("member", memberService.getMemberById(((MemberVO)session.getAttribute("member"))));
    return "member/myPage";
}
```



## 회원정보조회 주요기능2

1. 로그인 처리를 할 때 이미 Sessstion에 객체로 저장한 값을 memberVO 형식에 맞게 가져오고 그것을 member 변수에 넣어서 회원 정보 수정을 진행한다.
2. SQL에 업데이트문 쿼리를 사용하여 DB의 정보를 수정한다

### 회원정보 수정 Controller

```
@RequestMapping(value = "/modify", method = RequestMethod.POST)
public String modify(MemberVO memberVO, HttpSession httpSession ) throws Exception{

    Log.info("memberVO" + memberVO);

    //////////
    MemberVO member = (MemberVO)httpSession.getAttribute("member");

    if(memberVO.getName()!=null) {
        member.setName(memberVO.getName());
    }

    if(memberVO.getPassword()!=null) {
        member.setPassword(memberVO.getPassword());
    }

    if(memberVO.getNickname()!=null) {
        member.setNickname(memberVO.getNickname());
    }
    if(memberVO.getEmail()!=null) {
        member.setEmail(memberVO.getEmail());
    }
    if(memberVO.getBirthDay()!=null) {
        member.setBirthDay(memberVO.getBirthDay());
    }

    if(memberService.modify(member)==1){
```

### 회원정보 수정 Service

```
//회원 정보 수정
@Override
public int modify(MemberVO memberVO) throws Exception {

    return memberDAO.modify(memberVO);
}
```

### 회원정보 수정 DAO

```
//회원 정보 수정
@Override
public int modify(MemberVO memberVO) {
    return sqlSession.update(NAMESPACE + ".modify", memberVO);
}
```

### 회원정보 수정 Mapper

```
<update id="modify" parameterType="member">
    <![CDATA[
        UPDATE member
        SET    name =#{name}, password = #{password}, nickname = #{nickname}, email= #{email}, birthday = #{birthday}
        WHERE id = #{id}
    ]]>
</update>
```

## 회원정보조회 주요기능3

1. 팝업창이 뜬 후 사용자가 정보를 입력하고 바꾸기 버튼을 눌렀을 때 onclick 평선이 작동되고 jQuery가 작동되며 정보가 변경된다
2. 이름을 쓰지 않았다면 알람과 메시지가 출력된다

## 정보변경 modal 창 HTML

```
10 <script src="/resources/js/member.js"></script>
11
12 <p>기존 이름</p>
13
14 <input class="w-50" type="text" placeholder="이름" value="{member.name }" disabled="disabled">
15 <br/><br/><hr>
16 <p>바꿀 이름</p>
17
18 <input class="w-50" type="text" id="rename" placeholder="입력란" value="" />
19 <button type="button" class="btn btn-primary" onclick="fn_memberRegister();">바꾸기</button>
20
21 </body>
22 </html>
```

## 팝업창정보변경 jQuery

```
9
10 // 입력한 값을 가져온다.
11 let rename = $("#rename").val();
12
13 // 이름 항목에 값이 없으면 입력하도록 한다.
14 if($("#rename").val() == "") {
15     alert("이름은 필수 입력 항목입니다.");
16     $("#rename").focus();
17     return false;
18 }
19 $.ajax({
20     type: "POST",
21     url: "/member/modify",
22     data: {name:rename},
23     success: function(result) {
24         if(result == "Y") {
25             alert("이름을 바꾸었습니다.");
26             location.href = "/member/mypage";
27         }
28         else{
29             alert("실패하였습니다..");
30         }
31     },
32     error: function(result) {
33         alert("이름 바꾸기에 실패하였습니다.");
34     }
35 });
```

리뷰작성

- 1. 후기작성하기 버튼을 누르면 사용자가 상품에 대해서 구매후기를 작성하는 화면이 출력된다
- 2. 별 개수와(1~5개) 200자 이내로 글 입력이 가능하다
- 3. 사진을 직접 찍어서 파일을 업로드 할 수 있다.
- 4. 아래의 저장버튼을 눌러 저장 가능하며 이전화면으로 되돌아갈 수 있다.

DAILY

🔍

김현수님

로그아웃

New

남성

여성

남녀공용

Q&A

내 정보

주문 내역 조회

구매후기

구매후기(리뷰) 작성 화면

상품은 만족스러웠나요?  
(별 개수 선택)

별 개수를 선택해 주세요

어떤 점이 좋았나요?

200자 이내로 입력해주세요

상품 사진 첨부하기

파일 선택

선택된 파일 없음

저장

이전화면으로

리뷰작성 상세화면1

DAILY

Q

🛒

김현수님

로그아웃

New

남성

여성

남녀공용

Q&A

내 정보

주문 내역 조회

구매후기

구매후기(리뷰) 작성 화면

상품은 만족스러웠나요?  
(별 개수 선택)

별 개수를 선택해 주세요

별 개수를 선택해 주세요

☆

☆☆

☆☆☆

☆☆☆☆

☆☆☆☆☆

상품 사진 첨부하기

파일 선택

선택된 파일 없음

저장

이전화면으로

## 리뷰작성 주요기능

1. jQuery를 이용하여 좋아요 개수를 반영 할 수 있다
2. 상품의 사진 첨부 또한 가능하며 200자 이내로 리뷰를 작성할 수 있다

## 좋아요 별 개수 저장 jQuery

```
6<script>
7 //e.프리벤트디폴트 찾아서 저장 버튼에 넣어야함.
8
9 $(document).ready(function() {
10     $("#saveData").click(function() {
11         fn_star();
12         console.log($("#star").val());
13     })
14 })
15 function fn_star() { //선택스타로 가져오면 굳이
16
17     let starC = "";
18
19     if($("#1").is(":selected")) {
20         starC = 1;
21     } else if($("#2").is(":selected")) {
22         starC = 2;
23     } else if($("#3").is(":selected")) {
24         starC = 3;
25     } else if($("#4").is(":selected")) {
26         starC = 4;
27     } else if($("#5").is(":selected")) {
28         starC = 5;
29     }
30
31     $("#star").val(starC);
32 }
```

## 리뷰 작성화면 HTML


```
<body>
<form id="Writer" name="WrierForm" method="POST" >
  <div><h1>구매후기(리뷰) 작성 화면</h1></div>
  <div class="container text-center border-top border-5 border-dark w-100 " ><br><br>
    <div class="col">
      <div class="col" id="size">상품은 만족스러웠나요?<h5>(별 개수 선택)</h5></div>
      <div class="col" >
        <select name="score" class="form-select form-select-lg mb-3" id="selectStar" aria-label=".form-select-lg exam
          <option>별 개수를 선택해 주세요</option>
          <option id="1" value="1">★</option>
          <option id="2" value="2">★★</option>
          <option id="3" value="3">★★★</option>
          <option id="4" value="4">★★★★</option>
          <option id="5" value="5">★★★★★</option>
        </select>
        <!-- <input type="hidden" id="star" name="score" value="" /> -->
      </div><br><br>
      <div class="col" id="size">어떤 점이 좋았나요?</div>
      <div class="row" >
        <input type="text" name="content" placeholder="200자 이내로 입력해주세요" value="" ></input></div><br><br>
      <div class="col">
        상품 사진 첨부하기
      </div>
      <div class="input-group mb-3">
        <input type="file" id="picture" class="form-control" id="inputGroupFile01">
      </div> <br>
      <button class="btn btn-outline-secondary" type="submit" id="saveData" >저장</button>
      <button class="btn btn-outline-secondary" type="button" onclick="location.href='${contextPath}/member/reView'">
    </div>
```

구매후기정보

- 1. 구매후기작성 버튼을 누르면 다음 화면이 출력된다
- 2. 구매한 상품의 사진과 시간 그리고 사용자가 작성한 스타일후기를 볼 수 있다
- 3. 또한 사용자가 선택한 좋아요 개수가 나타난다 (1~5개)
- 4. "후기작성하러가기" 버튼을 클릭하여 후기 작성 페이지로 이동할 수 있다

구매후기정보 화면

구매 후기 작성

| 상품정보  | 구매/확정일 스타일 후기 좋아요 별 갯수 |       |          |                          |
|---|------------------------|-------|----------|--------------------------|
|    | 2022-11-15             | 스타일후기 | 좋아요 별 갯수 | <a href="#">후기작성하러가기</a> |
|   | 2022-11-15             | 스타일후기 | 좋아요 별 갯수 | <a href="#">후기작성하러가기</a> |
|  | 2022-11-15             | 스타일후기 | 좋아요 별 갯수 | <a href="#">후기작성하러가기</a> |
|  | 2022-11-15             | 스타일후기 | 좋아요 별 갯수 | <a href="#">후기작성하러가기</a> |
|  |                        |       |          |                          |

## 구매후기정보 주요기능

1. C태그를 이용한 for문의 활용으로 구매품목의 갯수에 맞추어 자동으로 목록이 생성된다
2. EL식의 활용으로 SQL에서 정보를 가져와 사용자에게 해당하는 보여준다

## 구매정보 화면 HTML

```
<tbody>
  <c:forEach begin="1" end="5" var="i">
    <tr>
      <td>
        <div class="a">
          
        </div>
      </td>
      <td>
        ${review.r_enroll }
      </td>
      <td>
        ${review.r_enroll }
      </td>
      <td>
        ${review.score }
      </td>
      <td class="text-break">
        <button type="button" onclick="location.href='${contextPath}/reView/reViewWriter'"
          class="btn btn-outline-secondary">
          후기작성하러가기
        </button>
      </td>
    </tr>
  </c:forEach>
</tbody>
```