



원데이 클래스에  
오신 것을 환영합니다.

# Contents

01



개발자 소개

02



개발개요

03



개발과정

04



개발기획

05



ERD 모델링 과정

06



등급별 주요기능

07



주요기능설명

08



디버깅

## 개발자 소개

Gil Tae  
Hyeung

## CONTACT

**Tel.** 010-xxxx-xxxx

**Email.** [r1605866@naver.com](mailto:r1605866@naver.com)

**GitHub.** <https://github.com/Giltaehyeong>

### One. Where did I make

[이젠 아카데미] Java Web Backend\_Developer 과정

### Two. What did I make

Team Project

One day Class 플랫폼 사이트

Assigned Task

회원가입 / 회원정보수정 / 회원탈퇴 / 로그인 / 마이페이지

On&Off 갤러리 게시판 / 클래스 관리자 게시판 List

Debugging

### Three. How did I make

With Java / JSP / Spring / MySQL

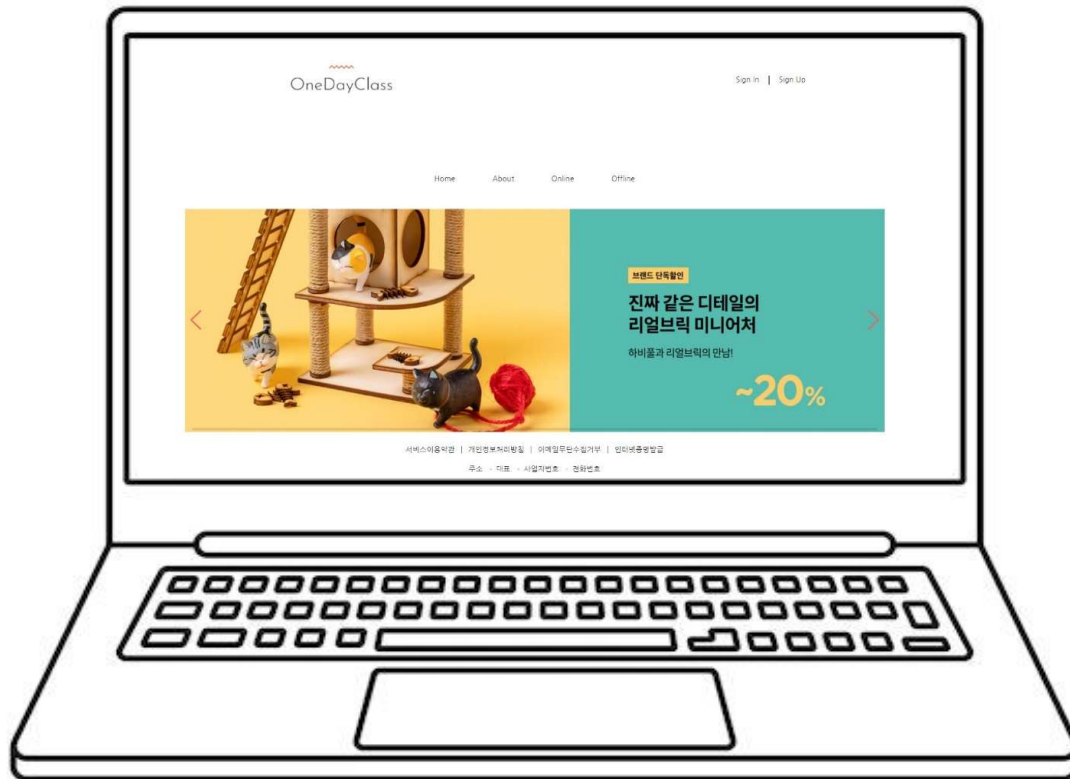
HTML5 / CSS / JavaScript / Jquery / Ajax

VSC / Eclipse / GitHub / Tomcat

## PART 2 개발 개요 - 프로젝트 소개

# OneDayClass

핸드메이드, 쿠킹, 드로잉 등 여러 종류의 클래스를 배우고 싶을 때, 온/오프라인 클래스에 등록 되어 있는 수업을 고객이 간편하게 신청하여 이용할 수 있는 플랫폼 사이트.



## Project

- 온/오프 클래스 수업 신청 / My 클래스 관리
- 리뷰 / Q&A / 장바구니 / 결제 / 회원관리 / 등업

## 개발 인원 및 기간

- 개발 기간 : 2021.12.27 ~ 2022.01.22
- 개발 인원 : 4명

## 적용 기술

- Java / JSP / Spring / MySQL
- HTML / CSS / JS / jQuery
- VSC / Eclipse / GitHub

## 깃 허브 링크

- Jsp Model1(기반 설명)  
[https://github.com/Giltaehyeong/Proj\\_OneDayClass](https://github.com/Giltaehyeong/Proj_OneDayClass)
- 동작 영상  
<https://youtu.be/kDs0r3Ea3LI>

## Front End



## Back End



## DataBase



PART 3    **개발과정 - 프로젝트 일정**

SUN	MON	TUE	WED	THU	FRU	SRI
	27	28	29	30	31	1
	기획		DB 설계 및 ERD 작성			
2	3	4	5	6	7	8
	메인페이지		로그인 및 회원가입 구현, DB연동			
9	10	11	12	13	14	15
	게시판 구현(등업, 리뷰, 문의, 클래스 게시판)					
16	17	18	19	20	21	22
	기능 페이지 구현(관리자, 마이페이지, 클래스, 결제관련)				디버깅	

## 개발 기획 - 아이디어 종합

## 원데이 클래스 홈페이지

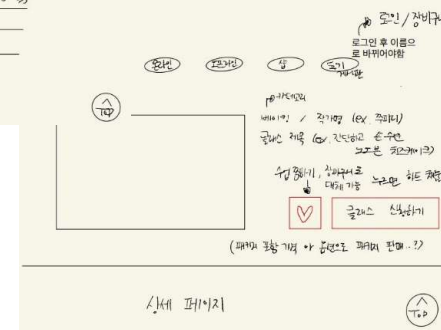
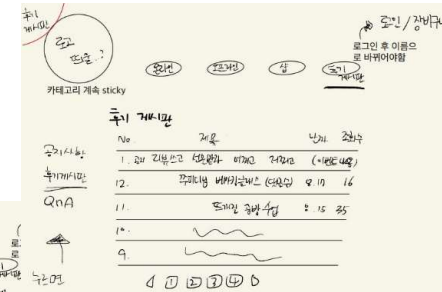
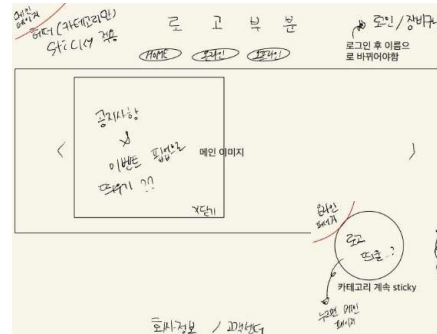
여러 분야의 클래스들은 모아놓고 회원이 원하는  
클래스를 수강 할 수 있고 클래스 자료 판매  
(기본재료를 포함 구성,  
클래스에 필요한 구성품만)  
↳ 기본재료 포함 X

그리고 홈페이지 쇼핑물게는 원데이 클래스 강사님의  
작품 판매, 후기 게시판에 글은 올리면 적립금 추가.

## 필요한 페이지

메인화면, 강사소개...?, 클래스 & 작품 판  
(클래스 상세  
페이지에 적어도  
필요함)

로그인, 회원가입창, 관리자 버전, 마이페이지  
후기게시판.



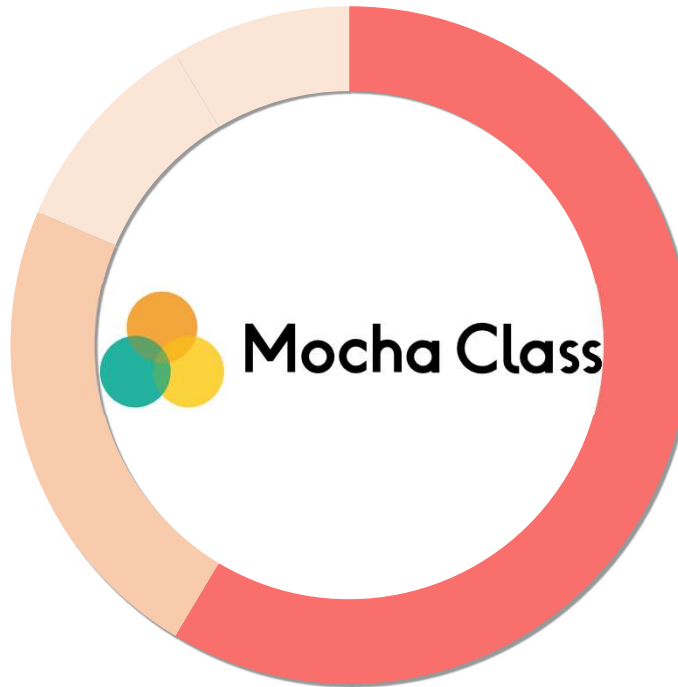
## 아이디어 회의

## • 개발 계기

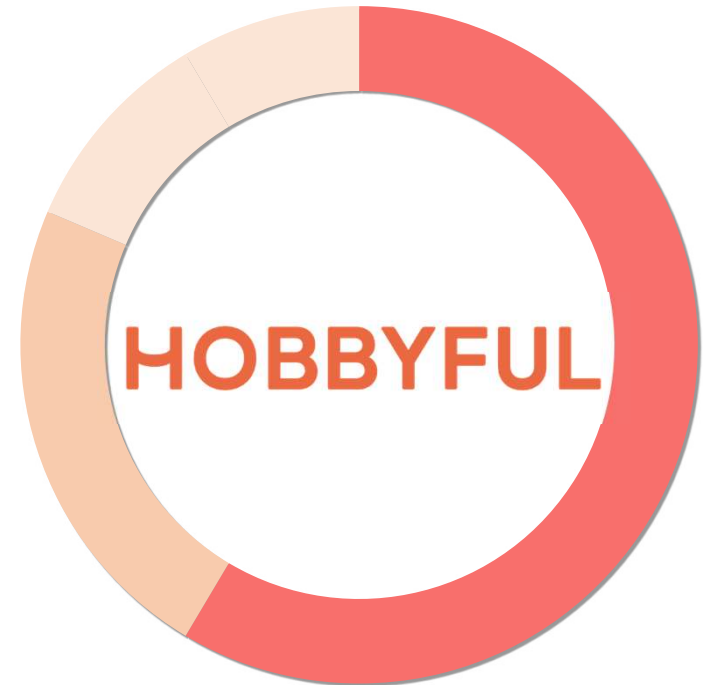
요새 다양한 원데이 클래스 수업이 있는데, 블로그나 인스타를 통해 예약을 하는 곳이 많습니다. 한 블로그에서 하나의 클래스만 볼 수 있고 수업 신청, 수강생 관리 등 불편하다는 단점이 있습니다. 이러한 이슈를 해결하기 위해 다양한 클래스들을 한 눈에 볼 수 있고, 나의 클래스 신청 내역도 보다 편리하게 통합하여 관리 할 수 있는 플랫폼 사이트를 개발하게 되었습니다.



아이디어스



모카클래스



하비풀

## 벤치마킹 사이트

기존 플랫폼 사이트인 아이디어스, 모카클래스, 하비풀 3개의 사이트를 벤치마킹하여 구현하였습니다.



## 개발기획 - 요구사항 정의서

요구사항 정의서

파일 수정 보기 삽입 서식 데이터 도구 확장 프로그램 도움말 및 초전에 마지막으로 수정했습니다.

100% 123% 기본값 (Arial) 10 B I U A

A	B	C	D	E
1	No	구분	요구사항 ID	요구사항 명
2	1	헤더	REQ-001	로그인 / 회원가입 / 장바구니 / 로그
3	2	헤더	REQ-002	카테고리
4	3	헤더	REQ-003	온라인 / 오프라인 button
5	4	헤더	REQ-004	헤더가늘
6	5	메인	REQ-005	온라인 페이지
7	6	메인	REQ-006	오프라인 페이지
8	7	메인	REQ-007	답글 쓰우기 & 이벤트 답글
9	8	메인	REQ-008	회원 이미지(슬라이드 스)
10	9	푸터	REQ-009	회사정보
11	10	메인	REQ-010	온라인 수업 정할가능
12	11	공용	REQ-011	온라인 클래스 카테고리
13	12	공용	REQ-012	추천리스트
14	13	공용	REQ-013	오프라인 클래스 카테고리
15	14	공용	REQ-014	추천리스트
16	15	공용	REQ-015	오프라인 수업 정할
17	16	공용	REQ-016	로그인(통합)
18	17	공용	REQ-017	회원가입(통합)
19	18	공용	REQ-018	회원가입(통합)
20	19	개인	REQ-019	마이페이지(개인)
21	20	개인	REQ-020	마이페이지(프로필설정)
22	21	개인	REQ-021	마이페이지(상터배제기)
23	22	개인	REQ-022	마이페이지(매뉴)
24	23	개인	REQ-023	마이페이지(주문내역)
25	24	개인	REQ-024	마이페이지(배출원)
26	25	개인	REQ-025	마이페이지(내정보관리)

요구사항 상세설명(프로세스 어썬구)

로그인, 회원가입, 장바구니, 로그 페이지로 넘어감.  
로그인시 화면에 o으로 바뀌게 함.

스티커 적용

해당 페이지로 넘어감.  
클릭되어 있는 헤더 카테고리가 글씨가 굵게  
로그 디자인

회사정보 노출 수 있게  
최신순, 리뷰순(정렬)  
수업 종류별 카테고리  
슬라이드식 구현  
지역별 카테고리, 수업 중  
슬라이드식 구현  
최신순, 리뷰순

로그인 화면 이전 페이지로

아이디, 비밀번호, 비밀번호

silsp/css/CSS\_QuizIstarbuzz\_Quiz (게시

silsp/js/p08\_GNB/GNB 세트), silsp/jque

silsp/js/p09\_Slideshow (가로이동), silsp

F	G	H
중요도	비고	
	silsp/css/CSS_Quiz/starbuzz_Quiz (게시판이동)	
	silsp/js/p08_GNB(GNB 세트), silsp/jquery/p03_AddClass(속성제거)	
	silsp/s/p09_Slideshow (가로이동), silsp/jquery/p11_Indicator(버튼이동)	
	silsp/jquery/JQ_Quiz/p02_GalleryBBS_zzupd/lbbs(갤러리게시판)	
	silsp/jquery/JQ_Quiz/p02_GalleryBBS_zzupd/lbbs(갤러리게시판)	
	회원가입 통합, join 등업 update, 1번 2번 silsp/js/JS_Quiz/p04_Login_Validation/member(다음 로그인, 로그인 오류시 나오는 중간페이지?) silsp/jsp_Model1/Proj_Ch08_Bean/src/main/webapp/join(회원가입 기능) 카카오 지도 api(주소입력) silsp/js/p12_JS_Object(정규표현식)	
	silsp/js/p11_LayerPopup(모달레이어) <회원탈퇴>	

## 요구사항 정의서 만들기 / 기획

- 아이디어를 종합해 홈페이지의 요구사항 정의서를 만들며 페이지 구분별로 항목을 나눠 기재하였습니다.
- 요구사항 상세설명에 홈페이지를 구현 할 때 참고할 수 있도록 구현 내용을 기재하였습니다.
- 중요도를 상/중/하로 구현목록의 우선순위를 정해 마감기한에 차질없이 수행할 수 있도록 분류하였습니다.
- 비고란엔 구글링을 통해 해당 페이지를 구현할 때 도움되는 링크를 기재하였습니다.

## ERD 모델링

## 테이블 컬럼 만들기

요구사항 정의서

파일 수정 보기 삽입 서식 데이터 도구 확장 프로그램 도움말 및 초면에 마지막으로 수정했습니다.

K23

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	회원가입	학생	강사	온라인과목	오프라인과목	댓글							
3	아이디	아이디	아이디	카테고리	카테고리	No							
4	비밀번호	비밀번호	비밀번호	과목명	과목명	아이디(FK)							
5	이름	이름	이름	날짜	날짜	내용							
6	핸드폰번호	번호	번호	가격	가격	작성일자							
7	유연번호	주소	주소	클래스	클래스								
8	주소	이메일	이메일	수강신청	수강신청								
9	상세주소	등급	등급	대표이미지	지역								
10	이메일주소	상호명	상호명	과목코드	과목코드								
11	등급												
12	상호명												
13	SNS												
14	관리자페이지	강사관리페이지	1:1 문의 게시판	리뷰 게시판	장바구니	관리자페이지							
15	로그인/로그아웃	로그인/로그아웃	No	No	비밀번호								
16	상세페이지	상세페이지	아이디(FK)	아이디(FK)	결제정보	Q&A 게시판							
17	주문	주문	결제금액	결제금액	장바구니No	강사신청관리							
18	결제금액		결제금액	결제금액		결제금액							
19			비밀번호	비밀번호									
20			작성일자	작성일자									
21													
22													
23													
24													
25													
26													
27													

## DB 정규화 작업

요구사항 정의서

파일 수정 보기 삽입 서식 데이터 도구 확장 프로그램 도움말 및 초면에 마지막으로 수정했습니다.

M49

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		회원가입	클래스 게시판	OnA 게시판	리뷰 게시판	등업 게시판						
3		아이디(PK)	클래스 게시판 번호(PK)	Qna 글번호(PK)	리뷰 글번호(PK)	등업 글번호(PK)						
4		비밀번호	고유코드	게시자 아이디	리뷰 아이디	등업 아이디						
5		이름	강사 이름	문의 제목	리뷰 제목	등업 제목						
6		핸드폰번호	사용자 아이디	문의 내용	리뷰 내용	등업 내용						
7		유연번호	클래스 카테고리	문의 작성 날짜	리뷰 작성 날짜	등업 작성 날짜						
8		주소	클래스 제목	상대적인 위치 값	리뷰 추천수	등업 상호명						
9		상세주소	클래스 상세내용	부모글 번호	리뷰 작성 날짜	등업 SNS						
10		이메일주소	클래스 날짜	답변의 길이	파일 이름	등업 작성 날짜						
11		등급	클래스 가격	답변 작성자 아이디	파일 크기 저장	상대적인 위치 값						
12		상호명	비밀번호	파일 이름	게시물 상태	부모글 번호						
13		SNS	검색일 이름	파일 크기 저장	클래스 게시판 번호(FK)	답변의 길이						
14			검색일 크기 저장	게시물 상태		크기 저장						
15			파일 이름	클래스 게시판 번호(FK)	리뷰 추천 히스토리	게시물 상태						
16			파일 크기 저장		아이디(FK)	작성일자						
17			최대 수강인원		리뷰 게시판 번호(FK)							
18			수강신청 인원									
19			지역									
20			on/off 유무	장바구니	결제확인 히스토리	결제						
21			게시물 상태	아이디(FK)	결제 번호(PK)	아이디(FK)						
22			추천수	클래스 게시판 번호(FK)	결제 아이디	클래스 게시판 번호(FK)						
23					결제 이름	강사 이름(FK)						
24			클래스 하루 히스토리		핸드폰 번호	클래스 제목(FK)						
25			아이디(FK)		유연번호	결제번호(FK)						
26			클래스 게시판 번호(FK)		주소	수강받는 수량						
27					상세주소	결제일자						
28					이메일주소							
29					소속							
30					휴게비							
31					휴게비금액							
32					결제일자							
33												
34												

정규화 작업(중복 테이블, 컬럼 제거 / 테이블 통합)

회원가입 / 학생 / 강사  
고객 마이페이지

회원가입

온라인과목/오프라인 과목/ 이미지

클래스 게시판

1:1 문의 게시판 / 이미지 / 댓글

Q&amp;A 게시판

리뷰 게시판 / 이미지

리뷰 게시판

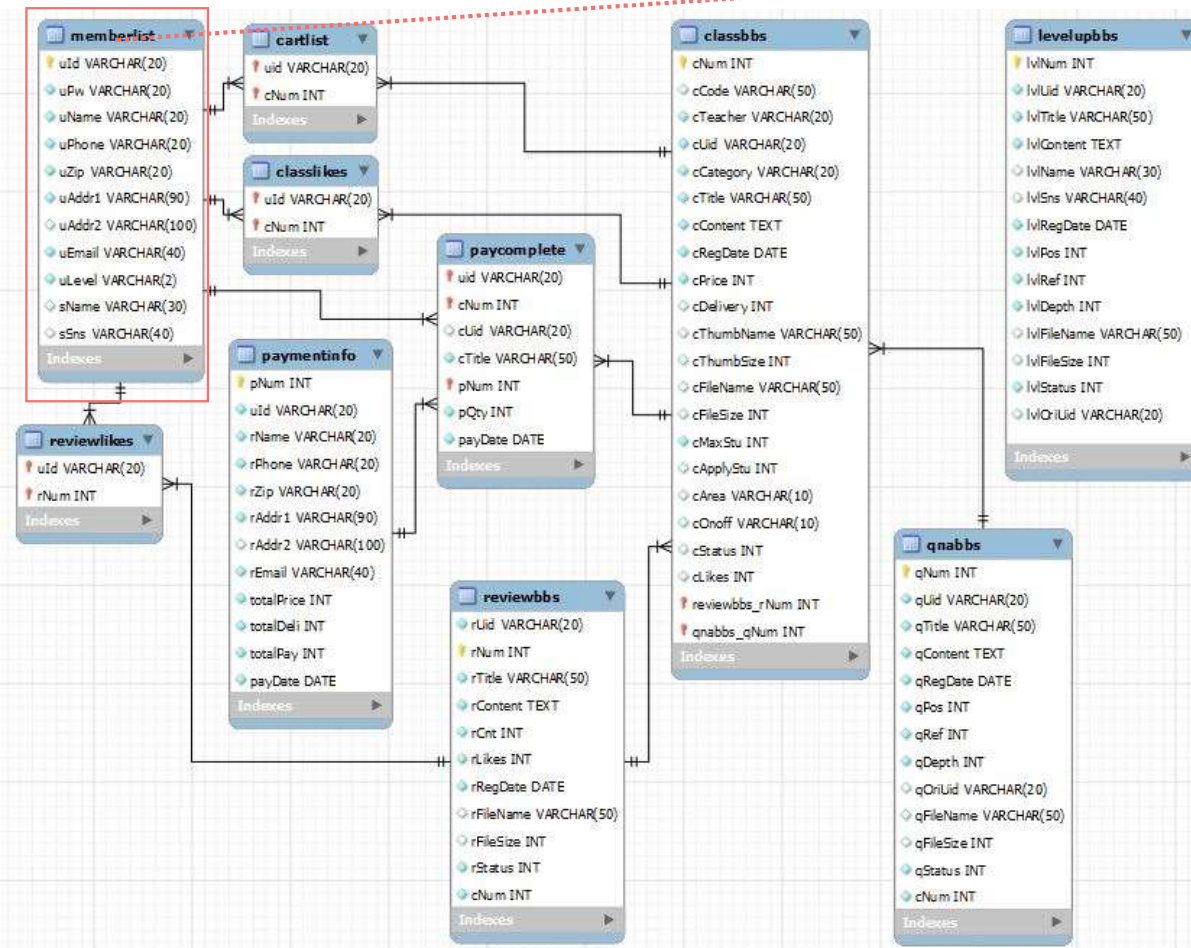
관리자페이지 / 관리자 문의 게시판  
이미지 / 댓글

등업 게시판

결제 / 주문내역

결제확인 리스트 / 결제

## ERD 모델링 - 회원 관리



10개 테이블 유기적 연결

## Memberlist 구현

## 회원관련 PageList

Login.jsp / 로그인 페이지

LoginProc.jsp / 로그인 프로세스 페이지

Logout.jsp / 로그아웃 페이지

Member.jsp / 회원가입

IdCheck.jsp / Id 중복확인

MemberProc.jsp / 가입 프로세스

Member\_Mod.jsp / 회원정보 수정

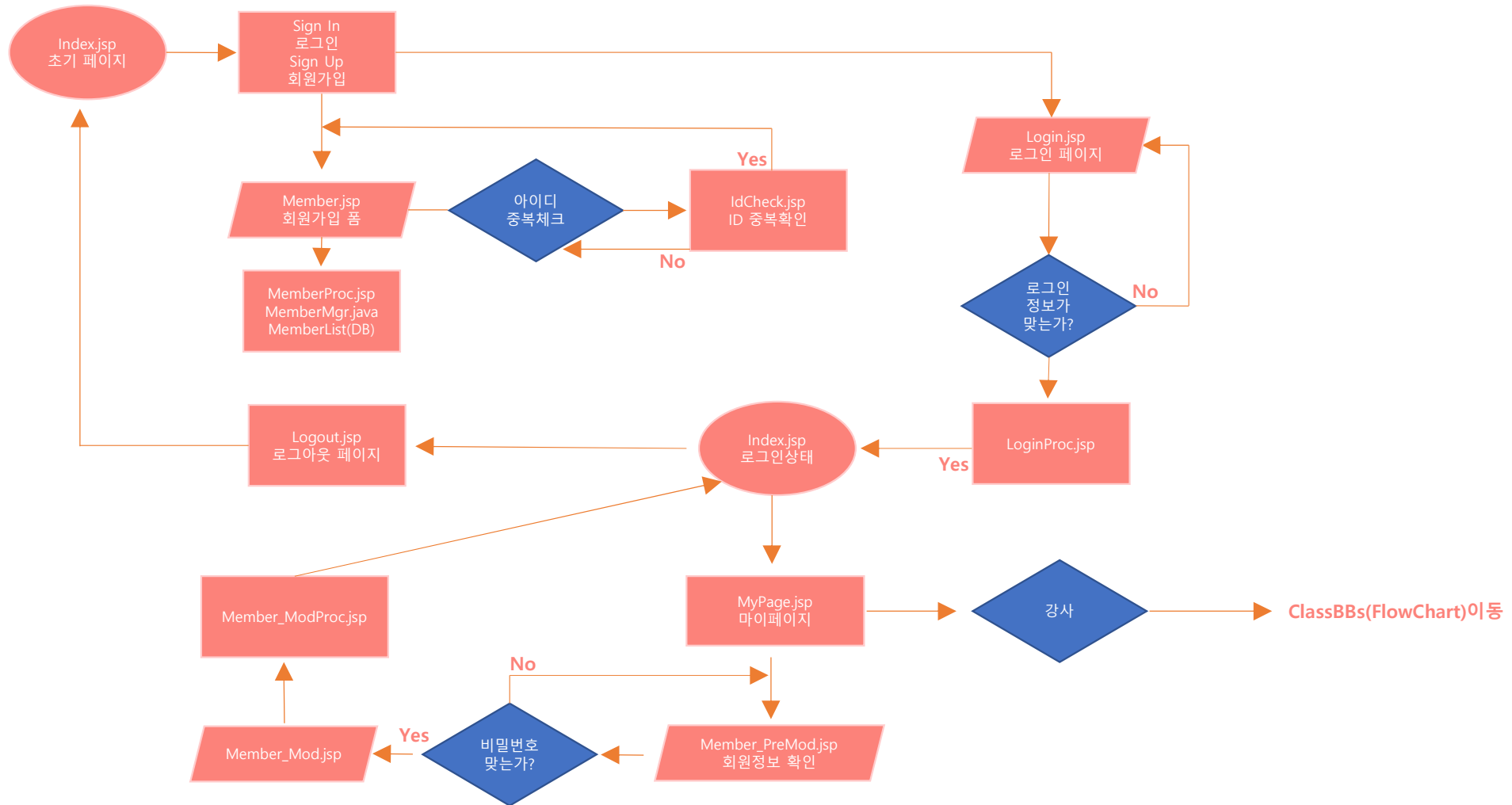
Member\_ModProc.jsp / 수정 프로세스

Member\_Del.jsp / 회원 탈퇴

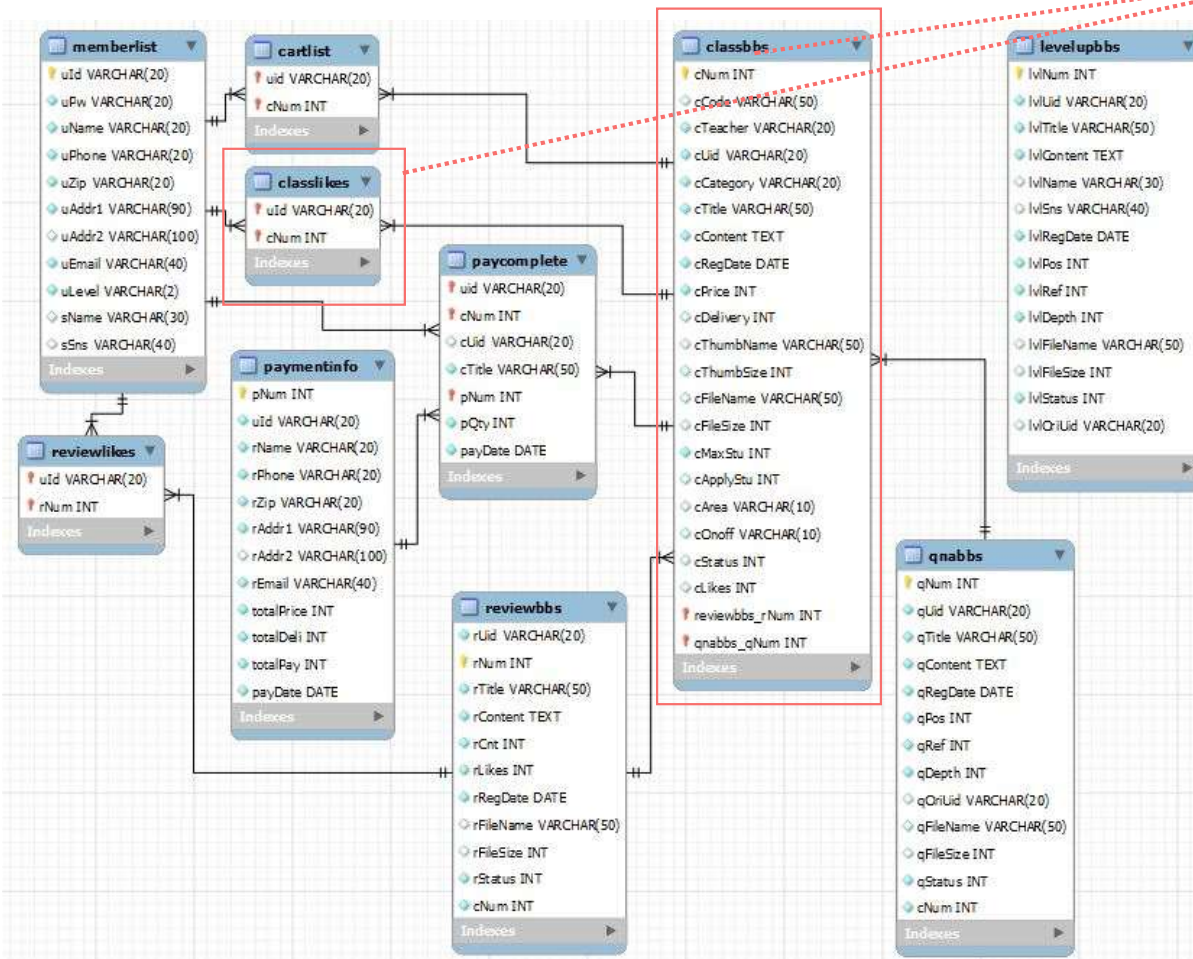
Member\_PreMod.jsp / 회원정보 확인 페이지

Mypage.jsp / 마이페이지

## FlowChat - Memberlist (참고)



## ERD 모델링 - 클래스 게시판



10개 테이블 유기적 연결

## ClassBBS

## 구현 PageList

ClassOnList.jsp / 온라인 이미지 게시판

ClassOffList.jsp / 오프라인 이미지 게시판

ClassList.jsp / 클래스 관리자 게시판

ClassApproval.jsp / 관리자 승인 페이지

ClassPost.jsp / 클래스 작성

ClassPostProc.jsp / 작성 프로세스

ClassUpdate.jsp / 클래스 수정

ClassUpdateProc.jsp / 수정 프로세스

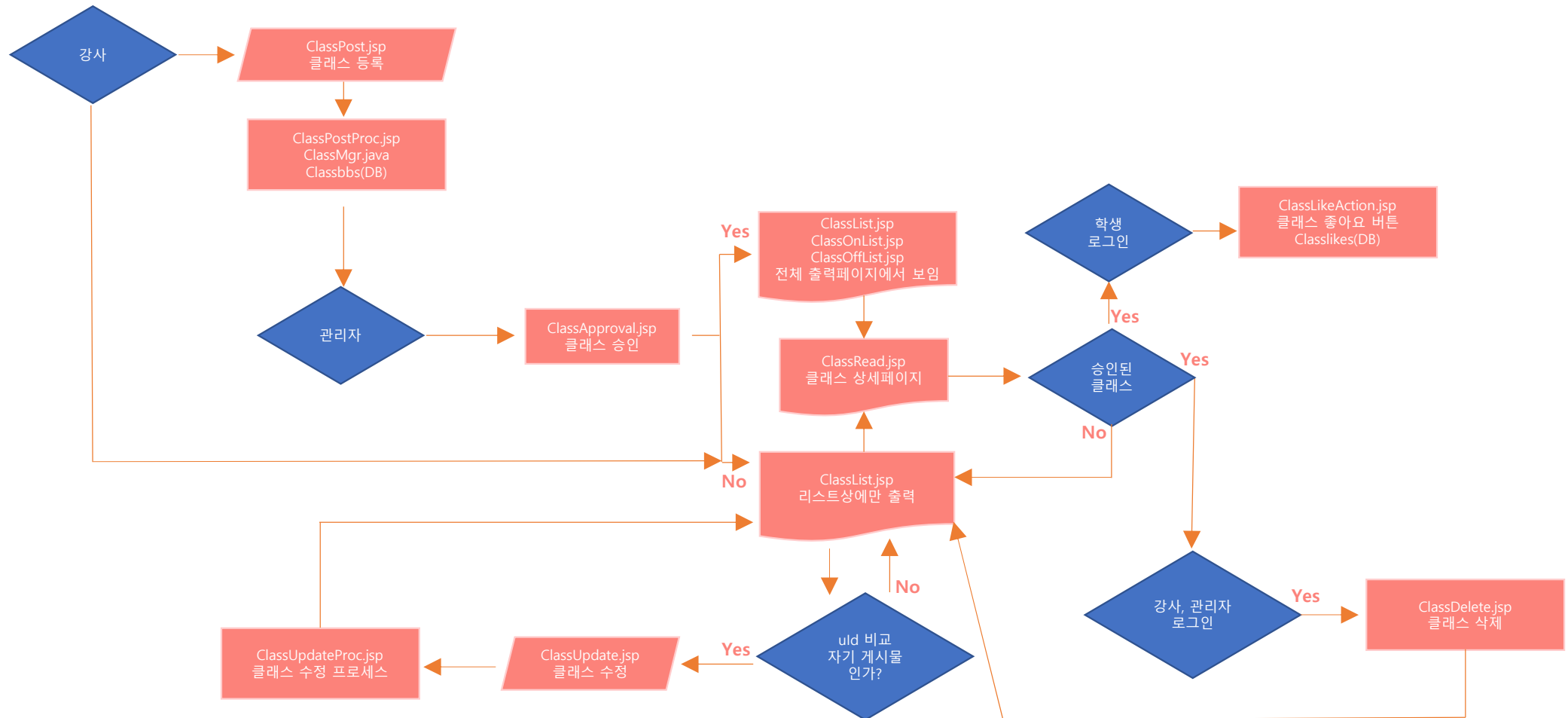
ClassDelete.jsp / 클래스 삭제 페이지

ClassRead.jsp / 클래스 상세페이지

ClassLikeAction.jsp / 좋아요 버튼



## FlowChat - Classbbs (참고)



## 등급별 주요기능

● ● ● ● ●

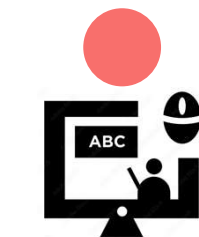
PERSONAL



회원가입



개인 로그인



클래스 신청



리뷰/문의 게시글작성



등업 신청

## 등급별 주요기능



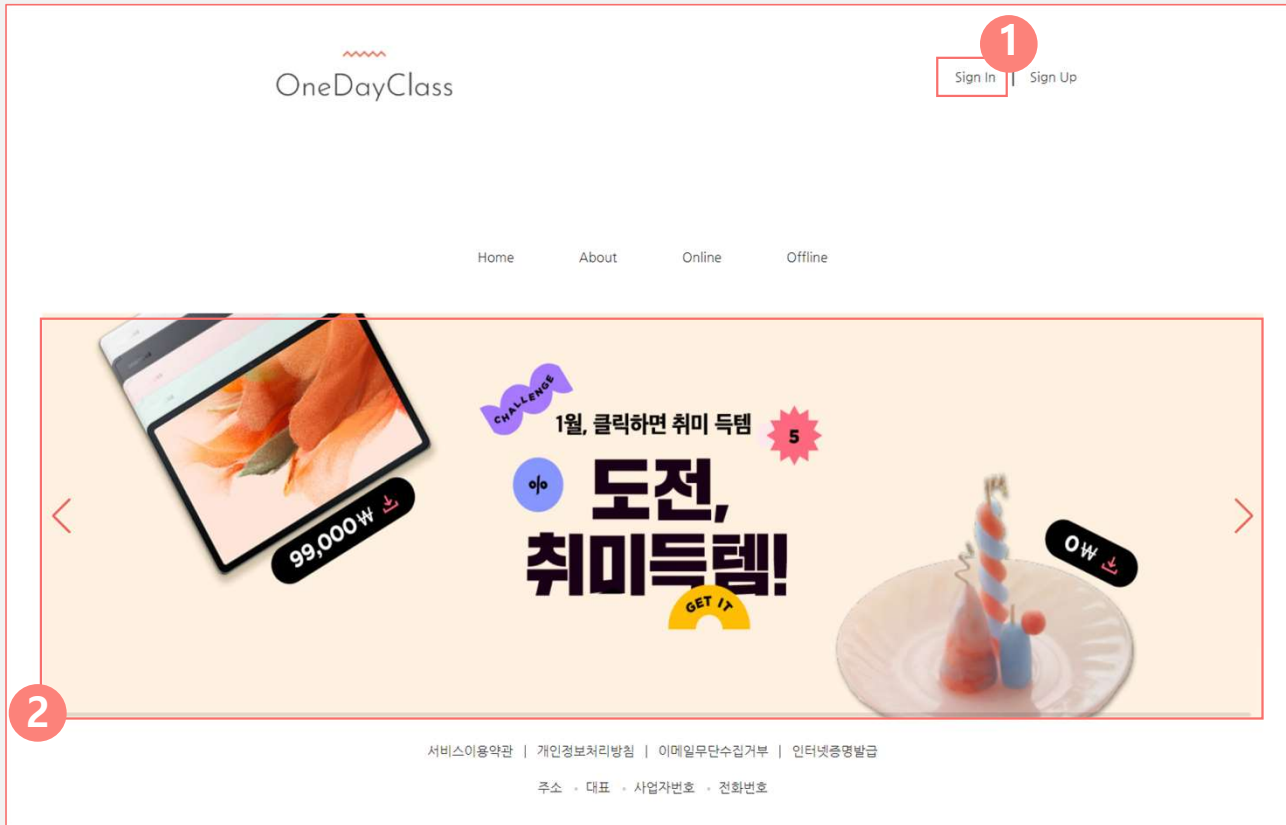


...

ADMIN



## 주요기능 - 메인 페이지



## 2 Swiper.js 기능 사용

Swiper.js 라이브러리를 사용하여 깔끔한 슬라이드 효과 구현



## 1 Sign In / 로그인 - 등급별 로그인

버튼 클릭 시 로그인 페이지로 이동

Login.jsp

길태형님 환영합니다. | 로그아웃 | 🛒

```
if (res) {
    String[] uStat = mMgr.getMemberStat(uid);
    int cartCnt = pMgr.getCartCnt(uid);
    session.setAttribute("idKey", uid);
    session.setAttribute("nameKey", uStat[0]);
    session.setAttribute("levelKey", uStat[1]);
    session.setAttribute("cartCntKey", cartCnt);
    //msg = "로그인에 성공하였습니다."
    response.sendRedirect("/Proj_OneDayClass/Index.jsp");
}
```

등급별 로그인 세션

OneDayClass 테스트님 환영합니다. | 로그아웃 | 🛒

```
<% if (iuid != null && iuLevel.equals("3")) { //관리자 로그인%>
```

관리자 로그인

OneDayClass 길태형 선생님 환영합니다. | 로그아웃

```
} else if (iuid != null && iuLevel.equals("2")) {
```

강사 로그인

OneDayClass 관리자님 안녕하세요. | 로그아웃

```
} else if (iuid != null) {
```

학생 로그인

→ 잘못 입력 시 로그인 불가

localhost:84 내용:  
아이디와 비밀번호를 확인해주세요.

확인

LoginProc.jsp

조건문을 사용해 DB에 저장되어 있는 uid, uPw 데이터와 비교해 같지 않다면 메시지 출력할 수 있도록 구현

uid 요소와 비교 후 결과값이 참인 경우 sendRedirect() 메서드 호출하여 Index.jsp 파일을 불러옴.

로그인 정보 출력을 위해 아이디, 이름, 등급, 장바구니 갯수를 페이지에 받아오게 구현

## PART 7 주요기능 - 회원가입

### 3 비밀번호 확인

비밀번호 재확인 요소를 필수 입력요소로 구현하여 미 입력 시 입력하도록 설정

### 1 Sign Up / 회원가입

버튼 클릭 시 회원가입 페이지로 이동

Member.jsp / 회원정보 입력 / \* 필수요소 모두 입력 시

MemberProc.jsp / InsertMember() 메서드를 호출해 DB에 전송

localhost:84 내용:  
회원가입을 완료!

회원가입 완료 알림창

### 2 중복확인

idCheck.jsp / 중복확인 페이지

테이블에 중복 uid가 없을 경우  
확인 버튼 활성화

테이블에 중복 uid가 있을 경우  
확인 버튼 비활성화

checkId() 메서드를 호출해 테이블을 조회 후 조건문으로 중복아이디 확인

```
<h2><%=uid %></h2>
<%
if (result) {
    out.println("<b>사용중인 아이디입니다.</b>");
} else {
    out.println("<b>사용가능한 아이디입니다.</b>");
}
%>
```

### 유효성 검사

\* 아이디  
@#5@#5 ID중복확인

클릭시

```
//// 아이디 = 영어, 숫자, 특수기호(underscore), 5글자 이상 20글자 이하
let uid = $("#uid").val();
let uidLen = uid.length;
let uidReg = /^[a-zA-Z0-9_]{5,20}$/;
if (uidLen < 5) {
    alert("ID는 영어, 숫자, 특수기호(underscore), 5글자 이상 20글자 이하");
    $("#uid").focus();
    return;
} else if (uidReg.test(uid)) {
    alert("ID는 영어, 숫자, 특수기호(underscore), 5글자 이상 20글자 이하");
    $("#uid").focus();
    return;
}
```

데이터 유효성 검사를 통해 올바르지 않을 경우 알림창 출력

localhost:84 내용:  
ID는 영어, 숫자, 특수기호(underscore), 5글자 이상 20글자 이하

## PART 7 주요 기능 - 회원가입

### \* 전화번호

전화번호를 입력해주세요.

### \* 주소

우편번호

우편번호찾기

주소

상세주소를 입력해주세요.

### \* 이메일

이메일 주소를 입력해주세요.

☐ 전부 동의합니다.

☐ 만 14세 이상입니다.

☐ 이용약관에 동의합니다.

☐ 개인정보수집·이용에 동의합니다.

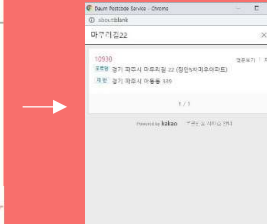
동의하고 가입하기

4

5

## 4 주소 찾기 / 다음 API

버튼 클릭 시 다음 API 실행



\* 주소

10930 Zip 우편번호찾기

경기 파주시 마두리길 22 uAddr1

101동 601호 uAddr2

InsertMember() 메서드를 통해 JDBC를 이용한 insert 구문을 이용하여 데이터 삽입

## 5 체크박스 동의

jQuery를 사용하여 모두체크 동작구현

☐ 전부 동의합니다.

체크박스 버튼에 id 속성부여.

체크 시

☒ 전부 동의합니다.

☒ 만 14세 이상입니다.

☒ 이용약관에 동의합니다.

☒ 개인정보수집·이용에 동의합니다.

jQuery 로직

```
$("#allChk").click(function(){
    let checked = $("#allChk").is(":checked");
    if(checked)
        $("input:checkbox").prop("checked", true);
    else
        $("input:checkbox").prop("checked", false);
});
```

View

```
<label>
  <input type="checkbox" id="allChk">
  <span> 전부 동의합니다. </span> <br> <br>
</label>
<label>
  <input type="checkbox" id="uChk1">
  <span> 만 14세 이상입니다. </span> <br>
</label>
<label>
  <input type="checkbox" id="uChk2">
  <span> 이용약관에 동의합니다. </span> <br>
</label>
<label>
  <input type="checkbox" id="uChk3">
  <span> 개인정보수집·이용에 동의합니다. </span>
</label>
```

## 주요기능 - 마이페이지(내 정보 관리)



1

Click

학생

회원정보 확인

test1

비밀번호

확인

마이페이지 접속 전 비밀번호 확인

Home About Online Offline

김태형

회원등급 학생

주문 내역

내 정보 관리

회원 정보 수정

회원 탈퇴

서비스이용약관 | 개인정보처리방침 | 이메일무단수집거부 | 인터넷중영발급

주소 : 대표 : 사업자번호 : 전화번호

Click

3

만약에 uLevel(등급)이 1(학생)이면

```
<%if (uLevel.equals("1")) {%>
<li class="subMenuLi"><a href="/Proj_OneDayClass/sign/Member_Mod.jsp" id="memModBtn">회원 정보 수정</a></li>
<li class="subMenuLi"><a href="/Proj_OneDayClass/sign/Member_Mod.jsp" id="memModBtn">회원 정보 수정</a></li>
<li class="subMenuLi"><a href="#" id="memDrop">회원 탈퇴</a></li>
<% } else { %>
```

Else / 2(선생)

```
<% } else { %>
<li class="subMenuLi"><a href="/Proj_OneDayClass/sign/Member_Mod.jsp" id="memModBtn">회원 정보 수정</a></li>
<li class="subMenuLi"><a href="#" id="memDrop">회원 탈퇴</a></li>
<% } %>
```

회원 정보 수정

아이디 test1

\* 패스워드

패스워드 확인

이름 김태형

\* 전화번호 010-1234-5678

\* 주소 10930 우편번호찾기

경기 파주시 마두리길 22

101동 601호

\* 이메일 r1605866@naver.com

정보수정 다시쓰기 메인으로 로그인

## 1 회원정보 수정

버튼 클릭 시 비밀번호 검사 후 회원 정보수정 페이지로 이동

MemberMod.jsp / 회원수정 페이지

Member\_PreMod.jsp / 회원정보확인 페이지

Member\_ModProc.jsp / modifyMember() 메서드 호출

## 2 Vector class / List 구현

데이터를 저장된 순서를 유지하며 불러오기 위해 사용

```
public Vector modifyMember(String uid) {
    Vector<MemberBean> vList = new Vector<>();
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;
    try {
        objConn = pool.getConnection();
        sql = "select * from memberlist where uid=?";
        objPstmt = objConn.prepareStatement(
            objPstmt.setString(1, uid);
        objRs = objPstmt.executeQuery();
        if (objRs != null) {
            while (objRs.next()) {
                MemberBean memBean = new MemberBean();
                memBean.setUid(objRs.getString("uid"));
                memBean.setPw(objRs.getString("upw"));
                memBean.setName(objRs.getString("uName"));
                memBean.setPhone(objRs.getString("uPhone"));
                memBean.setZip(objRs.getString("uZip"));
                memBean.setAddr1(objRs.getString("uAddr1"));
                memBean.setAddr2(objRs.getString("uAddr2"));
                memBean.setEmail(objRs.getString("uEmail"));
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return vList;
}
```

Model 요소를 Vector 객체로 선언

Memberlist의 uid 컬럼으로 조회 후 Vector 배열요소에 담다

```
<% if (sessionUid != null) { // 현재 로그인 상태이면 %>
<!-- // 로그인 상태 시작 // -->
<%
    MemberBean memBean = (MemberBean)vList.get(0);
    String uid = memBean.getUid();
    String upw = memBean.getUpw();
    String uPhone = memBean.getuPhone();
    String uEmail = memBean.getuEmail();
    String uZip = memBean.getuZip();
    String uAddr1 = memBean.getuAddr1();
    String uAddr2 = memBean.getuAddr2();
    %>
```

오브젝트 vList 데이터를 0번 요소부터 순서대로 출력

변경 가능한 요소는 input 요소에 value값 기재

불가능한 요소는 출력 요소만 기재

## 3 회원탈퇴

버튼 클릭 시 확인 유무 메시지 출력 후 삭제

localhost:84 내용:

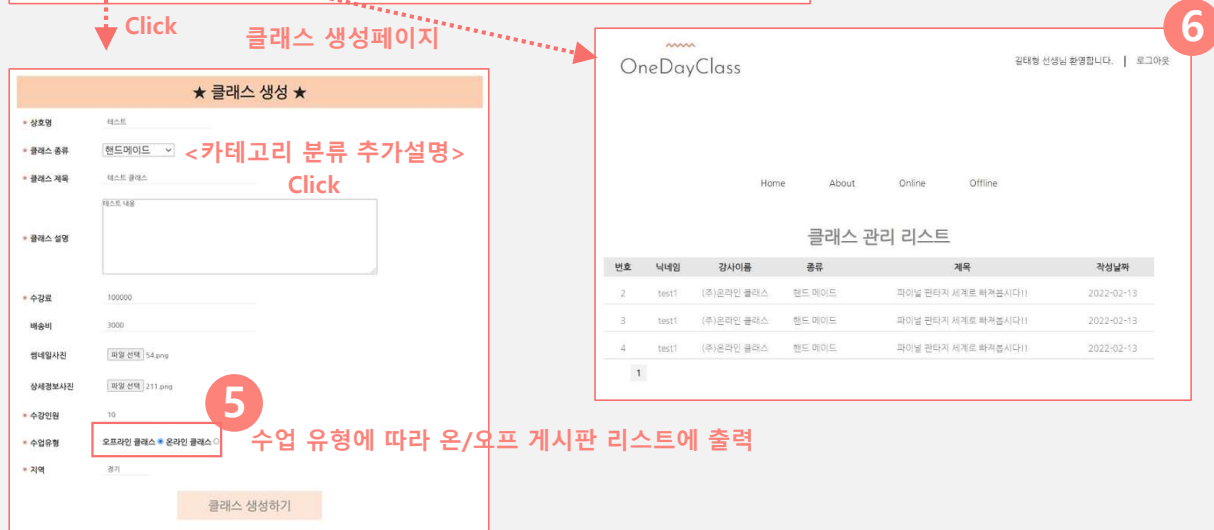
정말 삭제하시겠습니까?

확인 취소

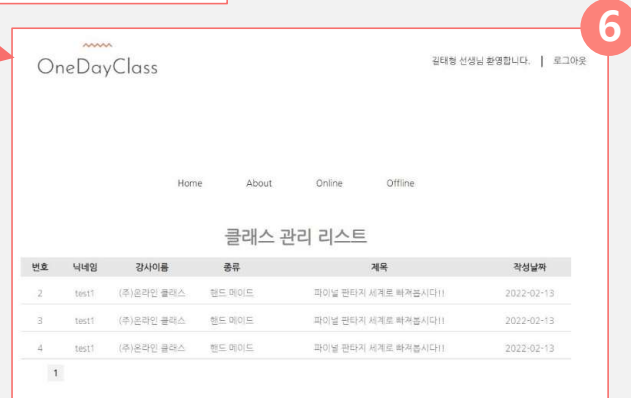
Member\_Del.jsp deleteMember() 메서드 호출

Delete 구문 Where uid = "?" 절 세션 요소에 연결되어있는 uid 요소와 비교하여 테이블에서 삭제함.

## PART 7 주요기능 - 마이페이지 (클래스 관리)



28페이지 추가설명



### 4 클래스 생성 페이지

버튼 클릭 시 클래스 등록페이지로 이동

ClassPost.jsp / 클래스 생성페이지 → `<%if(uId != null && str2.equals(uLevel)) { %>`

강사 이상 등급만 접근 가능.

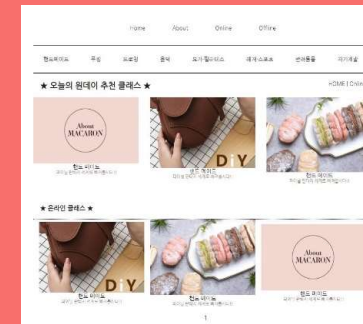
ClassPostProc.jsp / InsertBoard() 메서드를 호출하여 DB에 데이터 입력

### 5 클래스 생성 / 온&오프 체크

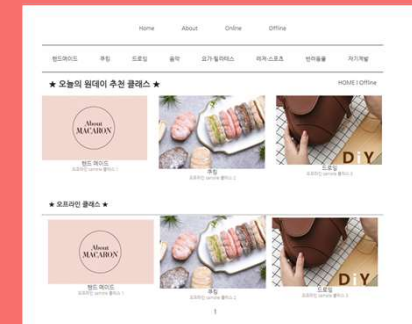
수업 유형 선택 시 해당 갤러리 리스트에 게시물 출력

↓ 온라인클래스

↓ 오프라인 클래스



ClassOnList.jsp



ClassOffList.jsp

### 6 클래스 관리 / 게시판 List

버튼 클릭 시 클래스 리스트 페이지로 이동

```
<%
if(uLevel.equals("3")){
vList = bMgr.getBoardadmin(start, end);
listSize = vList.size();
}else{
vList = bMgr.getBoardTeacher(start, end);
listSize = vList.size();
}
%>
```

getBoardadmin() 메서드 호출  
모든 게시물 볼 수 있음  
삭제 된 게시물 포함

getBoardTeacher() 메서드 호출  
cStatus < 3(비공개) 까지만 볼 수 있음

## 주요기능 - 클래스 생성 페이지

★ 클래스 생성 ★

\* 상호명

테스트

\* 클래스 종류

핸드메이드

\* 클래스 제목

테스트 클래스

\* 클래스 설명

테스트 내용

\* 수강료

100000

\* 배송비

3000

\* 썸네일사진

파일 선택 54.png

\* 상세정보사진

파일 선택 211.png

\* 수강인원

10

\* 수업유형

오프라인 클래스 온라인 클래스

\* 지역

경기

클래스 생성하기

다중 업로드 구현

```
multi = new MultipartRequest(req, path, // 저장소
    maxSize, // 파일크기
    encType, new DefaultFileRenamePolicy());
```

→ Cos 라이브러리의 DefaultFileRenamePolicy() 메서드를 사용하여 이름이 중복될 경우 저장 시 파일명 뒤에 숫자 하나를 붙여줌.

## 7 MultipartRequest 사용

파일 업로드를 위해 MultipartRequest 사용



Cos 라이브러리 추가 후

Build Path를 통해 jar 파일 추가

```
private static String encType = "UTF-8";
private static int maxSize = 10 * 1024 * 1024;
// 파일크기 : 10MB
```

파일 이름 문자열 타입의 인코딩 방식 "UTF-8"로 설정  
파일 사이즈는 10MB으로 설정

```
String path = req.getServletContext().getRealPath("/src/main/webapp/fileUpload/classfileupload/");
path = UtilMgr.replace(path, ".metadata\\.plugins\\org.eclipse.wst.server.core\\tmp0\\wtpwebapps\\", "");
```

Tomcat의 내장객체 getRealPath() 메서드를 호출하여 경로를 지정

```
// 오버로드 설정
File file = new File(path);
// classfileupload
if (!file.exists())
    file.mkdirs();
```

폴더가 없을 경우 자동생성 하기 위해  
file.mkdirs() 함수 사용

Insert 구문으로 테이블에 데이터 입력

```
multi = new MultipartRequest(req, path, // 저장소
    maxSize, // 파일크기
    encType, new DefaultFileRenamePolicy());

if (multi.getFileName("cThumbName") != null) {
    cThumbName = multi.getFileName("cThumbName");
    cThumbSize = (int) multi.getFile("cThumbName").length();
}

if (multi.getFileName("cFileName") != null) {
    // 이름
    cFileName = multi.getFileName("cFileName");
    // 용량
    cFileSize = (int) multi.getFile("cFileName").length();
}
```

저장소1

저장소2

작성 날짜를 기입하기 위해  
SQL now() 함수 사용

```
String sql = "insert into classbbs("
    + "cCode, cTeacher, cUId, cCategory, cTitle, cContent, cRegDate, cPrice, cDelivery, "
    + "cThumbName, cThumbSize, cFileName, cFileSize, "
    + "cMaxStu, cApplyStu, cOnoff, cArea, cStatus, cLikes) values ("
    + "?, ?, ?, ?, ?, ?, ?, ?, ?, ?, "
    + "?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1, multi.getParameter("cCode"));
pstmt.setString(2, multi.getParameter("cTeacher"));
pstmt.setString(3, multi.getParameter("cUId"));
pstmt.setString(4, multi.getParameter("cCategory"));
pstmt.setString(5, multi.getParameter("cTitle"));
pstmt.setString(6, multi.getParameter("cContent"));
pstmt.setDate(7, new Date());
pstmt.setInt(8, Integer.parseInt(multi.getParameter("cPrice")));
pstmt.setInt(9, Integer.parseInt(multi.getParameter("cDelivery")));
pstmt.setString(10, cThumbName);
pstmt.setInt(11, cThumbSize);
pstmt.setString(12, cFileName);
pstmt.setInt(13, Integer.parseInt(multi.getParameter("cMaxStu")));
pstmt.setInt(14, multi.getParameter("cOnoff"));
pstmt.setInt(15, multi.getParameter("cArea"));
```



## 주요기능 - 온/오프 게시판

Home About Online Offline

1

핸드메이드 쿠키 드로잉 음악 요가·필라테스 레저·스포츠 반려동물 자기계발

★ 오늘의 원데이 추천 클래스 ★

HOME | Offline

About MACARON

핸드 메이드  
오프라인 sample 클래스 1

쿠키  
오프라인 sample 클래스 2

드로잉  
오프라인 sample 클래스 3

★ 오프라인 클래스 ★

About MACARON

핸드 메이드  
오프라인 sample 클래스 1

쿠키  
오프라인 sample 클래스 2

드로잉  
오프라인 sample 클래스 3

1



## 1 카테고리 분류

메뉴 클릭 시 카테고리 별로 출력 될 수 있도록 구현

```
<li class="devTagList"><span onclick="cCategorySelect('1')> 핸드메이드</span></li>
<li class="devTagList"><span onclick="cCategorySelect('2')> 쿠키</span></li>
<li class="devTagList"><span onclick="cCategorySelect('3')> 드로잉</span></li>
<li class="devTagList"><span onclick="cCategorySelect('4')> 음악</span></li>
<li class="devTagList"><span onclick="cCategorySelect('5')> 요가·필라테스</span></li>
<li class="devTagList"><span onclick="cCategorySelect('6')> 레저·스포츠</span></li>
<li class="devTagList"><span onclick="cCategorySelect('7')> 반려동물</span></li>
<li class="devTagList"><span onclick="cCategorySelect('8')> 자기계발</span></li>
```

```
function cCategorySelect(c){
    $("#cCategorySel").val(c);
    $("#categoryform").submit();
}
```

ClassOnList.jsp

span 요소를 클릭하면 클릭한 값이 form을 통해 cCategorySel 값에 저장되고

classbbs.js

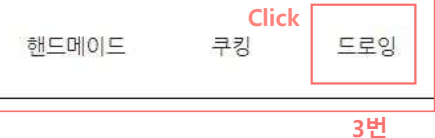
```
public Vector<ClassBean> getBoardList(String cCategorySel, String onoff, int start, int end) {
    Vector<ClassBean> vList = new Vector<>();
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;

    try {
        objConn = pool.getConnection(); // DB연동
        if (cCategorySel.equals("null") || cCategorySel.equals("")) {
            // 카테고리 메뉴 선택 안했을시
            sql = "select * from classbbs where cStatus<3 AND cOnoff = ? order by cNum asc limit ?, ?";
            objPstmt = objConn.prepareStatement(sql);
            objPstmt.setString(1, onoff);
            objPstmt.setInt(2, start);
            objPstmt.setInt(3, end);
        } else {
            // 카테고리 메뉴 클릭 시
            sql = "select * from classbbs where cCategory = ? AND cStatus<3 AND cOnoff = ? order by cNum asc limit ?, ?";
            objPstmt = objConn.prepareStatement(sql);
            objPstmt.setString(1, cCategorySel);
            objPstmt.setString(2, onoff);
            objPstmt.setInt(3, start);
            objPstmt.setInt(4, end);
        }
        objRs = objConn.executeQuery(sql);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }
    return vList;
}
```

getBoard() 메서드의 매개변수를 통해 sql문 조회 시

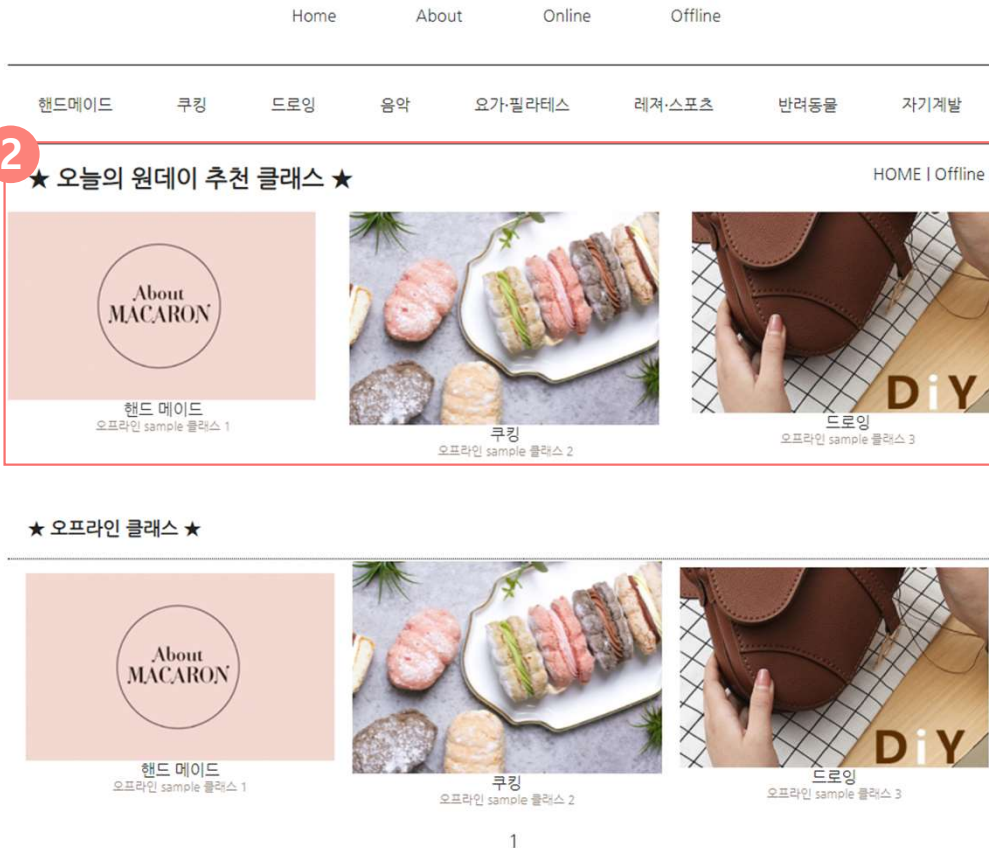
select 구문의 where cCategory = ? 전달되어 카테고리 번호에 해당되는 게시물만 출력하도록 구현

## ★ 오프라인 클래스 ★



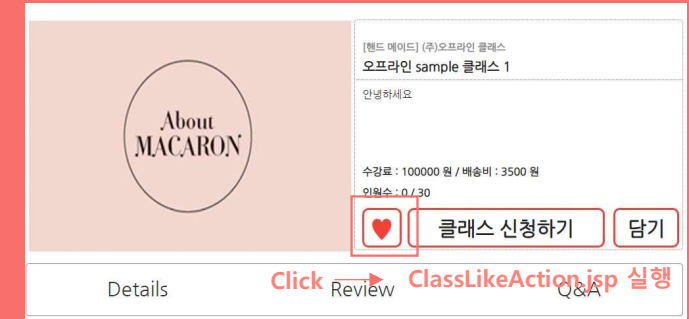


## 주요기능 - 온/오프 게시판



## 2 추천 클래스 List 출력

인기가 많은 게시물부터 차례대로 출력되도록 구현



↓ cLike 컬럼 카운트 증가



&lt;찜하기 추가설명&gt;

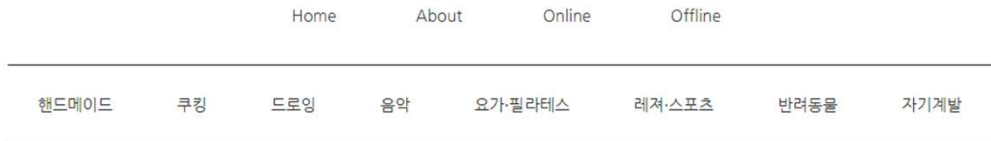
Click

```
if (cCategorySel.equals("null") || cCategorySel.equals("")) {
    // 카테고리 메뉴 선택 안했을시, desc 정렬
    sql = "select * from classbbs where cStatus<3 AND cOnoff = ? order by cLikes desc limit ?, ?";
    objPstmt = objConn.prepareStatement(sql);
    objPstmt.setString(1, onoff);
    objPstmt.setInt(2, start);
    objPstmt.setInt(3, end);
} else {
```

Order by 절 cLike 값의 desc(내림차순) 으로 정렬하여



## 주요기능 - 온/오프 게시판



## ★ 오늘의 원데이 추천 클래스 ★

HOME | Offline



핸드메이드  
오프라인 sample 클래스 1



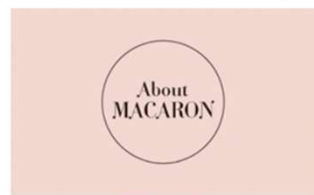
쿠키  
오프라인 sample 클래스 2



드로잉  
오프라인 sample 클래스 3

3

## ★ 오프라인 클래스 ★



핸드메이드  
오프라인 sample 클래스 1



쿠키  
오프라인 sample 클래스 2



드로잉  
오프라인 sample 클래스 3

1

## 3 일반 클래스 List 출력

게시물 리스트를 3x3 형식으로 출력 총 9개까지 출력

## ★ 오프라인 클래스 ★



1 2

```

<%>
<%>
for (int i = 0; i < listSize; i += 3) {
    <tr>
    <%>
    for (int j = 0; j < 3; j++) {
        다중 for문을 사용하여 3행 3열 List 구현
        ClassBean bean = vList.get(i + j);
        int cNum = bean.getNum();
        // 게시물 번호
        String cThumbName = bean.getThumbName();
        String cCategory = bean.getCategory();
        // 카테고리
        String cTitle = bean.getTitle();
        // 제목
        int cStatus = bean.getStatus();
        // 글 속성, 공개 비공개 여부
        String cOnoff = bean.getOnoff();
        // 클래스 on(N), off(Y) 여부
        if (cStatus == 2) {
            <%>

```

게시물 출력 순서

1	2	3
4	5	6
7	8	9

다음 페이지로 이동

10

&lt;페이징 설명 뒷장&gt;

# 주요기능 - 온 / 오프 게시판 / 페이징

페이징 기능을 사용하기 위한 변수 선언

4

```
int totalRecord = 0; // 전체 데이터 수(DB에 저장된 row 개수)
int numPerPage = 9; // 페이지당 출력하는 데이터 수(=게시글 숫자)
int pagePerBlock = 5; // 블록당 표시되는 페이지 수의 개수
int totalPage = 0; // 전체 페이지 수
int totalBlock = 0; // 전체 블록수
```

```
int nowPage = 1; // 현재 (사용자가 보고 있는) 페이지 번호
int nowBlock = 1; // 현재 (사용자가 보고 있는) 블록
int start = 0; // DB에서 데이터를 불러올 때 시작하는 인덱스 번호
int end = 9; // 시작하는 인덱스 번호부터 반환하는(=출력하는) 데이터 개수
// select * from T/N where... order by ... limit start, end;
int listSize = 0; // 1페이지에서 보여주는 데이터 수
```

페이지 나누기 번호 출력 시작

블록마다 시작되는 첫 페이지와 마지막 페이지 관련 작업

```
<table class="page">
<tbody>
<tr>
<!-- 페이징 시작 -->
<td colspan="2" id="pagingTd">
<%
int pageStart = (nowBlock - 1) * pagePerBlock + 1; // 하단 페이지 시작 번호
int pageEnd = (nowBlock < totalBlock) ? pageStart + pagePerBlock - 1 : totalPage; // 하단 페이지 끝 번호
if (totalPage != 0) { // 전체 페이지가 0이 아니라면 = 게시글이 1개라도 있다면
if (nowBlock > 1) { // 페이지 블록이 2이상이면 => 2개이상의 블록이 있어야 가능
12
<span class="moveBlockArea" onclick="moveBlock('<%=nowBlock - 1%>', '<%=pagePerBlock%>')"> &lt; &lt; /> <span>
<%
} else {
<span class="moveBlockArea"> < /> <span>
<%
}
<%
<span class="moveBlockArea"> < /> <span>
<%
}
-->
</td>
</tr>
</tbody>
</table>
```

```
// 페이지 나누기용 페이지 번호 출력 시작
for (; pageStart <= pageEnd; pageStart++) {
if (pageStart == nowPage) { // 현재 사용자가 보고 있는 페이지
<span class="nowPageNum"><%=pageStart%></span>
<%
} else { // 현재 사용자가 보고 있지 않은 페이지
<span class="pageNum" onclick="movePage('<%=pageStart%>')"><%=pageStart%></span>
<%
} // End If
} // End For 페이지 나누기용 페이지 번호 출력 끝
// End If #01_totalPage
</td>
</tr>
</tbody>
</table>
```

페이지 나누기용 번호 출력 시작

페이지 나누기용 출력 종료

페이징 기능 끝

## 4 pageing기능 구현

게시판 페이징 기능 설명

```
int pageStart = (nowBlock - 1) * pagePerBlock + 1; // 하단 페이지 시작 번호
```

Ex) 26개 자료 현재 기준

numPerPage : 5; 페이지당 출력 데이터 수  
pagePerBlock : 5; 블록 당 페이지 수  
nowBlock : 현재 블록  
totalBlock : 전체 블록

totalRecord : 26    totalPage : 6  
nowBlock : 1 => pageStart : 1    pageEnd : 5 (적용결과)  
nowBlock : 2 => pageStart : 6    pageEnd : 10( = totalPage)

```
int pageEnd = (nowBlock < totalBlock) ? pageStart + pagePerBlock - 1 : totalPage;
```

블록당 5페이지 출력 => pageStart    pageEnd  
1블럭 =>    1    5  
2블럭 =>    6    10

1 2 3 4 5 >    다음 → 이전    < 6

6번째 페이지부터 다음 블록으로 이동

★ 오프라인 클래스 ★



★ 오프라인 클래스 ★



< 6

6번 페이지 게시물 출력

1 2 3 4 5 > Click

## 주요기능 - 클래스 관리 리스트

OneDayClass 길태형 선생님 환영합니다. | 로그아웃

Home About Online Offline

### 클래스 관리 리스트

번호	닉네임	강사이름	종류	제목	작성날짜
2	test1	(주)온라인 클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-13
3	test1	(주)온라인 클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-13
4	test1	(주)온라인 클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-13

1

관리자님 안녕하세요. | 로그아웃

## 클래스 관리 리스트

1	번호	닉네임	강사이름	종류	제목	작성날짜
	1	test2	(주)오프라인클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-01
	2	test1	(주)온라인 클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-13
	3	test1	(주)온라인 클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-13
	4	test1	(주)온라인 클래스	핸드 메이드	파이널 판타지 세계로 빠져봅시다!!	2022-02-13

1

클래스 상세페이지로 이동

## 1 클래스 게시판 List

강사는 자기게시물만 출력, 관리자는 모두 출력

cStatus / 비공개(1) / 공개(2) / 삭제(3)  
테이블 컬럼에 3개의 상태 값 설정

강사

```
public Vector<ClassBean> getBoardTeacher(int start, int end) {
    Vector<ClassBean> vList = new Vector<>();
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;

    try {
        objConn = pool.getConnection(); // DB연동
        sql = "select * from classbbs where cStatus < 3 order by cNum asc limit ?, ?";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setInt(1, start);
        objPstmt.setInt(2, end);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }

    return vList;
}
```

cStatus < 3 조건을 통해  
강사는 삭제된 항목은 볼 수 없음

관리자

classbbs 테이블의 cNum  
ASC(오름차순) 출력

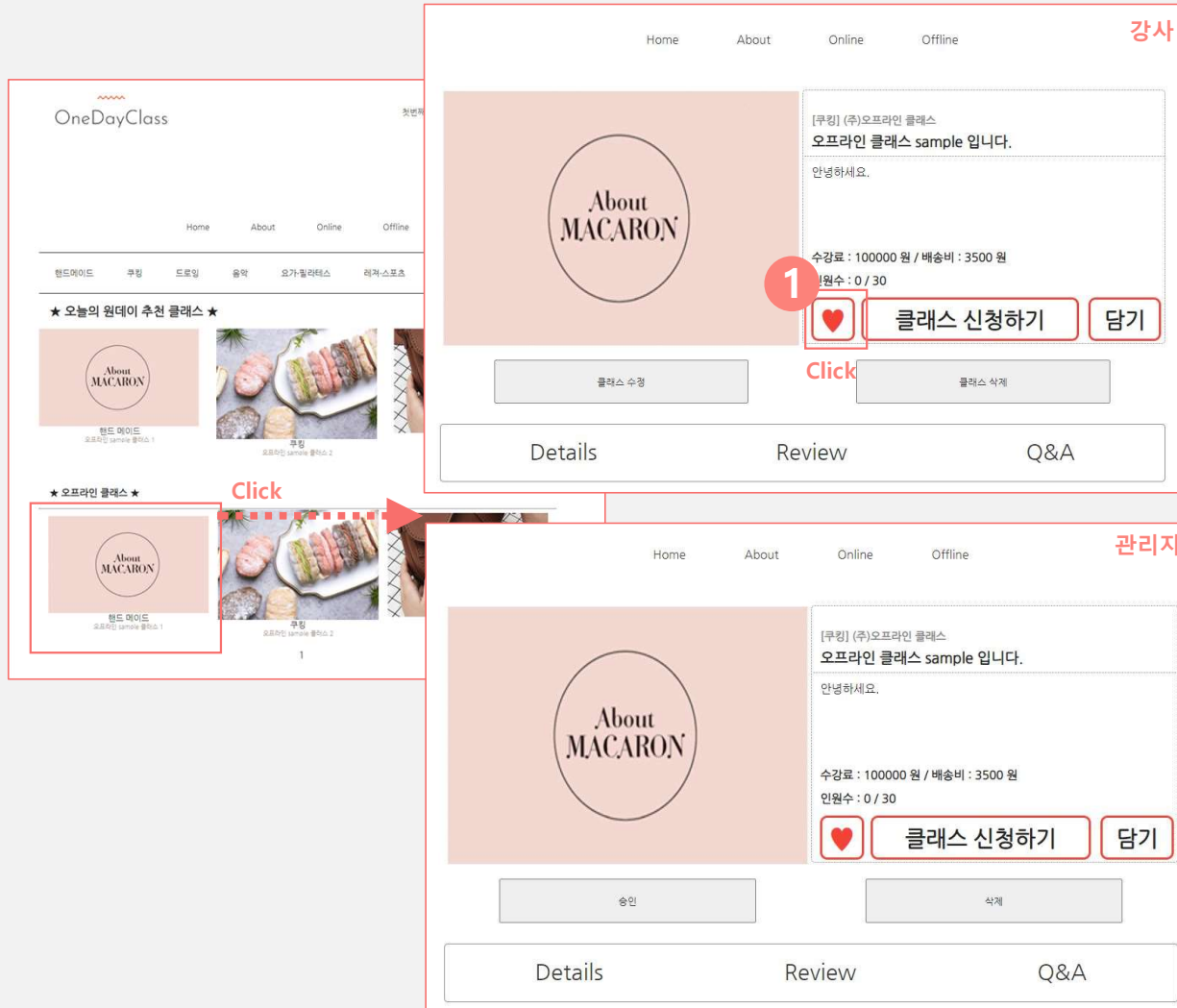
```
public Vector<ClassBean> getBoardadmin(int start, int end) {
    Vector<ClassBean> vList = new Vector<>();
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;

    try {
        objConn = pool.getConnection(); // DB연동
        sql = "select * from classbbs order by cNum asc limit ?, ?";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setInt(1, start);
        objPstmt.setInt(2, end);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }

    return vList;
}
```



## PART 7 주요기능 - 클래스 상세 페이지



### 1 찜하기 버튼

하트 클릭 시 UpdateLike() 메서드 실행

```
public int UpdateLike(ClassLikeBean classLikeBean) {
    int result = 0;

    Connection objConn = null;
    PreparedStatement objPstmt = null;
    String sql = null;

    try {
        objConn = pool.getConnection(); // DB연동
        sql = "insert into classLikes (uid, cNum) values(?,?)";
    } catch (Exception e) {
        System.out.println("SQL이슈 : " + e.getMessage());
    } finally {
        pool.freeConnection(objConn, objPstmt);
    }

    return result;
}
```

Memberlist의 uid 요소와  
ClassBBS의 cNum값을  
Forekey로 참조하여  
classLikes 테이블에 삽입

```
public int InsertLikeCnt(int cNum) {
    int result = 0;

    Connection objConn = null;
    PreparedStatement objPstmt = null;
    String sql = null;

    try {
        objConn = pool.getConnection(); // DB연동
        sql = "update classbbs set cLikes=cLikes+1 where cNum=?";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setInt(1, cNum);
        result = objPstmt.executeUpdate();
    } catch (Exception e) {
        System.out.println("SQL이슈 : " + e.getMessage());
    } finally {
        pool.freeConnection(objConn, objPstmt);
    }

    return result;
}
```

Constraint Primary key(uid, cNum) 값을  
지정하여 해당 테이블 PK요소로 만들.

InsertLikeCnt() 메서드 호출  
Update 구문을 사용하여  
Classbbs 테이블의 cLike 값을 1 증가시킴

```
result = objPstmt.executeUpdate();

} catch (Exception e) {
    System.out.println("SQL이슈 : " + e.getMessage());
} finally {
    pool.freeConnection(objConn, objPstmt);
}

return result;
```

각 메서드 실행 시  
objPstmt.executeUpdate()가 호출되었다면  
Result값에 1을 증가시켜 return

classLikes 테이블에 동일한 uid, cNum  
데이터가 있다면 objPstmt.executeUpdate()


```
<% 실행되지 않아 Result 값이 0
LikeBean.setId(uid);
LikeBean.setCNum(cNum);
int result = bMgr.UpdateLike(LikeBean);
if(result == 0){
    alert("이미 찜하신 강의입니다.");
    history.back();
} else{
    alert("찜완료 ^.^");
    //location.href = history.back(-1);
    bMgr.InsertLikeCnt(cNum);
    history.back();
}
%>
```

Result 값이 1인 경우

Result 값이 0인 경우  
중복 데이터로 인식

## 주요기능 - 클래스 상세 페이지

Home About Online Offline **강사**



[핸드 메이드] (주)오프라인 클래스2  
오프라인 클래스 sample2 입니다.

sample2

수강료 : 10000 원 / 배송비 : 3500 원  
인원수 : 0 / 20 **초과시**

**클래스 담기** **결제**

**3** 클래스 수정

Details Review Q&A

수강료 : 10000 원 / 배송비 : 3500 원  
인원수 : 30 / 30

**마감 되었습니다**

**ClassDelete.jsp 페이지 실행**  
(뒷장설명)

**ClassUpdate.jsp 페이지 실행**  
★ 클래스 수정 ★

\* 강사이름 (주)sample3

\* 클래스 종류 드로잉

\* 클래스 제목 오프라인 클래스 sample 입니다.

안녕하세요.

\* 클래스 설명

\* 수강료 30000

배송비 3000

\* 수강인원 30

\* 수업유형 오프라인 클래스 ☒ 온라인 클래스 ☐

\* 지역 ex)경기

**클래스 수정하기** **돌아가기**

/ClassRead.jsp?cNum=1&nowPage=1

```
int numParam = Integer.parseInt(request.getParameter("cNum")); //현재 페이지 번호
String nowPage = request.getParameter("nowPage");

ClassBean bean = (ClassBean)session.getAttribute("bean");
int cNum = bean.getCNum(); //현재 페이지 번호
String cTeacher = bean.getCTeacher();
String cUid = bean.getCUid();
String cCategory = bean.getCCategory();
String cTitle = bean.getCTitle(); //클래스 제목
String cContent = bean.getCContent(); //클래스 설명
String cRegDate = bean.getCRegDate(); //클래스 작성날짜

int cPrice = bean.getCPrice();
int cDelivery = bean.getCDelivery();

String cThumbName = bean.getCThumbName(); //썸네일 이름
int cThumbSize = bean.getCThumbSize(); //썸네일 크기저장
String cFileName = bean.getCFileName(); //파일이름
int cFileSize = bean.getCFileSize(); //크기저장

int cMaxStu = bean.getCMaxStu(); //최대 수강인원
int cApplyStu = bean.getCApplyStu();

String Area = bean.getCArea();
String cOnoff = bean.getCOnoff(); //onoff 여부
```

## 2 수강생 인원 마감

수강인원이 최대인원과 동일해지면 "마감 되었습니다" 메시지로 바뀌도록 구현

```
<% if(cApplyStu >= cMaxStu) { %>
<table>
<tr>
<td>
<button type="button" class="buy">마감 되었습니다</button>
</td>
</tr>
</table>
<% } else { %>
<table>
<tr>
<td>
<button type="button" class="buy" id="buy">클래스 신청하기</button>
</td>
</tr>
</table>
<button type="button" class="put" id="put" onClick="location.href='./payment/CartAction.jsp?cNum=<%= cNum %>'">결제</button>
</td>
</tr>
</table>
```

**cApplyStu(수강인원) 요소와 cMaxStu(총인원) 요소의 크기를 비교하여 크거나 같을 때 버튼 기능이 닫힐 수 있도록 구현**

## 3 클래스 수정

버튼 클릭시 ClassUpdate.jsp 페이지로 이동

```
////// 클래스 수정페이지 가기//////
$("#button#classmodbtn").click(function(){
let nowPage = $("#input#nowPage").val().trim();
let cNum = $("#input#cNum").val().trim();

let url = "/Proj_OneDayClass/classBBS/ClassUpdate.jsp?";
url += "cNum=" + cNum + "&nowPage=" + nowPage;
location.href = url;
});
```

```
<input type="hidden" id="cCode" name="cCode" size="27" class="umem">
<input type="hidden" name="cUid" id="cUid" value="<%=uid%>">
<input type="text" id="cTeacher" name="cTeacher" size="27" class="umem" maxlength="20" autofocus placeholder="선생님 이름을 입력하세요." value="<%=cTeacher%>">
</td>
</tr>
</table>

<table>
<tr>
<td>
<input type="button" class="update" id="update" value="수정하기">
</td>
</tr>
</table>

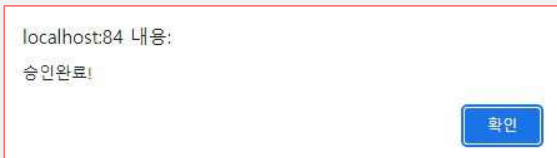
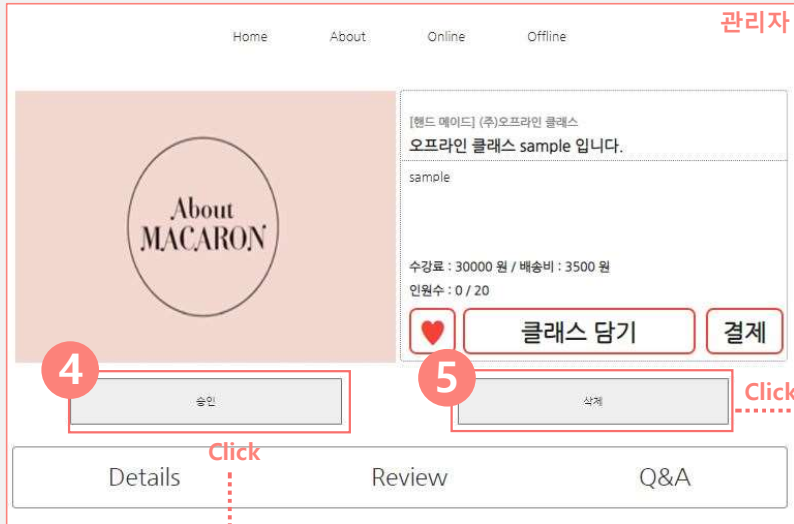
<table>
<tr>
<td>
<input type="button" class="updateFrm" id="updateFrm" value="수정하기" action="ClassUpdateProc.jsp" method="post" >
</td>
</tr>
</table>
```

**1. 현재 페이지의 Query String 값을 url 변수에 담아 cNum, nowPage에 전송**

**2. 현재 페이지의 담긴 bean 요소를 출력 블록에 value 값에 담아 불러옴**

**3. Input 요소에 담긴 내용들을 form요소를 통해 ClassUpdateProc.jsp 페이지로 이동**

## 주요기능 - 클래스 상세 페이지



관리자

```
if(uLevel.equals("3")) {
    int exeCnt = bMgr.approvalBoard(numParam);
    String url = "ClassList.jsp?cNum="+nowPage;
    <script>
        alert("승인완료!");
        location.href = "<%=url%>";
    </script>
} else {
    <script>
        alert("오류가 발생했습니다. 관리자에게 문의하세요.");
        history.back();
    </script>
}
%>
```

if. 관리자 or 자신의 게시물일 경우

localhost:84 내용:  
삭제되었습니다!

확인

else 관리자 or 자신의 게시물이 아닐 경우

localhost:84 내용:  
당신의 게시글이 아닙니다.

확인

```
int exeCnt = bMgr.deleteBoard(numParam);
String url = "ClassList.jsp?cNum="+nowPage;

if(uId.equals(cUid)||uLevel.equals("3")) {
    <script>
        alert("삭제되었습니다!");
        location.href = "<%=url%>";
    </script>
} else {
    <script>
        alert("당신의 게시글이 아닙니다.");
        history.back();
    </script>
}
%>
```

## 4 관리자 승인

버튼 클릭시 ClassApproval.jsp 페이지 실행

→ ClassApproval.jsp 의 approvalBoard(numParam); 메서드 호출

```
public int approvalBoard(int cNum) {
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;

    int exeCnt = 0; // 삭제 데이터 수, DB 삭제가 성공하였는지 여부 판별

    try {
        objConn = pool.getConnection(); // DB연동
        // sql = "delete from classbbs where cnum=?";
        sql = "update classbbs set cStatus=2 where cNum=?";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setInt(1, cNum);
        exeCnt = objPstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }
    return exeCnt;
}
```

1. 현재페이지의 numParm값을 매개변수로 받아
2. Where cNum = ? 번째 게시물의
3. cStatus 상태 값을 2(공개)로 변경.

## 5 강사, 관리자 클래스 삭제

버튼 클릭시 ClassDelete.jsp 페이지 실행

→ ClassDelete.jsp 의 deleteBoard(numParam); 메서드 호출

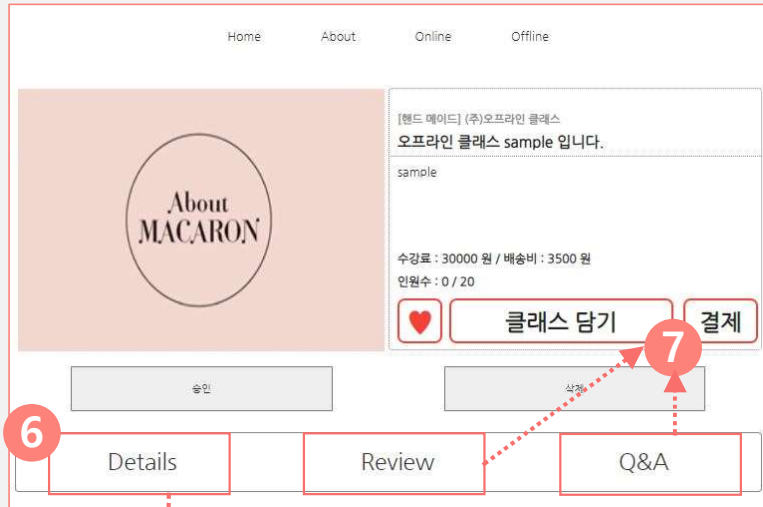
```
public int deleteBoard(int cNum) {
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;

    int exeCnt = 0; // 삭제 데이터 수, DB 삭제가 성공하였는지 여부 판별

    try {
        objConn = pool.getConnection(); // DB연동
        // sql = "delete from classbbs where cnum=?";
        sql = "update classbbs set cStatus=3 where cNum=?";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setInt(1, cNum);
        exeCnt = objPstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }
    return exeCnt;
}
```

1. 현재페이지의 numParm값을 매개변수로 받아
2. Where cNum = ? 번째 게시물의
3. cStatus 상태 값을 3(삭제)로 변경.

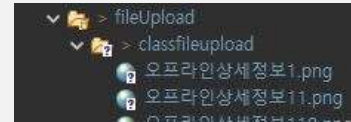
## 주요기능 - 클래스 상세 페이지



6

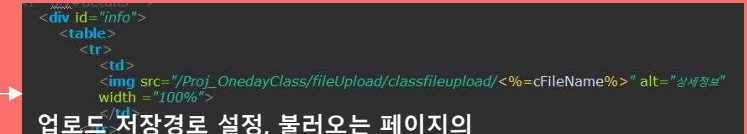
7

상세정보 이미지 출력



## 6 상세정보

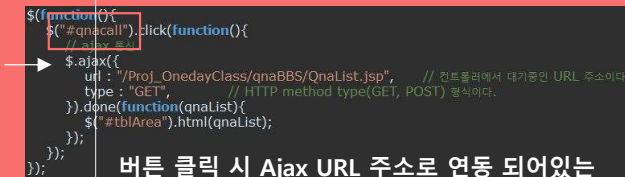
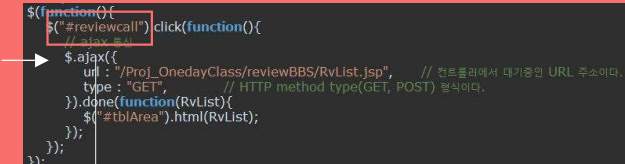
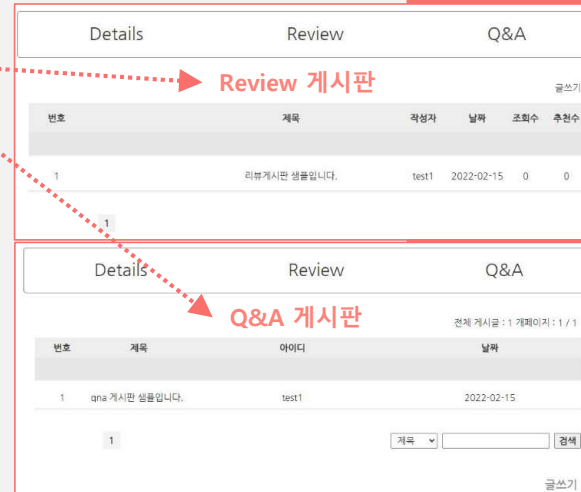
버튼 클릭 시 게시물 상세정보 이미지 출력



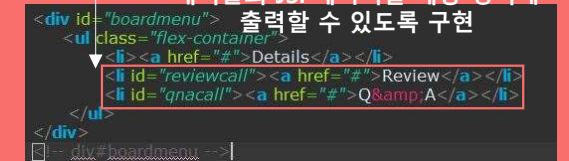
업로드 저장경로 설정, 불러오는 페이지의 cFileName 제목 요소와 저장 경로에 있는 같은 이름의 파일 이미지 출력

## 7 리뷰, Q&amp;A 게시판

버튼 클릭 시 Ajax으로 연동한 게시물 출력



버튼 클릭 시 Ajax URL 주소로 연동 되어있는 게시물의 JSP페이지를 해당 영역에 출력할 수 있도록 구현





## 주요기능 - 클래스 상세 페이지 / 기능

Home About Online Offline

About  
MACARON

[행드 메이드] (주)오프라인 클래스  
오프라인 클래스 sample 입니다.  
sample

수강료: 3000 원 / 배송비: 3500 원  
인원: 10명

8

Click

클래스 담기

결제

Click

삭제

게시물 번호(cNum)과 사용자 아이디(uid) 페이지번호를 참조하여 장바구니 리스트 출력

onClick="location.href='./payment/CartAction.jsp?A  
cNum=<%= cNum %>&uId=<%=uId%>&nowPage=<%=nowPage%>'"

상품명	가격	수량	총계	장바구니 합계
오프라인 클래스 sample 입니다.	100000원	- <input type="text"/> +	100000원	소계: 200000원
오프라인 클래스 sample 입니다.	100000원	- <input type="text"/> +	100000원	배송: 7000원
			총계:	207000원

Home About Online Offline

## 결제정보

---

**수령인 정보**

- \* 이름 : 김민준
- \* 전화번호 : 01073472134
- \* 주소 : 10300      우편번호 : 00000
- \* 결제하기 전에 반드시 입력해주세요.
- \* 10월 01일 ~ 10월 31일
- \* 이메일 : test@naver.com
- ☐ 신용카드를 이용하실까요?    Yes No

수량	상품명	가격	수량	총계
오르바이트 판매스 sample 1입(1개)		30000원	1	30000원
오르바이트 판매스 sample 2 입(1개)		10000원	1	10000원

결제하기

결제하기 버튼 클릭

Payment.jsp

길태형님 환영합니다. | 로그아웃 | 2

insertCard() 메서드 호출

localhost:84 내용:  
장바구니 취소가 되었습니다.

확인

deleteCard() 메서드 호출

길태형님 환영합니다. | 로그아웃 | 1

장바구니 0개

길태형님 환영합니다. | 로그아웃 | 0

장바구니

상품명	가격	수량	총계	장바구니 총계
장바구니가 비었습니다.				소계: 0원
				배송: 0원
				총계: 0원

결제하기

수령인 정보		결제 가격	
수령인이름	김태형	소재:	40000원
전화번호	01012341234	배송:	7000원
주소	10930	총계:	47000원
	경기 파주시 마두리길 22 101동 601호	<div>메인으로</div>	
이메일	test@nave.com		
결제시간	2022년 07월 17일 오전 01:02:44		
상품명	가격	수량	총계
오르래민 총레스 sample 입니다.	30000원	1	30000원
오르래민 총레스 sample) 입니다.	10000원	1	10000원

## 8 클래스 담기

## 버튼 클릭 시 CartAction.jsp 페이지 메서드 호출

```

if (pMgr.checkCart(cartBean)) { // 장바구니에 있을 때,
    if (pMgr.deleteCart(cartBean)) { // 삭제 되었을때
        msg = "장바구니 항목이 삭제됩니다.";
    } else {
        deleteCart() 메서드 호출 시
        msg = "오류가 생겼습니다. 다시 시도해보세요.";
    }
} else {
    if (pMgr.insertCart(cartBean)) { // 입력 되었을때
        insertCart() 메서드 호출 시
        msg = "장바구니에 추가되었습니다.";
    } else {
        msg = "오류가 생겼습니다. 다시 시도해보세요.";
    }
}

int cartCnt = pMgr.cntCart(uid);
session.setAttribute("cartCntKey", cartCnt);

```

```

<%
if (vList.isEmpty()) {
%>
<tr>
<td colspan="4">장바구니가 비었습니다.</td>
</tr>
<%
} else {
} else {
for (int i = 0; i < listSize; i++) {
ClassBean bean = vList.get(i);

int cNum = bean.getNum();
String cTitle = bean.getTitle();
String cUId = bean.getUId();
int cPrice = bean.getPrice();
int cDelivery = bean.getDelivery();
int cMaxStu = bean.getMaxStu();
int cApplyStu = bean.getApplyStu();
int availStu = cMaxStu - cApplyStu;

deliverySum += cDelivery;

%>
<tr>
<td><%=cTitle %></td>
<td><%=cPrice %>원</td>
<td>
<input type="hidden" id="cNum"<%=i %>" name="cNum"<%=i %>" value="<%=cNum %>"
<input type="hidden" id="cUId"<%=i %>" name="cUId"<%=i %>" value="<%=cUId %>"
<input type="hidden" id="cTitle"<%=i %>" name="cTitle"<%=i %>" value="<%=cTitle %>"
<input type="hidden" id="cPrice"<%=i %>" name="cPrice"<%=i %>" value="<%=cPrice %>"
<input type="hidden" id="availStu"<%=i %>" name="availStu"<%=i %>" value="<%=availStu %>"
|
<span class="btnminus" onclick="btnMinus(<%=i %>)"></span>
<input type="text" name="pQty"<%=i %>" id="pQty"<%=i %>" size="2"
maxlength="4" class="pQty" value="1" max="<%=availStu %>" min="1"
<span class="btnplus" onclick="btnPlus(<%=i %>)"></span>
</td>
<td>
<span id="res"<%=i %>"><%=cPrice %></span>원
<input type="hidden" name="resVal"<%=i %>" id="resVal"<%=i %>" value="<%=cPrice %>"
</td>
</tr>

```

## 디버깅 작업 - 버그 확인

디버깅					
폴더명	JSP 파일명	동작내용 (굵은글씨확인완료)	버그	확인사항	보완사항
adminBBS	AdminPage.jsp	관리자 페이지		<ul style="list-style-type: none"> <li>등록 관리 메뉴 작동완료</li> <li>클래스관리 메뉴 작동완료</li> <li>문의게시판 메뉴 작동완료</li> <li>리뷰게시판 메뉴 작동완료</li> <li>게시물 삭제 오류 발생,,</li> <li>여기선 Enter 이벤트 정상작동함</li> <li>페이징 기능 확인</li> <li>글내용 정상출력</li> <li>글목록 버튼 작동 확인</li> <li>삭제 오류 (AdminRvDelete.jsp 내용확인)</li> <li>메시지는 뜨는데 삭제는 안됨. rStatus에서 변경완료</li> </ul>	상세/확인사항
adminBBS	AdminQnaDelete.jsp	관리자 Qna 게시물 삭제	삭제 오류 발생		
adminBBS	AdminQnaList.jsp	관리자 Qna 리스트			
adminBBS	AdminQnaRead.jsp	관리자 Qna 상세 페이지	삭제 안됨		
adminBBS	AdminRvDelete.jsp	관리자 Review 게시물 삭제	삭제 안됨		
adminBBS	AdminRvList.jsp	관리자 Review 리스트	리스트상 이미지가 안보임		
adminBBS	AdminRvRead.jsp	관리자 Review 상세 페이지	삭제 안됨		

## 디버깅 작업 - 확인

- 폴더명 / JSP 파일명으로 분류해 해당 페이지의 기능을 직접 사용해보고 버그를 찾아 동작내용 / 버그내용 / 확인사항/ 보완사항 으로 나눠 정리하였습니다.
- 버그가 있는 페이지엔 빨간색 굵은 글씨, 이상 없는 페이지는 검정 굵은 글씨로 표기하여 정리하였습니다.

디버깅 작업 - 버그 수정

수정완료

<p>답변을 답변글 자동 삭제되고 qStatus 값 3으로 바뀌는거 확인</p> <p>첨부파일 다운로드 되는거 확인</p> <p>다른 사람 비공개글 제목 "비밀글입니다." 확인 qStatus 값 1 확인</p> <p>다른 사람 비공개답변 제목 "비밀글입니다." 확인 qStatus 값 2 확인</p> <p>비공개글 클릭 안되는거 확인</p> <p>다른 사람 글이면 수정 버튼 삭제버튼 안나오는거 확인</p> <p>페이지 기능 확인</p> <p>검색기능 동작 확인</p> <p>아이디 검색기능 동작 확인</p> <p>내용 검색기능 동작 확인</p> <p>검색부분에 내용입력하고 Enter누르면 오류뜸</p> <p>공개글, 비공개글 동작여부 확인</p> <p>데이터 전송 확인(DB 확인)</p> <p>글 내용 정상출력 확인</p> <p>리스트 버튼 동작</p> <p>답변 글 목록 완료, 비공개시 2 공개시 1 qStatus 값 확인</p> <p>학생 등급의 회원도 q&amp;a 답변을 달 수 있나요?(by.Sekdoltae)</p> <p>작성자로 자기 게시물에 답변을 작성할 수 있나요??(by.AceWaterZero)</p> <p>답변 글 삭제 안됨, 삭제버튼 클릭시 qStatus 값 변동없음</p> <p>글 내용 전송완료</p> <p>일반글 내용 수정 확인 답변글 수정 확인</p> <p>데이터 전송 확인(DB 확인)</p>	<p>답변 작성 페이지 내용 입력칸 박스가 너무 작음</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------

## 피드백

## 디버깅 작업 - 수정 / 피드백

- 버그를 수정한 페이지는 정상동작 확인 후 파란색 글씨로 기재 하였습니다.
- 보완사항 항목은 해당 페이지의 부족하거나 추가되었으면 하는 사항을 기재 하여 업데이트 하는 방식으로 구현하였습니다.



감사합니다