



원데이 클래스에
오신 것을 환영합니다.

Contents

01



개발자 소개

02



개발개요

03



개발과정

04



ERD 모델링 과정

05



주요기능설명

개발자 소개

Gil Tae
Hyeung

CONTACT

Tel. 010-xxxx-xxxx

Email. r160xxxx@gmail.com

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

One. Where did I make

[이젠 아카데미] [빅데이터전문가]빅데이터분석 머신러닝활용(자바, 파이썬)

Two. What did I make

Team Project

일상에서 취미를 만나는 가장 쉬운 방법 **OnedayClass** 플랫폼 사이트

Assigned Task

로그인&로그아웃 / 회원가입 / 회원정보수정 / 회원탈퇴 / 마이페이지
클래스 게시판 / 디버깅 / Github 종합

Three. How did I make

With Java / JSP / Spring Framework / MySQL

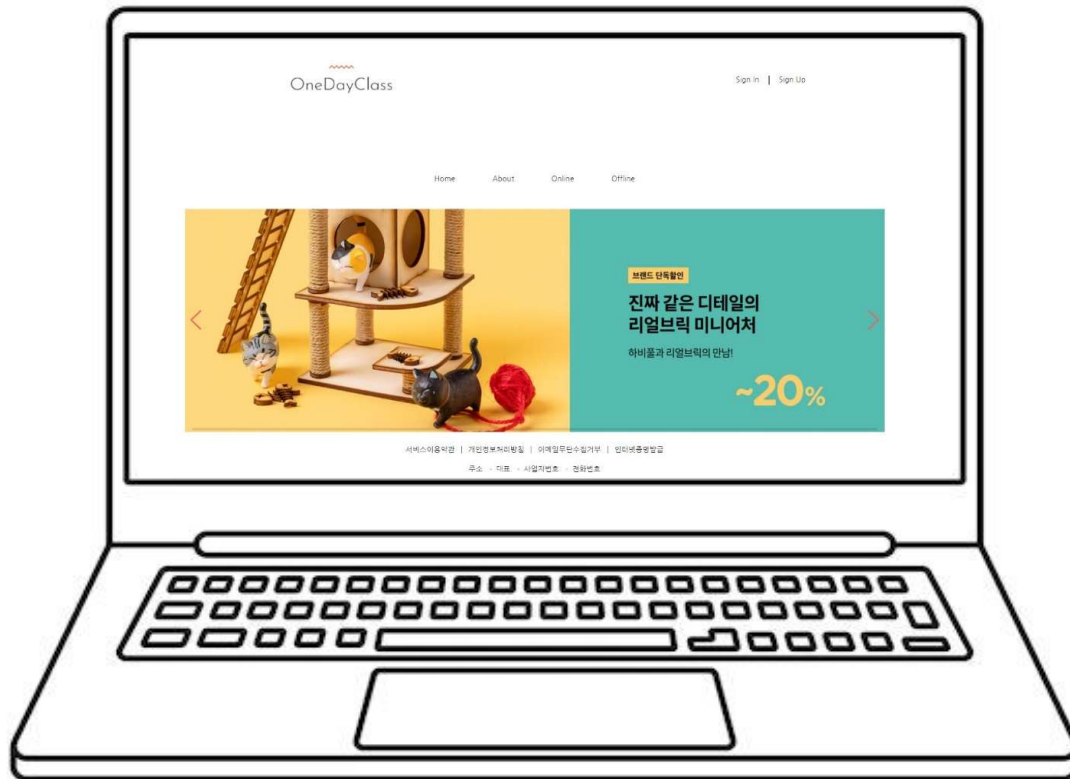
HTML5 / CSS / JavaScript / JQuery / Ajax

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 / SpringFramework / MySQL
- HTML / CSS / JS / jQuery
- VSC / Eclipse / GitHub

깃 허브 링크

- Jsp Model1(기반 설명)
https://github.com/Giltaehyeong/Proj_OneDayClass
- Spring Framework(maven) Migration(로그인 파트까지만)
https://github.com/Giltaehyeong/Proj_OneDayClass_Spring
- 동작 영상
<https://youtu.be/kDs0r3Ea3LI>

Front End



Back End



DataBase

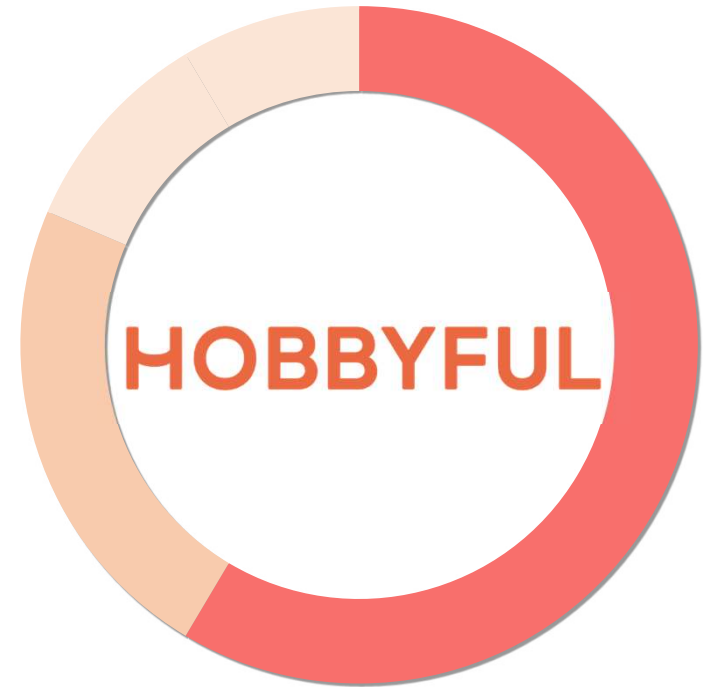




아이디어스



모카클래스



하비풀

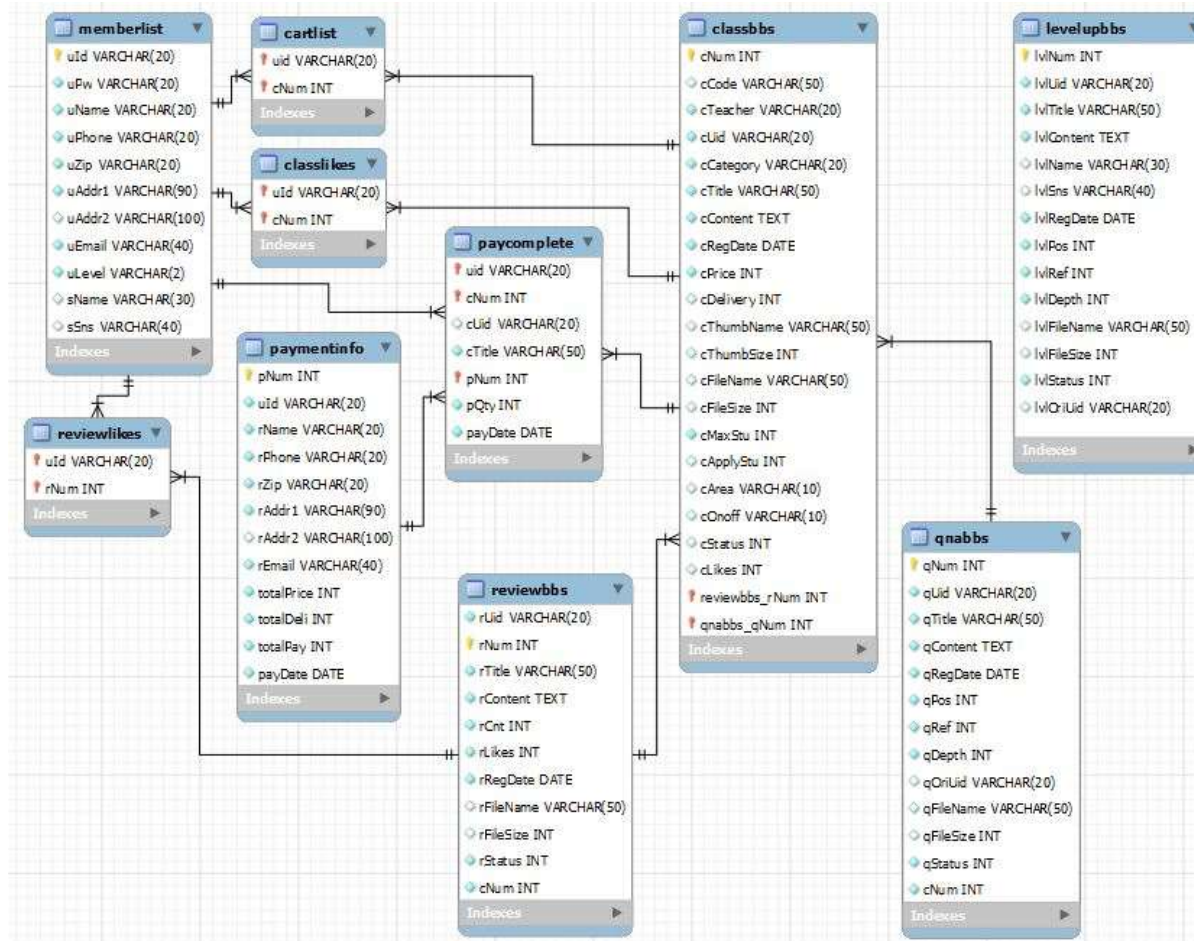
벤치마킹 사이트

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

PART 3 **개발과정 - 프로젝트 일정**

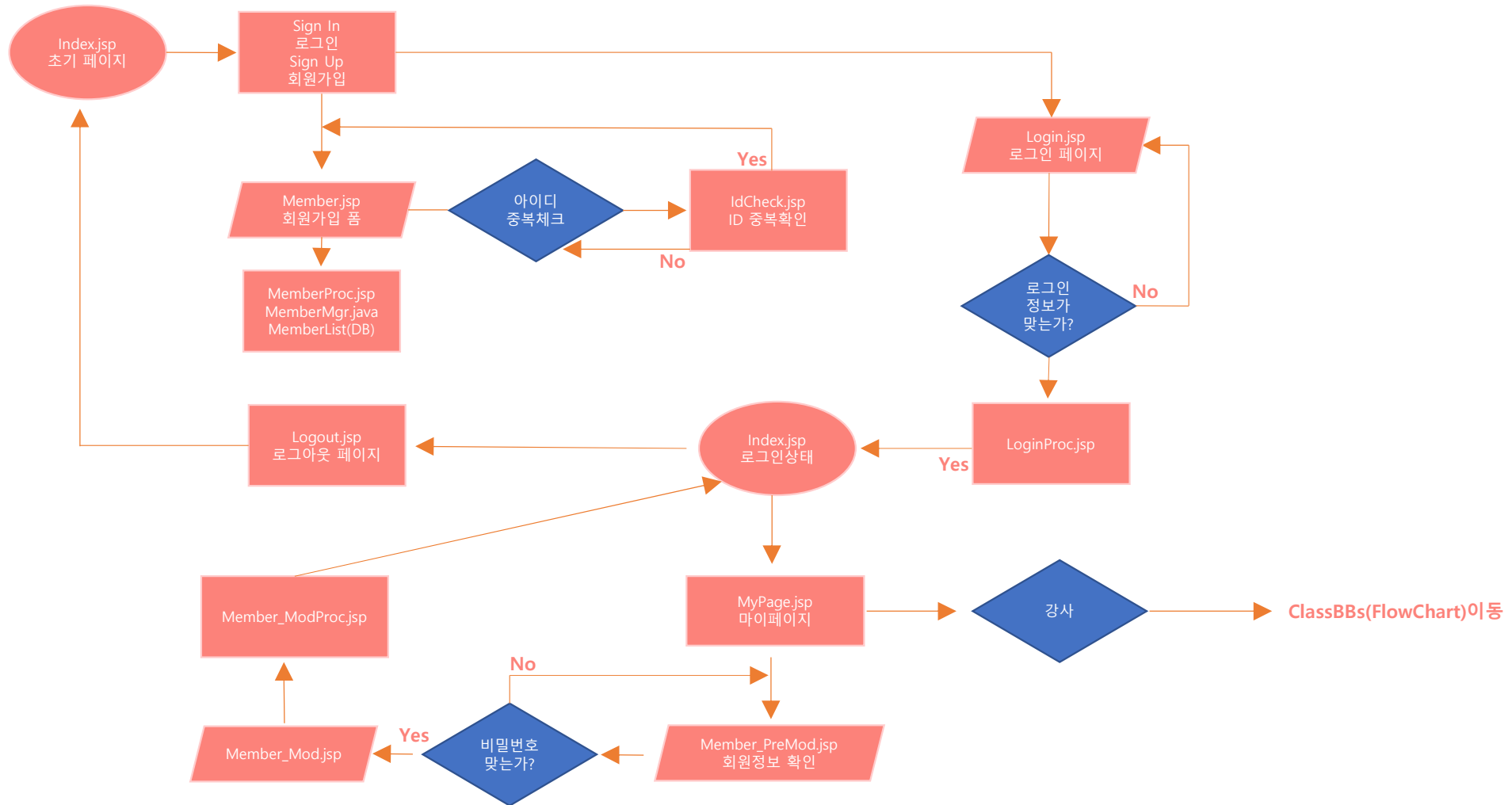
SUN	MON	TUE	WED	THU	FRU	SRI
	27	28	29	30	31	1
	기획		DB 설계 및 ERD 작성			
2	3	4	5	6	7	8
	메인페이지		로그인 및 회원가입 구현, DB연동			
9	10	11	12	13	14	15
	게시판 구현(등업, 리뷰, 문의, 클래스 게시판)					
16	17	18	19	20	21	22
	기능 페이지 구현 (관리자, 마이페이지, 클래스, 결제관련)				디버깅	

ERD 모델링

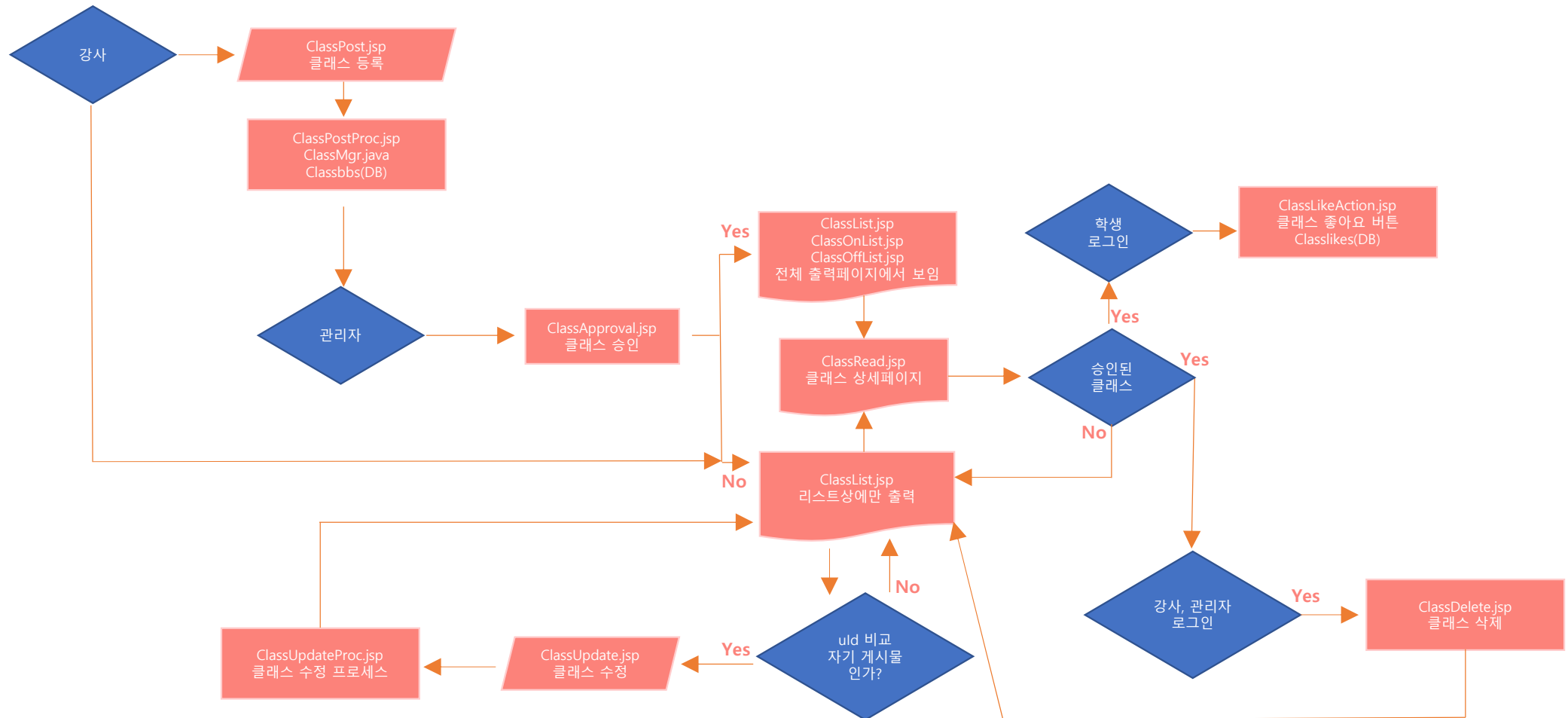


10개 테이블 유기적 연결

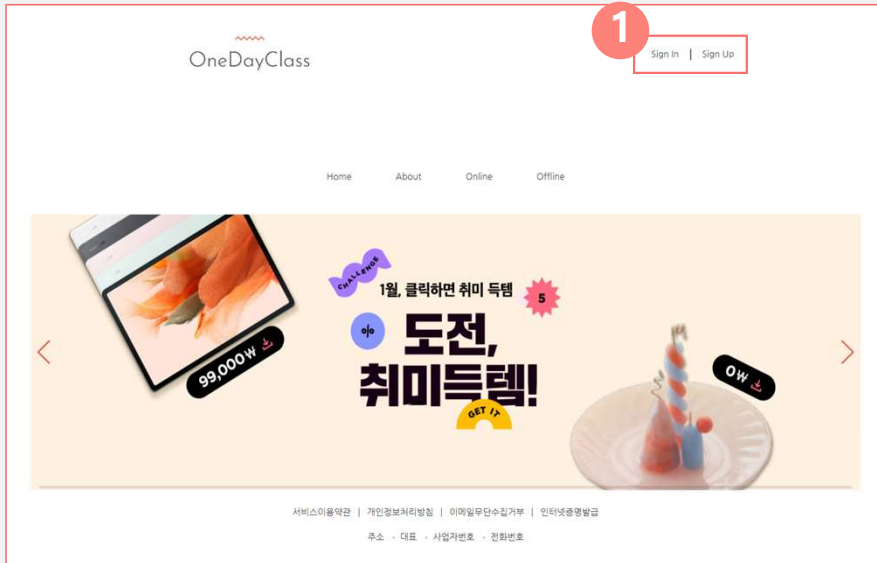
FlowChart - Memberlist



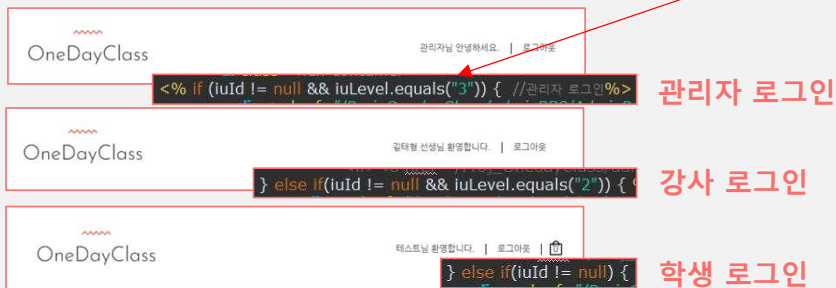
FlowChart - Classbbs



주요기능 - 메인 페이지



등급별 로그인 세션



관리자 로그인

강사 로그인

학생 로그인

1 Sign In / Logout

버튼 클릭 시 로그인 페이지로 이동

로그인

<form action = "LoginProc.jsp">

아이디

비밀번호

로그인 회원가입

Login.jsp

↓ LoginProc.jsp로 데이터 전송.

```
boolean res = mMgr.loginMember(uid, uPw);
if (res) {
    String[] uStat = mMgr.getMemberStat(uid);
    int cartCnt = pMgr.cntCart(uid);
    session.setAttribute("idKey", uid);
    session.setAttribute("nameKey", uStat[0]);
    session.setAttribute("levelKey", uStat[1]);
    session.setAttribute("cartCntKey", cartCnt);
    //msg = "로그인에 성공 하였습니다.";
    response.sendRedirect("/Proj_OneDayClass/Index.jsp");
} else {
    %>
}
```

Boolean형 res 변수에 메서드 실행 시 memberlist 테이블에서 uid, uPw가 조회된다면 next() 메서드를 사용해서 Select 결과의 존재 여부를 확인하고 Flag 값에 담아 반환

이후 res 값에 true값이 반환될 경우 세션에 값을 저장하기 위해 Session객체에 setAttribute("키", 값) 담아서 Index.jsp 페이지로 Redirect.

길태행님 환영합니다. | 로그아웃

```
public boolean loginMember(String uid, String uPw) {
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;

    String sql = null;
    boolean flag = false;

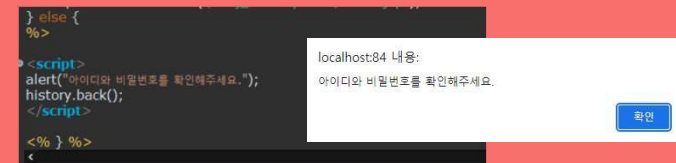
    try {
        objConn = pool.getConnection();
        sql = "select uid from memberlist where uid=? and uPw=?";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setString(1, uid);
        objPstmt.setString(2, uPw);

        flag = objPstmt.executeQuery().next();

    } catch (Exception e) {
        System.out.println("SQL 이슈4 : " + e.getMessage());
    } finally {
        pool.freeConnection(objConn, objPstmt, objRs);
    }

    return flag;
}
```

이후 res 값에 false값이 반환될 경우 코드 블록 밖의 스크립트 문을 실행시킨다.



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" %>
2 pageEncoding="UTF-8"%>
3
4 session.invalidate(); // 세션으로 로그인할때만
5 // invalidate() 메서드로 세션 소멸하는 것이 로그아웃이다.
6
7 <script>
8 //alert("로그아웃되었습니다.");
9 location.href="/Proj_OneDayClass/Index.jsp";
10 </script>
```

session.invalidate(); 메서드를 사용하여 세션을 소멸시킴.

Logout.jsp

주요기능 - 우편번호찾기

3 우편번호찾기/ 다음 API

* 전화번호

전화번호를 입력해주세요.

* 주소

우편번호

주소

상세주소를 입력해주세요.

* 이메일

이메일 주소를 입력해주세요.

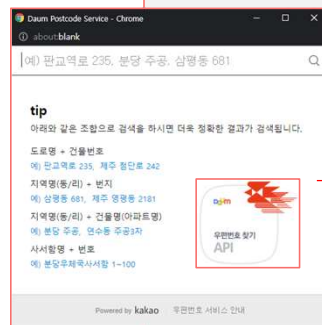
☐ 전부 동의합니다.

☐ 만 14세 이상입니다.

☐ 이용약관에 동의합니다.

☐ 개인정보수집·이용에 동의합니다.

동의하고 가입하기



서버에서 가져온 값이 value 값에 입력됨.

* 주소

10930

경기 파주시 마무리길 22

101동 601호

```

//////////////// 우편번호 주소 확인 시작 //////////////////
function zipCheck() {
    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

            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById("uZip").value = data.zonecode;
            document.getElementById("uAddr1").value = roadAddr;
            document.getElementById("uAddr2").focus(); // 상세주소로 포커스

        }).open();
    }
    우편번호 찾기 누르면 위 function 실행
    $("#zipBtn").click(function(){
        zipCheck();
    });
    ////////////////// 우편번호 주소 확인 끝 //////////////////

```

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>
</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>

```

PART 5 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 마무리길 22

101동 601호

* 이메일 r1605866@naver.com

정보수정 다시쓰기 메인으로 로그인아웃

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

게시판 리스트 출력을 위해 Vector 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();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }
    return vList;
}
```

Bean 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 5 주요기능 - 마이페이지 (클래스 관리)



★ 클래스 생성 ★

상호명: 테스트

클래스 종류: **핸드메이드** <카테고리 분류 추가설명> Click

클래스 제목: 테스트 클래스

클래스 설명:

수강료: 100000

배송비: 3000

병네일사진: 파일 선택

상세정보사진: 파일 선택

수강인원: 10

수업유형: **오프라인 클래스** * 온라인 클래스

지역: 경기

클래스 생성하기

28페이지 추가설명

OneDayClass 김태형 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

4 클래스 생성 페이지

버튼 클릭 시 클래스 등록페이지로 이동

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

강사 이상 등급만 접근 가능.

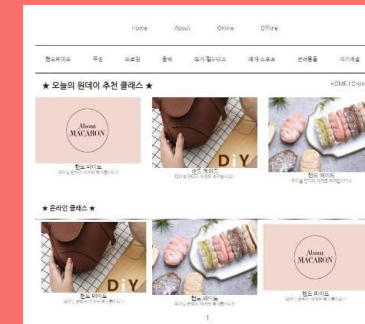
ClassPostProc.jsp / InsertBoard() 메서드를 호출하여 DB에 데이터 입력

5 클래스 생성 / 온&오프 체크

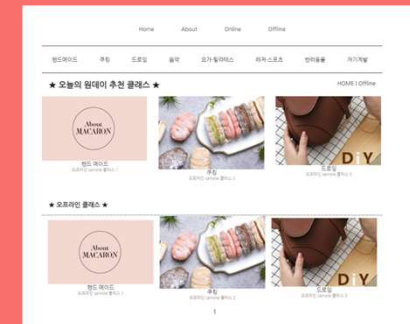
수업 유형 선택 시 해당 갤러리 리스트에 게시물 출력

↓ 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

* 클래스 종류 핸드메이드

* 클래스 제목

* 클래스 설명

* 수강료

* 배송비

* 썸네일사진

파일 선택 54.png

* 상세정보사진

파일 선택 211.png

* 수강인원

* 수업유형 오프라인 클래스 온라인 클래스

* 지역

다중 업로드 구현

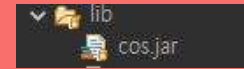
```
multi = new MultipartRequest(req, path, // 저장소
    maxSize, // 파일크기
    encType, new DefaultFileRenamePolicy());
```

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

클래스 생성하기

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

```
// 오버릭트 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.getFileSystemName("cThumbName") != null) {
    cThumbName = multi.getFileSystemName("cThumbName");
    cThumbSize = (int) multi.getFile("cThumbName").length();
}

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

    // 저장소1
    // 저장소2

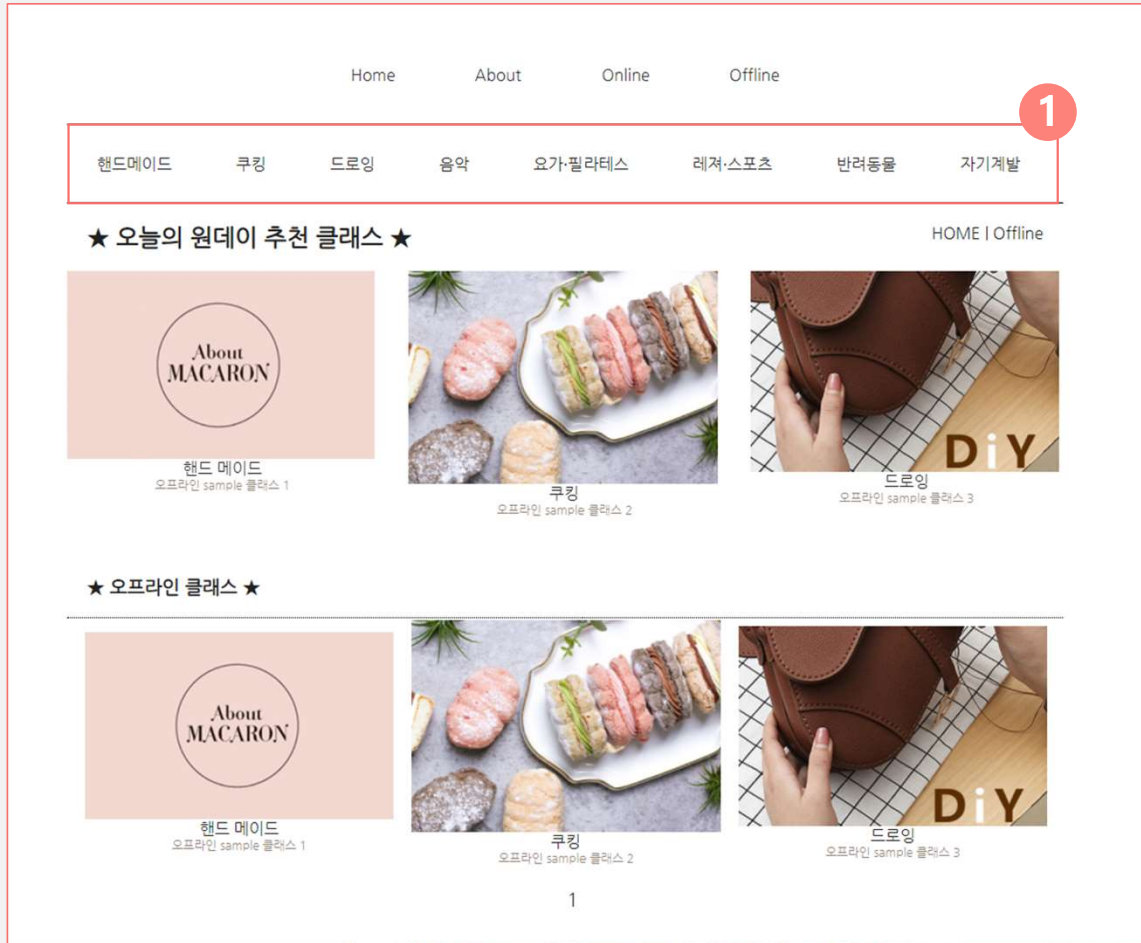
    pstmt = objConn.prepareStatement("insert into classbbs (cCode, cTeacher, cUId, cCategory, cContent, cRegDate, cPrice, cDelivery, "
        + "cThumbName, cThumbSize, cFileName, cFileSize, "
        + "cMaxStu, cApplyStu, cOnoff, cArea, cStatus, cLikes) values ("
        + "?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, "
        + "?, ?, ?, ?, ?, ?, ?, ?)");

    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("cContent"));
    pstmt.setString(6, multi.getParameter("cRegDate"));
    pstmt.setString(7, multi.getParameter("cPrice"));
    pstmt.setString(8, multi.getParameter("cDelivery"));
    pstmt.setString(9, cThumbName);
    pstmt.setInt(10, cThumbSize);
    pstmt.setString(11, cFileName);
    pstmt.setInt(12, cFileSize);
    pstmt.setInt(13, Integer.parseInt(multi.getParameter("cMaxStu")));
    pstmt.setInt(14, Integer.parseInt(multi.getParameter("cApplyStu")));
    pstmt.setInt(15, Integer.parseInt(multi.getParameter("cOnoff")));
    pstmt.setInt(16, Integer.parseInt(multi.getParameter("cStatus")));
    pstmt.setInt(17, Integer.parseInt(multi.getParameter("cLikes")));

    objConn.executeUpdate(pstmt);
    pstmt.close();
}
```

작성 날짜를 기입하기 위해
SQL now() 함수 사용

주요기능 - 클래스 게시판



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')> 쿠킹</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

```
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);

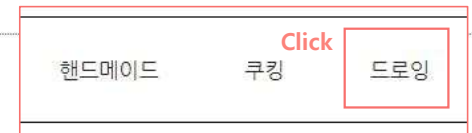
        while (objRs.next()) {
            ClassBean cBean = new ClassBean();
            cBean.setCNum(objRs.getInt("cNum"));
            cBean.setCCategory(objRs.getString("cCategory"));
            cBean.setCStatus(objRs.getInt("cStatus"));
            cBean.setCOnoff(objRs.getString("cOnoff"));
            cBean.setCTitle(objRs.getString("cTitle"));
            cBean.setCImage(objRs.getString("cImage"));
            cBean.setCDate(objRs.getString("cDate"));
            cBean.setCWriter(objRs.getString("cWriter"));
            cBean.setCReply(objRs.getInt("cReply"));
            cBean.setCReplyDate(objRs.getString("cReplyDate"));
            cBean.setCReplyContent(objRs.getString("cReplyContent"));
            cBean.setCReplyStatus(objRs.getInt("cReplyStatus"));
            cBean.setCReplyDate(objRs.getString("cReplyDate"));
            cBean.setCReplyContent(objRs.getString("cReplyContent"));
            cBean.setCReplyStatus(objRs.getInt("cReplyStatus"));
            vList.add(cBean);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        objConn.close();
        objPstmt.close();
        objRs.close();
    }

    return vList;
}
```

getBoard() 메서드의 매개변수를 통해 sql문 조회 시

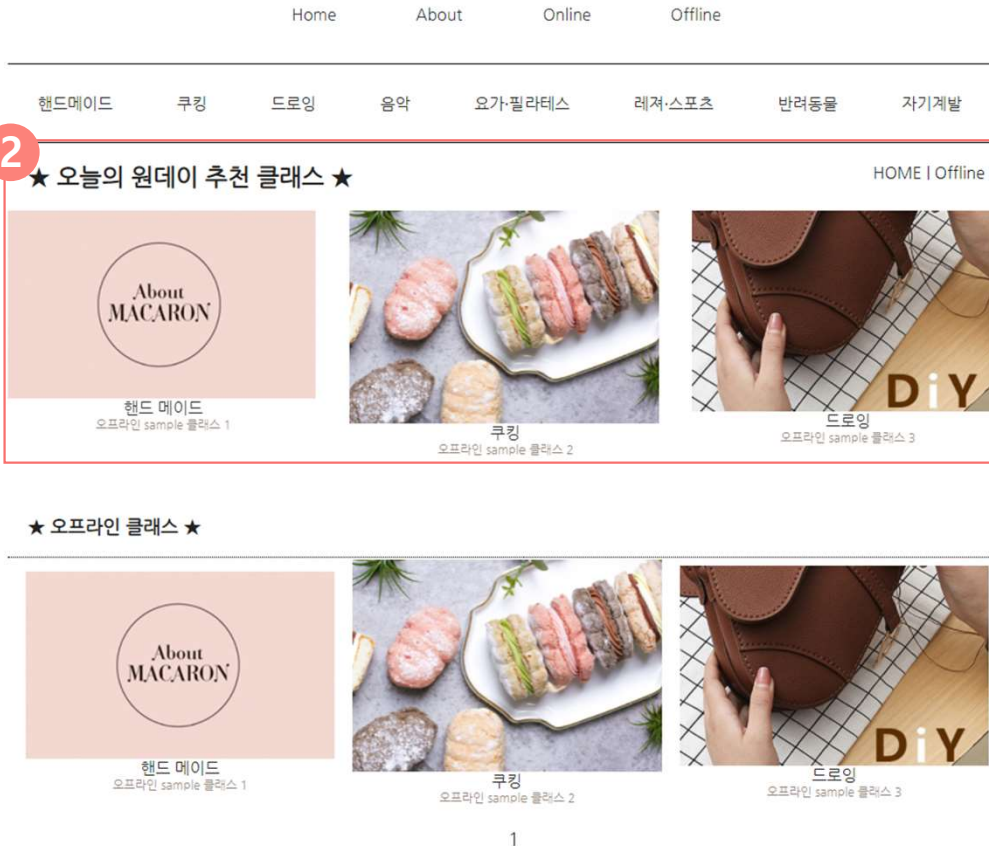
select 구문의 where cCategory = ? 전달되어 카테고리 번호에 해당되는 게시물만 출력하도록 구현

★ 오프라인 클래스 ★



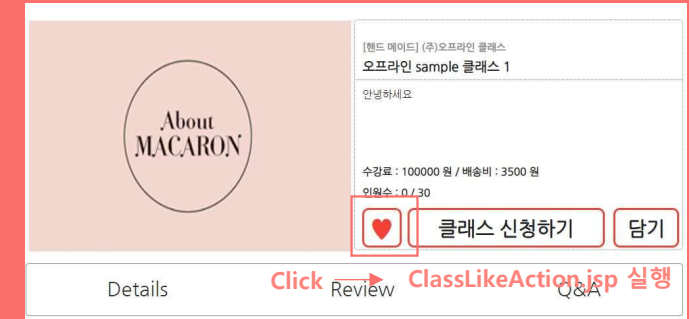
3번

주요기능 - 클래스게시판



2 추천 클래스 List 출력

인기가 많은 게시물부터 차례대로 출력되도록 구현



↓ cLike 컬럼 카운트 증가



<찜하기 추가설명>

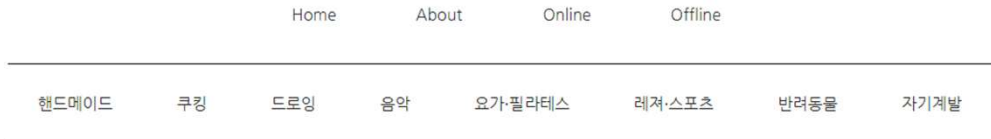
Click

```
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 - 클래스 게시판



★ 오늘의 원데이 추천 클래스 ★

HOME | Offline



핸드메이드
오프라인 sample 클래스 1



쿠킹
오프라인 sample 클래스 2



드로잉
오프라인 sample 클래스 3

3

★ 오프라인 클래스 ★



핸드메이드
오프라인 sample 클래스 1



쿠킹
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

<페이징 설명 뒷장>

주요기능 - 클래스 게시판 / 페이지징 기능

④ 게시판 페이지징 기능 구현

★ 오프라인 클래스 ★

출력 게시글 수 : 9개



1 Block

```
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 T1 where... order by ... limit start, end;
int listSize = 0; // 1페이지에서 보여주는 데이터 수
```

1 2 3 4 5 >

블록당 표시되는 페이지 수 : 5개

```
if (request.getParameter("nowPage") != null) {
    nowPage = Integer.parseInt(request.getParameter("nowPage"));
    // 전달받은 값의 페이지번호를 숫자로 만들어서 초기화 할 수 있음.
    start = (nowPage * numPerPage) - numPerPage;
    //      1      *      9      -      9
    //      0      0
    // 현재페이지 * 페이지당 출력하는 데이터 수 - 페이지당 출력하는 데이터 수
    // 데이터가 50개 => num : 9 8 7 6 5
    //                0 1 2 3 ~ 8
    // 페이지당 출력할 데이터 수 9개 => 1 ~ 9 이게 첫번째 페이지
    // 현재 페이지가 1페이지라면
    // 2페이지 10 ~ 18
    // 3페이지 19 ~ 27

    end = numPerPage;
} nowPage가 null이라면.
```

시작 : (현재 페이지 * 출력 데이터 수) - 출력데이터수 = 0

그래서 vList가 Null이라면
vList.isEmpty() list의 null값을
체크해주는 메서드를 사용하여
해당 div 요소 출력.

```
Vector<ClassBean> vList = null;
```

vList값에 null을 집어넣음.

```
<%
if (vList.isEmpty()) {
    %>
    <div>
    <div><%= "게시물이 없습니다." %></div>
    </div>
    <%
} else {
```

주요기능 - 클래스 게시판 / 페이징기능

근데 vList가 Null이 아니라면

listSize -> 페이지에서 보여주는 데이터 수만큼 div 영역에 게시물을 출력해준다.



레저·스포츠
오프라인 클래스 sample 입니다.

```

<%
for (int i = 0; i < listSize; i += 3) {
%>
<tr>
<%
for (int j = 0; j < 3; j++) {
    ClassBean bean = vList.get(i + j);
    int cNum = bean.getcNum();
    //게시글 넘버 두번째 for문이 돌 때 안에 있는 게시물을 3개 출력한다.
    String cThumbName = bean.getcThumbName();
    String cCategory = bean.getcCategory();
    //카테고리
    String cTitle = bean.getcTitle();
    //제목
    int cStatus = bean.getcStatus();
    //글 속성, 공개 비공개 여부
    String cOnoff = bean.getcOnoff();
    //클래스 on(N), off(Y) 여부
    if (cStatus == 2) {
%>
<td onclick="read('<%=cNum%>', '<%=nowPage%>')">
    <!--
-->

```

첫번째 for문이 돌 때 i 값에 3을 더하고

두번째 for문이 돌 때 안에 있는 게시물을 3개 출력한다.

그래서 i값과 j값을 더한값이 출력한 게시물
의 -1과 같다면 i를 j에 대입하고 break;

```

%>
<%
if ((i + j) == (listSize - 1)) {
    i += j;
    break;
}
}
%>
</tr>
<%
}
}

```

주요기능 - 클래스 게시판 / 페이징기능

```
totalRecord = bMgr.getTotalCount(cCategorySel);
```

```

//////////////////////////////////총 게시물 수(List.jsp) 시작//////////////////////////////////
public int getTotalCount(String cCategorySel) {

    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;
    int totalCnt = 0; 총 게시물 수

    try {
        objConn = pool.getConnection(); // DB연동

        if (cCategorySel.equals("null") || cCategorySel.equals("")) {
            sql = "select count(*) from classBbs where cStatus<3";
            objPstmt = objConn.prepareStatement(sql);
        } else {
            sql = "select count(*) from classBbs ";
            sql += "where cCategory = ? AND cStatus<3";
            objPstmt = objConn.prepareStatement(sql);
            objPstmt.setString(1, cCategorySel);
        }

        objRs = objPstmt.executeQuery();

        if (objRs.next()) {
            totalCnt = objRs.getInt(1);
        }

    } catch (Exception e) {
        System.out.println("SQL이슈4 : " + e.getMessage());
    } finally {
        pool.freeConnection(objConn, objPstmt, objRs);
    }

    return totalCnt; 총 게시물 수 반환
}
//////////////////////////////////총 게시물 수(List.jsp) 끝//////////////////////////////////

```

```
totalRecord = bMgr.getTotalCount(cCategorySel); //전체 데이터 수 반환
```

```

totalPage = (int) Math.ceil((double) totalRecord / numPerPage);
nowBlock = (int) Math.ceil((double) nowPage / pagePerBlock);
totalBlock = (int) Math.ceil((double) totalPage / pagePerBlock);

```

전체 페이지 개수를 계산하는 부분입니다.
 만약 122개의 레코드가 있고 122/10의 결과 값인 12.2를
 소수점을 올림시켜서 13개의 페이지가 만들어 집니다.
 13번째 페이지에선 2개의 레코드를 갖고 있습니다.

```
totalRecord = bMgr.getTotalCount(cCategorySel); //전체 데이터 수 반환
```

```

totalPage = (int) Math.ceil((double) totalRecord / numPerPage);
nowBlock = (int) Math.ceil((double) nowPage / pagePerBlock);
totalBlock = (int) Math.ceil((double) totalPage / pagePerBlock);

```

현재 블록을 계산하는 부분인데, 현재 블록은 현재 페이지를
 블록 당 페이지 수로 나누어서 계산하면 됩니다. totalPage
 와 마찬가지로 소수점일 경우 올림합니다.

```
totalRecord = bMgr.getTotalCount(cCategorySel); //전체 데이터 수 반환
```

```

totalPage = (int) Math.ceil((double) totalRecord / numPerPage);
nowBlock = (int) Math.ceil((double) nowPage / pagePerBlock);
totalBlock = (int) Math.ceil((double) totalPage / pagePerBlock);

```

전체 블록 수를 계산하는 부분입니다. 전체 블록의 개수는
 전체 페이지의 개수를 블록 당 페이지수로 나누어서 계산하
 면 됩니다. 원리는 전체 페이지 개수를 구하는 방법과 같습
 니다.

주요기능 - 클래스 게시판 / 페이징기능

```

<!-- 페이징 시작 -->
<td colspan="2" id="pagingTd">
  <%
    int pageStart = (nowBlock - 1) * pagePerBlock + 1;
    int pageEnd = (nowBlock < totalBlock) ? pageStart + pagePerBlock - 1 : totalPage;
  %>
  <span class="moveBlockArea" onclick="moveBlock('<%=nowBlock - 1%>', '<%=pagePerBlock%>')"> << </span>
  <%
  %>
  } else {
  %>
  <span class="moveBlockArea"> </span>
  <%
  %>
  }
  // 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 For 페이지 나누기용 페이지 번호 출력 끝

  if (totalBlock > nowBlock) { // 다음 블록이 남아 있다면
    <span class="moveBlockArea" onclick="moveBlock('<%=nowBlock + 1%>', '<%=pagePerBlock%>')"> > </span>
  } else {
    <span class="moveBlockArea"> </span>
  }
}
// End if#01_totalPage
%>

```

하단페이지 시작번호

하단 페이지 끝번호.

블록마다 시작되는 첫 페이지와 마지막 페이지 관련 작업

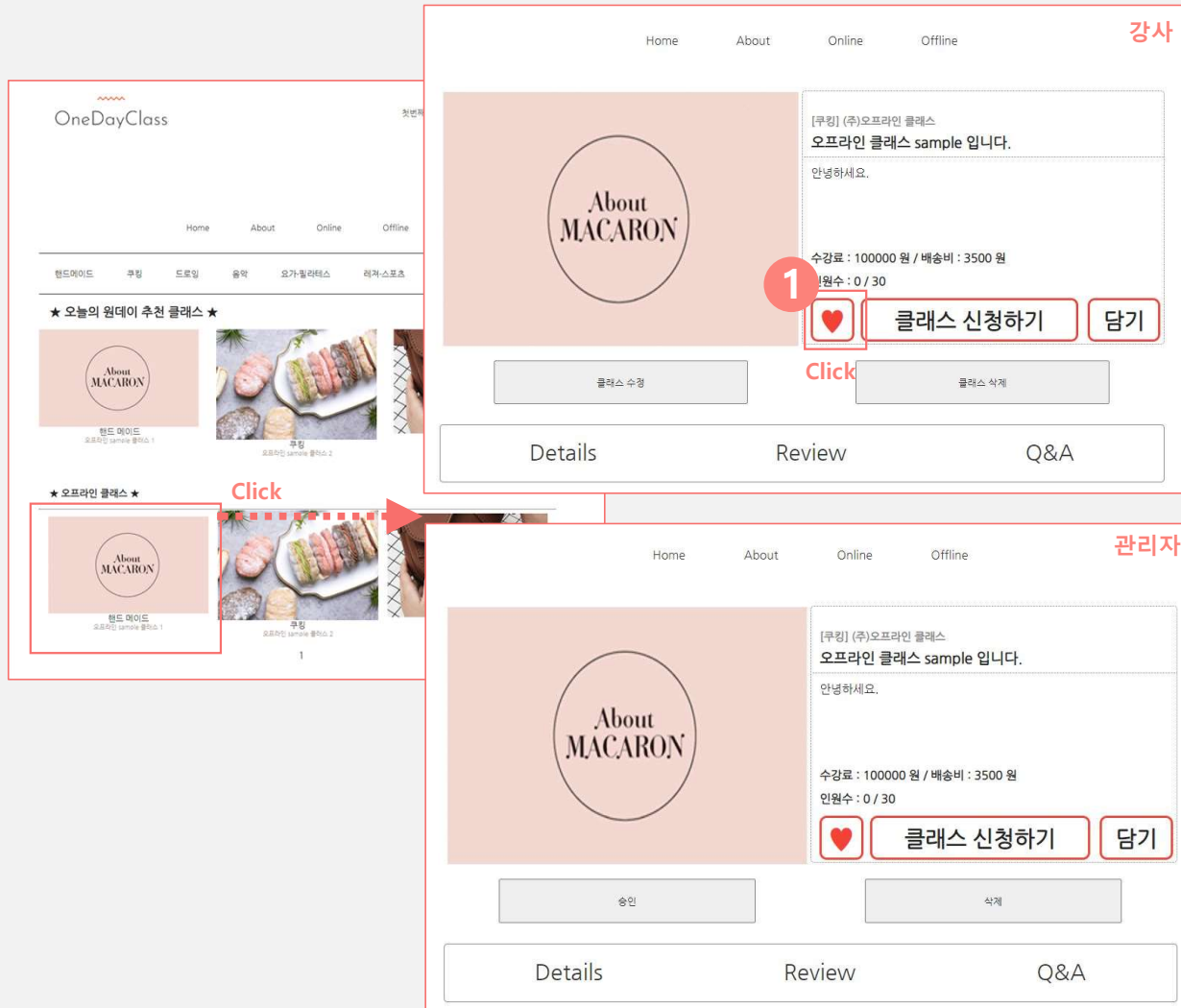
블럭당 5페이지 출력 =>

	pageStart	pageEnd
1블럭	1	5
2블럭	6	10

현재 표시되는 블록수가 전체 블록수보다 많다면

> Next 다음 페이지 생성

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(?,?)";
        objPstmt = objConn.prepareStatement(sql);
        objPstmt.setInt(1, classLikeBean.getUid());
        objPstmt.setInt(2, classLikeBean.getCNum());
        result = objPstmt.executeUpdate();
    } 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

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.setUid(uid);

LikeBean.setCNum(cNum);

int result = bMgr.UpdateLike(LikeBean);

if(result == 0){

<script>

alert("이미 찜하신 강의입니다.");

history.back();

</script>

<% 실행된 경우

Result 값이 1인 경우

중복 데이터로 인식

else{

<script>

alert("찜완료 ^.^");

//location.href = history.back(-1);

<script>

bMgr.InsertLikeCnt(cNum);

<script>

history.back();

</script>


<% 실행된 경우

Result 값이 1인 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

ClassDelete.jsp 페이지 실행 (뒷장설명)

Details Review Q&A

ClassUpdate.jsp 페이지 실행 ★ 클래스 수정 ★

* 강사이름 (주)sample3

* 클래스 종류 드로잉

* 클래스 제목 오프라인 클래스 sample 입니다.

* 클래스 설명

* 수강료 30000

* 배송비 3000

* 수강인원 30

* 수업유형 오프라인 클래스 ☒ 온라인 클래스 ☐

* 지역 ex)경기

클래스 수정하기 돌아가기

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

마감 되었습니다

ClassDelete.jsp 페이지 실행
(뒷장설명)

/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 클래스 번호
String cTeacher = bean.getCTeacher();
String cUid = bean.getCUid();
String cCategory = bean.getCCategory(); //클래스 카테고리
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 담힐 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" value="수정" />
</td>
</tr>
</table>

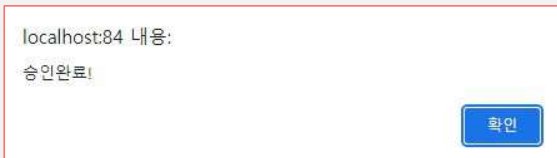
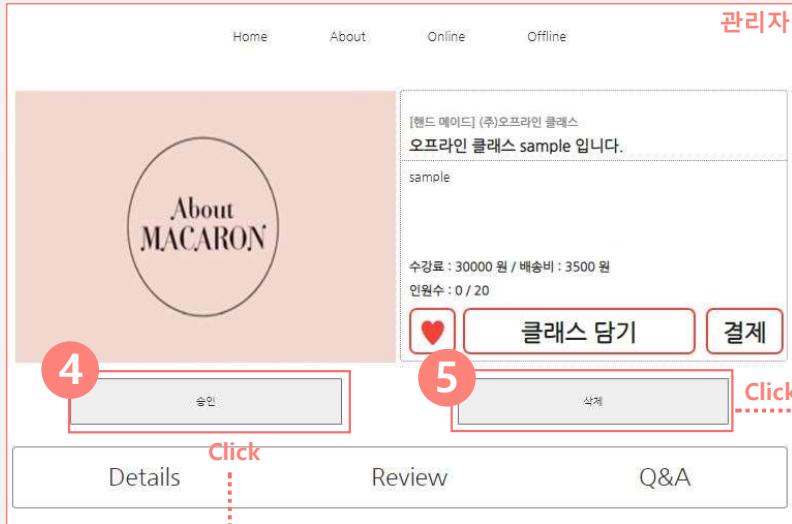
<table>
<tr>
<td>
<input type="button" class="updateFrm" id="updateFrm" action="ClassUpdateProc.jsp" method="post" value="수정하기" />
</td>
</tr>
</table>
```

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 게시물이 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 게시물이 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 실행되었는지 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();
    }
    
```

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

```

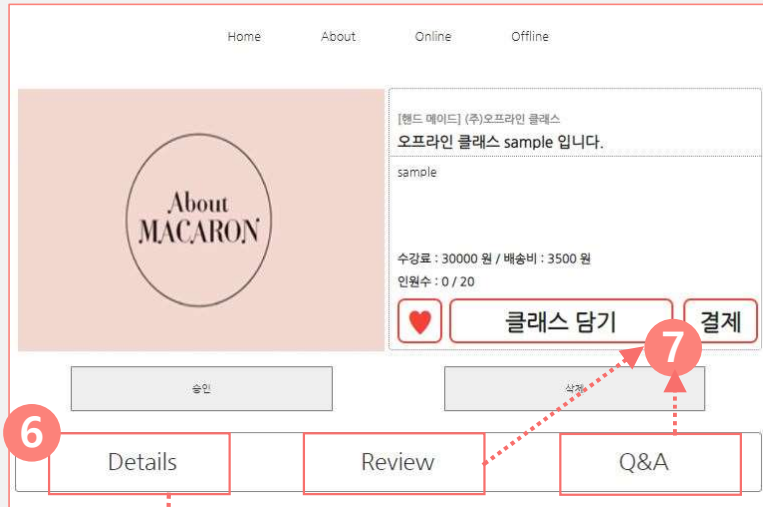
public int deleteBoard(int cNum) {
    Connection objConn = null;
    PreparedStatement objPstmt = null;
    ResultSet objRs = null;
    String sql = null;

    int exeCnt = 0; // 삭제 데이터 수, DB 삭제가 실행되었는지 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();
    }
    
```

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

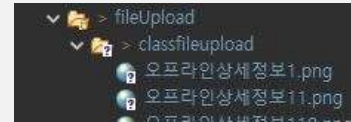
주요기능 - 클래스 상세 페이지



6

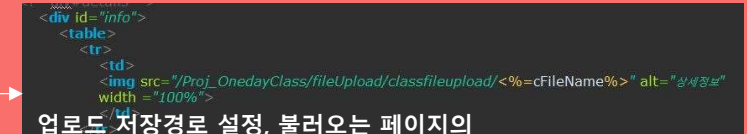
7

상세정보 이미지 출력



6 상세정보

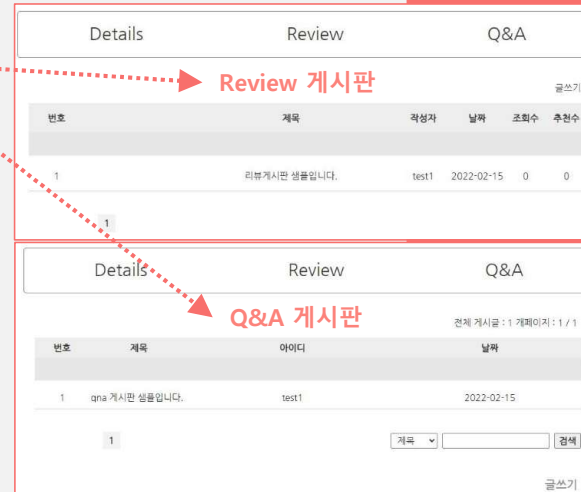
버튼 클릭 시 게시물 상세정보 이미지 출력



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

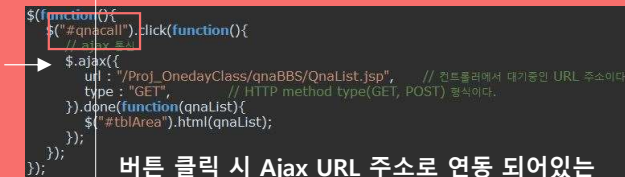
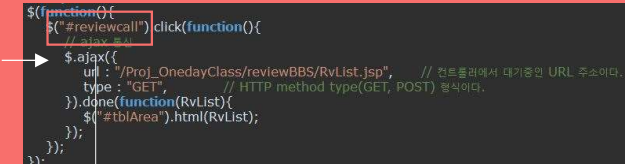
7 리뷰, Q&A 게시판

버튼 클릭 시 Ajax으로 연동한 게시물 출력

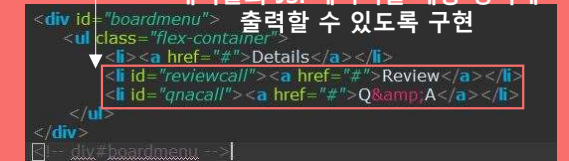


Review 게시판

Q&A 게시판



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

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

sample

수량 1000 원 / 배송비 : 3500 원

인원 10명

Click

클래스 담기 결제

Click

삭제

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

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

Cart.jsp

상품명	가격	수량	총계	장바구니 합계
오프라인 클래스 sample 입니다.	10000원	10000원	소계 : 20000원	
오프라인 클래스 sample 입니다.	10000원	10000원	배송 : 7000원	
			총계 : 207000원	

장바구니 페이지로 이동

결제하기

수령인 정보

결제정보

이름 김대형

전화번호 01012341234

주소 10000

상위주소 10000

이메일 test@naver.com

결제하기

수령인 정보

상품명	가격	수량	총계
오프라인 클래스 sample 입니다.	30000원	1	30000원
오프라인 클래스 sample2 입니다.	10000원	1	10000원

길태형님 환영합니다. | 로그아웃 | 2

insertCart() 메서드 호출

localhost:84 내용:
장바구니 취소가 되었습니다.

deleteCart() 메서드 호출

길태형님 환영합니다. | 로그아웃 | 1

장바구니 0개

길태형님 환영합니다. | 로그아웃 | 0

장바구니

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

결제하기

수령인 정보

결제 정보

수령인이름 길태형

전화번호 01012341234

주소 10930

상위주소 경기 파주시 마두리길 22

이메일 test@naver.com

결제시간 2022년 02월 17일 오전 01:02:44

상품명 가격 수량 총계

상품명	가격	수량	총계
오프라인 클래스 sample 입니다.	30000원	1	30000원
오프라인 클래스 sample2 입니다.	10000원	1	10000원

최종결제완료창 PaymentCompl.jsp

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 취소가 되었습니다.";
    } else {
        msg = "오류가 생겼습니다. 다시 시도해보세요.";
    }
} else {
    if(pMgr.insertCart(cartBean)) { // 입력 되었을때
        msg = "장바구니에 넣었습니다.";
    } else {
        msg = "오류가 생겼습니다. 다시 시도해보세요.";
    }
}
int cartCnt = pMgr.cntCart(uid);
session.setAttribute("cartCntKey", cartCnt);
}

sql = "delete from cartList where uid = ? AND cNum = ?";
objPstmt = objConn.prepareStatement(sql);
objPstmt.setString(1, bean.getId());
objPstmt.setInt(2, bean.getcNum());
int cnt = objPstmt.executeUpdate();
if (cnt > 0)
    flag = true;

sql = "insert into cartList values (?, ?)";
objPstmt = objConn.prepareStatement(sql);
objPstmt.setString(1, bean.getId());
objPstmt.setInt(2, bean.getcNum());
int cnt = objPstmt.executeUpdate();
if (cnt > 0)
    flag = true;

```

장바구니에 담긴 클래스가 없다
면 if문 조건식 실행

장바구니에 담긴 클래스가 있다
면 else문 조건식 실행

Vector Class를 사용한 배열출력<25페이지>

```

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

        int cNum = bean.getcNum();
        String cTitle = bean.getcTitle();
        String cUid = bean.getcUid();
        int cPrice = bean.getcPrice();
        int cDelivery = bean.getcDelivery();
        int cMaxStu = bean.getcMaxStu();
        int cApplyStu = bean.getcApplyStu();
        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 %>" />
            </td>
            <td>
                <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>
    }
}

```



감사합니다