

팀 자료 관리 서비스

TFMS : Team File Manage Service

B조 (MBTI조)

JSP와 서블릿 - 김병철 교수님 (목 7,8,9/사)

목차

1. 팀원 소개
2. 주제 소개
3. 목표 및 달성도
4. 세부 실행 내용
5. 구현 결과(+시연)
6. 성과 및 기대효과
7. 느낀점 및 개선사항

팀원소개

팀장. 장민관



프론트엔드, JSP보조,
서버 구현

팀원. 박주형



HTML, CSS, BootsStrap
프론트엔드

팀원. 최정규



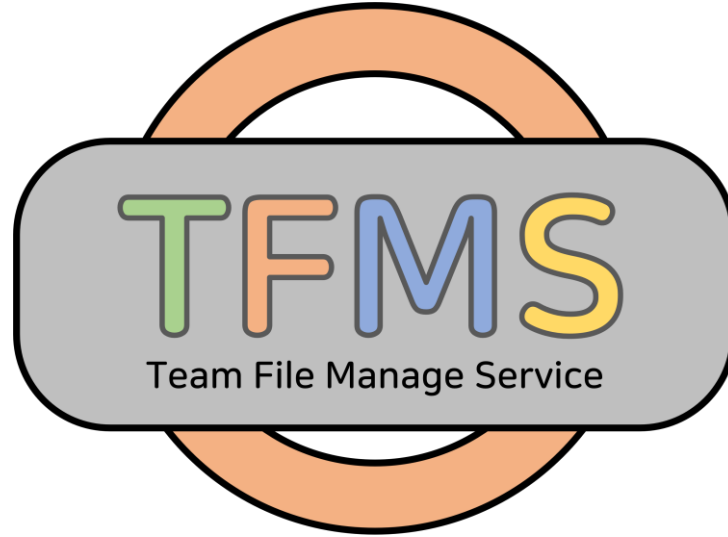
파일관리 백엔드(JSP)

팀원. 권진성



로그인,회원가입,
유저관리 백엔드(JSP)

주제소개

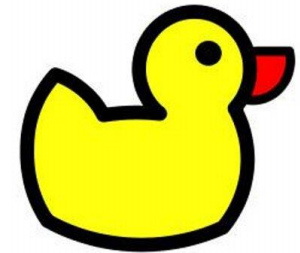
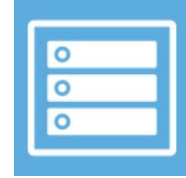
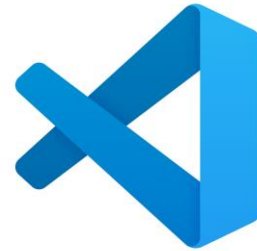
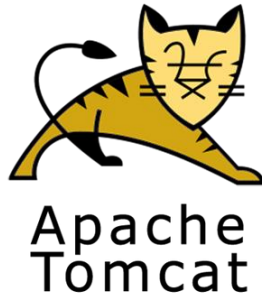
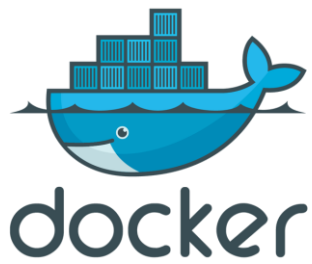


Big : 팀별 파일 관리 서비스

small : 서버를 통해 원거리에서 팀의 파일을 관리하는 서비스
여러 그룹을 참여 및 생성하여 팀별로 파일을 관리한다.
DB와 파일 입출력을 적절히 적용하여 구현한다.

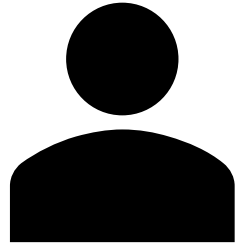
세부 실행 내용

서버 환경 - 사용 프로그램

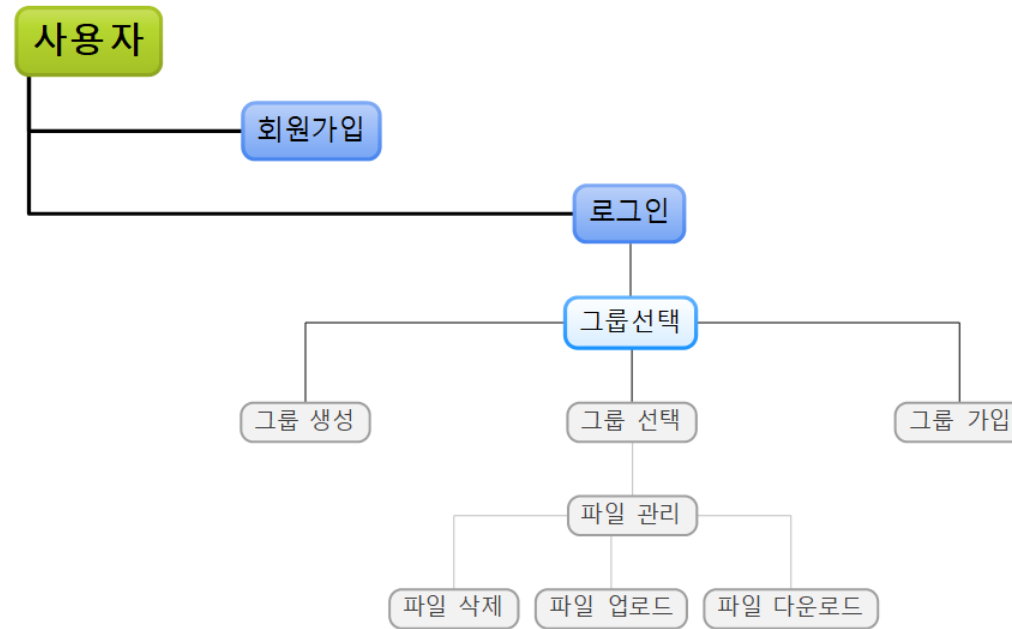


세부 실행 내용

기능구조도 : 사용자 시점

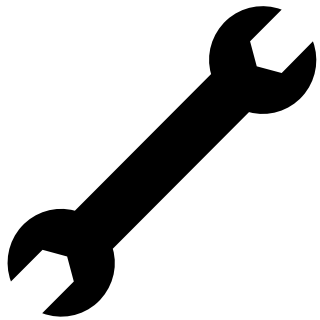


사용자

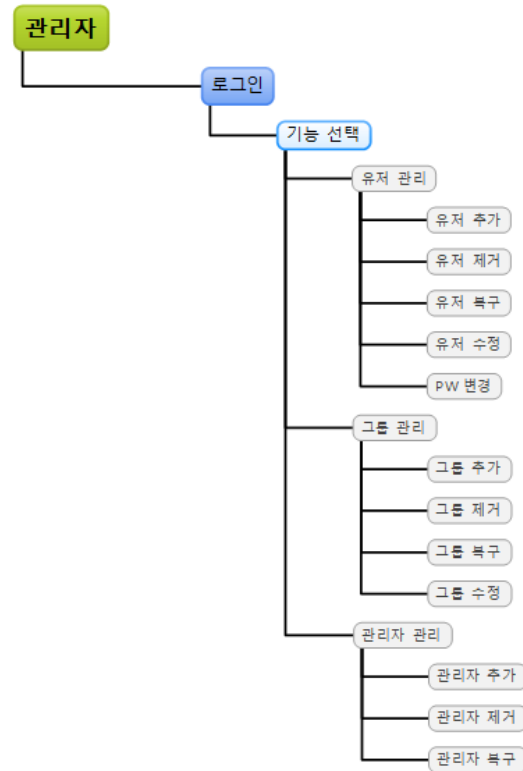


세부 실행 내용

기능구조도 : 관리자 시점

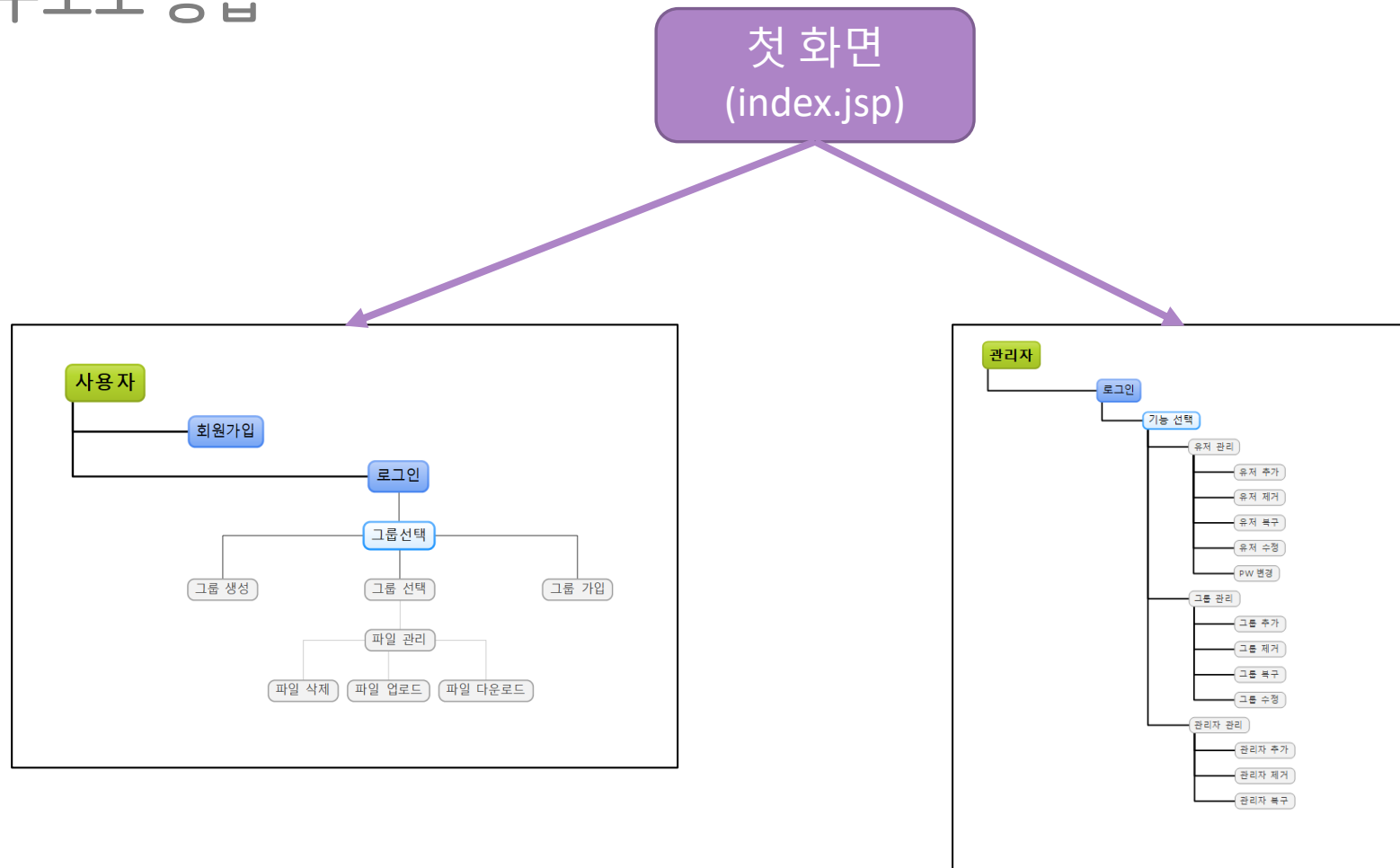


관리자



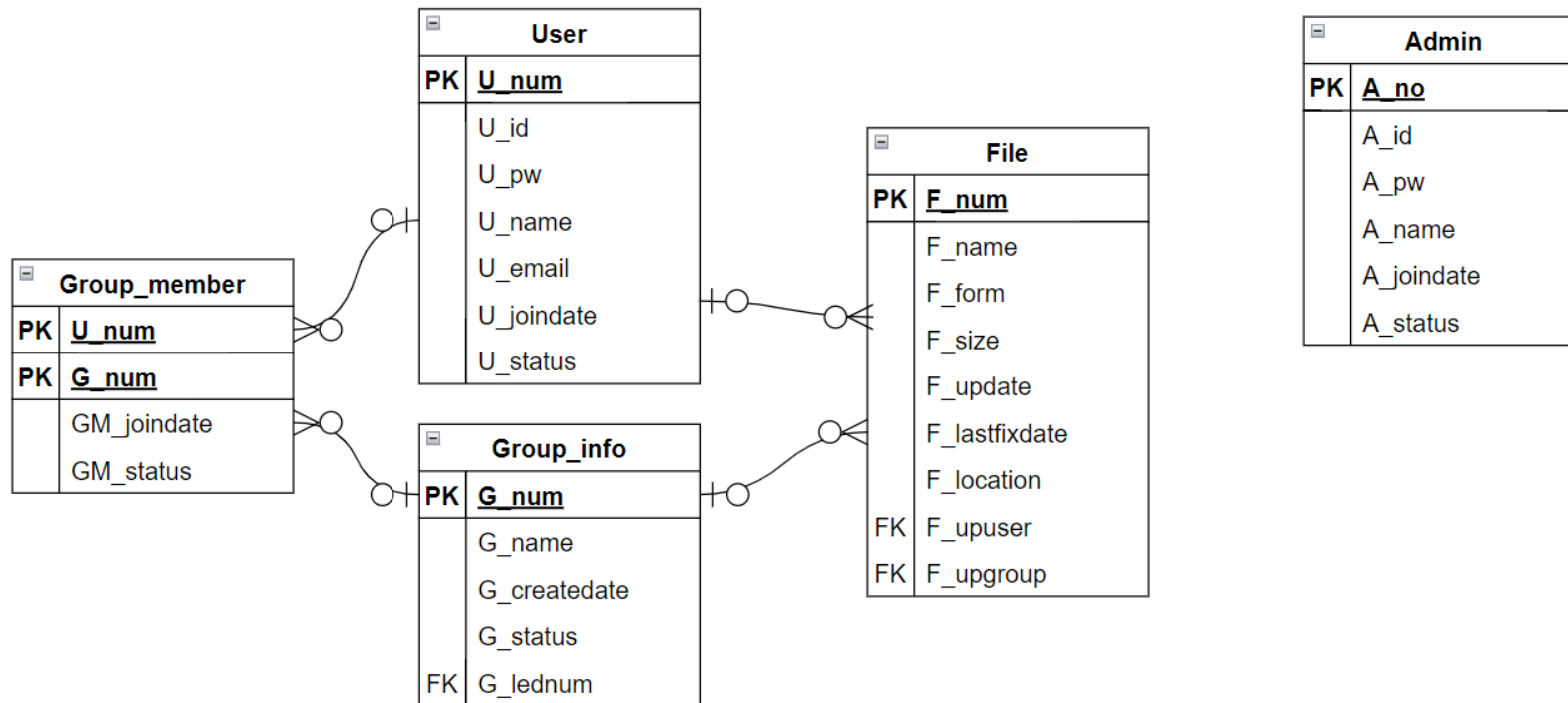
세부 실행 내용

기능구조도 종합



세부 실행 내용

DB 설계



구현 결과

DB구성

테이블

<input type="checkbox"/>	테이블	엔진?
<input type="checkbox"/>	admin	InnoDB
<input type="checkbox"/>	file	InnoDB
<input type="checkbox"/>	group_info	InnoDB
<input type="checkbox"/>	group_member	InnoDB
<input type="checkbox"/>	user	InnoDB
	총 5개	InnoDB

admin

열	형	주석
A_no	int(11) 자동 증가	
A_id	char(20)	
A_pw	char(20)	
A_name	char(20) NULL	
A_joindate	date NULL	
A_status	tinyint(1) NULL	

file

열	형	주석
F_num	int(11) 자동 증가	
F_name	char(100) NULL	
F_form	char(10) NULL	
F_size	char(10) NULL	
F_update	date NULL	
F_last_fix_date	date NULL	
F_upuser	int(11)	
F_upgroup	int(11)	
F_location	char(100) NULL	

group_info

열	형	주석
G_num	int(11) 자동 증가	
G_name	char(20) NULL	
G_createdate	date NULL	
G_lednum	int(11) NULL	
G_status	tinyint(1) NULL	

group_member

열	형	주석
G_num	int(11)	
U_num	int(11)	
GM_joindate	date NULL	
GM_status	tinyint(4) NULL [1]	

user

열	형	주석
U_num	int(11) 자동 증가	
U_id	char(30)	
U_pw	char(30)	
U_name	char(20) NULL	
U_email	char(50) NULL	
U_joindate	date NULL	
U_status	tinyint(1) NULL	

구현 결과

기본 디자인



TFMS
Team File Manage System

로그인

회원가입

관리자 로그인

Team B (MBTI 조)



TFMS:관리자

유저 관리

관리자명 : 관리자01 [로그아웃](#)

[이전으로](#) [유저 추가](#) [유저 복구](#)

번호	유저 고유번호	유저 아이디	이름	이메일	가입일자			
1	27	t1	test11	-	2022-12-18	수정	PW 변경	제거
2	28	aaa	asd	-	2022-12-18	수정	PW 변경	제거
3	30	qqqqqq	-	-	2022-12-18	수정	PW 변경	제거
4	31	sdf	-	-	2022-12-18	수정	PW 변경	제거
5	34	cjg721	최정규	cjg721@naver.com	2022-12-20	수정	PW 변경	제거

Team B (MBTI 조)



백석대학교
BAEKSEOK UNIVERSITY

구현 결과

기능 설명 - 회원가입



회원가입

ID

중복확인

PW

PW확인

사용자 이름

이메일

가입하기

[이전으로](#)

ID 확인 - Chrome

주의 요함 | tfms.duckdns.org:81/do/UIDCheckDo.jsp?idcheck=t1

ID 중복확인
"t1" 아이디는
중복 아이디로 사용 불가능합니다.
[이전으로](#)

ID 확인 - Chrome

주의 요함 | tfms.duckdns.org:81/do/UIDCheckDo.jsp?idcheck=t2

ID 중복확인
"t2" 아이디는
사용가능합니다.
[사용하기](#) [이전으로](#)

tfms.duckdns.org:81 내용:

가입하시겠습니까?

확인

취소

tfms.duckdns.org:81 내용:

패스워드 확인이 틀립니다.

확인

tfms.duckdns.org:81 내용:

가입되었습니다.

확인

구현 결과

기능 설명 - 회원가입

```
LocalDate now = LocalDate.now();
String uid = request.getParameter("user_id");
String upw = request.getParameter("user_pw");
String uname = request.getParameter("user_name");
String uemail = request.getParameter("user_email");
String udate = now.toString();//formatter.format(now);
String ustat = "1";
if(uname == ""){uname = null;}
if(uemail == ""){uemail = null;}
StringBuffer sql = new StringBuffer("insert into user(U_id,U_pw,U_name,U_email,U_joindate,U_status)");
sql.append("values(?,?,?,?,?,?)");
PreparedStatement pstmt = conn.prepareStatement(sql.toString());
pstmt.setString(1, uid);
pstmt.setString(2, upw);
pstmt.setString(3, uname);
pstmt.setString(4, uemail);
pstmt.setString(5, udate);
pstmt.setString(6, ustat);
pstmt.executeUpdate();
```

U_num	U_id	U_pw	U_name	U_email	U_joindate	U_status
27	t1	t1	test11	NULL	2022-12-18	1
28	aaa	aa	asd	NULL	2022-12-18	1
29	zzz	ddd	NULL	NULL	2022-12-18	0
30	qqqqqq	qqq	NULL	NULL	2022-12-18	1
31	sdf	sdf	NULL	NULL	2022-12-18	1
33	t2	qqqq	NULL	NULL	2022-12-20	1

회원가입 창에서 입력한 내용들을 가져오고 현재 시간을 가져온 후 유저 상태(U_status)를 1로 저장하게 됩니다. 이후 가져온 내용을 DB에 저장하게 된다.



백석대학교
BAEKSEOK UNIVERSITY

구현 결과

기능 설명 - 로그인



tfms.duckdns.org:81 내용:

로그인 할까요?

확인

취소

tfms.duckdns.org:81 내용:

로그인 실패
(아이디 또는 비밀번호가 다릅니다.)

tfms.duckdns.org:81 내용:

로그인 성공
(test11님 환영합니다.)

구현 결과

기능 설명 - 로그인

```
function confirmCheck(){
    var idvalue = document.getElementsByName("user_id").value;
    var pwvalue = document.getElementsByName("user_pw").value;
    if(idvalue == ""){
        alert("ID를 입력해주세요.");
        return false;
    }
    else{
        if(pwvalue == ""){
            alert("PW를 입력해주세요.");
            return false;
        }
    }
    var confirmresult = confirm("로그인 할까요?");
    if(confirmresult==false){
        return false;
    }else{return true;}
}
```

```
if ("<%=idResult%>" == 0) {
    alert("로그인 실패\n(아이디 또는 비밀번호가 다릅니다.)");
    window.history.back();
}
else {
    if ("<%=statusResult%>" == 0) {
        alert("로그인 실패\n(탈퇴 처리된 회원입니다.)");
        window.history.back();
    }
    else {
        if ("<%=upw%>" == "<%=pwResult%>") {
            localStorage.setItem('unum', "<%=usernumResult%>");
            sessionStorage.setItem('uname', "<%=nameResult%>");
            sessionStorage.setItem('uemail', "<%=emailResult%>");
            sessionStorage.setItem('ujoindate', "<%=userjdateResult%>");
            alert("로그인 성공\n(<%=nameResult%>님 환영합니다.)");
            location.href="/userside/Gselect.jsp?Unum=<%=usernumResult%>";
        }
        else {
            alert("로그인 실패\n(아이디 또는 비밀번호가 다릅니다.)");
            window.history.back();
        }
    }
}
```

아이디와 비밀번호 창이 비어있는지 확인과
로그인 실패에 관련된 코드 및 입력된 정보가 세션에 들어가는 코드



백석대학교
BAEKSEOK UNIVERSITY

구현 결과

기능 설명 - 로그아웃



그룹 선택

유저이름 : test11 [로그아웃](#)

tfms.duckdns.org:81 내용:
로그아웃 하시겠습니까?

확인

취소

로그아웃 처리시 사용자 확인 작업 이후 세션스토리지 내용을 지우고 로그아웃 합니다.

DB데이터 리스트화

```
// 세션생성
//session = request.getSession();

String unum = request.getParameter("Unum");
String sql1 = "Select count(G_num) as Gcount from group_member";
String sql2 = "Select * from group_info where G_status = 1 and
//sql 즉시 실행
Statement stmt1 = conn.createStatement();
//Select SQL문
ResultSet rs1 = stmt1.executeQuery(sql1);
rs1.next();
int gcount = Integer.parseInt(rs1.getString("gcount"));

Statement stmt2 = conn.createStatement();
//Select SQL문
ResultSet rs2 = stmt2.executeQuery(sql2);
//[0]: gnum [1]: gname [2]: gdate [3]: gleadnum
String[][] GroupList = new String[gcount][4];

//GroupList
for(int i=0;i<gcount;i++){
    rs2.next();
    GroupList[i][0] = rs2.getString("G_num");
    GroupList[i][1] = rs2.getString("G_name");
    GroupList[i][2] = rs2.getString("G_createdate");
    GroupList[i][3] = rs2.getString("G_lednum");
    if(GroupList[i][0] == null){GroupList[i][0] = "-";}
    if(GroupList[i][1] == null){GroupList[i][1] = "-";}
    if(GroupList[i][2] == null){GroupList[i][2] = "-";}
    if(GroupList[i][3] == null){GroupList[i][3] = "-";}
}
```

```

<tr>
    <th>번호</th>
    <th>그룹 번호</th>
    <th>그룹 이름</th>
    <th>생성 날짜</th>
    <th>릴장 번호</th>
</tr>
</thead>
<tbody>
    <%
        for(int j=0;j<gcount;j++){
            int count = j;
    %>
    <tr>
        <td><%=j+1%></td>
        <td><%=GroupList[j][0]%></td>
        <td><%=GroupList[j][1]%></td>
        <td><%=GroupList[j][2]%></td>
        <td><%=GroupList[j][3]%></td>
        <td><input type="button" value="선택" onclick="groupSelect(<%=GroupList[j][0]%>)"></td>
    </tr>
    <%
        }
    %>
</tbody>
</table>

```

구현 결과

유저/그룹 추가, 제거, 수정, 복구

이전으로 유저 추가 유저 복구								
번호	유저 고유번호	유저 아이디	이름	이메일	가입일자			
1	27	t1	test11	-	2022-12-18	수정	PW 변경	제거
2	28	aaa	asd	-	2022-12-18	수정	PW 변경	제거
3	30	qqqqqq	-	-	2022-12-18	수정	PW 변경	제거
4	31	sdf	-	-	2022-12-18	수정	PW 변경	제거
5	34	cjg721	최정규	cjg721@naver.com	2022-12-20	수정	PW 변경	제거

구현 결과

유저/그룹 추가, 제거, 수정, 복구

추가

```
LocalDate now = LocalDate.now();
String gname = request.getParameter("group_name");
String gleader = request.getParameter("leader_num");
String gdate = now.toString();
String gstatus = "1";
if(gleader == ""){gleader = null;}

//sql
StringBuffer sql = new StringBuffer("insert into group_info(G_name, G_leader, G_date, G_status)");
sql.append("values(?, ?, ?, ?)");
PreparedStatement pstmt = conn.prepareStatement(sql.toString());
pstmt.setString(1, gname);
pstmt.setString(2, gleader);
pstmt.setString(3, gdate);
pstmt.setString(4, gstatus);
pstmt.executeUpdate();
```

수정

```
String gno = request.getParameter("Gno");
String gname = request.getParameter("group_name");
String gleader = request.getParameter("leader_num");
if(gleader == ""){gleader = null;}

//sql
StringBuffer sql = new StringBuffer("update group_info set G_name = ?, G_leader = ?");
PreparedStatement pstmt = conn.prepareStatement(sql.toString());
pstmt.setString(1, gname);
pstmt.setString(2, gleader);
pstmt.executeUpdate();
```

제거

```
String gno = request.getParameter("Gno");

//sql
StringBuffer sql = new StringBuffer("update group_info set G_status = '0'");
sql.append("where G_num = ?");
PreparedStatement pstmt = conn.prepareStatement(sql.toString());
pstmt.setString(1, gno);
pstmt.executeUpdate();
```

복구

```
//sql
StringBuffer sql = new StringBuffer("update group_info set G_status = '1'");
sql.append("where G_num = ?");
PreparedStatement pstmt = conn.prepareStatement(sql.toString());
pstmt.setString(1, gno);
pstmt.executeUpdate();
```



구현 결과_파일 리스트

그룹번호 : 27

파일 업로드 [이전으로](#)

번호	파일번호	파일이름	파일형식	파일크기	올린날짜	
1	14	20221220_082050.png	png	35Kb	2022-12-20	<div>다운로드</div> 삭제
2	19	Homework_#13_Solution.pdf	pdf	63Kb	2022-12-20	<div>다운로드</div> 삭제
3	20	연구노트_0팀_학번_이름_0주차.hwp	hwp	13Kb	2022-12-20	<div>다운로드</div> 삭제

1)먼저 sql을 통해서 세션에 저장되어 있는 그룹넘버에 해당하는 파일들의 목록을 전부 가져온다.

2)해당 정보들을 반복을 통해 테이블에 리스트 형식으로 보여주고 여러 버튼들에 같이 넘겨줄 값들을 같이 넣어준다

3)파일 이름에는 링크를 걸어 파일을 다운로드할 수 있는 jsp파일로 이동한다.

구현 결과_파일 리스트_소스코드

```
sql = "select * from file where F_upgroup = " + gnum;

int rowCount = 1;
try {
    pstmt = conn.prepareStatement(sql);

    ResultSet result = pstmt.executeQuery();
}%
<%@ include file="/topBarUser.jsp" %>
<script>
    window.onload = function(){
        document.getElementById("barsiteinfo").innerText = "TFMS";
        document.getElementById("bartitle").innerText = "파일관리";
        document.getElementById("baruser").innerText = "유저이름 : " + sessionStorage.getItem("uname");
        document.getElementById("bargroup").innerText = "그룹번호 : " + <%=gnum%>;
    }
</script>
<h3>그룹번호 : <%=gnum%></h3>
<input type="button" value="파일 업로드" onclick="location.href='/userside/FUpload.jsp?Unum=<%=unum%>&Gnum=<%=gnum%>'">
<a href="/userside/Gselect.jsp?Unum=<%=unum%>">이전으로</a>
<table style="margin-left: auto; margin-right: auto;" class='table-bordered table-sm'>
    <thead>
        <tr>
            <th>번호</th>
            <th>파일번호</th>
            <th>파일이름</th>
            <th>파일형식</th>
            <th>파일크기</th>
            <th>올린날짜</th>
        </tr>
    </thead>
    <tbody>
```

```
<tbody>
    <% while(result.next()){ %>
    <tr>
        <td><%= rowCount %></td>
        <td><%= result.getInt("F_num") %></td>
        <td><%= result.getString("F_name") %></td>
        <td><%= result.getString("F_form") %></td>
        <td><%= result.getString("F_size") %></td>
        <td><%= result.getString("F_update") %></td>
        <td><input type="button" value="다운로드" onclick="location.href='FDownload_test_Do.jsp?Unum=<%=unum%>&Gnum=<%=gnum%>&Fno=<%= result.getInt("F_num") %>'></td>
        <td><a href="FDelete_test_Do.jsp?Unum=<%=unum%>&Gnum=<%=gnum%>&Fno=<%= result.getInt("F_num") %>">삭제</a></td>
    </tr>
    <%
        rowCount++;
    }
    result.close();
}
catch (Exception e) {
    out.println("[board] 테이블 조회에 문제가 있습니다. <hr>");
    out.println(e.toString());
    e.printStackTrace();
}
finally {
    if (pstmt != null) pstmt.close();
    if (conn != null) conn.close();
}
    %>
</tbody>
</table>
<script src="/scripts.js"></script>
<%@ include file="/bottomBar.jsp" %>
```



구현 결과_파일 업로드

※ 파일 업로드시 한글파일명 사용 불가

파일 선택	20221221_032052.png	업로드
이전으로		

- 1)파일 업로드는 수업시간에 사용했던 cos라이브러리를 이용해서 파일 업로드를 진행하였다.
- 2)먼저 파일정보를 db에 넣을 sql문을 작성하였다.
그다음 그룹넘버를 이름으로 가지는 경로를 넣어서 multipartRequest를 설정해주었다.
- 3) sql에 넣을 요소들을 정리해주었다. 파일의 이름에서 '.'을 기준으로 나눠 해당 파일의 형식을 가져오고 파일의 크기는 byte, Kb, Mb단위로 나눠서 저장해주었다.
- 4) 해당 요소들을 sql에 넣고 수행을 해주는 것으로 업로드를 마무리했다.

구현 결과_파일 업로드_소스코드

```
int rowCount = 0;
String unum = request.getParameter("Unum");
String gnum = request.getParameter("Gnum");
PreparedStatement pstmt = null;
Statement stmt = null;
StringBuffer SQL = new StringBuffer("insert into file(F_name, F_form, F_size, F_update, F_last_f
SQL.append("values (?, ?, ?, ?, ?, ?, ?, ?)");
try {
    String encoding = "UTF-8";
    int sizeLimit = 10 * 1024 * 1024;
    String relativeDirectory = "/data/"+gnum;
    ServletContext context = getServletContext();
    String realDirectory = context.getRealPath(relativeDirectory);
    MultipartRequest multi = new MultipartRequest(request, realDirectory, sizeLimit, encoding, n

    Date now = new Date();
    SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
    String today = sf.format(now);
    String f_name = multi.getFilesystemName("fileInput"); //파일이름
    String[] f_form = new String[2];
    f_form = f_name.split("[.]");
    String form = f_form[1];
    String f_size = String.valueOf(multi.getFile("fileInput").length()); // 파일의 크기를 반환
    int size = Integer.parseInt(f_size);

    if( (size/1024)>1){
        int size_kb = size/1024;
```

```
        if((size_kb/1024)>1){
            int size_mb = size_kb/1024;
            size_mb++;
            f_size= size_mb + "Mb";
        }else{
            size_kb++;
            f_size = size_kb+"Kb";
        }
    }else{
        f_size = f_size + "byte";
    }

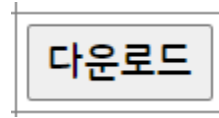
    pstmt = conn.prepareStatement(SQL.toString());
    //삽입할 학생 레코드 데이터 입력
    pstmt.setString(1, f_name);
    pstmt.setString(2, form);
    pstmt.setString(3, f_size);
    pstmt.setString(4, today);
    pstmt.setString(5, today);
    pstmt.setInt(6, Integer.parseInt(unum));
    pstmt.setInt(7, Integer.parseInt(gnum));
    pstmt.setString(8, relativeDirectory);

    rowCount = pstmt.executeUpdate();
    if (rowCount == 1) out.println("<hr>레코드 하나가 성공적으로 삽입 되었습니다.<hr>");
    else out.println("레코드 삽입에 문제가 있습니다.");

    stmt = conn.createStatement();
}
catch(Exception e) {
    out.println("조회에 문제가 있습니다. <hr>");
```

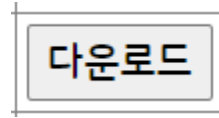


구현 결과_파일 다운로드



- 1) 다음 먼저 파일의 위치를 찾아야하기 때문에 해당 정보들을 DB에서 가져와서 파일의 실제 경로를 찾아준다.
- 2) 구한 경로를 통해 파일 객체를 생성하고 그 객체를 FileInputStream을 생성해서 바이트 단위로 읽어올 준비를 한다.
- 3) response헤더에 파일의 링크를 클릭했을 때 다운로드 화면이 출력되도록 처리를 해주고 파일의 크기도 같이 가져온다.
- 4) 그 다음 FileOutputStream에 response를 연결해서 연결했던 파일을 바이트단위로 가져오면서 파일의 다운로드가 완료된다.

구현 결과_파일 다운로드



- 1) 다음 먼저 파일의 위치를 찾아야하기 때문에 해당 정보들을 DB에서 가져와서 파일의 실제 경로를 찾아준다.
- 2) 구한 경로를 통해 파일 객체를 생성하고 그 객체를 FileInputStream을 생성해서 바이트 단위로 읽어올 준비를 한다.
- 3) response헤더에 파일의 링크를 클릭했을 때 다운로드 화면이 출력되도록 처리를 해주고 파일의 크기도 같이 가져온다.
- 4) 그 다음 FileOutputStream에 response를 연결해서 연결했던 파일을 바이트단위로 가져오면서 파일의 다운로드가 완료된다.

구현 결과_파일 다운로드_소스코드

```
try {

    pstmt = conn.prepareStatement(sql); // 해당 파일 정보 가져오기
    ResultSet result = pstmt.executeQuery();

    result.next();

    String f_name = result.getString("F_name");

    String relativeDirectory = result.getString("F_location");
    ServletContext context = getServletContext();
    String f_path = context.getRealPath(relativeDirectory); //경로 구하기
    f_path = f_path + "/" + f_name; //경로 끝

    try{
        myFile = new File (f_path); // 파일 인식
        f_in = new FileInputStream(myFile);
    }catch(FileNotFoundException fe){
        skip = true;
    }

    client = request.getHeader("User-Agent");

    // 파일 다운로드 헤더 지정
    response.reset() ;
    response.setContentType("application/octet-stream");
    response.setHeader("Content-Description", "JSP Generated Data");

    if(!skip){

        response.setHeader ("Content-Disposition", "attachment; filename="+
        response.setHeader ("Content-Length", ""+myFile.length() );

        f_os = response.getOutputStream();
        byte b[] = new byte[(int)myFile.length()];
        int leng = 0;
```

구현 결과_파일 삭제_소스코드

```
<%
String unum = request.getParameter("Unum");
String gnum = request.getParameter("Gnum");
String fno = request.getParameter("Fno");
PreparedStatement pstmt = null;
Statement stmt = null;
int rowCount = 0;

String sql_delete = "delete from file where F_num =" + fno;
String sql = "select * from file where F_num =" + fno;

try {

    pstmt = conn.prepareStatement(sql); // 해당 파일 정보 가져오기
    ResultSet result = pstmt.executeQuery();

    result.next();

    String relativeDirectory = result.getString("F_location");
    ServletContext context = getServletContext();
    String realDirectory = context.getRealPath(relativeDirectory); //경로 구하기

    realDirectory = realDirectory + "/" + result.getString("F_name");
    File deleteFile = new File (realDirectory);
    if(deleteFile.exists()){
        deleteFile.delete(); //파일이 존재하면 삭제
        //out.println("<hr>파일 삭제 성공.<hr>");
    }else{
        //out.println("<hr>파일 삭제 실패.<hr>");
    }

    pstmt = conn.prepareStatement(sql_delete);
    rowCount = pstmt.executeUpdate();
}
```

1)삭제는 해당하는 파일의 경로를 통해서 객체를 생성한다.

2)파일 객체를 .delete()함수를 사용하면 해당하는 파일이 삭제한다.

3)삭제가 되면 해당 파일의 DB정보도 sql명령문을 이용해 지워준다.



백석대학교
BAEKSEOK UNIVERSITY

성과 및 기대효과

- 팀의 자료관리 서비스를 제공한다.
- 프론트와 백엔드 구현을 해봄으로 웹 이해도를 높인다.
- 팀 프로젝트를 진행함으로 협업의 중요성을 깨닫는다.
- 서버 세팅 단계에서부터 진행하여 서버에 대한 이해도를 높인다.

감사합니다