

# 웹 프로그래밍 TP#2 최종 보고서

## 커뮤니티 웹사이트 만들기

분반 : 01반

학번 : 201302450

이름 : 이상현

### 1. 본 과제의 개요

\* 목표 및 본인이 구현한 시스템에 대한 간단한 설명

전체적으로 봤을 때 커뮤니티 웹사이트를 주제로 웹사이트를 구현한다. 서비스를 이용하는 클라이언트는 HW1과 유사하게 메일인증을 통한 회원가입을 하고 로그인을 하면 서비스를 사용할 수 있다.

이번 과제에서는 주로 커뮤니티 웹사이트의 기본적인 기능 구현을 목표로 하며 마이페이지, 익명게시판, 자유게시판, 갤러리, 방명록, 자료검색 등이 있다. 마이페이지에서는 회원가입 시 자신이 입력한 정보를 확인하고 정보를 수정할 수 있으며, 익명게시판에서는 게시글을 익명으로 올릴 수 있다. 자유게시판은 일반적인 게시판을 구현하였으며, 갤러리는 짧은 제목과, 내용, 그리고 사진을 첨부할 수 있도록 하였다. 방명록은 지난 팀 프로젝트1에서 구현하였던 게스트북을 응용하였다.

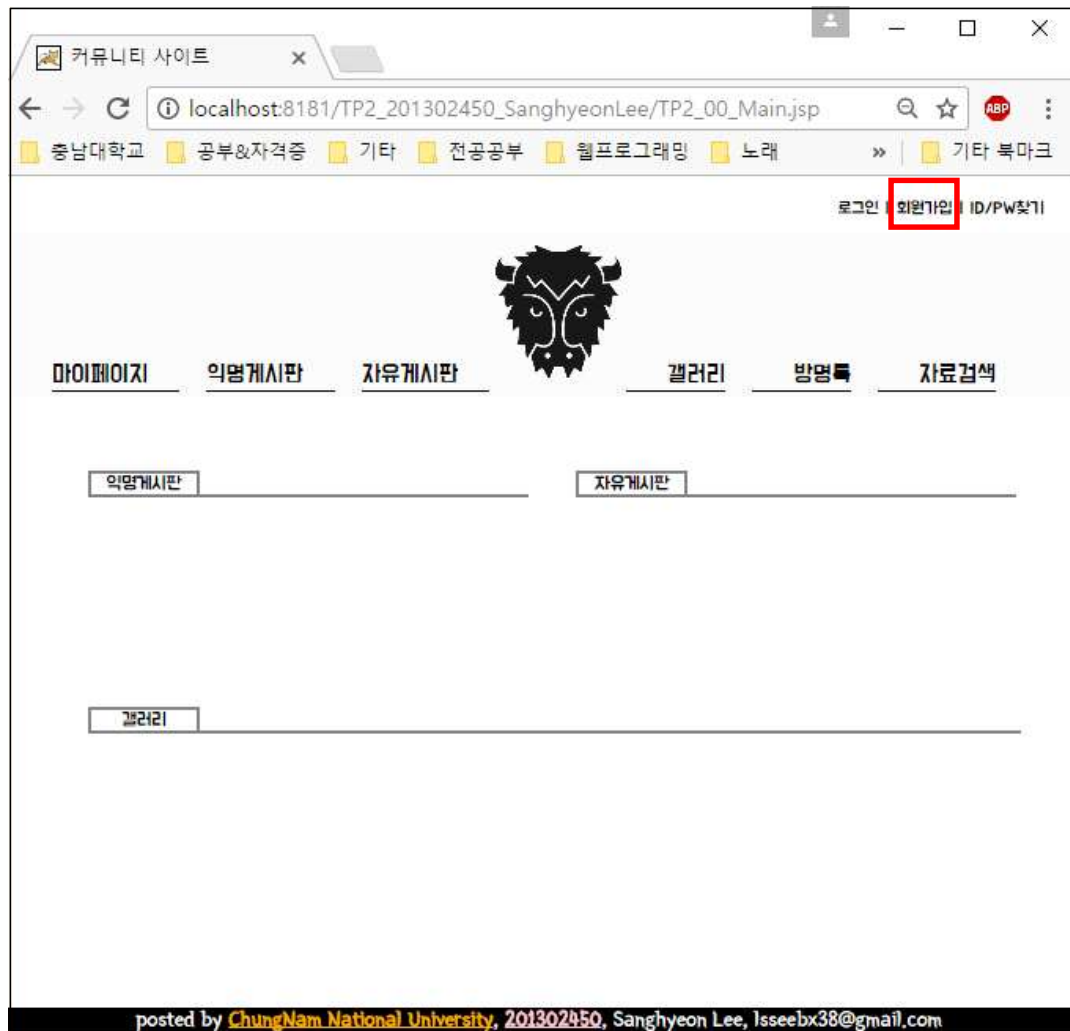
추가적으로 지난 설계에서 언급하지 않았던 웹소켓을 사용하여, 메인페이지를 제외한 각 페이지의 왼쪽 aside부분에 웹소켓을 통해 클라이언트 to 클라이언트 또는 클라이언트 to 서버 간 채팅을 가능하도록 구현하였다.

## 2. 본 시스템에서 구현한 기능 시나리오 (상세히 작성)

\* 각 기능별로 나누어 실제 실행 화면을 캡처하여 붙이고, 부가 설명을 기술

### 1) 회원가입, 로그인

먼저 데이터베이스에 아무것도 저장되어 있지 않을 때의 메인페이지이다. 메인페이지는 익명게시판, 자유게시판, 갤러리에 등록된 게시글 중 가장 최근에 등록된 5개의 글과 갤러리의 경우 4개의 이미지를 출력한다. 하지만 현재 아래의 사진에서는 데이터베이스에 저장된 것이 없으므로 출력되는 글이 없다.



클라이언트가 처음 이 서비스를 사용한다고 가정하고 회원가입을 누르면 회원가입을 할 수 있는 페이지로 이동한다.



마이페이지
익명게시판
자유게시판
갤러리
방명록
자료검색

2017 웹 프로그래밍 TP2

본 약관은 "TP2"서비스를 이용하기 위해서는 회원등록을 완료 해야합니다.

☒ 약관에 동의합니다.

ID	<input type="text" value="isseebx"/>	<input type="button" value="중복 체크"/>
Password	<input type="password" value="....."/>	
Password 확인	<input type="password" value="....."/>	
이름	<input type="text" value="이상현"/>	
전화번호	<input type="text" value="010-5680-3247"/>	
이메일	<input type="text" value="isseebx@naver.com"/>	
생년월일	<input type="text" value="2017-12-31"/>	
<input type="button" value="회원가입"/>		

유효성을 검사하는 알고리즘을 가지는 js 및 이에 사용되는 jsp파일들은 기존 HW1과제에서 했던 것을 응용하였다. 그때와 차이점이 있다면 회원정보의 저장을 텍스트파일이 아닌 외부 데이터베이스인 mysql을 사용하였다는 점이다. 유효성에 맞게 정보를 입력하고 회원가입 버튼을 누르면 데이터베이스에 입력한 정보가 삽입된다. 하지만 아직 인증이 되지 않았으므로, 인증하기 전까지는 사용가능 하지 않도록 activation이라는 필드를 두어 false로 저장한다.

아래는 쿼리문을 이용하여 데이터베이스에 삽입하는 java코드이다.

```
final String ACTIVATION = "false";
final String ID         = request.getParameter("id");
final String PW         = request.getParameter("pw");
final String NAME       = request.getParameter("name");
final String TEL        = request.getParameter("tel");
final String EMAIL      = request.getParameter("email");
final String BIRTH      = request.getParameter("birth");

Class.forName(driverName);
Connection conn = DriverManager.getConnection(URL, DBID, DBPW);

//String.format()을 이용하여 Member테이블에 삽입하는 insert쿼리를 작성
final String sql = String.format("insert into Member values('%s','%s','%s','%s','%s','%s','%s')",
    ACTIVATION, ID, PW, NAME, TEL, EMAIL, BIRTH);
final Statement stmt = conn.createStatement();
final int isUpdate   = stmt.executeUpdate(sql);
```

위의 자바코드를 통해 아래와 같이 데이터베이스의 Member 테이블에 클라이언트의 정보가 임시로 저장된다.

서버: localhost ▶ 데이터베이스: webprogrammingtermproject ▶ 테이블 : member "회원정보"

보기 구조 SQL 검색 삽입 내보내기 Import 테이블 작업 비우기 삭제

행(레코드) 보기 0 - 0 (1 합계, 질의 실행시간 0.0004 초)

SELECT +  
FROM 'member'  
LIMIT 0 , 30

보기: 30 행, 시작(행)위치 0  
수평(가로) 정렬 ( 100 칸이 넘으면 헤더 반복)

+ Options

	ACTIVATION	ID	PW	NAME	TEL	EMAIL	BIRTH
<input type="checkbox"/>	false	isseebx	qQ1!12	이상현	010-5680-3247	Isseebx@naver.com	2017-12-31

이 상태에서 로그인을 하려고 하면 activaiton이 false이므로 메일인증을 해달라고 요청을 하며 로그인을 못하도록 막는다.

← → ↻ localhost:8181/TP2\_201302450\_SanghyeonLee/MemberManagement/ 🔍 ☆ ABP ⋮

충남대학교 기타 북마크

localhost:8181 내용:

메일 인증이 필요한 아이디입니다.

확인

마이페이지 익명게시판 자유게시판 갤러리 방명록 자료검색

ID isseebx  
Password  
로그인 회원가입

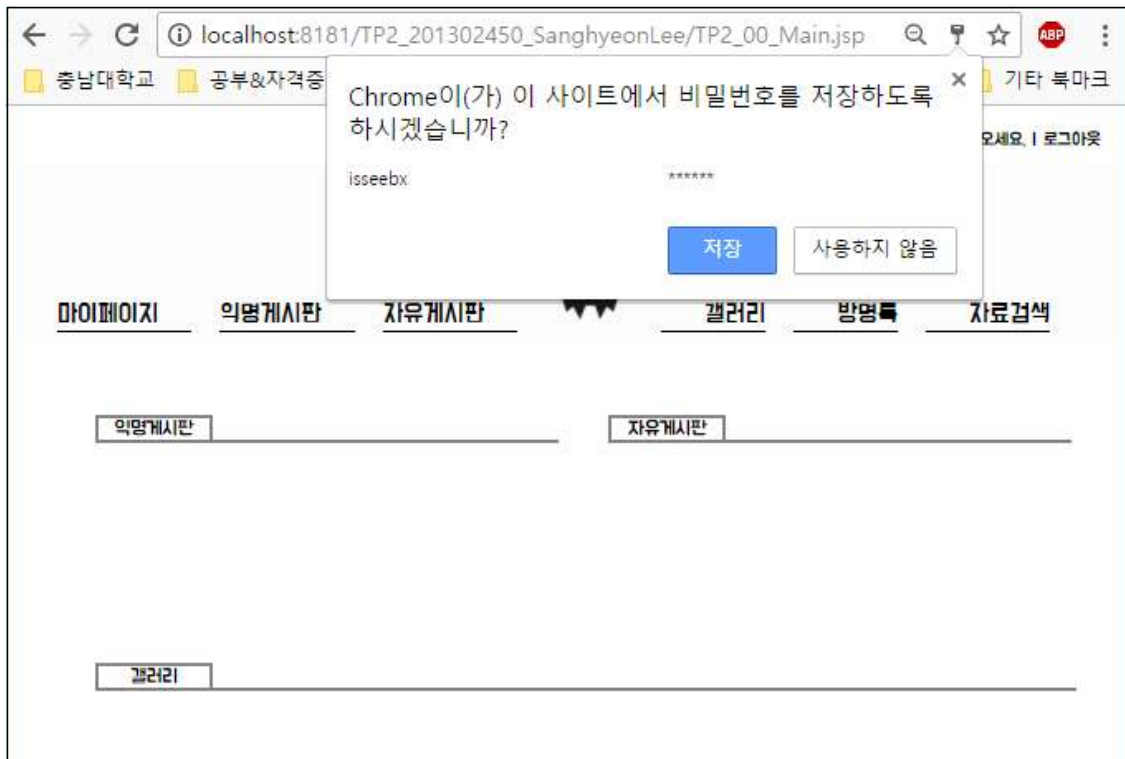
만약 메일을 통해 인증을 하였다면 아래와 같은 쿼리문을 가진 자바코드가 실행되고 데이터베이스 레코드의 해당 필드가 수정된다.

```
//String.format()을 이용하여 Member테이블에 Update하는 쿼리를 작성
final String SQL = String.format("UPDATE Member SET activation='%s' WHERE id='%s'", ACTIVATION, ID);

final Statement stmt = conn.createStatement();
final int isUpdate = stmt.executeUpdate(SQL);
```

	ACTIVATION	ID	PW	NAME	TEL	EMAIL	BIRTH
<input type="checkbox"/>	true	isseebx	qQ1!12	이상현	010-5680-3247	Isseebx@naver.com	2017-12-31

위에서 메일인증을 완료한 아이디로 로그인을 하면 정상적으로 로그인 이 되면서 아래의 사진과 같이 메인페이지로 전환이 된다. 로그인을 하였을 때 정보는 세션에서 관리하므로 크롬에서도 아래와 같이 비밀번호를 자동으로 저장하겠냐는 팝업창도 뜨는 것을 확인해 볼 수 있다.



클라이언트의 정보를 서버의 세션에 저장하는 자바코드는 다음과 같다.

```
else if(ACTIVATION.equals("true") && PW.equals(requestPw)){
    /* ID와 PW가 일치하는 record가 있을 경우 세션에 등록, 로그인 성공 진행 */
    result = "idExist";
    session.setAttribute("sessionId", ID);
    session.setAttribute("sessionName", NAME);
    break;
}
```

처음 로그인을 한 이후에는 다음 사진의 우측 상단과 같이 회원의 이름이 표시되며, 로그아웃 버튼이 생긴다. 로그아웃을 누를 경우 서버의 세션에 저장된 값들이 제거되며 로그인 창으로 전환된다.



이후 각 게시판에서 게시글 작성버튼을 눌렀을 경우 게시글 작성을 할 수 있는 페이지로 전환된다. 만약 로그인을 하지 않았다면 게시글 작성을 할 수 없으며, 게시글 보기는 가능하지만 호스트가 아니므로 게시글 삭제나 수정 또한 역시 불가능하다. 이에 대한 간략한 자바 코드는 다음과 같다.

```
<a id = "NewWrite"
  <%
    final String ID = (String)session.getAttribute("sessionId");
    if(ID == null){ out.print("onclick=\"alert('로그인을 해주세요')\""); } %>
href="<%
    if(ID == null){ out.print("#"); }
    else{ out.print("./TP2_07_NewWrite.jsp?boardname=AnonymousBoard"); }
%>">게시물등록</a>
```

또 로그아웃 시 사용되는 자바코드는 다음과 같다.

```
<%
    session.invalidate(); //session 전체 무효화.
    response.sendRedirect("./Login.jsp"); //메인 화면으로 이동
%>
```

## 2) 마이페이지

마이페이지에서는 회원가입 시 자신이 입력한 정보를 확인하고, 수정하여 데이터베이스에 반영할 수 있다. 만약 로그인을 하지 않았다면 회원정보가 출력되지 않으며 당연히 수정도 불가능하다. 이는 다음과 같다.

The screenshot shows the '마이페이지' (My Page) interface. At the top right, there are links for '로그인 | 회원가입 | ID/PW찾기'. The main navigation bar includes '마이페이지', '익명게시판', '자유게시판', '갤러리', '방명록', and '자료검색'. A modal window is displayed in the center with the title 'localhost:8181 내용:' and the message '로그인을 해주세요' (Please log in). There is an '확인' (Confirm) button at the bottom right of the modal. The background content is partially obscured by the modal.

만약 로그인을 하였다면 아래와 같이 데이터베이스에 저장된 정보들을 input태그의 value에 출력하여 정보를 수정한 후 데이터베이스에 반영이 가능하도록 해준다.

The screenshot shows the '마이페이지' (My Page) interface after a successful login. At the top right, it says '이상현님 어서오세요. | 로그아웃'. The main navigation bar is the same as the previous screenshot. The '회원 정보' (Member Information) section is now populated with the following details:

이름	이상현
이메일	lsseebx@naver.com
전화번호	010-5680-3247
생년월일	2017-12-31

At the bottom right of the member information section, there is a button labeled '개인정보 수정하기' (Edit Personal Information).



이상현님 어서오세요. | 로그아웃



마이페이지
익명게시판
자유게시판
갤러리
방명록
자료검색

어서오세요! 환영합니다.

회원 정보

이름	이상현
이메일	lsseebx123@naver.com
전화번호	010-1234-3247
생년월일	2015-10-29

만약 위와 같이 정보를 수정하고 '개인정보 수정하기' 버튼을 눌렀다면 아래와 같이 반영됨을 확인해 볼 수 있다.

	ACTIVATION	ID	PW	NAME	TEL	EMAIL	BIRTH
<input type="checkbox"/>  	true	lsseebx	qQ1!12	이상현	010-1234-3247	lsseebx123@naver.com	2015-10-29

이 과정은 아래의 자바코드로 실행되며 수정이 완료된 이후에는 갱신된 정보를 이용하여 다시 마이페이지 화면을 띄운다.

```

DBMemberUpdate dbMemberUpdate = DBMemberUpdate.getDBMemberUpdate();
dbMemberUpdate.updateMemberInformation(SQL);
%>
<script type='text/javascript'>alert('수정이 완료되었습니다');</script>
<%
/* 게시글수정 완료이후 수정된 마이페이지로 이동 */
String URL3 = "./TP2_01_MyPage.jsp";
response.sendRedirect(URL3);
%>

```



### 3) 익명게시판

익명게시판은 게시글을 작성하고 등록하되 작성자의 이름은 Anonymous로 감추어 주는 기능을 한다. 하지만 게시글의 수정이나 삭제에 대한 권한은 게시글을 작성한 호스트만 가능하도록 한다. 아래는 데이터베이스에 익명게시판에 해당하는 글의 레코드가 없을 경우 출력되는 화면이다.



이상현님 어서오세요. | 로그아웃

마이페이지 익명게시판 자유게시판 갤러리 방명록 자료검색

어서오세요! 환영합니다.

제목 작성자 날짜

제목 ▼ 검색어를 입력해주세요. 검색 **게시물등록**

게시물등록을 위해 하단의 버튼을 누르면 게시물을 작성하고 등록할 수 있는 아래의 페이지로 전환된다.



이상현님 어서오세요. | 로그아웃

마이페이지 익명게시판 자유게시판 갤러리 방명록 자료검색

제목 안녕하세요

첨부파일 파일 선택 선택된 파일 없음

반갑습니다.

send **게시물등록**

위에서 제목과 내용을 작성한 후 게시물등록 버튼을 누르면 다음과 같이 데이터베이스의 board테이블에 게시글에 대한 정보가 레코드로 삽입된다.

서버: localhost ▶ 데이터베이스: webprogrammingtermproject ▶ 테이블 : board

보기 구조 SQL 검색 삽입 내보내기 Import 테이블 작업 비우기 삭제

✓ 행(레코드) 보기 0 - 0 (1 합계, 질의 실행시간 0.0004 초)

```
SELECT *
FROM 'board'
LIMIT 0 , 30
```

보기: 30 행. 시작(행)위치 0  
수평(가로) 정렬 ( 100 칸이 넘으면 헤더 반복)

+ Options

	boardname	time	host	title	content	file
<input type="checkbox"/>	AnonymousBoard	17.06.22. 11:45:44	isseebx	안녕하세요	반갑습니다.	

모두 체크 / 모두 체크안함 선택한 것을:

그리고 데이터베이스에 글이 추가되었으므로 게시판페이지로 돌아갔을 때, 아래와 같이 등록된 글이 제목, 작성자, 날짜와 함께 표시된다.

이상현님 어서오세요. | 로그아웃

마이페이지 익명게시판 자유게시판 갤러리 방명록 자료검색

어서오세요! 환영합니다.

제목	작성자	날짜
안녕하세요	Anonymous	17.06.22. 11:45:44

제목 ▼ 검색어를 입력해주세요. 검색 게시물등록

제목은 a태그로 구성되어 있으며, 이를 클릭하면 기본키(primary key)인 작성시간(time)을 통해 데이터베이스에서 해당 글의 record를 검색하여 글을 자세히 볼 수 있도록 한다.

```
buf.append("<tr>");
buf.append("<td class='td1'><a href='./TP2_08_LookStory.jsp?time="+TIME+"&boardname="+boardName"+&id="+ID+"'>"+TITLE+"</a></td>");
buf.append("<td class='td2'>"+ID+"</td>");
buf.append("<td class='td3'>"+TIME+"</td>");
buf.append("</tr>");
```

a태그를 누르면 글을 보도록 html을 구성해주는 jsp를 호출하면서 boardname과 time을 파라미터로 전달한다. 여기서 time은 게시글들을 완전하게 구별해주는 기본 키의 역할을 하며, 다음의 메소드를 통해 해당 글에 대한 정보를 알아올 수 있다.

```
public ResultSet getStoryRecord(String time) throws ClassNotFoundException, SQLException{
    Connection conn = DBconn.getConnection();
    Statement st = conn.createStatement();
    String sql = "Select * from board where time='" + time + "'";
    ResultSet rs = st.executeQuery(sql);
    return rs;
}
```

게시글을 보는 페이지인 LookStory.jsp에서는 위의 메소드를 호출하여 레코드를 전달받고, 아래와 같이 테이블에 게시글의 정보를 출력해준다.

이성현님 어서오세요. | 로그아웃



마이페이지
익명게시판
자유게시판
갤러리
방명록
자료검색

어서오세요! 환영합니다.

제목	안녕하세요	작성자	Anonymous
첨부파일		시간	17.06.22. 11:45:44
반갑습니다.			

이 후 게시글을 작성한 호스트인 경우 글을 수정하거나 삭제할 수 있도록 하였다. 만약 수정이나 삭제를 요청했을 때 권한이 있는지의 여부는 자바스크립트의 ajax수행을 통해 확인한다.

```
<input type="button" value="수정" onclick="isHost('modify')"/>
<input type="button" value="삭제" onclick="isHost('remove')"/>
<input type="button" value="목록" onclick="location.href=
<%
final String BOARDNAME = request.getParameter("boardname");
if(BOARDNAME != null){
    if(BOARDNAME.equals("AnonymousBoard")) out.print("'TP2_02_AnonymousBoard.jsp'");
    else if(BOARDNAME.equals("FreeBoard")) out.print("'TP2_03_FreeBoard.jsp'");
    else if(BOARDNAME.equals("Gallery")) out.print("'TP2_04_Gallery.jsp'");
    else out.print("#");
}
%>"/>
```

위에서 isHost의 파라미터인 modify와 remove는 만약 현재 게시글의 호스트가 맞을 경우 수행하는 동작을 의미한다. 또 목록으로 돌아가는 버튼은 이전 페이지에서 전달받은 boardname을 이용하여 돌아갈 주소를 a태그로 설정하였다.

ajax는 jsp를 호출하여 응답을 받는데, 실질적인 기능은 자바코드인 DB MemberLoad클래스의 isHost메소드에서 하며 이는 다음의 자바코드를 통해 결과를 받는다.

```
public boolean isHost(String time, String id) throws ClassNotFoundException, SQLException{
    Connection conn = DBconn.getConnection();
    Statement st = conn.createStatement();
    String sql = "Select * from board where time='" + time + "'";
    ResultSet rs = st.executeQuery(sql);
    String hostId = "";
    boolean result = false;

    if(rs.next()){
        hostId = rs.getString("host");
        if(id == null) result = false;
        else if(id.equals(hostId)) result = true;
    }
    return result;
}
```

만약 글의 호스트가 수정버튼을 눌렀다면 위 과정을 통해 다음의 글을 수정할 수 있는 페이지가 화면에 출력된다. 글을 수정하고 게시물수정 버튼을 누르면 데이터베이스에 있는 record의 필드 값이 변경된다. 다음은 게시물을 수정하는 페이지의 화면이다.

이성현님 어서오세요. | 로그아웃



마이페이지

익명게시판

자유게시판

갤러리

방명록

자료검색

제목

안녕하세요

첨부파일

파일 선택

선택된 파일 없음

반갑습니다.

취소

게시물수정

send

만약 글의 호스트가 삭제버튼을 눌렀다면 자바스크립트를 통해 정말 삭제할 것인지 클라이언트에게 confirm을 띄운다. 이 후 확인버튼을 눌렀다면 다음 쿼리문을 수행하여 데이터베이스에서 글을 제거한다.

```
conn = DBconn.getConnection();
Statement st = conn.createStatement();
final String SQL = "Delete from Board where time='" + time + "'";
st.executeUpdate(SQL);
return true;
```

글을 삭제한 이후 데이터베이스를 확인해보면 다음과 같이 데이터베이스 board테이블이 텅 비어있음을 확인해 볼 수 있다. 물론 기존에 글이 2개 이상 저장되어 있었다면 해당 글에 대한 record만 삭제된다.




추가적으로 익명게시판은 게시판에 글이 출력될 때 누가 작성했는지 알 수 없어야 한다. 하지만 글을 작성한 호스트는 글을 삭제하거나 수정할 권한을 가지고 있어야 하므로, 데이터베이스에는 host의 id가 저장되어 있되, 다음과 같이 코드를 구현하여 출력될 때는 익명으로 표시되게 하였다.

```
if(boardName.equals("AnonymousBoard")) ID = "Anonymous"; //익명게시판이면 작성자를 Anonymous로 함
```

#### 4) 자유게시판

자유게시판은 앞의 익명게시판과 매우 유사하며, 차이점이 있다면 글의 host를 Anonymous로 감추었던 익명게시판과 달리, 자유게시판은 글의 host를 직접적으로 명시한다. 다음과 같이 글을 작성하였다고 하자.

이상현님 어서오세요. | 로그아웃



마이페이지   익명게시판   **자유게시판**   갤러리   방명록   자료검색

어서오세요! 환영합니다...

send

제목 자유게시판입니다.

첨부파일  선택된 파일 없음

안녕하세요...

그러면 아래와 같이 게시글들이 목록으로 출력된다.

이상현님 어서오세요. | 로그아웃



마이페이지   익명게시판   **자유게시판**   갤러리   방명록   자료검색





어서오세요! 환영합니다...

제목   작성자   날짜

자유게시판입니다.   isseebx   17.06.22. 20:05:23

제목 ▼   검색어를 입력해주세요.   검색   **게시물등록**

그리고 데이터베이스의 board 테이블에는 다음과 같이 저장된다. 익명 게시판과 차이가 있다면 boardname 필드의 값이 자유게시판을 의미하는 FreeBoard로 저장되는 점이다.

			boardname	time	host	title	content	file
<input type="checkbox"/>			AnonymousBoard	17.06.22. 20:07:25	isseebx	익명게시판이에요!	안녕하세요	
<input type="checkbox"/>			FreeBoard	17.06.22. 20:05:23	isseebx	자유게시판입니다.	안녕하세요.!	

이외에 글을 수정하거나 삭제하는 것은 익명게시판과 동일하다.



## 5) 갤러리

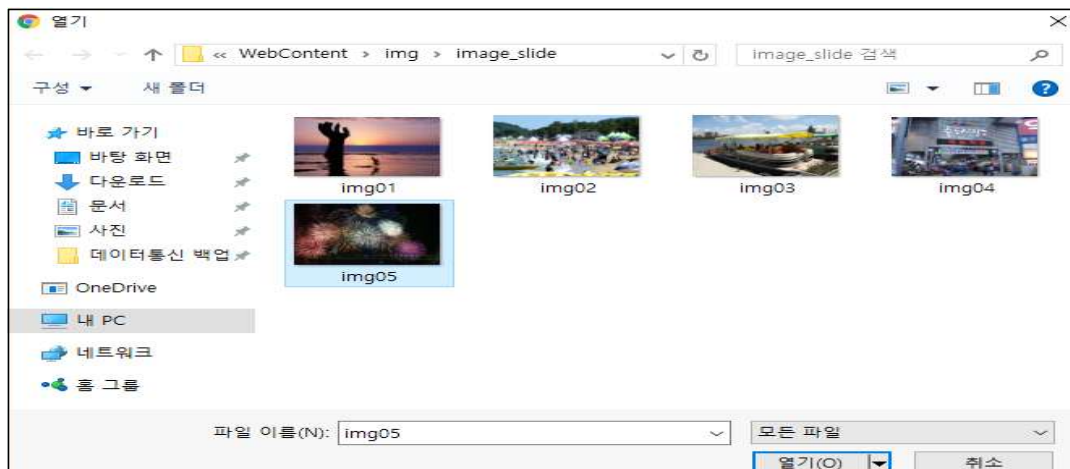
갤러리는 제목과 내용을 올리는 일반 게시판들과는 다르게 사진업로드 또한 가능하도록 구현되어 있다. 다음은 boardname=Gallery에 해당하는 board 테이블의 record들을 출력하는 갤러리의 화면이다. 물론 데이터베이스의 board 테이블에 갤러리의 글이 등록되지 않아 아무것도 출력되지 않은 상태이다.



위에서 게시물등록 버튼을 누르면 게시물을 등록할 수 있는 페이지로 전환된다. 그리고 제목과 내용은 일반 게시판들과 동일하게 작성 가능하다.



이후 파일선택 버튼을 클릭하여 업로드할 파일을 선택한다.



파일을 선택하면 다음과 같이 첨부하고자 하는 이미지의 파일명이 출력되고 게시물등록 버튼을 누른다.

이상현님 어서오세요. | 로그아웃

마이페이지 | 익명게시판 | 자유게시판 | **갤러리** | 방명록 | 자료검색

어서오세요! 환영합니다.

제목

첨부파일  img05.png

내용~

```
<form action="./InsertGalleryNewWrite.jsp" method="post" enctype="multipart/form-data">
```

여기서 파일을 서버로 전송하기위해 form태그의 enctype 속성의 값을 multipart/form-data로 설정해준다.

그리고 위 방법으로 encrypt되었기 때문에 일반적인 방법(request.getParameter 등)으로는 전달되는 정보를 서버에서 받아올 수 없다. 그래서 cos.jar 라이브러리를 이용하여 파라미터를 받아온다.

```
MultipartRequest multi = new MultipartRequest(request, savePath,
    sizeLimit, "euc-kr", new DefaultFileRenamePolicy());

// 파라미터 설정
String host = multi.getParameter("host");
String title = multi.getParameter("title");
String content = multi.getParameter("content");

// 넘어오는 파일 이름
String fileName = multi.getFilesystemName("upload");
```

또 게시물등록 버튼을 누르면 서버에서는 다음의 자바코드를 일부적으로 수행한다. 이미지 파일을 저장할 로컬 경로를 설정하고, 기본적인 호스트의 id와, 제목, 내용 그리고 파일 이름과 파일 사이즈를 추가적으로 받는다.

```
String savePath = "D:\\apache-tomcat-7.0.77\\webapps\\ROOT\\upload";
int sizeLimit = 5 * 1024 * 1024;

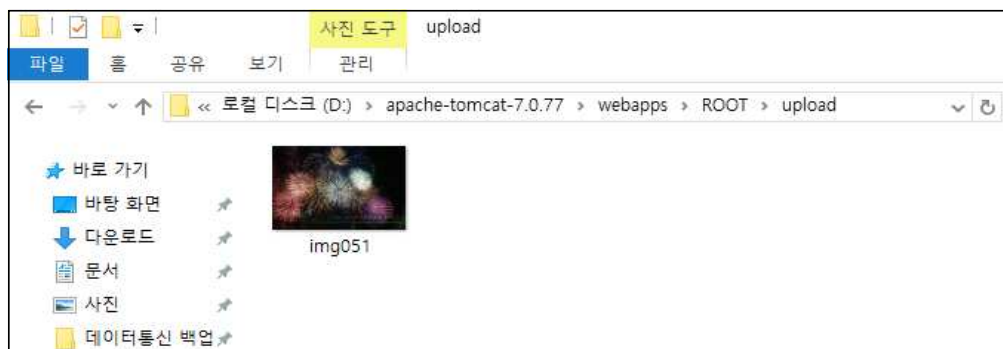
MultipartRequest multi = new MultipartRequest(request, savePath, sizeLimit);

// 파라미터 설정
String host = multi.getParameter("host");
String title = multi.getParameter("title");
String content = multi.getParameter("content");

// 넘어오는 파일 이름
String fileName = multi.getFilesystemName("upload");

// 파일 사이즈 받기
int size = (int)(new File(savePath + "/" + fileName).length());
```

이를 통해 아래 사진과 같이 로컬 경로(서버)에 이미지 파일이 업로드되어 저장되며, 자바코드로 쿼리문을 수행하여 데이터베이스에 레코드로 갤러리 글의 정보가 저장된다.



	num	host	title	content	upload	size
<input type="checkbox"/>	13	isseebx	불꽃놀이!!	우와~	img051.png	297035



## 6) 방명록

방명록은 텀프로젝트2에서 구현하였던 방명록을 그대로 가져와서 사용하였다. 커뮤니티 사이트의 성격상 잘 어울릴 것이라고 생각되어 가져왔으며, 오픈그래프프로토콜도 포함되어 있다. 기본화면은 다음과 같다.

이상현님 어서오세요. | 로그아웃



마이페이지

익명게시판

자유게시판

갤러리

방명록

자료검색


어서오세요! 환영합니다.

방명록

작성자	이상현
	안녕하세요! 들렀다가요!

방명록은 로그인을 하지 않은 사람도 글을 작성가능 하도록, 로그인을 했는지 여부를 체크하지 않으며, 작성자 또한 자신이 직접 입력이 가능하다. 등록하기 버튼을 누르면 다음과 같이 글이 테이블에 추가된다.

이상현님 어서오세요. | 로그아웃



마이페이지

익명게시판

자유게시판

갤러리

방명록

자료검색

어서오세요! 환영합니다.

방명록

작성자:이상현	안녕하세요! 들렀다가요!
<div>답글 등록하기</div>	
작성자	
내용	

등록하기

그리고 답글에서 적절한 URL을 입력하면 오픈그래프프로토콜 기능이 작동하여 해당 URL에 대한 미리보기 기능을 제공한다.

방명록	
작성자:이상현	안녕하세요! 들렀다가요!
답변 : 여기도 한번 가보세요! <a href="http://www.naver.com">www.naver.com</a> <a href="http://www.google.com">www.google.com</a> <a href="http://www.facebook.com">www.facebook.com</a>	
	Google Search the world's information,... <a href="http://www.google.com">http://www.google.com</a>
	Facebook Facebook 계정을 만들거나 로그인... <a href="https://www.facebook.com/">https://www.facebook.com/</a>
	네이버 네이버 메인에서 다양한 정보와 유... <a href="http://www.naver.com/">http://www.naver.com/</a>

## 7) 자료검색

자료검색 페이지는 전체 게시판(익명게시판, 자유게시판)에 등록되어 있는 게시글 중에 클라이언트가 원하는 자료만 출력하도록 하는 페이지이다. 이는 익명게시판, 자유게시판, 갤러리에도 구현이 되어있으며 이들에 차이점은 단순히 그것들은 해당 게시판에 한해서 글을 찾고, 자료검색 페이지에서는 전체 게시판에 한해서 글을 찾는다. 아래는 처음 페이지를 들어왔을 때 화면이며 다음과 같이 제목, 내용, 작성자를 기준으로 키워드를 포함하는 글이 있는지 확인 가능하다.



위에서 검색버튼을 눌렀을 때 수행되는 자바코드의 일부는 아래와 같다. 클라이언트가 설정한 criteria 필드가 keyword 문자열을 포함하고 있으면 해당 게시글을 검색하여 출력한다. 마지막에 order by time DESC는 최근에 작성한 글부터 출력하도록 한다.

```
String sql = "Select * from Board where boardname='" + boardName + "'";
boolean isSearch = false;
/* 검색의 경우 sql 추가 */
if(criteria != null && keyword != null){
    sql = sql + "and " + criteria + " like '%" + keyword + "%'";
    isSearch = true;
}
sql += " order by time DESC";
```



따라서 아래와 같이 작성자를 iss로 하고 검색버튼을 누르면 작성자인 host필드에 iss라는 문자열이 포함된 레코드만을 가져와서 화면에 출력한다. 물론 다른 게시판들처럼 제목이 a태그로 구성되어 있어 누르면 게시글을 볼 수 있다.

이상현님 어서오세요. | 로그아웃



마이페이지   익명게시판   자유게시판   갤러리   방명록   자료검색

어서오세요! 환영합니다.

제목	작성자	날짜
자유게시판입니다.	isseebx	17.06.22. 20:05:23

작성자 ▼ iss    검색

## 8) 메인페이지

메인페이지에서는 익명게시판, 자유게시판, 갤러리에 올라와있는 최신 글을 출력하고, 게시판을 따로 들어가지 않더라도 해당 글을 보여주는 페이지로 바로 전환이 가능하도록 구현되어 있다. 각 게시판에 게시글이 있을 때 메인페이지는 다음과 같이 출력된다.



여기서 각 게시판의 게시글들이 너무 많이 출력되면 보기좋지 않으므로 익명게시판과 자유게시판은 최대 5개, 갤러리는 최대 4개만 최신 글이 출력되도록 구현하였다. 이는 쿼리문에 limit를 추가하여 구현하였고 갤러리에 한하여 구성된 자바코드는 다음과 같다.

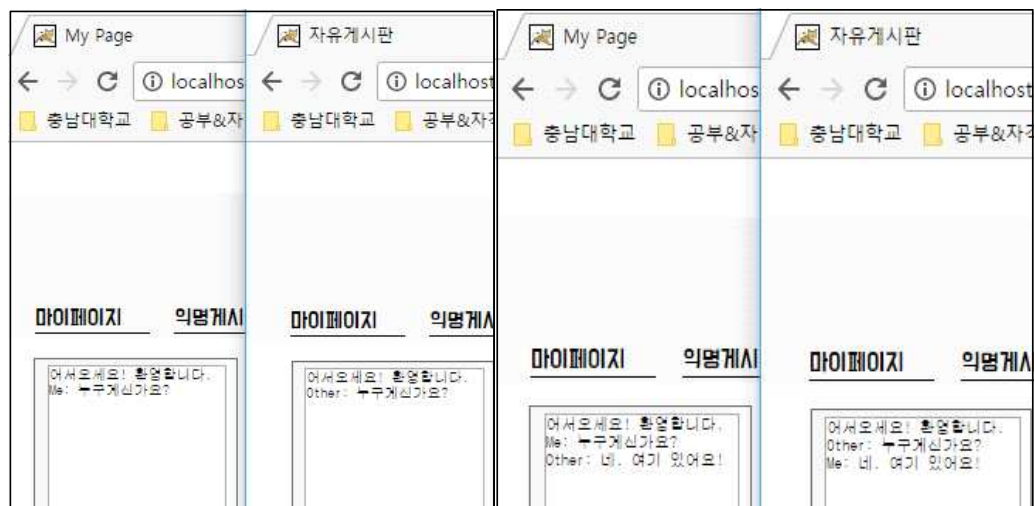
```
final int GALLERYLIMITS = 4;
String GallerySQL = "select * from upTable limit "+GALLERYLIMITS;
ResultSet rs = st.executeQuery(GallerySQL);
StringBuffer buf = new StringBuffer();
while(rs.next()){
    buf.append("<img src=/upload/" + rs.getString("upload") + ">");
}
result[index] = buf.toString();
```

## 9) 채팅기능

메인페이지를 제외한 왼쪽의 aside에는 강의시간에 배운 웹 소켓을 이용하여 클라이언트 간 소통이 가능하도록 구현하였다. 웹서비스를 이용하는 클라이언트 또는 로그인을 하지 않은 게스트라도 채팅을 할 수 있도록 하였으며, 내가 전송한 메시지는 Me: ~로 표현하고 그밖에 다른 이가 전송한 메시지는 Other: ~으로 표현한다.



2개의 웹페이지를 키고 확인해 보았을 때 다음과 같이 정상작동 함을 확인해 볼 수 있었다.

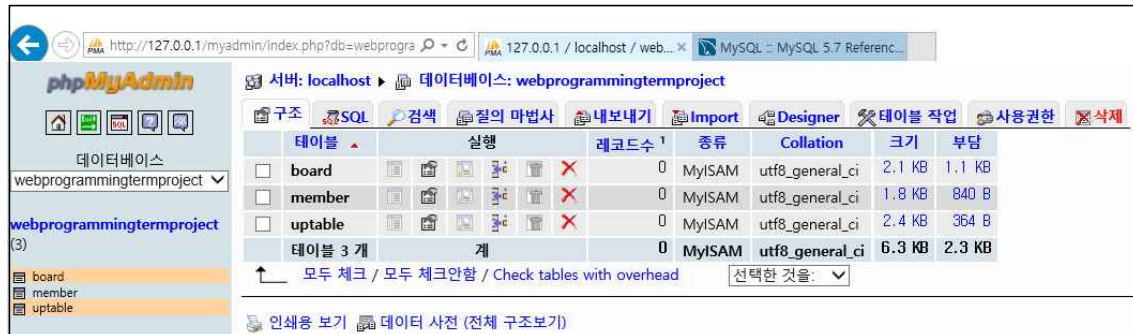


전체적인 웹소켓의 구현은 외부 사이트를 참고하여 작성하였다.

### 3. 본 시스템의 창의성

(비교할 수 있는 다른 시스템을 예를 들어 창의적이라고 생각하는 부분을 모두 적으세요.

그리고 왜 창의적이라고 생각하는지에 대한 이유도 적으세요)

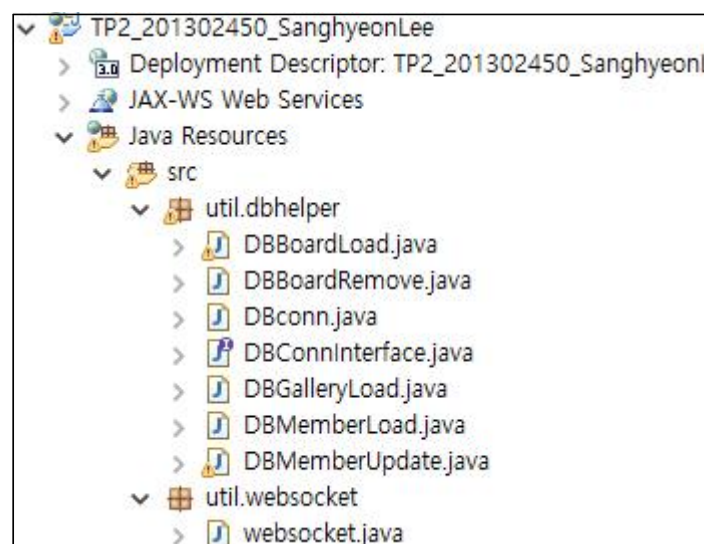


현실적으로 실용성을 가지고 체계적으로 관리하기 위해, 아직 학업에서 아직 배우지 않은 외부 데이터베이스인 MySql과 데이터베이스 시스템인 phpMyAdmin을 이용하여 웹페이지에 저장되는 모든 게시글과 회원에 대한 정보를 관리 하였다.

#### 1) 구현 기술의 창의성

\* 구현 코드의 예를 캡처한 후, 부가 설명 추가

JSP에서는 화면의 복잡성 및 readability를 고려하여 데이터베이스에 접속하는 코드를 모두 구현하지 않고, 다른 페이지에서도 사용할 만한 중복되는 코드는 따로 java파일을 두어 메소드를 정의하고, 객체를 만들고 사용하였다.



또한 다수의 클라이언트가 request를 요청하면 서버에서 각 클라이언트에 맞게 프로세스나 스레드를 생성해야 하기 때문에 서버의 부담이 많다고 생각하였다. 그래서 서버의 부담을 줄이기 위해 데이터베이스 접속을 담당하는 클래스의 객체가 요청 시마다 만들어지지 않고 최초에 만들어진 것을 계속 사용하게 하기 위해, 아래 사진처럼 클래스내에 정적인 객체변수를 하나두고 getConnection()을 통해 객체를 받아오도록 하였다. 이렇게 하면 최초 생성될 시에는 객체가 만들어져서 dbConn에 저장될 것이고, 이후 부터는 처음 생성된 객체를 계속해서 사용할 것이다.

```
public class DBConn {
    private static Connection dbConn;

    public static Connection getConnection() throws ClassNotFoundException, SQLException{
        if(dbConn == null){
            Class.forName(DBConnInterface.DRIVERNAME);
            System.out.println("Driver load");
            dbConn = DriverManager.getConnection(DBConnInterface.URL, DBConnInterface.DBI);
            System.out.println("DB connection success");
        }

        return dbConn;
    }
}
```

이를 응용하여 다른 클래스에서도 똑같이 모두 적용하였다.

```
public class DBBoardLoad {
    public static DBBoardLoad dbBoardLoad;

    public static DBBoardLoad getDBBoardLoad(){
        if(dbBoardLoad == null){
            dbBoardLoad = new DBBoardLoad();
        }
        return dbBoardLoad;
    }
}
```

## 2) 아이디어의 창의성

\* 아이디어의 예: 화면 구성의 편의성, 또는 기능의 편리함 등.

요즘에는 페이스북의 각 대학교 대나무숲과 같이 익명으로 글을 올리는 것이 대세처럼 퍼져가고 있다. 익명으로 글을 올리면 사람들은 마녀사냥을 당할 걱정이 없이 글을 편하게 올릴 수 있으므로, 이와 같이 익명으로 글을 올릴 수 있는 익명게시판을 자유게시판과 별도로 구현하였다.

또 커뮤니티 웹사이트의 특성상 단지 글을 올리는 것뿐만 아니라 실시간으로 소통을 하는 기능이 있으면 클라이언트 간 소통에 매우 편리할 것이라 생각하였다. 그래서 메인페이지를 제외한 각 페이지마다 클라이언트 간 채팅이 가능하도록 웹소켓을 이용하여 채팅을 구현하였다. 다만 이

는 최근의 기술로 직접 구현하기에는 어려움이 있었기에 외부 사이트를 많이 참고하였다.

#### **4. 참고 사이트 및 소스 (옵션)**

\* 구현과정에서 참고한 소스를 제공하는 웹 사이트와 기능 적기

예) .참고 사이트 : <http://choiyb2.tistory.com/81>

.이 사이트에서 제공하는 기능

- HTML5 Web Socket(웹 소켓)

\* 최종 보고서 제출 파일 이름 양식

**[WP]TP2\_Report\_학번\_이름.zip**