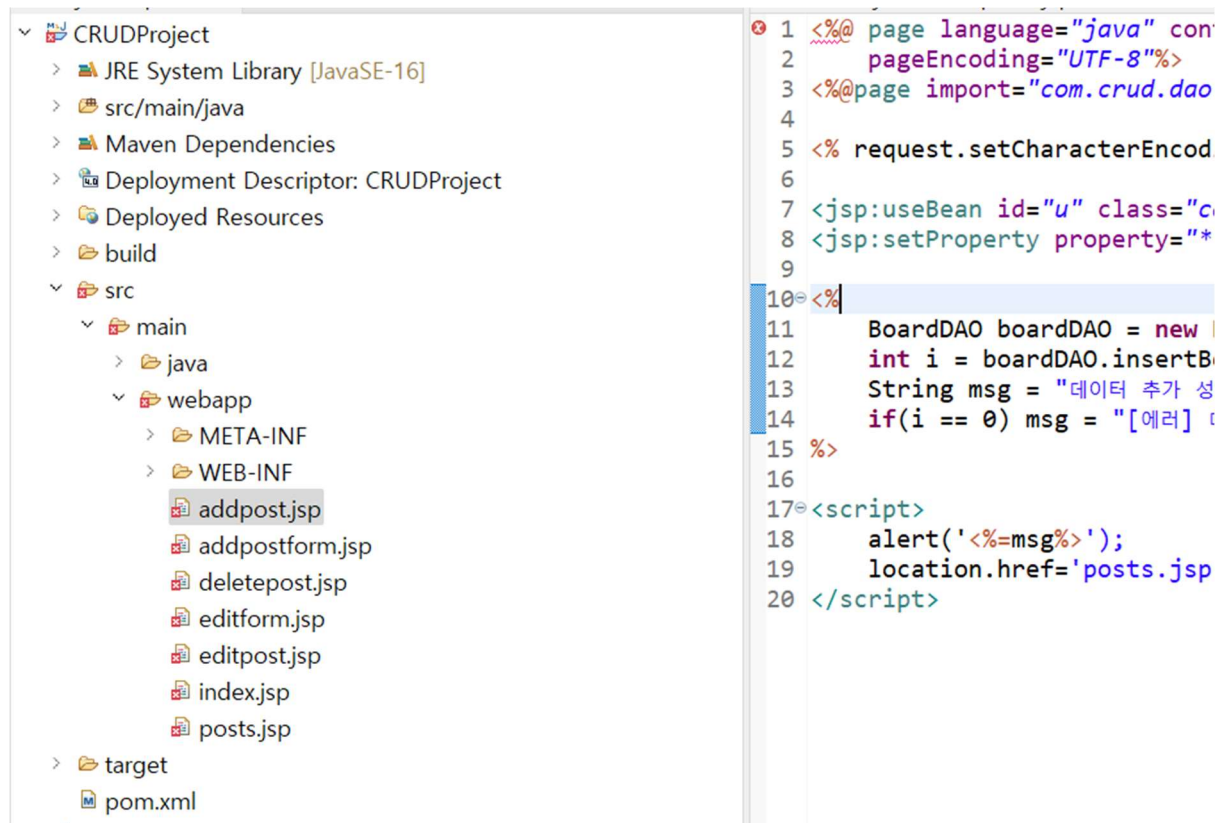


[LAB12]

1. Jsp CRUD 프로젝트 제작

다음과 같은 디렉토리 구조로 jsp 파일을 형성.



Dependency를 pom.xml에 추가.

```
</build>
<dependencies>
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.21</version>
</dependency>
<!-- https://mvnrepository.com/artifact/jstl/jstl -->
<dependency>
<groupId>jstl</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>
</dependencies>
```

JDBC로 DB 연결할 때 이미 만들어진 walab.handong.edu 로 연결 변경

```
package com.crud.common;

import java.sql.Connection;

public class JDBCUtil {
    public static Connection getConnection(){
        Connection con=null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con= DriverManager.getConnection("jdbc:mysql://walab.handong.edu/pp2021_1","pp2021_1","gbvMJnZu2kWwXgVs");
        }catch(Exception e){
            System.out.println(e);
        }
        return con;
    }
}
```

Board 테이블 구조와 동일한 명의 변수 이름으로 getter와 setter 설정

```
JDBCUtil.java  posts.jsp  CRUDProject/pom.xml  addpost.jsp  free board  BoardVO.java x
1 package com.crud.bean;
2
3 import java.util.Date;
4
5 public class BoardVO {
6     private int seq;
7     private String title;
8     private String writer;
9     private String content;
10    private Date regdate;
11    private int cnt;
12
13    public int getSeq() {
14        return seq;
15    }
16    public void setSeq(int seq) {
17        this.seq = seq;
18    }
19    public String getTitle() {
20        return title;
21    }
22    public void setTitle(String title) {
23        this.title = title;
24    }
25    public String getWriter() {
26        return writer;
27    }
}
```

2. 소스 분석 및 정리

실전 프로젝트 처음 STS로 진행했던 CRUD & MySQL 처리 문법과 동일하게 PreparedStatement를 이용해서 CRUD 기능 구현을 해놓은 자바 클래스 파일.

```
JDBCUtil.java  posts.jsp  CRUDProject/pom.xml  addpost.jsp  free board  BoardVO.java  BoardDAO.java
1 package com.crud.dao;
2
3 import java.sql.Connection;
11
12 public class BoardDAO {
13
14     Connection conn = null;
15     PreparedStatement stmt = null;
16     ResultSet rs = null;
17
18     private final String BOARD_INSERT = "insert into BOARD (title, writer, content) values (?, ?, ?)";
19     private final String BOARD_UPDATE = "update BOARD set title=?, writer=?, content=? where seq=?";
20     private final String BOARD_DELETE = "delete from BOARD where seq=?";
21     private final String BOARD_GET = "select * from BOARD where seq=?";
22     private final String BOARD_LIST = "select * from BOARD order by seq desc";
23
24 public int insertBoard(BoardVO vo) {
25     System.out.println("==> JDBC로 insertBoard() 기능 처리");
26     try {
27         conn = JDBCUtil.getConnection();
28         stmt = conn.prepareStatement(BOARD_INSERT);
29         stmt.setString(1, vo.getTitle());
30         stmt.setString(2, vo.getWriter());
31         stmt.setString(3, vo.getContent());
32         stmt.executeUpdate();
33         return 1;
34     } catch (Exception e) {
35         e.printStackTrace();
36     }
}
```

Addpost.jsp 파일

```
CRUDProject/pom.xml  addpost.jsp  free board  BoardVO.java  BoardDAO.java
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"%>
3  <%@page import="com.crud.dao.BoardDAO"%>
4
5  <% request.setCharacterEncoding("utf-8"); %>
6
7  <jsp:useBean id="u" class="com.crud.bean.BoardVO" />
8  <jsp:setProperty property="*" name="u"/>
9
10 <%
11     BoardDAO boardDAO = new BoardDAO();
12     int i = boardDAO.insertBoard(u);
13     String msg = "데이터 추가 성공 !";
14     if(i == 0) msg = "[에러] 데이터 추가 ";
15 %>
16
17 <script>
18     alert('<%=msg%>');
19     location.href='posts.jsp';
20 </script>
```

BoardDAO의 클래스 타입으로 boardDAO 오브젝트를 생성하고 insertBoard의 성공된 개수를 반환하는 i 변수를 통해서 데이터를 성공적으로 입력했는지 검증.

Addpostform.jsp 파일

```
CRUDProject/pom.xml  free board  BoardVO.java  BoardDAO.java  addpostform.jsp
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6    <meta charset="UTF-8">
7    <title>Insert title here</title>
8  </head>
9  <body>
10
11  <h1>Add New Post</h1>
12  <form action="addpost.jsp" method="post">
13  <table>
14    <tr><td>Title:</td><td><input type="text" name="title"/></td></tr>
15    <tr><td>Writer:</td><td><input type="text" name="writer"/></td></tr>
16    <tr><td>Content:</td><td><textarea cols="50" rows="5" name="content"></textarea></td></tr>
17    <tr><td><a href="posts.jsp">View All Records</a></td><td align="right"><input type="submit" value="Add Post"/></td></tr>
18  </table>
19  </form>
20
21  </body>
22  </html>
```

Html 문법으로 화면상에 표시되는 곳을 표현한 문법.

http://localhost:8080/CRUDProject/addpostform.jsp

Add New Post

Title:

Writer:

Content:

[View All Records](#)

Title:

Writer:

Content:

[View All Records](#)

각각의 타입을 text로 지정하고 textarea에 content를 작성 후 post.jsp 로 링크를 건 View all Record 와 submit 타입으로 지정된 Add Post 버튼을 추가한다.

```
CRUDProject/pom.xml free board BoardVO.java BoardDAO.java addpostform.jsp posts.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@page import="com.crud.dao.BoardDAO, com.crud.bean.BoardVO, java.util.*"%>
4 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
9 <title>free board</title>
10 <style>
11 #List {
12     font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
13     border-collapse: collapse;
14     width: 100%;
15 }
16 #List td, #List th {
17     border: 1px solid #ddd;
18     padding: 8px;
19     text-align: center;
20 }
21 #List tr:nth-child(even){background-color: #f2f2f2;}
22 #List tr:hover {background-color: #ddd;}
23 #List th {
24     padding-top: 12px;
25     padding-bottom: 12px;
26     text-align: center;
27     background-color: #0066b3;
28     color: white;
29 }
30 </style>
31 <script>
32     function delete_ok(id){
33         var a = confirm("정말로 삭제하겠습니까?");
34         if(a) location.href='deletepost.jsp?id=' + id;
35     }
36 </script>
37 </head>
```


화면 상으로 보여지는 아래와 같은 부분의 html과 css 파일을 담아놓은 파일

자유게시판						
id	Title	Writer	Content	Regdate	Edit	Delete
70				2021-11-20	Edit	Delete
69	마지막 글	전소영	저는 이제 실프를 끝내고 자러 갑니다. 내일은 기분 좋게 토커를 공부해보려 합니다. 여러분도 좋은 주말 보내세요 :)	2021-11-20	Edit	Delete
68	7,8,9	전소영	여기 댓글거 적으신 분 덕분에 배가 너무 더 매우 많이 고평정됩니다. 저는 현재 잘 수 있을까요 근데 이거 c.r.u.d 사이클로 10번씩이면 총 40번인가요 아 그냥 도합 10번인가 보군요! 희망이 보입니다. 제 눈앞에는 브이콘이 있지만, 참고 내일 아침을 기대하도록 합니다.	2021-11-20	Edit	Delete
67	마지막?	최해림	아마도 잘 수 있지 않을까요? 화이팅하세요	2021-11-20	Edit	Delete
66	5,6	전소영	이거 10번이나 어떻게 하지... addpost는 점점 내려간다... 그래도 페이지, 근데 화면 설계 되어보고싶어서 길게 적는 중인데 일일 길게 적으면 한 페이지 다 차지할 수도 있으려나 아 그러	2021-11-20	Edit	Delete

```
<body>
<h1>자유게시판</h1>
<%
    BoardDAO boardDAO = new BoardDAO();
    List<BoardVO> list = boardDAO.getBoardList();
    request.setAttribute("list",list);
%>
<table id="list" width="90%">
<tr>
    <th>Id</th>
    <th>Title</th>
    <th>Writer</th>
    <th>Content</th>
    <th>Regdate</th>
    <th>Edit</th>
    <th>Delete</th>
```

테이블을 Id, Title, Writer, Content, Register, Edit Delete 순으로 tr, th를 이용해 만든다.

```
<c:forEach items="${list}" var="u">
    <tr>
        <td>${u.getSeq()}</td>
        <td>${u.getTitle()}</td>
        <td>${u.getWriter()}</td>
        <td>${u.getContent()}</td>
        <td>${u.getRegdate()}</td>
        <td><a href="editform.jsp?id=${u.getSeq()}">Edit</a></td>
        <td><a href="javascript:delete_ok('${u.getSeq()}')">Delete</a></td>
    </tr>
</c:forEach>
</table>
<br/><a href="addpostform.jsp">Add New Post</a>
```

List 배열 요소를 순서대로 반복해서 처리할 수 있는 태그

U 라는 변수(var)에 Collection 객체 \${list} 를 하나씩 가져오며 각각의 테이블에 getter 메소드를 사용해서 화면상에 출력한다. Edit의 경우 하이퍼레퍼런스(href)를 만들어 editform.jsp 파일에 id 를 getSeq() 의 리턴값으로 주게 된다. Delete의 경우 href를 javascript 문법을 이용해 getSeq() 리턴값으로 보내고 변수 a에 confirm 창을 띄워 true false 값을 입력받고 만약 사용자가 true를 입력했다면 deletepost.jsp 파일로 입력 받은 id와 함께 보내게 된다. 이때 사용되는 location.href 는 새로운 페이지로 이동한다는 문법이다.

```
<script>
    function delete_ok(id){
        var a = confirm("정말로 삭제하겠습니까?");
        if(a) location.href='deletepost.jsp?id=' + id;
    }
</script>
```

Deletepost.jsp 파일

```
CRUDProject/po... free board BoardVO.java BoardDAO.java addpostform.jsp deletepost.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@page import="com.crud.dao.BoardDAO, com.crud.bean.BoardVO"%>
4 <%
5   String sid = request.getParameter("id");
6   if (sid != ""){
7     int id = Integer.parseInt(sid);
8     BoardVO u = new BoardVO();
9     u.setSeq(id);
10    BoardDAO boardDAO = new BoardDAO();
11    boardDAO.deleteBoard(u);
12  }
13  response.sendRedirect("posts.jsp");
14 %>
```

http 요청의 파라미터 값을 얻기 위해 request.getParameter 를 사용하여 사용자가 입력한 id가 무엇인지 알아내고 sid에 String 타입으로 저장을 한다. Integer 형변환을 통해서 id 에 정수로 저장을 하고 BoardVO 타입의 오브젝트 u를 만들어 setter 메소드로 id를 저장한다. 그리고 BoardDAO 클래스로 boardDAO 오브젝트를 하나 형성하고 해당 오브젝트 u를 deleteBoard 오브젝트를 이용해서 제거해준다.

```
// 글 삭제
public void deleteBoard(BoardVO vo) {
    System.out.println("==> JDBC로 deleteBoard() 기능 처리");
    try {
        conn = JDBCUtil.getConnection();
        stmt = conn.prepareStatement(BOARD_DELETE);
        stmt.setInt(1, vo.getSeq());
        stmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private final String BOARD_DELETE = "delete from BOARD where seq=?";
```

deleteBoard에서는 BoardVO 타입으로 넘겨받은 삭제할 id의 오브젝트를 미리 만들어 놓은 PreparedStatement를 통해서 ? 자리에 getSeq() 메소드로 채워주고 stmt를 executeUpdate하게 된다.

Editform.jsp 파일

```
free board BoardDAO.java addpostform.jsp deletepost.jsp index.jsp editform.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@page import="com.crud.dao.BoardDAO, com.crud.bean.BoardVO"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <meta charset="UTF-8">
8     <title>Edit Form</title>
9 </head>
10 <body>
11
12 <%
13     BoardDAO boardDAO = new BoardDAO();
14     String id=request.getParameter("id");
15     BoardVO u=boardDAO.getBoard(Integer.parseInt(id));
16 %>
17
18 <h1>Edit Form</h1>
19 <form action="editpost.jsp" method="post">
20     <input type="hidden" name="seq" value="<%=u.getSeq() %>" />
21     <table>
22     <tr><td>Title:</td><td><input type="text" name="title" value="<%= u.getTitle() %>" /></td></tr>
23     <tr><td>Writer:</td><td><input type="text" name="writer" value="<%= u.getWriter() %>" /></td></tr>
24     <tr><td>Content:</td><td><textarea cols="50" rows="5" name="content"><%= u.getContent() %>
25     <tr><td colspan="2"><input type="submit" value="Edit Post"/>
26     <input type="button" value="Cancel" onclick="history.back()" /></td></tr>
27     </table>
28 </form>
29
30 </body>
```

기본적으로 addpostform.jsp 파일과 동일하다. Form action 태그를 통해 form data를 서버로 보낼 때 해당 데이터가 도착할 URL을 입력하고 post 형식으로 보내게 된다. 이전 addpostform 과 다르게 수정을 진행할 때는 작성해놨던 value를 가져와야 하므로 getter 메소드를 통해 사용자가 이전에 작성했던 내용들을 화면상에 같이 출력해서 보여준다. Submit 타입으로 Edit Post를 value로 지정하고 button 타입으로는 Cancel value를 설정하여 사용자가 클릭을 하게 될 경우 뒤로 이동하도록 만들어준다.

Editpost.jsp 파일

```
addpostform.jsp deletepost.jsp index.jsp editform.jsp editpost.jsp addpost.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@page import="com.crud.dao.BoardDAO"%>
4
5 <% request.setCharacterEncoding("utf-8"); %>
6
7 <jsp:useBean id="u" class="com.crud.bean.BoardVO" />
8 <jsp:setProperty property="*" name="u"/>
9
10 <%
11   BoardDAO boardDAO = new BoardDAO();
12   int i=boardDAO.updateBoard(u);
13   response.sendRedirect("posts.jsp");
14 %>

public int updateBoard(BoardVO vo) {
    System.out.println("==> JDBC로 updateBoard() 가능 처리");
    try {
        conn = JDBCUtil.getConnection();
        stmt = conn.prepareStatement(BOARD_UPDATE);
        stmt.setString(1, vo.getTitle());
        stmt.setString(2, vo.getWriter());
        stmt.setString(3, vo.getContent());
        stmt.setInt(4, vo.getSeq());

        System.out.println(vo.getTitle() + "-" + vo.getWriter() + "-" + vo.getContent());
        stmt.executeUpdate();
        return 1;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return 0;
}
```

BoardDAO 의 클래스 타입으로 boardDAO 오브젝트를 형성하고 com.crud.bean.BoardVO 에 있는 클래스를 u 변수 명으로 id 를 입력받고 해당 u를 updateBoard 메소드를 이용해서 성공적으로 업데이트가 되면 1을 리턴받고 그렇지 않다면 0을 리턴받는다. 그리고 특정 조건에서 post.jsp 넘어가고 싶을 때 response.sendRedirect 메소드를 사용하고 이동을 한다.

[jsp action tag]

```
<jsp:forward page="second.jsp" />
```

Forward 태그로 현재 페이지를 다른 페이지로 전환할 때 사용되는 태그

```
<jsp:forward page="second.jsp">
  <jsp:param name="id" value="abcd" />
  <jsp:param name="pw" value="1234" />
</jsp:forward>
```

Forward 액션 태그와 param 태그를 통해서 다른 페이지에 데이터를 전달할 수 있다.

```
<jsp:include page="second.jsp" />
```

Jsp 페이지 내에 다른 페이지를 삽입하는 액션 태그

[jsp directive 태그]

Jsp directive는 container 에 방향이나 명령을 제공하게 된다. 특정 부분에 대해서 어떻게 처리를 할지를 컴퓨터에게 알려주는 태그가 된다. 그렇기 때문에 여러 개의 속성값(attribute)을 가진다. 또한 속성과 속성 사이에는 key-value 페어로 값을 지정할 수 있다.

Directive	Description
<%@ page ... %>	페이지 종속적인 속성들을 정의합니다. scripting language, error page, and buffering requirements 같은 것들을 말합니다.
<%@ include ... %>	로드될 때 동적으로 파일을 include합니다.
<%@ taglib ... %>	페이지에서 사용될 사용자정의 액션을 포함하는 태그 라이브러리를 선언합니다.

Page Directive 태그

기본문법은 아래와 같습니다.

```
<%@ page attribute="value" %>
```

XML형식은 아래와 같습니다.

```
<jsp:directive.page attribute="value" />
```

Include Directive 태그

문법은 아래와 같습니다.

```
<%@ include file="relative url" %>
```

XML형식은 아래와 같습니다.

```
<jsp:directive.include file="relative url" />
```

Tagible Directive 태그

문법은 아래와 같습니다.

```
<%@ taglib uri="uri" prefix="prefixOfTag" %>
```

XML형식은 아래와 같습니다.

```
<jsp:directive.taglib uri="uri" prefix="prefixOfTag" />
```

[Tag library]

JSP Standard Tag Library에서 Tag library 란 자바 코드를 캡슐화하는 기술을 직접 만드는 기술을 말한다. JSTL 은 자주 쓰는 Taglib 를 인터페이스화하고 구현한 것들을 모아둔 lib이다.

[EL expression language]

자바스크립트에서 확장된 Xpath 에서 힌트를 얻어 만들어진 언어이다. 값이 없는 변수(NULL)에 대해서 좀 더 관대하며 데이터 형 변환을 자동으로 해준다. 이러한 특징들은 문자열을 처리하는데 주요하게 사용된다. 서버로 전달될 때 항상 문자열로 전달이 되는데 서버 프로그램에서 int, boolean 으로 사용해야 될 때가 존재한다. 이럴 때 EL을 사용하면 값이 없거나 형 변환을 해야 하는 경우에 신경 쓸 필요없이 사용할 수 있는 편리함이 있다.

EL 표현식

문자

- `${true}`
- `${false}`
- `${123}`
- `${3.14}`
- `{"JAVA"}`
- `{'java'}`

EL연산자

- 산술연산자: +, -, *, /, %, mode
- 논리연산자: &&, ||, !, and, or, not
- 비교연산자: ==, !=, <, >, <=, >=, eq, ne, lt, gt, le, ge
- empty연산자: 값이 null이거나 공백문자인지를 판단하는 연산자
`${empty ""}` → true
- `${empty null}` → true

내장객체	설명
pageContext	JSP 내장 객체 pageContext와 같습니다.
pageScope	pageContext에 등록된 데이터의 이름과 값을 저장하고 있는 map객체
requestScope	HttpServletRequest에 등록된 데이터의 이름과 값을 저장하고 있는 map객체
sessionScope	HttpSession에 등록된 데이터의 이름과 값을 저장하고 있는 map객체
applicationScope	ServletContext에 등록된 데이터의 이름과 값을 저장하고 있는 map 객체

param	QueryString의 이름과 값을 저장하고 있는 map객체
paramValues	같은 이름으로 전달된 질의 문자열의 이름과 값들을 저장하고 있는 map 객체
header	요청정보 헤더의 정보를 이름과 값으로 저장하고 있는 map객체
headerValues	요청정보 헤더의 정보들을 이름과 값들을 저장하고 있는 map 객체
cookie	요청을 보낸 클라이언트의 쿠키 이름과 값을 저장하고 있는 map객체
initParam	웹 애플리케이션에 저장한 초기 파라미터의 이름과 값을 저장하고 있는 map 객체

3. 실행 테스트

Add New Post

Title:

Writer:

Content:

Create 기능

[View All Records](#) Add Post

자유게시판

Id	Title	Writer	Content	Regdate	Edit	Delete
75	12주차 실습	김현욱	Create 기능	2021-11-20	Edit	Delete

Edit Form

Title:

Writer:

Content:

Update 기능

Edit Post Cancel

Id	Title	Writer	Content	Regdate	Edit	Delete
75	12주차 실습	김현욱	Update 기능	2021-11-20	Edit	Delete

자유게시판

Id	Title	Writer	Content	Regdate	Edit	Delete
74	12주차 실습	김현욱	실습 추가하고 다시 삭제하고 추가 TEST 중입니다 (수정4)	2021-11-20	Edit	Delete

자유게시판

Id	Title	Writer	Content	Regdate	Edit	Delete
75	12주차 실습	김현욱	Update 기능	2021-11-20	Edit	Delete

localhost:8080의 메시지
정말로 삭제하겠습니까?

확인 취소

기능확인 캡처

Id	Title	Writer	Content	Regdate	Edit	Delete
73	12주차 실습	김현욱	실습 과제 TEST 중입니다	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
73	12주차 실습	김현욱	실습 과제 TEST 중입니다 (수정 1회)	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
73	12주차 실습	김현욱	실습 과제 TEST 중입니다 (수정 2회)	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
73	12주차 실습	김현욱	실습 과제 TEST 중입니다 (수정 4회)	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
72				2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
74	12주차 실습	김현욱	실습 추가하고 다시 삭제하고 추가 TEST 중입니다	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
74	12주차 실습	김현욱	실습 추가하고 다시 삭제하고 추가 TEST 중입니다 (수정1)	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
74	12주차 실습	김현욱	실습 추가하고 다시 삭제하고 추가 TEST 중입니다 (수정2)	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
74	12주차 실습	김현욱	실습 추가하고 다시 삭제하고 추가 TEST 중입니다 (수정3)	2021-11-20	Edit	Delete

Id	Title	Writer	Content	Regdate	Edit	Delete
74	12주차 실습	김현욱	실습 추가하고 다시 삭제하고 추가 TEST 중입니다 (수정4)	2021-11-20	Edit	Delete