

# 학교 교실 관리 페이지

이건수

github:HTTPS://GITHUB.COM/GUN1740/MOVIE-RESERVATION.GIT

01

기획 방향

02

제작 구조

03

개발일정/개발환경

04

개발 구조

05

구현 페이지

06

페이지 상세설명

01

# 기획방향

# 학교 특정 교실 사용 일정 공유

\* 홈페이지 방식으로 선생님들과 학생사이의 교실이동 공유

- 선생님/관리자계정 학생들/회원시스템 운영
- 캘린더 , 게시판 기능 추가

\* 설명

학교에서 사용되고 있는 특정한 교실들 (ex: 음악실 , 체육실 , 운동장 , 과학실등) 관리자 계정을 통하여 선생님들이 캘린더로 드래그하여 특정 시간 , 하루 등 예약합니다

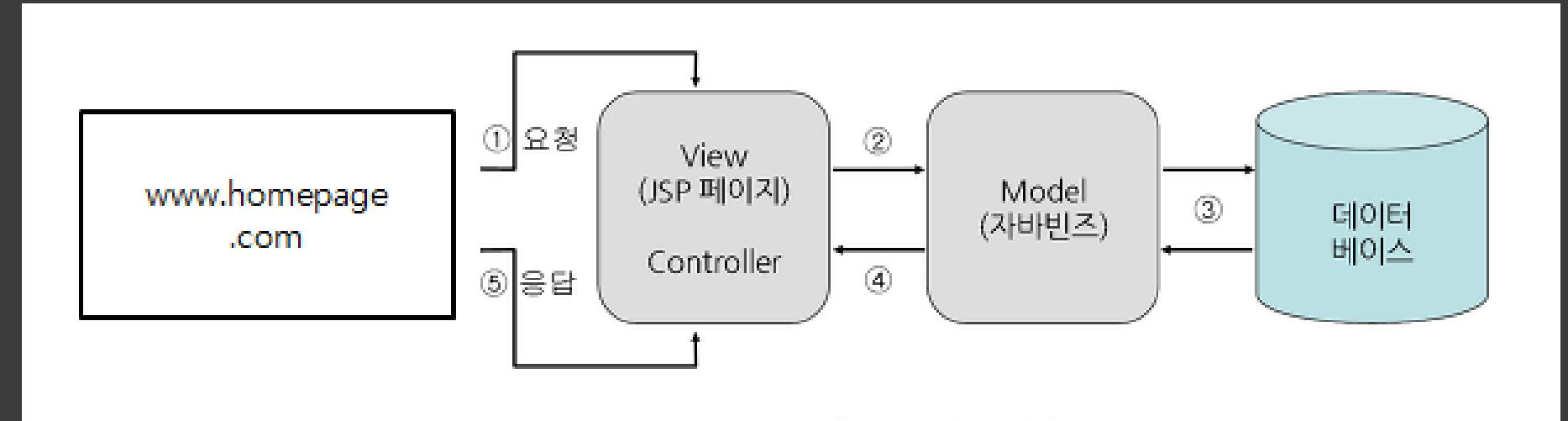
학생들은 선생님들이 예약한 캘린더 시간을 보고 이동수업때 이동합니다 또한 게시판 기능으로 건의사항기능을 추가하여 건의사항이 있을시 게시판 처럼 작성하여 사용 할 수 있습니다.

02 ■

# 제작구조

# MVC 모델로 제작

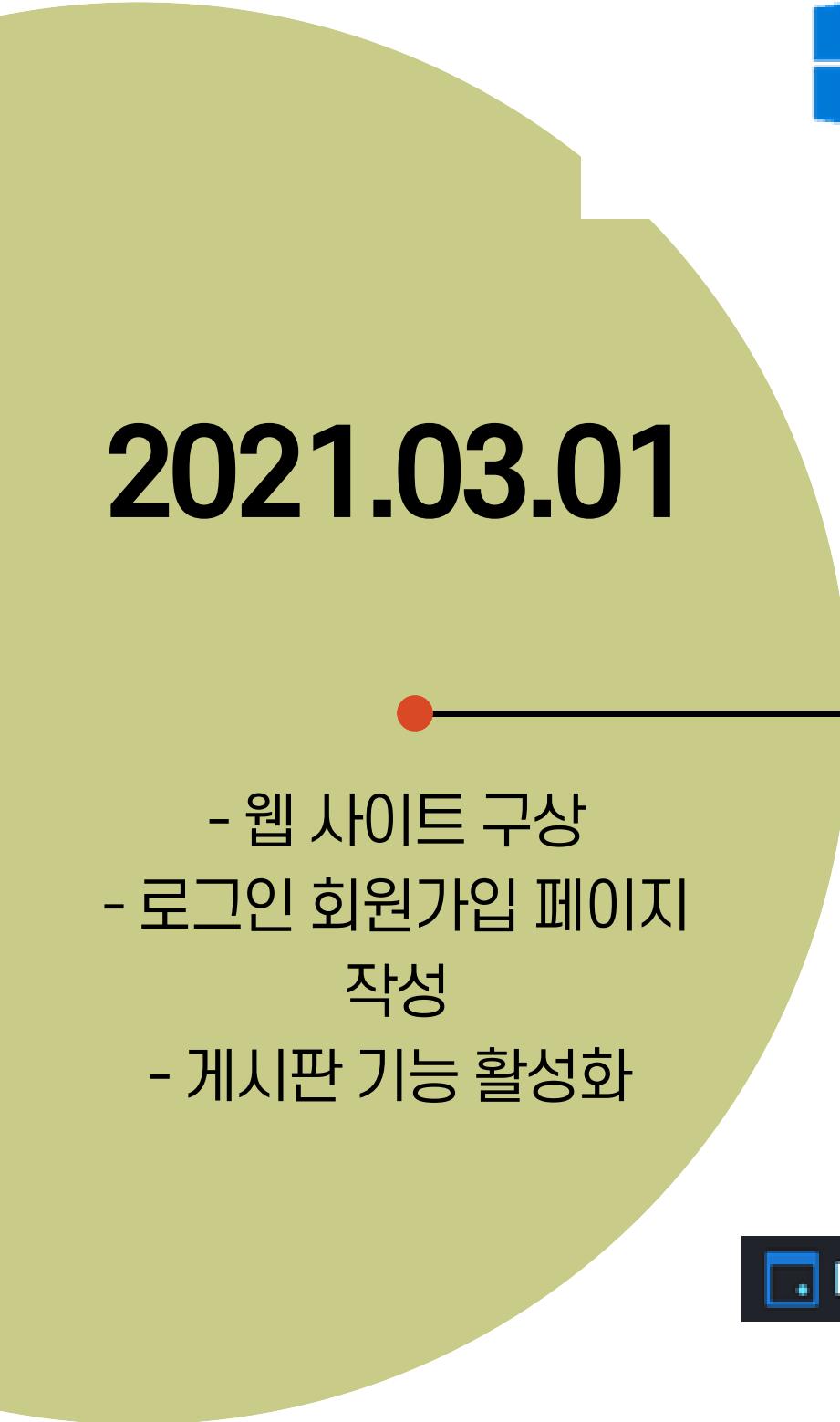
MVC(Model-View-Controller)를 사용하여 모든 클라이언트 요청과 응답을 JSP가 담당하는 구조로 사용하였습니다



이 패턴을 사용시 사용자 인터페이스로부터 비즈니스 로직을 분리하여 시각적 요소나 그 이면에서 실행되는 여러가지 로직을 서로 영향 없이 수정할수 있어 시간적으로 단축이 되고 페이지를 쉽게 구현 할 수 있습니다.

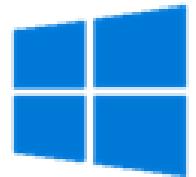
03 ■

# 개발일정 개발환경



2021.03.01

- 웹 사이트 구상
- 로그인 회원가입 페이지 작성
- 게시판 기능 활성화



Windows 10 pro



Version: 2020-12 (4.18.0)  
Build id: 20201210-1552



Bootstrap 3.3.1

 ORACLE

Oracle Database 11g



DBeaver

Dbeaver 7.2.5



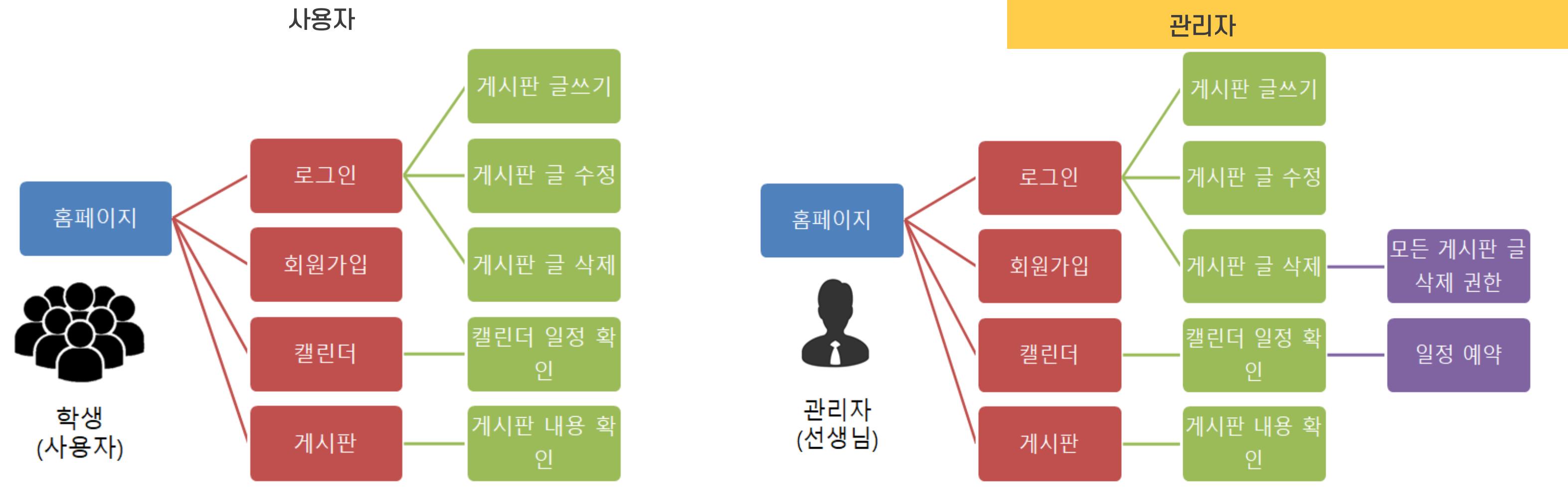
FullCalendar OpenSource



04

개발구조

# BASIC FLOW(UML)



# DB 구조

User1					
Table 명	User 정보				
NO	Column명	설명	Data Type	Nullable	PK/FK
1	UserID	아이디	varchar(20)	N	PK/FK
2	UserPassword	비밀번호	varchar(20)	N	
3	UserName	이름	varchar(20)	N	
4	UserGender	성별	varchar(20)	N	
5	UserEmail	이메일	varchar(50)	N	

CALENDAR
SCHEDULE_CODE
ABC CLASS_NAME
ABC START_DATE
ABC END_DATE
ABC ALLDAY

Calendar					
Table 명	Table 설명 캘린더 정보				
NO	Column명	설명	Data Type	Nullable	PK/FK
1	schedule_code	번호	int	N	PK/FK
2	class_name	교실이름	varchar(20)	N	
3	start_date	사용시작날짜/시간	varchar2(50)	N	
4	end_date	사용끝 날짜/시간	varchar2(50)	N	
5	allday	당일 풀 사용	varchar(50)	N	

BBS
BBSID
BBSTITLE
USERID
BBSDATE
BBSCONTENT
BBSAVAILABLE

BBS					
Table 명	Table 설명 게시판				
NO	Column명	설명	Data Type	Nullable	PK/FK
1	BBSID	번호	varchar(50)	N	PK/FK
2	BBSTITLE	제목	varchar(50)	N	
3	USERID	아이디	varchar(50)	N	
4	BBSDATE	날짜	varchar(50)	N	
5	BBSCONTENT	설명	varchar(50)	N	
6	BBSAVAILABLE	접속/비접속	varchar(50)	N	

# DIAGRAM

<b>Bbs</b>	<b>Calendar</b>	<b>User</b>	<b>UserDAO</b>	<b>CalendarDAO</b>
<pre>&lt;&lt;Java Class&gt;&gt; <b>Bbs</b>  ▫ bbsID: int ▫ bbsTitle: String ▫ userID: String ▫ bbsDate: String ▫ bbsContent: String ▫ bbsAvailable: int  <b>c</b>Bbs() <b>g</b>getBbsID():int <b>g</b>setBbsID(int):void <b>g</b>getBbsTitle():String <b>g</b>setBbsTitle(String):void <b>g</b>getUserID():String <b>g</b>setUserID(String):void <b>g</b>getBbsDate():String <b>g</b>setBbsDate(String):void <b>g</b>getBbsContent():String <b>g</b>setBbsContent(String):void <b>g</b>getBbsAvailable():int <b>g</b>setBbsAvailable(int):void</pre>	<pre>&lt;&lt;Java Class&gt;&gt; <b>Calendar</b>  ▫ schedule_code: int ▫ class_name: String ▫ userID: String ▫ start_date: String ▫ end_date: String ▫ allday: String  <b>c</b>Calendar() <b>g</b>getSchedule_code():int <b>g</b>setSchedule_code(int):void <b>g</b>getClass_name():String <b>g</b>setClass_name(String):void <b>g</b>getUserID():String <b>g</b>setUserID(String):void <b>g</b>getBbsDate():String <b>g</b>setBbsDate(String):void <b>g</b>getBbsContent():String <b>g</b>setBbsContent(String):void</pre>	<pre>&lt;&lt;Java Class&gt;&gt; <b>User</b>  ▫ userID: String ▫ userPassword: String ▫ userName: String ▫ userGender: String ▫ userEmail: String  <b>c</b>User() <b>g</b>getUserID():String <b>g</b>setUserID(String):void <b>g</b>getClassName():String <b>g</b>setClassName(String):void <b>g</b>getStart_date():String <b>g</b>setStart_date(String):void <b>g</b>getEnd_date():String <b>g</b>setEnd_date(String):void</pre>	<pre>&lt;&lt;Java Class&gt;&gt; <b>UserDAO</b>  ▫ conn: Connection ▫ pstmt: PreparedStatement ▫ rs: ResultSet  <b>c</b>UserDAO() <b>g</b>login(String,String):int <b>g</b>join(User):int</pre>	<pre>&lt;&lt;Java Class&gt;&gt; <b>CalendarDAO</b>  ▫ conn: Connection ▫ pstmt: PreparedStatement ▫ rs: ResultSet  <b>c</b>CalendarDAO() <b>g</b>Calendarwrite(int,String,String,String,String):int <b>g</b>join(User):int</pre>
			<pre>&lt;&lt;Java Class&gt;&gt; <b>BbsDAO</b>  ▫ conn: Connection ▫ rs: ResultSet  <b>c</b>BbsDAO() <b>g</b>getDate():String <b>g</b>getNext():int <b>g</b>write(String,String,String):int <b>g</b>getList(int):ArrayList&lt;Bbs&gt; <b>g</b>nextPage(int):boolean <b>g</b>getBbs(int):Bbs <b>g</b>update(int,String,String):int <b>g</b>delete(int):int</pre>	

클래스	설명
Bbs	게시판
BbsDAO	게시판DAO

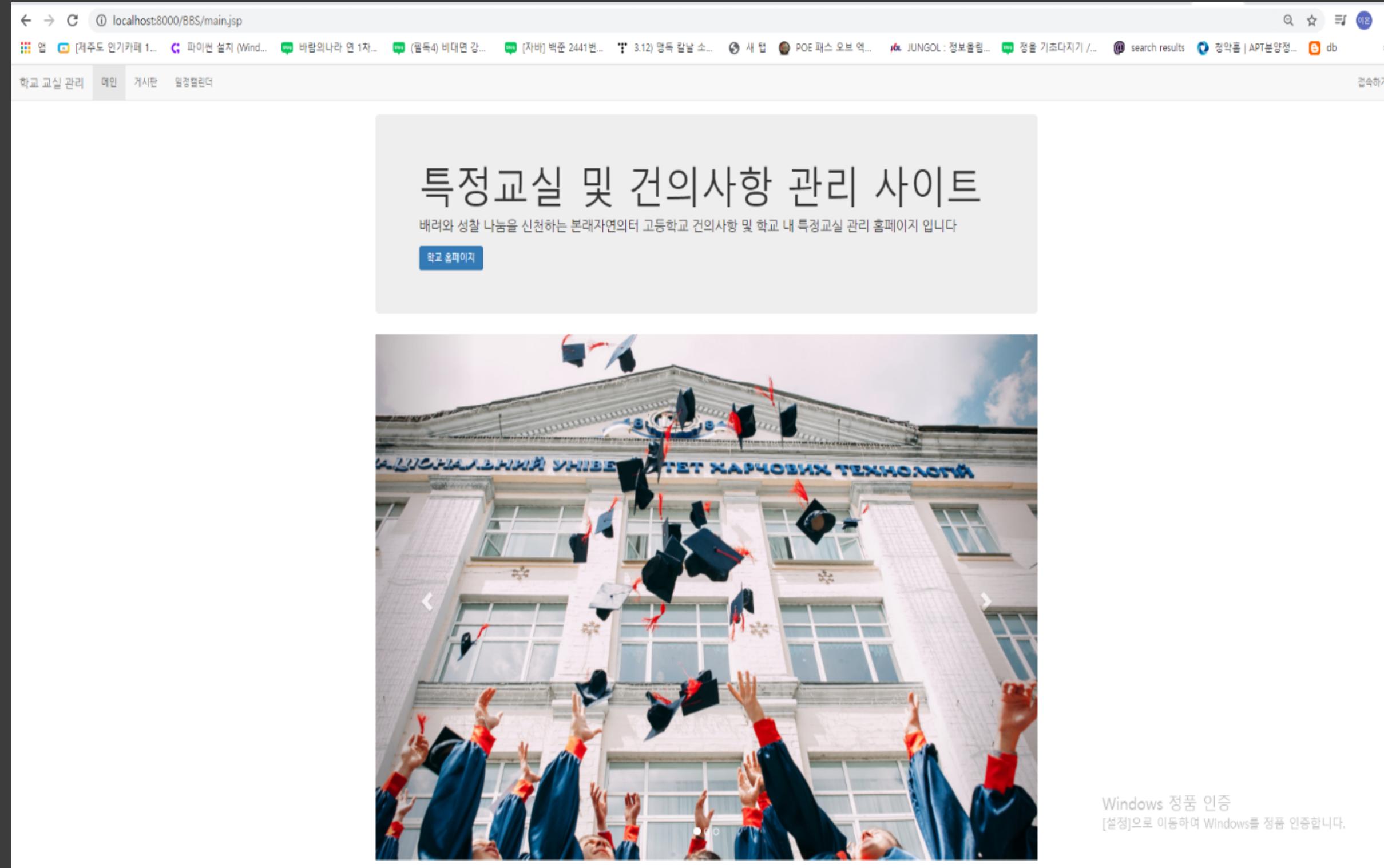
클래스	설명
User	사용자
UserDA	사용자DAO

클래스	설명
Calendar	캘린더
CalendarD	캘린더DAO

05.

# 구현페이지

# 홈페이지

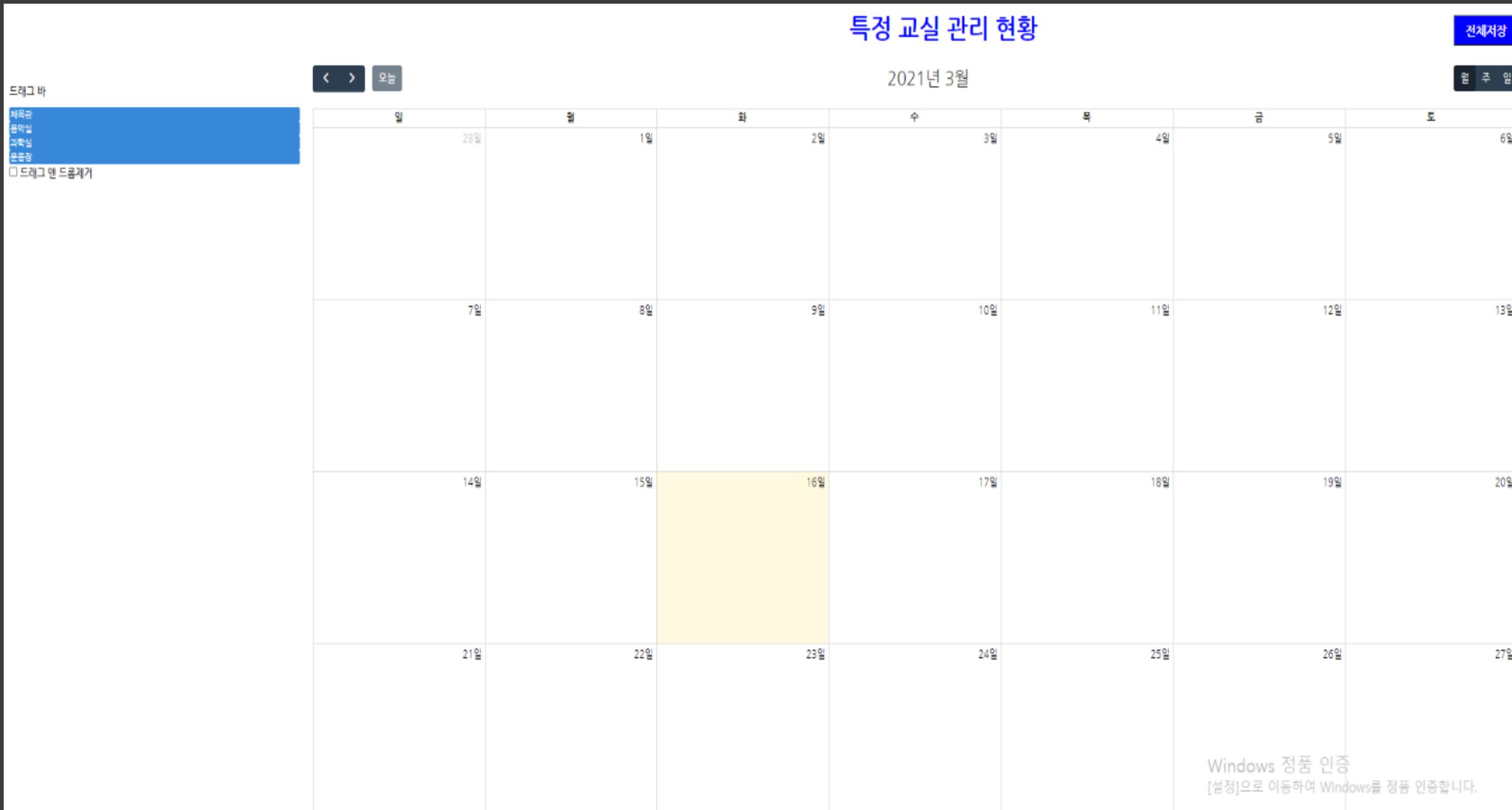


# 게시판

번호	제목	작성자	작성일
11	안녕하세요	111	2021-03-16 00:30:54
10	안녕하세요	111	2021-03-16 00:30:52
9	안녕하세요	111	2021-03-16 00:30:51
8	안녕하세요	111	2021-03-16 00:30:48
7	안녕하세요	777	2021-03-16 00:30:24
6	안녕하세요	777	2021-03-16 00:30:22
5	안녕하세요	777	2021-03-16 00:30:20
4	안녕하세요	777	2021-03-16 00:30:19
3	안녕하세요	777	2021-03-16 00:30:17
2	안녕하세요	777	2021-03-16 00:30:13

[다음](#)[글쓰기](#)

# 캘린더



06.

# 페이지 상세설명

# MAIN.JSP

```
<nav class="navbar navbar-default">
  <div class="navbar-header">
    <button type="button" class="navbar-toggle collapsed"
      data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"
      aria-expanded="false">
      <span class="icon-bar"></span> <span class="icon-bar"></span> <span
        class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="main.jsp">학교 교실 관리</a>
  </div>
  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
      <li class="active"><a href="main.jsp">메인</a></li>
      <li><a href="bbs.jsp">게시판</a></li>
      <li><a href="calendar.jsp">일정캘린더</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li class="dropdown"><a href="#" class="dropdown-toggle"
        data-toggle="dropdown" role="button" aria-haspopup="true"
        aria-expanded="false">접속하기<span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="Login.jsp">로그인</a></li>
          <li><a href="join.jsp">회원가입</a></li>
        </ul>
      </li>
    </ul>
  </div>
</nav>
```



## NAVBAR

navbar 기능을 사용하여 사용자의 화면 크기에 맞게 변경  
하도록 진행 하였습니다

큰 화면으로 접속하면 접속하기라는 클릭창이 나오고 접  
클릭하게 되면 로그인, 회원가입 기능만 보이게 됩니다

작은 화면으로 접속하게 dropdown 기능을 사용하여 3  
개의 작은 줄이 형성되고 클릭시 메인, 게시판, 일정캘린더  
등 전체 구성 메뉴들을 확인 할 수 있습니다.

# MAIN.JSP

```

<div class = "container">
<div class="jumbotron">
<div class="container">
<h1> 특정교실 및 건의사항 관리 사이트 </h1>
<p>배려와 성찰 나눔을 신천하는 본래자연의터 고등학교 건의사항 및 학교 내 특정교실 관리 홈페이지 입니다</p>
<p><a class="btn btn-primary btn-pull" href="http://www.gwangju-highschool.hs.kr/main.php" role="button">학교 홈페이지</a></p>
</div>
</div>
</div>
<div class="container">
<div id="myCarousel" class="carousel slide" data-ride="carousel">
<ol class="carousel-indicators">
<li data-target="#myCarousel" data-slide-to="0" class="active"></li>
<li data-target="#myCarousel" data-slide-to="1" ></li>
<li data-target="#myCarousel" data-slide-to="2" ></li>
</ol>
<div class="carousel-inner">
<div class="item active">
<img src = "images/1.jpg">
</div>
<div class="item">
<img src = "images/2.jpg">
</div>
<div class="item">
<img src = "images/3.jpg">
</div>
</div>
<a class="left carousel-control" href="#myCarousel" data-slide="prev">
<span class="glyphicon glyphicon-chevron-left"></span>
</a>
<a class="right carousel-control" href="#myCarousel" data-slide="next">
<span class="glyphicon glyphicon-chevron-right"></span>
</a>

```



- btn btn primary btn pull 기능을 사용하여 '학교 홈페이지'를 클릭할 때 학교 홈페이지로 이동할 수 있게 구성하였습니다.
- item 클래스를 사용하여 이미지를 불러오고 carousel slide 를 사용하여 사진이 슬라이드로 넘어갈 수 있게 구성 하였습니다 item

# LOGIN.JSP, LOGINACTION.JSP, LOGOUT.JSP

로그인 화면

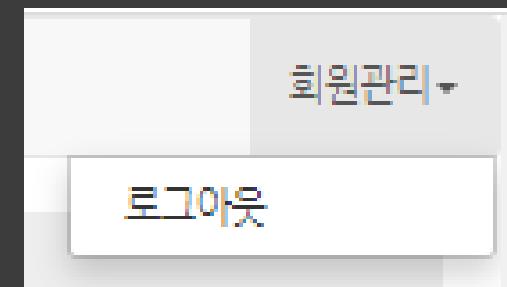



```
UserDAO userDAO = new UserDAO();
int result = userDAO.login(user.getUserID(), user.getPassword());
if(result ==1){
    session.setAttribute("userID", user.getUserID());
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("location.href='main.jsp'");
    script.println("</script>");
}
else if(result==0){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('비밀번호가 틀립니다.')");
    script.println("history.back()");
    script.println("</script>");
}
else if(result==-1){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('존재하지 않는 아이디입니다.')");
    script.println("history.back()");
    script.println("</script>");
}
else if(result== -2){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('데이터 오류가 발생했습니다')");
    script.println("history.back()");
    script.println("</script>");
}
```

```
<div class = "container">
<div class = "col-lg-4"></div>
<div class = "col-lg-4">
<div class = "jumbotron" style="padding-top: 20px">
<form method="post" action="loginAction.jsp">
<h3 style="text-align: center;">로그인 화면</h3>
<div class="form-group">
<input type = "text" class="form-control" placeholder="아이디" name="userID" maxlength="20">
</div>
<div class="form-group">
<input type = "password" class="form-control" placeholder="비밀번호" name="userPassword" maxlength="20">
</div>
<input type="submit" class="btn btn-primary form-control" value="로그인">
</form>
</div>
</div>
```

```
public int login(String userID, String userPassword) {
    String SQL = "SELECT userPassword FROM user1 WHERE userID = ?";
    try {
        pstmt = conn.prepareStatement(SQL);
        pstmt.setString(1, userID);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            if(rs.getString(1).equals(userPassword))
                return 1;
            else
                return 0;
        }
        return -1;
    } catch(Exception e) {
        e.printStackTrace();
    }
    return -2;
}
```

```
<%>
session.invalidate();
%>
<script>
    location.href = 'main.jsp';
</script>
</body>
</html>
```



UserDAO 에 login 메소드 만들어 SQL문장으로 DB에 있는 USER ID를 찾습니다  
 rs.getString(1).equals(userPassword) 기능으로 DB에 있는 값을 찾아서 result 값으로 변경합니다.

result 값을 LoginAction 에 UserDAO 불러오고 result 값이 1이면 접속완료 0이면 비밀번호가 오류 -1이면 존재하지 않는 아이디 -2면 데이터베이스 오류로 작성 하였습니다

로그인이 되고나면 회원관리 창에 로그아웃버튼이 활성화가 되고 클릭시 로그아웃 됩니다

# JOIN.JSP, JOINACTION.JSP

```

if(userID != null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('이미 로그인이 되어 있습니다.'");
    script.println("location.href = 'main.jsp'");
    script.println("</script>");
}

if(user.getUserID()==null || user.getUserPassword()==null || user.getUserName()==null ||
user.getUserGender()==null || user.getUserEmail()==null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('입력이 안 된 사항이 있습니다.'");
    script.println("history.back()");
    script.println("</script>");

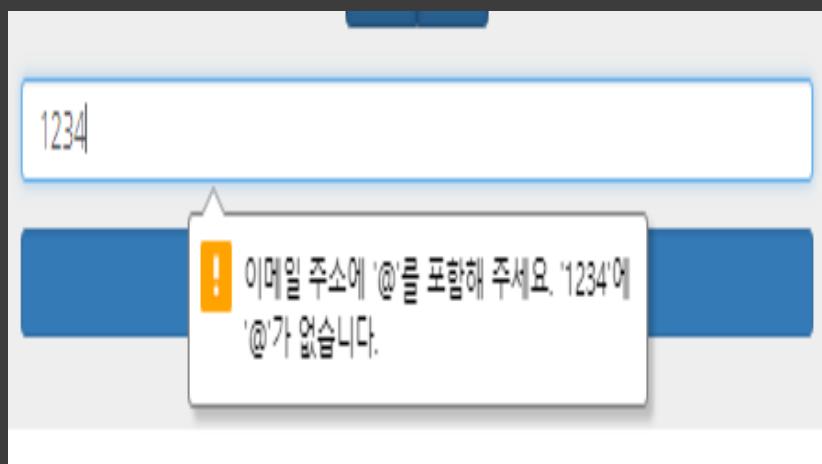
} else{
    UserDAO userDAO = new UserDAO();
    int result = userDAO.join(user);
    if(result ==-1){
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('이미 존재하는 아이디입니다')");
        script.println("history.back()");
        script.println("</script>");

    } else {
        session.setAttribute("userID", user.getUserID());
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("location.href = 'main.jsp'");
        script.println("</script>");
```

```

<div class="jumbotron" style="padding-top: 20px">
<form method="post" action="joinAction.jsp">
    <h3 style="text-align: center;">회원가입 화면</h3>
    <div class="form-group">
        <input type="text" class="form-control" placeholder="아이디"
               name="userID" maxlength="20">
    </div>
    <div class="form-group">
        <input type="password" class="form-control" placeholder="비밀번호"
               name="userPassword" maxlength="20">
    </div>
    <div class="form-group">
        <input type="text" class="form-control" placeholder="이름"
               name="userName" maxlength="20">
    </div>
    <div class="form-group" style="text-align: center;">
        <div class="btn-group" data-toggle="buttons">
            <label class="btn btn-primary active"> <input
                type="radio" name="userGender" autocomplete="off" value="남자"
                checked>남자
            </label>
            <label class="btn btn-primary active"> <input
                type="radio" name="userGender" autocomplete="off" value="여자">여자
            </label>
        </div>
    </div>

    <div class="form-group">
        <input type="email" class="form-control" placeholder="이메일"
               name="userEmail" maxlength="20">
    </div>
```



localhost:8000 내용:  
입력이 안 된 사항이 있습니다.

localhost:8000 내용:  
이미 존재하는 아이디입니다

UserDAO join 메소드를 생성하고 회원가입에 입력한 TEXT 문서를 POST 방식을 사용하여 데이터를 넘기고 SQL문장을 사용하여 USER1 테이블에 입력 됩니다

input type을 email 방식으로 사용하여 이메일 양식이 아니면 사진 문구가 나올 수 있게 구성하였습니다.

5개 양식중 1개라도 입력을 하지 않고 회원가입을 클릭시 user.getUser(회원가입항목) == null 인지 확인하여 "입력이 안 된 사항이 있습니다" 라는 얼럿을 호출하게 됩니다.

else문구에 UserDAO 클래스를 호출하여 result 값으로 -1이 나오면 존재하는 아이디가 있다라는 얼럿창이 호출됩니다.

회원가입이 정상적으로 완료되면 main.jsp로 이동될수 있도록 작성 하였습니다.

# LOGOUTACTION.JSP

```

if(userID != null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('이미 로그인이 되어 있습니다.'");
    script.println("location.href = 'main.jsp'");
    script.println("</script>");
}

if(user.getUserID()==null || user.getUserPassword()==null || user.getUserName()==null ||
user.getUserGender()==null || user.getUserEmail()==null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('입력이 안 된 사항이 있습니다.'");
    script.println("history.back()");
    script.println("</script>");

} else{
    UserDAO userDAO = new UserDAO();
    int result = userDAO.join(user);
    if(result ==-1){
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('이미 존재하는 아이디입니다')");
        script.println("history.back()");
        script.println("</script>");

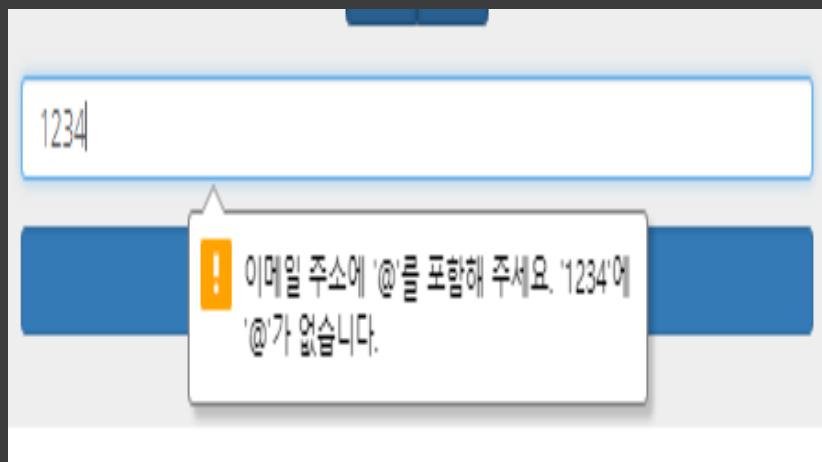
    } else {
        session.setAttribute("userID", user.getUserID());
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("location.href = 'main.jsp'");
        script.println("</script>");
```

```

<div class="jumbotron" style="padding-top: 20px">
<form method="post" action="joinAction.jsp">
    <h3 style="text-align: center;">회원가입 화면</h3>
    <div class="form-group">
        <input type="text" class="form-control" placeholder="아이디"
               name="userID" maxlength="20">
    </div>
    <div class="form-group">
        <input type="password" class="form-control" placeholder="비밀번호"
               name="userPassword" maxlength="20">
    </div>
    <div class="form-group">
        <input type="text" class="form-control" placeholder="이름"
               name="userName" maxlength="20">
    </div>
    <div class="form-group" style="text-align: center;">
        <div class="btn-group" data-toggle="buttons">
            <label class="btn btn-primary active"> <input
                type="radio" name="userGender" autocomplete="off" value="남자"
                checked>남자
            </label>
            <label class="btn btn-primary active"> <input
                type="radio" name="userGender" autocomplete="off" value="여자">여자
            </label>
        </div>
    </div>
    <div class="form-group">
        <input type="email" class="form-control" placeholder="이메일"
               name="userEmail" maxlength="20">
    </div>
```

회원가입 화면

아이디  
비밀번호  
이름  
남자 여자  
이메일  
회원가입



localhost:8000 내용:  
입력이 안 된 사항이 있습니다.

localhost:8000 내용:  
이미 존재하는 아이디입니다

확인

UserDAO join 메소드를 생성하고 회원가입에 입력한 TEXT 문서를 POST 방식을 사용하여 데이터를 넘기고 SQL문장을 사용하여 USER1 테이블에 입력 됩니다

input type을 email 방식으로 사용하여 이메일 양식이 아니면 사진 문구가 나올 수 있게 구성하였습니다.

5개 양식중 1개라도 입력을 하지 않고 회원가입을 클릭시 user.getUser(회원가입항목) == null 인지 확인하여 "입력이 안 된 사항이 있습니다" 라는 얼럿을 호출하게 됩니다.

else문구에 userDAO클래스를 호출하여 result 값으로 -1이 나오면 존재하는 아이디가 있다라는 얼럿창이 호출됩니다.

회원가입이 정상적으로 완료되면 main.jsp로 이동될수 있도록 작성 하였습니다.

# BBS.JSP

```

<div class="container">
<div class="row">
<table class = "table table-striped" style="text-align : center; border: 1px solid #dddddd">
<thead>
<tr>
<th style = "background-color:#eeeeee; text-align: center;">번호</th>
<th style = "background-color:#eeeeee; text-align: center;">제목</th>
<th style = "background-color:#eeeeee; text-align: center;">작성자</th>
<th style = "background-color:#eeeeee; text-align: center;">작성일</th>
</tr>
</thead>
<tbody>
<%
BbsDAO bbsDAO = new BbsDAO();
ArrayList<Bbs> list = bbsDAO.getList(pageNumber);
for(int i=0; i<list.size(); i++){
%>
<tr><%= list.get(i).getBbsID() %></td>
<td><a href="view.jsp?bbsID=<%= list.get(i).getBbsID()%>"><%= list.get(i).getBbsTitle().replaceAll(" ", "&nbsp;" )>
<td><%= list.get(i).getUserID() %></td>
<td><%= list.get(i).getBbsDate() %></td>
</tr>
<%
}
%>

```

번호	제목	작성자	작성일
23	777	777	2021-03-16 02:05:11
22	777	777	2021-03-16 02:05:08
21	777	777	2021-03-16 02:05:04
20	777	777	2021-03-16 02:05:01
19	777	777	2021-03-16 02:04:57
18	777	777	2021-03-16 02:04:54
17	777	777	2021-03-16 02:04:51
16	111	111	2021-03-16 02:04:23
15	111	111	2021-03-16 02:04:20
14	111	111	2021-03-16 02:04:17

[글쓰기]

[다음]

```

public ArrayList<Bbs> getList(int pageNumber){
    String SQL = "SELECT * FROM (SELECT * FROM BBS WHERE bbsID < ? "
                + "AND bbsAvailable = 1 ORDER BY bbsID DESC) WHERE ROWNUM <= 10";
    ArrayList<Bbs> list = new ArrayList<Bbs>();
    try {
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, getNext()-(pageNumber-1)*10);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            Bbs bbs = new Bbs();
            bbs.setBbsID(rs.getInt(1));
            bbs.setBbsTitle(rs.getString(2));
            bbs.setUserID(rs.getString(3));
            bbs.setBbsDate(rs.getString(4));
            bbs.setBbsContent(rs.getString(5));
            bbs.setBbsAvailable(rs.getInt(6));
            list.add(bbs);
        }
    } catch(Exception e) {
        e.printStackTrace();
    }
    return list;
}

```

```

public boolean nextPage(int pageNumber) {
    String SQL = "SELECT * FROM BBS WHERE bbsID < ? AND bbsAvailable=1";
    try {
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, getNext()-(pageNumber -1) *10);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            return true;
        }
    } catch(Exception e) {
        e.printStackTrace();
    }
    return false;
}

```

```

if(pageNumber !=1){
%>
<a href="bbs.jsp?pageNumber=<%=pageNumber -1%>" 
class="btn btn-success btn-arrow-left">이전</a>
<%
} if(bbsDAO.nextPage(pageNumber+1)){
%>
<a href="bbs.jsp?pageNumber=<%=pageNumber +1%>" 
class="btn btn-success btn-arrow-left">다음</a>
<%
}

```

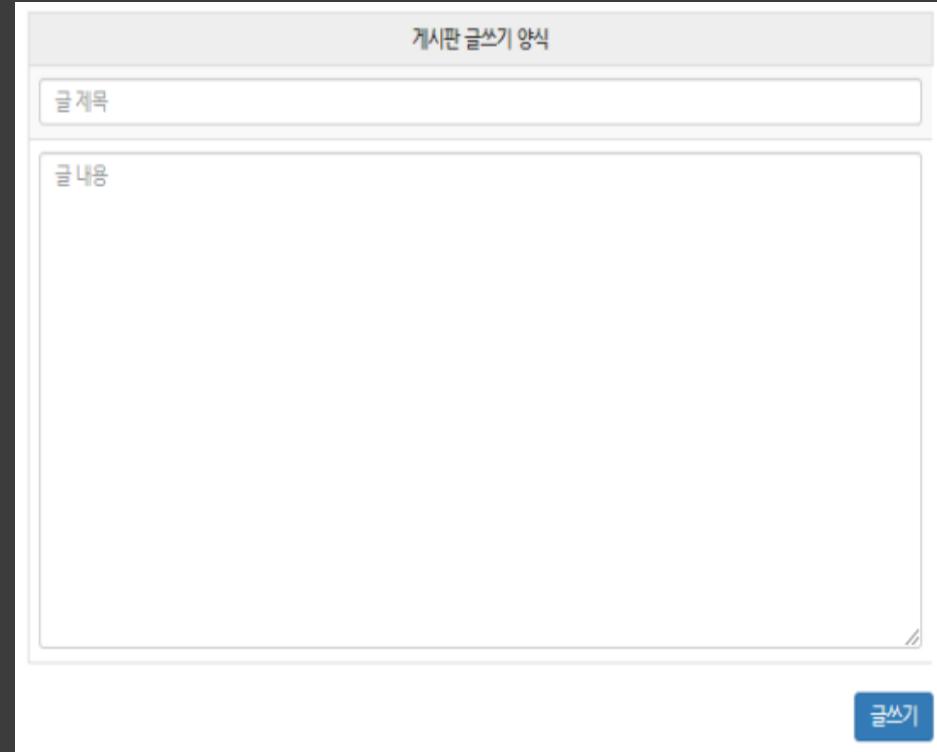
bbs.jsp에 번호,제목,작성자,작성일을 보일수 있는 테이블을 작성하였습니다. BbsDAO에 ArrayList 기능을 사용하고 sql rownum을 사용하여 게시물을 최신글로 10개만 보이게 작성하였고 nextPage라는 boolean 메소드를 생성하여 11개가 될때 제일 이전에 작성된 글부터 뒤로 넘어가도록 만들었습니다. 다음버튼을 누르면 이전에 있는 글부터 볼수 있고 이전버튼을 누를시 최신글을 확인 할 수 있습니다. 제목값에 replaceAll를 작성하여 특수기호들이 작성되도 또한 크로스사이트 스크립트를 방지 하도록 하였습니다.

# WRITE.JSP, WRITEACTION.JSP

```
<div class="container">
  <div class="row">
    <form method="post" action="writeAction.jsp">
      <table class="table table-striped"
        style="text-align: center; border: 1px solid #dddddd">
        <thead>
          <tr>
            <th colspan="2"
              style="background-color: #eeeeee; text-align: center;">게시판
              글쓰기 양식</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td><input type="text" class="form-control"
              placeholder="글 제목" name="bbsTitle" maxlength="50"></td>
            </tr>
            <tr>
              <td><textarea class="form-control" placeholder="글 내용"
                name="bbsContent" maxlength="2048" style="height: 350px;"></textarea></td>
            </tr>
          </tbody>
        </table>
        <input type="submit" class="btn btn-primary pull-right" value="글쓰기">
      </form>
```

```
String userID =null;
if(session.getAttribute("userID") !=null){
  userID = (String) session.getAttribute("userID");
}
if(userID == null){
  PrintWriter script = response.getWriter();
  script.println("<script>");
  script.println("alert('로그인을 하세요.'");
  script.println("location.href = 'login.jsp'");
  script.println("</script>");
}
else{
  if(bbs.getBbsTitle()==null || bbs.getBbsContent()==null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('입력이 안 된 사항이 있습니다.'");
    script.println("history.back()");
    script.println("</script>");
  }
  else{
    BbsDAO bbsDAO = new BbsDAO();
    int result = bbsDAO.write(bbs.getBbsTitle(), userID, bbs.getBbsContent());
    if(result ==-1){
      PrintWriter script = response.getWriter();
      script.println("<script>");
      script.println("alert('글쓰기에 실패했습니다')");
      script.println("history.back()");
      script.println("</script>");
    }
  }
}

public int write (String bbsTitle, String userID , String bbsContent) {
  String SQL = "INSERT INTO BBS VALUES (?, ?, ?, ?, ?, ?, ?)";
  try {
    PreparedStatement pstmt = conn.prepareStatement(SQL);
    pstmt.setInt(1, getNext());
    pstmt.setString(2, bbsTitle);
    pstmt.setString(3, userID);
    pstmt.setString(4, getDate());
    pstmt.setString(5, bbsContent);
    pstmt.setInt(6, 1);
    return pstmt.executeUpdate();
  }catch(Exception e) {
    e.printStackTrace();
  }
  return -1;
}
```



Write.jsp로 글제목, 글내용, 글쓰기 버튼을 구현하였고 Bbs.DAO를 이용하여 글쓰기가 완료되면 DB BBS테이블로 입력됩니다. 로그인을 안하고 글쓰기를 하면 로그인을 하세요라는 얼럿이 발생하고 로그인 창으로 넘어가게 됩니다.

글제목, 글내용에 replaceAll()를 작성하여 특수기호들이 작성되도 또한 크로스사이트 스크립트를 방지하도록 하였습니다

# VIEW.JSP,DELETEACTION.JSP

게시판글보기 양식	
글제목	777
작성자	777
작성일자	2021-03-16 02:05:11
내용	777
목록	수정
삭제	글쓰기

게시판글보기 양식	
글제목	111
작성자	111
작성일자	2021-03-16 02:04:23
내용	1111
목록	글쓰기

```

<a href = "bbs.jsp" class="btn btn-primary">목록</a>
<%
if(userID !=null && userID.equals(bbs.getUserID())){
%>
<a href = "update.jsp?bbsID=<%=bbsID %>" class="btn btn-primary">수정</a>
<a onclick="return confirm('정말 삭제하시겠습니까?')" href = "deleteAction.jsp?bbsID=<%=bbsID %>">삭제</a>
<br>
<%
}
%>
<input type="submit" class="btn btn-primary pull-right" value="글쓰기">

public int delete(int bbsID) {
    String SQL = "UPDATE BBS SET bbsAvailable= 0 WHERE bbsID =?";
    try {
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, bbsID);

        return pstmt.executeUpdate();
    }catch(Exception e) {
        e.printStackTrace();
    }
    return -1;
}
}

```

```

if (result == -1) {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('글 삭제에 실패했습니다')");
    script.println("history.back()");
    script.println("</script>");
} else {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("location.href = 'bbs.jsp'");
    script.println("</script>");
}

```

view.jsp는 게시판에 작성된 글을 클릭하여 내용을 확인 할 수 있는 기능을 구현 하였습니다

게시판에 작성된 로그인이 되어 있지 않거나 bbsID 와 현재 접속된 UserID 가 같지 않을 경우는 아래 그림과 같이 목록 글쓰기 버튼만 활성화 하였고 bbsID 와 UserID가 같을 경우 수정과 삭제 버튼을 활성화 했습니다.

bbs테이블에 있는 bbsAvailable는 실시간 사용자가 접속해있으면 0 아니면 1을 생성합니다 삭제버튼을 누를시 얼럿창이 활성화 되고 확인을 누르면 BbsDA O에 delete메소드에 SQL문구로 DB에 있는 데이터를 삭제합니다.

# UPDATE.JSP, UPDATEACTION.JSP

게시판 글 수정 양식

안녕하세요

```

if(bbs.getBbsTitle()==null || bbs.getBbsContent()==null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('입력이 안 된 사항이 있습니다.'");
    script.println("history.back()");
    script.println("</script>");

} else{
    BbsDAO bbsDAO = new BbsDAO();
    int result = bbsDAO.write(bbs.getBbsTitle(), userID, bbs.getBbsContent());
    if(result == -1){
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('글쓰기에 실패했습니다')");
        script.println("history.back()");
        script.println("</script>");
    } else {

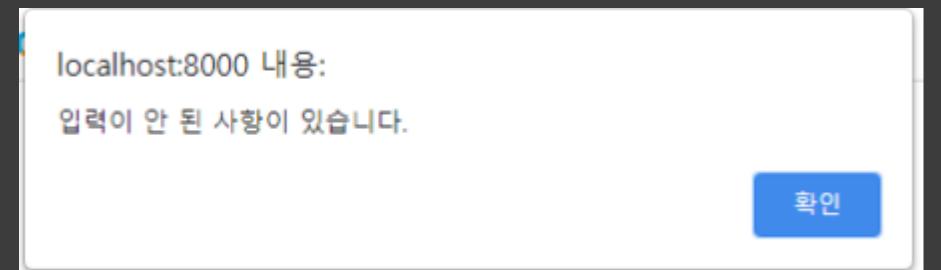
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("location.href = 'bbs.jsp'");
        script.println("</script>");
    }
}

```

```

<tr>
    <td><input type="text" class="form-control"
        placeholder="글 제목" name="bbsTitle" maxlength="50"
        value="<%bbs.getBbsTitle()%>"></td>
</tr>
<tr>
    <td><textarea class="form-control" placeholder="글 내용"
        name="bbsContent" maxlength="2048" style="height: 350px;">
        <%bbs.getBbsContent()%></textarea></td>
</tr>

```



view.jsp에서 제공하는 수정 버튼을 눌러서 활성화가 되면  
기존에 있는 글들을 update.jsp에서 불러옵니다

```

public int update(int bbsID, String bbsTitle, String bbsContent) {
    String SQL = "UPDATE BBS SET bbsTitle=? , bbsContent =? WHERE bbsID =?";
    try {
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        pstmt.setString(1, bbsTitle);
        pstmt.setString(2, bbsContent);
        pstmt.setInt(3, bbsID);

        return pstmt.executeUpdate();

    }catch(Exception e) {
        e.printStackTrace();
    }
    return -1;
}

```

그 이후 수정하고 viewAction.jsp에서 타이틀과 내용 둘중에  
하나라도 공백으로 입력이 되었을 경우 입력이 안된 사항이  
있습니다라는 얼럿이 나오도록 구성 하였습니다

타이틀과 내용 두개다 정상적으로 수정이 되면 BbsDAO에  
update 메소드에서 SQL 문장을 사용하여 업데이트를 진행  
하고 다시 게시판으로 들어가면 수정된 문장으로 확인 할 수  
있습니다.

# CALENDAR.JSP



**특정 교실 관리 현황**

전체저장

드래그 바

일	월	화	수	목	금	토
28일	1일	2일	3일	4일	5일	6일
7일	8일	9일	10일	11일	12일	13일
14일	15일	16일	17일	18일	19일	20일
21일	22일	23일	24일	25일	26일	27일
28일	29일	30일	31일	1일	2일	3일
4일	5일	6일	7일	8일	9일	10일

체육관  
음악실  
과학실  
운동장

□ 드래그 앤 드롭제거

# OPENSOURCE

FullCalendar 이라는 캘린더 ope  
nsource를 사용하였습니다

**특정 교실 관리 현황**

전체저장

드래그 바

	3, 14. (일)	3, 15. (월)	3, 16. (화)	3, 17. (수)	3, 18. (목)	3, 19. (금)	3, 20. (토)
증일							
오전 6시							
오전 7시			오전 6:30 체육관				
오전 8시				오전 8:30 음악실	오전 8:30 음악실		
오전 9시				오전 9시			
오전 10시							
오전 11시							
오후 12시							
오후 1시							

# CALENDAR.JSP

```

var calendar = null;
document.addEventListener('DOMContentLoaded', function() {
    var Calendar = FullCalendar.Calendar;
    var Draggable = FullCalendar.Draggable;

    var containerEl = document.getElementById('external-events');
    var calendarEl = document.getElementById('calendar');
    var checkbox = document.getElementById('drop-remove');
    new Draggable(containerEl, {
        itemSelector : '.fc-event',
        eventData : function(eventEl) {
            return {
                title : eventEl.innerText
            };
        }
    });

    calendar = new Calendar(calendarEl, {
        headerToolbar: {
            left: 'prev,next today',
            center: 'title',
            right: 'dayGridMonth,timeGridWeek,timeGridDay'
        },
        editable: true,
        droppable: true,
        drop: function(info) {
            if (checkbox.checked) {
                info.draggedEl.parentNode.removeChild(info.draggedEl);
            }
        },
        locale : 'ko'
    });

    calendar.render();
});

```

```

<strong>드래그 바</strong>
</p>

<div>
    class='fc-event fc-h-event fc-daygrid-event fc-daygrid-block-event'
    <div class='fc-event-main'>체육관</div>
</div>
<div>
    class='fc-event fc-h-event fc-daygrid-event fc-daygrid-block-event'
    <div class='fc-event-main'>음악실</div>
</div>
<div>
    class='fc-event fc-h-event fc-daygrid-event fc-daygrid-block-event'
    <div class='fc-event-main'>과학실</div>
</div>
<div>
    class='fc-event fc-h-event fc-daygrid-event fc-daygrid-block-event'
    <div class='fc-event-main'>운동장</div>
</div>

function allSave(){
    var allEvent = calendar.getEvents();
    console.log(allEvent);

    var events = new Array();
    for(var i=0; i < allEvent.length; i++){
        var obj = new Object();
        obj.title = allEvent[i]._def.title;
        obj.allDay = allEvent[i]._def.allDay;
        obj.start = allEvent[i]._instance.range.start;
        obj.end = allEvent[i]._instance.range.end;

        events.push(obj);
    }
    var jsondata = JSON.stringify(events);
    console.log(jsondata);
    savedata(jsondata);
}

```

▶ (3) [EventApi, EventApi, EventApi]	<a href="#">calendar.jsp:126</a>
[{"title":"음악실","allDay":false,"start":"2021-03-16T08:30:00.000Z","end":"2021-03-16T09:30:00.000Z"}, {"title":"체육관","allDay":false,"start":"2021-03-16T06:30:00.000Z","end":"2021-03-16T07:30:00.000Z"}, {"title":"음악실","allDay":false,"start":"2021-03-18T08:30:00.000Z","end":"2021-03-18T09:30:00.000Z"}]	<a href="#">calendar.jsp:140</a>

기존 navbar를 사용하였고 캘린더(calendar), 다음달/이전달(headerToolbar), 일/주/월(headerToolbar) 기능들을 활성화 하였습니다

먼저 fullcalendar에서 제공하는 main.js,main.css를 폴더에 넣고 locale:'ko'를 사용하여 한글화 하였고 var checkbox를 사용하여 드래그바를 만들어서 캘린더에 놓을 수 있도록 만들었습니다.

function allSave메소드를 만들어서 전체 저장을 누르면 Array배열 형태로 값을 받게 되고 console.log를 활용하여 값이 정상적으로 Array배열에 들어오는지 확인 합니다.

# JSP 기능 정리

bbs.jsp
calendar.jsp
calendarAction.jsp
deleteAction.jsp
index.jsp
join.jsp
joinAction.jsp
login.jsp
loginAction.jsp
logoutAction.jsp
main.jsp
update.jsp
updateAction.jsp
view.jsp
write.jsp
writeAction.jsp

jsp	기능
bbs	게시판
calendar	캘린더
calendarAction	캘린더 기능
index	메인화면으로 이동
join	회원가입
joinAction	회원가입 기능
login	로그인

jsp	기능
loginAction	로그인 기능
logoutAction	로그아웃 기능
main	홈페이지 메인화면
update	게시물 글 수정
update	게시물 글 수정 기능
view	게시물 글 보기
delectAction	게시물 삭제

jsp	기능
write	게시물 글쓰기
WriteAction	게시물 글쓰기 기능

# THANK YOU

이건수

github  
GITHUB: [HTTPS://GITHUB.COM/GU-N1740/MOVIE-RESERVATION.GIT](https://github.com/GU-N1740/MOVIE-RESERVATION.GIT)