

BITCAMP JEJU

Web MVC

1조 박훈 이병무 장서현 전형진

01

메뉴 구조
ER 다이어그램

02

스토리 보드

03

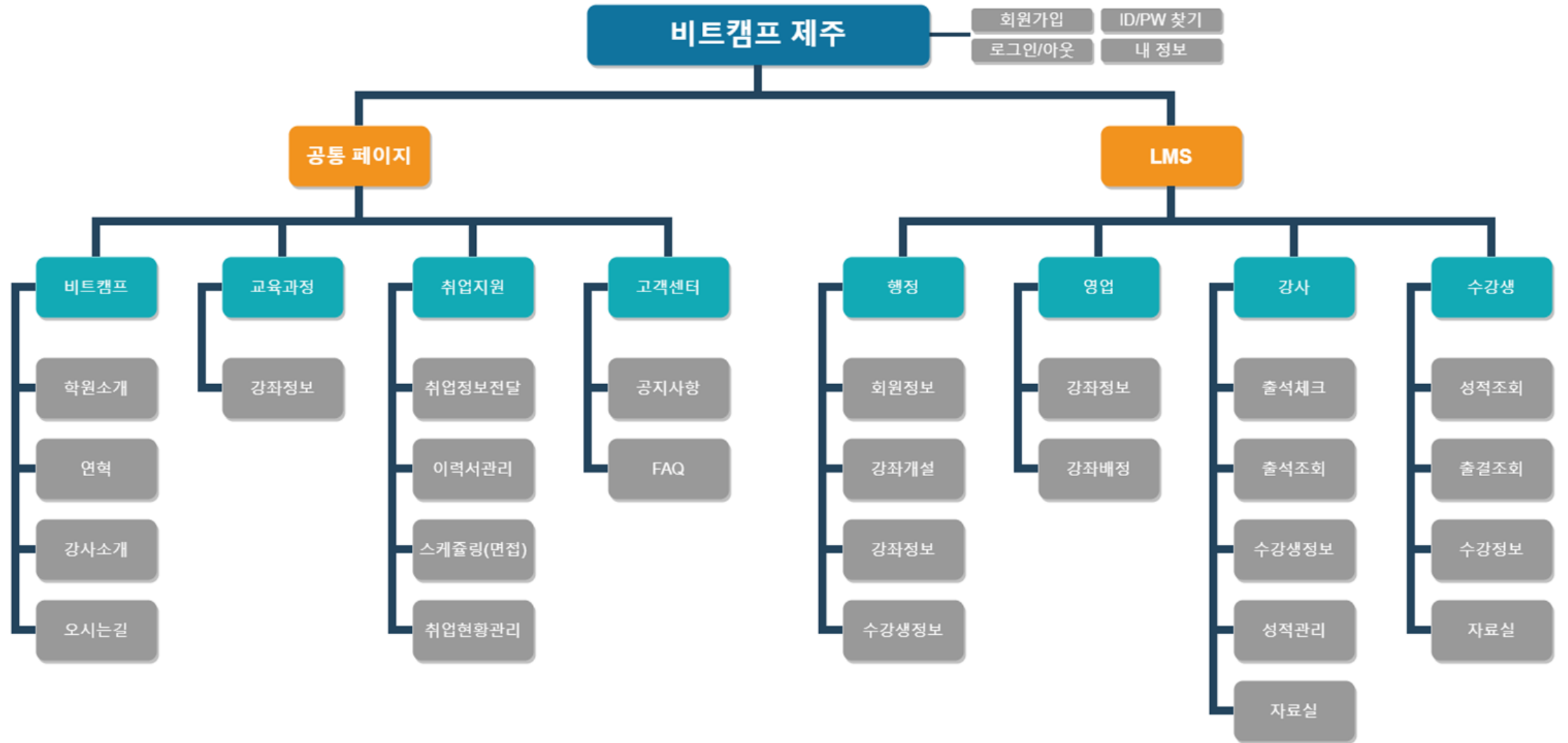
클래스 다이어
그램

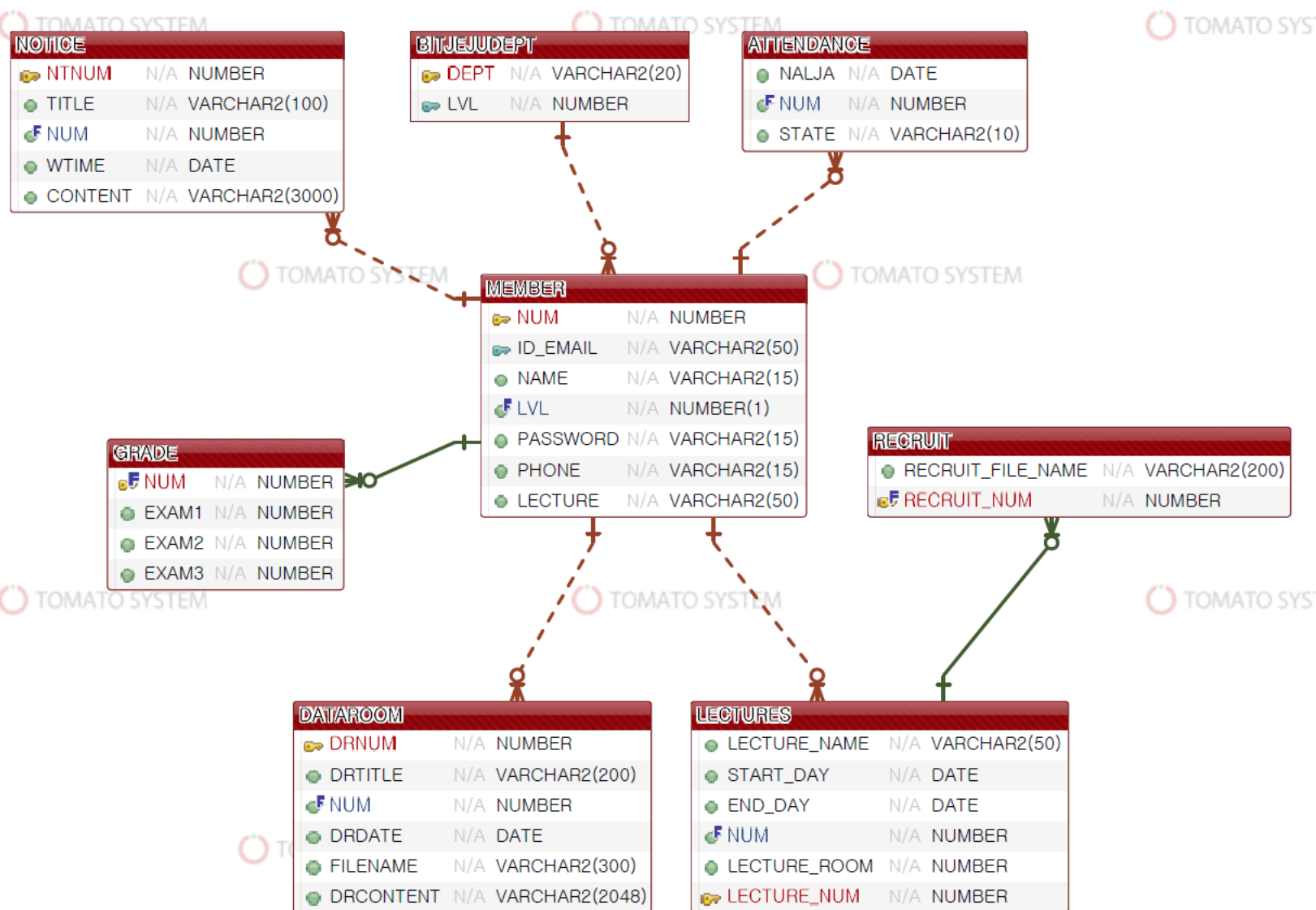
04

시퀀스 다이어
그램

05

구현

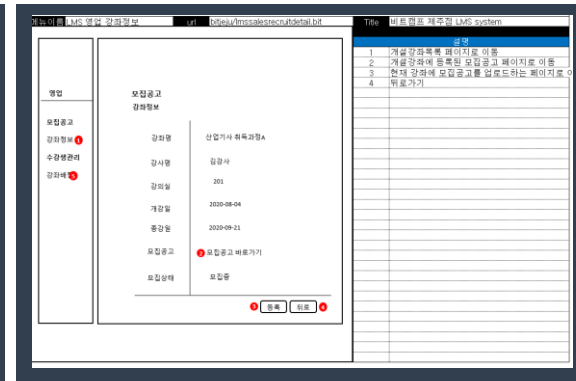
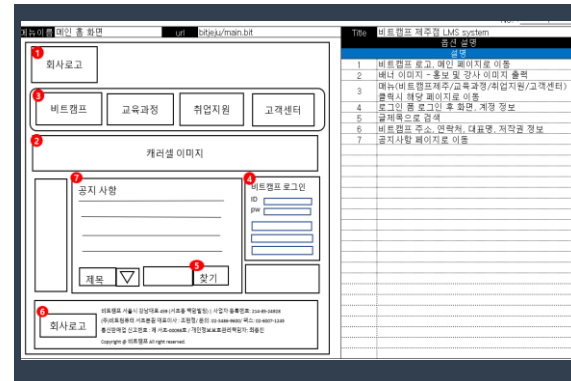




화면설계

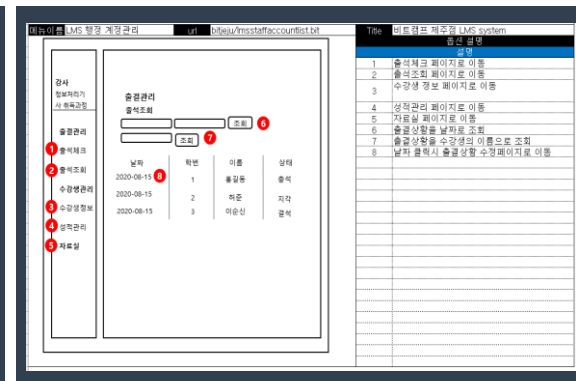
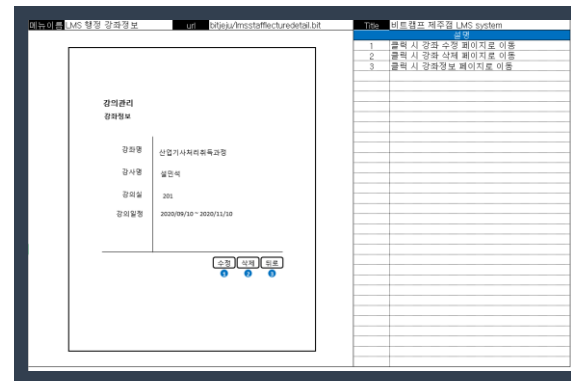
메인 화면

메뉴 : 비트캠프 교육과정 취업지원 고객센터
로그인



행정

계정관리 강좌개설 수강생관리



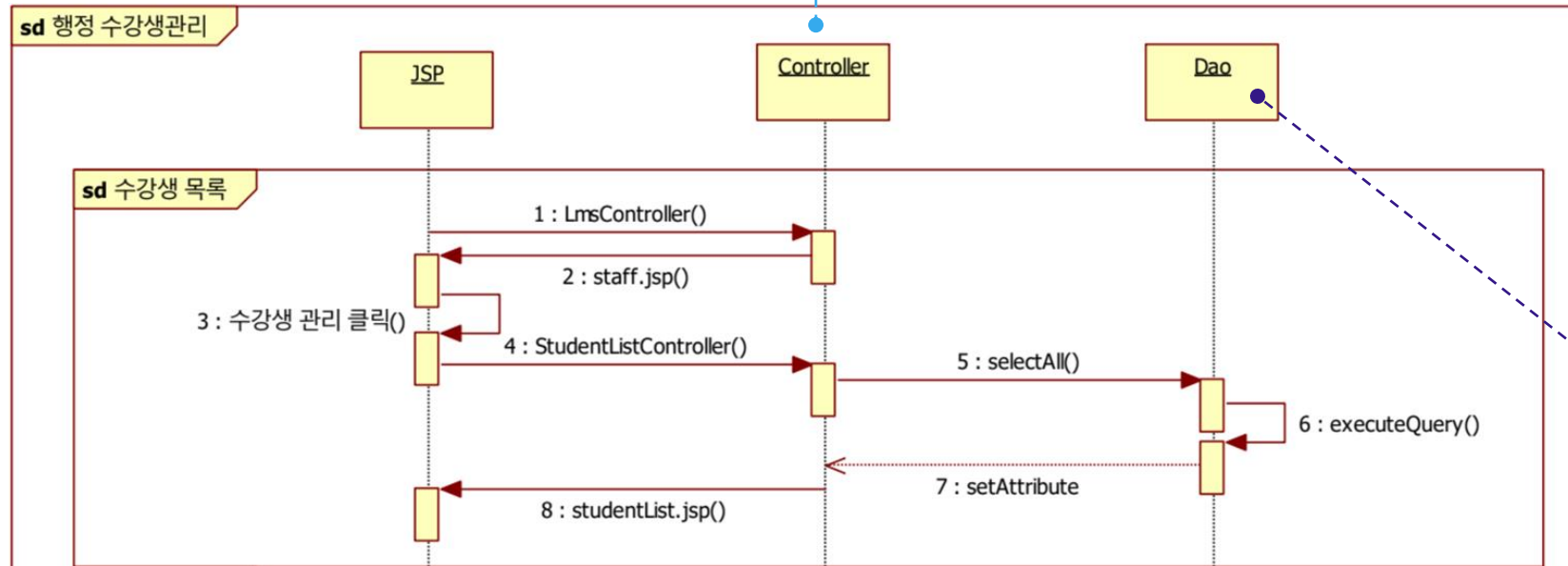
Bitcamp JEJU

영업

모집공고 강좌배정

강사, 수강생

출결 관리 성적관리 자료실



시퀀스 다이어그램

예시

페이지 별로 나누어 작성

행정, 영업, 강사, 수강생, 메인
페이지
총 다섯개의 시퀀스 다이어그램 작성

라이프라인

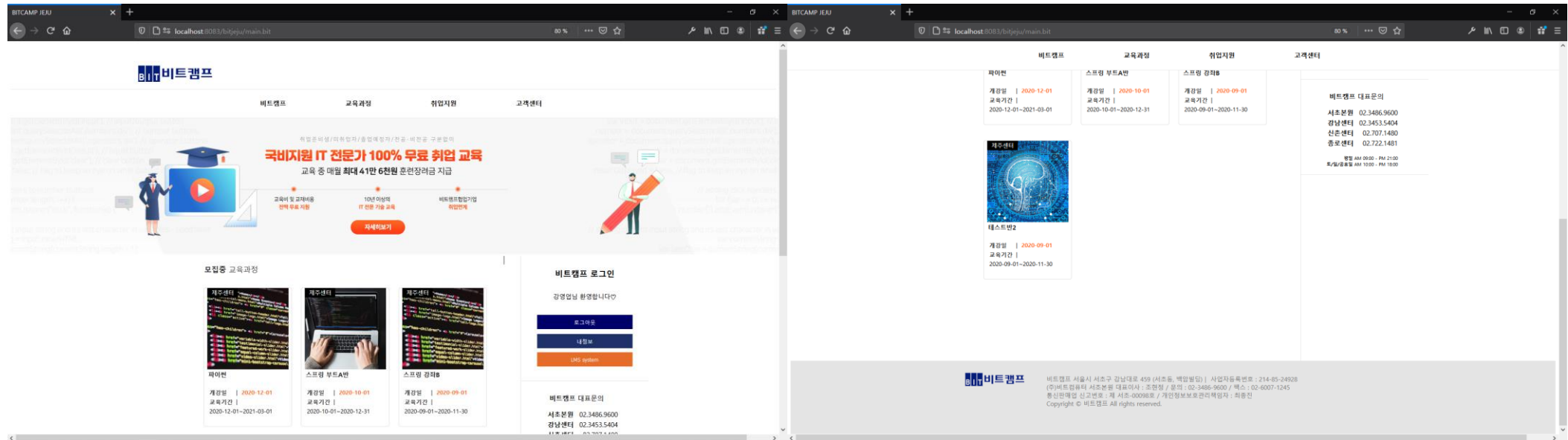
파일 별로 라이프 라인 설정

각 페이지의 JSP, Controller, Dao를 라이프라인으로 설정하여 표시



구현

담당 파트만 게시



- 광고 캐러셀
- 로그인 및 LMS 접속 버튼
- 고정 메뉴 : 비트캠프 교육과정 취업지원 고객센터
- 모집중인 수강정보가 화면 중앙에 위치

- 모든 페이지에서 로그인폼이 보인다.
- Footer에 로그인폼이 포함되어 있어서 모든 페이지에서 로그인 가능하다.
- 로그인은 Ajax처리하고 페이지 이동없이 로그인 처리를 하였다.

```
$('#formlogin').submit(function(){ //로그인버튼 눌렀을 때 이벤트.
    var emailVal = $('#emailid').val(); //id창에 입력된 값
    var regExp = /^[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*.[a-zA-Z]{2,3}$/i; //정규표현식 이메일
    var param = 'emailid='+$('#emailid').val()+'&pw='+$('#pw').val(); //ajax로 보낼 파라미터 id, pw

    if($('#emailid').val()="" || $('#pw').val()==""){ //아이디나 비번 입력안한경우.
        alert("아이디 혹은 비밀번호를 확인해주세요.");
    }else if(emailVal.match(regExp) == null){ //이메일 양식이 맞는지 클라이언트에서 먼저체크.
        alert("아이디는 이메일 형식입니다.");
    }else{
        $.ajax('login.bit',{
            'method':'post',
            'data':param,
            'success':function(data){
                if($(data).find('fail').text()='fail'){
                    console.log('로그인실패')
                    alert('아이디와 비밀번호를 확인해주세요');
                    //id pw를 서버에 보냈으나 회원정보와 일치하지 않음
                    //location.replace('main.bit');
                }else{
                    //location.href='main.bit'; //login성공
                    //비동기통신이므로 강제로 페이지 이동시켜주지않으면 로그인창이 '환영합니다'로 안바뀜
                    location.reload(true);
                }
            } //success
        }); //ajax
    } //else
    return false;
}); //submit
```

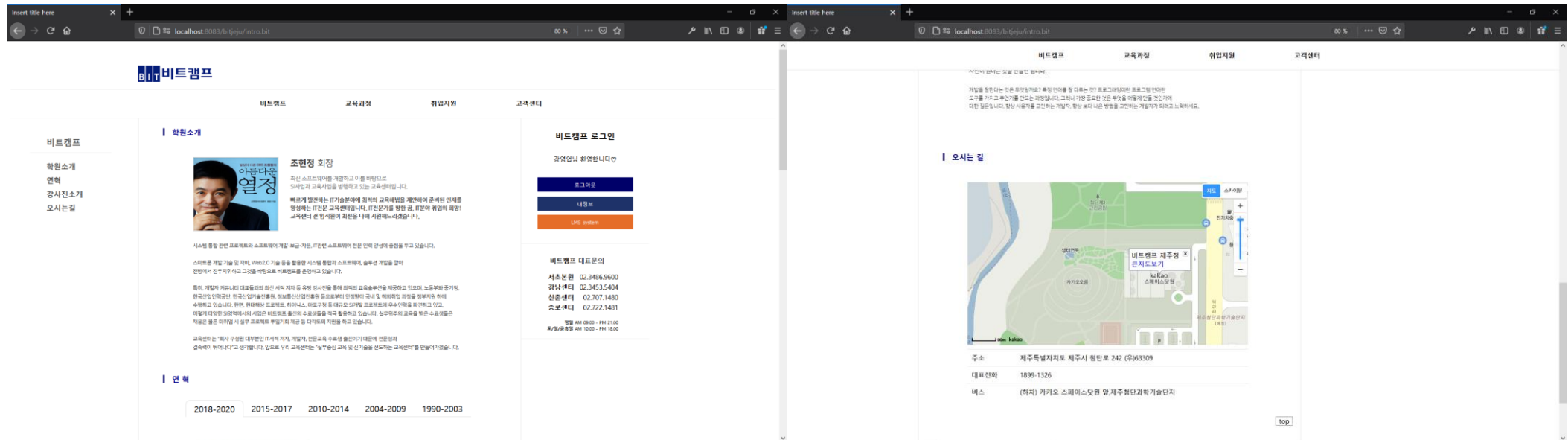
- 모집공고내용을 DB에서 받아와서
jstl로 반복한다.
- Thumbnail클래스의 div에는
Jquery로 클릭 이벤트 부여해서
모집요강 페이지로 이동시킨다.
- 상단 헤더와 푸터, 로그인폼은
템플릿으로 따로 만들어 모든
JSP파일에서 공유한다.

```
&nbsp;   <!-- *****content start***** -->

<div id="curriculum-main">

    <div id="tile-div">
        <div class="title_letter">&nbsp;&nbsp;&nbsp;<strong>모집중</strong> 교육과정</div>
    </div>
    <div id="lecturecontent">
        <!-- *****col-xs-4를 foreach로 db에서 모집공고 받아와서 반복.***** -->

        <c:forEach items="${lecture}" var="bean">
            <c:set value="${bean.recruit_num}" var="idx"></c:set>
            <div class="col-xs-4">
                <div class="thumbnail" onclick="curriDetail(${idx})">
                    <div class="box-tag box-tag-normal">제주센터</div>
                    
                    <!-- 강좌이미지 모집공고이미지 -->
                    <div class="caption">
                        <h3><span class="lecture-name">${bean.recruit_name }</span></h3>
                    </div>
                    <div class="text-left price-bar">
                        <table id="edu-duration">
                            <tr>
                                <th>개강일</th>
                                <td>&#124;&nbsp;&nbsp;&nbsp;<span id="orange-date">${bean.start_day }</span></td>
                            </tr>
                            <tr>
                                <th>교육기간</th>
                                <td>&#124;&nbsp;&nbsp;&nbsp;</td>
                            </tr>
                            <tr>
                                <td colspan="2">${bean.start_day}~${bean.end_day }</td>
                            </tr>
                        </table>
                    </div>
                </div>
            </div>
        </c:forEach>
    </div>
</div>
```



- 스크롤을 내리면 고정메뉴가 상단에 픽스된다.
- 왼쪽 메뉴를 누르면 해당정보로 스크롤이 이동한다.
- 학원소개, 강사진 소개, 연혁, 카카오맵 api

```
function Tabs(){
    $('#Tab>ul>li').each(function(idx, ele){ /*하나 누를때마다 눌린 탭과 나머지탭의 border와 배경색, 이벤트 변경, 이미지 로드*/
        $(this).on('click', function(){
            $(this).css('border','1px solid #dddddd').css('border-top-right-radius','10px')
            .css('border-top-left-radius','10px').css('border-bottom','0px');

            $('#Tab>.tab_list>li').not($(this)) //안눌린 것들은 밑경계선 보이게.
            .css('border','0px solid')
            .css('border-bottom','1px solid #dddddd');

            $(this).css('background-color','white');//배경 흰색

            $('#Tab>.tab_list>li').not($(this)).on('mouseover',function(){ /*마우스 이벤트*/
                $(this).css('background-color','aliceblue')
            });
            $('#Tab>.tab_list>li').not($(this)).on('mouseleave',function(){
                $(this).css('background-color','white')
            });

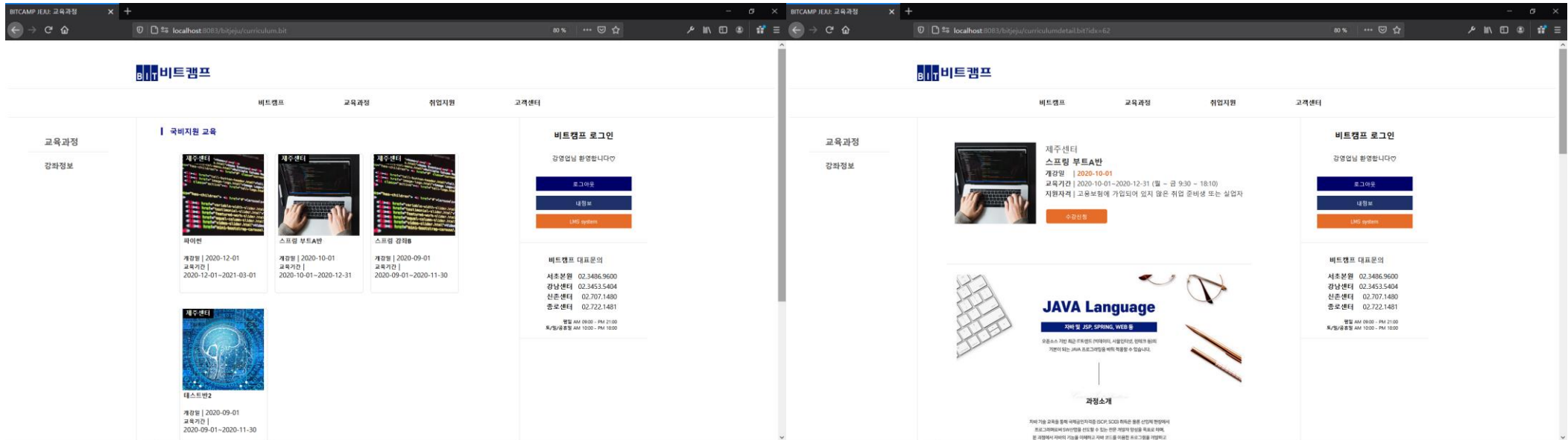
            $('.etcimg').attr('src','img/history'+idx+'.jpg');//탭 년도별로 이미지 부여
        }); //li click
    });
} //Tabs
```

- 연혁과 강사소개는 자바스크립트로 동적으로 구현하였다.
- 원하는 연도와 강사 탭을 누르면 해당하는 내용이 div에 나타난다.

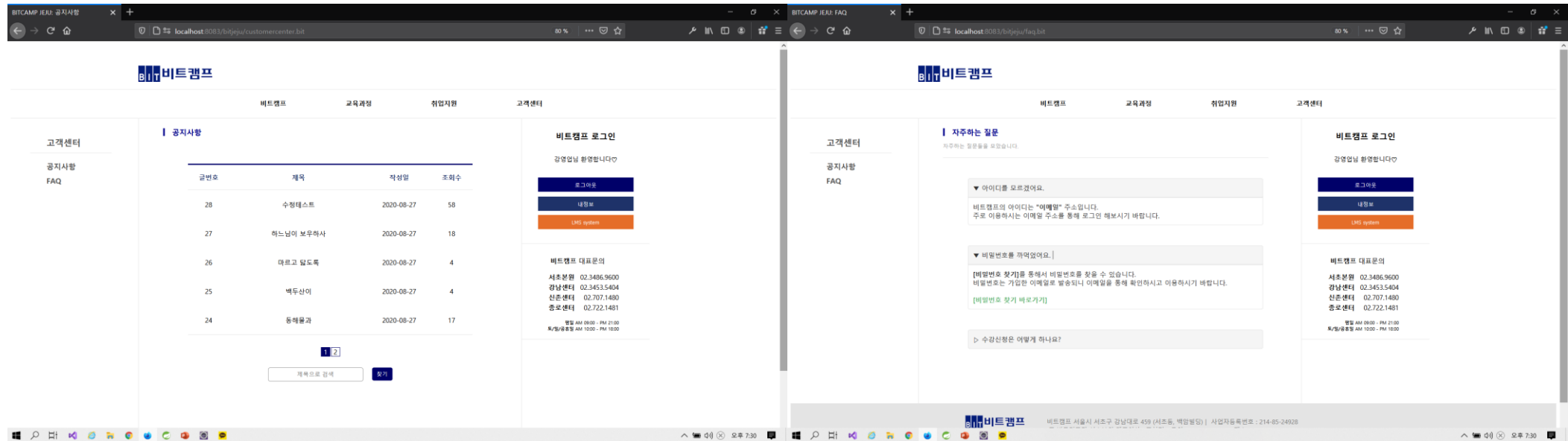
- 연혁과 강사소개는 첫번째 탭을
기본으로 보이게 클릭을 실행시킨다.
- 카카오맵을 실행시킨다.
- 왼쪽에 메뉴를 클릭했을 때 해당하는
요소로 이동하게끔 조건을 만들어
클릭을 실행시킨다.

```
var aTag = "${a}";
$(document).ready(function(){
    console.log(aTag);
    Stu();
    Tabs();
    $('#Tab>.tab_list>li:nth-child(1)').click();
    $('#teacher>.tab_teacher_list>li:nth-child(1)').click();
    kakaoMap();

    if(aTag==1){
        //$('#intro-a-1').get(0).click();
    }else if(aTag==2){
        $('#intro-a-2').get(0).click();
    }else if(aTag==3){
        $('#intro-a-3').get(0).click();
    }else if(aTag==4){
        $('#intro-a-4').get(0).click();
    }
});
```



- 모집중인 강좌 정보가 화면 중앙에 배치
- 강좌 클릭하면 상세보기 페이지로 이동
- 수강신청은 로그인을 해야 하며 해당과목을 수강하고 있지 않는 학생만 가능하다.



- 공지사항 게시판(페이징처리, 게시물 검색)
- Session의 로그인 유저가 행정팀일 때만 글작성 가능
- FAQ는 자주 묻는 질문 세가지를 JQuery로 클릭 했을 때 내용이 보이도록 구성

- 페이지 번호(pageNum), 게시글 수(cardinality), 첫 번호(start)와 마지막 번호(end), 한 페이지에 보여줄 번호 수(lang)
- 반복문으로 페이지 번호 링크를 생성하는 함수를 만들고 게시판 하단 div에 append 시킨다.
- 페이지 로딩시 전체 게시글 수와 한페이지에 보일 게시글 수를 이용해 전체 페이지번호 수를 구한다.
- Start와 end를 정한 뒤 위에서 정의한 pages() 함수를 실행시킨다.

```
//*****↓페이지처리*****  
var pageNum, cardinality, start, end, lang;  
pageNum = "${pageNum}";  
cardinality = "${totalList}";  
function pages() {  
    var pageLink = '';  
    for (var i = start; i < end; i++) { //페이지 링크 번호 5개씩 생성.  
        pageLink += '<span class="page_num"><a href="#">&nbsp; + i  
            + '&nbsp;</a></span>';  
        console.log(i);  
    }  
    console.log(start, 'start');  
    $('<div id="page_num" />').html(pageLink).appendTo('#pagenum');  
    //페이지링크 문자열을 게시판 테이블 뒤에 삽입.  
}  
$(function() {  
    lang = 10; //한 페이지당 페이지 링크번호 수. 10개  
    var left = 0;  
    var post = 5; //한페이지에 보이는 게시글 수  
    if (cardinality % post != 0) { //게시글 나누기 5의 나머지가 있는 경우 1페이지 추가  
        left = 1;  
    }  
    /*  
    77개 /10 =7  
    나머지 +1페이지  
    */  
    var totalPage = (cardinality - cardinality % post) / post + left;  
    console.log('totalpage', totalPage);  
    start = 1 + parseInt((pageNum - 1) / lang) * lang;  
    end = start + lang; //페이지 번호링크 끝  
    if (end > totalPage) {  
        end = totalPage + 1;  
    }  
    pages();  
});
```

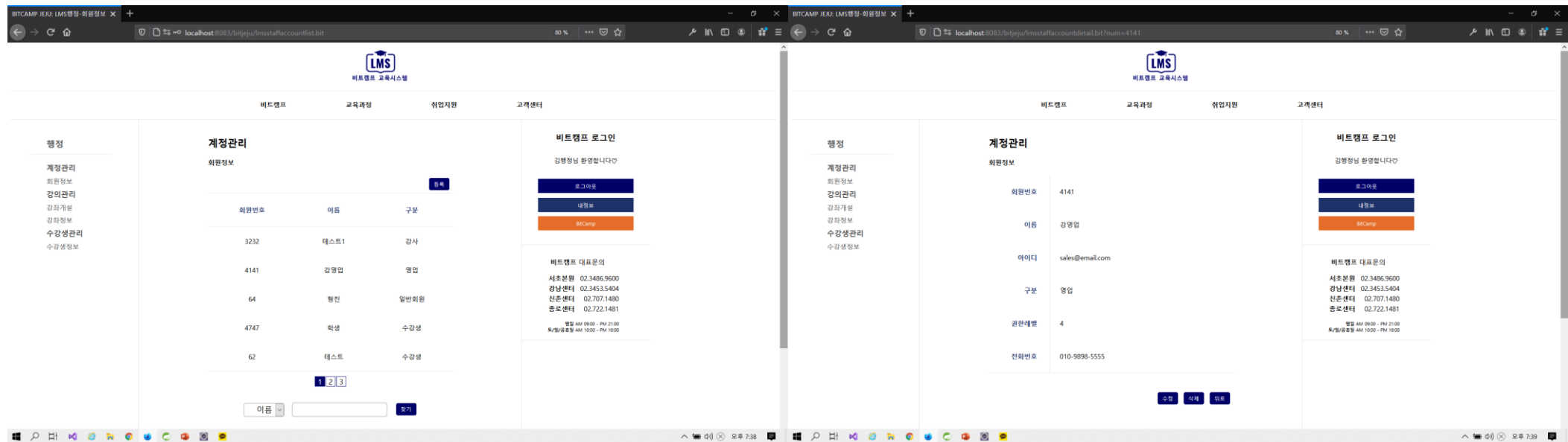
```
$('.faq-question').each(function(){
    $(this).on('click',function(){
        if($(this).find('.triangle').text() == '▶'){

            $(this).find('.triangle').text('▼');
            $(this).css('border-bottom-left-radius','0px')
                .css('border-bottom-right-radius','0px')
                .css('box-shadow','0px');
            $(this).next().slideDown(500);

        }else if($(this).find('.triangle').text() == '▼'){

            $(this).find('.triangle').text('▶');
            $(this).next().slideUp(500,function(){
                $(this).prev().css('border-bottom-left-radius','5px')
                    .css('border-bottom-right-radius','5px')
                    .css('box-shadow','0px');
            }); //slideup
        } //else
    }) //click
}); //each
```

- FAQ 글을 눌렀을 때 삼각형 모양의 색깔 유무로 내용이 show()인지 hide()지 구별한다.
- 처음 눌렀을 때는 내용을 slideDown()한다.
- 내용이 나와있을 때 클릭하면 slideUp()한다.



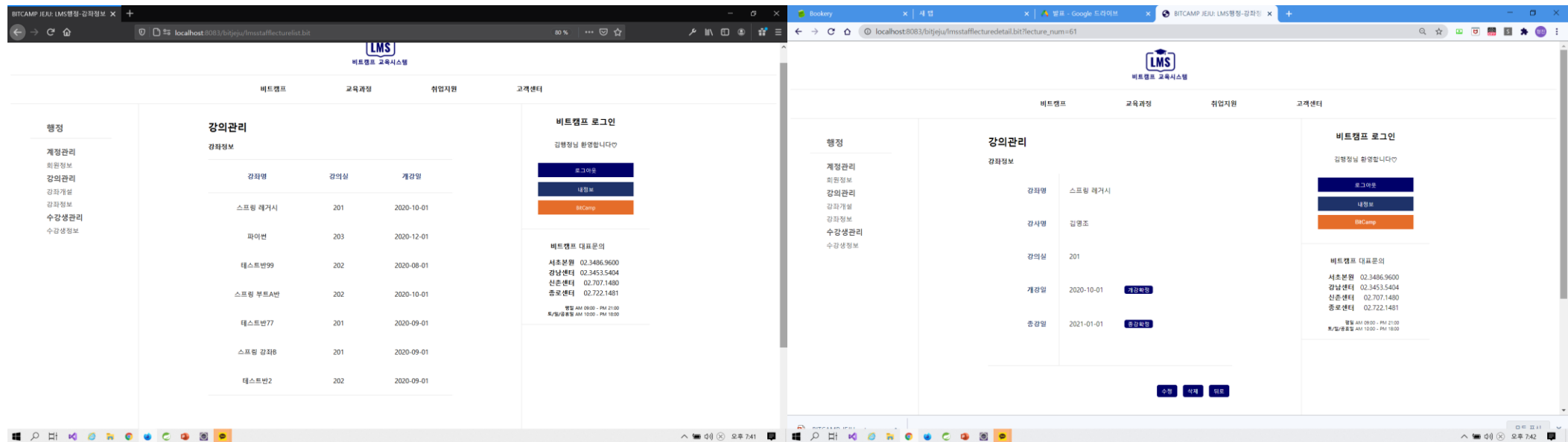
- 행정팀 LMS 계정관리 – 모든 회원정보 열람 가능, 정보 수정 및 삭제, 등록이 가능하다.
- 페이징 처리
- 회원 이름, 분류로 검색 가능

```
public ArrayList<MemberDto> selectAll(int pageNum, String key, String word) throws SQLException {
    ArrayList<MemberDto> list = new ArrayList<MemberDto>();
    int startNum = 1 + (pageNum - 1) * 5;
    int endNum = 5 + (pageNum - 1) * 5; // 페이지당 게시글 수
    String sql = "select * from (select num,id_email, name, dept, lvl, phone, lecture, rownum as rwn from ";
    sql += "(select * from member natural join bitjejuddept where " + key + " like '%" + word + "%')) ";
    sql += "where rwn between " + startNum + " and " + endNum;
    System.out.println(sql);

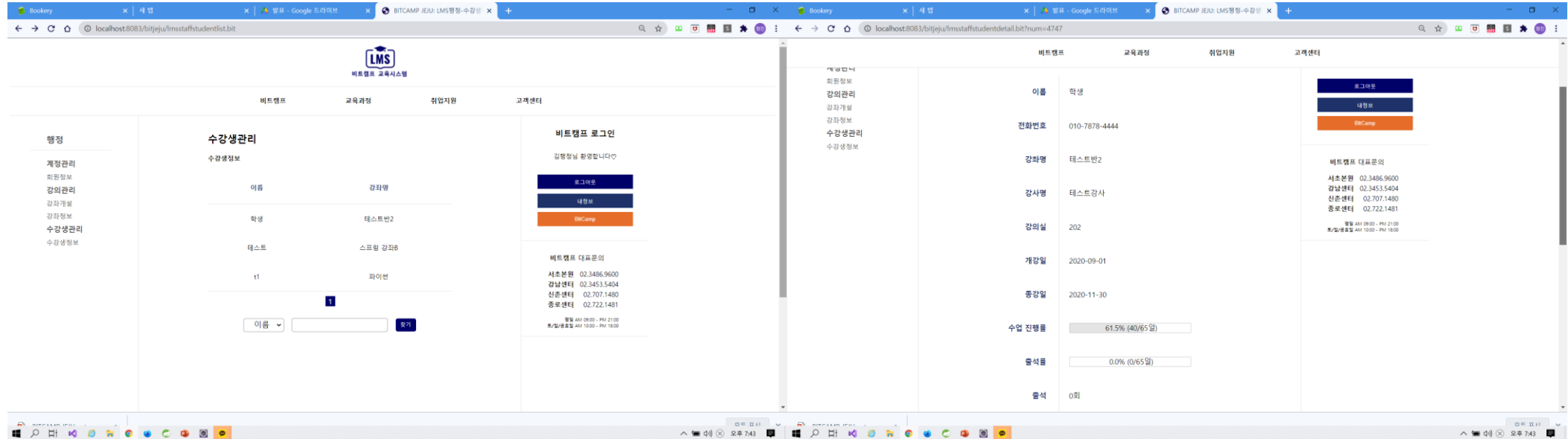
    MemberDto bean = null;
    pstmt = conn.prepareStatement(sql);
    rs = pstmt.executeQuery();

    while (rs.next()) {
        bean = new MemberDto();
        bean.setNum(rs.getInt("num"));
        bean.setId_email(rs.getString("id_email"));
        bean.setName(rs.getString("name"));
        bean.setDept(rs.getString("dept"));
        bean.setLvl(rs.getInt("lvl"));
        bean.setPhone(rs.getString("phone"));
        bean.setLecture(rs.getString("lecture"));
        list.add(bean);
    } // while
    if (pstmt != null)
        pstmt.close();
    if (conn != null)
        conn.close();
    return list;
} // selectAll
```

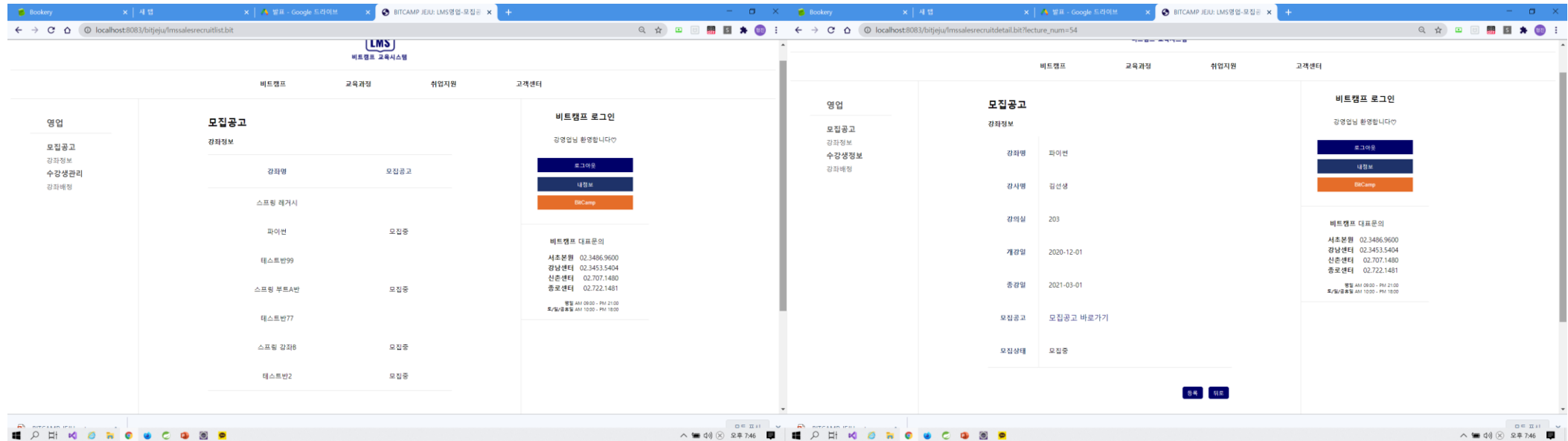
- 항상 페이지 번호와 검색옵션(key 제목 등), 검색단어를 파라미터로 받아 SQL문에 적용한다.
- 회원정보Dto 객체에 ResultSet을 담아 ArrayList로 반환한다.
- 뷰에서 list를 받아 게시판에 회원정보를 출력한다.



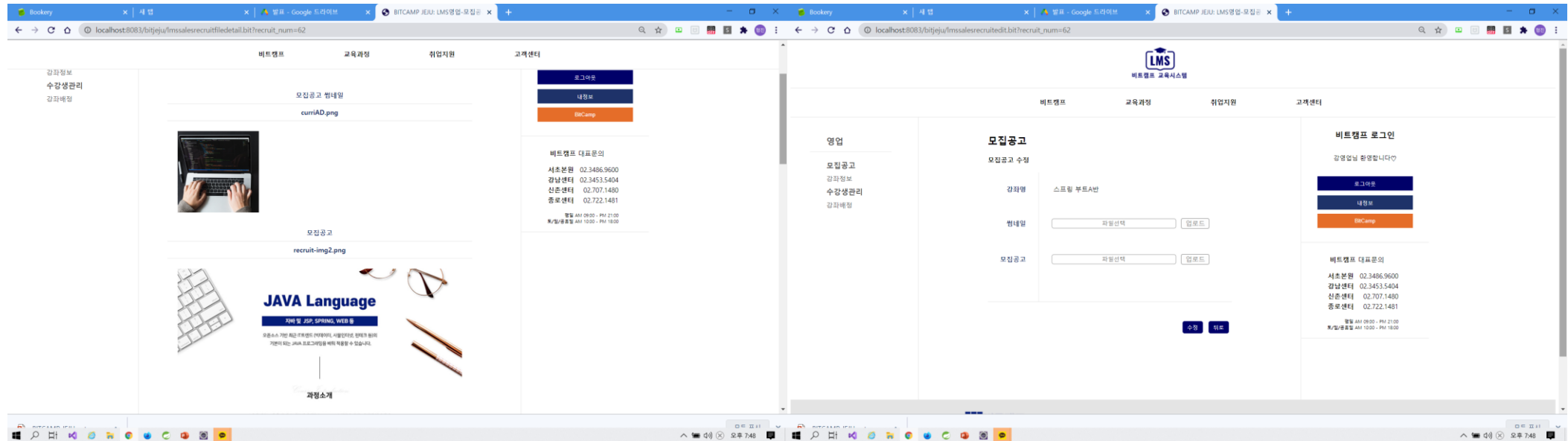
- 강좌 개설 가능
- 강좌 개강, 종강 확정 가능
- 강좌 정보 수정, 삭제 가능



- 수강생 전체 조회 가능
- 수강생 정보 열람가능
- 출석률이 80% 미만인 학생은 삭제 가능



- 행정팀이 개설한 강좌정보 열람 가능
- 개설된 강좌에 모집공고 업로드 가능



- 모집공고 썸네일과 모집공고 이미지 업로드
- 수정 가능
- 수정하면 기존에 서버에 저장한 이미지 파일은 삭제하고 새로 업로드 하는 방식

- 뷰에서 업로드한 파일을 컨트롤러에서 post로 받는다.
- Cos.jar 라이브러리의 multipartRequest 클래스를 이용해 파일을 업로드한다.
- 썸네일과 모집공고 이미지 두개 를 받아서 업로드 한다.
- 사진이 업로드되면 해당 경로(url)은 recruit테이블에 저장한다.
- 업로드 완료 후 모집공고 리스트 페이지로 이동시킨다.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("utf-8");
    String path = request.getRealPath("../recruit");
    File file = new File(path);

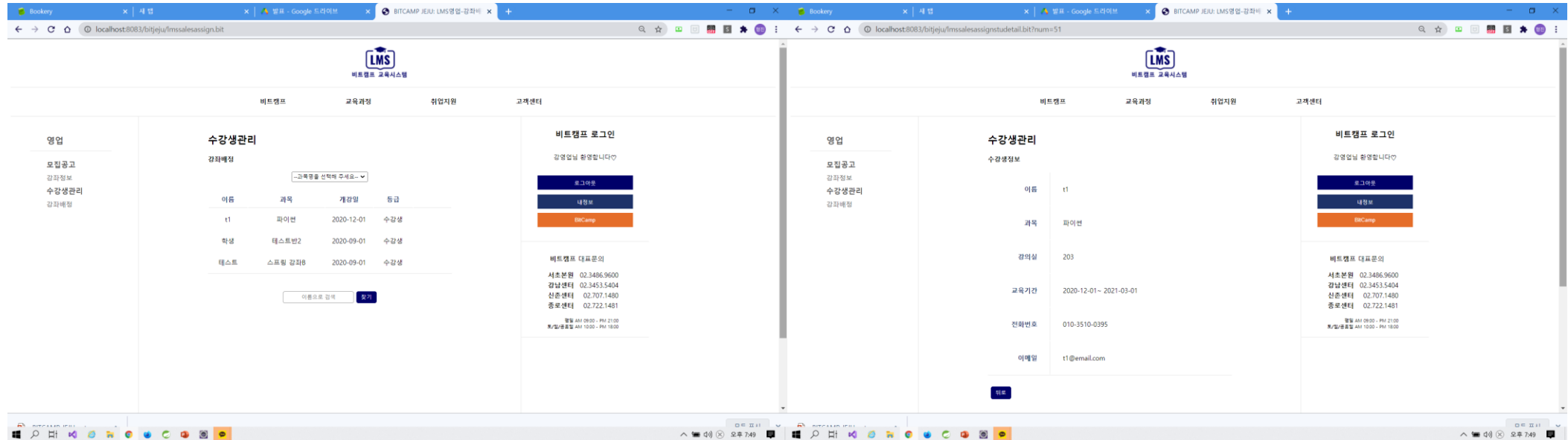
    if(!file.exists()) {
        file.mkdirs();
    }
    int fsize = 10*1024*1024;
    //*****파일 업로드*****
    DefaultFileRenamePolicy frp = new DefaultFileRenamePolicy();//파일명이 같으면 rename해준다.

    MultipartRequest mpReq = new MultipartRequest(request, path, fsize, "utf-8", frp);// 업로드하는 파일 받는 객체
    //서블릿 요청, 경로, 파일크기, 인코딩 설정, rename객체

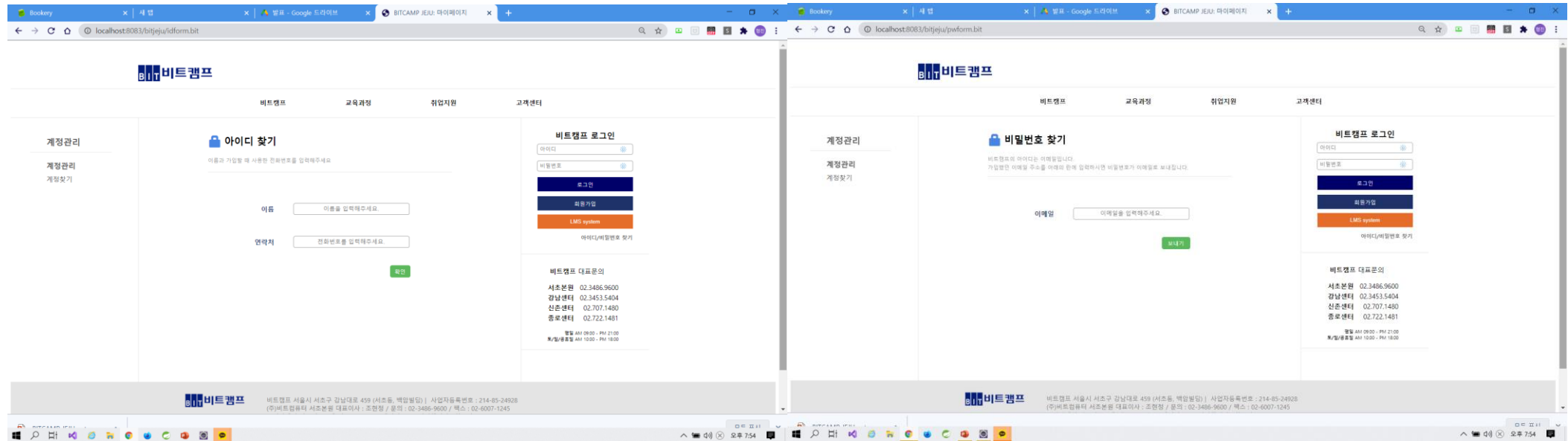
    Enumeration en = mpReq.getParameterNames();
    String lecture_name = null;
    while(en.hasMoreElements()) {
        String lecture = (String)en.nextElement();//강좌이름
        lecture_name = mpReq.getParameter(lecture);
    }
    String thumbnail_name = mpReq.getFileName("thumbnail");//저장된파일명
    String file_name = mpReq.getFileName("recruitfile");//저장된파일명

    System.out.println(lecture_name); //모집공고 강좌
    System.out.println(thumbnail_name); //목록에서 보이는 썸네일 이미지
    System.out.println(file_name);//모집공고 이미지파일

    RecruitDao dao = new RecruitDao();
    try {
        //*****DB에 모집공고 정보 저장*****
        dao.recruitUpload(lecture_name, file_name, thumbnail_name);//강좌명으로 강좌번호찾기, 파일명,썸네일명 저장
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    request.getRequestDispatcher("/lmsalesrecruitlist.bit").forward(request, response);//모집공고강좌정보메뉴로 이동.
}
```



- 홈페이지에서 수강신청한 학생들 목록을 조회
- 수강 확정시켜서 회원등급을 일반에서 수강생으로 전환
- 학생 개인정보 조회
- 학생 검색 가능



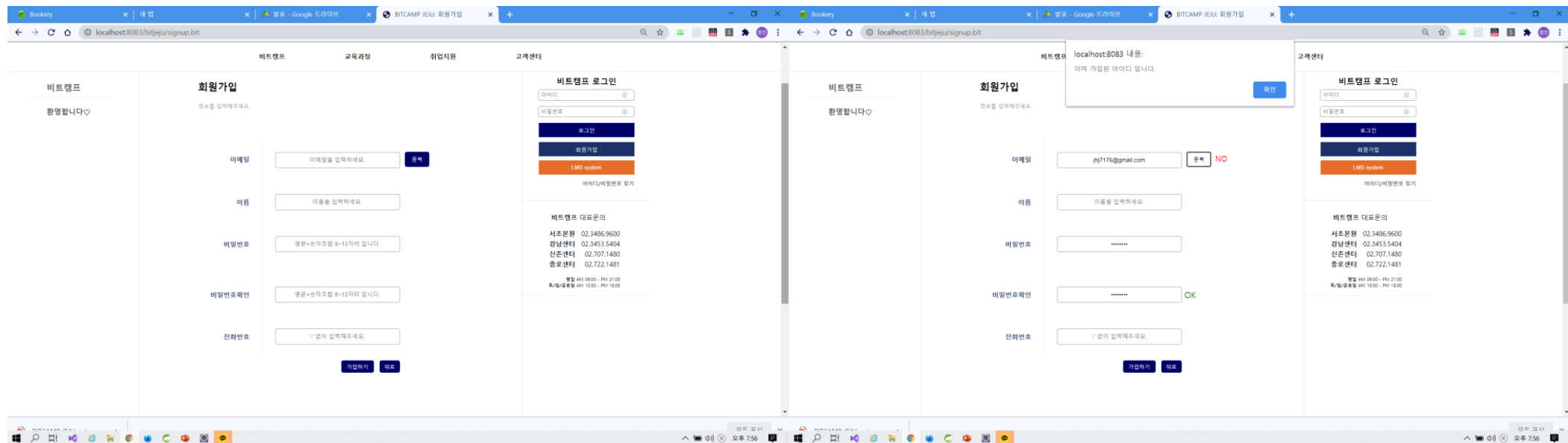
- 아이디찾기는 이름과 연락처로 조회한 뒤 페이지에 표시
- 비밀번호 찾기는 가입한 이메일로 DB에서 조회 후 해당 비밀번호를 발송

- 패스워드 찾기를 원하는 유저의 Email을 파라미터로 받는다.
- 회원 테이블에서 해당하는 id의 email을 검색해서 반환되는 값이 있다면 회원의 비밀번호를 user_password에 저장한다.
- Javax-mail.jar 라이브러리를 이용했다.
- DB에서 찾은 비밀번호를 메일 내용에 넣어 메일 발송한다.

```
public void sendEmail(String id_email) {
    String user_password = null;
    String sql = "select password from member where id_email=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id_email);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            user_password = rs.getString("password");
        }
        prop = new Properties();
        prop.put("mail.smtp.host", "smtp.gmail.com");
        prop.put("mail.smtp.port", 465);
        prop.put("mail.smtp.auth", "true");
        prop.put("mail.smtp.ssl.enable", "true");
        prop.put("mail.smtp.ssl.trust", "smtp.gmail.com");

        session = Session.getDefaultInstance(prop, new javax.mail.Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(user, password);
            }
        }); // session

        MimeMessage message = new MimeMessage(session);
        message.setFrom(new InternetAddress(user));
        // 수신자메일주소
        message.addRecipient(Message.RecipientType.TO, new InternetAddress(id_email));
        // Subject
        message.setSubject("bitjeju 비밀번호 입니다."); // 메일 제목을 입력
        // Text
        message.setText(user_password); // 메일 내용을 입력
        // send the message
        Transport.send(message); // 전송
        System.out.println("message sent successfully ... ");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



- 이메일을 아이디로 사용하며 ajax로 중복확인
- 비밀번호 영문 + 숫자 조합에 8~12자 조건(JavaScript로 검증)

- 회원가입 폼 검증
- Email은 정규표현식을 사용해 검사한다.
- Input 요소의 값이 비어 있는 경우
- Id가 email이 아닌 경우
- 비밀번호가 영문+숫자 조합이 아닌 경우
- 비밀번호 길이가 8~12자를 벗어난 경우
- 비밀번호와 두번째 비밀번호 확인 input 요소의 값이 일치하지 않는 경우
- 이메일과 비밀번호가 일치 하지 않는 경우는 span태그를 이용해 경고 메시지 출력한다.
- 모든 검사를 통과하면 ajax로 서버에 가입정보를 전송한 뒤 페이지를 이동한다.

```

(function(){
    $('#signupbtn').on('click',function(){ //등록버튼 눌렀을 때 이벤트.
        console.log('서브밋');
        var emailVal = $('#signupemailid').val(); //id창에 입력된 값
        var regExp = /^[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*[a-zA-Z]{2,3}$/i;
        var memberInfo = 'emailid='+$('#signupemailid').val()+'&pw='+$('#signuppw').val();
        memberInfo += '&phone='+$('#phone').val()+'&name='+$('#name').val();
        console.log(memberInfo);
        /*
            비밀번호는 영문 + 숫자 조합, 8~12자리
        */
        var passwordVal=$('#signuppw').val(); //비밀번호
        var signuppw2 = $('#signuppw2').val(); //비밀번호2

        if($('#signupemailid').val()="" || $('#signuppw').val()="" || $('#name').val()="" || $('#phone').val()==""){
            alert("모든 정보를 입력해야 합니다.");
        }else if(emailVal.match(regExp) == null){
            alert("아이디는 이메일 형식 입니다.");
        }else if(pwCheck(passwordVal)){
            //비밀번호 검증 하기.   영문이나 숫자가 포함되어야함., 포함되어있으면 false 영어나숫자만 true
            alert('비밀번호는 영문 + 숫자 조합이어야 합니다. ');
        }else if(passwordVal.length<8 || passwordVal.length>12){
            //비밀번호 길이는 8자~12자까지.
            alert('비밀번호는 8 ~ 12자리 입니다. ');
        }else if($('#overlapCK').text()=='NO'){
            alert('아이디를 확인해주세요');
        }else if(passwordVal!=signuppw2){
            alert('비밀번호가 일치하지 않습니다. ');
        }else{
            $.ajax('signup.bit',{
                'method':'post',
                'data':memberInfo,
                'success':function(){
                    location.href='signedup.bit';
                } //success
            }); //ajax
        }
    } //else
    return false;
}); //submit

```

- Email를 입력한 뒤 중복버튼을 누른다.
- 입력한 email이 회원테이블에 있는지 검사한다.
- Ajax로 페이지 이동없이 email를 서버로 보낸다.
- 중복검사후 결과를 xml형식으로 반환한다.
- 결과가 ok이면 ok라는 메시지가 span요소로 중복버튼 옆에 나타난다.
- 반대인 경우는 no 메시지가 span요소로 중복버튼 옆에 나타난다.

```
$('#overlap').on('click',function(){ // 아이디 중복검사 버튼
    var chk_id=$('#signuemailid').val()

    if(chk_id==''){
        alert('아이디를 입력해주세요.');        return;
    }else{
        $.ajax('overlapcheck.bit',{ //idoverlapcheck.java
            'method':'post',
            'data':'id_email='+chk_id,
            'success':function(data){
                overlapCheck=$(data).find('overlap').text();
                if(overlapCheck=='ok'){
                    console.log('성공',$(data).find('overlap').text());
                    $('#overlapCK').text('OK').css('color','green')
                                        .css('font-weight','500')
                                        .css('font-size','120%');
                }else if(overlapCheck=='no'){
                    $('#overlapCK').text('NO').css('color','red')
                                        .css('font-weight','500')
                                        .css('font-size','120%');
                    alert('이미 가입된 아이디 입니다.');                }
            }
        })//ajax
    }
})
```

- 비밀번호 input요소에는 keyup 이벤트를 부여한다.
- 영문+숫자 조합이 아니거나 길이가 8자~12자 를 벗어나면 경고메세지가 span요소로 하단에 나타난다.
- 두번째 비밀번호 확인에서도 keyup 이벤트를 이용해 비밀번호를 칠 때마다 위의 값과 일치하는지 검사한다.
- 이상이 없으면 ok 메시지를 span요소로 우측에 나타낸다.

```
$('#signuppw').on('keyup',function(){
    signuppw1 = $('#signuppw').val(); //비밀번호1
    if(pwCheck(signuppw1)){
        $('#passwordCheck').text('비밀번호는 영문+숫자 조합이어야 합니다').css('font-size','80%')
        .css('color','red').css('vertical-align','middle').css('padding-left','10px');
    }else if(signuppw1.length<8||signuppw1.length>12){
        $('#passwordCheck').text('비밀번호는 8~12 자리입니다').css('font-size','80%')
        .css('color','red').css('vertical-align','middle').css('padding-left','10px');
    }else{
        $('#passwordCheck').text(' ');
    }
});

$('#signuppw2').on('keyup',function(){
    signuppw1 = $('#signuppw').val(); //비밀번호1
    signuppw2 = $('#signuppw2').val(); //비밀번호2
    if(signuppw2==''){
        $('#changeptxt').text('');
    }else if((signuppw1==signuppw2) &&(signuppw2!=null)){
        $('#changeptxt').text('OK').css('font-size','130%')
        .css('color','green').css('vertical-align','middle');
    }else{
        $('#changeptxt').text('NO').css('font-size','130%')
        .css('color','red').css('vertical-align','middle');
    }
});
```




감사합니다.