

MVC PROJECT

A Team

프로젝트 결과 보고서

Ezen_Developer

문다운 Team Project Manager



01 팀 구성원 및 역할

Ezen_Developer



문다운

PM | BACKEND

- 역할분배
- 문서초안
- 일정관리



김민정

PL | FRONTEND

- 기획
- 요구분석
- 화면편집



박건도

PA | FRONTEND

- 문서관리
- 화면편집
- 디자인



윤대희

PA | BACKEND

- DB 설계
- 개인과제
- 테스트

{ 공통역할 화면 설계 / 프로그램 설계 / 프로젝트 개발 }

Project

MVC 패턴을 활용한 학사 출결 관리 시스템

Project Name

Ezen_Univ

MVC 패턴을 활용한 학사 출결 관리 시스템



PROBLEM

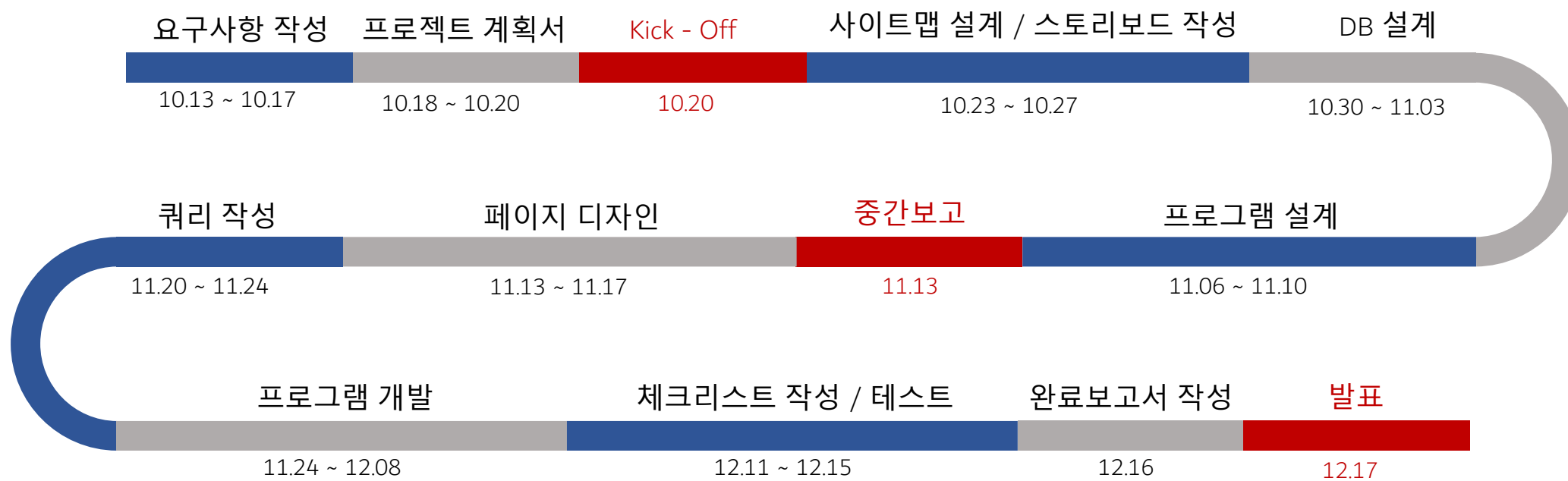
- ✖ 종이 출석부
- ✖ 복잡한 구조

OBJECTIVE

- ✓ 출결 관리 효율 향상
- ✓ 쉽고 빠른 사용

03 프로젝트 일정

Ezen_Developer



개발도구

프로그래밍



DB 개발 및 설계



그래픽 편집



관리도구

서버



형상관리



브라우저



프로그래밍 언어

프로그램



DB



웹 프로그램



디자인



05 요구사항

Ezen_Developer

NO	기능적 요구 사항	학생	교수
1.회원 관리			
1.1	로그인정보 초기설정	로그인정보 초기설정	로그인정보 초기설정
1.2	개인 회원 정보 조회	개인 회원 정보 조회	개인 회원 정보 조회
1.3	회원정보 수정	회원정보 수정	회원정보 수정
1.4	로그인	로그인	로그인
1.5	로그아웃	로그아웃	로그아웃
1.6	아이디 찾기	아이디 찾기	아이디 찾기
1.7	비밀번호 찾기	비밀번호 찾기	비밀번호 찾기
2. 강의 관리			
2.1	강의목록	수강중 강의	강의를 강의
2.2	휴보강 관리	휴보강 관리	휴보강 관리
2.3	시간표	수강중 강의 시간표	강의를 강의 시간표
3.공지 사항			
3.1	공지사항 입력	공지사항 입력	공지사항 입력
3.2	공지사항 목록	공지사항 목록	공지사항 목록
3.3	공지사항 수정	공지사항 수정	공지사항 수정
3.4	공지사항 삭제	공지사항 삭제	공지사항 삭제
3.5	공지사항 상세보기	공지사항 상세보기	공지사항 상세보기
4.내비게이션			
4.1	마이페이지	마이페이지	마이페이지
4.2	출석관리	출석관리	출석관리
4.3	공지사항	공지사항	공지사항
4.4	강의관리	강의관리	강의관리
5.출석관리			
5.1	출석현황조회	출석현황조회	출석현황조회
5.2	출석관리	출석관리	출석관리
5.3	출석 미달 관리	출석 미달 강의 조회	출석 미달 조회
5.4	공결처리관리	공결처리관리	공결처리관리
6. 마이페이지			
6.1	개인정보	개인정보	개인정보
6.2	개인정보 수정	개인정보 수정	개인정보 수정
6.3	강의목록	강의목록	강의목록
7. 메인화면			
7.1	내비게이션	내비게이션	내비게이션
7.2	공지사항	공지사항	공지사항
7.3	출석관리	출석관리	출석하기
7.4	시간표	시간표	시간표



NO	기능적 요구 사항	학생	교수	관리자
1.회원 관리				
1.1	회원가입신청	회원가입	회원가입	
1.2	회원가입 승인			회원가입 승인
1.3	개인 회원 정보 조회	개인 회원 정보 조회	개인 회원 정보 조회	
1.4	회원정보 수정	회원정보 수정	회원정보 수정	
1.5	비밀번호 수정	비밀번호 수정	비밀번호 수정	
1.6	로그인	로그인	로그인	로그인
1.7	로그아웃	로그아웃	로그아웃	로그아웃
1.8	아이디 찾기	아이디 찾기	아이디 찾기	
1.9	비밀번호 찾기	비밀번호 찾기	비밀번호 찾기	
2. 강의 관리				
2.1	강의목록	강의목록	강의목록	
2.2	강의등록			강의등록
2.3	학생 수강 리스트 등록			학생 수강 리스트 등록
2.4	강의시간 설정			강의시간 설정
2.5	개강날짜 등록			개강날짜 등록
2.6	시간표 확인	시간표 확인	시간표 확인	
3. 출석관리				
3.1	출석처리		출석처리	
3.2	출석조회	출석조회	출석조회	
3.3	결석률 조회	결석률 조회		
3.4	출석률 조회		출석률 조회	
3.5	출석미달 조회	출석미달 조회	출석미달 조회	
3.6	강의별 출석상태 횟수 출력	강의별 출석상태 횟수 출력	강의별 출석상태 횟수 출력	
3.7	교과목명 선택후 검색		교과목명 선택후 검색	
4. 휴보강 공지				
4.1	휴보강공지 등록		휴보강공지 등록	
4.2	휴보강공지 목록	휴보강공지 목록	휴보강공지 목록	
4.3	휴보강공지 상세조회	휴보강공지 상세조회	휴보강공지 상세조회	
4.4	휴보강공지 수정		휴보강공지 수정	
4.5	휴보강공지 삭제		휴보강공지 삭제	
5. 메인화면 출력				
5.1	휴보강공지 목록	휴보강공지 목록	휴보강공지 목록	
5.2	매인 배너 출력	매인 배너 출력		매인 배너 출력
5.3	개인 정보 출력	개인 정보 출력		개인 정보 출력
5.4	시간표 출력	시간표 출력		시간표 출력

휴보강 기능 삭제
공결처리 기능 삭제

회원가입 기능 추가
관리자 추가

NO	기능적 요구 사항	학생	교수	관리자
1.회원 관리				
1.1	회원이입신청	회원이입	회원이입	
1.2	회원이입 승인			회원이입 승인
1.3	개인 회원 정보 조회	개인 회원 정보 조회	개인 회원 정보 조회	
1.4	회원정보 수정	회원정보 수정	회원정보 수정	
1.5	비밀번호 수정	비밀번호 수정	비밀번호 수정	
1.6	로그인	로그인	로그인	로그인
1.7	로그아웃	로그아웃	로그아웃	로그아웃
1.8	아이디 찾기	아이디 찾기	아이디 찾기	
1.9	비밀번호 찾기	비밀번호 찾기	비밀번호 찾기	
2. 강의 관리				
2.1	강의목록	강의목록	강의목록	
2.2	강의등록			강의등록
2.3	학생 수강 리스트 등록			학생 수강 리스트 등록
2.4	강의시간 설정			강의시간 설정
2.5	개강날짜 등록			개강날짜 등록
2.6	시간표 확인	시간표 확인	시간표 확인	
3. 출석관리				
3.1	출석처리		출석처리	
3.2	출석조회	출석조회	출석조회	
3.3	결석률 조회	결석률 조회		
3.4	출석률 조회		출석률 조회	
3.5	출석미달 조회	출석미달 조회	출석미달 조회	
3.6	강의별 출석상태 횟수 출력	강의별 출석상태 횟수 출력	강의별 출석상태 횟수 출력	
3.7	교과목명 선택후 검색		교과목명 선택후 검색	
4. 휴보강 공지				
4.1	휴보강공지 등록		휴보강공지 등록	
4.2	휴보강공지 목록	휴보강공지 목록	휴보강공지 목록	
4.2	휴보강공지 상세조회	휴보강공지 상세조회	휴보강공지 상세조회	
4.3	휴보강공지 수정		휴보강공지 수정	
4.4	휴보강공지 삭제		휴보강공지 삭제	
5. 메인화면 출력				
5.1	휴보강공지 목록	휴보강공지 목록	휴보강공지 목록	
5.2	메인 배너 출력	메인 배너 출력	메인 배너 출력	
5.3	개인 정보 출력	개인 정보 출력	개인 정보 출력	
5.4	시간표 출력	시간표 출력	시간표 출력	

주요 기능

관리자

- 회원가입 승인
- 강의 등록
- 학생 수강 리스트 등록

교수

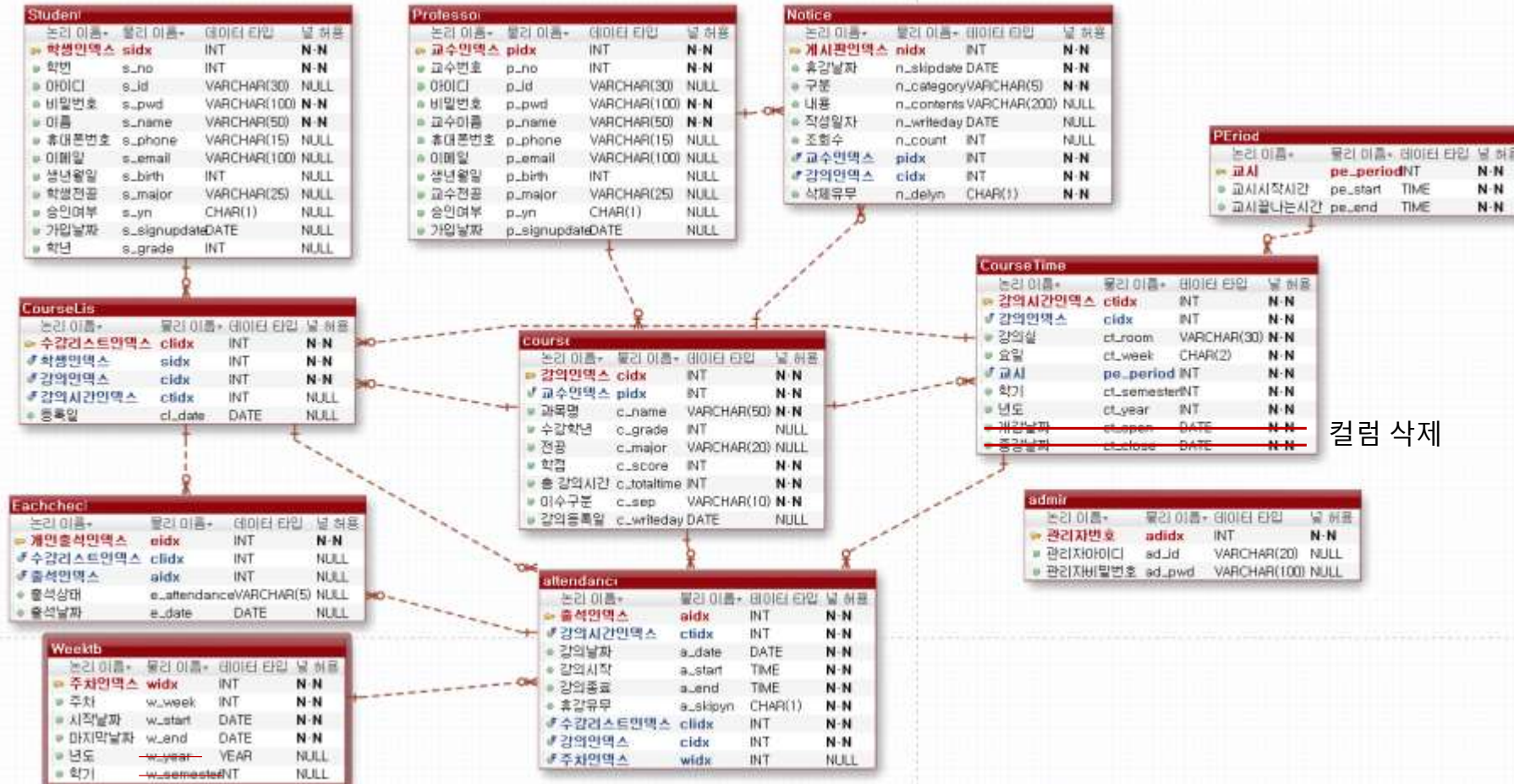
- 출결 조회
- 출석 관리
- 휴보강 공지

학생

- 출석조회

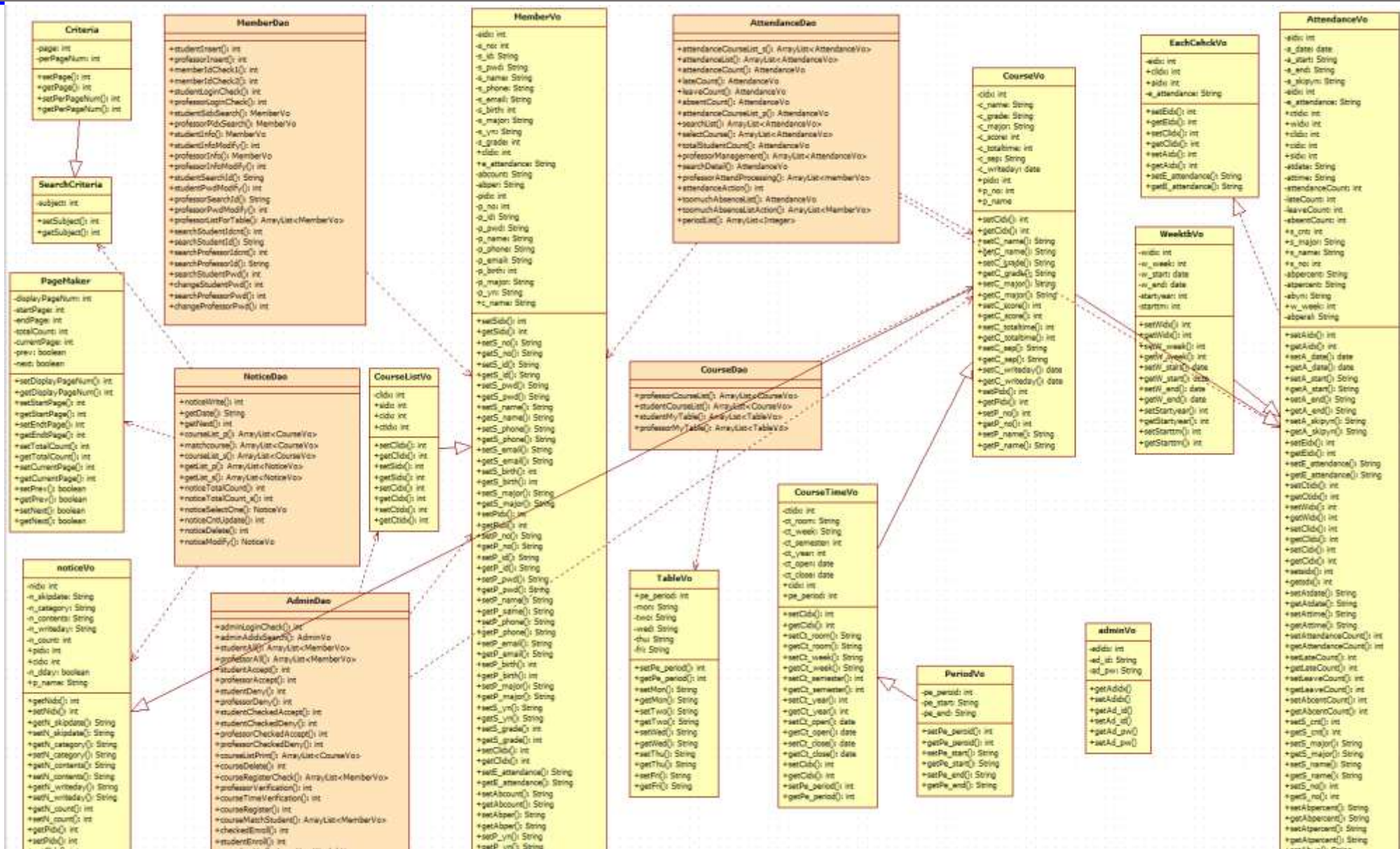
06 DB 설계 (ERD)

Ezen_Developer



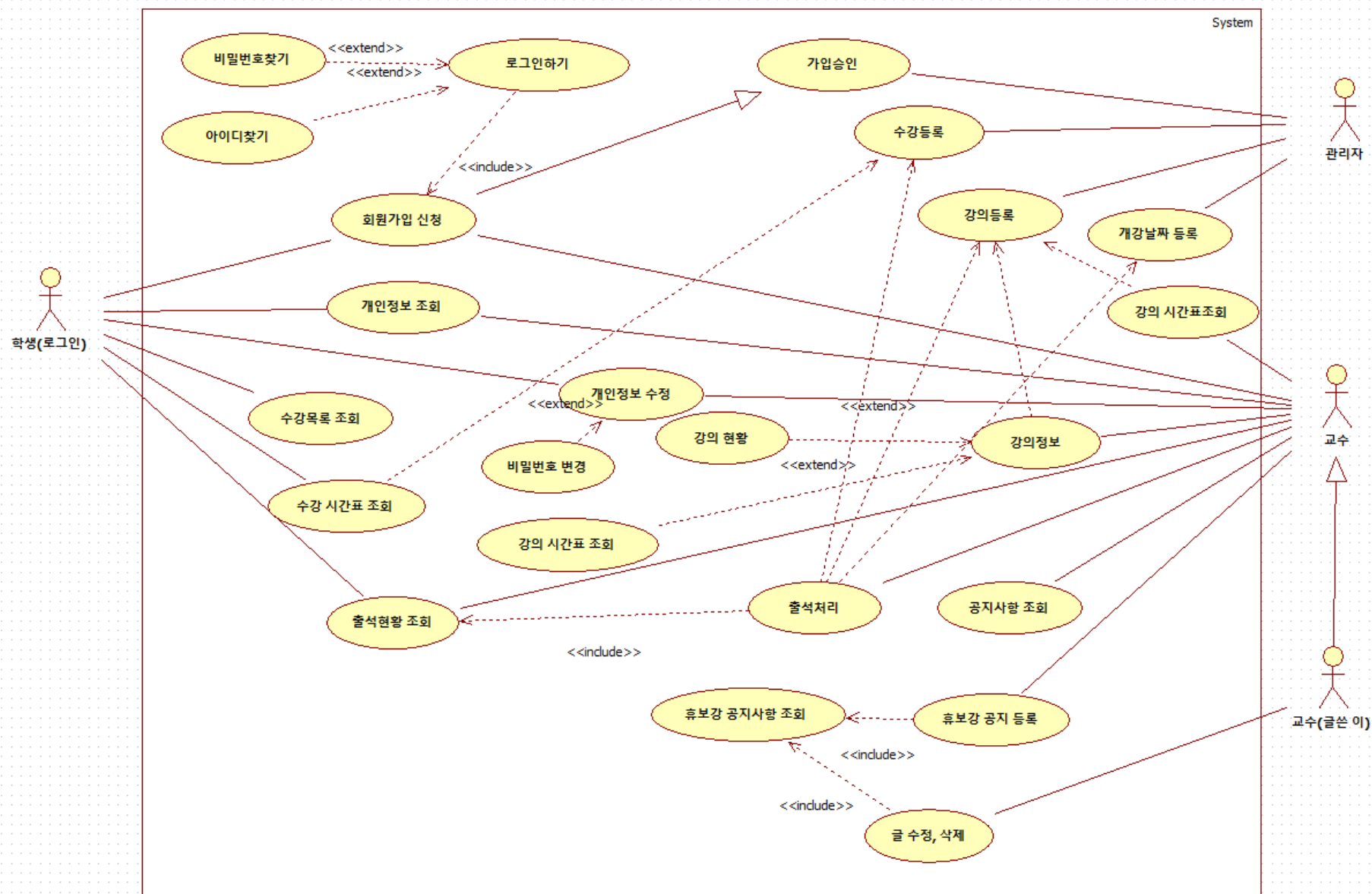
컬럼 삭제

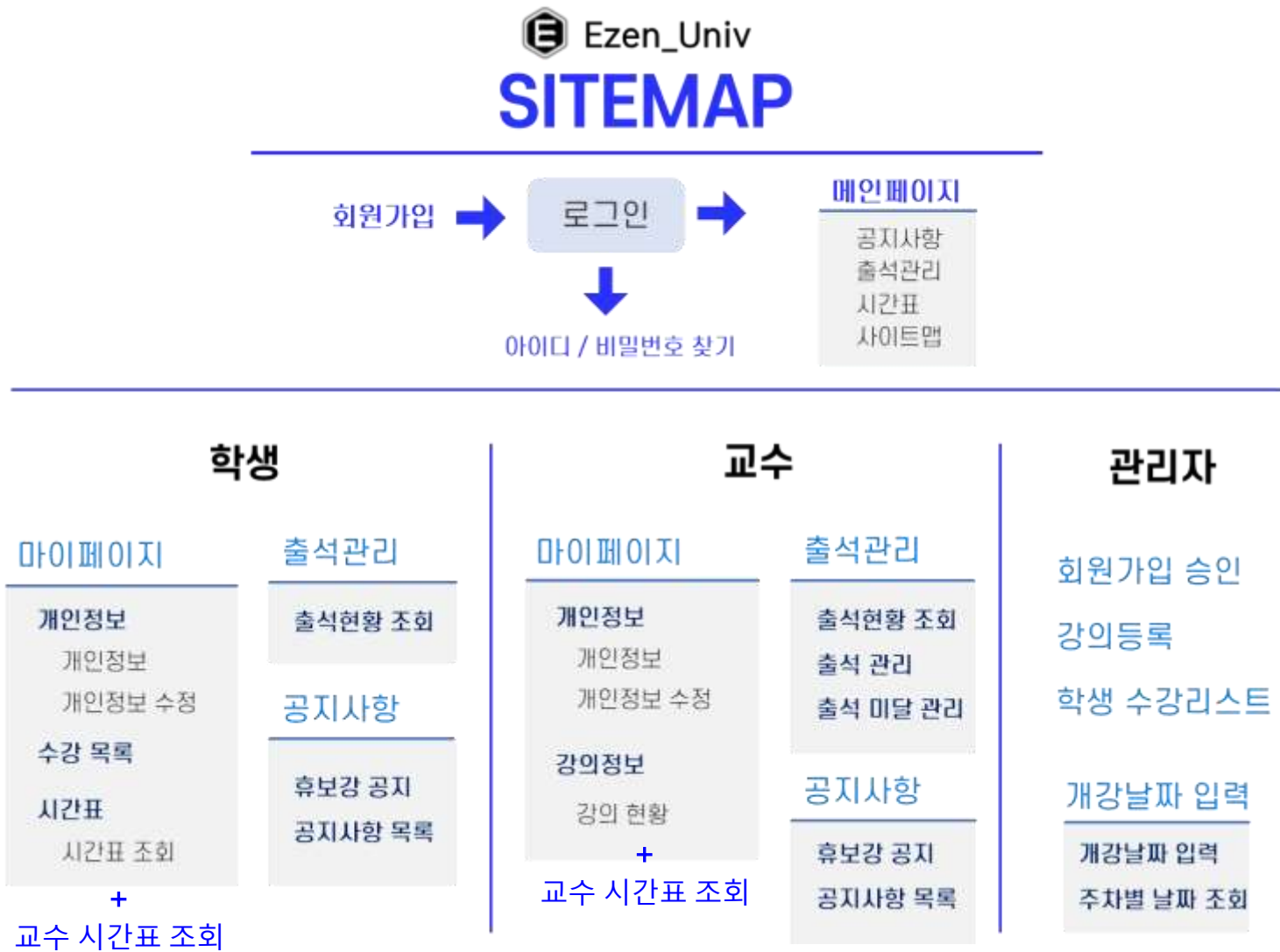
w_year -> startyear
w_semester -> starttm



08 Use Case 다이어그램

Ezen_Developer





Ezen_Developer

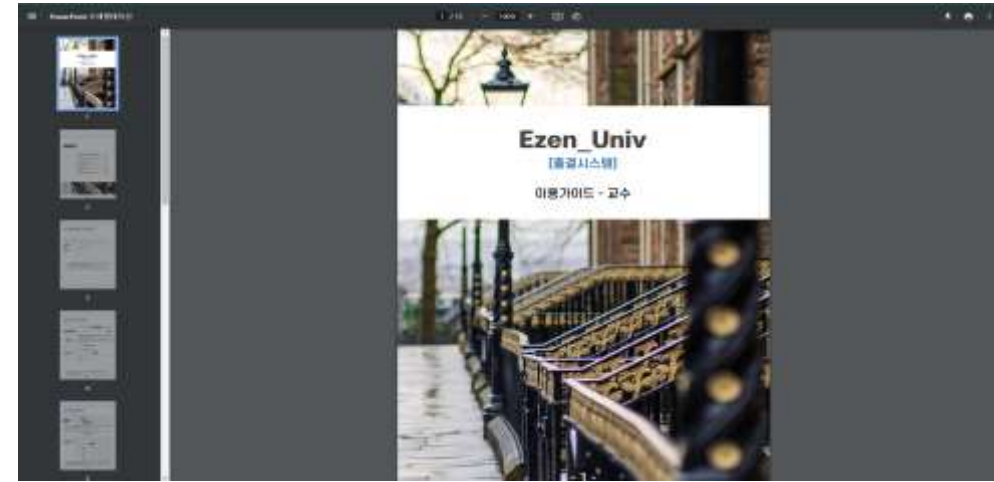


10 UI 페이지 흐름 (공통)

Ezen_Developer



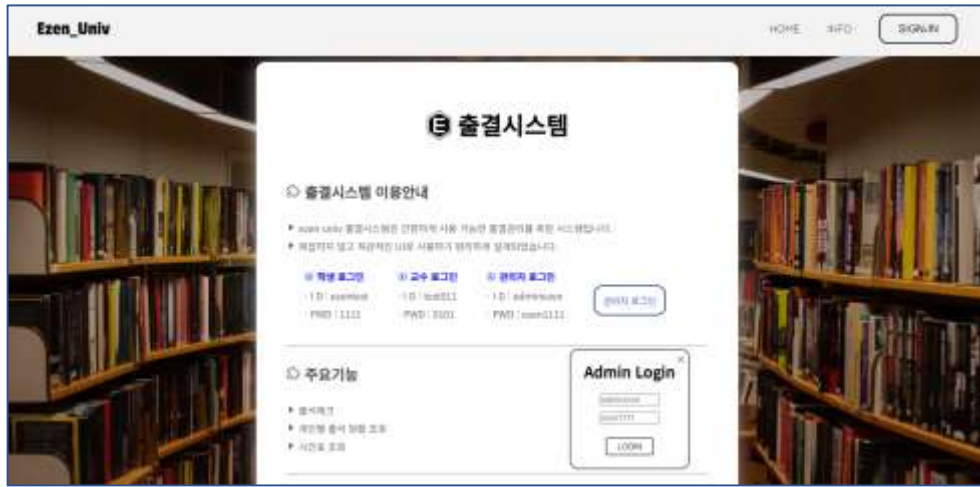
페이지정보



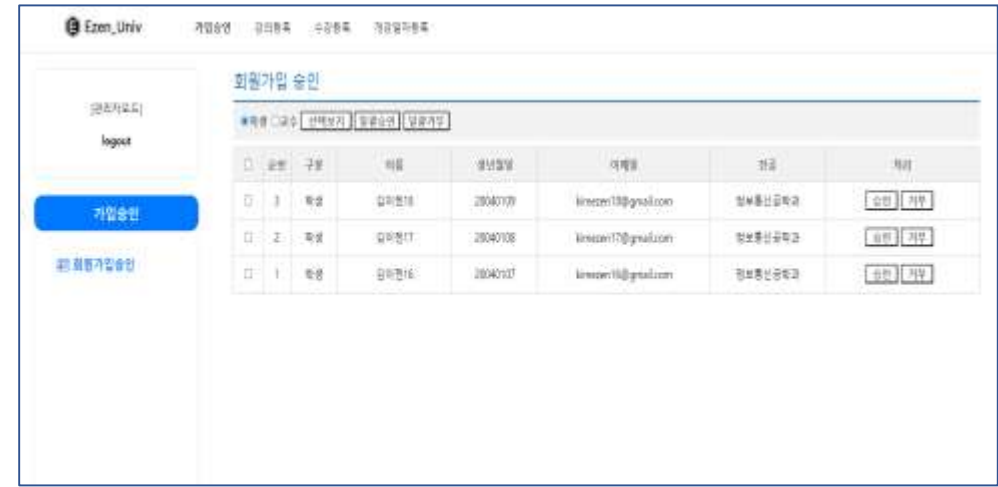
이용가이드

10 UI 페이지 흐름 (관리자)

Ezen_Developer



로그인



가입승인



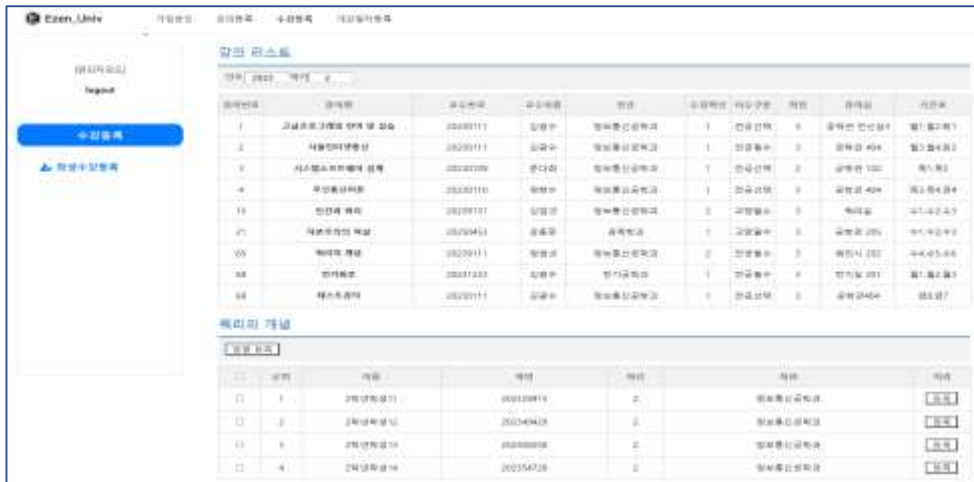
강의등록



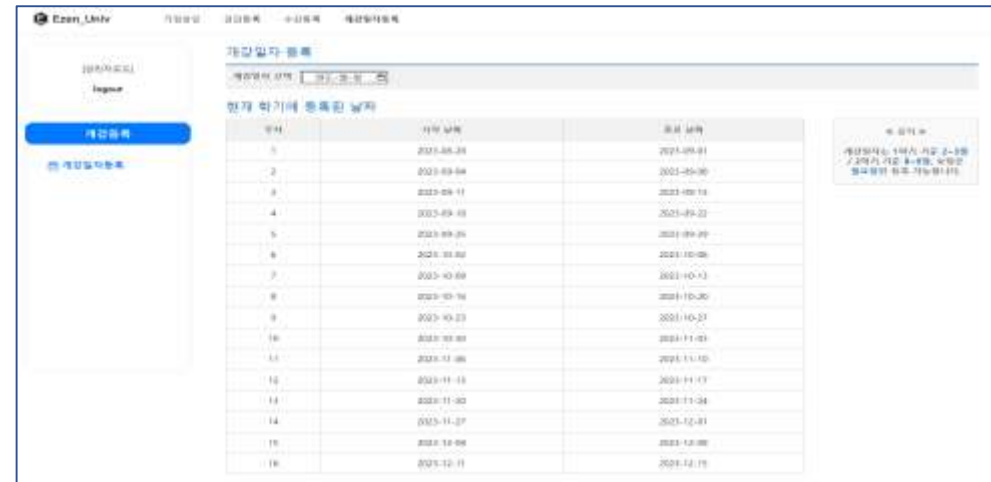
강의등록

10 UI 페이지 흐름 (관리자)

Ezen_Developer



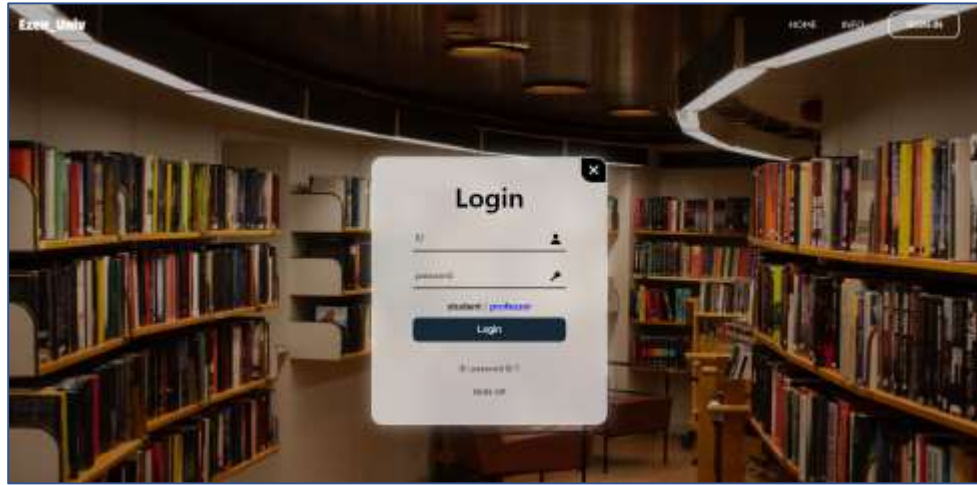
수강등록



개강날짜 등록

10 UI 페이지 흐름 (교수)

Ezen_Developer



로그인



메인페이지



마이페이지 - 개인정보



마이페이지 - 강의현황

10 UI 페이지 흐름 (교수)

Ezen_Developer

Ezen_Univ

마이페이지 | 출석관리 | 강의사항

교수 시간표 조회

날짜: 2023.10.01 | 학기: 3 | 교수: 김민준 | **조회**

교과	교과명	교과번호	교과시간	교과시간	교과시간	교과시간
1	고급프로그래밍 언어 및 알고리즘	1001	1001	1001	1001	1001
2	고급프로그래밍 언어 및 알고리즘	1002	1002	1002	1002	1002
3	고급프로그래밍 언어 및 알고리즘	1003	1003	1003	1003	1003
4	고급프로그래밍 언어 및 알고리즘	1004	1004	1004	1004	1004
5	고급프로그래밍 언어 및 알고리즘	1005	1005	1005	1005	1005
6	고급프로그래밍 언어 및 알고리즘	1006	1006	1006	1006	1006
7	고급프로그래밍 언어 및 알고리즘	1007	1007	1007	1007	1007
8	고급프로그래밍 언어 및 알고리즘	1008	1008	1008	1008	1008
9	고급프로그래밍 언어 및 알고리즘	1009	1009	1009	1009	1009
10	고급프로그래밍 언어 및 알고리즘	1010	1010	1010	1010	1010

마이페이지 - 교수 시간표 조회

Ezen_Univ

마이페이지 | 출석관리 | 강의사항

출석현황 조회

날짜: 2023.10.01 | 학기: 3 | 교수: 김민준 | **조회**

교과	교과명	교과번호	교과시간	교과시간	교과시간	교과시간	교과시간
1	고급프로그래밍 언어 및 알고리즘	1001	1001	1001	1001	1001	1001
2	고급프로그래밍 언어 및 알고리즘	1002	1002	1002	1002	1002	1002
3	고급프로그래밍 언어 및 알고리즘	1003	1003	1003	1003	1003	1003
4	고급프로그래밍 언어 및 알고리즘	1004	1004	1004	1004	1004	1004
5	고급프로그래밍 언어 및 알고리즘	1005	1005	1005	1005	1005	1005
6	고급프로그래밍 언어 및 알고리즘	1006	1006	1006	1006	1006	1006
7	고급프로그래밍 언어 및 알고리즘	1007	1007	1007	1007	1007	1007
8	고급프로그래밍 언어 및 알고리즘	1008	1008	1008	1008	1008	1008
9	고급프로그래밍 언어 및 알고리즘	1009	1009	1009	1009	1009	1009
10	고급프로그래밍 언어 및 알고리즘	1010	1010	1010	1010	1010	1010

출석관리 - 출석현황 조회

Ezen_Univ

마이페이지 | 출석관리 | 강의사항

출석관리

날짜: 2023.10.01 | 학기: 3 | 교수: 김민준 | **조회**

교과	교과명	교과번호	교과시간	교과시간	교과시간	교과시간	교과시간
1	고급프로그래밍 언어 및 알고리즘	1001	1001	1001	1001	1001	1001
2	고급프로그래밍 언어 및 알고리즘	1002	1002	1002	1002	1002	1002
3	고급프로그래밍 언어 및 알고리즘	1003	1003	1003	1003	1003	1003
4	고급프로그래밍 언어 및 알고리즘	1004	1004	1004	1004	1004	1004
5	고급프로그래밍 언어 및 알고리즘	1005	1005	1005	1005	1005	1005
6	고급프로그래밍 언어 및 알고리즘	1006	1006	1006	1006	1006	1006
7	고급프로그래밍 언어 및 알고리즘	1007	1007	1007	1007	1007	1007
8	고급프로그래밍 언어 및 알고리즘	1008	1008	1008	1008	1008	1008
9	고급프로그래밍 언어 및 알고리즘	1009	1009	1009	1009	1009	1009
10	고급프로그래밍 언어 및 알고리즘	1010	1010	1010	1010	1010	1010

출석관리 - 출석 관리



Ezen_Univ

마이페이지 | 출석관리 | 강의사항

출석관리

날짜: 2023.10.01 | 학기: 3 | 교수: 김민준 | **조회**

교과	교과명	교과번호	교과시간	교과시간	교과시간	교과시간	교과시간
1	고급프로그래밍 언어 및 알고리즘	1001	1001	1001	1001	1001	1001
2	고급프로그래밍 언어 및 알고리즘	1002	1002	1002	1002	1002	1002
3	고급프로그래밍 언어 및 알고리즘	1003	1003	1003	1003	1003	1003
4	고급프로그래밍 언어 및 알고리즘	1004	1004	1004	1004	1004	1004
5	고급프로그래밍 언어 및 알고리즘	1005	1005	1005	1005	1005	1005
6	고급프로그래밍 언어 및 알고리즘	1006	1006	1006	1006	1006	1006
7	고급프로그래밍 언어 및 알고리즘	1007	1007	1007	1007	1007	1007
8	고급프로그래밍 언어 및 알고리즘	1008	1008	1008	1008	1008	1008
9	고급프로그래밍 언어 및 알고리즘	1009	1009	1009	1009	1009	1009
10	고급프로그래밍 언어 및 알고리즘	1010	1010	1010	1010	1010	1010

출석관리 - 출석 관리

10 UI 페이지 흐름 (교수)

Ezen_Developer



출석관리 - 출석 미달 관리



공지사항 - 휴보강 글쓰기



공지사항 - 휴보강 공지



공지사항 - 휴보강 상세보기

10 UI 페이지 흐름 (학생)

Ezen_Developer



페인페이지



마이페이지 - 개인정보



마이페이지 - 수강목록



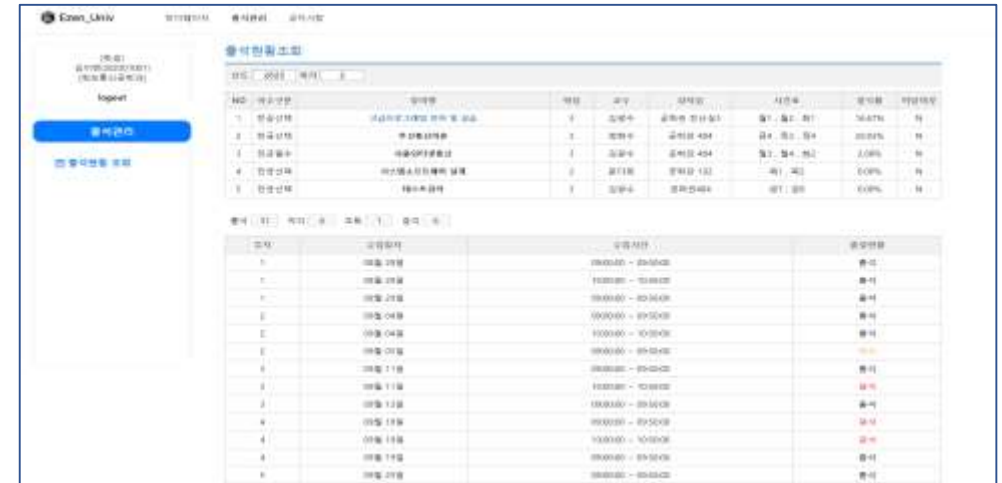
마이페이지 - 시간표 조회

10 UI 페이지 흐름 (학생)

Ezen_Developer



마이페이지 - 교수 시간표 조회



출석관리 - 출석현황 조회



공지사항 - 휴보강 공지



공지사항 - 휴보강 공지 상세보기

11 주요 기능 관리자 - 회원가입 승인

Ezen_Developer

회원가입 승인

선택	구분	이름	생년월일	이메일	학과	처리
<input type="checkbox"/>	학생	유성현	20041003	kimezen10@gmail.com	바이오메디칼공학부	<input type="button" value="승인"/> <input type="button" value="거부"/>
<input type="checkbox"/>	학생	윤창호	20040903	kimezen9@gmail.com	바이오메디칼공학부	<input type="button" value="승인"/> <input type="button" value="거부"/>
<input type="checkbox"/>	학생	백포남	20040703	kimezen8@gmail.com	바이오메디칼공학부	<input type="button" value="승인"/> <input type="button" value="거부"/>
<input type="checkbox"/>	학생	주영호	20040603	kimezen7@gmail.com	바이오메디칼공학부	<input type="button" value="승인"/> <input type="button" value="거부"/>
<input type="checkbox"/>	학생	전현아	20040503	kimezen6@gmail.com	바이오메디칼공학부	<input type="button" value="승인"/> <input type="button" value="거부"/>
<input type="checkbox"/>	학생	이합택	20040403	kimezen5@gmail.com	바이오메디칼공학부	<input type="button" value="승인"/> <input type="button" value="거부"/>

관리자

- 일괄승인
- 일괄거부
- 승인 거부
- 선택보기
- 가입 미승인 회원들

- JavaScript / jQuery
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

- 가입승인 클릭시 승인이 안된 회원들이 보여진다.
- 선택보기를 통해 학생과 교수 목록을 나누어 보기에 편하게 만들었다.
- 가입이 승인이 된 회원들은 회원번호를 부여받게 된다.
- 가입이 거부된 회원들의 정보는 삭제된다.

11 주요 기능 관리자 - 회원가입 승인

Ezen_Developer

```
// "선택보기" 버튼 클릭 이벤트 핸들러
$("#view").on("click", function() {
    const selectedValue = $("input[name='MemberList']:checked").val();

    // 상태를 보컬 스토리지에 저장
    localStorage.setItem('selectedMember', selectedValue);

    // 목록보기 설정
    if (selectedValue === 'studentAll') {
        $('#professorAlllist').hide();
        $('#professorCheckedBtn').hide();
        $('#studentAlllist').show();
        $('#studentCheckedBtn').show();
        $('#professorAlllist input[type="checkbox"]').prop('checked', false);
    } else {
        $('#studentAlllist').hide();
        $('#studentCheckedBtn').hide();
        $('#professorAlllist').show();
        $('#professorCheckedBtn').show();
        $('#studentAlllist input[type="checkbox"]').prop('checked', false);
    }
});

// 학생 일괄승인 버튼 클릭 이벤트
$('#submitButton1').on('click', function(event) {
    if ($('#input[name="student"]:checked').length === 0) {
        alert("한 명 이상의 학생을 선택하세요."); // 알림창 표시
        return false; // 폼 제출 방지
    }
    let confirmation = confirm("일괄승인하겠습니까?"); // 확인 창 표시
    if (confirmation) {
        let newAction = `${pageContext.request.contextPath}/admin/acceptStudentAllOk.do`;
        $('#studentAccept').prop('action', newAction);
        $('#studentAccept').submit(); // 폼 제출
    }
});
```

```
function acceptStudentOk(sidx){
    let confirmation = confirm("해당 학생의 회원가입을 승인하겠습니까?"); // 확인 창 표시
    if (confirmation) {
        // 학생 회원가입 승인
        $.ajax({
            type: "post",
            url: `${pageContext.request.contextPath}/admin/acceptStudentOk.do?sidx="+sidx,
            data: {sidx: sidx},
            dataType: "json",
            cache: false,
            success: function(data){
                //alert("승인성공");
            }
        });
    }
}
```

- 선택보기를 이벤트로 해당 리스트 <DIV> 숨기거나 보여주는 식으로 작동

Controller

```
String[] selectedOptions = request.getParameterValues("student");
int[] intSelectedOptions = new int[selectedOptions.length];

for (int i = 0; i < selectedOptions.length; i++) {
    intSelectedOptions[i] = Integer.parseInt(selectedOptions[i]);
}

int value = 0;
AdminDao add = new AdminDao();
value = add.studentCheckedAccept(intSelectedOptions);
```

- 일괄 승인과 일괄 거부는 form으로 제출

- 개별 승인, 거부는 ajax로 처리

```
public int studentCheckedAccept(int[] selectedData) {
    PreparedStatement selectS_no = null;
    PreparedStatement insertS_no = null;
    ResultSet rs = null;
    int exec = 0;
    int currentYear = Year.now().getValue();

    String selectCnt = "select count(*) as cnt from student where s_no=?";
    String selectS_no = "update student set s_yym='Y', s_no=? where sidx=?";

    try {
        // 트랜잭션 시작
        conn.setAutoCommit(false);

        for (int value : selectedData) {
            int s_no;

            // s_no value에 따라 새로운 s_no 값으로 중복 체크
            do {
                s_no = currentYear * 100000 + (int) (Math.random() * 89999);

                // 중복 체크
                selectS_no = conn.prepareStatement(selectCnt);
                selectS_no.setInt(1, s_no);
                rs = selectS_no.executeQuery();
            } while (rs.next() && rs.getInt("cnt") > 0);

            // 중복된 s_no를 없애기
            insertS_no = conn.prepareStatement(selectS_no);
            insertS_no.setInt(1, s_no);
            insertS_no.setInt(2, value);
            int rowAffected = insertS_no.executeUpdate();
            exec += rowAffected;
        }

        // 트랜잭션 커밋
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        if (rs != null) rs.close();
        if (selectS_no != null) selectS_no.close();
        if (insertS_no != null) insertS_no.close();
    }

    return exec;
}
```

- 컨트롤러를 통해 넘어온 해당 회원(들)의 인덱스(들)를 통해 데이터베이스에 넘길 준비를 한다.

- 그 전에 회원에게 랜덤으로 번호를 부여해 그 값이 중복이 있는지 반복문을 돌려 데이터베이스를 확인한다. 없으면 승인여부 N/Y와 함께 업데이트 된다.

11 주요 기능 관리자 - 강의등록

Ezen_Developer

Ezen_Univ
가일승근
강의등록
수강등록
개강일자등록

(관리자로그인)
logout

강의등록

새로입강의등록

강의등록현황

년도 2023 학기 2

강의번호	강의명	교수번호	교수이름	성명	수강학년	태수구분	학점	강의실	시간표	처리
1	고급프로그래밍 언어 및 실습	20230111	김광수	정보통신공학과	1	전공선택	3	공학관 전산실3	월1.월2.화1	삭제
2	서버관리및통신	20230111	김광수	정보통신공학과	1	전공필수	3	공학관 404	월1.월4.화2	삭제
3	시스템소프트웨어 설계	20230109	문다희	정보통신공학과	1	전공선택	2	공학관 102	목1.목2	삭제
4	무선통신이론	20230110	정현수	정보통신공학과	1	전공선택	3	공학관 404	목3.목4.금4	삭제
15	임상과 적회	20230131	김영호	정보통신공학과	2	교양필수	3	취미실	수1.수2.수3	삭제
21	지능정보의 역할	20230433	김종훈	경제학과	1	교양필수	3	공학관 305	수1.수2.수3	삭제
65	취미와 개발	20230111	김영호	정보통신공학과	2	전공필수	3	취미실 202	수4.수5.수6	삭제
88	전기회로	20231223	김광수	전기공학과	1	전공필수	3	전기실 301	월1.월2.월3	삭제
89	테스트강의	20230111	김광수	정보통신공학과	1	전공선택	3	공학관404	금요일7	삭제

강의등록

강의등록

강의명	이강에 직접 입력하세요	한정	목록을 선택하세요
태수구분	목록을 선택하세요	교수번호	전공 선택시 목록선택
수강학년	목록을 선택하세요	담당교수	교수번호 선택시 자동입력
학점	목록을 선택하세요	등록의시간	목록을 선택하세요

다음

강의등록-시간표

강의실	요일	교시	학기	년도	처리

강의실
이강에 직접 입력

요일
목록선택

교시
1교시 2교시 3교시 4교시 5교시 6교시 7교시 8교시 9교시

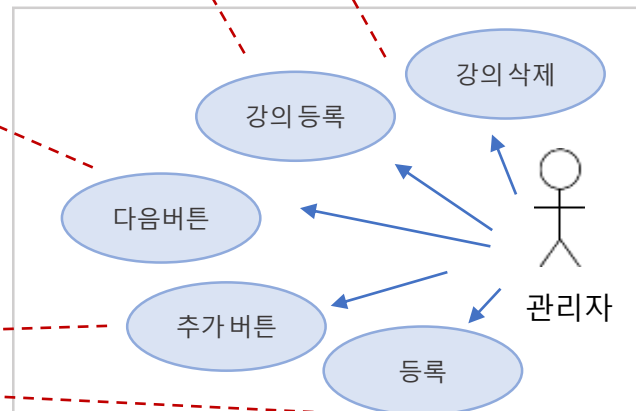
학기
목록선택

년도
목록선택

삭제 등록

추가

- 강의등록 클릭시 강의등록 현황이 보여진다.
- 해당 강의를 삭제할 수 있으며, 강의 등록버튼을 누르면 현황표 밑으로 감춰 놓았던 강의 등록 테이블이 생긴다.
- 다음 버튼을 누르면, 강의등록 시간표 테이블로 바뀌고 시간표를 입력하고 추가 버튼을 누르면 테이블에 등록이 된다. 여러 개를 등록할 수 있다.



- JavaScript / jQuery
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

11 주요 기능 관리자 - 강의등록

Ezen_Developer

```
$('#nextBtn').on('click', function() {  
    //alert("확인");  
    let p_no = $('#p_no').val();  
    let p_name = $('#p_name').val();  
    let c_grade = $('#c_grade').val();  
    let c_major = $('#c_major').val();  
  
    if(c_major == null || c_major === ""){  
        alert("전공을 선택하세요");  
        return;  
    }else if(p_no == null || p_no === ""){  
        alert("교수번호를 선택하세요");  
        return;  
    }else if(p_name == null || p_name === ""){  
        alert("교수이름을 입력하세요");  
        return;  
    }else if(c_grade == null || c_grade === ""){  
        alert("수강학년을 선택하세요");  
        return;  
    }  
    $.ajax({  
        type: "post",  
        url: "${pageContext.request.contextPath}/admin/professorVerification.do",  
        dataType: "json",  
        data: {  
            "p_no": p_no,  
            "p_name": p_name,  
        },  
        cache: false,  
        success: function(data) {  
            if(data.value==1){  
                $('#courseListInput').hide();  
                $('#courseTimeListInput').show();  
            }  
        }  
    });  
}
```

- 다음 버튼을 눌렀을 시 시간표 테이블 중복값을 최대한 없애기 위해 필수 요소의 항목을 입력.

```
$('#previousBtn').on('click', function() {  
    $('#courseTimeListInput').hide();  
    $('#courseListInput').show();  
    $('#timeTable tbody').empty();  
});
```

- 이전 버튼을 눌렀을 시 시간표 테이블을 빈 값으로 만들어 중복된 값으로 꼬이지 않게 방지

전공	바이오메디컬공학부
교수번호	전공 조회시 목록선택
담당교수	전공 조회시 목록선택
총강의시간	목록을 선택하세요

바이오메디컬공학부
20235146
강세연

```
// 전공 선택시 교수번호 가져오는 함수  
$('#c_major').on('change', function() {  
    $('#p_name').val("");  
    $('#p_no').val("");  
    let c_major = $('#c_major').val();  
  
    $.ajax({  
        type: "post",  
        url: "${pageContext.request.contextPath}/admin/registerView.do",  
        data: {  
            "c_major": c_major  
        },  
        dataType: "json",  
        cache: false,  
        success: function(data) {  
            let professorNumberOption = $("#p_no");  
  
            // 기존 datalist 비우기  
            professorNumberOption.empty();  
  
            // 첫 번째 옵션 추가  
            professorNumberOption.append('<option value="" disabled selected>전공 조회시 목록선택</option>');  
  
            // JSON 데이터를 datalist에 추가  
            $.each(data, function(index, item) {  
                let NumberOption = $('#<option>').attr("value", item.p_no).text(item.p_no);  
                professorNumberOption.append(NumberOption);  
            });  
  
            // 강의등록 조회버튼 시 교수번호와 교수이름 가져오는 함수  
            $('#p_no').on('change', function() {  
                $('#p_name').val("");  
                let p_no = $('#p_no').val();  
  
                $.ajax({  
                    type: "post",  
                    url: "${pageContext.request.contextPath}/admin/registerProfessorView.do",  
                    data: {  
                        "p_no": p_no  
                    },  
                    dataType: "json",  
                    cache: false,  
                    success: function(data) {  
                        if(data.value != undefined && data.value != null && data.value != ""){  
                            $('#p_name').val(data.value); // p_name 값을 input에 넣음  
                        }else alert("해당 교수번호로 작성되는 교수이름이 없습니다.");  
                    },  
                    error: function(request, status, error) {  
                        alert("통신오류 실패");  
                        alert("code:" + request.status + "\n" + "message:" + request.responseText + "\n" + "error:" + error);  
                    }  
                });  
            });  
        }  
    });  
});
```

- 해당 전공을 선택시 교수번호들을 데이터를 가져와 옵션 값에 넣어준다.

- 해당 교수번호에 해당하는 교수이름을 자동선택하는 함수.

11 주요 기능 관리자 - 강의등록

Ezen_Developer

```
// 기존 강의들의 각 행을 읽어와서 중복 확인
$( "#timeTable tbody tr" ).each(function() {
    let existingCourseRoom = $(this).find("td:eq(0)").text();
    let existingWeek = $(this).find("td:eq(1)").text();
    let existingPeriod = $(this).find("td:eq(2)").text();
    let existingSemester = $(this).find("td:eq(3)").text();
    let existingYear = $(this).find("td:eq(4)").text();

    if (
        existingCourseRoom === courseRoomValue &&
        existingWeek === weekValue &&
        existingPeriod === periodValue &&
        existingSemester === semesterValue &&
        existingYear === yearValue
    ) {
        isDuplicate = true;
        return false; // each loop 종료
    }
});

// 중복된 값이 있다면 알림을 표시하고 함수 종료
if (isDuplicate) {
    alert("이미 강의등록-시간표에 입력되어 있습니다.");
    return;
}

// 중복된 값이 없다면 데이터를 행 추가 요청 계속 수행

let cells = $( "#timeTable tbody td" );
let allCellsFilled = true;

cells.each(function() {
    if ($(this).text() === "" || $(this).html() === "&nbsp;") {
        $(this).text(courseRoomValue);
        $(this).next().text(weekValue);
        $(this).next().next().text(periodValue);
        $(this).next().next().next().text(semesterValue);
        $(this).next().next().next().next().text(yearValue);
        allCellsFilled = false;
        return false; // loop 종료
    }
});

if (allCellsFilled) {
    // 새로운 행 추가
    let newRow = "<tr><td>" + courseRoomValue + "</td>" +
        "<td>" + weekValue + "</td>" +
        "<td>" + periodValue + "</td>" +
        "<td>" + semesterValue + "</td>" +
        "<td>" + yearValue + "</td>" +
        "<td><button class='deleteRow'>삭제</button></td></tr>";
    $( "#timeTable" ).append(newRow);
}

// 입력값 초기화
$("#ct_week").val("");

// 새로운 행의 마지막 셀에 추가
$( "#timeTable tbody tr" ).each(function() {
    if ($(this).find("td").length > 0) {
        let lastCell = $(this).find("td").last();
        if (lastCell.find(".deleteRow").length === 0) {
            lastCell.append("<button class='deleteRow'>삭제</button>");
        }
    }
});
```

- 1차적으로 강의 시간표 테이블에 등록시 값이 없거나 같은 행에 중복된 값을 걸러내는 기능의 함수.

select COUNT(*) AS cnt
from course a

JOIN coursetime b ON a.cidx = b.cidx
JOIN professor c ON c.pidx = a.pidx
where ((REPLACE(ct_room, ' ', '') = REPLACE(?, ' ', '')) OR
(c_major = ? AND c_grade = ?) OR p_no = ?
AND ct_week = ?
AND pe_period = ?
AND ct_semester = ?
AND ct_year = ?

- 2차적으로 강의 시간표에 추가 버튼 클릭시 등록된 중복 값을 걸러내는 쿼리

```
$( "#timeTable tr:eq(0)" ).each(function() {
    let rowData = $(this).find("td");

    // 빈 값 확인
    if ($(rowData).filter(':empty').length > 0) {
        alert("시간표를 입력하지 않았습니다.");
        return false;
    }

    let courseDetails = {
        room: $(rowData[0]).text(),
        day: $(rowData[1]).text(),
        period: $(rowData[2]).text(),
        semester: $(rowData[3]).text(),
        year: $(rowData[4]).text()
    };
    tableData.push(courseDetails);
});

$.fn.action = "<$.request.getContextPath()%/admin/courseRegisterAction.do";
$.fn.method = "post";
let tableInput = $( "<input>" )
    .attr("type", "hidden")
    .attr("name", "tableData")
    .val(JSON.stringify(tableData));

$(fn).append(tableInput);
$.submit();
}
```

- 결과적으로 중복된 것이 단 하나라도 없으면 시간표 테이블에 배열로 된 데이터들을 JSON.stringify로 form 제출시 같이 추가하여 제출한다.

```
ArrayList<CourseTimeVo> ctv = new ArrayList<>();
// JSONArray를 순회하며 객체 가져오기
for (Object obj : jsonArray) {
    JSONObject jsonObject = (JSONObject) obj;

    // 가져온 객체에서 필요한 값을 얻기
    String ct_room = (String) jsonObject.get("room");
    String ct_week = (String) jsonObject.get("day");
    int pe_period = Integer.parseInt((String) jsonObject.get("period"));
    int ct_semester = Integer.parseInt((String) jsonObject.get("semester"));
    int ct_year = Integer.parseInt((String) jsonObject.get("year"));

    // CourseTimeVo 객체 생성
    CourseTimeVo courseTimeVo = new CourseTimeVo();
    courseTimeVo.setCt_room(ct_room);
    courseTimeVo.setCt_week(ct_week);
    courseTimeVo.setPe_period(pe_period);
    courseTimeVo.setCt_semester(ct_semester);
    courseTimeVo.setCt_year(ct_year);

    // 리스트에 추가
    ctv.add(courseTimeVo);
}
```

- Controlloer에서 직렬화된 테이블 시간표로 파싱.역직렬하여 값들을 Vo에 담는다.
- 해당 값들을 가지고 DAO 가서 데이터 베이스에 집어 넣는다.

11 주요 기능 관리자 - 수강등록

Ezen_Developer

Ezen_Univ

가장유연한 관리자

로그아웃

수강등록

학생수강등록

강의 리스트

년도: 2023 학기: 2

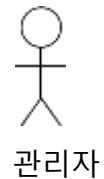
강의번호	강의명	교수번호	교수이름	전공	수강학년	학수구분	학점	강의실	시간표
1	고급프로그래밍 언어 및 실습	20230111	김영수	정보통신공학과	1	전공선택	3	공학관 301실	월1,월2,월3
2	사물인터넷통신	20230111	김영수	정보통신공학과	1	전공필수	3	공학관 404	월3,월4,월5
3	시스템소프트웨어 설계	20230109	문다희	정보통신공학과	1	전공선택	2	공학관 102	목1,목2
4	무선통신이론	20230110	김영수	정보통신공학과	1	전공선택	3	공학관 404	목2,목4,금4
15	인간과 컴터	20239151	김영수	정보통신공학과	2	교양필수	3	공학관 404	수1,수2,수3
21	자본주의의 역설	20230459	김영수	경제학과	1	교양필수	3	공학관 205	수1,수2,수3
65	컴퓨터 개론	20239111	김영수	정보통신공학과	2	전공필수	3	공학관 202	수4,수5,수6
68	전기회로	20231223	김영수	전기공학과	1	전공필수	3	공학관 201	월1,월2,월3
69	테스트공학	20230111	김영수	정보통신공학과	1	전공선택	3	공학관 404	금8,금9
70	데이터공학	20238970	최장준	바이오메디칼공학과	1	전공필수	3	공학관 305	월1,월2,월3

고급프로그래밍 언어 및 실습

강의 등록

수강번호	이름	학번	학년	학과	처리
1	이대훈	202348754	1	정보통신공학과	등록
2	김다희	202314331	1	정보통신공학과	등록
3	김다희	202309705	1	정보통신공학과	등록

- 수강등록 클릭시 수강등록 현황을 보여준다. 기존 강의등록을 가져오는 쿼리와 같다.
- 강의 이름을 클릭하면 해당 수강해야할 학생들 목록이 뜬다.
- 해당 학생들의 학과의 강의로서, 등록 가능한 버튼만 있다.



관리자

수강 선택

수강해야할
학생들 목록

수강 등록

- JavaScript / jQuery
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

11 주요 기능 관리자 - 수강등록

Ezen_Developer

// 해당 수강에 등록해야할 학생들 리스트

```
$(".course-name").click(function () {  
    //alert("확인");  
    let cidx = $(this).closest('tr').data('id');  
    let c_name = $(this).closest('tr').find('.course-name').text();  
    let c_major = $(this).closest('tr').find('.major-name').text();  
    let c_grade = $(this).closest('tr').find('.grade').text();  
  
    $.ajax({  
        type: 'POST',  
        url: "${pageContext.request.contextPath}/admin/courseMatchStudent.do",  
        data: {  
            "cidx": cidx,  
            "c_major": c_major,  
            "c_grade": c_grade  
        },  
        success: function(data) {  
            console.log(data);  
            //alert("등록 확인");  
            $("#studentlist h3").text(c_name); //과목 리스트 h3에 불러온 과목명 입력  
            courseMatchStudent(data, cidx); //학생들 리스트 갱신과 함께 cidx값도 같이 넘기기  
        },  
        error: function(error) {  
            alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);  
        }  
    });  
});  
  
return;
```

- 강의 이름 클릭 이벤트로 발생시, 수강 인원 데이터를 가져오는 위해 필수 value 값들을 ajax로 보낸다.

// 학생들 리스트 테이블

```
function courseMatchStudent(data, cidx){  
    //console.log("해당 courseId:"+courseId);  
    let parsedData = JSON.parse("["+data+"]");  
    let str="";  
  
    parsedData.forEach(function (item, index){  
        let acceptBtn= "<button type='button' id='btn' onclick='courseAccept(\""+item.sidx+"\",\""+cidx+"')'>등록</button>";  
        let autoIncrementNumber = index + 1; //인덱스 자동증가  
  
        str += "<tr><td><input type='checkbox' name='studentCheckbox' value='"+item.sidx+"'></td> *  
        "<td>" + autoIncrementNumber + "</td>" +  
        "<td>" + item.c_name + "</td>" +  
        "<td>" + item.s_no + "</td>" +  
        "<td>" + item.s_grade + "</td>" +  
        "<td>" + item.s_major + "</td>" +  
        "<td>" + acceptBtn + "</td></tr>";  
    });  
  
    $("#batchBtn").val(cidx);  
    $("#studentlist tbody").html(str); // 학생리스트 tbody 부분과 입력  
    $("#studentlist").show(); // <div id="studentlist"> 학생 목록 표 활성화
```

- 해당 데이터를 테이블을 만들어 DIV tbody 부분에 입력

- 기존 숨겨 놓았던 학생들 리스트는 테이블은 show로 바꿔준다.

// 일괄 처리 버튼 클릭 이벤트

```
$("#batchBtn").click(function () {  
    if ($("#input[name='studentCheckbox']:checked").length === 0) {  
        alert("한 명 이상의 학생을 선택하세요."); // 일괄할 표시  
        return false; // 볼 재발 방지  
    }  
    let confirmation = confirm("일괄 등록하시겠습니까?"); // 확인 창 표시  
    if (confirmation) {  
        let cidx = $(this).val();  
        let selectedStudent = [];  
        //console.log("선택한 버튼의 value 값:", cidx);  
  
        // 선택된 체크박스의 값을 배열에 추가  
        $("#[name='studentCheckbox']:checked").each(function () {  
            selectedStudent.push($(this).val());  
        });  
  
        // 배열을 JSON 문자열로 변환  
        let jsonIds = JSON.stringify(selectedStudent);  
  
        // AJAX를 통해 서버로 전송  
        $.ajax({  
            type: 'POST',  
            url: "${pageContext.request.contextPath}/admin/checkedEnrollAction.do",  
            data: {  
                selectedStudent : jsonIds,  
                cidx : cidx  
            },  
            success: function (data) {  
                alert("해당 학생들의 등록이 완료되었습니다");  
            }  
        });  
    }  
});
```

- 일괄처리 하기 위한 클릭 이벤트와 ajax

// 등록 버튼 이벤트

```
function courseAccept(sidx,cidx){  
    let confirmation = confirm("해당 학생을 등록하시겠습니까?"); // 확인 창 표시  
    if (confirmation) {  
        $.ajax({  
            type: 'POST',  
            url: "${pageContext.request.contextPath}/admin/EnrollAction.do",  
            data: {  
                "sidx": sidx,  
                "cidx": cidx  
            },  
            success: function(data) {  
                alert("해당 학생의 등록이 완료되었습니다");  
            }  
        });  
    }  
}
```

- 개별 등록 클릭 이벤트와 ajax

11 주요 기능 관리자 - 수강등록

Ezen_Developer

```
String selectedStudent = request.getParameter("selectedStudent");
String cidx = request.getParameter("cidx");

JSONParser parser = new JSONParser();
JSONArray jsonArray = null;

// selectedStudent를 JSON 배열로 파싱
try {
    jsonArray = (JSONArray) parser.parse(selectedStudent);
} catch (org.json.simple.parser.ParseException e) {
    e.printStackTrace();
}

ArrayList<MemberVo> mv = new ArrayList<>();
// JSON 배열을 순회하며 각 값을 처리
for (Object obj : jsonArray) {
    String sidxString = (String) obj;
    int sidx = Integer.parseInt(sidxString);

    // MemberVo 객체 생성
    MemberVo memberVo = new MemberVo();
    memberVo.setSidx(sidx);

    mv.add(memberVo);
}

int value = 0;
AdminDao add = new AdminDao();
value = add.checkedEnroll(mv, Integer.parseInt(cidx));

String str = "{ \"value\": \"\" + value + \"\" }";
out.println(str);
```

- 대부분의 많은 데이터 값들은 JSON으로 파싱하여 리스트에 넣는다.

Controller

```
public ArrayList<MemberVo> courseMatchStudentlist(int cidx, String c_major, int c_grade) {
    ArrayList<MemberVo> sMatchList = new ArrayList<MemberVo>();
    ResultSet rs = null;

    String sql = "select a.sidx,a.s_name,a.s_no,a.s_grade,a.s_major\n"
        + "from student a left join courselist b on a.sidx=b.sidx\n"
        + "left join course c on b.cidxc=c.cidc\n"
        + "where (b.sidx not in (select sidx from courselist where cidx =?) or b.cidc is null)\n"
        + "and c_yn='Y' and a.s_major=? and a.s_grade=?\n"
        + "group by a.sidx,a.s_name,a.s_no,a.s_grade,a.s_major";

    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, cidx);
        pstmt.setString(2, c_major);
        pstmt.setInt(3, c_grade);
        rs = pstmt.executeQuery();

        while(rs.next()){
            MemberVo mv = new MemberVo();

            mv.setSidx(rs.getInt("a.sidx"));
            mv.setS_name(rs.getString("a.s_name"));
            mv.setS_no(rs.getInt("a.s_no"));
            mv.setS_grade(rs.getInt("a.s_grade"));
            mv.setS_major(rs.getString("a.s_major"));

            sMatchList.add(mv);
        }
    }
```

- 데이터 베이스에서 리스트를 가져올 때 DAO 모습이다.

DAO

```
public int checkedEnroll(ArrayList<MemberVo> mv, int cidx) {
    PreparedStatement selectStmt = null;
    PreparedStatement insertStmtSidxCidx = null;
    ResultSet rs = null;
    int exec = 0;

    try {
        conn.setAutoCommit(false);

        // ctidx 값을 가져오기
        String selectCidx = "SELECT ctidx FROM coursetime a JOIN course b ON a.cidc = b.cidc WHERE b.cidc = ?";
        selectStmt = conn.prepareStatement(selectCidx);
        selectStmt.setInt(1, cidx);
        rs = selectStmt.executeQuery();

        // ctidx 값을 저장할 리스트
        ArrayList<Integer> ctidxList = new ArrayList<>();
        while (rs.next()) {
            int ctidx = rs.getInt("ctidx");
            ctidxList.add(ctidx);
        }

        // sidxArray와 ctidxList를 순회하며 데이터 입력
        String insertcourseList = "INSERT INTO courselist(sidx, cidc, ctidx) VALUES (?, ?, ?)";
        insertStmtSidxCidx = conn.prepareStatement(insertcourseList);

        for (MemberVo memberVo : mv) {
            int sidc = memberVo.getSidx();

            // ctidxList를 기준으로 sidx를 여러번 insert
            for (int ctidx : ctidxList) {
                insertStmtSidxCidx.setInt(1, sidc);
                insertStmtSidxCidx.setInt(2, cidc);
                insertStmtSidxCidx.setInt(3, ctidx);

                exec += insertStmtSidxCidx.executeUpdate();
            }
        }

        conn.commit();
    }
```

- 일괄 처리 등록할 때 트랜잭션을 사용한다.

DAO

11 주요 기능 교수 - 출석현황 조회

Ezen_Developer

출석현황조회

년도: 2023 학기: 2

NO	이수구분	강의명	세부전공	수강학년	강의실	시간표	수강인원	출석률
1	한공선택	고급프로그래밍 언어 및 실습	정보통신공학과	1	공학관 전상실3	월1, 월2, 화1	16	75.00%
2	한공필수	사물인터넷통신	정보통신공학과	1	공학관 404	월3, 월4, 화2	16	93.75%
3	한공선택	태스크관리	정보통신공학과	1	공학관404	연8, 연7	19	100.00%

강의명: 고급프로그래밍 언어 및 실습

No	학생	1교시	2교시	3교시	4교시	5교시	6교시	7교시	8교시	9교시	10교시	11교시	12교시	13교시	14교시	15교시	16교시	17교시	18교시	19교시	20교시
1	김이현 (202301001)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	출석	출석	출석	출석	출석	출석	출석	출석	출석	
2	김이현2 (202301002)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	출석	출석	출석	출석	출석	출석	출석	출석	출석	
3	김이현3 (202301003)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	출석	출석	출석	출석	출석	출석	출석	출석	출석	
4	김이현4 (202301004)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	출석	출석	출석	출석	출석	출석	출석	출석	출석	
5	김이현5 (202301005)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	출석	출석	출석	출석	출석	출석	출석	출석	출석	
6	김이현6 (202301006)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	결석	출석	결석	출석	출석	출석	출석	출석	출석	
7	김이현7 (202301007)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	결석	출석	결석	출석	출석	출석	출석	출석	출석	
8	김이현8 (202301008)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	결석	출석	결석	출석	출석	출석	출석	출석	출석	
9	김이현9 (202301009)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	결석	출석	결석	출석	출석	출석	출석	출석	출석	
10	김이현10 (202301010)	출석	출석	출석	출석	출석	결석	출석	결석	결석	출석	결석	출석	결석	출석	출석	출석	출석	출석	출석	

- 출석현황 조회 클릭시 강의목록 출력
- 강의 목록에 출석률도 함께 출력된다. (출석률은 (100 - 결석률)로 계산)
- 조회하고자 하는 강의 클릭시 하단에 전체 출결 상황이 출력되고 아직 출결이 이루어지지 않은 수업은 빈칸으로 처리된다.
- Ajax를 사용하여 새로고침 없이 하단 에 출결현황이 출력될 수 있도록 한다.



교수

강의 선택

출석현황
보기

- JavaScript / jQuery
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

11 주요 기능 교수 - 출석현황 조회

Ezen_Developer

I. 출결표 상단 출결시간(일자, 교시) 출력

학명	고급프로그래밍 언어 및 실습									
학생	08.28(월) 1교시	08.28(월) 2교시	08.29(화) 1교시	09.04(월) 1교시	09.04(월) 2교시	09.05(화) 1교시	09.11(월) 1교시	09.11(월) 2교시	09.12(화) 1교시	

```
function getAttendanceTable(cidx){
    $.ajax({
        type : "post",
        url : "${pageContext.request.contextPath}/attendance/getAttendanceTable.do",
        data : {
            "cidx" : cidx
        },
        dataType : "json",
        success : function(data){
            var str = "";
            str = "<thead><tr>";

            $(data).each(function(){
                str = str + "<td style='width: 60px; height: 50px;'>"+this.a_date+"("+"this.ct_week+"")<br>"+this.pe_period+"교시</td>";
            });

            str = str + "</tr></thead><tbody></tbody>";
            $("#list_table").html("<table class='listNo' id='listNo'>"+str+"</table>");

            getStudentList(cidx);

            return ;
        },
        error: function(request, status, error){
            alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
        }
    });
}
```

- 출결시간 출력 후에 학생목록 출력 함수 실행

CourseTime				
논리 이름	물리 이름	데이터 타입	날	허용
강의시간인덱스	ctidx	INT	N	N
강의인덱스	cidx	INT	N	N
강의실	ct_room	VARCHAR(30)	N	N
요일	ct_week	CHAR(2)	N	N
교시	pe_period	INT	N	N
학기	ct_semester	INT	N	N
년도	ct_year	INT	N	N
개강날짜	ct_open	DATE	N	N
종강날짜	ct_close	DATE	N	N

Weektb				
논리 이름	물리 이름	데이터 타입	날	허용
주차인덱스	widx	INT	N	N
주차	w_week	INT	N	N
시작날짜	w_start	DATE	N	N
마지막날짜	w_end	DATE	N	N
년도	w_year	YEAR	NULL	
학기	w_semester	INT	NULL	

```
select
    date_format((case
        when ct.ct_week='월' then date_add(w.w_start, interval 0 day)
        when ct.ct_week='화' then date_add(w.w_start, interval 1 day)
        when ct.ct_week='수' then date_add(w.w_start, interval 2 day)
        when ct.ct_week='목' then date_add(w.w_start, interval 3 day)
        when ct.ct_week='금' then date_add(w.w_start, interval 4 day)
        end),'%m.%d') as wk, ct.ct_week, ct.pe_period
from weektb w, coursetime ct
where ct.cidx=? and w.startyear=? and w.starttm=?
order by 1, 3
```

- 해당 년도, 학기, 강의의 모든 날짜 출력

-> CourseTime의 요일과 Weektb의 주차를 이용하여
날짜 계산

11 주요 기능 교수 - 출석현황 조회

Ezen_Developer

2. 출결표 왼쪽 학생목록 출력 및 각 td에 id 부여

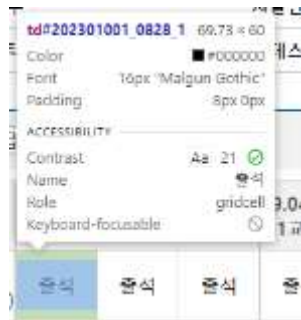
```
function getStudentList(cidx){
    $.ajax({
        type: "post",
        url: "${pageContext.request.contextPath}/attendance/getStudentList.do",
        data: {
            "cidx": cidx
        },
        dataType: "json",
        success: function(data){
            var studentlist = "<table style='width: 80px;'><thead><tr>"
            + "<td style='width: 30px; height: 60px;'>No</td>"
            + "<td style='width: 50px;'>학생</td></tr></thead><tbody>";
            $(data).each(function(){
                var str = "<tr>";
                studentlist = studentlist + "<tr><td style='height:60px;'></td><td>"+this.s_name+"<br>"+this.s_no+"</td></tr>"
                var count = this.c_totaltime;
                for(var i=0; i<count; i++){
                    var tablehead = $('#listNo thead');
                    var headtr = tablehead.children();
                    var td = headtr.children();
                    var tddate = td.eq(i).text();
                    var onlydate = tddate.substring(0,2)+tddate.substring(3,5)+" "+tddate.substring(8,9);
                    str = str + "<td id='"+this.s_no+"_"+onlydate+"' style='height: 60px;'></td>";
                }
                str = str + "</tr>";
                $('#listNo tbody').append(str);
            });
            studentlist = studentlist + "</tbody></table>";
            $('#list_table_student').html(studentlist);
        },
        error: function(request, status, error){
            alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
        }
    });
}
```

- 목록출력, id 부여된 td 추가 후 각 출결상태 호출

No	학생
1	김미현1 (202301001)
2	김미현2 (202301002)
3	김미현3 (202301003)
4	김미현4 (202301004)
5	김미현5 (202301005)
6	김미현6 (202301006)
7	김미현7 (202301007)
8	김미현8 (202301008)
9	김미현9 (202301009)

```
select s.s_name, s.s_no, c.c_totaltime
from courselist cl
join student s on cl.sidx=s.sidx
join course c on cl.cidx = c.cidx
where cl.cidx=1
group by s.s_name, s.s_no order by 2
```

- 해당 과목을 수강하는 학생 목록 출력



- 이전에 출결시간이 출력된 테이블 하단에 '학번_날짜_교시'로 이루어진 id 가 부여된 td를 append 해준다.
- 방금 불러온 학번과 이전에 생성된 시간td에 text()를 이용해서 날짜와 교시 정보를 얻는다.
- 반복문 조건은 각 강의시간의 totaltime(한학기 총 강의시간) 만큼 하도록 한다.

3. 각 출결상태 칸(td)에 정보 입력

```
function getEachCheck(cidx){
$.ajax({
type: "post",
url: "${pageContext.request.contextPath}/attendance/getEachCheck.do",
data: {
"cidx": cidx
},
dataType: "json",
success: function(data){
var td_id = "";

$(data).each(function(){
var attend = this.e_attendance;

td_id = "#"+this.s_no + "_" + this.a_date + "_" + this.pe_period;
$(td_id).html(attend);

var attend_style = $(td_id);
if(attend == "결석"){
attend_style.css("color", "red");
}else if(attend == "지각" || attend == "조퇴"){
attend_style.css("color", "orange");
}

});
return;
},
error: function(request, status, error){
alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
}
});
}
```

- 출결 상태에 따른 css 변경

No	학생	08.28(월) 1교시	08.28(월) 2교시	08.29(화) 1교시	09.04(월) 1교시	09.04(월) 2교시	09.05(화) 1교시	09.11(월) 1교시	09.11(월) 2교시
1	김이현 (202301001)	출석	출석	출석	출석	출석	지각	출석	결석
2	김이현2 (202301002)	출석	출석	출석	출석	출석	지각	출석	결석
3	김이현3 (202301003)	출석	출석	출석	출석	출석	지각	출석	결석
4	김이현4 (202301004)	출석	출석	출석	출석	출석	지각	출석	결석
5	김이현5 (202301005)	출석	출석	출석	출석	출석	출석	출석	출석

controller

```
else if(location.equals("/getEachCheck.do")) {
int cidx = Integer.parseInt(request.getParameter("cidx"));

AttendanceDao ad = new AttendanceDao();
List<AttendanceVo> eachlist = ad.getEachCheck(cidx);
int s_no = 0;
String a_date = "";
String a_start = "";
String e_attendance = "";
String str = "";
int pe_period = 0;
for(int i=0; i<eachlist.size(); i++) {
String comma = "";
if (i == eachlist.size() - 1) {
comma = "";
} else {
comma = ",";
}

s_no = eachlist.get(i).getS_no();
a_date = eachlist.get(i).getA_date();
a_date = a_date.substring(5, 7)+a_date.substring(8,10);
a_start = eachlist.get(i).getA_start();
a_start = a_start.substring(0,5);

if(a_start.equals("09:00")) {
pe_period = 1;
}else if(a_start.equals("10:00")) {
pe_period = 2;
}else if(a_start.equals("11:00")) {
pe_period = 3;
}else if(a_start.equals("12:00")) {
pe_period = 4;
}else if(a_start.equals("13:00")) {
pe_period = 5;
}else if(a_start.equals("14:00")) {
pe_period = 6;
}else if(a_start.equals("15:00")) {
pe_period = 7;
}else if(a_start.equals("16:00")) {
pe_period = 8;
}else if(a_start.equals("17:00")) {
pe_period = 9;
}

e_attendance = eachlist.get(i).getE_attendance();

str = str + "{ \"s_no\" : \""+s_no+"\", \"a_date\" : \""+a_date+"\", \"pe_period\" : \""+pe_period+"\", \"e_attendance\" : \""+e_attendance+"\" }"+comma;
}

PrintWriter out = response.getWriter();
out.println("[\""+str+"\"]");
}
```

select s.s_no, a.a_date, a.a_start, ec.e_attendance
from eachcheck ec
join attendance a on a.aidx = ec.aidx
join courselist cl on cl.cidx = a.cidx
join course c on cl.cidx = c.cidx
join student s on s.sidx = cl.sidx
where c.cidx=?

- 해당 과목의 출결상황과 학번, 날짜, 교시를 불러온 후에 컨트롤러에서 원하는 형태로 다듬어서 보낸다.
- a_start(출석테이블에서의 시작 시간이 저장된 컬럼)을 이용해서 해당 교시를 정해준다.

- td_id 변수에 "#학번_날짜_교시"를 대입한 후에 해당 id를 가지고있는 td에 attend 값이 입력되도록 한다.

11 주요 기능 교수 - 출석관리

Ezen_Developer

Ezen_Univ 마이페이지 출석관리 공지사항

[교수]
김광수(20230111)
[정보통신공학과]
logout

출석관리

출석현황 조회
출석 관리

출석관리

강의명: 고급프로그래밍 언어 및 실습 수강 인원: 16

주차	수업일자	수업시간	출석	조퇴	지각	결석	출결처리
1.	08월28일(월)	1교시(09:00:00 ~ 09:50:00)	16	0	0	0	출결조회
1.	08월28일(월)	2교시(10:00:00 ~ 10:50:00)	16	0	0	0	출결조회
1.	08월29일(화)	1교시(09:00:00 ~ 09:50:00)	16	0	0	0	출결조회
2.	09월04일(월)	1교시(09:00:00 ~ 09:50:00)	12	0	0	4	출결조회

Ezen_Univ 마이페이지 출석관리 공지사항

[교수]
김광수(20230111)
[정보통신공학과]
logout

출석관리

출석현황 조회
출석 관리
출석 미달 관리

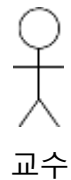
출석관리

강의명: 고급프로그래밍 언어 및 실습

2주차 / 2023-09-04 / 1교시 (09:00:00 ~ 09:50:00) 전체출석 저장

NO	이름	학번	출결구분	처리
1	김어현	202301001	출석	출석 지각 조퇴 결석
2	김어현2	202301002	출석	출석 지각 조퇴 결석
3	김어현3	202301003	출석	출석 지각 조퇴 결석
4	김어현4	202301004	출석	출석 지각 조퇴 결석

- 강의 선택시 해당 강의의 한 학기 수업 일자를 하단에 전부 출력하고 각 수업의 출석/조퇴/지각/결석 카운팅과 출결 버튼이 출력된다.
- 출석처리가 완료된 수업은 출결조회, 출석처리가 완료되지 않은 수업은 출결처리 버튼으로 생성된다.
- 출결처리 화면으로 이동 후 학생별 출결 현황을 선택하여 저장할 수 있다.
- 저장 버튼 클릭시 입력된 출결구분을 배열로 보내서 처리한다.



교수

강의 선택

출결처리
화면 이동

개인별 출결
선택

출결처리

- JavaScript / jQuery
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

11 주요 기능 교수 - 출석관리

Ezen_Developer

I. 강의 선택시 해당강의의 전체 수업일자 출력

출석관리

강의명 고급프로그래밍 언어 및 실습 ~ 출 수업 인원 16

주차	수업일자	수업시간	출석	조퇴	지각	결석	출석지리
1	08월28일(월)	1교시(09:00:00 ~ 09:50:00)	16	0	0	0	출결표지
1	08월28일(월)	2교시(10:00:00 ~ 10:50:00)	16	0	0	0	출결표지
1	08월29일(화)	1교시(09:00:00 ~ 09:50:00)	16	0	0	0	출결표지
2	09월04일(월)	1교시(09:00:00 ~ 09:50:00)	12	0	0	4	출결표지
2	09월04일(월)	2교시(10:00:00 ~ 10:50:00)	12	0	0	4	출결표지
2	09월05일(화)	1교시(09:00:00 ~ 09:50:00)	10	0	4	2	출결표지
3	09월11일(월)						
3	09월11일(월)						
3	09월12일(화)						
4	09월18일(월)	1교시(09:00:00 ~ 09:50:00)	0	0	0	16	출결표지
4	09월18일(월)	2교시(10:00:00 ~ 10:50:00)	0	0	0	16	출결표지

<select>에서 change 이벤트가 일어날 때 마다 Ajax로 하단에 수업일자 테이블 생성

```
function professorManagement(cidx){
$.ajax({
  type: "get",
  url: "=request.getContextPath() ?/attendance/professorManagement.do?cidx=" + cidx,
  dataType: "json",
  success: function(data){
    var totalcnt = parseInt($('#totalSeount').val(), 10);
    $('#mytable').empty();
    var str = "<table><thead><tr><td style='width:50px'>주차</td><td style='width:250px'>수업일자</td><td style='width:400px'>수업시간</td><td style='width:50px'>출석</td><td style='width:50px'>조퇴</td><td style='width:50px'>지각</td><td style='width:50px'>결석</td><td style='width:50px'>출석지리</td></tr></thead><tbody>";
    $.each(data, function(i) {
      var cnt = 0;
      str += "<tr><td>"+data[i].w_weeks+"</td><td>"+data[i].wk+"</td><td>"+data[i].ct_weeks+"</td><td>"+data[i].pe_period+"</td><td>"+data[i].pe_start+" ~ "+data[i].pe_end+"</td><td>"+data[i].att+"</td><td>"+data[i].early+"</td><td>"+data[i].late+"</td><td>"+data[i].absent+"</td><td>"+data[i].absent+"</td></tr>";
      cnt += parseInt(data[i].att, 10) + parseInt(data[i].early, 10) + parseInt(data[i].late, 10) + parseInt(data[i].absent, 10);
      if(cnt==totalcnt){
        str += "<td><button type='button' class='btn' style='background-color:#F2F2F2;'>출결표지</button></td></tr>";
      }else{
        str += "<td><button type='button' class='btn' style='background-color:#0067B3; color:white; border:1px solid #000080;'>출결표지</button></td></tr>";
      }
    });
    str += "</tbody></table>";
    $('#mytable').append(str);
  },
  error: function(request, status, error){
    alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
  }
});
return;
}
```

```
CREATE PROCEDURE professorManagement(IN selected_cidx int, IN syear YEAR, IN stm INT )
BEGIN
DECLARE i int DEFAULT selected_cidx;
DECLARE sy year DEFAULT syear;
DECLARE st int DEFAULT stm;

select a.cidx,a.w_week,a.ct_week ,a.wk,a.pe_period,p.pe_start, p.pe_end,
ifnull(tt.att,0) as att, ifnull(tt.early,0) as early, ifnull(tt.late,0) as late,
ifnull(tt.absent,0) as absent
from (select w.w_week, ct.cidx, w.widx, ct.ct_week,
date_format((case
when ct.ct_week='월' then date_add(w.w_start, interval 0 day)
when ct.ct_week='화' then date_add(w.w_start, interval 1 day)
when ct.ct_week='수' then date_add(w.w_start, interval 2 day)
when ct.ct_week='목' then date_add(w.w_start, interval 3 day)
when ct.ct_week='금' then date_add(w.w_start, interval 4 day)
end),'%m월%d일') as wk, ct.pe_period
from weektb w, coursetime ct
where ct.cidx=i and w.startyear=2023 and w.starttm=2
order by 1, 2, 3)a
join period p on a.pe_period=p.pe_period
left join (SELECT
SUM(CASE WHEN e.e_attendance = '출석' THEN 1 ELSE 0 END) AS 'att',
SUM(CASE WHEN e.e_attendance = '지각' THEN 1 ELSE 0 END) AS 'late',
SUM(CASE WHEN e.e_attendance = '조퇴' THEN 1 ELSE 0 END) AS 'early',
SUM(CASE WHEN e.e_attendance = '결석' THEN 1 ELSE 0 END) AS 'absent',
a.cidx, a.a_start,a.widx,date_format(a.a_date,'%m월%d일') as dates
FROM eachcheck e
JOIN (
SELECT aidx, clidx,a_start, cidx,widx, a_date
FROM attendance
order by aidx
) a ON e.aidx = a.aidx group by a.cidx, a.a_start,a.widx,a.a_date) tt
on a.cidx=tt.cidx and p.pe_start=tt.a_start and tt.widx=a.widx and tt.dates= a.wk
order by 1, 2, 3;
```

Call professorManagement(?,?,?)

- Stored Procedure 생성 후 호출

11 주요 기능 교수 - 출석관리

Ezen_Developer

2. 출결구분 입력

출석관리

관리자 | 고급프로그램입력 | 2주차 / 2023-09-04 / 1교시 (09:00:00~09:50:00)

NO	이름	학번	출결구분	처리	비고
1	김민준	202301001	<input type="radio"/> 출석 <input checked="" type="radio"/> 결석 <input type="radio"/> 지각 <input type="radio"/> 조퇴 <input type="radio"/> 결석	전체출석	
2	김민준2	202301002	<input type="radio"/> 출석 <input checked="" type="radio"/> 결석 <input type="radio"/> 지각 <input type="radio"/> 조퇴 <input type="radio"/> 결석		
3	김민준3	202301003	<input type="radio"/> 출석 <input checked="" type="radio"/> 결석 <input type="radio"/> 지각 <input type="radio"/> 조퇴 <input type="radio"/> 결석		
4	김민준4	202301004	<input type="radio"/> 출석 <input checked="" type="radio"/> 결석 <input type="radio"/> 지각 <input type="radio"/> 조퇴 <input type="radio"/> 결석		

```
$(document).ready(function(){
    $('#allatt').on('click',function(){
        var size = $('#input:checkbox[name="student"]').length;
        for(var i=1; i<size; i++){
            var ra = "#att"+i;
            var cs = "input[name='attendvalue'+i+'\\'][value='출석']";
            var cs1 = "input[name='attendvalue'+i+'\\'][value='지각']";
            var cs2 = "input[name='attendvalue'+i+'\\'][value='조퇴']";
            var cs3 = "input[name='attendvalue'+i+'\\'][value='결석']";
            var tex = "#value"+i;
            $(ra).prop('checked', true);
            var val = "input[name='attendvalue'+i+'\\']";
            var val1 = $(val).val();
            var ne = $(cs).next();
            var ne1 = $(cs1).next();
            var ne2 = $(cs2).next();
            var ne3 = $(cs3).next();
            ne.css('background-color', '#459860');
            ne.css('border', '1px solid #459860');
            ne.css('color', 'white');
            $(tex).val('출석');
            ne1.css('background-color', 'white');
            ne1.css('border', '1px solid #555555');
            ne1.css('color', '#555555');

            ne2.css('background-color', 'white');
            ne2.css('border', '1px solid #555555');
            ne2.css('color', '#555555');

            ne3.css('background-color', 'white');
            ne3.css('border', '1px solid #555555');
            ne3.css('color', '#555555');
        }
    });
});
```

전체출석 클릭시 전체
출결구분 및 처리
'출석' 으로 변경

```
$('#input[name="attendvalue${i.count}"]').each(function() {
    var value = $(this).val();
    var checked = $(this).prop('checked');

    var $label = $(this).next();

    if (checked && value === '출석') {
        $label.css('background-color', '#459860');
        $label.css('border', '1px solid #459860');
        $label.css('color', 'white');
    } else if (checked && (value === '지각' || value === '조퇴')) {
        $label.css('background-color', '#E3C22A');
        $label.css('border', '1px solid #E3C22A');
        $label.css('color', 'white');
    } else if (checked && value === '결석') {
        $label.css('background-color', '#D95321');
        $label.css('border', '1px solid #D95321');
        $label.css('color', 'white');
    } else {
        $label.css('background-color', 'white');
        $label.css('border', '1px solid #555555');
        $label.css('color', '#555555');
    }
});
```

처리된 출결을 조회할 때
저장된 값에 따른 버튼 CSS

페이지 로드 시 체크된
상태를 감지하여 CSS
변경

```
$('#input[name="attendvalue${i.count}"]').change(function() {
    $('#input[name="attendvalue${i.count}"]').each(function() {
        var value = $(this).val();
        var checked = $(this).prop('checked');

        var $label = $(this).next();

        if (checked && value === '출석') {
            $label.css('background-color', '#459860');
            $label.css('border', '1px solid #459860');
            $label.css('color', 'white');
        } else if (checked && (value === '지각' || value === '조퇴')) {
            $label.css('background-color', '#E3C22A');
            $label.css('border', '1px solid #E3C22A');
            $label.css('color', 'white');
        } else if (checked && value === '결석') {
            $label.css('background-color', '#D95321');
            $label.css('border', '1px solid #D95321');
            $label.css('color', 'white');
        } else {
            $label.css('background-color', 'white');
            $label.css('border', '1px solid #555555');
            $label.css('color', '#555555');
        }
    });
});
```

출결처리 버튼(라디오)의 체크
상태가 변경될 때의 이벤트 핸들러

3. 출결처리 (jsp)

출석관리

관리자 | 고급프로그램 빌드 및 실행

2주차 / 2023-09-04 / 1교시 (09:00:00~09:50:00) [전체출석] [재입] [비고]

NO	이름	학번	출결구분	처리
1	김이현	202301001	출석	출석 지각 조퇴 결석
2	김이현2	202301002	지각	출석 지각 조퇴 결석
3	김이현3	202301003	조퇴	출석 지각 조퇴 결석
4	김이현4	202301004	결석	출석 지각 조퇴 결석

```
$(document).ready(function(){
    $("#attbtn").on("click",function(){
        var fm = document.attendancefrm;
        var ctidx = fm.ctidx.value;
        var a_date = fm.a_date.value;
        var pe_start = fm.pe_start.value;
        var pe_end = fm.pe_end.value;
        var cidx = fm.cidx.value;
        var widx = fm.widx.value;
        var period = fm.period.value;
        var checkbox = $("input[name='student']:checked");
        var jsonArray = new Array();
```

저장 버튼 클릭

```
1 checkbox.each(function (i) {
    var jsonObj = new Object();
    var tr = checkbox.parent().parent().eq(i);
    var td = tr.children();
    var attendvalue = td.eq(4).children().val();

    jsonObj.clidx = $(this).val();
    jsonObj.attendvalue = attendvalue;
    jsonObj = JSON.stringify(jsonObj);
    jsonArray.push(JSON.parse(jsonObj));

});

let arrays = JSON.stringify(jsonArray);
```

1. JSON 라이브러리를 사용하여 JSONObject 객체를 생성
2. 테이블 왼쪽의 체크박스가 체크된 목록들의 정보를 jsonArray에 push
3. 배열을 json형식으로 변환하여 arrays 변수에 입력

2

```
$.ajax({
    type : "get",
    url : "<%=request.getContextPath()%>/attendance/attendanceAction.do",
    data : {
        ctidx:ctidx,
        a_date:a_date,
        pe_start:pe_start,
        pe_end:pe_end,
        cidx:cidx,
        widx:widx,
        period:period,
        Array:arrays
    },
    dataType : "json",
    success : function(data){
        if(data.value == 0){
            alert("입력오류");
        }else{
            alert("출석이 완료되었습니다.");
            var loc = "<%=request.getContextPath()%>/attendance/attendancePreManagement.do";
            location.href=loc;
        }
    },
    error : function(request, status, error){
        alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
    }
});
return;
```

출결처리에 필요한 정보들을 Ajax로 전달

처리 후 페이지 이동
처리

3. 출결처리 (controller)

```
int ctidx = Integer.parseInt(request.getParameter("ctidx"));
String a_date = request.getParameter("a_date");
String pe_start = request.getParameter("pe_start");
String pe_end = request.getParameter("pe_end");
int cidx = Integer.parseInt(request.getParameter("cidx"));
int widx = Integer.parseInt(request.getParameter("widx"));
int period = Integer.parseInt(request.getParameter("period"));
```

1

```
AttendanceVo av = new AttendanceVo();
av.setCtidx(ctidx);
av.setA_date(a_date);
av.setA_start(pe_start);
av.setA_end(pe_end);
av.setCidx(cidx);
av.setWidx(widx);
av.setPe_period(period);
```

```
String attendValue = request.getParameter("Array");
```

```
JSONParser parser = new JSONParser();
JSONArray jsonArray = null;
```

```
try {
    jsonArray = (JSONArray) parser.parse(attendValue);
} catch (ParseException e) {
    e.printStackTrace();
}
```

2

```
ArrayList<EachCheckVo> list = new ArrayList<>();
```

3

```
for(int i=1; i<jsonArray.size(); i++){
    JSONObject insertData = (JSONObject) jsonArray.get(i);
    int clidx = Integer.parseInt((String) insertData.get("clidx"));
    String attvalue = (String)insertData.get("attendvalue");

    EachCheckVo ecv = new EachCheckVo();
    ecv.setCidx(clidx);
    ecv.setE_attendance(attvalue);
    list.add(ecv);
}
```

4

```
AttendanceDao ad = new AttendanceDao();
int value = ad.attendanceAction(list, av);
```

5

1. 전달받은 파라미터 중 array를 제외한 기본 출석 정보들은 AttendanceVo에 입력
2. String 타입으로 받은 배열을 JSONArray로 변경
3. array에 담긴 값을 list로 담기 위한 ArrayList<EachCheckVo> list 생성
4. array 크기만큼 for문 안에서 각 값을 EachEcheckVo에 담은 후 list에 add
5. 출결일자 정보가 담긴 av와 학생출결 정보가 담긴 list를 Dao로 전달

11 주요 기능 교수 - 출석관리

Ezen_Developer

3. 출결처리 (Dao & Query)

```
for(int i=0; i<list.size(); i++) {  
    pstmt = conn.prepareStatement(checkcnt);  
    pstmt.setInt(1, av.getCidx());  
    pstmt.setString(2, av.getA_date());  
    pstmt.setInt(3, av.getPe_period());  
    pstmt.setInt(4, list.get(i).getCidx());  
    cnt = pstmt.executeQuery();  
    int counting = 0;  
    int cntaidx = 0;  
    while(cnt.next()) {  
        counting = cnt.getInt("cnt");  
        cntaidx = cnt.getInt("a.aidx");  
    }  
}
```

select count(*) cnt, a.aidx
from eachcheck ec
join attendance a on ec.aidx = a.aidx
join coursetime ct on a.ctidx=ct.ctidx
where a.cidx=? and a.a_date=? and ct.pe_period=? and ec.cidx=?
group by a.aidx

해당 출석일자에 대해서 학생별로 cidx를
조회하여
데이터가 없으면 insert, 있으면 update 실행

```
if(counting == 0) {  
    pstmt = conn.prepareStatement(insertAtt);  
    pstmt.setInt(1, av.getCidx());  
    pstmt.setString(2, av.getA_date());  
    pstmt.setString(3, av.getA_start());  
    pstmt.setString(4, av.getA_end());  
    pstmt.setInt(5, list.get(i).getCidx());  
    pstmt.setInt(6, av.getCidx());  
    pstmt.setInt(7, av.getWidx());  
    pstmt.executeUpdate();  
  
    pstmt = conn.prepareStatement(maxAidx);  
    rs = pstmt.executeQuery();  
    int aidx = 0;  
    while(rs.next()) {  
        aidx = rs.getInt("maxval");  
    }  
  
    pstmt = conn.prepareStatement(insertEach);  
    pstmt.setInt(1, list.get(i).getCidx());  
    pstmt.setInt(2, aidx);  
    pstmt.setString(3, list.get(i).getE_attendance());  
    value = pstmt.executeUpdate();  
} else {  
    pstmt = conn.prepareStatement(updateatt);  
    pstmt.setString(1, list.get(i).getE_attendance());  
    pstmt.setInt(2, cntaidx);  
    value = pstmt.executeUpdate();  
}
```

insert into attendance(ctidx, a_date, a_start, a_end, a_skipyn, cidx, cidx, widx)
values(?,?,?,?,?, 'N', ?, ?, ?)

해당 출석정보가 담긴 attendance 데이터 입력

select max(aidx) as maxval from attendance

방금 입력된 데이터의 인덱스 조회 후 그
인덱스 값을 포함한 출석정보 eachcheck에 입력

insert into eachcheck(cidx, aidx, e_attendance) values(?, ?, ?)

update eachcheck set e_attendance=? where aidx=?

존재하는 출석정보에 대한 attendance 값을
입력된 정보로 변경

```
}  
conn.commit();
```

11 주요 기능 교수 - 후보강 공지

Ezen_Developer

Ezen_Univ 마이페이지 줄식관리 공지사항

[교수]
김광수(20230111)
[정보통신공학과]
logout

공지사항

후보강 공지

후보강 공지

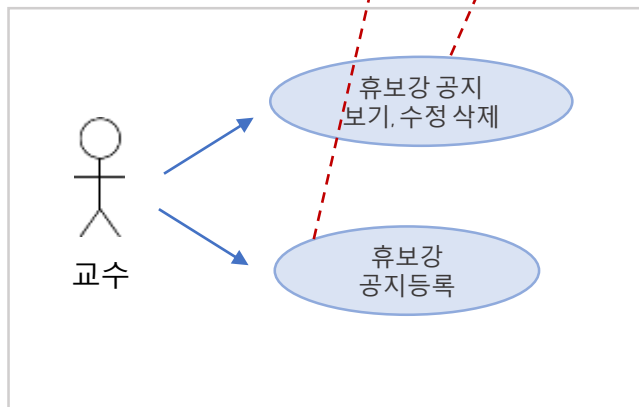
No	제목	작성일	조회수
1236	[휴강][2023-12-12][고급프로그래밍 언어 및 실습]	2023-12-11	0
1235	[휴강][2023-11-17][고급프로그래밍 언어 및 실습]	2023-12-01	5
1234	[휴강][2023-12-11][고급프로그래밍 언어 및 실습]	2023-11-30	4
1233	[휴강][2023-12-18][고급프로그래밍 언어 및 실습]	2023-11-27	19
1232	[휴강][2023-11-14][고급프로그래밍 언어 및 실습]	2023-11-27	53
1228	[휴강][2023-11-20][고급프로그래밍 언어 및 실습]	2023-11-24	39
1226	[휴강][2023-11-14][고급프로그래밍 언어 및 실습]	2023-11-24	23
1225	[휴강][2023-12-04][고급프로그래밍 언어 및 실습]	2023-11-24	41
1224	[휴강][2023-11-27][고급프로그래밍 언어 및 실습]	2023-11-24	24
1223	[휴강][2023-12-04][고급프로그래밍 언어 및 실습]	2023-11-24	15

글쓰기

강의를 선택해주세요

1 2 3 4 5 6 7 8 9 10 >

- 공지사항 클릭시 공지목록 출력
- 공지제목 클릭시 해당 글 보기 목록으로 이동
- 페이징 구현
- Ajax를 사용하여 강의 selectBox에 교수가 강의하는 강의만 출력 후 검색하면 해당 강의에 대한 공지만 출력.
- 새로운 글(3일 미만)인 게시글에 대해서 제목 앞 new 표시 출력



- JavaScript
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

11 주요 기능 교수 - 휴보강 등록

Ezen_Developer

휴보강 공지등록

제목: [선택] 카테고리: 고급프로그래밍 언어 및 실습 날짜: 2023-12-18

오늘 일거요

목록 등록

```
insert into notice(n_skipdate,  
n_category, n_contents, n_count,  
pidx, cidx, n_delyn)\r\nVALUES(?,?,?,0,?,?,?, 'N');
```

- 교수가 가지고 있는 cidx에 대해서만 출력되게함.

```
<select name="coursetype">  
  <c:forEach var="cv" items="${courselist}">  
    <option value="${cv.cidx}">${cv.c_name}</option>  
  </c:forEach>  
</select>
```

- White-space: pre를 써 글쓰기에서 띄어쓰기, 줄바꿈을 자유롭게 함
- Replace 를 통해 글내용에서 <script> 등 을 컴파일하지 못하도록 제한해 보안을 강화함

```
<td class="text-box">  
  <p style="white-space: pre;">${fn:replace(fn:replace(nv.n_contents, '<', '&lt;'), '>', '&gt;')}</p>  
</td>
```

11 주요 기능 교수 - 휴보강 등록

Ezen Developer

후보강 공지사항

제목 :

[휴강][2023-12-18][고급프로그래밍 언어 및 실습]

초회수 : 3

작성일자 : 2023-12-11 15:24:16

작성자 : 김광우

오늘 일해요

목록

수정

삭제

```
update notice set
n_delyn = 'Y'
where nidx = ?
```

공지 수첩

제목

공간

고급프로그래밍 언어 및 실습

2023-12-18

오늘 일거요

- 수정을 누르면 그 게시글에 대한 저장
됐던 입력값들이 불러와지고 게시글을
수정할 수 있게 함

```
update notice set
n_skipdate = ?
n_category = ?
n_contents = ?
cidx = ?
where nidx = ? and pidx = ?"
```

```
<form name="frm">  
    <input type="hidden" name="nidx" value="<%=nv.getNidx()%>">  
    <table class="noticeWrite">  
  
        <tr class="subject">  
            <td><strong>제목</strong>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>  
            <td>  
  
                <select name="noticeType">  
                    <option value="<%=nv.getN_category()%>"><%=nv.getN_category()%></option>  
                    <option value="유급">유급</option>  
                    <option value="보급">보급</option>  
                </select>  
            </td>  
            <td>  
                <select name="courseType">  
                    <c:forEach var="cv" items="${courselist}">  
                        <option value="<${cv.cid}>">${cv.c_name}</option>  
                    </c:forEach>  
                </select>  
            </td>  
            <td>  
                <input type="date" name="when" value="<%=nv.getN_skipdate()%>">  
            </td>  
        </tr>  
        <tr>  
            <td>  
            </td>  
        </tr>  
    </table>  
  
    <td>  
  
        <textarea name="contents" class="text-area"><%=nv.getN_contents() %></textarea>  
    </td>  
</tr>
```

controller

```
String subject = request.getParameter("subject");
if (subject == null) subject = "0";
String page = request.getParameter("page");
if (page == null) page = "1";
```

```
String sub = request.getParameter("subject");
request.setAttribute("sub", sub);
```

```
SearchCriteria scri = new SearchCriteria();
scri.setPage(Integer.parseInt(page));
```

```
int cidx = Integer.parseInt(subject);
scri.setSubject(cidx);
```

```
PageMaker pm = new PageMaker();
pm.setScri(scri);
```

```
int pidx = ((Integer)(session.getAttribute("pidx"))).intValue();
```

```
NoticeDao nd = new NoticeDao();
ArrayList<CourseVo> courselist = nd.courselist_p(pidx);
```

```
ArrayList<NoticeVo> alist = nd.getList_p(pidx, scri);
```

```
int cnt = nd.noticeTotalCount_p(pidx, scri);
pm.setTotalCount(cnt);
```

```
request.setAttribute("pm", pm);
request.setAttribute("alist", alist);
request.setAttribute("courselist", courselist);
```

```
String path = "/notice/noticelist_p.jsp";
//화면용도의 주소는 포워드로 토스해서 해당 컨트롤러로 보낸다
RequestDispatcher rd = request.getRequestDispatcher(path);
rd.forward(request, response);
```

- 페이징 구현을 위한 pm, 공지 목록 출력을 위한 alist, 과목 검색을 위한 courselist를 setAttribute.
- 이후 Jstl 을 이용해 jsp파일에 변수를 가져와 사용

jsp

```
<td class="sub">
  <select class="searchsubject" name="subject" id="subject">
    <option value="0">검색어를 선택해주세요</option>
    <c:forEach var="cv" items="${courselist}">
      <option value="${cv.cidx}" <c:if test="${subb == cv.cidx}">selected</c:if>${cv.c_name}</option>
    </c:forEach>
  </select>
</td>

<c:forEach var="nv" items="${alist}">
  <tr>
    <td>
      ${nv.nidx}
    </td>
    <td style="text-align: left; padding-left: 20px" class="subject">
      <a href="${pageContext.request.contextPath}/notice/noticeContents_p.do?nidx=${nv.nidx}">
        <c:if test="${nv.n_dday eq 'true'}>
          <mark><span style="color: orange; font-weight: bold;">new</span></mark>
        </c:if>
        ${nv.n_subject}
      </a>
      <c:if test="${nv.prev == true}">
        <a href="${pageContext.request.contextPath}/notice/noticelist_p.do?page=${pm.startPage-1}${(param)}">
          <span class="material-symbols-outlined">
            arrow_back_ios_new
          </span>
        </a>
      </c:if>
    </td>
    <td>
      <c:forEach var="i" begin="${pm.startPage}" end="${pm.endPage}" step="1">
        <c:choose>
          <when test="${i == pm.scri.page}">
            <a href="${pageContext.request.contextPath}/notice/noticelist_p.do?page=${i}${(param)}">div class="active">${i}</div></a> <endop>
          </c:when>
          <otherwise>
            <a href="${pageContext.request.contextPath}/notice/noticelist_p.do?page=${i}${(param)}">div class="normal">${i}</div></a> <endop>
          </c:otherwise>
        </c:choose>
      </c:forEach>
    </td>
    <td>
      <c:if test="${pm.next == true}&&pm.endPage>0">
        <a href="${pageContext.request.contextPath}/notice/noticelist_p.do?page=${pm.endPage+1}${(param)}">
          <span class="material-symbols-outlined">
            arrow_forward_ios
          </span>
        </a>
      </c:if>
    </td>
  </tr>
</c:forEach>
</td>
</tr>
```

11 주요 기능 교수 - 휴보강 목록

Ezen_Developer

Ezen_Univ

마이페이지

출석관리

공지사항

[교수]
김광수(20230111)
[정보통신공학과]

logout

공지사항

휴보강 공지

휴보강 공지

No	제목	작성일	조회수
1235	[휴강][2023-11-17][고급프로그래밍 언어 및 실습]	2023-12-01	5
1234	[휴강][2023-12-11][고급프로그래밍 언어 및 실습]	2023-11-30	4
1233	[휴강][2023-12-18][고급프로그래밍 언어 및 실습]	2023-11-27	19
1232	[휴강][2023-11-14][고급프로그래밍 언어 및 실습]	2023-11-27	53
1228	[휴강][2023-11-20][고급프로그래밍 언어 및 실습]	2023-11-24	39
1226	[휴강][2023-11-14][고급프로그래밍 언어 및 실습]	2023-11-24	23
1225	[휴강][2023-12-04][고급프로그래밍 언어 및 실습]	2023-11-24	41
1224	[휴강][2023-11-27][고급프로그래밍 언어 및 실습]	2023-11-24	24
1223	[휴강][2023-12-04][고급프로그래밍 언어 및 실습]	2023-11-24	15
1222	[휴강][2023-11-20][고급프로그래밍 언어 및 실습]	2023-11-21	149

관리하기

사물인터넷통신
강의를 선택해주세요
고급프로그래밍 언어 및 실습
사물인터넷통신
테스트강의

```
<td class="sub">
<select class="searchsubject" name="subject" id="subject">
<option value="0">강의를 선택해주세요</option>
<c:forEach var="cv" items="${courselist}">
<option value="${cv.cid}" <c:if test="${subb == cv.cid}">selected</c:if>>${cv.c_name}</option>
</c:forEach>
</select>
</td>
```

- 교수가 가지고 있는 cid에 대해서만 출력되게함.

- 검색기능 구현
- Str을 sql문 where 조건에 넣어 해당 cid만 공지목록에 출력 되게함

- 페이징 구현

```
private void calcData() {
//1.기본적으로 1에서 10까지 나타내게 설정
endPage = (int)(Math.ceil(scri.getPage()/(double)displayPageNum)*displayPageNum);
//2.endPage를 설정했으면 시작페이지도 설정
startPage = (endPage-displayPageNum)+1;
//3.현재 페이지 값을 받는다
int tempEndPage = (int)Math.ceil(totalCount/(double)scri.getPerPageNum());
//4.설정endPage와 실제endPage 비교한다
if (endPage > tempEndPage) {
endPage = tempEndPage;
}
//5.이전다음버튼 문구
prev = (startPage == 1 ? false:true);
next = (endPage*scri.getPerPageNum() >= totalCount ? false:true);
}
```

```
.pagination .act{
font-weight: bold;
color: #0078ff;
border-radius: 5px;
border: 2px solid #0078ff;
padding: 5px 7px;
}

.pagination a {
display: block;
font-size: 15px;
text-decoration: none;

padding: 5px 7px;
color: #ccc;
display: inline-block;
}
```

- 현재페이지 위치에 따른 css 변경

```
String str = "";
if (scri.getSubject() != 0) {
str = " and B.cid=" + scri.getSubject() + "\r\n";
}
String SQL = "SELECT nidx,CONCAT('[' + A.n_category, ']', '[' + A.n_skipdate, ']', '[' + B.c_name, '']) as subject,A.n_writeday,A.n_count\r\n"
+ "FROM notice A\r\n"
+ "join course B\r\n"
+ "on A.cid = B.cid\r\n"
+ "where A.pidx = ? and A.n_delyn = 'N'\r\n"
+ str
+ "order by nidx desc limit ?,?";
```

11 주요 기능 교수 - 휴보강 목록

Ezen_Developer

- 공지등록일이 3일 미만인 경우 제목 앞에 new 표시

```
NoticeVo nv = new NoticeVo();
nv.setNidx(rs.getInt("nidx"));
nv.setN_subject(rs.getString("subject"));
nv.setN_writeday(rs.getString("A.n_writeday"));
nv.setN_count(rs.getInt("A.n_count"));

int wyear = Integer.parseInt(nv.getN_writeday().substring(8,4));
int wmonth = Integer.parseInt(nv.getN_writeday().substring(5,7));
int wday = Integer.parseInt(nv.getN_writeday().substring(8,10));
LocalDate date1 = LocalDate.of(wyear, wmonth, wday);

LocalDate currentDate = LocalDate.now();

long daysDifference = ChronoUnit.DAYS.between(date1, currentDate);

if(Path.of(daysDifference)<3) {
    nv.setN_dday(true);
}else {
    nv.setN_dday(false);
}

alist.add(nv);
}

<c:if test='${nv.n_dday eq 'true'}'>
<mark><span style="color: orange; font-weight: bold;">new</span></mark>
</c:if>
${nv.n_subject}
```

제목	작성일	조회수
new [휴강][2023-12-12][고급프로그래밍 언어 및 실습]	2023-12-11	0
[휴강][2023-11-17][고급프로그래밍 언어 및 실습]	2023-12-01	5
[휴강][2023-12-11][고급프로그래밍 언어 및 실습]	2023-11-30	4
[휴강][2023-12-18][고급프로그래밍 언어 및 실습]	2023-11-27	19
[휴강][2023-11-14][고급프로그래밍 언어 및 실습]	2023-11-27	53
[휴강][2023-11-20][고급프로그래밍 언어 및 실습]	2023-11-24	39
[휴강][2023-11-14][고급프로그래밍 언어 및 실습]	2023-11-24	23
[휴강][2023-12-04][고급프로그래밍 언어 및 실습]	2023-11-24	41
[휴강][2023-11-27][고급프로그래밍 언어 및 실습]	2023-11-24	24
[휴강][2023-12-04][고급프로그래밍 언어 및 실습]	2023-11-24	15

```
select DISTINCT
nidx,CONCAT('[',B.n_category,'],['',B.n
_skipdate,'],['',A.c_name,']')as
subject,B.n_writeday,B.n_count\r\n
from course A\r\n
join notice B\r\n
on A.cidx = B.cidx\r\n
join courselist C\r\n
on A.cidx = C.cidx\r\n
where C.sidx=? and B.n_delyn='N'
order by nidx desc limit ?,?";
```

- 제목에 휴강or보강, 해당날짜, 과목명이 출력

```
SELECT NOW()

update Notice set\r\n
n_count = n_count+1\r\n
where nidx = ?";
```

- 작성일과 조회수 출력

controller

```
else if(location.equals("noticeDelete.do")) {
    HttpSession session = request.getSession();
    PrintWriter out = response.getWriter();

    String nidxx = request.getParameter("nidxx");
    int pidxx = ((Integer)(session.getAttribute("pidxx"))).intValue();
    int nidxx_int = Integer.parseInt(nidxx);
    NoticeDao nd = new NoticeDao();
    int value=0;

    NoticeVo nv = nd.noticeSelectOne(nidxx_int);
    value = nd.noticeDelete(nidxx_int);
    request.setAttribute("nv", value);

    if(pidxx != nv.getPidxx()) {
        out.println("<script>alert('권한이 없습니다.');
```

```
else if(location.equals("noticeModify.do")) {
    HttpSession session = request.getSession();
    PrintWriter out = response.getWriter();

    if(session.getAttribute("pidxx")==null) {
        String path = request.getContextPath();
        out.println("<script>alert('로그인이 필요합니다');
```

- 교수여야 글쓰기 페이지에 접근할 수 있도록 설정
- 교수더라도 글쓴이가 아니면 수정,삭제 페이지에 해당 경로에 어떤 방식으로든 접근하지 못하도록 설정

11 주요 기능 학생 - 출석현황 조회

Ezen_Developer

Ezen_Univ 마이페이지 출석관리 출석사항

(학생)
김이영(202301001)
[정보통신공학과]

logout

출석관리

출석현황 조회

년도 : 2023 학기 : 2

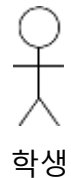
출석현황조회

NO	이수구분	강의명	학점	교수	강의부	시간표	결석률	미달여부
1	전공선택	고급프로그래밍 언어 및 알고	3	김영수	공학관 303실3	월1, 월2, 화1	16.67%	미
2	전공선택	무선통신이론	3	정현수	공학관 404	문4, 목3, 목4	20.83%	미
3	전공필수	사물인터넷통신	3	김영수	공학관 404	월3, 월4, 화2	2.09%	미
4	전공선택	시스템소프트웨어 설계	3	문다희	공학관 102	목1, 목2	0.00%	미
5	전공선택	데이터베이스	3	김영수	공학관 404	문3, 목3	0.00%	미

출석 : 31 지각 : 4 조퇴 : 1 결석 : 6

주차	수업일자	수업시간	출석현황
1	09월 28일	09:00:00 ~ 09:50:00	출석
1	09월 28일	10:00:00 ~ 10:50:00	출석
1	09월 29일	09:00:00 ~ 09:50:00	출석
2	09월 04일	09:00:00 ~ 09:50:00	출석
2	09월 04일	10:00:00 ~ 10:50:00	출석
2	09월 05일	09:00:00 ~ 09:50:00	지각
3	09월 11일	09:00:00 ~ 09:50:00	출석
3	09월 11일	10:00:00 ~ 10:50:00	결석
3	09월 12일	09:00:00 ~ 09:50:00	출석
4	09월 18일	09:00:00 ~ 09:50:00	출석
4	09월 18일	10:00:00 ~ 10:50:00	결석
4	09월 19일	09:00:00 ~ 09:50:00	출석
5	09월 25일	09:00:00 ~ 09:50:00	출석
5	09월 25일	10:00:00 ~ 10:50:00	출석
5	09월 26일	09:00:00 ~ 09:50:00	출석
6	10월 02일	09:00:00 ~ 09:50:00	출석
6	10월 02일	10:00:00 ~ 10:50:00	출석

- 출석현황 조회 클릭시 수강목록 출력
- 수강 목록에 결석률과 미달여부도 함께 출력된다.
 - 결석률은 (결석횟수+지각횟수*0.5+조퇴횟수*0.5)/전체수업시간*100)으로 계산
 - 미달여부는 결석률이 25% 초과할 때 Y로 출력된다.
- 조회하고자 하는 강의 클릭시 하단에 당일 날짜까지의 출결현황이 조회된다.
- 해당 강의 당일 날짜까지의 출석 / 지각 / 조퇴 / 결석 횟수가 카운팅된다.
- Ajax를 사용하여 새로고침 없이 하단에 출결현황이 출력될 수 있도록 한다.



학생

강의 선택

출석현황
보기

- JavaScript / jQuery
 - Event 처리
 - Ajax를 사용한 화면처리
- JSP / SQL
 - MVC 패턴으로 처리

11 주요 기능 학생 - 출석현황 조회

Ezen_Developer

1. 해당 년도 및 학기 수강목록 출력

출석현황조회

년도	2023	학기	2					
NO	이수구분	강의명	학점	교수	강의실	시간표	결석률	미달여부
1	전공선택	고급프로그래밍 언어 및 실습	3	김광수	공학관 전산실3	월1, 월2, 화1	16.67%	N
2	전공선택	무선통신이론	3	정현수	공학관 404	금4, 목3, 목4	20.83%	N

```
<c:forEach var="av" items="${list}" varStatus="i">
...<tr>
...<td></td>
...<td>${av.c_sep}</td>
...<td><button type="button" id="c_name" class="c_name${i.count}" value="${av.cid}" onclick="search_detail('${av.cid}')">${av.c_name}</button></td>
...<td>${av.c_score}</td>
...<td>${av.p_name}</td>
...<td>${av.ct_room}</td>
...<td>${av.c_times}</td>
...<td>${av.abpercent}</td>
...<c:choose>
...<c:when test="${av.abyn eq 'Y'}">
...<td style="font-weight:bold; color:red">${av.abyn}</td>
...</c:when>
...<c:otherwise>
...<td>${av.abyn}</td>
...</c:otherwise>
...</c:choose>
...</tr>
...</script>
...$(document).ready(function(){
...    $("button").click(function(){
...        var c_name = $(this).val();
...        $(this).css("color", "#0067B3");
...        $("button").not("[value='"+c_name+"']").css("color", "black");
...    });
...});
...</script>
</c:forEach>
```

controller

```
if(location.equals("attendanceSituation_s.do"))-{
    HttpSession session = request.getSession();
    PrintWriter out = response.getWriter();
    if(session.getAttribute("sid") == null){
        String path = request.getContextPath();
        out.println("<script>alert('로그인이 필요합니다'); location.href='"+path+"/index.jsp';</script>");
    }
    int sid = ((Integer)(session.getAttribute("sid"))).intValue();
    AttendanceDao ad = new AttendanceDao();
    LocalDate now = LocalDate.now();
    int year = now.getYear();
    int month = now.getMonthValue();
    int semester = 0;
    if(month >= 1 && month <= 7){
        semester = 1;
    }else{
        semester = 2;
    }
    ArrayList<AttendanceVo> slist = ad.attendanceCourseList_s(sid, year, semester);
    request.setAttribute("list", slist);
    request.setAttribute("year", year);
    request.setAttribute("semester", semester);
    String path = "/attendance/attendanceSituation_s.jsp";
    RequestDispatcher rd = request.getRequestDispatcher(path);
    rd.forward(request, response);
    out.println(sid);
}
```

- 해당 학생의 수강과목에 연관되는 정보(이수구분, 학점, 교수, 강의실, 시간표, 결석률, 미달여부 등)를 불러와서 컨트롤러에서 slist에 담고 list로 보냄
-> jquery로 출력

11 주요 기능 학생 - 출석현황 조회

Ezen_Developer

2. 강의 클릭시 출결현황 카운팅과 목록 동시에 출력

NO	이수구분	강의명	학점	교수	강의실	시간표	결석률	비밀여부
1	전공선택	코딩프로그래밍 언어 및 실습	3	김광수	공학관 전산동5	월1, 월2, 화1	16.67%	N

출석 31 | 지각 4 | 출결 1 | 결석 8

주차	수업일자	수업시간	출결현황
1	08월 28일	09:00:00 - 09:50:00	출석
1	08월 28일	10:00:00 - 10:50:00	출석
1	08월 29일	09:00:00 - 09:50:00	출석
2	09월 04일	09:00:00 - 09:50:00	출석
2	09월 04일	10:00:00 - 10:50:00	출석
2	09월 05일	09:00:00 - 09:50:00	지각
3	09월 11일	09:00:00 - 09:50:00	출석
3	09월 11일	10:00:00 - 10:50:00	결석

- onclick 이벤트를 사용하여 강의 클릭시 출결현황 카운팅과 목록 동시에 출력

```
function search_detail(cidx){  
    countAction(cidx);  
    listAction(cidx);  
    return;  
}
```

```
<c:forEach var="av" items="${list}" varStatus="i">  
    <tr>  
        <td></td>  
        <td>${av.c_sep}</td>  
        <td><button type="button" id="c_name" class="c_name${i.count}" value="${av.cidx}" onclick="search_detail('${av.cidx}')">${av.c_name}</button></td>  
        <td>${av.c_score}</td>  
        <td>${av.p_name}</td>  
        <td>${av.ct_room}</td>  
        <td>${av.c_times}</td>  
        <td>${av.abpercent}</td>  
        <c:choose>  
            <c:when test="${av.abym eq 'Y'}">  
                <td style="font-weight:bold; color:red">${av.abym}</td>  
            <c:when>  
                <c:otherwise>  
                <td>${av.abym}</td>  
            </c:otherwise>  
        </c:choose>  
    </tr>  
</script>  
$(document).ready(function(){  
    $("button").click(function(){  
        var c_name= $(this).val();  
        $(this).css('color','0067B3');  
        $("button").not("[value='"+c_name+"']").css('color','black');  
    });  
</script>  
</c:forEach>
```

11 주요 기능 학생 - 출석현황 조회

Ezen_Developer

3. 해당 강의 출석 / 지각 / 조퇴 / 결석 카운팅 출력

출석 31 지각 4 조퇴 1 결석 6

```
select count(*) as attendance
from attendance A
join course C on A.cidx = C.cidx
join Eachcheck B on B.aidx = A.aidx
join courselist cl on cl.clidx = A.cidx
where C.cidx=? and cl.sidx=? and
B.e_attendance='출석'
```

- 해당 학생의 해당 강의 출결현황에서 개별 상태(출석, 지각, 조퇴, 결석) 횟수를 계산한다.

```
function countAction(cidx){
    $.ajax({
        type: "get",
        url: "<%=request.getContextPath()%>/attendance/attendanceCount.do?cidx="+cidx,
        dataType: "json",
        cache: false,
        success: function(data){
            $('#details_first').empty();
            str = 출석 <input type="text" name="attendance" value="+data.attcnt+ disabled/>&nbsp;
            +지각 <input type="text" name="late" value="+data.latecnt+ disabled/>&nbsp;
            +조퇴 <input type="text" name="leave_early" value="+data.leavecnt+ disabled/>&nbsp;
            +결석 <input type="text" name="absent" value="+data.absentcnt+ disabled/>
            $('#details_first').append(str);
        },
        error: function(request, status, error){
            alert("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
        }
    });
    return;
}
```

controller

```
}else if(location.equals("attendanceCount.do")){
    HttpSession session = request.getSession();
    PrintWriter out = response.getWriter();
    if(session.getAttribute("sidx")==null){
        String path = request.getContextPath();
        out.println("<script>alert('로그인이 필요합니다'); location.href='"+path+"/index.jsp';</script>");
    }
    AttendanceDao ad = new AttendanceDao();
    int sidx = ((Integer)(session.getAttribute("sidx"))).intValue();
    int cidx = Integer.parseInt(request.getParameter("cidx"));
    AttendanceVo av = ad.attendanceCount(sidx, cidx);
    AttendanceVo alv = ad.lateCount(sidx, cidx);
    AttendanceVo aleve = ad.leaveCount(sidx, cidx);
    AttendanceVo abv = ad.absentCount(sidx, cidx);
    int attcnt = 0;
    int latecnt = 0;
    int leavecnt = 0;
    int absentcnt = 0;
    attcnt = av.getAttendanceCount();
    latecnt = alv.getLateCount();
    leavecnt = aleve.getLeaveCount();
    absentcnt = abv.getAbsentCount();
    String str = "";
    str = str + "{\\"attcnt\":"+attcnt+",\\"latecnt\":"+latecnt+",\\"leavecnt\":"+leavecnt+",\\"absentcnt\":"+absentcnt+"}";
    out.println(str);
}
```

- 해당 강의의 출석, 지각, 조퇴, 결석 카운팅 값을 불러온 후에 컨트롤러에서 원하는 형태로 다듬어서 보낸다.

- 해당 강의의 출석, 지각, 조퇴, 결석 카운팅 값을 input box 에 넣어서 append준다.

11 주요 기능 학생 - 출석현황 조회

Ezen_Developer

4. 해당 강의 출석현황 출력

주차	수업일자	수업시간	출결현황
1	08월 28일	09:00:00 ~ 09:50:00	출석
1	08월 28일	10:00:00 ~ 10:50:00	출석
1	08월 29일	09:00:00 ~ 09:50:00	출석
2	09월 04일	09:00:00 ~ 09:50:00	출석
2	09월 04일	10:00:00 ~ 10:50:00	출석
2	09월 05일	09:00:00 ~ 09:50:00	지각
3	09월 11일	09:00:00 ~ 09:50:00	출석
3	09월 11일	10:00:00 ~ 10:50:00	결석

```
function listAction(cidx){
    $.ajax({
        type: "get",
        url: "<%=request.getContextPath()%>/attendance/attendanceList.do?cidx="+cidx,
        dataType: "json",
        cache: false,
        success: function(data){
            $('#attendanceList').empty();
            var str2="";
            str2+="
```

- 출결 상태에 따른 css 변경

controller

```
else if(location.equals("attendanceList.do")){
    HttpSession session = request.getSession();
    PrintWriter out = response.getWriter();
    if(session.getAttribute("sidx")==null){
        String path = request.getContextPath();
        out.println("<script>alert('로그인이 필요합니다'); location.href='"+path+"/index.jsp';</script>");
    }
    AttendanceDao ad = new AttendanceDao();
    int sidx = ((Integer)(session.getAttribute("sidx"))).intValue();
    int cidx = Integer.parseInt(request.getParameter("cidx"));
    ArrayList<AttendanceVo> alist = ad.attendanceList(sidx, cidx);
    request.setAttribute("alist", alist);
    int listCnt = alist.size();
    int widex = 0;
    String atdate = "";
    String attime = "";
    String e_attendance = "";
    String str2 = "";
    for(int i = 0; i < listCnt; i++){
        widex = alist.get(i).getWidex();
        atdate = alist.get(i).getAtdate();
        attime = alist.get(i).getAttime();
        e_attendance = alist.get(i).getE_attendance();
        String comma = "";
        if(i == listCnt-1){ // 마지막 행이면 , (comma)가 아닌 공백을 출력함
            comma = " ";
        } else {
            comma = ",";
        }
        str2 = str2 + "<td>"+widex+"</td>"+atdate+"</td>"+attime+"</td>"+e_attendance+"</td>"+comma;
    }
    //for 구분자 대입으로 출력하기
    str2+="
```

- 해당 강의의 주차, 수업일자, 수업시간, 출결현황을 불러온 후에 컨트롤러에서 원하는 형태로 다듬어서 보낸다.

DB 설계시 약어사용

잘못된 Vo구조

과한 욕심으로 인한 요구사항 미충족