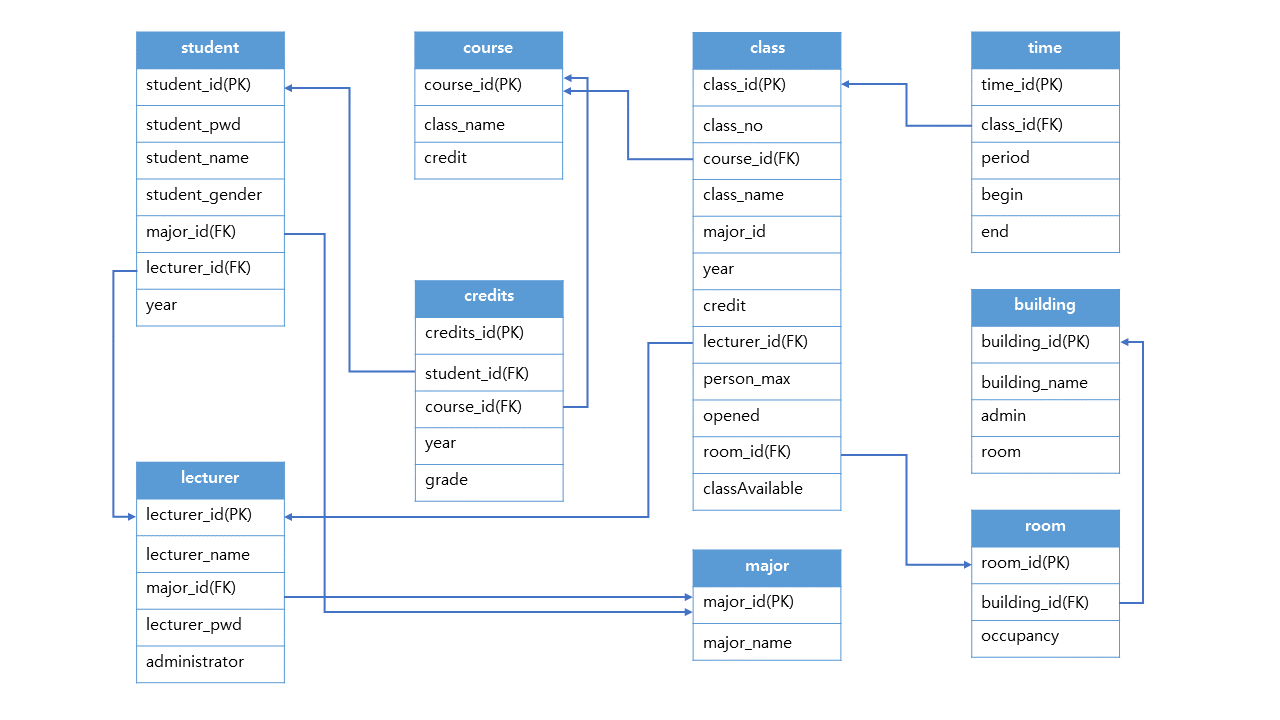
**[데이터베이스시스템 및 응용] PROJECT1 WIKI**

2020029607 김채은

* **Design**

- DB Schema



- 사이트 디자인 구성 계획

메인 페이지에서 로그인과 회원가입이 가능하고, 로그인 후에는 학생 계정과 관리자 계정의 메인 페이지가 달라진다.

학생 계정으로 로그인했을 경우, 메인 페이지에는 로그아웃 버튼과 수강신청 버튼이 있고, 수강신청 버튼을 누르면 과목 검색, 신청 결과 확인, 희망 수업 확인, 시간표 확인이 가능한 페이지로 넘어간다. 과목 검색 페이지에서는 수업을 검색할 수 있고, 검색 결과로 나온 수업을 수강 신청하거나 수강 희망할 수 있다. 신청 결과 페이지에서는 학생이 신청한 과목을 볼 수 있고, 희망 수업 페이지에서는 학생이 희망한 과목을 볼 수 있다. 시간표 페이지에서는 학생이 신청한 과목을 바탕으로 만들어진 시간표를 확인할 수 있다.

관리자 계정으로 로그인했을 경우, 로그아웃, 학생정보, 과목정보, OLAP 버튼이 각각 존재한다. 학생 정보를 누르면 학생들의 시간표, 성적, 지도교수와 재학 상태를 나타내는 표를 볼 수 있다. 이때, 학생의 성적은 과목 이름, 학수 번호, 년도와 함께 볼 수 있다. 과목 정보를 누르면 과목 정보가 뜨고, 해당 과목을 증원하거나 학생의 수강 허용을 반영할 수 있다. 그리고 특정 과목을 설강하거나 폐강하는 것이 가능하다.

* **Implement**

- SQL문 수도 코드

drop database if exists db2020029607;

create database if not exists db2020029607;

use db2020029607;

create table student(student\_id VARCHAR(20), student\_pwd VARCHAR(40), student\_name VARCHAR(20), student\_gender VARCHAR(20), major\_id INT, lecturer\_id INT, year INT, primary key(student\_id));

create table class(class\_id VARCHAR(20), class\_no VARCHAR(20), course\_id VARCHAR(20), class\_name VARCHAR(40), major\_id INT, year INT, credit INT, lecturer\_id INT, person\_max INT, opened INT, room\_id INT, classAvailable INT, primary key (class\_id));

create table course(course\_id VARCHAR(20), class\_name VARCHAR(40), credit INT, primary key(course\_id));

create table credits(credits\_id INT, student\_id INT, course\_id VARCHAR(20), year INT, grade VARCHAR(10), primary key(credits\_id));

create table lecturer(lecturer\_id VARCHAR(20), lecturer\_name VARCHAR(20), lecturer\_pwd VARCHAR(40), major\_id INT, administrator INT, primary key(lecturer\_id));

create table major(major\_id INT, major\_name VARCHAR(40), primary key(major\_id));

create table time(time\_id INT, class\_id VARCHAR(20), period INT, begin VARCHAR(40), end VARCHAR(40), primary key (time\_id));

create table building(building\_id INT, building\_name VARCHAR(20), admin VARCHAR(20), room INT, primary key(building\_id));

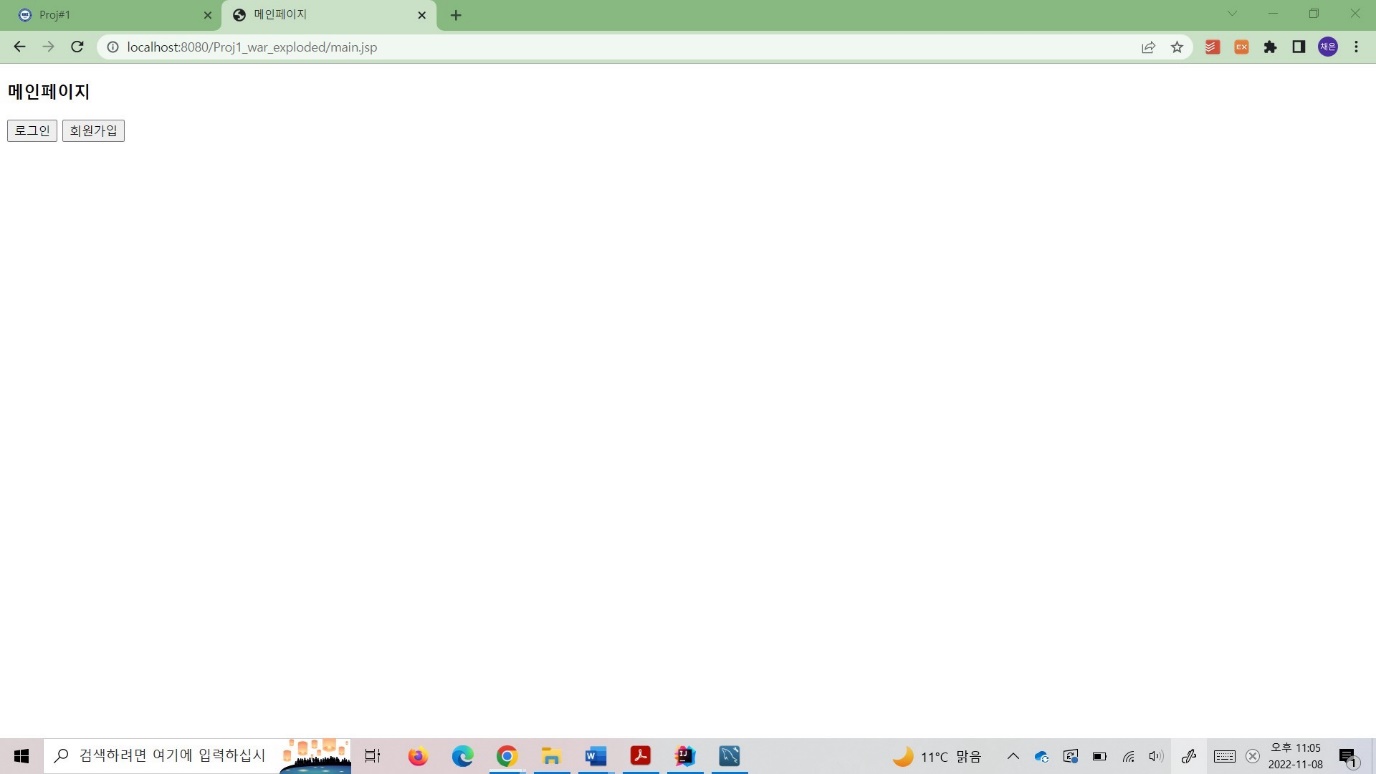
create table room(room\_id INT, building\_id INT, occupancy INT, primary key(room\_id));

- 입력한 SQL csv 파일

Building, class, course, credits, lecturer, major, room, student, time 파일(엑셀 파일로 제출)

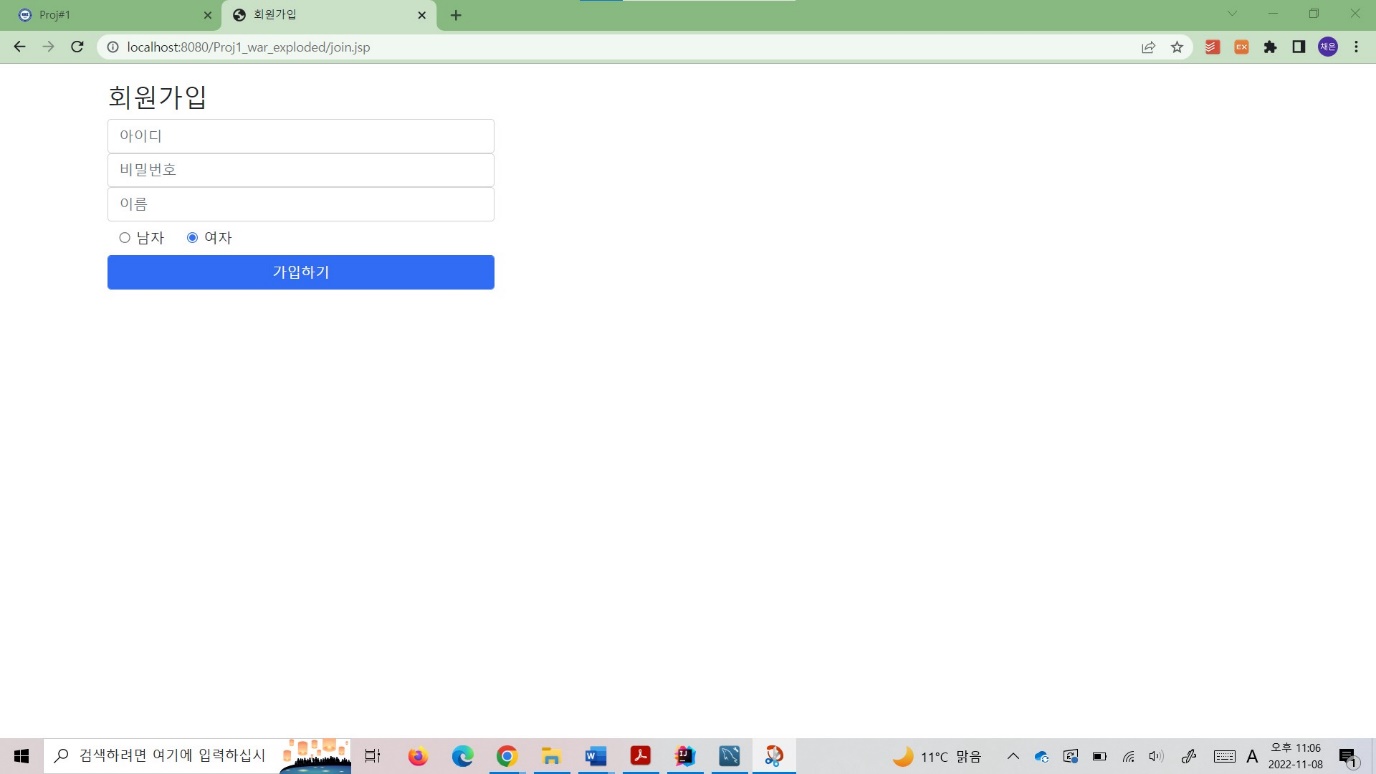
* **Result**

1. 로그인 전 메인페이지



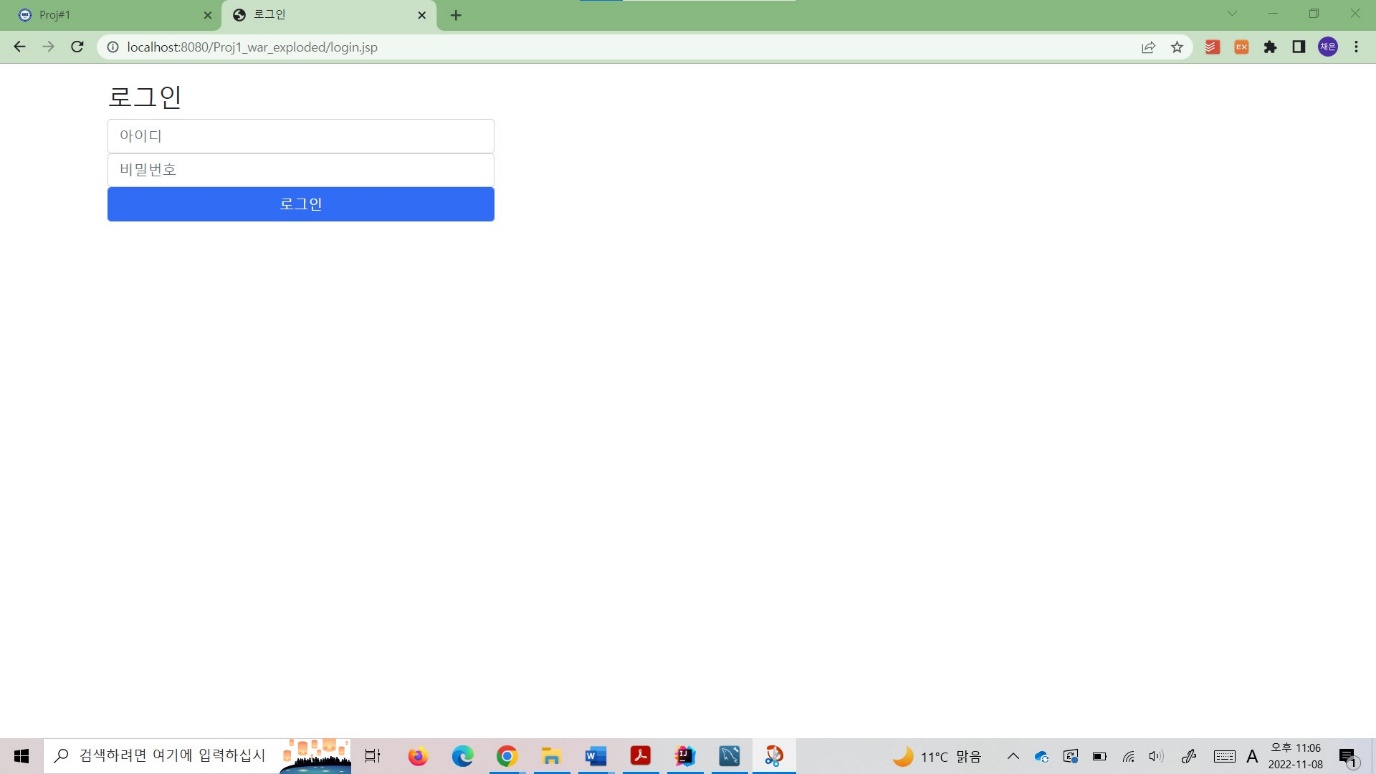
: 로그인 하기 전에는 로그인 버튼과 회원가입 버튼 2개를 배치하여 로그인 혹은 회원가입이 가능하다.

2. 회원가입 페이지



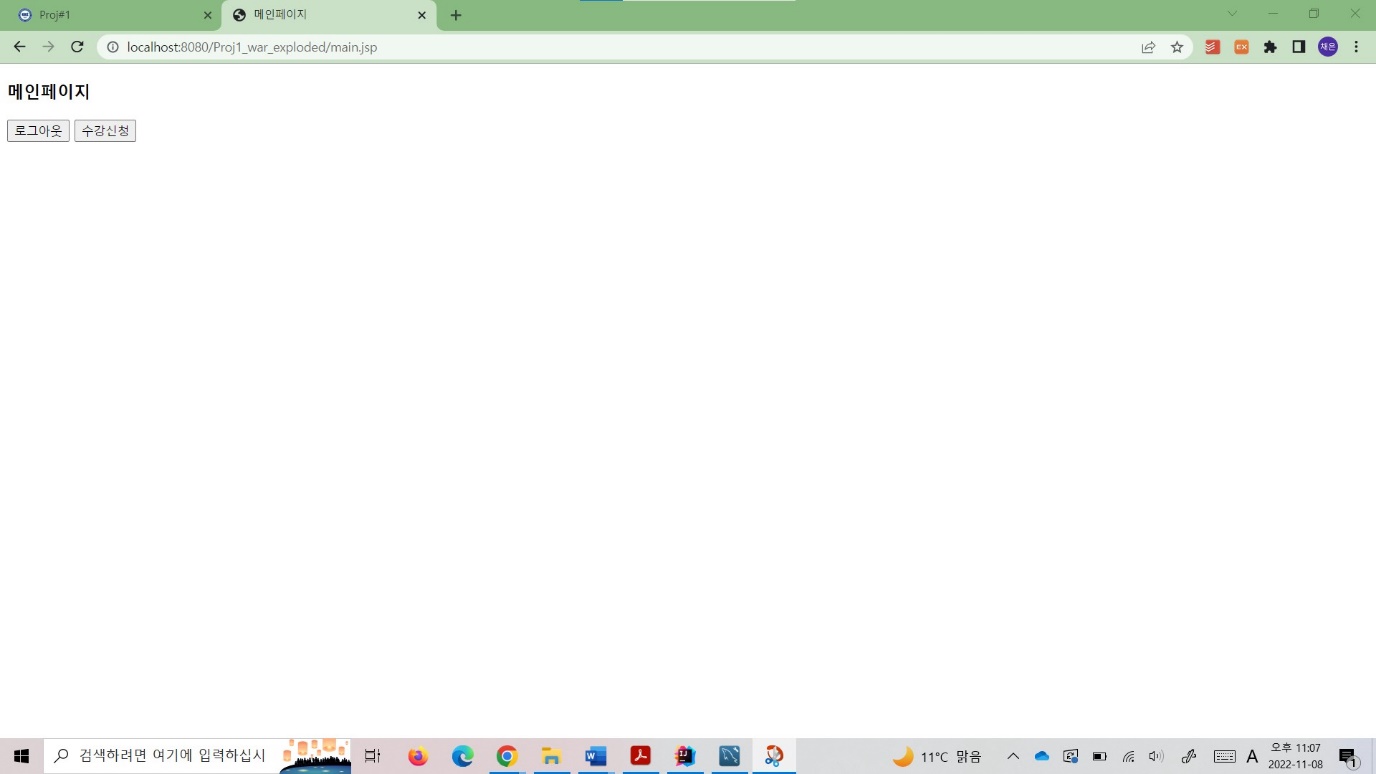
: 아이디, 비밀번호, 이름, 성별을 입력하면 가입하기 버튼을 눌러 회원가입을 진행할 수 있다. 회원가입을 하면 무조건 학생 계정으로 회원가입 되도록 만들었고, 관리자 계정은 따로 회원가입 할 수 없다.

3. 로그인 페이지



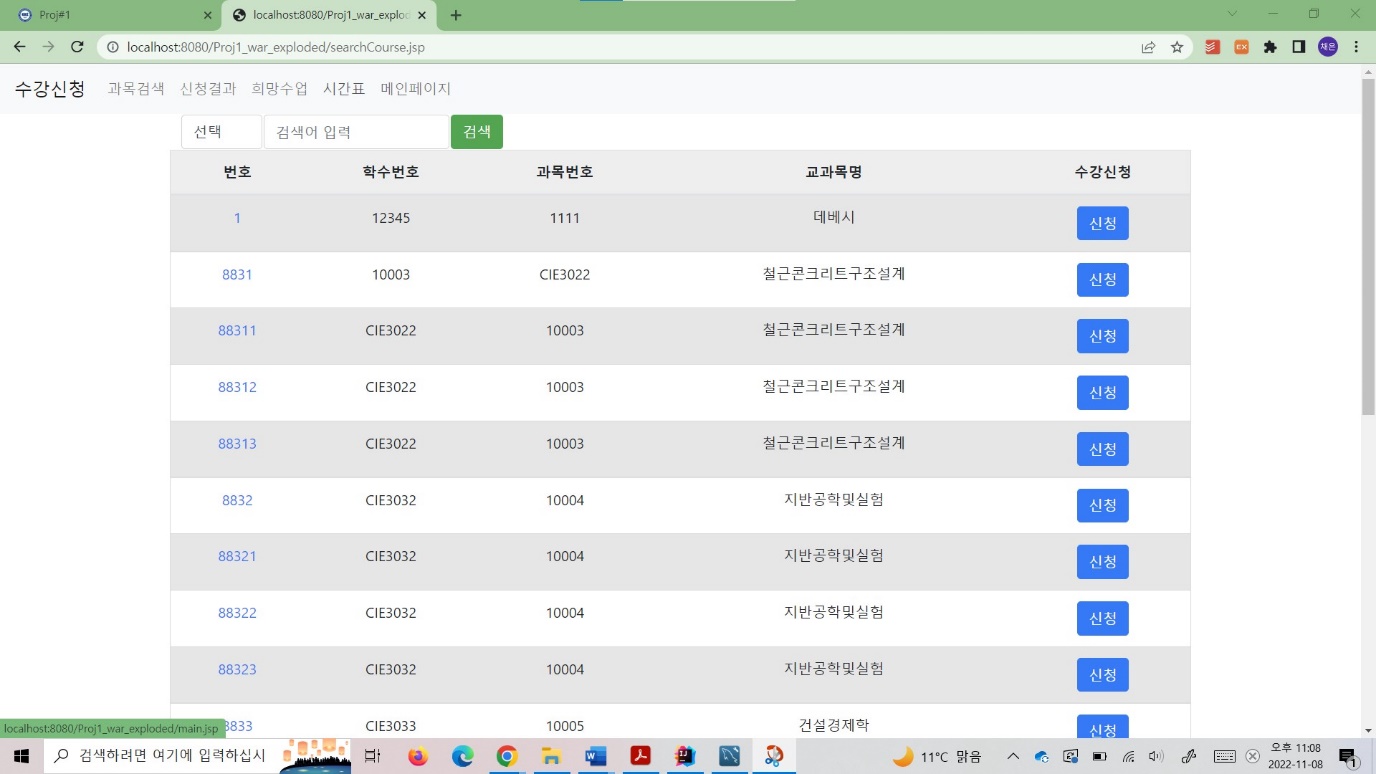
: 아이디와 비밀번호를 둘 다 입력하면 로그인 버튼을 눌러 로그인을 진행할 수 있다. 이때 어떤 아이디로 로그인하는가에 따라 학생 계정과 관리자(교강사) 계정으로 나뉘어서 로그인된다.

4. 학생 계정으로 로그인 했을 시 메인 페이지



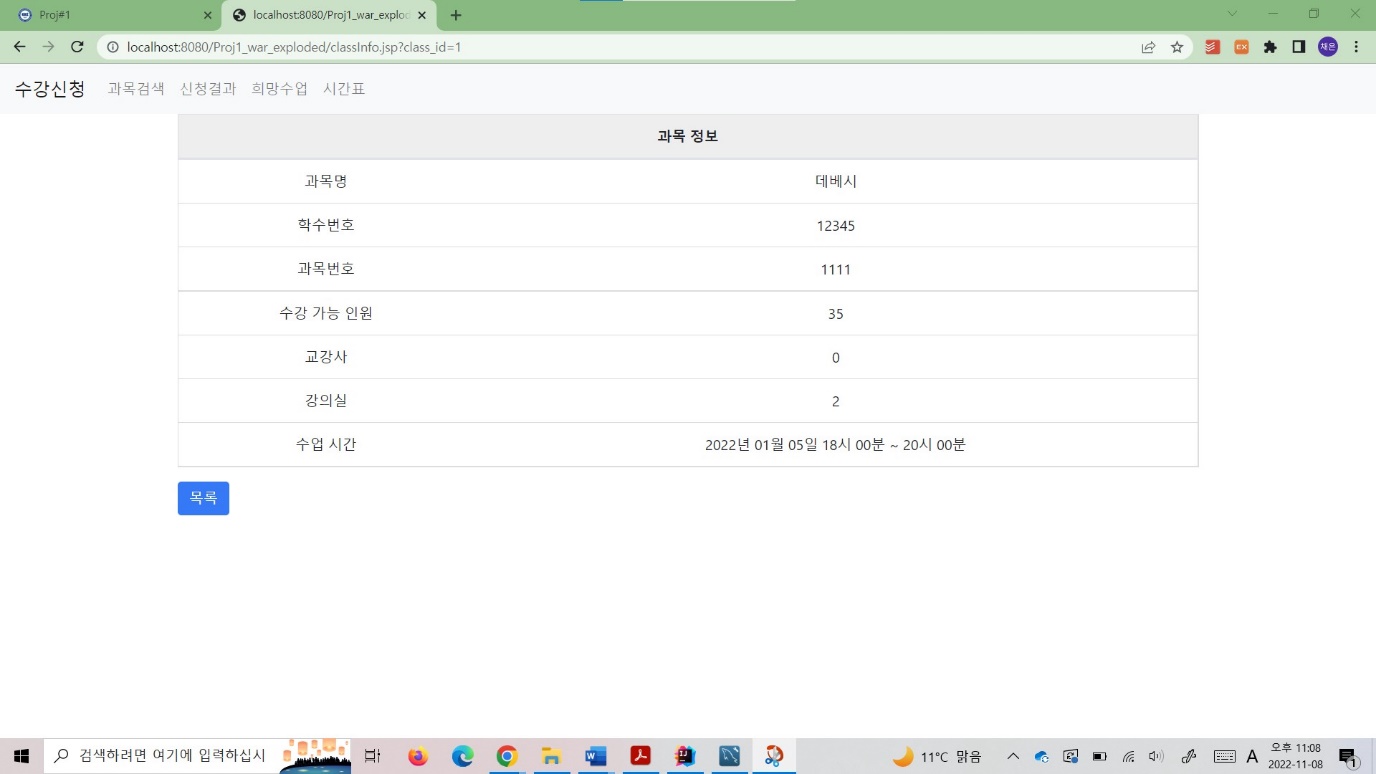
: 학생 계정으로 로그인 했을 시 로그아웃 버튼과 수강신청 버튼이 존재한다. 로그아웃 버튼을 누르면 로그아웃이 되면서 처음의 메인 페이지로 돌아간다. 수강신청 버튼을 누르면 수강신청과 과목 검색을 할 수 있는 페이지로 넘어간다.

5. 학생 계정으로 로그인 했을 시 과목 검색 페이지



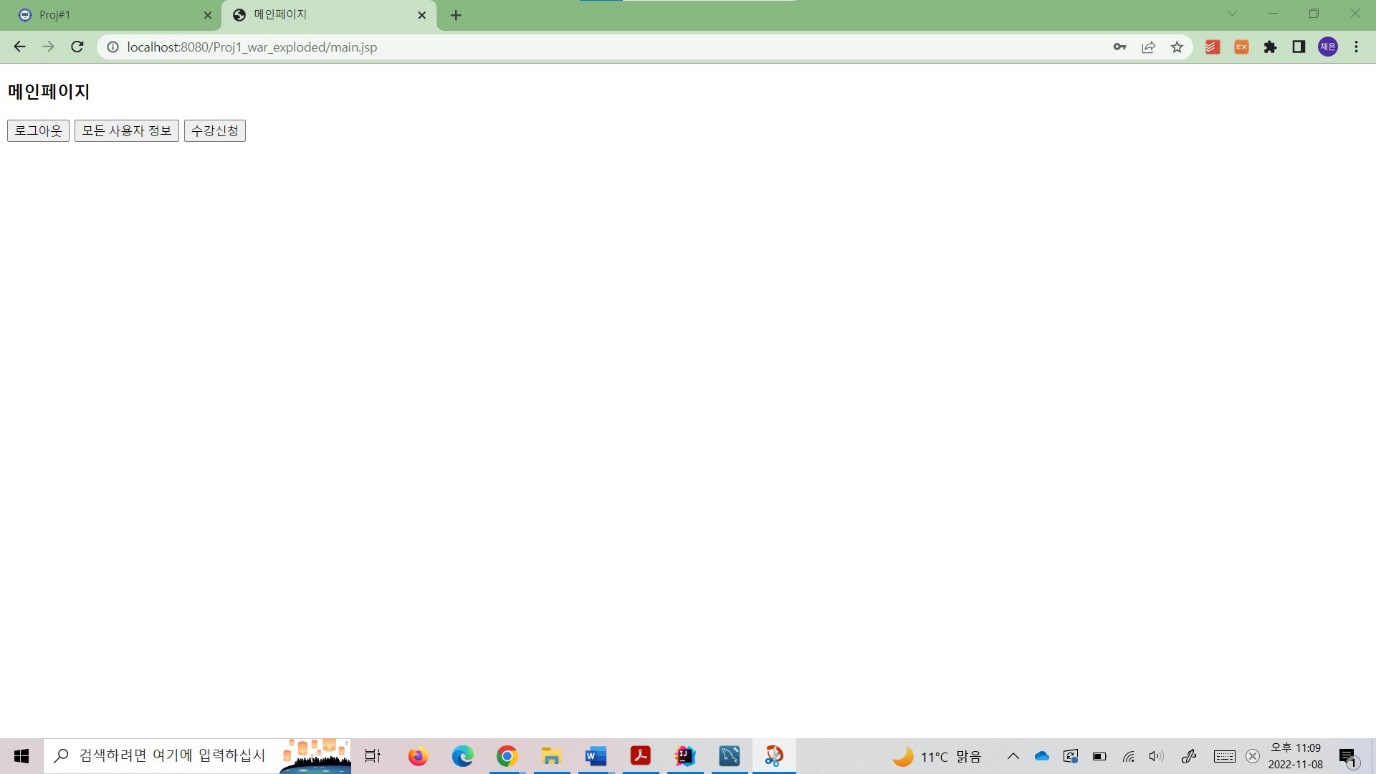
: 학생 계정으로 로그인 하고 수강 신청 버튼을 누른 뒤 과목 검색 창을 누르면 위의 화면으로 이동한다. 이 페이지에서는 과목 검색, 과목 정보 확인이 가능하다. 선택 부분에서 학수번호, 과목번호, 교과목 명을 선택해서 각각에 대해 검색할 수 있다.

6. 학생 계정으로 로그인 했을 시 과목 정보 페이지



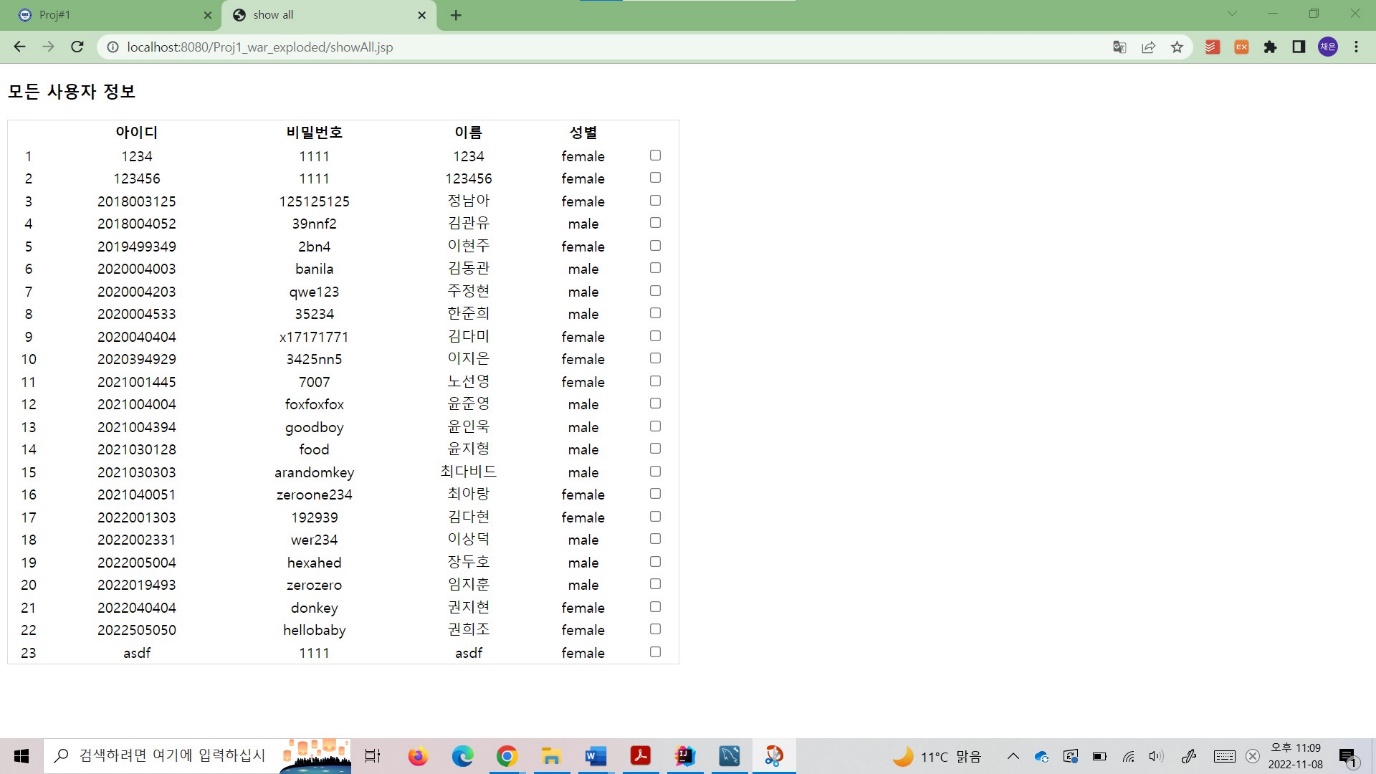
: 학생 계정으로 로그인 한 다음 과목 검색 페이지에서 번호를 누르면 해당 과목의 정보 페이지로 넘어간다. 과목 정보 페이지에는 과목명, 학수번호, 과목번호, 수강 가능 인원, 교강사, 강의실, 수업 시간이 출력된다. 이때 과목명과 학수번호, 과목 번호는 필수적으로 값이 출력되지만 수강 가능 인원, 교강사, 강의실은 입력하지 않았을 경우 0으로 출력되며, 수업 시간을 입력하지 않은 경우에는 아예 출력되지 않는다.

7. 교강사 계정으로 로그인 했을 시 메인 페이지



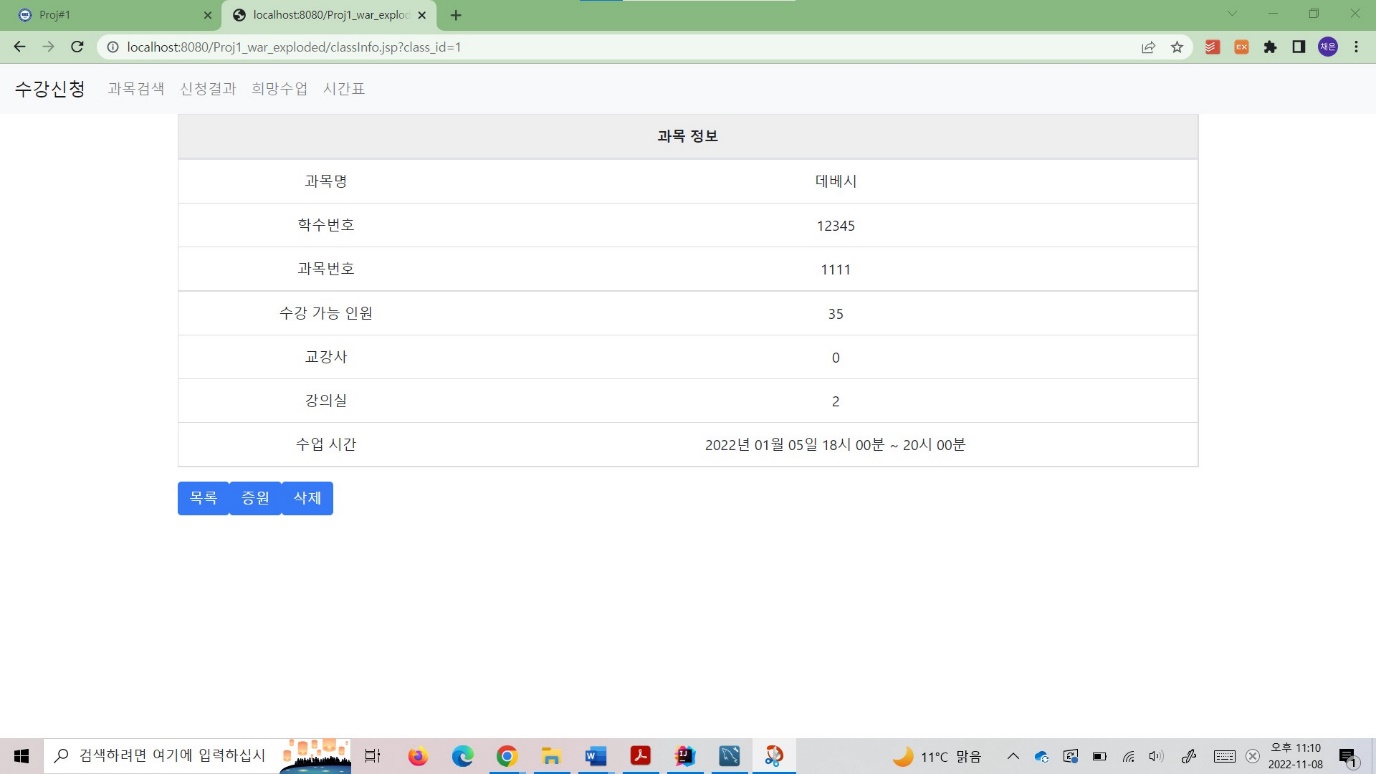
: 학생 계정으로 로그인 한 경우와는 달리 모든 사용자 정보를 볼 수 있는 버튼이 추가된다.

8. 모든 사용자 정보 출력 페이지



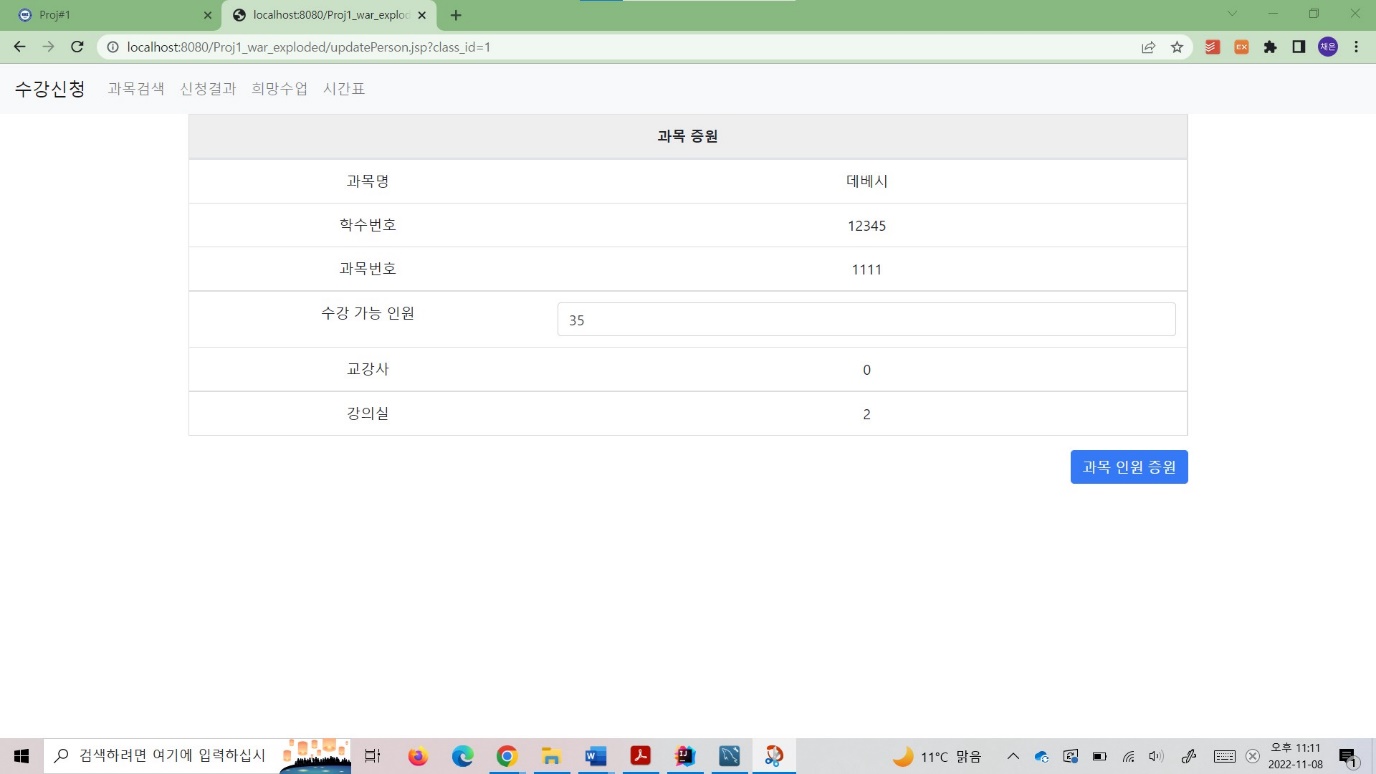
: 사용자 정보 출력 페이지에는 학생들의 정보만 출력된다. 학생의 아이디, 비밀번호, 이름, 성별이 출력된다.

9. 교강사 계정 로그인 후 과목 정보 페이지



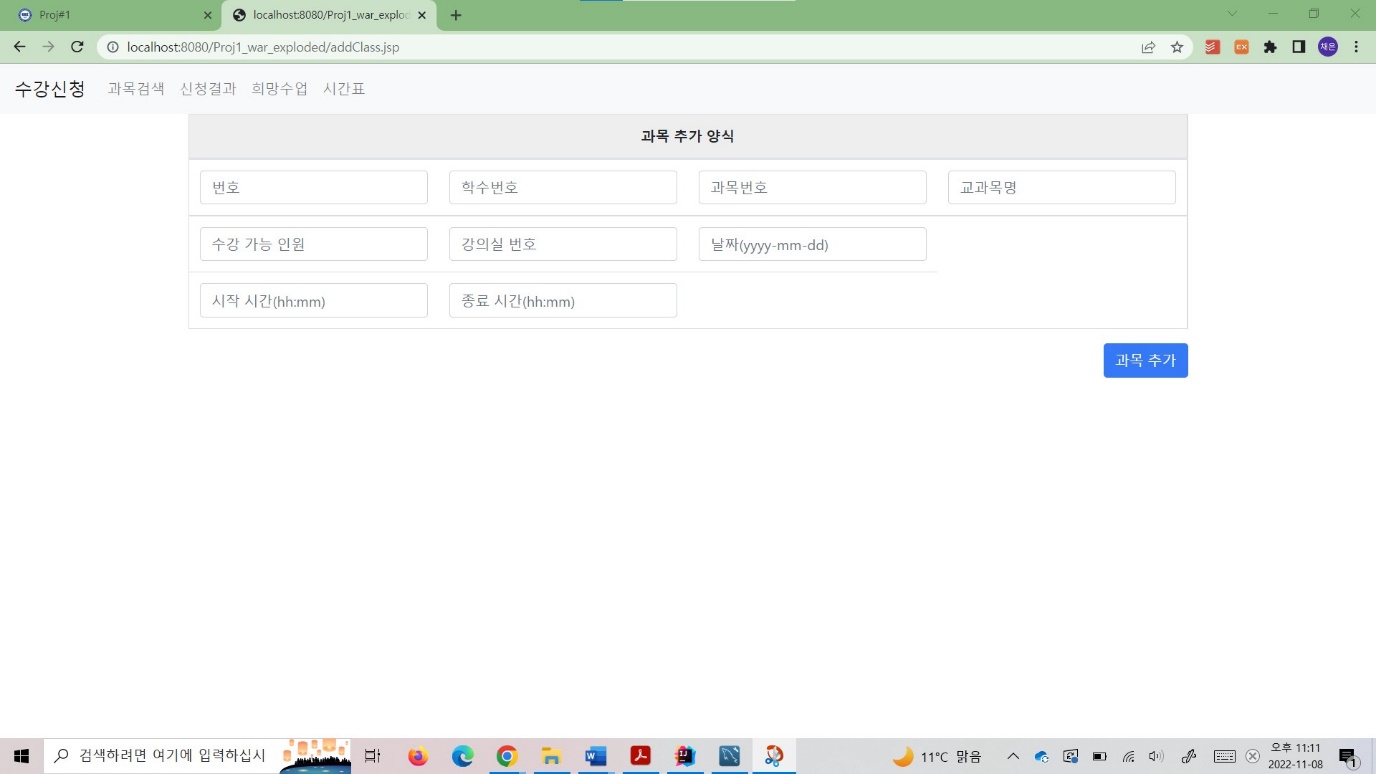
: 교강사 계정으로 로그인 한 다음 과목 검색 페이지에서 번호를 누르면 과목 정보 페이지로 넘어간다. 이때 학생 계정과는 다르게 과목 증원 버튼과 과목 삭제 버튼이 존재한다. 과목 삭제 버튼을 누르면 과목을 폐강할 수 있다.

10. 과목 증원 페이지



: 과목 증원 페이지에서는 수강 가능 인원을 조정할 수 있다. 이때 수강 가능 인원이 강의실의 최대 수용 가능 인원 수를 초과할 시, 과목 증원은 불가능하다.

11. 과목 추가 페이지



: 교강사 계정으로 로그인 했을 경우, 과목 검색 페이지 맨 아래부분에 과목 추가 버튼이 생성된다. 이 버튼을 누르면 위의 과목 추가 페이지로 넘어간다. 이때 번호, 학수번호, 과목번호, 교과목명은 필수로 입력해야 과목 추가가 가능하다. 또한 수강 가능 인원이 강의실의 최대 수용 가능 인원을 초과할 시 과목 추가가 불가능하다. 그리고 과목의 수강 날짜가 일요일인 경우에는 과목 추가가 불가능하다.

* **Trouble shooting**

1. 관리자 계정과 일반 사용자 계정 분리하는 것에 어려움을 겪음.

- 대부분의 경우에는 일반 사용자 테이블에 하나의 column을 추가해서 그 column의 값이 1이 되는 계정을 관리자 계정으로 설정해서 관리자 계정을 구분하는 경우가 많았다. 그러나 관리자 계정과 일반 사용자 계정의 table을 나눠서 제작하다 보니 로그인 시 이를 구분하는 과정에서 어려움을 겪었다. 그래서 교강사 계정의 DAO에서도 login 함수를 새로 만든 다음, lecturerDAO와 studentDAO의 결과를 다른 int 값에 넣어 그 결과에 따라 학생과 교강사 계정을 구분했다.

1. 웹을 처음 제작해보는 터라 시간이 부족해서 다양한 기능을 만들지 못함.

- 웹을 이전에 제작해 본 경험이 없었기 때문에 하나하나 다 공부를 해서 만들어야 했는데, 그 과정에서 시간이 오래 걸려서 처음에 만들고자 했던 기능을 다 제작하지 못한 점이 아쉬웠다.

1. 서로 다른 table을 함께 사용해서 기능을 개발하는 부분에서 어려움을 겪음.

- 과목의 수강 가능 인원 수를 늘리는 과정에서 해당 강의실의 최대 수용 가능 인원 수를 넘으면 안된다는 조건을 걸었다. 그런데 과목의 수강 가능 인원 수는 class table에, 강의실의 최대 수용 가능 인원 수는 room table에 있다. 그래서 이 둘을 비교하는 과정에서 어려움을 겪었다. 그래서 roomDAO에서 room\_id가 주어졌을 때 강의실의 최대 수용 가능 인원 수를 출력하는 함수를 새로 만든 다음 그 결과값과 입력된 person\_max의 값을 비교하는 방식을 사용했다.