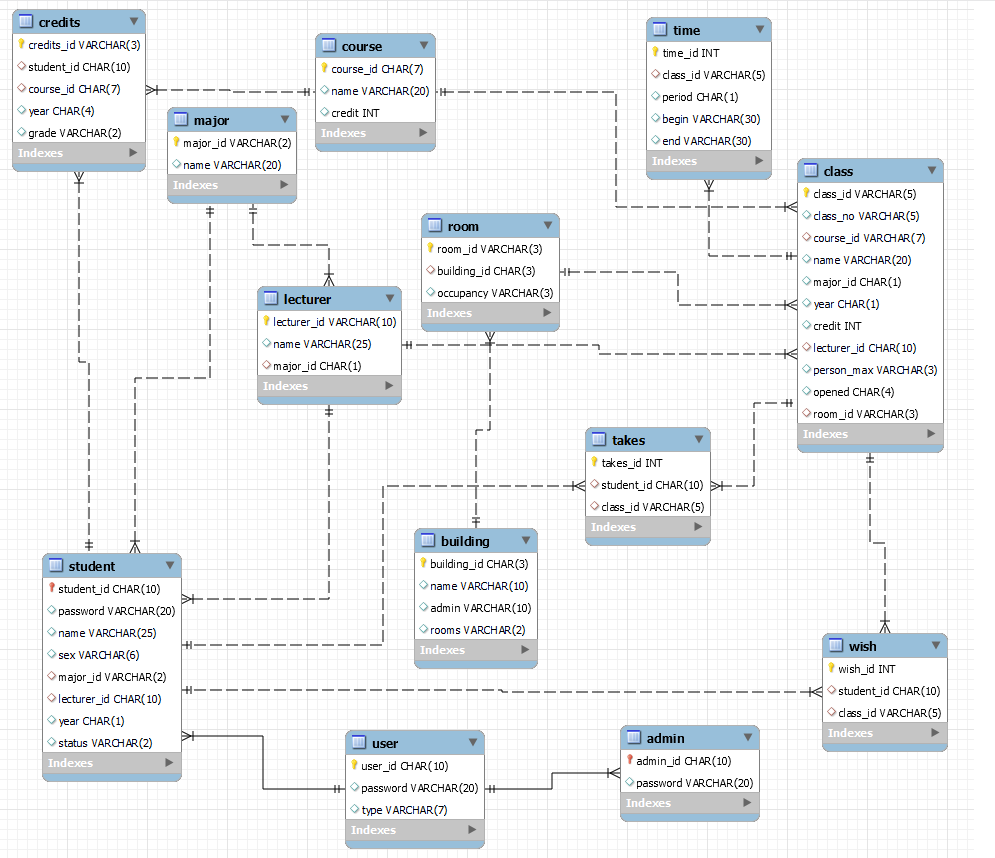
1. 스키마 디자인



주어진 조건에 따라 쿼리문들을 작성해 가면서 필요한 테이블이 무엇이 있는지 고민을 해보았다.

기존에 주어진 테이블에 새롭게 추가된 테이블들은 takes, user, admin, wish 이다.

기존 스키마에서는 student테이블에 status를 추가하였다. (학생의 재학 휴학 자퇴 여부)

Takes는 ‘학생’이 수강신청한 ‘과목’들을 나타내고, 따라서 왜래키 student\_id, 왜래키 class\_id를 부여하였다.

Wish는 ‘학생’이 희망하는 ‘과목’을 나타내고 따라서 왜래키 student\_id, 왜래키 class\_id를 부여했다.

User는 ‘학생과 관리자 모두’를 나타낸다

Admin은 ‘관리자’를 나타내고 따라서 왜래키 admin\_id를 부여했다.

Foreign key를 부여하는 과정에서 on delete cascade를 부여하였다. (부모 테이블에서 삭제가 일어나면 자식 테이블에도 삭제가 이루어져야 하기 때문)

1. UI 계획

한양대학교 수강신청을 참고해서 UI를 만들 계획이다. CSS와 JS를 사용할 예정이고 부가적인 라이브러리는 필요한 경우에만 사용할 예정이다. 웹 개발에서 디자인을 미리 설계하고 개발하는 것은 매우 중요하다. 무작정 개발을 시작했다가 뒤늦게 리팩토링 하기 위해서 코드를 수정하는 작업이 상당히 오래 걸리기 때문이다. 따라서 생각할 수 있는 모든 페이지를 구상한 뒤 페이지들을 어떻게 패키지로 묶을 것이며, 페이지 별로 어느 부분이 공통적인 속성을 갖고 페이지 모듈화가 올바르게 이루어 질지 고민하는 작업을 거쳤다. (페이지별 디자인은 따로 파일로 첨부했습니다)

1. 개발 환경

IDE – IntelliJ IDEA 2022.2

Database – MySQL 8.0

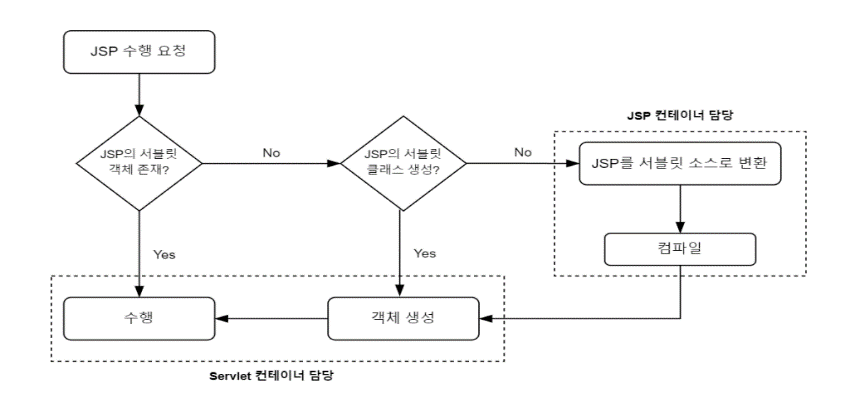
Server - Tomcat 9.0.68

SDK - Amazon Corretto version 15.0.2

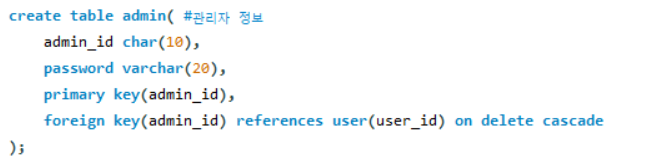
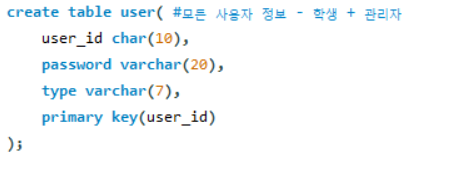
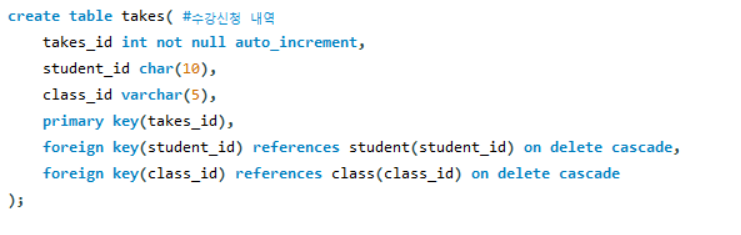
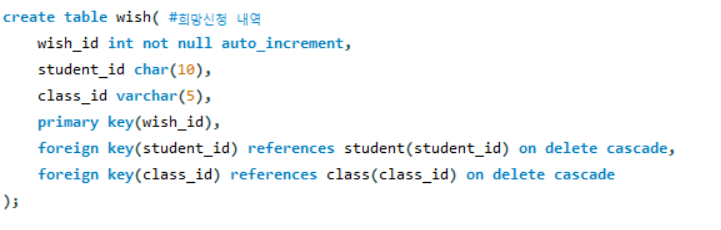
Connector – mysql-connector-java-8.0.20.jar

1. 코드 동작 흐름도

DAO객체는 데이터베이스와의 연결을 위한 메소드들을 가지고 있으며 데이터베이스 커넥션 조건들까지도 포함하고 있다. DTO는 우리가 DAO에 있는 메소드를 호출하기 위하여 값을 DTO에 담아서 인자로 넘기거나 DAO에 있는 실행 결과가 DTO에 담겨져서 나온다. 운송해주는 역할을 해준다. DAO의 메소드 안에는 쿼리문들이 있어 데이터베이스에다가 질의를 실행해주는 메소드들로 표현이 되어있다. 화면 내에서의 동적인 흐름들은 JS로 제어가 가능하다. 특정 조건에서 경고창을 표시하는 것이나 특정 태그의 visibility를 조절할 수 있다. JSP의 구체적인 동작 흐름도는 다음과 같다.



1. 스키마에 알맞은 input dataset



위는 기존 존재하는 테이블에 새롭게 추가한 테이블들이다. 다른 테이블에 대한 데이터들은 기존에 존재하는 csv 파일로 사용하였으며 wish 와 takes는 홈페이지에서 희망 신청, 수강 신청에 따라 추가된다. User 과 admin은 따로 데이터를 만들어서 넣어줬다. User에는 기존에 있는 student 테이블의 데이터와 새롭게 추가한 admin의 2021058995(내 학번) 과 비밀번호를 넣었다. Student 테이블에는 status를 일괄적으로 “재학”으로 해놨다.

1. 구현된 코드 설명

큰 틀에서 공통적으로 포함할 수 있는 코드들은 webapp바로 하위에 놓고 구체적으로 student와 admin에 따라 포함해야 하는 코드들은 각각의 student 디렉토리와 admin 디렉토리 하위에 놓았다.

1. 로그인 창

로그인을 하는 페이지는 두가지 기능이 있다.

첫번째는 수업 검색이고 두번째는 로그인이다..

검색은 학생, 관리자, 로그인 안 한 사람 모두에게 사용되는 기능이므로 search.jsp를 webapp 하위에 놓았다. 로그인 과정에서는 아이디와 비밀번호 둘 중 하나라도 비어 있으면 입력 경고를 띄우고, 둘 다 입력했을 때 아이디가 데이터베이스에 없다면 아이디가 없는 경고를, 아이디는 존재하지만 비밀번호가 일치하지 않는 경우에는 비밀번호가 일치하지 않는다는 경고를 JS의 alert기능을 통해 표시하였다. 또한 DAO에서 checkType이라는 함수가 있는데, 이 함수는 user\_id를 받아 type이 학생인지 관리자인지 검사를 진행한다. 따라서 학생이면 studenthome.jsp로 관리자면 adminhome.jsp로 이동시켰다. 상세 내용은 joincheck.jsp에서 확인해볼 수 있다.

로고와 이름은 모든 페이지에 공통적으로 적용되며 클릭 시 메인 화면으로 이동시킨다.

1. 학생 메인 화면

학생 메인 화면에서 총 4가지 기능(수강편람, 수강신청, 희망수업, 신청내역)이 가능했고 로그인 창에서는 로그아웃이 기능이 있다.

-수강편람

이번 과제에서 가장 까다로웠던 부분이었던 것 같다. 결국 다른 기능들도 비슷한 메커니즘으로 동작하기 때문에 수강편람 부분을 끝내고 나서야 다른 부분은 수월하게 할 수 있었다. 수강편람에서의 가장 큰 딜레마는 쿼리 처리 vs 자바 처리였다. 칼럼들이 각각 서로 다른 여러 개의 database 테이블에 걸쳐 있는 것 뿐만 아니라 특히 신청인원/수강정원을 표시하는 것은 sql의 group by를 사용하거나 따로 이 칼럼에 대해서만 값을 별도로 구하는 작업을 해야 했었다. JSP 코드들을 보면 search.jsp에서 튜플들을 for문 루프를 돌리면서 출력하는 과정이기 때문에 깔끔한 방법으로 진행하고자 한 ‘searchDTO의 오브젝트 하나는 한 튜플에 대한 정보’를 지니도록 만들었다. 결국 SQL 코드가 상당히 복잡해지는 결과를 만들었지만 jsp가 복잡해지는 것보다 더 나은 선택이었던 것 같다. 약간 까다로웠던 부분들 중 하나는 time테이블과 관련된 부분이었던 것 같다. Time 테이블을 보면 period 값에 따라 같은 class\_id를 지닐 수 있었기 때문에 다른 테이블들과 조인하는 과정에서 생각을 많이 하게 만들었다. 튜플들의 정보를 확인한 결과 period1이 없이는 period2가 존재하지 않는다는 사실을 이용하여 class테이블에 period가 1인 time테이블을 이너 조인 한 뒤 다시 period가 2인 time테이블을 아우터 조인 함으로써 한 테이블에 period1과 2에 대한 속성을 지니도록 하였다. 리팩토링의 아쉬운 부분들 중 하나는 시간을 변환하는 과정이었다. ISO8061을 변환하는 과정들을 따로 클래스를 만들어서 import하는 방식으로 진행하였더라면 코드가 더 가독성있게 구성되었을 것 같다. 너무 많은 페이지에서 비슷한 작업을 수행한지 한참 뒤라 고치기에는 시간이 부족했다. (나중에 해볼 것)

검색 필드의 조건은 세 개, 검색 필드는 하나이므로 select와 option태그를 이용한 뒤 변수 searchFiled와 searchWord를 사용하였다. DAO에 있는 selectList 함수에서는 SQL문을 if문을 활용하여 searchField에 따라 어떤 형식의 검색이 가능한지 확인하였다.

-수강신청

수강편람에서 희망, 재수강 대상 과목 칼럼이 추가되었다. 검색 과정은 수강편람에서와 동일하다. 수강신청한 뒤 신청내역에 목록을 보여주는 작업과 희망 신청을 한 뒤 희망수업에 목록을 보여주는 작업이 상당히 유사하였다. 기존에 수강편람 검색 쿼리에서 아주 조금씩만 덧붙이면 해결이 되었다.

-희망수업

-신청내역

1. 어드민 메인 화면

어드민 메인 화면에서도 총 4가지 기능(수강편람, 통계, 학생 정보 조회/변경, 강좌 개설/삭제)이 가능했고 학생과 마찬가지로 로그인 창에서는 로그아웃 기능이 있다.

-수강편람

학생과 유일하게 다른 점은 신청 칼럼을 제외한 것이다.

-통계

학생들의 정보 조회 기능을 구현하는 과정에서 학생들의 평균 점수를 구하는 SQL을 미리 작성했기 때문에 이부분은 생각보다 빠르게 해결됐다. 학생들의 평균 성적을 구한 테이블을 학생들의 id를 기준으로 credits테이블과 이너 조인을 한 뒤 과목으로 group by를 하면서 avg(평점평균 – 과목학점)을 수행하면 각 과목별로 값이 커짐에 따라 해당 과목에서 학생들이 점수를 잘 못받았다는 사실을 알려주는 테이블이 완성된다.

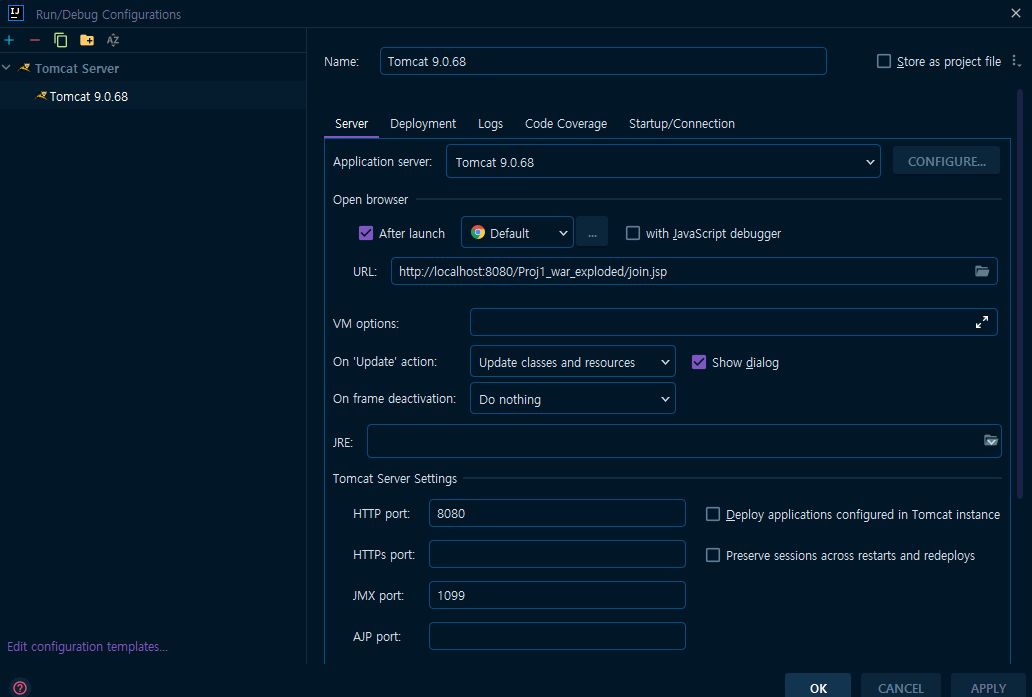
-학생 정보 조회/변경

학생 정보를 조회하는 검색필드 조건은 학생ID와 학생 이름이며 검색 조건 특성상 완전 일치 검색으로 구현하였다. 테이블 헤더 중에 ‘금 학기 시간표, 성적, 상태’ 기능을 새롭게 구현하였다. 금 학기 시간표는 조회 버튼을 누르면 학생이 수강신청한 시간표가 나온다. 성적은 (학점 수\*parsing한성적)의 총합 / 학점 수 총합을 통해 학생의 평균 학점을 SQL 쿼리로 구현하였다. 조회버튼을 클릭하면 학생의 성적들에 대한 상세한 자료가 나온다. 이 부분은 credits테이블에서 해당 학생의 id와 일치하는 값들로 가져왔다. 상태는 재학 휴학 자퇴 세가지 조건이 있으며 Update문을 사용하여 student 테이블의 status 속성을 바꾸는 기능을 구현하였다.

-강좌 개설/삭제

수강 편람 기능과 매우 유사하며 삭제하기라는 테이블 헤더가 존재하였다. 강좌를 개설하는 과정에서는 개설 버튼을 누르면 개설하고자 하는 강좌의 조건들을 입력할 수 있도록 하였다. 강좌 개설에 있었던 조건 3에 대해서는 E-learning 부분을 체크하도록 만들었고 애초에 18:00 이후에 개설이 불가능하도록 만들었다. 조건2에 대해서는 SQL 테이블 생성 과정에서 on delete cascade를 통해 가능하였다.

1. 빌드 과정 및 실행 과정



서버를 새롭게 켰을 때 join.jsp가 열리도록 URL을 설정해주었다. 다른 configuration들은 기존의 설정과 크게 다르지 않다.

1. 각 요구사항에 대한 실행 결과

메인 화면



로그아웃 상태에서 수강편람

테이블이(가) 표시된 사진

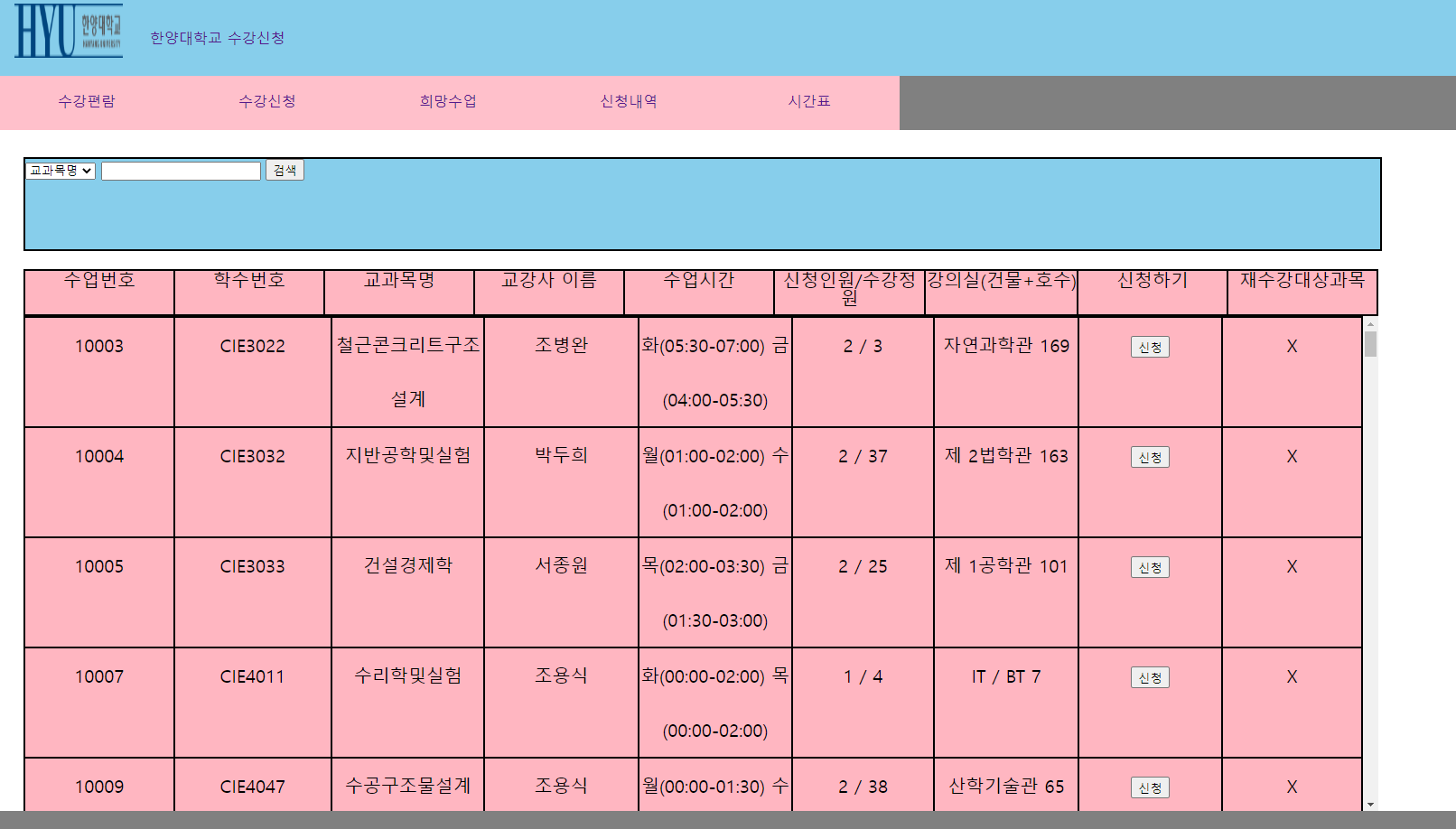
자동 생성된 설명

1. 학생의 입장에서 차례대로 확인해볼 것이다.

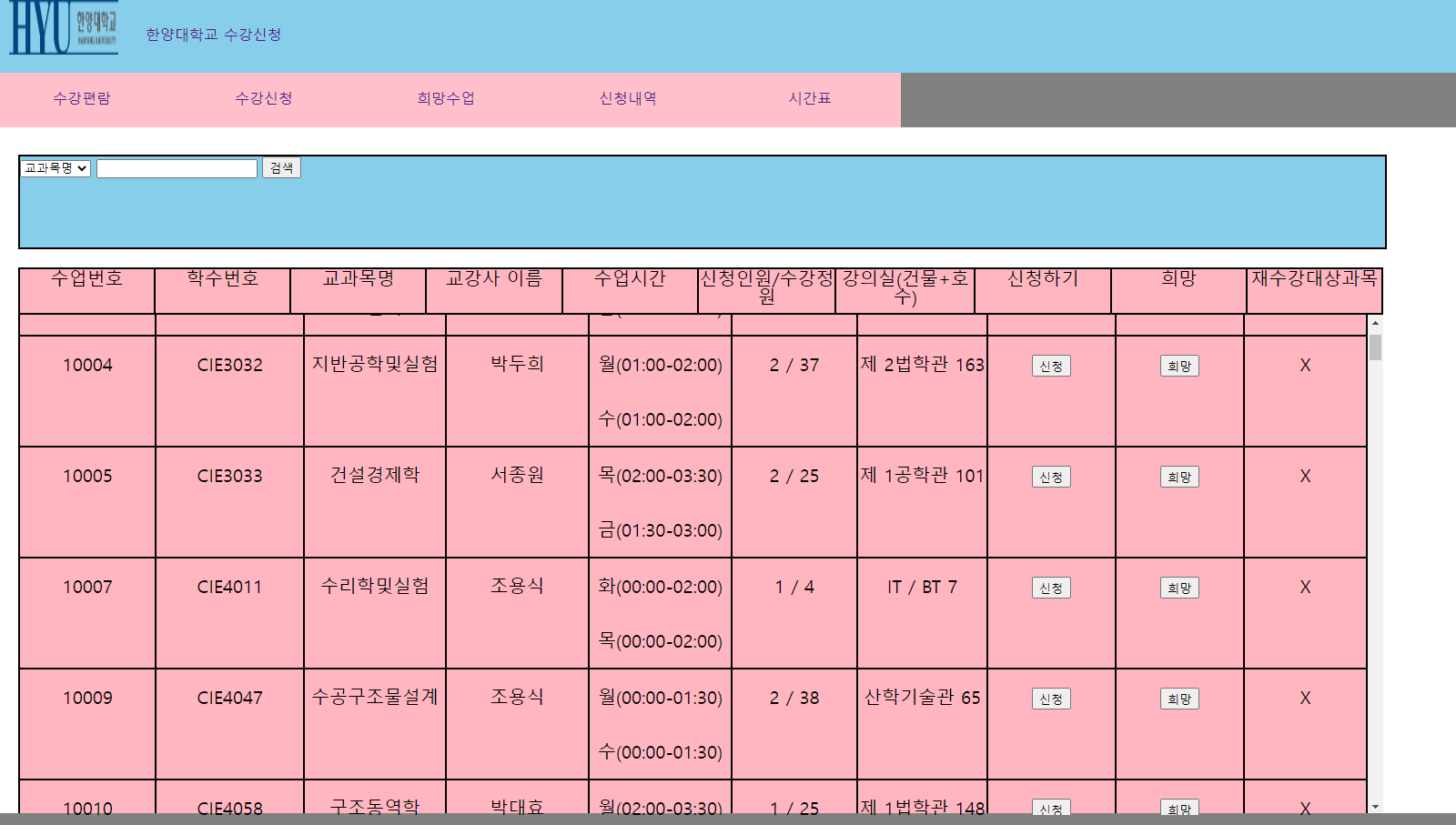
학생 메인



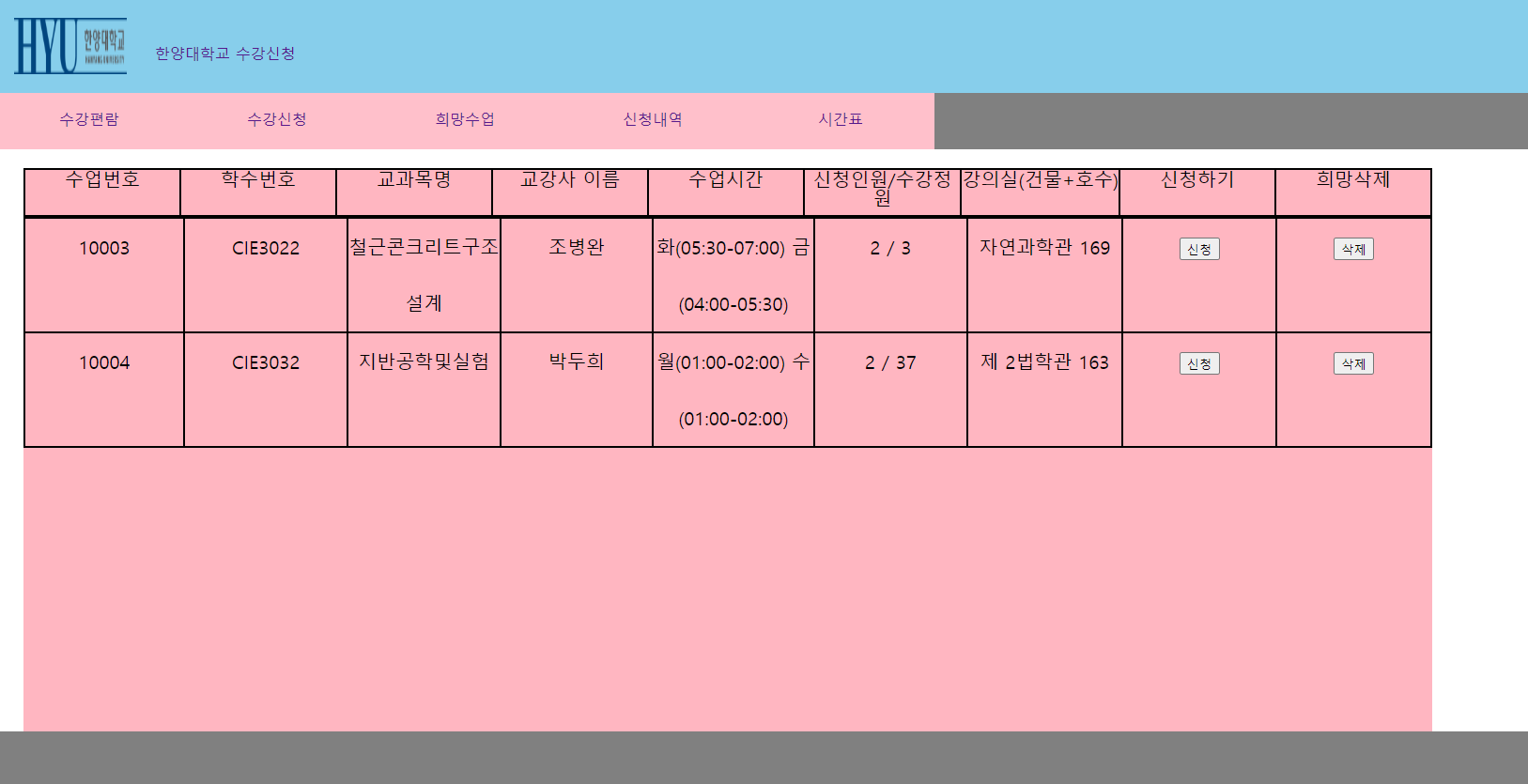
학생 수강편람



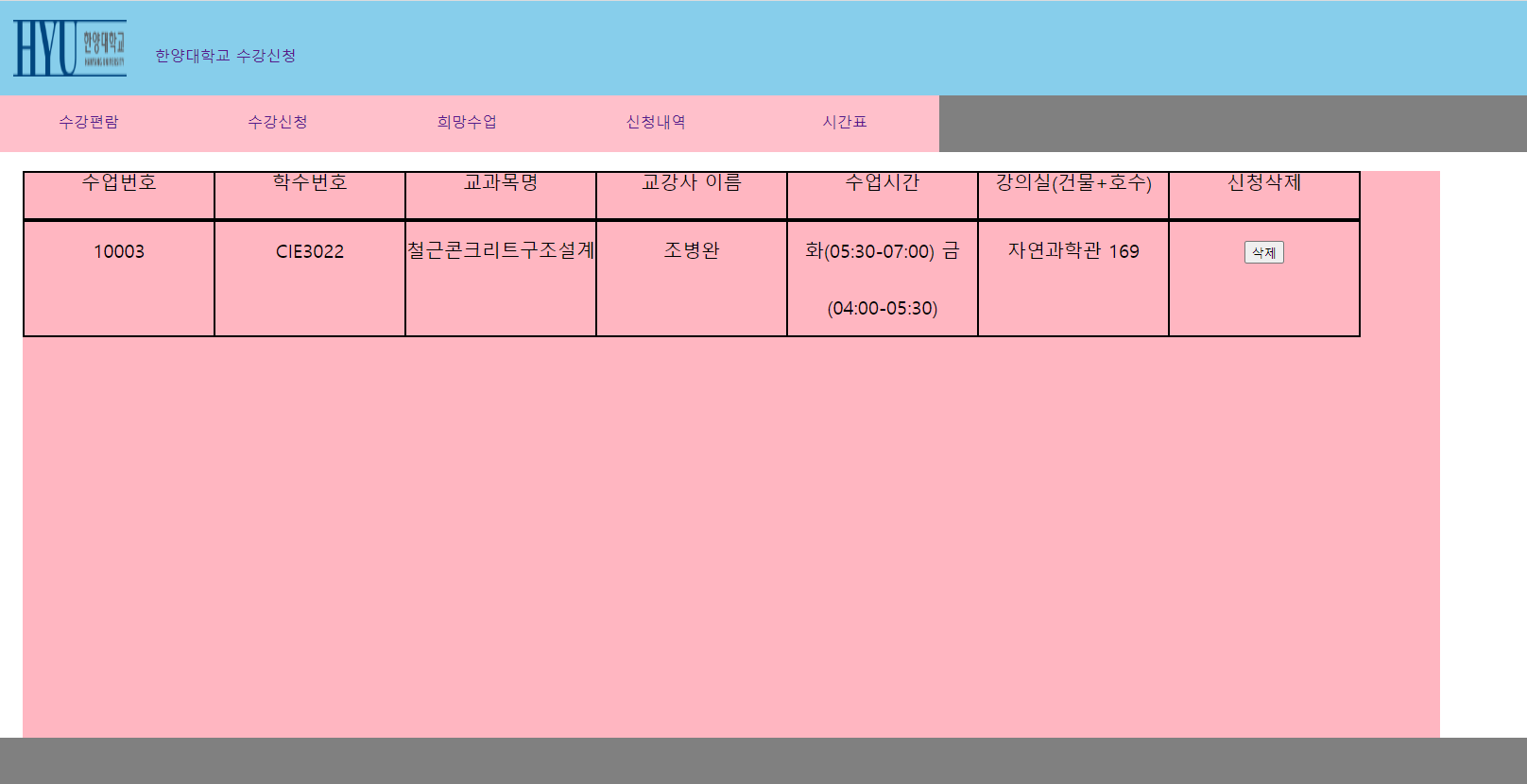
학생 수강신청



희망수업(추가한 뒤)

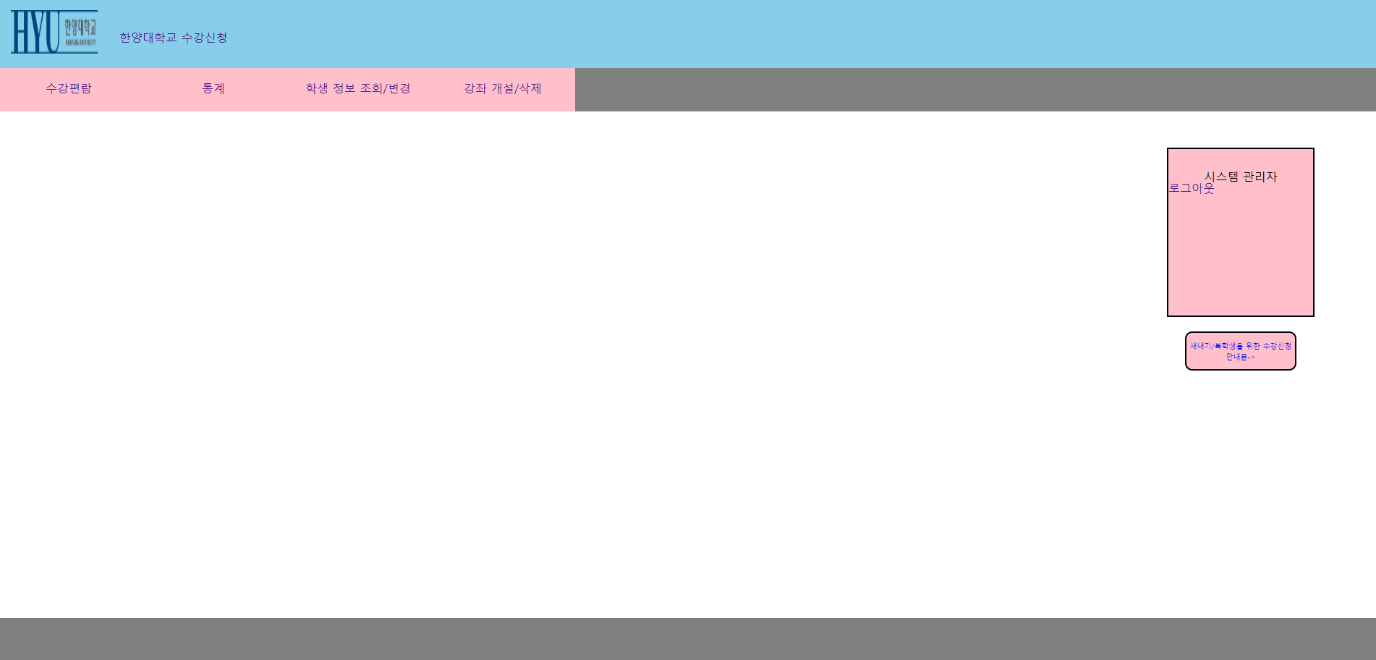


신청내역(신청한 뒤)

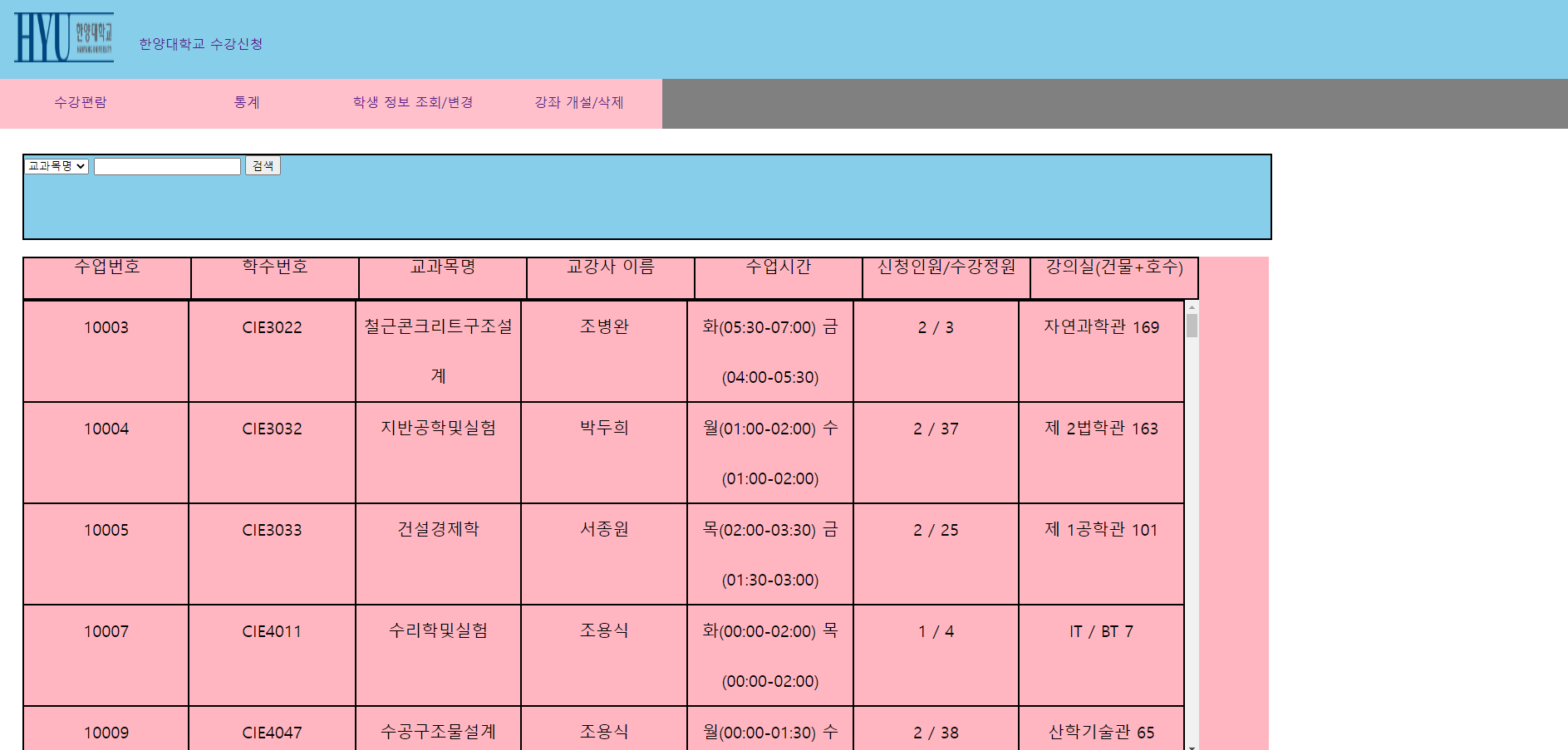


1. 관리자의 입장에서 확인해볼 것이다.

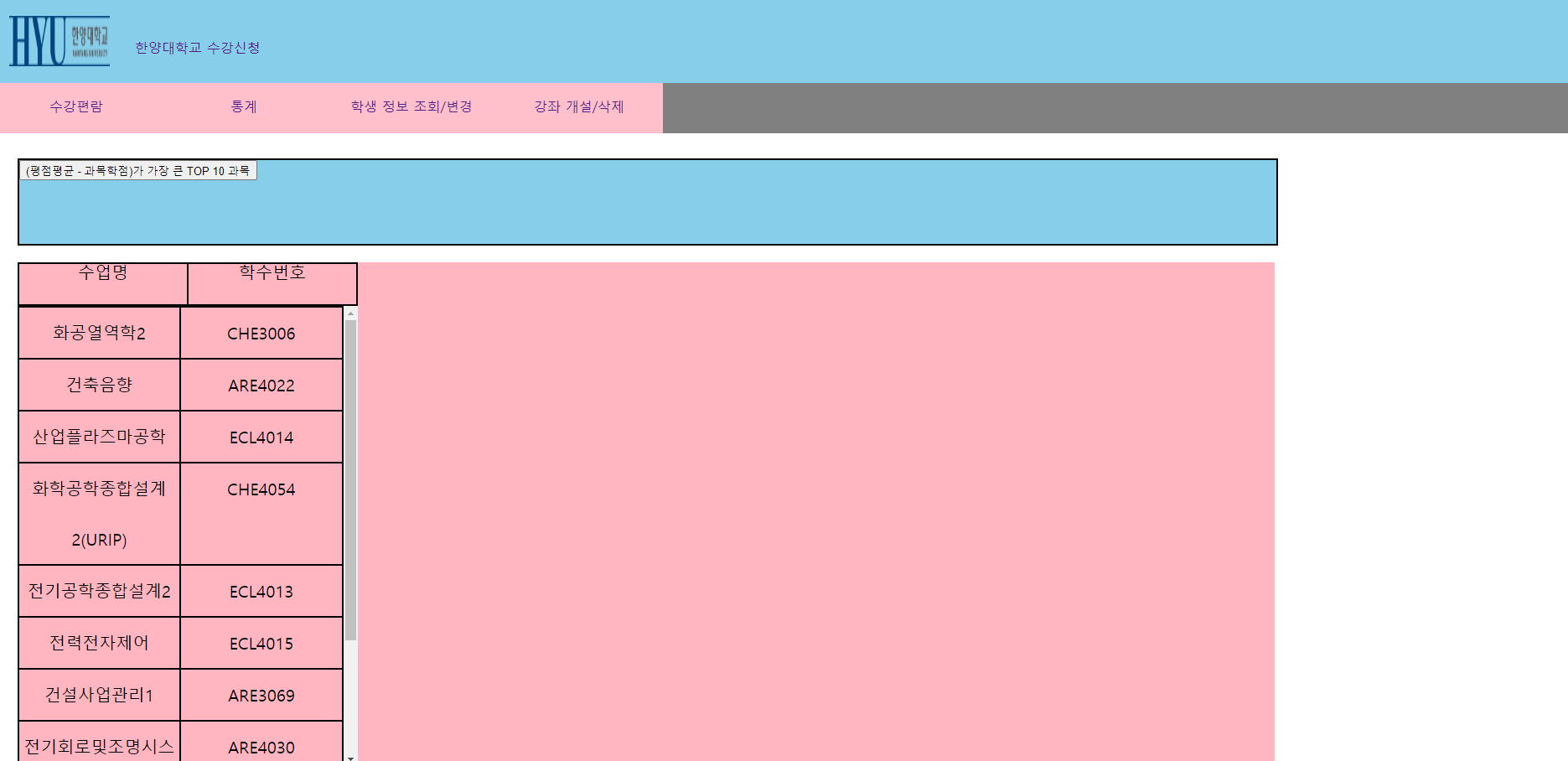
관리자 메인 화면



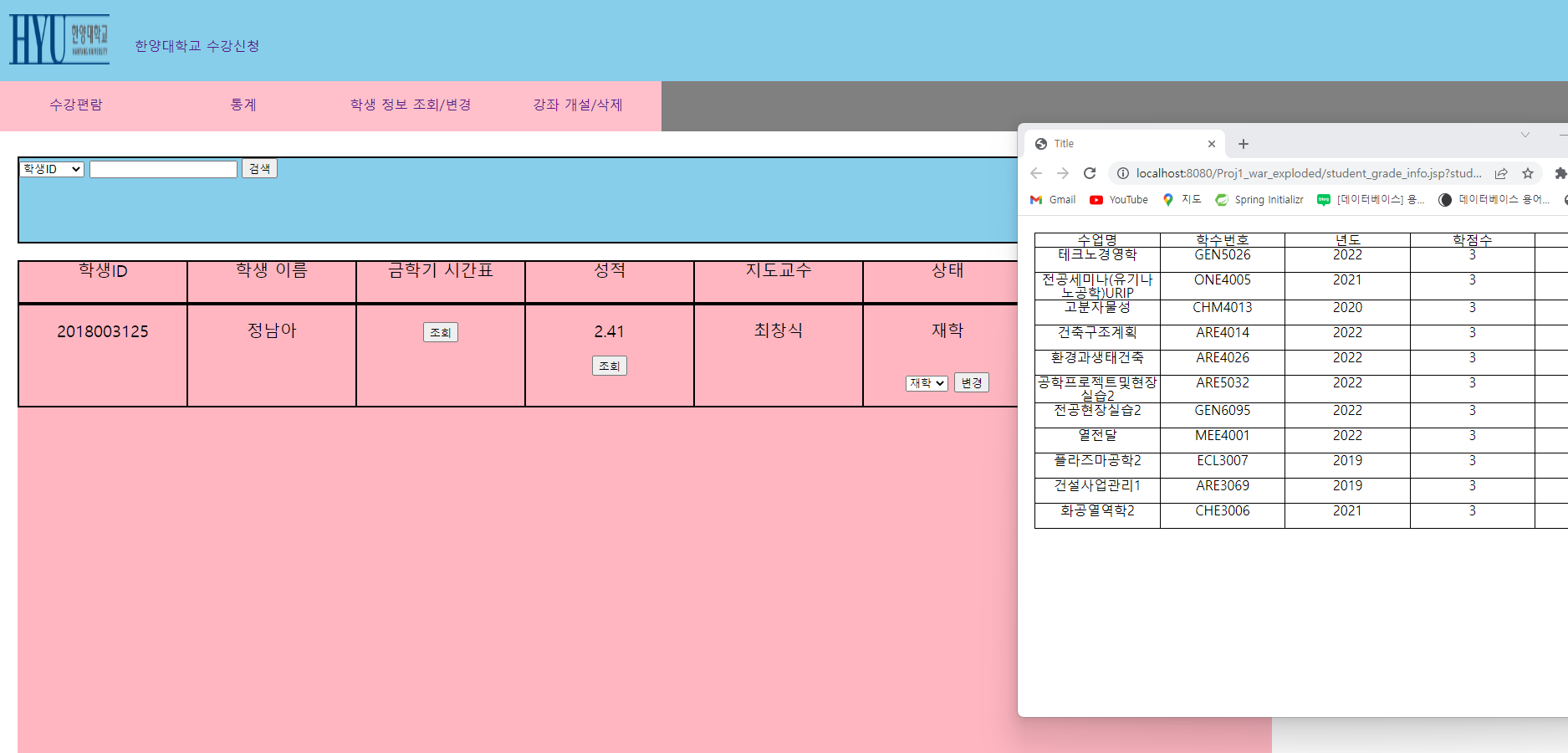
관리자 수강편람



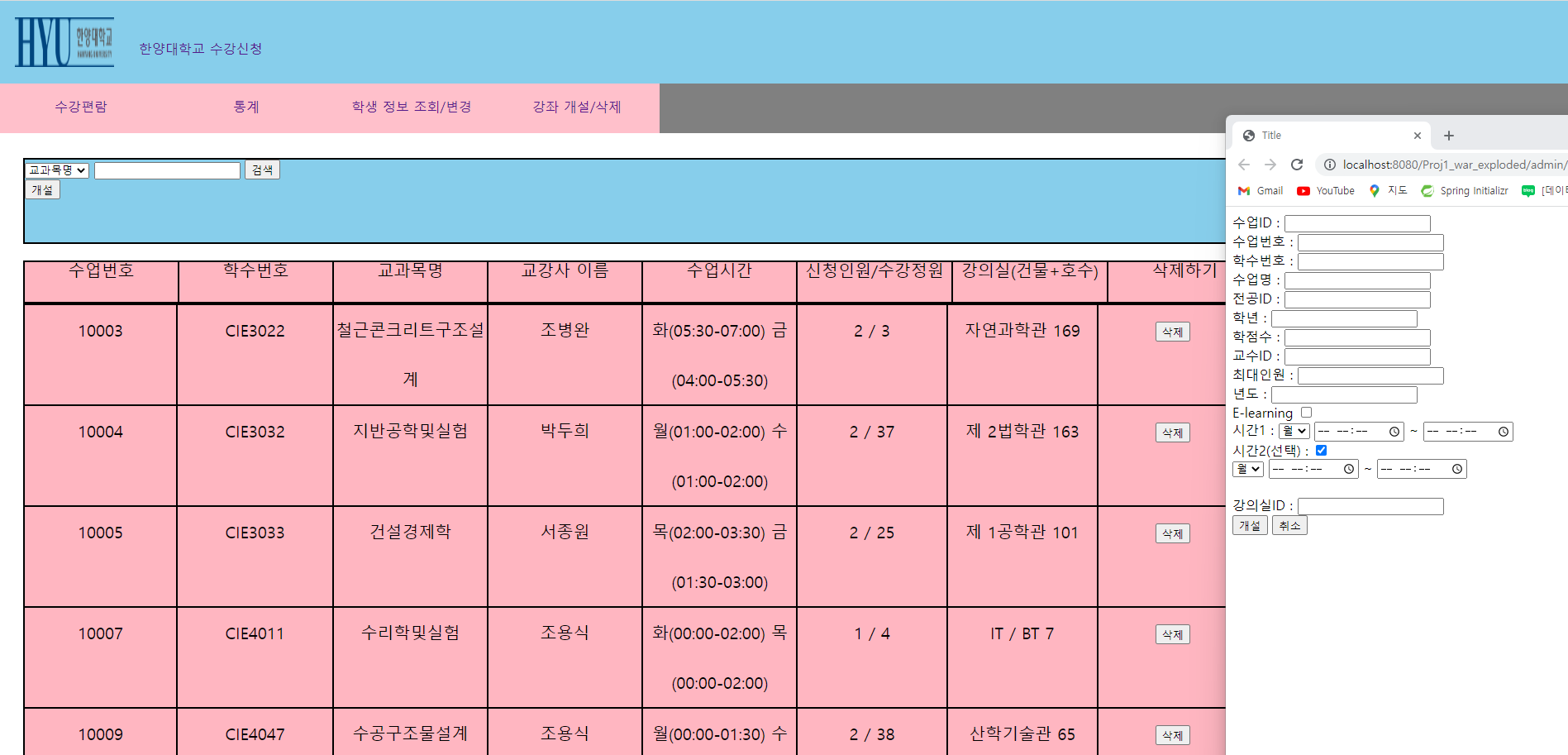
관리자 통계



학생정보 조회/변경



강좌개설/삭제



1. Trouble shooting

-스키마를 디자인 하는 과정에서 발생한 오류

마주한 첫 번째 오류는 “Incorrect foreign key definition for ‘foreign key without name’: reference and table don’t match”였다.

오탈자가 있어서 발생한 문제라고 생각했지만 신기하게도 foreign키의 참조 대상이 아닌 table들’만’ 올바르게 생성된 것이 확인되었고 아직 참조하는 테이블이 생성되지 않은 상태에서 테이블을 생성하려고 해서 발생한 오류였다.

마주한 두 번째 오류는 ERROR 1046:No database selected였다.

이 에러는 foreign key로 참조하고자 하는 테이블의 attribute를 적지 않고 테이블 이름만 적어서 발생한 문제였다.

Ex) foreign key (room\_id) references room(room\_id) on delete cascade

초록색 표시된 곳을 처음에 안 적었었다.

테이블csv 값들을 import 하는 과정에서 왜래키를 고려하여 import하지 않아서 발생한 오류도 발생하였다. (테이블을 생성하는 과정도 왜래키를 고려하여 차례대로 생성해야 한다.)

-JSP 코드를 구현하면서 발생한 오류

가끔 DAO에 있는 메소드를 수정한 다음 build하고 run하면서 no such method exception이 날 때가 있었다. 맞게 했다고 생각하고 서버를 껐다가 다시 키니까 정상적으로 작동하였다.

가끔 CSS 작업이 웹 브라우저 상에서 반영이 안되는 경우가 발생했는데, 이 경우는 웹 브라우저의 캐쉬를 삭제하니까 정상적으로 CSS가 들어갔다. 이전에 실행했던 내용의 정보를 기억하고 있어서 그렇다고 한다.

이외에 특별한 오류는 없었다.

10)느낀 점

웹에 대한 디자인 패턴을 공부한 뒤 가독성을 개선시키면 좋을 것 같다는 생각을 하였다. (Gof의 디자인 패턴을 읽을 계획이다)

웹 개발이 상당한 시간이 필요하다는 사실도 깨닫게 되었다. 페이지 하나를 개발하는데 시간이 많이 걸렸지만 페이지가 수십장이나 되기 때문이다.

그리고 무엇보다도 자바스크립트를 제대로 잘 알고 있는 것이 웹 개발에 상당히 도움이 된다는 사실도 알게 되었다. 자바스크립트로 할 수 있는 기능들이 너무 많았지만 문법들을 제대로 알고 있지 않아 자료들을 참고하는데 어려움이 있었다.