모의 수강신청 사이트 제작

작성자: 2021038131 장준혁

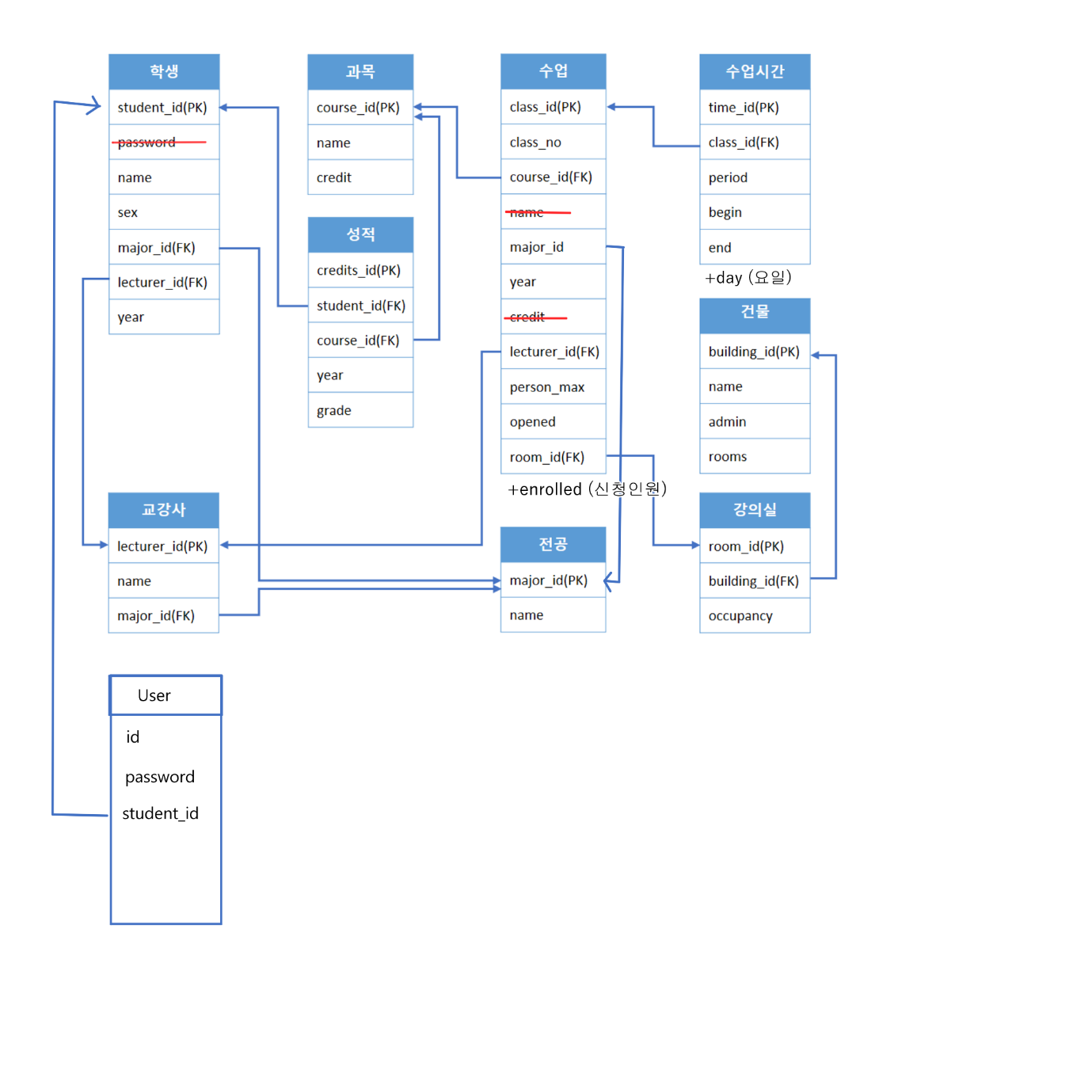
연락처: [ggogono3@naver.com](mailto:ggogono3@naver.com)

시작일: 2022년 10월 13일

종료일: 2022년 10월 26일

프로젝트 다운로드: <https://github.com/ClearSky-S/sugang>

# Schema 디자인



실제 구현은 sugang/home/models.py에서 확인 가능하다.

대부분은 과제 가이드라인을 기반으로 일부 수정했다.

수업 스키마의 name과 credit 어트리뷰트는 과목과 중복된다. 따라서 제거했다. 수업시간 스키마에서 begin과 end에서 요일 정보의 중복이 발생한다. 이를 day라는 별도의 어트리뷰트를 만들어서 중복을 제거했다. 수업 테이블에 현재 수강신청을 한 학생 수에 대한 어트리뷰트를 추가했다.

학생의 테이블에서 비밀번호를 암호화하지 않고 그대로 노출하고 있어서 password를 제거했다. PBKDF2 알고리즘으로 SHA256 해시를 적용해서 password를 암호화한 후 별로의 계정 정보(auth\_user) 테이블에 저장한다. 또한 실제 학교 계정에서 초기화되지 않은 계정의 아이디는 학번과 같지만 이후 학번과 다르게 변경할 수 있다. 따라서 로그인하는 아이디도 별도의 테이블에서 관리해야한다.

DB Insert 작업의 편의성을 위해 모델들의 의존 관계에 따라 우선순위(Tier)를 부여했다.

티어가 낮은 모델은 더 높은 티어의 모델에 의존적이다.

DB 삽입 명령을 수행할 때 높은 티어부터 실행해야한다.

Tier 1 Model: 전공, 건물, 과목

Tier 2 Model: 교강사, 강의실

Tier 3 Model: 학생, 수업

Tier 4 Model: 성적, 수업시간, +auth\_user

DB 구조를 생성하는 코드는 다음과 같다.

길이상 일부만 첨부했다.

**from** **django.db** **import** models

*# Create your models here.*

**from** **django.template.defaultfilters** **import** default

*# 티어가 낮은 모델은 더 높은 티어의 모델에 의존적이다.*

*# 즉, DB 삽입 명령을 수행할 때 높은 티어부터 실행해야한다.*

*# Tier 1 Model: 전공, 건물, 과목*

*# Tier 2 Model: 교강사, 강의실*

*# Tier 3 Model: 학생, 수업*

*# Tier 4 Model: 성적, 수업시간, +auth\_user*

*# ----------- Tier 1 ---------------*

*# Tier 1 Model: 전공, 건물, 과목*

**class** **Major**(models.Model):

*"""*

*major\_id,name*

*1,건설환경공학과*

*"""*

major\_id = models.IntegerField(primary\_key=**True**)

name = models.CharField(max\_length=100)

**class** **Building**(models.Model):

*"""*

*building\_id,name,admin,rooms*

*305,IT / BT,공과대학,19*

*"""*

building\_id = models.IntegerField(primary\_key=**True**)

name = models.CharField(max\_length=100)

admin = models.CharField(max\_length=100)

rooms = models.IntegerField()

**class** **Course**(models.Model):

*"""*

*course\_id,name,credit*

*CIE3022,철근콘크리트구조설계,3*

*"""*

course\_id = models.CharField(primary\_key=**True**, max\_length=20)

name = models.CharField(max\_length=100)

credit = models.IntegerField()

*# ----------- Tier 2 ---------------*

*# Tier 2 Model: 교강사, 강의실*

**class** **Lecturer**(models.Model):

*"""*

*lecturer\_id,name,major\_id*

*2001001001,조병완,1*

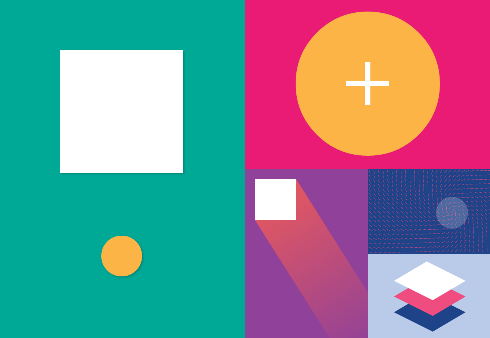
*"""*

lecturer\_id = models.IntegerField(primary\_key=**True**)

name = models.CharField(max\_length=100)

major = models.ForeignKey(Major, on\_delete=models.CASCADE)

# UI 디자인



구글의 Material Design 가이드라인을 따른다.



CSS 프레임워크인 Bootstrap을 기반으로 작업한다.



Hanyang Blue (#1D2475)를 대표색으로 사용한다. (ex. NavBar 배경색 등에 적용)

반응형 웹 기술을 통해 모바일 친화적인 UI를 구성한다.

(ex. 모바일 화면에서는 가로로 배치된 레이아웃이 세로로 변경된다.

Navbar의 요소들이 모바일 화면에서는 햄버거 버튼으로 축소된다.)

# Input data set

DB 삽입 작업에는 기존에 작성된 CSV 파일을 그대로 이용한다. 파이썬의 CSV 라이브러리를 이용해서 CSV 파일을 읽어온 후 현재 설계된 DB 스키마에 따라 데이터를 직접 파싱해 DB에 데이터를 삽입한다.

아래는 DB Insertion을 위해 CSV를 파싱하는 코드의 일부이다. 길이상 일부만 첨부했다.

Sugang/management/commands/insert\_test\_data.py 파일에서 전체를 확인할 수 있다.

**from** **django.core.management.base** **import** BaseCommand, CommandError

**from** **home.models** **import** \*

**import** **csv**

**class** **Command**(BaseCommand):

help = 'inserts test data set'

**def** handle(self, \*args, \*\*options):

print("insert major.csv")

**with** open('test\_data\csv\major.csv', 'r') **as** csvfile:

r = csv.reader(csvfile)

**for** row **in** r:

**if** row[0] == 'major\_id':

**continue**

major = Major()

*"""*

*major\_id,name*

*1,건설환경공학과*

*"""*

major.major\_id = row[0]

major.name = row[1]

major.save()

# 구현된 코드 설명

장고는 MVT 패턴을 따른다.

서버에 요청이 들어오면 url을 urls.py 파일에 따라 파싱하고 연결된 views.py의 함수로 보내고 views.py는 models.py를 통해 데이터베이스 관련 작업을 하고 유저가 원하는 데이터를 template에 적용해서 렌더링한 후 유저에게 응답을 보낸다. 그 외 자세한 사항들은 필요한 경우 코드에 주석이 있고 대부분의 코드들은 그 자체로 하는 일을 알 수 있고 복잡하지 않기 때문에 자세한 설명은 생략한다.

# 빌드 및 실행 과정

1. >>> git clone https://github.com/ClearSky-S/sugang.git
2. 파이썬을 설치한다. 이때 환경변수 등록을 꼭 해야한다. (설치 시 ADD PATH 옵션 체크)
3. >>>pip install django 명령어를 통해 장고를 설치한다.
4. .db와 같이 로컬에서 따로 설정해야 하는 파일들은 편의상 미리 만들어서 확장자 뒤에 .example을 붙여 놓았으니 빠르게 .example을 확장자 뒤에서 제거하고 사용하면 된다.
5. 기본 example DB를 사용할 경우 관리자 계정/비밀번호는 admin/3412 이다.
6. Db.sqlite3.example 대신 직접 만들고 싶다면 뒷부분을 이어서 따라하면 된다.
7. >>> python manage.py migrate (db 테이블 생성)
8. >>> python manage.py insert\_test\_data (csv를 파싱해서 데이터 삽입)
9. >>> python manage.py createsuperuser (관리자 계정 생성)

# 요구사항

1. 수강편람 기능

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

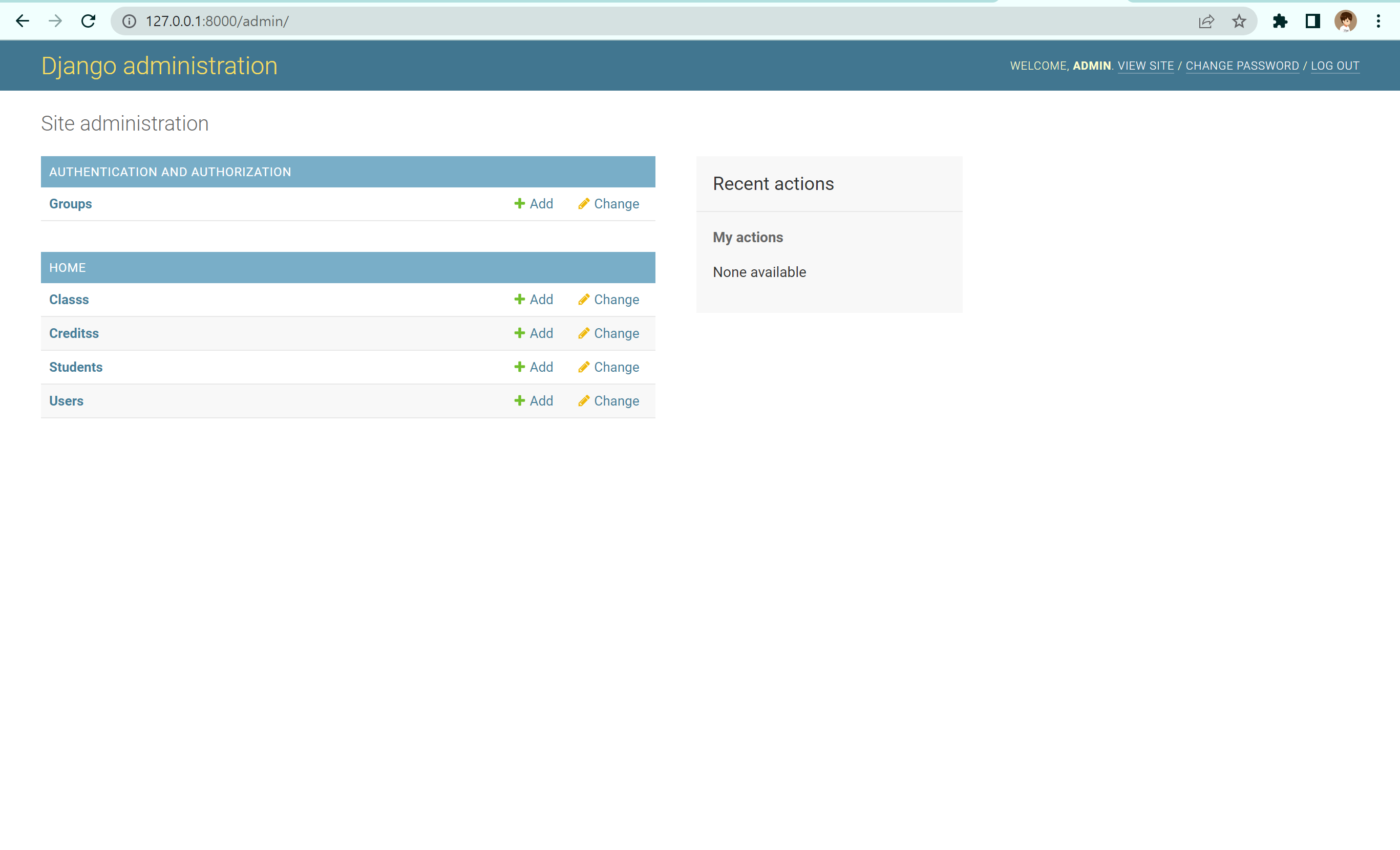
자동 생성된 설명



1. 관리자

텍스트이(가) 표시된 사진

자동 생성된 설명



주소창에 admin/timetable/학번 입력 시 학생의 시간표를 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

성적 변경 및 조회 페이지

텍스트이(가) 표시된 사진

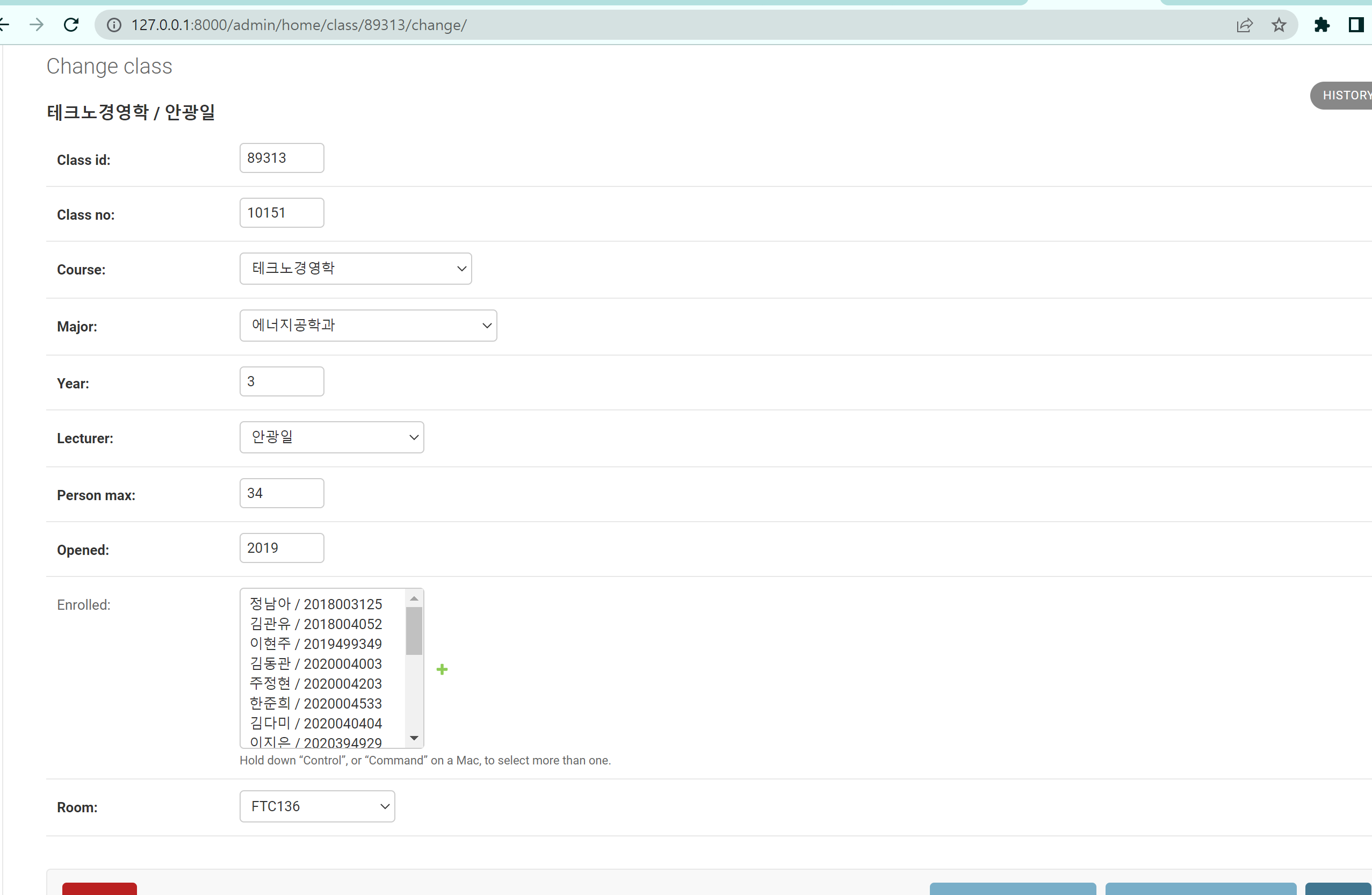
자동 생성된 설명

지도 교수 등 학생 정보 변경

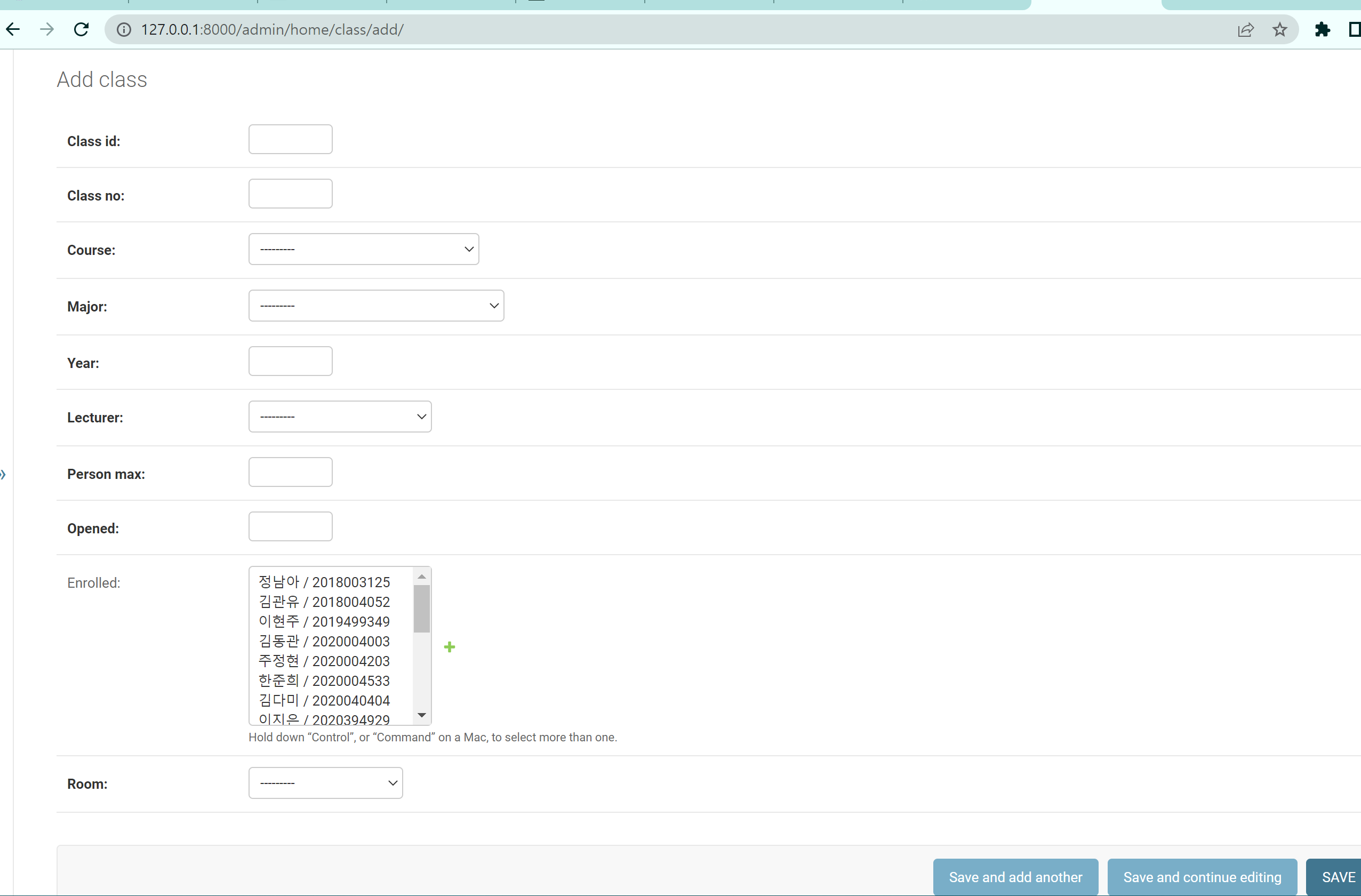
텍스트이(가) 표시된 사진

자동 생성된 설명

과목 정보 조회 및 변경



과목 설강



OLAP

테이블이(가) 표시된 사진

자동 생성된 설명

로그인/로그아웃

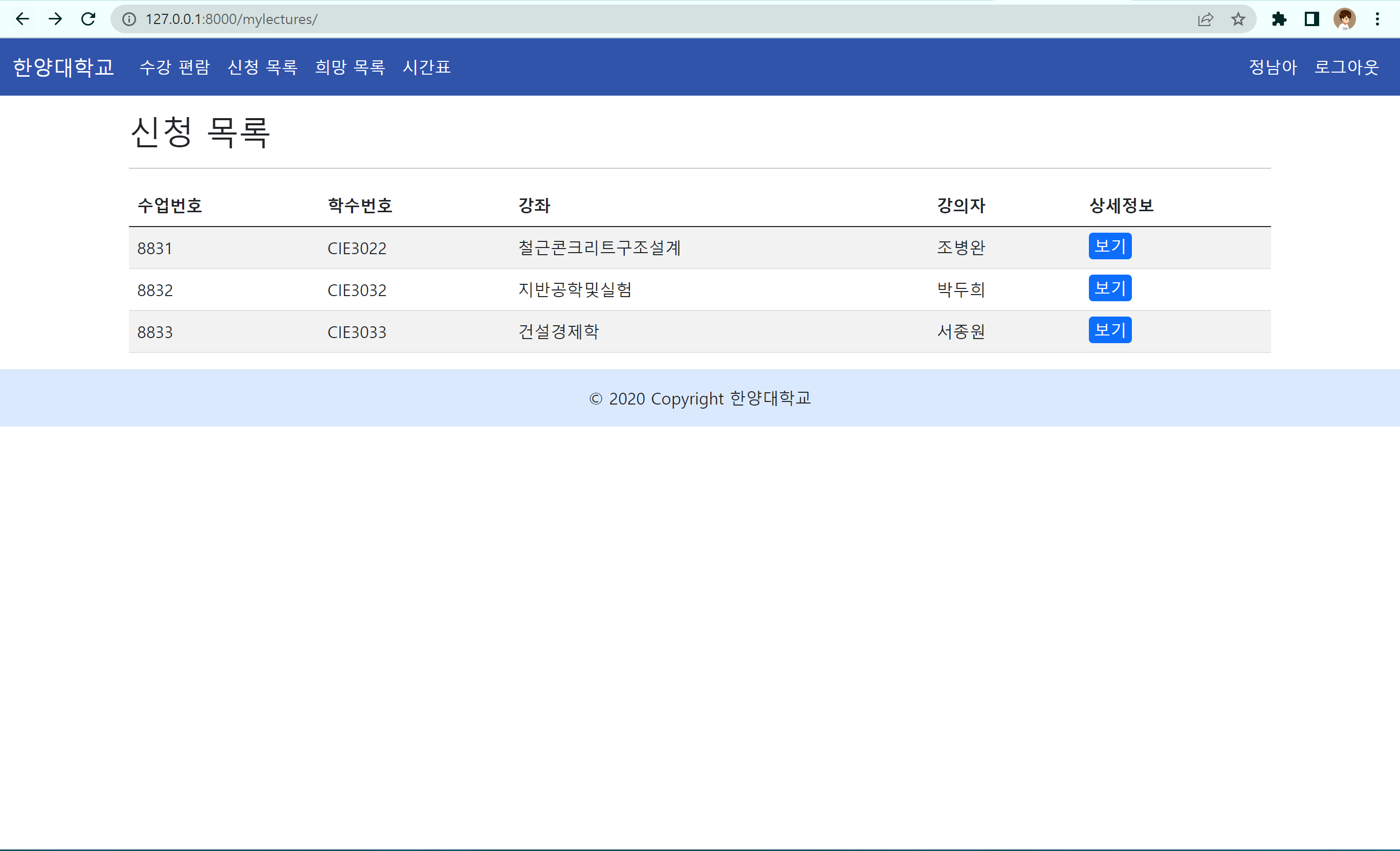
텍스트이(가) 표시된 사진

자동 생성된 설명

수강신청 및 취소

테이블이(가) 표시된 사진

자동 생성된 설명





희망수업

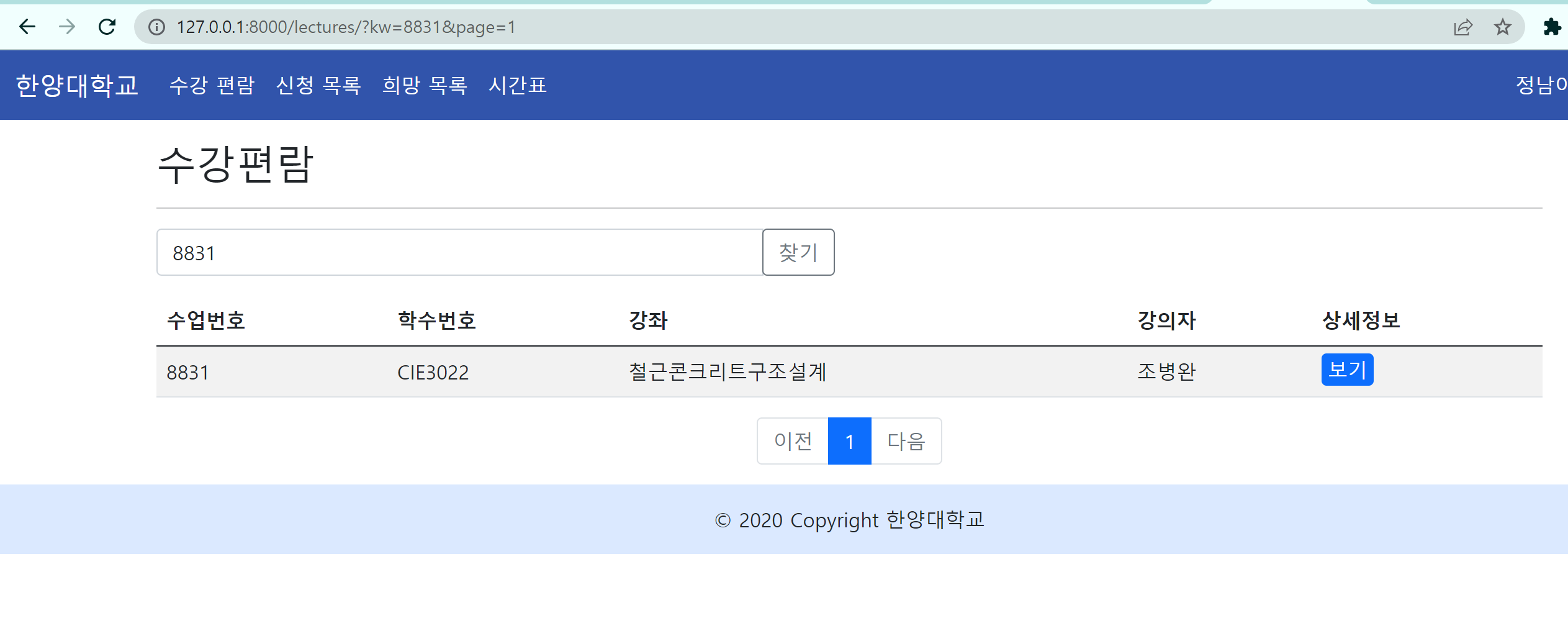
텍스트이(가) 표시된 사진

자동 생성된 설명

시간표 생성



검색기능

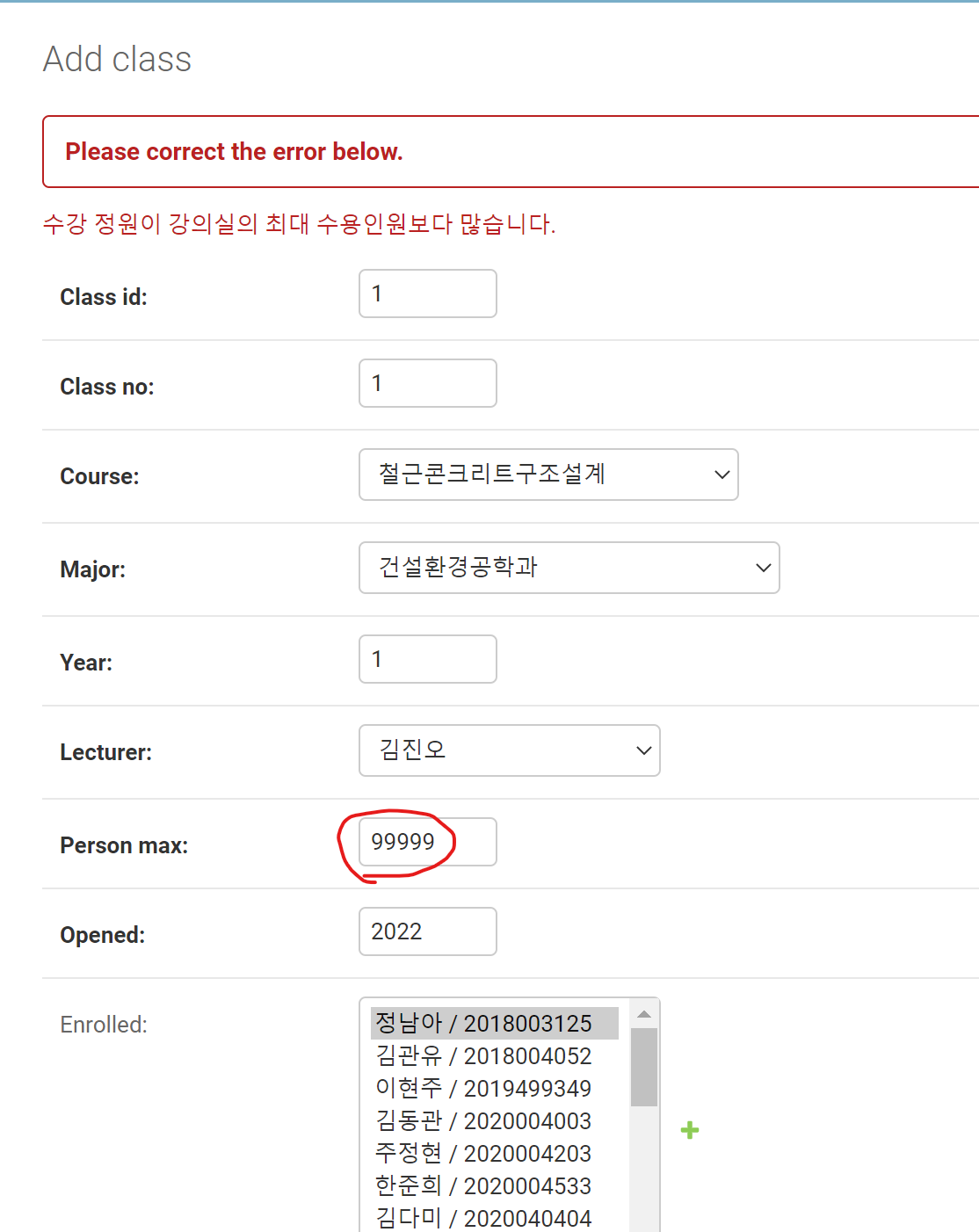




텍스트이(가) 표시된 사진

자동 생성된 설명

관리기능1- 수강정원, 강의실 수용 인원



관리기능2-과목삭제, cascade

수강신청한 기록 모두 삭제

아래 이미지에서 Class-student relationship이 강의를 신청한 학생을 뜻함.

텍스트이(가) 표시된 사진

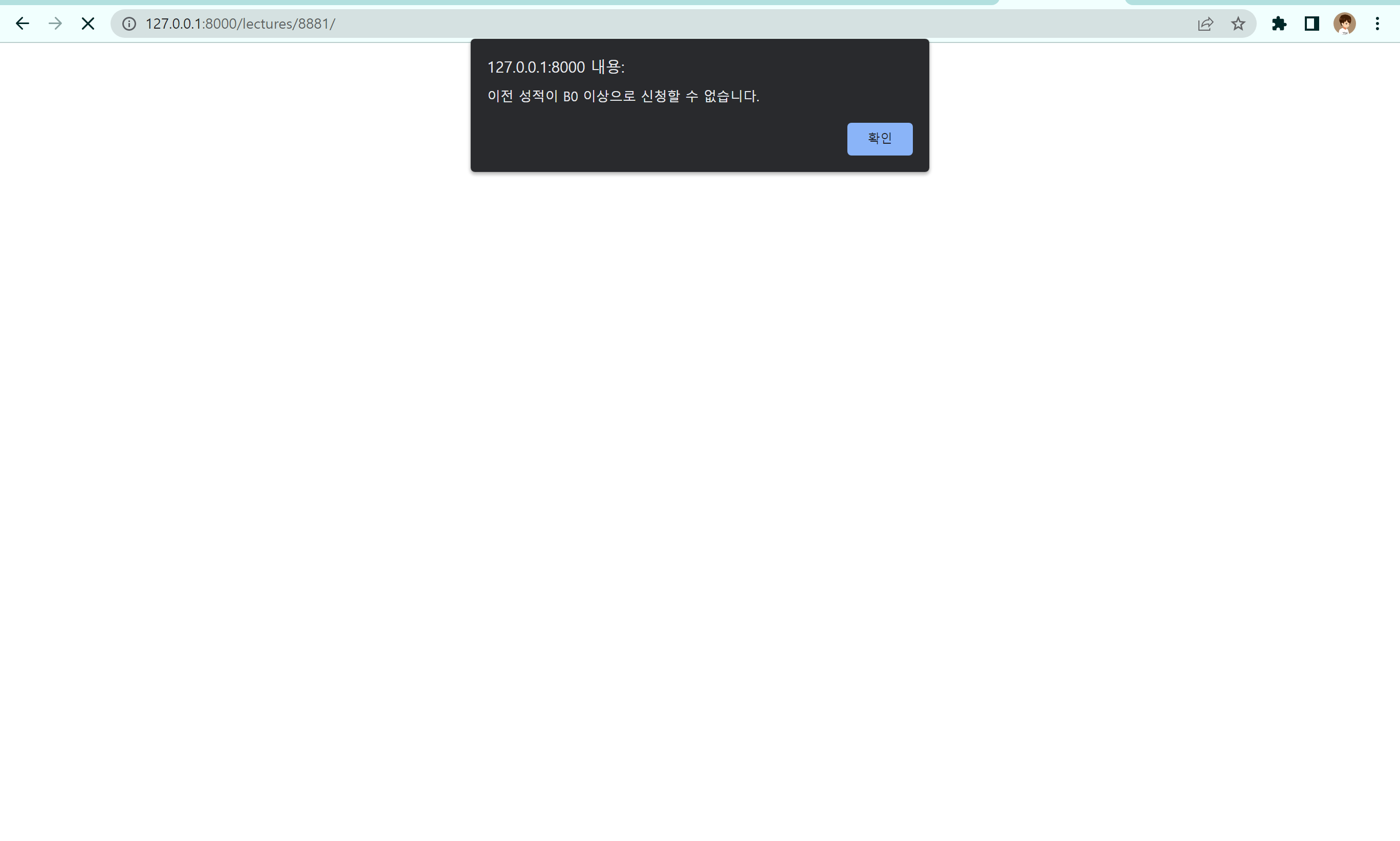
자동 생성된 설명

관리 기능3- 주말 수업은 온라인으로



수강신청1. 성적 높으면 재수강 불가





수강신청 기능 2- 정원 초과시 신청 불가

텍스트이(가) 표시된 사진

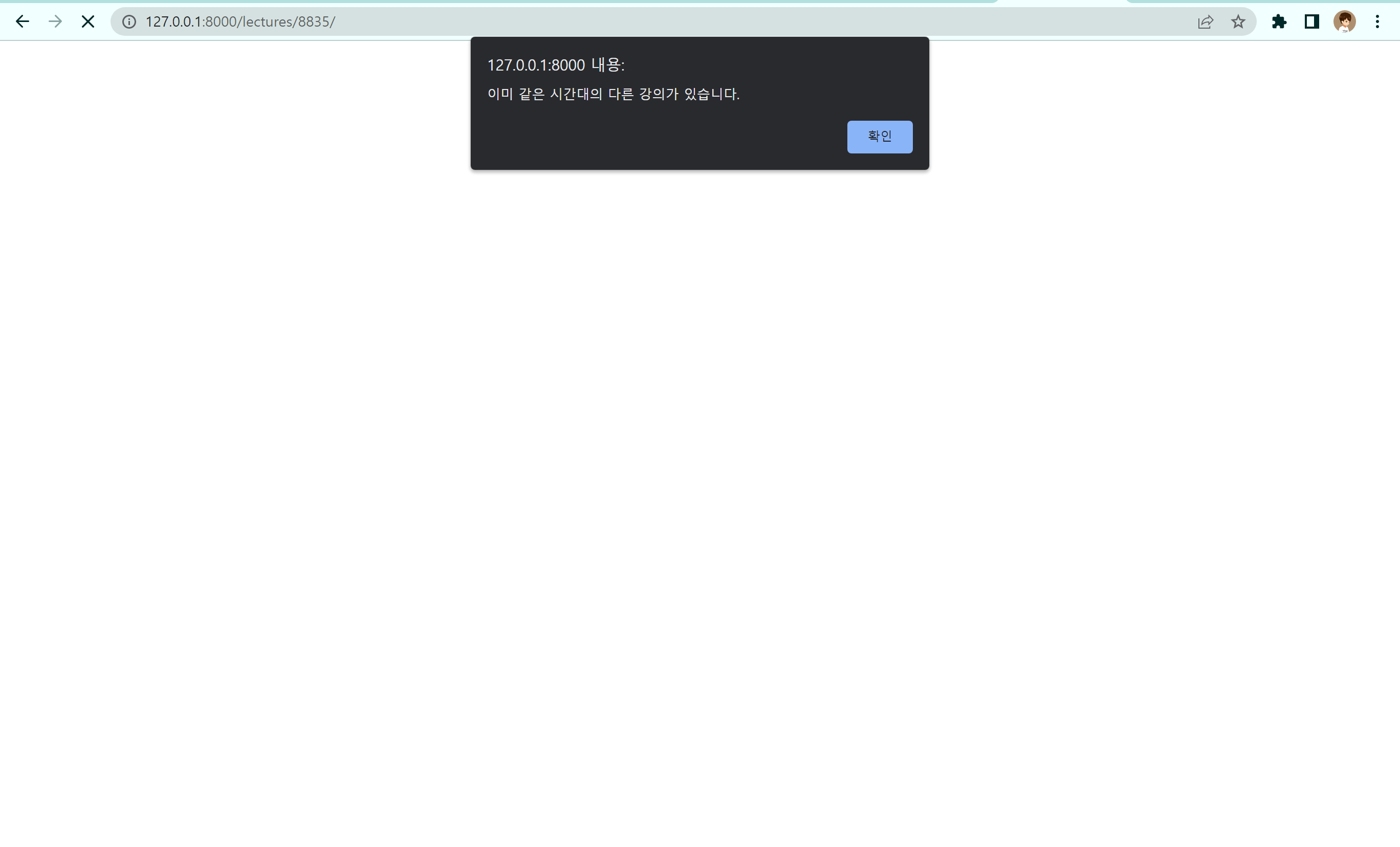
자동 생성된 설명



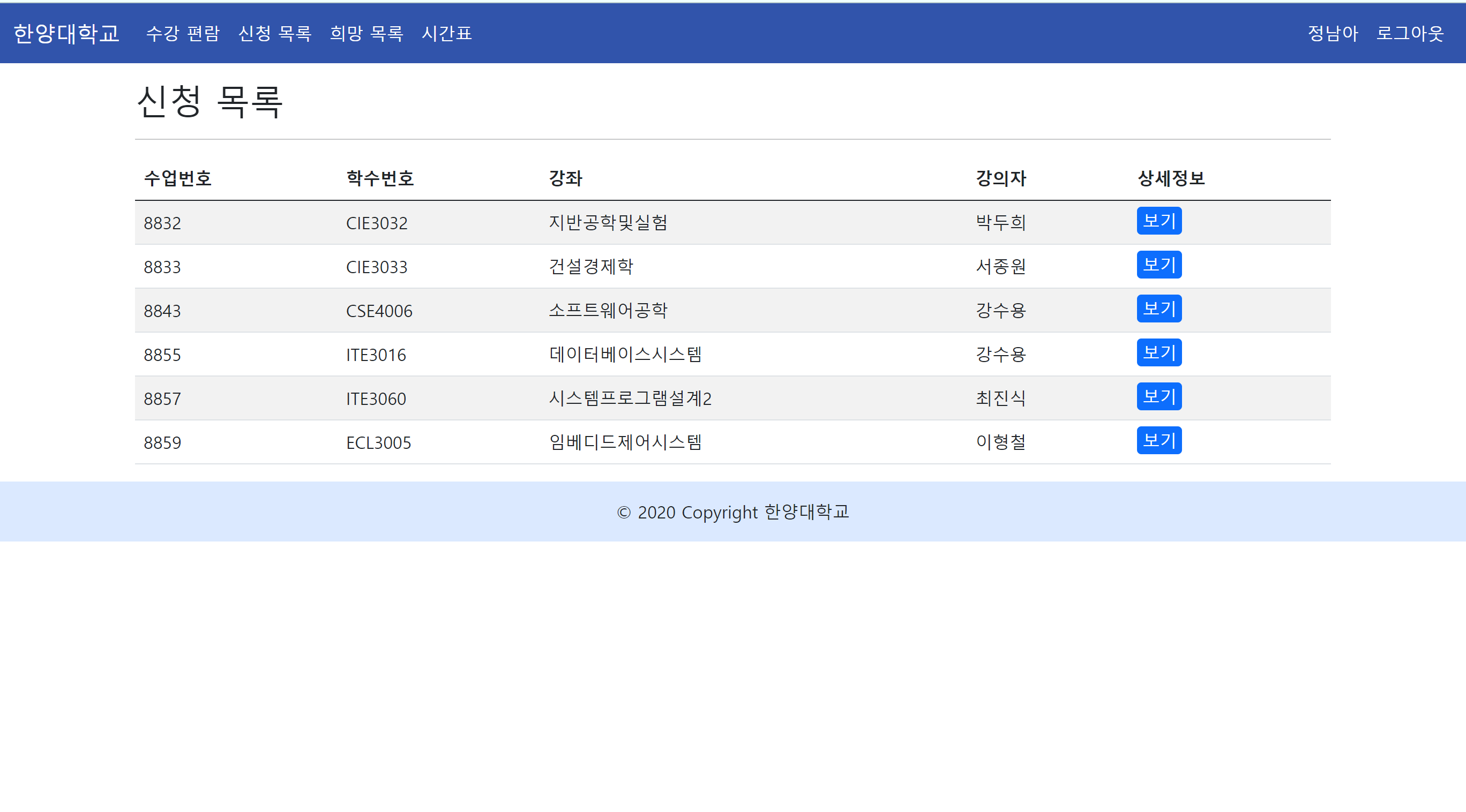
수강신청3- 겹치는 시간 신청 불가





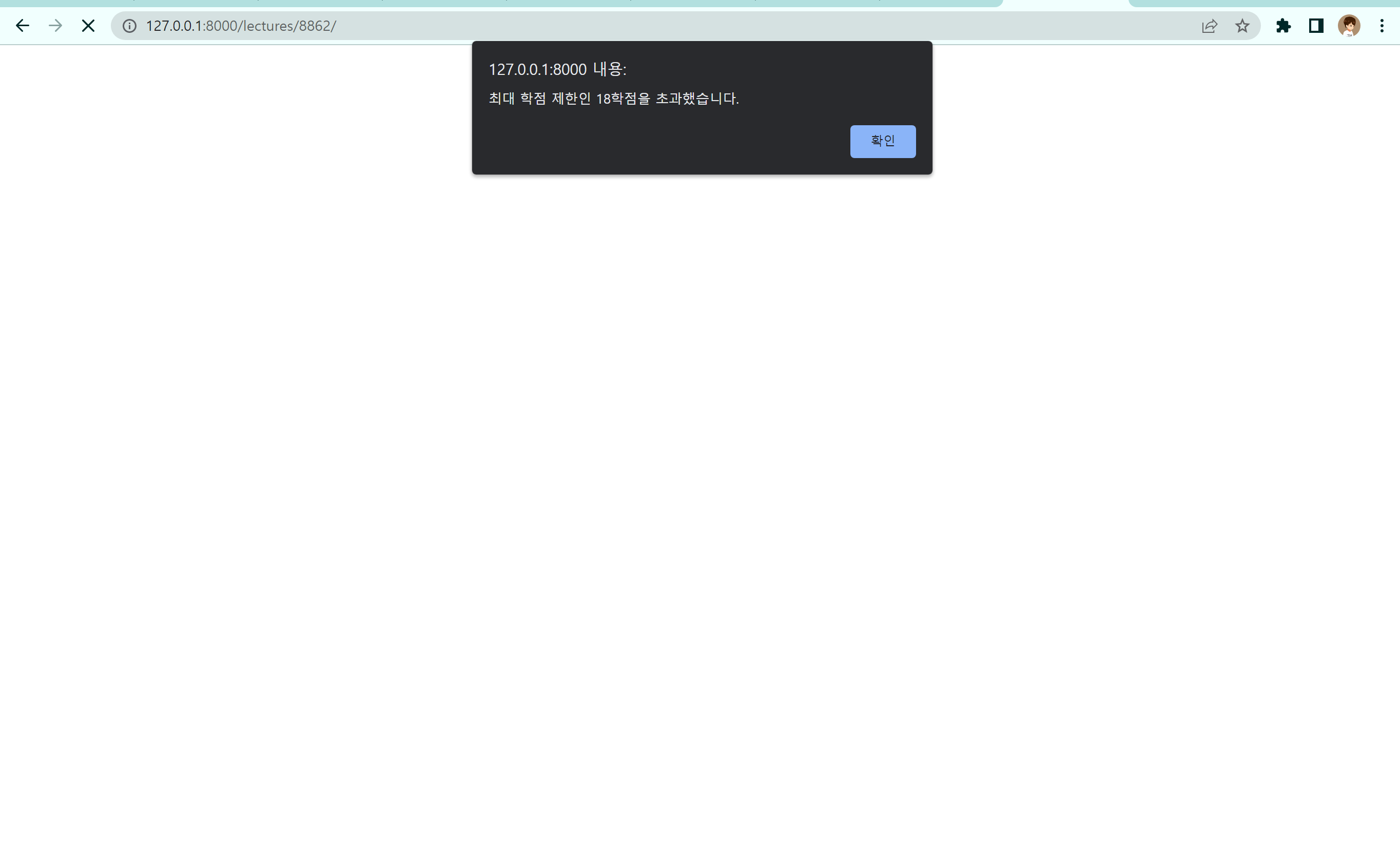


수강신청4- 최대 18학점



테이블이(가) 표시된 사진

자동 생성된 설명



학생시간표



관리자 시간표

주소창에 admin/timetable/학번 입력 시 학생의 시간표를 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

# 추가 기능

없음.

# URL 설계

// 실제 구현은 sugang/urls.py와 home/urls.py 에 있다.

path('', views.home, name='home'), # 메인페이지, 로그인 X일 경우 로그인 페이지로 리다이렉트 됨

path('login/', auth\_views.LoginView.as\_view(template\_name='home/login.html'), name='login'), # 로그인

path('logout/', auth\_views.LogoutView.as\_view(), name='logout'),

path('user/', views.home, name='user'), # 사용자 정보 조회 및 수정/ 미구현

path('lectures/', views.lectures, name='lectures'), # 강의 목록 조회 및 수강신청

path('lectures/<int:class\_id>/', views.enroll, name='enroll'), # 강의 정보 및 신청

path('mylectures/', views.mylectures, name='mylectures'), # 신청한 강의 확인

path('wishlist/', views.wishlist, name='wishlist'), # 신청한 강의 확인

path('wishlist/<int:class\_id>/', views.wishEnroll, name='wishEnroll'),

path('timetable/', views.timetable, name='timetable'), # 시간표 확인

path('cancel/<int:class\_id>/', views.cancel, name='cancel'), # 강의 취소 post method 만 허용

path('admin/statistics/', views.statistics, name='statistics'), # 분석 정보 제공

path('admin/timetable/<int:student\_id>/', views.timetable, name='admin/timetable'), # 분석 정보 제공

# 문제 해결

DB의 예시 데이터로 CSV 가 주어져 있는데 이것을 사람이 일일이 보면서 SQL 삽입 명령을 치는 것은 지루한 일이다. 파이썬 CSV 라이브러리를 통해서 CSV 파일을 읽어오고 파싱해서 자동으로 DB에 데이터를 삽입하는 코드를 작성했다.

시간표라고 하면 에브리타임이나 고등학교학교 시간표 같은 정석적인 시간표가 고정관념적으로 생각난다. 처음에는 시간표 자바스크립트 라이브러리를 사용하려고 했는데 생각해보면 그러한 형태의 시간표를 꼭 고수할 필요가 없다. 그러한 시간표는 낭비되는 공간이 많고 목, 금요일과 같이 오른쪽에 위치한 날짜의 시간은 왼쪽의 시간 레이블과 거리가 멀어서 정확한 시간이 헷깔린다. 그래서 아래와 같이 수업이 있는 시간만 표시하는 시간표를 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명