

# kweb 면제 과제

📅 Date	@2024년 3월 10일
☰ Tags	

## Kweb : 간단한 온라인 교육 플랫폼 제작



### 제출 사항

- 필수 구현 5개
- 선택 구현 : 학생 (최근 업데이트)
- 선택 : 리포트

## 도메인 모델링

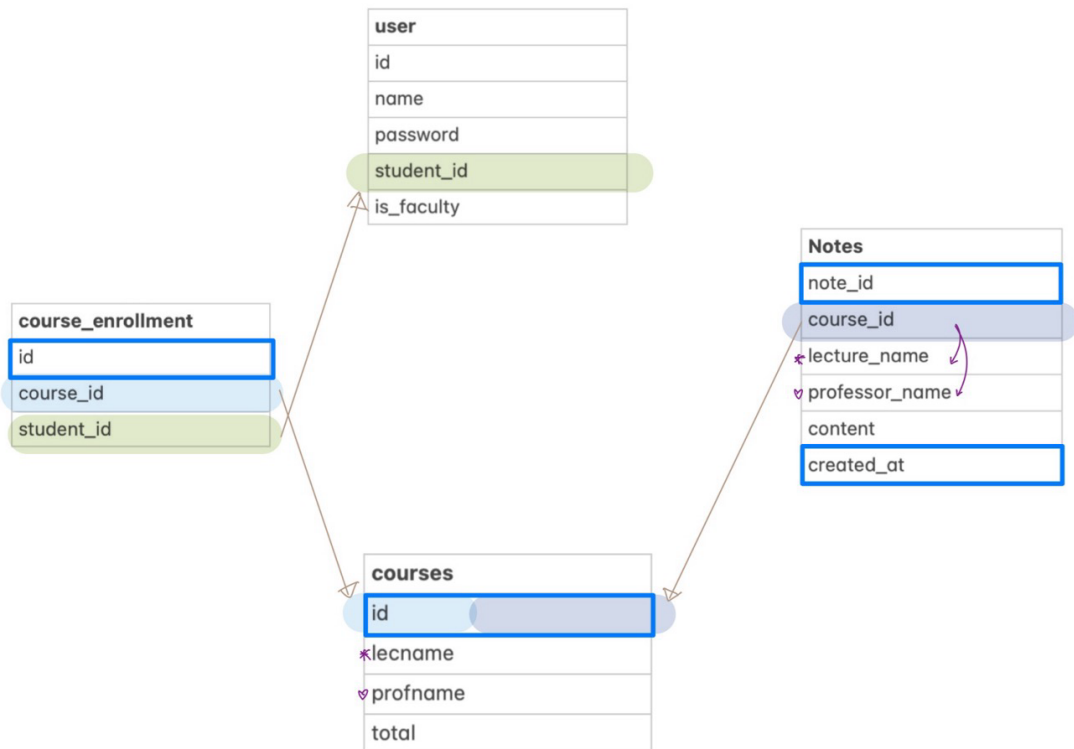
### entity 설명

- 회원 (user)
  - id, password, name, student\_id, is\_faculty(학생/교수 여부 : 1이면 교수, 0이면 학생) 정보가 저장된다.
  - 교수는 여러 강의를 할 수 있고, 학생은 여러 강의를 들을 수 있다. (user와 courses는 1대다 관계이다.)
  - 교수는 강의하는 과목에서 여러 게시글을 만들 수 있다. 학생은 수강하는 과목에서 여러 게시글을 확인할 수 있다. (user와 Notes는 1대다 관계이다.)
    - 이때 '수강하는 과목' 정보를 관리해야 한다. [course\_enrollment] 동일한 게시글을 해당 과목을 듣는 모든 학생들이 읽을 수 있다. (일대다 다대일 관계를 구성)
- courses
  - 강의 번호, 강의 제목, 교수자 이름, 최대 수강 신청 인원 수가 저장된다.
  - 강의마다 게시물 여러 개 작성이 가능하니, courses와 Notes의 관계는 1대다 관계이다.
- Notes

- 노트 번호, 강의 번호, 강의 제목, 교수자 이름, 내용, 생성 일시가 저장된다.

## DB 스키마 구조 + 모델링 기타 사항

- 외래키 설정을 통해, 해당 정보들을 함부로 삭제할 수 없게하고, 매핑관계의 명확성을 밝혔다.

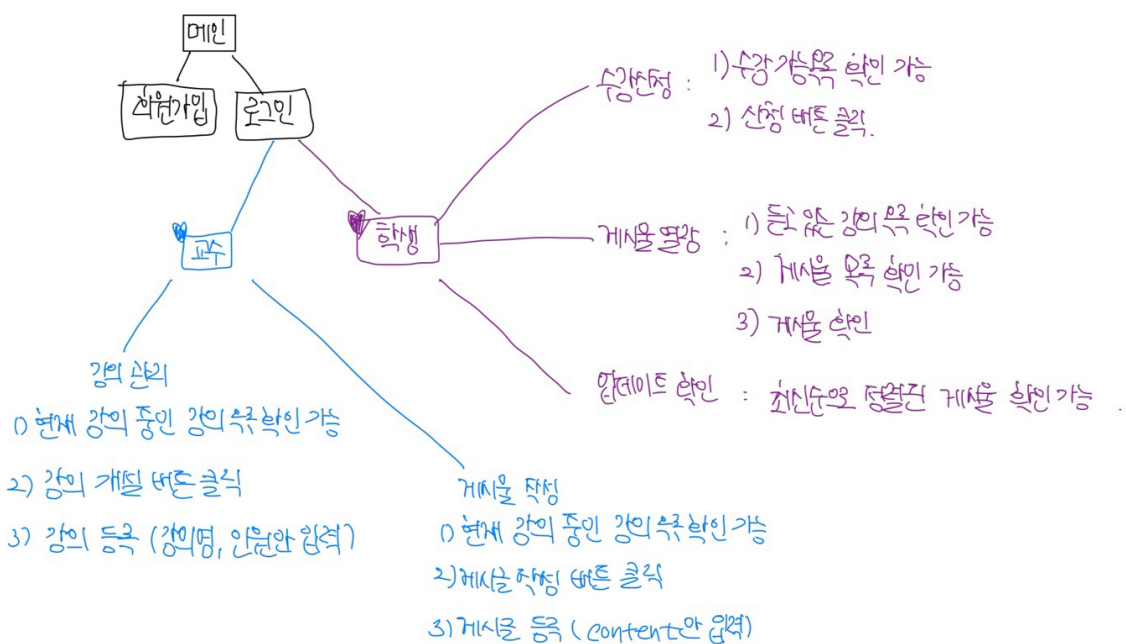


- 4가지 종류의 데이터베이스 테이블.
- 형광펜, 화살표가 외래키 설정임을 보인다. 화살표로 찢린 쪽이 원래 주인이라고 볼 수 있게 그림을 그림을 그려두었다.
  - 파란색 박스로 그려둔 쪽은, 자동으로 생성되는 값들이다. (1씩 증가하는 정수이거나, 타임 스탬프 자동으로 생성하는 종류이다.)
  - Notes를 생성할 시, 해당 course\_id에 따른 lecture\_name과 professor\_name은 데이터베이스에 접근하여 정보를 읽어낸 후 값을 가져와 저장하는 형식이다. (해당 포맷에 맞춰 자동으로 가져올 값이기 때문에 보라색으로 그림에서는 표시해두었다.)

## 구현

- 개발 언어 : javascript
- framework : node.js + bootstrap
- database : mysql

## 웹 애플리케이션 사이트 맵



중간에 오류가 나거나, 동작을 다했을 시 '처음으로 돌아가기'버튼을 누르거나 혹은 자동으로 하트가 그려져있는 페이지가 보이게끔 설정해두었습니다.

## API 명세



lib\_login (회원가입, 로그인과 관련된 구현들)

### auth.js

#### 1. 로그인 API

- **엔드포인트:** `/auth/login`
- **메서드:** GET
- **요청:** 사용자의 아이디와 비밀번호를 포함하는 POST 요청
- **응답:** 로그인에 성공하면 세션에 사용자 정보를 저장하고, 메인 페이지(로그인 된 계정에 맞는 선택 가능 페이지)로 리다이렉트됩니다. 로그인에 실패하면 오류 메시지를 표시하고 로그인 페이지로 리다이렉트됩니다.

## 2. 로그인 프로세스 API

- **엔드포인트:** `/auth/login_process`
- **메서드:** POST
- **요청:** 사용자의 아이디와 비밀번호를 포함하는 POST 요청
- **응답:** 로그인에 성공하면 세션에 사용자 정보를 저장하고 메인 페이지로 리다이렉트됩니다. 로그인에 실패하면 오류 메시지를 표시하고 로그인 페이지로 리다이렉트됩니다.

## 3. 회원가입 API

- **엔드포인트:** `/auth/register`
- **메서드:** GET
- **요청:** 사용자의 아이디, 비밀번호, 이름, 학번, 교수 여부를 포함하는 POST 요청
- **응답:** 회원가입에 성공하면 로그인 페이지로 리다이렉트됩니다. 이미 가입된 아이디나 학번인 경우 오류 메시지를 표시하고 회원가입 페이지로 리다이렉트됩니다.

## 4. 회원가입 프로세스 API

- **엔드포인트:** `/auth/register_process`
- **메서드:** POST
- **요청:** 사용자의 아이디, 비밀번호, 이름, 학번, 교수 여부를 포함하는 POST 요청
- **응답:** 회원가입에 성공하면 메인 페이지로 리다이렉트됩니다. 이미 가입된 아이디나 학번인 경우 오류 메시지를 표시하고 회원가입 페이지로 리다이렉트됩니다.



professor (교수자 권한 접속 시 실행 가능 기능 구현)

pr.js

## 1. 교수 권한 접속 API

- 엔드포인트: `/professor/select`
- 메서드: GET
- 요청: 없음
- 응답: 교수의 이름을 표시하고(제대로된 접속 확인 용도), 강의 관리와 강의 게시물 작성을 위한 링크를 포함한 페이지를 반환합니다.
- +) 교수자 권한으로 접속한 경우, 여기서 보이는 페이지가 메인 페이지입니다.

## 2. 강의 관리 API

- 엔드포인트: `/professor/lecture_control`
- 메서드: GET
- 요청: 없음
- 응답: 해당 교수가 강의하는 강의의 목록을 표시한 페이지를 반환합니다.

## 3. 강의 개설 페이지 API

- 엔드포인트: `/professor/make_lecture`
- 메서드: GET
- 요청: 없음
- 응답: 강의 개설을 위한 폼이 있는 페이지를 반환합니다.
- +) 강의명, 최대 인원을 입력 받는 기능을 합니다.

## 4. 강의 개설 API

- 엔드포인트: `/professor/make`
- 메서드: POST
- 요청: 강의명( `lecname` ), 최대 인원( `total` ), 교수 이름( `name` )을 포함하는 POST 요청
- 응답: 성공적으로 강의를 등록되면 알림 메시지를 표시하고 메인 페이지로 리다이렉트됩니다. 이미 만들어진 강의인 경우 오류 메시지를 표시하고 이전 페이지로 리다이렉트됩니다.
- +) 이때, 입력한 강의 명과 최대 인원을 사용하며, 교수 이름의 경우 session에 저장된 정보를 계속해서 가져옵니다.

## 5. 강의 게시물 작성 페이지 API

- **엔드포인트:** `/professor/write`
- **메서드:** POST
- **요청:** 없음
- **응답:** 교수가 강의하는 강의의 목록과 게시물 작성을 위한 폼이 있는 페이지를 반환합니다.
- +) 게시물 내용만 입력으로 받습니다.

## 6. 게시물 작성 API

- **엔드포인트:** `/professor/write_enter`
- **메서드:** POST
- **요청:** 강의 ID( `course_id` ), 강의명( `lecname` ), 교수명( `profname` ), 내용( `content` )을 포함하는 POST 요청
- **응답:** 성공적으로 게시물이 등록되면 알림 메시지를 표시하고 게시물 선택 페이지로 리다이렉트됩니다. 오류가 발생한 경우 오류 메시지를 표시하고 이전 페이지로 리다이렉트됩니다.
- +) `course_id`에 맞춰 `lecname`, `profname`은 자동으로 입력됩니다.



student (학생 권한 접속 시 실행 가능 기능 구현)

### 1. 학생 권한 접속

- **엔드포인트:** `/student/select`
- **메서드:** GET
- **요청:** 없음
- **응답:** 학생의 이름과 선택할 수 있는 작업 목록이 있는 페이지를 반환합니다.
- +) 해당 페이지가 학생 접속의 메인 페이지입니다.

### 2. 강의 수강 신청

- **엔드포인트:** `/student/sugang_preprocess`
- **메서드:** GET
- **요청:** 없음

- **응답:** 강의 수강 신청을 위한 페이지를 반환하고, 수강 가능한 강의 목록을 표시합니다.

### 3. 수강 신청 프로세스

- **엔드포인트:** `/student/sugang_process`
- **메서드:** POST
- **요청:** 학생 ID와 수강하려는 강의 ID를 포함하는 POST 요청
- **응답:** 수강 신청이 완료되면 알림을 표시하고 학생 메인 페이지로 리다이렉트됩니다.

### 4. 강의 게시물 열람

- **엔드포인트:** `/student/see`
- **메서드:** POST
- **요청:** 없음
- **응답:** 학생이 수강 중인 강의 목록을 가져와서 보여줍니다. 수강 중인 강의가 없을 경우 적절한 메시지를 표시합니다.

### 5. 강의 게시물 조회

- **엔드포인트:** `/student/see_lecture`
- **메서드:** POST
- **요청:** 선택된 강의의 ID를 포함하는 POST 요청
- **응답:** 선택된 강의의 게시물 목록을 가져와서 보여줍니다. 게시물이 없는 경우 적절한 메시지를 표시합니다.

### 6. 게시물 상세 조회

- **엔드포인트:** `/student/lecture`
- **메서드:** POST
- **요청:** 선택된 게시물의 ID를 포함하는 POST 요청
- **응답:** 선택된 게시물의 내용을 보여줍니다.

### 7. 강의 게시물 업데이트 확인

- **엔드포인트:** `/student/update`
- **메서드:** POST
- **요청:** 없음

- **응답:** 학생이 수강 중인 강의의 게시물이 업데이트된 내용을 보여줍니다. (선택 구현) 업데이트된 게시물이 없을 경우 적절한 메시지를 표시합니다.



views (랜더링 용 ejs 파일들을 모아놓은 것)

- 적절한 값을 입력받거나, 버튼의 조작을 통해 원하는 구현의 함수를 실행시키기 좋게 만들어놓았다.
- listing 하는 게 많은 경우 (table구경) 의 구현을 주로 만들어 놓았다.