면제 과제 리포트

2024320147 최민석

목차

1. 도메인 모델링
2. DB 스키마 구조
3. 웹 애플리케이션 사이트 맵
4. API 명세
5. 모델링에 대한 기타 사항

도메인 모델링

본 과제를 통해 구현한 기능은 필수 과제로 등록된 다섯 가지 기능으로 다음과 같다.

1. 로그인

* 아이디, 비밀번호, 이름, 학번, 학생/교수 선택

1. 교수자의 강의 관리

* 새로운 강의 등록
* 등록된 강의 조회

1. 강의 게시물 작성

* 서식이 있는 텍스트로 작성할 수 있어야 함

1. 학생 수강 신청

* 강의 목록 조회
* 조회된 강의를 수강 신청

1. 신청된 강의 게시물 열람

* 신청한 강의, 게시물 목록 조회
* 게시물 내용 조회

각 기능은 다음과 같은 조건이 필요하다.

1. 로그인

* 조건을 입력하는 회원 가입 구현 필요
* 회원 정보를 저장할 수 있어야 함
* 비밀번호를 암호화하여 저장하여야 함
* 로그인 시 저장된 회원 정보를 대조하고 불러올 수 있어야 함

1. 강의 관리

* 교수로 등록된 회원만 가능
* 새로운 강의를 등록하고 이를 저장할 수 있어야 함
* 목록을 조회할 때 회원의 강의 정보를 불러올 수 있어야 함

1. 게시물 작성

* 제목, 본문 등을 서식이 있는 텍스트로 작성할 수 있어야 함
* 조회하였을 때 rich text를 불러올 수 있어야 함
* 게시물 정보를 저장하여야 함

1. 수강 신청

* 학생으로 등록된 회원만 가능
* 저장된 정보에서 교수와 강의 목록을 불러올 수 있어야 함
* 수강 신청 후 신청 정보를 저장할 수 있어야 함
* 중복 신청을 감지할 수 있어야 함

1. 게시물 열람

* 조회하였을 때 rich text를 불러올 수 있어야 함
* 알맞은 교수, 강의, 게시물을 찾아 불러와야 함

이를 간단히 그림으로 나타내면 다음과 같다.

|  |
| --- |
|  |

대략적인 내부 로직은 다음과 같다.

1. 서버를 실행하면 main.html이 열린다.
2. 회원 가입 시 여러 조건을 입력할 수 있으며, 가입과 동시에 회원 정보(info.json)와 폴더, 그 아래의 g.json이 생성된다.
3. 교수자의 경우 강의, 게시물을 만들 수 있다. 만드는 도중 정보는 localstorage에 임시 저장되었다가 info.json과 g.json에 최종 저장된다.
4. 학생의 경우 수강 신청 및 조회가 가능하다. 도중의 정보는 localstorage에 임시 저장되며 완료 시 g.json에 반영된다.

DB 스키마 구조

기본적으로 모든 회원 정보는 json 형식으로 저장한다. 회원 정보는 교수자, 학생 관계없이 다음과 같은 파일 경로로 저장된다.

Info

ㄴinfo.json

ㄴ(회원 id)

ㄴg.json

Info.json은 회원의 정보를 저장하는 부분이다. 각 요소는 다음과 같이 저장된다.

{

"id": "회원의 id",

"password": “회원의 암호화된 비밀번호",

"name": "회원의 이름",

"number": "회원의 학번",

"student": 회원이 학생이면 true 아니면 false,

"g": [

"등록한 강의 목록 (학생의 경우 공란)"

]

}

g.json은 교수자와 학생이 다르다. 우선 교수자의 경우 다음과 같다.

{

"title": "강의 이름",

"inside": [등록된 게시물 이름],

"insideText": [등록된 게시물의 내용]

}

학생의 경우는 다음과 같다.

{

"title": "학생 id",

"inside": “수강 신청한 강의 이름”,

"insideText": ‘위 강의의 교수자’

}

웹 애플리케이션 사이트 맵

사이트 맵의 경우 node.js의 sitemap을 이용하여 구성하였다. 사이트 맵과 구성 코드는 g ithub 과제 제출란에 추가하였다.

API 명세

다음은 api의 기능, 입출력 등의 정보이다. 아래와 같다.

1. /sign\_up

* Post
* id, password, name, number, student 받음
* message(가입 성공) return
* 회원 정보 폴더, 파일 등 생성

1. /login

* Post
* Id, password 입력
* Return message(로그인 성공), id, name, student여부

1. /stu

* Post
* Id 입력
* Return message(성공), 교수자 목록

1. /stun

* Post
* Id, 선택된 교수 id 입력
* Return message, 교수의 강의 목록

1. /stunn

* Post
* Input id, 교수자 id, 강의 title
* Return message
* 신청한 강의를 파일에 저장

1. /stuu

* Post
* Id 입력
* Return message, 신청한 강의와 교수자

1. /stuuu

* Post
* 교수자 id, 강의 title 입력
* Return message, 게시글 목록

1. /stuuuu

* Post
* Input 교수자 id, 강의 title, 게시글 title
* Return 게시글 내용

1. /pro

* Post
* Input id
* Return message, id, name, student 여부, 등록한 강의 목록

1. /pron

* Post
* Input id, 등록할 강의 title
* Return id, name, student 여부, 강의 목록
* 등록한 강의 정보를 파일에 저장

1. /pronn

* Post
* Input 강의 title, id
* Return message, 선택된 강의 title, 게시글 title, 게시글 내용

1. /pronnn

* Post
* Input id, 강의 title, 게시글 title, 내용
* Return message
* 입력된 게시글의 내용과 제목을 파일에 저장

1. /pronnnn

* Post
* Input id, 강의 title, 게시글 title
* Return 게시들 내용

모델링에 대한 기타 사항

앞으로 구현하거나 보완할 수 있는 점을 설명하고자 한다.

1. 선택 구현 목록(수강 학생 관리, 최근 업데이트, 실제 환경 배포)
2. 이름이 같은 교수, 학생에 대한 표기 처리
3. 비밀번호에 pepper 추가로 보안 강화
4. 각종 디자인을 더 직관적으로

이 외에도 다양한 개선점이 있을 것이다.