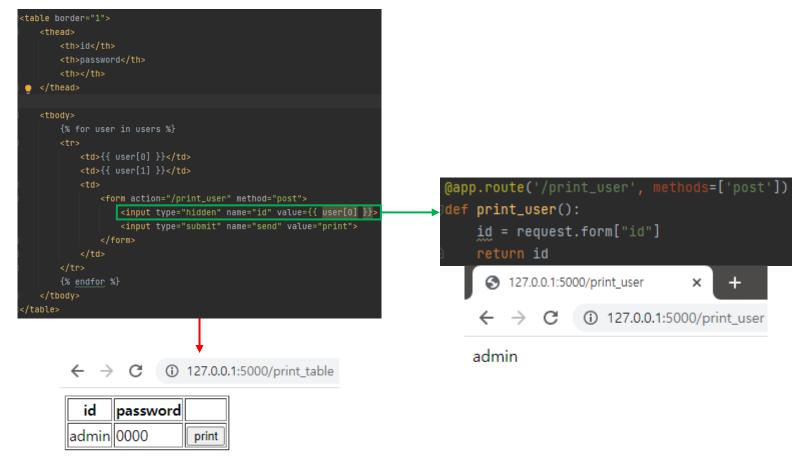


DB Application Programming

- Term Project -

A Useful Skill: 'hidden field'

- <form> tag can be placed in tag
- type="hidden" in <input> tag allows to put a value in a variable without displaying it on a webpage





DB Application Programming

- Term project의 목표는 **인터넷 강의 사이트**의 제작
- 테이블은 제공하는 파일(term.sql)을 기본으로 사용하되, 이후 설명할 본인만의 추가 기능 구현에 필요하다면 테이블의 추가나 수정을 허용함
- 에러가 있어 실행이 불가능한 코드를 제출할 경우 0점 처리
- Term project는 기본 과제의 3배의 배점을 가짐(30점)
 - 기본 기능에 대한 코딩(10점)
 - 본인만의 추가 기능 코딩(10점)
 - 보고서(10점)

Table Description(term.sql)

```
create table users
(
id varchar(15) not null,
password varchar(20) not null,
primary key (id)
);
```

```
users 테이블
```

id: 사용자의 id

password: 사용자의 password

```
create table subject
(
    code varchar(2) not null,
    subject_name varchar(20) not null,
    primary key (code)
);
```

```
■ subject 테이블
```

code: 과목 코드

• subject_name: 강의 과목명

```
create table lecture
(

code varchar(2) not null,
name varchar(20) not null,
price int not null check (price>=0),
tutor varchar(15) not null,
primary key (code, name, price, tutor),
foreign key (code) references subject (code),
foreign key (tutor) references users (id)
);
```

■ lecture 테이블

• code: 강의 과목 종류 식별 코드

• name: 강의의 이름

• price: 강의의 금액

• tutor: 강사 id



Table Description(term.sql)

```
create table enrollment (

tutor varchar(15) not null,
tutee varchar(20) not null,
name varchar(2) not null,
code varchar(2) not null,
lecture_price int null check (lecture_price>=0),
foreign key (tutor) references users (id),
foreign key (tutee) references users (id),
foreign key (code) references subject (code)
);
```

■ enrollment 테이블

• tutor: 강사 id

• tutee: 학생 id

▶ code: 강의 과목 종류 식별 코드

● lecture_price: 강의 등록 금액

name: 강의명

```
create table rating_info (
    rating varchar(10) not null,
    condition int not null check (condition>=0),
    discount numeric(4,2) not null check (100>discount and discount>=0),
    primary key (rating)
);
```

■ rating_info 테이블

• rating: 사용자의 등급

 condition: 등급의 조건(credit이 조건보다 크면 해당 등급을 부여)

• discount: 할인율(%)

```
create table account
(

id varchar(15) not null,
credit int not null check (credit>=0),
rating varchar(10) not null,
role varchar(10) not null check (role in ('tutor', 'tutee')),
primary key (id),
foreign key (id) references users (id),
foreign key (rating) references rating_info (rating)
);
```

account 테이블

• id: 사용자의 id

• credit: 사용자의 충전 금액

• rating: 사용자의 등급

role: 강사/학생 여부

- Term project에서는 디자인에 대한 점수가 없으므로 배경 이미지나 색상 등은 신경 쓰지 않아도 됨
- 본 자료에 나오는 웹페이지 이미지들은 전부 예시이므로 기능만 제대로 구현했다면 똑같이 따라할 필요는 없음
- 또한, 편의를 위해 테이블에 약간의 tuple이 기본으로 들어가 있음(term.sql 참고)



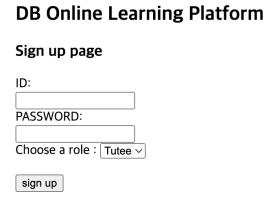
■ 웹페이지에 접속하면 가장 먼저 로그인이나 회원가입을 할 수 있는 페이지가 나옴

DB Online Learning Platform



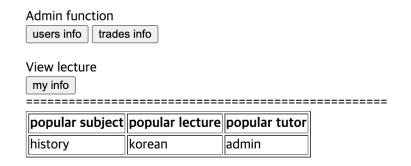
- 로그인
 - 입력된 정보와 users 테이블에 들어있는 id와 password가 전부 일치했을 때만 로그인에 성공함
 - 그 외에는 로그인에 실패함

회원가입 버튼 클릭 시 회원 가입 페이지로 이동



- 회원가입
 - 중복된 id에 대한 회원가입은 불가능
 - password의 중복은 허용
 - tutor/tutee 선택하여 가입할 수 있고, 해당 정보는 데이터베이스에 저장됨
 - 초기 설정으로 credit=10000, rating='welcome'이 주어짐
- Role을 고르는 부분은 select tag를 활용

로그인에 성공하면 강의 결제를 위한 페이지가 나옴



User Info

current user: admin / Logout

credit: 10001000 rating: gold

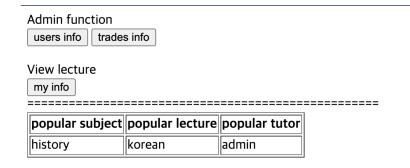
Lectures

add				
code	name	price	tutor	
00	korean history	1000	admin	register

- Admin function
 - 'admin'의 id로 로그인했을 때만 사용할 수 있는 기능(password는 '0000')
 - 모든 사용자의 정보를 볼 수 있음
 - 모든 강의 등록 내역을 볼 수 있음
- 등록 사이트의 현재 정보
 - 현재 수강 등록 횟수가 가장 많은 subject, lecture, tutor를 표시



■ 로그인에 성공하면 강의 결제를 위한 페이지가 나옴



User Info

current user: postgres / Logout

credit: 74005 rating: bronze

Lectures

add				
code	name	price	tutor	
00	korean history	1000	admin	register

- 접속 중인 사용자의 정보
 - id, credit, rating, role 등의 사용자의 정보 표시
- 로그아웃 하여 로그인 페이지로 돌아갈 수 있어야 함
- Lectures
 - 현재 수강 가능한 강의 정보를 표시 (code, name, price, tutor)
 - 강사들은 강의를 추가하거나 등록할 수 있음
 - 학생들은 강의를 구매할 수 있음

Admin function

users info trades info

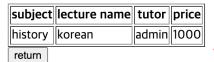


현재 로그인한 계정과 관련된 강의들을볼 수 있어야 함

My lecture

subject	lecture name	tutee	price
language	Korean	dyj	30
mathematics	math	dyj	150

Registered Lecture



Tutor

My lecture

code	lecture name	tutor	price
mathematics	math	namil	150
language	Korean	namil	30
return			

Tutee

- Tutor의 경우
 - 자신의 강의인 My lecture
 - 자신이 구매한 Registered Lecture
- Tutee인 경우
 - 자신이 구매한 My lecture



■ 사용자가 강의를 등록할 경우 해당 강의의 정보를 작성하는 페이지가 나옴

Subject List OO history O1 mathematics O2 language Code: Name: Price: add cancel

- 등록할 강의의 정보(code, name, price)를 작성하고 추가 할 수 있음
 - tutor는 강의를 등록하려는 사용자의 id
- 만약 (code, name, price)가 일치하는 강의가 있어도 강사가 다르다면 별개의 강의로 취급됨
- 취소하고 이전 페이지로 돌아갈 수 있음

■ 사용자가 강의를 결제할 경우 결제 확인 페이지가 나옴

Lecture INFO

code	name	price	tutor
00	korean history	1000	admin

Your credit: 75000 Your rating: bronze

Total price: 1000 Discount price: 5

Final price: 995

confirm cancel

tutor	tutee	subject	lecture name	price
admin	postgres	history	korean history	1000
admin	postgres	history	korean	1000
return				

- 강의 정보를 확인하고 결제함
 - 이때, 결제 금액이 자신의 credit을 초과하는 경우에는 결제에 실패함
 - 강사는 자신의 강의에 대해 결제할 수 없음
 - 이미 결제한 강의를 또 결제할 수 없음
- 결제가 진행되는 경우 수강자의 rating에 따라 할인이 됨
 - 수강자는 할인된 금액만큼 credit에서 차감
 - 강사는 할인 전 금액만큼 credit에 가산
 - 그리고 변경된 credit에 맞게 rating_info 테이블에서 condition을 확인하고 강사와 수강자의 rating을 설정함
- 결제가 완료되면 그 정보는 enrollment 테이블에 저장됨(할인 전 금액)



Additional Functions

- 앞서 설명한 기본 기능들을 제외하고 자신만의 추가 기능을 구현함
 - Unique 할수록 좋음
 - ex) admin은 강의 학생 수를 제한할 수 있음, 강사는 강의를 삭제할 수 있음, ...
- 최소 2개 이상의 추가 기능을 구현 해야함
 - 2개보다 많이 구현한다고 추가 점수는 없음
- 만약 추가 기능의 구현을 위해 테이블의 추가나 수정이 필요하다면 term.sql을 완전히 바꾸는 수준만 아니면 허용함



Report Requirements

- DB application에서 사용한 schema diagram의 그림
 - 만약 테이블을 추가했거나 수정했다면 그 부분도 포함해서 그려야 함
 - Sql 파일을 수정하여 작성했다면 sql 파일도 첨부
- 구현한 함수들에 대한 간단한 설명
 - 함수가 어떤 기능을 제공하는지
 - 어떤 SQL을 사용했는지
 - 추가 기능으로 어떤 것을 구현했는지
- 사용한 html 파일들의 간단한 설명
- 한글 또는 영어로 작성, 페이지 수의 제한은 없음
- PDF **파일**로 저장 !!



Homework

- Complete term project
- Submit your project file on blackboard
 - 10:29:59, June 1st, 2023
 - Only studentNumber_name.zip files are accepted
 - No late submission

