

[2023] 데이터베이스 Term Project 보고서

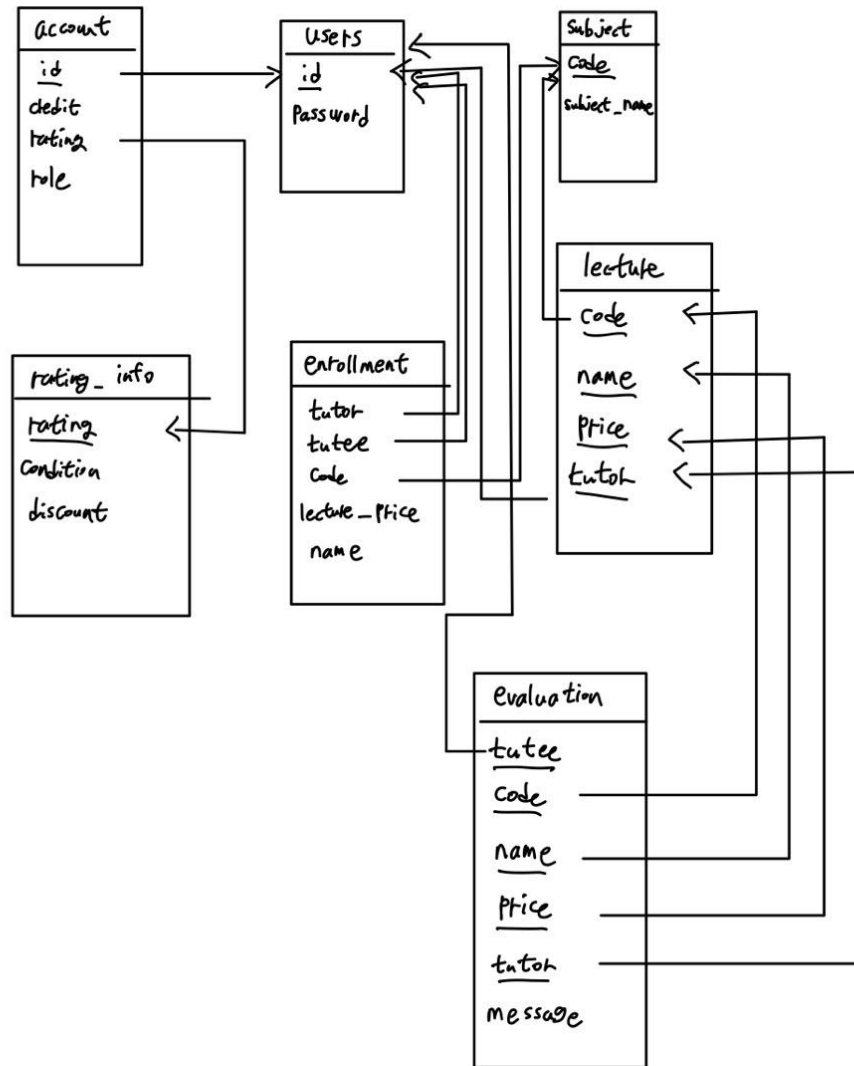
학과 : 컴퓨터학과

학번 : 2021320126

이름 : 장지윤

제출 날짜 : 2023-05-31

1. Schema Diagram



2. 함수 소개

1. rating_update(value)

i. 기능

Account Schema에 credit 변화가 생겼을 때 credit에 해당하는 rating을 return 해주는 함수이다.

2. re_turn()

i. 기능

에러가 생기거나 로그아웃을 하는 경우, main.html로 돌아가도록 하는 함수이다.

3. register()

i. 기능

form으로 id, password, send 값이 request된다. send 값은 'login' 또는 'sign up' 이 return된다. Send 값이 'login'인 경우, users 데이터베이스 내에 id와 password가 일치하는 element가 존재하면 해당 계정의 정보와, 해당 계정이 수강하고 있지 않는 강의 정보, 현 강의 사이트 내 가장 인기있는 강의 정보와 함께 login_success.html로 return하도록 하는 함수이다 users 데이터베이스 내에 id와 password가 일치하는 element가 존재하지 않다면 main.html로 돌아가도록 한다. 또한 send 값이 'sign up'인 경우에는 회원가입을 진행할 수 있는 sign_up.html로 넘어가도록 하는 함수이다..

ii. SQL

1. "select * from users;"

A. id, password가 일치하는 계정이 데이터베이스 내에 있는지 확인

2. "select subject_name, lecture_name, tutor from enrollment, subject where enrollment.code = subject.code group by subject_name, tutor, lecture_name having count(tutee) >= all(select count(tutee) from enrollment group by code, tutor, lecture_name);"

A. 현 강의 사이트 내 가장 인기있는 강의정보 확인

3. "select * from account where id ='{ }';".format(id)

A. 해당 계정에 account 정보를 확인

4. "select code, name, price, tutor from lecture where (code, name, price, tutor) not in (select code, lecture_name, lecture_price, tutor from enrollment where tutee = '{ }';".format(id)

A. 해당 계정이 수강하고 있지 않은 강의정보 확인

4. sign_up()

i. 기능

본 사이트에 회원 가입을 해 사이트 이용을 가능하게 하기 위함. 회원 가입할 때 role을 tutor 혹은 tutee로 선택 가능함. 데이터베이스 내에 같은 id를 가진 element가 없도록 이미 존재하는 id로 회원가입 신청을 하면 회원가입 없이 main.html로 넘어감. 데이터베이스 내에 같은 id가 없다면 회원가입이 되고 main.html로 넘어감.

ii. SQL

1. "select * from users;"

A. 회원가입을 요청한 id가 데이터베이스 내에 존재하지 않는지 확인

2. "insert into users values('{}', '{}');".format(id, password)

A. id가 데이터베이스 내에 존재하지 않다면, users 데이터베이스 내에 id, password를 추가

3. "insert into account values('{}', {}, '{}', '{}');".format(id, 1000, "welcome", role)

A. 회원가입한 계정에 기본 account 정보를 account schema에 추가

5. account_termination() (추가 기능)

i. 기능

Main.html에서 회원탈퇴 사이트로 이동하는 함수이다.

6. account_termination_real() (추가 기능)

i. 기능

Id, password를 form으로 request 받고 먼저 해당 (id, password)가 users 데이터베이스에 존재하는지 확인한다. 만약 존재하지 않는다면 회원탈퇴 없이 main.html로 이동한다. 존재한다면 evaluation, enrollment, lecture, account, users schema 순으로 id가 들어간 모든 element를 삭제하고 main.html로 이동한다. 데이터베이스 내 foreign key를 고려하여 해당 순으로 삭제를 진행한다.

ii. SQL

1. "select id, password from users where id = '{}' and password = '{}';".format(id, password)

- A. 회원탈퇴할 id와 password가 users 데이터베이스에 존재하는지 확인
- 2. Delete from evaluation where tutee = '{}' or tutor = '{}';".format(id, id)
 - A. Evaluation 데이터베이스에서 해당 id가 작성하거나 해당 id가 진행한 강의라면 모두 삭제
- 3. Delete from enrollment where tutee = '{}' or tutor = '{}';".format(id, id)
 - A. Enrollment 데이터베이스에서 해당 id가 강의를 수강했거나 강의를 하는 경우 모든 element 삭제
- 4. Delete from lecture where tutor = '{}';".format(id)
 - A. 해당 id가 강의를 하는 경우 lecture 데이터베이스에서 모든 element 삭제
- 5. Delete from account where id = '{}';".format(id)
 - A. 해당 id에 대한 account 정보 삭제
- 6. Delete from users where id = '{}' and password = '{}';".format(id, password)
 - A. 해당 id에 대한 users 정보 삭제

7. admin_function()

i. 기능

Admin만이 사용할 수 있는 기능으로, admin이 아닌 경우 main.html로 이동한다. Admin인 경우 전체 계정 정보를 확인할 수 있는 기능과 전체 강의 trade 정보를 확인할 수 있는 기능을 제공하는 함수이다. 전체 계정 정보 확인은 "admin_user_info.html"로 전체 강의 trade 정보 확인은 "admin_trades_info.html"로 이동한다.

ii. SQL

- 1. "select * from account;"
 - A. 모든 계정의 account 정보 확인
- 2. "select * from enrollment;"
 - A. 모든 강의 trade 정보 확인

8. user_info()

i. 기능

Login한 상황에서 로그아웃해 "main.html"로 이동하는 함수이다.

9. my_info()

i. 기능

계정이 tutor인 경우, 내가 강의하는 강의를 등록한 강의등록 정보와 내가 등록한 강의에 강의 정보를 확인할 수 있다. 계정이 tutee인 경우, 내가 등록한 강의에 강의 정보를 확인할 수 있는 함수이다.

ii. SQL

1. "select subject_name, lecture_name, tutee, lecture_price from subject, enrollment where subject.code = enrollment.code and tutor = '{}';".format(id)

A. Tutor가 내가 강의하는 강의를 등록한 강의 등록 정보를 확인

2. "select subject_name, lecture_name, tutor, lecture_price from subject, enrollment where subject.code = enrollment.code and tutee = '{}';".format(id)

A. Tutor 또는 tutee가 내가 등록한 강의에 강의 정보를 확인

10. Lectures_()

i. 기능

해당 계정의 role이 tutor면 강의를 추가할 수 있는 사이트(add_lecture.html)로 이동하며 role이 tutee면 강의를 추가할 수 없으므로 main.html로 이동하는 함수이다.

ii. SQL

1. "select * from subject;"

A. 현재 subject로 등록된 과목 코드와 강의 과목명 확인

11. add_lecture()

i. 기능

Tutor가 자신이 강의할 강의를 추가하는 함수이다. 이때 등록하려는 code가 subject 데이터베이스 내에 존재하는 code여야한다.

ii. SQL

1. "select code from subject;"

A. 등록하려는 강의 code가 Subject 데이터베이스 내에 존재하는 code 인지 확인

2. "insert into lecture values('{', '{', {}, '{'}');".format(code, name, price, tutor)

A. 등록하려는 강의 정보를 lecture 데이터베이스에 저장

12. register_lecture()

i. 기능

내가 수강하고 있지 않은 강의를 새로 수강 등록하는 사이트로 이동하는 함수이다.

ii. SQL

1. "select id, credit, rating, role from account where id = '{'}';".format(id)

A. 해당 계정의 account 정보를 확인

2. "select rating, condition, discount from rating_info where rating = '{'}';".format(rating)

A. 해당 계정의 rating에 대한 rating_info 정보 확인

13. register_confirm

i. 기능

나의 rating 정보를 계산하여 할인율을 반영해 가격을 표시하고 결제를 진행하는 함수이다. 결제가 진행되면 해당 계정의 credit이 할인된 가격만큼 차감되고 결제하는 강의에 tutor의 credit은 원래 강의 가격만큼 증가하며 enrollment_confirm.html로 이동한다. 만약, 해당 계정이 결제하려는 강의에 tutor이거나 할인된 가격보다 credit이 작거나 이미 등록한 강의라면 결제되지 않고 main.html로 이동한다.

ii. SQL

1. "select tutee from enrollment where tutor = '{'}' and code = '{'}' and lecture_name = '{'}';".format(tutor, code, name)

- A. 해당 계정이 이미 등록한 강의가 아닌지 확인하기 위해 해당 강의를 등록한 모든 tutee 정보 확인
2. "update account set credit = credit - {} where id = '{}';".format(final_price, id)
 - A. 결제가 진행되며 해당 계정 credit을 할인 가격만큼 차감
3. "update account set credit = credit + {} where id = '{}';".format(price, tutor)
 - A. 결제가 진행되며 등록하는 강의 tutor 계정 credit을 원래 가격만큼 증가
4. "update account set rating = '{}' where id = '{}';".format(rating_update(credit - final_price), id)
 - A. 해당 계정 credit이 변경되었기에 이에 맞는 rating으로 update
5. "select * from account where id = '{}';".format(tutor)
 - A. 등록하는 강의 tutor 계정에 credit 확인
6. "update account set rating = '{}' where id = '{}';".format(rating_update(tutor_credit + price), tutor)
 - A. 등록하는 강의 tutor 계정에 credit이 변경되었기에 이에 맞는 rating으로 update
7. "insert into enrollment values('{}', '{}', '{}', '{}', {});".format(id, tutor, code, name, price)
 - A. 강의 등록을 위해 enrollment 데이터베이스에 강의 등록 내역 추가
8. "select * from enrollment where tutee = '{}';".format(id)
 - A. 강의 등록이 잘 되었는지 확인

14. evaluation() (추가 기능)

i. 기능

내가 등록한 강의에 대해서는 강의 평가를 쓸 수 있고, 내가 등록하지 않은 강의에 대해서는 강의 평가를 볼 수 있는 사이트로 이동하는 함수이다.

ii. SQL

1. "select code, lecture_name, lecture_price, tutor from enrollment where tutee = '{}';".format(id)

A. 내가 등록한 강의 내역 확인

2. "select code, name, price, tutor from lecture where (code, name, price, tutor) not in (select code, lecture_name, lecture_price, tutor from enrollment where tutee = '{}');".format(id)

A. 내가 등록하지 않은 강의 내역 확인

15. lecture_eval() (추가 기능)

i. 기능

강의 평가 사이트에서 form request로 send가 'view' 또는 'write'가 return되고 view 또는 write 하려는 강의 정보가 return된다. Send가 'view'인 경우에는 해당 강의에 대한 평가를 볼 수 있는 view_lecture_eval.html로 이동하는 함수이다. Send가 'write'인 경우에는 해당 강의에 대한 평가를 쓸 수 있는 write_lecture_eval.html로 이동한다. 만약 해당 계정이 본 강의에 대해서 이미 강의 평가를 작성했다면 return.html로 이동한다.

ii. SQL

1. "select message from evaluation where code = '{} and name = '{} and price = {} and tutor = '{}';".format(code, name, price, tutor)

A. send가 'view'일 때, 해당 강의에 대한 모든 강의 평가 확인

2. "select tutee, code from evaluation where tutee = '{} and code = '{} and name = '{} and price = {} and tutor = '{}';".format(id, code, name, price, tutor)

A. send가 'write'일 때, 해당 계정이 이미 강의 평가를 작성하였는지 확인

16. write_eval() (추가 기능)

i. 기능

강의 평가를 작성해 evaluation 데이터베이스에 저장하는 함수이다. 작성 후,

return.html로 이동하게 된다.

ii. SQL

1. "insert into evaluation values('{}', '{}', '{}', {}, '{}', '{}');" .format(id, code, name, price, tutor, message)

A. 해당 강의 평가를 evaluation 데이터베이스에 저장한다.

3. HTML 파일 소개

1. main.html

처음 들어가게 되는 사이트로, 로그인 또는 회원가입 사이트 이동, 회원탈퇴 사이트 이동할 수 있다. App.py에 id, password를 전달하고 send로 'login' 혹은 'sign up'을 전달한다. 회원탈퇴는 send로 'Account Termination'을 전달한다.

2. login_success.html

'admin' id만 사용할 수 있는 Admin function이 있고 Admin function은 모든 사용자의 정보를 보거나 모든 강의 등록 내역을 볼 수 있다. 또한, 나의 lecture를 볼 수 있는 my info 버튼이 있으며 현재 등록 사이트의 수강 등록 횟수가 가장 많은 subject, lecture, tutor를 표시한다. 해당 계정의 id, credit, rating, role 등의 사용자 정보를 표시한다. 로그아웃하여 main.html로 넘어갈 수 있다. 현재 수강 가능한 강의 정보를 표시하며 강사들은 강의를 추가하는 add 버튼을 사용할 수 있고 강사와 학생 모두 강의를 register할 수 있다. 또한 강의 평가를 쓰고 볼 수 있는 강의 평가 사이트로 이동할 수 있는 Check Evaluation 버튼이 있다.

3. sign_up.html

회원가입하는 사이트로, 중복된 id에 대한 회원가입은 불가능하며 tutor/tutee를 선택하여 가입할 수 있다.

4. account_termination.html

회원탈퇴하는 사이트로, id password를 입력해 해당 계정을 삭제할 수 있다. id password가 틀릴 경우 main.html 사이트로 돌아간다.

5. admin_trades_info.html

admin만 들어갈 수 있는 사이트로 모든 강의 등록 정보가 표기된다. 또한, 다시 login_success.html 사이트로 돌아갈 수 있는 return 버튼이 있다.

6. admin_user_info.html

admin만 들어갈 수 있는 사이트로 모든 사용자 정보가 표기된다. 또한, 다시 login_success.html 사이트로 돌아갈 수 있는 return 버튼이 있다.

7. my_info_tutor.html

사용자가 tutor일 때, 자신의 강의를 등록한 학생과 함께 my lecture와 자신이 등록한 Registered lecture를 볼 수 있다.

8. my_info_tutee.html

사용자가 tutee일 때, 자신이 등록한 Registered lecture를 볼 수 있다.

9. add_lecture.html

tutor가 강의의 정보를 작성해 새로운 강의를 등록하는 홈페이지이다. code, name, price를 입력해 add 버튼을 누르면 새로운 강의로 등록된다. 취소하고 이전 페이지로 돌아갈 수 있는 cancel 버튼이 있다. Add 버튼을 눌렀지만, 등록하려는 code가 subject에 존재하지 않으면 강의를 등록되지 않으며 return.html 사이트로 이동한다.

10. Register_lecture.html

새로운 강의를 등록할 수 있는 홈페이지이다. 등록하려는 강의 정보와 사용자의 credit, rating과 함께 rating에 따른 할인율과 최종 가격이 나온다 confirm 버튼을 누르면 강의를 등록되며 등록이 잘 되었는지 확인하는 enrollment_confirm.html 사이트로 이동한다. 취소하고 이전 페이지로 돌아갈 수 있는 cancel 버튼이 있다.

11. enrollment_confirm.html

Register_lecture.html 사이트에서 강의를 등록하고 강의를 잘 등록되었는지 확인하는 사이트이다. 내가 등록한 모든 강의를 확인할 수 있다. login_success.html 사이트로 이동할 수 있는 return 버튼이 있다.

12. eval.html

강의 평가 사이트이다. 내가 등록한 강의 정보와 내가 등록하지 않은 강의 정보가 표로 나타난다. 등록한 강의 정보에 대해서는 강의 평가를 쓸 수 있는 write_lecture_eval.html로 이동하는 write 버튼이 있으며 등록하지 않은 강의 정보에 대해서는 강의 평가를 볼 수 있는 view_lecture_eval.html로 이동하는 view 버튼이 있다. Write 버튼을 눌렀지만 이미 해당 강의에 대해서 강의 평가를 작성한 경우에는 return.html 사이트로 이동하게 된다. 또한 이전 사이트로 돌아갈 수 있는 return 버튼이 있다.

13. view_lecture_eval.html

내가 클릭한 강의에 강의 평가를 볼 수 있다. 또한, Login_success.html 사이트로 갈 수 있는 return 버튼이 있다.

14. write_lecture_eval.html

내가 클릭한 강의에 강의 평가를 쓸 수 있다. 강의 평가를 쓰고 write 버튼을 누르면 강의 평가가 데이터베이스에 등록된다. 또한, 강의 평가 작성을 취소하고 Login_success.html 사이트로 갈 수 있는 cancel 버튼이 있다.

15. return.html

return 버튼을 통해 login_success.html 사이트로 돌아갈 수 있다.