**도서관대출\_정진묵\_최종 발표 자료**

< 목차 >

* 프로젝트 수행 과정

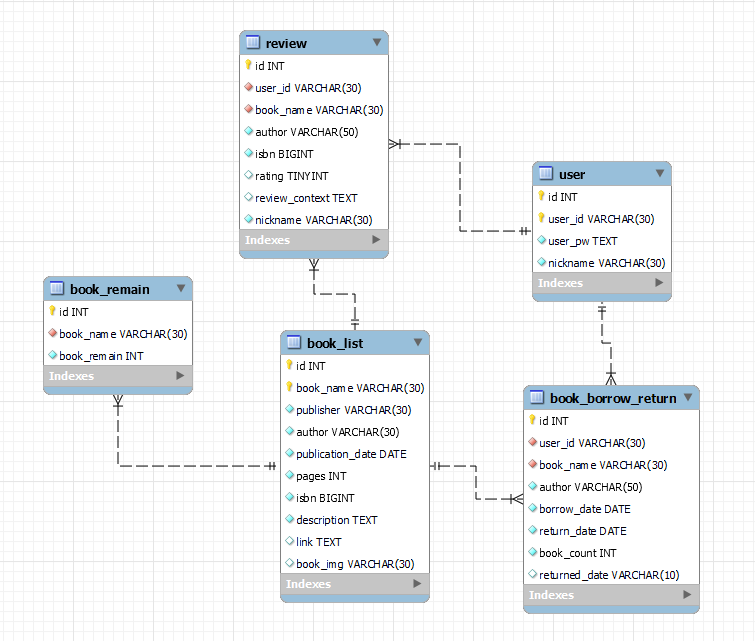
1. 데이터베이스 설계
2. main\_view.py 관련파일 설계
3. book\_info\_view.py 관련파일 설계
4. book\_checkout\_return\_view.py 관련파일 설계

* 라이브 데모
* 어려웠던 점
* 배운 점
* 프로젝트를 통해 느낀 점

**< 프로젝트 수행 과정 >**

**1. 데이터베이스 설계**

* 데이터 베이스 구조

=> 모든 테이블은 아이디나 책 제목 등의 중복을 방지하기 위해 자동으로 증가하는 id값을 기본키로 설정하였습니다.

- user 테이블 : 이메일 주소로 된 아이디인 user\_id를 복합키로 갖고, 비밀번호와 닉네임 속성을 갖는 테이블입니다.

- book\_list 테이블 : 책 제목인 book\_name을 복합키로 갖고 출판사, 작가, 출판일, 페이지수, isbn코드, 책 설명, 책 링크, 책 이미지의 주소 링크를 속성으로 갖는 테이블입니다.

- book\_borrow\_return 테이블 : 유저 아이디와 책 제목을 외래키로 갖고 작가와 빌린 날짜, 반납한 날짜, 빌린 권수, 반납을 했는지를 체크하는 returned\_date를 속성으로 갖는다. 이때 returned\_date의 기본값은 no로 설정하였다. 기본적으로 반납일은 빌린 날을 기준으로 2주 뒤 날짜로 설정하였고, 후에 반납을 하게 되면 returned\_date값이 yes로 바뀌면서 return\_date인 반납 날짜는 당일 날짜로 변경됩니다.

- book\_remain 테이블 : 책 제목을 외래키로 갖고 남은 권수를 나타내는 속성을 갖는 테이블이다. 남은 권수에 해당하는 book\_remain속성은 기본값으로 10권을 갖는다. 모든 책은 10권씩 있다고 가정합니다.

- review 테이블 : 책에 관한 리뷰에 대한 테이블이다. 리뷰를 작성한 유저의 아이디와 책 제목 속성을 외래키로 갖고 책정보와 유저의 닉네임에 대한 속성을 갖는 테이블입니다.

**2. main\_view.py 파일 설계**

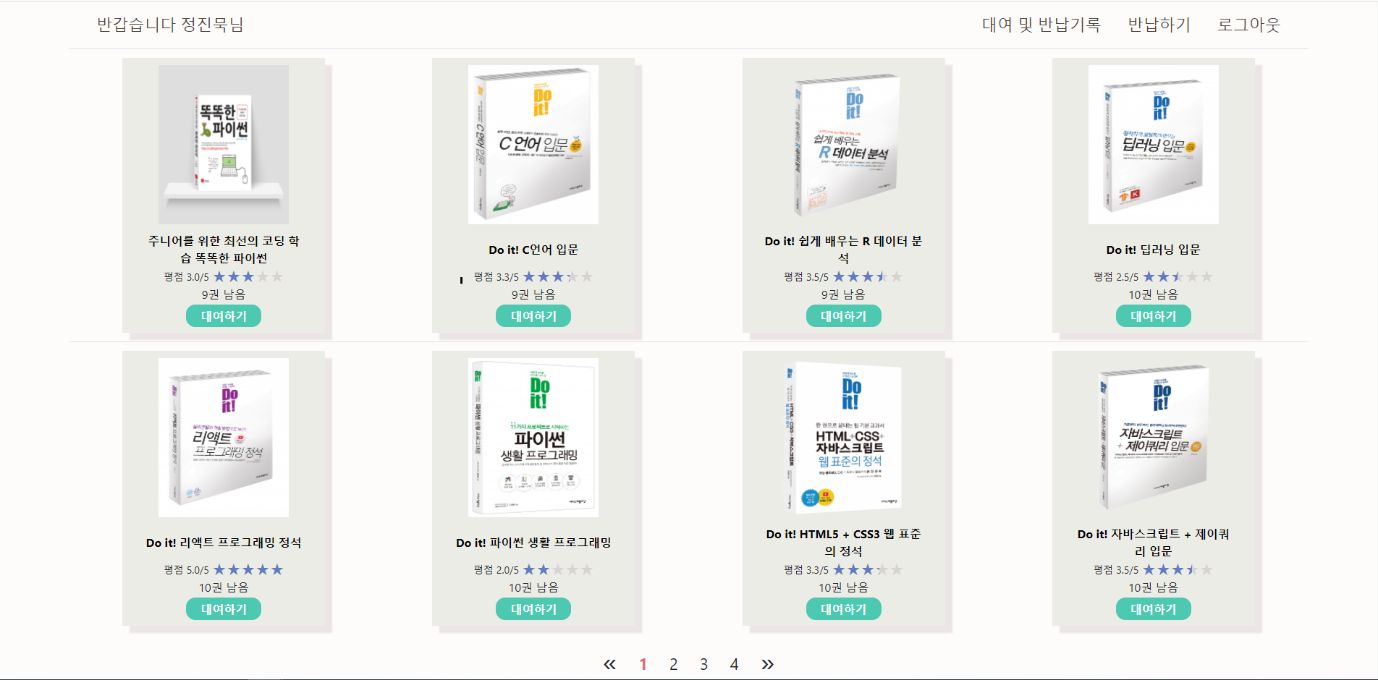
실내, 하얀색, 다른, 여러개이(가) 표시된 사진

자동 생성된 설명2-1) 메인 페이지

- pagination 기능을 이용해 32권의 책을 한 페이지에 8권씩 보이게 하였습니다. => javascript의 click 이벤트와 이에 따른 테이블에 .notshow클래스명을 이용해 display: none; 으로 설정하여 각 페이지에 맞는 테이블 행만 클래스명을 지워주도록 이벤트를 설정했습니다.

- 로그인이 되어있지 않은 경우 위의 사진처럼 상단 네비게이션 바에 로그인과 회원가입 버튼만 보이고 좌측에는 레이서 도서관이라는 문구만 보이게 설정했습니다.

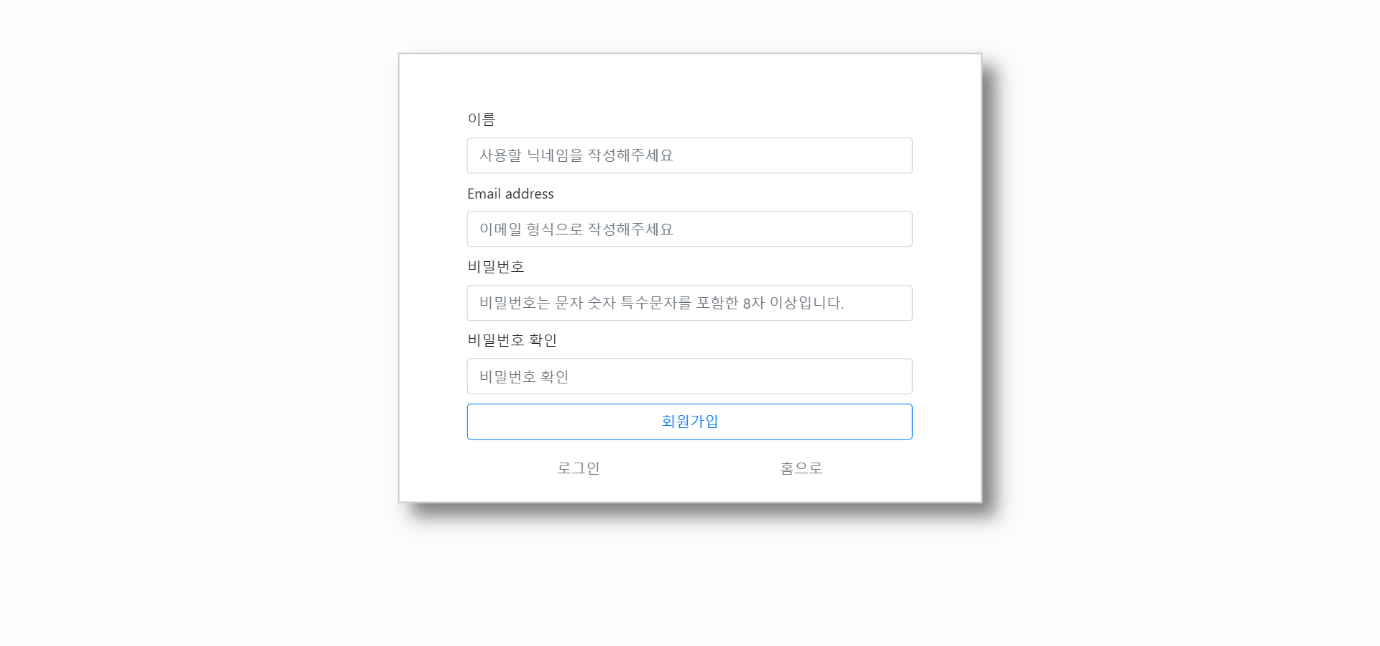
- 하나의 책 마다 책 사진, 책 제목, 평점, 남은 권수, 로그인이 되어있다면 대출하기 버튼이 보이도록 설정하였다.

- 책의 평점은 해당 책에 해당하는 모든 평점들의 평균의 소수 첫째 자리까지 반올림하여 나타낸 점수입니다. 이 점수에 맞춰 5점 만점에 해당 %만큼 별의 색이 채워지도록 만들었습니다.

- 위 사진은 로그인이 된 상태의 메인 페이지이다. 상단 네비게이션 바의 내용이 바뀐 것을 확인할 수 있다. 왼쪽에는 유저의 닉네임이 나오고 오른쪽 상단에는 대여 및 반납기록, 반납하기, 로그아웃 메뉴가 존재하는 것을 알 수 있습니다.

- 대여 및 반납기록 페이지의 경우 유저가 대여한 후 페이지에 책이 표시되는 것이 아니라 대여 후 반납하기 페이지에서 책을 반납한 경우에 대여 및 반납기록에 표시가 됩니다.

- 반납하기 페이지에는 반납기한이 2주로 표현되어 소개가 되고 있는 상태이고 반납이 이루어진 후에는 대여 및 반납기록에 반납날짜가 해당 반납일로 바뀌어서 표시가 됩니다.

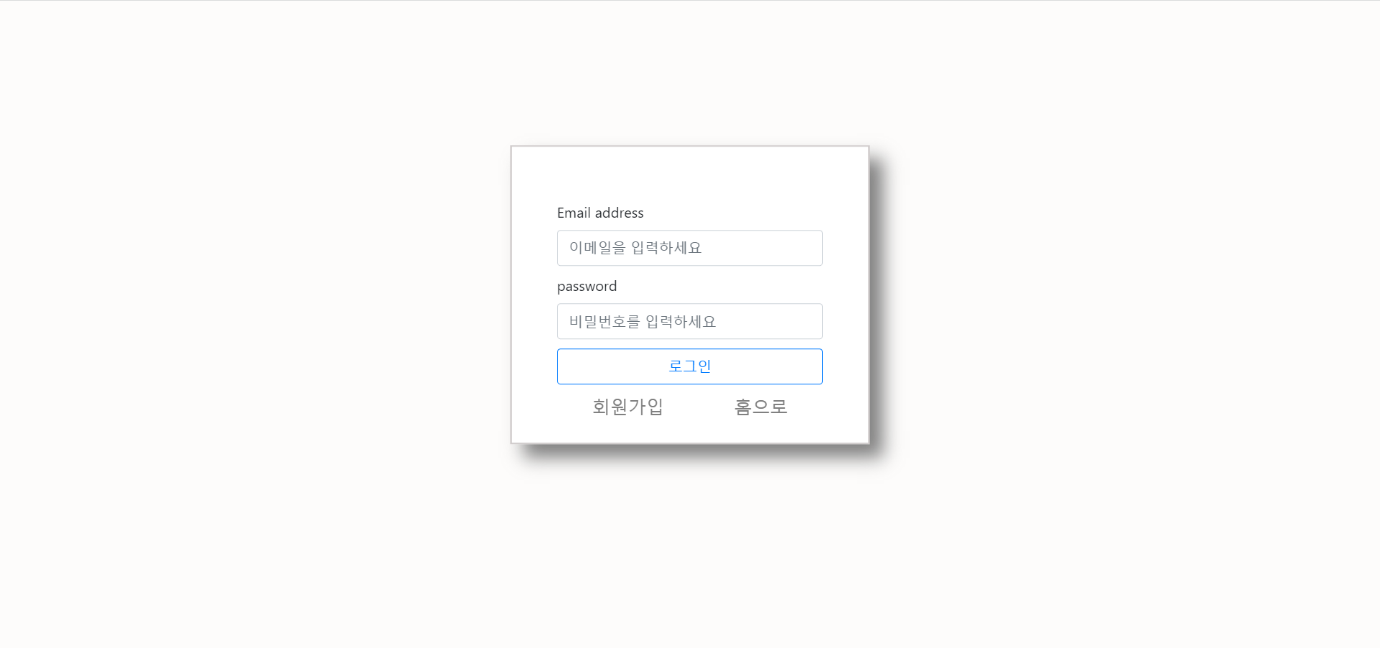
2-2) 회원가입 페이지

- 회원가입시 닉네임, 이메일 아이디, 비밀번호를 설정해 주어야 합니다.

- 이메일 형식의 경우 python 정규 표현식을 이용하여 input태그의 pattern 속성에 작성해 주었습니다.

- 비밀번호의 경우 특수문자, 영문자, 숫자를 모두 사용하여 8자리 이상 작성하도록 규칙을 설정하였고 이는 flask 서버에서 따로 비밀번호를 확인하는 함수를 만들어서 구현하였습니다. 이때 re모듈의 re.search함수를 이용했습니다.

- 만약 4가지 입력 중 하나라도 잘못된 입력이 들어온다면 flask의 flash를 이용하여 잘못된 내용에 대해 설명하는 문구가 나오도록 구현했습니다.

2-3) 로그인 페이지

- 이메일과 비밀번호 중 하나라도 틀린 입력이 있다면 flash를 통해 틀린부분을 알려주도록 구현했습니다.

- 비밀번호의 경우 flask\_bcrypt 모듈을 이용하였습니다.

- 로그인된 유저의 정보 중 이메일과 닉네임 정보를 flask session에 저장하였습니다.

2-4) 로그아웃 페이지

- 로그아웃 메뉴를 누르면 session에서 이메일 정보와 닉네임 정보를 None으로 바꿔주는 방법을 통해 구현했습니다.

**3. book\_info\_view.py 파일 설계**

텍스트이(가) 표시된 사진

자동 생성된 설명

- 책 상세 페이지와 리뷰 작성에 대한 부분입니다.

- 책의 제목을 클릭하면 해당하는 책의 상세 페이지로 Book\_list테이블의 id값을 이용해서 이동하도록 설정하였습니다.

- 평점과 리뷰를 남기는 부분은 로그인이 된 상태에만 보이도록 설정하였습니다.

- 전체적인 페이지는 책 사진, 책 정보를 테이블 태그를 통해 나타내고 그 밑에 책 소개 페이지와 댓글 목록들을 최신 댓글 순서로 불러오게 되어있고 마지막으로 평점과 리뷰를 남길 수 있는 부분이 있다.

- 평점은 5점만점으로 줄 수 있고 기본적으로 3점이 세팅이 되어있는 상태입니다.

- 리뷰를 작성하게 되면 Review테이블에 데이터가 저장되고 댓글 목록에서 확인할 수 있게 됩니다. 이때 작성된 평점들의 평균이 메인페이지 부분에 나타나게 됩니다.

- 평점 부분은 input태그의 checkbox타입을 이용했는데 특정 박스보다 작은 값들을 가지는 태그들은 모두 선택되도록 설정을 해주었고 데이터를 flask에 넘겨줄 때는 모든 값들을 리스트 형태로 넘겨주고 결과는 가장 큰 값만 따로 빼서 이용했습니다.

**4. book\_checkout\_return\_view.py 파일 설계**

4-1) 대여하기 페이지

- 메인 페이지에서 책 별로 대여하기 버튼을 클릭하면 위의 페이지로 넘어갈 수 있습니다. 대여하기 페이지는 책 제목과 몇 권의 책을 대여할 것인지 정할 수 있습니다.

- 이때 남아있는 책보다 많은 양의 책을 대여하게 된다면 경고창이 발생하며 더 이상 숫자를 늘릴 수 없습니다. 마찬가지로 1권 이하의 책으로 버튼을 눌러서 줄이면 경고창이 발생하게 됩니다.

- 대여하기 버튼을 클릭하면 원하는 권수만큼 대여하게 되고 Remain테이블에 수정된 책의 권수가 저장이 되며 Book\_borrow\_return 테이블에서도 대여한 책과 유저정보가 저장되게 됩니다.

4-2) 반납하기 페이지

- 책의 제목과 몇 권의 책을 빌렸는지 확인 가능하며 반납기한의 경우 대여일로부터 2주 뒤로 자동 설정되어 있습니다.

- 반납하기 버튼을 누르면 반납이 완료되고 반납한 책은 대여 및 반납기록에 남게 됩니다.

- Book\_remain테이블에서 반납을 완료한 책의 권수만큼 다시 업데이트를 진행합니다.

4-3) 대여 및 반납기록 페이지

텍스트, 하얀색이(가) 표시된 사진

자동 생성된 설명- 반납을 완료한 책들의 목록만 나오도록 하였습니다. Book\_borrow\_table에서 returned\_date의 값이 yes인 책들만 출력됩니다.

- 반납일의 경우 Book\_borrow\_table에서 returned\_date의 값이 no라면 borrow\_date로부터 2주 뒤로 처음에 설정되지만 반납을 완료하는 순간 returned\_date의 값이 yes로 변경되면서 return\_date의 값이 당일의 날짜로 업데이트 됩니다.

기본적인 디자인은 메인페이지와 동일합니다.

**<라이브 데모>**

- Microsoft azure 가상머신을 통해 배포하였습니다.

<http://elice-kdt-3rd-vm-098.koreacentral.cloudapp.azure.com:5000/>

**< 프로젝트 진행하며 어려웠던 점들 >**

- 개인적으로 db에서 데이터를 가져와 기능을 구현하는 것보다 디자인 구현이 더 어렵게 느껴졌습니다. => 전체적인 디자인 색을 정하는데 많이 참고한 사이트입니다.

<https://mycolor.space/>

- 평점 비율에 맞게 별모양으로 색을 칠하는 부분과 댓글 작성시 별점을 주는 부분에서 많은 고생을 했습니다. => 참고했던 블로그 입니다.

<https://melthleeth.tistory.com/entry/HTML-CSS%EB%A1%9C-%EB%B3%84%EC%B0%8D%EA%B8%B0-Star-Rating>

<https://blogpack.tistory.com/695>

**< 프로젝트를 통해 배운 점들 >**

- 데이터베이스 설계가 매우 중요한 것 같습니다.

- 수업시간에 배운 내용을 갖고 거의 모든 기능을 구현했는데 다른 레이서분들 질문하는 내용들을 살펴보면 정말 아직 배울 내용이 많이 남아있는 것 같고 더 열심히 해야겠다는 생각이 들었습니다.

- 서버 배포할 때 생각보다 다양한 오류들이 발생하는 것 같습니다.

- bootstrap이 굉장히 많은 시간을 단축시켜준다는 것을 느꼈습니다.

- 백엔드가 개인적으로 맞고 재밌다고 느낀 것이 가장 중요한 것 같습니다.

**< 프로젝트를 진행하며 느낀 점 >**

개인적으로 처음부터 끝까지 혼자 프로젝트를 진행한 것이 처음이었기 때문에 첫날과 둘째날 정말 많은 고민을 한 것 같습니다. 특히 데이터베이스를 어떻게 설계해야 좋을지에 대한 고민을 정말 많이 했고 .gitignore파일도 처음 만들어보고 flask파일도 처음부터 작성한게 태어나서 처음 한 일들이었는데 많은 오류를 겪었고 하나하나 해결할 때마다 정말 기분좋게 프로젝트를 진행했던 것 같습니다. 막상 프로젝트가 끝나니 부족한 부분이 더욱 많이 보이고 구현하고 싶은 기능들도 더 늘어난 것 같습니다.

아직은 너무나 부족한 사람이지만 이번 프로젝트를 진행하며 스스로 문제를 해결해가며 경험한 과정들을 바탕으로 앞으로 즐겁게 백엔드 공부를 계속 할 수 있었으면 좋겠고, 원하는 기능들을 앞으로 채워가며 성장하는 레이서가 될수있으면 좋겠습니다.