

Introduction to Databases, Spring 2019

Final Project Report

Student ID 2014310407

Name 이 준 혁

1. Overview

1) Benchmarking site: 네이버 웹툰 페이지: <https://comic.naver.com/webtoon/weekday.nhn>

2) Objective:

이 프로젝트의 목적은 실제와 비슷한 웹툰 페이지를 구현하는 것이다. 기존 웹툰 페이지와 같이, 웹툰, 작가, 웹툰의 내용, 사용자로 이루어져 있다. 각 사용자는 로그인을 해서 쿠키라는 재화를 구매할 수 있고, 이 쿠키를 바탕으로 웹툰을 구매하여 볼 수 있다. 또한 자신이 좋아하는 웹툰을 관심 웹툰으로 지정할 수 있다.

더 나아가, 웹툰은 각각 게시되는 요일이 있고, 각 웹툰에는 매 화마다 평점이 매겨진다. 모든 웹툰은 1차적으로는 요일별로 전시되어 있다. 여기에서 웹툰의 이름 순으로 정렬해서 보거나, 평점 순으로 웹툰을 정렬해서 볼 수도 있다.



[그림 1-1] 요일 별 웹툰 전시의 예(<https://comic.naver.com/webtoon/weekday.nhn>)

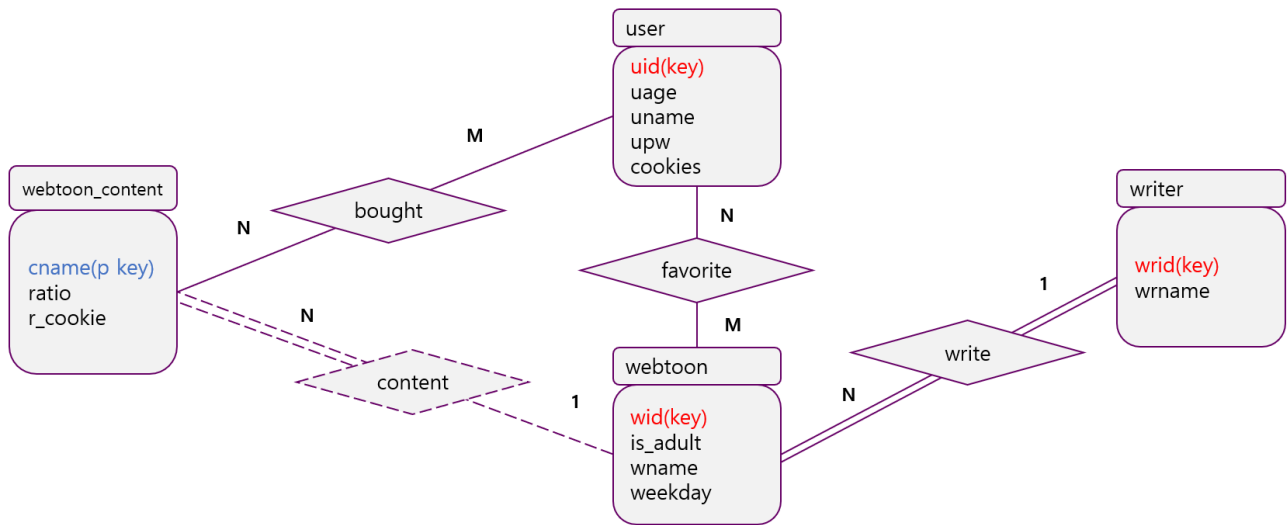
또한 다음과 같이 특정 작가의 웹툰 혹은 특정 웹툰을 이름을 통해 검색해서 볼 수 있게 서비스를 제공할 것이다.



[그림 1-3] 웹툰 검색 결과 예시(<https://comic.naver.com/webtoon/weekday.nhn>)

마지막으로, 성인만 볼 수 있는 웹툰이 존재하는데, 이는 미성년자가 볼 수 없도록 제한하는 서비스 역시 제공할 것이다.

2. ER-Diagram



[그림 2-1] ER diagram

1) Entity

(1) **User:** 웹툰을 보는 사용자를 나타내는 Entity이다. 다음과 같은 5개의 attribute를 가진다. Uid를 key로 가진다.

- age: 사용자의 나이

- uid: 사용자를 식별하기 위한 id, 각 사용자별로 유일하게 설정해야 한다. 사용자가 직접 회원가입시 기입하는 id이다.

- cookie: 사용자가 보유하고 있는 쿠키의 수(쿠키는 웹툰 구매를 위한 재화이다.)

- pw: uid와 더불어 사용자가 로그인을 하기 위한 비밀번호이다. 사용자가 직접 회원가입시 기입하는 비밀번호이다.

- uname: 사용자가 직접 입력하는 사용자의 이름(닉네임)이다.

(2) **webtoon:** 웹툰을 나타내는 Entity이다. 다음과 같은 4개의 attribute를 가진다. Wid를 key로 가진다.

- wid: 웹툰을 구별하기 위한 id이다. 웹툰 별로 유일하게 하나씩 가진다.

- Is_adult: 성인 웹툰 여부를 구분하기 위한 attribute이다. 성인 웹툰일 시 True, 아닌 경우 False이다. 성인 웹툰의 경우 사용자의 나이가 19세 미만인 경우 접근 자체가 불가능하다.

- wname: 웹툰의 이름이다. 웹툰의 이름 순 정렬에 사용된다.

- weekday: 웹툰의 요일을 나타낸다. 한 웹툰이 여러 요일에 연재될 수도 있기 때문에, multi-value 로 설계되었다.

(3) Webtoon_content: 웹툰이 있어야 지만 생성될 수 있는 weak entity type이다. 웹툰의 내용들(각 화의 내용들) 다음과 같은 2개의 attribute를 가진다. Cname을 partial key로 가진다.

- cname: 각 화의 제목을 나타낸다. 웹툰에서 유일하게 결정되는 partial key를 나타낸다.

- Ratio: 각 화의 평점을 나타낸다. 이후 평점 순으로 웹툰을 정렬할 때 사용된다.

- r_cookie: 각 화를 구매하는 데에 사용하는 쿠키의 수이다.

(4) Writer: 웹툰을 작성하는 작가를 나타낸다. Wrid를 key로 가진다. 다음과 같은 두 개의 attribute를 가진다.

- wrid: 작가를 유일하게 결정하는 id이다.

- wrname: 작가의 이름이다. 작가를 통해 웹툰을 검색할 때 이용된다.

2) Relation

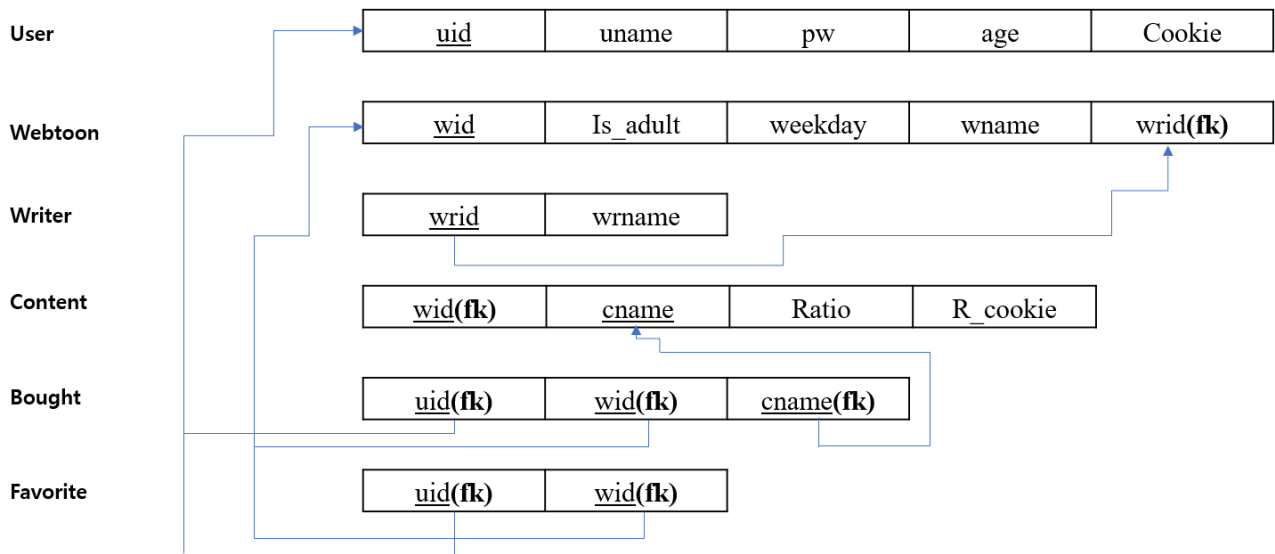
(1) Bought: 어떤 유저가 어떤 웹툰의 몇 화를 샀는지를 나타내는 relation이다. 한 명의 유저가 여러 웹툰을 구매할 수 있고, 한 웹툰을 여러 유저가 구매할 수도 있기 때문에, M:N의 관계를 가진다.

(2) Content: webtoon과 그 내용들, 즉 각 화들을 저장해 놓는 관계이다. 웹툰 없이 내용만 존재할 수 없으므로, weak entity relation을 나타내며, 모든 화는 웹툰 하나에 속해 있지만 웹툰은 여러 화를 가지고 있으므로, N:1의 관계를 가진다.

(3) Favorite: 유저가 관심을 표시한 웹툰 간의 관계를 나타낸다. 한 명의 유저가 여러 웹툰에 관심 표시를 할 수도 있고, 한 웹툰이 여러 사용자에게 의해 관심 표시를 받을 수 있으므로, M:N 관계를 가진다.

(4) Write: 웹툰과 그 웹툰을 그린 작가 간의 관계를 나타낸다. 모든 작가는 적어도 하나의 웹툰을 그려야 하고, 모든 웹툰은 정확하게 한 명의 작가만 가지게 설계했다. 따라서 둘 다 total participation을 가지고, 1:N의 관계를 가진다.

3. Relational model



[그림 3-1] Relational model

1) **User**: ER diagram과 동일하게, uid, uname, pw, age, cookie attribute를 가진다. Uid를 primary key로 가진다.

2) **Webtoon**: ER diagram과 동일하게, wid, is_adult, weekday, wname를 attribute로 가지고, wid를 primary key로 가진다. 또 작가 id를 나타내는 wrid를 primary key로 가진다.

3) **Writer**: ER diagram과 동일하게, wrid, wrname을 attribute로 가지고, wrid를 primary key로 가진다.

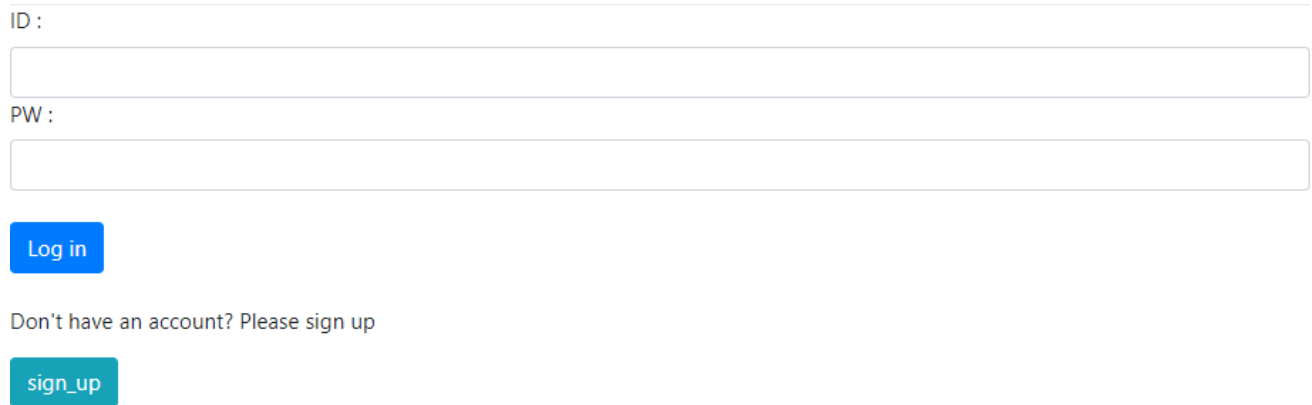
4) **Webtoon_content**: Primary key로 웹툰 id를 나타내는 wid와 화 이름을 나타내는 cname을 가진다. 이 때 wid는 webtoon relation에서 가져온 primary key이다. 또한 Ratio와 r_cookie를 attribute로 가진다.

Bought: 유저, 웹툰, 웹툰의 화를 나타내는 uid, wid, cname을 primary key로 가진다.

Favorite: 유저, 웹툰을 나타내는 uid, wid을 primary key로 가진다.

4. Implementation and Results

1) Function 1: sign up & Log in & Log out



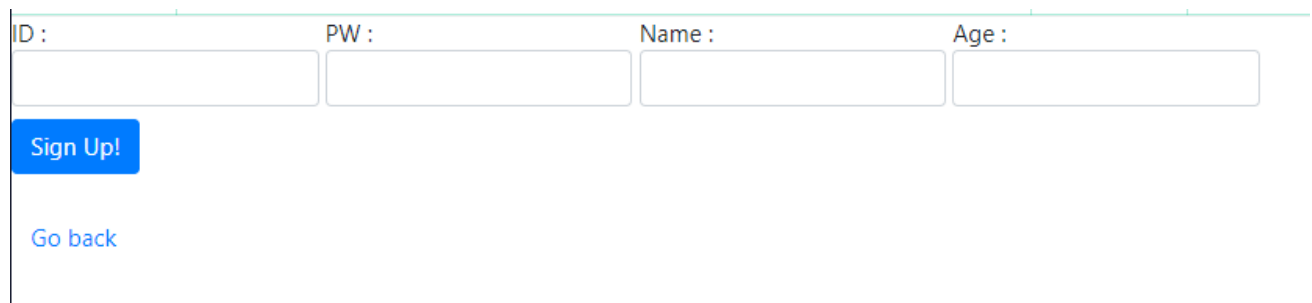
The login page form consists of two input fields: 'ID :' and 'PW :'. Below the 'PW :' field is a blue 'Log in' button. Underneath the button is the text 'Don't have an account? Please sign up', followed by a teal 'sign_up' button.

[그림 4-1] Login page

아이디와 비밀번호를 입력함으로 인해 로그인을 할 수 있다. 아이디 혹은 비밀번호가 잘못되면 다음과 같은 오류 메시지와 함께 로그인을 허용하지 않는다.

이 때 select 문을 이용하는데, select 문의 결과가 없다면, 입력된 id/pw에 해당하는 유저가 없다는 것이고, 이는 아이디 혹은 password가 잘못된 것이라는 의미이므로, 입력이 잘못되었다는 문구를 출력 후 다시 로그인하게 했다.

로그인에 성공하면 user table의 정보로 세션을 만든 후에, 메인 화면으로 이동한다. 반면 Sign_up을 누를 경우 sign_up.html으로 이동한다.



The sign up page form has four input fields: 'ID :', 'PW :', 'Name :', and 'Age :'. Below these fields is a blue 'Sign Up!' button. At the bottom left, there is a blue 'Go back' link.

[그림 4-2] Sign up page

Sign_up 화면에서는 ID, pw, 이름, 나이를 입력함으로 인해서 로그인을 한다. 이 때 Insert 문을 통해서 해당 값을 user table에 추가하는데, 만약 table의 모든 값이 not null constraint로 설정되어 있기 때문에 값이 하나라도 비어 있다는 에러가 나오면, 빈칸을 채우라는 오류 메시지를 내보냈다.

빈 값이 없어도, id는 primary key 이므로, insert 하려고 하는 row의 id가 중복되었다는 에러가 나온 경우, 이미 해당 아이디가 존재한다는 오류 메시지를 내보냈다.

마지막으로 메인 화면에서 로그아웃 버튼을 누른다면 로그아웃을 해서 다시 로그인 페이지로 갈 수 있다. 이는 존재하는 session을 파괴하면서 로그아웃을 하도록 구현했다.

2) Function 2: webtoon search



[그림 4-3] Search page

Main 화면에서 search 버튼을 누를 경우 다음과 같은 search page를 마주할 수 있다. 여기서 웹툰의 이름을 이용해 검색어가 들어간 웹툰들을 검색할 수 있다. 이는 LIKE와 %기호를 select문에 넣으면서 구현했다.

webtoon search

webtoon name :

Search

2 webtoons are found

웹툰 이름	작가 이름	요일	성인 웹툰 여부	관심 웹툰 추가
Trinity	writer1	Sunday	No	add favorite
Aladdin	writer3	Tuesday	No	add favorite

[Go to home](#)

[그림 4-4] 'in'으로 검색했을 때 결과

다음과 같이 'in'으로 검색한 경우 in이 들어간 모든 웹툰의 목록이 나온다. 웹툰 제목을 클릭하면 웹툰 화 목록을 볼 수 있다. 이 경우 세션에서 유저의 나이를 가져와 유저가 미성년자인데 성인 웹툰에 접근할 시 미성년자는 관람할 수 없다는 메시지를 주며 접근을 허용하지 않는다. 이 이외의 경우에는 웹툰 아이디에 해당하는 모든 화들을 보여준다. 이 역시 select 문을 이용해 구현했다.

Trinity

3 webtoon stories are exist

웹툰 화	평점	요구 쿠키 수	구매
story1	2	0	purchase
story2	2	0	purchase
story3	4	0	purchase

[그림 4-5] 'Trinity' 웹툰을 클릭했을 경우

3) Function 3: Favorite webtoons

웹툰 검색 후에 add favorite를 누르게 된다면, insert 문을 이용해 favorite table에 유저의 id와 웹툰의 id를 저장한다. 만약 이미 있는 웹툰을 추가하려고 한다면 이미 관심 웹툰에 존재한다는 문구를 출력하며 추가를 허용하지 않는다.

만약 관심 웹툰에 추가가 되면, 잘 추가되었다는 문구를 띄우며 나의 관심목록(favorite)에 추가되었음을 알 수 있다. 이는 Main에서 favorite을 눌러 나의 관심목록을 확인해 보면 다음과 같은 창에서 확인할 수 있다.



[그림 4-6] Favorite webtoon page

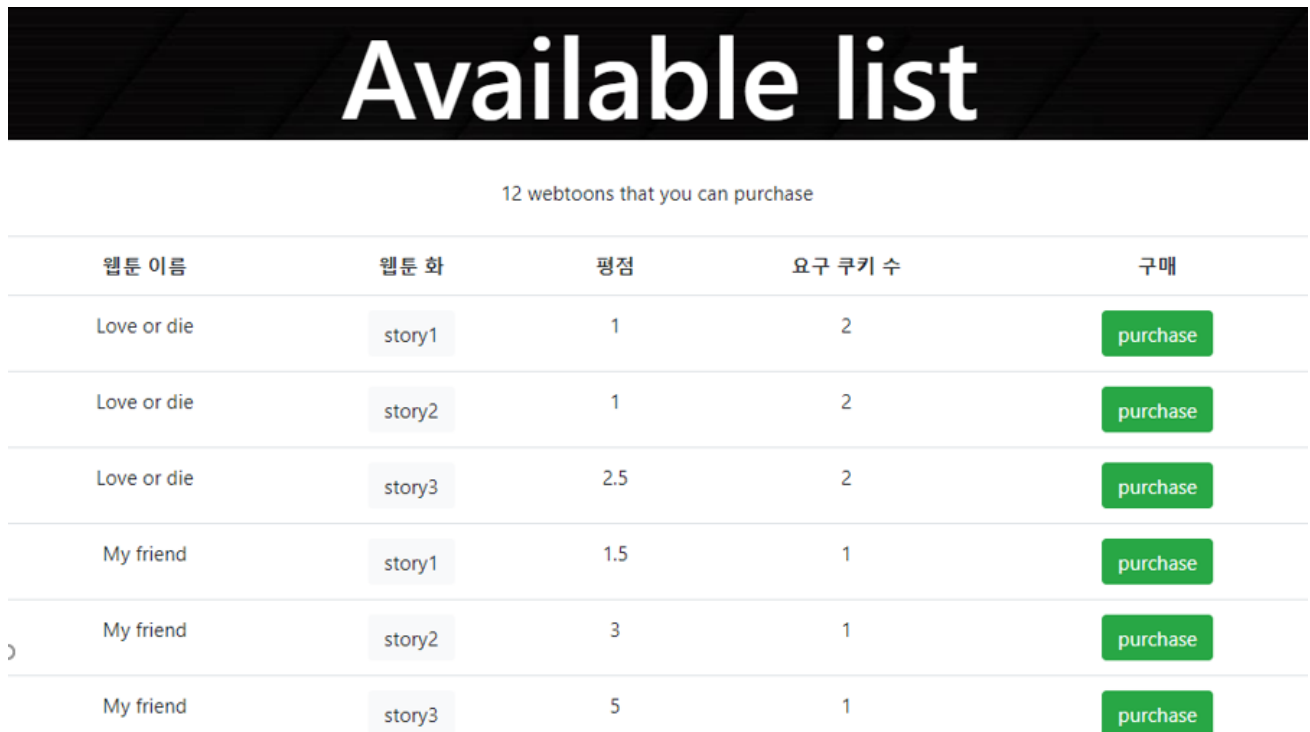
위 리스트는 favorite 테이블에 있는 웹툰들을 가져오는데, favorite 테이블에서는 웹툰의 아이디만 알 수 있으므로 웹툰 정보를 가지고 있는 webtoon 테이블과 natural join을 함으로 인해 웹툰의 추가적인 정보 역시 가져온다. 여기서 웹툰의 이름을 누르는 경우에도 웹툰을 검색한 후에 그 이름을 누르는 것과 같다.

4) Function 4: Purchase webtoon

모든 웹툰은 구매 후에 볼 수 있으며, 그 전까지는 볼 수 없다. 웹툰 구매에는 쿠키라는 재화가 사용된다. 다음과 같은 두 가지 방법으로 웹툰 구매 페이지를 볼 수 있다.

- (1) 그림 [4-5]에서처럼 웹툰 화 목록을 본 경우
- (2) 메인 화면에서 purchase 버튼을 눌러 구매 가능 웹툰 목록을 본 경우

(2)번과 같은 경우는, 사용자가 아직 구매하지 않은 모든 웹툰 목록을 보여줬다. 이를 위해 `not exist` 문과 `select` 문을 이용해, `webtoon_content` table에서 아직 `bought`에 있지 않은 `row` 들을 뽑아냈다.



웹툰 이름	웹툰 화	평점	요구 쿠키 수	구매
Love or die	story1	1	2	<button>purchase</button>
Love or die	story2	1	2	<button>purchase</button>
Love or die	story3	2.5	2	<button>purchase</button>
My friend	story1	1.5	1	<button>purchase</button>
My friend	story2	3	1	<button>purchase</button>
My friend	story3	5	1	<button>purchase</button>

[그림 4-7] Available list (하락)

웹툰을 구매하려면 `purchase` 버튼을 누르면 된다. 버튼을 누르면, 웹툰의 `id`, 몇 화인지를 나타내는 이름, 필요한 쿠키 수가 `php` 파일로 전송된다. `Php` 파일에서는 `select` 문을 이용해서는 웹툰의 정보를, `session`을 이용해서는 사용자의 정보를 가져온다.

그리고 가져온 정보들을 이용해 다음과 같은 두 가지를 확인한다.

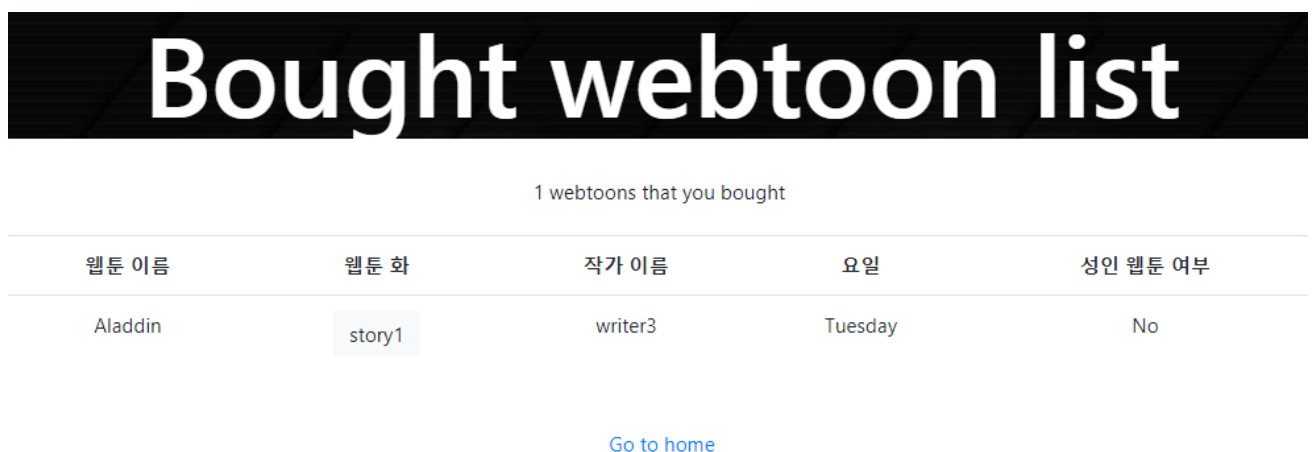
- (1) 웹툰이 성인 웹툰이면서, 사용자의 나이가 19세 미만인지를 체크한다. 이 경우에 해당한다면 이전과 같이 미성년자에게는 허락되지 않는다는 오류 메시지를 주면서 구매를 허용하지 않는다.
- (2) 유저의 잔여 쿠키 수가 웹툰의 쿠키 수보다 적은 경우에는, 쿠키가 부족하다는 메시지를 띄우며 구매를 허용하지 않는다.

위 두 가지 경우가 아닌 경우에는, `insert` 문을 이용해 구매 정보를 저장하는 `bought` 테이블에 구매 정보를 저장한다. 만약 미리 구매한 웹툰을 다시 구매하려고 한다면, `primary key constraint`를 위배하는 것이므로, 이미 구매했다는, 다시 살 수 없다는 오류 메시지를 보낸다.

웹툰을 정상적으로 구매했다면, 정상적으로 구매가 되었다는 문구와 구매가 된다. 이 경우에는 update문을 이용해 유저의 잔여 쿠키에서 웹툰 구매에 필요한 쿠키를 뺀 값으로 user 테이블을 업데이트 한다. 세션의 값 역시 업데이트를 해 주고 종료한다.

구매한 웹툰은 메인 화면에서 bought를 누를 경우 그 목록을 볼 수 있다. 해당 리스트는 bought 테이블에 있는 웹툰들을 가져오는데, bought 테이블에서는 웹툰의 아이디만 알 수 있으므로 웹툰 정보를 가지고 있는 webtoon 테이블과 natural join을 함으로 인해 웹툰의 추가적인 정보 역시 가져온다.

여기서 웹툰 화 이름을 누른 경우 웹툰을 실제로 볼 수 있다. 이를 위해서는 content/웹툰id_웹툰화 이름.jpg 파일을 보여주는 형식이고, 실제로 이 파일이 존재해야만 볼 수 있다.



[그림 4-8] Bought webtoon page

5) Function 5: Charge cookie & Info check

웹툰 구매에서 쿠키 부족으로 웹툰을 사지 못하거나 메인 화면에서 charge 버튼을 누르는 경우 웹툰 구매에 필요한 쿠키를 살 수 있는 페이지로 이동한다.

Money that you want to charge (1 cookie per 100 won) :

Charge Now!

[Go to home](#)

[그림 4-9] Charge page

충전할 금액을 입력하는 란에 값을 입력하면 100원당 1개의 쿠키를 얻을 수 있다. 100원 미만은 버림 한다.

값을 입력하면 해당 값을 100으로 나눈 값을 계산한 후, update문을 이용해 user table의 cookies 값을 업데이트하고, 세션의 값 역시 업데이트한다.

또한 메인 화면에서 info 버튼을 누르면 자신의 정보(이름, 나이, 그리고 잔여 쿠키 수)를 확인할 수 있다. 이 역시 select 문으로 구현 되어있다.

Information	
이름	junhyuk
나이	16
쿠키 수	5

[그림 4-7] Info page

충전을 하는 경우 위 쿠키수가 증가하는 것을, 웹툰 구매 시 쿠키수가 감소하는 것을 확인할 수 있다.

5. Challenges and failures

1) Failures

- (1) 작가를 통한 검색: 이름을 통해 검색하는 것과 같이 했다면 어렵지는 않았지만 시간이 부족했다.
- (2) 이름 순 웹툰 정렬: 모든 웹툰의 리스트를 보이는 것은 가능하다. 여기서 정렬 기능은 order by로 가능할 듯 했으나 역시 시간 문제로 완성하지 못했다.
- (3) 평점 순 웹툰 정렬: 각 화들의 평점을 모두 다 합쳐야 하고, 이를 저장하는 새로운 attribute를 만드는 과정이 필요해서 구조적 문제 때문에 시간이 부족했다.
- (4) 요일별로 나누기: Group by를 통해 쉽게 할 수는 있었겠지만, 이는 html 파일로 display하는 것이 또 다른 어려운 문제였다.

2) Challenges

- (1) constraint: 나이가 0보다 커야 한다, 음수를 충전할 수 없다 등의 데이터를 consistent하게 만드는 제약 조건을 넣지 못함. Check를 통해 넣을 수 있었겠지만, 역시 table을 통째로 바꿔야 해서 시간이 촉박했습니다.
- (2) Further functions: 위에서 언급하지 않은 관심 웹툰 삭제 등 데이터를 삭제하는 내용 혹은 관리자 모드를 만들어 웹툰을 직접 추가하는 내용도 넣고 싶었으나, 여유롭지 못한 시간 덕분에 기존에 구현하고 싶었던 내용들도 다 만들지 못했습니다. 추후에 시간이 나면 마저 완성해 보겠습니다.