

# 도서관 대여 서비스 프로젝트

**2기 – 2반 김민성**

# 0. Intro : 사용한 프로그램과 패키지

## 개발 환경: Visual Studio Code

### Python

- Flask
- SQLAlchemy

### HTML

- Vanilla HTML
- Jinja2
- CSS
- JavaScript

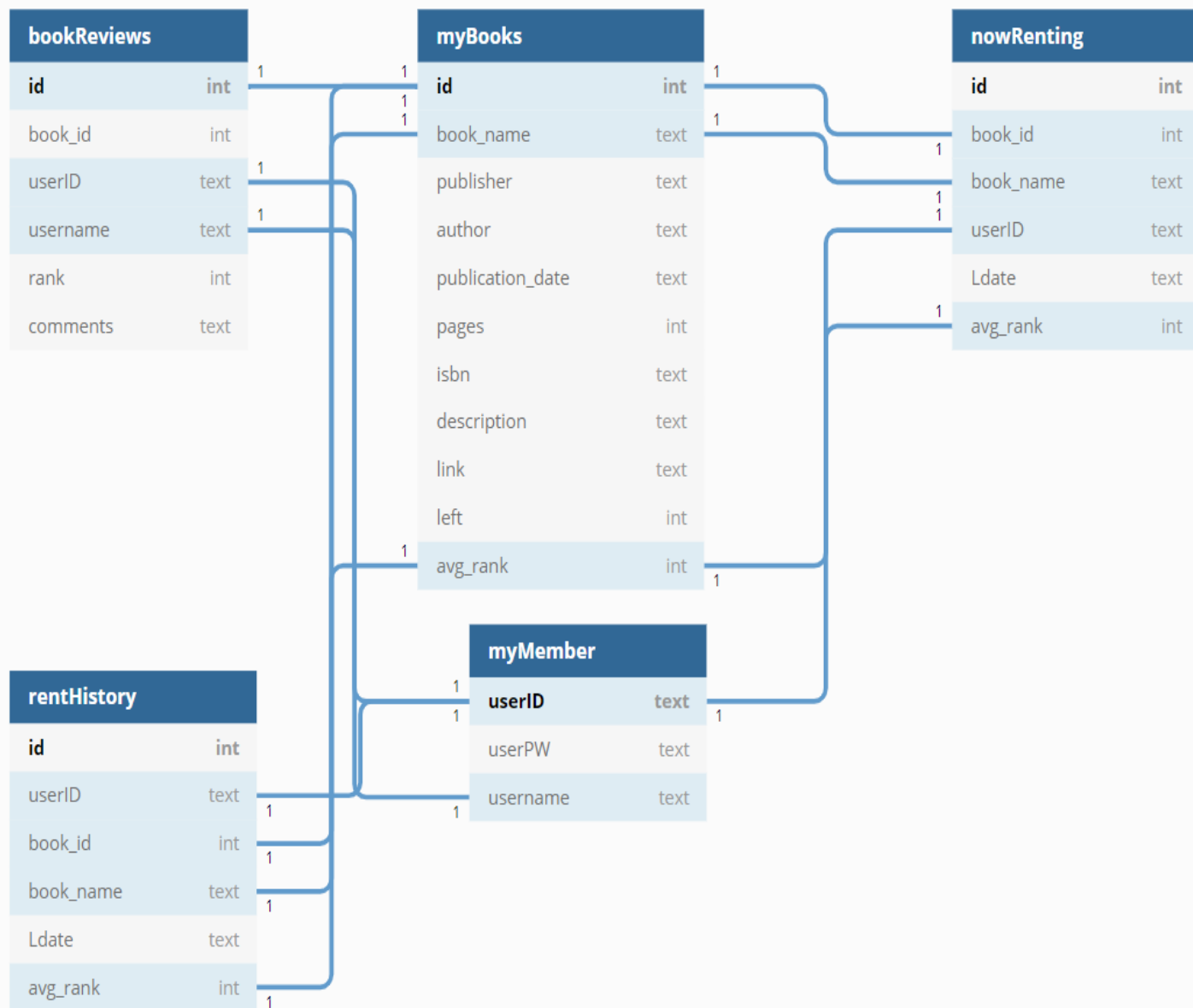
### DB

- DB Browser for SQLite

# 0. Intro : 전체 디렉토리 구조

gitty	
app.py	- Flask 서버 구동 및 기본 configuration이 들어있는 파일
db_connect.py	- SQLAlchemy 패키지를 db 객체에 복제하는 파일
models.py	- db에 맞춘 테이블, 컬럼이 선언된 파일
MyERD.jpg	- ERD 구조를 보여주는 파일
MyLibrary.db	- 프로젝트 전체에서 사용할 데이터베이스
README.md	- README 파일
_blueprints.py	- 전반적인 기능이 모두 들어있는 블루프린트 파일
static	
myCSS.css	- CSS 속성 파일
book_img	- 책 이미지가 든 폴더
js	- 메인페이지에 별점 표시를 하기 위한 js파일
star.js	- 반납페이지에 별점 표시를 하기 위한 js파일
star_bannap.js	- 대여기록 페이지에 별점 표시를 하기 위한 js파일
star_history.js	
templates	
bannap.html	- 반납하기 페이지
basic.html	- 모든 페이지가 상속받는 뼈대 페이지
book_intro.html	- 책 소개 페이지
history.html	- 대여 기록 페이지
login.html	- 로그인 페이지
main.html	- 메인 페이지
submit.html	- 회원가입 페이지
_navbar.html	- 내비게이션 바(basic.html에 등록하여 상속시킴)
__pycache__	

# 0. Intro : ERD



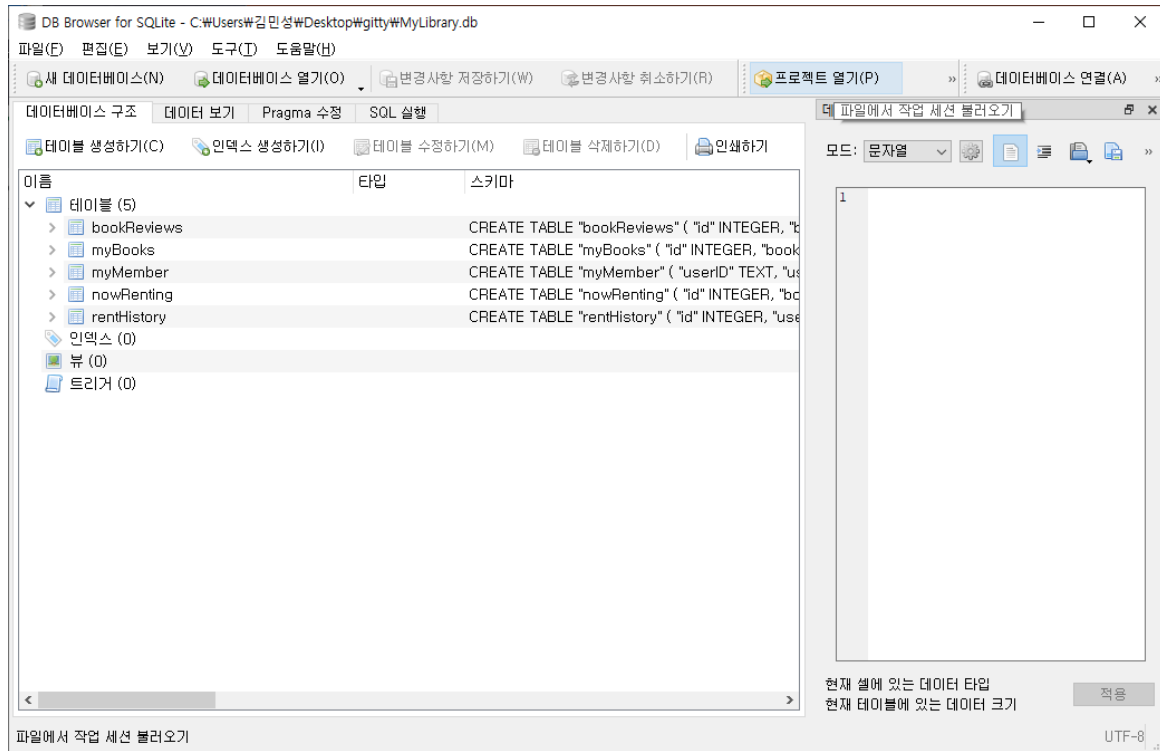
## 테이블 설명

- myMember : 회원 정보 테이블
- myBooks : 도서관에 비치된 책 정보 테이블
- bookReviews : 모든 책에 대한 모든 리뷰 기록 테이블
- rentHistory : 모든 회원에 대한 모든 대여 기록 테이블
- nowRenting : 회원이 대여한 책 정보 테이블

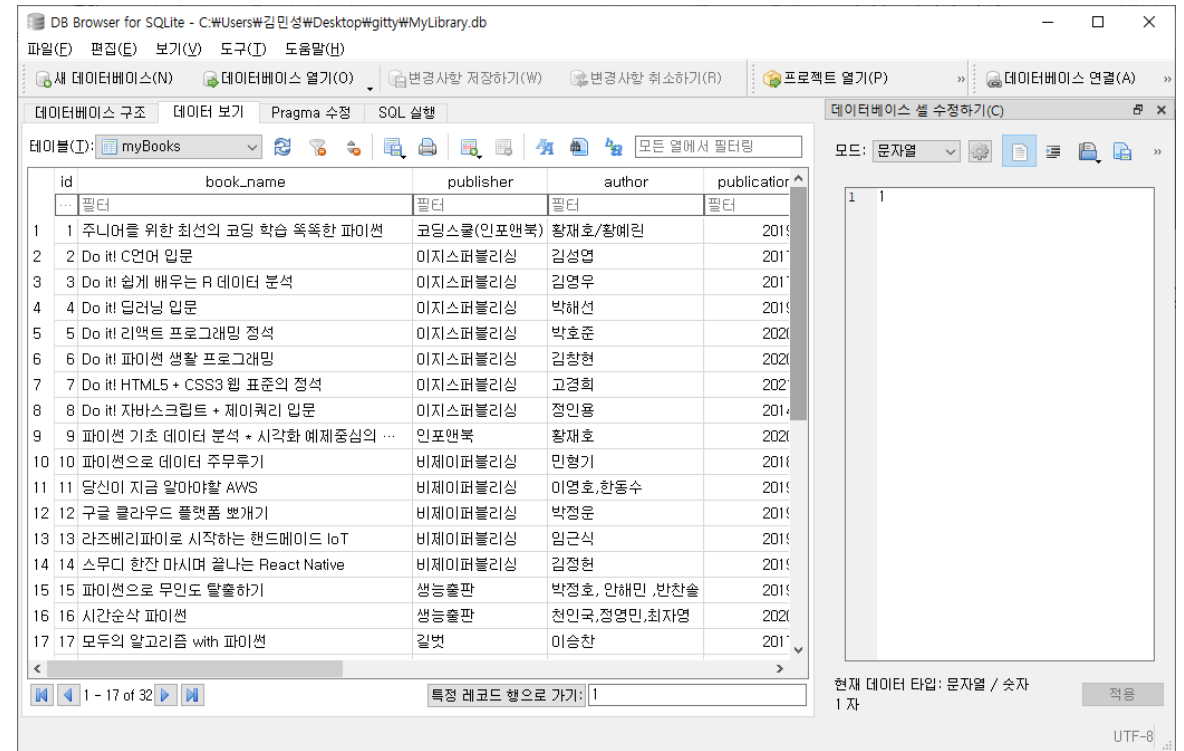
# 1. 프로젝트 진행 경과 : DB 구축하기

- 최우선적으로 데이터베이스 구축이 필요하다 생각하여 가장 먼저 작업함
- DB 구축에 사용된 프로그램 : DB Browser for SQLite
- DB Browser for SQLite: <https://sqlitebrowser.org/>
- .db파일을 손쉽게 작성할 수 있음 (테이블 작성부터 데이터 삽입까지)
- MySQL을 이용해 연결하는 데 익숙치 못해 구글링하여 찾은 프로그램

# 1. 프로젝트 진행 경과 : DB 구축하기



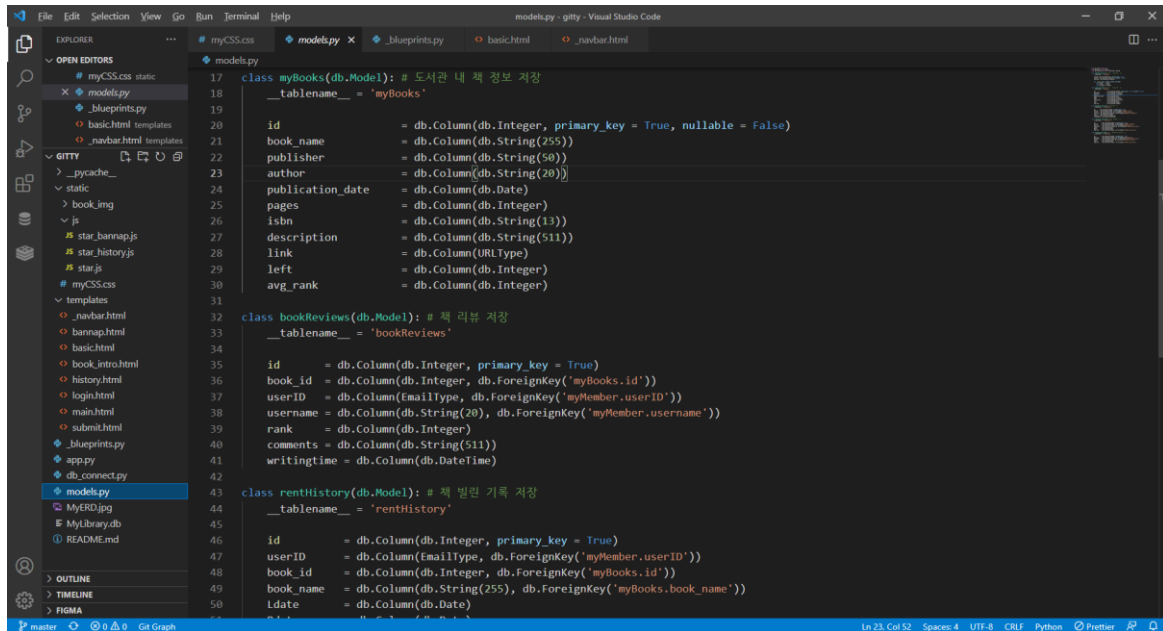
▲ 테이블을 만든 상태



▲ 원본 .csv파일을 임포트해 테이블에 맞게 DB를 생성

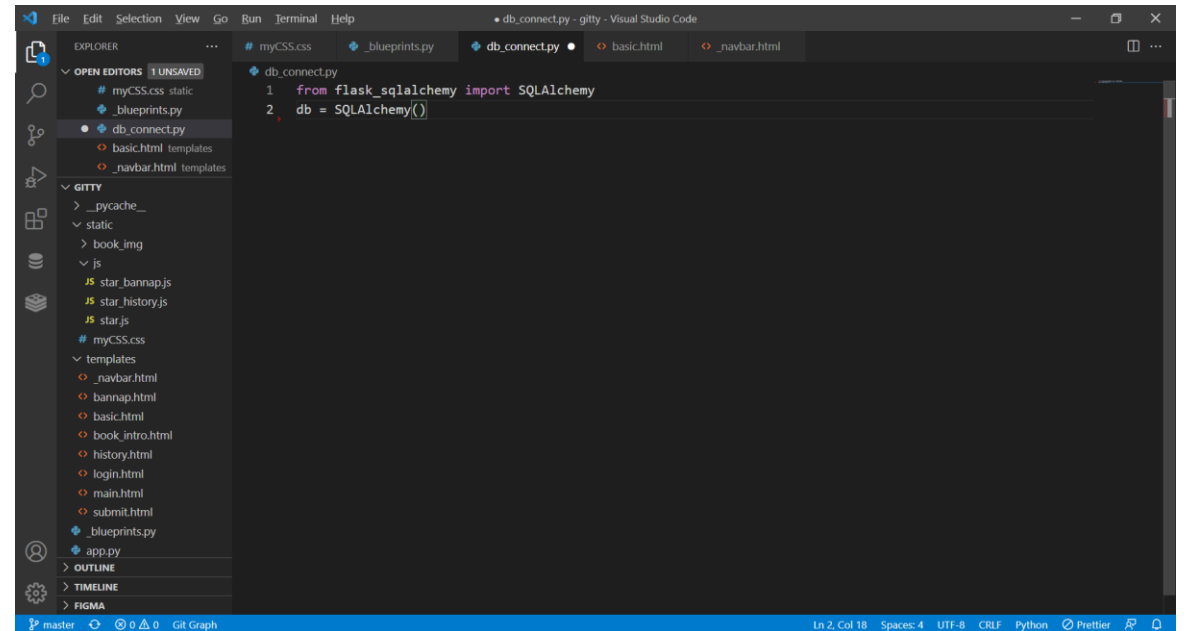
# 1. 프로젝트 진행 경과 : Models.py 작성

- DB 테이블에 맞게 테이블과 컬럼 선언
- SQLAlchemy를 간단히 import하기 위해 db\_connect.py 파일 생성



```
17 class myBooks(db.Model): # 도서권 내 책 정보 저장
18     __tablename__ = 'myBooks'
19
20     id = db.Column(db.Integer, primary_key = True, nullable = False)
21     book_name = db.Column(db.String(255))
22     publisher = db.Column(db.String(50))
23     author = db.Column(db.String(20))
24     publication_date = db.Column(db.Date)
25     pages = db.Column(db.Integer)
26     isbn = db.Column(db.String(13))
27     description = db.Column(db.String(511))
28     link = db.Column(db.URLType)
29     left = db.Column(db.Integer)
30     avg_rank = db.Column(db.Integer)
31
32 class bookReviews(db.Model): # 책 리뷰 저장
33     __tablename__ = 'bookReviews'
34
35     id = db.Column(db.Integer, primary_key = True)
36     book_id = db.Column(db.Integer, db.ForeignKey('myBooks.id'))
37     userID = db.Column(db.Integer, db.ForeignKey('myMember.userID'))
38     username = db.Column(db.String(20), db.ForeignKey('myMember.username'))
39     rank = db.Column(db.Integer)
40     comments = db.Column(db.String(511))
41     writingtime = db.Column(db.DateTime)
42
43 class rentHistory(db.Model): # 책 빌린 기록 저장
44     __tablename__ = 'rentHistory'
45
46     id = db.Column(db.Integer, primary_key = True)
47     userID = db.Column(db.Integer, db.ForeignKey('myMember.userID'))
48     book_id = db.Column(db.Integer, db.ForeignKey('myBooks.id'))
49     book_name = db.Column(db.String(255), db.ForeignKey('myBooks.book_name'))
50     ldate = db.Column(db.Date)
```

▲ Models.py 파일



```
1 from flask_sqlalchemy import SQLAlchemy
2 db = SQLAlchemy()
```

▲ db\_connect.py 파일

# 1. 프로젝트 진행 경과 : app.py 작성

- 기본 configuration이 들어있는 app.py를 작성
- 절대 경로에서 db파일을 가져올 수 있도록 설정

```
app.py
1  from flask import Flask
2  from db_connect import db
3  import os
4  import _blueprints
5
6  def create_app():
7      app = Flask(__name__)
8      BASE_DIR = os.path.dirname(__file__) # 폴더 구조가 달라져도 현재 폴더를 가져와서 사용 가능토록 설정
9
10     app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///{}'.format(os.path.join(BASE_DIR, 'MyLibrary.db'))
11     app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
12     app.config['SESSION_TYPE'] = 'filesystem'
13     app.config['SECRET_KEY'] = "ejrwrjsq32354621"
14
15     db.init_app(app) # db와 app 연결
16     app.register_blueprint(_blueprints.bp) # 블루프린트 가져오기
17
18     return app
19
20 if __name__ == "__main__":
21     create_app().run(debug = True, port = 5000)
```

▲ app.py 파일



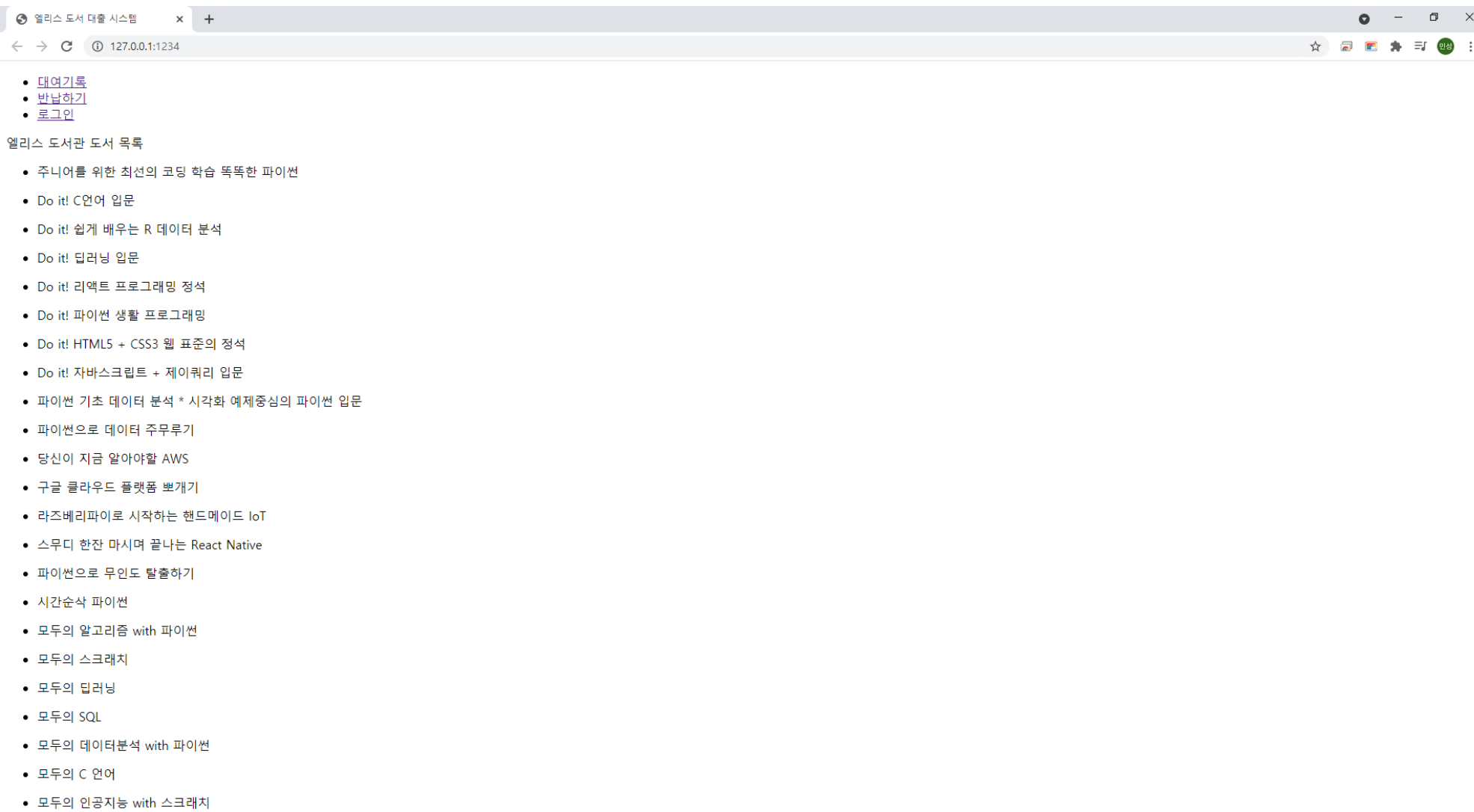
# 1. 프로젝트 진행 경과 : Back to Front

## 첫째 주(08.16 ~ 08.21) 작업 내용 요약

- 우선적으로 백엔드 Flask 코드 부분부터 작업
- 기능부터 구현하고 속성 및 페이지에 기능 적용을 하는 게 더 쉽다고 판단
- HTML 파일은 연결만 해 두고 코드 완성 시 하나하나 실행하며 기능 테스트
- 파일은 app.py와 \_blueprint.py 2개로 나눔
- app.py : 구동할 때 기초적인 configuration만 설정한 파일
- \_blueprint.py: 전체적인 기능이 담긴 파일

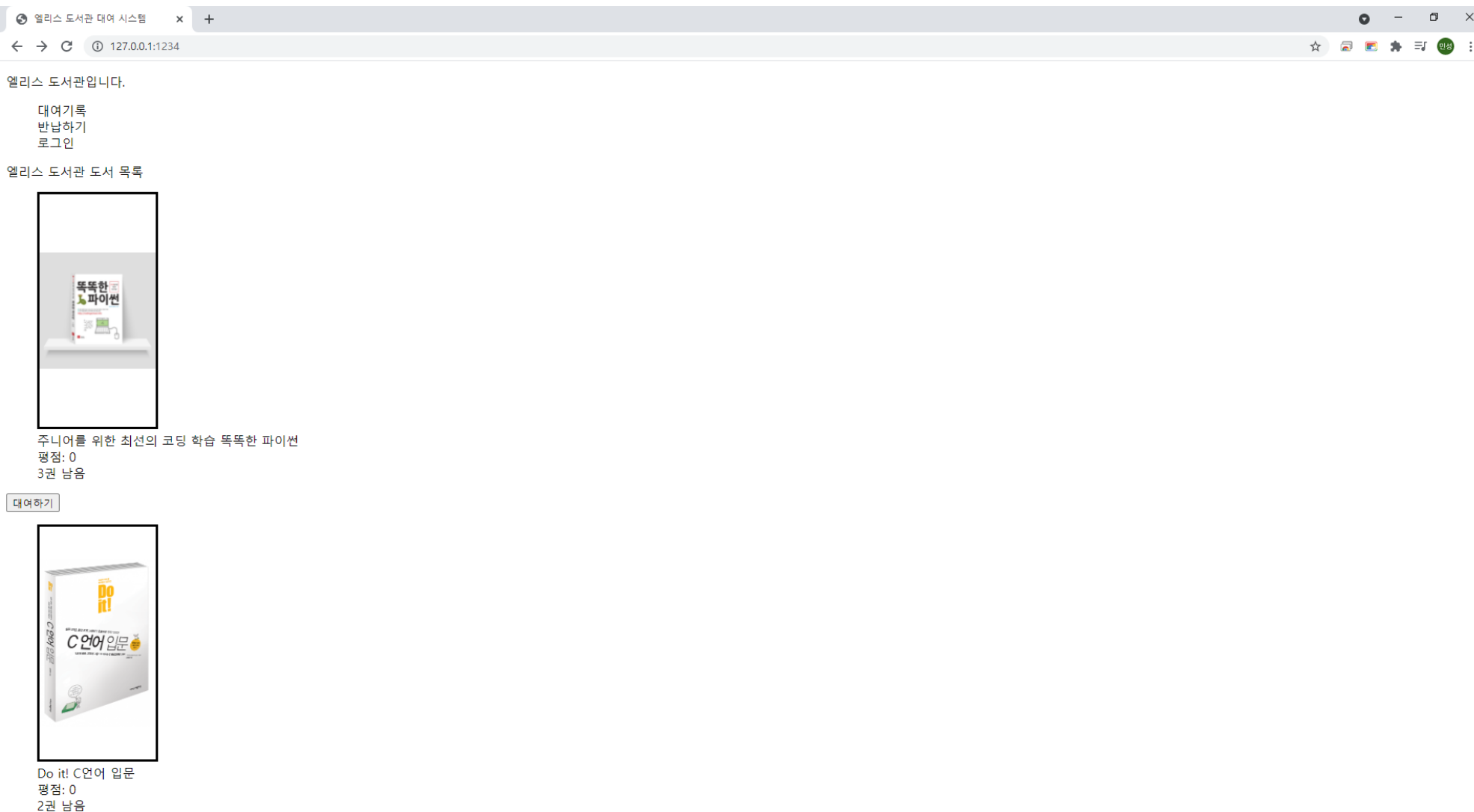
# 1. 프로젝트 진행 경과 : Back to Front

## 8.16 스켈레톤 코드 작성만 한 상태



# 1. 프로젝트 진행 경과 : Back to Front

## 8.20 메인페이지에 책 표지, 대여하기 버튼 출력



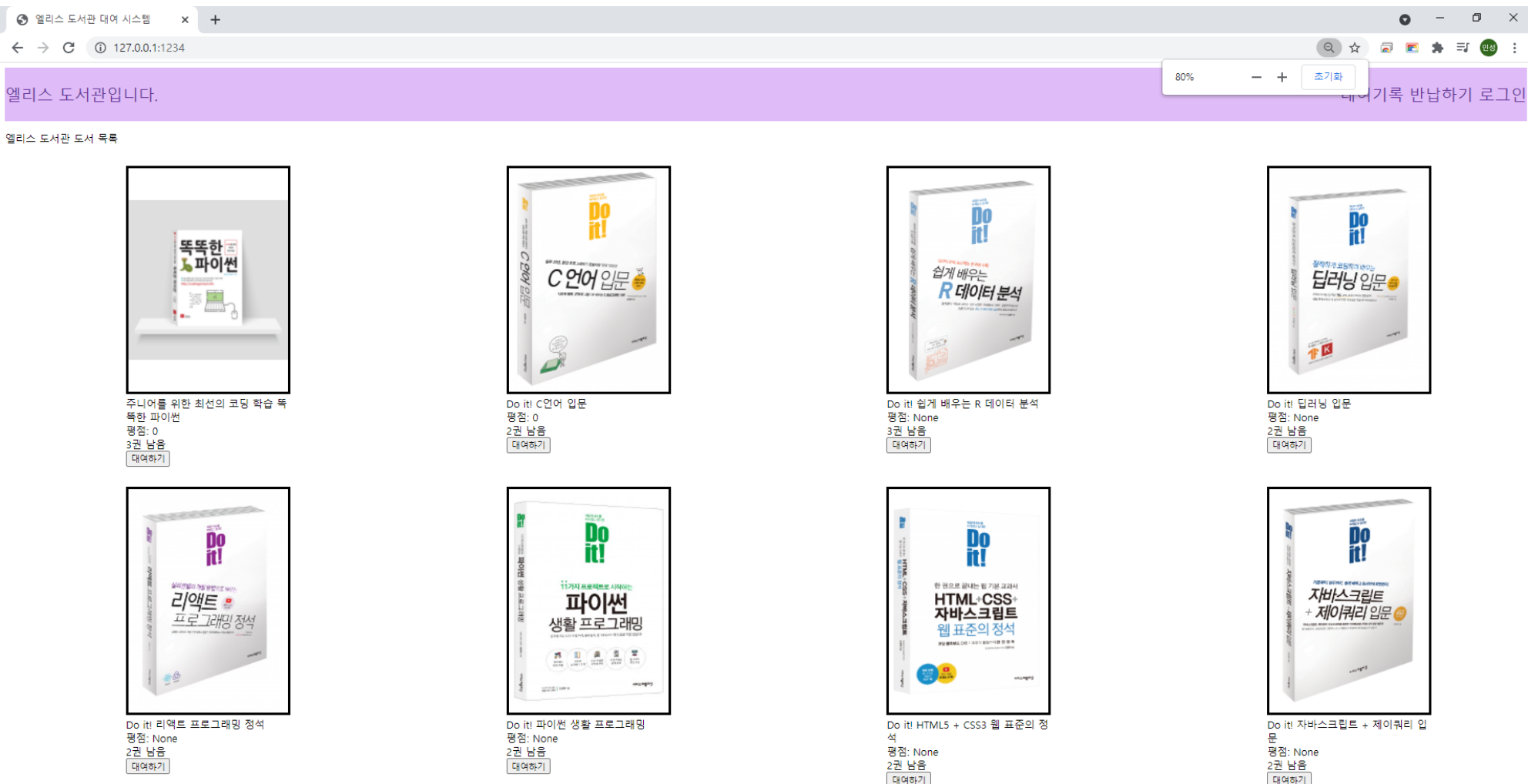
# 1. 프로젝트 진행 경과 : Back to Front

## 둘째 주(08.22 ~ 08.27) 작업 내용 요약

- 실제 유저가 보는 부분인 HTML 구조를 잡음
- Bootstrap을 이용하는 방법도 생각해 봤지만 복잡한 것 같아서 패스
- HTML 파일 요소 내 CSS 속성을 하나하나 설정
- 백엔드에서 구현하기 힘든 부분은 JavaScript를 이용
- 웹사이트를 직접 이용해보며 예기치 못한 동작 확인 및 코드 수정
- 거의 완성이 되었을 때 가상머신에 Git을 이용하여 프로젝트를 업로드

# 1. 프로젝트 진행 경과 : Back to Front

## 8.22 메인페이지 Grid 설정, 페이지네이션



# 1. 프로젝트 진행 경과 : Back to Front

## 8.27 마무리 작업 완료

엘리스 도서관 대여 시스템

← → ↺ 127.0.0.1:5000

67% - + 초기화

엘리스 도서관입니다. 대여기록 반납하기 로그인

엘리스 도서관 도서 목록

주니어를 위한 최선의 ...  
★★★★☆  
3권 남음  
대여하기

Do it! C언어 입문  
★★★★☆  
2권 남음  
대여하기

Do it! 쉽게 배우는 R ...  
★★★★☆  
9권 남음  
대여하기

Do it! 딥러닝 입문  
★★★★☆  
2권 남음  
대여하기

Do it! 리액트 프로그...  
☆☆☆☆☆  
2권 남음  
대여하기

Do it! 파이썬 생활 프...  
★★★★☆  
2권 남음  
대여하기

Do it! HTML5 + CSS...  
☆☆☆☆☆  
2권 남음  
대여하기

Do it! 자바스크립트 + ...  
★★★★☆  
2권 남음  
대여하기

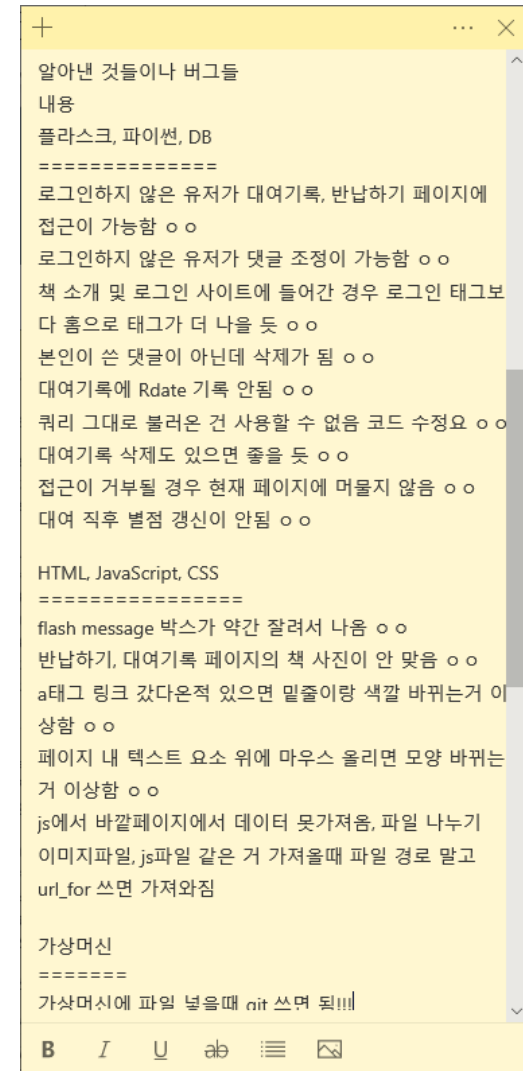
1 2 3 4

## 2. 배운 점 : Self-Feedback

### 버그 및 알아낸 점 기록하기

- 진행하는 동안 알아낸 것들이나 예기치 못한 실행결과 기록
- 매일 작업을 시작할 때 회고록 작성과 같이 참고하니 도움이 됨

그동안 기록한 것들 ►



## 2. 배운 점 : Googling

### 구글링으로 해결한 것들

1. DB Browser : <https://sqlitebrowser.org/>
2. flash message 띄우기  
<https://flask.palletsprojects.com/en/2.0.x/patterns/flashing/>  
<https://flask-docs-kr.readthedocs.io/ko/latest/patterns/flashing.html>
3. url\_for 사용법  
<https://exchangeinfo.tistory.com/65>  
[https://runebook.dev/ko/docs/flask/api/index#flask.url\\_for](https://runebook.dev/ko/docs/flask/api/index#flask.url_for)
4. CSS grid 정렬 : <https://studiomeal.com/archives/533>
5. CSS 자동 줄바꿈 금지 및 내용 많으면 ... 표시 : <https://ggmouse.tistory.com/54>
6. CSS 마우스 커서 모양 설정 : [https://www.w3schools.com/cssref/tryit.asp?filename=trycss\\_cursor](https://www.w3schools.com/cssref/tryit.asp?filename=trycss_cursor)
7. CSS 버튼 만들 수 있는 웹사이트 : <https://www.bestcssbuttongenerator.com/>
8. Pagination  
[https://flask-sqlalchemy.palletsprojects.com/en/2.x/api/?highlight=pagination#flask\\_sqlalchemy.Pagination](https://flask-sqlalchemy.palletsprojects.com/en/2.x/api/?highlight=pagination#flask_sqlalchemy.Pagination)  
<https://stackoverflow.com/questions/18468887/flask-sqlalchemy-pagination-error>  
<https://betterprogramming.pub/simple-flask-pagination-example-4190b12c2e2e>
9. SQLAlchemy 관련 에러 해결 : <https://stackoverflow.com/questions/51390141/sqlalchemy-returns-weird-data>
10. app.py에서 db파일을 절대경로로 호출해 가져오기: <https://wikidocs.net/81074>
11. Regex 작성할 수 있는 웹사이트 : <https://regexr.com/>
12. ERD 작성할 수 있는 웹사이트 : <https://dbdiagram.io/home>
13. MDN : <https://developer.mozilla.org/ko/>
- ...



### 3. 라이브 데모

<http://kdt-1st-project-52.koreacentral.cloudapp.azure.com:5000/>

## 4. Q & A

Q & A

끝. 2주간 모두 고생하셨습니다~!

감사합니다 😊