Database Design Project Final Report

Sidney Suen (1001525)

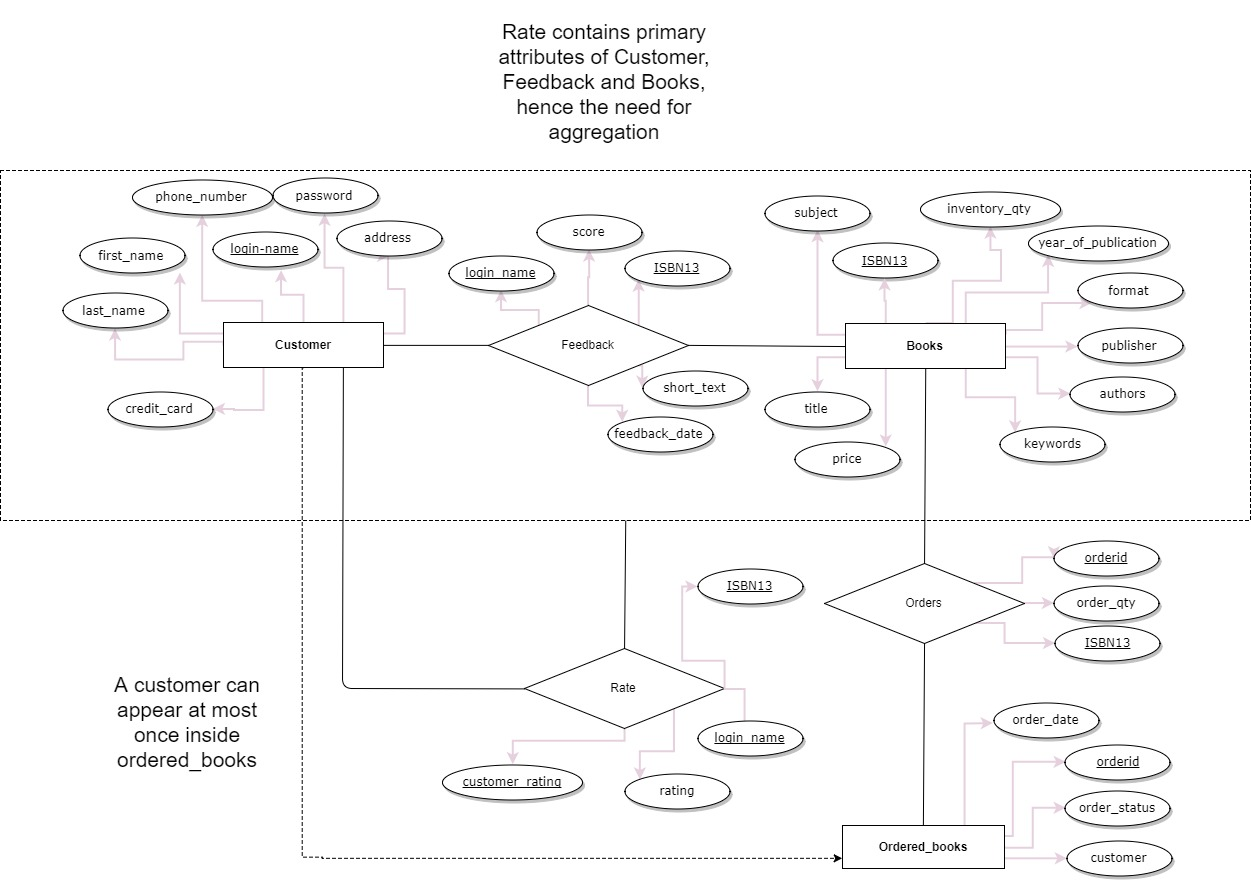
Archit Atul Date (1001695)

Amish Bhandari (1001614)

Arshi Dalvi (1001768)

Nickson Guay (1000998)

ER Diagram



Notes:

* Deleting customers was not specified in the project specs; most websites would archive all customer data nonetheless.
* Ordered\_books is a table mapping a user to an ordered, which uniquely identifies an order. Hence, a username corresponds to at most 1 orderid.
* For more details in running the code, please refer to the README file.

Relational Schema (SQL DDL Code)

#create database bookstore;

use bookstore;

create table Books (

ISBN13 char(17),

title char(50),

authors char(50),

publisher char(50),

year\_of\_publication integer,

inventory\_qty integer,

price numeric(6,2),

format char(9) check (format = 'hardcover' or format = 'softcover'),

keywords char(20),

subject char(20),

primary key (ISBN13));

create table Customers (

login\_name char(20),

password char(20),

first\_name char(20),

last\_name char(20),

credit\_card char(20),

address char(50),

phone\_number char(20),

primary key (login\_name));

create table Feedback (

ISBN13 char(17),

login\_name char(20),

score integer,

short\_text char (100),

feedback\_date date,

primary key (ISBN13, login\_name),

foreign key (ISBN13) references Books(ISBN13),

foreign key (login\_name) references Customers(login\_name));

create table Rate (

login\_name char(20),

ISBN13 char(17),

customer\_rating char(20),

rating integer check (rating = 0 or rating = 1 or rating = 2),

CONSTRAINT chk\_same CHECK (login\_name <> customer\_rating),

primary key (ISBN13,login\_name,customer\_rating),

foreign key (customer\_rating) references Customers(login\_name),

foreign key (login\_name,ISBN13) references Feedback(login\_name, ISBN13));

create table Ordered\_books (

orderid integer,

customer char(20) not null,

order\_date date,

order\_status char(10),

primary key (orderid),

foreign key (customer) references Customers(login\_name));

create table Orders (

orderid integer,

ISBN13 char(17),

order\_qty integer,

primary key (orderid,ISBN13),

foreign key (orderid) references Ordered\_books(orderid),

foreign key (ISBN13) references Books(ISBN13));

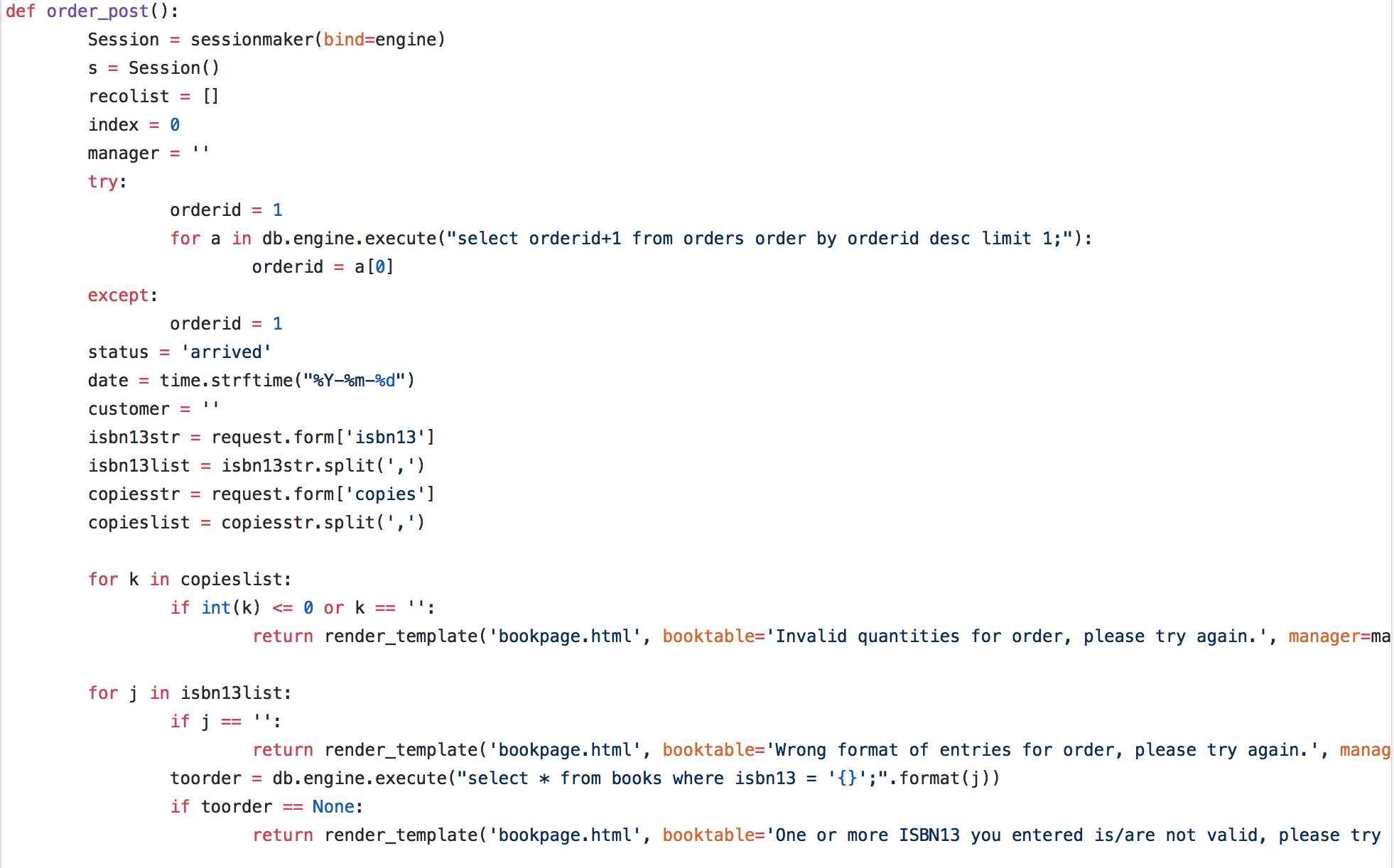
Implementation

1. Website visitor can register and login. Based on credentials they can either login as a user or a manager. The login name is checked for uniqueness.





1. After registration, a user can order one or more books. A user may order multiple copies of a book, one or more times.





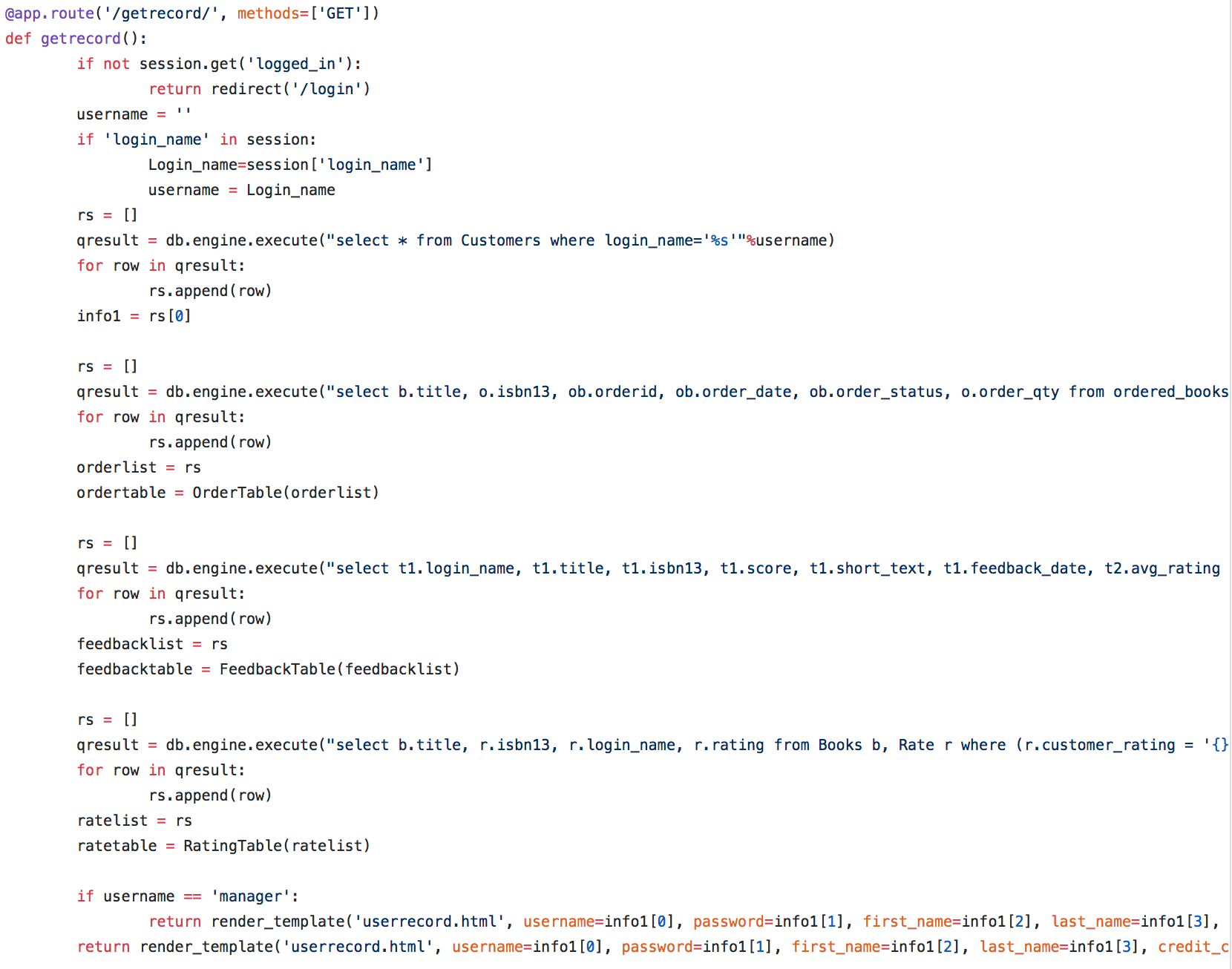
1. User records: Upon user demand, you should print the full record of a user:

• his/her account information

• his/her full history of orders (book name, number of copies, date etc.)

• his/her full history of feedbacks

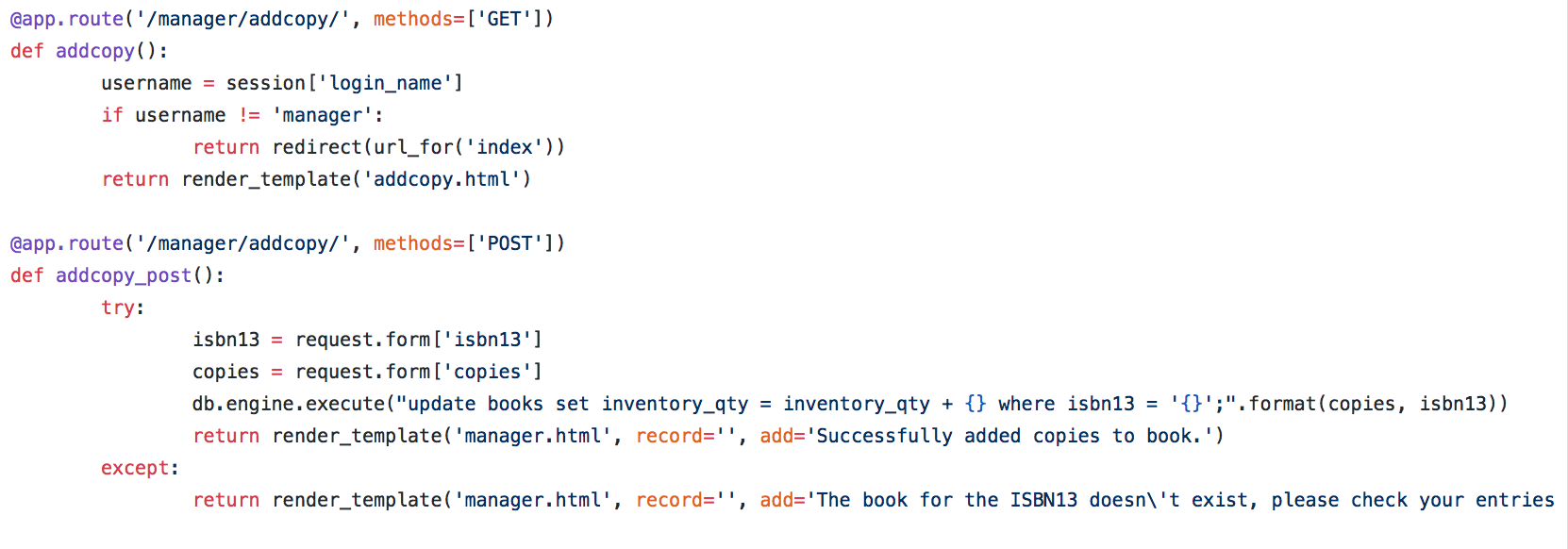
• the list of all the feedbacks he/she ranked with respect to usefulness



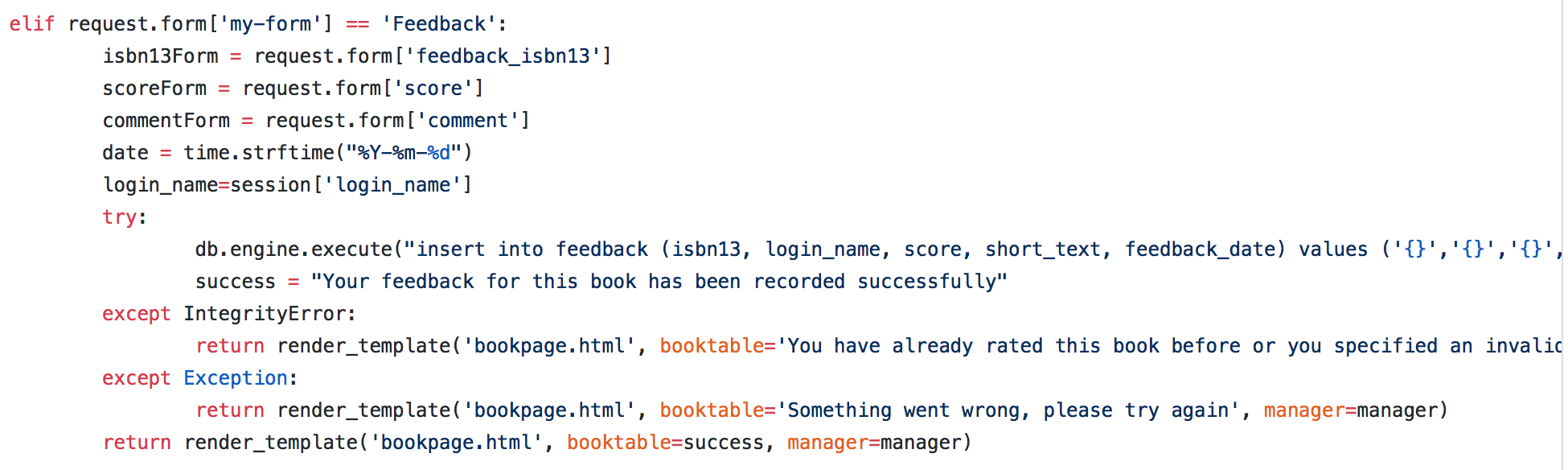
1. New book: The store manager can add new book into the database along with its details.



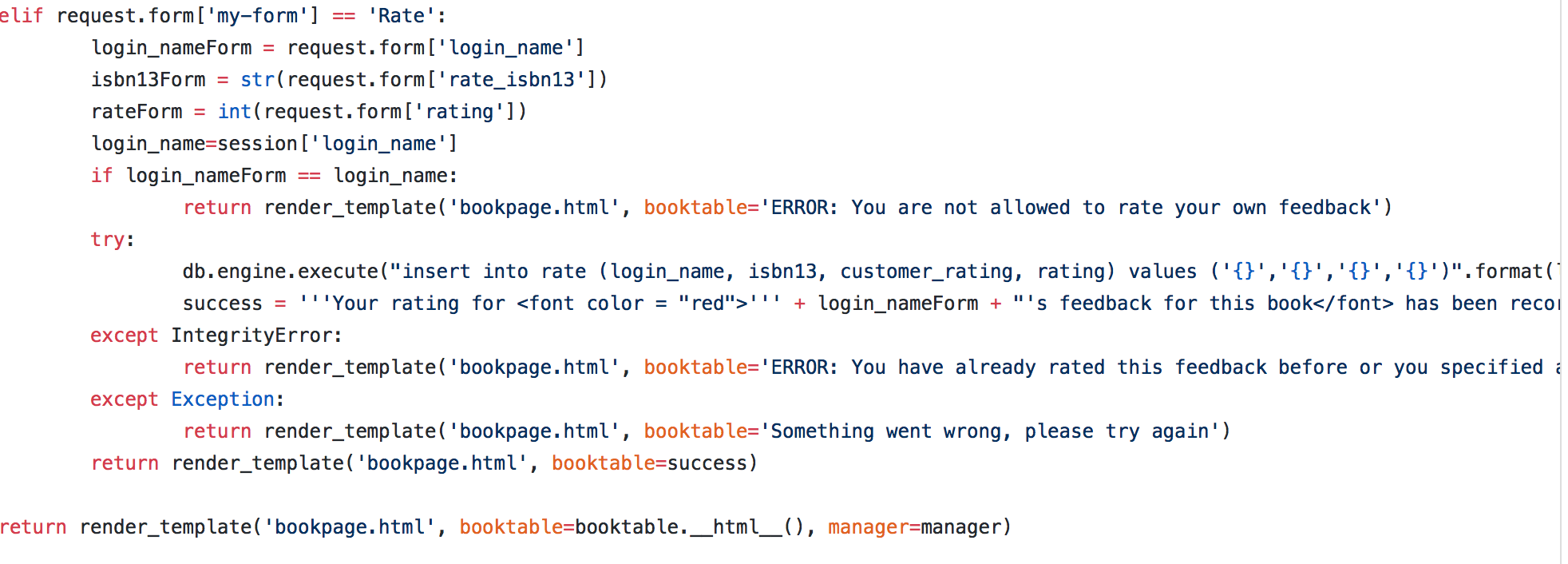
1. The manager can increment the number of books in the library



1. Feedback recordings: Users can record their feedback for a book. Details like date, numerical score and comments can be recorded. No changes is allowed and only one feedback per user is allowed.



1. Users can rate the usefulness of other’s feedback. Users are not allowed to rate their own feedback, and are not allowed to rate a feedback twice.



1. User will be able to browse books, and request for which books using conjunctive queries on the authors, and/or publisher, and/or title, and/or subject. Users can also specify results to be sorted by a) year, or b) by the average score of the feedbacks





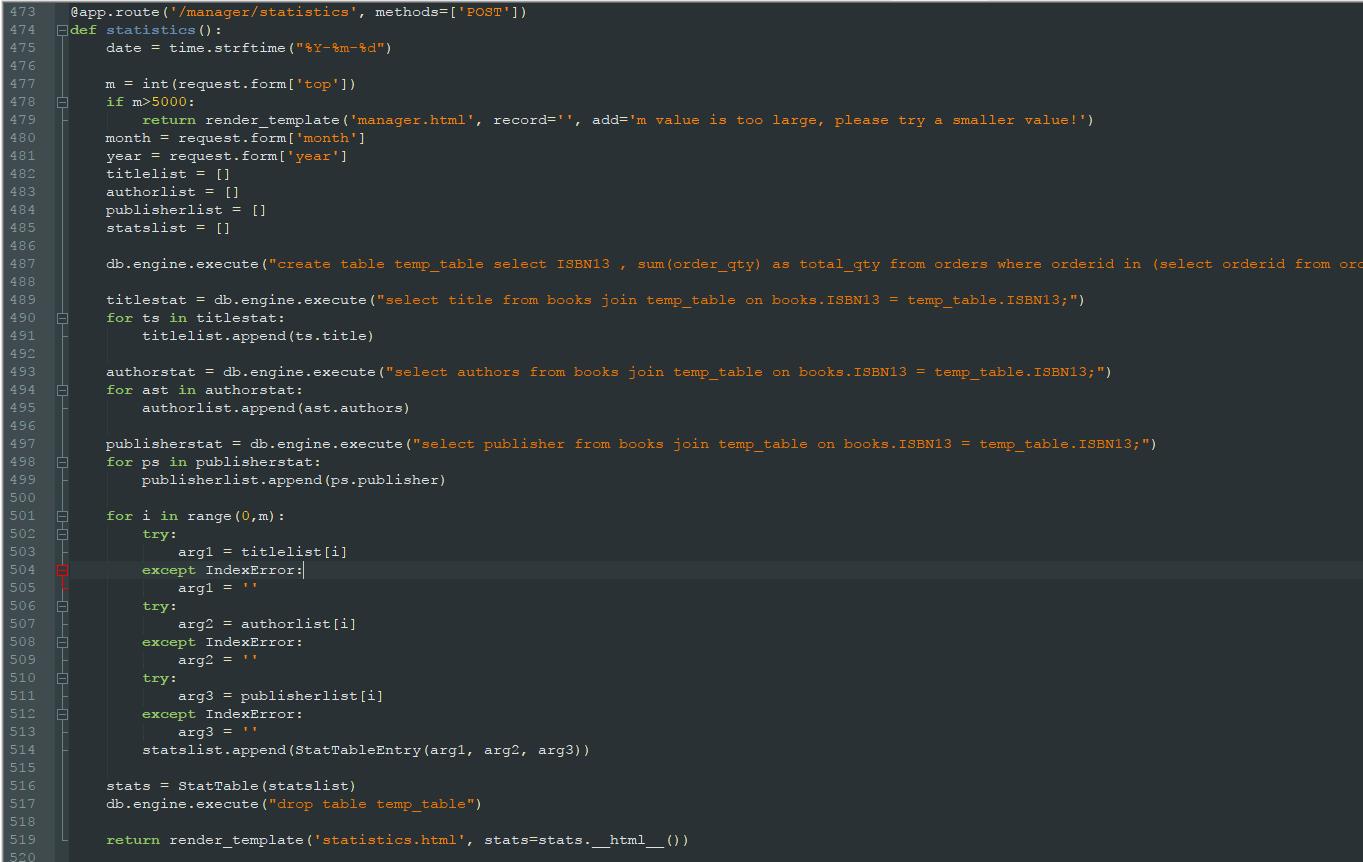
1. For a given book, user can ask for top n most useful feedback. N is specified by the user.



1. Book recommendation – when user orders copy of a book say book A, the website recommends a list of other suggested books. A book B is suggested if there exist a user that have purchased both book A and book B. The suggested books are sorted based on decreasing sales count.



1. Statistics – Every month the store manager wants
   * The list of the m most popular books (copies sold this month)
   * The list of m most popular authors
   * The list of m most popular publishers



**Executing the SQL query (line 487):**

db.engine.execute("create table temp\_table select ISBN13 , sum(order\_qty) as total\_qty from orders where orderid in (select orderid from ordered\_books where year(order\_date) = '%s' and month(order\_date) = '%s') group by ISBN13 order by total\_qty desc limit %s" % (year, month, m))

**Screen Dumps:**

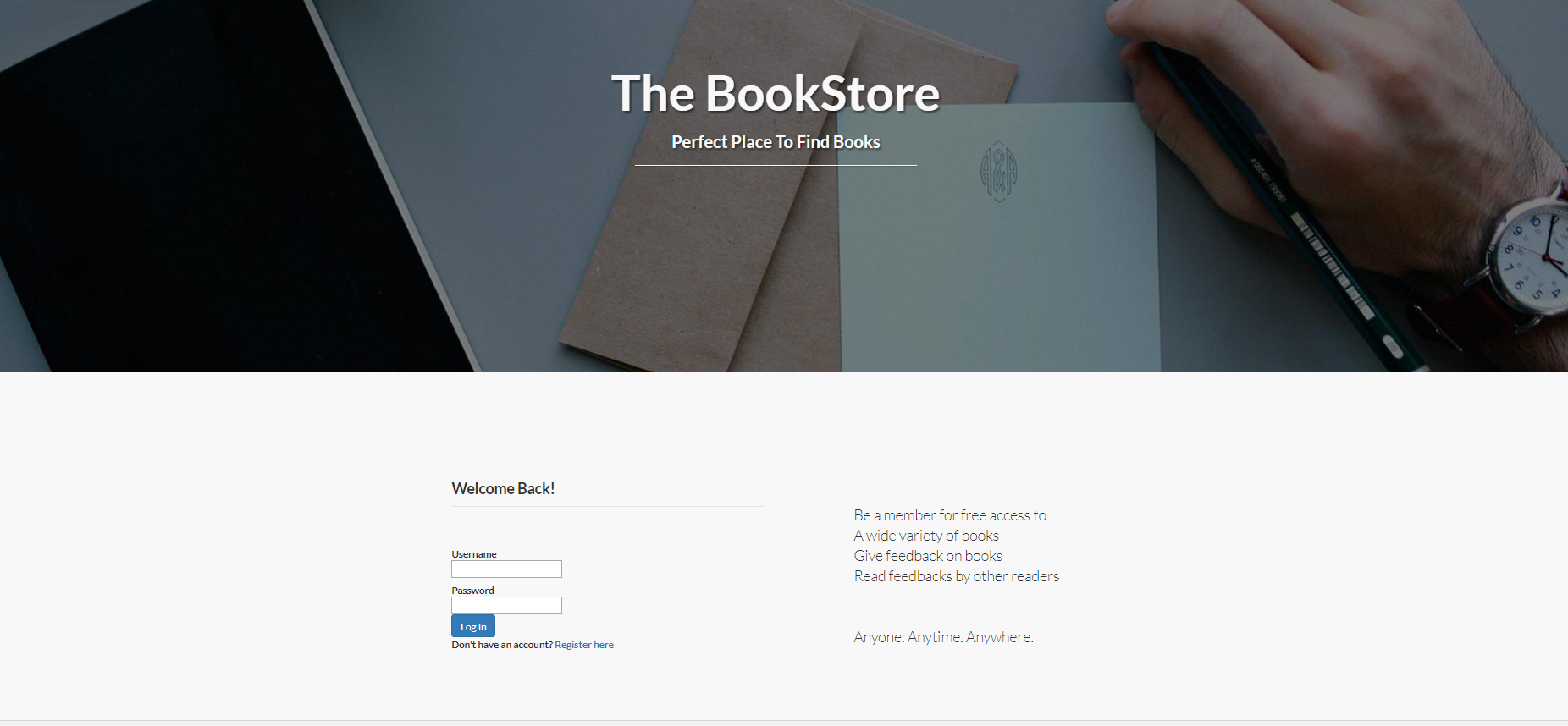


Figure 1: Login page

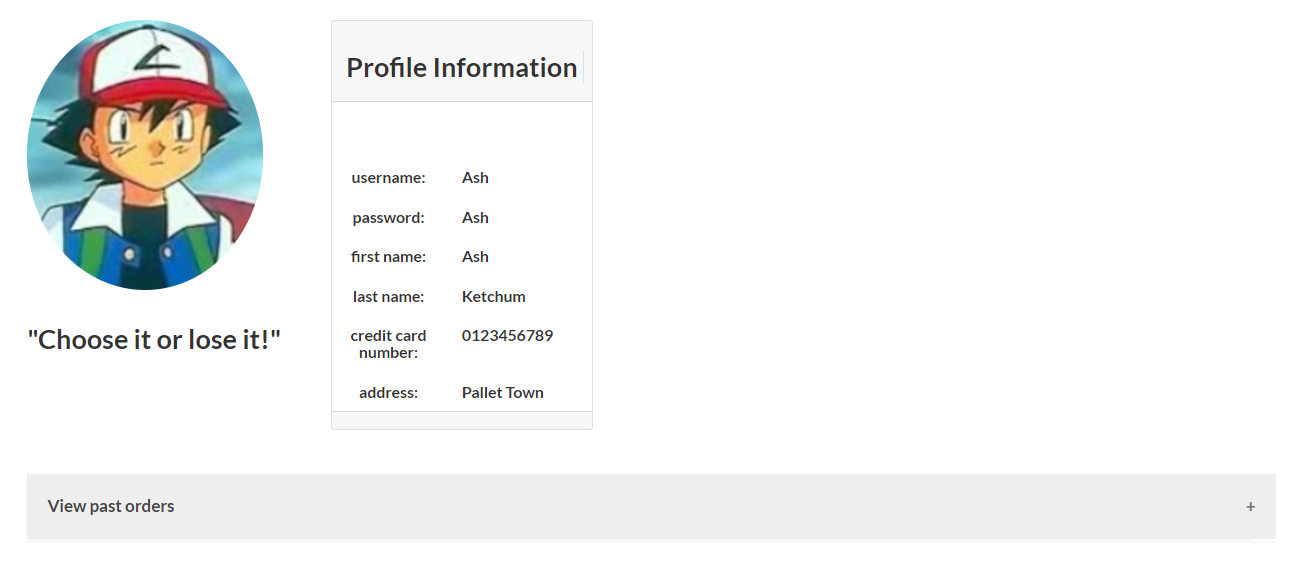


Figure 2: User profile

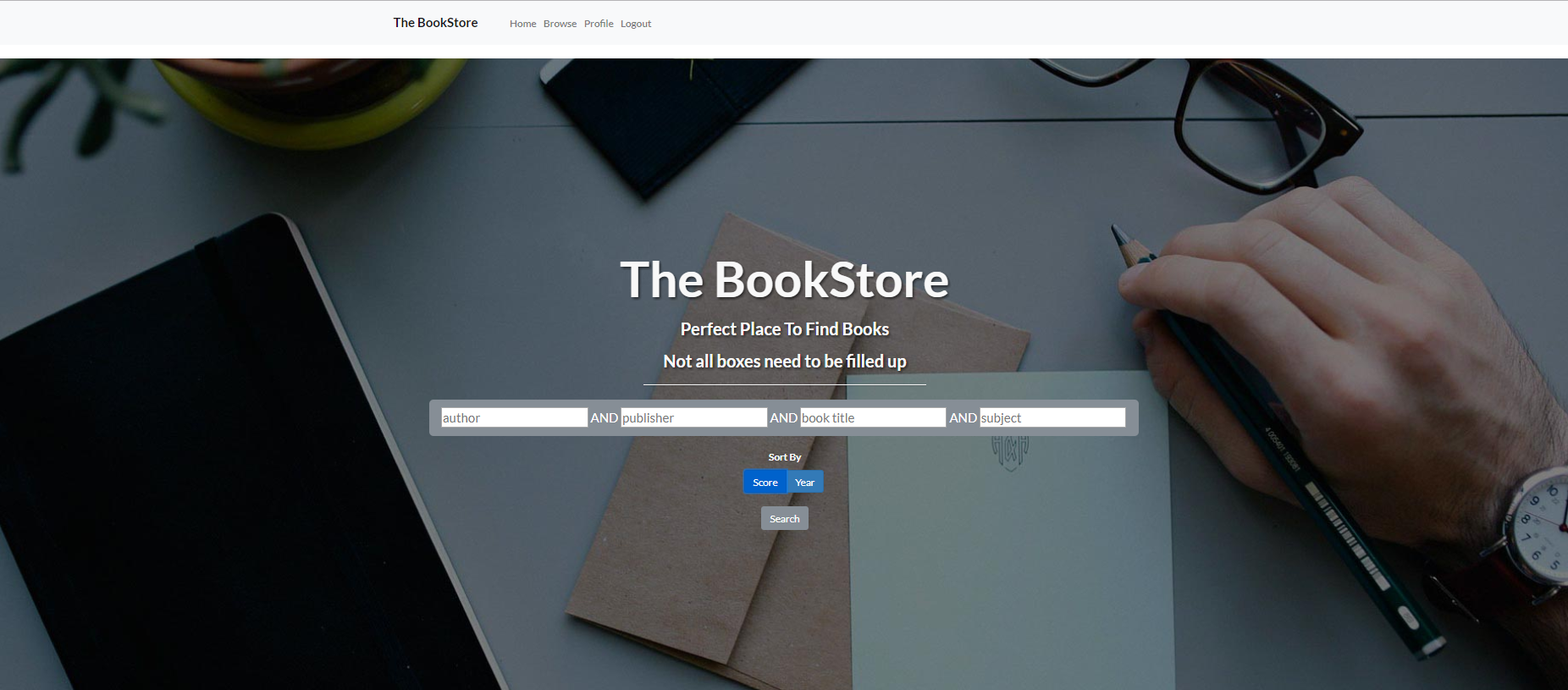


Figure 3: Browse books. Default selection will return all books

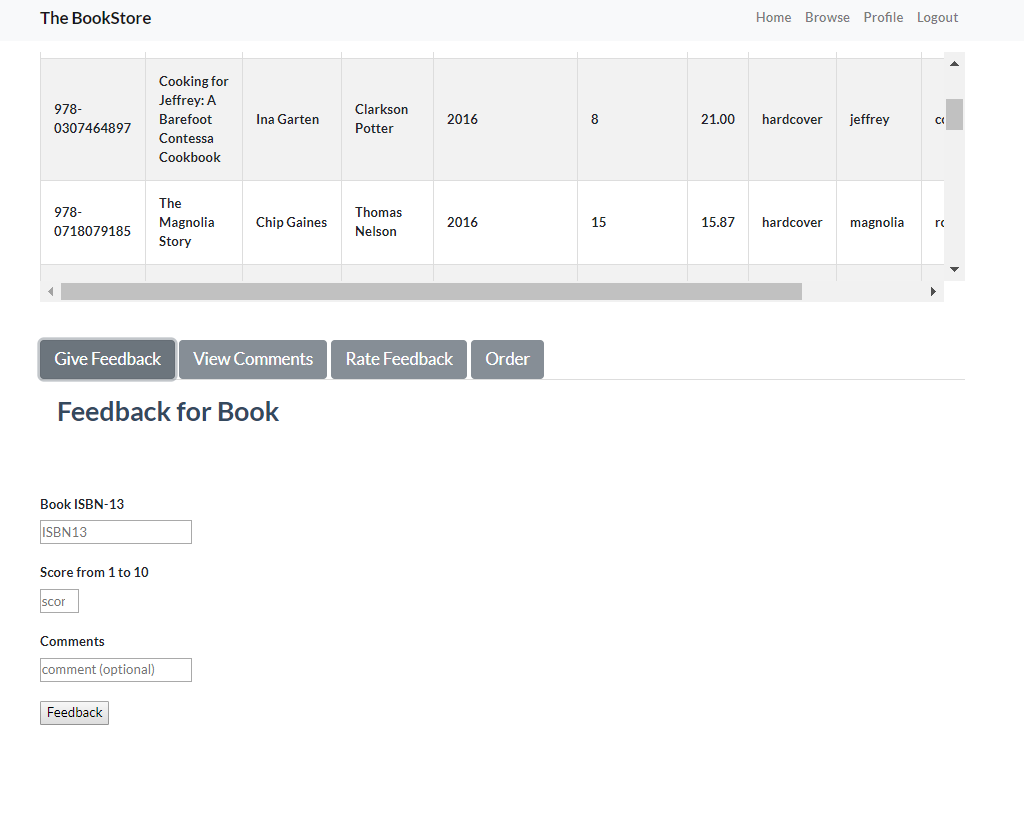


Figure 4: Users may comment/ rate feedback/View comments / order books