COMPSCI 235 Assignment2 Report

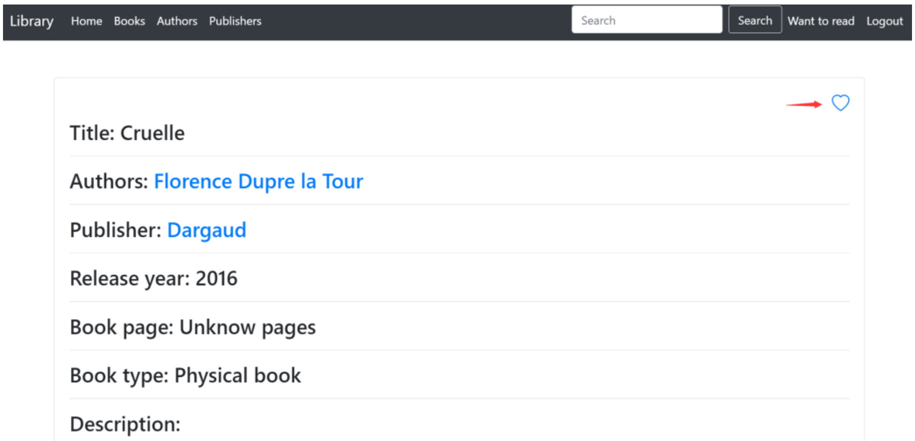
Our web application is a user-friendly, safe, powerful and self-defensive library system. It supports many essential functions that a library should have and some additional cool features that make the system unique. This report will introduce the new cool features and present our key design decisions throughout the development process.

Cool features:

The want-to-read List:

Our book system may have lots of books available to users. Users may interest in one book but don't have time to read it right now. Users can memorize the name or write the name somewhere else. However, it is very annoying to memorize an interesting book name and forget it when ready to read. On the other side, write the book name somewhere else seems like a good idea. However, this will greatly reduce the user experience, and we don't want this to happen as we aim to provide the best experience to users.

A want-to-read list is a place where user can put their favourite books inside. When logged in users are viewing one book's details, they can see a hollow heart on the top-right corner:



Title: Cruelle

Authors: Florence Dupre la Tour

Publisher: Dargaud

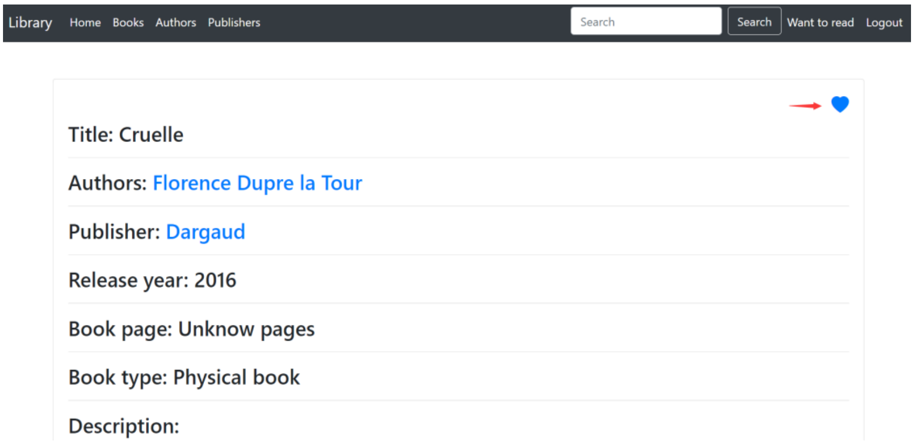
Release year: 2016

Book page: Unknow pages

Book type: Physical book

Description:

If users are interested in this book, they can click on the heart and add this book to the want-to-read list. The heart will then be filled to indicating the user that the action is complete like this:



Title: Cruelle

Authors: Florence Dupre la Tour

Publisher: Dargaud

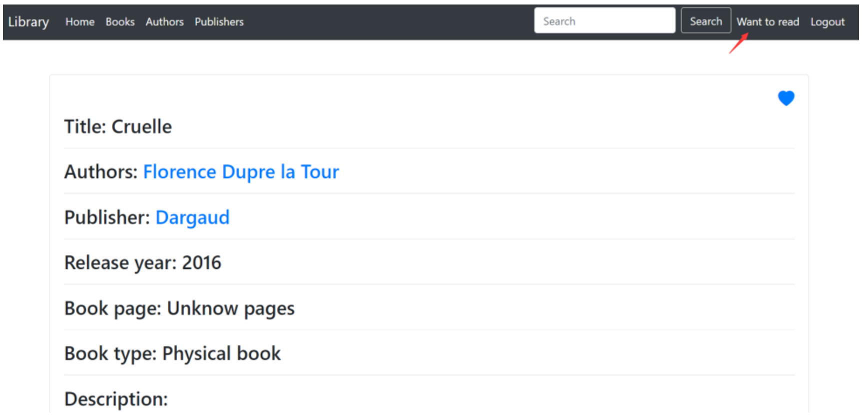
Release year: 2016

Book page: Unknow pages

Book type: Physical book

Description:

Users can add whatever number of books into the want-to-read list. To view all books in the list, just simply click on the “Want to read” button on the top-right corner:



Title: Cruelle

Authors: Florence Dupre la Tour

Publisher: Dargaud

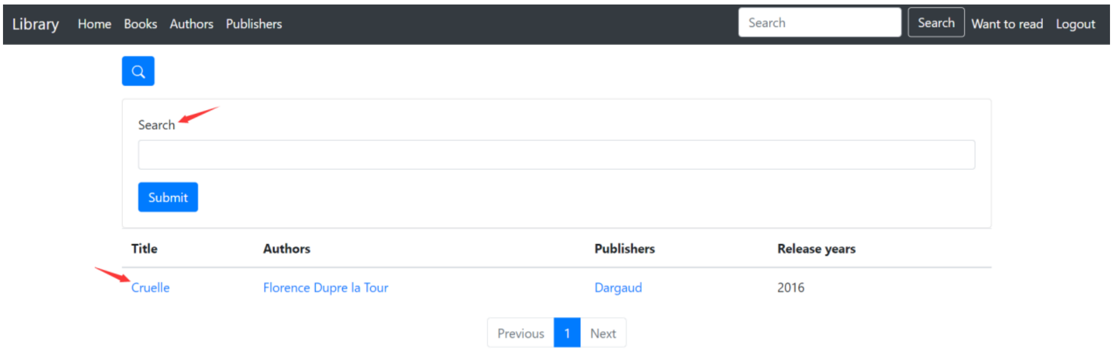
Release year: 2016

Book page: Unknow pages

Book type: Physical book

Description:

Users can view favourite books as usual in “Want to read”, search is also available for users to only search books in the want-to-read list.

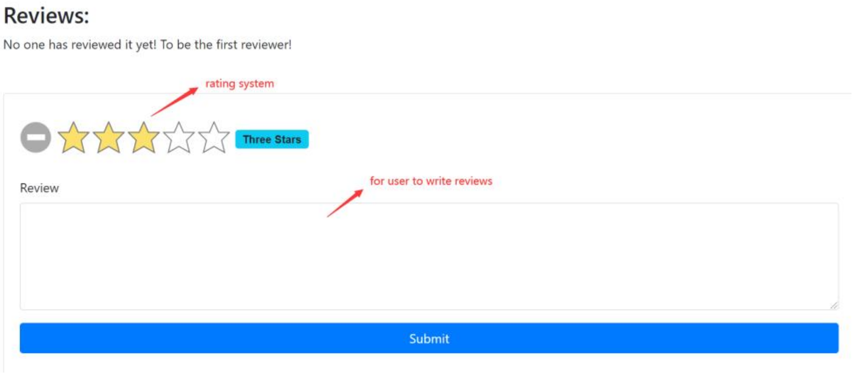


Title	Authors	Publishers	Release years
Cruelle	Florence Dupre la Tour	Dargaud	2016

Note that only logged-in users can add books to the want-to-read list. If users don’t log in, click the heart will redirect them to the login page.

Reviews and Rating System:

Logged-in users can post a review for an article. A box will be loaded for users to write comments at the end of each book. The user can also rate this book from 1 star to 5 stars.



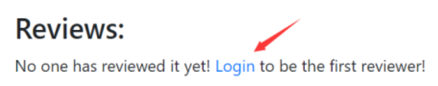
Reviews:

No one has reviewed it yet! To be the first reviewer!

rating system

for user to write reviews

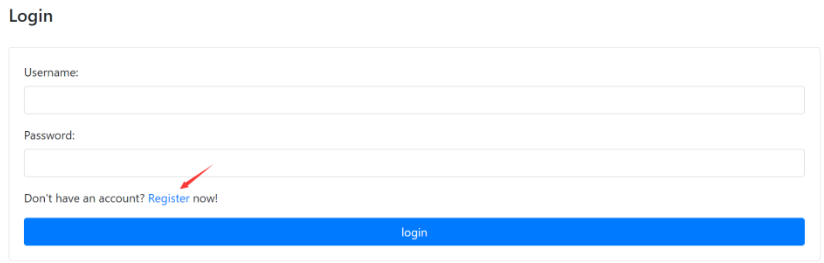
However, if the user didn’t log in, the box will not be loaded; instead, we will prompt the user to log in:



Reviews:

No one has reviewed it yet! Login to be the first reviewer!

If the user doesn’t have an account, after clicking login, they can click on “register”:



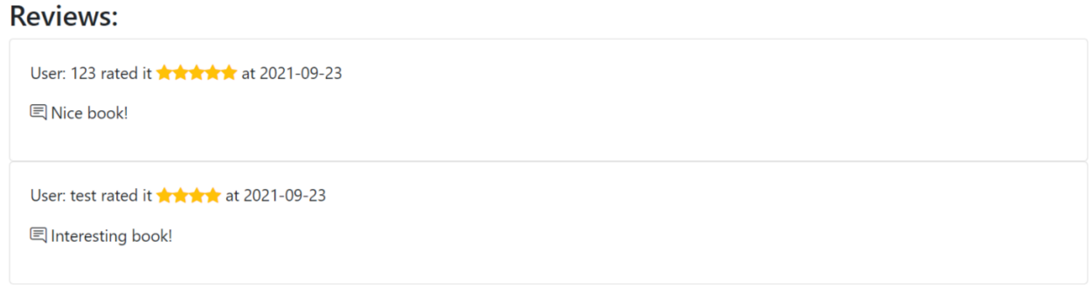
Login

Username:

Password:

Don't have an account? Register now!

Users can see all reviews at the bottom of each book; logged-in and not logged-in users can see reviews for all books.



Reviews:

User: 123 rated it ★★★★★ at 2021-09-23

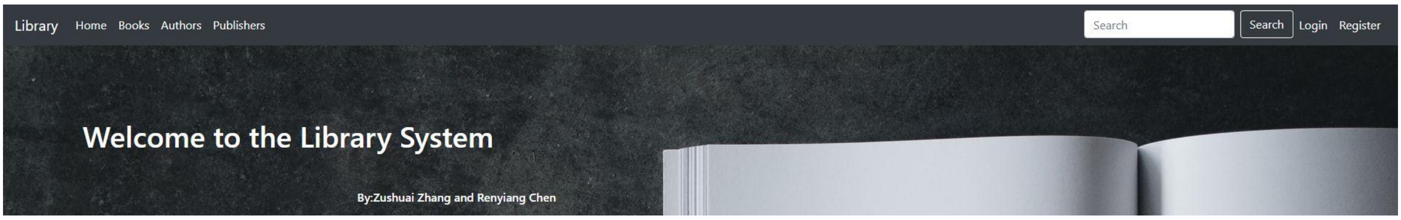
Nice book!

User: test rated it ★★★★ at 2021-09-23

Interesting book!

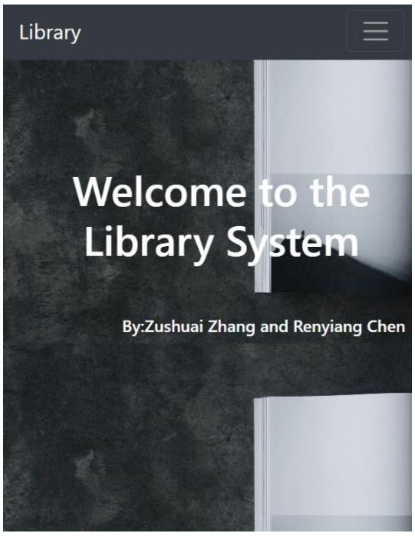
Mobile Terminal Adaption:

People are using various devices nowadays. Most people are likely to access our library system via phones or tablet PCs with a smaller screen than computers. If we do not adapt mobile terminals, users may need to enlarge the screen to use some functionalities such as search. We did such adaption by responsive design through an external front-end framework called bootstrap, and all information are now displayed more compact for small screen users:



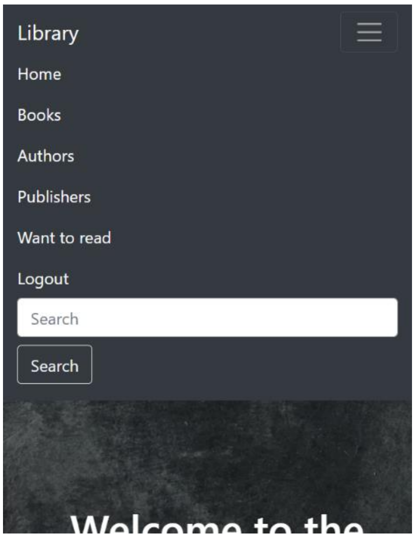
Welcome to the Library System

By:Zushuai Zhang and Renyiang Chen



Welcome to the Library System

By:Zushuai Zhang and Renyiang Chen



Design decision:

Architectural pattern:

Using the architectural design pattern discussed during the lecture, we put specific files in specific folders, and each folder corresponds to one part of the architectural pattern. This makes the relationship between each functionality clear and makes web application development easier.

The Single Responsibility principle:

Each class/module, such as a specific blueprint (books_blueprint, authors_blueprint, etc.), is designated to be responsible for only one part of a single functionality; this makes our code easy to maintain, flexible and expansible since any changes to one part will not affect other functionalities.

Repository pattern

We used a repository pattern to make our service layer focus on business logic rather than dealing with fetching data.

Dependency Inversion principle

As stated by the Dependency Inversion Principle, we should not implement dependency between high-level modules and low-level modules, so an abstract repository interface is implemented between the service layer and the memory repository so that the service layer does not depend on memory repository; service layer and memory repository are both depend on the abstract repository. The benefit is that this allows us to change the type of repositories freely, such as changes to a database repository.

Principles of Agile Software

We follow this principle. A key point about this principle is that a working software is the most important thing, it measures the progress of development. To achieve this, we developed a homepage first, then gradually added more features and functionalities such as displaying books, publishers, etc. This makes our development fast, and we can test our application at a very early stage.