**Introduction**

A storytelling website built on Express backend, Pug template, bootstrap4 framework frontend and MongoDB database. The application is designed to provide a core set of features to meet the basic needs of users. The main functions include user identity verification, book publishing and evaluation, book genre query, etc.

**Background**

Public data sources used

APIs: User registration, user login, log out, search books by category, add books, delete books, edit books, comment on books, view my books

User input：  
 Login: username, password

Registration: username, nickname, password, confirm password, role

Comments: Ratings, evaluation content

Competing applications all have functions such as user registration, login, and book information viewing.

**Core Functions**

What makes the proposed app relevant/distinctive?

Provides a user-friendly library management and sharing platform. It may lie in the ease with which users can search for books, post reviews, rate books, and interact with other users. One of its possible features is to provide a user-friendly interface and interactive functions that allow users to easily manage books, browse other people's reviews, and share their reading experiences.

What business processes needed

User registration and login will be an important part of the business process, and users will need to be able to register for an account and log in to use the functionality of the application. In addition, book search, book publishing, book evaluation, etc. are also important business processes.

Consider integration with existing services

Later, third-party APIs can be accessed, such as social media sharing and identity authentication services. These services can enhance user experience and provide additional functionality.

What profile information is required? Where will it be stored?

Users may be required to provide basic personal information (such as username, password, user nickname, etc.). This information is stored in mongodb database.

Development time and cost

Development took 7 days.

Will the proposed app meet the brief?

It can realize user registration, search according to different genre of books, publish books, evaluate books, etc.

**Advanced Functions**

Features planned to be added in the future:

Personalized recommendations: Recommend books to users based on their interests and reviews.

Multi-language support: Provides multi-language interface to meet the needs of international users.。

**Data Protection**

Personal data visible to users:

Administrators can view the books they have published, edit and delete them

Data Security:

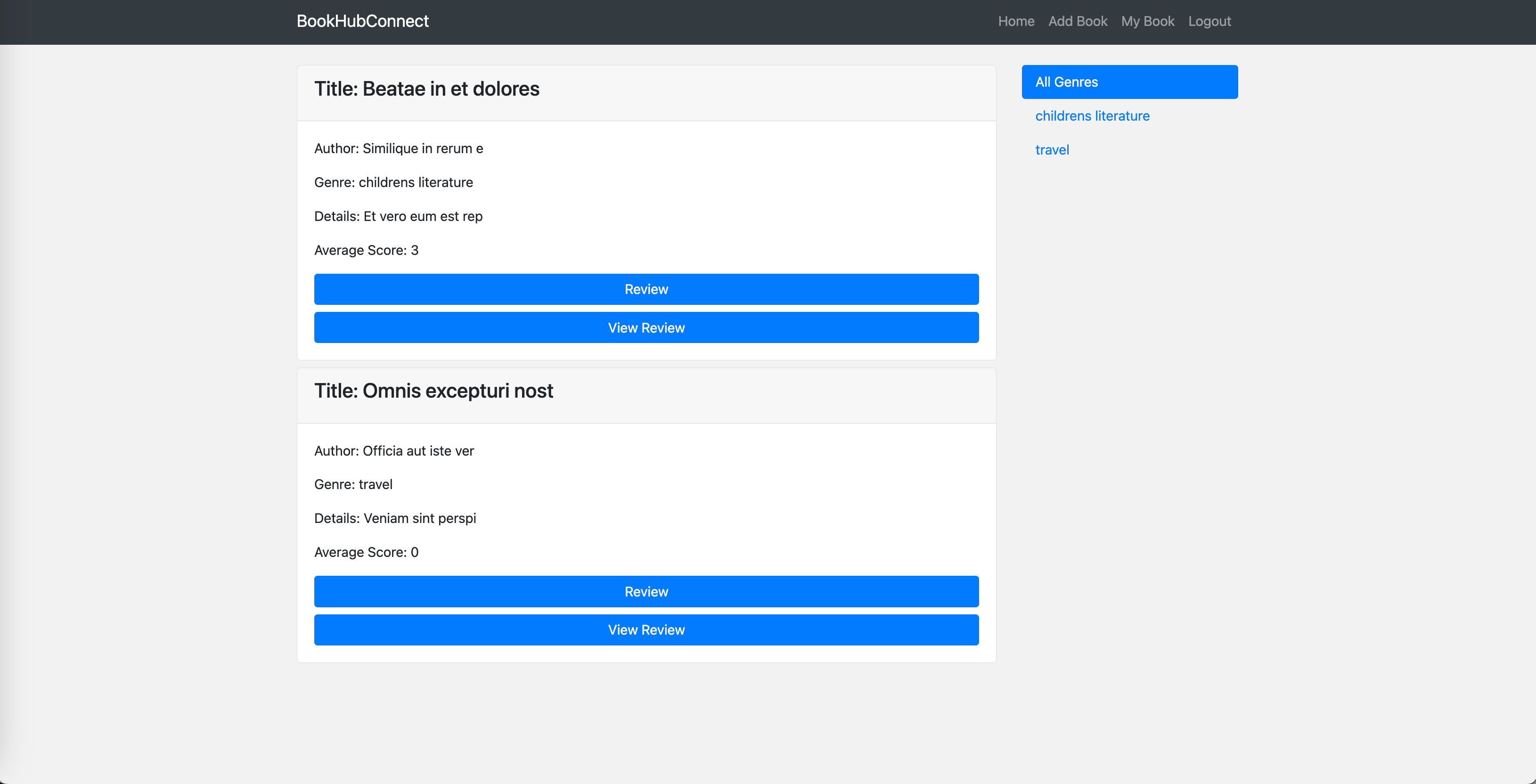
User registration data, password md5 encryption

**Implementation**

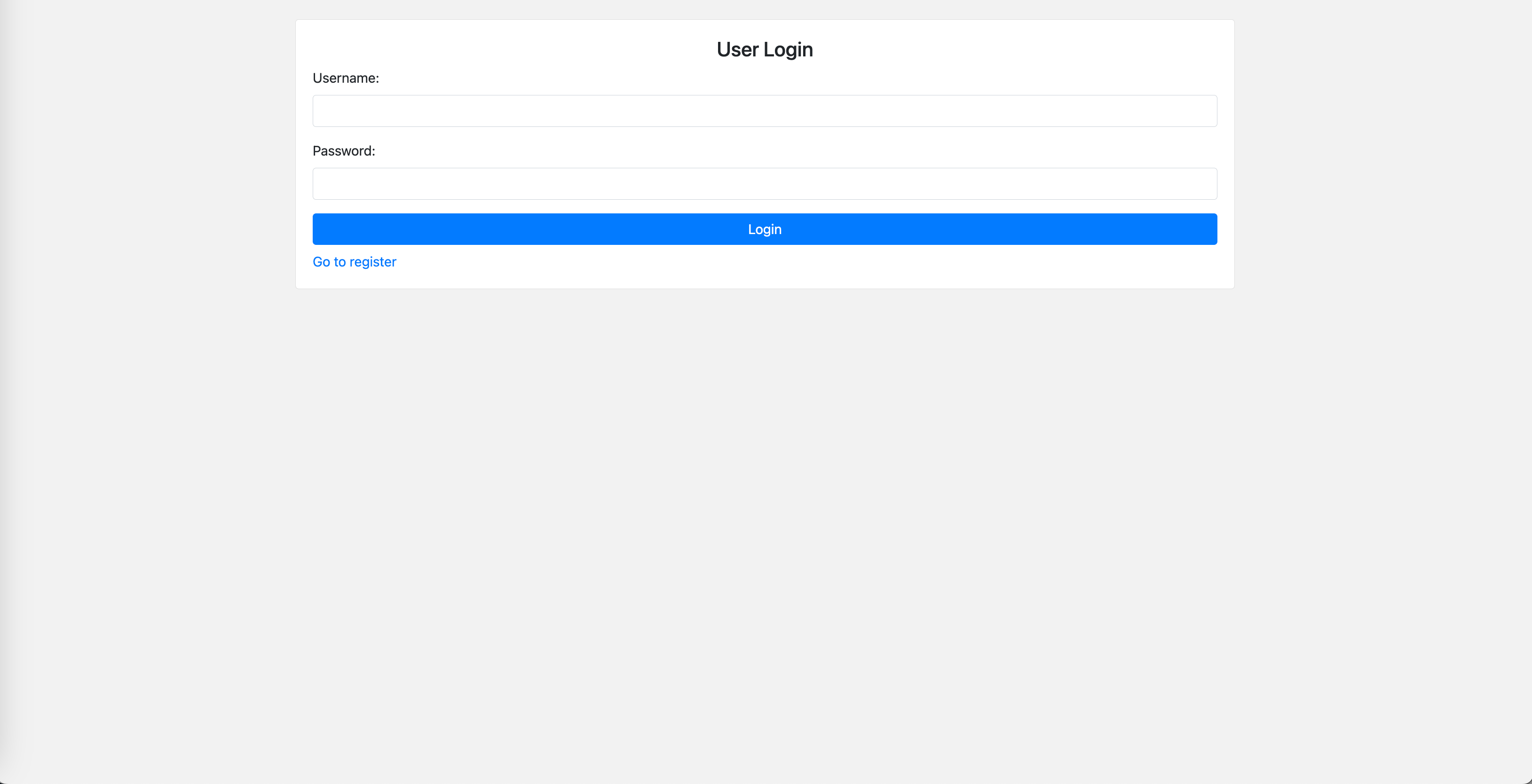
Website layout description:

The top is the navigation bar, the left is the website name BookHubConnect, and the right is the menu bar. The menu bar displays Home, Login, and Register options when not logged in. After logging in, the user displays the Home and Logout options, and the administrator displays the Add Book, My Book, Logout, the left side of the home page is the book list, and the right side is the category list. There are view comment buttons and add comment buttons in the book items. The add comment button will only be displayed after logging in.

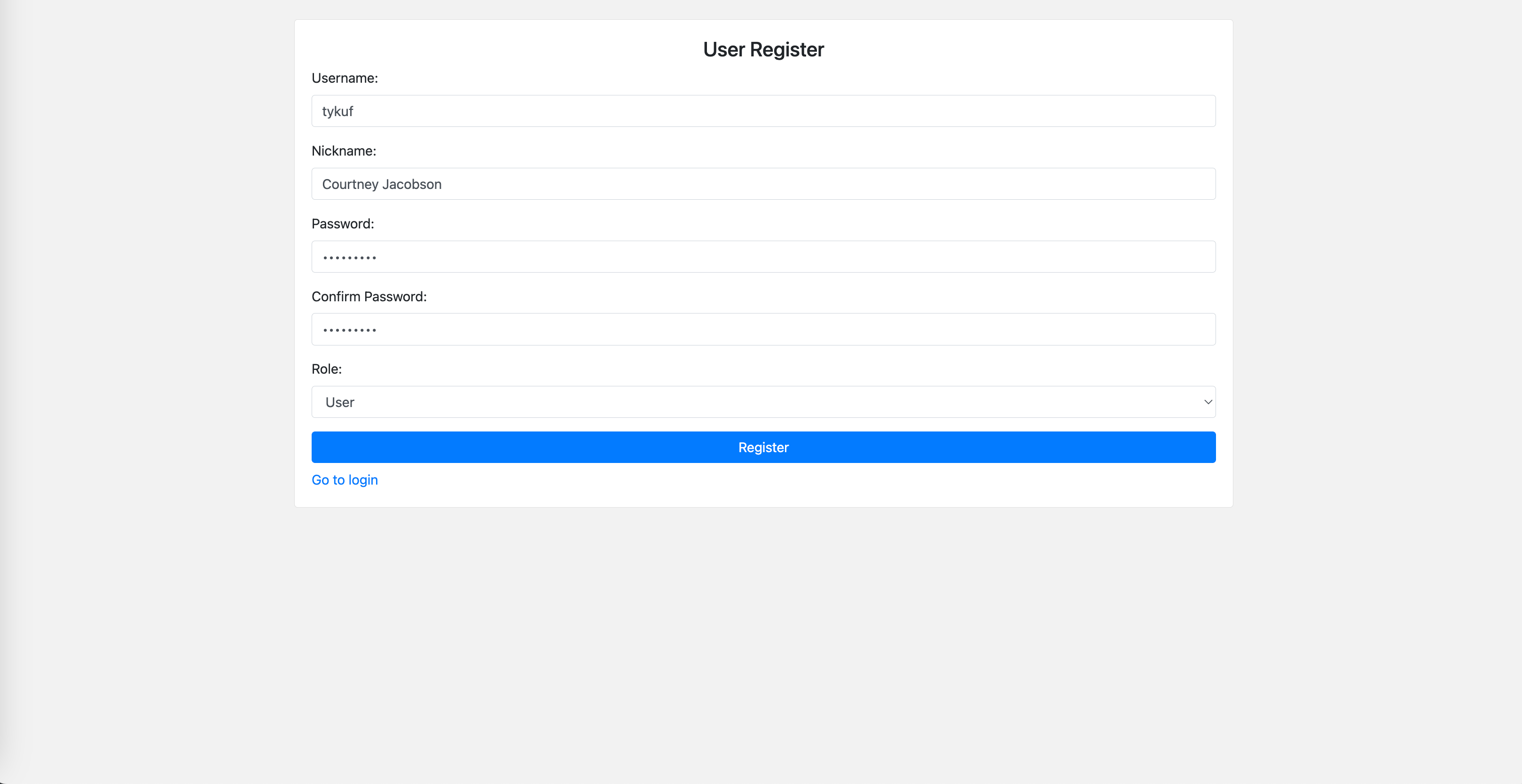
Home page renderings:



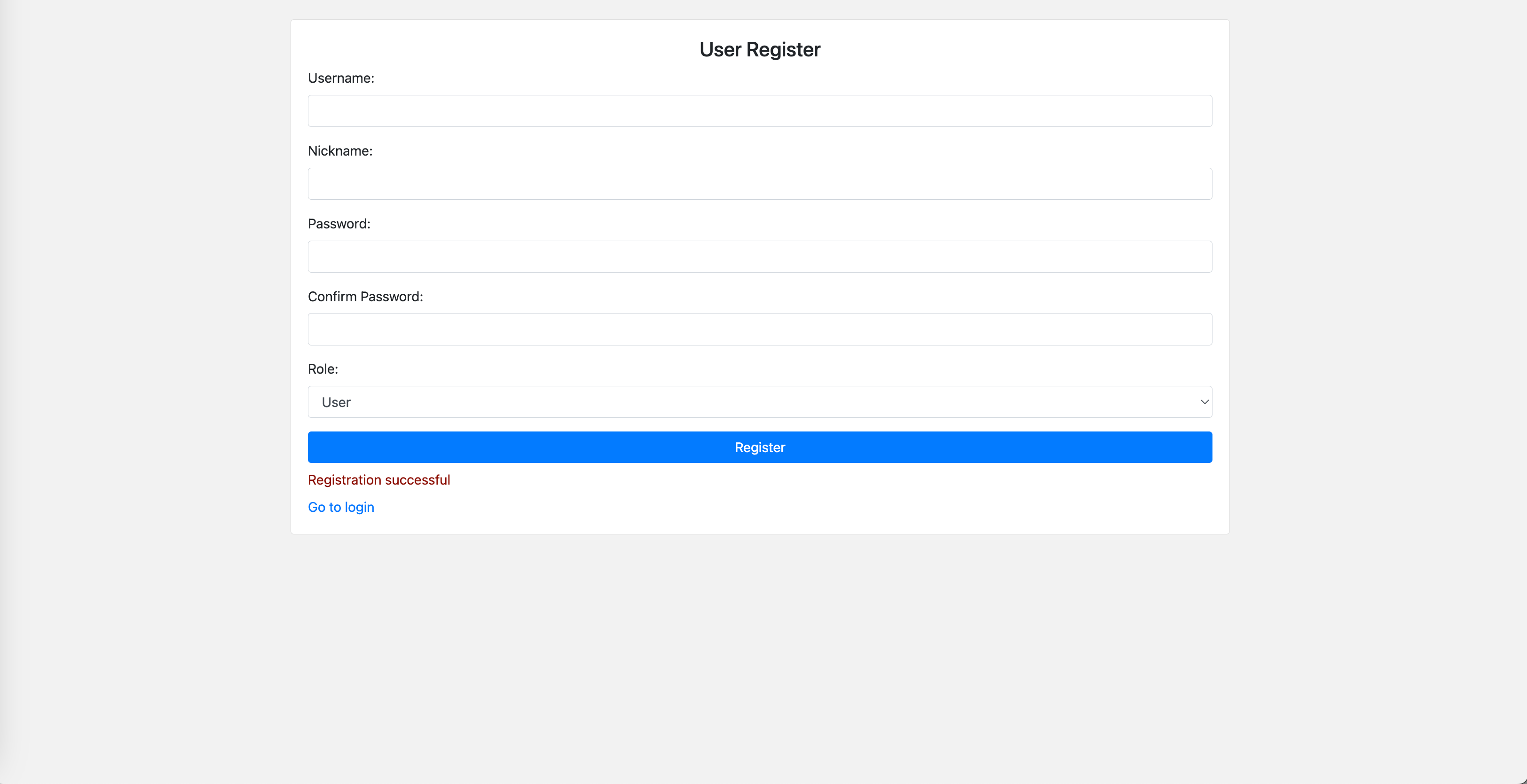
Login renderings:



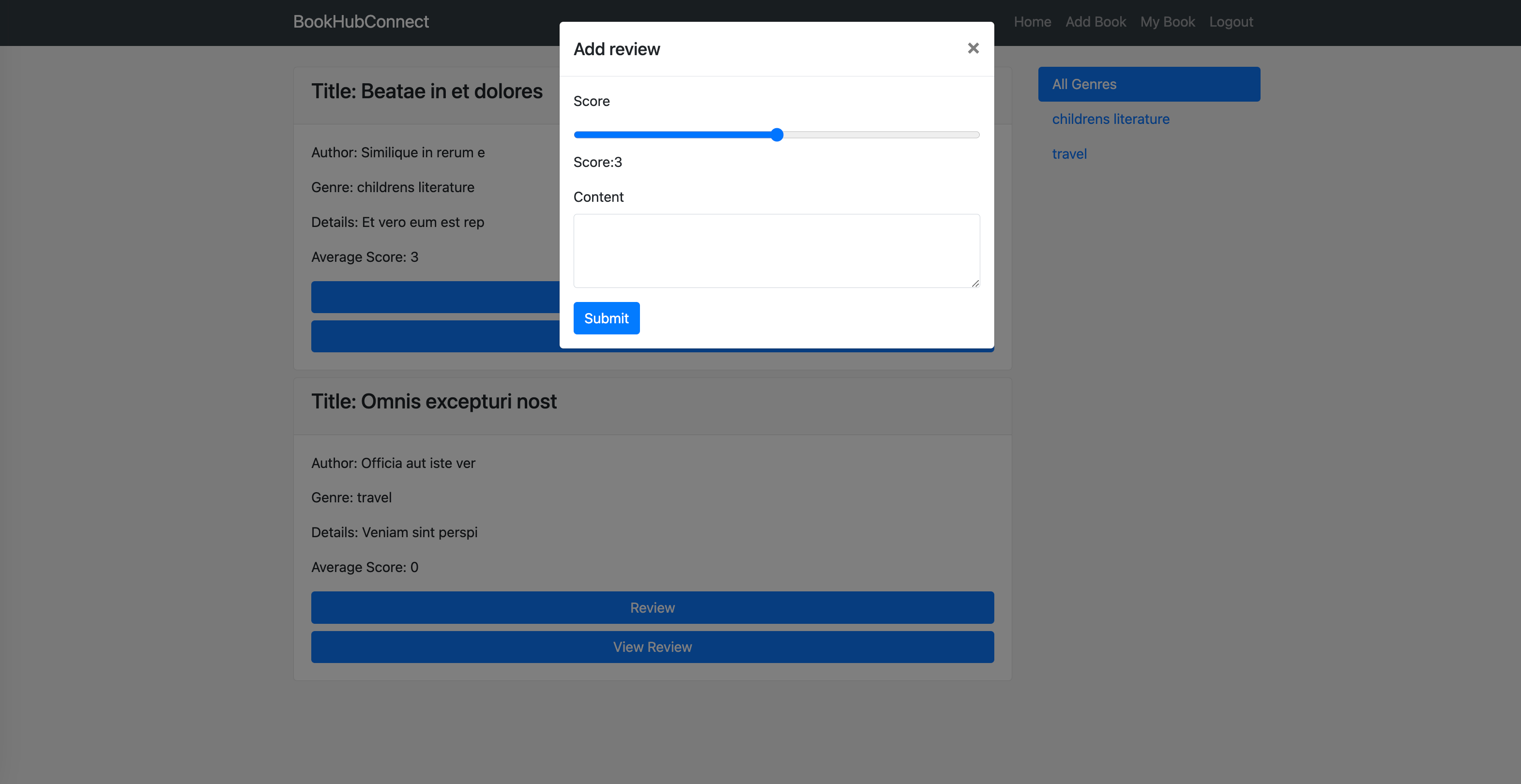
User registration:



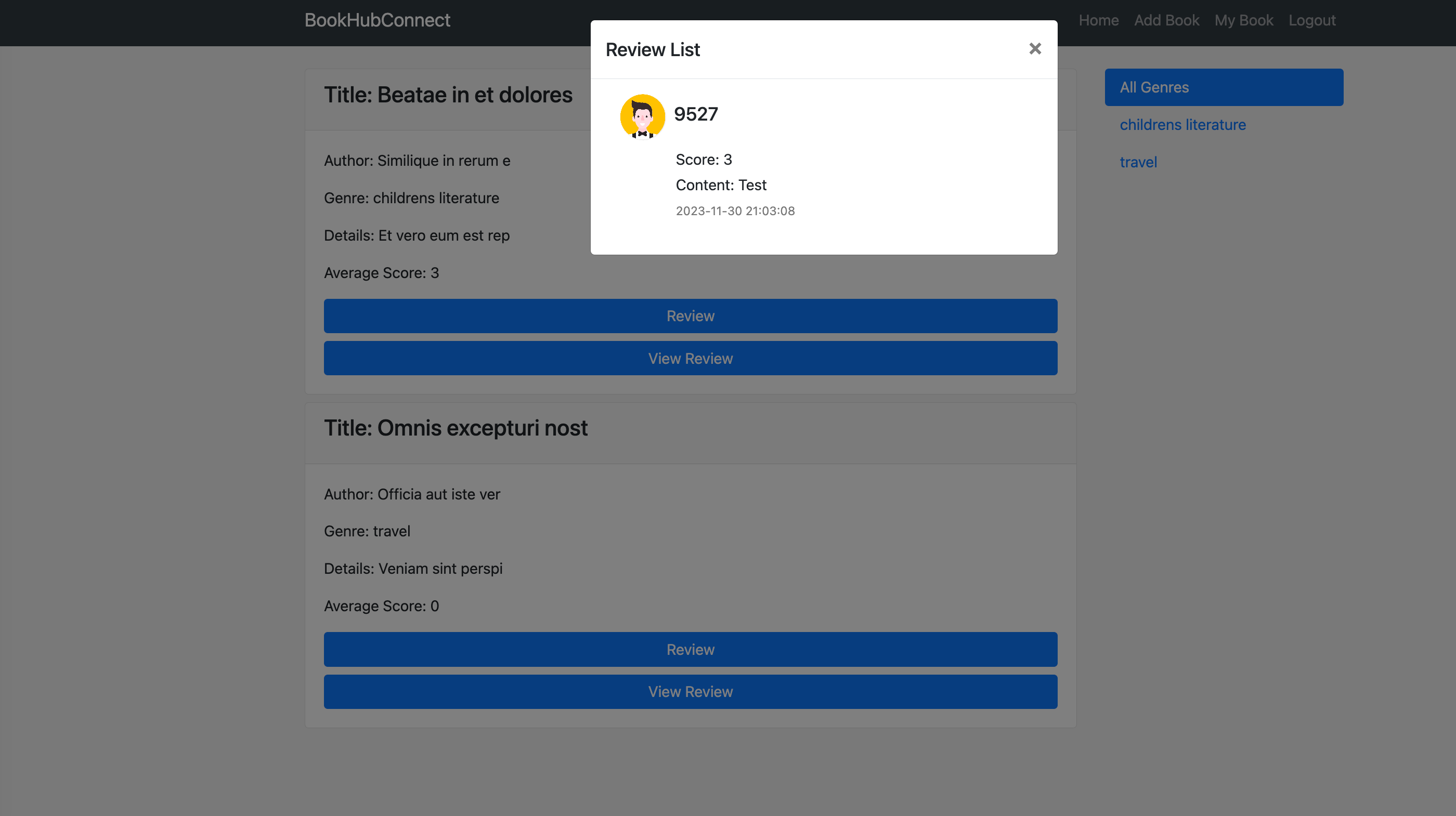
Registration success:



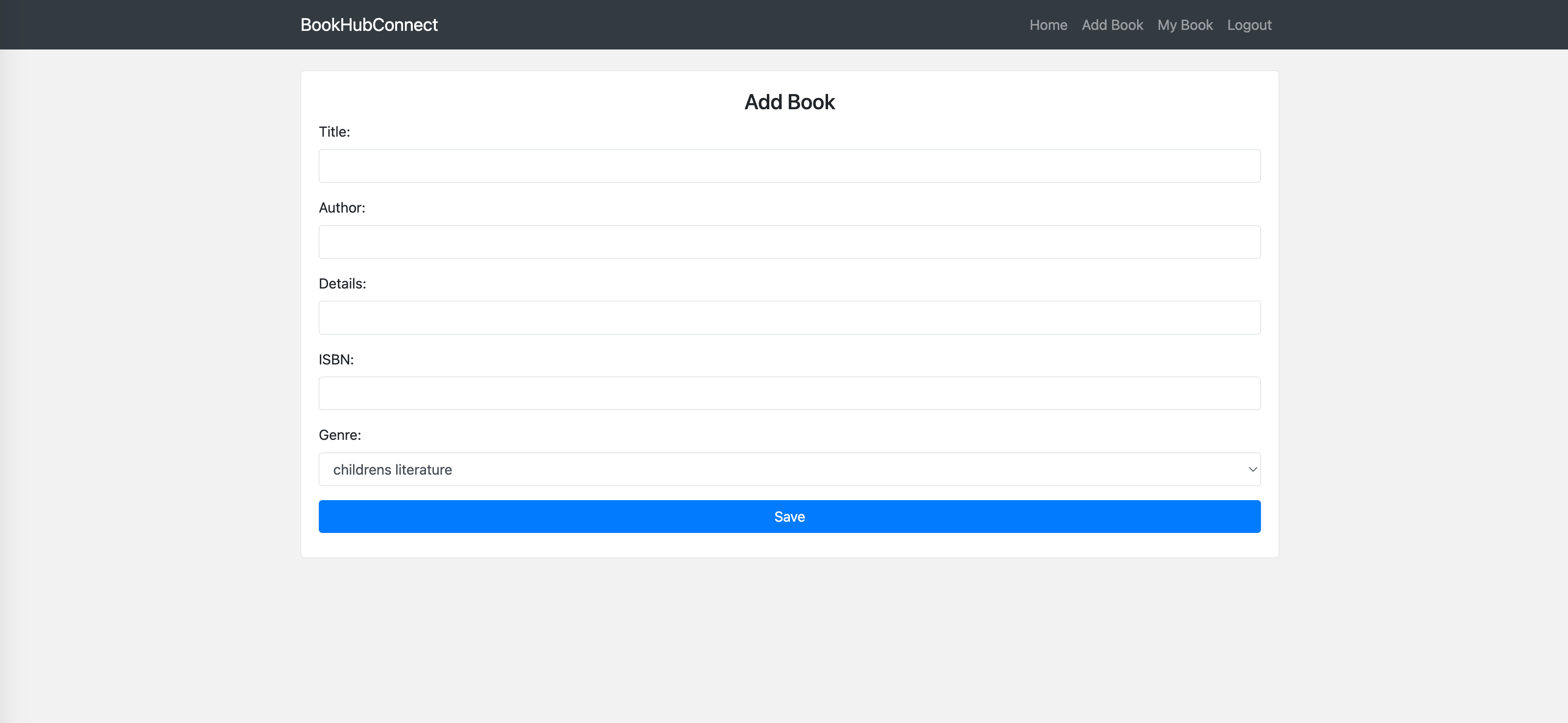
Add reviews:



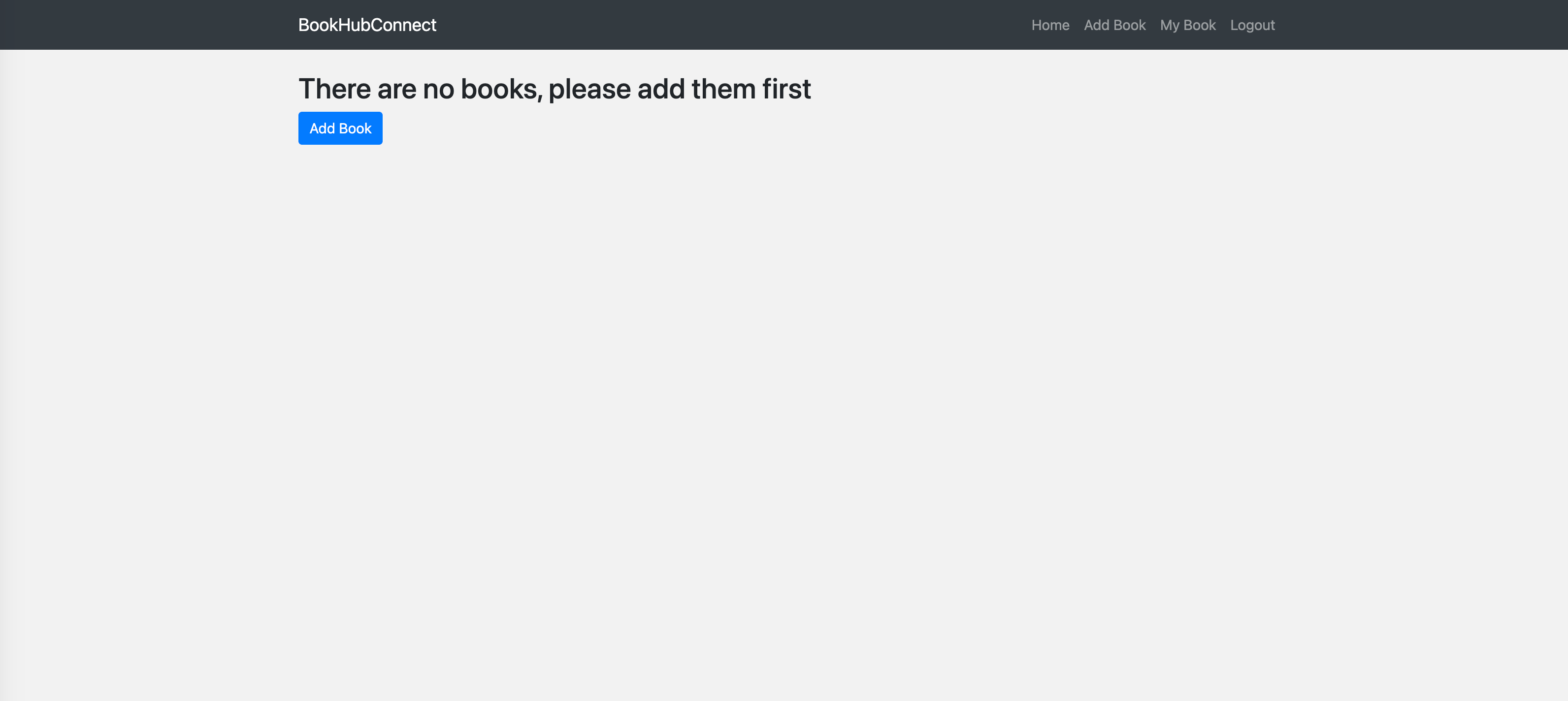
View reviews:

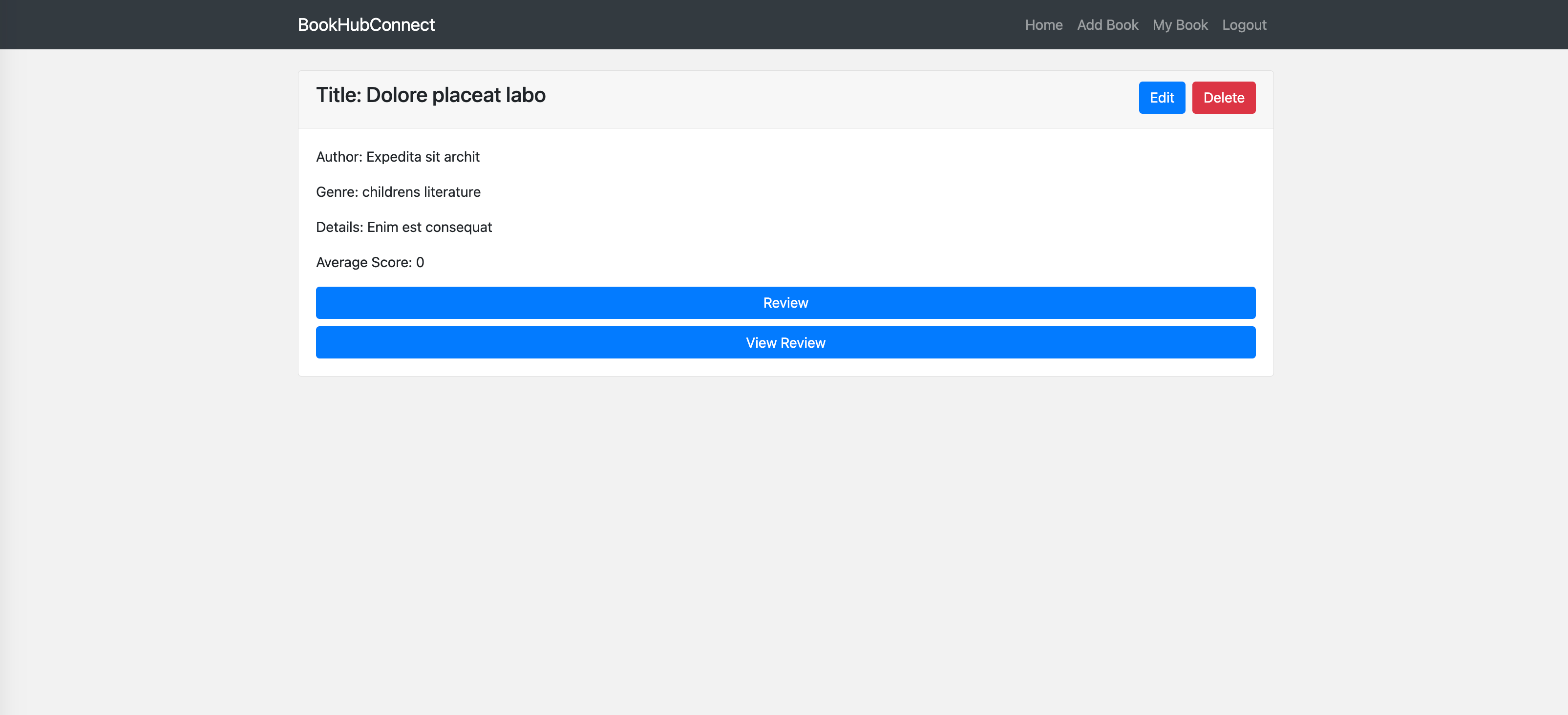


Add books:



My books:





**Technology Stack**

Frontend：

Pug Template Engine: A template engine for generating HTML pages on the server side, simplifying the HTML creation process.

Bootstrap 4: A front-end UI framework that provides responsive design, components, and styles for creating modern user interfaces.

Backend：

Express.js: A lightweight Node.js web application framework for building reliable and high-performance backend services.

MongoDB: A non-relational database used to store and manage application data.

Other tools and techniques:

Mongoose: A library for object modeling with MongoDB databases in a Node.js environment.

**Data organisation**

Collections:

users: Contains the user's personal information.

books: stores detailed information about books.

reviews: Record information such as user reviews and ratings of books.

Genres: stores category information of books, name.

Structure:

users: username、password、nickname、role。

books: name、author、details、isbn、user、genre。

reviews: book、user、score、content、createTime。

genres: name

Relationships between collections:

There is a one-to-many relationship between users and reviews. One user can rate multiple books.

There is a one-to-many relationship between books and reviews, and a book can have multiple reviews.

There is a many-to-one relationship between books and genres. A category can have multiple books.

API design：

GET /user/login View login page

POST /user/login login operation

GET /user/register View the registration page

POST /user/register registration operation

GET /user/logout Log out

POST /review/add comment operation

GET /review/list View review list

GET /book/list book list

GET /book/myself View my published books

GET /book/add View the add book page

POST /book/add add book operation

GET /book/remove/:id delete book operation

GET /book/update/:id Edit book page

POST /book/update/:id Edit book operation

Query and UI forms:

Book classification inquiries can be made through the classification menu bar on the right

Simulation servers supported during the design process:

No mock data is used, and the Fake Filler browser plug-in is used to quickly fill the form during testing.

**Services**

Single server: Use Apache, Nginx, IIS single server. This method is relatively low-cost, but it has greater risks. If there is a failure, the entire service may stop running.

Multi-server: Improve application stability and reliability by setting up multiple servers. Traffic can be distributed through load balancing, but maintenance costs are high.

Advantages of cloud services:

Scalability and elasticity: Cloud services allow resources to be quickly expanded and contracted based on demand.

High availability: Cloud providers typically offer infrastructure with high availability, reducing the risk of application outages.

Cloud service comparison:

AWS (Amazon Web Services): Provides a wide range of cloud services, including computing, storage, databases, etc. Pay as you go and choose a variety of services based on your needs.

Azure (Microsoft Azure): Microsoft's cloud service platform provides services similar to AWS. Integrates well with other Microsoft products.

Google Cloud Platform (GCP): A cloud service platform provided by Google with high scalability and performance.

**Scalability**

Possible problems:

Number of users: As the number of users increases, the system needs to handle more requests. If the system architecture is not horizontally scalable, performance degradation and response time delays may result.

Data volume: As data volume increases, database queries and processing can become more complex and time-consuming. If the database design is not reasonable or there are no suitable indexes, it may lead to performance bottlenecks.

solution:

Horizontal expansion: By adding more server instances to share the load, load balancing technology can be used to distribute traffic to different servers.

**Tracking and statistics**

Can track website performance data, such as page loading time, server response time, etc., to optimize performance

Cannot track personal privacy information that users access on the website, etc.

**Conclusions**

The bootstrap framework and pug template engine are used to greatly increase development efficiency. At the same time, bootstrap's own style makes the web page look more beautiful.