TRƯỜNG ĐẠI HỌC CẦN THƠ

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**🖎🕮✍**



**BỘ MÔN TIN HỌC ỨNG DỤNG**

**BÁO CÁO ĐỒ ÁN HỌC PHẦN JAVA NÂNG CAO TN208**

**TÌM HIỂU CÔNG NGHỆ JAVA WEB VỚI SPRING BOOT VÀ ANGULAR**

**Giảng viên hướng dẫn: Sinh viên thực hiện:**

**Ths. Vũ Duy Linh Phạm Hoàng Duy – B1706568**

**CẦN THƠ, 6/2021**

1. **Giới thiệu đề tài**

Vận dụng kiến thức tìm hiểu được để xây dựng ứng dụng quản lý sách cơ bản trên nên tảng web.

Mục đích: mở rộng kiến thức, nâng cao kỹ năng lập trình đồng thời tìm hiểu các công nghệ web phổ biến hiện nay.

* Phạm vi đề tài:

|  |  |
| --- | --- |
| Phạm vi kỹ thuật | |
| Ngôn ngữ, framework | Java 11, Spring Boot 2.4, TypeScript 4.1, Angular 11, HTML5, CSS3 |
| Mô hình lập trình | Web API Models & Controllers |
| DBMS | MySQL Server 8.0, MySQL WorkBench 8.0, MySQL CMD |
| Công cụ soạn thảo và biên dịch | Eclipse 2021, Visual Studio Code 1.57 |
| Trình kiểm thử API | Postman, Thunder Client (VS Code extension) |

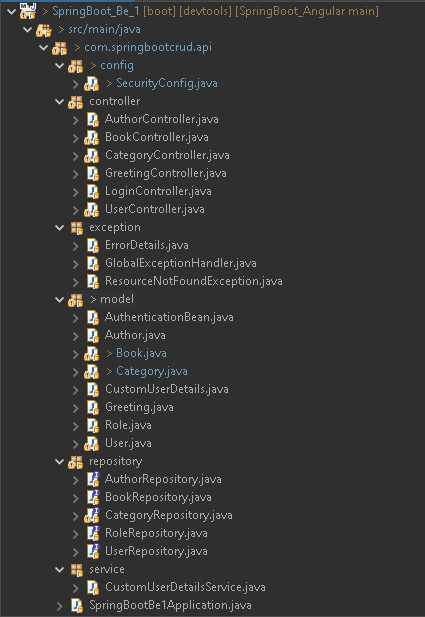
* Phạm vi người dùng: người dùng trang quản trị - admin
* Thực hiện đề tài: Phạm Hoàng Duy, MSSV: B1706568

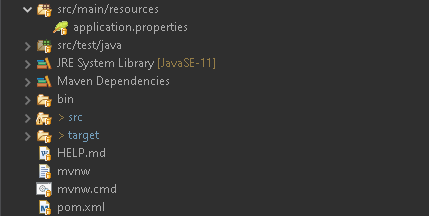
|  |  |
| --- | --- |
| Phone / Zalo | 0764417600 |
| Github | https://github.com/phamhoangduy207 |
| Project repository | https://github.com/phamhoangduy207/SpringBoot\_Angular |

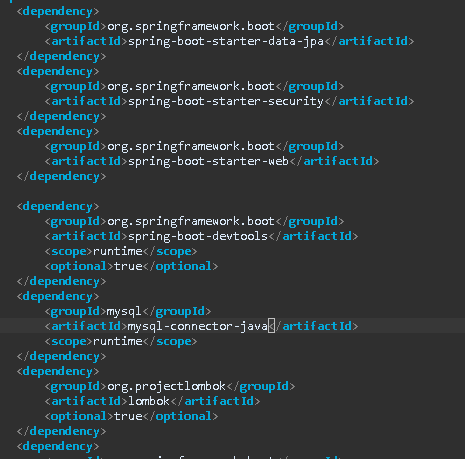
1. **Kết quả thực hiện**

**Các điểm quan trọng:**

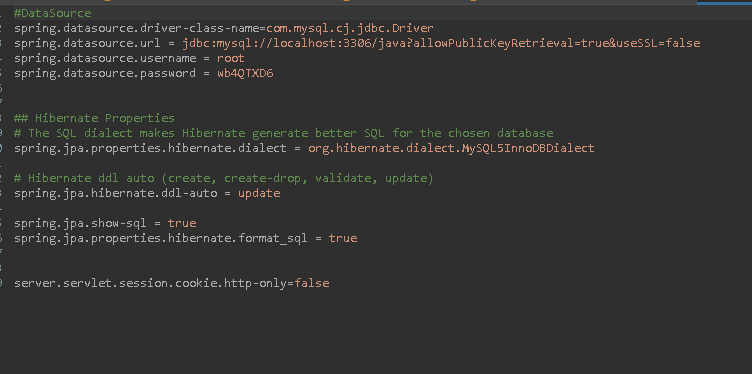
* Cầu trúc của backend project

****

****

****

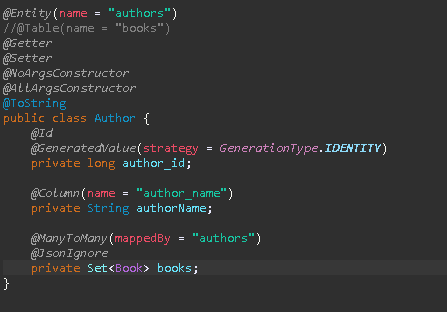
Hình Các plugin được sử dụng



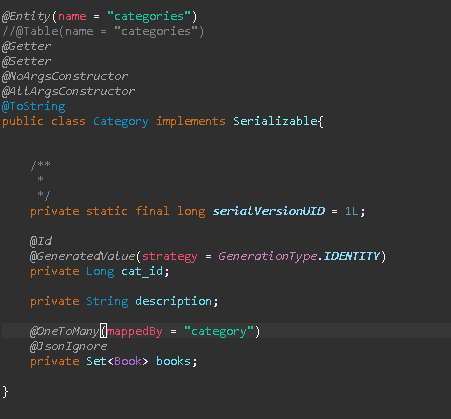
Hình Kết nối cơ sở dữ liệu MySQL



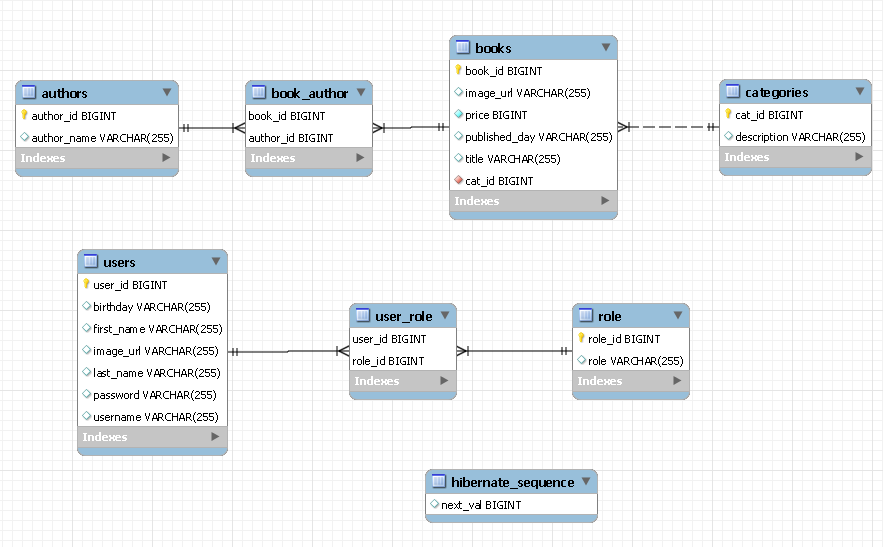
Hình Model Book



Hình Model Author



Hình Model Category

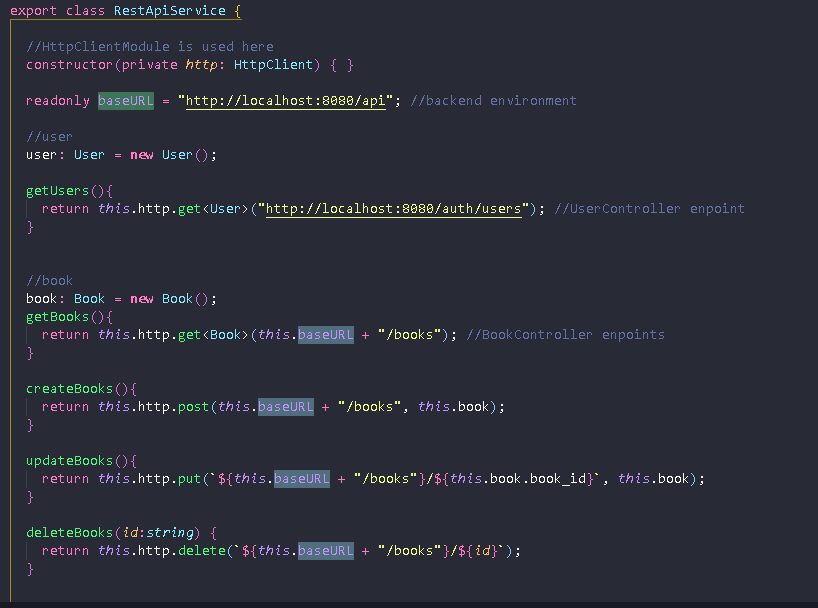


Hình Cơ sở dữ liệu nhận được sau khi khởi động chương trình

Các model Author và Category liên kết với Book lần lượt theo quan hệ ManyToMany và OneToMany, cấu hình Hibernate cho phép tự động cập nhật CSDL.

* Mối liên kết giữa Angular app và server:

Sử dụng HttpClientModule để gửi các request (get, post, put, delete) tới server

****

Hình Lớp gọi các API enpoints

Sử dụng HttpInterceptor để tự động truyền JSESSIONID vào mỗi request được thực hiện

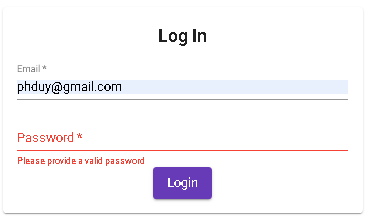
****

Hình Lớp hỗ trợ truyền header

1. **Chức năng xác thực**: cấu hình Spring Security yêu cầu xác thực với tất cả các truy cập vào đường dẫn như sau

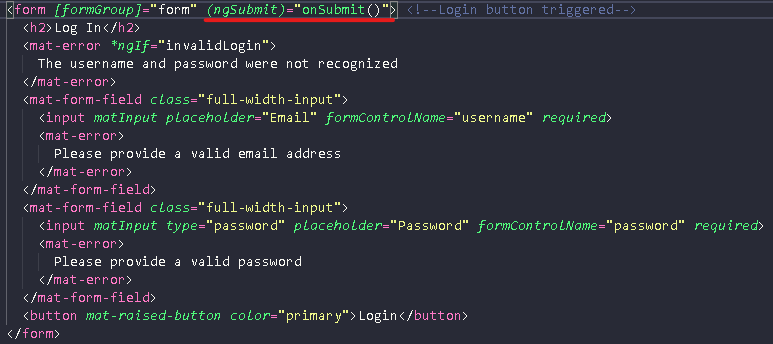
****

Hình Cấu hình HttpSecurity



Hình Giao diện đăng nhập

Khi kích hoạt nút Login gọi thực hiện hàm onSubmit, hàm này tiếp tục gọi phương thức authenticationService(username, password) được viết ở lớp service với các tham số đầu vào là thông tin đăng nhập trên giao diện.



Hình Giao diện đăng nhập - HTML

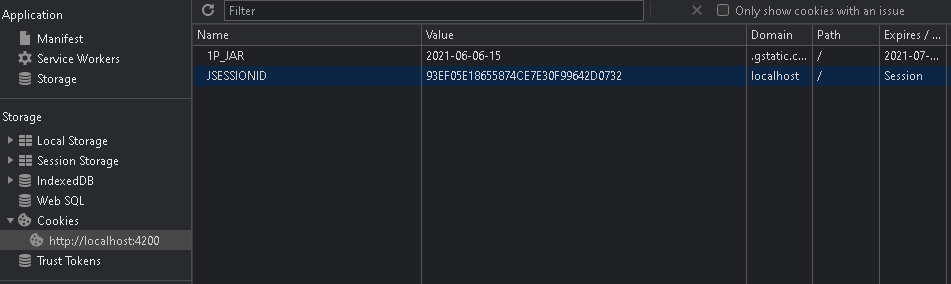


Hình Phương thức onSubmit()

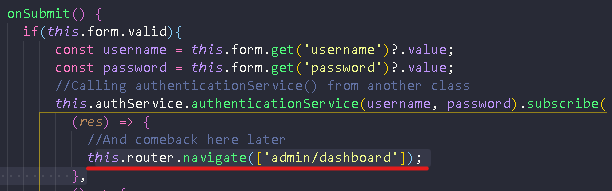


Hình Phương thức authenticationService(username, password)

authenticationService() gửi một Get request tới <http://localhost:8080/api/greeting> (tương ứng với GreetingController) cùng với header Authorization có giá trị là chuỗi mã hóa được tạo dựa trên username và password. Ghi chú: có thể send request tới bất cứ controller nào đang được bảo về bởi HttpSecurity, khi đủ điều kiện sẽ được trả về tự động 1 JSESSIONID.



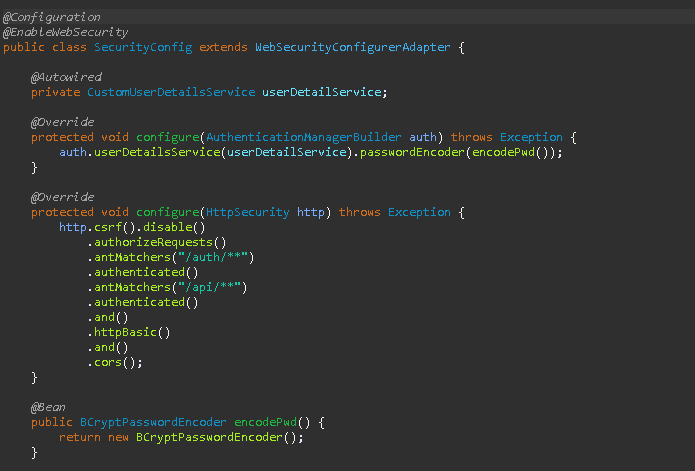
Hình JSESSIONID sau khi đăng nhập thành công được lưu vào cookies

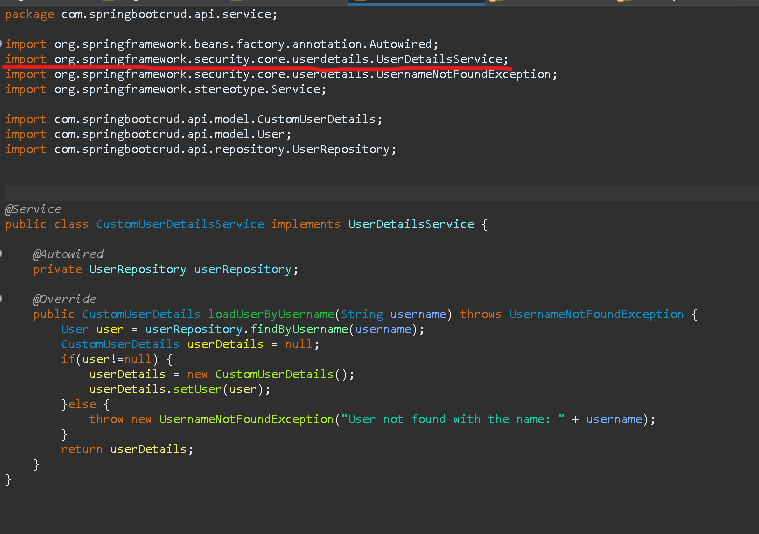


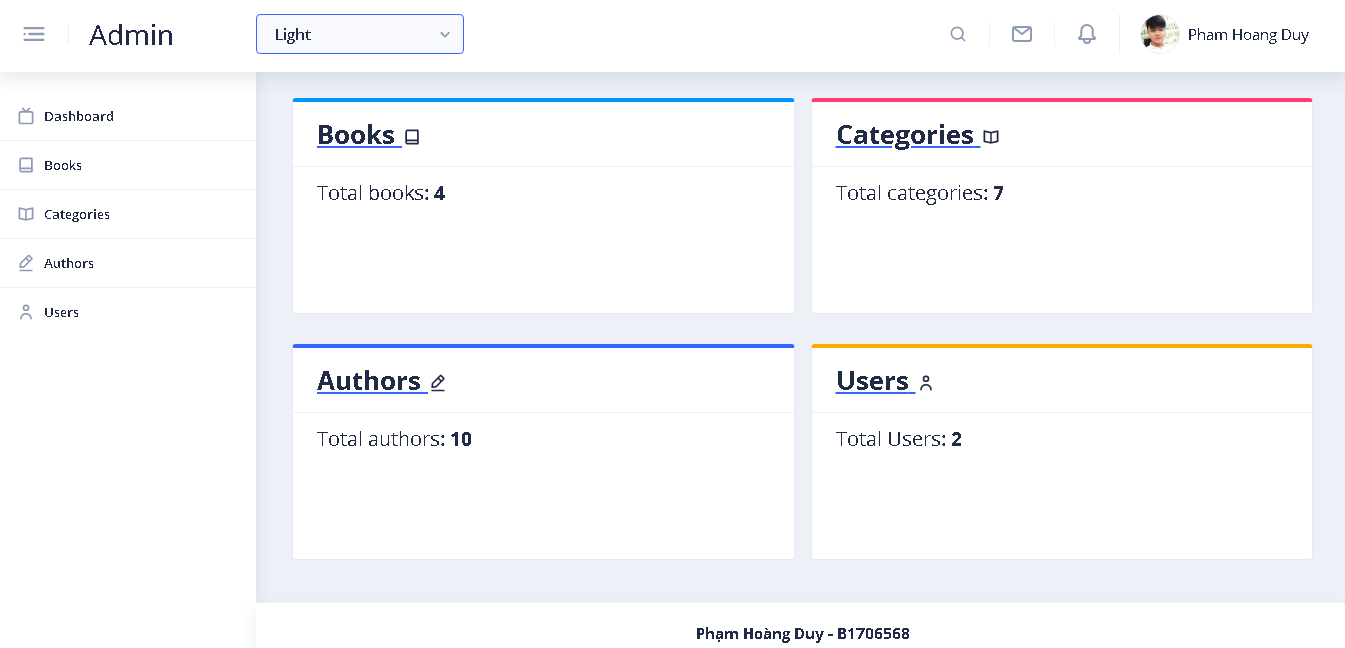
Hình Tiếp tục thực hiện phần còn lại của onSubmit()

Và tiếp tục điều hướng tới dashboard.

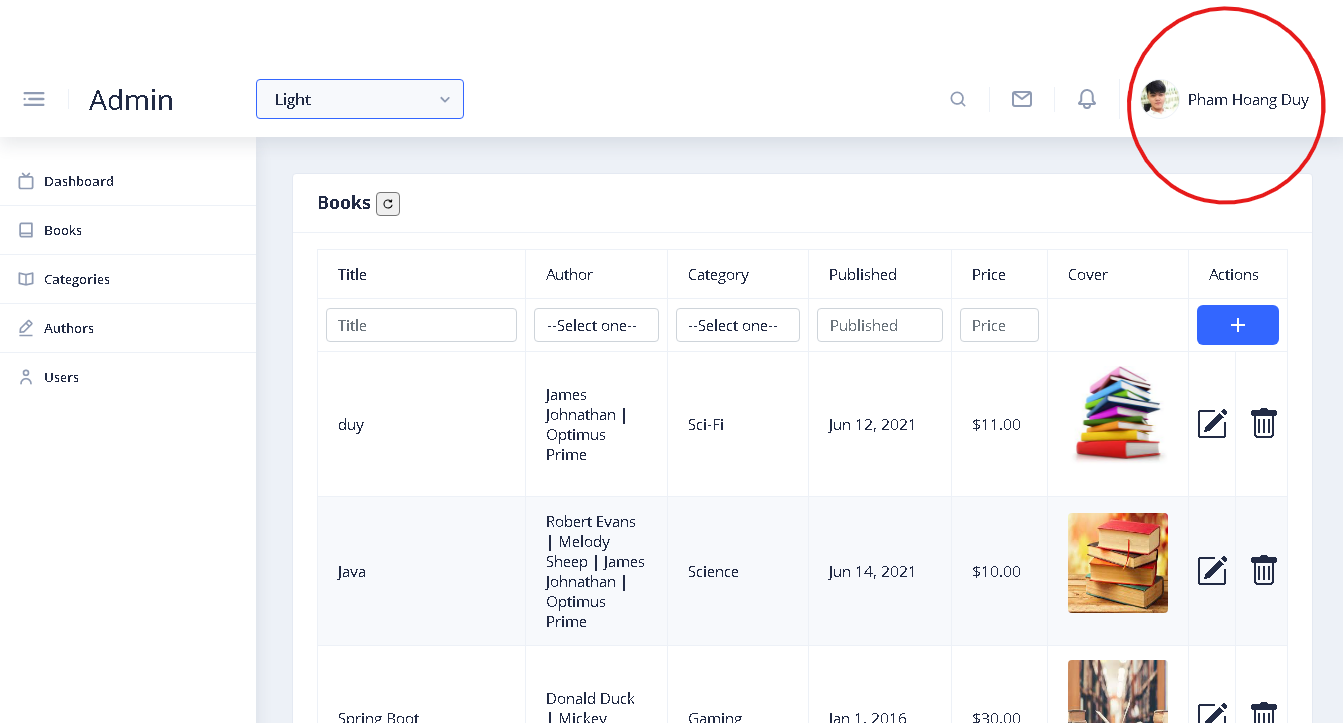
Điều này có nghĩa là kịch bản xác thực được xây dựng hầu hết ở phía frontend, vì JSESSIONID sẽ được trả về cho bất cứ request nào có token hợp lệ nên không cần phải viết thêm xử lý ở backend như 1 LoginController. Tuy nhiên, để so sánh thông tin đăng nhập với database cần sử dụng phương thức userDetailsService() của AuthenticationManagerBuilder với tham số đầu vào là 1 CustomUserDetailsService được xây dựng dựa trên interface UserDetailsService và model User.



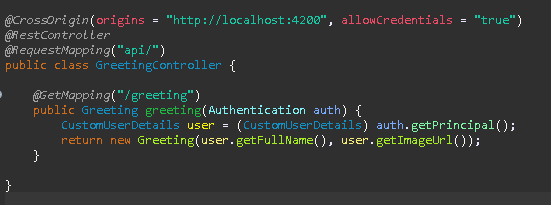




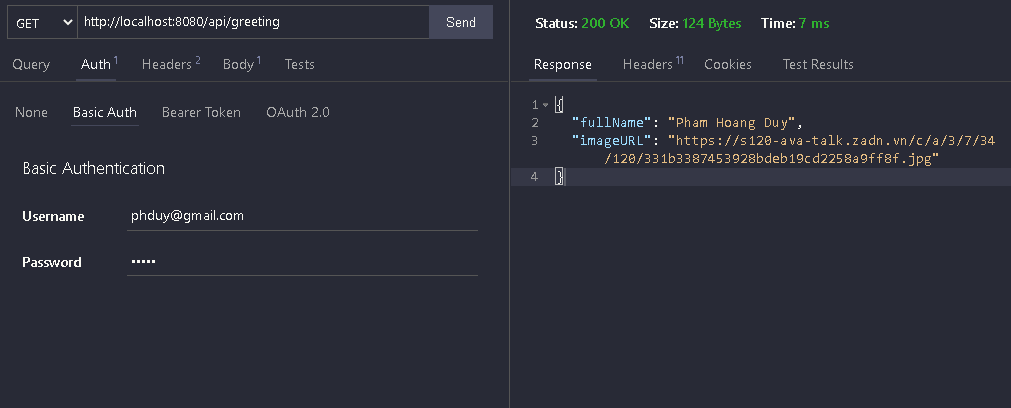
Hình Giao diện dashboard



Dữ liệu người dùng hiển thị ở góc phải là giá trị nhận về từ login request ở trên, bao gồm họ tên và image URL.



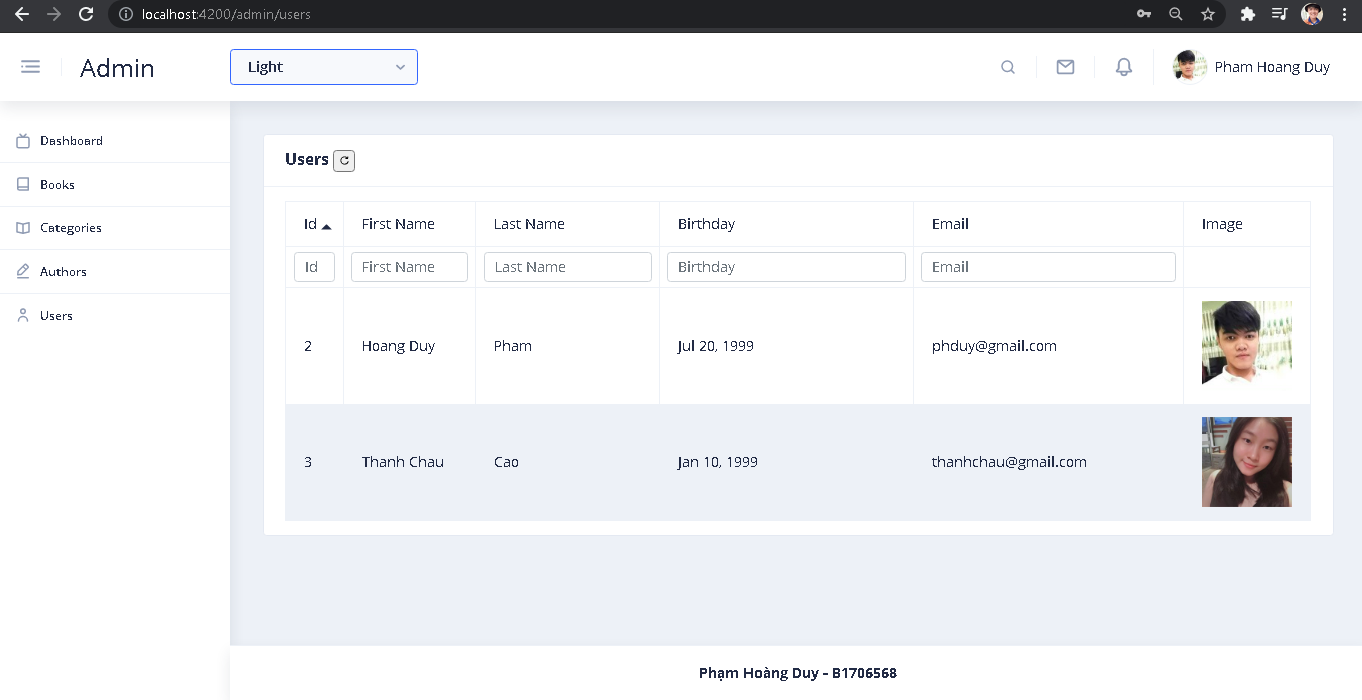
Hình GreetingController

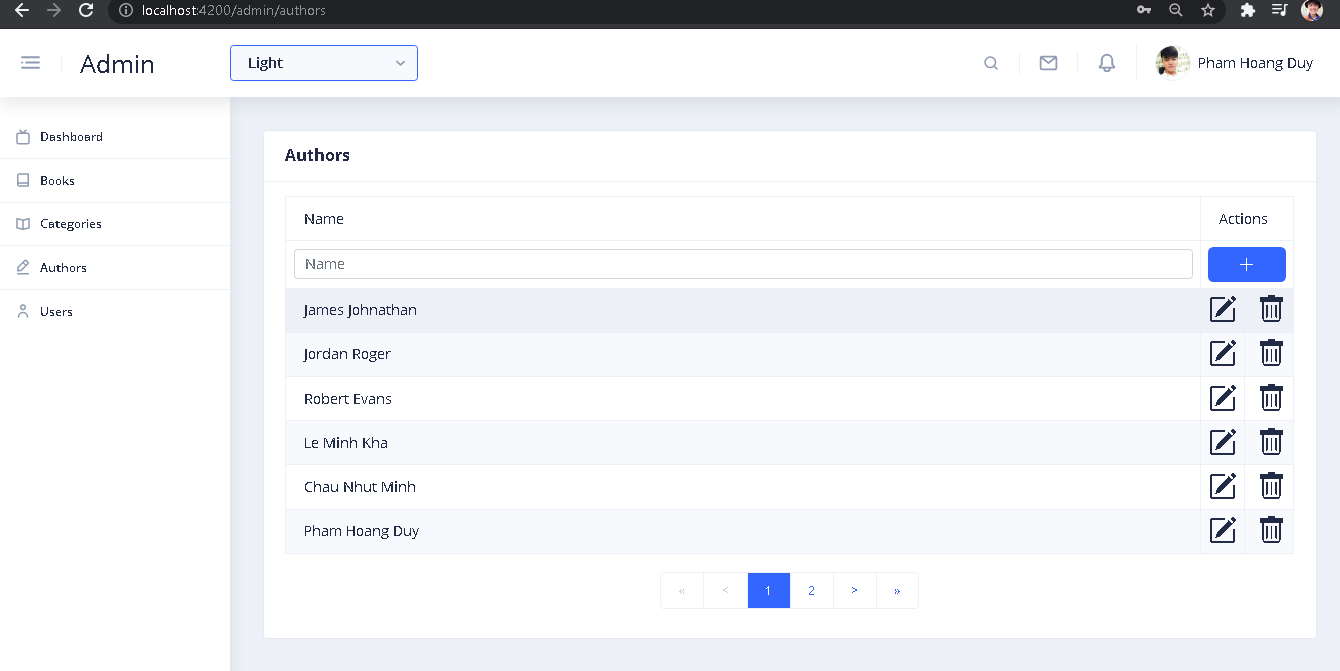


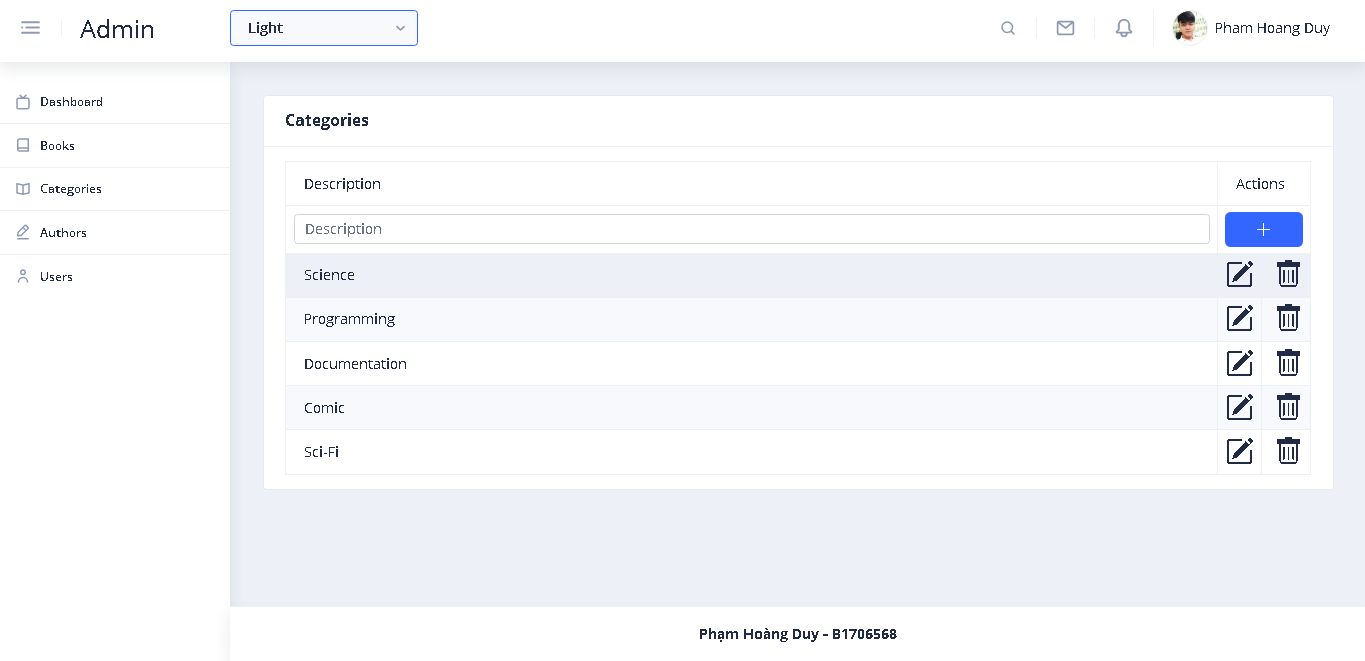
Hình Dữ liệu JSON từ GreetingController

Khi chọn đăng xuất, clear cookies và điều hướng ngược lại trang đăng nhập.

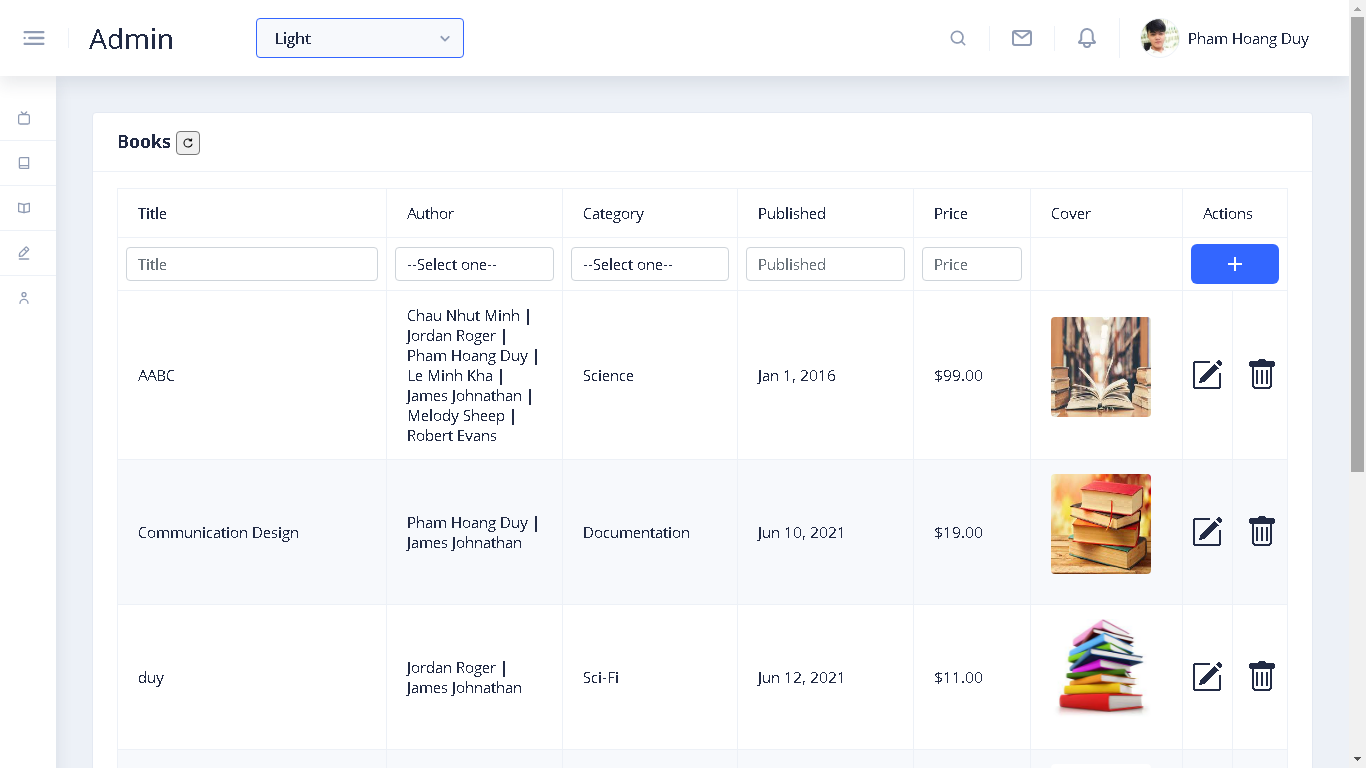
1. Chức năng quản lý:

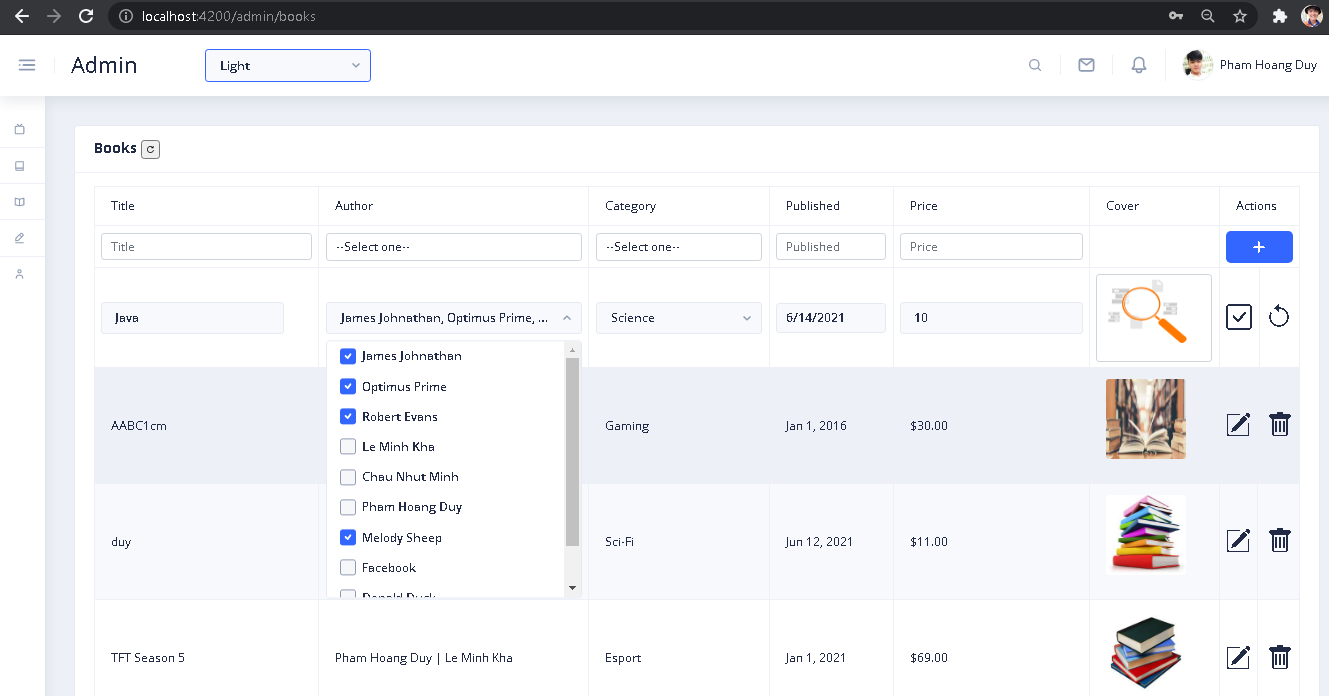
* Xem thông tin user:
* Quản lý tác giả và thể loại:

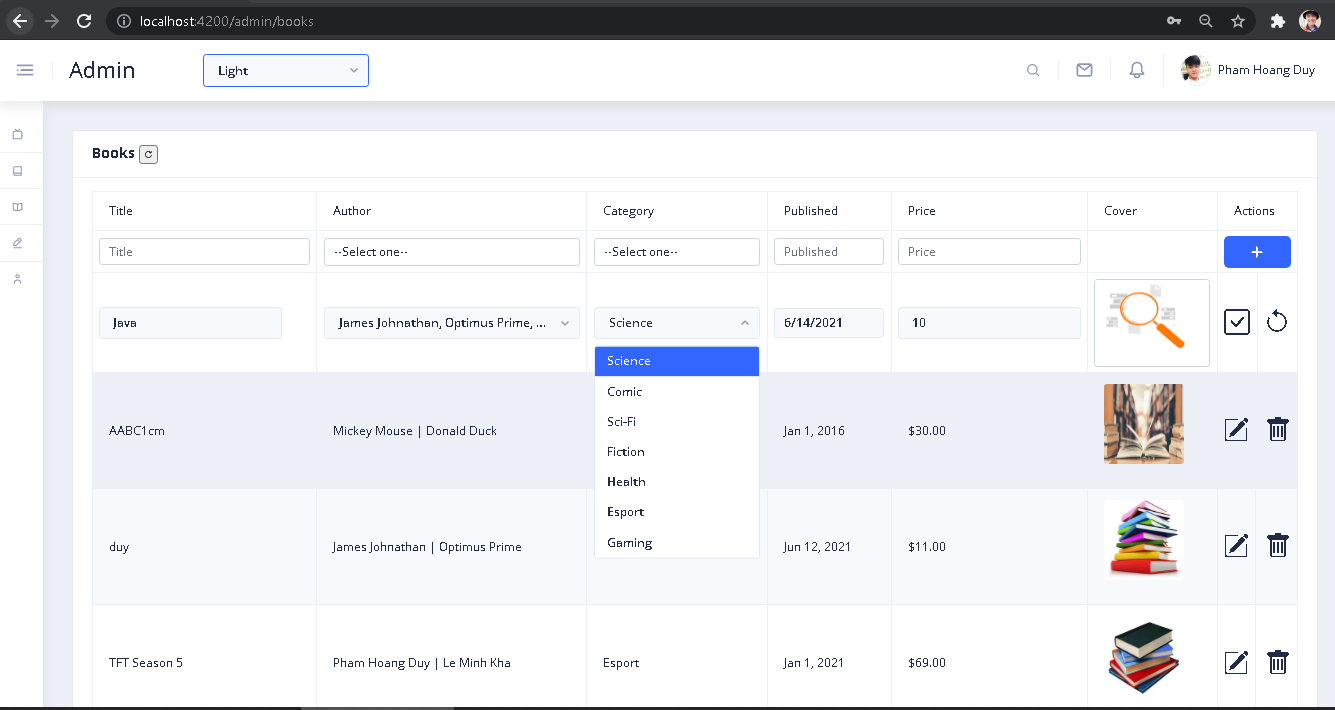


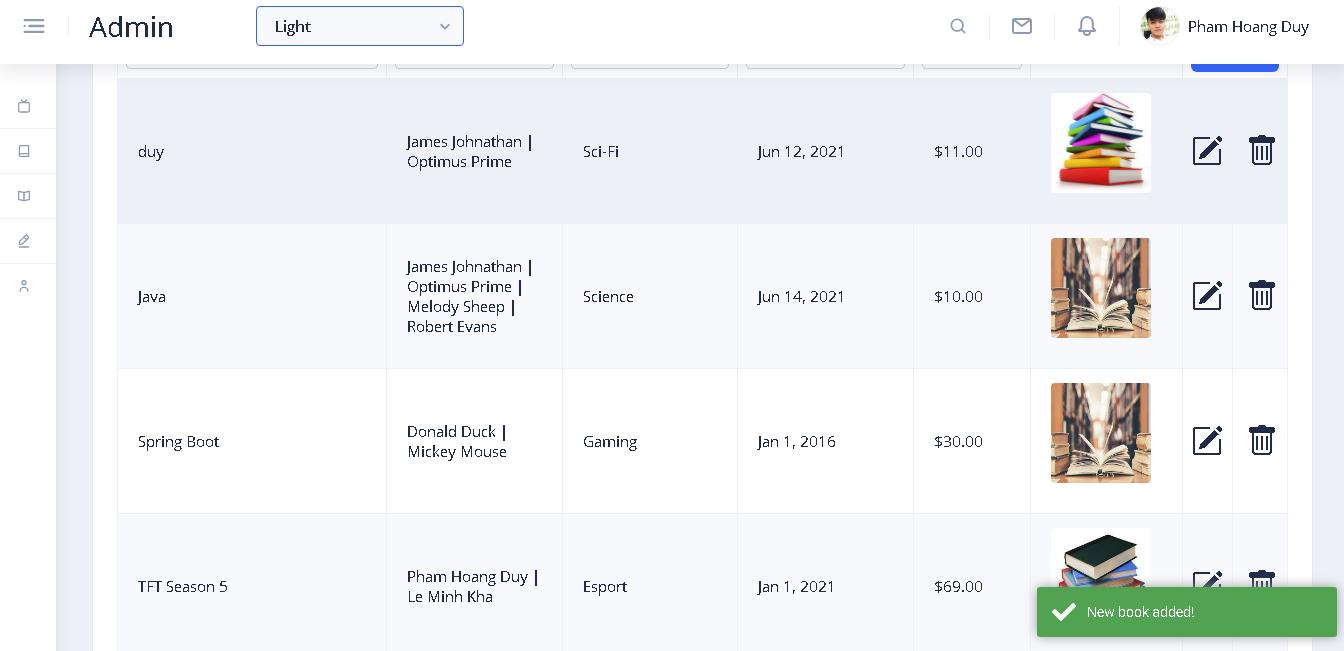


* Quản lý sách:









Các giao diện quản lý tích hợp đầy đủ các chức năng thêm, sửa, xóa, tìm kiếm, phần trang và sort header. Riêng trang quản lý sách ở các trường khóa ngoại Author và Category dữ liệu bộ lọc và editor được giới hạn trong các dropdown. Có thể điều chỉnh casdeType tùy theo nhu cầu sử dụng.

1. **Kết luận và hướng phát triển**
2. **Kết luận**

Cơ bản hoàn thành các mục tiêu đề ra: sử dụng được Spring Boot cùng với Angular, quản lý dữ liệu bằng JPA/Hibernate tuy nhiên ở giai đoạn chỉnh sửa vẫn còn một vài lỗi nhỏ liên quan tới cascadeType.

1. **Hướng phát triển**

Sửa lỗi, mở rộng dự án: viết trang thương mạng với tính năng giỏ hàng, thanh toán, v.v.