

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO MÔN HỌC**  
**CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI**

**WEB QUẢN LÝ MƯỢN/TRẢ SÁCH**

*Giảng viên hướng dẫn:*

**TS. Nguyễn Bảo Ân**

*Sinh viên thực hiện:*

**Lê Hà Duy - 110122060**

**Đỗ Thị Kim Hương - 110122083**

**Lớp: DA22TTD**

**Vĩnh Long, tháng 7 năm 2025**

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO MÔN HỌC**  
**CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI**

**WEB QUẢN LÝ MƯỢN/TRẢ SÁCH**

*Giảng viên hướng dẫn:*

**TS. Nguyễn Bảo Ân**

*Sinh viên thực hiện:*

**Lê Hà Duy - 110122060**

**Đỗ Thị Kim Hương - 110122083**

**Lớp: DA22TTD**

**Vĩnh Long, tháng 7 năm 2025**

[illegible]

**i**

[illegible]

**Giảng viên phản biện**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn đến quý thầy cô trường Đại học Trà Vinh nói chung, đặc biệt là các thầy cô trong khoa Kỹ thuật và Công nghệ đã tạo điều kiện thuận lợi để chúng em có cơ hội học tập, thực hành, và tiếp cận với môi trường làm việc thực tế. Nhờ đó, chúng em đã tích lũy được nhiều kinh nghiệm quý giá, giúp chúng em tự tin hơn khi bước vào môi trường làm việc sau này.

Chúng em cũng xin cảm ơn TS. Nguyễn Bảo Ân – người đã tận tình hướng dẫn và hỗ trợ nhóm em xuyên suốt quá trình thực hiện đề tài. Nhờ những chia sẻ, chỉ dẫn cụ thể từ cô mà nhóm em đã có thể hoàn thành báo cáo đúng tiến độ, đồng thời tích lũy được nhiều kiến thức hữu ích cho bản thân.

Dù đã nỗ lực để hoàn thành tốt nhất có thể, nhưng do giới hạn về thời gian và kinh nghiệm nên đề tài vẫn còn những thiếu sót. Nhóm chúng em rất mong nhận được sự góp ý, nhận xét từ quý thầy cô để có thể hoàn thiện đề tài tốt hơn trong tương lai.

Chúng em xin chân thành cảm ơn!!

## MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU .....</b>	<b>1</b>
1.1 Tổng quan về dự án .....	1
1.2 Mục tiêu của ứng dụng .....	1
1.3 Lý do chọn đề tài .....	1
<b>CHƯƠNG 2: PHÂN TÍCH YÊU CẦU .....</b>	<b>3</b>
2.1 Yêu cầu chức năng .....	3
2.2 Yêu cầu phi chức năng .....	4
<b>CHƯƠNG 3: THIẾT KẾ HỆ THỐNG .....</b>	<b>6</b>
3.1 Kiến trúc tổng thể .....	6
3.2 Cấu trúc thư mục .....	7
3.2 Thiết kế cơ sở dữ liệu .....	8
3.3 Thiết kế API .....	9
3.4 Thiết kế giao diện (UI/UX) .....	12
3.4.1 Thiết kế giao diện UI .....	12
3.4.2 Thiết kế giao diện UX .....	18
<b>CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG .....</b>	<b>28</b>
4.1 Danh sách công nghệ sử dụng .....	28
4.2 Quy trình CI/CD với GitHub Actions .....	29
4.3 Cấu hình Docker và quy trình triển khai .....	29
<b>CHƯƠNG 5: QUẢN LÝ DỰ ÁN .....</b>	<b>31</b>
5.1 Sử dụng Jira trong quản lý tiến độ .....	31
5.2 Phân công nhiệm vụ các thành viên nhóm .....	31
<b>CHƯƠNG 6: KIỂM THỬ .....</b>	<b>35</b>
6.1 Chiến lược kiểm thử và công cụ sử dụng .....	35
6.2 Kết quả kiểm thử API .....	36
<b>CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN .....</b>	<b>39</b>
7.1 Những khó khăn gặp phải .....	39
7.2 Bài học rút ra .....	39
7.3 Đề xuất cải tiến trong tương lai .....	39
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>40</b>
<b>PHỤ LỤC .....</b>	<b>41</b>
1. Yêu cầu hệ thống .....	41
2. Clone mã nguồn từ GitHub .....	41

3. Cài đặt và cấu hình backend .....	41
4. Khởi chạy ứng dụng bằng Docker .....	41

## DANH MỤC HÌNH ẢNH

Hình 3.1 Sơ đồ kiến trúc .....	6
Hình 3.2 Cấu trúc thư mục hệ thống .....	8
Hình 3.3 Mô hình thực thể kết hợp .....	9
Hình 3.4 Form đăng ký với Figma .....	12
Hình 3.5 Form đăng nhập với Figma .....	12
Hình 3.6 Giao diện trang chủ với Figma .....	13
Hình 3.7 Giao diện trang danh sách Sách với Figma .....	13
Hình 3.8 Giao diện trang nhà xuất bản với Figma .....	14
Hình 3.9 Giao diện trang tác giả với Figma .....	14
Hình 3.10 Giao diện trang thể loại với Figma .....	15
Hình 3.11 Giao diện phiếu mượn sách với Figma .....	15
Hình 3.12 Giao diện phiếu trả sách với Figma .....	16
Hình 3.13 Giao diện phí phạt với Figma .....	16
Hình 3.14 Giao diện biểu đồ lượt mượn với Figma .....	17
Hình 3.15 Giao diện trang quản lý độc giả với Figma .....	17
Hình 3.16 Giao diện trang quản lý nhân viên với Figma .....	18
Hình 3.17 Form đăng ký .....	18
Hình 3.18 Form đăng nhập .....	19
Hình 3.19 Giao diện trang chủ .....	19
Hình 3.20 Giao diện trang chủ .....	20
Hình 3.21 Giao diện trang Danh sách Sách .....	20
Hình 3.22 Giao diện trang Nhà xuất bản .....	21
Hình 3.23 Giao diện trang Tác giả .....	22
Hình 3.24 Giao diện trang Thể loại sách .....	22
Hình 3.25 Giao diện Phiếu mượn sách .....	23
Hình 3.26 Giao diện Phiếu trả sách .....	24
Hình 3.27 Giao diện trang Phí phạt .....	24
Hình 3.28 Giao diện Biểu đồ lượt mượn theo tháng .....	25
Hình 3.29 Giao diện Quản lý Độc giả .....	26
Hình 3.30 Giao diện Quản lý Nhân viên .....	26
Hình 5.1 Phân công nhiệm vụ các thành viên ở Sprint 1 trên Jira .....	32
Hình 5.2 Phân công nhiệm vụ các thành viên ở Sprint 2 trên Jira .....	32
Hình 5.3 Phân công nhiệm vụ các thành viên ở Sprint 3 trên Jira .....	32



Hình 5.4 Phân công nhiệm vụ các thành viên ở Sprint 4 trên Jira .....	33
Hình 5.5 Phân công nhiệm vụ các thành viên ở Sprint 4 trên Jira .....	33
Hình 6.1 Postman kiểm tra Đăng ký .....	36
Hình 6.2 Postman kiểm tra Đăng nhập .....	37
Hình 6.3 Postman lấy Danh sách Sách .....	37
Hình 6.4 Postman kiểm tra thêm sách .....	38

## DANH MỤC BẢNG

Bảng 3.1 Các API liên quan Auth .....	9
Bảng 3.2 Các API liên quan Book .....	10
Bảng 3.3 Các API liên quan Borrow .....	10
Bảng 3.4 Các API liên quan Notification .....	10
Bảng 3.5 Các API liên quan Category .....	10
Bảng 3.6 Các API liên quan Employee .....	11
Bảng 3.7 Các API liên quan Fine .....	11
Bảng 3.8 Các API liên quan Publisher .....	11
Bảng 3.9 Các API liên quan Reader .....	11
Bảng 3.10 Các API liên quan Return .....	11
Bảng 5.1 Chi tiết và độ ưu tiên công việc .....	33

## CHƯƠNG 1: GIỚI THIỆU

### 1.1 Tổng quan về dự án

Tên dự án: Web Quản lý Mượn/Trả Sách

Chủ đề: Xây dựng hệ thống hỗ trợ quản lý hoạt động mượn - trả sách, giúp tự động hóa các quy trình như đăng ký mượn, trả sách, kiểm tra tình trạng sách, quản lý độc giả, theo dõi hạn trả và tính năng thông báo quá hạn.

### 1.2 Mục tiêu của ứng dụng

Ứng dụng "Quản lý Mượn/Trả Sách" được xây dựng nhằm mục đích hỗ trợ việc quản lý hiệu quả quá trình mượn và trả sách của độc giả. Cụ thể, ứng dụng hướng đến các mục tiêu sau:

Quản lý thành viên: Cung cấp các chức năng cho phép người dùng đăng ký tài khoản mới, đăng nhập an toàn thông qua xác thực JWT, dễ dàng cập nhật thông tin cá nhân và quản lý hồ sơ của họ.

Tự động hóa quy trình quản lý mượn – trả sách, thay thế thao tác thủ công bằng hệ thống phần mềm để sử dụng.

Quản lý sách: Cho phép thủ thư thêm, sửa, xóa thông tin sách một cách linh hoạt, đồng thời theo dõi trạng thái hiện tại của từng cuốn sách (có sẵn để mượn hoặc đang được mượn) để tối ưu hóa việc sử dụng tài nguyên.

Theo dõi hoạt động mượn trả của từng độc giả: ghi nhận thời gian mượn, thời hạn trả, kiểm tra tình trạng quá hạn.

Cảnh báo quá hạn và hỗ trợ gửi thông báo nhắc nhở đến độc giả.

Quản lý độc giả, lưu trữ thông tin thành viên, thống kê số lượt mượn và lịch sử mượn sách.

Thống kê và báo cáo trực quan: Cung cấp các biểu đồ và số liệu thống kê chi tiết về lượt mượn sách theo tháng.

Tối ưu hóa trải nghiệm người dùng (UX): Thiết kế giao diện thân thiện, dễ điều hướng, đảm bảo ứng dụng có thể truy cập và sử dụng hiệu quả trên mọi thiết bị (máy tính, máy tính bảng, điện thoại di động) thông qua thiết kế responsive.

### 1.3 Lý do chọn đề tài

Trong bối cảnh nhu cầu đọc sách và tra cứu tài liệu ngày càng tăng, việc quản lý hiệu quả hoạt động mượn và trả sách trở nên cần thiết, đặc biệt đối với các đơn vị,

tổ chức hoặc cá nhân có số lượng sách lớn. Các phương pháp quản lý truyền thống dựa trên ghi chép giấy hoặc phần mềm đơn giản thường gặp nhiều bất cập như thất thoát sách, khó kiểm soát thời hạn mượn hoặc mất nhiều thời gian tra cứu thông tin. Vì vậy, việc xây dựng một hệ thống quản lý mượn/trả sách hiện đại, tự động hóa và dễ sử dụng là cần thiết nhằm nâng cao hiệu quả quản lý, tiết kiệm thời gian và đảm bảo tính chính xác trong quá trình theo dõi, sử dụng sách.

## CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

### 2.1 Yêu cầu chức năng

#### Quản lý tài khoản người dùng

Đăng ký thành viên mới: Người dùng có thể tự đăng ký tài khoản mới thông qua giao diện thân thiện, cung cấp các thông tin cơ bản cần thiết như tên, email và mật khẩu. Hệ thống sẽ kiểm tra tính hợp lệ của dữ liệu đầu vào.

Đăng nhập với xác thực JWT: Người dùng sẽ đăng nhập bằng tên đăng nhập/email và mật khẩu. Backend sẽ tạo ra một JSON Web Token (JWT) sau khi xác thực thành công, được sử dụng cho các yêu cầu tiếp theo.

Đăng xuất: Cho phép người dùng kết thúc phiên làm việc.

Middleware bảo vệ các route riêng tư: Các endpoint API yêu cầu xác thực sẽ được bảo vệ bởi middleware. Middleware này sẽ kiểm tra JWT trong mỗi yêu cầu. Nếu JWT hợp lệ, yêu cầu sẽ được xử lý; ngược lại, yêu cầu sẽ bị từ chối với lỗi xác thực.

#### Quản lý sách

Hiện thị danh sách sách với trạng thái (Available / Borrowed): Hệ thống sẽ hiển thị toàn bộ danh sách sách trong thư viện, kèm theo thông tin chi tiết và trạng thái hiện tại của từng cuốn sách (có sẵn để mượn hoặc đang được mượn).

Thêm sách mới vào hệ thống: nhân viên có quyền thêm các cuốn sách mới vào hệ thống, nhập đầy đủ các thông tin như tên sách, tác giả, nhà xuất bản, số trang, năm xuất bản, và thể loại.

Chỉnh sửa thông tin sách hiện có: Cho phép nhân viên cập nhật thông tin của các cuốn sách đã có trong hệ thống, bao gồm việc sửa lỗi hoặc bổ sung chi tiết.

Xóa sách khỏi hệ thống: Cung cấp chức năng để loại bỏ thông tin của một cuốn sách khỏi cơ sở dữ liệu, thường áp dụng cho các sách bị mất, hỏng hoặc không còn được lưu trữ trong thư viện.

#### Quản lý phiếu mượn/trả

Tạo phiếu mượn sách: Ghi nhận thông tin khi một độc giả mượn sách, bao gồm mã độc giả, mã sách, ngày mượn, ngày hẹn trả và số tiền đặt cọc (nếu có).

Cập nhật trạng thái phiếu mượn (open/closed/overdue): Phiếu mượn sẽ tự động hoặc được cập nhật thủ công trạng thái: “open” (đang mượn), “closed” (đã trả sách), hoặc “overdue” (quá hạn chưa trả).

Ghi nhận phiếu trả sách: Khi độc giả trả sách, hệ thống sẽ ghi nhận ngày trả thực tế, tình trạng sách khi trả và tính toán phí phạt nếu có.

### **Hệ thống thông báo**

Gửi email nhắc nhở quá hạn trả sách: Hệ thống tự động quét các phiếu mượn có trạng thái “quá hạn chưa trả”, sau đó gửi email thông báo nhắc nhở đến độc giả tương ứng.

Thiết lập nhắc nhở thành viên sắp tới hạn trả sách: Gửi email thông báo cho độc giả trước ngày hẹn trả một khoảng thời gian nhất định (ví dụ: trước 3 ngày) để nhắc nhở họ trả sách đúng hạn.

### **Báo cáo và thống kê**

Hiện thị số lượng phiếu mượn theo trạng thái (open/closed/overdue): Cung cấp một cái nhìn tổng quan về tình hình mượn/trả sách thông qua số liệu cụ thể cho từng trạng thái phiếu mượn.

Tạo biểu đồ lượt mượn theo tháng: Hiện thị biểu đồ cột biểu diễn số lượng sách được mượn trong từng tháng, giúp dễ dàng theo dõi xu hướng hoạt động.

### **Các tính năng khác**

Tìm kiếm nhanh sách: Cung cấp chức năng tìm kiếm mạnh mẽ cho phép người dùng tìm kiếm sách dựa trên các tiêu chí khác nhau (tên, mã, tác giả, v.v.).

Hiện thị trạng thái loading, thông báo (toast) khi mượn-trả, tìm kiếm nhanh: Cải thiện trải nghiệm người dùng bằng cách hiển thị các thông báo trực quan về quá trình xử lý (ví dụ: Đang tải..., mượn sách thành công!), giúp người dùng biết được trạng thái của hành động họ vừa thực hiện.

## **2.2 Yêu cầu phi chức năng**

### **Hiệu suất**

API phản hồi nhanh chóng, đảm bảo trải nghiệm mượt mà cho người dùng: Hệ thống được thiết kế để tối ưu hóa thời gian phản hồi của API, đảm bảo rằng mọi thao tác từ người dùng, từ việc tìm kiếm sách đến thực hiện giao dịch mượn/trả, đều diễn ra mượt mà và không bị gián đoạn, ngay cả khi có lượng lớn người dùng truy cập đồng thời.

### **Bảo mật**

Sử dụng JWT để xác thực và ủy quyền người dùng: Toàn bộ quá trình xác thực và ủy quyền đều dựa trên JWT, một tiêu chuẩn bảo mật mạnh mẽ, giúp đảm bảo tính toàn vẹn và an toàn cho phiên làm việc của người dùng.

### **Khả năng mở rộng**

Sử dụng Docker để dễ dàng triển khai trong môi trường container: Ứng dụng được đóng gói trong Docker containers, giúp cô lập môi trường chạy, đảm bảo tính nhất quán giữa môi trường phát triển, kiểm thử và triển khai, đồng thời tối ưu hóa quá trình mở rộng tài nguyên.

### **Khả năng sử dụng**

Giao diện người dùng trực quan, dễ hiểu và dễ thao tác: Thiết kế UI/UX tập trung vào sự đơn giản, rõ ràng, giúp người dùng mới cũng có thể dễ dàng làm quen và sử dụng ứng dụng mà không cần nhiều hướng dẫn.

Tối ưu hóa cho các thiết bị di động (responsive design): Giao diện sẽ tự động điều chỉnh để phù hợp với kích thước màn hình của các thiết bị khác nhau (máy tính để bàn, máy tính bảng, điện thoại thông minh), đảm bảo trải nghiệm nhất quán và tối ưu.

### **Khả năng duy trì**

Mã nguồn rõ ràng, có cấu trúc tốt, dễ đọc và dễ bảo trì: Mã nguồn được viết theo các tiêu chuẩn lập trình tốt, có chú thích đầy đủ và tuân thủ các quy tắc về cấu trúc thư mục, giúp các nhà phát triển dễ dàng hiểu và bảo trì trong tương lai.

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

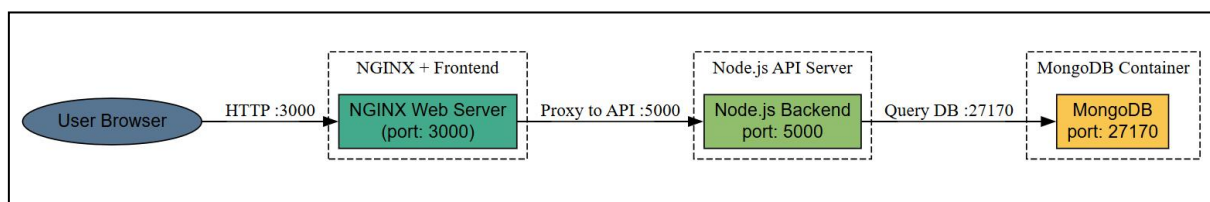
### 3.1 Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình kiến trúc Client-Server ba lớp, phân tách rõ ràng giữa giao diện người dùng, logic nghiệp vụ và lưu trữ dữ liệu. Điều này giúp tăng cường tính module hóa, khả năng mở rộng và dễ dàng bảo trì.

**Frontend (Client-side):** Là lớp giao diện người dùng, nơi độc giả và nhân viên tương tác trực tiếp với hệ thống. Nó được xây dựng bằng các công nghệ web tiêu chuẩn như HTML, CSS và JavaScript. Lớp này chịu trách nhiệm hiển thị giao diện người dùng, thu thập input từ người dùng và gửi các yêu cầu đến Backend thông qua các lời gọi API. Giao diện người dùng sẽ được thiết kế responsive để tương thích với nhiều kích thước màn hình thiết bị.

**Backend (Server-side):** Là lớp Logic nghiệp vụ (Application), được phát triển bằng Node.js với framework Express.js. Lớp này đóng vai trò trung tâm, tiếp nhận các yêu cầu từ Frontend, xử lý logic nghiệp vụ phức tạp (xác thực người dùng, tính toán trạng thái sách, xử lý giao dịch mượn/trả), và tương tác với Cơ sở dữ liệu. Backend cung cấp một tập hợp các RESTful API để Frontend có thể giao tiếp.

**Cơ sở dữ liệu:** Lớp này chịu trách nhiệm lưu trữ và quản lý toàn bộ dữ liệu của hệ thống, bao gồm thông tin sách, độc giả, nhân viên, phiếu mượn, phiếu trả,... Hệ thống sử dụng HQTCSDL phi quan hệ (MongoDB), một cơ sở dữ liệu NoSQL, cho phép lưu trữ dữ liệu linh hoạt và mở rộng dễ dàng.



Hình 3.1 Sơ đồ kiến trúc

#### Luồng xử lý tổng quát

**User Browser (trình duyệt người dùng):** Người dùng gửi yêu cầu HTTP đến cổng 3000.

**NGINX Web Server (port: 3000):** Đây là nơi phục vụ giao diện người dùng (frontend), NGINX đồng thời đảm nhận vai trò reverse proxy để chuyển tiếp các yêu



cầu API đến backend, nếu trình duyệt truy cập /api, NGINX sẽ proxy yêu cầu đến port 5000 (Node.js backend).

Node.js API Server (port: 5000): Chịu trách nhiệm xử lý logic nghiệp vụ và yêu cầu API từ frontend (qua NGINX). Khi cần truy xuất dữ liệu, backend sẽ gửi truy vấn tới MongoDB.

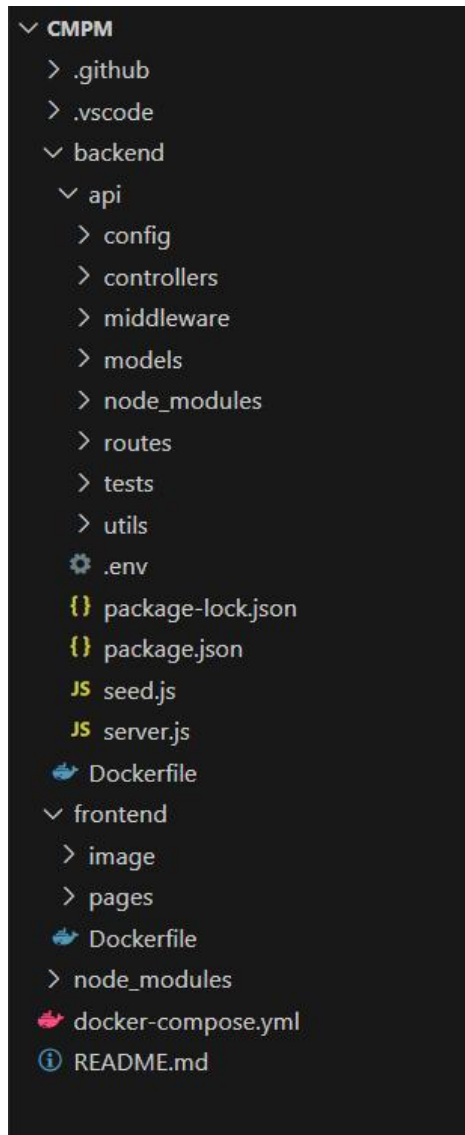
MongoDB Container (port: 27170): Cơ sở dữ liệu NoSQL dùng để lưu trữ thông tin của ứng dụng. Backend truy vấn đến port 27170 để lấy hoặc ghi dữ liệu.

### 3.2 Cấu trúc thư mục

Hệ thống được tổ chức thành hai phần chính: backend và frontend, cùng với các tệp cấu hình hỗ trợ triển khai bằng Docker.

Thư mục backend, được xây dựng bằng Node.js. Bên trong có thư mục api chứa toàn bộ mã nguồn ứng dụng phía server, bao gồm các thư mục như config (cấu hình hệ thống), controllers (xử lý logic nghiệp vụ), middleware (xử lý xác thực, phân quyền), models (định nghĩa các schema dữ liệu với MongoDB), routes (các API endpoint), utils (các hàm tiện ích dùng chung) và tests (chứa các bài test). Ngoài ra, các tệp như server.js dùng để khởi chạy server, seed.js dùng để khởi tạo dữ liệu mẫu vào cơ sở dữ liệu, và .env chứa các biến môi trường quan trọng như PORT và DB\_URI. Backend còn có một Dockerfile để tạo image Docker phục vụ cho việc triển khai.

Thư mục frontend là nơi xây dựng giao diện người dùng bằng ngôn ngữ HTML, CSS, Javascript. Chứa thư mục pages (tương ứng với các route giao diện), image (chứa hình ảnh tĩnh) và một Dockerfile để đóng gói frontend vào container, có thể kết hợp với NGINX để serve ra ngoài.



Hình 3.2 Cấu trúc thư mục hệ thống

### 3.2 Thiết kế cơ sở dữ liệu

#### Mô hình thực thể kết hợp (ERD)

Xây dựng cơ sở dữ liệu để quản lý việc mượn/trả sách, trong đó bao gồm các bảng chính như sau:

DOCGIA: Lưu thông tin người dùng trong hệ thống. Mỗi độc giả có các thuộc tính như: MaDG, TenDG, GioiTinh, NgaySinh, DC, SDT, Email.

THELOAISACH: cho phép tổ chức sách theo từng danh mục. Mỗi thể loại có thể chứa nhiều sách, với các thuộc tính như: MaTL, TenTL.

SACH: Là bảng trung tâm của hệ thống. Các thuộc tính bao gồm: MaSach, TenSach, SoTrang.

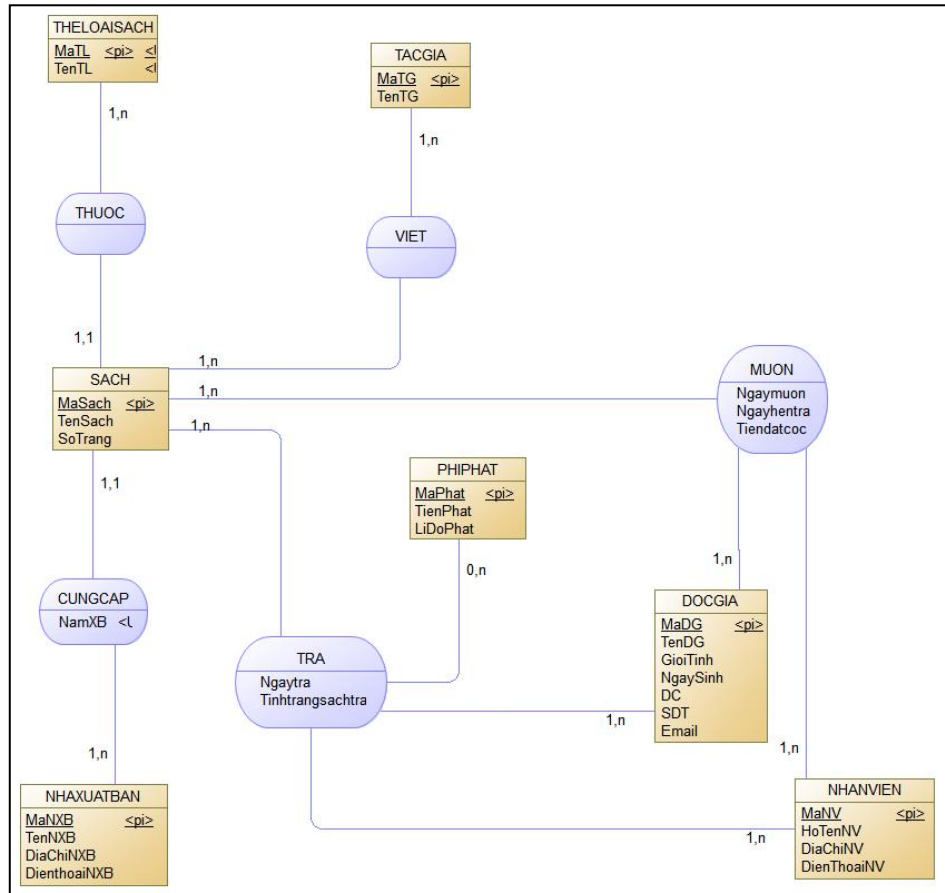
Ngoài ra, hệ thống còn có các bảng hỗ trợ khác để phản ánh quá trình mượn và trả như:

MUON: Ghi lại lịch sử mượn sách (ngày mượn, ngày hẹn trả, tiền đặt cọc).

TRA: Ghi nhận thông tin trả sách (ngày trả, tình trạng sách khi trả).

PHIPHAT: Quản lý các khoản phạt nếu độc giả trả sách trễ hoặc làm hỏng.

TACGIA, NHAXUATBAN, NHANVIEN: Lưu thông tin tác giả, nhà xuất bản và nhân viên xử lý nghiệp vụ mượn trả.



Hình 3.3 Mô hình thực thể kết hợp

### 3.3 Thiết kế API

Các endpoint chính của hệ thống:

Bảng 3.1 Các API liên quan Auth

Method	Endpoint	Mô tả
POST	/api/auth/register	Đăng ký tài khoản mới
POST	/api/auth/login	Đăng nhập
POST	/api/auth/logout	Đăng xuất
GET	/api/auth/verify	Kiểm tra token

Bảng 3.2 Các API liên quan Book

Method	Endpoint	Mô tả
GET	/api/books	Lấy danh sách sách
GET	/api/books/:id	Lấy chi tiết 1 sách
POST	/api/books	Thêm sách mới
PUT	/api/books/:id	Cập nhật sách
DELETE	/api/books/:id	Xoá sách
GET	/api/books/status/:status	Lọc sách theo trạng thái
GET	/api/books/stats/status	Thống kê sách theo trạng thái
POST	/api/books/reminders/overdue	Gửi nhắc nhở sách quá hạn

Bảng 3.3 Các API liên quan Borrow

Method	Endpoint	Mô tả
POST	/api/borrows/add	Thêm phiếu mượn mới
GET	/api/borrows	Lấy danh sách phiếu mượn
PUT	/api/borrows/:id	Cập nhật phiếu mượn
DELETE	/api/borrows/:id	Xoá phiếu mượn
GET	/api/borrows/stats	Thống kê theo trạng thái phiếu mượn
GET	/api/borrows/stats/monthly	Thống kê theo tháng

Bảng 3.4 Các API liên quan Notification

Method	Endpoint	Mô tả
POST	/api/books/reminders/overdue	Gửi email nhắc sách quá hạn

Bảng 3.5 Các API liên quan Category

Method	Endpoint	Mô tả
POST	/api/categories/add	Thêm thể loại mới
GET	/api/categories	Lấy danh sách thể loại
PUT	/api/categories/:maTheLoai	Cập nhật thể loại
DELETE	/api/categories/:maTheLoai	Xoá thể loại

*Bảng 3.6 Các API liên quan Employee*

Method	Endpoint	Mô tả
POST	/api/employees/add	Thêm nhân viên mới
GET	/api/employees	Lấy danh sách nhân viên
PUT	/api/employees/:manv	Cập nhật nhân viên
DELETE	/api/employees/:manv	Xóa nhân viên

*Bảng 3.7 Các API liên quan Fine*

Method	Endpoint	Mô tả
POST	/api/fines/add	Thêm phí phạt
GET	/api/fines	Lấy danh sách phí phạt
PUT	/api/fines/:maphat	Cập nhật phí phạt
DELETE	/api/fines/:maphat	Xoá phí phạt

*Bảng 3.8 Các API liên quan Publisher*

Method	Endpoint	Mô tả
POST	/api/publishers/add	Thêm nhà xuất bản
GET	/api/publishers	Lấy danh sách NXB
PUT	/api/publishers/:manxb	Cập nhật NXB
DELETE	/api/publishers/:manxb	Xoá NXB

*Bảng 3.9 Các API liên quan Reader*

Method	Endpoint	Mô tả
POST	/api/readers/add	Thêm độc giả mới
GET	/api/readers	Lấy danh sách độc giả
PUT	/api/readers/:madocgia	Cập nhật độc giả
DELETE	/api/readers/:madocgia	Xóa độc giả


*Bảng 3.10 Các API liên quan Return*

Method	Endpoint	Mô tả
POST	/api/returns/add	Thêm phiếu trả

GET	/api/returns	Lấy danh sách phiếu trả
PUT	/api/returns/:id	Cập nhật phiếu trả
DELETE	/api/returns/:id	Xoá phiếu trả

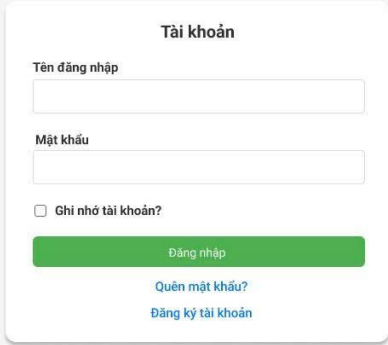
### 3.4 Thiết kế giao diện (UI/UX)

#### 3.4.1 Thiết kế giao diện UI



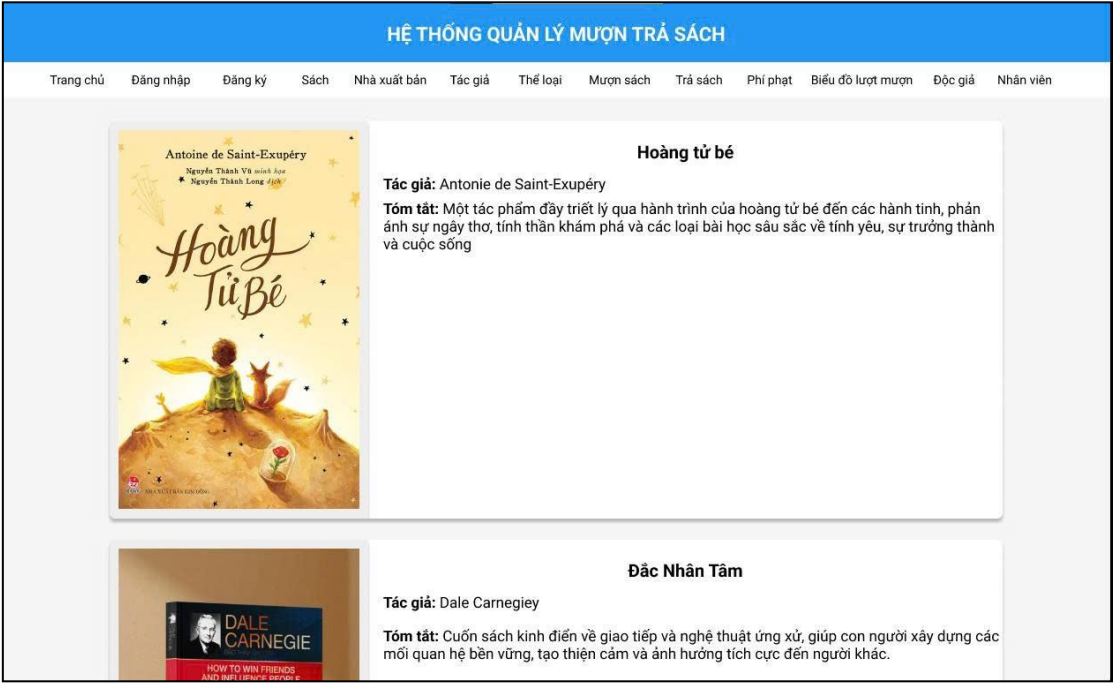
Form đăng ký thành viên (Registration Form) hiển thị các trường nhập liệu: Họ tên, Số điện thoại, Email, và Mật khẩu. Mỗi trường có một nhãn và một ô nhập liệu. Dưới cùng có hai nút: "Hủy bỏ" (Hủy bỏ) và "Đăng ký" (Đăng ký).

Hình 3.4 Form đăng ký với Figma

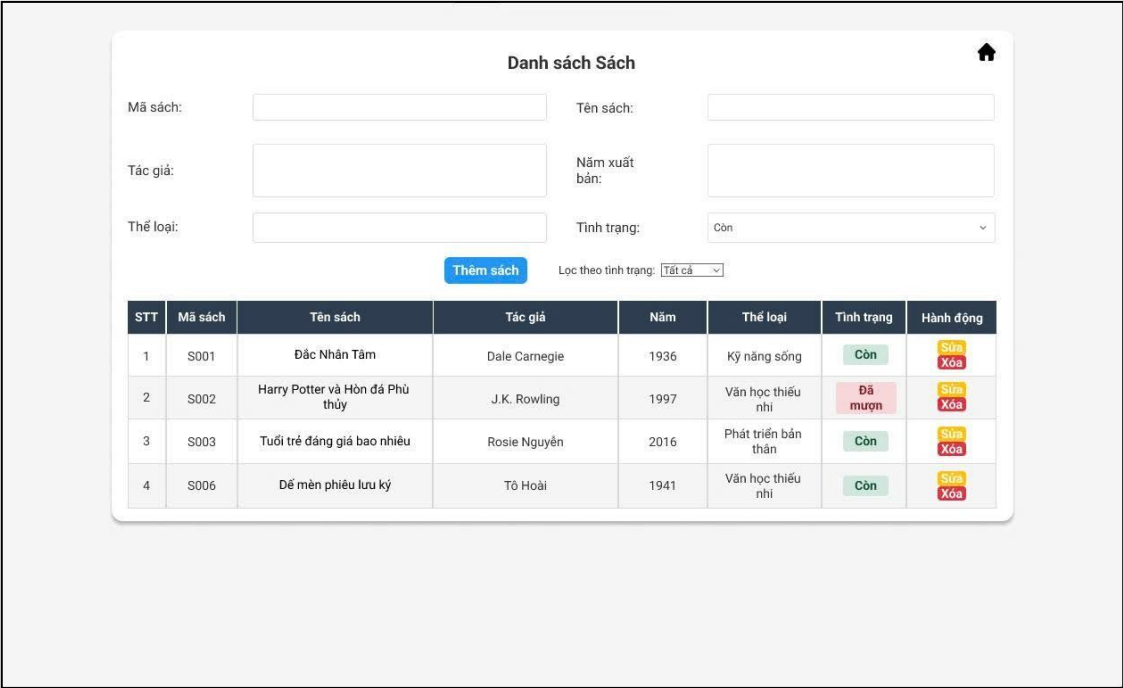


Form đăng nhập tài khoản (Login Form) hiển thị các trường nhập liệu: Tên đăng nhập và Mật khẩu. Dưới các trường nhập liệu là một checkbox "Ghi nhớ tài khoản?". Dưới cùng có một nút "Đăng nhập" (Đăng nhập) và hai liên kết: "Quên mật khẩu?" (Quên mật khẩu?) và "Đăng ký tài khoản" (Đăng ký tài khoản).

Hình 3.5 Form đăng nhập với Figma



Hình 3.6 Giao diện trang chủ với Figma



Hình 3.7 Giao diện trang danh sách Sách với Figma

Nhà xuất bản

Mã NXB:VD: NXB01

Tên NXB:VD: Nhà xuất bản Trẻ

Địa chỉ:VD: 123 Lê Lợi, Q1

SĐT:VD: 0912345678

Thêm

Sửa

Xóa

Mã NXB	Tên NXB	Địa chỉ	SĐT
NXB01	NXB Trẻ	161B Lý Chính Thắng, P. Võ Thị Sáu, Q.3, TP. HCM	(028) 3932 5286
NXB02	NXB Kim Đồng	55 Quang Trung, P. Nguyễn Du, Q. Hai Bà Trưng, Hà Nội	(024) 3943 7878
NXB03	NXB Hội Nhà Văn	9/27 Hoàng Cầu, Đống Đa, Hà Nội	(024) 3514 2436
NXB04	NXB Lao Động	175 Giảng Võ, Ba Đình, Hà Nội	(024) 3846 0590
NXB05	NXB Tổng hợp TP.HCM	62 Nguyễn Thông, Q.3, TP. HCM	(028) 3933 0690
NXB06	NXB Dân Trí	55 Quang Trung, Q. Hai Bà Trưng, Hà Nội	(024) 3944 0846
NXB07	NXB Thế Giới	46 Trần Hưng Đạo, Q. Hoàn Kiếm, Hà Nội	(024) 3825 4068

Hình 3.8 Giao diện trang nhà xuất bản với Figma

Tác giả

Mã tác giả:VD: TG01

Tên tác giả:VD: Nguyễn Nhật Ánh

Thêm

Sửa

Xóa

Mã tác giả	Tên tác giả
TG01	Nguyễn Nhật Ánh
TG02	Tô Hoài
TG03	Antoine de Saint-Exupéry
TG04	Gosho Aoyama
TG05	J.K. Rowling
TG06	Paulo Coelho
TG07	Ichiro Kishimi, Fumitake Koga
TG08	Phùng Quán
TG09	Nguyễn Ngọc Tư

Hình 3.9 Giao diện trang tác giả với Figma



Thế loại sách

Mã thế loại:  
VD: TL01

Tên thế loại:  
VD: Văn học

Thêm

Sửa

Xóa

Mã thế loại	Tên thế loại
TL01	Văn học thiếu nhi
TL02	Truyện tranh

Hình 3.10 Giao diện trang thế loại với Figma

Phiếu Mượn sách

Mã phiếu mượn:  
VD: M01

Mã đọc giả:  
VD: DG01

Mã sách:  
VD: S001

Tiền đặt cọc(VND):  
VD: 50000

Ngày mượn:  
dd / mm / yyyy

Ngày hạn trả:  
dd / mm / yyyy

Xác nhận trả sách

Thống kê Phiếu Mượn

Đang mượn	Đã trả sách	Quá hạn chưa trả
5	0	4

Danh sách phiếu mượn

Mã phiếu	Mã đọc giả	Mã sách	Tiền đặt cọc	Ngày mượn	Ngày hạn trả	Thao tác
M04	DG06	S014	55.000 VND	10/4/2025	2/6/2025	<div>SửaXóa</div>
M05	DG05	S006	50.000 VND	5/7/2025	15/7/2025	<div>SửaXóa</div>
M01	DG07	S005	70.000 VND	10/7/2025	26/7/2025	<div>SửaXóa</div>
M02	DG08	S003	60.000 VND	1/7/2025	10/7/2025	<div>SửaXóa</div>
M03	DG10	S008	55.000 VND	3/6/2025	7/7/2025	<div>SửaXóa</div>

Hình 3.11 Giao diện phiếu mượn sách với Figma

Phiếu Trả sách

Mã trả:  
VD: T01

Mã đọc giả:  
VD: DG01

Tình trạng sách trả:  
--Chọn--

Mã phạt:  
VD: P01 hoặc để trống nếu không phạt

Ngày trả:  
dd / mm / yyyy

Xác nhận trả sách

Mã trả	Mã đọc giả	Tình trạng	Mã phạt	Ngày trả	Hành động
T05	DG01	Bình thường	-	11/7/2025	Sửa Xóa
T04	DG06	Bị hỏng	P01	1/6/2025	Sửa Xóa
T03	DG03	Bị hỏng	P02	10/2/2025	Sửa Xóa
T02	DG02	Bị hỏng	P03	2/7/2025	Sửa Xóa
T01	DG01	Bình thường	-	5/7/2025	Sửa Xóa

Hình 3.12 Giao diện phiếu trả sách với Figma

Quản lý Phí phạt

Mã phạt:  
VD: P01

Mã trả:  
VD: T01

Tiền phạt:  
VD: 50000

Lý do phạt:  
VD: Trả sách trễ

Thêm

Sửa

Xóa

Mã phạt	Mã trả	Tiền phạt	Lý do phạt
P02	T02	10000	Bìa bị gấp
P03	T03	20000	Rách góc

Hình 3.13 Giao diện phí phạt với Figma



Hình 3.14 Giao diện biểu đồ lượt mượn với Figma

Quản lý Độc giả

Mã độc giả:  Họ tên:

Ngày sinh:  Giới tính:

Địa chỉ:  SĐT:

Email:

Mã độc giả	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT	Email
DG01	Trần Văn An	2018-09-02	Nam	123 Nguyễn Trãi, Hà Nội	0987654321	antran@gmail.com
DG02	Nguyễn Thị Bình	2005-06-14	Nữ	465 Lê Lợi, TPHCM	0912345678	binhnguyen@yahoo.com
DG03	Lê Hoàng	1998-09-23	Nam	789 Lạc Long Quân, Đà Nẵng	0909090909	lehoang@gmail.com
DG04	Phạm Thu Trang	2001-07-05	Nữ	321 Cách Mạng, Cần Thơ	0978123456	trangpham@gmail.com
DG05	Vũ Đức Minh	2003-10-27	Nam	22 Nguyễn Văn Cừ, Hải Phòng	0934567890	minhvuduc@gmail.com
DG06	Ngô Thanh Mai	1999-03-11	Nữ	11 Trần Hưng Đạo, Nha Trang	0967123456	mainth@gmail.com

Hình 3.15 Giao diện trang quản lý độc giả với Figma

Quản lý Nhân viên

Mã nhân viên:VD: NV01

Họ tên:VD: Trần Văn B

Địa chỉ:VD: 456 Trần Hưng Đạo, Q5

SĐT:VD: 0912345678

Thêm

Sửa

Xóa

Mã nhân viên	Họ tên	Địa chỉ	SĐT
NV01	Lê Thị Lan	1 Nguyễn Huệ, Hà Nội	0901234567
NV02	Nguyễn Văn Nam	22 Trần Phú, TP.HCM	0912345678
NV03	Phạm Thị Hòa	15 Quang Trung, Đà Nẵng	0923456789
NV04	Hồ Văn Hưng	88 Lê Lợi, Huế	0934567890
NV05	Trịnh Thị Mai	66 Trần Hưng Đạo, Cần Thơ	0945678901

Hình 3.16 Giao diện trang quản lý nhân viên với Figma

3.4.2 Thiết kế giao diện UX

Đăng ký thành viên

Họ tên

Nhập họ tên

Số điện thoại

Nhập số điện thoại

Email

Nhập email

Mật khẩu

Nhập mật khẩu

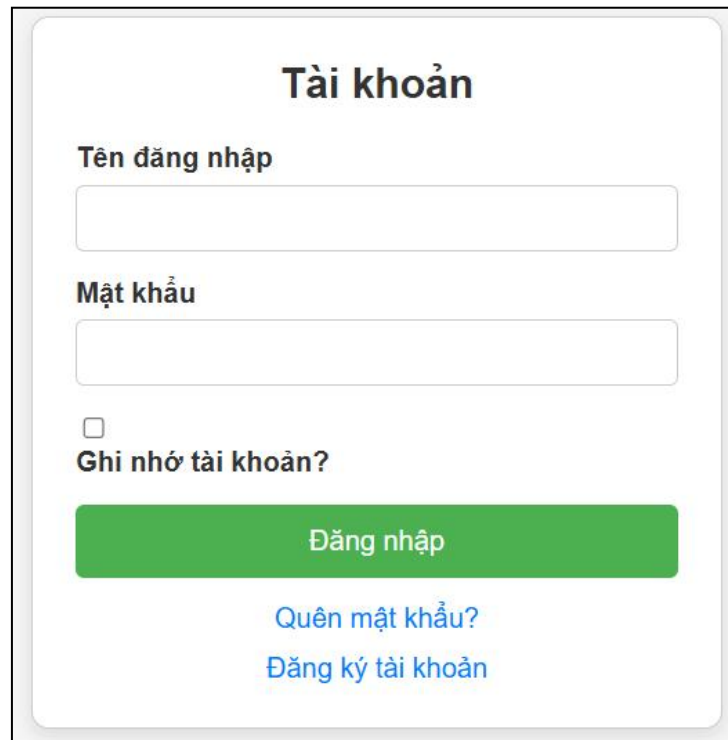
Hủy bỏ

Đăng ký

Hình 3.17 Form đăng ký

Form đăng ký thành viên: Đây là nơi người dùng có thể tạo tài khoản để tham gia vào hệ thống quản lý thư viện. Mục đích của form đăng ký là giúp độc giả dễ dàng kết nối với hệ thống, lưu trữ và quản lý thông tin cá nhân, đồng thời thuận tiện trong việc mượn, trả sách cũng như sử dụng các tiện ích và dịch vụ hỗ trợ đọc sách mà thư viện cung cấp. Giao diện được thiết kế đơn giản, thân thiện với người dùng, gồm các

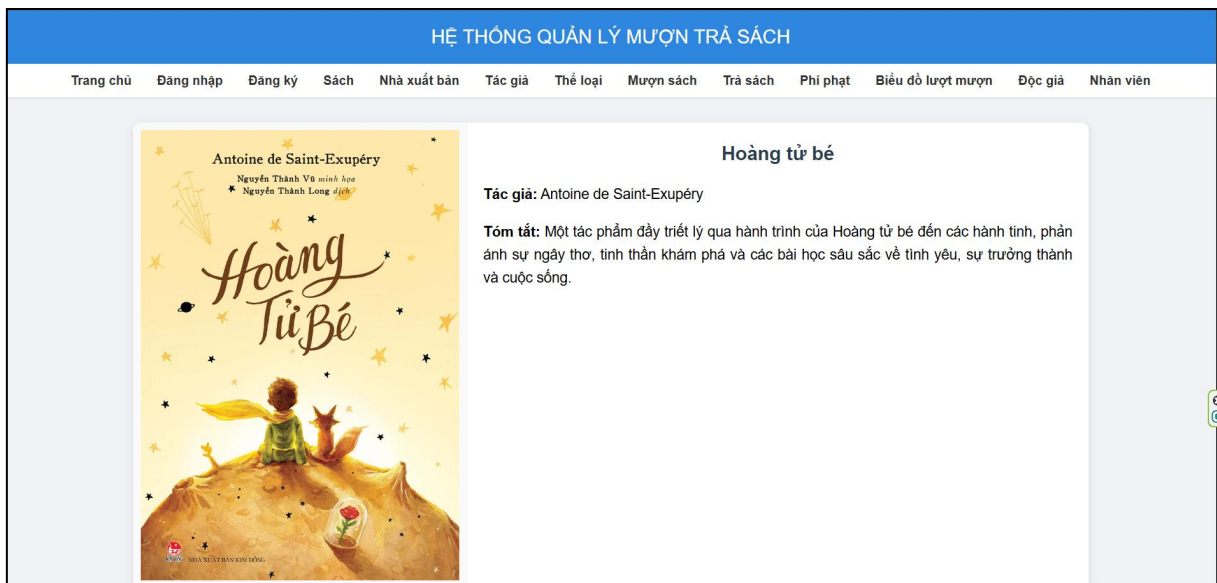
trường nhập thông tin cơ bản như họ tên, số điện thoại, email và mật khẩu, giúp quá trình đăng ký trở nên nhanh chóng và dễ dàng.



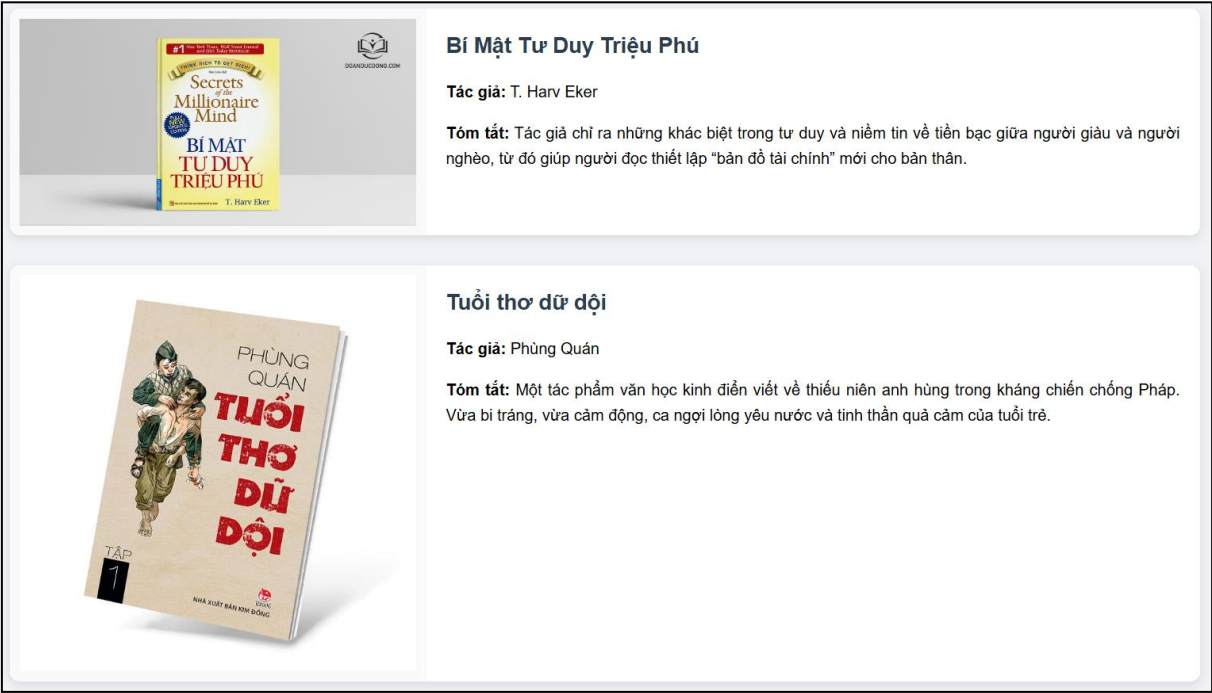
The image shows a login form titled "Tài khoản" (Account). It contains two input fields: "Tên đăng nhập" (Username) and "Mật khẩu" (Password). Below the password field is a checkbox labeled "Ghi nhớ tài khoản?" (Remember account?). A green button labeled "Đăng nhập" (Login) is positioned below the checkbox. At the bottom of the form, there are two links: "Quên mật khẩu?" (Forgot password?) and "Đăng ký tài khoản" (Register account).

Hình 3.18 Form đăng nhập

Form đăng nhập: Đây là giao diện cho phép người dùng truy cập vào hệ thống bằng cách nhập tên đăng nhập và mật khẩu đã được đăng ký trước đó. Việc đăng nhập giúp xác thực danh tính người dùng, từ đó đảm bảo tính bảo mật và cá nhân hóa trải nghiệm.

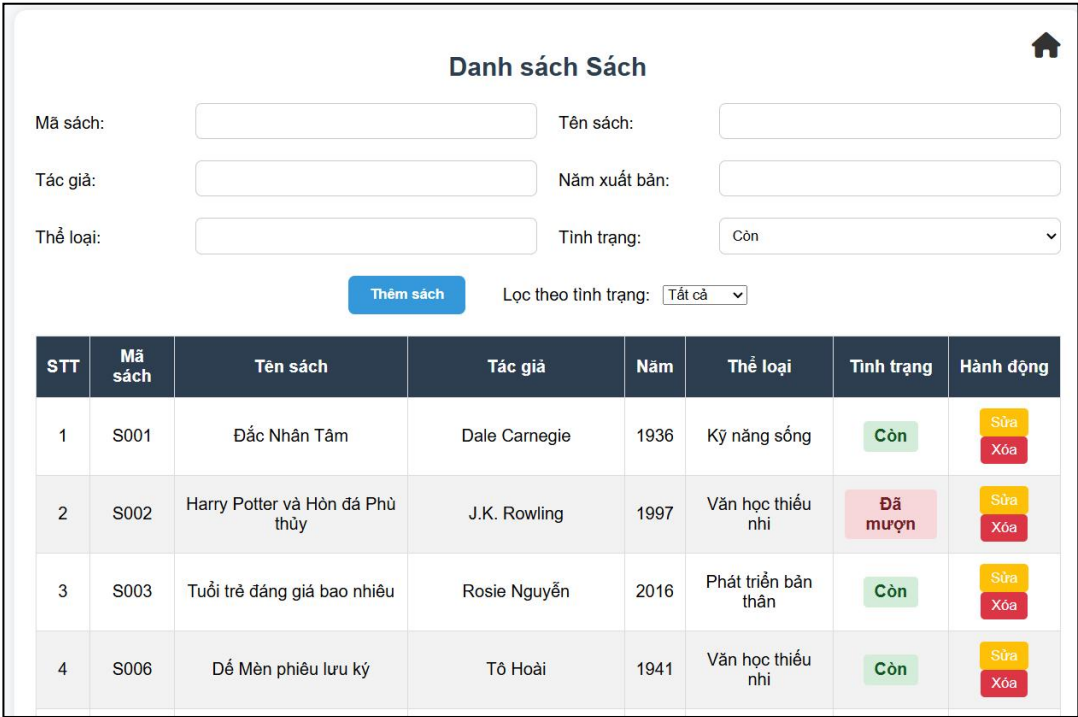


Hình 3.19 Giao diện trang chủ



Hình 3.20 Giao diện trang chủ


Trang chủ: Là giao diện chính của hệ thống quản lý mượn trả sách, hiển thị thông tin giới thiệu các đầu sách nổi bật cùng hình ảnh, tác giả và tóm tắt nội dung. Giao diện được thiết kế trực quan với thanh điều hướng ở đầu trang giúp người dùng dễ dàng truy cập các chức năng như Mượn sách, Trả sách, Phí phạt, Biểu đồ lượt mượn,...Trang chủ là một phần quan trọng của hệ thống.



Hình 3.21 Giao diện trang Danh sách Sách

Trang Danh sách Sách: Hiển thị và quản lý thông tin các đầu sách trong hệ thống. Người dùng có thể xem danh sách các sách có trong hệ thống, tìm kiếm, lọc tình trạng sách theo trạng thái còn hoặc đã mượn. Nhân viên hệ thống có thể thực hiện thao tác thêm, sửa, xóa sách dễ dàng. Bảng danh sách thể hiện rõ ràng thông tin chi tiết và hỗ trợ quản lý sách hiệu quả.

### Nhà xuất bản



Mã NXB:

Tên NXB:

Địa chỉ:

SĐT:

Thêm

Sửa

Xóa

Mã NXB	Tên NXB	Địa chỉ	SĐT
NXB01	NXB Trẻ	161B Lý Chính Thắng, P. Võ Thị Sáu, Q.3, TP. HCM	(028) 3932 5286
NXB02	NXB Kim Đồng	55 Quang Trung, P. Nguyễn Du, Q. Hai Bà Trưng, Hà Nội	(024) 3943 7878
NXB03	NXB Hội Nhà Văn	9/27 Hoàng Cầu, Đống Đa, Hà Nội	(024) 3514 2436
NXB04	NXB Lao Động	175 Giảng Võ, Ba Đình, Hà Nội	(024) 3846 0590
NXB05	NXB Tổng hợp TP.HCM	62 Nguyễn Thông, Q.3, TP. HCM	(028) 3933 0690
NXB06	NXB Dân Trí	55 Quang Trung, Q. Hai Bà Trưng, Hà Nội	(024) 3944 0846
NXB07	NXB Thế Giới	46 Trần Hưng Đạo, Q. Hoàn Kiếm, Hà Nội	(024) 3825 4068

Hình 3.22 Giao diện trang Nhà xuất bản

Trang Nhà xuất bản: Đây là giao diện dùng để quản lý thông tin các nhà xuất bản trong hệ thống. Người dùng xem thông tin các nhà xuất bản. Nhân viên hệ thống có thể thêm mới, chỉnh sửa hoặc xóa nhà xuất bản thông qua các trường nhập liệu như mã NXB, tên NXB, địa chỉ và số điện thoại. Dữ liệu được hiển thị dưới dạng bảng, giúp dễ dàng theo dõi và tra cứu. Giao diện hỗ trợ thao tác nhanh chóng, thuận tiện cho việc quản lý nguồn cung cấp sách trong thư viện.

### Tác giả

**Mã tác giả:**

**Tên tác giả:**

Thêm
Sửa
Xóa

Mã tác giả	Tên tác giả
TG01	Nguyễn Nhật Ánh
TG02	Tô Hoài
TG03	Antoine de Saint-Exupéry
TG04	Gosho Aoyama
TG05	J.K. Rowling
TG06	Paulo Coelho
TG07	Ichiro Kishimi, Fumitake Koga
TG08	Phùng Quán
TG09	Nguyễn Ngọc Tư

Hình 3.23 Giao diện trang Tác giả

Trang Tác giả: Đây là giao diện dùng để quản lý thông tin các tác giả trong hệ thống thư viện. Người dùng có thể xem danh sách các tác giả hiện có, trong khi nhân viên hệ thống có thể thêm mới, chỉnh sửa hoặc xóa tác giả thông qua các trường nhập liệu như mã tác giả và tên tác giả. Dữ liệu được hiển thị dưới dạng bảng rõ ràng, hỗ trợ tra cứu và quản lý thuận tiện. Giao diện được thiết kế trực quan, giúp việc thao tác nhanh chóng và dễ dàng trong quá trình quản lý thông tin tác giả.

### Thẻ loại sách

**Mã thẻ loại:**

**Tên thẻ loại:**


Thêm
Sửa
Xóa

Mã thẻ loại	Tên thẻ loại
TL01	Văn học thiếu nhi
TL02	Truyện tranh
TL03	Văn học triết lý
TL04	Tâm lý học
TL05	Văn học hiện đại
TL06	Phát triển bản thân
TL07	Kỹ năng sống
TL08	Lịch sử

Hình 3.24 Giao diện trang Thẻ loại sách



Trang Thẻ loại sách: Đây là giao diện dùng để quản lý thông tin các thẻ loại sách trong hệ thống thư viện. Người dùng có thể xem danh sách thẻ loại hiện có, nhân viên hệ thống có thể thêm mới, chỉnh sửa hoặc xóa thẻ loại thông qua các trường nhập như mã thẻ loại và tên thẻ loại. Danh sách được hiển thị dưới dạng bảng, giúp tra cứu nhanh và thuận tiện trong việc phân loại sách. Giao diện đơn giản, rõ ràng, hỗ trợ hiệu quả cho việc quản lý và sắp xếp kho sách theo từng chủ đề cụ thể.



## Phiếu Mượn Sách

**Mã phiếu mượn:**

**Mã độc giả:**

**Mã sách:**

**Tiền đặt cọc (VNĐ):**

**Ngày mượn:**

**Ngày hẹn trả:**

Thêm phiếu mượn

Thống kê Phiếu Mượn

Đang mượn	Đã trả sách	Quá hạn chưa trả
1	0	8

Danh sách phiếu mượn

Mã phiếu	Mã độc giả	Mã sách	Tiền đặt cọc	Ngày mượn	Ngày hẹn trả	Thao tác
M04	DG04	S007	50.000 VNĐ	1/7/2025	18/7/2025	<div>SửaXóa</div>
M05	DG01	S004	55.000 VNĐ	30/6/2025	25/7/2025	<div>SửaXóa</div>

Hình 3.25 Giao diện Phiếu mượn sách

Phiếu Mượn Sách: Đây là giao diện cho phép quản lý các giao dịch mượn sách trong hệ thống thư viện. Người dùng có thể tạo mới phiếu mượn bằng cách nhập thông tin như mã phiếu mượn, mã độc giả, mã sách, số tiền đặt cọc, ngày mượn và ngày hẹn trả. Dữ liệu phiếu mượn được hiển thị dưới dạng bảng giúp tra cứu dễ dàng, đồng thời nhân viên có thể quản lý sửa và xóa phiếu mượn.

Giao diện còn tích hợp mục thống kê phiếu mượn, hiển thị số lượng phiếu đang mượn, đã trả và số phiếu quá hạn chưa trả, giúp quản lý tình hình mượn sách hiệu quả. Tất cả được thiết kế rõ ràng, trực quan, thuận tiện cho nhân viên hệ thống trong quá trình theo dõi và xử lý mượn – trả sách.

Phiếu Trả Sách

Mã trả:  
VD: T01

Mã độc giả:  
VD: DG01

Tình trạng sách trả:  
-- Chọn --

Mã phạt:  
VD: P001 hoặc để trống nếu không phạt

Ngày trả:  
dd/mm/yyyy

Xác nhận trả sách

Danh sách trả sách

Mã trả	Mã độc giả	Tình trạng	Mã phạt	Ngày trả	Hành động
T01	DG01	Bình thường	-	5/7/2025	<div>Sửa</div> <div>Xóa</div>
T02	DG02	Bị hỏng	P01	2/7/2025	<div>Sửa</div> <div>Xóa</div>
T03	DG03	Bị hỏng	P02	10/2/2025	<div>Sửa</div> <div>Xóa</div>
T04	DG06	Bị hỏng	P03	1/6/2025	<div>Sửa</div> <div>Xóa</div>
T05	DG01	Bình thường	-	11/7/2025	<div>Sửa</div> <div>Xóa</div>

Hình 3.26 Giao diện Phiếu trả sách

Phiếu Trả Sách: Đây là giao diện giúp quản lý quá trình trả sách của độc giả trong hệ thống thư viện. Người dùng có thể xác nhận trả sách bằng cách nhập các thông tin như mã trả, mã độc giả, tình trạng sách trả, mã phạt (nếu có) và ngày trả. Bảng danh sách bên dưới hiển thị các phiếu trả đã được ghi nhận, bao gồm tình trạng sách, mức phạt và ngày trả tương ứng.

Chức năng sửa và xóa được tích hợp để dễ dàng điều chỉnh dữ liệu khi cần. Giao diện được thiết kế rõ ràng, giúp quản lý và theo dõi việc trả sách nhanh chóng, chính xác, đồng thời đảm bảo xử lý các trường hợp hư hỏng hay quá hạn một cách hiệu quả.

Quản lý Phí phạt

Mã phạt:  
VD: P01

Mã trả:  
VD: T01

Tiền phạt:  
VD: 50000

Lí do phạt:  
VD: Trả sách trễ

Thêm

Sửa

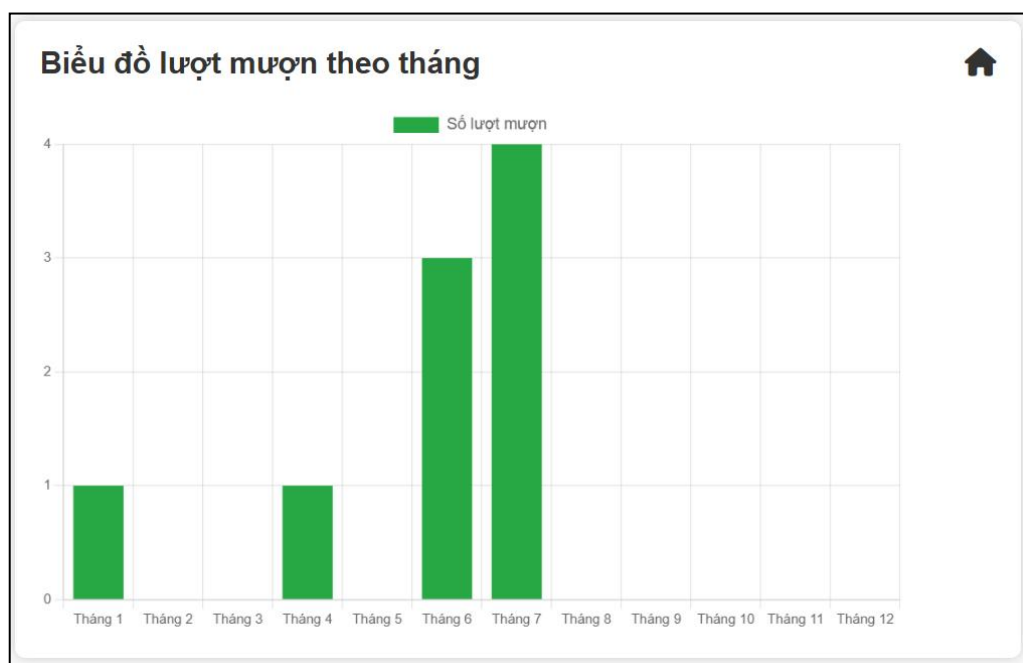
Xóa

Mã phạt	Mã trả	Tiền phạt	Lí do phạt
P01	T02	50000	Bìa bị gấp
P02	T03	20000	Rách góc
P03	T04	25000	Trả sách trễ

Hình 3.27 Giao diện trang Phí phạt

Trang Quản lý Phí phạt: Đây là giao diện hỗ trợ quản lý các khoản phạt phát sinh khi độc giả trả sách không đúng quy định. Người dùng có thể xem và nhân viên có thể nhập các thông tin như mã phạt, mã trả, số tiền phạt và lý do phạt (ví dụ: sách bị rách, gấp mép, trả trễ hạn...). Dữ liệu được trình bày dưới dạng bảng, hiển thị rõ ràng từng khoản phạt tương ứng với phiếu trả.


Trang cũng cung cấp các chức năng thêm, sửa và xóa phí phạt, giúp nhân viên hệ thống cập nhật nhanh chóng và chính xác. Giao diện được thiết kế trực quan, hỗ trợ theo dõi và xử lý vi phạm một cách minh bạch và hiệu quả.



Hình 3.28 Giao diện Biểu đồ lượt mượn theo tháng

Trang Biểu đồ lượt mượn theo tháng: Đây là giao diện trực quan giúp thống kê và hiển thị số lượt mượn sách theo từng tháng trong năm dưới dạng biểu đồ cột. Mỗi cột tương ứng với một tháng và thể hiện số lượng sách được mượn trong khoảng thời gian đó.

Giao diện này giúp người quản lý hệ thống dễ dàng theo dõi xu hướng mượn sách, từ đó có thể đưa ra các chiến lược bổ sung sách, tổ chức sự kiện khuyến đọc vào các thời điểm phù hợp. Việc sử dụng biểu đồ giúp việc phân tích dữ liệu trở nên sinh động, rõ ràng và hiệu quả hơn.



### Quản lý Độc giả

Mã độc giả:

Ngày sinh:

Địa chỉ:

Email:

Họ tên:

Giới tính:

SĐT:


Thêm
Sửa
Xóa

Mã độc giả	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT	Email
DG01	Trần Văn An	2018-09-02	Nam	123 Nguyễn Trãi, Hà Nội	0987654321	antran@gmail.com
DG02	Nguyễn Thị Bình	2005-06-14	Nữ	456 Lê Lợi, TP.HCM	0912345678	binhnguyen@yahoo.com
DG03	Lê Hoàng	1998-09-23	Nam	789 Lạc Long Quân, Đà Nẵng	0909090909	lehoang@gmail.com
DG04	Phạm Thu Trang	2001-07-05	Nữ	321 Cách Mạng, Cần Thơ	0978123456	trangpham@gmail.com

*Hình 3.29 Giao diện Quản lý Độc giả*

Trang Quản lý Độc giả: Giao diện này cho phép người quản lý và nhân viên hệ thống thực hiện các thao tác thêm, sửa, xóa thông tin độc giả một cách dễ dàng. Nhân viên hệ thống có thể nhập các thông tin như mã độc giả, họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại và email.

Bảng dữ liệu bên dưới hiển thị danh sách độc giả đã đăng ký, giúp quản lý thông tin cá nhân một cách trực quan và khoa học.



### Quản lý Nhân viên

Mã nhân viên:

Địa chỉ:

Họ tên:

SĐT:

Thêm
Sửa
Xóa

Mã nhân viên	Họ tên	Địa chỉ	SĐT
NV01	Lê Thị Lan	1 Nguyễn Huệ, Hà Nội	0901234567
NV02	Nguyễn Văn Nam	22 Trần Phú, TP.HCM	0912345678
NV03	Phạm Thị Hòa	15 Quang Trung, Đà Nẵng	0923456789
NV04	Hồ Văn Hưng	88 Lê Lợi, Huế	0934567890
NV05	Trịnh Thị Mai	66 Trần Hưng Đạo, Cần Thơ	0945678901

*Hình 3.30 Giao diện Quản lý Nhân viên*

Trang Quản lý Nhân viên: Trang này hỗ trợ việc quản lý danh sách nhân viên trong hệ thống thư viện. Quản lý hệ thống có thể dễ dàng thêm, sửa, xóa thông tin

nhân viên thông qua các trường nhập liệu như: mã nhân viên, họ tên, địa chỉ và số điện thoại.

Danh sách nhân viên được trình bày dưới dạng bảng, giúp theo dõi, tìm kiếm và quản lý thông tin nhân sự một cách trực quan. Giao diện thân thiện, dễ sử dụng, hỗ trợ hiệu quả trong công tác phân công và tổ chức hoạt động thư viện.

## CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG

### 4.1 Danh sách công nghệ sử dụng

#### Ngôn ngữ lập trình

JavaScript: Dùng cho backend.

HTML, CSS, JavaScript: Dùng cho frontend.

MongoDB Query Language (MQL): Truy vấn dữ liệu trong MongoDB.

#### Framework/ Thư viện

##### *Frontend:*

Chart.js: Dùng để vẽ biểu đồ cột.

Font Awesome: Dùng để hiển thị icon fa-home.

Fetch API: Dùng để gửi yêu cầu HTTP đến server API (fetch(...)).

LocalStorage API: Dùng để lấy thông tin vai trò người dùng (localStorage.getItem(role)).

##### *Backend:*

Mongoose: Kết nối MongoDB, quản lý schema/model.

Process: Quản lý biến môi trường, thoát tiến trình.

Nodemailer: Gửi email nhắc nhở người dùng (sách sắp hết hạn, sách quá hạn).

Bcryptjs: Mã hóa mật khẩu khi đăng ký, kiểm tra mật khẩu khi đăng nhập.

JWT (json web token): Xác thực người dùng qua JWT.

Dotenv: Load biến môi trường từ .env.

Cors: Cho phép frontend truy cập tài nguyên backend.

Express: Xây dựng REST API.

#### Cơ sở dữ liệu

MongoDB: NoSQL Database đóng vai trò là nơi lưu trữ toàn bộ dữ liệu động của hệ thống bao gồm tài khoản người dùng, sách, phiếu mượn, thông báo, thống kê...

#### Kiểm thử

Postman: kiểm thử API

Swagger: tạo tài liệu API tự động và hiển thị giao diện để test API ngay trên trình duyệt.

GitHub Actions: định nghĩa các workflow tự động hoá các hoạt động trong phát triển phần mềm.

**Triển khai:** Docker dùng để đóng gói và triển khai ứng dụng

## 4.2 Quy trình CI/CD với GitHub Actions

### Kiểm thử Backend(CI)

```
name: Node.js CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  test:
    runs-on: ubuntu-latest

    services:
      mongo:
        image: mongo
        ports: ['27017:27017']
        options: >-
          --health-cmd "mongosh --eval 'db.stats()'"
          --health-interval 10s
          --health-timeout 5s
          --health-retries 5

    steps:
      - uses: actions/checkout@v3
      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: 18
      - name: Install dependencies
        run: |
          cd backend/api
          npm install
      - name: Run tests
        run: |
          cd backend/api
          npm test
```

*Quy trình:*

Kích hoạt khi có push hoặc pull request với các thay đổi trong nhánh main.

Khởi chạy môi trường Ubuntu và container MongoDB.

Thiết lập môi trường Node.js phiên bản 18.

Cài đặt phụ thuộc cho backend tại thư mục backend/api bằng lệnh npm install.

Thực thi kiểm thử tự động với lệnh npm test.

## 4.3 Cấu hình Docker và quy trình triển khai

### Cấu hình

*Dockerfile cho Backend:* Chứa các lệnh để xây dựng image cho ứng dụng Node.js backend. Nó sẽ bao gồm việc cài đặt Node.js runtime, sao chép mã nguồn của

backend vào image, cài đặt các gói phụ thuộc (npm install), và định nghĩa lệnh để khởi chạy ứng dụng Express.

*Dockerfile cho Frontend:* Chứa các lệnh để xây dựng image cho ứng dụng frontend tĩnh (HTML/CSS/JS). Dùng Nginx để phục vụ các file frontend đã được build.

*Docker Compose:* Sử dụng file docker-compose.yml để định nghĩa và chạy đa container ứng dụng, gồm:

mongo: Dịch vụ cơ sở dữ liệu MongoDB (port 27017), gắn volume mongodb\_data để lưu trữ dữ liệu vĩnh viễn.

backend: Chạy Node.js API (port 5000), đọc biến môi trường để kết nối MongoDB, phụ thuộc vào dịch vụ mongo.

frontend: Chạy ứng dụng giao diện người dùng (port 3000, ánh xạ vào cổng 80 trong container), phụ thuộc vào backend.

### **Quy trình triển khai**

*Hệ thống sẽ được triển khai với:*

Backend API chạy tại: http://localhost:5000

Frontend chạy tại: http://localhost:3000

MongoDB: mongodb://mongo:27017/db



## CHƯƠNG 5: QUẢN LÝ DỰ ÁN

### 5.1 Sử dụng Jira trong quản lý tiến độ

Dự án được quản lý một cách hiệu quả bằng Jira, một công cụ quản lý dự án Agile hàng đầu, giúp nhóm duy trì sự tổ chức, minh bạch và theo dõi tiến độ công việc một cách chặt chẽ.

**Lập kế hoạch Sprint:** Các sprint được tạo và lên kế hoạch rõ ràng với các mục tiêu cụ thể và danh sách các nhiệm vụ (tasks) chi tiết. Mỗi sprint kéo dài một khoảng thời gian cố định (1 hoặc 2 tuần) để nhóm tập trung hoàn thành một tập hợp các tính năng.

**Phân chia nhiệm vụ:** Mỗi task trong Jira được tạo với mô tả rõ ràng, tiêu chí hoàn thành và được gán cho từng thành viên trong nhóm dựa trên kỹ năng và khối lượng công việc.

**Theo dõi tiến độ:** Nhóm sử dụng Scrum Board của Jira để trực quan hóa tiến độ của từng sprint, di chuyển các task qua các cột “To Do”, “In Progress”, “Done”. Burndown Chart cũng được sử dụng để theo dõi lượng công việc còn lại và dự đoán khả năng hoàn thành sprint đúng hạn. Các cuộc họp Daily Scrum được tổ chức thường xuyên để cập nhật tiến độ và giải quyết các trở ngại.

Giao diện Scrum Board trong Jira, hiển thị các nhiệm vụ được phân công và trạng thái tiến độ của từng sprint[1].

### 5.2 Phân công nhiệm vụ các thành viên nhóm

Việc phân công nhiệm vụ rõ ràng giúp tối ưu hóa hiệu suất làm việc nhóm và đảm bảo mọi khía cạnh của dự án đều được bao quát. Hệ thống được hoàn thành trong 5 Sprint với mỗi nhiệm vụ phân công cho từng thành viên cụ thể như sau:

#### ***Sprint 1 (12/5 - 19/5):***

**Mục tiêu:** Xây dựng nền tảng xác thực người dùng và chức năng quản lý hồ sơ cơ bản.

**Công việc chính:** Tạo form Đăng ký thành viên Thư viện, tạo form Đăng nhập, tạo API xác thực JWT (đăng ký / đăng nhập / xác thực), thiết kế giao diện trang chủ và thêm chức năng Đăng xuất, triển khai middleware bảo vệ các route riêng tư và tạo trang Hồ sơ thành viên và cập nhật thông tin.

<input checked="" type="checkbox"/> SCRUM-6	Tạo trang Hồ sơ thành viên và cập nhật thông tin	HK Huong Kim	HK Huong Kim		SCRUM Sprint 1
<input checked="" type="checkbox"/> SCRUM-5	Middleware bảo vệ route riêng tư	L lehaduy2004	HK Huong Kim		SCRUM Sprint 1
<input checked="" type="checkbox"/> SCRUM-4	Thiết kế giao diện trang chủ và thêm chức năng đăng xuất	HK Huong Kim	HK Huong Kim		SCRUM Sprint 1
<input checked="" type="checkbox"/> SCRUM-3	Tạo API xác thực JWT (đăng ký/đăng nhập/xác thực)	L lehaduy2004	HK Huong Kim		SCRUM Sprint 1
<input checked="" type="checkbox"/> SCRUM-2	Tạo form Đăng nhập	HK Huong Kim	HK Huong Kim		SCRUM Sprint 1
<input checked="" type="checkbox"/> SCRUM-1	Tạo form Đăng ký thành viên	HK Huong Kim	L lehaduy2004		SCRUM Sprint 1

Hình 5.1 Phân công nhiệm vụ các thành viên ở Sprint 1 trên Jira

### Sprint 2 (19/5 - 2/6):

Mục tiêu: Hoàn thiện chức năng quản lý sách và hiển thị danh sách.

Công việc chính: Thiết kế giao diện trang Danh sách Sách, tạo API CRUD cho sách (create, read, update, delete), hiển thị danh sách sách theo trạng thái (Available / Borrowed), thêm chức năng Thêm, Sửa, Xóa sách.

Work	Assignee	Reporter	Status	Sprint
<input checked="" type="checkbox"/> SCRUM-10	Thêm chức năng thêm, sửa, xóa sách	HK Huong Kim		SCRUM Sprint 2
<input checked="" type="checkbox"/> SCRUM-9	Hiển thị danh sách theo trạng thái (Available / Borrowed)	L lehaduy2004		SCRUM Sprint 2
<input checked="" type="checkbox"/> SCRUM-8	Tạo API CRUD cho task (create, read, update, delete)	L lehaduy2004		SCRUM Sprint 2
<input checked="" type="checkbox"/> SCRUM-7	Thiết kế giao diện trang Danh sách Sách	HK Huong Kim		SCRUM Sprint 2

Hình 5.2 Phân công nhiệm vụ các thành viên ở Sprint 2 trên Jira

### Sprint 3 (2/6 - 16/6):

Mục tiêu: Phát triển hệ thống thông báo và báo cáo cơ bản.

Công việc chính: Tạo API gửi mail nhắc nhở quá hạn trả sách (notification system), hiển thị số lượng phiếu mượn theo trạng thái (open/closed/overdue), tạo biểu đồ lượt mượn theo tháng, thiết lập chức năng nhắc thành viên sắp tới hạn trả.

Work	Assignee	Reporter	Status	Sprint
<input checked="" type="checkbox"/> SCRUM-22	Thiết lập nhắc thành viên sắp tới hạn trả	L lehaduy2004		SCRUM Sprint 3
<input checked="" type="checkbox"/> SCRUM-21	Tạo biểu đồ lượt mượn theo tháng	L lehaduy2004		SCRUM Sprint 3
<input checked="" type="checkbox"/> SCRUM-20	Hiển thị số lượng phiếu mượn theo trạng thái (open/closed/overdue)	HK Huong Kim		SCRUM Sprint 3
<input checked="" type="checkbox"/> SCRUM-19	Tạo API gửi mail nhắc nhở quá hạn trả sách (notification system)	L lehaduy2004		SCRUM Sprint 3

Hình 5.3 Phân công nhiệm vụ các thành viên ở Sprint 3 trên Jira

### Sprint 4 (16/6 - 30/6):

Mục tiêu: Cải thiện trải nghiệm người dùng và thiết kế responsive.

Công việc chính: Thiết kế layout tổng thể của Thư viện (Header, Sidebar: Sách, Báo cáo, Phiếu mượn...), đảm bảo responsive cho điện thoại và máy tính bảng, cải thiện trải nghiệm người dùng.

Work	Assignee	Reporter	Status	Sprint
<input checked="" type="checkbox"/> SCRUM-25 Cải thiện trải nghiệm người dùng (loading, toast mượn-trả, tìm kiếm nhanh)	 lehaduy2004	 Hương Kim	<span>DONE</span>	SCRUM Sprint 4
<input checked="" type="checkbox"/> SCRUM-24 Responsive cho điện thoại và máy tính bảng	 lehaduy2004	 Hương Kim	<span>DONE</span>	SCRUM Sprint 4
<input checked="" type="checkbox"/> SCRUM-23 Thiết kế layout Thư viện (Header, Sidebar: Sách, Báo cáo, Phiếu mượn...)	 Hương Kim	 Hương Kim	<span>DONE</span>	SCRUM Sprint 4

Hình 5.4 Phân công nhiệm vụ các thành viên ở Sprint 4 trên Jira

**Sprint 5 (30/6 - 14/7):**

Mục tiêu: Hoàn thiện kiểm thử và triển khai ứng dụng.

Công việc chính: Viết unit test toàn diện cho backend, kiểm thử chức năng frontend (manual và automation), triển khai backend lên Docker, triển khai frontend lên Docker, kiểm thử hệ thống tổng thể (system testing).

Work	Assignee	Reporter	Status	Sprint
<input checked="" type="checkbox"/> SCRUM-30 Kiểm thử hệ thống	 lehaduy2004	 Hương Kim	<span>DONE</span>	SCRUM Sprint 5
<input checked="" type="checkbox"/> SCRUM-29 Triển khai frontend lên Docker	 Hương Kim	 Hương Kim	<span>DONE</span>	SCRUM Sprint 5
<input checked="" type="checkbox"/> SCRUM-28 Triển khai backend lên Docker	 Hương Kim	 Hương Kim	<span>DONE</span>	SCRUM Sprint 5
<input checked="" type="checkbox"/> SCRUM-27 Kiểm thử chức năng frontend	 Hương Kim	 Hương Kim	<span>DONE</span>	SCRUM Sprint 5
<input checked="" type="checkbox"/> SCRUM-26 Viết unit test cho backend	 lehaduy2004	 Hương Kim	<span>DONE</span>	SCRUM Sprint 5

Hình 5.5 Phân công nhiệm vụ các thành viên ở Sprint 4 trên Jira

Bảng 5.1 Chi tiết và độ ưu tiên công việc

Sprint	Mô tả	Story Points	Độ ưu tiên
Sprint 1	Tạo form Đăng ký thành viên Thư viện	3	Trung bình
Sprint 1	Tạo form Đăng nhập	2	Trung bình
Sprint 1	Tạo API xác thực JWT (đăng ký/đăng nhập/xác thực)	4	Cao
Sprint 1	Thiết kế giao diện trang chủ và thêm chức năng đăng xuất	1	Cao
Sprint 1	Middleware bảo vệ route riêng tư	3	Cao
Sprint 1	Tạo trang Hồ sơ thành viên và cập nhật thông tin	3	Trung bình
Sprint 2	Thiết kế giao diện trang Danh sách Sách	4	Trung bình

Sprint 2	Tạo API CRUD cho task (create, read, update, delete)	2	Cao
Sprint 2	Hiển thị danh sách theo trạng thái (Available / Borrowed)	2	Trung bình
Sprint 2	Thêm chức năng thêm, sửa, xóa sách	3	Trung bình
Sprint 3	Tạo API gửi mail nhắc nhở quá hạn trả sách (notification system)	4	Cao
Sprint 3	Hiển thị số lượng phiếu mượn theo trạng thái (open/closed/overdue)	3	Trung bình
Sprint 3	Tạo biểu đồ lượt mượn theo tháng	4	Cao
Sprint 3	Thiết lập nhắc thành viên sắp tới hạn trả	2	Cao
Sprint 4	Thiết kế layout Thư viện (Header, Sidebar: Sách, Báo cáo, Phiếu mượn...)	3	Cao
Sprint 4	Responsive cho điện thoại và máy tính bảng	5	Trung bình
Sprint 4	Cải thiện trải nghiệm người dùng (loading, toast mượn-trả, tìm kiếm nhanh)	4	Trung bình
Sprint 5	Viết unit test cho backend	2	Cao
Sprint 5	Kiểm thử chức năng frontend	3	Cao
Sprint 5	Triển khai backend lên Docker	2	Trung bình
Sprint 5	Triển khai frontend lên Docker	2	Cao
Sprint 5	Kiểm thử hệ thống	3	Trung bình

## CHƯƠNG 6: KIỂM THỬ

### 6.1 Chiến lược kiểm thử và công cụ sử dụng

Để đảm bảo chất lượng và độ tin cậy của Hệ thống Quản lý Mượn/Trả Sách, một chiến lược kiểm thử toàn diện đã được áp dụng, bao gồm nhiều cấp độ kiểm thử và sử dụng các công cụ chuyên biệt.

#### *Kiểm thử đơn vị (Unit Testing):*

Mục tiêu: Kiểm tra từng thành phần nhỏ nhất và độc lập của mã nguồn để đảm bảo rằng mỗi đơn vị hoạt động đúng theo đặc tả.

Công cụ sử dụng: Jest cho JavaScript backend, Jest được lựa chọn vì tính dễ sử dụng, tốc độ nhanh và khả năng tạo báo cáo độ bao phủ mã.

Phạm vi: Tập trung vào các logic nghiệp vụ cốt lõi trong backend (controllers, models, middleware,..) để đảm bảo tính đúng đắn của dữ liệu và quy trình xử lý.

#### *Kiểm thử API (API Testing):*

Mục tiêu: Xác minh các endpoint của backend API hoạt động đúng cách, trả về dữ liệu chính xác, xử lý các trường hợp thành công và thất bại một cách phù hợp. Kiểm tra tính bảo mật, hiệu suất và độ tin cậy của API.

Công cụ sử dụng: Postman. Postman là một công cụ mạnh mẽ, cung cấp giao diện thân thiện để gửi các yêu cầu HTTP (GET, POST, PUT, DELETE), kiểm tra phản hồi, quản lý biến môi trường, và tạo các bộ sưu tập kiểm thử.

Quy trình: Tạo các bộ sưu tập yêu cầu (Postman Collections) cho từng nhóm API (Auth, Book, Borrow/Return, Notification,...). Mỗi request được cấu hình với các tham số, body request và headers cần thiết (bao gồm cả JWT cho các API được bảo vệ). Các bài kiểm thử kịch bản (test scripts) được viết trong Postman để tự động xác minh mã trạng thái HTTP, cấu trúc phản hồi JSON và giá trị dữ liệu[2].

#### *Kiểm thử tích hợp (Integration Testing):*

Mục tiêu: Kiểm tra sự tương tác và luồng dữ liệu giữa các module hoặc thành phần khác nhau của hệ thống (Frontend giao tiếp với Backend, Backend tương tác với Cơ sở dữ liệu). Mục đích là phát hiện lỗi do sự không tương thích giữa các thành phần.

Công cụ sử dụng: Có thể sử dụng các framework test tích hợp của ngôn ngữ lập trình (Node.js API), kết hợp với các kịch bản kiểm thử thủ công và việc sử dụng lại các request Postman để kiểm tra chuỗi các hành động liên quan.

### *Kiểm thử hệ thống (System Testing):*

Mục tiêu: Kiểm tra toàn bộ hệ thống như một thể thống nhất để đảm bảo nó đáp ứng tất cả các yêu cầu chức năng và phi chức năng đã đề ra, hoạt động chính xác trong môi trường thực tế.

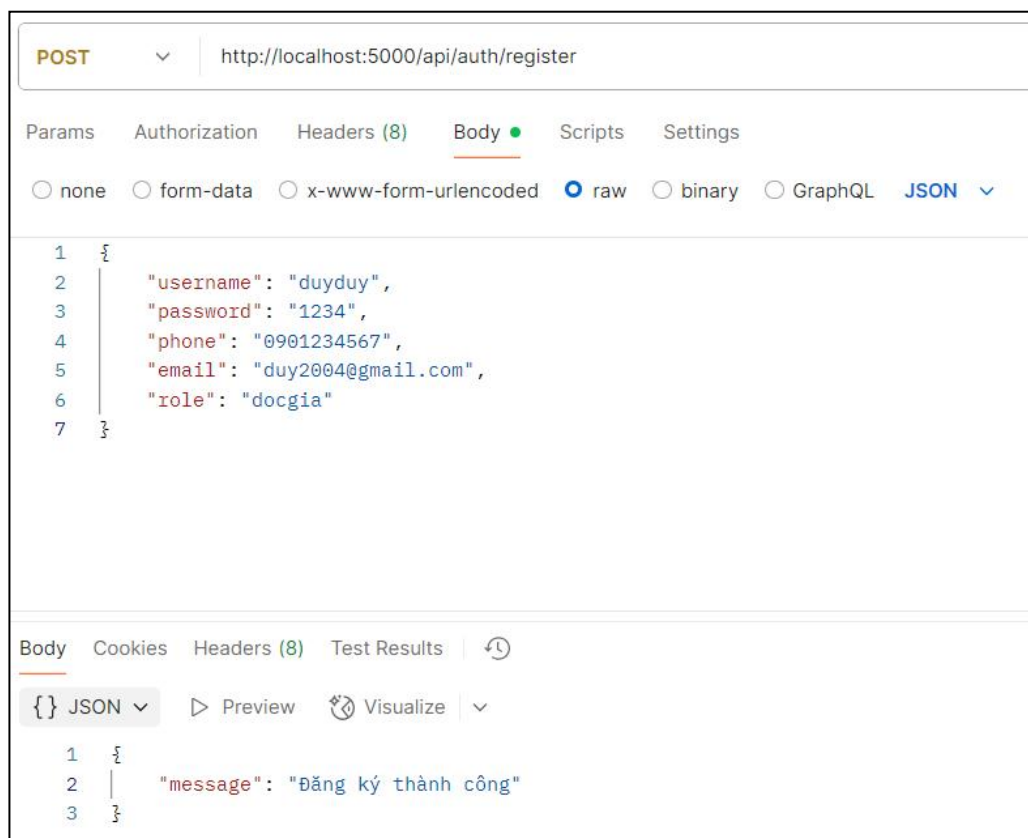
Quy trình: Mô phỏng các kịch bản người dùng cuối một cách toàn diện, từ việc đăng ký, mượn sách, trả sách, đến việc nhận thông báo và xem báo cáo. Kiểm tra tính tương thích trên các trình duyệt và thiết bị khác nhau.

## 6.2 Kết quả kiểm thử API

### API Đăng ký thành công

Mô tả: Đảm bảo người dùng mới có thể tạo tài khoản thành công và nhận được phản hồi xác nhận từ server.

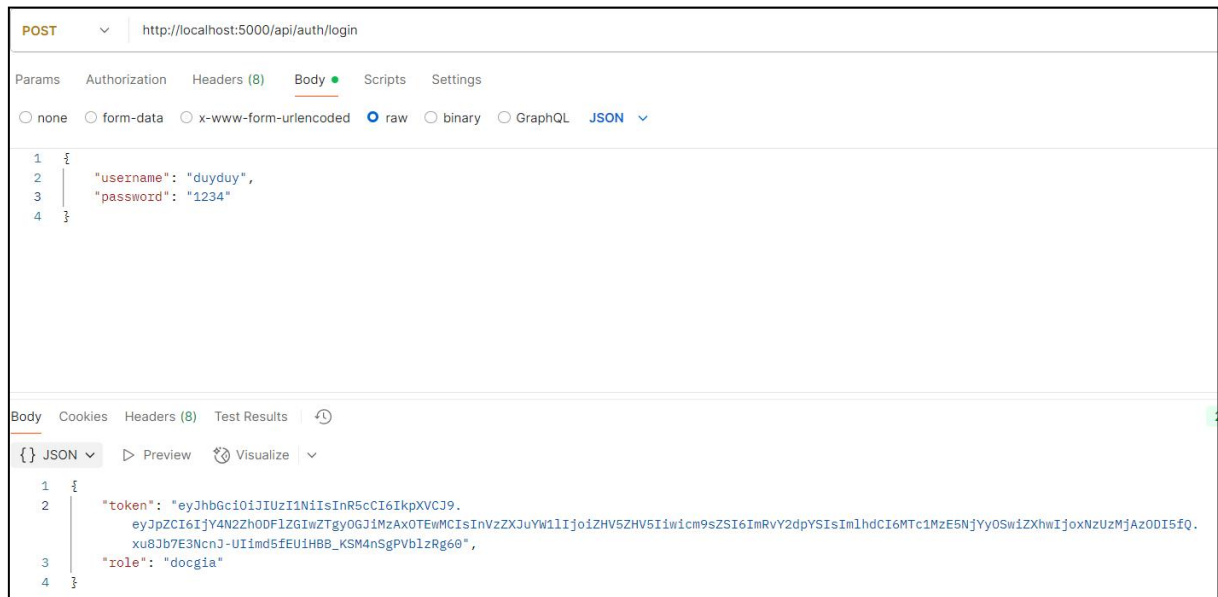
Mục tiêu: Kiểm tra tính năng đăng ký hoạt động đúng, đảm bảo người dùng nhận được thông báo rõ ràng khi đăng ký thành công, xác minh rằng tài khoản được lưu vào cơ sở dữ liệu với vai trò đúng.



Hình 6.1 Postman kiểm tra Đăng ký

### API Đăng nhập thành công

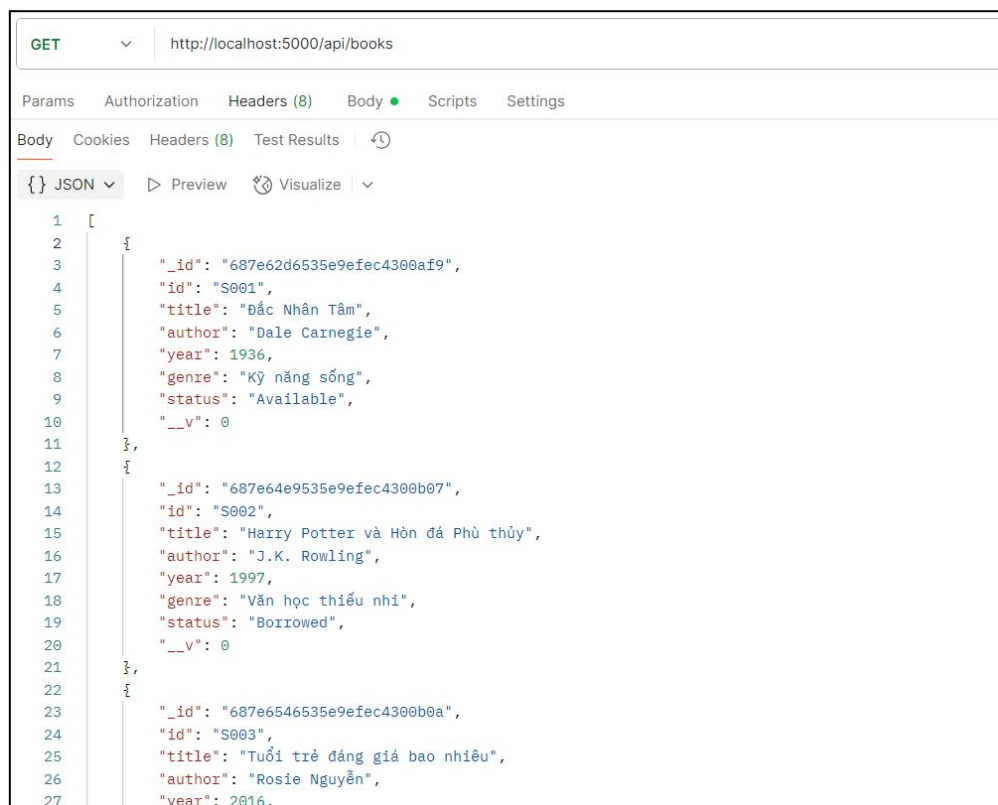
Mô tả: Xác minh rằng người dùng có thể đăng nhập với thông tin hợp lệ và nhận được JWT để xác thực các yêu cầu tiếp theo.



Hình 6.2 Postman kiểm tra Đăng nhập

### API Lấy danh sách sách

Mô tả: Đảm bảo hệ thống trả về danh sách sách chính xác, hỗ trợ lọc theo các tiêu chí đã định.

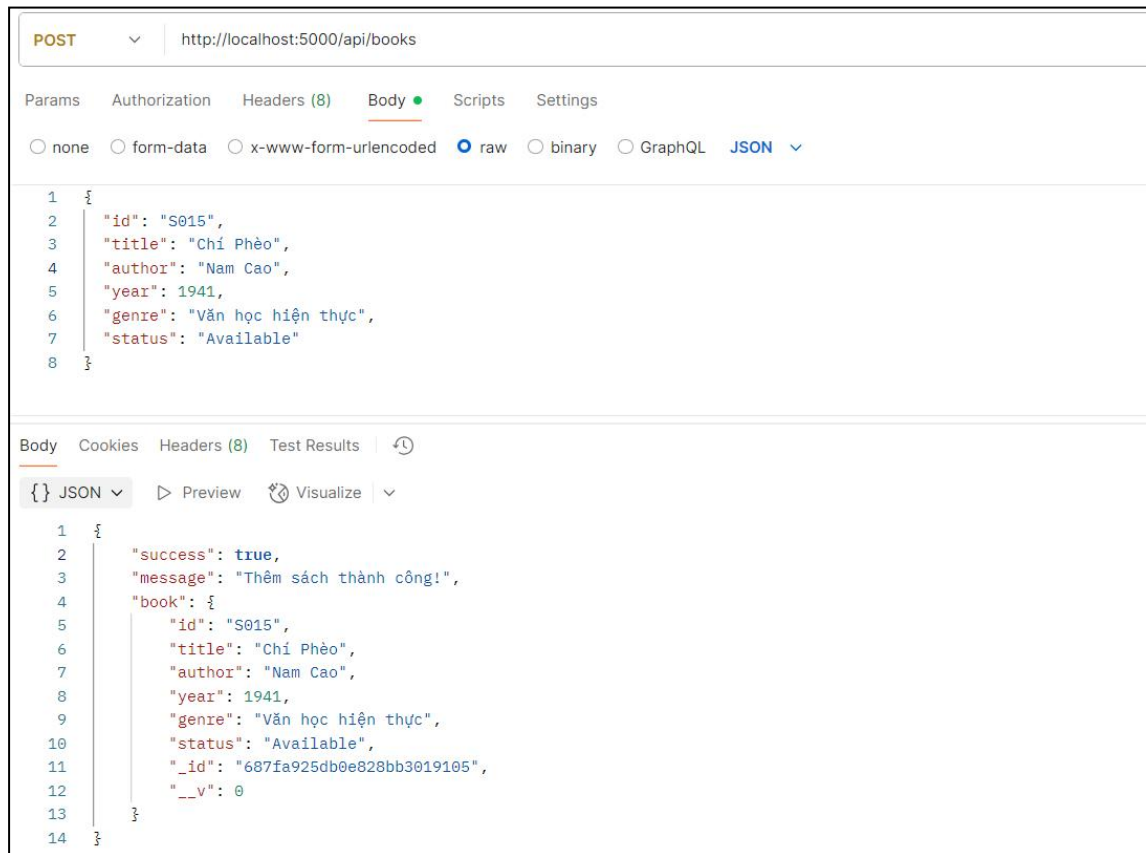


Hình 6.3 Postman lấy Danh sách Sách



## API Thêm sách thành công

Mô tả: Xác minh rằng nhân viên hệ thống có thể thêm sách mới vào hệ thống một cách thành công và dữ liệu được lưu trữ chính xác.



Hình 6.4 Postman kiểm tra thêm sách



## CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN

### 7.1 Những khó khăn gặp phải

Quản lý mã nguồn trên GitHub: Khó khăn trong pull/merge, xung đột mã nguồn khi nhiều người cùng làm việc.

Thiết lập CI/CD với GitHub Actions: Viết các workflow YAML để tự động build, test, deploy còn nhiều lỗi do thiếu quyền hoặc cú pháp sai. Cần hiểu sâu GitHub Actions và Docker để thiết lập chuẩn.

Xử lý lỗi & debug API: Lỗi giữa frontend và backend (CORS, JWT, dữ liệu sai định dạng, lỗi DB) khó phát hiện nếu thiếu kinh nghiệm hoặc công cụ hỗ trợ. Tốn thời gian dò lỗi và phân tích log.

### 7.2 Bài học rút ra

Quản lý dự án hiệu quả: Dùng Jira, lập kế hoạch sprint chi tiết, họp daily giúp quản lý tiến độ và phân công rõ ràng.

CI/CD mang lại lợi ích lớn: Tự động kiểm thử & triển khai với GitHub Actions giúp phát hiện lỗi sớm, tăng tốc độ phát hành.

API rõ ràng, có tài liệu: Tuân thủ RESTful và dùng Swagger giúp giảm hiểu lầm, dễ tích hợp và kiểm thử.

### 7.3 Đề xuất cải tiến trong tương lai

Tìm kiếm toàn văn: Cho phép tra cứu sách nâng cao theo nhiều tiêu chí.

Thanh toán trực tuyến: Tích hợp VNPay, PayPal để đóng phạt tiện lợi.

Dashboard phân tích dữ liệu: Hiển thị các chỉ số mượn sách, thói quen người dùng.

Cải tiến UI/UX liên tục: Thu thập phản hồi, phân tích hành vi để tối ưu trải nghiệm.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] “Jira là gì? Tổng quan về Jira”, [<https://viblo.asia/p/jira-la-gi-tong-quan-ve-jira-gGJ590XpIX2>] [truy cập 20/05/2025].
- [2] “Postman là gì? Tìm hiểu thành phần, tính ứng dụng và cơ sở chức năng của Postman”, [<https://fptshop.com.vn/tin-tuc/danh-gia/postman-la-gi-168171>] [truy cập 20/05/2025].

## PHỤ LỤC

### **Yêu cầu hệ thống**

Để có thể chạy được hệ thống quản lý công việc, máy tính cần đảm bảo các yêu cầu sau:

Đã cài đặt Node.js phiên bản 14 trở lên.

Có trình quản lý gói npm (đi kèm với Node.js).

Đã cài đặt MongoDB (nếu chạy cơ sở dữ liệu cục bộ).

Đã cài đặt VS code và Docker.

Trình duyệt web hiện đại như Google Chrome hoặc Firefox.

Công cụ Git để clone mã nguồn từ GitHub

### **Clone mã nguồn từ GitHub**

Người dùng cần mở terminal hoặc Git Bash và thực hiện lệnh:

```
git clone https://github.com/LeHaDuy2k4/DoAnKetThucMon\_CNPM.git
```

```
cd DoAnKetThucMon_CNPM
```

### **Cài đặt và cấu hình backend**

Truy cập vào thư mục backend và tiến hành cài đặt các thư viện cần thiết:

```
cd backend
```

```
npm install
```

### **Khởi chạy ứng dụng bằng Docker**

Người dùng có thể sử dụng Docker để chạy toàn bộ hệ thống. Trong thư mục gốc của dự án, sử dụng tệp docker-compose.yml có sẵn và chạy lệnh:

```
docker-compose up --build
```

Truy cập và sử dụng ứng dụng

Backend chạy tại địa chỉ: <http://localhost:5000>

Frontend (giao diện người dùng) chạy tại địa chỉ: <http://localhost:3000>

Tại giao diện người dùng, người dùng có thể thực hiện các chức năng:

Đăng ký/ Đăng nhập tài khoản.

Xem Danh sách Sách, nhà xuất bản, tác giả, thể loại, phí phạt.

Thêm phiếu mượn sách, trả sách.

Link GitHub: [https://github.com/LeHaDuy2k4/DoAnKetThucMon\\_CNPM.git](https://github.com/LeHaDuy2k4/DoAnKetThucMon_CNPM.git)