

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP LỚN
PHÁT TRIỂN ỨNG DỤNG WEB
Nhóm học phần: CT449-03



ĐỀ TÀI: QUẢN LÝ MƯỢN SÁCH

Giảng viên hướng dẫn:
Ths. Nguyễn Minh Trung

Sinh viên thực hiện: Trương Hoàng Duy

MSSV: B2007173

Lớp: DI20Z6A1

Khóa: 46

Cần Thơ, tháng 25/4/2024

LỜI CẢM ƠN

Để hoàn thành được bài báo cáo này, em xin chân thành cảm ơn Ban Giám hiệu, các khoa, phòng và quý thầy, cô của trường Công nghệ thông tin và Truyền thông, những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy Ths. Nguyễn Minh Trung - người đã trực tiếp giảng dạy và hướng dẫn em thực hiện bài báo cáo này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

Trong quá trình thực hiện bài báo cáo này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý của thầy để bài báo cáo ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC SƠ ĐỒ.....	v
DANH MỤC BIỂU BẢNG.....	vi
DANH MỤC HÌNH ẢNH	vii
TÓM TẮT.....	viii
PHẦN GIỚI THIỆU.....	1
1. Đặt vấn đề	1
2. Mục tiêu đề tài.....	2
PHẦN NỘI DUNG.....	4
CHƯƠNG 1.....	4
ĐẶC TẢ YÊU CẦU.....	4
CHƯƠNG 2.....	5
CƠ SỞ LÝ THUYẾT	5
1. Tổng quan về MEVN STACK:	5
1.1 Khái niệm:	5
1.2 Ưu và nhược điểm của công nghệ MEVN STACK:.....	5
2. MongoDB:.....	7
3. Restful API:	7
4. Express.js:	8
5. Vue.js:.....	9
6. Node.js:.....	10
7. Tổng quan về Mongoose:.....	11
8. Các công cụ hỗ trợ	11
8.1 Visual Studio Code	11
8.2. Postman	12
CHƯƠNG 3.....	14
CÁC SƠ ĐỒ CỦA WEBSITE.....	14

1. Sơ đồ Use case.....	14
1.1 Sơ đồ Use case tổng quát của hệ thống	14
1.1.1 Đặc tả sơ đồ chức năng use case tổng quát.....	14
1.2 Sơ đồ use case quản lý sách	16
1.2.1 Đặc tả usecase quản lý sách	16
1.3 Sơ đồ use case quản lý nhà xuất bản.....	18
1.3.1 Đặc tả use case quản lý nhà xuất bản	18
1.4 Sơ đồ use case quản lý độc giả.....	20
1.4.1 Đặc tả use case quản lý độc giả.....	20
1.5 Sơ đồ use case quản lý mượn sách	22
1.5.1 Đặc tả use case quản lý mượn sách	22
1.6 Sơ đồ use case xem danh mục sách	23
1.6.1 Đặc tả use case xem danh mục sách.....	24
1.7 Sơ đồ use case đăng kí mượn sách	24
1.7.1 Đặc tả use case đăng kí mượn sách	25
1.8 Sơ đồ use case đăng kí trả sách	26
1.8.1 Đặc tả use case đăng kí trả sách	26
2. Cơ sở dữ liệu	27
2.1 Mô hình cơ sở dữ liệu CDM	27
2.2 Mô tả các lớp:.....	27
2.3 Mô tả Sach.....	28
2.4 Mô tả NhaXuatBan.....	28
2.5 Mô tả DocGia	29
2.6 Mô tả TheoDoiMuonSach.....	29
2.7 Mô tả NhanVien.....	30
CHƯƠNG 4.....	31
KẾT QUẢ ĐẠT ĐƯỢC	31
1. Giao diện Website ở phía độc giả:	31

1.1 Giao diện hiển thị danh mục các sách hiện có	31
1.2 Giao diện hiển thị form đăng kí mượn sách.....	31
1.3 Giao diện hiển thị form đăng kí trả sách.....	32
2. Giao diện Website ở phía người quản lý:	32
2.1 Giao diện hiển thị form thêm mới sách và chỉnh sửa sách	32
2.2 Giao diện hiển thị danh mục các sách hiện có	33
2.3 Giao diện hiển thị form đăng kí mượn sách.....	33
2.4 Giao diện hiển thị form đăng kí trả sách.....	33
2.5 Giao diện hiển thị danh sách các độc giả đang mượn	34
2.6 Giao diện hiển thị form thêm nhà xuất bản	34
2.7 Giao diện hiển thị form chỉnh sửa nhà xuất bản	34
2.8 Giao diện hiển thị danh sách các nhà xuất bản	35
2.9 Giao diện hiển thị form thêm độc giả	35
2.10 Giao diện hiển thị form chỉnh sửa độc giả.....	35
2.11 Giao diện hiển thị danh sách các độc giả.....	36
TÀI LIỆU THAM KHẢO.....	37

DANH MỤC SƠ ĐỒ

Sơ đồ 1. Sơ đồ use case tổng quát	14
Sơ đồ 2: Sơ đồ use case Quản lý sách	16
Sơ đồ 3: Sơ đồ use case quản lý nhà xuất bản	18
Sơ đồ 4: Sơ đồ use case quản lý độc giả.....	20
Sơ đồ 5: Sơ đồ use case quản lý mượn sách.....	22
Sơ đồ 6: Sơ đồ use case xem danh mục các sách	23
Sơ đồ 7: Sơ đồ use case đăng kí mượn sách.....	24
Sơ đồ 8: Sơ đồ use case đăng kí trả sách	26

DANH MỤC BIỂU BẢNG

Bảng 1: Đặc tả sơ đồ chức năng use case tổng quát	15
Bảng 2: Đặc tả chức năng quản lý sách	17
Bảng 3: Đặc tả chức năng quản lý nhà xuất bản.....	19
Bảng 4: Đặc tả chức năng quản lý độc giả	21
Bảng 5: Đặc tả chức năng quản lý mượn sách	23
Bảng 6: Đặc tả chức năng xem danh mục sách	24
Bảng 7: Đặc tả chức năng đăng kí mượn sách	25
Bảng 8: Đặc tả chức năng đăng kí trả sách.....	26
Bảng 9: Tên các bảng trong cơ sở dữ liệu CDM	27
Bảng 10: Mô tả danh mục các sách	28
Bảng 11: Mô tả danh sách các nhà xuất bản.....	28
Bảng 12: Mô tả danh sách các độc giả	29
Bảng 13: Mô tả danh sách các độc giả đang mượn sách	29
Bảng 14: Mô tả danh sách các độc nhân viên.....	30

DANH MỤC HÌNH ẢNH

Hình 1. MEVN STACK	5
Hình 2. MongoDB	7
Hình 3. Vue.js	9
Hình 4. Node.js	10
Hình 5: Visual Studio Code	11
Hình 6: Postman.....	12
Hình 7: Mô hình CDM	27
Hình 8: Danh mục các sách hiện có.....	31
Hình 9: Form đăng kí mượn sách	31
Hình 10: Form đăng kí trả sách	32
Hình 11: Form thêm mới sách	32
Hình 12: Form chỉnh sửa sách	32
Hình 13: Danh mục các sách hiện có.....	33
Hình 14: Form đăng kí mượn sách	33
Hình 15: Form đăng kí trả sách	33
Hình 16: Danh sách các độc giả đang mượn	34
Hình 17: Form thêm nhà xuất bản	34
Hình 18: Form chỉnh sửa nhà xuất bản.....	34
Hình 19: Danh sách các nhà xuất bản.....	35
Hình 20: Form thêm độc giả.....	35
Hình 21: Form chỉnh sửa độc giả	35
Hình 22: danh sách các độc giả	36

TÓM TẮT

Công nghệ thông tin đang trở thành trụ cột quan trọng của xã hội hiện đại, mang lại nhiều cơ hội và thách thức. Với sự phát triển nhanh chóng của Internet và nền kinh tế số, việc áp dụng công nghệ để quản lý và cung cấp dịch vụ ngày càng trở nên quan trọng. Trong lĩnh vực thư viện, việc quản lý sách và nguồn tài nguyên trở nên phức tạp hơn khi nhu cầu mượn sách và tìm kiếm thông tin trực tuyến tăng cao.

Với xu hướng này, việc xây dựng một "Hệ thống quản lý thư viện trực tuyến" trở thành một nhu cầu thiết yếu. Dự án này sử dụng công nghệ MEVN STACK (MongoDB, Express.js, Vue.js, Node.js) để xây dựng một ứng dụng web hiện đại, linh hoạt và dễ dàng mở rộng.

Mục tiêu của dự án "Hệ thống quản lý thư viện trực tuyến" là tạo ra một nền tảng giúp nhân viên thư viện có thể quản lý sách, nhà xuất bản và các thông tin liên quan một cách hiệu quả. Đồng thời, người dùng có thể truy cập vào hệ thống để tìm kiếm sách, xem thông tin chi tiết và mượn sách dễ dàng từ bất kỳ đâu thông qua Internet.

Với việc ứng dụng công nghệ MEVN STACK, dự án sẽ mang lại giao diện người dùng thân thiện, trải nghiệm mượt mà và đảm bảo an ninh thông tin cho tất cả người dùng. Hệ thống sẽ giúp việc quản lý thư viện trở nên nhanh chóng, chính xác và tiết kiệm thời gian, đồng thời tối ưu hóa quá trình mượn và trả sách cho độc giả.

Để đạt được kết quả mong muốn, dự án sẽ tiến hành nghiên cứu sâu rộng về các công nghệ và công cụ cần thiết, phát triển giao diện người dùng thân thiện và đơn giản, cũng như xây dựng các chức năng quản lý sách và người dùng một cách toàn diện và linh hoạt.

Nhờ vào sự kết hợp giữa các công nghệ MongoDB, Express.js, Vue.js và Node.js, dự án "Hệ thống quản lý thư viện trực tuyến" không chỉ đáp ứng được nhu cầu hiện tại mà còn có khả năng mở rộng và phát triển trong tương lai.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

Mô hình quản lý mượn sách trong thư viện không chỉ đơn thuần là việc lưu trữ và cung cấp sách cho độc giả mà còn góp phần quan trọng vào việc khuyến khích đọc sách và nâng cao kiến thức của cộng đồng. Tuy nhiên, việc quản lý sách và người mượn trong một thư viện truyền thống thường gặp phải nhiều thách thức như:

Hiệu suất và Hiệu quả: Trong một thư viện lớn, việc quản lý hàng ngàn cuốn sách và người mượn đồng thời đòi hỏi một hệ thống quản lý hiệu quả. Việc này không chỉ giúp tăng cường hiệu suất làm việc mà còn đảm bảo rằng mọi cuốn sách đều được quản lý một cách chính xác và khoa học.

Dễ dàng tiếp cận và Tìm kiếm: Đối với người mượn, việc tìm kiếm và mượn sách nhanh chóng và dễ dàng là rất quan trọng. Hệ thống quản lý mượn sách cần cung cấp một giao diện trực quan, giúp người dùng có thể tìm kiếm sách một cách dễ dàng và biết được trạng thái của sách đã mượn.

An toàn và Bảo mật: Việc bảo mật thông tin của độc giả và dữ liệu sách là ưu tiên hàng đầu. Hệ thống cần đảm bảo rằng thông tin cá nhân của độc giả và dữ liệu sách đều được bảo vệ một cách an toàn và bảo mật.

Với những thách thức và nhu cầu trên, việc xây dựng một "Hệ thống quản lý mượn sách trực tuyến" đã áp dụng công nghệ MEVN STACK (MongoDB, Express.js, Vue.js, Node.js) nhằm giải quyết các vấn đề trên, tạo ra một nền tảng quản lý mượn sách hiện đại, linh hoạt và dễ dàng mở rộng. Điều này giúp cải thiện trải nghiệm người dùng, nâng cao hiệu suất làm việc và đảm bảo an toàn thông tin cho tất cả người dùng trong thư viện.

2. Mục tiêu đề tài

- Mục tiêu tổng quát:

Xây dựng website quản lý mượn sách trực tuyến với các chức năng hoàn chỉnh, dễ sử dụng và thuận tiện cho cả người quản lý và người mượn sách, giúp độc giả có thể tìm và mượn để có được những quyển sách mà họ thật sự cảm thấy hứng thú.

- Mục tiêu cụ thể:

Hệ thống quản lý mượn sách gồm có 2 website: **Website độc giả** và **Website quản trị**.

 Website độc giả:

- Website có 3 phần chính:

1. Danh mục sách.
2. Đăng kí mượn sách.
3. Đăng kí trả sách

- Chức năng:

1. Danh mục sách: liệt kê tất cả các sách hiện có và độc giả có thể tìm kiếm sách ở phần danh mục này.
2. Đăng kí mượn sách: đây là phần mà độc giả có thể đăng kí để mượn những quyển sách mà họ đã chọn.
3. Đăng kí trả sách: Đây là phần mà độc giả có thể trả lại những quyển sách mà họ đã mượn.

 Website quản trị:

- Website quản trị có những phần chính:

1. Quản lý sách và danh mục sách.
2. Quản lý mượn sách và danh sách các độc giả đang mượn sách.
3. Quản lý nhà xuất bản và danh sách các nhà xuất bản.
4. Quản lý độc giả và danh sách các độc giả.

- Chức năng:

1. Quản lý sách và danh mục sách: tìm kiếm sách dựa theo tên hoặc các ký tự liên quan, thêm sách, chỉnh sửa sách, xóa sách.
2. Quản lý mượn sách và danh sách các độc giả đang mượn sách: thực hiện được các chức năng tương tự như trang độc giả là đăng kí mượn sách và đăng kí trả sách, theo dõi và tìm kiếm các độc giả hiện đang mượn.
3. Quản lý nhà xuất bản và danh sách các nhà xuất bản: tìm kiếm nhà xuất bản dựa theo tên hoặc các ký tự liên quan, thêm nhà xuất bản, chỉnh sửa nhà xuất bản, xóa nhà xuất bản.
4. Quản lý độc giả và danh sách các độc giả: tìm kiếm độc giả dựa theo tên hoặc các ký tự liên quan, thêm độc giả, chỉnh sửa độc giả, xóa độc giả.

PHẦN NỘI DUNG

CHƯƠNG 1

ĐẶC TẢ YÊU CẦU

Việc xây dựng "Hệ thống quản lý mượn sách trực tuyến" nhằm tạo ra một nền tảng thuận tiện và hiệu quả cho việc quản lý, tìm kiếm và mượn sách trong thư viện. Đây cũng là một cách hiện đại hóa quy trình quản lý sách, tăng cường tính chất lượng và dễ dàng truy cập cho độc giả.

- Giao diện và Chức năng
 - Giao diện đơn giản và trực quan: Hệ thống cung cấp giao diện người dùng thân thiện, hiển thị rõ ràng thông tin về các cuốn sách, bao gồm tên sách, tác giả, mô tả chi tiết và hình ảnh sách rõ nét. Mỗi cuốn sách sẽ được gán một mã sách sách riêng biệt để dễ dàng quản lý và tìm kiếm.
 - Tìm kiếm sách: Độc giả có thể tìm kiếm sách theo tên, tác giả.
 - Quy trình mượn sách: Độc giả chọn sách mình muốn mượn và điền các thông tin để đăng kí mượn sách.
- Quản lý hệ thống
 - Quản lý sách: Người quản lý (Admin) có quyền thêm mới sách vào hệ thống khi có sách mới, cập nhật thông tin sách, và xóa sách không còn hoặc đã lỗi.
 - Quản lý nhà xuất bản: Người quản lý (Admin) có quyền thêm mới nhà xuất bản vào hệ thống khi có nhà xuất bản mới, cập nhật thông tin nhà xuất bản, và xóa nhà xuất bản.
 - Quản lý độc giả: Admin có thể quản lý thông tin của độc giả, bao gồm thông tin mượn sách và trạng thái ngày mượn và ngày trả. Họ cũng có quyền thêm mới, chỉnh sửa hoặc xóa tài khoản độc giả khi cần thiết.

Nhờ vào việc áp dụng công nghệ MEVN STACK, "Hệ thống quản lý mượn sách trực tuyến" không chỉ giúp nâng cao chất lượng quản lý sách mà còn tối ưu hóa trải nghiệm người dùng, đồng thời cung cấp các công cụ quản lý hiệu quả cho người quản lý.

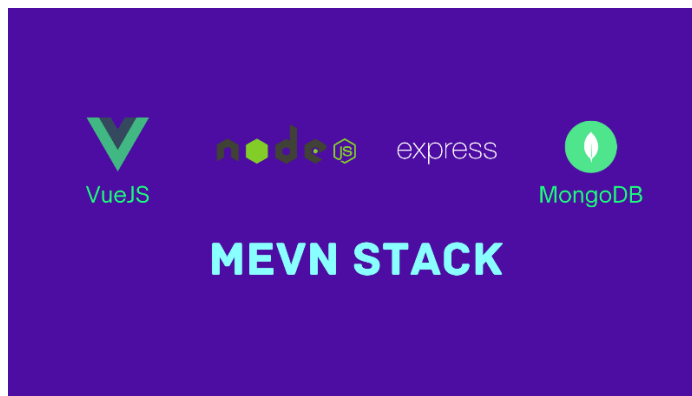
CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

1. Tổng quan về MEVN STACK:

1.1 Khái niệm:

MEVN STACK là sự kết hợp giữa của **MongoDB**, **Express.js**, **Vue.js**, **Node.js** và khiến cho việc xây dựng những ứng dụng web trở nên mạnh mẽ và đơn giản hơn bao giờ hết.



Hình 1. MEVN STACK

1.2 Ưu và nhược điểm của công nghệ MEVN STACK:

+ Ưu điểm:

- **Dễ dàng xây dựng ứng dụng web động:** MEVN stack cung cấp một giải pháp đầy đủ cho việc xây dựng các ứng dụng web động. Bằng cách kết hợp MongoDB để lưu trữ dữ liệu, Express.js để quản lý tương tác với dữ liệu, Vue.js để tạo giao diện người dùng và Node.js để xây dựng máy chủ, MEVN stack cung cấp một cách tiếp cận đơn giản và hiệu quả để phát triển ứng dụng web động.
- **Tính linh hoạt:** Mỗi thành phần trong MEVN stack đều rất linh hoạt và cho phép các nhà phát triển tùy chỉnh ứng dụng của mình theo nhu cầu của họ. MongoDB cung cấp một cách tiếp cận linh hoạt cho việc lưu trữ dữ liệu, Express.js cho phép các nhà phát triển xây dựng các API RESTful động, Vue.js cho phép các nhà phát triển tạo giao diện người dùng phức tạp và Node.js cho phép các nhà phát triển xây dựng ứng dụng web động đa nền tảng.
- **Hiệu năng cao:** MEVN stack sử dụng Node.js để xây dựng máy chủ, cung cấp khả năng xử lý đồng thời cao. Điều này giúp các ứng dụng xây dựng

bằng MEVN stack có thể xử lý nhiều yêu cầu cùng một lúc, tăng hiệu suất và tốc độ phản hồi.

- **Cộng đồng lớn:** MEVN stack được sử dụng rộng rãi trong cộng đồng phát triển web, vì vậy có rất nhiều tài liệu và hỗ trợ từ cộng đồng. Điều này giúp các nhà phát triển dễ dàng tìm kiếm thông tin và giải quyết các vấn đề khi phát triển ứng dụng.
- **Nhược điểm:**
- **Không phù hợp cho các ứng dụng lớn:** MEVN stack không phù hợp cho các ứng dụng lớn và phức tạp, do các công nghệ trong stack này chủ yếu được sử dụng cho các ứng dụng web đơn giản và vừa phải. Nếu bạn đang phát triển một ứng dụng web lớn và phức tạp hơn, bạn có thể cần sử dụng các công nghệ khác hoặc tùy chỉnh MEVN stack để đáp ứng nhu cầu của mình.
- **Cần có kiến thức về nền tảng JavaScript:** Để phát triển các ứng dụng với MEVN stack, bạn cần có kiến thức về nền tảng JavaScript, bao gồm cả các framework Vue.js và Node.js. Nếu bạn không có kinh nghiệm với JavaScript, việc học và sử dụng MEVN stack có thể khó khăn.
- **Độ phức tạp của MongoDB:** Mặc dù MongoDB là một cơ sở dữ liệu NoSQL rất phổ biến, nhưng nó có một số đặc điểm phức tạp, đặc biệt là khi sử dụng các tính năng phức tạp như đa lưu trữ hoặc quản lý giao dịch. Điều này có thể khiến cho việc phát triển và bảo trì ứng dụng với MEVN stack trở nên khó khăn hơn.
- **Khả năng mở rộng giới hạn:** MEVN stack cung cấp một giải pháp đơn giản và nhanh chóng cho việc xây dựng ứng dụng web động, tuy nhiên, khả năng mở rộng của nó có thể bị giới hạn. Nếu ứng dụng của bạn phát triển mạnh mẽ và có nhu cầu mở rộng, bạn có thể cần phải tùy chỉnh MEVN stack hoặc chuyển sang các công nghệ khác để đáp ứng nhu cầu của mình.

2. MongoDB:



Hình 2. MongoDB

- Là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, phát triển bởi MongoDB. Các ứng dụng được phát triển trên MongoDB có thể được triển khai trên các nền tảng khác nhau, bao gồm các môi trường đám mây và máy chủ riêng.

- MongoDB được sử dụng rộng rãi trong các ứng dụng web, đặc biệt là các ứng dụng có nhu cầu xử lý dữ liệu lớn và phức tạp, như các ứng dụng thương mại điện tử, các hệ thống giám sát và các ứng dụng chạy trên đám mây.

3. Restful API:

Khái niệm:

RESTful API (Representational State Transfer) là một kiểu kiến trúc ứng dụng mạng dựa trên giao thức HTTP, giúp các ứng dụng giao tiếp với nhau thông qua các phương thức HTTP như GET, POST, PUT, DELETE. RESTful API giúp đơn giản hóa việc truyền tải dữ liệu giữa client và server dưới dạng JSON hoặc XML.

Ưu điểm và hạn chế:

+ Ưu điểm:

- Dễ hiểu và sử dụng: RESTful API sử dụng các phương thức HTTP thông dụng, giúp dễ dàng triển khai và tích hợp.
- Tính nhất quán: RESTful API đảm bảo tính nhất quán trong việc truy cập và quản lý dữ liệu.
- Khả năng mở rộng: RESTful API hỗ trợ khả năng mở rộng tốt, giúp dễ dàng thêm các tính năng mới.
- Bảo mật: RESTful API hỗ trợ nhiều cơ chế xác thực và ủy quyền, giúp bảo vệ dữ liệu và tài nguyên.

- Ngôn ngữ và nền tảng độc lập: RESTful API có thể hoạt động trên nhiều ngôn ngữ và nền tảng khác nhau.

+ Hạn chế:

- Trạng thái phiên (stateless): RESTful API không lưu trữ trạng thái phiên giữa các yêu cầu, điều này có thể gây khó khăn trong việc xử lý các tác vụ phức tạp.
- Tốc độ truy vấn: RESTful API có thể chậm hơn so với các giao thức khác như gRPC hay GraphQL, đặc biệt khi cần truy vấn nhiều dữ liệu liên quan.
- Over-fetching và under-fetching: RESTful API có thể trả về nhiều dữ liệu hơn hoặc ít hơn so với nhu cầu của client, dẫn đến lãng phí tài nguyên và băng thông.
- Khó khăn trong phiên bản hóa: Việc cập nhật và quản lý phiên bản của RESTful API có thể gây khó khăn, khiến cho việc duy trì và nâng cấp hệ thống trở nên phức tạp.

4. Express.js:

- Express.js là một framework web được xây dựng trên nền tảng Node.js. Nó được phát triển nhằm giúp cho việc xây dựng ứng dụng web với Node.js trở nên đơn giản và nhanh chóng hơn. Express.js cung cấp cho các nhà phát triển các tính năng và công cụ để xử lý các yêu cầu HTTP, xử lý các phản hồi, quản lý session và cookie, và nhiều tính năng khác.
- Express.js được thiết kế để đơn giản hóa việc xây dựng các ứng dụng web và API với Node.js. Nó cho phép các nhà phát triển tạo ra các định tuyến (routing) dựa trên các yêu cầu HTTP, tạo middleware để xử lý các yêu cầu, và cung cấp các công cụ để quản lý các kết nối và phiên làm việc.

5. Vue.js:



Hình 3. Vue.js

- Vue.js là một framework JavaScript mã nguồn mở được sử dụng để xây dựng các ứng dụng web động và giao diện người dùng (UI). Nó được thiết kế để cung cấp một cách tiếp cận dễ dàng cho việc xây dựng các ứng dụng phức tạp và tương tác với người dùng.

- Vue.js cho phép các nhà phát triển tạo ra các thành phần (components) tái sử dụng để xây dựng giao diện người dùng và quản lý trạng thái ứng dụng trong một cách hiệu quả. Nó cũng cung cấp các công cụ để liên kết dữ liệu (data binding) và xử lý sự kiện (event handling), giúp cho việc phát triển các ứng dụng web trở nên đơn giản và nhanh chóng hơn.

- Vue.js có thiết kế linh hoạt và có thể tích hợp vào các ứng dụng web hiện có một cách dễ dàng. Nó hỗ trợ các tính năng như hot-reload (tự động tải lại khi có thay đổi), Vuex (trạng thái quản lý), Vue Router (định tuyến) và nhiều tính năng khác để giúp cho việc phát triển các ứng dụng web trở nên dễ dàng và linh hoạt hơn.

6. Node.js:



Hình 4. Node.js

- Node.js là một nền tảng phát triển được xây dựng trên Chrome's V8 JavaScript engine để cho phép các nhà phát triển sử dụng JavaScript để phát triển các ứng dụng trên môi trường máy chủ. Nó cho phép các nhà phát triển xây dựng các ứng dụng web và các dịch vụ mạng khác một cách dễ dàng và hiệu quả.

- Node.js cho phép các nhà phát triển xây dựng các ứng dụng web với hiệu suất cao bằng cách sử dụng JavaScript để xử lý các yêu cầu từ phía máy khách và chuyển tiếp chúng đến máy chủ. Nó cũng cung cấp các tính năng để xử lý các yêu cầu không đồng bộ (asynchronous), giúp cho các ứng dụng web trở nên nhanh chóng và hiệu quả hơn.

- Node.js có thiết kế linh hoạt và có thể tích hợp vào các ứng dụng web hiện có một cách dễ dàng. Nó có thể được sử dụng để xây dựng các ứng dụng web đơn giản hoặc các ứng dụng web phức tạp, và hỗ trợ các tính năng như mã nguồn mở, quản lý gói, và phát triển cộng đồng.

7. Tổng quan về Mongoose:

- Mongoose là một thư viện ODM (Object-Document Mapping) cho MongoDB và Node.js, cho phép lập trình viên dễ dàng tương tác với cơ sở dữ liệu MongoDB thông qua các đối tượng JavaScript.
- Mongoose cung cấp các tính năng như:
 - + **Validation:** Kiểm tra tính hợp lệ của dữ liệu trước khi lưu vào cơ sở dữ liệu.
 - + **Query building:** Xây dựng các truy vấn đến cơ sở dữ liệu MongoDB một cách dễ dàng.
 - + **Middleware:** Cho phép định nghĩa các hàm middleware được thực hiện trước hoặc sau các hoạt động như lưu hoặc xóa đối tượng.
 - + **Schema definition:** Cho phép định nghĩa các lược đồ (schema) cho các đối tượng được lưu trữ trong cơ sở dữ liệu.
 - + **Population:** Cho phép lấy dữ liệu từ các bảng liên quan đến nhau (references) trong cơ sở dữ liệu MongoDB.

8. Các công cụ hỗ trợ

8.1 Visual Studio Code



Hình 5: Visual Studio Code

Visual Studio Code (VS Code) là một trình biên soạn mã nguồn mở và miễn phí được phát triển bởi Microsoft. Nó được thiết kế để hỗ trợ một loạt ngôn ngữ lập trình và công nghệ web, bao gồm cả JavaScript, TypeScript, CSS, HTML, và nhiều hơn nữa. VS Code có sự kết hợp hoàn hảo giữa tính năng mạnh mẽ và hiệu suất tối ưu, giúp người phát triển tăng cường năng suất làm việc.

Một số tính năng nổi bật của VS Code bao gồm:

- Giao diện người dùng thân thiện: VS Code có giao diện sử dụng người dùng thân thiện và dễ dàng để thao tác.
- Hỗ trợ ngôn ngữ đa dạng: Nó hỗ trợ nhiều ngôn ngữ lập trình và framework, có thể được mở rộng thông qua các tiện ích mở rộng (extensions).
- Debugging và Testing: VS Code cung cấp các công cụ mạnh mẽ để debug và kiểm thử mã nguồn một cách dễ dàng.
- IntelliSense: Tính năng này giúp tự động hoàn thành mã, kiểm tra lỗi và đề xuất cú pháp trong quá trình viết mã.
- Intégration với Git: VS Code tích hợp sẵn với Git, cho phép bạn quản lý phiên bản mã nguồn và thực hiện các thao tác git một cách trực quan.

8.2. Postman



Hình 6: Postman

Postman là một công cụ hỗ trợ phát triển và kiểm thử API (Application Programming Interface) đến từ Postman Technologies. Với giao diện người dùng thân thiện và các tính năng mạnh mẽ, Postman giúp các nhà phát triển và kiểm thử viên dễ dàng tạo, kiểm tra, và quản lý các yêu cầu API.

Một số tính năng và ưu điểm của Postman bao gồm:

- Giao diện người dùng dễ sử dụng: Postman có giao diện trực quan và thân thiện, giúp người dùng dễ dàng tạo và quản lý các yêu cầu API.
- Hỗ trợ nhiều phương thức HTTP: Postman hỗ trợ nhiều phương thức HTTP như GET, POST, PUT, DELETE và nhiều phương thức khác.
- Kiểm thử API đa dạng: Bạn có thể kiểm thử API của bạn cho mọi trường hợp, từ kiểm tra yêu cầu đơn giản đến các scenario phức tạp với nhiều biến số.

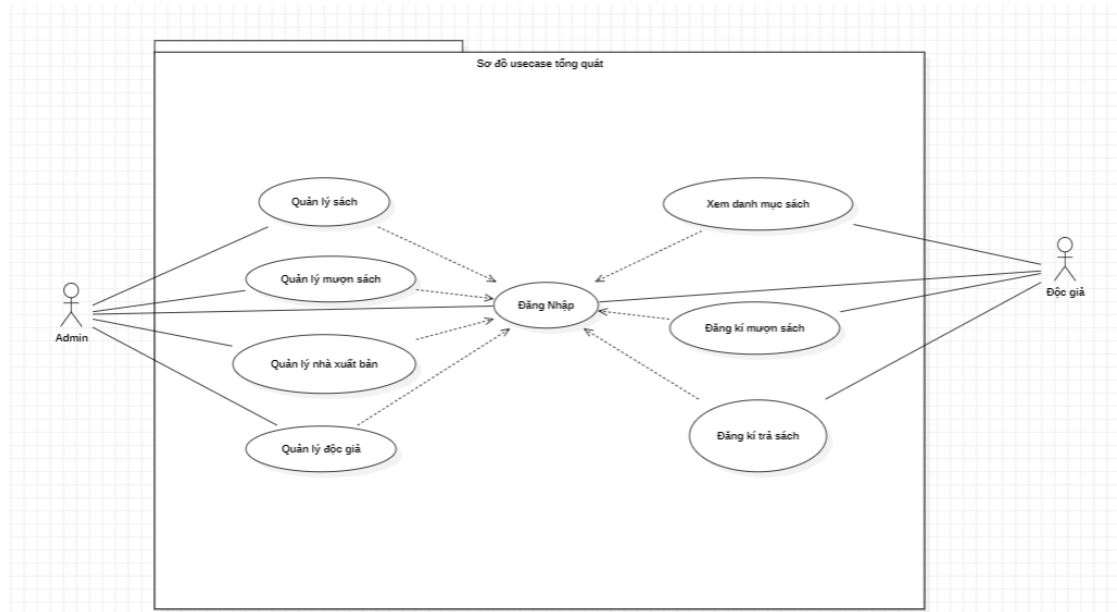
- Môi trường và biến: Postman cho phép bạn quản lý môi trường và biến, giúp dễ dàng thay đổi các giá trị trong các yêu cầu API mà không cần chỉnh sửa mã.
- Tích hợp với Git và cơ sở dữ liệu: Postman có tích hợp với Git, cho phép bạn lưu trữ và quản lý các bộ kiểm thử API trên kho lưu trữ của mình.
- Mô phỏng và kiểm thử tự động: Postman cho phép bạn tạo các bộ kiểm thử tự động và mô phỏng các yêu cầu API để giúp bạn đảm bảo rằng API của bạn hoạt động đúng như mong đợi.

CHƯƠNG 3

CÁC SƠ ĐỒ CỦA WEBSITE

1. Sơ đồ Use case

1.1 Sơ đồ Use case tổng quát của hệ thống



Sơ đồ 1. Sơ đồ use case tổng quát

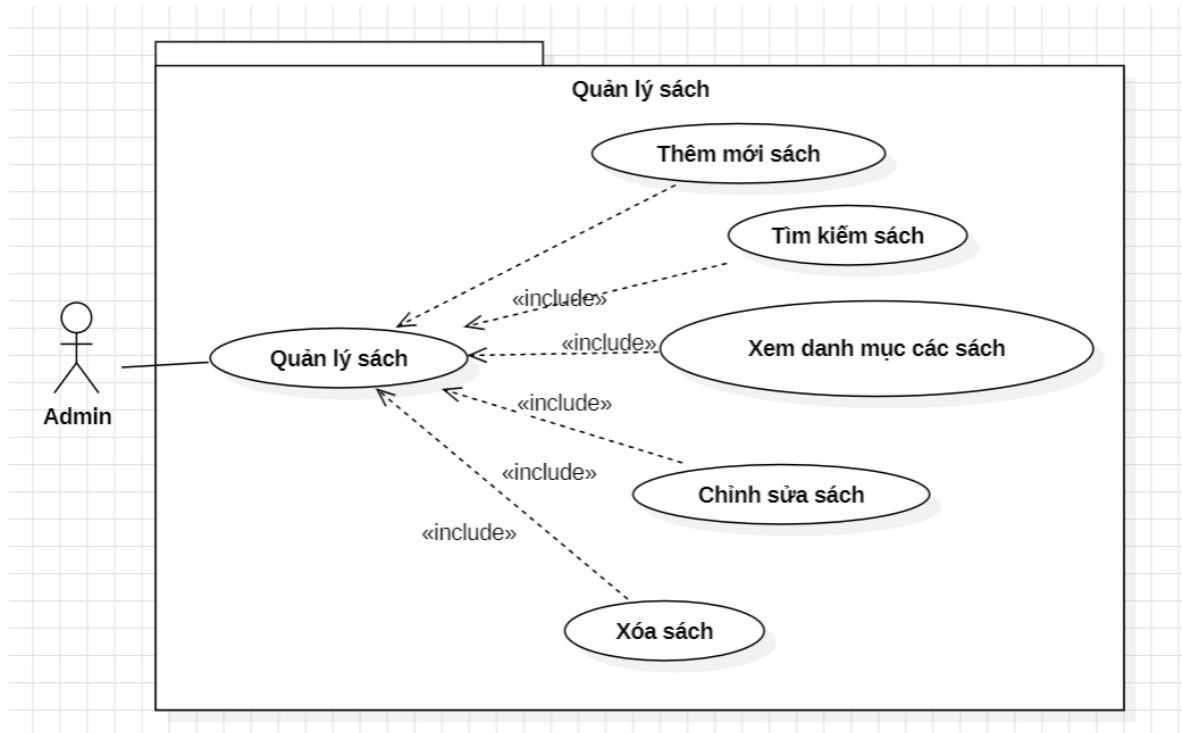
1.1.1 Đặc tả sơ đồ chức năng use case tổng quát

STT	Tên use case	Diễn giải
1	Đăng nhập	Đăng nhập vào website để có thể thao tác đầy đủ chức năng của website.
2	Quản lý sách	Admin có thể thêm, sửa và xóa sách trên website. Có thể thêm các sách mới, điều chỉnh các thông tin của sách, xem danh mục các sách và tìm kiếm sách.

3	Quản lý mượn sách	Admin có thể theo dõi trạng thái danh sách các độc giả đang mượn và có thể thực hiện đăng kí mượn hoặc trả sách cho các độc giả.
4	Quản lý nhà xuất bản	Admin có thể thêm, sửa và xóa nhà xuất bản trên website. Có thể thêm các nhà xuất bản mới, điều chỉnh các thông tin của nhà xuất bản, xem danh sách các nhà xuất bản và tìm kiếm nhà xuất bản
5	Quản lý độc giả	Admin có thể thêm, sửa và xóa độc giả trên website. Có thể thêm các độc giả mới, điều chỉnh các thông tin của độc giả, xem danh sách các độc giả và tìm kiếm độc giả.
6	Xem danh mục sách	Độc giả có thể xem danh mục các sách hiện tại có trên website.
7	Đăng kí mượn sách	Độc giả có thể đăng kí mượn sách trên website.
8	Đăng kí trả sách	Độc giả có thể đăng kí trả sách trên website.

Bảng 1: Đặc tả sơ đồ chức năng use case tổng quát

1.2 Sơ đồ use case quản lý sách



Sơ đồ 2: Sơ đồ use case Quản lý sách

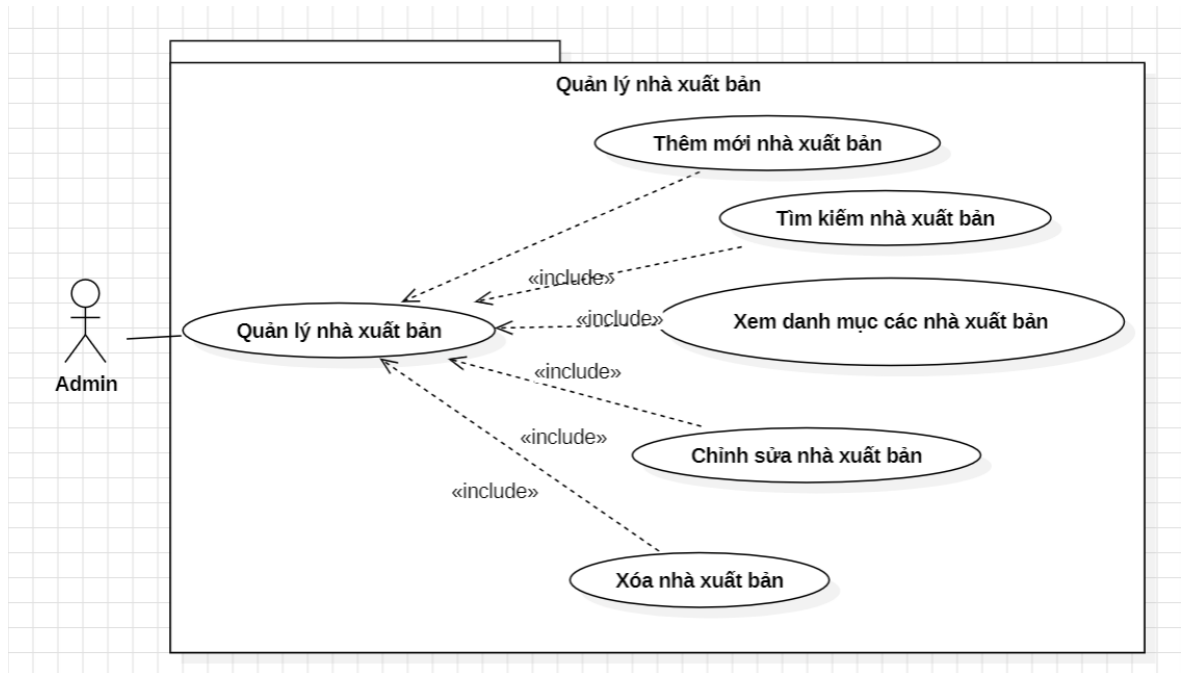
1.2.1 Đặc tả usecase quản lý sách

Tác nhân	Admin
Mô tả	Cho phép người quản trị tìm kiếm, xem danh mục các sách, chỉnh sửa và xóa sách.
Điều kiện	Người quản lý đăng nhập thành công vào hệ thống
Luồng sự kiện chính	<ul style="list-style-type: none"> - Tìm kiếm sách + Admin nhập từ khóa sách cần tìm + Giao diện hiển thị các sách trùng với từ khóa. - Thêm mới sách + Hệ thống hiển thị form thêm mới sách. + Người quản trị sẽ nhập các thông tin như: Mã sách, tên sách, đơn giá, số quyền, năm xuất bản... + Sau đó nhấn vào nút thêm sách.

	<ul style="list-style-type: none"> + Nếu thành công hệ thống hiển thị thông tin thêm mới sách thành công. - Chỉnh sửa sách + Người quản trị bấm vào nút chỉnh sửa sách. + Nhập vào những thông tin cần cập nhật và nhấn cập nhật. + Nếu thành công hệ thống hiển thị thông báo cập nhật thành công sách. - Xóa sách + Người quản trị chọn sách cần xóa. + Nhấn nút xóa để loại bỏ sách. + Hệ thống hiển thị xóa sách thành công. + Hệ thống sẽ cập nhật lại danh sách các sách. + Usecase kết thúc.
Luồng điều kiện rẽ nhánh	<ul style="list-style-type: none"> - Hệ thống sẽ không cập nhật thông tin nếu người quản trị nhập sai hoặc thiếu. - Người quản trị cần nhập lại thông tin. - Quay lại luồng xử lý chính.
Kết quả	Các thông tin của sách sẽ được cập nhật vào cơ sở dữ liệu

Bảng 2: Đặc tả chức năng quản lý sách

1.3 Sơ đồ use case quản lý nhà xuất bản



Sơ đồ 3: Sơ đồ use case quản lý nhà xuất bản

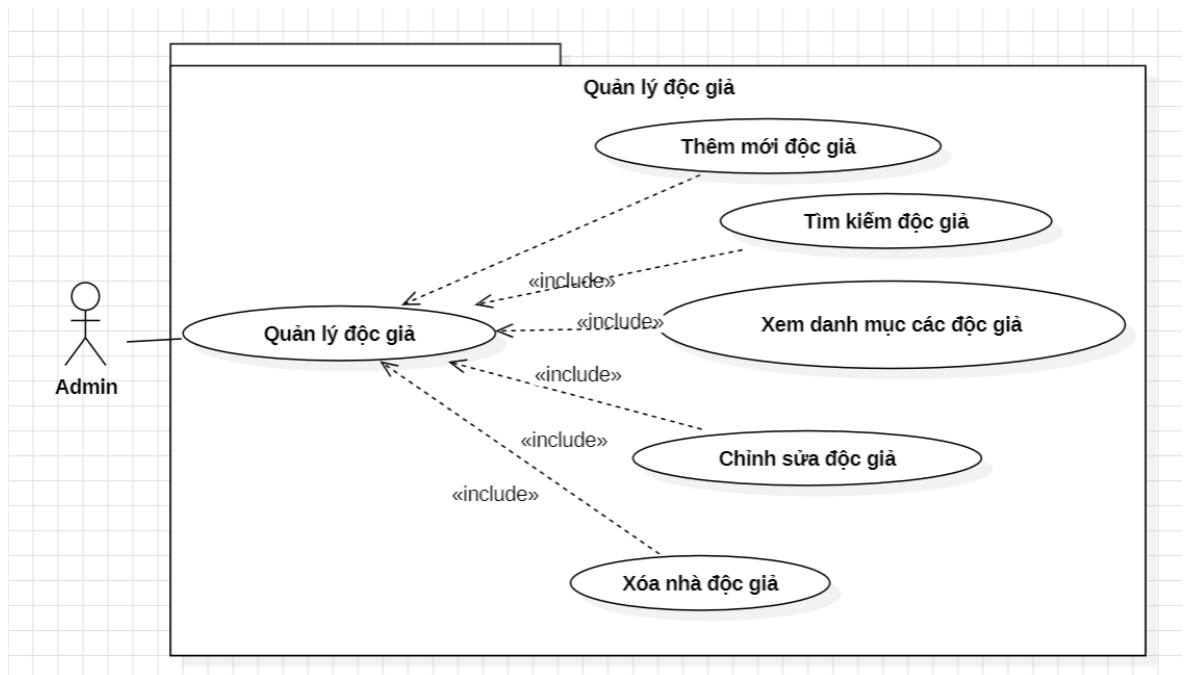
1.3.1 Đặc tả use case quản lý nhà xuất bản

Tác nhân	Admin
Mô tả	Cho phép người quản trị tìm kiếm, xem danh sách các nhà xuất bản, chỉnh sửa và xóa nhà xuất bản.
Điều kiện	Người quản lý đăng nhập thành công vào hệ thống
Luồng sự kiện chính	<ul style="list-style-type: none"> - Tìm kiếm nhà xuất bản + Admin nhập từ khóa sách cần tìm + Giao diện hiển thị các nhà xuất bản trùng với từ khóa. - Thêm mới nhà xuất bản + Hệ thống hiển thị form thêm mới nhà xuất bản. + Người quản trị sẽ nhập các thông tin như: Mã nhà xuất bản, tên nhà xuất bản, địa chỉ. + Sau đó nhấn vào nút thêm nhà xuất bản.

	<ul style="list-style-type: none"> + Nếu thành công hệ thống hiển thị thông tin thêm mới nhà xuất bản thành công. - Chỉnh sửa nhà xuất bản + Người quản trị bấm vào nút chỉnh sửa nhà xuất bản. + Nhập vào những thông tin cần cập nhật và nhấn cập nhật. + Nếu thành công hệ thống hiển thị thông báo cập nhật thành công nhà xuất bản. - Xóa nhà xuất bản + Người quản trị chọn nhà xuất bản cần xóa. + Nhấn nút xóa để loại bỏ nhà xuất bản. + Hệ thống hiển thị xóa nhà xuất bản thành công. + Hệ thống sẽ cập nhật lại danh sách các nhà xuất bản. + Usecase kết thúc.
Luồng điều kiện rẽ nhánh	<ul style="list-style-type: none"> - Hệ thống sẽ không cập nhật thông tin nếu người quản trị nhập sai hoặc thiếu. - Người quản trị cần nhập lại thông tin. - Quay lại luồng xử lý chính
Kết quả	Các thông tin của nhà xuất bản sẽ được cập nhật vào cơ sở dữ liệu

Bảng 3: Đặc tả chức năng quản lý nhà xuất bản

1.4 Sơ đồ use case quản lý độc giả



Sơ đồ 4: Sơ đồ use case quản lý độc giả

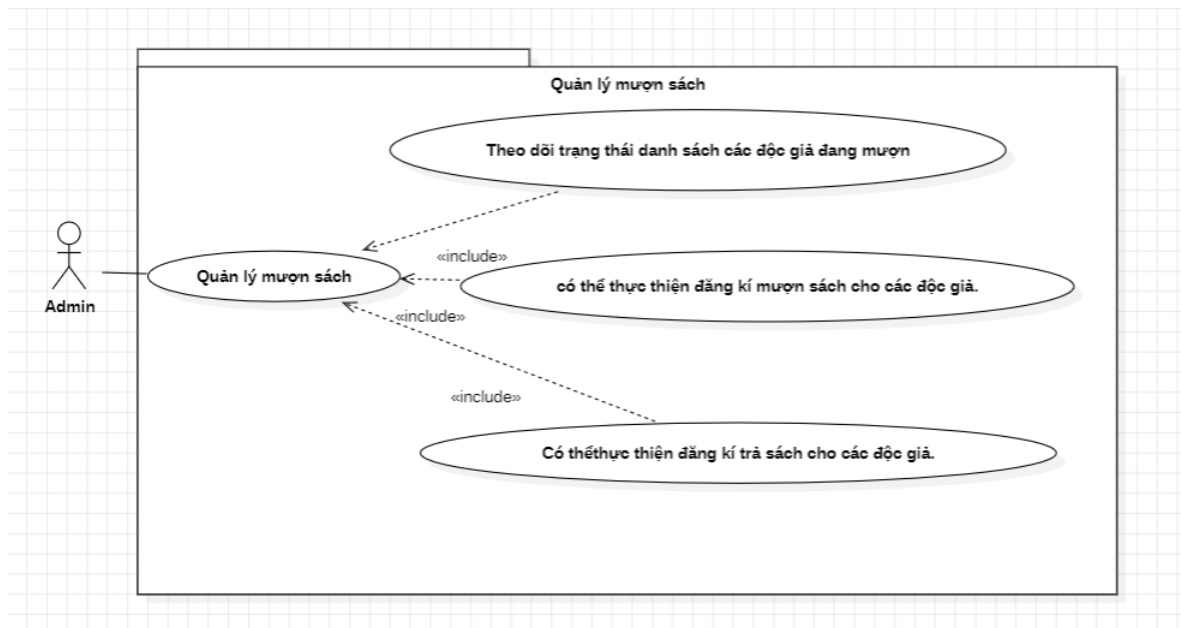
1.4.1 Đặc tả use case quản lý độc giả

Tác nhân	Admin
Mô tả	Cho phép người quản trị tìm kiếm, xem danh sách các độc giả, chỉnh sửa và xóa độc giả.
Điều kiện	Người quản lý đăng nhập thành công vào hệ thống
Luồng sự kiện chính	<ul style="list-style-type: none"> - Tìm kiếm độc giả + Admin nhập từ khóa độc giả cần tìm + Giao diện hiển thị các độc giả trùng với từ khóa. - Thêm mới độc giả + Hệ thống hiển thị form thêm mới độc giả. + Người quản trị sẽ nhập các thông tin như: Mã độc giả, họ lót, tên, ngày sinh, phái, địa chỉ... + Sau đó nhấn vào nút thêm độc giả.

	<ul style="list-style-type: none"> + Nếu thành công hệ thống hiển thị thông tin thêm mới độc giả thành công. - Chỉnh sửa độc giả + Người quản trị bấm vào nút chỉnh sửa sách. + Nhập vào những thông tin cần cập nhật và nhấn cập nhật. + Nếu thành công hệ thống hiển thị thông báo cập nhật thành công độc giả. - Xóa độc giả + Người quản trị chọn độc giả cần xóa. + Nhấn nút xóa để loại bỏ độc giả. + Hệ thống hiển thị xóa độc giả thành công. + Hệ thống sẽ cập nhật lại danh sách các độc giả. + Usecase kết thúc.
Luồng điều kiện rẽ nhánh	<ul style="list-style-type: none"> - Hệ thống sẽ không cập nhật thông tin nếu người quản trị nhập sai hoặc thiếu. - Người quản trị cần nhập lại thông tin. - Quay lại luồng xử lý chính.
Kết quả	Các thông tin của độc giả sẽ được cập nhật vào cơ sở dữ liệu

Bảng 4: Đặc tả chức năng quản lý độc giả

1.5 Sơ đồ use case quản lý mượn sách



Sơ đồ 5: Sơ đồ use case quản lý mượn sách

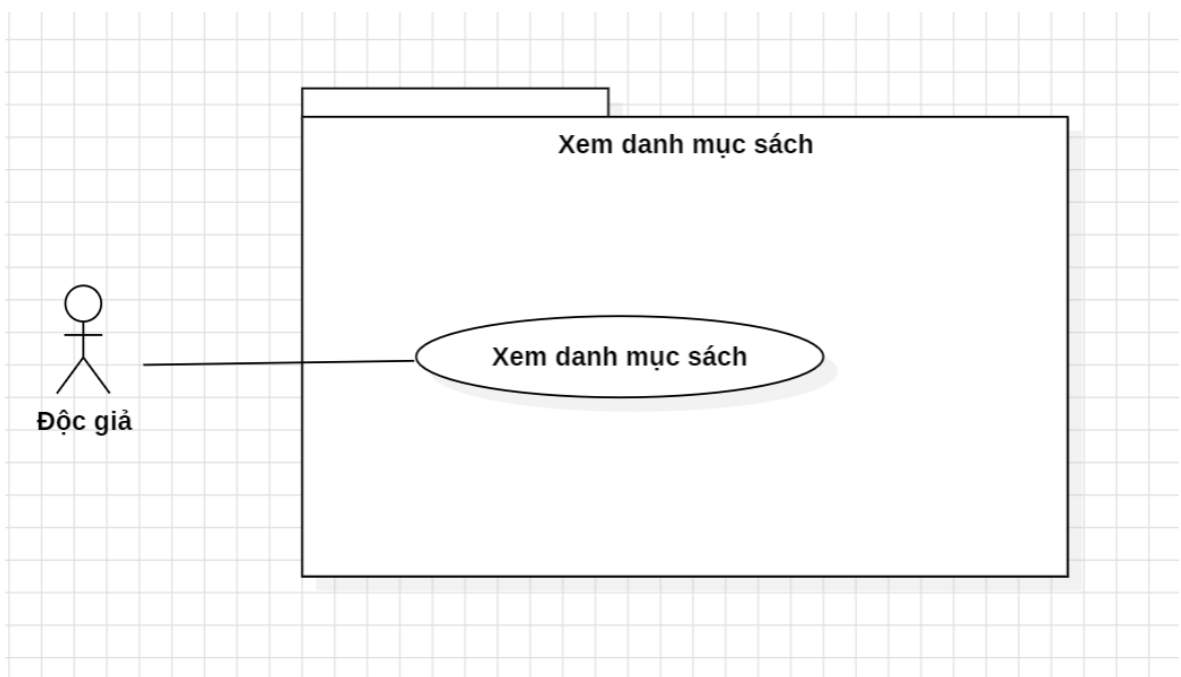
1.5.1 Đặc tả use case quản lý mượn sách

Tác nhân	Admin
Mô tả	Cho phép người quản trị theo dõi danh sách và tìm kiếm các độc giả đang mượn sách, có thể hỗ trợ độc giả đăng kí mượn và trả sách.
Điều kiện	Người quản lý đăng nhập thành công vào hệ thống.
Luồng sự kiện chính	<p>Tìm kiếm và theo dõi danh sách các độc giả:</p> <ul style="list-style-type: none"> + Người quản trị có thể tìm độc giả dựa trên các thông tin như mã số độc giả, mà sách đang mượn cũng như ngày mượn và ngày trả sách. <p>Thực hiện hỗ trợ độc giả đăng mượn sách</p> <ul style="list-style-type: none"> + Người quản trị có thể sử dụng mã độc giả và mã sách cũng như ngày trả sách mà độc giả cung cấp để hỗ trợ độc giả mượn sách. <p>Thực hiện hỗ trợ độc giả đăng trả sách</p>

	+ Người quản trị cũng có thể sử dụng mã độc giả và mã sách mà độc giả cung cấp hoặc các thông tin có trên danh sách các độc giả đang mượn để hỗ trợ độc giả đăng kí trả sách.
Luồng điều kiện rẽ nhánh	
Kết quả	Các thông tin mượn sách của độc giả sẽ được thêm vào cơ sở dữ liệu nếu đăng kí mượn sách và ngược lại sẽ bị xóa đi nếu đăng kí trả sách.

Bảng 5: Đặc tả chức năng quản lý mượn sách

1.6 Sơ đồ use case xem danh mục sách



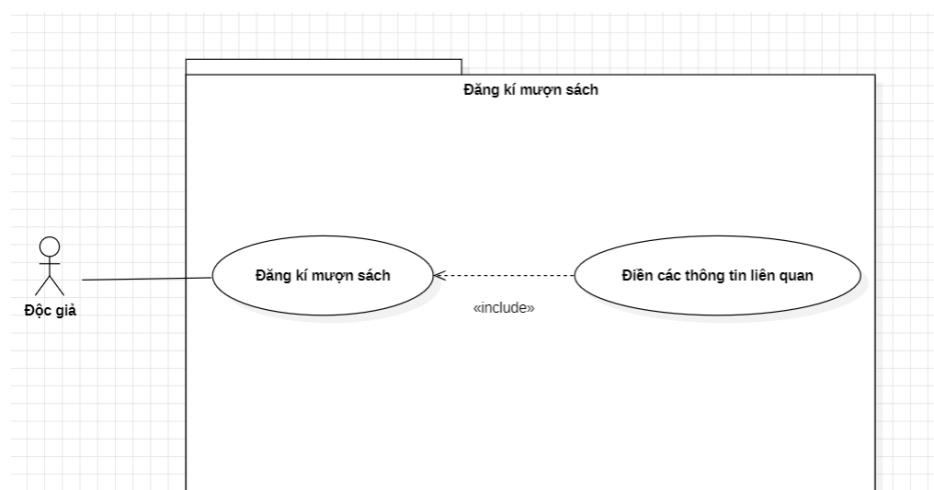
Sơ đồ 6: Sơ đồ use case xem danh mục các sách

1.6.1 Đặc tả use case xem danh mục sách

Tác nhân	Độc giả
Mô tả	Cho phép độc giả xem danh mục và tìm kiếm các sách hiện đang có trong danh mục các sách có thể mượn
Điều kiện	Độc giả đăng nhập thành công vào hệ thống.
Luồng sự kiện chính	<p>Tìm kiếm và xem danh mục các sách hiện đang có</p> <p>+ Độc giả nhập từ khóa sách cần tìm</p> <p>+ Giao diện hiển thị các sách trùng với từ khóa.</p> <p>+ Độc giả có thể xem được các thông tin liên quan tới sách</p>
Luồng điều kiện rẽ nhánh	
Kết quả	Độc giả có thể xem và tìm kiếm được các thông tin về sách.

Bảng 6: Đặc tả chức năng xem danh mục sách

1.7 Sơ đồ use case đăng kí mượn sách



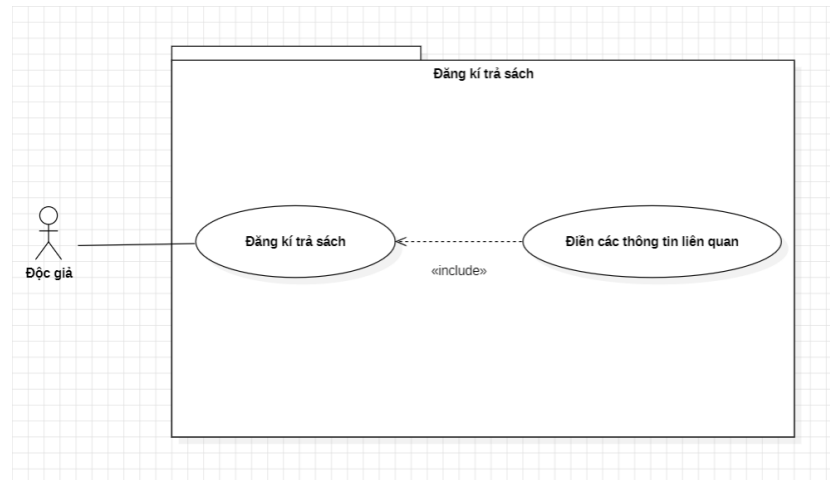
Sơ đồ 7: Sơ đồ use case đăng kí mượn sách

1.7.1 Đặc tả use case đăng kí mượn sách

Tác nhân	Độc giả
Mô tả	Cho phép độc giả đăng kí mượn sách
Điều kiện	Độc giả đăng nhập thành công vào hệ thống.
Luồng sự kiện chính	<p>Thực hiện đăng kí mượn sách</p> <p>+ Độc giả điền các thông tin như mã độc giả, mã sách mà độc giả chọn, ngày mượn và ngày trả dự kiến mà độc giả chọn vào form đăng kí mượn sách</p> <p>+ Nhấn nút đăng kí để mượn sách</p> <p>+ Hệ thống thông báo độc giả mượn sách thành công.</p>
Luồng điều kiện rẽ nhánh	<p>- Hệ thống sẽ không cập nhật thông tin nếu độc giả nhập sai hoặc thiếu.</p> <p>- Độc giả cần nhập lại thông tin.</p> <p>- Quay lại luồng xử lý chính.</p>
Kết quả	Các thông tin đăng ký mượn sách sẽ được cập nhật vào cơ sở dữ liệu

Bảng 7: Đặc tả chức năng đăng kí mượn sách

1.8 Sơ đồ use case đăng kí trả sách



Sơ đồ 8: Sơ đồ use case đăng kí trả sách

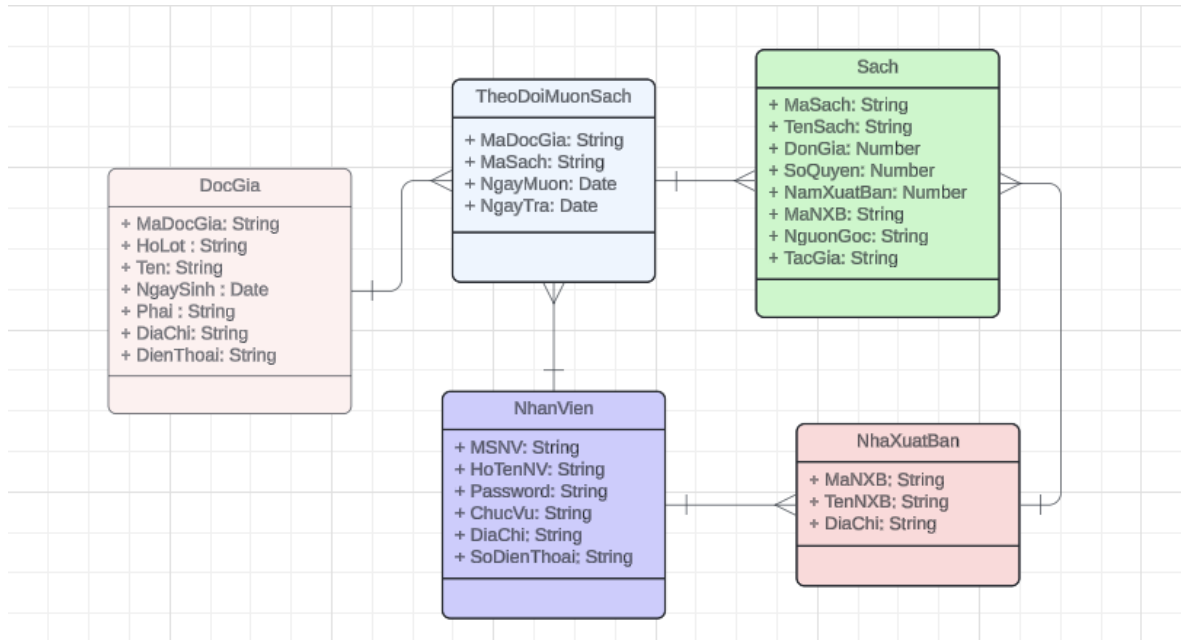
1.8.1 Đặc tả use case đăng kí trả sách

Tác nhân	Độc giả
Mô tả	Cho phép độc giả đăng kí trả sách
Điều kiện	Độc giả đăng nhập thành công vào hệ thống.
Luồng sự kiện chính	Thực hiện đăng kí trả sách + Độc giả điền các thông tin như mã độc giả, mã sách vào form đăng kí trả sách + Nhấn nút đăng kí để trả sách + Hệ thống thông báo độc giả trả sách thành công.
Luồng điều kiện rẽ nhánh	- Hệ thống sẽ không cập nhật thông tin nếu độc giả nhập sai hoặc thiếu. - Độc giả cần nhập lại thông tin. - Quay lại luồng xử lý chính.
Kết quả	Các thông tin đăng ký trả sách sẽ được cập nhật vào cơ sở dữ liệu

Bảng 8: Đặc tả chức năng đăng kí trả sách

2. Cơ sở dữ liệu

2.1 Mô hình cơ sở dữ liệu CDM



Hình 7: Mô hình CDM

2.2 Mô tả các lớp:

STT	Tên bảng	Diễn giải
1	Sach	Danh mục các sách
2	NhaXuatBan	Danh sách các nhà xuất bản
3	DocGia	Danh sách các độc giả
4	TheoDoiMuonSach	Danh sách các độc giả đang mượn sách
5	NhanVien	Danh sách các nhân viên

Bảng 9: Tên các bảng trong cơ sở dữ liệu CDM

2.3 Mô tả Sách

Thuộc Tính	Kiểu	Khóa chính	Khóa ngoại	Mô tả
MaSach	String	X		Mã sách
TenSach	String			Tên sách
DonGia	Number			Đơn giá sách
SoQuyên	Number			Số quyển
NamXuatBan	Number			Năm xuất bản
MaNXB	String		X	Mã nhà xuất bản
NguonGoc	String			Nguồn gốc sách
TacGia	String			Tác giả của sách

Bảng 10: Mô tả danh mục các sách

2.4 Mô tả NhaXuatBan

Thuộc Tính	Kiểu	Khóa chính	Khóa ngoại	Mô tả
MaNXB	String	X		Mã nhà xuất bản
TenNXB	String			Tên nhà xuất bản
DiaChi	String			Địa chỉ nhà xuất bản

Bảng 11: Mô tả danh sách các nhà xuất bản

2.5 Mô tả DocGia

Thuộc Tính	Kiểu	Khóa chính	Khóa ngoại	Mô tả
MaDocGia	String	X		Mã độc giả
HoLot	String			Họ lót độc giả
Ten	String			Tên độc giả
NgaySinh	Date			Ngày sinh độc giả
Phai	String			Giới tính độc giả
DiaChi	String			Địa chỉ độc giả
DienThoai	String			Số điện thoại độc giả

Bảng 12: Mô tả danh sách các độc giả

2.6 Mô tả TheoDoiMuonSach

Thuộc Tính	Kiểu	Khóa chính	Khóa ngoại	Mô tả
MaDocGia	String		X	Mã độc giả
MaSach	String		X	Mã sách
NgayMuon	Date			Ngày mượn
NgayTra	Date			Ngày trả

Bảng 13: Mô tả danh sách các độc giả đang mượn sách

2.7 Mô tả NhanVien

Thuộc Tính	Kiểu	Khóa chính	Khóa ngoại	Mô tả
MSNV	String	X		Mã số nhân viên
HoTenNV	String			Họ tên nhân viên
Password	String			Mật khẩu
DiaChi	String			Địa chỉ nhân viên
DienThoai	String			Số điện thoại nhân viên

Bảng 14: Mô tả danh sách các độc nhân viên

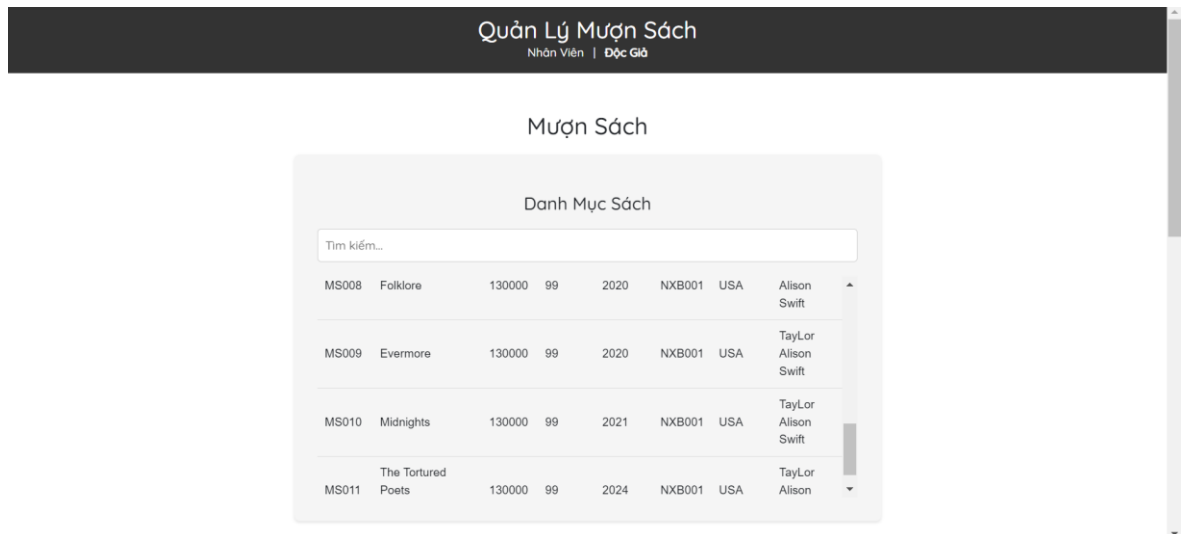
CHƯƠNG 4

KẾT QUẢ ĐẠT ĐƯỢC

Website được xây dựng với giao diện thân thiện và đơn giản, mang tới trải nghiệm tốt cho người.

1. Giao diện Website ở phía độc giả:

1.1 Giao diện hiển thị danh mục các sách hiện có



Hình 8: Danh mục các sách hiện có

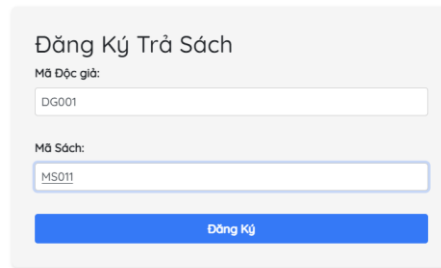
1.2 Giao diện hiển thị form đăng kí mượn sách

The screenshot displays the 'Đăng Ký Mượn Sách' (Book Borrowing Registration) form. The form has the following fields and controls:

- Mã Độc Giả:** A text input field containing 'DG001'.
- Mã Sách:** A text input field containing 'MS011'.
- Ngày Mượn:** A date picker field showing '18/04/2024'.
- Ngày Trả:** A date picker field showing '11/05/2024'.
- Đăng ký:** A blue button at the bottom of the form.

Hình 9: Form đăng kí mượn sách

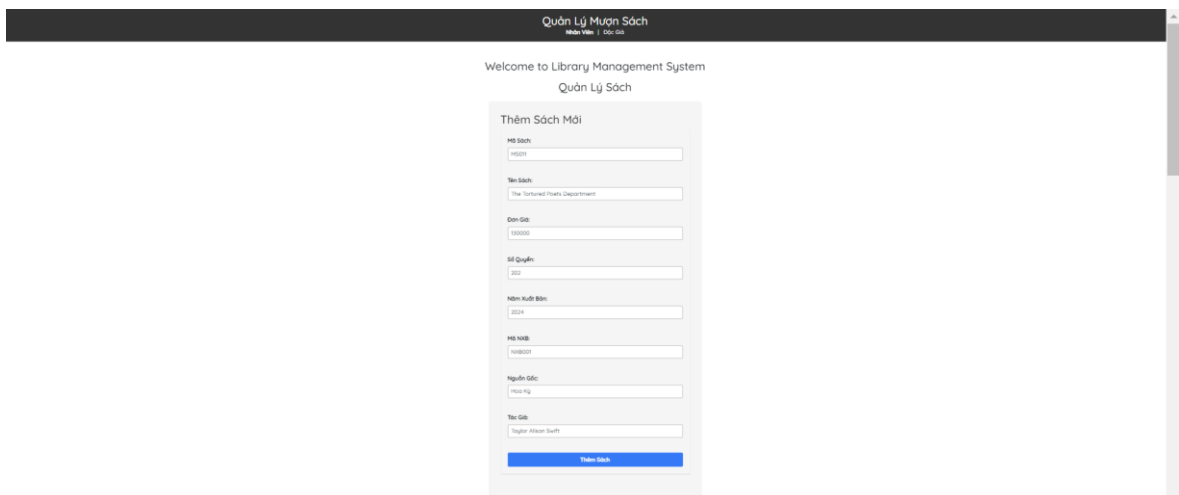
1.3 Giao diện hiển thị form đăng kí trả sách



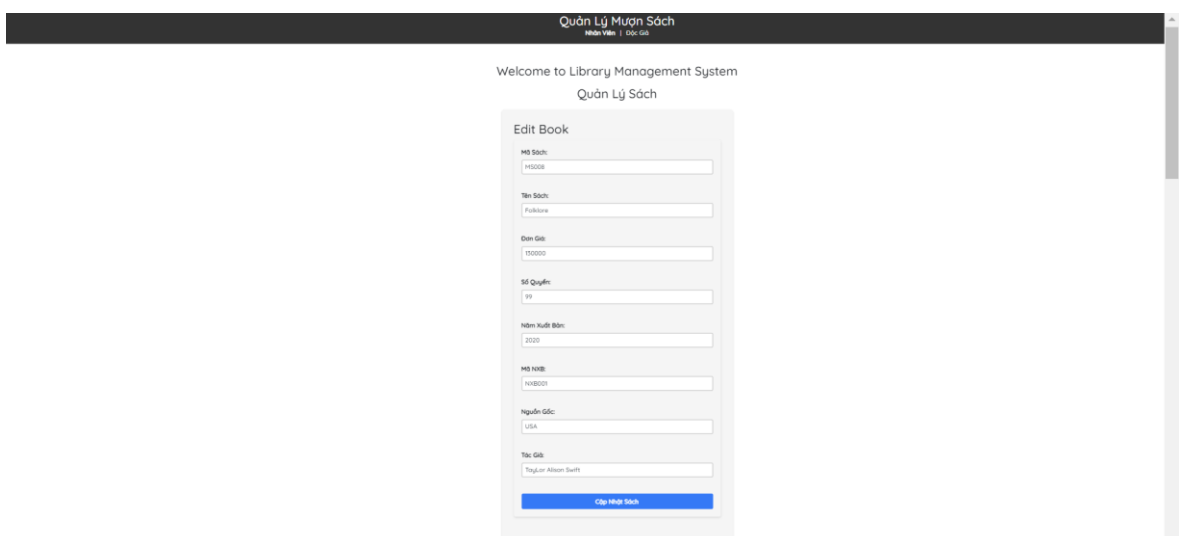
Hình 10: Form đăng kí trả sách

2. Giao diện Website ở phía người quản lý:

2.1 Giao diện hiển thị form thêm mới sách và chỉnh sửa sách



Hình 11: Form thêm mới sách



Hình 12: Form chỉnh sửa sách

2.2 Giao diện hiển thị danh mục các sách hiện có

Danh Mục Sách									
Tìm kiếm...									
MS007	Lover	130000	99	2018	NXB001	USA	TayLor Alison Swift	Chỉnh Sửa	Xóa
MS008	Folklore	130000	99	2020	NXB001	USA	TayLor Alison Swift	Chỉnh Sửa	Xóa
MS009	Evermore	130000	99	2020	NXB001	USA	TayLor Alison Swift	Chỉnh Sửa	Xóa
MS010	Midnights	130000	99	2021	NXB001	USA	TayLor Alison Swift	Chỉnh Sửa	Xóa
MS011	The Tortured Poets Department	130000	99	2024	NXB001	USA	TayLor Alison Swift	Chỉnh Sửa	Xóa

Hình 13: Danh mục các sách hiện có

2.3 Giao diện hiển thị form đăng kí mượn sách

Mượn Sách

Đăng Ký Mượn Sách

Mã Độc Giả:

Mã Sách:

Ngày Mượn:

Ngày Trả:

Đăng ký

Hình 14: Form đăng kí mượn sách

2.4 Giao diện hiển thị form đăng kí trả sách

Đăng Ký Trả Sách

Mã Độc giả:

Mã Sách:

Đăng Ký

Hình 15: Form đăng kí trả sách

2.5 Giao diện hiển thị danh sách các độc giả đang mượn

Danh Sách Các Độc Giả Đang Mượn Sách

Tìm kiếm...

DG001 đã mượn sách có mã là: MS011 vào ngày: 2024-04-19 và sẽ trả vào ngày: 2024-12-30
DG002 đã mượn sách có mã là: MS001 vào ngày: 2024-04-19 và sẽ trả vào ngày: 2024-07-12
MS002 đã mượn sách có mã là: MS002 vào ngày: 2024-04-19 và sẽ trả vào ngày: 2024-07-12
MS003 đã mượn sách có mã là: MS003 vào ngày: 2024-04-19 và sẽ trả vào ngày: 2024-10-12
MS004 đã mượn sách có mã là: MS004 vào ngày: 2024-04-19 và sẽ trả vào ngày: 2024-11-12

Hình 16: Danh sách các độc giả đang mượn

2.6 Giao diện hiển thị form thêm nhà xuất bản

Quản Lý Nhà Xuất Bản

Thêm NXB

Mã NXB:
NXB001

Tên NXB:
Hoa Kỳ

Địa Chỉ:
USA

Thêm NXB

Hình 17: Form thêm nhà xuất bản

2.7 Giao diện hiển thị form chỉnh sửa nhà xuất bản

Quản Lý Nhà Xuất Bản

Chỉnh Sửa NXB

Mã NXB:
NXB003

Tên NXB:
Thụy Sĩ

Địa Chỉ:
Switzerland

Cập Nhật NXB

Hình 18: Form chỉnh sửa nhà xuất bản

2.8 Giao diện hiển thị danh sách các nhà xuất bản

Danh Sách Nhà Xuất Bản

Tìm kiếm...			
Mã NXB	Tên NXB	Địa Chỉ	Thao Tác
NXB001	Hoa Kỳ	USA	Chỉnh Sửa Xóa
NXB002	Pháp	France	Chỉnh Sửa Xóa
NXB003	Thụy Sĩ	Switzerland	Chỉnh Sửa Xóa
NXB004	Nhật Bản	Japan	Chỉnh Sửa Xóa

Hình 19: Danh sách các nhà xuất bản

2.9 Giao diện hiển thị form thêm độc giả

Quản Lý Độc Giả

Thêm Độc Giả

Mã Độc Giả:

Họ Tên:

Tên:

Ngày Sinh:

Phái:

Địa Chỉ:

Điện Thoại:

[Thêm Độc Giả](#)

Hình 20: Form thêm độc giả

2.10 Giao diện hiển thị form chỉnh sửa độc giả

Quản Lý Độc Giả

Chỉnh Sửa Độc Giả

Mã Độc Giả:

Họ Tên:

Tên:

Ngày Sinh:

Phái:

Địa Chỉ:

Điện Thoại:

[Cập Nhật](#)

Hình 21: Form chỉnh sửa độc giả

2.11 Giao diện hiển thị danh sách các độc giả

Danh Sách Độc Giả							
Tìm kiếm...							
Mã Độc Giả	Họ Lót	Tên	Ngày Sinh	Phái	Địa Chỉ	Điện Thoại	Thao Tác
MG001	Trương Hoàng	Duy	2002-12-30	Nam	An Giang	0987654321	<button>Chỉnh Sửa</button> <button>Xóa</button>
MG002	Hồ Thị Kim	Thoa	2002-03-29	Nu	Cần Thơ	0987654321	<button>Chỉnh Sửa</button> <button>Xóa</button>
MG003	Trương Ngọc	Yến	2002-11-23	Nu	Cần Thơ	0987654321	<button>Chỉnh Sửa</button> <button>Xóa</button>
MG004	Trần Kim	Ngọc	2002-09-02	Nu	Côn Đảo	0987654321	<button>Chỉnh Sửa</button> <button>Xóa</button>
.....	<button>Chỉnh Sửa</button> <button>Xóa</button>

Hình 22: danh sách các độc giả

TÀI LIỆU THAM KHẢO

[1] Link tham khảo các tài liệu liên quan đến HTML, CSS, JS, BOOTRASP,...
<https://www.w3schools.com/>

[2] Slides bài giảng và tài liệu hướng dẫn thực hành của thầy Nguyễn Minh Trung
Đường link github:

https://github.com/hoangduytruong13/B2007173_TRUONGHOANGDUY.git