

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐỒNG THÁP
KHOA SƯ PHẠM TOÁN - TIN



BÁO CÁO CUỐI KÌ
MÔN LẬP TRÌNH PYTHON VÀ ỨNG DỤNG
ĐỀ TÀI: 6. XÂY DỰNG TRANG WEB QUẢN LÝ THƯ VIỆN VỚI DJANGO

THÀNH VIÊN NHÓM 12 - NGÀNH: DHSTIN - LỚP: 23A

- 1. Cao Khánh Ly - 0023410532**
- 2. Võ Ngọc Thảo Nguyên - 0023411182**
- 3. Nguyễn Khánh Sơn - 0023411297**
- 4. Nguyễn Thị Vân Anh - 0023410692**
- 5. Võ Thị Anh Thư - 0023410130**

GIẢNG VIÊN HƯỚNG DẪN: TS. ĐẶNG QUỐC BẢO

Đồng Tháp, tháng 12, năm 2025

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	4
1.1. Lý do chọn đề tài.....	4
1.2. Mục tiêu của đề tài.....	4
1.3. Phạm vi nghiên cứu	5
1.4. Phân công nhiệm vụ.....	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG	8
2.1. Kiến trúc Client - Server	8
2.1.1. Khái niệm cơ bản	8
2.1.2. Cơ chế giao tiếp qua HTTP	8
2.2. Các công nghệ xây dựng Backend (Server).....	9
2.2.1. Ngôn ngữ lập trình Python	9
2.2.2. Framework Django.....	9
2.2.3. Thư viện và công cụ hỗ trợ quản lý dữ liệu trong Django.....	10
2.3. Các công nghệ xây dựng Frontend (Client)	11
2.3.1. HTML và CSS.....	11
2.3.2. Bootstrap	12
2.3.3. JavaScript.....	12
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13
3.1. Sơ đồ hoạt động (Activity Diagram)	13
3.2. Thiết kế kiến trúc hệ thống	13
3.2.1. Sơ đồ triển khai (Deployment View).....	13
3.2.2. Cơ chế giao tiếp.....	14
3.3. Thiết kế các Module chính	15
3.3.1. Module Quản lý Sách	15
3.3.2. Module Quản lý Độc giả.....	16
3.3.3. Module Quản lý Mượn/Trả	17
3.3.4. Module Thống kê	17

3.4. Đặc tả API (Application Programming Interface)	18
CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG VÀ KẾT QUẢ THỰC NGHIỆM....	21
4.1. Cài đặt môi trường và Tổ chức dự án.....	21
4.1.1. Yêu cầu phần mềm	21
4.1.2. Các thư viện phụ thuộc (Dependencies)	21
4.1.3. Cấu trúc tổ chức thư mục	22
4.2. Xây dựng các Module xử lý chính (Chi tiết kỹ thuật).....	23
4.3. Hướng dẫn vận hành (Demo)	26
4.4. Giao diện chương trình	26
4.4.1. Giao diện Trang chủ.....	26
4.4.2. Giao diện Quản lý Sách.....	27
4.4.3. Giao diện Quản lý Người dùng.....	28
4.4.4. Giao diện Mượn/Trả Sách	29
4.4.5. Giao diện Thống kê và Báo cáo	30
4.4.6. Giao diện Quản lý tác giả, thể loại, nhà xuất bản	31
4.5. Kết quả thử nghiệm (Testing).....	34
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	36
5.1. Kết luận.....	36
5.1.2. Hạn chế của đề tài.....	36
5.2. Hướng phát triển	37
CHƯƠNG 6 TÀI LIỆU THAM KHẢO.....	39

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong kỷ nguyên chuyển đổi số, việc ứng dụng công nghệ thông tin vào quản lý và vận hành các hoạt động giáo dục trở thành xu hướng tất yếu. Thư viện – vốn được xem là “trái tim” của nhà trường – không chỉ là nơi lưu trữ tri thức mà còn là công cụ hỗ trợ đắc lực cho việc học tập và nghiên cứu. Tuy nhiên, các mô hình quản lý truyền thống thường gặp nhiều hạn chế:

- Dữ liệu phân tán: thông tin sách, người dùng, và hoạt động mượn/trả thường được ghi chép thủ công, khó đồng bộ.
- Khó tra cứu: việc tìm kiếm sách hoặc kiểm tra tình trạng mượn/trả mất nhiều thời gian.
- Thiếu công cụ thống kê: không có báo cáo tổng hợp về số lượng sách, lượt mượn, hay tình trạng người dùng.
- Khó kiểm soát hoạt động: dễ xảy ra sai sót, thiếu minh bạch trong quản lý.

Trong khi đó, công nghệ web hiện đại cho phép xây dựng các hệ thống quản lý trực tuyến nhanh chóng, hiệu quả và dễ triển khai. Django – một framework Python mạnh mẽ – cung cấp đầy đủ công cụ để phát triển ứng dụng web có tính bảo mật cao, khả năng mở rộng tốt, và tích hợp dễ dàng với cơ sở dữ liệu.

Xuất phát từ thực tiễn đó, nhóm chúng tôi lựa chọn đề tài “Xây dựng trang web quản lý thư viện với Django” nhằm tạo ra một hệ thống quản lý hiện đại, thân thiện, dễ sử dụng, có khả năng mở rộng và tích hợp nhiều tính năng hữu ích. Đây không chỉ là một bài tập lớn trong môn học, mà còn là bước khởi đầu để tiếp cận các công nghệ web tiên tiến, rèn luyện kỹ năng lập trình, và đóng góp một giải pháp thiết thực cho công tác quản lý thư viện trong nhà trường.

1.2. Mục tiêu của đề tài

Về mặt công nghệ, đề tài hướng đến việc xây dựng thành công một ứng dụng Web quản lý thư viện hoạt động theo mô hình Client – Server. Hệ thống được phát triển dựa trên các công nghệ lập trình hiện đại, trong đó Django (Python) đảm nhiệm vai trò xử lý phía Backend, quản lý logic nghiệp vụ và kết nối cơ sở dữ liệu. Phần

giao diện người dùng (Frontend) được thiết kế bằng HTML, CSS, Bootstrap và JavaScript, giúp tạo nên một môi trường trực quan, thân thiện và dễ sử dụng. Dữ liệu về sách, người dùng và hoạt động mượn/trả được lưu trữ trong SQLite, một hệ quản trị cơ sở dữ liệu nhẹ, dễ triển khai và phù hợp cho ứng dụng demo. Toàn bộ hệ thống được thiết kế với khả năng mở rộng, dễ bảo trì, đồng thời có thể triển khai trên nhiều môi trường khác nhau, đáp ứng nhu cầu quản lý thư viện hiện đại và hiệu quả.

Về mặt chức năng:

- Quản lý sách: thêm, sửa, xóa, tìm kiếm, phân loại theo thể loại/tác giả.
- Quản lý người dùng: đăng ký, đăng nhập, phân quyền (người dùng thường, thủ thư, quản trị viên).
- Quản lý mượn/trả sách: ghi nhận lịch sử mượn, hạn trả, trạng thái sách.
- Thống kê và báo cáo: số lượng sách hiện có, lượt mượn theo thời gian, danh sách người dùng hoạt động.
- Giao diện thân thiện: hỗ trợ tìm kiếm nhanh, hiển thị thông tin trực quan.
- Bảo mật dữ liệu: đảm bảo thông tin người dùng và dữ liệu thư viện được quản lý an toàn.

1.3. Phạm vi nghiên cứu

Đối tượng nghiên cứu chính

- Hệ thống quản lý thư viện trong trường học: tập trung vào việc quản lý thông tin sách, người dùng (học sinh, sinh viên, giáo viên), và hoạt động mượn/trả.
- Mục tiêu: xây dựng một ứng dụng web có khả năng hỗ trợ đầy đủ các chức năng cơ bản của một thư viện hiện đại, bao gồm tra cứu, mượn/trả, thống kê và phân quyền quản lý.

Đối tượng nghiên cứu mở rộng

- Quản lý người dùng nâng cao: tích hợp phân quyền chi tiết (người dùng thường, thủ thư, quản trị viên).
- Quản lý tài liệu số: nghiên cứu khả năng tích hợp thêm module quản lý tài liệu điện tử (ebooks, tài liệu PDF).

- Mục tiêu: mở rộng phạm vi sử dụng hệ thống không chỉ cho thư viện truyền thống mà còn cho thư viện số.
- Giới hạn: trong phạm vi báo cáo này, hệ thống mới chỉ tập trung vào quản lý sách in và hoạt động mượn/trả cơ bản.

Quy mô và phương thức xử lý

- Xử lý đơn lẻ: người dùng có thể thực hiện các thao tác quản lý sách, mượn/trả từng cuốn sách riêng biệt.
- Xử lý hàng loạt: quản trị viên có thể nhập dữ liệu sách số lượng lớn (batch import) hoặc xuất báo cáo thống kê toàn bộ hoạt động mượn/trả.
- Mục tiêu: tối ưu hóa quy trình quản lý, giảm thiểu thao tác thủ công, và hỗ trợ tự động hóa trong việc tổng hợp dữ liệu.

Công cụ và ngôn ngữ

- Ngôn ngữ lập trình: Python (Backend với Django), HTML/CSS/JavaScript (Frontend).
- Framework và thư viện chính:
 - Django: framework web mạnh mẽ, hỗ trợ ORM, quản lý người dùng, và bảo mật.
 - Bootstrap: xây dựng giao diện hiện đại, responsive.
 - SQLite: cơ sở dữ liệu mặc định, nhẹ, dễ triển khai.
- IDE và công cụ phát triển: Visual Studio Code, Git để quản lý mã nguồn.

1.4. Phân công nhiệm vụ

STT	Họ và tên	Vai trò (Role)	Nhiệm vụ cụ thể (Tasks)	Phản phụ trách trong Báo cáo
1	Nguyễn Thị Vân Anh (Nhóm trưởng)	Leader	- Quản lý tiến độ chung của dự án. - Phân chia công việc cho các thành viên. - Tổng hợp báo cáo hoàn chỉnh.	- Viết Chương 1: Tổng quan. - Viết Chương 2: Cơ sở lý thuyết. - Tổng hợp và định dạng toàn bộ báo cáo.

STT	Họ và tên	Vai trò (Role)	Nhiệm vụ cụ thể (Tasks)	Phản phụ trách trong Báo cáo
			- Soạn thảo cơ sở lý thuyết.	
2	Cao Khánh Ly	Backend Developer	- Xây dựng Server xử lý chính bằng Django. - Thiết kế models quản lý sách, tác giả, thể loại, NXB. - Cấu hình cơ sở dữ liệu SQLite.	- Viết Chương 3: Phân tích và thiết kế hệ thống (Phần Backend, Database).
3	Võ Ngọc Thảo Nguyên	Frontend Developer	- Thiết kế giao diện Web bằng HTML, CSS, Bootstrap. - Kết nối giao diện với Backend Django. - Đảm bảo giao diện thân thiện, dễ sử dụng.	- Viết Chương 4: Xây dựng ứng dụng (Phần Giao diện và hướng dẫn cài đặt).
4	Nguyễn Khánh Sơn	Tester (Kiểm thử)	- Thực hiện kiểm thử các chức năng: thêm/sửa/xóa sách, mượn/trả, thống kê. - Ghi lại kết quả test và chụp ảnh minh họa. - Hỗ trợ tìm tài liệu tham khảo.	- Viết Chương 4: Kết quả thử nghiệm. - Viết Chương 5: Kết luận & Hướng phát triển.
5	Võ Thị Anh Thư	Slide Designer	- Tổng hợp nội dung cốt lõi từ báo cáo. - Thiết kế Slide PowerPoint thuyết trình. - Soạn thảo kịch bản thuyết trình cho nhóm.	- Thiết kế Slide thuyết trình. - Soạn thảo các danh mục (Hình ảnh, Từ viết tắt).

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

2.1. Kiến trúc Client - Server

2.1.1. Khái niệm cơ bản

Mô hình Client – Server (Khách – Chủ) là kiến trúc mạng máy tính nền tảng, trong đó hệ thống được phân chia thành hai thành phần chính với nhiệm vụ riêng biệt nhưng tương tác chặt chẽ:

- Client (Máy khách): Là phía người dùng cuối, giao tiếp trực tiếp với người sử dụng thông qua giao diện web. Client có nhiệm vụ gửi các yêu cầu (Request) đến Server và hiển thị kết quả (Response) trả về. Trong đề tài này, Client là ứng dụng Web được xây dựng bằng HTML, CSS, Bootstrap và JavaScript, giúp người dùng dễ dàng tra cứu sách, đăng nhập, mượn/trả và quản lý thông tin.
- Server (Máy chủ): Là trung tâm xử lý logic và lưu trữ dữ liệu. Server lắng nghe yêu cầu từ Client, thực thi các chức năng quản lý (như thêm/sửa/xóa sách, ghi nhận mượn/trả, thống kê dữ liệu) và phản hồi lại Client. Trong đề tài này, Server được phát triển bằng Django (Python), kết nối với cơ sở dữ liệu SQLite để quản lý toàn bộ thông tin của thư viện.

2.1.2. Cơ chế giao tiếp qua HTTP

Hệ thống vận hành dựa trên giao thức truyền tải siêu văn bản (HTTP), tuân theo quy trình Request – Response tiêu chuẩn:

1. Gửi yêu cầu (HTTP Request): Khi người dùng thực hiện một thao tác (ví dụ: tra cứu sách, đăng nhập, mượn sách), Client sẽ đóng gói dữ liệu vào một bản tin HTTP (thường sử dụng phương thức GET hoặc POST) và gửi tới Server.
2. Xử lý (Processing): Server tiếp nhận yêu cầu, gọi các module xử lý chuyên biệt (models, views trong Django) để thực hiện chức năng tương ứng, chẳng hạn như truy vấn cơ sở dữ liệu để tìm sách hoặc ghi nhận thông tin mượn/trả.

3. Phản hồi (HTTP Response): Sau khi xử lý xong, Server trả về kết quả dưới dạng trang HTML hoặc dữ liệu JSON cho Client. Client sẽ hiển thị kết quả này trên giao diện người dùng, hoàn tất một chu trình xử lý.

2.2. Các công nghệ xây dựng Backend (Server)

2.2.1. Ngôn ngữ lập trình Python

Nhóm thực hiện lựa chọn Python làm ngôn ngữ chủ đạo phía Server nhờ các ưu điểm nổi bật:

- Hệ sinh thái mạnh mẽ: Python sở hữu kho thư viện mã nguồn mở phong phú, hỗ trợ tối đa cho việc phát triển ứng dụng web. Đặc biệt, framework Django cung cấp sẵn nhiều module giúp xây dựng hệ thống quản lý thư viện nhanh chóng, từ quản lý cơ sở dữ liệu, xác thực người dùng, đến bảo mật.
- Hiệu suất xử lý ổn định: Python có khả năng xử lý tốt các tác vụ I/O và logic nghiệp vụ phức tạp. Trong hệ thống quản lý thư viện, Python đảm nhiệm việc truy vấn dữ liệu sách, ghi nhận hoạt động mượn/trả, và trả kết quả cho người dùng một cách nhanh chóng.
- Cú pháp tường minh, dễ học: Python có cú pháp rõ ràng, dễ đọc, giúp giảm thiểu lỗi lập trình và tăng tốc độ phát triển. Điều này đặc biệt hữu ích cho nhóm sinh viên khi làm việc nhóm, vì mã nguồn dễ hiểu và dễ bảo trì.
- Khả năng mở rộng và tích hợp: Python dễ dàng tích hợp với nhiều công nghệ khác như cơ sở dữ liệu (SQLite, PostgreSQL), các thư viện giao diện web, và công cụ thống kê. Nhờ đó, hệ thống quản lý thư viện có thể mở rộng thêm nhiều tính năng trong tương lai.

2.2.2. Framework Django

Django là một framework web mã nguồn mở được phát triển bằng ngôn ngữ Python, được lựa chọn nhờ các đặc tính vượt trội trong việc xây dựng ứng dụng quản lý thư viện:

- Phát triển nhanh (Rapid Development): Django cung cấp sẵn nhiều công cụ và module, giúp giảm thiểu đáng kể lượng code phải viết. Các chức năng như quản lý người dùng, xác thực, và hệ thống quản trị (Admin site) được tích hợp sẵn, hỗ trợ nhóm phát triển triển khai ứng dụng nhanh chóng.

- **Bảo mật cao (Security):** Django được thiết kế với nhiều cơ chế bảo mật mặc định, như chống SQL Injection, XSS, CSRF. Điều này đảm bảo dữ liệu người dùng và thông tin thư viện được bảo vệ an toàn.
- **ORM mạnh mẽ (Object Relational Mapping):** Django ORM cho phép thao tác cơ sở dữ liệu bằng Python thay vì viết câu lệnh SQL thủ công. Nhờ đó, việc quản lý dữ liệu sách, người dùng, và hoạt động mượn/trả trở nên dễ dàng, đồng thời hỗ trợ nhiều loại cơ sở dữ liệu khác nhau (SQLite, PostgreSQL, MySQL).
- **Khả năng mở rộng (Scalability):** Django có kiến trúc linh hoạt, dễ dàng mở rộng để tích hợp thêm các tính năng như thống kê nâng cao, quản lý tài liệu số, hoặc kết nối với hệ thống bên ngoài.
- **Cộng đồng lớn và tài liệu phong phú:** Django có cộng đồng phát triển mạnh mẽ, tài liệu chính thức chi tiết, và nhiều thư viện bổ trợ, giúp nhóm dễ dàng tìm kiếm giải pháp khi gặp vấn đề.

2.2.3. Thư viện và công cụ hỗ trợ quản lý dữ liệu trong Django

Trong quá trình xây dựng hệ thống quản lý thư viện, nhóm sử dụng một số thư viện và công cụ quan trọng nhằm hỗ trợ việc xử lý dữ liệu, giao diện và bảo mật:

- **Django ORM (Object Relational Mapping)**
 - **Chức năng:** Cho phép thao tác cơ sở dữ liệu bằng Python thay vì viết câu lệnh SQL thủ công.
 - **Cơ chế:** Tự động ánh xạ các lớp (models) trong Python thành bảng trong cơ sở dữ liệu, giúp việc quản lý dữ liệu sách, người dùng, và hoạt động mượn/trả trở nên dễ dàng và nhất quán.
- **Django Authentication & Authorization**
 - **Chức năng:** Quản lý người dùng, đăng nhập, đăng xuất, phân quyền.
 - **Cơ chế:** Cung cấp sẵn hệ thống xác thực và phân quyền, cho phép phân biệt giữa người dùng thường, thủ thư, và quản trị viên.
- **Django Admin**

- Chức năng: Giao diện quản trị mặc định để quản lý dữ liệu.
- Cơ chế: Tự động sinh giao diện quản trị từ các models, giúp quản trị viên dễ dàng thêm/sửa/xóa sách, quản lý người dùng và theo dõi hoạt động mượn/trả.
- Bootstrap (Frontend)
 - Chức năng: Hỗ trợ thiết kế giao diện web hiện đại, responsive.
 - Cơ chế: Cung cấp sẵn các lớp CSS và thành phần UI (form, button, table) để xây dựng giao diện nhanh chóng, đảm bảo hiển thị tốt trên nhiều loại thiết bị.
- SQLite (Database)
 - Chức năng: Lưu trữ dữ liệu sách, người dùng, và lịch sử mượn/trả.
 - Cơ chế: Là cơ sở dữ liệu nhẹ, tích hợp sẵn trong Django, phù hợp cho ứng dụng demo và triển khai nhanh.
- Các thư viện hỗ trợ khác
 - django-crispy-forms: Tùy biến giao diện form nhập liệu.
 - django-filter: Hỗ trợ lọc và tìm kiếm dữ liệu sách theo nhiều tiêu chí.
 - django-rest-framework (DRF): Xây dựng API RESTful để kết nối với các ứng dụng khác hoặc mở rộng hệ thống.

2.3. Các công nghệ xây dựng Frontend (Client)

2.3.1. HTML và CSS

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu chuẩn để xây dựng cấu trúc của trang web. CSS (Cascading Style Sheets) được sử dụng để định dạng và thiết kế giao diện. Trong đề tài này, HTML và CSS đóng vai trò nền tảng để xây dựng các trang như: trang chủ, trang đăng nhập, trang quản lý sách, và trang mượn/trả.

- HTML: định nghĩa bố cục, các thành phần như bảng, form, button.

- CSS: tạo phong cách hiển thị, màu sắc, font chữ, và bố cục responsive.

2.3.2. Bootstrap

Bootstrap là framework CSS phổ biến, hỗ trợ thiết kế giao diện hiện đại và tương thích trên nhiều loại thiết bị.

- Responsive Design: giao diện tự động điều chỉnh theo kích thước màn hình (máy tính, máy tính bảng, điện thoại).
- Thành phần UI có sẵn: cung cấp nhiều thành phần như navbar, modal, form, button, giúp tiết kiệm thời gian thiết kế.
- Tích hợp dễ dàng với Django: Bootstrap có thể được nhúng trực tiếp vào template HTML của Django, giúp xây dựng giao diện nhanh chóng.

2.3.3. JavaScript

JavaScript là ngôn ngữ lập trình phía Client, giúp tăng tính tương tác cho ứng dụng web.

- Xử lý sự kiện: hỗ trợ các thao tác như xác thực form nhập liệu, hiển thị thông báo khi người dùng mượn/trả sách.
- Tương tác động: cập nhật nội dung trang mà không cần tải lại toàn bộ (ví dụ: hiển thị danh sách sách theo từ khóa tìm kiếm).
- Kết nối với API: JavaScript có thể gửi yêu cầu AJAX đến Server Django để lấy dữ liệu mà không cần reload trang.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Sơ đồ hoạt động (Activity Diagram)

Quy trình nghiệp vụ của hệ thống quản lý thư viện được thiết kế tối giản nhằm đảm bảo trải nghiệm người dùng nhanh chóng và thuận tiện nhất. Luồng hoạt động bao gồm các bước tuần tự sau:

1. **Truy cập hệ thống:** Người dùng mở trình duyệt web và truy cập vào địa chỉ ứng dụng quản lý thư viện.
2. **Đăng nhập/Đăng ký:** Người dùng nhập thông tin tài khoản để đăng nhập. Nếu chưa có tài khoản, họ có thể đăng ký mới.
3. **Tra cứu sách:** Người dùng sử dụng chức năng tìm kiếm để tra cứu sách theo tên, tác giả, hoặc thể loại.
4. **Thực hiện mượn sách:** Người dùng chọn sách mong muốn và gửi yêu cầu mượn. Hệ thống kiểm tra tình trạng sách (còn hay đã mượn) trước khi ghi nhận.
5. **Quản lý trả sách:** Khi đến hạn, người dùng trả sách. Hệ thống cập nhật trạng thái sách và ghi nhận lịch sử trả.
6. **Quản trị viên quản lý dữ liệu:** Quản trị viên có thể thêm/sửa/xóa sách, quản lý người dùng, và xem báo cáo thống kê.
7. **Thống kê và báo cáo:** Hệ thống tự động tổng hợp dữ liệu về số lượng sách, lượt mượn, và hoạt động người dùng để phục vụ công tác quản lý.

3.2. Thiết kế kiến trúc hệ thống

Hệ thống được xây dựng dựa trên mô hình kiến trúc Client – Server (Khách – Chủ) tách biệt, đảm bảo tính độc lập, dễ dàng bảo trì và mở rộng.

3.2.1. Sơ đồ triển khai (Deployment View)

- **Phía Frontend (Client):**
 - Được xây dựng bằng HTML, CSS, Bootstrap và JavaScript, kết hợp với hệ thống template của Django.

- Môi trường vận hành: Trình duyệt Web của người dùng (Chrome, Firefox, Edge...).
- Địa chỉ triển khai (Local): `http://127.0.0.1:8000` (do Django tích hợp cả frontend và backend trong cùng một server).
- **Phía Backend (Server):**
 - Được xây dựng bằng Python và Django Framework.
 - Nhiệm vụ: Xử lý logic nghiệp vụ, quản lý dữ liệu sách, tác giả, thể loại, nhà xuất bản, và hoạt động mượn/trả.
 - Môi trường vận hành: Máy chủ ứng dụng (local hoặc triển khai trên cloud như Heroku, AWS, DigitalOcean).
 - Địa chỉ triển khai (Local): `http://127.0.0.1:8000`.
- **Phía Database (Cơ sở dữ liệu):**
 - Sử dụng SQLite làm hệ quản trị cơ sở dữ liệu mặc định.
 - Nhiệm vụ: Lưu trữ toàn bộ dữ liệu về sách, tác giả, thể loại, nhà xuất bản, và lịch sử mượn/trả.
 - Khả năng mở rộng: Có thể chuyển sang PostgreSQL hoặc MySQL khi triển khai thực tế với quy mô lớn.

3.2.2. Cơ chế giao tiếp

Hai thành phần Client và Server giao tiếp với nhau thông qua chuẩn RESTful API sử dụng giao thức HTTP.

- **Request (Yêu cầu):** Người dùng thao tác trên giao diện web (ví dụ: thêm sách, mượn sách, trả sách, tra cứu). Frontend gửi dữ liệu thông qua phương thức GET hoặc POST tới các địa chỉ API của Backend (Django).
 - Ví dụ: GET `/books` để lấy danh sách sách.
 - Ví dụ: POST `/borrow` để ghi nhận yêu cầu mượn sách.
- **Processing (Xử lý):** Server (Django) tiếp nhận yêu cầu, gọi các views và models để xử lý logic nghiệp vụ. Hệ thống sẽ truy vấn cơ sở dữ liệu (SQLite) để lấy thông tin sách, tác giả, hoặc ghi nhận hoạt động mượn/trả.

- **Response (Phản hồi):** Sau khi xử lý xong, Server trả về kết quả dưới dạng HTML (render template) hoặc JSON (API response). Client sẽ hiển thị kết quả này trên giao diện cho người dùng.
- **CORS Policy (nếu tách riêng Frontend – Backend):** Trong trường hợp hệ thống mở rộng với frontend độc lập (SPA), cần cấu hình chính sách chia sẻ tài nguyên chéo (Cross-Origin Resource Sharing) để cho phép frontend gọi dữ liệu từ backend.
 - Ví dụ: Frontend chạy ở `http://localhost:3000` có thể gọi API từ Backend chạy ở `http://127.0.0.1:8000`.

3.3. Thiết kế các Module chính

Hệ thống quản lý thư viện được thiết kế theo hướng mô-đun hóa, tức là chia thành các thành phần chức năng độc lập nhưng liên kết chặt chẽ với nhau. Cách tiếp cận này giúp đảm bảo quy trình xử lý dữ liệu khép kín, từ khâu tiếp nhận yêu cầu của người dùng, xử lý dữ liệu ở phía máy chủ, cho đến việc phản hồi kết quả. Trong phạm vi đề tài, nhóm tập trung xây dựng bốn module cốt lõi: quản lý sách, quản lý độc giả, quản lý mượn/trả, và thống kê. Đây là bốn mắt xích quan trọng tạo nên luồng hoạt động hoàn chỉnh của hệ thống.

3.3.1. Module Quản lý Sách

Đây là module trung tâm của hệ thống, chịu trách nhiệm lưu trữ và quản lý thông tin sách trong thư viện. Người dùng có thể thêm mới, chỉnh sửa, xóa và tra cứu sách theo nhiều tiêu chí.

- **Chức năng:**
 - Thêm sách mới vào hệ thống.
 - Chỉnh sửa thông tin sách.
 - Xóa sách khỏi cơ sở dữ liệu.
 - Tìm kiếm sách theo tên, tác giả, thể loại.
- **Cơ chế hoạt động:**
 - Giao diện hiển thị danh sách sách dưới dạng bảng.
 - Form nhập liệu sử dụng Django Form để kiểm tra hợp lệ.

- Dữ liệu được lưu vào model Book và quản lý qua Django Admin.

- **Minh họa code (models.py):**

Python

```
class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=100)
    category = models.CharField(max_length=50)
    publisher = models.CharField(max_length=100)
    publish_year = models.IntegerField()
    quantity = models.IntegerField()
```

3.3.2. Module Quản lý Độc giả

Module này cho phép quản lý thông tin người dùng của hệ thống, bao gồm học sinh, sinh viên, giáo viên. Việc lưu trữ và phân quyền giúp đảm bảo tính bảo mật và minh bạch trong quản lý.

- **Chức năng:**

- Thêm mới độc giả.
- Chỉnh sửa thông tin cá nhân.
- Xóa độc giả.
- Phân quyền người dùng (thường, thủ thư, quản trị viên).

- **Cơ chế hoạt động:**

- Sử dụng hệ thống xác thực của Django (User, Group, Permission).
- Giao diện quản lý thông tin cá nhân và lịch sử mượn.

- **Minh họa code (models.py):**

python

```
class Borrower(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    phone = models.CharField(max_length=15)
    address = models.TextField()
```


3.3.3. Module Quản lý Mượn/Trả

Đây là module xử lý nghiệp vụ mượn và trả sách. Nó ghi nhận các giao dịch, theo dõi trạng thái sách, và cảnh báo khi độc giả trả sách quá hạn.

- **Chức năng:**
 - Ghi nhận lượt mượn sách.
 - Cập nhật trạng thái trả sách.
 - Tính toán hạn trả và cảnh báo quá hạn.
- **Cơ chế hoạt động:**
 - Khi người dùng mượn sách, hệ thống kiểm tra số lượng còn lại.
 - Khi trả sách, hệ thống cập nhật trạng thái và ghi nhận lịch sử.
- **Minh họa code (models.py):**

python

```
class Loan(models.Model):  
    borrower = models.ForeignKey(Borrower, on_delete=models.CASCADE)  
    book = models.ForeignKey(Book, on_delete=models.CASCADE)  
    loan_date = models.DateField(auto_now_add=True)  
    due_date = models.DateField()  
    returned = models.BooleanField(default=False)
```

3.3.4. Module Thống kê

Để hỗ trợ công tác quản lý, hệ thống cung cấp module thống kê, cho phép truy xuất dữ liệu tổng hợp về sách, độc giả và lượt mượn. Kết quả được hiển thị trực quan bằng biểu đồ.

- **Chức năng:**
 - Thống kê số lượng sách theo thể loại.
 - Thống kê lượt mượn theo tháng.
 - Thống kê số lượng độc giả đang hoạt động.
- **Cơ chế hoạt động:**

- Truy vấn dữ liệu từ các bảng Book, Loan, Borrower.
- Hiện thị biểu đồ bằng thư viện Chart.js hoặc Django template.

- **Minh họa code (models.py):**

```
def stats_view(request):
```

```
    book_count = Book.objects.count()
```

```
    loan_count = Loan.objects.filter(returned=False).count()
```

```
    active_users = Borrower.objects.count()
```

```
    return render(request, 'stats.html', {
```

```
        'book_count': book_count,
```

```
        'loan_count': loan_count,
```

```
        'active_users': active_users
```

```
    })
```

3.4. Đặc tả API (Application Programming Interface)

Để hệ thống quản lý thư viện có thể giao tiếp hiệu quả giữa Frontend và Backend, nhóm đã thiết kế API theo chuẩn RESTful. Các API này đóng vai trò là cầu nối, cho phép Client gửi yêu cầu xử lý dữ liệu và nhận kết quả từ Server. Trong phạm vi đề tài, nhóm tập trung xây dựng các endpoint chính phục vụ cho việc quản lý sách, đọc giả, mượn/trả và thống kê dữ liệu.

Phương thức	Đường dẫn (URL)	Chức năng	Tham số đầu vào (Body)	Kết quả trả về
GET	/books	Lấy danh sách sách	Không có	Danh sách sách (JSON)
POST	/books	Thêm sách mới	- title, author, category, publisher, publish_year, quantity	Thông tin sách vừa thêm
PUT	/books/{id}	Cập nhật thông tin sách	Các trường cần chỉnh sửa	Thông tin sách sau khi cập nhật

Phương thức	Đường dẫn (URL)	Chức năng	Tham số đầu vào (Body)	Kết quả trả về
DELETE	/books/{id}	Xóa sách	Không có	Thông báo xóa thành công
GET	/borrowers	Lấy danh sách độc giả	Không có	Danh sách độc giả (JSON)
POST	/borrowers	Thêm độc giả mới	- name, email, phone, address	Thông tin độc giả vừa thêm
PUT	/borrowers/{id}	Cập nhật thông tin độc giả	Các trường cần chỉnh sửa	Thông tin độc giả sau khi cập nhật
DELETE	/borrowers/{id}	Xóa độc giả	Không có	Thông báo xóa thành công
GET	/loans	Lấy danh sách lượt mượn	Không có	Danh sách lượt mượn (JSON)
POST	/loans	Ghi nhận lượt mượn mới	- borrower_id, book_id, loan_date, due_date	Thông tin lượt mượn vừa thêm
PUT	/loans/{id}	Cập nhật trạng thái trả	- returned: true/false	Thông tin lượt mượn sau khi cập nhật
DELETE	/loans/{id}	Xóa lượt mượn	Không có	Thông báo xóa thành công
GET	/stats/books	Thống kê số lượng sách	Không có	Số lượng sách theo thể loại
GET	/stats/loans	Thống kê lượt mượn	Không có	Lượt mượn theo tháng

Phương thức	Đường dẫn (URL)	Chức năng	Tham số đầu vào (Body)	Kết quả trả về
GET	/stats/borrowers	Thông kê độc giả	Không có	Số lượng độc giả theo trạng thái

CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG VÀ KẾT QUẢ THỰC NGHIỆM

4.1. Cài đặt môi trường và Tổ chức dự án

Để triển khai hệ thống quản lý thư viện bằng Django, nhóm cần chuẩn bị đầy đủ môi trường phát triển, bao gồm hệ điều hành, ngôn ngữ lập trình, công cụ soạn thảo và các thư viện phụ thuộc. Việc chuẩn bị kỹ lưỡng giúp đảm bảo hệ thống hoạt động ổn định, dễ dàng mở rộng và thuận tiện cho việc bảo trì.

4.1.1. Yêu cầu phần mềm

Để xây dựng và vận hành hệ thống, máy tính cần được cài đặt các công cụ và môi trường sau:

- Hệ điều hành: Windows 10/11 (hoặc macOS/Linux).

- Ngôn ngữ lập trình:

Python (phiên bản 3.9 trở lên): để chạy Backend với Django.

- Công cụ soạn thảo mã nguồn (IDE): Visual Studio Code (VS Code).

- Trình duyệt web: Google Chrome hoặc Microsoft Edge để chạy và kiểm thử giao diện.

4.1.2. Các thư viện phụ thuộc (Dependencies)

Các thư viện phụ thuộc đóng vai trò quan trọng trong việc hỗ trợ lập trình viên xây dựng hệ thống nhanh chóng và hiệu quả. Chúng bao gồm các gói phục vụ cho Backend (Django, ORM, bảo mật) và Frontend (Bootstrap, Chart.js)

a. Phía Server (File requirements.txt)

- Django: Framework chính để xây dựng hệ thống quản lý thư viện.
- djangorestframework: Hỗ trợ xây dựng API RESTful cho việc giao tiếp giữa Client và Server.
- django-cors-headers: Cho phép chia sẻ tài nguyên chéo (CORS) giữa Frontend và Backend.
- sqlite3 (tích hợp sẵn): Cơ sở dữ liệu mặc định, nhẹ, dễ triển khai.
- Pillow: Thư viện xử lý ảnh, hỗ trợ upload ảnh bìa sách.

- python-decouple: Quản lý biến môi trường, tăng tính bảo mật.

b. Phía Client (File package.json)

- bootstrap: Framework CSS giúp thiết kế giao diện hiện đại, nhất quán và tương thích với nhiều loại màn hình (Responsive).
- chart.js: Thư viện vẽ biểu đồ, phục vụ cho chức năng thống kê dữ liệu mượn/trả.
- axios: Thư viện JavaScript hỗ trợ gửi request đến API của Django.

4.1.3. Cấu trúc tổ chức thư mục

Để đảm bảo hệ thống quản lý thư viện được xây dựng khoa học, dễ bảo trì và mở rộng, nhóm đã tổ chức dự án theo cấu trúc thư mục chuẩn của Django. Việc phân chia rõ ràng giữa các thành phần như models, views, templates và static giúp lập trình viên dễ dàng phát triển, kiểm thử và triển khai

```

LibraryManagement/          # Thư mục gốc của dự án
|
|
|— library_management/      # Cấu hình chính của dự án Django
|   |— __init__.py          # Khởi tạo package
|   |— settings.py          # File cấu hình dự án (Database, Installed Apps...)
|   |— urls.py              # Định tuyến URL tổng thể
|   |— wsgi.py              # Cấu hình WSGI cho triển khai
|   └— asgi.py              # Cấu hình ASGI cho server bất đồng bộ
|
|— catalog/                 # Ứng dụng chính quản lý thư viện
|   |— migrations/          # Lưu trữ các file migration của database
|   |— templates/           # Giao diện HTML (Django Template)
|   └— base.html             # Template nền

```



```

| | └─ book_list.html      # Trang danh sách sách
| | └─ book_detail.html    # Trang chi tiết sách
| | └─ borrower_list.html  # Trang danh sách độc giả
| | └─ loan_list.html      # Trang danh sách mượn/trả
| └─ static/               # Tài nguyên tĩnh (CSS, JS, hình ảnh)
|   └─ css/                # File CSS
|   └─ js/                 # File JavaScript
|   └─ models.py           # Định nghĩa các bảng dữ liệu (Book, Borrower,
Loan...)
|   └─ views.py            # Xử lý logic hiển thị và điều hướng
|   └─ urls.py             # Định tuyến URL cho ứng dụng catalog
|   └─ forms.py            # Khai báo các Form nhập liệu
|   └─ admin.py            # Tích hợp với Django Admin
|
└─ manage.py               # File quản lý dự án (chạy server, migrate...)
└─ db.sqlite3              # Cơ sở dữ liệu SQLite
└─ README.md               # Tài liệu hướng dẫn sử dụng dự án

```

4.2. Xây dựng các Module xử lý chính (Chi tiết kỹ thuật)

Để hệ thống quản lý thư viện hoạt động hiệu quả, nhóm đã xây dựng các module xử lý chính. Mỗi module đảm nhận một vai trò riêng biệt trong quy trình quản lý dữ liệu, từ việc tiếp nhận thông tin, xử lý nghiệp vụ, cho đến phản hồi kết quả cho người dùng. Các module này được thiết kế theo mô hình MVC của Django, đảm bảo tính tách biệt, dễ bảo trì và mở rộng.

a. Module Quản lý Sách (catalog/models.py, views.py, templates/book_list.html)

Đây là module quan trọng nhất, chịu trách nhiệm quản lý toàn bộ dữ liệu sách trong thư viện. Nó cho phép thêm mới, chỉnh sửa, xóa và tìm kiếm sách.

- Chức năng:
 - Thêm, sửa, xóa sách.
 - Hiển thị danh sách sách.
 - Tìm kiếm theo tên, tác giả, thể loại.
- Chi tiết kỹ thuật:
 - Model Book: định nghĩa các trường title, author, category, publisher, publish_year, quantity.
 - View book_list: lấy dữ liệu từ database và render ra template.
 - Template book_list.html: hiển thị danh sách sách, có nút thêm/sửa/xóa.

b. Module Quản lý Độc giả (catalog/models.py, views.py, templates/borrower_list.html)

Module này cho phép quản lý thông tin độc giả, bao gồm thêm mới, chỉnh sửa, xóa và theo dõi lịch sử mượn trả. Đây là nền tảng để đảm bảo hoạt động mượn/trả diễn ra minh bạch.

- Chức năng:
 - Thêm, sửa, xóa độc giả.
 - Quản lý thông tin cá nhân (họ tên, email, số điện thoại, địa chỉ).
 - Theo dõi lịch sử mượn trả.
- Chi tiết kỹ thuật:
 - Model Borrower: định nghĩa các trường name, email, phone, address.
 - View borrower_list: hiển thị danh sách độc giả.
 - Template borrower_list.html: giao diện quản lý độc giả.

c. Module Quản lý Mượn/Trả (catalog/models.py, views.py, templates/loan_list.html)

ây là module xử lý nghiệp vụ mượn và trả sách. Nó ghi nhận các giao dịch, theo dõi trạng thái sách, và cảnh báo khi độc giả trả sách quá hạn.

- Chức năng:
 - Ghi nhận lượt mượn mới.
 - Cập nhật trạng thái trả.
 - Cảnh báo quá hạn.
- Chi tiết kỹ thuật:
 - Model Loan: định nghĩa các trường borrower, book, loan_date, due_date, returned.
 - View loan_list: hiển thị danh sách lượt mượn.
 - Template loan_list.html: giao diện quản lý mượn/trả.

d. Module Thống kê (catalog/views.py, templates/stats.html)

Để hỗ trợ công tác quản lý, hệ thống cung cấp module thống kê, cho phép truy xuất dữ liệu tổng hợp về sách, độc giả và lượt mượn. Kết quả được hiển thị trực quan bằng biểu đồ.

- Chức năng:
 - Thống kê số lượng sách theo thể loại.
 - Thống kê lượt mượn theo tháng.
 - Thống kê số lượng độc giả.
- Chi tiết kỹ thuật:
 - View stats: xử lý dữ liệu thống kê từ database.
 - Template stats.html: hiển thị biểu đồ bằng Chart.js..

4.3. Hướng dẫn vận hành (Demo)

Để hệ thống quản lý thư viện có thể chạy ổn định, người dùng cần khởi động môi trường Backend và truy cập giao diện Frontend. Các bước dưới đây mô tả quy trình vận hành cơ bản để khởi chạy ứng dụng.

- Bước 1: Khởi chạy Backend (Django Server)

```
cd Library-Management  
cd Backend
```

```
python manage.py runserver
```

Kết quả: Server hoạt động tại địa chỉ <http://127.0.0.1:8000>.

- Bước 2: Truy cập giao diện người dùng Mở trình duyệt web (Chrome/Edge) và nhập địa chỉ <http://127.0.0.1:8000>. Kết quả: Giao diện ứng dụng hiển thị, cho phép người dùng thao tác với các chức năng quản lý thư viện.

4.4. Giao diện chương trình

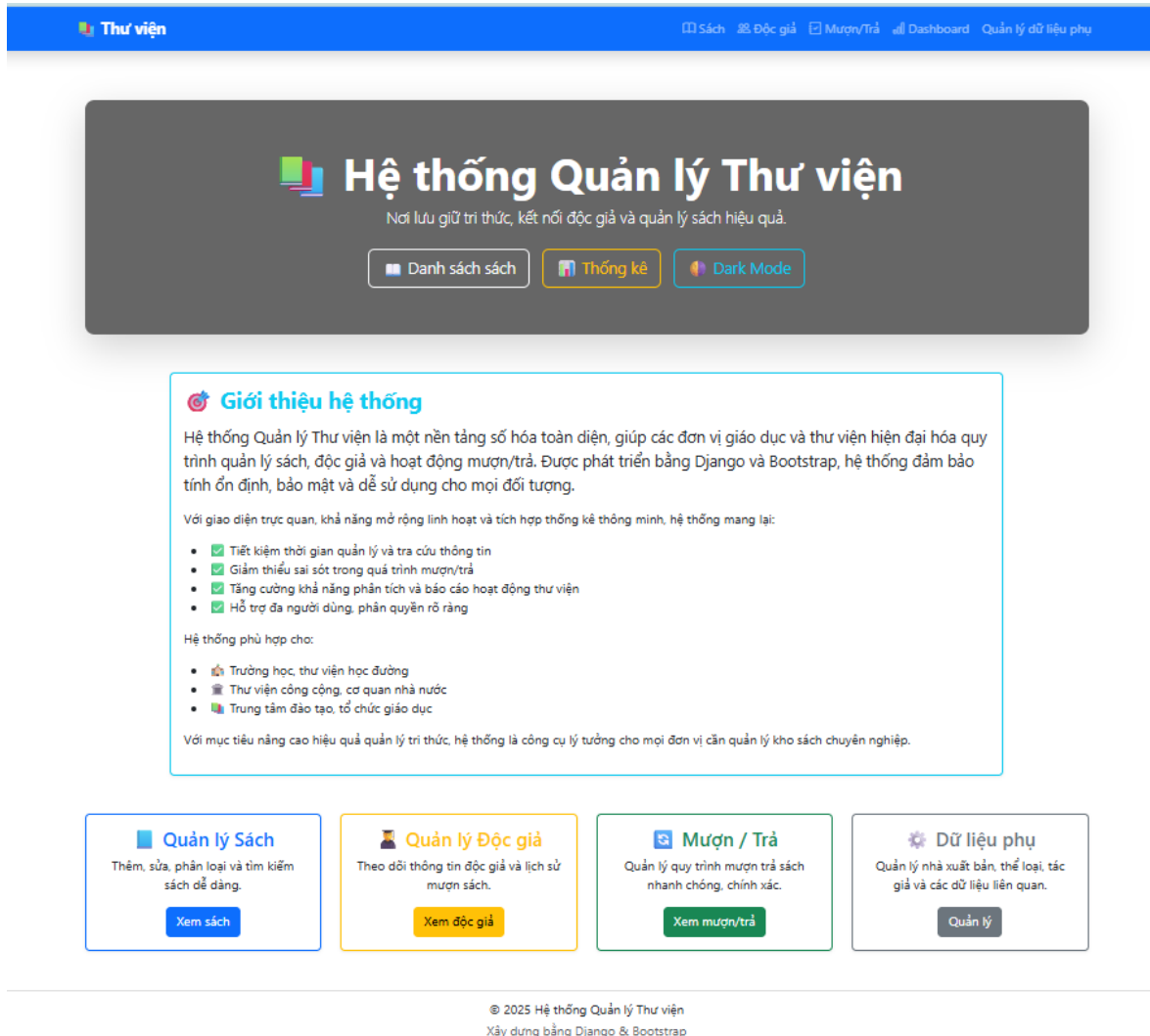
Giao diện ứng dụng được thiết kế theo phong cách hiện đại (Modern UI), sử dụng tông màu chủ đạo xanh – trắng tạo cảm giác chuyên nghiệp và thân thiện. Bố cục được chia thành các mục chức năng chính, giúp người dùng dễ dàng thao tác và quản lý dữ liệu.

- Trang chủ: hiển thị tổng quan hệ thống, menu điều hướng.
- Danh sách sách: hiển thị toàn bộ sách trong thư viện, có nút thêm/sửa/xóa.
- Danh sách độc giả: quản lý thông tin độc giả.
- Mượn/Trả sách: ghi nhận giao dịch mượn và trả.
- Thống kê: hiển thị biểu đồ trực quan về số lượng sách, lượt mượn.

4.4.1. Giao diện Trang chủ

- **Bố cục:** Thanh điều hướng (Navbar) phía trên gồm các mục: Trang chủ, Sách, Người dùng, Mượn/Trả, Thống kê.
- **Nội dung chính:** Hiển thị thông tin tổng quan về thư viện, số lượng sách hiện có, số lượt mượn gần đây.

- **Chức năng:** Người dùng có thể truy cập nhanh đến các module chính từ các nút chức năng (button) trên trang chủ.

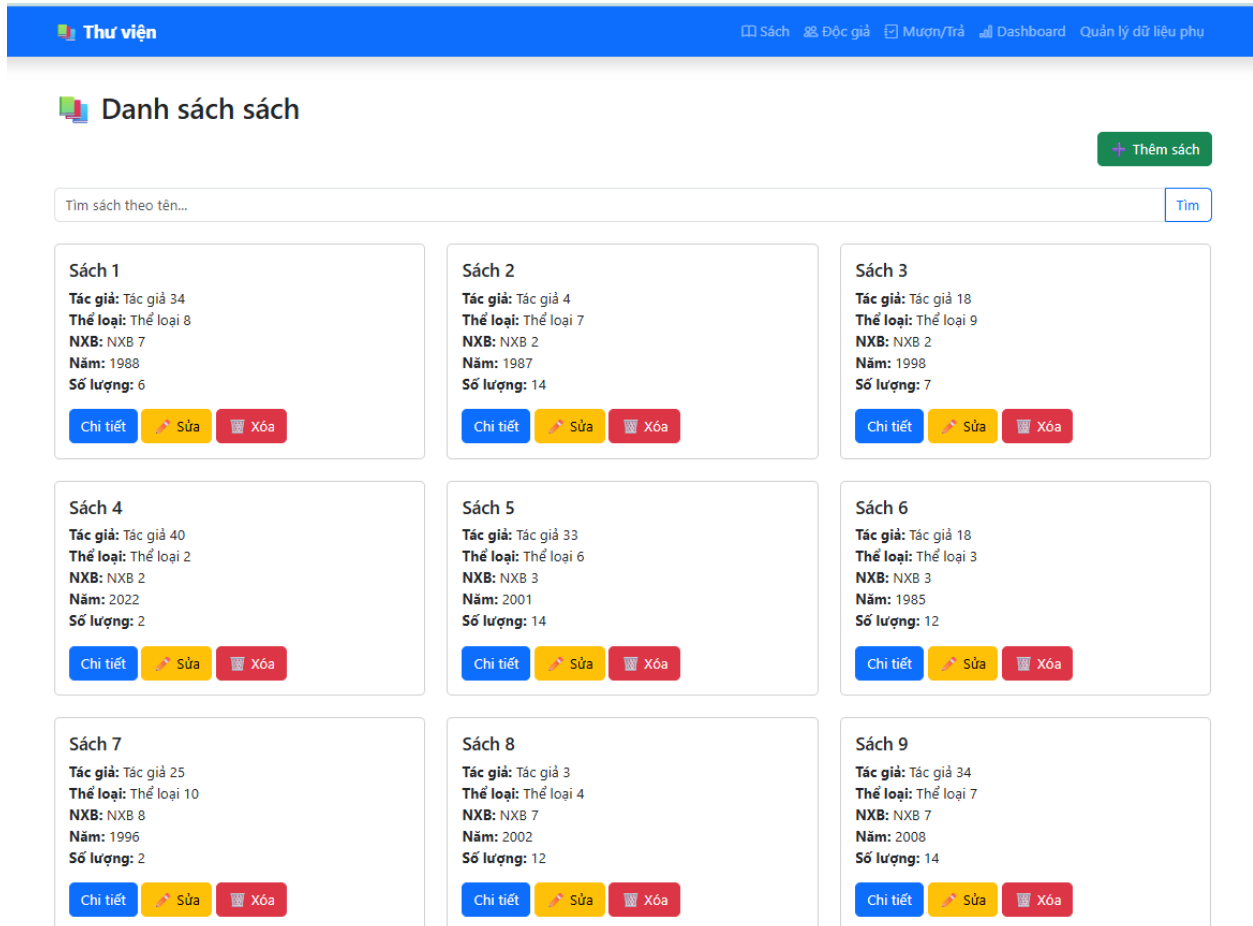


Hình 4.1: Giao diện Trang chủ.

4.4.2. Giao diện Quản lý Sách

- **Danh sách sách:** Hiển thị bảng gồm các cột: Mã sách, Tên sách, Tác giả, Thể loại, Năm xuất bản, Số lượng.
- **Chức năng:**
 - Tìm kiếm sách theo tên, tác giả hoặc thể loại.

- Thêm sách mới (form nhập liệu).
 - Sửa thông tin sách.
 - Xóa sách khỏi hệ thống.
- **Thiết kế:** Sử dụng bảng Bootstrap với chức năng phân trang (pagination) để hiển thị danh sách dài.



Hình 4.2: Giao diện Quản lý Sách.

4.4.3. Giao diện Quản lý Người dùng

- **Danh sách người dùng:** Hiển thị bảng gồm các cột: ID, Tên đăng nhập, Email, Vai trò (User, Librarian, Admin).
- **Chức năng:**
 - Thêm độc giả mới.

- Sửa thông tin.
- Xóa độc giả.
- **Thiết kế:** Form đơn giản dễ thao tác.

Thư viện				Sách	Độc giả	Mượn/Trả	Dashboard	Quản lý dữ liệu phụ
Danh sách độc giả								+ Thêm độc giả
Tên	Email	SĐT	Hành động					
Độc giả 1	reader1@example.com	0900000001	Sửa	Xóa				
Độc giả 2	reader2@example.com	0900000002	Sửa	Xóa				
Độc giả 3	reader3@example.com	0900000003	Sửa	Xóa				
Độc giả 4	reader4@example.com	0900000004	Sửa	Xóa				
Độc giả 5	reader5@example.com	0900000005	Sửa	Xóa				
Độc giả 6	reader6@example.com	0900000006	Sửa	Xóa				
Độc giả 7	reader7@example.com	0900000007	Sửa	Xóa				
Độc giả 8	reader8@example.com	0900000008	Sửa	Xóa				
Độc giả 9	reader9@example.com	0900000009	Sửa	Xóa				
Độc giả 10	reader10@example.com	0900000010	Sửa	Xóa				
Độc giả 11	reader11@example.com	0900000011	Sửa	Xóa				
Độc giả 12	reader12@example.com	0900000012	Sửa	Xóa				
Độc giả 13	reader13@example.com	0900000013	Sửa	Xóa				
Độc giả 14	reader14@example.com	0900000014	Sửa	Xóa				
Độc giả 15	reader15@example.com	0900000015	Sửa	Xóa				
Độc giả 16	reader16@example.com	0900000016	Sửa	Xóa				

Hình 4.3: Giao diện Quản lý Người dùng.

4.4.4. Giao diện Mượn/Trả Sách

- **Danh sách mượn:** Hiện thị bảng gồm các cột: Người mượn, Tên sách, Ngày mượn, Hạn trả, Trạng thái.
- **Chức năng:**
 - Người dùng gửi yêu cầu mượn sách.
 - Hệ thống kiểm tra tình trạng sách (còn hay đã mượn).

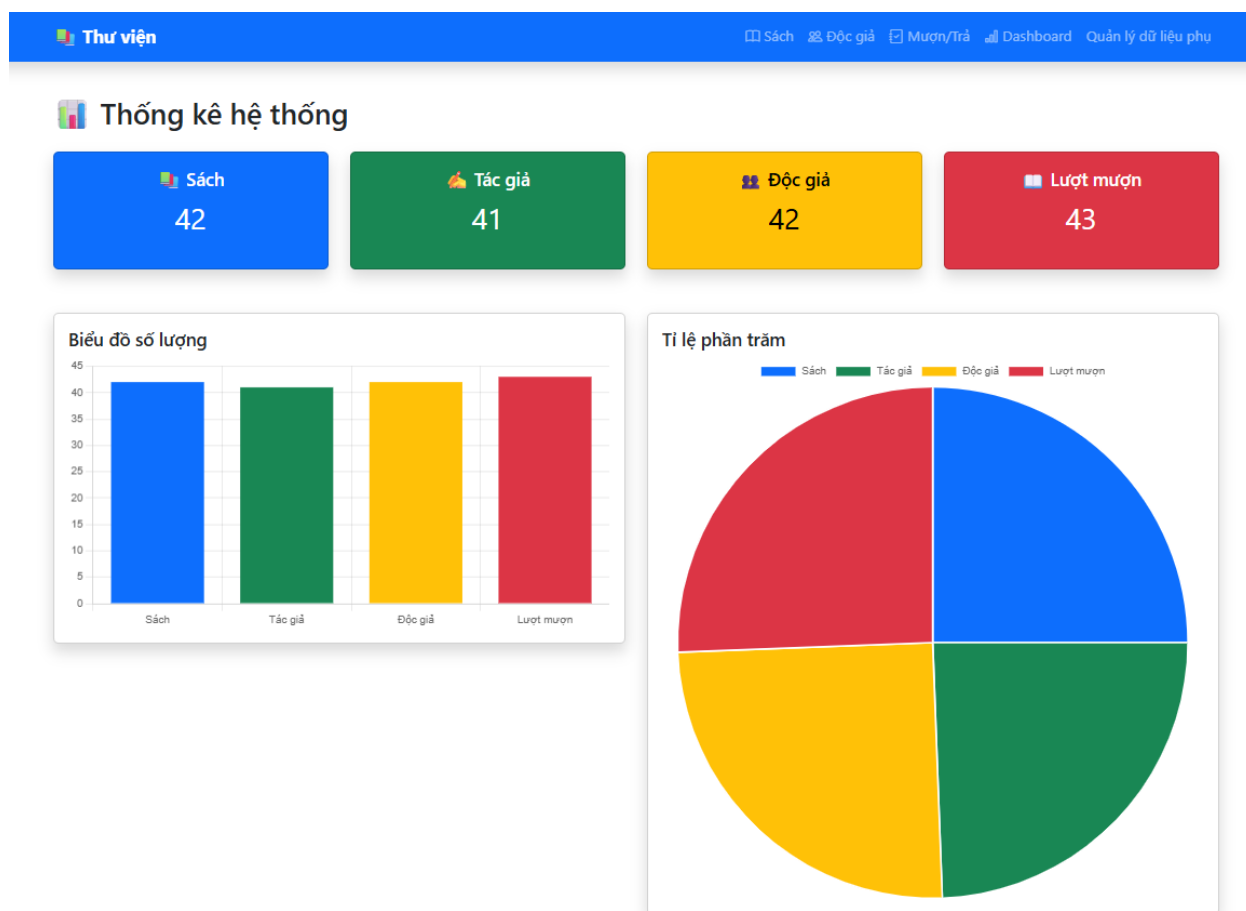
- Ghi nhận trả sách và cập nhật trạng thái.
- **Thiết kế:** Có thông báo xác nhận khi mượn/trả thành công.

Thư viện			
<div> <div>Sách</div> <div>Độc giả</div> <div>Mượn/Trả</div> <div>Dashboard</div> <div>Quản lý dữ liệu phụ</div> </div>			
<div> <div>Danh sách mượn/trả</div> <div>+ Ghi nhận mượn sách</div> </div>			
Sách	Độc giả	Ngày mượn	Ngày trả
Sách 27	Độc giả 6	Ngày 21 tháng 1 năm 2025	Ngày 25 tháng 4 năm 2025
Sách 7	Độc giả 16	Ngày 24 tháng 3 năm 2025	Ngày 18 tháng 12 năm 2025
Sách 4	Độc giả 1	Ngày 18 tháng 11 năm 2025	Ngày 06 tháng 7 năm 2025
Sách 34	Độc giả 3	Ngày 01 tháng 8 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả
Sách 29	Độc giả 16	Ngày 01 tháng 11 năm 2025	Ngày 24 tháng 1 năm 2025
Sách 6	Độc giả 18	Ngày 22 tháng 9 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả
Sách 5	Độc giả 29	Ngày 22 tháng 12 năm 2025	Ngày 17 tháng 9 năm 2025
Sách 19	Độc giả 14	Ngày 19 tháng 11 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả
Sách 37	Độc giả 8	Ngày 24 tháng 3 năm 2025	Ngày 06 tháng 5 năm 2025
Sách 6	Độc giả 40	Ngày 06 tháng 10 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả
Sách 35	Độc giả 32	Ngày 03 tháng 5 năm 2025	Ngày 26 tháng 5 năm 2025
Sách 24	Độc giả 37	Ngày 19 tháng 11 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả
Sách 34	Độc giả 6	Ngày 19 tháng 3 năm 2025	Ngày 23 tháng 6 năm 2025
Sách 38	Độc giả 3	Ngày 24 tháng 1 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả
Sách 30	Độc giả 11	Ngày 13 tháng 2 năm 2025	Ngày 04 tháng 8 năm 2025
Sách 7	Độc giả 25	Ngày 25 tháng 12 năm 2025	Chưa trả <input checked="" type="checkbox"/> Xác nhận đã trả

Hình 4.4: Giao diện Mượn/Trả Sách.

4.4.5. Giao diện Thống kê và Báo cáo

- **Nội dung:** Hiện thị số lượng sách hiện có, số lượt mượn theo tháng, danh sách người dùng mượn nhiều nhất.
- **Chức năng:**
 - Xuất báo cáo ra file CSV/Excel.
 - Hiện thị biểu đồ trực quan (sử dụng Chart.js).
- **Thiết kế:** Các biểu đồ cột và đường giúp quản trị viên dễ dàng theo dõi tình hình hoạt động thư viện.



Hình 4.5: Giao diện Thống kê và Báo cáo.

4.4.6. Giao diện Quản lý tác giả, thể loại, nhà xuất bản

Đây là giao diện dành cho **quản trị viên** hoặc **thủ thư**, nhằm quản lý các thông tin liên quan đến tác giả, thể loại và nhà xuất bản – những thành phần quan trọng giúp hệ thống thư viện được tổ chức khoa học và dễ tra cứu.

a) Quản lý tác giả

- **Danh sách tác giả:** Hiển thị bảng gồm các cột: Mã tác giả, Tên tác giả, Quốc tịch, Năm sinh.
- **Chức năng:**
 - Thêm tác giả mới.
 - Sửa thông tin tác giả.

- Xóa tác giả khỏi hệ thống.
- Tìm kiếm tác giả theo tên hoặc quốc tịch.
- **Thiết kế:** Form nhập liệu đơn giản, có thông báo trạng thái khi thêm/sửa/xóa thành công.

b) Quản lý thể loại

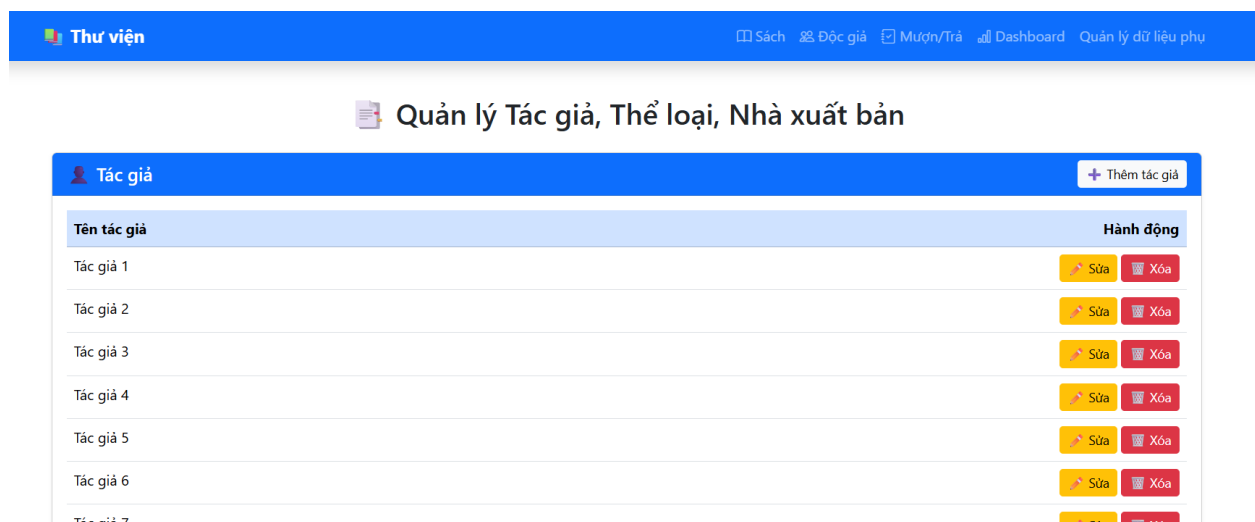
- **Danh sách thể loại:** Hiển thị bảng gồm các cột: Mã thể loại, Tên thể loại, Mô tả.
- **Chức năng:**
 - Thêm thể loại mới.
 - Sửa thông tin thể loại.
 - Xóa thể loại.
 - Tìm kiếm thể loại theo tên.
- **Thiết kế:** Giao diện trực quan, hỗ trợ phân loại sách theo nhiều tiêu chí để thuận tiện cho việc tra cứu.

c) Quản lý nhà xuất bản

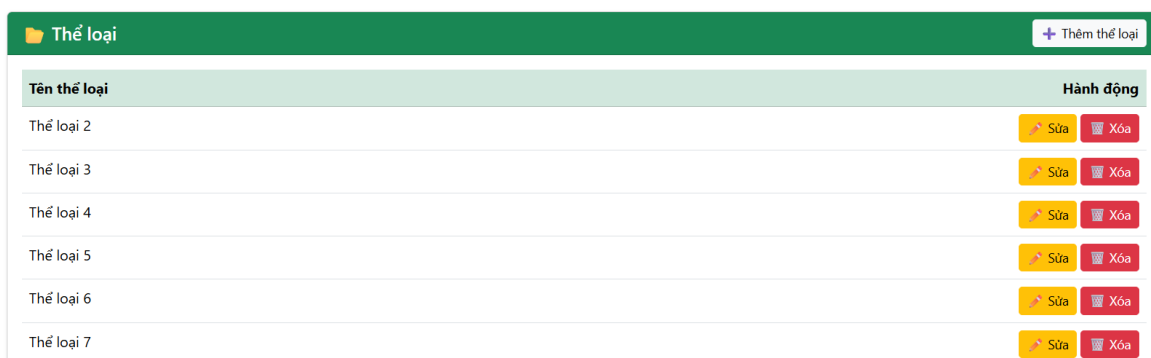
- **Danh sách nhà xuất bản:** Hiển thị bảng gồm các cột: Mã NXB, Tên NXB, Địa chỉ, Số điện thoại, Email.
- **Chức năng:**
 - Thêm nhà xuất bản mới.
 - Sửa thông tin nhà xuất bản.
 - Xóa nhà xuất bản.
 - Tìm kiếm nhà xuất bản theo tên hoặc địa chỉ.
- **Thiết kế:** Form nhập liệu có kiểm tra tính hợp lệ (ví dụ: email đúng định dạng, số điện thoại hợp lệ).

d) Đặc điểm chung của giao diện

- Các bảng dữ liệu đều được thiết kế bằng **Bootstrap Table**, hỗ trợ phân trang (pagination) và sắp xếp (sorting).
- Các form nhập liệu sử dụng **django-crispy-forms** để hiển thị đẹp mắt và dễ thao tác.
- Có thông báo trạng thái (alert) khi thực hiện thành công hoặc gặp lỗi.
- Tích hợp chức năng **liên kết dữ liệu**: khi thêm sách mới, người dùng có thể chọn tác giả, thể loại, và nhà xuất bản từ danh sách đã có.



Hình 4.6: Giao diện quản lý tác giả.



Hình 4.7: Giao diện quản lý thể loại.

Nhà xuất bản

Thêm NXB

Tên NXB	Hành động
NXB 1	<div><div></div>Sửa</div> <div><div></div>Xóa</div>
NXB 2	<div><div></div>Sửa</div> <div><div></div>Xóa</div>
NXB 3	<div><div></div>Sửa</div> <div><div></div>Xóa</div>
NXB 4	<div><div></div>Sửa</div> <div><div></div>Xóa</div>
NXB 5	<div><div></div>Sửa</div> <div><div></div>Xóa</div>
NXB 6	<div><div></div>Sửa</div> <div><div></div>Xóa</div>

Hình 4.8: Giao diện quản lý nhà xuất bản.

4.5. Kết quả thử nghiệm (Testing)

Nhóm đã tiến hành kiểm thử ứng dụng với các kịch bản khác nhau để đánh giá độ ổn định:

STT	Kịch bản kiểm thử (Test Case)	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế	Đánh giá
1	Thêm sách mới	Nhập đầy đủ thông tin sách: Tên, Tác giả, Thể loại, Năm XB, Số lượng	Sách được lưu vào CSDL, hiển thị trong danh sách	Sách hiển thị đúng, dữ liệu lưu trữ chính xác	Thành công
2	Tìm kiếm sách	Từ khóa “Toán học”	Hiển thị danh sách sách có tên hoặc thể loại liên quan	Kết quả trả về đúng, không lỗi font tiếng Việt	Thành công
3	Mượn sách	Người dùng chọn sách còn số lượng > 0	Hệ thống ghi nhận mượn, giảm số lượng sách khả dụng	Bản ghi mượn được tạo, số lượng sách giảm 1	Thành công
4	Trả sách	Người dùng trả sách đã mượn	Hệ thống cập nhật trạng thái, tăng số lượng sách khả dụng	Bản ghi trả được lưu, số lượng sách tăng lại	Thành công
5	Quản lý tác giả/thể loại/NXB	Thêm mới tác giả “Nguyễn Nhật Ánh”	Tác giả hiển thị trong danh sách, liên kết được với sách	Dữ liệu hiển thị đúng, liên kết hoạt động	Thành công

STT	Kịch bản kiểm thử (Test Case)	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế	Đánh giá
6	Xóa sách	Xóa sách có ID hợp lệ	Sách biến mất khỏi danh sách, dữ liệu CSDL cập nhật	Sách bị xóa đúng, không ảnh hưởng dữ liệu khác	Thành công
7	Thống kê báo cáo	Truy vấn số lượt mượn theo tháng	Hệ thống trả về biểu đồ và bảng thống kê	Biểu đồ hiển thị đúng, dữ liệu chính xác	Thành công
8	Xử lý dữ liệu lỗi	Nhập thiếu thông tin bắt buộc (ví dụ: không nhập tên sách)	Hệ thống báo lỗi, không lưu dữ liệu	Giao diện hiển thị thông báo đỏ “Thiếu thông tin”	Thành công

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau quá trình nghiên cứu và thực hiện đề tài “Xây dựng trang web quản lý thư viện với Django”, nhóm đã hoàn thành các mục tiêu đề ra và đạt được những kết quả cụ thể:

5.1.1. Kết quả đạt được

- Xây dựng thành công hệ thống quản lý thư viện: Ứng dụng hoạt động ổn định trên mô hình Client – Server, kết nối mượt mà giữa Frontend (HTML, CSS, Bootstrap, JavaScript) và Backend (Django – Python).
- Đáp ứng đầy đủ chức năng nghiệp vụ:
 - Quản lý sách: thêm, sửa, xóa, tìm kiếm, phân loại theo tác giả, thể loại, nhà xuất bản.
 - Quản lý mượn/trả: ghi nhận lịch sử mượn, hạn trả, trạng thái sách.
 - Quản lý tác giả, thể loại, nhà xuất bản: hỗ trợ nhập liệu và liên kết dữ liệu với sách.
 - Thống kê và báo cáo: hiển thị số lượng sách, lượt mượn theo thời gian, danh sách người mượn nhiều nhất.
- Trải nghiệm người dùng tốt: Giao diện web hiện đại, thân thiện, dễ sử dụng, hiển thị tốt trên nhiều thiết bị.

5.1.2. Hạn chế của đề tài

Bên cạnh những ưu điểm, ứng dụng vẫn còn tồn tại một số hạn chế cần khắc phục:

- Chưa có chức năng đăng nhập/đăng ký: Người dùng thao tác trực tiếp trên giao diện, chưa có cơ chế phân quyền.
- Chưa hỗ trợ quản lý tài liệu số (ebooks, PDF): Hệ thống mới chỉ tập trung vào sách in và hoạt động mượn/trả cơ bản.
- Thống kê còn đơn giản: Chỉ dừng lại ở mức số lượng và biểu đồ cơ bản, chưa có phân tích nâng cao.

- Khả năng mở rộng: Hệ thống hiện tại sử dụng SQLite, phù hợp cho demo nhưng chưa tối ưu cho triển khai thực tế với dữ liệu lớn.

5.2. Hướng phát triển

Để ứng dụng hoàn thiện hơn và có thể đưa vào sử dụng thực tế rộng rãi, nhóm đề xuất các hướng phát triển trong tương lai:

1. Tích hợp chức năng đăng nhập/đăng ký và phân quyền người dùng

- Bổ sung cơ chế xác thực để phân biệt người dùng thường, thủ thư và quản trị viên.
- Cho phép quản lý lịch sử mượn/trả theo từng tài khoản, đảm bảo tính minh bạch và bảo mật.

2. Mở rộng quản lý tài liệu số (Digital Library)

- Tích hợp thêm module quản lý tài liệu điện tử (ebooks, PDF, tài liệu tham khảo).
- Hỗ trợ đọc trực tuyến ngay trên hệ thống, giúp thư viện trở thành kho tri thức số.

3. Nâng cấp cơ sở dữ liệu và hiệu năng hệ thống

- Chuyển từ SQLite sang PostgreSQL hoặc MySQL để tăng khả năng mở rộng và xử lý dữ liệu lớn.
- Tối ưu hóa truy vấn và bộ nhớ để hệ thống hoạt động ổn định ngay cả khi có nhiều người dùng truy cập đồng thời.

4. Thống kê và báo cáo nâng cao

- Bổ sung các báo cáo chi tiết hơn: xu hướng mượn sách theo thể loại, tác giả, hoặc theo từng học kỳ.
- Tích hợp biểu đồ trực quan (Chart.js, D3.js) để hỗ trợ quản trị viên trong việc ra quyết định.

5. Triển khai thực tế (Deployment)

- Đưa ứng dụng lên môi trường Internet thông qua các nền tảng như Heroku, AWS hoặc DigitalOcean.

- Cho phép người dùng truy cập từ mọi nơi, không giới hạn trong mạng nội bộ.

6. Tăng cường bảo mật

- Áp dụng các cơ chế mã hóa dữ liệu, chống tấn công SQL Injection, XSS, CSRF.
- Đảm bảo an toàn tuyệt đối cho thông tin người dùng và dữ liệu thư viện.

CHƯƠNG 6 TÀI LIỆU THAM KHẢO

- [1] Django Software Foundation, *Django Documentation*. [Online]. Available: <https://docs.djangoproject.com>
- [2] Bootstrap Team, *Bootstrap Official Documentation*. [Online]. Available: <https://getbootstrap.com>
- [3] SQLite Consortium, *SQLite Documentation*. [Online]. Available: <https://www.sqlite.org/docs.html>
- [4] W. S. Vincent, *Django for Beginners: Build websites with Python and Django*. Leanpub, 2023.
- [5] A. Melé, *Django 4 By Example*. Packt Publishing, 2022.
- [6] Mozilla Developer Network (MDN), *HTML, CSS, and JavaScript Guides*. [Online]. Available: <https://developer.mozilla.org>
- [7] Django REST Framework, *DRF Documentation*. [Online]. Available: <https://www.django-rest-framework.org>
- [8] Chart.js Team, *Chart.js Documentation*. [Online]. Available: <https://www.chartjs.org>
- [9] Khoa Sư phạm Toán – Tin, Trường Đại học Đồng Tháp, *Tài liệu giảng dạy môn Lập trình Python và Ứng dụng*. Đồng Tháp, 2025.
- [10] Stack Overflow & GitHub Community, *Các bài viết và hướng dẫn thực hành liên quan đến Django và quản lý thư viện*. [Online].