**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**NGUYỄN VĂN ĐỨC**

XÂY D**Ự**NG H**Ệ** TH**Ố**NG QUẢN LÝ THƯ VIỆN

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2025BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**NGUYỄN VĂN ĐỨC**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ THƯ VIỆN**

**Mã số sinh viên: 2251012038**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: Ths. NGUYỄN THỊ MAI TRANG**

**TP. HỒ CHÍ MINH, 2025**

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án “Xây dựng hệ thống Quản lý thư viện”, em đã nhận được rất nhiều sự hỗ trợ và giúp đã quý báu từ thầy cô, bạn bè và gia đình.

Quý thầy cô trong Khoa Công Nghệ Thông Tin, đặc biệt là Ths. Nguyễn Thị Mai Trang, người đã tận tình hướng dẫn, định hướng và góp ý xuyên suốt quá trình thực hiện đồ án. Những kiến thức và kinh nghiệm mà cô đã chia sẻ là nền tảng quan trọng giúp em hoàn thiện đề tài này.

Cuối cùng, con xin bày tỏ lòng biết ơn đến gia đình, bạn bè đã luôn là nguồn động viên, hỗ trợ to lớn về mặt tinh thần, giúp con vững bước để hoàn thành tốt công việc của mình.

Mặc dù đã rất cố gắng, nhưng chắc chắn bài làm vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp của quý thầy cô và các bạn.

Em xin chân thành cảm ơn!

TP HCM, Tháng 8 Năm 2025

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc205302421)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc205302422)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc205302423)

[MỤC LỤC 4](#_Toc205302424)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc205302425)

[DANH MỤC HÌNH VẼ 8](#_Toc205302426)

[DANH MỤC BẢNG 9](#_Toc205302427)

[MỞ ĐẦU 10](#_Toc205302428)

[Chương 1. GIỚI THIỆU ĐỀ TÀI 11](#_Toc205302429)

[1.1. Giới thiệu đề tài 11](#_Toc205302430)

[1.2. Lý do chọn đề tài 11](#_Toc205302431)

[1.3. Bố cục báo cáo 12](#_Toc205302432)

[Chương 2. CƠ SỞ LÝ THUYẾT 12](#_Toc205302433)

[2.1. Gi**ớ**i thi**ệ**u Spring Boot 12](#_Toc205302434)

[2.1.1. Định nghĩa 12](#_Toc205302435)

[2.1.2. Tính chất 12](#_Toc205302436)

[2.1.3. Đặc điểm nổi bật 12](#_Toc205302437)

[2.2. Giới thiệu ReactJS 12](#_Toc205302438)

[2.2.1. Định nghĩa 12](#_Toc205302439)

[2.2.2. Virtual DOM 12](#_Toc205302440)

[2.2.3. Tính năng nổi bật 12](#_Toc205302441)

[2.3. RESTFULL API 12](#_Toc205302442)

[2.3.1. Restfull Api là gì 12](#_Toc205302443)

[2.3.2. Cách thức hoạt động 12](#_Toc205302444)

[2.3.3. Ưu điểm 12](#_Toc205302445)

[2.4. MySQL 12](#_Toc205302446)

[2.5. Tổng quan mô hình MVC (Model – View – Controller) 12](#_Toc205302447)

[2.6. Git và Github 12](#_Toc205302448)

[2.7. Taiwind CSS 12](#_Toc205302449)

[2.7.1. Ưu điểm 12](#_Toc205302450)

[2.7.2. Nhược điểm 12](#_Toc205302451)

[2.7.3. Cách thiết lập 12](#_Toc205302452)

[Chương 3. PHÁT TRIỂN HỆ THỐNG PHÂN TÍCH VÀ THIẾT KẾ 13](#_Toc205302453)

[3.1. Giới thiệu hệ thống 13](#_Toc205302455)

[3.1.1. Giới thiệu hệ thống quản lý thư viện 13](#_Toc205302456)

[3.1.2. Tóm tắt chức năng của hệ thống 13](#_Toc205302457)

[3.2. Phân tích thiết kế 13](#_Toc205302458)

[3.2.1. Các tác nhân của hệ thống 13](#_Toc205302459)

[3.2.2. Sơ đồ lớp 13](#_Toc205302460)

[3.2.3. Sơ đồ usecase tổng quát 13](#_Toc205302461)

[3.2.4. Đặc tả usecase 13](#_Toc205302462)

[3.2.5. Sơ đồ Activity quan trọng 13](#_Toc205302463)

[3.2.6. Xây dựng cơ sở dữ liệu 13](#_Toc205302464)

[3.2.7. Sơ đồ tuần tự 13](#_Toc205302465)

[Chương 4. CÁC CHƯC NĂNG HỆ THỐNG ĐẠT ĐƯỢC 14](#_Toc205302466)

[4.1. Chức năng quản trị (quản trị viên, kiểm duyệt viên) 14](#_Toc205302468)

[4.1.1. Thống kê 14](#_Toc205302469)

[4.1.2. Quản lý sách 14](#_Toc205302470)

[4.1.3. Quản lý loại sách 14](#_Toc205302471)

[4.1.4. Quản lý mượn/trả sách 14](#_Toc205302472)

[4.1.5. Quản lý người dùng 14](#_Toc205302473)

[4.2. Chức năng người dùng (khách hàng, người đọc sách) 14](#_Toc205302474)

[4.2.1. Trang chủ 14](#_Toc205302475)

[4.2.2. Đăng nhập 14](#_Toc205302476)

[4.2.3. Đăng kí 14](#_Toc205302477)

[4.2.4. Đặt lại mật khẩu 14](#_Toc205302478)

[4.2.5. Đổi mật khẩu 14](#_Toc205302479)

[4.2.6. Lọc và tìm sách 14](#_Toc205302480)

[4.2.7. Bình luận và đánh giá sách 14](#_Toc205302481)

[Chương 5. HƯỚNG KẾT LUẬN VÀ PHÁT TRIỂN 14](#_Toc205302482)

[5.1. Kết quả đạt được và hạn chế 14](#_Toc205302487)

[5.1.1. Kết quả đạt được 14](#_Toc205302488)

[5.1.2. Một vài hạn chế 14](#_Toc205302489)

[5.2. Kết luận 14](#_Toc205302490)

[5.3. Hướng phát triển 14](#_Toc205302491)

[TÀI LIỆU THAM KHẢO 15](#_Toc205302492)

[PHỤ LỤC 16](#_Toc205302493)

DANH MỤC TỪ VIẾT TẮT

DANH MỤC HÌNH VẼ

[Hình 1.1: Tên hình 1 10](#_Toc367742554)

DANH MỤC BẢNG

[Bảng 1.1: Tên bảng 1 10](#_Toc367742567)

[Bảng 2.1: Tên bảng 1 11](#_Toc367742568)

MỞ ĐẦU

Trong thời đại chuyển đổi số, Công nghệ Thông tin ngày càng đóng vai trò then chốt trong việc hỗ trợ và tự động hóa các hoạt động trong nhiều lĩnh vực, đặc biệt là giáo dục. Hầu hết các tổ chức, trường học hiện nay đều hướng tới việc số hóa hệ thống quản lý nhằm tiết kiệm thời gian, nâng cao hiệu quả và đảm bảo độ chính xác trong công việc.

Trong đó, **quản lý thư viện** là một nghiệp vụ quan trọng, có liên quan trực tiếp đến hoạt động học tập và nghiên cứu của học sinh, sinh viên cũng như giảng viên. Tuy nhiên, nhiều thư viện vẫn đang sử dụng phương pháp quản lý thủ công hoặc bán thủ công, dẫn đến việc tra cứu, mượn – trả sách gặp nhiều khó khăn, mất thời gian và dễ xảy ra sai sót. Từ thực tế đó, em quyết định thực hiện đề tài: **“Xây dựng hệ thống quản lý thư viện”.**

Mục tiêu của hệ thống là hỗ trợ quản lý các đầu sách, bạn đọc, quy trình mượn – trả và thống kê thông tin một cách khoa học, chính xác và dễ sử dụng. Ứng dụng này được xây dựng như một giải pháp đơn giản nhưng có khả năng mở rộng, phù hợp với các thư viện quy mô vừa và nhỏ.

Đề tài là cơ hội để em vận dụng kiến thức đã học vào thực tiễn, đồng thời rèn luyện tư duy hệ thống và kỹ năng lập trình. Mặc dù thời gian thực hiện còn hạn chế, em đã cố gắng hoàn thiện sản phẩm ở mức tốt nhất có thể. Rất mong nhận được sự đóng góp ý kiến từ thầy cô và mọi người để hệ thống ngày càng hoàn thiện hơn.

# Chương 1. GIỚI THIỆU ĐỀ TÀI

* 1. Giới thiệu đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng công nghệ thông tin vào quản lý thư viện đã trở thành một xu hướng tất yếu. Thư viện truyền thống với hình thức quản lý thủ công bằng sổ sách, giấy tờ không còn đáp ứng hiệu quả nhu cầu tra cứu, mượn – trả sách của người dùng. Những bất cập như thất lạc thông tin, tra cứu chậm, thống kê không chính xác,.. gây ra nhiều khó khăn cho cả thủ thư và người đọc.

Để khắc phục những hạn chế đó, việc xây dựng một Hệ thống quản lý thư viện là cần thiết. Hệ thống này nhằm tin học hóa các nghiệp vụ quản lý thư viện như: quản lý sách, tác giả, thể loại, độc giả, mượn – trả sách, thống kê, xử lý vi phạm,.. giúp tiết kiệm thời gian, nâng cao hiệu quả và độ chính xác trong công tác quản lý.

Hệ thống sẽ cung cấp giao diện thân thiện, dễ sử dụng cho cả quản trị viên (quản lý sách, thành viên, báo cáo thống kê...) và người dùng (tìm kiếm sách, xem thông tin, đăng ký mượn trả,..). Qua đó, góp phần nâng cao chất lượng phục vụ và trải nghiệm của bạn đọc trong thư viện hiện đại.

* 1. Lý do chọn đề tài

Hiện nay, nhiều trường học, trung tâm đào tạo và tổ chức vẫn đang sử dụng phương pháp quản lý thư viện truyền thống, thủ công. Điều này không chỉ làm mất nhiều thời gian mà còn dễ gây sai sót, thất lạc dữ liệu, khó kiểm soát số lượng sách và người mượn. Đặc biệt, khi số lượng sách và người dùng tăng lên, khối lượng công việc càng trở nên phức tạp và khó quản lý.

Với nền tảng kiến thức đã học và niềm yêu thích với lĩnh vực phát triển phần mềm quản lý, em lựa chọn thực hiện đề tài “Xây dựng hệ thống quản lý thư viện” nhằm cung cấp một giải pháp công nghệ giúp tự động hóa các quy trình trong thư viện. Hệ thống không những đáp ứng các chức năng cơ bản mà còn hướng đến khả năng mở rộng, tích hợp tìm kiếm thông minh và nâng cao trải nghiệm người dùng.

* 1. Bố cục báo cáo

Bài báo cáo đồ án ngành gồm có 5 phần chính:

* Chương 1 – Giới thiệu đề tài
* Chương 2 – Cơ sở lý thuyết
* Chương 3 – Phát triển hệ thống: phân tích và thiết kế
* Chương 4 – Triển khai hệ thống: chức năng của hệ thống sau khi phát triển
* Chương 5 – Kết luận và hướng phát triển

# Chương 2. CƠ SỞ LÝ THUYẾT

* 1. Gi**ớ**i thi**ệ**u Spring Boot
     1. Định nghĩa

Spring Boot được xây dựng dựa trên nền của Spring Framework, phát triển bởi ngôn ngữ Java. Nó là một phiên bản có tính tự động hoá cao hơn và đơn giản hơn của Spring. Spring Boot khiến cho việc tạo và khởi chạy ứng dụng chỉ diễn ra trong vòng vài phút.[1]

* + 1. Tính chất

Độ hoàn thiện cao: Spring Boot cung cấp rất nhiều tính năng được tích hợp sẵn (out-of-the-box) như hệ thống cấu hình tự động (auto-configuration), máy chủ nhúng (embedded server), công cụ quản lý phụ thuộc, giúp lập trình viên nhanh chóng xây dựng ứng dụng mà không cần cấu hình thủ công quá nhiều[2].

Linh hoạt: Spring Boot phù hợp cho nhiều loại ứng dụng khác nhau, từ hệ thống đơn giản đến các hệ thống phức tạp, quy mô lớn. Người dùng có thể dễ dàng tùy biến các thành phần trong hệ thống theo nhu cầu thực tế[2].

Bảo mật: Spring Boot hỗ trợ tích hợp Spring Security – một mô-đun bảo mật mạnh mẽ giúp bảo vệ ứng dụng khỏi các lỗ hổng phổ biến như XSS, CSRF, SQL Injection... Người dùng cũng có thể dễ dàng cấu hình xác thực, phân quyền người dùng trong hệ thống[2].

Khả năng mở rộng: Spring Boot được thiết kế với kiến trúc mô-đun, hỗ trợ tốt cho việc mở rộng và tích hợp với các công nghệ khác như Spring Cloud, Docker, Kafka, v.v[2].

Khả năng tích hợp: Spring Boot hỗ trợ tích hợp với nhiều hệ quản trị cơ sở dữ liệu (MySQL, PostgreSQL, MongoDB...), công cụ logging, API RESTful, và các thư viện bên thứ ba thông qua hệ thống quản lý phụ thuộc (Maven hoặc Gradle) [2].

* + 1. Đặc điểm nổi bật

Tự động cấu hình (Auto Configuration): Spring Boot tự động cấu hình các thành phần của ứng dụng dựa trên các thư viện hiện có trong classpath. Điều này giúp giảm thiểu đáng kể công việc cấu hình thủ công, đặc biệt là so với Spring truyền thống.

Cung cấp các máy chủ nhúng như Tomcat, Jetty, hoặc Undertow, cho phép chạy ứng dụng mà không cần triển khai trên máy chủ bên ngoài, giúp phát triển và kiểm thử nhanh chóng.

Sử dụng hệ thống quản lý phụ thuộc (Maven hoặc Gradle) với các "starter" (như spring-boot-starter-web, spring-boot-starter-data-jpa) để tự động kéo các thư viện cần thiết, giảm xung đột phiên bản.

Cung cấp các công cụ như Spring Boot DevTools (tự động reload khi mã thay đổi), cấu hình dựa trên file properties/YAML, và Spring Initializr để khởi tạo dự án nhanh.

Hỗ trợ tích hợp với nhiều công nghệ như Spring Cloud, Docker, Kafka, và các cơ sở dữ liệu (MySQL, PostgreSQL, MongoDB), phù hợp cho các hệ thống từ nhỏ đến lớn.

Tích hợp Spring Security, hỗ trợ xác thực, phân quyền, và bảo vệ ứng dụng khỏi các lỗ hổng như XSS, CSRF, hoặc SQL Injection.

Cung cấp các công cụ mạnh mẽ để phát triển API RESTful nhanh chóng, dễ dàng tích hợp với frontend (như ReactJS) thông qua JSON.

* 1. Giới thiệu ReactJS
     1. Định nghĩa

ReactJS (thường gọi là React) là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook vào năm 2013, dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web, đặc biệt là các ứng dụng đơn trang (Single Page Applications - SPA).

* + 1. Virtual DOM

Virtual DOM (Document Object Model ảo) là một trong những tính năng cốt lõi giúp React trở nên nổi bật so với các thư viện hoặc framework khác. Virtual DOM là một bản sao nhẹ của DOM thực tế, được lưu trữ trong bộ nhớ và hoạt động như một lớp trung gian giữa trạng thái ứng dụng và giao diện hiển thị trên trình duyệt.

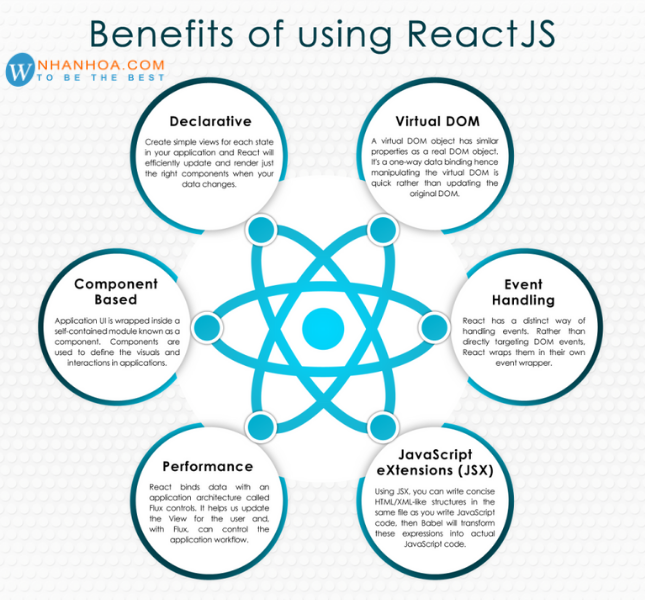
* + 1. Tính năng nổi bật

Quản lý trạng thái hiệu quả: React hỗ trợ quản lý trạng thái (state) thông qua các hook như useState và useEffect, hoặc các thư viện quản lý trạng thái bên thứ ba như Redux hoặc Context API. Điều này rất hữu ích trong việc quản lý trạng thái động.

Tích hợp dễ dàng: React có thể tích hợp với các backend như Spring Boot thông qua API RESTful, sử dụng các thư viện như Axios hoặc Fetch để gửi và nhận dữ liệu.

Cộng đồng lớn và hệ sinh thái phong phú: React có một cộng đồng phát triển rộng lớn, với nhiều thư viện hỗ trợ như React Router (định tuyến), Material-UI (giao diện), và Formik (xử lý biểu mẫu). Điều này giúp đẩy nhanh quá trình phát triển và đảm bảo tính ổn định của hệ thống.

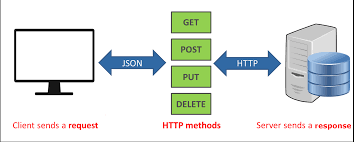
Khả năng mở rộng: React hỗ trợ xây dựng các ứng dụng lớn bằng cách sử dụng các công cụ như Next.js (cho server-side rendering) hoặc tích hợp với các dịch vụ như Firebase, AWS để mở rộng tính năng.



*Hình 2.1. Sơ đồ tóm tắt lợi ích của ReactJs*

* 1. RESTFULL API
     1. Restfull Api là gì

RESTful API (Giao diện Lập trình Ứng dụng theo phong cách REST) là một tập hợp các quy tắc thiết kế cho phép các ứng dụng giao tiếp với nhau thông qua giao thức HTTP, sử dụng các phương thức chuẩn như GET, POST, PUT, DELETE để thao tác trên các tài nguyên (resources). REST (Representational State Transfer) là một kiến trúc được Roy Fielding đề xuất vào năm 2000, nhấn mạnh tính đơn giản, khả năng mở rộng và không lưu trạng thái (stateless) [5].

* + 1. Cách thức hoạt động

*Hình 2.2. Hình kiến trúc RESTful API*

RESTful API hoạt động dựa trên các nguyên tắc cốt lõi của kiến trúc REST, sử dụng các tài nguyên được định danh bởi URI (Định danh Tài nguyên Thống nhất) và các phương thức HTTP để thực hiện các thao tác.

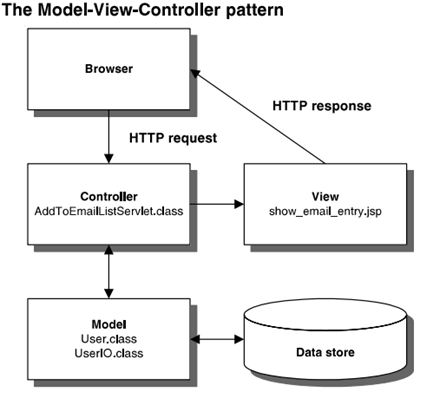
* + 1. Ưu điểm
  + Đơn giản và dễ sử dụng: RESTful API sử dụng các phương thức HTTP chuẩn (GET, POST, PUT, DELETE), dễ hiểu và dễ tích hợp với các công nghệ như ReactJS và Spring Boot.
  + Khả năng mở rộng: thiết kế không lưu trạng thái và dựa trên tài nguyên cho phép hệ thống dễ dàng xử lý số lượng lớn yêu cầu từ người dùng, phù hợp khi hệ thống thư viện mở rộng quy mô.
  + Tính linh hoạt: hỗ trợ nhiều định dạng dữ liệu (JSON, XML) và tích hợp được với nhiều hệ quản trị cơ sở dữ liệu (MySQL, MongoDB) hoặc dịch vụ bên thứ ba.
  + Hiệu suất cao: do không lưu trạng thái, RESTful API giảm tải cho server, giúp xử lý nhanh các yêu cầu như tìm kiếm sách hoặc quản lý giao dịch mượn/trả.
  + Tái sử dụng: các API được thiết kế theo tài nguyên có thể được sử dụng lại cho nhiều chức năng, ví dụ: API /api/books hỗ trợ cả tìm kiếm sách và hiển thị chi tiết sách.
  + Hỗ trợ đa nền tảng: RESTful API cho phép tích hợp với nhiều loại client (web, mobile), tạo điều kiện để hệ thống quản lý thư viện phát triển thêm ứng dụng di động trong tương lai.
  1. MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở, được phát triển bởi Oracle. MySQL sử dụng ngôn ngữ truy vấn có cấu trúc (SQL - Structured Query Language) để quản lý và thao tác dữ liệu. Với hiệu suất cao, độ tin cậy và tính dễ sử dụng, MySQL là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất, được sử dụng rộng rãi trong các ứng dụng web, bao gồm hệ thống quản lý thư viện [9].

* 1. Tổng quan mô hình MVC (Model – View – Controller)

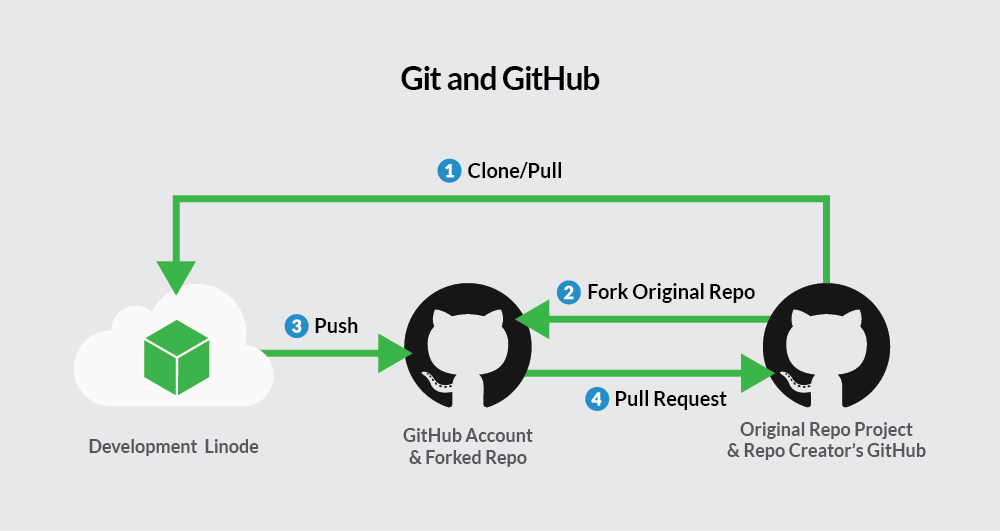
Mô hình MVC (Model – View – Controller) là một mẫu thiết kế kiến trúc phần mềm, chia ứng dụng thành ba thành phần chính: Model, View, và Controller. Mô hình này giúp tách biệt logic nghiệp vụ, giao diện người dùng, và xử lý yêu cầu, từ đó tăng tính bảo trì và khả năng mở rộng của ứng dụng [10].

* Model: Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.
* View: Chịu trách nhiệm hiển thị dữ liệu cho người dùng.
* Controller: Đóng vai trò trung gian, tiếp nhận yêu cầu từ người dùng, tương tác với Model để xử lý dữ liệu, và trả kết quả về View.



*Hình 2.3. Hình mô hình Model-View-Controller*

* 1. Git và Github
* Git: Là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System), cho phép nhiều lập trình viên làm việc trên cùng một dự án, theo dõi lịch sử thay đổi mã nguồn, và quản lý các phiên bản khác nhau của mã [11].
* **GitHub:** Là một nền tảng trực tuyến dựa trên Git, cung cấp nơi lưu trữ mã nguồn (repository), hỗ trợ cộng tác, quản lý dự án, và tích hợp các công cụ như CI/CD (Continuous Integration/Continuous Deployment) [12].



*Hình 2.4. Hình mô tả github*

* 1. Taiwind CSS

Tailwind CSS là một framework CSS theo phong cách tiện ích (utility-first), được thiết kế để xây dựng giao diện người dùng nhanh chóng và hiệu quả thông qua các lớp tiện ích được định nghĩa sẵn. Trong hệ thống quản lý thư viện, Tailwind CSS được sử dụng để tạo giao diện responsive và thân thiện cho các chức năng như tìm kiếm sách, quản lý độc giả, và hiển thị lịch sử mượn/trả, tích hợp với ReactJS ở phía frontend.

* + 1. Ưu điểm
* **Tăng tốc độ phát triển giao diện: Tailwind CSS cung cấp hàng loạt lớp tiện ích (như flex, grid, text-2xl, bg-green-500) để định kiểu trực tiếp trong HTML/JSX, giúp giảm thời gian viết CSS tùy chỉnh**
* **Tùy chỉnh linh hoạt: Thông qua tệp cấu hình tailwind.config.js, Tailwind CSS cho phép tùy chỉnh màu sắc, kích thước, hoặc khoảng cách theo nhu cầu của hệ thống quản lý thư viện, ví dụ: thêm màu sắc riêng cho thương hiệu thư viện.**
* **Hỗ trợ responsive design: Tailwind CSS tích hợp các lớp responsive (như sm:, md:, lg:) để xây dựng giao diện thích ứng với nhiều thiết bị.**
* **Tối ưu kích thước tệp: Tailwind CSS sử dụng công cụ PurgeCSS để loại bỏ các lớp không sử dụng khi triển khai, tạo ra tệp CSS nhỏ gọn, giúp tăng tốc độ tải trang web.**
* **Cộng đồng và tài liệu phong phú: Tailwind CSS có tài liệu chi tiết và cộng đồng hỗ trợ lớn, giúp lập trình viên dễ dàng học và áp dụng, đặc biệt khi tích hợp với ReactJS trong hệ thống quản lý thư viện.**
  + 1. Nhược điểm
* **Sử dụng nhiều lớp tiện ích có thể làm mã JSX trở nên phức tạp và khó đọc.**
* **Lập trình viên mới cần thời gian để làm quen với hệ thống lớp tiện ích và cách sử dụng hiệu quả, đặc biệt khi làm việc với các dự án lớn như hệ thống quản lý thư viện.**
* **Việc tùy chỉnh Tailwind CSS đòi hỏi chỉnh sửa tệp tailwind.config.js. Nếu không tối ưu, có thể tạo ra tệp CSS lớn, ảnh hưởng đến hiệu suất.**
* **Với các giao diện yêu cầu thiết kế độc đáo hoặc phức tạp, Tailwind CSS có thể không linh hoạt bằng cách viết CSS tùy chỉnh hoặc sử dụng các framework như Bootstrap.**
  + 1. Cách thiết lập
  + Trong dự án ReactJS, cài đặt Tailwind CSS và các phụ thuộc bằng npm:

* npm install -D tailwindcss postcss autoprefixer
* npx tailwindcss init
  + Cấu hình file tailwind.config.js:

/\*\* @type {import('tailwindcss').Config} \*/

export default {

content: ["./src/\*\*/\*.{js,jsx}"],

theme: {

extend: {},

},

plugins: [],

}

* + Thêm các chỉ thị Tailwind vào đầu file index.css:

@tailwind base;

@tailwind components;

@tailwind utilities;

# Chương 3. PHÁT TRIỂN HỆ THỐNG PHÂN TÍCH VÀ THIẾT KẾ

1. 1. Giới thiệu hệ thống
      1. Giới thiệu hệ thống quản lý thư viện

Hệ thống Quản lý Thư viện được phát triển trong đồ án này là một giải pháp công nghệ nhằm tạo ra cầu nối hiệu quả giữa bạn đọc và kho tài nguyên tri thức của thư viện. Hệ thống giúp người dùng tra cứu và tiếp cận tài liệu một cách thuận tiện, nhanh chóng với thông tin chính xác, đồng thời hỗ trợ cán bộ thư viện trong công tác quản lý nghiệp vụ một cách chuyên nghiệp và hiệu quả. Nền tảng được tích hợp đầy đủ các chức năng cần thiết cho tất cả các nhóm đối tượng người dùng.

Về phía độc giả, hệ thống cung cấp một giao diện tra cứu trực tuyến (OPAC) trực quan, với danh mục tài liệu đa dạng bao gồm sách, tạp chí, luận văn, tài liệu điện tử... Mỗi tài liệu đều được mô tả chi tiết kèm theo hình ảnh bìa, tóm tắt, thông tin tác giả, và vị trí chính xác trên kệ sách. Bạn đọc có thể dễ dàng tìm kiếm tài liệu theo các yêu cầu khác nhau thông qua bộ lọc nâng cao (lọc theo nhan đề, tác giả, chủ đề, năm xuất bản) hay lưu lại các tài liệu quan tâm vào danh sách đọc cá nhân. Ngoài ra, hệ thống còn cho phép bạn đọc đánh giá, bình luận về các tài liệu đã đọc, giúp những người dùng khác có thêm nguồn tham khảo chất lượng trước khi lựa chọn.

Về phía cán bộ thư viện (thủ thư), hệ thống cung cấp một không gian làm việc chuyên nghiệp để quản lý toàn bộ kho tài liệu số (nghiệp vụ biên mục) và theo dõi tình trạng của từng tài liệu (đang trên kệ, đã được mượn, quá hạn). Quy trình quản lý lưu thông (mượn, trả, gia hạn tài liệu) và quản lý bạn đọc được thiết kế tinh gọn, khoa học và dễ dàng thao tác, giúp giảm thiểu sai sót và tiết kiệm thời gian.

Về phía quản trị hệ thống, hệ thống phân quyền rõ ràng cho từng vai trò. Cán bộ quản lý có thể xem các báo cáo, thống kê trực quan về tình hình hoạt động của thư viện (ví dụ: tài liệu được mượn nhiều nhất, tần suất mượn trả), từ đó có cái nhìn tổng quan để đưa ra các chiến lược phát triển phù hợp. Đặc biệt, hệ thống cung cấp quyền kiểm soát toàn diện cho nhóm quản trị viên (Admin), cho phép quản lý tài khoản người dùng, phân quyền chi tiết, cấu hình các chính sách của thư viện và đảm bảo hệ thống vận hành ổn định, an toàn.

* + 1. Tóm tắt chức năng của hệ thống
* Ch**ứ**c n**ă**ng c**ủ**a độc giả
  + Chức năng đăng nhập: Khách hàng đăng nhập tài khoản nếu đã có tài khoản trước đó.
  + Tra cứu tài liệu: Tìm kiếm sách một cách nhanh chóng thông qua thanh tìm kiếm.
  + Xem thông tin chi tiết: Xem đầy đủ thông tin mô tả của từng tài liệu, bao gồm ảnh bìa, tóm tắt nội dung, số lượng còn lại.
  + Quản lý tài khoản cá nhân: Xem lịch sử mượn, trả tài liệu, kiểm tra thời gian đến hạn của sách
  + Bình luận và đánh giá sách
* Chức năng của thủ thư
  + Quản lý sách: thêm, sửa, xóa thông tin sách
  + Quản lý danh mục, thể loại
  + Thực hiện giao dịch cho mượn, trả và gia hạn cho độc giả
  + Quản lý tài khoản độc giả: thêm, sửa, xóa thông tin độc giả, tra cứu thông tin và lịch sử hoạt động
* Chức năng của người quản trị
  + Quản lý người dùng và phân quyền: Tạo và quản lý các tài khoản trong hệ thống
  + Xem các báo cáo, thống kê sách được mượn nhiều nhất, số lượt mượn
  1. Phân tích thiết kế
     1. Các tác nhân của hệ thống
     2. Sơ đồ lớp
     3. Sơ đồ usecase tổng quát
     4. Đặc tả usecase
        1. Đăng nhập
        2. Đăng kí tài khoản
        3. Chức năng quên mật khẩu
        4. Chức năng tìm kiếm và lọc sách
        5. Chức năng đăng xuất
        6. Chức năng quản lý thông tin cá nhân
        7. Chức năng quản lý sách
        8. Chức năng quản lý người dùng
        9. Chức năng mượn/trả sách
        10. Chức năng bình luận, đánh giá sách
     5. Sơ đồ Activity quan trọng
        1. Đăng nhập, đặt lại mật khẩu
        2. Đăng kí tài khoản
        3. Tìm kiếm và lọc sách
        4. Đăng xuất
        5. Đổi mật khẩu
        6. Bình luận và đánh giá
        7. Mượn/ trả sách
        8. Bình luận, đánh giá
        9. Quản lý thông tin cá nhân
     6. Xây dựng cơ sở dữ liệu
     7. Sơ đồ tuần tự

1. * + 1. Sơ đồ tuần tự cho chức năng đăng nhập
       2. Sơ đồ tuần tự cho chức năng đăng kí
       3. Sơ đồ tuần tự cho chức năng đặt lại mật khẩu
       4. Sơ đồ tuần tự cho chức năng tìm kiếm và lọc
       5. Sơ đồ tuần tự cho chức năng quản lý thông tin cá nhân
       6. Sơ đồ tuần tự cho chức năng quản lý sách
       7. Sơ đồ tuần tự cho chức năng mượn/trả sách
       8. Sơ đồ tuần tự cho chức năng bình luận, đánh giá
       9. Sơ đồ tuần tự cho chức năng quản lý người dùng

# Chương 4. CÁC CHƯC NĂNG HỆ THỐNG ĐẠT ĐƯỢC

1. 1. Chức năng quản trị (quản trị viên, kiểm duyệt viên)
      1. Thống kê
      2. Quản lý sách
      3. Quản lý loại sách
      4. Quản lý mượn/trả sách
      5. Quản lý người dùng
   2. Chức năng người dùng (khách hàng, người đọc sách)
      1. Trang chủ
      2. Đăng nhập
      3. Đăng kí
      4. Đổi mật khẩu
      5. Lọc và tìm sách
      6. Bình luận và đánh giá sách

# Chương 5. HƯỚNG KẾT LUẬN VÀ PHÁT TRIỂN

4. 1. Kết quả đạt được và hạn chế
      1. Kết quả đạt được
      2. Một vài hạn chế
   2. Kết luận
   3. Hướng phát triển

# TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE – *bỏ dòng này khi viết báo cáo*)

1. “Spring Boot là gì giải thích phần mềm…”[Trực tuyến].Địa chỉ https://topdev.vn/blog/gioi-thieu-ve-spring-boot-spring-boot-la-gi/.[ Truy cập ngày 04/08/2025].
2. “Tìm hiểu về spring boot và cách ứng dụng…”[Trực tuyến]. Địa chỉ https://spring.io/projects/spring-boot. [Truy cập ngày 05/08/2025].
3. Hướng nội , “ReactJS là gì.” [Trực tuyến]. Địa chỉ <https://200lab.io/blog/reactjs-la>- gi// [Truy cập ngày 22/09/2024].
4. Lê Đức Mạnh, ‘Restful API là gì’ [Trực tuyến]. Địa chỉ <https://viblo.asia/p/restful>- api-la-gi-1Je5EDJ4lnL [Truy cập ngày 23.09.2024].
5. “Mô hình MVC là gì? Ví dụ về cách hoạt động dễ hiểu” [Trực tuyến]. Địa chỉ <https://topdev.vn/blog/kien-truc-model-view>- controller/?utm\_source=google&utm\_medium=cpc&utm\_campaign=pmax- branding&utm\_content=performance&gad\_source=1&gclid=CjwKCAjwx4O4BhAnE iwA42SbVPazp\_w0BNsYrOKESeJCYv8YZjF8p- j1b38V628DFuUF6Lwc\_u7VGxoCt3QQAvD\_BwE [Truy cập ngày 23/09/2024].
6. “Tailwind CSS: Hướng dẫn 3 cách thiết lập Tailwind chi tiết” [Trực tuyến]. Địa chỉ <https://itviec.com/blog/tailwind-css/> [Truy cập ngày 23/09/2024]
7. “10 lưu ý về thiết kế API đúng” [Trực tuyến]. Địa chỉ <https://hue.codegym.vn/blog/bai-viet-chuyen-mon/10-luu-y-de-thiet-ke-api-dung/> [Truy cập ngày 23/09/2024]
8. “MVC là gì: Tổng quan MVC và ứng dụng mô hình MVC trong lập trình” [Trực tuyến]. Địa chỉ <https://itviec.com/blog/mvc-la-gi/> [Truy cập ngày 23/09/2024]
9. “Github là gì? Hướng dẫn sử dụng Github cho người mới bắt đầu” [Trực tuyến]. Địa chỉ <https://stringee.com/vi/blog/post/huong-dan-su-dung-github> [Truy cập ngày 23/09/2024]
10. [9] Tài liệu MySQL chính thức: https://dev.mysql.com/doc/ (truy cập ngày 05/08/2025).
11. [10] Tài liệu về mô hình MVC: https://viblo.asia/p/mo-hinh-mvc-la-gi-va-cach-hoat-dong-cua-no-1VgZvNNoZAw (truy cập ngày 05/08/2025).
12. [11] Tài liệu Git chính thức: https://git-scm.com/doc (truy cập ngày 05/08/2025).
13. [12] Hướng dẫn GitHub: https://docs.github.com/vi (truy cập ngày 05/08/2025).
14. [13] Bài viết về MySQL trên Viblo: https://viblo.asia/p/tim-hieu-ve-mysql-co-ban-3Q75wQNW5Wb (truy cập ngày 05/08/2025).

# PHỤ LỤC