

**TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN TỐT NGHIỆP

**XÂY DỰNG HỆ THỐNG QUẢN LÝ
THIẾT BỊ VÀ QUÀ TẶNG CHO TRUNG TÂM
NGOẠI NGỮ - TIN HỌC VICTORY**

Giảng viên hướng dẫn : ThS NHAN MINH PHÚC

Sinh viên thực hiện: LÂM HUỆ TRUNG

Mã số sinh viên: 110121255

Lớp : DA21TTC

Khoa : 2021-2025

Trà Vinh, tháng..... năm.....

**TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN TỐT NGHIỆP

**XÂY DỰNG HỆ THỐNG QUẢN LÝ
THIẾT BỊ VÀ QUÀ TẶNG CHO TRUNG TÂM
NGOẠI NGỮ - TIN HỌC VICTORY**

Giảng viên hướng dẫn : ThS NHAN MINH PHÚC

Sinh viên thực hiện: LÂM HUỆ TRUNG

Mã số sinh viên: 110121255

Lớp : DA21TTC

Khoa : 2021-2025

Trà Vinh, tháng..... năm.....

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển không ngừng, việc tự động hóa các quy trình quản lý ngày càng trở nên thiết yếu để nâng cao hiệu quả và tối ưu hóa nguồn lực. Đặc biệt, tại các trung tâm đào tạo như Trung tâm Ngoại ngữ - Tin học Victory, nhu cầu xây dựng một hệ thống quản lý thiết bị và quản lý hiệu quả là vô cùng cần thiết nhằm hỗ trợ công tác quản trị, giảm thiểu sai sót và nâng cao chất lượng dịch vụ.

Hệ thống quản lý được đề xuất trong đề tài này được phát triển dựa trên các công nghệ hiện đại bao gồm Node.js và Express cho backend, MongoDB làm cơ sở dữ liệu, Handlebars (HBS) cho giao diện người dùng, và Bootstrap 5 để đảm bảo thiết kế giao diện thân thiện, đáp ứng tốt trên nhiều thiết bị. Kiến trúc MVC (Model-View-Controller) kết hợp với mô hình CQRS (Command Query Responsibility Segregation) được áp dụng để tăng tính mô-đun, dễ bảo trì và mở rộng của hệ thống.

Đề tài hướng tới việc xây dựng một hệ thống quản lý giúp nhân viên trung tâm dễ dàng theo dõi, quản lý thiết bị và quản lý hiệu quả, đồng thời cung cấp giao diện trực quan cho người dùng. Hệ thống không chỉ hỗ trợ tối ưu hóa quy trình làm việc mà còn đảm bảo tính chính xác và minh bạch trong việc lưu trữ, cập nhật thông tin. Ngoài ra, với xu hướng tích hợp các công nghệ tiên tiến, việc sử dụng MongoDB cho phép lưu trữ dữ liệu linh hoạt, trong khi CQRS giúp tách biệt rõ ràng giữa các thao tác đọc và ghi, nâng cao hiệu suất xử lý.

Đề tài này không chỉ mang lại giá trị thực tiễn cho Trung tâm Ngoại ngữ - Tin học Victory mà còn là cơ hội để nghiên cứu và ứng dụng các công nghệ hiện đại vào giải quyết các bài toán thực tế trong lĩnh vực quản lý.

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn đến toàn thể quý thầy cô, giảng viên Trường Kỹ thuật và Công nghệ, đặc biệt là các thầy cô ở Khoa Công nghệ Thông tin đã tận tâm chỉ dạy trong suốt quá trình học tập tại trường và đã tạo điều kiện tốt nhất để em hoàn thành trọn vẹn bài báo cáo này.

Tiếp theo, em xin tỏ lòng biết ơn sâu sắc đến Thầy Nhan Minh Phúc – Giảng viên Trường Kỹ thuật và Công nghệ, trong quá trình hướng dẫn đồ án đã vô cùng tâm huyết trong việc truyền đạt kiến thức, đưa ra các hướng đi phù hợp và luôn tận tình, sẵn sàng hướng dẫn, chỉ bảo trong suốt quá trình thực hiện đồ án này.

Trong bài báo cáo, do lượng kiến thức và kinh nghiệm còn khiêm tốn, vẫn còn một số sai sót nhỏ không đáng kể. Do đó, em kính mong quý thầy cô thông cảm, góp ý để em có thể tiếp thu và cải thiện cho những nghiên cứu trong tương lai. Sau tất cả, kính chúc các thầy cô luôn dồi dào sức khỏe.

Em xin chân thành cảm ơn!

Trà Vinh, ngày tháng ... năm 2025

Sinh viên thực hiện

Lâm Huệ Trung

NHẬN XÉT

Nội dung quyền khóa luận phù hợp với yêu cầu đề ra, sản phẩm tốt xây dựng đầy đủ các chức năng đã nêu trong đề cương chi tiết, đủ điều kiện bảo vệ khóa luận.

Giảng viên hướng dẫn (ký và ghi rõ họ tên)

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP
(Của giảng viên hướng dẫn)

Họ và tên sinh viên: Lâm Huệ Trung MSSV: 11021255

Ngành: Công nghệ Thông tin..... Khóa: 2021

Tên đề tài: Xây dựng hệ thống quản lý thiết bị và quà tặng cho Trung tâm Ngoại ngữ - Tin học Victory

Họ và tên Giáo viên hướng dẫn: Nhan Minh Phúc.....

Chức danh: Giảng viên Học vị: Thạc sĩ

NHẬN XÉT

1. Nội dung đề tài: Đề tài thiết thực, giải quyết được bài toán quản lý nội bộ tại các trung tâm đào tạo.

2. Ưu điểm:

Sử dụng công nghệ phổ biến và phù hợp với các hệ thống hiện đại: Node.js + Express (backend), MongoDB (cơ sở dữ liệu NoSQL linh hoạt), Bootstrap 5 và Handlebars (frontend). Kiến trúc hệ thống được tổ chức rõ ràng theo mô hình MVC, đồng thời ứng dụng CQRS giúp tách biệt xử lý đọc và ghi, thuận tiện cho mở rộng và bảo trì. Giao diện có tính responsive, tương thích trên nhiều thiết bị.

3. Khuyết điểm:

Thiết kế giao diện người dùng (UI/UX) còn đơn giản, bối rối chưa thực sự tối ưu ở một số trang.

4. Điểm mới đề tài:

Kết hợp quản lý thiết bị và quà tặng trong một hệ thống duy nhất – phù hợp với đặc thù trung tâm đào tạo. Ứng dụng mô hình kiến trúc CQRS, hiếm gặp ở cấp độ đồ án sinh viên, thể hiện tư duy hệ thống khá tốt.

5. Giá trị thực trên đề tài:

Hệ thống có thể triển khai sử dụng tại Trung tâm Ngoại ngữ – Tin học Victory với tùy biến nhỏ.

6. Đề nghị sửa chữa bổ sung: Như nội dung góp ý trực tiếp

8. Đánh giá: Sinh viên thể hiện tinh thần học hỏi tốt, đề tài có tính ứng dụng cao, hoàn thiện đầy đủ chức năng cơ bản, có khả năng triển khai thật.

Trà Vinh, ngày tháng năm 2025
Giảng viên hướng dẫn
(Ký & ghi rõ họ tên)

NHẬN XÉT

Giảng viên chấm

UBND TỈNH TRÀ VINH **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ **Độc lập - Tự do - Hạnh phúc**

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét:

Chức danh: Học vị:

Chuyên ngành:

Cơ quan công tác:

Tên sinh viên:

Tên đề tài đồ án, khóa luận tốt nghiệp:

I. Ý KIẾN NHÂN XÉT

1. Nội dung:

2. Điểm mới các kết quả của đồ án, khóa luận:

.....

3. Ứng dụng thực tế:

II. CÁC VĂN ĐỀ CẦN LÀM RÕ

(Các câu hỏi của giáo viên phản biện)

III. KẾT LUẬN

(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đồ án khóa luận tốt nghiệp)

.....
.....
.....

....., ngày tháng năm 20..

Người nhận xét

(Ký & ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. ĐẶT VẤN ĐỀ.....	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu	1
1.3. Nội dung.....	1
1.4. Đối tượng và phạm vi nghiên cứu	2
1.5. Phương pháp nghiên cứu	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	4
2.1. Mô hình MVC.....	4
2.1.1. Giới thiệu	4
2.1.2. Thành phần trong mô hình MVC.....	4
2.1.3. Luồng xử lý trong MVC	5
2.1.4. Lý do chọn mô hình MVC	6
2.2. CQRS pattern	7
2.2.1. Giới thiệu	7
2.2.2. Command và Query	7
2.2.3. Lý do sử dụng CQRS.....	8
2.3. JWT	8
2.3.1. Giới thiệu	8
2.3.2. Thành phần của JWT	9
2.4. RESTful API.....	9
2.4.1. Tổng quan về RESTful API.....	9
2.4.2. Các thành phần của RESTful API	10
2.4.3. RESTful hoạt động như thế nào	11
2.4.4. Authentication và dữ liệu trả về.....	11
2.5. Các công nghệ xây dựng trang web.....	12
2.5.1. Ngôn ngữ HTML	12
2.5.2. Bảng định kiểu CSS	15
2.5.3. Ngôn ngữ mở rộng của CSS	16
2.5.4. Ngôn ngữ Javascript	17
2.5.5. Bootstrap Framework	18
2.5.6. Node.js	22
2.5.7. Express	24
2.5.8. Hệ quản trị cơ sở dữ liệu MongoDB.....	27
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	29

3.1. Mô tả bài toán	29
3.2. Phân tích yêu cầu và lựa chọn công nghệ	30
3.2.1. Phân tích yêu cầu	30
3.2.2. Lựa chọn công nghệ.....	31
3.3. Thiết kế kiến trúc hệ thống	33
3.3.1. Thiết kế kiến trúc hệ thống	33
3.3.2. Thiết kế cơ sở dữ liệu.....	34
3.4. Xây dựng API phía máy chủ (Back-end).....	39
3.5. Thiết kế giao diện (Front-end)	40
3.5.1. Thiết kế khung giao diện làm việc chính.....	40
3.5.2. Thiết kế khung giao diện đăng nhập	41
3.5.3. Thiết kế giao diện danh sách hiển thị	41
3.5.4. Thiết kế giao diện thêm mới và chỉnh sửa	42
3.5.5. Thiết kế giao diện xem thông tin chi tiết	43
3.5.6. Thiết kế giao diện thống kê.....	43
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	44
4.1. Bộ dữ liệu thử nghiệm	44
4.1.1. Bộ dữ liệu thử nghiệm chức năng quản lý người dùng	44
4.1.2. Bộ dữ liệu thử nghiệm chức năng quản lý thiết bị.....	45
4.1.3. Bộ dữ liệu thử nghiệm chức năng quản lý quà tặng	47
4.2. Kết quả thực nghiệm với vai trò quản lý thiết bị	48
4.2.1. Chức năng đăng nhập	48
4.2.2. Chức năng xem thông tin và thay đổi mật khẩu	48
4.2.3. Chức năng thêm mới thiết bị	49
4.2.4. Chức năng tìm kiếm, chỉnh sửa, xoá xem thông tin chi tiết thiết bị	50
4.2.5. Chức năng thêm phòng	52
4.2.6. Chức năng tìm kiếm, chỉnh sửa, xoá xem thông tin chi tiết phòng	53
4.2.7. Chức năng thêm thiết bị vào phòng	54
4.2.8. Chức năng chuyển, xoá thiết bị của phòng	55
4.2.9. Chức năng mượn thiết bị	55
4.2.10. Chức năng xem đơn mượn và trả thiết bị	58
4.2.11. Chức năng thống kê quản lý thiết bị	59
4.3. Kết quả thực nghiệm với vai trò quản lý quà tặng.....	59
4.3.1. Chức năng thêm quà tặng	59
4.3.2. Chức năng tìm kiếm, chỉnh sửa, xoá xem thông tin chi tiết quà tặng	61
4.3.3. Chức năng tạo đơn yêu cầu quà tặng	62

4.3.4. Chức năng xem thông tin đơn và duyệt đơn yêu cầu	64
4.3.5. Chức năng hoàn trả quà tặng	65
4.3.6. Chức năng thống kê quản lý quà tặng.....	66
4.4. Kết quả thực nghiệm với vai trò quản trị viên	67
4.4.1. Chức năng thêm mới tài khoản người dùng	67
4.4.2. Chức năng xem thông tin, chỉnh sửa, xoá tài khoản người dùng	68
4.4.3. Chức năng vô hiệu hoá, khôi phục tài khoản.....	69
4.4.4. Chức năng thêm mới giảng viên.....	70
4.4.5. Chức năng chỉnh sửa, xoá giảng viên	72
4.5. Các chức năng khác	72
4.5.1. Chức năng thông báo	72
4.5.2. Chức năng gửi mail.....	74
4.6. Kết quả thực nghiệm hệ thống quản lý thiết bị và quà tặng	74
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	77
5.1. Kết luận.....	77
5.2. Hướng phát triển	77
DANH MỤC TÀI LIỆU THAM KHẢO.....	78
PHỤ LỤC 1: CÁC CÔNG CỤ, PHẦN MỀM HỖ TRỢ XÂY DỰNG TRANG WEB	
1. Visual Studio Code	79
2. MongoDB Compass.....	79
3. Postman.....	80

DANH MỤC CÁC HÌNH ẢNH

Hình 2.1 Các thành phần của mô hình MVC.....	4
Hình 2.2 Luồng xử lý trong MVC	6
Hình 2.3 Command và query trong CQRS Pattern	8
Hình 2.4 REST API Design	10
Hình 2.5 Cách thức vận hành của RESTful	11
Hình 3.1 Kiến trúc hệ thống.....	33
Hình 3.2. Sơ đồ quan hệ thực thể.....	34
Hình 3.3 Bản phác thảo khung giao diện chính.....	40
Hình 3.4 Bản phát thảo giao diện đăng nhập	41
Hình 3.5 Bản phát thảo giao diện danh sách hiển thị	42
Hình 3.6 Bản phát thảo giao diện thêm mới và chỉnh sửa.....	42
Hình 3.7 Thiết kế giao diện xem thông tin chi tiết	43
Hình 3.8 Bảng phát thảo giao diện thống kê.....	43
Hình 4.1 Giao diện thực tế chức năng đăng nhập.....	48
Hình 4.2 Giao diện thực thể chức năng xem thông tin và thay đổi mật khẩu	49
Hình 4.3 Giao diện thực tế chức năng thêm thủ công thiết bị	50
Hình 4.4 Giao diện thực tế chức năng thêm thiết bị bằng file	50
Hình 4.5 Giao diện thực tế của danh sách thiết bị	51
Hình 4.6 Giao diện thực tế của chức năng xem thông tin chi tiết thiết bị	51
Hình 4.7 Giao diện thực tế của chức năng chỉnh sửa thông tin thiết bị.....	52
Hình 4.8 Giao diện thực tế của chức năng thêm phòng thủ công	52
Hình 4.9 Giao diện thực tế của chức năng thêm phòng bằng file.....	53
Hình 4.10 Giao diện thực tế chức năng quản lý phòng	53
Hình 4.11 Giao diện thực tế của chức năng thêm thiết bị vào phòng	54
Hình 4.12 Giao diện thực tế của chức năng xem thiết bị của phòng	54
Hình 4.13 Giao diện thực tế của chức năng chuyển phòng	55
Hình 4.14 Giao diện thực tế của chức năng mượn trả thiết bị	56
Hình 4.15 Giao diện thực tế của chức năng chọn thiết bị để mượn	56
Hình 4.16 Giao diện thực tế của chức năng chọn giảng viên mượn thiết bị	57
Hình 4.17 Giao diện thực tế của chức năng chọn phòng mượn thiết bị	58
Hình 4.18 Giao diện thực tế của chức năng xem thông tin đơn mượn thiết bị	59
Hình 4.19 Giao diện thực tế chức năng thêm thủ công quà tặng	60
Hình 4.20 Giao diện thực tế của chức năng thêm quà tặng bằng file	60
Hình 4.21 Giao diện thực tế của danh sách quà tặng	61
Hình 4.22 Giao diện thực tế của chức năng xem thông tin chi tiết quà tặng	61
Hình 4.23 Giao diện thực tế của chức năng chỉnh sửa thông tin quà tặng	62
Hình 4.24 Giao diện thực tế của danh sách đơn yêu cầu quà tặng	62
Hình 4.25 Giao diện thực tế của modal thêm đơn yêu cầu quà tặng từ PDF	63
Hình 4.26 Giao diện thực tế của chọn quà tặng để mượn.....	63
Hình 4.27 Giao diện thực tế của chức năng chọn mục đích sử dụng và người yêu cầu	64
Hình 4.28 Giao diện thực tế chức năng xem thông tin đơn yêu cầu quà tặng	65
Hình 4.29 Giao diện cảnh báo khi nhấn vào nút duyệt đơn.....	65
Hình 4.30 Giao diện thực tế của đơn yêu cầu quà tặng có hoàn trả	66
Hình 4.31 Giao diện thực tế của chức năng thông kê quà tặng	66
Hình 4.32 Giao diện thực tế của chức năng thêm tài khoản người dùng	68
Hình 4.33 Giao diện thực tế chức năng chỉnh sửa thông tin tài khoản	69
Hình 4.34 Giao diện thực tế chức năng xem thông tin chi tiết tài khoản	69

Hình 4.35 Giao diện thực tế của tài khoản bị vô hiệu hoá.....	70
Hình 4.36 Giao diện thực tế của danh sách giảng viên.....	70
Hình 4.37 Giao diện thực tế chức năng thêm thủ công giảng viên.....	71
Hình 4.38 Giao diện thực tế của chức năng thêm giảng viên bằng file.....	71
Hình 4.39 Giao diện thực tế của chức năng chỉnh sửa, xoá giảng viên.....	72
Hình 4.40 Giao diện thực tế của chức năng thông báo.....	73
Hình 4.41 Mail thông báo khi được cấp tài khoản	74
Hình 4.42 Cây thư mục	75
Hình 4.43 Giao diện Terminal khi chạy dự án thành công	76

DANH MỤC CÁC BẢNG, SƠ ĐỒ

Bảng 2.1. Các thẻ HTML thông dụng	14
Bảng 3.1 Các chức năng trong hệ thống quản lý thiết bị và quà tặng	30
Bảng 3.2 Các công nghệ được lựa chọn sử dụng.....	32
Bảng 3.3 Bảng dữ liệu tài khoản người dùng	35
Bảng 3.4 Bảng dữ liệu thông tin người dùng.....	35
Bảng 3.5 Bảng dữ liệu yêu cầu mượn trả thiết bị	35
Bảng 3.6 Bảng dữ liệu thiết bị	36
Bảng 3.7 Bảng dữ liệu mục thiết bị	36
Bảng 3.8 Bảng dữ liệu quà tặng.....	37
Bảng 3.9 Bảng dữ liệu vị trí lưu trữ	37
Bảng 3.10 Bảng dữ liệu thông báo	37
Bảng 3.11 Bảng dữ liệu đơn yêu cầu quà tặng	38
Bảng 3.12 Bảng dữ liệu thông tin phòng	38
Bảng 3.13 Bảng dữ liệu giảng viên.....	38
Bảng 4.1 Bộ dữ liệu người dùng thử nghiệm	44
Bảng 4.2 Bộ dữ liệu giảng viên thử nghiệm	45
Bảng 4.3 Bộ dữ liệu thiết bị thử nghiệm.....	45
Bảng 4.4 Bộ dữ liệu phòng học thử nghiệm	46

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
API	Application Programming Interface – Giao diện lập trình ứng dụng
CQRS	Command and Query Responsibility Segregation – Tách biệt truy vấn và cập nhật
CRUD	Create, Read, Update and Delete – Tạo, đọc, cập nhật, xóa
CSS	Cascading Style Sheets – Ngôn ngữ tạo kiểu cho HTML
CSV	Comma-Separated Values – Dữ liệu giá trị phân cách bằng dấu phẩy
DB	Database – Cơ sở dữ liệu
HTML	HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản
HTTP	Hypertext Transfer Protocol – Giao thức truyền tải siêu văn bản
JSON	JavaScript Object Notation – Định dạng dữ liệu kiểu đối tượng
JSP	JavaServer Pages – Trang máy chủ Java
JWT	JSON Web Token – Chuỗi mã hóa xác thực dạng JSON
MVC	Model View Controller – Mô hình phân tách giao diện và xử lý
NoSQL	Not Only SQL – Cơ sở dữ liệu phi quan hệ
PDF	Portable Document Format – Định dạng tài liệu di động
UI/UX	User Interface / User Experience – Giao diện người dùng / Trải nghiệm người dùng
URL	Uniform Resource Locator – Định vị tài nguyên thống nhất

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1. Lý do chọn đề tài

Sự phát triển mạnh mẽ của công nghệ thông tin đã đặt ra yêu cầu cấp thiết về việc tự động hóa các quy trình quản lý, đặc biệt tại các trung tâm đào tạo như Trung tâm Ngoại ngữ - Tin học Victory. Việc quản lý thiết bị như máy tính, máy chiếu hay quà tặng khuyễn khích cho học viên chủ yếu được thực hiện thủ công, dẫn đến nhiều bất cập về thời gian, công sức và nguy cơ sai sót. Do đó, việc xây dựng một hệ thống quản lý tự động hóa không chỉ đáp ứng nhu cầu thực tiễn mà còn góp phần nâng cao hiệu quả vận hành và chất lượng dịch vụ của trung tâm.

Việc lựa chọn đề tài này xuất phát từ mong muốn ứng dụng các công nghệ hiện đại để giải quyết bài toán quản lý một cách tối ưu. Sử dụng Node.js và Express cho backend, MongoDB làm cơ sở dữ liệu, Handlebars cùng Bootstrap 5 cho giao diện người dùng, kết hợp với kiến trúc MVC và CQRS, hệ thống được thiết kế để đảm bảo tính mạnh mẽ, dễ bảo trì và khả năng mở rộng. Những công nghệ này không chỉ phù hợp với yêu cầu kỹ thuật mà còn mang lại giao diện thân thiện, trực quan, giúp nhân viên trung tâm dễ dàng theo dõi và quản lý thông tin một cách chính xác, minh bạch.

Hơn nữa, đề tài mang lại giá trị thực tiễn rõ rệt khi cung cấp một giải pháp công nghệ giúp tối ưu hóa quy trình làm việc tại Trung tâm Victory. Hệ thống không chỉ hỗ trợ quản lý thiết bị và quà tặng mà còn có tiềm năng mở rộng để quản lý các tài nguyên khác trong tương lai. Đồng thời, quá trình nghiên cứu và phát triển hệ thống là cơ hội để khám phá, ứng dụng các kiến trúc phần mềm tiên tiến, qua đó nâng cao kỹ năng lập trình và tư duy hệ thống. Với ý nghĩa thực tiễn và học thuật, đề tài mang đến một giải pháp hiệu quả, đáp ứng nhu cầu quản lý ngày càng cao của trung tâm và đóng góp vào sự phát triển bền vững.

1.2. Mục tiêu

Nghiên cứu các thức hoạt động và các tính năng cần thiết cho hệ thống quản lý thiết bị và quà tặng của Trung tâm Ngoại ngữ - Tin học Victory, hỗ trợ Trung tâm trong việc quản lý thiết bị - quà tặng tối ưu và hiệu quả.

1.3. Nội dung

- Tìm hiểu các tính năng cần thiết của một hệ thống quản lý thiết bị và quà tặng.

- Tìm hiểu các phương pháp thống kê, quản lý thiết bị bị hiệu quả, tối ưu hiệu suất hoạt động.
- Tìm hiểu mô hình MVC và CQRS, RestFul API.
- Nghiên cứu ngôn ngữ NodeJS, framework Express, hệ quản trị NoSQL MongoDB.
- Vận dụng kiến thức trên để thiết kế và xây dựng hệ thống quản lý thiết bị và quà tặng cho Trung tâm Ngoại ngữ - Tin học Victory.

1.4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Quy trình quản lý thiết bị và quà tặng của trung tâm Ngoại ngữ - Tin học Victory.
- Hệ thống quản lý thiết bị và quà tặng cho Trung tâm Ngoại ngữ - Tin học Victory, tập trung vào tự động hóa và tối ưu hóa quy trình quản lý.
- Các loại thiết bị như máy tính, máy chiếu, thiết bị giảng dạy và quà tặng khuyến khích dành cho học viên.
- Các công nghệ và kiến trúc phần mềm được ứng dụng, bao gồm: Node.js và Express cho backend, MongoDB làm cơ sở dữ liệu, Handlebars và Bootstrap 5 cho giao diện người dùng, kiến trúc MVC kết hợp CQRS để đảm bảo tính mô-đun và khả năng mở rộng.

Phạm vi nghiên cứu:

- Thiết kế và phát triển hệ thống quản lý đáp ứng nhu cầu thực tế của Trung tâm Ngoại ngữ - Tin học Victory.
- Triển khai hệ thống trên nền tảng web, sử dụng các công nghệ Node.js, Express, MongoDB, Handlebars, và Bootstrap 5.
- Áp dụng kiến trúc MVC và CQRS để tối ưu hóa cấu trúc và hiệu suất hệ thống.
- Đề tài giới hạn phạm vi không bao gồm tích hợp với các hệ thống bên ngoài, không quản lý các loại tài nguyên khác ngoài thiết bị và quà tặng, Không nghiên cứu sâu về tối ưu hóa hiệu năng hạ tầng hoặc tích hợp công nghệ trí tuệ nhân tạo.

1.5. Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

- Nghiên cứu các tài liệu về hệ thống quản lý thiết bị MISA.
- Nghiên cứu các tài liệu về NodeJS và Express.
- Nghiên cứu các tài liệu về hệ quản trị NoSQL MongoDB.
- Nghiên cứu các tài liệu về RestFul API.
- Nghiên cứu các tài liệu về mô hình MVC và SQRS.
- Nghiên cứu các tài liệu cần thiết về thiết kế và xây dựng website.

Phương pháp thực nghiệm:

- Xây dựng mô hình cơ sở dữ liệu và kiến trúc hệ thống.
- Cài đặt các nền tảng và công cụ cần thiết.
- Từng bước phát triển tính năng và giao diện cho hệ thống.
- Kiểm thử và triển khai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Mô hình MVC

2.1.1. Giới thiệu

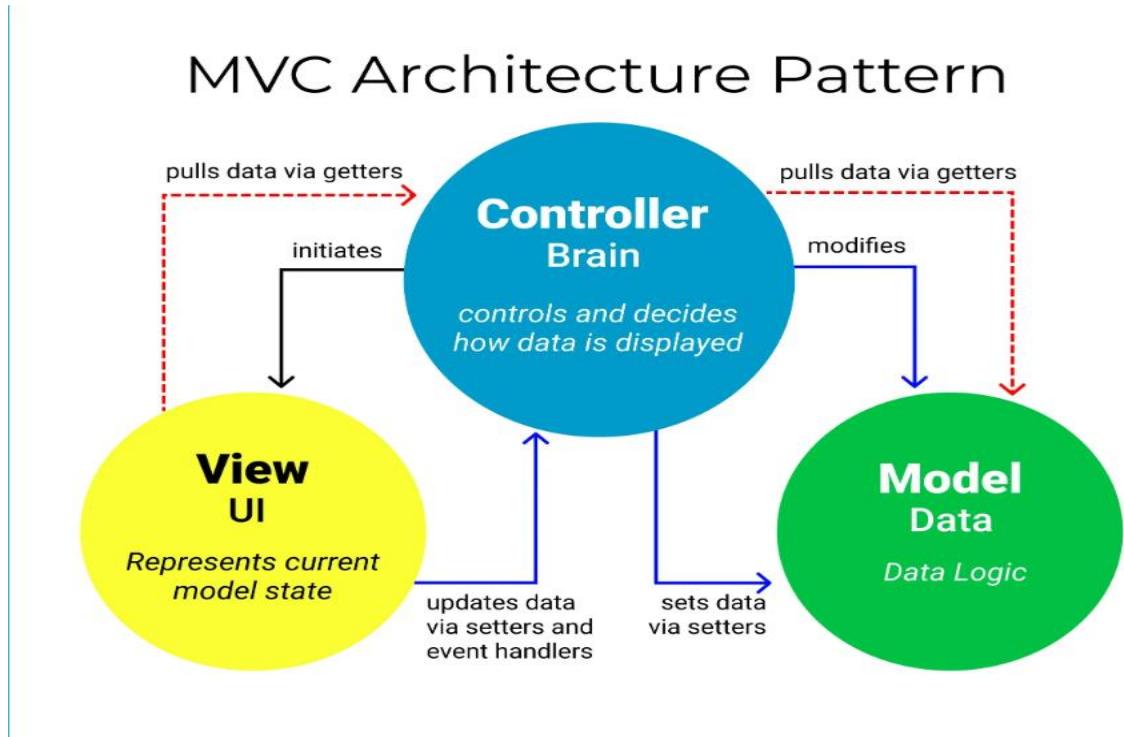
MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong việc phát triển ứng dụng để tạo ra các giao diện người dùng trực quan và có khả năng tương tác cao. MVC chia một ứng dụng thành 3 phần chính và mỗi phần có một vai trò riêng biệt:

Model đại diện cho dữ liệu và quy tắc nghiệp vụ của ứng dụng.

View chịu trách nhiệm hiển thị dữ liệu cho người dùng trực quan và tương tác.

Controller đóng vai trò là cầu nối giữa Model và View, xử lý các yêu cầu từ người dùng và cập nhật giao diện tương ứng.

2.1.2. Thành phần trong mô hình MVC



Hình 2.1 Các thành phần của mô hình MVC

Mô hình MVC gồm 3 thành phần bên trong không thể thiếu khi áp dụng mô hình này:

Model là lớp đại diện cho dữ liệu của ứng dụng. Nó có thể là một cơ sở dữ liệu, một file cấu hình hoặc một đối tượng phức tạp. Model chịu trách nhiệm lưu trữ, truy xuất và cập nhật dữ liệu.

View là lớp giao diện người dùng. Nó hiển thị dữ liệu từ Model cho người dùng và cho phép người dùng tương tác với ứng dụng. View thường được xây dựng bằng các ngôn ngữ template như HTML, JSP hoặc React.

Controller là lớp điều khiển luồng của ứng dụng. Nó nhận các yêu cầu từ người dùng, cập nhật Model và chọn View phù hợp để hiển thị.

Mối quan hệ giữa cả 3 như sau:

Controller: Cập nhật Model dựa trên các sự kiện từ View và chọn View phù hợp để hiển thị.

Model: Cung cấp dữ liệu cho View và Controller.

View: Hiển thị dữ liệu từ Model và gửi các sự kiện (event) đến Controller.

2.1.3. Luồng xử lý trong MVC

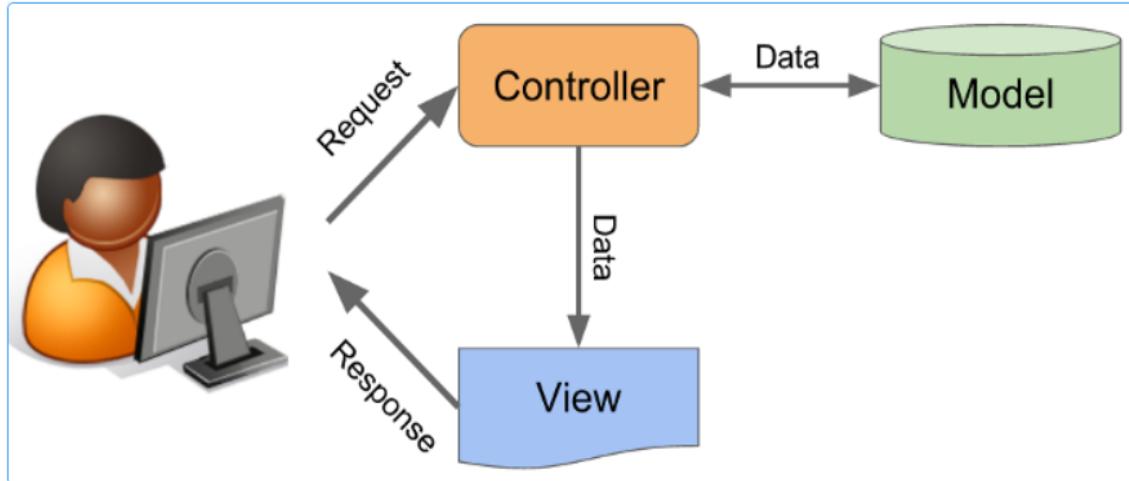
Luồng xử lý trong của mô hình MVC, có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.

Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.

Model chuẩn bị data và gửi lại cho Controller.

Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trả lại View và hiển thị cho người dùng trên trình duyệt.



Hình 2.2 Luồng xử lý trong MVC

2.1.4. Lý do chọn mô hình MVC

Mô hình MVC đã trở thành một tiêu chuẩn trong phát triển web hiện đại. Với cấu trúc phân chia rõ ràng giữa các thành phần mang đến nhiều lợi ích cho người sử dụng. Trong phần này, chúng ta sẽ tìm hiểu về tại sao nên áp dụng mô hình MVC vào dự án.

Quy trình phát triển nhanh hơn

MVC hỗ trợ phát triển nhanh chóng và song song. Nếu một mô hình MVC được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên View và một developer khác có thể làm việc với Controller để tạo logic nghiệp vụ cho ứng dụng web đó.

Do đó, ứng dụng mô hình MVC có thể được hoàn thành nhanh hơn ba lần so với các ứng dụng mô hình khác.

Khả năng cung cấp nhiều chế độ view

Trong mô hình MVC có thể tạo nhiều View cho chỉ một mô hình. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng MVC để phát triển chắc chắn là một giải pháp tuyệt vời.

Tuy nhiên, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.

Các sửa đổi không ảnh hưởng đến toàn bộ mô hình

Đối với bất kỳ ứng dụng web nào, người dùng có xu hướng thay đổi thường xuyên. Có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng...

Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh hưởng đến toàn bộ kiến trúc.

MVC Model trả về dữ liệu mà không cần định dạng

MVC pattern có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.

Ví dụ: tất cả loại dữ liệu đều có thể được định dạng bằng HTML. Ngoài ra, nó cũng có thể được định dạng bằng Macromedia Flash hay Dream Viewer.

Nền tảng MVC thân thiện với SEO

Nền tảng MVC hỗ trợ phát triển các trang web thân thiện với SEO. Nền nền tảng này có thể dễ dàng phát triển các URL thân thiện với SEO tạo ra nhiều lượt truy cập hơn.

Những ngôn ngữ như JavaScript hay jQuery có thể được tích hợp với MVC. Từ đó phát triển nhiều ứng dụng web giàu tính năng, đặc biệt là với mô hình MVC trong Java.

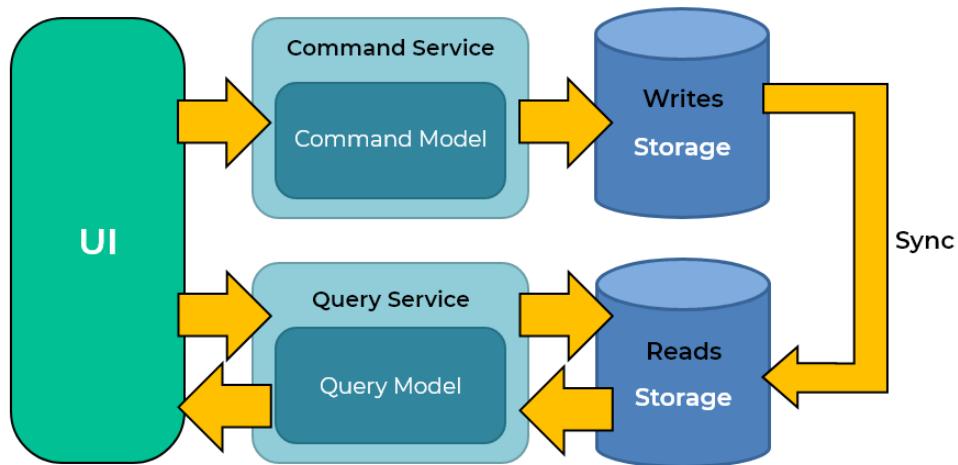
2.2. CQRS pattern

2.2.1. Giới thiệu

CQRS là viết tắt của Command and Query Responsibility Segregation. Bóc tách từng chữ ra thì Segregation là phân tách, Responsibility là trách nhiệm. Vậy CQRS có nghĩa là phân tách trách nhiệm giữa Command và Query.

2.2.2. Command và Query

CQRS pattern hướng mục tiêu chia tách Command và Query ra thành 2 phần khác biệt. Lưu ý là commands sẽ bao gồm các HTTP method: POST, PUT, DELETE và PATCH. Còn phần query theo đúng nghĩa tên gọi của gói, chỉ là GET method, lấy data từ DB.



Hình 2.3 Command và querry trong CQRS Pattern

2.2.3. Lý do sử dụng CQRS

CQRS pattern tách GET và các Http method còn lại ra thành hai phần. Mà GET sử dụng để truy vấn data, còn các method khác là để thao tác với dữ liệu.

Do nhu cầu giữa việc truy vấn dữ liệu và thao tác với dữ liệu khác nhau. Một ứng dụng có thể thực hiện cả triệu câu query GET hàng ngày nhưng chỉ thao tác insert vài câu.

Các tiếp cận cả Command và Queries trên cùng một services có thể gây ảnh hưởng tới performance.

Sự khác biệt về quyền được thực hiện Commands và Queries khác nhau đôi khi cũng làm services trở nên phức tạp.

Dữ liệu có thể bị xung đột khi thực hiện song song cả truy vấn và thao tác.

2.3. JWT

2.3.1. Giới thiệu

JSON Web Token (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

2.3.2. Thành phần của JWT

Header: Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

Payload: Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.

Reserved claims: Đây là một số metadata được định nghĩa trước, trong đó một số metadata là bắt buộc, số còn lại nên tuân theo để JWT hợp lệ và đầy đủ thông tin

Public Claims - Claims được cộng đồng công nhận và sử dụng rộng rãi.

Private Claims - Claims tự định nghĩa (không được trùng với Reserved Claims và Public Claims), được tạo ra để chia sẻ thông tin giữa 2 parties đã thỏa thuận và thống nhất trước đó.

Signature: Chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:

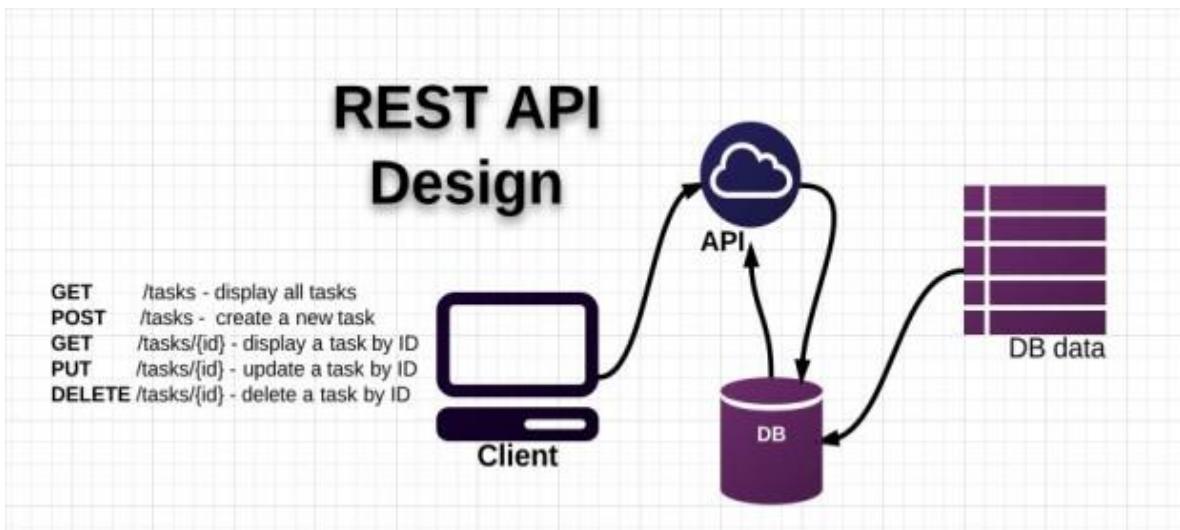
```
HMACSHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    secret)
```

Do bản thân Signature đã bao gồm cả header và payload nên Signature có thể dùng để kiểm tra tính toàn vẹn của dữ liệu khi truyền tải.

2.4. RESTful API

2.4.1. Tổng quan về RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 2.4 REST API Design

2.4.2. Các thành phần của RESTful API

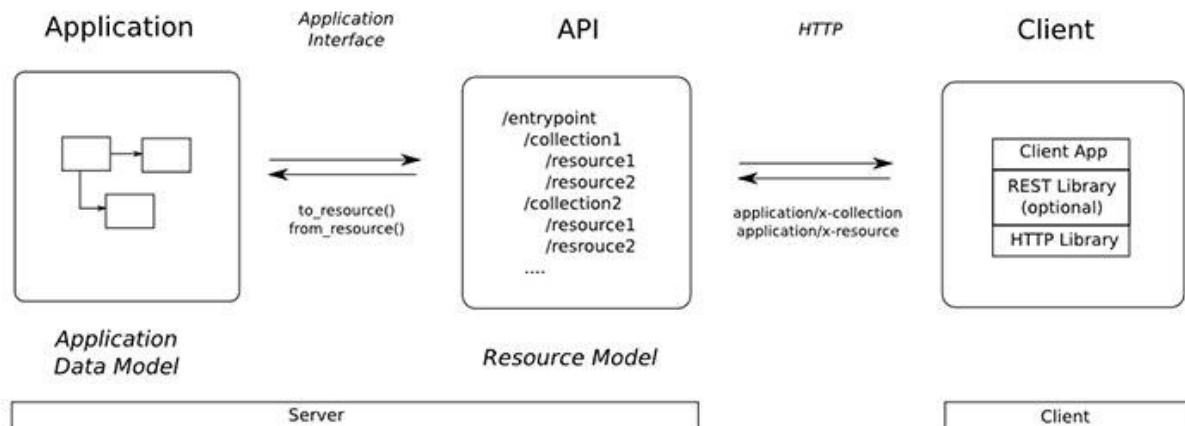
API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu cần cho ứng dụng ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản lý các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

2.4.3. RESTful hoạt động như thế nào



Hình 2.5 Cách thức vận hành của RESTful

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

POST (CREATE): Tạo mới một Resource.

PUT (UPDATE): Cập nhật thông tin cho Resource.

DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

2.4.4. Authentication và dữ liệu trả về

RESTful API không sử dụng session và cookie, nó sử dụng một `access_token` với mỗi request. Dữ liệu trả về thường có cấu trúc như sau:

```
{"data" : {  
    "id": "1",  
    "name": "TopDev"  
}  
}
```

Khi request một API nào đó thường thì sẽ có vài status code để nhận biết sau:

200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.

201 Created – Trả về khi một Resouce vừa được tạo thành công.

204 No Content – Trả về khi Resource xoá thành công.

304 Not Modified – Client có thể sử dụng dữ liệu cache.

400 Bad Request – Request không hợp lệ.

401 Unauthorized – Request cần có auth.

403 Forbidden – bị từ chối không cho phép.

404 Not Found – Không tìm thấy resource từ URI.

405 Method Not Allowed – Phương thức không cho phép với user hiện tại.

410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.

415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.

422 Unprocessable Entity – Dữ liệu không được xác thực.

429 Too Many Requests – Request bị từ chối do bị giới hạn.

2.5. Các công nghệ xây dựng trang web

2.5.1. Ngôn ngữ HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu chuẩn được sử dụng để xây dựng và định dạng nội dung trên các trang web. HTML lần đầu được phát triển vào năm 1993 bởi Tim Berners-Lee, người sáng tạo ra World Wide Web. HTML đã trải qua nhiều phiên bản và cập nhật, từ HTML 1.0 cho đến phiên bản hiện tại là HTML5 giúp cải thiện tính năng, khả năng tương tác và hỗ trợ các công nghệ hiện đại hơn trên web.

HTML rất quan trọng vì nó giúp các trình duyệt web hiểu và hiển thị nội dung theo đúng cấu trúc mà nhà phát triển mong muốn. Nó cũng hỗ trợ việc tạo ra các siêu liên kết (hyperlinks), cho phép người dùng điều hướng giữa các trang web một cách dễ dàng. Nhờ sự đơn giản và tính phổ biến, HTML là ngôn ngữ cốt lõi mà mọi nhà phát triển web đều phải

năm vững khi xây dựng trang web hoặc ứng dụng trực tuyến.

Vai trò chính của HTML là:

- *Tạo cấu trúc nội dung*: HTML xác định cách các phần tử (như đoạn văn, tiêu đề, liên kết, hình ảnh, v.v.) được tổ chức và hiển thị trên một trang web.
- *Kết hợp với CSS và JavaScript*: HTML cung cấp cấu trúc cơ bản, trong khi CSS định nghĩa về hình thức và JavaScript thêm các tính năng tương tác.

Một tài liệu HTML cơ bản thường có cấu trúc sau:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Example Heading</h1>
</body>
</html>
```

Các phần chính:

<!DOCTYPE html>: Khai báo kiểu tài liệu, thông báo cho trình duyệt biết rằng tài liệu này sử dụng HTML5.

<html>: Thủ gốc chứa toàn bộ nội dung của tài liệu HTML.

<head>: Chứa thông tin siêu dữ liệu về tài liệu như tiêu đề (hiển thị trên tab trình duyệt), charset và các liên kết đến tài nguyên như CSS.

<body>: Chứa nội dung thực sự hiển thị trên trang web, bao gồm các phần tử như văn bản, hình ảnh, liên kết, biểu mẫu,...

Các thẻ HTML thông dụng được trình bày trong bảng 2.1:

Bảng 2.1. Các thẻ HTML thông dụng

Tên thẻ	Mô tả, chức năng
<h1>...<h6>	Tiêu đề, từ h1 (tiêu đề chính) đến h6 (tiêu đề phụ nhỏ nhất)
<p>	Đoạn văn
<a>	Thẻ liên kết (anchor), cho phép tạo liên kết đến các trang khác hoặc phần khác của cùng một trang
	Thẻ nội tuyến dùng để gói một đoạn văn bản mà không ảnh hưởng đến cách dàn trang
<div>	Thẻ khống dùng để tạo một khống hoặc vùng trên trang
	Thẻ hiển thị hình ảnh
<video>	Thẻ nhúng video
	Danh sách không có thứ tự (bullet points)
	Danh sách có thứ tự (số hoặc chữ cái)
	Mục trong danh sách
<form>	Tạo biểu mẫu để nhập dữ liệu
<input>	Thẻ nhập liệu trong biểu mẫu như trường văn bản, nút radio, hộp kiểm tra
<textarea>	Vùng nhập liệu nhiều dòng
<button>	Nút bấm
<table>	Tạo bảng
<tr>	Hàng trong bảng
<td>	Ô dữ liệu trong bảng
<th>	Ô tiêu đề cột hoặc hàng

<header>	Phần đầu trang hoặc đầu của một phần
<footer>	Phần cuối trang hoặc cuối của một phần
<article>	Một bài viết hoặc khối nội dung độc lập
<section>	Một phần hoặc khối nội dung có chủ đề riêng
<nav>	Điều hướng, thường chứa liên kết đến các phần khác của trang hoặc trang web
<canvas>	Tạo không gian đồ họa động, hỗ trợ vẽ đồ họa 2D và 3D thông qua JavaScript

2.5.2. Bảng định kiểu CSS

CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để mô tả cách trình bày, định dạng và bố cục của các phần tử trong một tài liệu HTML. Trong khi HTML định nghĩa cấu trúc và nội dung của trang web, CSS được sử dụng để kiểm soát giao diện của trang, bao gồm màu sắc, phông chữ, khoảng cách, kích thước, vị trí của các phần tử và nhiều thuộc tính khác.

CSS hoạt động bằng cách áp dụng các quy tắc (rules) lên các phần tử HTML. Mỗi quy tắc bao gồm một bộ chọn (selector) để chọn các phần tử cụ thể và một tập hợp các thuộc tính (properties) cùng giá trị (values) để định nghĩa kiểu dáng cho các phần tử đó.

Một số đặc điểm chính của CSS:

- *Tách biệt nội dung và trình bày*: CSS giúp tách riêng phần trình bày ra khỏi nội dung giúp việc quản lý và thay đổi giao diện trang web trở nên dễ dàng hơn mà không ảnh hưởng đến cấu trúc HTML.

- *Thừa kế và phân tầng (Cascading)*: CSS tuân theo nguyên tắc phân tầng, nghĩa là khi có nhiều quy tắc được áp dụng lên cùng một phần tử, các quy tắc có độ ưu tiên cao hơn sẽ được áp dụng. CSS cũng hỗ trợ thừa kế, cho phép các thuộc tính của một phần tử cha được kế thừa bởi các phần tử con.

- *Định dạng đa dạng*: CSS cho phép định dạng phong phú và linh hoạt như màu sắc, phông chữ, căn lề, khoảng cách giữa các phần tử, độ rộng, độ cao, bo tròn góc, hiệu ứng đổ

bóng, hiệu ứng chuyển động và nhiều thuộc tính khác.

- *Khả năng đáp ứng (Responsive)*: CSS hỗ trợ các kỹ thuật tạo ra giao diện linh hoạt, phù hợp với nhiều kích thước màn hình khác nhau, từ điện thoại di động đến máy tính để bàn, thông qua việc sử dụng các đơn vị đo lường linh hoạt và các truy vấn phương tiện (media queries).

Có ba cách chính để tích hợp CSS vào tài liệu HTML:

- *Inline CSS*: Được định nghĩa trực tiếp trong thuộc tính style của một thẻ HTML.
- *Internal CSS*: Được đặt trong thẻ `<style>` trong phần `<head>` của tài liệu HTML.
- *External CSS*: CSS được viết trong một file riêng biệt với đuôi `.css` và liên kết với tài liệu HTML bằng thẻ `<link>`.

2.5.3. Ngôn ngữ mở rộng của CSS

SASS/SCSS là một chương trình tiền xử lý CSS (*CSS preprocessor*). Nó hỗ trợ viết CSS theo cách của một ngôn ngữ lập trình, có cấu trúc rõ ràng, ràng buộc, dễ phát triển và bảo trì code hơn. Ngoài ra nó có rất nhiều các thư viện hỗ trợ kèm theo hỗ trợ viết code CSS một cách dễ dàng vào đơn giản hơn. Có rất nhiều loại CSS Preprocessor trong đó bao gồm SASS, Stylus hay LESS.

SASS và SCSS về bản chất vẫn đề là giống nhau, chỉ khác nhau ở cách viết. Sass là chữ viết tắt của Syntactically Awesome Style Sheets, chương trình tiền xử lý bằng ngôn ngữ kịch bản (Preprocessor Scripting Language), sẽ được biên dịch thành CSS. Nghĩa là, xây dựng style bằng SASS, rồi SASS sẽ render thành file CSS.

SASS bản thân có hai kiểu viết khác nhau, một kiểu như là HAML, Pug – Sử dụng indent (cách thụt đầu dòng) để phân tách các khối code, sử dụng xuống dòng để phân biệt rules, có phần mở rộng là `.sass`.

```
.header
  color: #f3f3f3;

.header_inner
  border: 1px solid #f3f3f3
```

2.5.4. Ngôn ngữ Javascript

JavaScript là một ngôn ngữ lập trình động, phổ biến và mạnh mẽ được sử dụng rộng rãi để phát triển các ứng dụng web, ứng dụng di động, trò chơi và nhiều loại phần mềm khác. Ban đầu, JavaScript được phát triển vào năm 1995 bởi Brendan Eich tại Netscape nhằm bổ sung khả năng tương tác cho các trang web nhưng kể từ đó, nó đã phát triển và trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Tính năng nổi bật của JavaScript

- *Ngôn ngữ lập trình phía client*: JavaScript chủ yếu được sử dụng để lập trình phía client (trên trình duyệt web) giúp tạo ra các tương tác động trên giao diện người dùng mà không cần phải tải lại toàn bộ trang web. Nó cho phép các nhà phát triển tạo ra các ứng dụng web phong phú và tương tác từ việc kiểm tra đầu vào của người dùng đến thay đổi giao diện một cách động.

- *Tính năng đa nền tảng*: JavaScript hoạt động trên hầu hết các trình duyệt web hiện đại mà không cần phải cài đặt thêm bất kỳ phần mềm hoặc plugin nào. Điều này giúp nó trở thành một ngôn ngữ đa nền tảng, cho phép viết một lần và chạy trên nhiều hệ điều hành và trình duyệt khác nhau.

- *Hỗ trợ lập trình hướng đối tượng*: JavaScript là một ngôn ngữ lập trình hỗ trợ lập trình hướng đối tượng, cho phép các nhà phát triển tạo ra các đối tượng, kế thừa và tái sử dụng mã. Các khái niệm như lớp (class), đối tượng (object), kế thừa (inheritance) đều có thể được áp dụng trong JavaScript thông qua phiên bản ES6 trở về sau.

- *Lập trình sự kiện*: JavaScript cung cấp khả năng lập trình sự kiện giúp các ứng dụng web có thể phản hồi lại các hành động của người dùng như nhấp chuột, di chuyển chuột, nhập liệu trên bàn phím hoặc các sự kiện khác. Điều này giúp tạo ra trải nghiệm người dùng linh hoạt và tương tác.

- *Ngôn ngữ không đồng bộ*: JavaScript có khả năng xử lý các tác vụ không đồng bộ một cách hiệu quả thông qua các tính năng như callback, promises và async/await. Điều này giúp nó xử lý tốt các tác vụ I/O (Input/Output) như gọi API, đọc file mà không làm gián đoạn các tác vụ khác.

- *Khả năng mở rộng mạnh mẽ*: Ngoài việc lập trình phía client, với sự ra đời của Node.js,

JavaScript đã mở rộng sang lĩnh vực lập trình phía server, cho phép xây dựng các ứng dụng server-side mạnh mẽ và hiệu quả. Node.js cũng cung cấp một môi trường thực thi JavaScript trên máy chủ giúp tạo ra các ứng dụng thời gian thực như chat và streaming.

Ưu điểm của JavaScript

- *Tốc độ*: Vì JavaScript là ngôn ngữ được thực thi trực tiếp trên trình duyệt mà không cần biên dịch trước nên nó rất nhanh trong việc thực thi mã và phản hồi lại các hành động của người dùng.

- *Dễ học*: Cú pháp của JavaScript tương đối đơn giản và dễ tiếp cận, đặc biệt là đối với những người mới học lập trình.

- *Khả năng tương tác mạnh mẽ*: JavaScript dễ dàng tương tác với các ngôn ngữ và công nghệ khác như HTML và CSS để tạo ra các ứng dụng web phức tạp.

- *Hệ sinh thái phong phú*: JavaScript có một hệ sinh thái phong phú với nhiều thư viện và framework nổi tiếng như React, Angular, Vue.js giúp đơn giản hóa việc phát triển các ứng dụng lớn.

Một số thách thức của JavaScript

- *Hiệu năng trong các ứng dụng lớn*: Khi ứng dụng web lớn dần, quản lý mã JavaScript trở nên phức tạp, điều này có thể ảnh hưởng đến hiệu năng nếu không được tối ưu hóa đúng cách.

- *Tính không đồng bộ*: Mặc dù JavaScript có khả năng xử lý các tác vụ không đồng bộ, việc quản lý các tác vụ này đôi khi có thể phức tạp, đặc biệt là với những người mới làm quen.

Với khả năng lập trình linh hoạt và đa năng, JavaScript không chỉ giúp tạo ra các trang web tương tác mà còn mở rộng sang lĩnh vực phát triển di động, desktop và server. Hệ sinh thái JavaScript không ngừng phát triển giúp các nhà phát triển có thêm nhiều công cụ mạnh mẽ để xây dựng các ứng dụng phức tạp, hiệu quả và mang tính tương tác cao.

2.5.5. Bootstrap Framework

Bootstrap là một framework front-end mã nguồn mở được thiết kế bởi Twitter, giúp các nhà phát triển dễ dàng xây dựng giao diện người dùng (UI) cho các trang web và ứng

dụng di động. Ra đời năm 2011, Bootstrap nhanh chóng trở thành công cụ không thể thiếu trong cộng đồng lập trình web nhờ khả năng đơn giản hóa quy trình thiết kế và hỗ trợ tương thích tốt trên nhiều nền tảng.

Framework này cung cấp một bộ thành phần giao diện như nút, biểu mẫu, thanh điều hướng, bảng, và hệ thống lưới linh hoạt, giúp tạo nên các trang web hiện đại và tối ưu cho mọi kích thước màn hình. Bootstrap sử dụng HTML, CSS và JavaScript để mang lại trải nghiệm người dùng mượt mà và đồng nhất trên nhiều trình duyệt và thiết bị.

Điểm nổi bật của Bootstrap là khả năng tùy chỉnh mạnh mẽ, cho phép nhà phát triển điều chỉnh các thành phần để phù hợp với yêu cầu thiết kế cụ thể của dự án. Nhờ vào tính đơn giản, hiệu quả và linh hoạt, Bootstrap trở thành lựa chọn ưa thích không chỉ cho người mới học mà còn cho các lập trình viên chuyên nghiệp trong việc xây dựng giao diện nhanh chóng và hiệu quả.

Cài đặt Bootstrap

Bootstrap có thể được cài đặt theo hai cách:

Tải xuống thư viện: Bootstrap có thể được tải xuống dưới dạng thư viện CSS và JavaScript tại <https://getbootstrap.com/docs/5.0/getting-started/download/>.

Sử dụng CDN: Bootstrap có thể được sử dụng thông qua CDN (Content Delivery Network).

```
<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c  
ss"           rel="stylesheet"           integrity="sha384-  
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohpuuC0mLASjC"  
crossorigin="anonymous">  
  
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.  
min.js"           integrity="sha384-  
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtTaxVXM"  
crossorigin="anonymous"></script>
```

Các thành phần chính của Bootstrap

Định dạng văn bản – Typography: Bootstrap cung cấp các lớp CSS để định dạng văn bản trên trang web.

Thiết kế biểu mẫu – Form: Bootstrap cung cấp các lớp CSS để tạo các biểu mẫu web dễ sử dụng. Buttons: Bootstrap cung cấp các lớp CSS để tạo các nút bấm web hấp dẫn.

Cú pháp:

```
<button type="button"> Án </button>
```

Ví dụ: tạo 1 button để đăng nhập với class **.butttton**

Sử dụng class **.btn-lg** cho các nút lớn hoặc **.btn-sm** cho các nút nhỏ

class **.btn-primary** là lớp màu sắc của nút, trong trường hợp này là màu chủ đạo (primary color), thường là màu xanh dương.

Có thể thay thế class **.primary** bằng các giá trị màu sắc khác như **.success**, **.danger**, **.warning**, **.info** để đặt màu sắc khác cho nút.

```
<button type="button" class="btn btn-primary">Đăng Nhập</button>
```

Tables: Bootstrap cung cấp các lớp CSS để tạo các bảng web dễ đọc.

Cú pháp:

```
<table class="table"> </table>
```

Ví dụ: Tạo table sinh viên sử dụng class **.table**

```
<div>

    <h2>Student Table</h2>

    <table class="table">

        <thead>

            <tr>

                <th scope="col">#</th>
                <th scope="col">Student ID</th>
                <th scope="col">Name</th>
                <th scope="col">Age</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>S123456789</td>
                <td>John Doe</td>
                <td>22</td>
            </tr>
            <tr>
                <td>2</td>
                <td>S123456780</td>
                <td>Jane Doe</td>
                <td>23</td>
            </tr>
        </tbody>
    </table>
</div>
```

```

        </tr>

    </thead>

    <tbody>

        <tr>

            <th scope="row">1</th>

            <td>101</td>

            <td>John Doe</td>

            <td>20</td>

        </tr>

        <tr>

            <th scope="row">2</th>

            <td>102</td>

            <td>Jane Smith</td>

            <td>22</td>

        </tr>

    </tbody>

</table>

</div>

```

Navigationbar: Bootstrap cung cấp các lớp CSS để tạo các thanh điều hướng web hiệu quả.

Cú pháp

```
<nav class="navbar-nav"> </nav>
```

Ví dụ: tạo một navigationbar cho trang quản lý sản phẩm với class **.navbar**

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

    <ul class="navbar-nav">

        <li class="nav-item active">
```

```

        <a class="nav-link" href="#">Trang chủ</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">Thể loại</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">liên hệ</a>
    </li>
</ul>
</nav>

```

Tài liệu Bootstrap

Bootstrap có tài liệu phong phú và chi tiết, bao gồm hướng dẫn sử dụng, ví dụ và hướng dẫn. Tài liệu có sẵn trực tuyến và có thể được tải xuống dưới dạng PDF.

Tóm lại, Bootstrap là một framework mã nguồn mở mạnh mẽ và linh hoạt có thể giúp các nhà phát triển tạo các trang web và ứng dụng web đáp ứng nhanh chóng và dễ dàng.

2.5.6. Node.js

Node.js là một môi trường chạy JavaScript phía máy chủ (server-side) dựa trên nền tảng V8 JavaScript Engine của Google Chrome. Được phát triển bởi Ryan Dahl vào năm 2009, Node.js cho phép các nhà phát triển viết mã JavaScript để chạy trên máy chủ, mở rộng khả năng của JavaScript từ việc chỉ chạy trên trình duyệt web sang việc xây dựng các ứng dụng máy chủ (server-side applications).

Đặc điểm chính của Node.js:

- *Nền tảng phi đồng bộ và không chặn (Asynchronous and Non-blocking)*: Node.js sử dụng mô hình phi đồng bộ và không chặn, cho phép xử lý nhiều yêu cầu đồng thời mà không phải đợi các hoạt động I/O (như đọc/ghi tệp tin, truy vấn cơ sở dữ liệu) hoàn tất. Điều này giúp tăng cường hiệu suất và khả năng mở rộng của các ứng dụng web, đặc biệt là trong các ứng dụng cần xử lý lượng lớn dữ liệu hoặc nhiều kết nối đồng thời.

- *Mô hình sự kiện (Event-driven)*: Node.js dựa trên mô hình sự kiện, trong đó các hoạt động như xử lý yêu cầu HTTP, đọc tệp tin và truy vấn cơ sở dữ liệu được thực hiện qua các sự kiện và callback functions. Điều này giúp quản lý hiệu quả các tác vụ bất đồng bộ và xử lý nhiều kết nối đồng thời mà không làm giảm hiệu suất của ứng dụng.

- *Cung cấp một môi trường chạy JavaScript trên máy chủ*: Trước khi Node.js ra đời, JavaScript chủ yếu được sử dụng trên trình duyệt web. Node.js mở rộng khả năng của JavaScript để chạy trên máy chủ, cho phép phát triển các ứng dụng web, dịch vụ API và các ứng dụng mạng một cách đồng nhất bằng một ngôn ngữ lập trình duy nhất.

- *Quản lý gói qua npm (Node Package Manager)*: Node.js đi kèm với npm, một hệ thống quản lý gói mạnh mẽ cho phép các nhà phát triển dễ dàng cài đặt, quản lý và chia sẻ các thư viện và công cụ. npm có một kho lưu trữ khổng lồ với hàng triệu gói phần mềm giúp tiết kiệm thời gian phát triển và tăng cường khả năng mở rộng của các ứng dụng.

- *Mô hình đơn luồng (Single-threaded)*: Node.js sử dụng mô hình đơn luồng với vòng lặp sự kiện (event loop) để xử lý các tác vụ. Mặc dù Node.js chỉ sử dụng một luồng chính nhưng nó có thể xử lý hàng triệu kết nối đồng thời nhờ vào cơ chế phi đồng bộ và không chặn.

- *Hiệu suất cao với V8 Engine*: Node.js dựa trên V8 JavaScript Engine của Google, nổi tiếng với hiệu suất cao trong việc thực thi mã JavaScript. V8 thực hiện biên dịch mã JavaScript thành mã máy (machine code) giúp tăng tốc việc thực thi và cải thiện hiệu suất tổng thể của ứng dụng.

Các ứng dụng phổ biến của Node.js:

- *Ứng dụng web*: Node.js rất phù hợp để phát triển các ứng dụng web, bao gồm cả các trang web tĩnh và động, nhờ vào khả năng xử lý nhiều yêu cầu đồng thời và hỗ trợ các thư viện web như Express.js.

- *Dịch vụ API*: Node.js được sử dụng để xây dựng các dịch vụ API RESTful hoặc GraphQL, nhờ vào khả năng xử lý các yêu cầu HTTP và tương tác với cơ sở dữ liệu một cách hiệu quả.

- *Ứng dụng mạng thời gian thực*: Node.js thường được sử dụng trong các ứng dụng mạng thời gian thực như chat rooms, gaming servers và các ứng dụng video streaming, nhờ vào

khả năng xử lý các kết nối đồng thời và sự kiện.

- *Công cụ dòng lệnh (CLI tools)*: Node.js cũng được sử dụng để xây dựng các công cụ dòng lệnh mạnh mẽ, nhờ vào khả năng tương tác với hệ thống tệp tin và thực thi các lệnh từ dòng lệnh.

Node.js đã thay đổi cách các nhà phát triển xây dựng và triển khai ứng dụng máy chủ bằng cách mang lại một môi trường phát triển đồng nhất và hiệu suất cao. Với khả năng xử lý đồng thời tốt, mô hình sự kiện mạnh mẽ và hệ sinh thái phong phú, Node.js là một lựa chọn lý tưởng cho việc phát triển các ứng dụng web hiện đại, dịch vụ API và nhiều ứng dụng khác trong môi trường máy chủ.

2.5.7. Express

Express hay Express.js là một framework ứng dụng web cho Node.js, được phát triển để tạo ra các ứng dụng web và dịch vụ API một cách dễ dàng và nhanh chóng. Được phát hành lần đầu tiên vào năm 2010, Express đã trở thành một trong những framework phổ biến nhất cho Node.js nhờ vào sự đơn giản, hiệu quả và tính mở rộng của nó.

Đặc điểm nổi bật của Express.js:

- *Thiết kế đơn giản và nhẹ nhàng*: Express cung cấp một cấu trúc nhẹ nhàng và đơn giản giúp các nhà phát triển nhanh chóng thiết lập và triển khai các ứng dụng web và dịch vụ API. Với ít cấu hình cần thiết, Express giúp giảm bớt sự phức tạp trong việc phát triển và quản lý ứng dụng, tập trung vào việc cung cấp các tính năng cơ bản và linh hoạt để phát triển ứng dụng.

- *Mô hình Middleware mạnh mẽ*: Một trong những tính năng mạnh mẽ của Express là hệ thống middleware. Middleware là các hàm xử lý yêu cầu HTTP (request) và phản hồi (response) mà có thể được chèn vào giữa các bước xử lý của một ứng dụng. Điều này cho phép các nhà phát triển thực hiện các tác vụ như xác thực, xử lý lỗi, phân tích dữ liệu và nhiều chức năng khác một cách dễ dàng và có tổ chức. Middleware có thể được sử dụng toàn cục hoặc cho các tuyến đường (routes) cụ thể, mang lại khả năng kiểm soát và tùy chỉnh cao.

- *Quản lý tuyến đường (Routing)*: Express cung cấp một hệ thống quản lý tuyến đường linh hoạt, cho phép định nghĩa các tuyến đường HTTP cho các phương thức như GET,

POST, PUT, DELETE và các phương thức khác. Chúng ta có thể định nghĩa các tuyến đường với các tham số động và xử lý các yêu cầu từ người dùng một cách rõ ràng và có tổ chức. Express hỗ trợ việc phân chia các tuyến đường thành các tệp riêng biệt giúp quản lý mã nguồn trở nên dễ dàng hơn.

- *Tính mở rộng cao*: Express có khả năng mở rộng tốt thông qua việc tích hợp với các thư viện và công cụ khác. Chúng ta có thể dễ dàng kết hợp Express với các thư viện như Mongoose để tương tác với MongoDB, Passport để xử lý xác thực người dùng hoặc các công cụ khác để tăng cường chức năng của ứng dụng. Điều này giúp chúng ta xây dựng các ứng dụng phức tạp hơn mà vẫn giữ cho mã nguồn dễ quản lý và bảo trì.

- *Hỗ trợ nhiều engine template*: Express hỗ trợ nhiều engine template khác nhau như Pug (trước đây là Jade), EJS và Handlebars. Điều này giúp người dùng dễ dàng tạo ra các trang web động bằng cách kết hợp dữ liệu từ máy chủ với các mẫu giao diện. Việc tích hợp các engine template giúp tăng cường khả năng phát triển giao diện người dùng một cách linh hoạt và hiệu quả.

- *Quản lý yêu cầu và phản hồi (Request/Response)*: Express cung cấp các đối tượng yêu cầu (request) và phản hồi (response) để sử dụng giúp xử lý dữ liệu gửi từ người dùng và gửi dữ liệu phản hồi một cách hiệu quả. Chúng ta có thể dễ dàng truy cập các tham số của yêu cầu, xử lý dữ liệu từ biểu mẫu và gửi phản hồi với các mã trạng thái HTTP khác nhau, tiêu đề và nội dung.

- *Hỗ trợ xử lý lỗi (Error Handling)*: Express cung cấp cơ chế xử lý lỗi tích hợp, cho phép người dùng dễ dàng quản lý và phản hồi các lỗi xảy ra trong ứng dụng. Chúng ta có thể định nghĩa các hàm xử lý lỗi middleware để xử lý các lỗi phát sinh trong các bước xử lý yêu cầu, từ đó cung cấp thông tin lỗi chi tiết và thân thiện với người dùng.

- *Tích hợp với cơ sở dữ liệu*: Express có thể được tích hợp với nhiều hệ quản trị cơ sở dữ liệu (DBMS) khác nhau như MongoDB, MySQL, PostgreSQL và SQLite thông qua các thư viện và ORM (Object-Relational Mapping). Điều này giúp dễ dàng kết nối và tương tác với cơ sở dữ liệu, thực hiện các thao tác như lưu trữ, truy vấn và quản lý dữ liệu một cách hiệu quả.

Lợi ích của Express:

- *Nhanh chóng và dễ dàng để bắt đầu*: Với một cấu trúc đơn giản, Express.js được thiết kế với cấu trúc tối giản và linh hoạt, không yêu cầu cấu hình phức tạp hoặc thiết lập môi trường nặng nề. Các nhà phát triển có thể dễ dàng tạo ra một ứng dụng web cơ bản chỉ với vài dòng mã. Điều này giúp giảm thời gian học tập và triển khai cho những người mới bắt đầu, đồng thời giúp các dự án nhỏ được triển khai nhanh chóng.

- *Tài liệu phong phú*: Express cung cấp tài liệu chi tiết và dễ hiểu, bao gồm hướng dẫn sử dụng, các ví dụ mã nguồn và các bài viết giải thích về các tính năng chính. Điều này giúp các nhà phát triển nắm bắt nhanh chóng cách sử dụng các tính năng của Express và giải quyết các vấn đề mà họ gặp phải trong quá trình phát triển. Tài liệu phong phú giúp các nhà phát triển, đặc biệt là những người mới, có thể nhanh chóng làm quen và áp dụng Express vào dự án của mình.

- *Hỗ trợ nhiều middleware*: Express.js cho phép sử dụng các middleware để xử lý yêu cầu và phản hồi, thực hiện các tác vụ như xác thực, xử lý lỗi và phân tích dữ liệu. Các middleware có thể được sử dụng để thêm chức năng mới vào ứng dụng mà không cần phải thay đổi mã nguồn chính. Điều này giúp ứng dụng có thể mở rộng dễ dàng theo nhu cầu, chẳng hạn như thêm các tính năng bảo mật hoặc phân tích người dùng.

- *Dễ dàng tích hợp với thư viện và công cụ khác*: Express có khả năng tích hợp với nhiều thư viện và công cụ khác, chẳng hạn như cơ sở dữ liệu, hệ thống xác thực và các công cụ phân tích. Chúng ta có thể sử dụng các thư viện bên ngoài như Mongoose để kết nối và tương tác với MongoDB, Passport để xử lý xác thực người dùng hoặc các công cụ khác để nâng cao chức năng của ứng dụng. Điều này giúp người dùng có thể xây dựng các ứng dụng phức tạp và tùy chỉnh theo nhu cầu cụ thể của dự án.

- *Quản lý yêu cầu và phản hồi hiệu quả*: Nhờ vào sự đơn giản trong cách Express xử lý các yêu cầu và phản hồi, việc xây dựng các API và ứng dụng web trở nên trực quan và dễ quản lý. Chúng ta có thể dễ dàng định nghĩa các tuyến đường (routes) và các hàm xử lý (handlers) để xử lý các yêu cầu từ người dùng giúp giảm thiểu sự phức tạp và tăng hiệu quả phát triển ứng dụng.

- *Cộng đồng lớn*: Express.js có một cộng đồng người dùng và nhà phát triển rộng lớn và

năng động. Cộng đồng này đóng góp vào việc phát triển và duy trì Express, đồng thời cung cấp hỗ trợ qua các diễn đàn, nhóm thảo luận và các mạng xã hội. Việc có một cộng đồng lớn giúp đảm bảo rằng có thể dễ dàng tìm kiếm sự trợ giúp và giải pháp cho các vấn đề người dùng gặp phải.

- *Tài liệu phong phú và hỗ trợ*: Tài liệu của Express bao gồm hướng dẫn chi tiết, các ví dụ mã nguồn và các bài viết hướng dẫn từ cơ bản đến nâng cao. Ngoài ra, nhiều trang web, blog và khóa học trực tuyến cung cấp các hướng dẫn và mẹo bổ sung để giúp lập trình viên khai thác tối đa các tính năng của Express. Việc có tài liệu phong phú và sự hỗ trợ từ cộng đồng giúp giảm bớt khó khăn trong việc học và phát triển ứng dụng, đồng thời cung cấp nhiều nguồn tài nguyên để giải quyết các vấn đề kỹ thuật.

Express phát huy được sức mạnh và sự linh hoạt của mình trong việc xây dựng các ứng dụng web và dịch vụ API bằng Node.js. Với thiết kế đơn giản, khả năng mở rộng cao và tính dễ sử dụng, Express là một lựa chọn lý tưởng cho các nhà phát triển khi xây dựng các ứng dụng từ nhỏ đến lớn. Việc kết hợp Express với Node.js cung cấp một nền tảng mạnh mẽ và hiệu quả cho việc phát triển các ứng dụng máy chủ hiện đại.

2.5.8. Hệ quản trị cơ sở dữ liệu MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, được sử dụng để lưu trữ dữ liệu dạng tài liệu. MongoDB được phát triển bởi MongoDB Inc. và được cấp phép theo Giấy phép Công cộng phía Máy chủ (SSPL).

MongoDB lưu trữ dữ liệu trong các tài liệu JSON. Tài liệu JSON là một tập hợp các cặp khóa-giá trị, trong đó khóa là một chuỗi và giá trị có thể là một chuỗi, số, mảng, hoặc đối tượng JSON.

MongoDB sử dụng một mô hình tài liệu linh hoạt, cho phép lưu trữ dữ liệu theo bất kỳ cách nào phù hợp với nhu cầu. Điều này làm cho MongoDB trở nên linh hoạt và dễ sử dụng cho nhiều loại ứng dụng khác nhau.

Một số tính năng chính của MongoDB bao gồm:

Cơ sở dữ liệu hướng tài liệu: MongoDB lưu trữ dữ liệu trong các tài liệu JSON linh hoạt.

Cấu trúc dữ liệu linh hoạt: MongoDB cho phép lưu trữ dữ liệu theo bất kỳ cách nào phù hợp với nhu cầu.

Tính khả mở: MongoDB có thể được sử dụng trên nhiều nền tảng khác nhau, bao gồm Linux, macOS, và Windows.

Tính sẵn sàng cao: MongoDB có thể được sử dụng để tạo các ứng dụng có sẵn cao.

Tính bảo mật: MongoDB cung cấp các tính năng bảo mật mạnh mẽ để bảo vệ dữ liệu.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả bài toán

Trung tâm Ngoại ngữ - Tin học Victory có nhu cầu xây dựng một hệ thống quản lý thiết bị dạy học và quà thưởng nhằm tự động hóa, tối ưu hóa quy trình quản lý, đồng thời đảm bảo tính chính xác và minh bạch. Hệ thống được thiết kế để hỗ trợ quản lý các thiết bị như máy tính, máy chiếu, bảng trắng và quà tặng khuyến khích dành cho học viên, đối tác hoặc khách đến làm việc. Với mục tiêu nâng cao hiệu quả vận hành, hệ thống cung cấp các tính năng quản lý thiết bị theo phòng học, chuyển thiết bị giữa các phòng, theo dõi tình trạng thiết bị, quản lý mượn trả, cũng như thông kê và báo cáo chi tiết về thiết bị và quà thưởng. Hệ thống còn tích hợp phân quyền người dùng, tìm kiếm thiết bị nhanh chóng và thông báo khi kho thiết bị hoặc quà thưởng sắp cạn.

Để đáp ứng yêu cầu quản lý nhân sự và tài khoản, hệ thống cho phép quản lý tài khoản người dùng và thông tin giảng viên một cách linh hoạt. Người quản trị có thể thêm, sửa, xóa tài khoản người dùng, quản lý trạng thái tài khoản như kích hoạt hoặc khóa khi cần thiết. Tương tự, hệ thống hỗ trợ thêm, sửa, xóa thông tin giảng viên, cho phép nhập danh sách giảng viên qua file CSV và cung cấp giao diện để xem danh sách cũng như chi tiết thông tin giảng viên. Những tính năng này giúp trung tâm quản lý đội ngũ nhân sự hiệu quả, đảm bảo dữ liệu được cập nhật chính xác và dễ dàng truy xuất.

Trong quản lý thiết bị, hệ thống cung cấp các chức năng toàn diện để đáp ứng nhu cầu của trung tâm. Thiết bị được quản lý theo phòng học, cho phép xem thông tin về số lượng, loại thiết bị và thêm thiết bị vào từng phòng. Hệ thống hỗ trợ chuyển thiết bị giữa các phòng, lưu trữ lịch sử chuyển đổi để theo dõi. Ngoài ra, hệ thống theo dõi tình trạng thiết bị như mới, hỏng, cần bảo trì hoặc đang cho mượn, đồng thời cung cấp báo cáo thống kê thiết bị theo loại như máy tính, máy chiếu hay bảng trắng. Giảng viên có thể mượn và trả thiết bị, với hệ thống ghi nhận tình trạng thiết bị khi trả lại và lưu trữ lịch sử mượn trả để quản lý chặt chẽ. Tính năng tìm kiếm thiết bị theo tên, loại, tình trạng hoặc phòng học giúp nhân viên dễ dàng định vị thiết bị cần thiết.

Về quản lý quà tặng, hệ thống cho phép quản lý nhập, xuất và theo dõi tình trạng tồn kho quà tặng. Quà tặng có thể được xuất cho học viên xuất sắc, đối tác hoặc khách đến làm

việc, với quy trình duyệt đơn yêu cầu xuất quà thông qua file PDF được nhập bởi giáo viên. Admin nhận thông báo và duyệt đơn, đảm bảo quy trình minh bạch và thuận tiện. Hệ thống cũng hỗ trợ nhập file PDF để nghị xuất quà và xuất file báo cáo xuất quà, giúp quản lý các giao dịch quà tặng một cách hiệu quả. Thông báo tự động khi kho quà thường sắp hết giúp trung tâm kịp thời bổ sung nguồn lực.

Hệ thống cung cấp các báo cáo và thống kê chi tiết để hỗ trợ quản lý và ra quyết định. Các báo cáo bao gồm thống kê số lượng thiết bị và quà thường trong kho, chi tiết các giao dịch nhập xuất, tình trạng mượn trả thiết bị theo thời gian, cũng như báo cáo thiết bị theo loại để hỗ trợ phân bổ hợp lý. Báo cáo về quà thường giúp trung tâm nắm bắt số lượng tồn kho và lịch sử xuất quà, từ đó tối ưu hóa việc nhập xuất. Với các tính năng này, hệ thống không chỉ đáp ứng nhu cầu quản lý thiết bị và quà thường tại Trung tâm Victory mà còn mang lại sự tiện lợi, chính xác và hiệu quả trong vận hành.

3.2. Phân tích yêu cầu và lựa chọn công nghệ

3.2.1. Phân tích yêu cầu

Hệ thống quản lý thiết bị và quà tặng cho Trung tâm Ngoại ngữ - Tin học Victory được thiết kế với ba vai trò người dùng chính: quản trị viên (admin), người quản lý thiết bị, và người quản lý quà tặng. Mỗi vai trò được phân quyền rõ ràng để thực hiện các chức năng phù hợp, đảm bảo quản lý hiệu quả thiết bị dạy học, quà thường, và thông tin liên quan. Các chức năng chính của hệ thống được liệt kê chi tiết trong bảng 3.1, thể hiện phân quyền cho từng vai trò.

Bảng 3.1 Các chức năng trong hệ thống quản lý thiết bị và quà tặng

STT	Chức năng	Người quản lý thiết bị	Người quản lý quà tặng	Quản trị viên
1	Đăng nhập	✓	✓	✓
2	Quản lý tài khoản người dùng (thêm, sửa, xóa, kích hoạt/khóa)			✓
3	Quản lý giảng viên (thêm, sửa, xóa, nhập CSV, xem danh sách)			✓
4	Quản lý thiết bị theo phòng học	✓		✓

5	Chuyển thiết bị giữa các phòng	✓		✓
6	Thống kê tình trạng thiết bị (mới, cho mượn,...)	✓		✓
7	Quản lý mượn/trả thiết bị (ghi nhận tình trạng, lịch sử)	✓		✓
8	Tìm kiếm thiết bị (theo tên, loại, tình trạng, phòng học)	✓		✓
9	Thống kê thiết bị theo loại	✓		✓
10	Quản lý quà tặng (thêm, sửa, xoá)		✓	✓
11	Xuất quà tặng (học viên, đối tác, khách)		✓	✓
12	Nhập/duyệt đơn yêu cầu xuất quà (file PDF)		✓	✓
13	Hoàn trả quà tặng		✓	✓
14	Báo cáo kho thiết bị (nhập, xuất,...)	✓		✓
15	Báo cáo mượn/trả thiết bị (theo thời gian, tình trạng)	✓		✓
16	Báo cáo thống kê thiết bị theo loại	✓		✓
18	Báo cáo quà thường (nhập, xuất,...)		✓	✓
19	Thông báo	✓	✓	✓
20	Gửi mail	✓	✓	✓

3.2.2. Lựa chọn công nghệ

Hệ thống được xây dựng với các thành phần chính sau:

- **Front-end:** Giao diện người dùng hiển thị thông tin thiết bị, quà tặng, báo cáo, và hỗ trợ các thao tác quản lý của quản trị viên, người quản lý thiết bị, và người quản lý quà tặng.
- **Back-end:** Cung cấp API để xử lý logic nghiệp vụ, quản lý dữ liệu và giao tiếp giữa front-end và cơ sở dữ liệu.

- Database:** Lưu trữ thông tin về thiết bị, quà tặng, tài khoản người dùng, giảng viên, phòng học, lịch sử mượn/trả, và các giao dịch nhập/xuất.
- Thông báo:** Hệ thống thông báo tự động khi kho thiết bị hoặc quà tặng sắp cạn, đảm bảo quản lý kịp thời.

Các công nghệ được lựa chọn dựa trên các tiêu chí: Mã nguồn mở hoặc miễn phí để tối ưu chi phí. Dễ phát triển, triển khai và bảo trì. Hỗ trợ nhiều thư viện và công cụ phát triển. Cộng đồng người dùng lớn, tài liệu phong phú. Khả năng mở rộng và tích hợp tốt. Phù hợp với quy mô nhỏ và vừa của trung tâm. Dựa trên các tiêu chí này, các công nghệ được lựa chọn được trình bày trong bảng 3.2.

Bảng 3.2 Các công nghệ được lựa chọn sử dụng

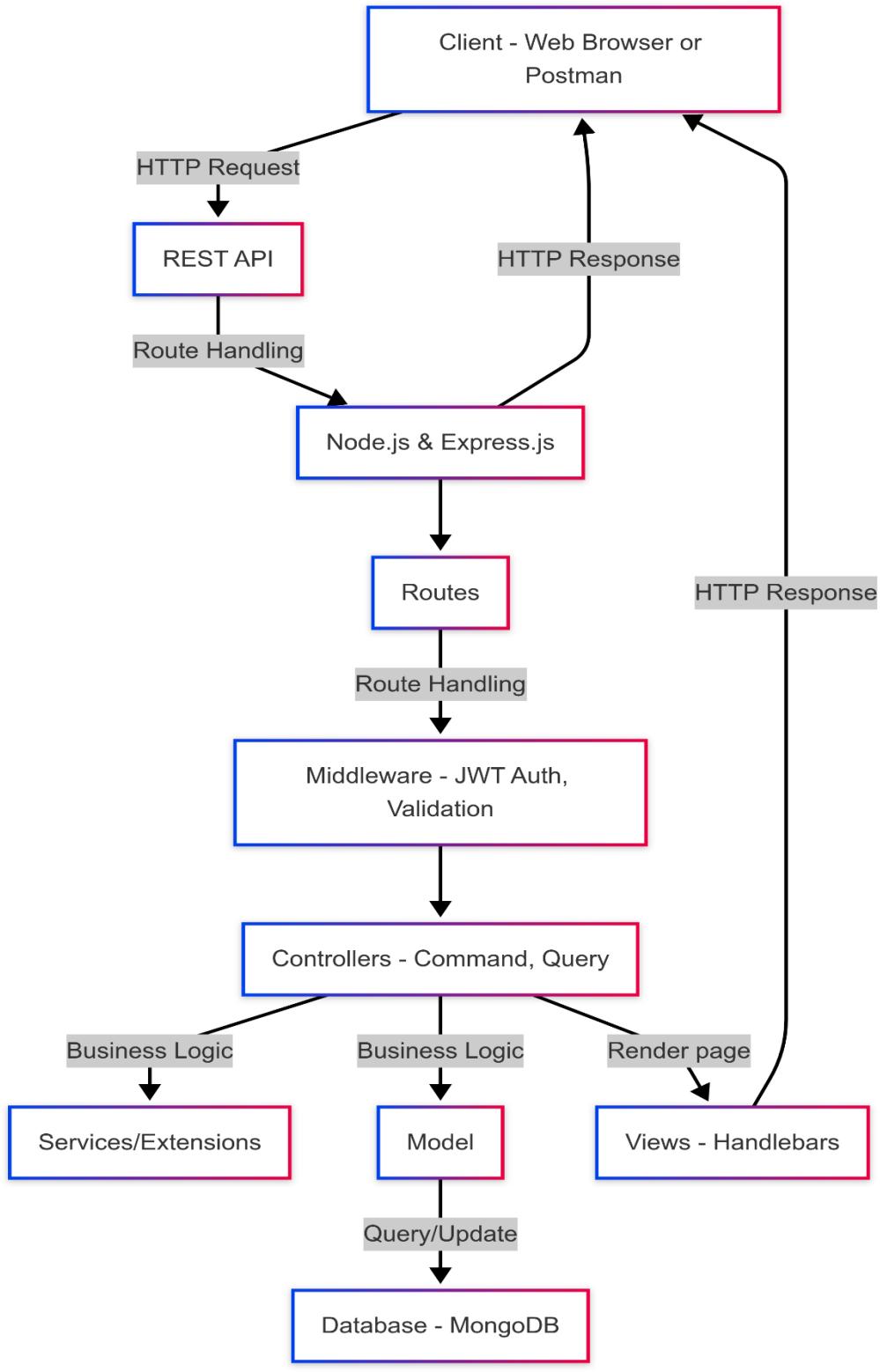
STT	Module	Công nghệ/Giải pháp	Diễn giải
1	Front-end	Handlebars	Công cụ template engine nhẹ, dễ sử dụng, phù hợp cho giao diện động và tích hợp với Node.js.
		Bootstrap 5	Framework CSS phổ biến, hỗ trợ giao diện responsive, thân thiện với người dùng trên nhiều thiết bị.
2	Back-end	Node.js	Môi trường runtime JavaScript phía server, hiệu năng cao, hỗ trợ nhiều thư viện.
		Express	Framework JavaScript cho Node.js, giúp phát triển API nhanh chóng và dễ dàng.
3	Database	MongoDB	Cơ sở dữ liệu NoSQL mã nguồn mở, linh hoạt trong lưu trữ dữ liệu phi cấu trúc, dễ mở rộng.
4	Kiến trúc	MVC + CQRS	Mô hình MVC tổ chức code rõ ràng, kết hợp CQRS tách biệt thao tác đọc/ghi để tăng hiệu suất và bảo trì.
5	Thông báo	Node.js (tích hợp)	Tích hợp các thư viện như nodemailer hoặc push notification để gửi thông báo khi kho sắp cạn.

Các công nghệ này đảm bảo hệ thống đáp ứng đầy đủ các yêu cầu chức năng, đồng thời dễ dàng triển khai, bảo trì và mở rộng trong tương lai, phù hợp với nhu cầu quản lý thiết

bị và quà tặng tại Trung tâm Ngoại ngữ - Tin học Victory.

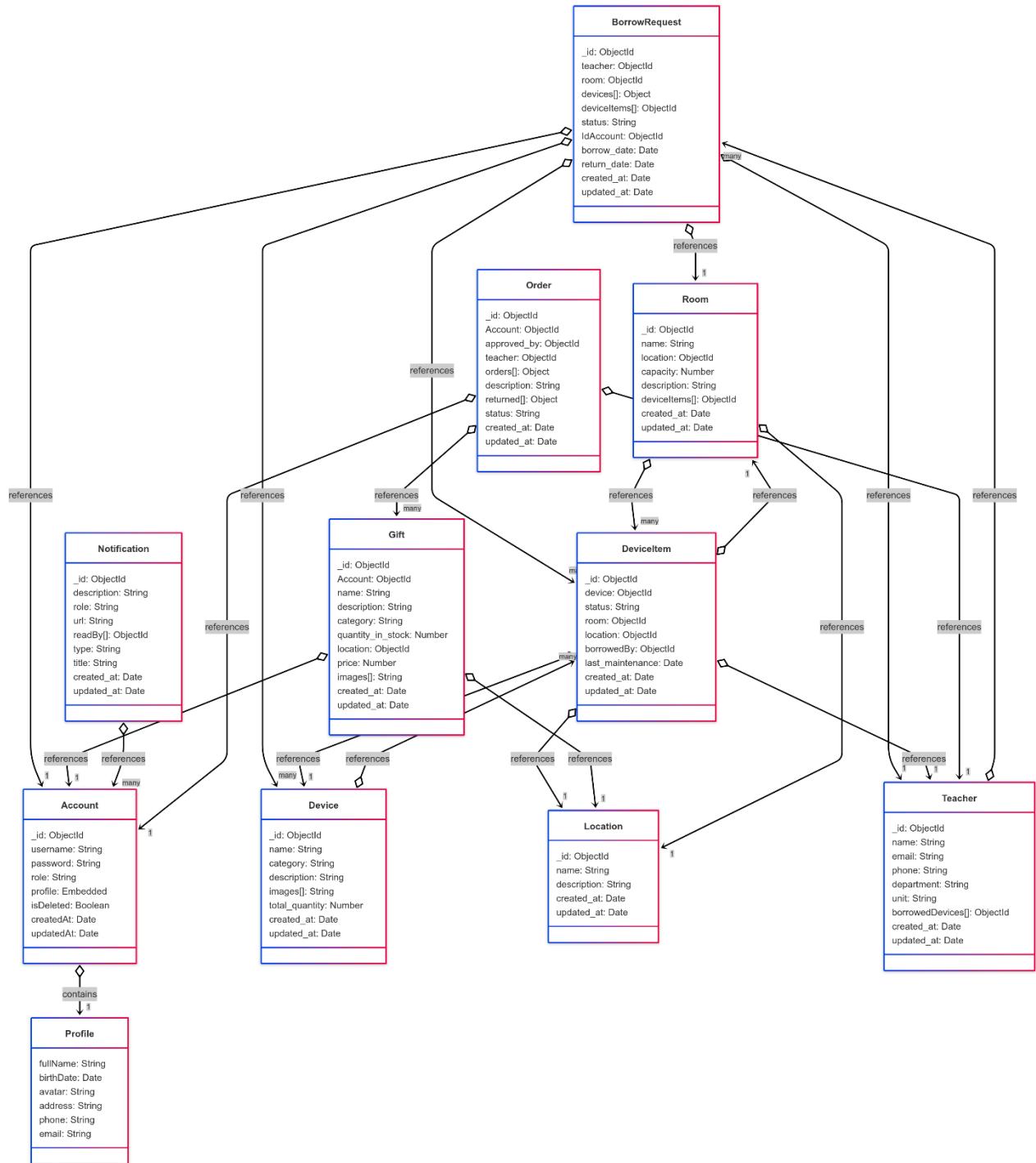
3.3. Thiết kế kiến trúc hệ thống

3.3.1. Thiết kế kiến trúc hệ thống



Hình 3.1 Kiến trúc hệ thống

3.3.2. Thiết kế cơ sở dữ liệu



Hình 3.2. Sơ đồ quan hệ thực thể

Chi tiết mỗi bảng sẽ có các thuộc tính, kiểu dữ liệu, ràng buộc được nêu rõ trong các bảng bên dưới (*từ Bảng 3.3 đến Bảng 3.13*).

*Mô tả bảng Account (Tài khoản người dùng)

Bảng 3.3 Bảng dữ liệu tài khoản người dùng

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID duy nhất của tài khoản
username	String	Required, Unique, Trim	Tên đăng nhập, không trùng lặp
password	String	Required	Mật khẩu của tài khoản
role	String	Required, Enum	Vai trò (system_admin, device manager, gift manager, user)
profile	Embedded	-	Thông tin cá nhân nhúng (xem bảng Profile)
isDeleted	Boolean	Default: false	Trạng thái xóa mềm
createdAt	Date	Tự động tạo (timestamps)	Thời gian tạo tài khoản
updatedAt	Date	Tự động cập nhật (timestamps)	Thời gian cập nhật cuối

*Mô tả bảng Profile (Thông tin người dùng)

Bảng 3.4 Bảng dữ liệu thông tin người dùng

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
fullName	String	Required, Trim	Họ và tên đầy đủ
birthDate	Date	-	Ngày sinh
avatar	String	Default: null	Đường dẫn ảnh đại diện
address	String	-	Địa chỉ
phone	String	Match: /^[^d]{10,15}\\$/	Số điện thoại (10-15 chữ số)
email	String	Match: email regex	Email hợp lệ

*Mô tả bảng BorrowRequest (Yêu cầu mượn trả thiết bị)

Bảng 3.5 Bảng dữ liệu yêu cầu mượn trả thiết bị

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID yêu cầu mượn thiết bị
teacher	ObjectId	Required, Ref: Teacher	Giáo viên yêu cầu
room	ObjectId	Ref: Room, Optional	Phòng nhận thiết bị (null nếu giữ thiết bị ngoài phòng)
devices	Array[Object]	Required	Danh sách { device: ObjectId, quantity: Number }
deviceItems	Array[ObjectId]	Ref: DeviceItem	Danh sách thiết bị cụ thể được cấp phát
status	String	Enum: [Đang mượn, Đã trả, Quá hạn], Default: "Đang mượn"	Trạng thái mượn thiết bị

<code>idAccount</code>	<code>ObjectId</code>	Required, Ref: <code>Account</code>	Người thực hiện yêu cầu
<code>borrow_date</code>	<code>Date</code>	Default: <code>Date.now</code>	Ngày mượn thiết bị
<code>return_date</code>	<code>Date</code>	Optional	Ngày trả thiết bị
<code>created_at</code>	<code>Date</code>	Default: <code>Date.now</code>	Ngày tạo yêu cầu
<code>updated_at</code>	<code>Date</code>	Tự động cập nhật	Ngày cập nhật cuối

*Mô tả bảng Device (Thiết bị)

Bảng 3.6 Bảng dữ liệu thiết bị

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
<code>id</code>	<code>ObjectId</code>	Tự động tạo	ID thiết bị
<code>name</code>	<code>String</code>	Required, Trim	Tên thiết bị
<code>category</code>	<code>String</code>	Required, Enum	Loại thiết bị (Máy chiếu, Laptop,...)
<code>description</code>	<code>String</code>	Default: "", Trim	Mô tả thiết bị
<code>images</code>	<code>Array[String]</code>	Default: []	Danh sách ảnh thiết bị
<code>total_quantity</code>	<code>Number</code>	Required, Min: 0, Default: 0	Tổng số thiết bị hiện có
<code>created_at</code>	<code>Date</code>	Default: <code>Date.now</code>	Thời gian tạo
<code>updated_at</code>	<code>Date</code>	Tự động cập nhật	Cập nhật cuối

*Mô tả bảng DeviceItem (Mục thiết bị)

Bảng 3.7 Bảng dữ liệu mục thiết bị

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
<code>id</code>	<code>ObjectId</code>	Tự động tạo	ID thiết bị cụ thể
<code>device</code>	<code>ObjectId</code>	Required, Ref: <code>Device</code>	Thuộc thiết bị nào
<code>status</code>	<code>String</code>	Required, Enum, Default: "Mới"	Trạng thái thiết bị (Mới, Hoạt động,...)
<code>room</code>	<code>ObjectId</code>	Ref: Room, Optional	Phòng đang đặt
<code>location</code>	<code>ObjectId</code>	Ref: Location, Optional	Vị trí đặt thiết bị
<code>borrowedBy</code>	<code>ObjectId</code>	Ref: Teacher, Optional	Giáo viên đang mượn
<code>last_maintenance</code>	<code>Date</code>	Default: null	Ngày bảo trì cuối
<code>created_at</code>	<code>Date</code>	Default: <code>Date.now</code>	Thời gian tạo
<code>updated_at</code>	<code>Date</code>	Tự động cập nhật	Cập nhật cuối

*Mô tả bảng Gift (Quà tặng)

Bảng 3.8 Bảng dữ liệu quà tặng

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID quà tặng
Account	ObjectId	Required, Ref: Account	Người quản lý quà
name	String	Required, Trim	Tên quà tặng
description	String	Default: "", Trim	Mô tả quà tặng
category	String	Required, Enum	Loại quà (Học viên, Đối tác, ...)
quantity in stock	Number	Default: 0, Min: 0	Số lượng tồn kho
location	ObjectId	Required, Ref: Location	Vị trí lưu quà
price	Number	Default: 0	Giá trị quà tặng
images	Array[String]	Default: []	Hình ảnh quà tặng
created at	Date	Default: Date.now	Tạo lúc
updated at	Date	Tự động cập nhật	Cập nhật cuối

*Mô tả bảng Location (Vị trí)

Bảng 3.9 Bảng dữ liệu vị trí lưu trữ

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID vị trí
name	String	Required, Unique, Trim	Tên vị trí
description	String	Default: "", Trim	Mô tả
created at	Date	Default: Date.now	Thời gian tạo
updated at	Date	Tự động cập nhật	Thời gian cập nhật cuối

*Mô tả bảng Notification (Thông báo)

Bảng 3.10 Bảng dữ liệu thông báo

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID thông báo
description	String	Default: "", Trim	Nội dung thông báo
role	String	Required, Enum	Vai trò nhận thông báo
url	String	Default: "", Trim	Liên kết nếu có
readBy	Array[Object Id]	Ref: Account	Những người đã đọc
type	String	Enum, Default: "info"	Loại thông báo (info, warning,...)
title	String	Default: "", Trim	Tiêu đề thông báo
created_at	Date	Tự động tạo (timestamps)	Thời gian tạo
updated_at	Date	Tự động cập nhật	Cập nhật cuối

*Mô tả bảng Order (Đơn yêu cầu quà tặng)

Bảng 3.11 Bảng dữ liệu đơn yêu cầu quà tặng

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID đơn hàng
Account	ObjectId	Required, Ref: Account	Người đặt hàng
approved_by	ObjectId	Ref: Account, Default: null	Người phê duyệt
teacher	ObjectId	Required, Ref: Teacher	Giáo viên nhận quà
orders	Array[ObjectId,quantity]	Required, Ref: Gift	Quà tặng, số lượng
description	String	Default: ""	Mục đích sử dụng
returned	Array[ObjectId,quantity, returned at]	Not required	Thông tin hoàn trả
status	String	Enum, Default: "Chưa duyệt"	Trạng thái duyệt
created_at	Date	Default: Date.now	Tạo lúc
updated_at	Date	Tự động cập nhật	Cập nhật cuối

*Mô tả bảng Room (Phòng)

Bảng 3.12 Bảng dữ liệu thông tin phòng

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	ObjectId	Tự động tạo	ID phòng
name	String	Required, Unique, Trim	Tên phòng
location	ObjectId	Ref: Location, Optional	Vị trí đặt phòng
capacity	Number	Min: 1, Optional	Sức chứa
description	String	Default: "", Trim	Mô tả phòng
deviceItems	Array[ObjectId]	Ref: DeviceItem	Thiết bị chứa trong phòng
created_at	Date	Default: Date.now	Tạo lúc
updated_at	Date	Tự động cập nhật	Cập nhật cuối

*Mô tả bảng Teacher (Giảng viên)

Bảng 3.13 Bảng dữ liệu giảng viên

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	Tự động tạo	ID duy nhất của giáo viên
name	String	Required, Trim	Họ và tên giáo viên
email	String	Required, Unique, Match: email regex	Email giáo viên hợp lệ
phone	String	Match: /^\d{10,15}\$/, Optional	Số điện thoại (10–15 chữ số)
department	String	Optional	Khoa / Bộ môn giáo viên thuộc về
unit	String	Optional	Đơn vị công tác

borrowedDevices	Array[ObjectId]	Ref: "BorrowRequest"	Đơn mượn
created_at	Date	Default: Date.now	Thời gian tạo hồ sơ giáo viên
updated_at	Date	Tự động cập nhật	Thời gian cập nhật cuối cùng

3.4. Xây dựng API phía máy chủ (Back-end)

Xây dựng tổng cộng 69 API chia làm 8 nhóm chính:

Nhóm API Người dùng: Nhóm API này chịu trách nhiệm quản lý người dùng, bao gồm đăng nhập, đăng xuất, tạo mới, cập nhật, thay đổi mật khẩu, vô hiệu hóa, xóa và khôi phục tài khoản người dùng. Hệ thống hỗ trợ cả upload avatar và mã hóa mật khẩu.

Nhóm API Thiết bị: Nhóm API này quản lý toàn bộ thông tin về thiết bị, bao gồm thêm mới, cập nhật, xóa, lấy danh sách, xem chi tiết và import dữ liệu thiết bị từ file.

Nhóm API Phòng học & Thiết bị trong phòng: Nhóm API này chịu trách nhiệm quản lý phòng học và việc gán thiết bị vào từng phòng, bao gồm: thêm thiết bị, xóa thiết bị, chuyển phòng, cập nhật thông tin phòng, import danh sách phòng từ file CSV, và xem chi tiết các thiết bị có trong từng phòng.

Nhóm API Giảng viên: Nhóm API này hỗ trợ quản lý thông tin giảng viên, bao gồm: thêm mới, cập nhật, xóa, import từ file CSV, và truy vấn danh sách giảng viên với khả năng phân trang và lọc theo bộ môn.

Nhóm API quản lý quà tặng: Nhóm API này hỗ trợ quản lý các hoạt động liên quan đến quà tặng: thêm mới, cập nhật, xóa, xem danh sách quà, import quà từ file, xử lý đơn nhập quà tặng, duyệt đơn nhập và hoàn trả. Quà tặng có thể bao gồm hình ảnh minh họa.

Nhóm API Mượn – Trả Thiết Bị: Nhóm API này hỗ trợ quy trình mượn và trả thiết bị, bao gồm: lấy danh sách thiết bị sẵn sàng cho mượn, tạo đơn mượn, trả thiết bị, xem danh sách và chi tiết các đơn mượn. Hệ thống đảm bảo kiểm soát tình trạng thiết bị và trạng thái của đơn mượn.

Nhóm API Thống kê: Nhóm API này hỗ trợ truy xuất dữ liệu thống kê phục vụ quản trị hệ thống. Bao gồm các thống kê liên quan đến mượn – trả thiết bị, tình trạng thiết bị, quà tặng và người dùng có hoạt động nổi bật.

Nhóm API Thông báo: Nhóm API này hỗ trợ tạo, đánh dấu đọc và hiển thị thông báo.

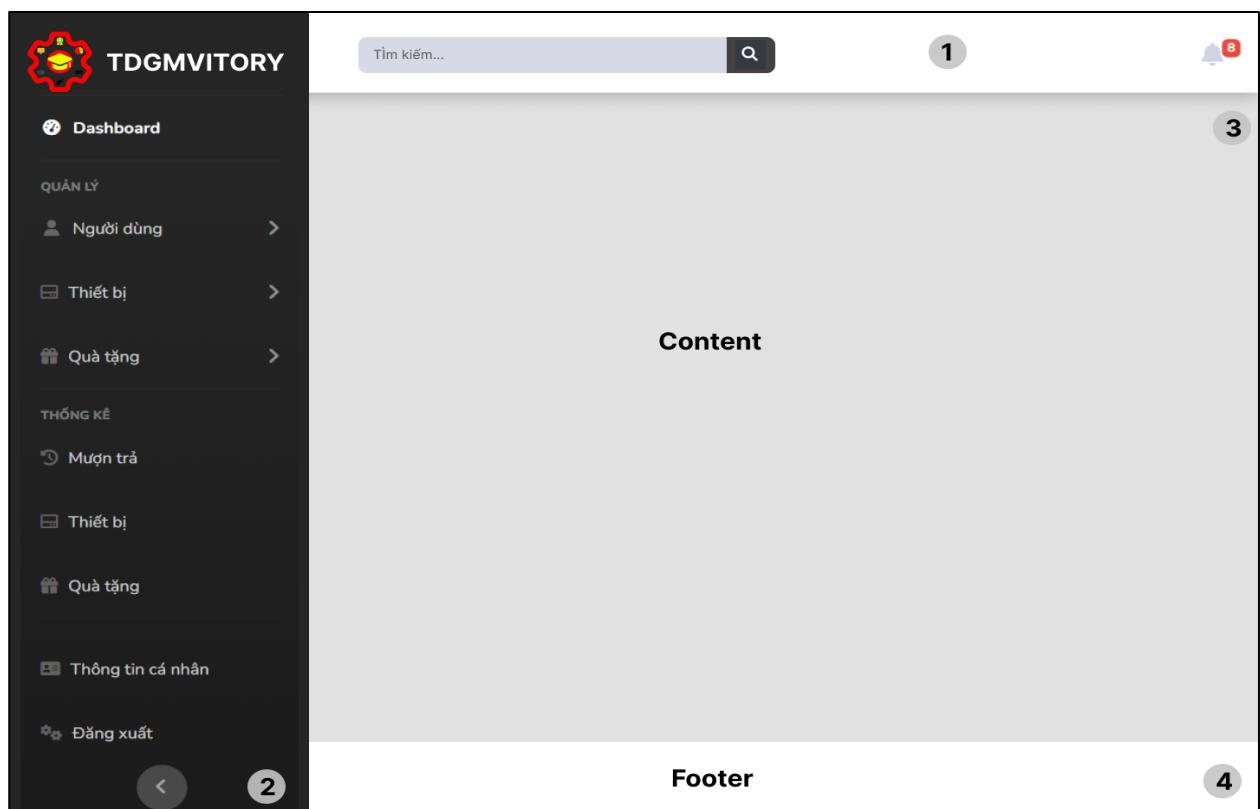
Tài liệu API: <https://tdgm-victory.onrender.com/api-docs/>.

3.5. Thiết kế giao diện (Front-end)

3.5.1. Thiết kế khung giao diện làm việc chính

Khung giao diện chính được thiết kế với bốn thành phần chính:

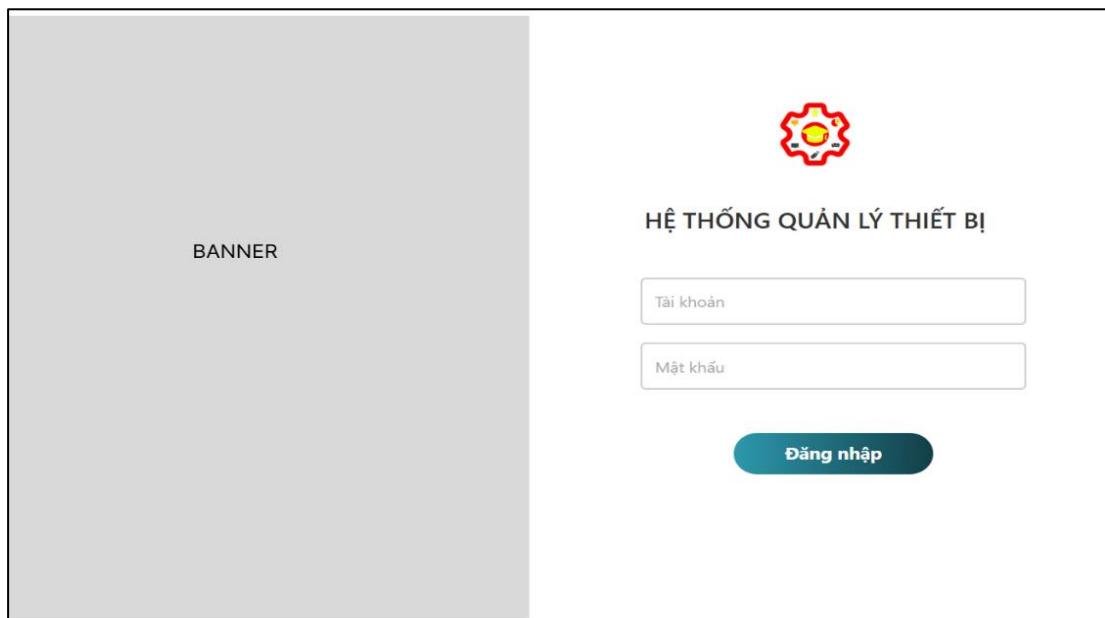
- *Header (1)*: Cho phép người dùng dễ dàng truy cập các tính năng toàn cục như xem thông báo, tìm kiếm.
- *Navbar (2)*: Chứa danh sách các liên kết điều hướng (menu) đến các trang quan trọng như: Trang chủ, Quản lý người dùng, Quản lý thiết bị, Quản lý quà tặng, Thống kê, ... Giúp người dùng dễ dàng truy cập các phần khác nhau của nền tảng.
- *Content (3)*: Vùng hiển thị nội dung chính của trang web, Hiển thị thông tin chính theo từng trang mà người dùng đã chọn từ Navbar, ví dụ: Danh sách thiết bị, Danh sách đơn yêu cầu quà tặng, Danh sách quà tặng, Danh sách mượn trả thiết bị,...
- *Footer (4)*: Phần chân trang, có thể chứa thông tin liên hệ, bản quyền.



Hình 3.3 Bản phác thảo khung giao diện chính

3.5.2. Thiết kế khung giao diện đăng nhập

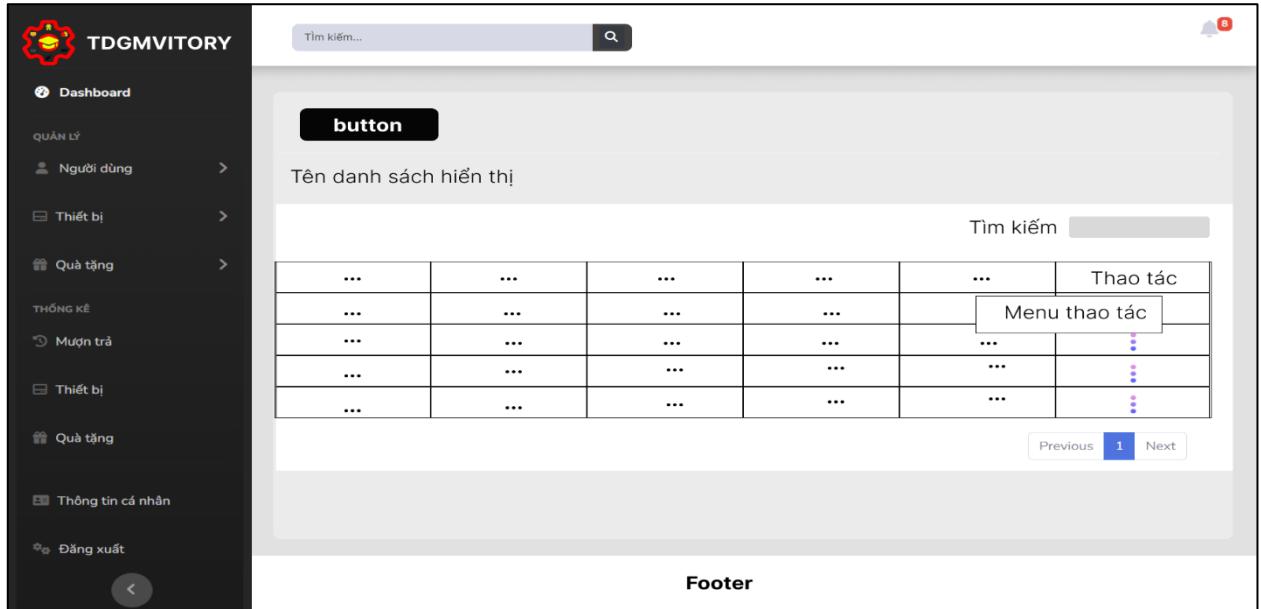
Để sử dụng các chức năng của hệ thống người dùng cần phải thực hiện đăng nhập. Giao diện đăng nhập được chia làm 2 phần, phía bên trái là banner, phía bên phải là form đăng nhập. Form đăng nhập bao gồm tên hệ thống, label, ô input nhập tài khoản, ô input nhập mật khẩu và nút đăng nhập.



Hình 3.4 Bản phát thảo giao diện đăng nhập

3.5.3. Thiết kế giao diện danh sách hiển thị

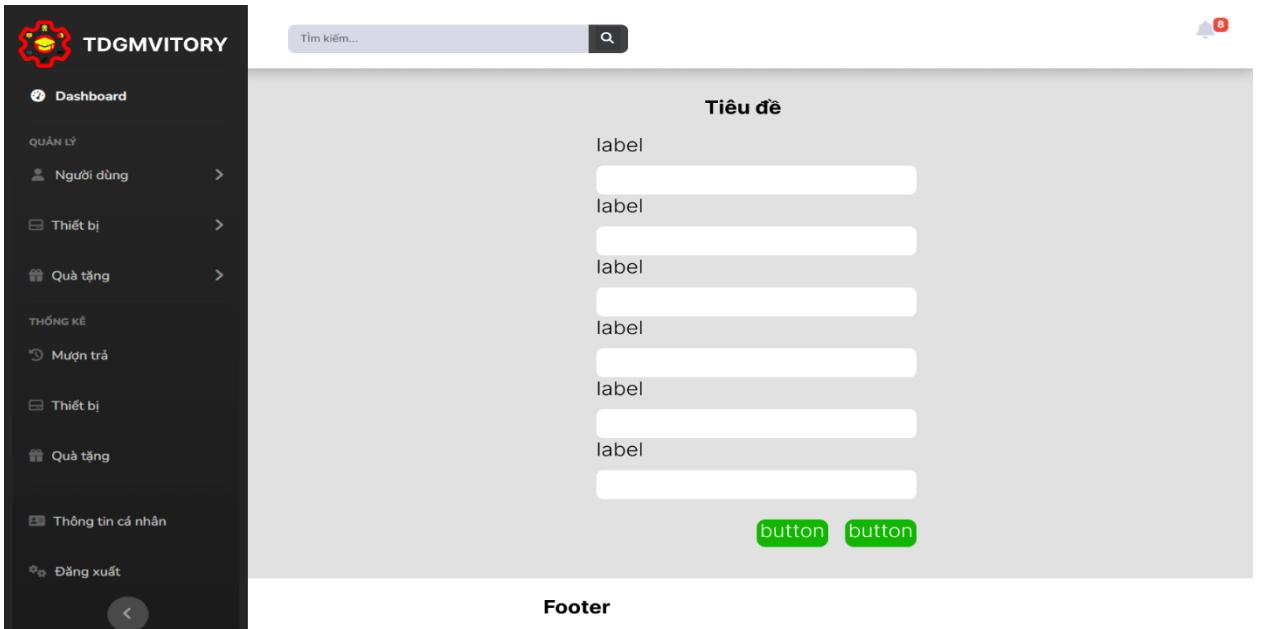
Giao diện danh sách hiển thị giống với giao diện điện tử có header, navbar, footer nhưng khác ở phần content. Danh sách hiển thị dưới dạng bảng bao gồm: Tên danh sách đang hiển thị, ô input tìm kiếm, các cột tiêu đề, các dòng dữ liệu. Dưới bảng có nút phân trang và nút quay lại. Cột cuối cùng trong bảng chứa các nút thao tác như chỉnh sửa, xoá,.. Tuỳ theo một số giao diện sẽ có nút thao tác nằm trên bảng để thực hiện các chức năng như thêm mới, tải mẫu file (mẫu file csv để nhập lên hệ thống),...



Hình 3.5 Bản phát thảo giao diện danh sách hiển thị

3.5.4. Thiết kế giao diện thêm mới và chỉnh sửa

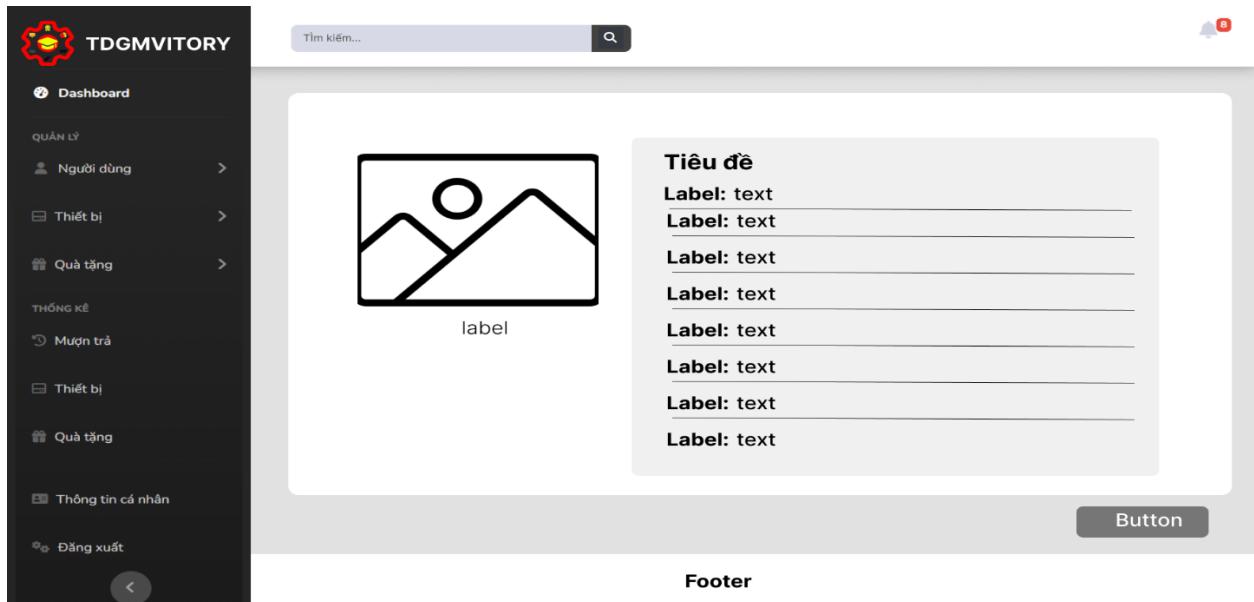
Giao diện danh sách hiển thị giống với giao diện điện tử có header, navbar, footer nhưng khác ở phần content. Giao diện thêm mới, chỉnh sửa hiển thị các thành phần như tiêu đề, ô input, label, button lưu và button quay lại,...



Hình 3.6 Bản phát thảo giao diện thêm mới và chỉnh sửa

3.5.5. Thiết kế giao diện xem thông tin chi tiết

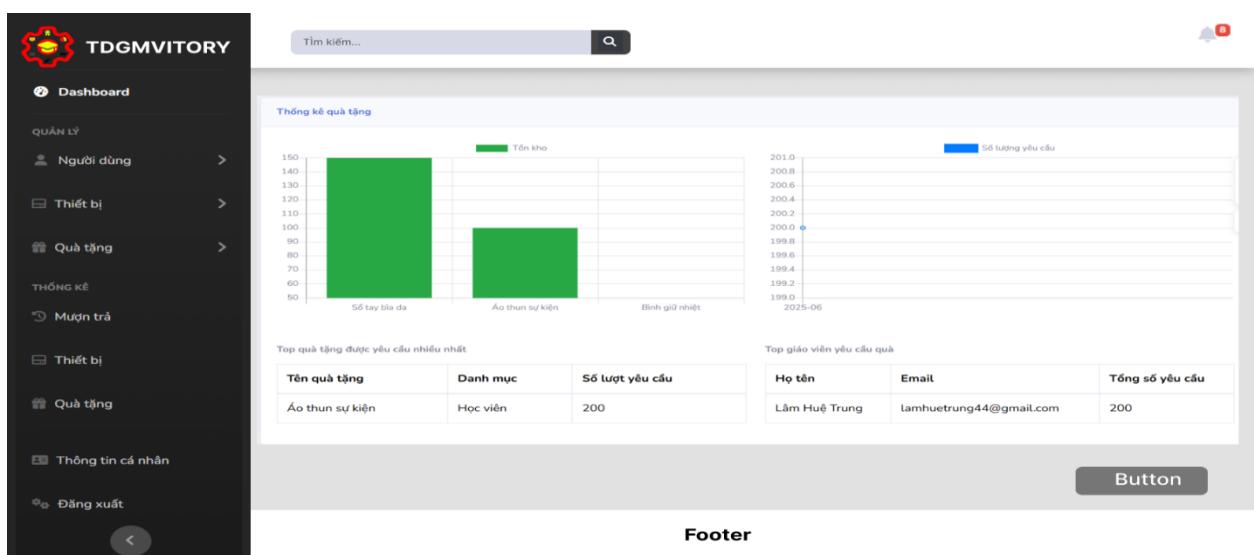
Giao diện danh sách hiển thị giống với giao diện điện tử gồm có header, navbar, footer nhưng khác ở phần content. Phần content được chia làm 2 phần: Phía bên trái hiển thị hình ảnh và tên của thông tin đang xem. Phía bên trái hiển thị thông tin chi tiết bao gồm: Tiêu đề, label, text,... Dưới cùng có nút quay lại.



Hình 3.7 Thiết kế giao diện xem thông tin chi tiết

3.5.6. Thiết kế giao diện thống kê

Giao diện danh sách hiển thị giống với giao diện điện tử gồm có header, navbar, footer nhưng khác ở phần content. Phần content bao gồm: tiêu đề, biểu đồ, bảng biểu,...



Hình 3.8 Bảng phát thảo giao diện thống kê

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Bộ dữ liệu thử nghiệm

4.1.1. Bộ dữ liệu thử nghiệm chức năng quản lý người dùng

Bảng 4.1 Bộ dữ liệu người dùng thử nghiệm

Vai trò	Thông tin chi tiết
Quản trị viên	Tài khoản: LamHueTrung Mật khẩu: Lht080103* Thông tin chung: <ul style="list-style-type: none">- Họ tên: Lâm Huệ Trung- Ngày sinh: 8/1/2003- Số điện thoại: 0763849007- Email: lamhuetrung@gmail.com- Địa chỉ: Trà Vinh
Quản lý thiết bị	Tài khoản: NhanMinhPhuc Mật khẩu: NhanMinhPhuc* Thông tin chung: <ul style="list-style-type: none">- Họ tên: Nhan Minh Phúc- Ngày sinh: 1/1/1990- Số điện thoại: 0714523698- Email: nhanminhphuc@tvu.edu.vn- Địa chỉ: Trà Vinh
Quản lý quà tặng	Tài khoản: NguyenHoangDuyThien Mật khẩu: NguyenHoangDuyThien* Thông tin chung: <ul style="list-style-type: none">- Họ tên: Nguyễn Hoàng Duy Thiện- Ngày sinh: 1/1/1990- Số điện thoại: 0714523697- Email: thiennhd@tvu.edu.vn- Địa chỉ: Trà Vinh

Bảng 4.2 Bộ dữ liệu giảng viên thử nghiệm

Họ tên	Email	Số điện thoại	Chuyên ngành	Đơn vị
Lâm Nam Phong	namphongtctv@gmail.com	0965123789	Công nghệ thông tin	Phòng HCTH
Trần Thị Minh Thư	thu.tran@tvu.edu.vn	0912345678	Quản trị kinh doanh	Phòng Kế hoạch
Lê Văn Hùng	hung.le@tvu.edu.vn	0978456123	Kỹ thuật điện tử	Phòng Kỹ thuật
Phạm Quốc Bảo	bao.pham@tvu.edu.vn	0909123456	Hành chính	Phòng HCTH
Nguyễn Thị Kim Oanh	oanh.nguyen@tvu.edu.vn	0987766554	Marketing	Phòng Truyền thông
Võ Minh Tâm	tam.vo@tvu.edu.vn	0933445566	Công nghệ thông tin	Phòng IT
Đỗ Thị Hồng Hạnh	hanh.do@tvu.edu.vn	0911223344	Kế toán	Phòng Kế toán
Hồ Văn Duy	duy.ho@tvu.edu.vn	0922334455	Thiết kế đồ họa	Phòng Thiết kế
Trịnh Quốc Việt	viet.trinh@tvu.edu.vn	0966887788	Quản lý dự án	Phòng Dự án
Lý Ngọc Lan	lan.ly@tvu.edu.vn	0944556677	Nhân sự	Phòng Nhân sự

4.1.2. Bộ dữ liệu thử nghiệm chức năng quản lý thiết bị

Bảng 4.3 Bộ dữ liệu thiết bị thử nghiệm

Tên thiết bị	Loại thiết bị	Số lượng	Mô tả thiết bị	Link hình ảnh trên máy
Remote máy lạnh Daikin	Khác	10	Điều khiển máy lạnh treo tường	C:\Users\HP\Desktop\Victory\Devices\remote_maylanh.jpg
Remote máy chiếu Epson	Khác	10	Điều khiển từ xa máy chiếu	C:\Users\HP\Desktop\Victory\Devices\remote_projector.jpg
Ô cảm điện đa năng	Khác	4	Cáp nguồn cho các thiết bị điện trong lớp	C:\Users\HP\Desktop\Victory\Devices\ocam_danang.jpg
Loa SoundMax A500	Loa	20	Loa treo tường phục vụ giảng dạy	C:\Users\HP\Desktop\Victory\Devices\loa_soundmax.jpg

Máy chiếu Epson X500	Máy chiếu	10	Trình chiếu bài giảng gắn trên trần	C:\Users\HP\Desktop\Victory\Devices\maychieuepson.jpg
Laptop Dell Latitude	Máy tính	10	Laptop giảng viên để trình bày bài giảng	C:\Users\HP\Desktop\Victory\Devices\laptopdell.jpg
Bảng trắng 1.5m	Bảng trắng	10	Bảng kính treo tường dùng bút lông	C:\Users\HP\Desktop\Victory\Devices\bangtrang.jpg
Bộ phát Wifi TP-Link	Thiết bị mạng	10	Router wifi gắn trần cho lớp học	C:\Users\HP\Desktop\Victory\Devices\wifi_tplink.jpg
Micro không dây	Khác	10	Micro đeo tai dành cho giảng viên	C:\Users\HP\Desktop\Victory\Devices\micro.jpg
Webcam Full HD	Khác	10	Webcam phục vụ học trực tuyến	C:\Users\HP\Desktop\Victory\Devices\webcam.jpg

Bảng 4.4 Bộ dữ liệu phòng học thử nghiệm

Tên phòng	Vị trí tòa nhà	Sức chứa
Phòng A42.101	Khu 2 TVU	100
Phòng A42.102	Khu 2 TVU	90
Phòng A42.103	Khu 2 TVU	60
Phòng A42.104	Khu 2 TVU	40
Phòng A42.201	Khu 2 TVU	50
Phòng A42.202	Khu 2 TVU	70
Phòng A42.203	Khu 2 TVU	30
Phòng A42.204	Khu 2 TVU	35
Phòng A42.301	Khu 2 TVU	80
Phòng A42.302	Khu 2 TVU	120

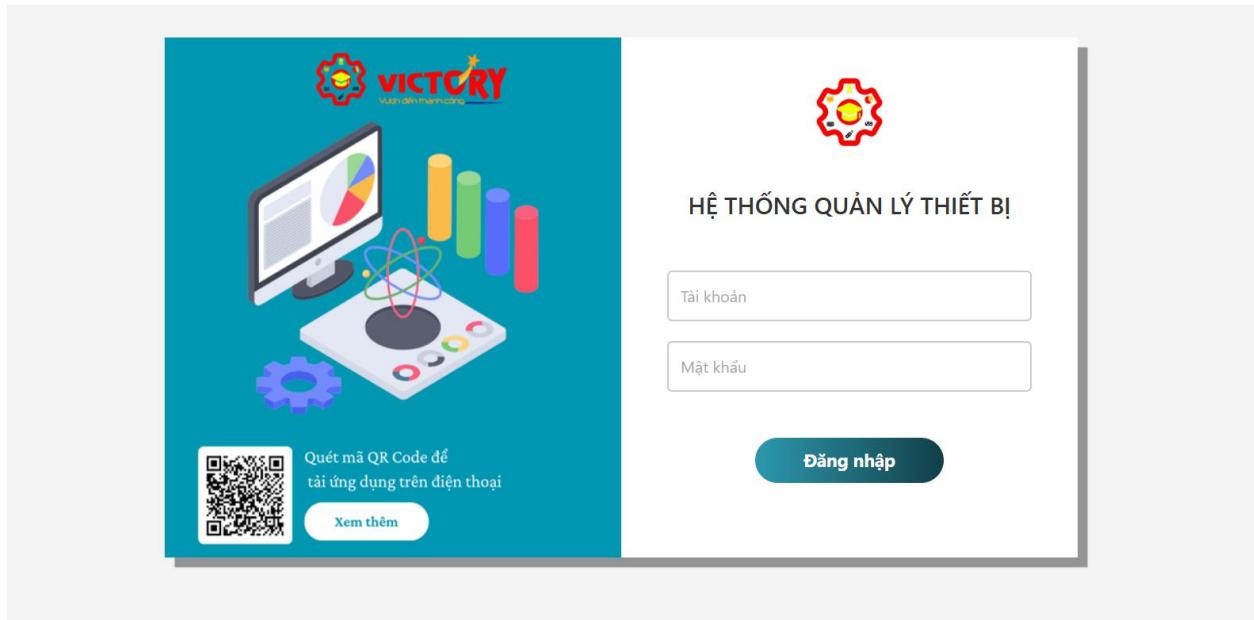
4.1.3. Bộ dữ liệu thử nghiệm chức năng quản lý quà tặng

Tên quà tặng	Loại quà tặng	Số lượng	Giá (VNĐ)	Mô tả	Link hình ảnh trên máy
Áo thun sự kiện	Học viên	100	120000	Áo cotton 100% dùng trong các sự kiện nội bộ	C:\Users\HP\Desktop\Victory\Gifts\aothun_sukien.jpg
Bình giữ nhiệt In Logo	Học viên	80	150000	Bình giữ nhiệt 500ml, khắc logo trung tâm	C:\Users\HP\Desktop\Victory\Gifts\binhgiunhiet.jpg
Sổ tay da cao cấp	Đối tác	50	200000	Sổ tay bìa da sang trọng, in dập chìm logo Victory	C:\Users\HP\Desktop\Victory\Gifts\sotay_da.jpg
Túi canvas đeo chéo	Học viên	70	100000	Túi vải canvas thời trang dùng tặng học viên xuất sắc	C:\Users\HP\Desktop\Victory\Gifts\tuicanvas.jpg
Kỷ niệm chương pha lê	Đối tác	30	300000	Biểu trưng pha lê tặng đối tác hoặc dịp kỷ niệm đặc biệt	C:\Users\HP\Desktop\Victory\Gifts\phanle_kyniem.jpg

4.2. Kết quả thực nghiệm với vai trò quản lý thiết bị

4.2.1. Chức năng đăng nhập

Để bắt đầu sử dụng các tính năng trên hệ thống, người dùng bắt buộc đăng nhập. Ở bước này người dùng phải nhận tài khoản và mật khẩu đã được hệ thống gửi về Gmail.

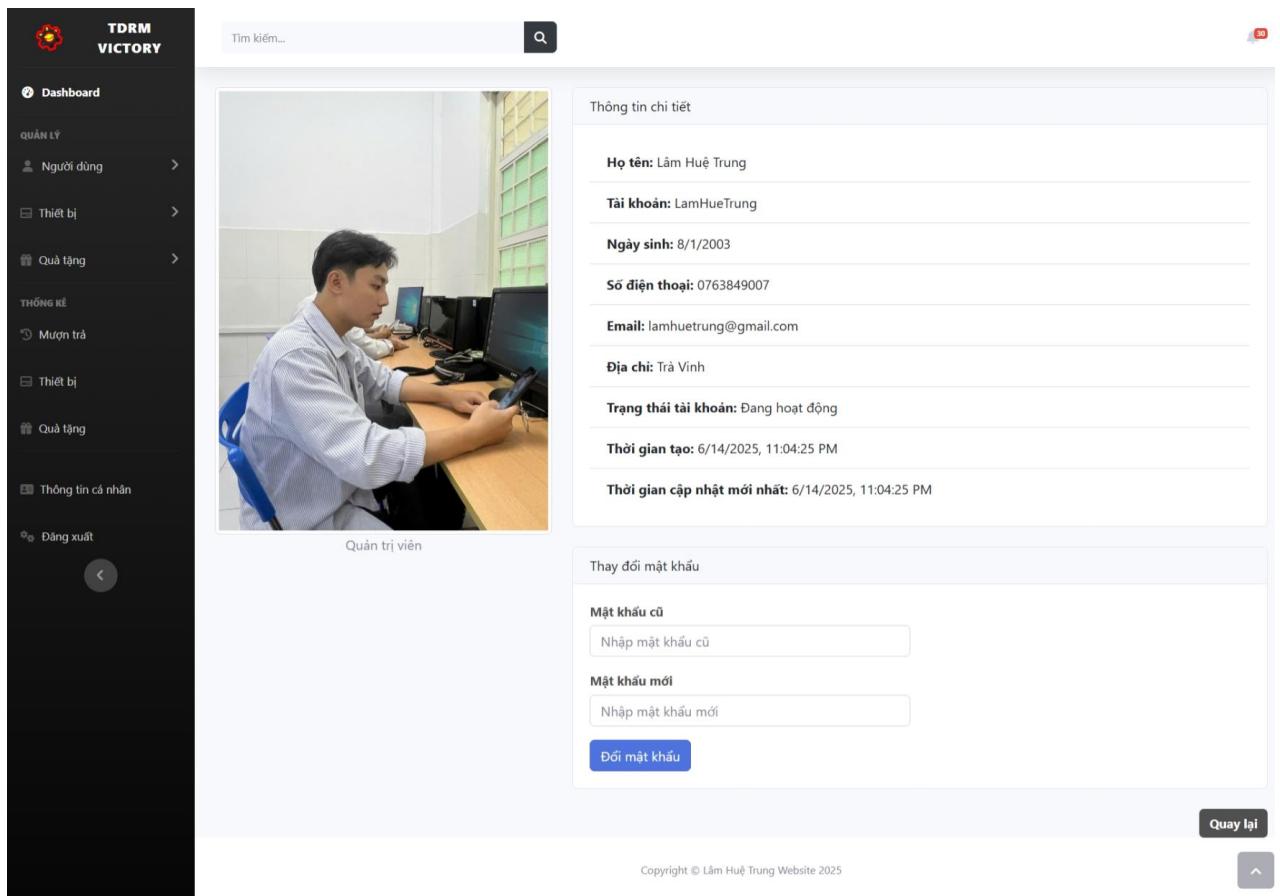


Hình 4.1 Giao diện thực tế chức năng đăng nhập

Người dùng cần điền thông tin tài khoản và mật khẩu vào ô input và nhấn vào nút đăng nhập. Hệ thống sẽ kiểm tra giá trị mà người dùng nhập vào nếu sai sẽ báo lỗi và yêu cầu nhập lại. Nếu thông tin chính xác, hệ thống sẽ hiển thị thông báo thành công và chuyển hướng người dùng đến giao diện chính, hiển thị các quyền có thể sử dụng. Đồng thời, sẽ tạo ra token với giới hạn là 5 giờ, nếu quá thời hạn sẽ quay lại giao diện đăng nhập.

4.2.2. Chức năng xem thông tin và thay đổi mật khẩu

Sau khi đăng nhập vào hệ thống thành công, người dùng có thể thay đổi mật khẩu để bảo mật tài khoản. Người dùng nhấn vào chức năng thông tin chi tiết trong thanh navbar, sau đó kéo xuống phần thay đổi mật khẩu, nhận mật khẩu cũ và mật khẩu mới vào và nhấn nút đổi mật khẩu.



Hình 4.2 Giao diện thực thể chức năng xem thông tin và thay đổi mật khẩu

Hệ thống sẽ kiểm tra tài khoản cũ, nếu tài khoản cũ trùng khớp và mật khẩu mới đảm bảo bảo mật gồm 1 chữ cái in hoa, chữ số, ký tự đặc biệt và có độ dài hơn 8 ký tự thì hệ thống sẽ tiến hành thay đổi mật khẩu, thông báo về Gmail và yêu cầu đăng nhập lại.

4.2.3. Chức năng thêm mới thiết bị

Để thêm mới thiết bị, người dùng nhấn vào Thiết bị -> Quản lý thiết bị, nhấn vào nút “Thêm thiết bị mới” trên giao diện. Hệ thống sẽ chuyển hướng đến giao diện thêm mới thiết bị. Người dùng có 2 lựa chọn:

- Nhấn vào nút “Tạo thủ công” và điền các trường như: Tên thiết bị, mô tả, số lượng, loại thiết bị, tải hình ảnh và nhấn vào nút “Tạo mới” để hoàn tất thao tác. Hệ thống sẽ tiến hành kiểm tra dữ liệu và hiển thị lỗi (nếu có). Nếu không có lỗi hệ thống sẽ hiển thị thông báo thành công và trả về giao diện danh sách thiết bị.

The screenshot shows the 'Thêm mới thiết bị' (Add new device) page. On the left is a dark sidebar with navigation links: Dashboard, Quản lý (Management), Thiết bị (Device), Quà tặng (Gift), THỐNG KÊ (Statistics), Mượn trả (Borrow/Return), Thiết bị (Device), Quà tặng (Gift), Thông tin cá nhân (Personal information), and Đăng xuất (Logout). The main area has a search bar at the top. Below it is a title 'Thêm mới thiết bị'. There are two buttons: 'Tạo thủ công' (Create manually) and 'Tạo bằng file' (Create by file). The 'Tạo bằng file' button is highlighted. The form fields include:

- Tên thiết bị: Example: Remote
- Mô tả: (empty input field)
- Số lượng: Example: 1-50
- Loại thiết bị: Máy tính
- Ảnh thiết bị: Choose File (No file chosen)

At the bottom right are 'Tạo mới' (Create new) and 'Quay lại' (Back) buttons.

Hình 4.3 Giao diện thực tế chức năng thêm thủ công thiết bị

- Nhấn vào nút “Tạo bằng file” để thêm nhiều thiết bị bằng file .csv. Người dùng cần tải file theo định dạng. Nhấn vào nút “File mẫu” để tải file mẫu về máy. Sau khi tải file với các thông tin đã điền, nhấn nút “Tải lên” để hoàn tất thao tác.

The screenshot shows the same 'Thêm mới thiết bị' page as above, but the 'Tạo bằng file' tab is selected. The 'Tạo thủ công' tab is now greyed out. The 'Tài file (CSV)' input field contains 'Choose File No file chosen'. At the bottom right are three buttons: 'File mẫu' (highlighted in blue), 'Tải lên' (Upload), and 'Quay lại' (Back).

Hình 4.4 Giao diện thực tế chức năng thêm thiết bị bằng file

4.2.4. Chức năng tìm kiếm, chỉnh sửa, xoá xem thông tin chi tiết thiết bị

Khi thêm mới thiết bị thành công, người dùng có thể tìm kiếm thiết bị tại ô tìm kiếm trên giao diện danh sách thiết bị và xem lại thông tin chi tiết của thiết bị. Trường hợp sai một số thông tin, người dùng có thể chỉnh sửa thông tin hoặc xoá thiết bị. Tại giao diện danh

sách thiết bị, người dùng nhấn vào ký hiệu 3 chấm tại cột thao tác, hộp thoại hiện lên các thao tác như: Xem chi tiết, chỉnh sửa, xoá.

Tên thiết bị	Loại thiết bị	Tổng số lượng	Ngày nhập	Thao tác
Bảng trắng A1	Bảng trắng	20	14/06/2025	⋮⋮⋮
Laptop Dell	Máy tính	10	14/06/2025	⋮⋮⋮
Máy chiếu Sony	Máy chiếu	5	14/06/2025	⋮⋮⋮

Hình 4.5 Giao diện thực tế của danh sách thiết bị

#	Trạng thái	Phòng	Vị trí
1	Mới	Kho chính	Kho chính
2	Mới	Kho chính	Kho chính
3	Mới	Kho chính	Kho chính
4	Mới	Kho chính	Kho chính
5	Mới	Kho chính	Kho chính

Hình 4.6 Giao diện thực tế của chức năng xem thông tin chi tiết thiết bị

Tên thiết bị
Bảng trắng A1

Mô tả
Bảng trắng lớp học

Loại thiết bị
Bảng trắng

Tổng số lượng
20

Ảnh thiết bị
Choose File No file chosen

Lưu thay đổi Quay lại

Hình 4.7 Giao diện thực tế của chức năng chỉnh sửa thông tin thiết bị

4.2.5. Chức năng thêm phòng

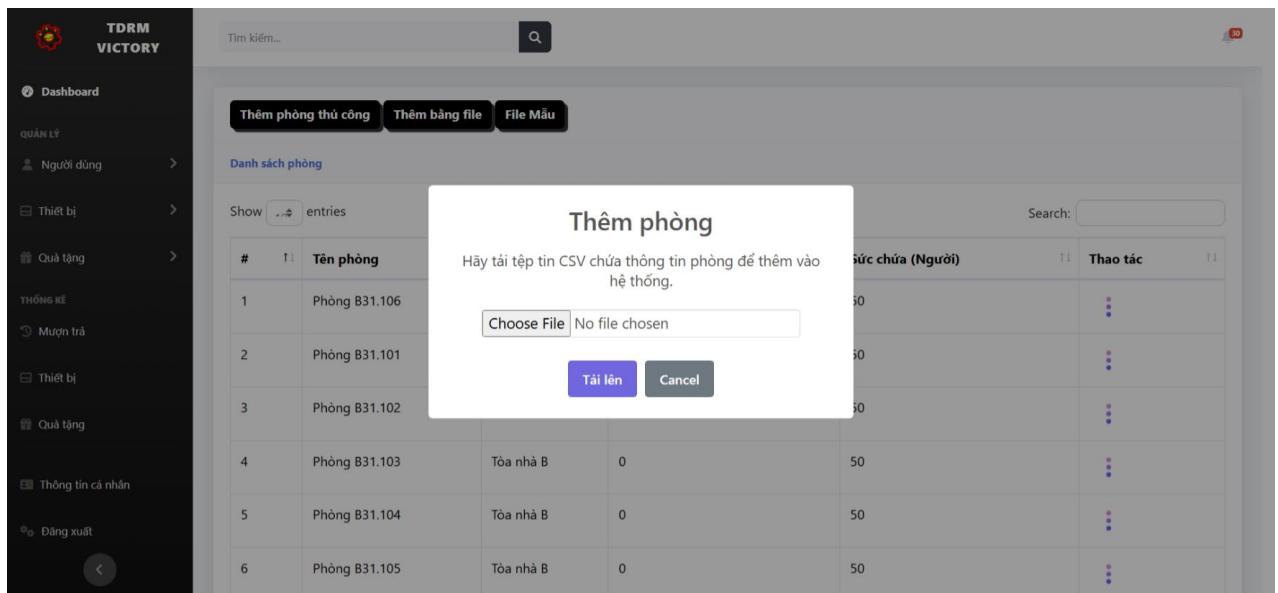
Người dùng nhấn vào Thiết bị -> quản lý theo phòng, tại giao diện danh sách phòng người dùng có 2 lựa chọn:

- Nhấn vào nút “Thêm thủ công”, hệ thống sẽ hiển thị modal thêm phòng mới, nhận thông tin đầy đủ và chọn nút “Tạo phòng” để hoàn tất thao tác.

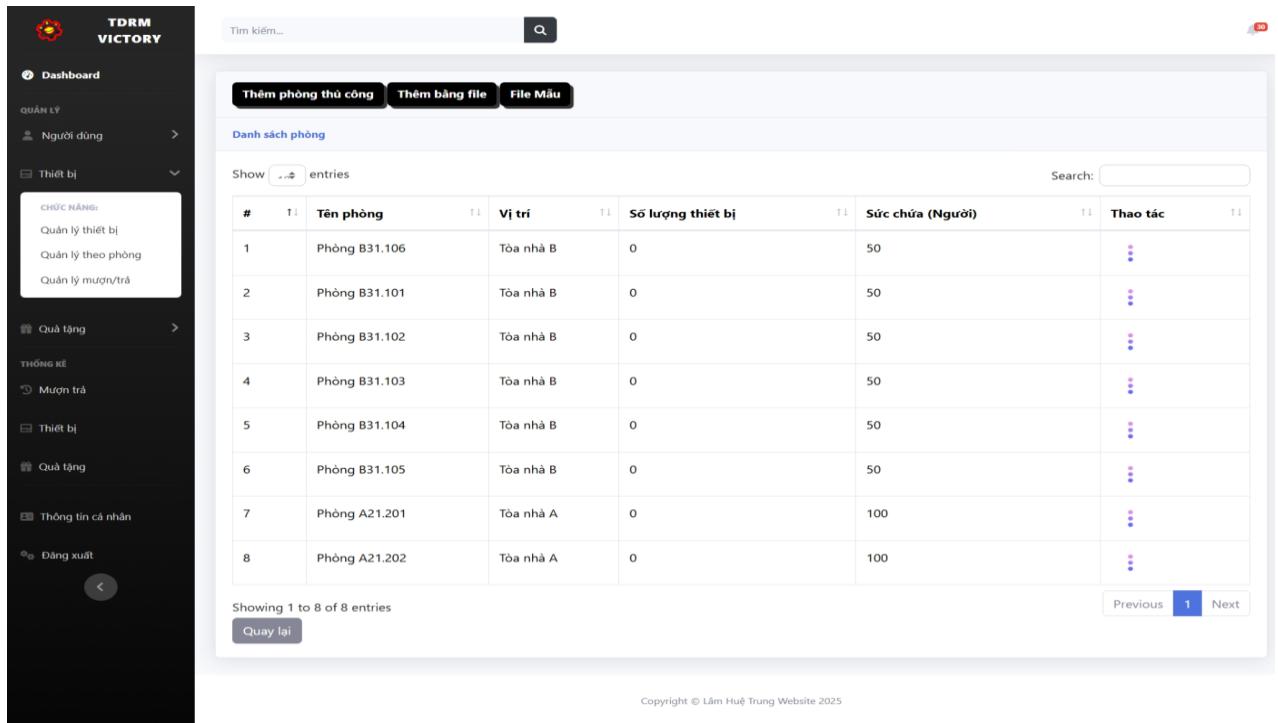
#	Tên phòng	Vị trí	Sức chứa (Người)	Thao tác
1	Phòng B31.106	Chọn tòa nhà	50	⋮
2	Phòng B31.101	Chọn tòa nhà	50	⋮
3	Phòng B31.102	Chọn tòa nhà	50	⋮
4	Phòng B31.103	Tòa nhà B	0	⋮
5	Phòng B31.104	Tòa nhà B	0	⋮
6	Phòng B31.105	Tòa nhà B	0	⋮

Hình 4.8 Giao diện thực tế của chức năng thêm phòng thủ công

- Nhấn vào nút “Thêm bằng file” và tải file .csv đúng với định dạng file mẫu. Hệ thống sẽ thực hiện thêm các thông tin phòng trong file vào database và trả về giao diện danh sách các phòng.



Hình 4.9 Giao diện thực tế của chức năng thêm phòng bằng file



Hình 4.10 Giao diện thực tế chức năng quản lý theo phòng

4.2.6. Chức năng tìm kiếm, chỉnh sửa, xoá xem thông tin chi tiết phòng

Khi thêm mới phòng thành công, người dùng có thể tìm kiếm thiết bị tại ô tìm kiếm trên giao diện danh sách phòng và xem lại thông tin chi tiết của từng phòng. Trường hợp sai một số thông tin, người dùng có thể chỉnh sửa thông tin hoặc xoá phòng. Tại giao diện danh sách phòng, người dùng nhấn vào ký hiệu 3 chấm tại cột thao tác, hộp thoại hiện lên các thao tác như: Sửa phòng, xoá phòng,...

4.2.7. Chức năng thêm thiết bị vào phòng

Sau khi tạo phòng, người dùng có thể thêm thiết bị cho phòng bằng cách nhấn vào ký hiệu ba chấm tại cột thao tác của phòng đó. Hệ thống sẽ chuyển hướng đến giao diện thêm thiết bị cho phòng.

Số lượng mượn	Tên thiết bị	Loại thiết bị	Số lượng có thể mượn	Hình ảnh
	Laptop Dell	Máy tính	10	
	Máy chiếu Sony	Máy chiếu	5	
	Bảng trắng A1	Bảng trắng	20	

Hình 4.11 Giao diện thực tế của chức năng thêm thiết bị vào phòng

Tại giao diện này, người dùng chọn thiết bị và nhập số lượng vào cột số lượng (có thể chọn nhiều thiết bị cùng lúc) và nhấn vào nút “Thêm thiết bị” để hoàn tất thao tác. Hệ thống sẽ lưu lại thao tác và chuyển hướng đến giao diện xem thiết bị của phòng (chức năng này có hiển thị tại menu thao tác của quản lý theo phòng).

#	Trạng thái	Tên thiết bị	Loại thiết bị	Ngày thêm vào phòng	Thao tác
1	Đang sử dụng	Laptop Dell	Máy tính	14/06/2025	
2	Đang sử dụng	Laptop Dell	Máy tính	14/06/2025	

Hình 4.12 Giao diện thực tế của chức năng xem thiết bị của phòng

4.2.8. Chức năng chuyển, xoá thiết bị của phòng

Tại giao diện xem thiết bị của phòng, người dùng có thể thao tác chuyển thiết bị cho phòng khác hoặc xoá thiết bị ra khỏi phòng. Để chuyển thiết bị sang phòng khác, người dùng nhấn vào nút “Chuyển phòng” trong menu thao tác của thiết bị cần chuyển.

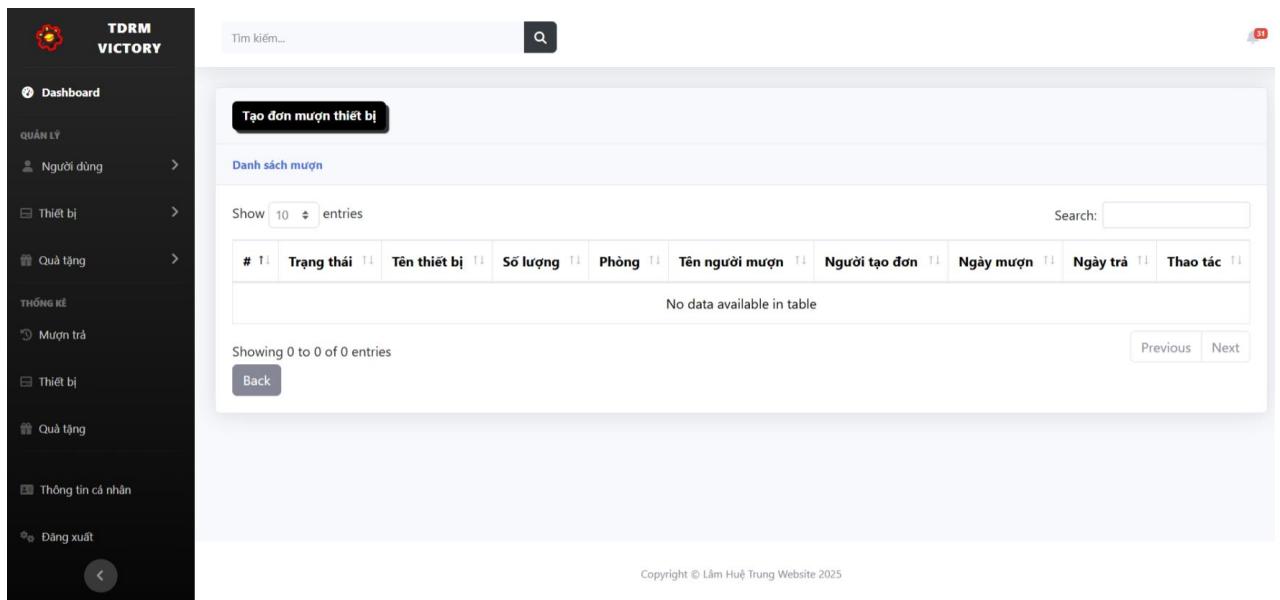
Tên phòng	Vị trí	Số lượng thiết bị	Sức chứa (Người)	
Kho chính	Kho chính	0	10000	<button>Chuyển</button>
Phòng A21.201	Tòa nhà A	0	100	<button>Chuyển</button>
Phòng A21.202	Tòa nhà A	0	100	<button>Chuyển</button>
Phòng B31.101	Tòa nhà B	0	50	<button>Chuyển</button>
Phòng B31.102	Tòa nhà B	0	50	<button>Chuyển</button>
Phòng B31.103	Tòa nhà B	0	50	<button>Chuyển</button>
Phòng B31.104	Tòa nhà B	0	50	<button>Chuyển</button>
Phòng B31.105	Tòa nhà B	0	50	<button>Chuyển</button>
Phòng B31.106	Tòa nhà B	2	50	<button>Chuyển</button>

Hình 4.13 Giao diện thực tế của chức năng chuyển phòng

Tại giao diện chuyển phòng, người dùng nhấn vào nút chuyển của phòng cần chuyển đến để hoàn tất thao tác. Muốn xoá thiết bị ra khỏi phòng làm tương tự các thao tác xoá khác.

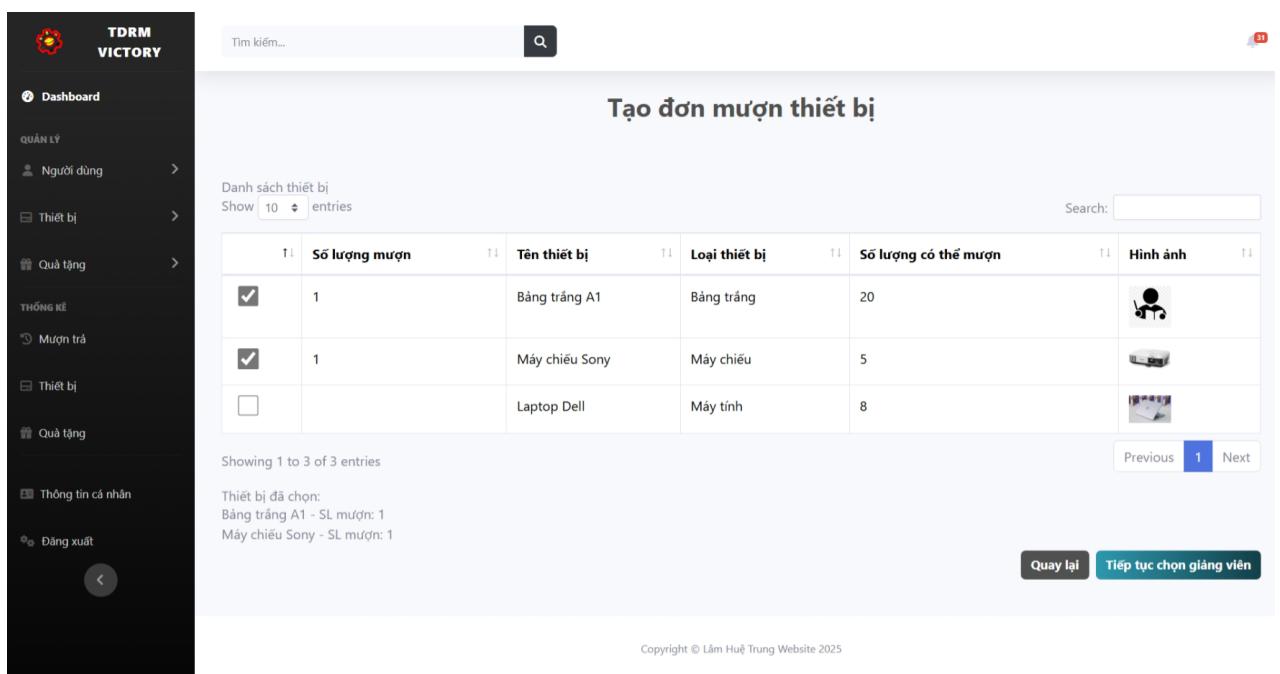
4.2.9. Chức năng mượn thiết bị

Nhấn vào Thiết bị -> Quản lý mượn/trả, để chuyển hướng đến trang giao diện mượn trả thiết bị.



Hình 4.14 Giao diện thực tế của chức năng mượn trả thiết bị

Tại giao diện này, người dùng nhấn vào nút tạo đơn mượn thiết bị để bắt đầu quy trình mượn thiết bị.



Hình 4.15 Giao diện thực tế của chức năng chọn thiết bị để mượn

Người dùng chọn thiết bị và nhập số lượng thiết bị cần mượn vào cột số lượng mượn sau đó nhấn nút “Tiếp tục chọn giảng viên”.

Tùy chọn	Họ tên	Email	Số điện thoại	Phòng ban
<input checked="" type="radio"/>	Lâm Huệ Trung	lamhueptrung44@gmail.com	0763849007	Công nghệ thông tin
<input type="radio"/>	Phạm Thị Trúc Mai	pttmai@tvu.edu.vn	0936010206	Công nghệ thông tin
<input type="radio"/>	Nguyễn Trần Diễm Hạnh	diemhanh_tvu@tvu.edu.vn	0917145587	Công nghệ thông tin
<input type="radio"/>	Trần Văn Nam	namtv@tvu.edu.vn	0365583414	Công nghệ thông tin
<input type="radio"/>	Trịnh Quốc Việt	tqviettv@tvu.edu.vn	0354696999	Công nghệ thông tin
<input type="radio"/>	Trâm Hoàng Nam	tramhoangnam@tvu.edu.vn	0977810235	Công nghệ thông tin
<input type="radio"/>	Nguyễn Nhứt Lam	lamnn@tvu.edu.vn	0919556441	Công nghệ thông tin
<input type="radio"/>	Nguyễn Bá Nhiệm	nhiemnb@tvu.edu.vn	0983303609	Công nghệ thông tin
<input type="radio"/>	Nguyễn Bảo Ân	annb@tvu.edu.vn	0908961632	Công nghệ thông tin
<input type="radio"/>	Thạch Kông Saoane	oane@tvu.edu.vn	61499815272	Công nghệ thông tin

Showing 1 to 10 of 10 entries

Previous 1 Next

Hoàn tất mượn cá nhân Quay lại Tiếp tục chọn phòng

Hình 4.16 Giao diện thực tế của chức năng chọn giảng viên mượn thiết bị

Tiếp tục chọn giảng viên mượn thiết bị và nhấn nút “Hoàn tất mượn cá nhân” để thực hiện tạo đơn mượn thiết bị cho cá nhân giảng viên. Nếu giảng viên mượn cho phòng thì nhấn nút “Tiếp tục chọn phòng” để tiếp tục thao tác.

Tạo đơn mượn thiết bị

	Tên phòng	Vị trí	Số lượng thiết bị	Sức chứa (Người)
<input checked="" type="radio"/>	Phòng B31.106	Tòa nhà B	2	50
<input type="radio"/>	Phòng B31.101	Tòa nhà B	0	50
<input type="radio"/>	Phòng B31.102	Tòa nhà B	0	50
<input type="radio"/>	Phòng B31.103	Tòa nhà B	0	50
<input type="radio"/>	Phòng B31.104	Tòa nhà B	0	50
<input type="radio"/>	Phòng B31.105	Tòa nhà B	0	50
<input type="radio"/>	Phòng A21.201	Tòa nhà A	0	100
<input type="radio"/>	Phòng A21.202	Tòa nhà A	0	100

Showing 1 to 8 of 8 entries

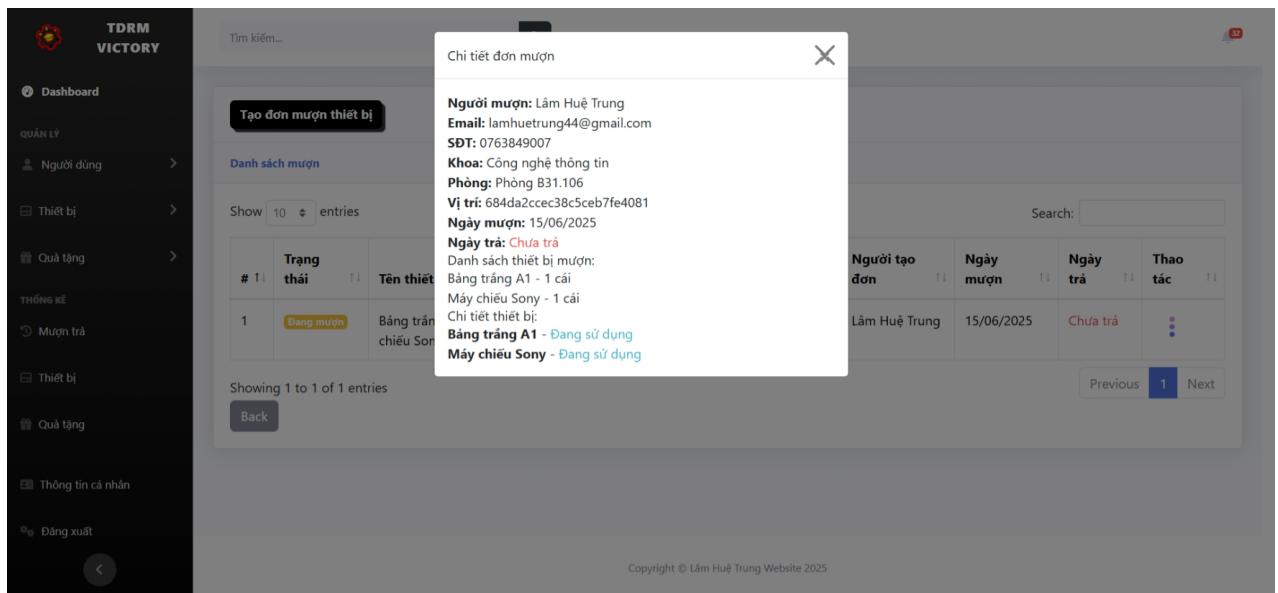
Copyright © Lâm Huệ Trung Website 2025

Hình 4.17 Giao diện thực tế của chức năng chọn phòng mượn thiết bị

Tiếp tục chọn phòng và nhấn nút “Hoàn tất” để hoàn tất thao tác tạo đơn mượn thiết bị cho phòng.

4.2.10. Chức năng xem đơn mượn và trả thiết bị

Nhấn vào nút “Xem chi tiết” trong menu thao tác của đơn để xem chi tiết đơn mượn hoặc nhấn nút “Trả thiết bị” để thực hiện chức năng trả thiết bị.



Hình 4.18 Giao diện thực tế của chức năng xem thông tin đơn mượn thiết bị

Sau khi nhân viên trực phòng đã nhận được thiết bị mà giảng viên mượn, nhân viên trực sẽ kiểm tra thiết bị và nhấn vào nút “Trả thiết bị”. Sau khi trả thiết bị thành công thì các thiết bị sẽ được đưa về kho chính và lưu lịch sử mượn/trả thiết bị lại để thống kê.

4.2.11. Chức năng thống kê quản lý thiết bị

Chức năng thống kê bao gồm thống kê thiết bị và thống kê mượn trả. Các thống kê này được hiển thị dưới dạng bảng biểu, biểu đồ cột, biểu đồ đường, biểu đồ tròn,...

- Thống kê thiết bị bao gồm: Tổng số thiết bị theo loại, Trạng thái thiết bị, thống kê thiết bị theo phòng, Thiết bị đang trong Kho chính,...

- Thống kê mượn trả bao gồm: Biểu đồ mượn thiết bị theo thời gian, Top 10 thiết bị được mượn nhiều nhất, Biểu đồ mượn theo phòng, Danh sách thiết bị đang mượn,...

4.3. Kết quả thực nghiệm với vai trò quản lý quà tặng

Tài khoản với quyền quản lý quà tặng cũng có các chức năng đăng nhập, xem thông tin cá nhân và thay đổi mật khẩu giống với quản lý thiết bị. Sự khác nhau nằm ở tài khoản quản lý quà tặng là chỉ sử dụng được các chức năng liên quan đến quà tặng.

4.3.1. Chức năng thêm quà tặng

Để thêm mới quà tặng, người dùng nhấn vào Quà tặng -> Quản lý quà tặng, nhấn vào nút “Thêm quà tặng mới” trên giao diện. Hệ thống sẽ chuyển hướng đến giao diện thêm mới quà tặng. Người dùng có 2 lựa chọn:

- Nhấn vào nút “Tạo thủ công” và điền các trường như: Tên quà tặng, mô tả, danh mục, giá, số lượng, tải hình ảnh và nhấn vào nút “Tạo mới” để hoàn tất thao tác. Hệ thống sẽ tiến hành kiểm tra dữ liệu và hiển thị lỗi (nếu có). Nếu không có lỗi hệ thống sẽ hiển thị thông báo thành công và trả về giao diện danh sách quà tặng.

Hình 4.19 Giao diện thực tế chức năng thêm thủ công quà tặng

- Nhấn vào nút “Tạo bằng file” để thêm nhiều thiết bị bằng file .csv. Người dùng cần tải file theo định dạng. Nhấn vào nút “File mẫu” để tải file mẫu về máy. Sau khi tải file với các thông tin đã điền, nhấn nút “Tải lên” để hoàn tất thao tác.

Hình 4.20 Giao diện thực tế của chức năng thêm quà tặng bằng file

4.3.2. Chức năng tìm kiếm, chỉnh sửa, xoá xem thông tin chi tiết quà tặng

Khi thêm mới quà tặng thành công, người dùng có thể tìm kiếm quà tặng tại ô tìm kiếm trên giao diện danh sách quà tặng và xem lại thông tin chi tiết của quà tặng. Trường hợp sai một số thông tin, người dùng có thể chỉnh sửa thông tin hoặc xoá quà tặng. Tại giao diện danh sách quà tặng, người dùng nhấp vào ký hiệu 3 chấm tại cột thao tác, hộp thoại hiện lên các thao tác như: Xem chi tiết, chỉnh sửa, xoá.

Tên quà tặng	Danh mục	Số lượng trong kho	Ngày nhập	Người nhập	Thao tác
Bình giữ nhiệt	Đối tác	410	14/06/2025	Lâm Huệ Trung	
Số tay bia da	Học viên	0	14/06/2025	Lâm Huệ Trung	
Áo thun sự kiện	Học viên	100	14/06/2025	Lâm Huệ Trung	

Hình 4.21 Giao diện thực tế của danh sách quà tặng

Hình 4.22 Giao diện thực tế của chức năng xem thông tin chi tiết quà tặng

Tên quà tặng
Bình giữ nhiệt

Mô tả
Bình giữ nhiệt inox cao cấp

Loại quà tặng
Đối tác

Tổng số lượng
410

Giá
200000

Ảnh quà tặng
Choose File | No file chosen

Lưu thay đổi | Quay lại

Hình 4.23 Giao diện thực tế của chức năng chỉnh sửa thông tin quà tặng

4.3.3. Chức năng tạo đơn yêu cầu quà tặng

Để bắt đầu sử dụng chức năng này, người dùng cần tải file mẫu về bằng cách nhấn vào Quà tặng -> Quản lý đơn yêu cầu, tại giao diện danh sách đơn yêu cầu quà tặng nhấn vào nút “File mẫu đơn yêu cầu” để tải về máy.

Thêm đơn yêu cầu | File mẫu đơn yêu cầu

Danh sách đơn yêu cầu quà tặng

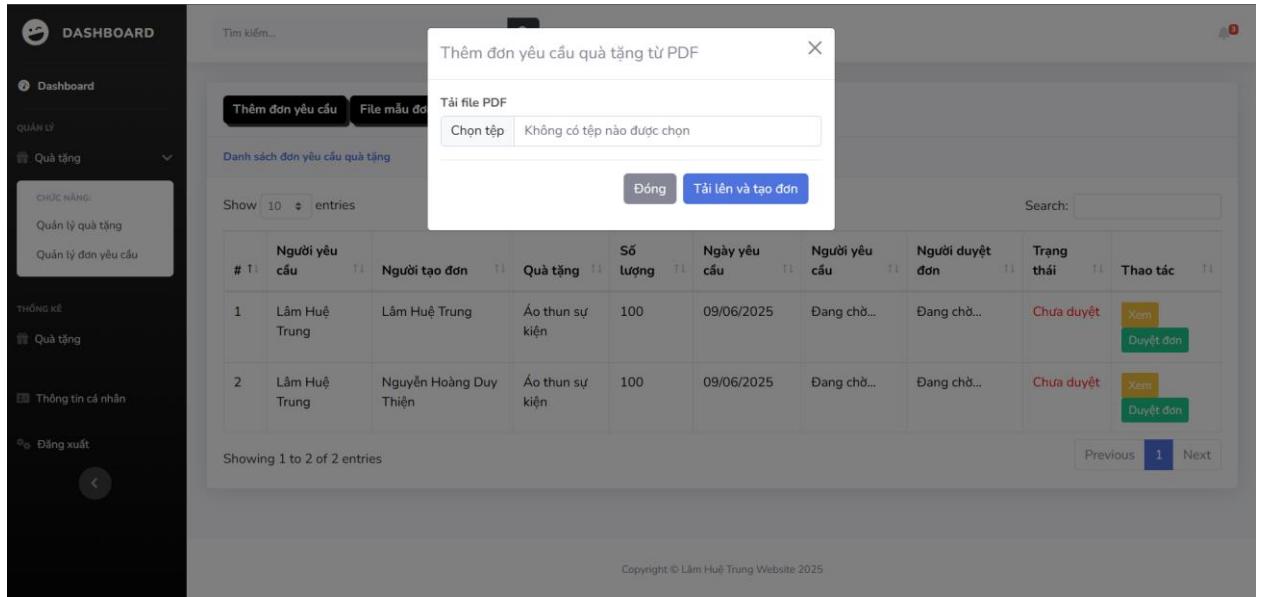
#	Người yêu cầu	Người tạo đơn	Ngày yêu cầu	Ngày duyệt đơn	Người duyệt đơn	Trạng thái	Thao tác
1	Lâm Huệ Trung	Lâm Huệ Trung	15/06/2025	Đang chờ...	Đang chờ...	Chưa duyệt	Xem Duyệt đơn
2	Lâm Huệ Trung	Lâm Huệ Trung	15/06/2025	15/06/2025	Lâm Huệ Trung	Đã duyệt	Xem Hoàn trả
3	Lâm Huệ Trung	Lâm Huệ Trung	15/06/2025	15/06/2025	Lâm Huệ Trung	Có hoàn trả	Xem

Showing 1 to 3 of 3 entries | Previous | Next

Hình 4.24 Giao diện thực tế của danh sách đơn yêu cầu quà tặng

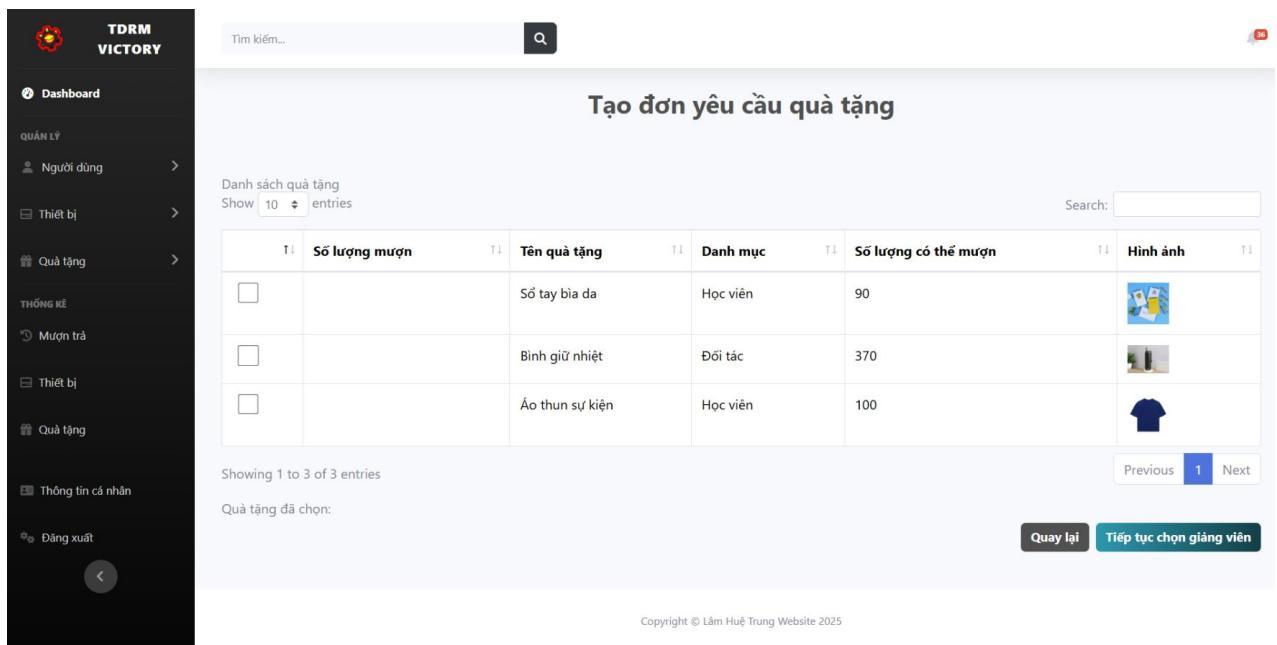
Sau khi tải file mẫu về người dùng cần đưa các thông tin theo mẫu yêu cầu vào file và export ra file pdf để tải lên hệ thống. Để bắt đầu tạo đơn yêu cầu, người dùng nhấn vào nút

“Thêm đơn yêu cầu” và chọn “Tạo bằng file” để hiển thị modal thêm đơn yêu cầu quà tặng từ PDF. Để hoàn tất thao tác, người dùng tải file lên modal và nhấn vào nút “Tải lên và tạo đơn”. Hệ thống sẽ kiểm tra loại file, thông tin trong file. Nếu không có lỗi hệ thống sẽ bắt đầu tạo đơn từ những thông tin trong file.



Hình 4.25 Giao diện thực tế của modal thêm đơn yêu cầu quà tặng từ PDF

Người dùng có thể tạo đơn thủ công trên hệ thống, nhấn vào nút “Thêm đơn yêu cầu” và chọn “Tạo thủ công” để chuyển hướng đến trang tạo đơn yêu cầu quà tặng.



Hình 4.26 Giao diện thực tế của chọn quà tặng để mượn

Người dùng nhập số lượng mượn và nhấn nút “Tiếp tục chọn giảng viên” để tiếp tục thao tác.

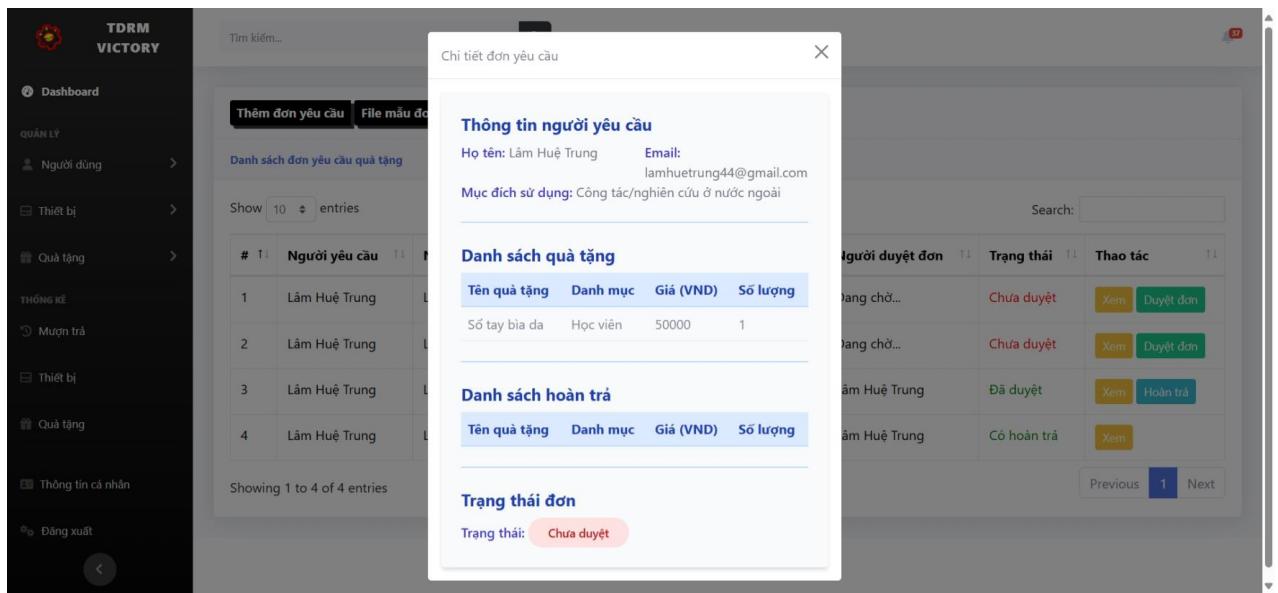
	Họ tên	Email	Số điện thoại	Phòng ban
<input checked="" type="radio"/>	Lâm Huệ Trung	lamhuetrung44@gmail.com	0763849007	Công nghệ thông tin
<input type="radio"/>	Phạm Thị Trúc Mai	pttmai@tvu.edu.vn	0936010206	Công nghệ thông tin
<input type="radio"/>	Nguyễn Trần Diễm Hạnh	diemhanh_tvu@tvu.edu.vn	0917145587	Công nghệ thông tin
<input type="radio"/>	Trần Văn Nam	namtv@tvu.edu.vn	0365583414	Công nghệ thông tin
<input type="radio"/>	Trịnh Quốc Việt	tqviettv@tvu.edu.vn	0354696999	Công nghệ thông tin
<input type="radio"/>	Trâm Hoàng Nam	tramhoangnam@tvu.edu.vn	0977810235	Công nghệ thông tin
<input type="radio"/>	Nguyễn Nhứt Lam	lamnn@tvu.edu.vn	0919556441	Công nghệ thông tin
<input type="radio"/>	Nguyễn Bá Nhiệm	nhiemnb@tvu.edu.vn	0983303609	Công nghệ thông tin
<input type="radio"/>	Nguyễn Bảo Ân	annb@tvu.edu.vn	0908961632	Công nghệ thông tin
<input type="radio"/>	Thạch Kặng Saoane	oane@tvu.edu.vn	61499815272	Công nghệ thông tin

Hình 4.27 Giao diện thực tế của chức năng chọn mục đích sử dụng và người yêu cầu

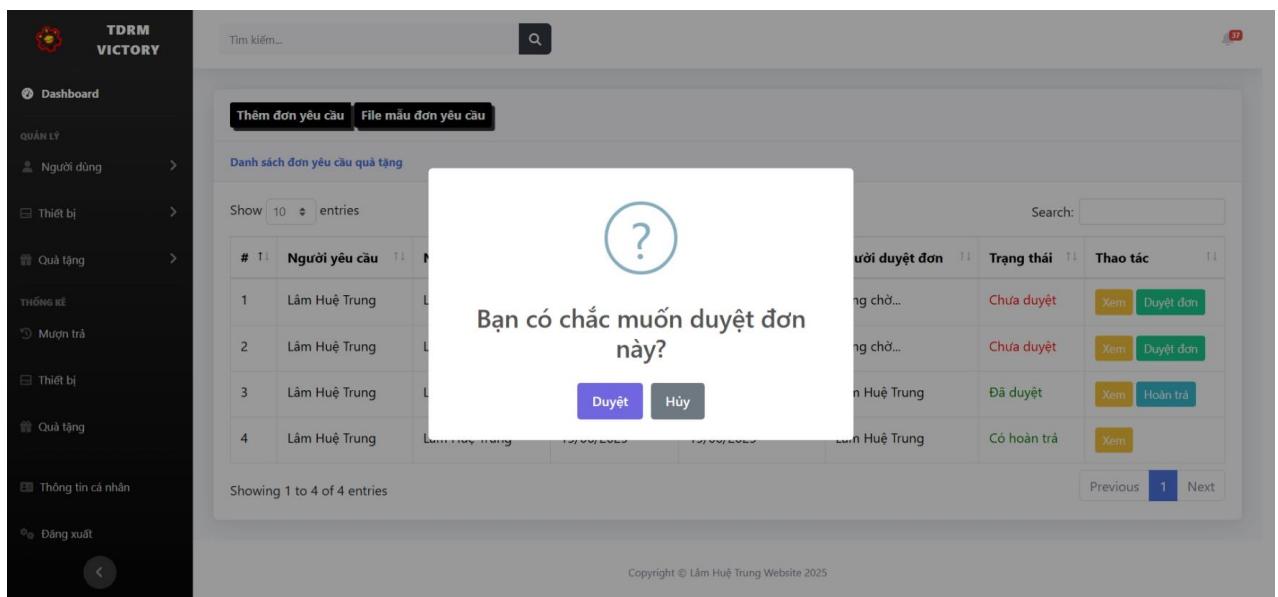
Tại giao diện chọn mục đích sử dụng và người yêu cầu, người dùng có thể chọn một hoặc nhiều mục đích sử dụng và chọn người yêu cầu xuất quà tặng, sau đó nhấn nút “Hoàn tất” để hoàn tất thao tác.

4.3.4. Chức năng xem thông tin đơn và duyệt đơn yêu cầu

Tại giao diện danh sách đơn yêu cầu quà tặng, người dùng có thể nhấp vào nút “Xem” để xem thông tin chi tiết của đơn yêu cầu. Sau khi kiểm tra thông tin đơn yêu cầu chính xác thì người dùng có thể nhấp vào nút duyệt đơn để bắt đầu duyệt đơn xuất quà tặng.



Hình 4.28 Giao diện thực tế chức năng xem thông tin đơn yêu cầu quà tặng

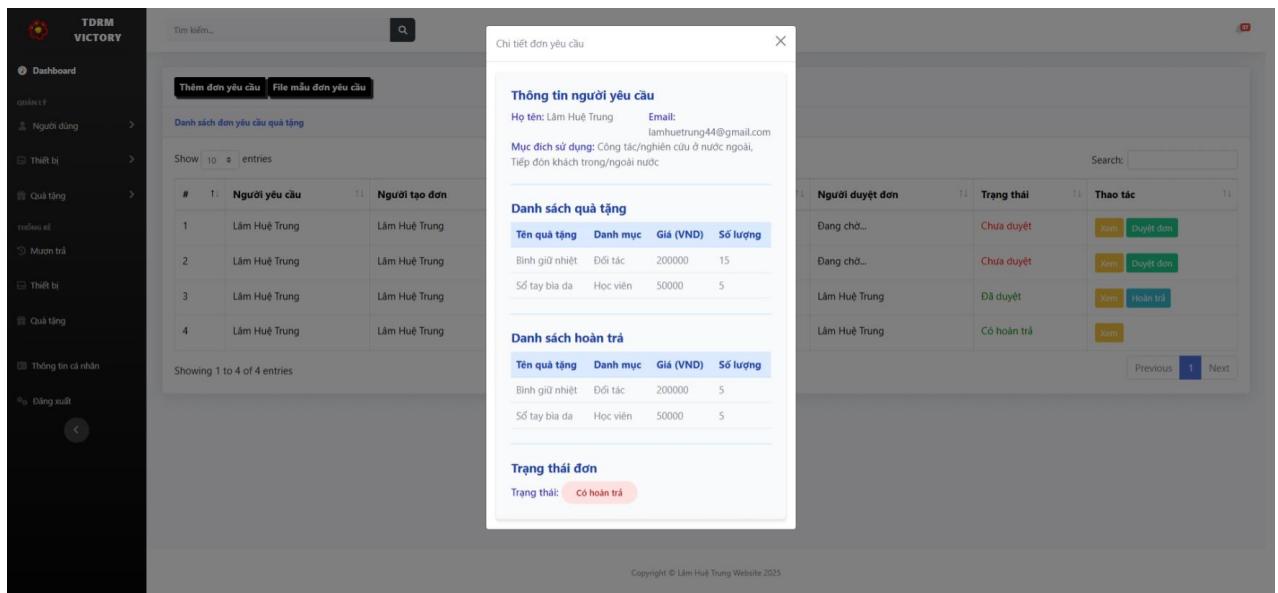


Hình 4.29 Giao diện cảnh báo khi nhấn vào nút duyệt đơn

Khi nhấn vào nút duyệt đơn, nếu người dùng đã chắc chắn không có sai sót thì chọn nút “Duyệt” để hoàn tất thao tác.

4.3.5. Chức năng hoàn trả quà tặng

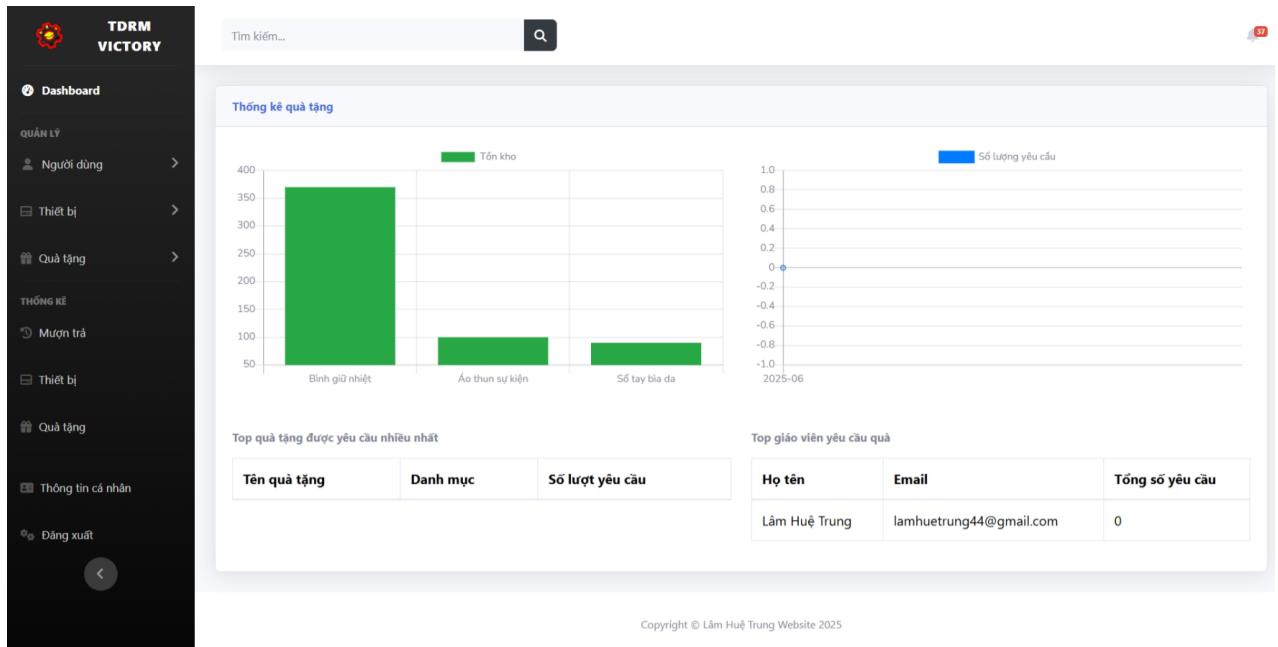
Đối với quà tặng đã được duyệt trong đơn yêu cầu nhưng không sử dụng hết, có thể hoàn trả lại. Tại giao diện danh sách đơn yêu cầu quà tặng, nhấn vào nút “Hoàn trả” ở cột thao tác của đơn yêu cầu và nhập số lượng trả, cuối cùng nhấn vào nút “Hoàn trả” để hoàn tất thao tác.



Hình 4.30 Giao diện thực tế của đơn yêu cầu quà tặng có hoàn trả

4.3.6. Chức năng thống kê quản lý quà tặng

Để xem các thông kê về quà tặng, người dùng nhấn vào Quà tặng trong phần thống kê. Giao diện hiển thị các bảng biểu, biểu đồ về các thông kê như: Thông kê tồn kho, thống kê số lượng đơn yêu cầu, top quà tặng được yêu cầu nhiều nhất, top giáo viên yêu cầu quà,...



Hình 4.31 Giao diện thực tế của chức năng thống kê quà tặng

4.4. Kết quả thực nghiệm với vai trò quản trị viên

Vai trò quản trị viên sở hữu tài khoản có toàn quyền trên hệ thống “Quản lý thiết bị và quà tặng của trung tâm Ngoại ngữ - Tin học Victory”. Tài khoản quản trị viên sẽ được tự động tại khi deploy hệ thống với các thông tin:

- Tài khoản: LamHueTrung
- Mật khẩu: Lht080103*
- Họ và tên: Lâm Huệ Trung
- Ngày sinh: 1/8/2003
- Số điện thoại: 0763849007
- Email: lamhuetrung@gmail.com

Quản trị viên có quyền sử dụng toàn bộ chức năng của quản lý thiết bị và quản lý quà tặng. Ngoài ra, quản trị viên còn có chức năng quản lý người dùng và quản lý giảng viên như: Thêm mới, xem thông tin, chỉnh sửa, xoá vĩnh viễn tài khoản người dùng và giảng viên, vô hiệu hóa tài khoản hoặc khôi phục tài khoản.

4.4.1. Chức năng thêm mới tài khoản người dùng

Quản trị viên có quyền cấp tài khoản quản lý thiết bị hoặc quản lý quà tặng. Để sử dụng tính năng này, người dùng nhấn vào Người dùng -> Thêm mới. Tại giao diện Thêm người dùng, nhập các thông tin như: Tài khoản, quyền hạn, họ tên, ngày sinh, số điện thoại,... và nhấn nút “Tạo mới” để hoàn tất thao tác. Mật khẩu mặc định của tài khoản là tài khoản + dấu “*”.

Hình 4.32 Giao diện thực tế của chức năng thêm tài khoản người dùng

4.4.2. Chức năng xem thông tin, chỉnh sửa, xoá tài khoản người dùng

Khi thêm mới tài khoản thành công, người dùng có thể tìm kiếm tài khoản tại ô tìm kiếm trên giao diện danh sách tài khoản và xem lại thông tin chi tiết của tài khoản. Trường hợp sai một số thông tin, người dùng có thể chỉnh sửa thông tin hoặc xoá tài khoản. Tại giao diện danh sách tài khoản, người dùng nhấn vào ký hiệu 3 chấm tại cột thao tác, hộp thoại hiện lên các thao tác như: Xem thêm, chỉnh sửa, xoá.

Hình 4.33 Giao diện thực tế chức năng chỉnh sửa thông tin tài khoản

Hình 4.34 Giao diện thực tế chức năng xem thông tin chi tiết tài khoản

4.4.3. Chức năng vô hiệu hóa, khôi phục tài khoản

Trong một vài trường hợp như người dùng vi phạm chính sách đang bị kỷ luật hoặc nghỉ việc, quản trị viên có thể vô hiệu hóa tài khoản của người dùng để tránh tình trạng rò rỉ thông tin. Nếu người dùng trở lại công việc và hoạt động bình thường trở lại thì quản trị viên có thể khôi phục tài khoản đó.

Hình 4.35 Giao diện thực tế của tài khoản bị vô hiệu hóa

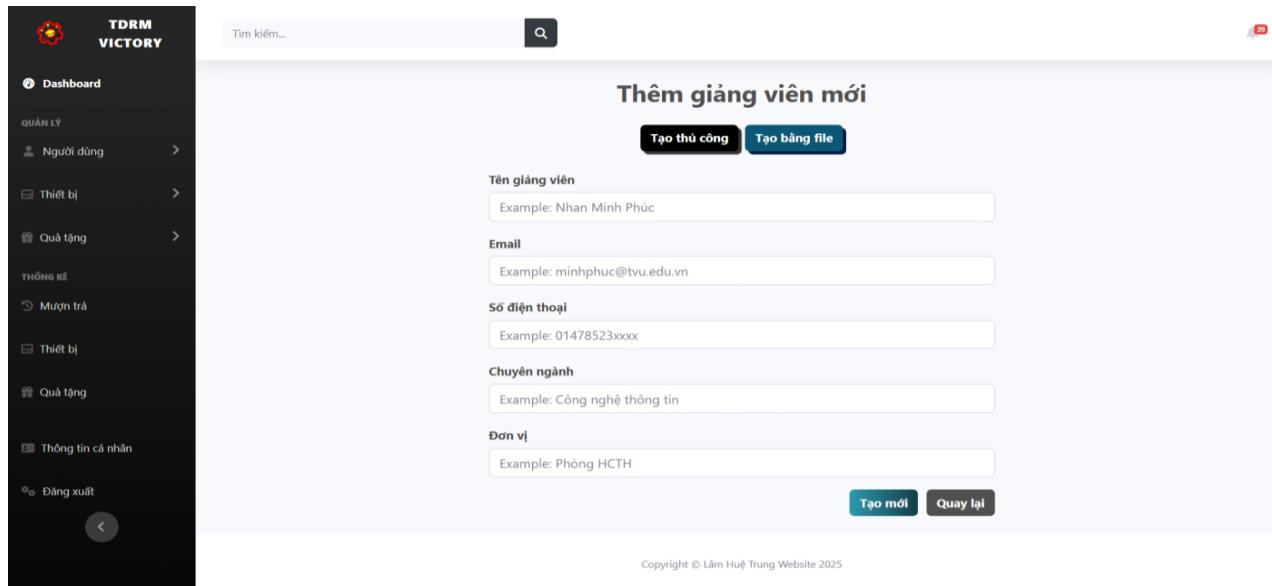
4.4.4. Chức năng thêm mới giảng viên

Tại thanh navbar, nhấn chọn Người dùng -> Quản lý giảng viên để vào giao diện danh sách giảng viên. Nhấn vào nút “Thêm giảng viên”, hệ thống sẽ chuyển hướng đến giao diện thêm giảng viên mới.

Hình 4.36 Giao diện thực tế của danh sách giảng viên

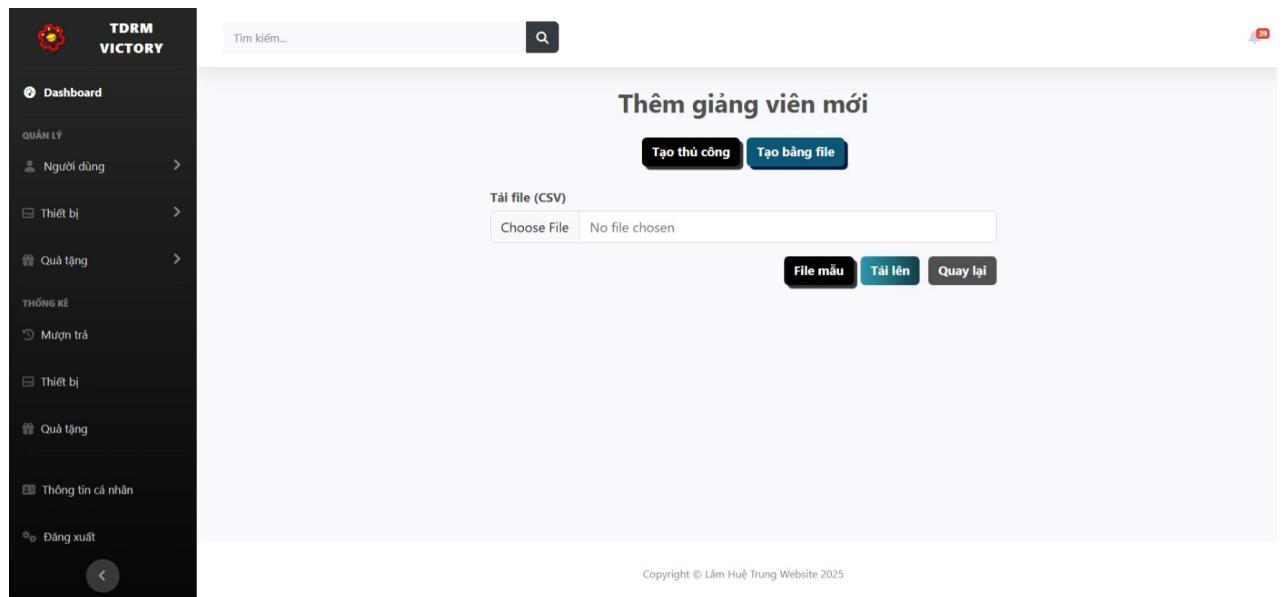
Tại giao diện thêm giảng viên mới, người dùng có 2 lựa chọn:

- Nhấn vào nút “Tạo thủ công” và điền các trường như: Tên giảng viên, email, số điện thoại, chuyên ngành và nhấn vào nút “Tạo mới” để hoàn tất thao tác. Hệ thống sẽ tiến hành kiểm tra dữ liệu và hiển thị lỗi (nếu có). Nếu không có lỗi hệ thống sẽ hiển thị thông báo thành công và trả về giao diện danh sách giảng viên.



Hình 4.37 Giao diện thực tế chức năng thêm thủ công giảng viên

- Nhấn vào nút “Tạo bằng file” để thêm nhiều thiết bị bằng file .csv. Người dùng cần tải file theo định dạng. Nhấn vào nút “File mẫu” để tải file mẫu về máy. Sau khi tải file với các thông tin đã điền, nhấn nút “Tải lên” để hoàn tất thao tác.



Hình 4.38 Giao diện thực tế của chức năng thêm giảng viên bằng file

4.4.5. Chức năng chỉnh sửa, xoá giảng viên

Khi thêm mới giảng viên thành công, quản trị viên có thể tìm kiếm giảng viên tại ô tìm kiếm trên giao diện danh sách giảng. Trường hợp sai một số thông tin, quản trị viên có thể chỉnh sửa thông tin hoặc xoá giảng viên. Tại giao diện danh sách giảng viên, quản trị viên nhấn vào ký hiệu 3 chấm tại cột thao tác, hộp thoại hiện lên các thao tác như: chỉnh sửa, xoá.

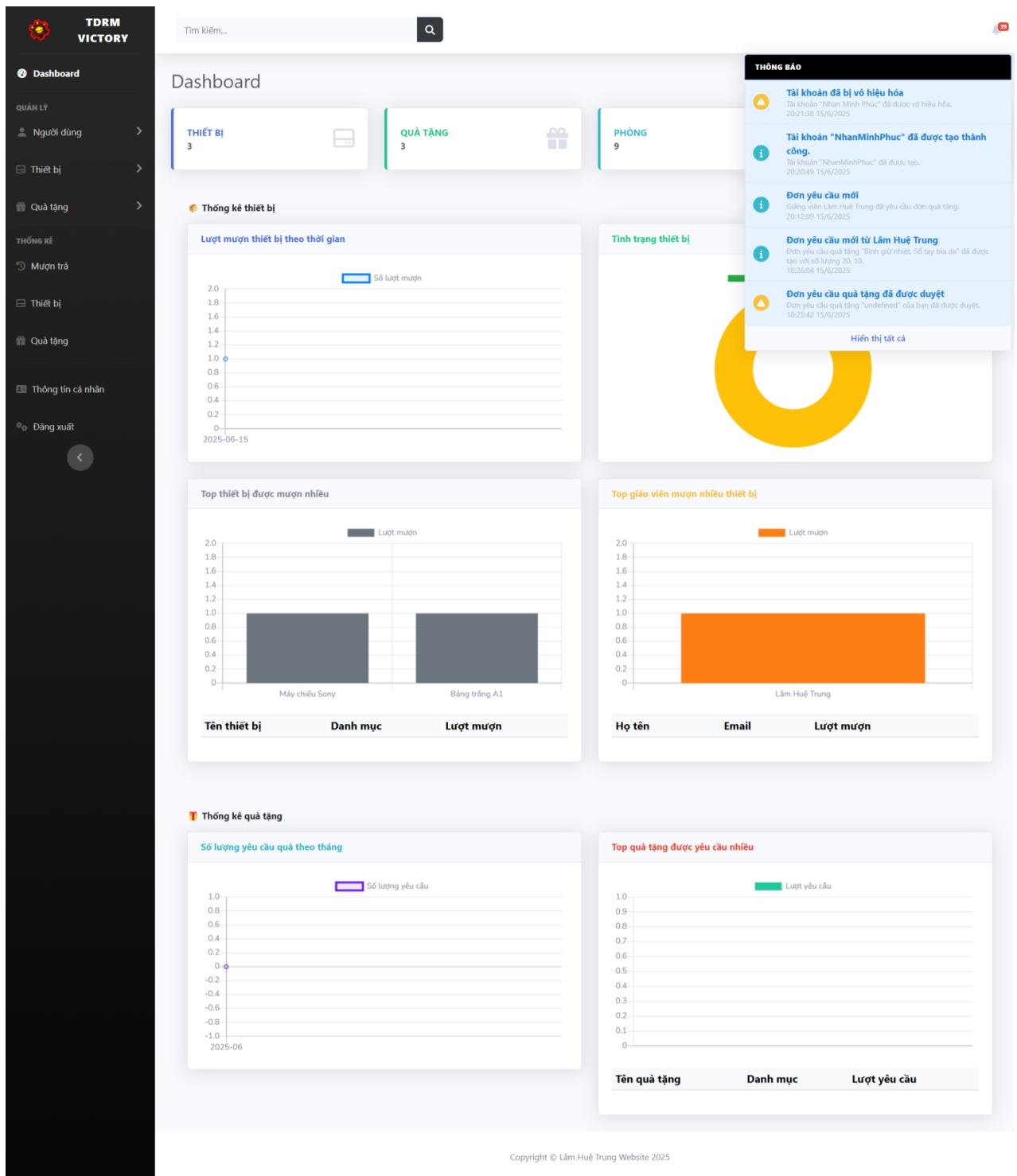
#	Họ tên	Email	Số điện thoại	Phòng ban	Thao tác
1	Lâm Huệ Trung	lamhuetrung44@gmail.com	0763849007	Công nghệ thông tin	⋮
2	Phạm Thị Trúc Mai	pttmai@tvu.edu.vn	0936010206	⋮	Chỉnh sửa Xoá
3	Nguyễn Trần Diễm Hạnh	diemhanh_tvu@tvu.edu.vn	0917145587	Công nghệ thông tin	⋮
4	Trần Văn Nam	namtv@tvu.edu.vn	0365583414	Công nghệ thông tin	⋮
5	Trịnh Quốc Việt	tqviettv@tvu.edu.vn	0354696999	Công nghệ thông tin	⋮
6	Trâm Hoàng Nam	tramhoangnam@tvu.edu.vn	0977810235	Công nghệ thông tin	⋮
7	Nguyễn Nhứt Lam	lamnn@tvu.edu.vn	0919556441	Công nghệ thông tin	⋮
8	Nguyễn Bá Nhiệm	nhiemnb@tvu.edu.vn	0983303609	Công nghệ thông tin	⋮
9	Nguyễn Bảo Ân	annb@tvu.edu.vn	0908961632	Công nghệ thông tin	⋮
10	Thạch Kặng Saoane	oane@tvu.edu.vn	61499815272	Công nghệ thông tin	⋮

Hình 4.39 Giao diện thực tế của chức năng chỉnh sửa, xoá giảng viên

4.5. Các chức năng khác

4.5.1. Chức năng thông báo

Khi thực hiện các tác vụ như thêm mới, chỉnh sửa, xoá thiết bị hoặc quà tặng, mượn trả thiết bị, tạo đơn yêu cầu hoặc duyệt đơn, người dùng mới,... Hệ thống sẽ thông báo thông tin đến các tài khoản liên quan. Để xem thông báo, người dùng nhấn vào biểu tượng  trên thanh header.

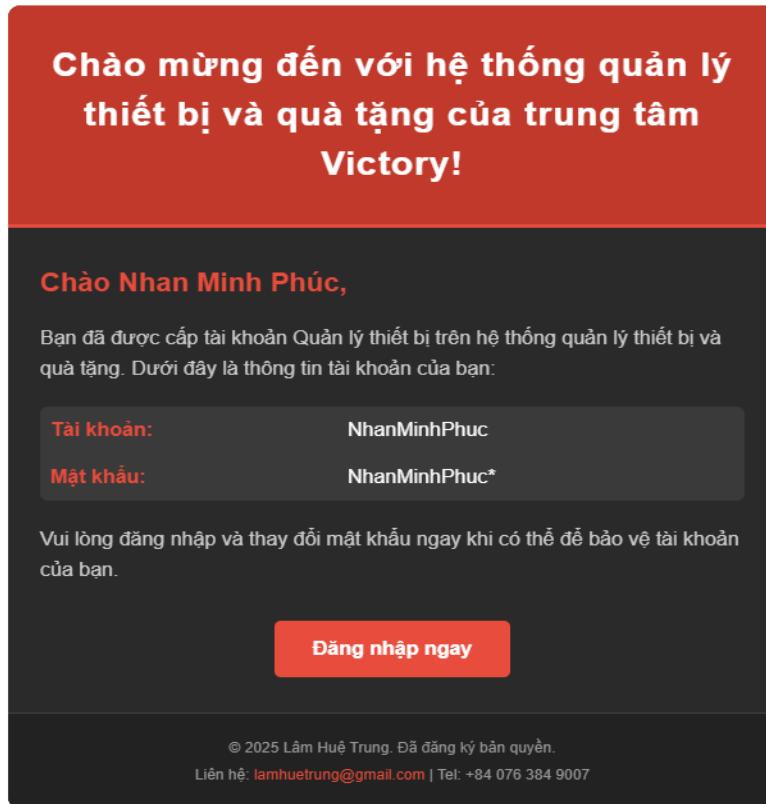


Hình 4.40 Giao diện thực tế của chức năng thông báo

Đối với các thông báo thay đổi thông tin thì khi nhấn vào thông báo, hệ thống sẽ chuyển hướng đến sự thay đổi đó.

4.5.2. Chức năng gửi mail

Khi tạo mới tài khoản, hệ thống sẽ gửi mail về cho người dùng để thông báo người dùng đã được cấp tài khoản.



Hình 4.41 Mail thông báo khi được cấp tài khoản

Khi thay đổi mật khẩu, tài khoản bị vô hiệu hóa hoặc khôi phục cũng gửi mail cho người dùng. Đối với chức năng mượn trả thiết bị và yêu cầu quà tặng cũng tương tự, chỉ khác nội dung mail gửi về.

4.6. Kết quả thực nghiệm hệ thống quản lý thiết bị và quà tặng

Hệ thống “Quản lý thiết bị và quà tặng của Trung tâm Ngoại ngữ - Tin học Victory” được xây dựng với kiến trúc MVC kết hợp với CQRS. Với việc sử dụng ngôn ngữ chính là NodeJS để phát triển, nên dễ mở rộng và phát triển với nhiều thư viện hỗ trợ.

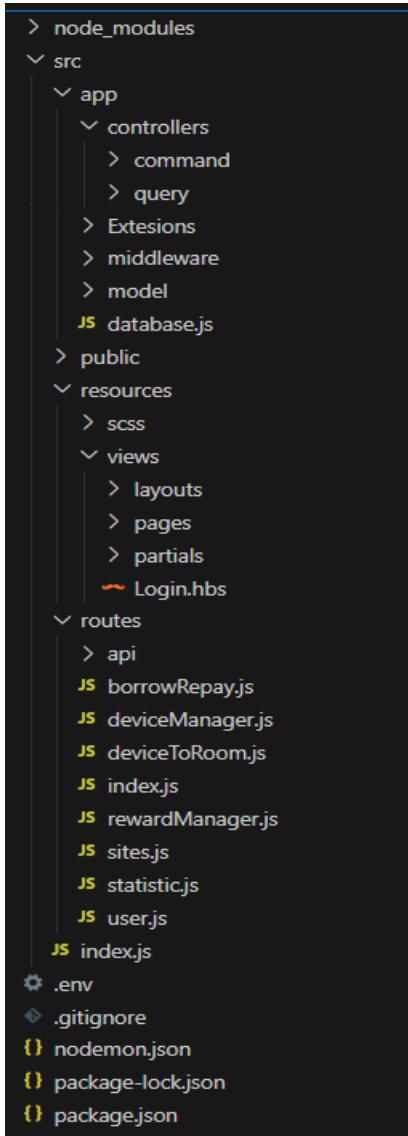
Khi đã có môi trường NodeJS và Git, mở Terminal (giao diện dòng lệnh) trong VS Code và thực hiện lần lượt các bước sau:

Bước 1: Clone dự án về máy

```
Git clone
https://github.com/LamHueTrung/tn_da21ttc_110121255_LamHueTrung_tdgm.git
```

Bước 2: Di chuyển vào thư mục của dự án

```
cd tn_da21ttc_110121255_LamHueTrung_tdgm
```



Hình 4.42 Cây thư mục

- *src*: thư mục chứa mã nguồn chính của dự án.
 - + *app*: chứa các thành phần như controller (xử lý), extesions (các phương thức hỗ trợ như upload ảnh, file, v.v), middleware xác thực token, model và connect database.
 - + *public*: chứa các hình ảnh, file được hiển thị công khai.
 - + *resources*: chứa các file giao diện.
 - + *route*: chứa các file cấu hình đường dẫn.
 - + *index.js*: file cấu hình chính của dự án.
- *.env*: file chứa các biến môi trường quan trọng của dự án. Vì lý do bảo mật, file này không được đưa lên Github. Hãy tạo một file mới (cùng cấp với thư mục *src*) với tên *.env* và có nội dung sau:

```

MONGODB_URI_LOCAL=mongodb://localhost:27017/Device
RewardManager

MONGODB_URI_ONLINE=url mongodb online

JWT_SECRET_KEY=1101212551amhuetrung08012003

AES_SECRET_KEY=3269ceed2bcea81c3c5c7c6955e991b1326
9ceed2bcea81c3c5c7c6955e991b1

AES_IV=46682df79d776d8c46682df79d776d8c

API_URL=http://localhost:3000

# cần phải xác thực 2 bước

USERNAME_GOOGLE=lamhuetrung@gmail.com

APP_PASSWORD_GOOGLE=f j h o d b i o q w v g l b r t
  
```

Bước 3: Cài đặt các gói thư viện

Ở bước này, chỉ cần chạy khi lần đầu chạy dự án để trình quản lý gói tải những thư viện được sử dụng trong dự án xuống. Ở những lần sau, không cần thực hiện bước này.

```
npm i
```

Bước 4: Chạy dự án

```
npm start
```

Sau khi thực hiện câu lệnh bên dưới, thông báo sẽ hiện ra như sau:

```
> datt-da2ittc-lamhuetrung-teachingDeviceRewardManager-nodejs@1.0.0 start
> nodemon --inspect ./src/index.js

[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,json,scss,hbs
[nodemon] starting `node --inspect ./src/index.js`
Debugger listening on ws://127.0.0.1:9229/2900307a-49a8-4249-9a8a-46143df2c417
For help, see: https://nodejs.org/en/docs/inspector
(node:29588) [MONGODB DRIVER] Warning: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:29588) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Example app listening on port 3000
MongoDB đã được kết nối.
Kết nối thành công tới MongoDB: Online
Tài khoản admin đã tồn tại.
{
  _id: new ObjectId('57d3cf0fa662a440760af4a6'),
  username: 'LamHuettTrung',
  password: '49f1fad56b80e1247fcbc46f3eafbc9fe',
  role: 'system_admin',
  profile: {
    fullName: 'Lâm Huệ Trung',
    birthDate: '2003-08-01T00:00:00Z',
    avatar: '/img/systemAdminIn.jpg',
    address: 'Trà Vinh',
    phone: '0763849007',
    _id: new ObjectId('57d3cf0fa662a440760af4a7')
  },
  isDeleted: false,
  createdAt: 2025-03-14T06:39:11.864Z,
  updatedAt: 2025-03-14T06:39:11.864Z,
  __v: 0
}
```

Hình 4.43 Giao diện Terminal khi chạy dự án thành công

Truy cập vào <http://localhost:3000> để truy cập vào hệ thống với tên tài khoản và mật khẩu đăng nhập được hiển thị trong terminal. Khi này, mọi thay đổi trong mã nguồn sẽ được cập nhật ngay lập tức trên trình duyệt.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Hệ thống “Quản lý thiết bị và quà tặng của Trung tâm Ngoại ngữ - Tin học Victory” đã được xây dựng thành công và đáp ứng được các nhu cầu cần thiết như:

- Chức năng quản lý người dùng
- Chức năng quản lý thiết bị
- Chức năng quản lý mượn trả thiết bị
- Chức năng quản lý quà tặng
- Chức năng quản lý đơn yêu cầu quà tặng
- Chức năng thống kê

Hệ thống chạy ổn định, xử lý các yêu cầu nhanh chóng, giao diện thân thiện và dễ sử dụng, dễ mở rộng.

5.2. Hướng phát triển

Với hướng phát triển, có thể mở rộng các chức năng theo chiều sâu (tối ưu hơn các tính năng hiện có) và theo chiều ngang (phát triển thêm các tính năng liên quan):

- *Theo chiều sâu:*

+ Cải thiện hiệu suất của hệ thống để xử lý dữ liệu lớn hơn. Cải thiện các chức năng như quản lý lịch sử đăng nhập, quản lý kho,...

+ Cải thiện khả năng bảo mật của hệ thống.

- *Theo chiều ngang:*

+ Tích hợp với các hệ thống quản lý khác của Trung tâm như quản lý lịch dạy học, quản lý lịch thi cử, quản lý đề thi,...

+ Phát triển thêm các tính năng hỗ trợ cho việc quản lý thiết bị trong các phòng làm việc, quà tặng.

+ Bổ sung tính năng chat để việc trao đổi giữa các người dùng với nhau được dễ dàng, thuận tiện hơn.

+ Tính năng hỗ trợ với AI có thể được chú ý phát triển để giúp đưa ra các hỗ trợ hữu ích một cách nhanh chóng ngay từ giai đoạn ban đầu.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] bootstrap, "getbootstrap.com," bootstrap document, 2025. [Online]. Available: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>.
- [2] Expressjs, "Expressjs.com," Express.js API Reference, 2024. [Online]. Available: [https://expressjs.com/en/5x/api.html..](https://expressjs.com/en/5x/api.html)
- [3] MISA, "https://emis.misa.vn/" Ứng dụng công nghệ thông tin trong quản lý thiết bị dạy học, 2025. [Online]. Available: https://emis.misa.vn/qlth/documents/Thiet_Bi.pdf.
- [4] MongoDB, "MongoDB.com," MongoDB Documentation, 2024. [Online]. Available: <https://www.mongodb.com/docs/..>
- [5] Nodejs, "Nodejs.org," Node.js documentation, 2024. [Online]. Available: <https://nodejs.org/docs/latest/api/documentation.html..>
- [6] npmjs, "https://www.npmjs.com/", crypto-js, 2024. [Online]. Available: <https://www.npmjs.com/package/crypto-js>.
- [7] npmjs, "npmjs.com," Express Handlebars, 2024. [Online]. Available: <https://www.npmjs.com/package/express-handlebars>.
- [8] npmjs, "npmjs.com," sass, 2024. [Online]. Available: <https://www.npmjs.com/package/sass>.
- [9] ViinDo, "https://viindoo.com/" 5 Phần mềm quản lý thiết bị trường học hiệu quả, chính xác, 2023. [Online]. Available: <https://viindoo.com/vi/blog/quan-tri-doanh-nghiep-3/phan-mem-quan-ly-thiet-bi-truong-hoc-613>.
- [10] pham.tien.khanh, "Gửi thư với NodeMailer," 2020. [Online]. Available: <https://viblo.asia/p/gui-thu-voi-nodemailer-Az45baD6lxY>.
- [11] MongoDb, "MongoDB Compass là gì? Hướng dẫn sử dụng," websitehcm.com, 2022. [Online]. Available: <https://websitehcm.com/mongodb-compass-la-gi-huong-dan-su-dung/>.
- [12] N. L. H. Thương, "MongoDB là gì? 9 Phần mềm quản trị Mongodbs nên sử dụng," prodima.vn, [Online]. Available: <https://prodima.vn/mongodb-la-gi/#:~:text=MongoDB%20Compass%20cung%20c%E1%BA%A5p%20ch%E1%BBA9c,b%E1%BB%99%20d%E1%BB%AF%20li%E1%BB%87u%20tr%E1%BB%B1c%20quan..>

PHỤ LỤC 1: CÁC CÔNG CỤ, PHẦN MỀM HỖ TRỢ XÂY DỰNG TRANG WEB

1. Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí và đa nền tảng, được phát triển bởi Microsoft. Được ra mắt lần đầu vào năm 2015, VS Code nhanh chóng trở thành một trong những công cụ phát triển phổ biến nhất trong cộng đồng lập trình viên nhờ vào sự linh hoạt, hiệu suất cao và khả năng tùy biến mạnh mẽ. Trình soạn thảo này hỗ trợ nhiều ngôn ngữ lập trình khác nhau như JavaScript, Python, C++, Java, HTML/CSS và nhiều ngôn ngữ khác thông qua hệ thống mở rộng phong phú.

Một trong những điểm mạnh lớn nhất của Visual Studio Code là sự tích hợp liền mạch với các hệ thống quản lý mã nguồn như Git, cho phép các nhà phát triển dễ dàng quản lý phiên bản, thực hiện commit, push và pull mà không cần rời khỏi trình soạn thảo.

Visual Studio Code có kho tiện ích mở rộng (extensions) phong phú, cho phép người dùng tùy chỉnh và mở rộng chức năng của trình soạn thảo theo nhu cầu. Người dùng có thể cài đặt các tiện ích hỗ trợ từ đơn giản như đổi màu giao diện, đến phức tạp hơn như tích hợp các công cụ phân tích mã, debugger hay hỗ trợ lập trình với framework cụ thể. VS Code cũng được tối ưu hóa để chạy mượt mà trên cả các máy tính có cấu hình thấp, làm cho nó trở thành lựa chọn lý tưởng cho cả những người mới học lập trình và các chuyên gia trong ngành.

Ngoài ra, với sự hỗ trợ mạnh mẽ từ cộng đồng, Visual Studio Code luôn được cập nhật và phát triển không ngừng giúp nó thích nghi với sự phát triển nhanh chóng của các công nghệ lập trình hiện đại. VS Code là sự kết hợp hoàn hảo giữa một trình soạn thảo mã nhẹ nhàng nhưng đầy đủ tính năng của một môi trường phát triển tích hợp (IDE), tạo điều kiện tối ưu cho việc phát triển phần mềm trên nhiều nền tảng khác nhau.

2. MongoDB Compass

MongoDB Compass là một GUI cho MongoDB. Nó còn được gọi là MongoDB GUI. MongoDB cho phép người dùng phân tích nội dung dữ liệu được lưu trữ của họ mà không cần biết trước về cú pháp truy vấn MongoDB. Khi tìm hiểu khám phá dữ liệu của mình trong

môi trường trực quan, có thể sử dụng Compass GUI để tối ưu hóa hiệu suất, quản lý chỉ mục và triển khai xác thực Document.

MongoDB Compass cung cấp chức năng đồ họa mạnh mẽ cho người dùng mà không cần ngôn ngữ truy vấn. Phần mềm này cũng hỗ trợ hiển thị cấu trúc và phân tích các tài liệu bên trong GUI trực quan chỉ khoảng vài giây. MongoDB compass có 1 số đặc điểm:

Người dùng dễ dàng khám phá toàn bộ dữ liệu trực quan.

Cho phép xem hiệu suất truy vấn.

Cung cấp nhanh thông tin chi tiết về hiệu năng truy vấn và tình trạng server.

Không cần phải viết dòng lệnh.

Tăng hiệu quả tiếp cận và tương tác tốt hơn với CRUD.

Giúp người dùng xem xét và đưa ra quyết định index, xác thực tài liệu...

3. Postman

Postman là một công cụ mạnh mẽ và phổ biến được sử dụng để phát triển, kiểm thử và quản lý các API (Application Programming Interface). Được phát hành lần đầu vào năm 2012, Postman đã trở thành công cụ không thể thiếu đối với các lập trình viên và tester trong quá trình xây dựng và phát triển phần mềm dựa trên API. Với giao diện đồ họa trực quan và dễ sử dụng, Postman cho phép người dùng tạo, gửi và phân tích các yêu cầu HTTP/HTTPS một cách nhanh chóng mà không cần phải viết mã hay dùng các công cụ dòng lệnh phức tạp.

Một trong những tính năng chính của Postman là hỗ trợ nhiều phương thức yêu cầu HTTP khác nhau như GET, POST, PUT, DELETE, PATCH và nhiều phương thức khác. Điều này cho phép các lập trình viên mô phỏng và kiểm thử các yêu cầu đến API một cách chính xác, cũng như kiểm tra cách thức API phản hồi với các dữ liệu đầu vào khác nhau. Giao diện của Postman cung cấp cho người dùng khả năng cấu hình chi tiết yêu cầu với các tham số, tiêu đề (headers), nội dung thân (body), cũng như quản lý cookies và xác thực người dùng.

Postman không chỉ dừng lại ở việc gửi yêu cầu và nhận phản hồi, nó còn cung cấp nhiều tính năng nâng cao giúp tối ưu hóa quy trình phát triển API. Một trong số đó là Postman Collections, cho phép người dùng tổ chức và lưu trữ các nhóm yêu cầu liên quan vào một tập hợp để tái sử dụng hoặc chia sẻ với đội nhóm. Postman cũng hỗ trợ environment variables

(biến môi trường), cho phép người dùng tạo các môi trường khác nhau (như phát triển, thử nghiệm và sản xuất) với các giá trị biến cù thể giúp dễ dàng chuyển đổi giữa các cấu hình khác nhau khi kiểm thử.

Postman còn có khả năng tự động hóa kiểm thử API. Người dùng có thể viết các đoạn mã kiểm thử bằng JavaScript trong Postman để tự động xác nhận phản hồi API có đúng như mong đợi hay không, từ đó giúp đảm bảo chất lượng API trong suốt quá trình phát triển. Tính năng này rất hữu ích cho việc thực hiện kiểm thử hồi quy và kiểm tra tính toàn vẹn của API sau mỗi lần cập nhật.

Ngoài ra, Postman hỗ trợ tích hợp CI/CD (Continuous Integration/Continuous Deployment) giúp tự động hóa quá trình kiểm thử API và triển khai trên các môi trường khác nhau một cách liền mạch. Điều này giúp cải thiện hiệu suất làm việc và đảm bảo các API luôn hoạt động đúng đắn trước khi triển khai lên môi trường sản xuất.

Với cộng đồng người dùng rộng lớn và tài liệu hỗ trợ phong phú, Postman không chỉ là công cụ hỗ trợ phát triển và kiểm thử API mà còn là một nền tảng hợp tác mạnh mẽ, cho phép các đội phát triển làm việc cùng nhau hiệu quả hơn trong quá trình xây dựng các giải pháp dựa trên API. Với những tính năng vượt trội và tính thân thiện với người dùng, Postman đã trở thành lựa chọn hàng đầu cho các nhà phát triển API trên toàn thế giới.