

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH

HỌC KỲ I, NĂM HỌC 2024-2025

XÂY DỰNG KHOÁ HỌC ONLINE BẰNG NODEJS & EXPRESS

Giáo viên hướng dẫn:

TS. Nguyễn Bảo Ân

Sinh viên thực hiện:

Họ tên: Lâm Huệ Trung

MSSV: 110121255

Lớp: DA21TTC

Trà Vinh, tháng..... năm.....

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

LỜI CẢM ƠN

Kính gửi: thầy Nguyễn Bảo Ân,

Trước hết, cho phép em được gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Bảo Ân đã tận tình giảng dạy, hướng dẫn và giúp đỡ em trong suốt quá trình thực hiện đồ án cơ sở ngành.

Thầy đã dành thời gian quý báu của mình để chia sẻ những kiến thức bổ ích, giúp em hoàn thành đồ án một cách tốt nhất. Thầy cũng đã luôn động viên, khích lệ em trong suốt quá trình thực hiện đồ án, giúp em vượt qua những khó khăn và hoàn thành đồ án đúng tiến độ.

Em cũng xin gửi lời cảm ơn tất cả các bạn đã giúp đỡ và hỗ trợ em khi gặp những trở ngại và khó khăn trong quá trình thực hiện đồ án.

Em xin hứa sẽ tiếp tục nỗ lực học tập và phấn đấu.

Trân trọng,

Lâm Huệ Trung

MỤC LỤC

CHƯƠNG 1 TỔNG QUAN.....	11
1.1 Mô tả bài toán	11
1.2 Công cụ và cách thức thực hiện.....	12
1.3 Xác định mô hình dữ liệu	14
1.4 Các thông tin cần lưu trữ trong đề tài bao gồm	14
1.4.1 Thông tin tài khoản và người dùng.....	14
1.4.2 Thông tin khoá học	15
1.4.3 Thông tin chương học	16
1.4.4 Thông tin bài học	16
1.4.5 Thông tin đăng ký khoá học	17
1.4.6 Thông tin tiến độ học	17
1.4.7 Thông tin lượt truy cập	17
1.5 Tương tác dữ liệu bằng NodeJs & Express và MongoDB	17
CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT	19
2.1 Mô hình MVC	19
2.1.1 Giới thiệu	19
2.1.2 Thành phần trong mô hình MVC.....	19
2.1.3 Luồng xử lý trong MVC	20
2.1.4 Lý do chọn mô hình MVC.....	21
2.2 CQRS pattern.....	23
2.2.1 Giới thiệu	23
2.2.2 Command và Query	23
2.2.3 Lý do sử dụng CQRS	23
2.3 JWT	24
2.3.1 Giới thiệu	24
2.3.2 Thành phần của JWT	24
2.4 Tổng quan về Bootstrap Framework	25
2.4.1 Cài đặt Bootstrap	25
2.4.2 Các thành phần chính của Bootstrap.....	26
2.5 Xây dựng chức năng bằng NodeJs & framework Express.....	29
2.5.1 Nền tảng NodeJs	29
2.5.2 Cài đặt các thư viện hỗ trợ	31
2.5.3 Chạy ứng dụng Node.js	32

2.5.4	Tài liệu Node.js.....	32
2.5.5	Kiến thức về framework Express.js	34
2.6	Hệ quản trị cơ sở dữ liệu MongoDB	35
CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU		37
3.1	Mô tả website.....	37
3.2	Đặc tả các chức năng hệ thống	37
3.2.1	Chức năng của sinh viên.....	37
3.2.2	Chức năng của leader.....	37
3.2.3	Chức năng của quản trị viên	38
3.2.4	Quản lý người dùng	38
3.3	Thiết kế dữ liệu.....	38
3.3.1	Danh sách thực thể.....	38
3.3.2	Mô hình dữ liệu.....	40
3.4	Kiến trúc hệ thống	40
3.5	Thiết kế giao diện	41
3.5.1	Sơ đồ website	41
3.5.2	Giao diện trang.....	41
CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU		48
4.1	Sản phẩm đạt được sau quá trình nghiên cứu.....	48
4.2	Về hiệu năng	48
4.3	Về trải nghiệm người dùng	48
4.4	Kết quả thử nghiệm	48
4.4.1	Chức năng người dùng.....	49
4.4.2	Chức năng người quản trị	52
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		62
5.1	Kết luận.....	62
5.1.1	Kết quả đạt được	62
5.1.2	Những đóng góp mới	62
5.2	Hướng phát triển	62
DANH MỤC TÀI LIỆU THAM KHẢO		63

DANH SÁCH HÌNH ẢNH VÀ BẢNG BIÊU

Hình 2. 1 Các thành phần của mô hình MVC	19
Hình 2. 2 Luồng xử lý trong MVC	21
Hình 2. 3 Command và querry trong CQRS Pattern.....	23
Hình 2. 4 Mô hình MVC	29
Hình 3. 1 Sơ đồ website người quản trị	41
Hình 3. 2 Sơ đồ website người dùng.....	41
Hình 3. 3 Giao diện trang đăng ký	42
Hình 3. 4 Giao diện trang đăng nhập của người dùng	42
Hình 3. 5 Giao diện trang chủ người dùng.....	43
Hình 3. 6 Giao diện đăng ký khoá học.....	43
Hình 3. 7 Giao diện học khoá học	44
Hình 3. 8 Giao diện bài viết	44
Hình 3. 9 Giao diện lộ trình học.....	45
Hình 3. 10 Giao diện đăng nhập trang quản trị	45
Hình 3. 11 Giao diện chính trang quản trị.....	46
Hình 3. 12 Giao diện thêm và sửa thông tin	46
Hình 3. 13 Giao diện danh sách hiển thị	47
Hình 3. 14 Giao diện thống kê	47
Hình 4. 1 Giao diện trang chủ người dùng.....	49
Hình 4. 2 Trang đăng ký	49
Hình 4. 3 Trang đăng nhập.....	50
Hình 4. 4 Trang xem thông tin khoá học và đăng ký.....	50
Hình 4. 5 Trang tham gia học.....	51
Hình 4. 6 Trang lộ trình học tập	51
Hình 4. 7 Trang bài viết	52
Hình 4. 8 Trang đăng nhập quản trị viên	52
Hình 4. 9 Trang quản trị	53
Hình 4. 10 Trang thêm người dùng.....	53
Hình 4. 11 Trang danh sách người dùng	54
Hình 4. 12 Trang cập nhật thông tin người dùng	54
Hình 4. 13 Giao diện xem thông tin người dùng	55
Hình 4. 14 Trang thêm khoá học.....	55
Hình 4. 15 Trang danh sách khoá học	56

Hình 4. 16 Trang thông tin chi tiết khoá học	56
Hình 4. 17 Trang cập nhật thông tin khoá học	57
Hình 4. 18 Trang danh sách chương học	57
Hình 4. 19 Thêm chương học.....	58
Hình 4. 20 Trang cập nhật chương học	58
Hình 4. 21 Danh sách bài học thuộc chương	59
Hình 4. 22 Giao diện thêm bài học	59
Hình 4. 23 Trang cập nhật bài học	60
Hình 4. 24 Trang thông tin bài học	60
Hình 4. 25 Trang thống kê người dùng	61
Hình 4. 26 Trang thống kê khoá học.....	61
Bảng 3.1 Danh sách thực thể.....	39

TÓM TẮT ĐỀ ÁN CƠ SỞ NGÀNH

Đề tài: Xây dựng website khoá học online bằng NodeJs & Express.

Mục tiêu: Xây dựng website khoá học online bằng NodeJs & Express theo hình thức self-learning. Sử dụng nền tảng NodeJs, framework Express cùng với Handlebar, Bootstrap và cơ sở dữ liệu MongoDB xây dựng website theo mô hình MVC.

Hướng tiếp cận: Nghiên cứu lý thuyết: Tìm hiểu các tài liệu, nghiên cứu trước đây về Node.js & Express.js, Handlerbar & Bootstrap, MongoDB và các website khoá học hiện đang có trên internet. Nghiên cứu thực nghiệm: Xây dựng website khoá học online bằng NodeJs & Express.

Cách giải quyết vấn đề: NodeJs là một nền tảng phát triển ứng dụng web và mobile hiệu quả, linh hoạt và dễ sử dụng. Express là một framework của NodeJs giúp phát triển ứng dụng web back-end một cách nhanh chóng và dễ dàng. Handlebars là một công cụ phía máy khách (mặc dù nó cũng có thể được sử dụng trên máy chủ) công cụ tạo khuôn template cho JavaScript. Bootstrap là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,... MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL hiệu quả và linh hoạt.

Kết quả đạt được: Website khoá học self-learning được xây dựng thành công, đáp ứng được các yêu cầu đề ra. Ứng dụng có giao diện thân thiện, dễ sử dụng. Ứng dụng có khả năng mở rộng, dễ dàng tích hợp với các hệ thống khác.

Bài học kinh nghiệm: Cần tìm hiểu kỹ các tài liệu, nghiên cứu trước đây về đề tài nghiên cứu. Cần lập kế hoạch thực hiện chi tiết và khoa học. Cần kiên trì và nỗ lực để hoàn thành đề tài.

MỞ ĐẦU

Việc tự học trong môi trường học rất quan trọng đối với các ngành nói chung và ngành công nghệ thông tin nói riêng. Kỹ năng tự học sẽ giúp cho sinh viên có thêm nhiều kiến thức về các khía cạnh mà giáo viên đề cập đến. Trên internet có rất nhiều website tự học như F8, CodeLearn, Coursera, AWS, youtube và nhiều website khác hướng dẫn lập trình. Vì thế, tự học lập trình online qua nền tảng internet đã không còn xa lạ với sinh viên.

Do đó, việc nghiên cứu và xây dựng website khoá học self-learning online là một đề tài khả thi và có tính thực tế cao. Website khoá học self-learning online có thể giúp cho sinh viên học tập một cách hiệu quả, tiết kiệm thời gian và công sức, đồng thời giúp sinh viên nâng cao kiến thức và khả năng về lập trình.

Mục đích của đề tài là xây dựng website khoá học self-learning online bằng NodeJs & Express, Handlerbar & Bootstrap, MongoDB theo mô hình MVC.

Đối tượng nghiên cứu của đề tài NodeJS & Express, Handlerbar & Bootstrap, MongoDB và các website self-learning hiện đang được nhiều sinh viên tin dùng.

Phạm vi nghiên cứu của đề tài bao gồm:

- Nghiên cứu về NodeJs & Express, Handlerbar & Bootstrap, MongoDB.
- Phân tích yêu cầu, thiết kế hệ thống, thiết kế giao diện và chức năng website khoá học self-learning online.
- Xây dựng website khoá học self-learning online.
- Kiểm website khoá học self-learning online.

CHƯƠNG 1 TỔNG QUAN

1.1 Mô tả bài toán

Việc tự học trong môi trường học rất quan trọng đối với các ngành nói chung và ngành công nghệ thông tin nói riêng. Kỹ năng tự học sẽ giúp cho sinh viên có thêm nhiều kiến thức về các khía cạnh mà giáo viên đề cập đến. Vì thế, sinh viên đang có xu thế tìm kiếm các trang website để học về lập trình. Để đáp ứng nhu cầu này, việc xây dựng website khoá học self-learning online là một giải pháp hiệu quả.

Website khoá học self-learning online có thể giúp cho sinh viên học tập một cách hiệu quả, tiết kiệm thời gian và công sức, đồng thời giúp sinh viên nâng cao kiến thức và khả năng về lập trình. website khoá học self-learning online bằng NodeJs & Express, Handlerbar & Bootstrap, MongoDB theo mô hình MVC.

Website sẽ có các chức năng chính sau:

Quản lý người dùng sử dụng website.

Quản lý khoá học, chương và bài học có trên website.

Thống kê lượt truy cập của người dùng.

Thống kê khoá học.

Tra cứu khoá học và xem thông tin khoá học.

Đăng ký khoá học.

Đánh giá tiến độ học của người dùng.

Đăng nhập bằng Google.

Việc xây dựng website khoá học self-learning online có thể mang lại nhiều lợi ích cho nhà trường, đặc biệt là khoa kỹ thuật và công nghệ, bao gồm:

Tạo nền tảng tự học cho sinh viên.

Nâng cao kiến thức, kỹ năng lập trình của sinh viên.

Tiết kiệm thời gian và nguồn lực.

1.2 Công cụ và cách thức thực hiện

Để thực hiện đề tài xây dựng website khoá học online bằng NodeJs & Express, cần tập trung nghiên cứu các kiến thức về nền tảng NodeJs, framework Express, Handlerbar và Bootstrap và hệ quản trị cơ sở dữ liệu MongoDB, cấu trúc dữ liệu, thuật toán, kỹ thuật lập trình web, các khoá học, các yêu cầu chức năng và giao diện của website. Cụ thể, cần nghiên cứu các nội dung sau:

NodeJs: NodeJs là một nền tảng JavaScript chạy trên máy chủ, được sử dụng để phát triển các ứng dụng web và ứng dụng dịch vụ web. NodeJs sử dụng mô hình event-driven, trong đó các sự kiện được xử lý một cách không đồng bộ. Điều này giúp NodeJs có thể xử lý nhiều yêu cầu cùng lúc một cách hiệu quả. NodeJs là một lựa chọn phù hợp cho việc phát triển chức năng của ứng dụng quản lý thực tập vì nó có khả năng xử lý các yêu cầu đồng thời và có thể mở rộng dễ dàng.

Express: Express là một framework web dựa trên NodeJs giúp phát triển các ứng dụng web back-end một cách nhanh chóng và dễ dàng. Express cung cấp nhiều tính năng hữu ích cho việc phát triển các ứng dụng web back-end, bao gồm routing, middleware, session management,... cung cấp các tính năng và chức năng cần thiết để phát triển các ứng dụng web RESTful. Express sẽ được sử dụng để xây dựng giao diện API cho ứng dụng quản lý thực tập.

Handlebars được viết bằng JavaScript, Handlebars là trình biên dịch nhận bất kỳ biểu thức HTML và biên dịch chúng thành hàm JavaScript. Hàm JavaScript này nhận một tham số, một đối tượng, dữ liệu từ server và trả về một chuỗi có các giá trị HTML và thuộc tính của đối tượng được chèn vào HTML.

Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế responsive dễ dàng, thuận tiện và nhanh chóng hơn.

MongoDB: MongoDB là một cơ sở dữ liệu NoSQL phân tán, được sử dụng để lưu trữ dữ liệu của ứng dụng. Mongodbs sử dụng mô hình document-based, trong đó dữ liệu được lưu trữ dưới dạng các document. Điều này giúp Mongodbs có thể lưu trữ các loại dữ liệu khác nhau một cách linh hoạt. Mongodbs sẽ được sử dụng để

lưu trữ thông tin về sinh viên thực tập, giáo viên hướng dẫn và thông tin chương trình thực tập.

Ưu điểm của việc sử dụng các công nghệ này:

NodeJs: NodeJs có khả năng xử lý các yêu cầu đồng thời và có thể mở rộng dễ dàng, phù hợp với việc phát triển website khoá học có số lượng sinh viên lớn.

Express: Express cung cấp các tính năng và chức năng cần thiết để phát triển các ứng dụng web RESTful, giúp việc phát triển website khoá học trở nên dễ dàng và hiệu quả hơn.

MongoDB: MongoDB là một cơ sở dữ liệu NoSQL phân tán, phù hợp với việc lưu trữ dữ liệu của website khoá học, giúp website có khả năng mở rộng dễ dàng.

Cấu trúc dữ liệu: Đây là một phần quan trọng của khoa học máy tính, liên quan đến việc lưu trữ và tổ chức dữ liệu. Cấu trúc dữ liệu cần được nghiên cứu kỹ lưỡng, bao gồm các kiến thức về các loại cấu trúc dữ liệu, các ưu điểm và nhược điểm của từng loại cấu trúc dữ liệu,...

Thuật toán: Đây là một tập hợp các quy tắc được sử dụng để giải quyết một vấn đề. Thuật toán cần được nghiên cứu kỹ lưỡng, bao gồm các kiến thức về các loại thuật toán, các ưu điểm và nhược điểm của từng loại thuật toán,...

Kỹ thuật lập trình web: Đây là một tập hợp các kỹ năng và kiến thức cần thiết để phát triển các ứng dụng web. Kỹ thuật lập trình web cần được nghiên cứu kỹ lưỡng, bao gồm các kiến thức về HTML, CSS, JavaScript, NodeJs & Express, Handlerbar, Bootstrap, MongoDB,...

Ngoài ra, cần nghiên cứu thêm các kiến thức liên quan đến đề tài như:

Chức năng cần có của website khoá học self-learning: Website sẽ có các chức năng chính sau: Quản lý người dùng sử dụng website, quản lý khoá học, chương và bài học có trên website, thống kê lượt truy cập của người dùng, thống kê khoá học, tra cứu khoá học và xem thông tin khoá học, đăng ký khoá học, đánh giá tiến độ học của người dùng, đăng nhập bằng Google.

Mô hình dữ liệu thường được sử dụng trong hệ thống quản lý sinh viên thực tập : Mô hình quan hệ, mô hình NoSQL.

Biện pháp bảo mật cần được thực hiện: Sử dụng thuật toán mã hóa, kiểm soát truy cập, thiết lập tường lửa.

Triển khai và vận hành hệ thống: Cài đặt hệ thống, thiết lập cấu hình hệ thống, tạo tài khoản cho người dùng, đào tạo người dùng và đào tạo người quản trị.

Quản lý hệ thống: Theo dõi tình trạng hoạt động của hệ thống, tạo bản sao lưu dữ liệu, cập nhật hệ thống.

Yêu cầu về giao diện của website: Cần nghiên cứu các yêu cầu về giao diện của website, để có thể thiết kế một website có giao diện đẹp mắt và dễ sử dụng.

Việc nghiên cứu kỹ các kiến thức này sẽ giúp cho việc thực hiện đề tài được thuận lợi và đạt kết quả tốt.

1.3 Xác định mô hình dữ liệu

Đề tài xây dựng website khoá học self-learning online cần lưu trữ các thông tin về tài khoản, thông tin người dùng, khoá học, chương học, bài học, thông tin đăng ký, đánh giá tiến độ. Các thông tin này cần được lưu trữ để phục vụ cho các chức năng chính của website, bao gồm:

Quản lý tài khoản: Thêm, cập nhật, xoá thông tin tài khoản.

Quản lý khoá học: Thêm, cập nhật, xoá, tìm kiếm khoá học.

Quản lý chương học: Thêm, cập nhật, xoá, tìm kiếm chương học.

Quản lý bài học: Thêm, cập nhật, xoá, tìm kiếm bài học.

Sinh viên đăng ký khoá học.

Đánh giá tiến độ học của sinh viên.

Thống kê người dùng và khoá học.

1.4 Các thông tin cần lưu trữ trong đề tài bao gồm

1.4.1 Thông tin tài khoản và người dùng

UserName: Là tài khoản người dùng dùng để đăng nhập vào.

Password: Là mật khẩu để người dùng đăng nhập vào.

Role: Là quyền truy cập của tài khoản, gồm có 3 quyền: System admin, Sub admin và User.

GoogleId: Nếu đăng nhập bằng tài khoản google sẽ lưu id lại.

isDeleted: Là trạng thái tài khoản gồm có đang hoạt động và bị vô hiệu hóa.

FullName: Là họ tên của người dùng.

BirthDate: Là ngày tháng năm sinh của người dùng.

Specialty: Là chuyên ngành hoặc chuyên môn của người dùng.

Avatar: Là đường dẫn đến file ảnh đại diện của người dùng.

Address: Là địa chỉ của người dùng.

Phone: Là số điện thoại của người dùng.

Timestamps: Gồm có CreatedAt (thời gian tạo), UpdatedAt (thời gian sửa).

1.4.2 Thông tin khoá học

Name: Là tên khoá học.

Description: Là mô tả về khoá học.

Benefits: Là các lợi ích sau khi học khoá học.

Level: Là cấp độ của khoá học, gồm có Beginner, Intermediate, Advanced.

Price: giá của khoá học, nếu giá bằng 0 thì khoá học miễn phí.

Author: Là tác giả tạo khoá học, có thể là System admin hoặc Sub admin.

Image: Là hình ảnh tượng trưng của khoá học.

TotalLessons: Là tổng số bài học của khoá học.

TotalChapters: Là tổng số chương học của khoá học.

Chapters: Là các chương của khoá học.

Category: Là lĩnh vực của khoá học.

Duration: Là tổng thời gian của khoá học.

isDeleted: Là trạng thái khoá học gồm có đang hoạt động và bị vô hiệu hoá.

Timestamps: Gồm có CreatedAt (thời gian tạo), UpdatedAt (thời gian sửa).

1.4.3 Thông tin chương học

CourseId: Là id khoá học của chương học.

Title: Là tên của chương học.

Lessons: Là các bài học thuộc chương học.

ChapterOrder: Là thứ tự của chương học.

isDeleted: Là trạng thái chương học gồm có đang hoạt động, bị vô hiệu hoá.

Timestamps: Gồm có CreatedAt (thời gian tạo), UpdatedAt (thời gian sửa).

1.4.4 Thông tin bài học

ChapterId: Là id chương của bài học.

Name: Là tên bài học.

Description: Là mô tả của bài học.

VideoUrl: Là đường dẫn của video bài học được nhúng từ youtube.

Duration: Là thời gian video.

LessonsOrder: Là thứ tự của bài học.

isDeleted: Là trạng thái bài học gồm có đang hoạt động và bị vô hiệu hoá.

Timestamps: Gồm có CreatedAt (thời gian tạo), UpdatedAt (thời gian sửa).

1.4.5 Thông tin đăng ký khoá học

UserId: Là id người dùng đăng ký khoá học.

CourseId: Là id khoá học mà người dùng đăng ký.

RegistrationDate: Là ngày đăng ký, tự động lấy giá trị ngày giờ.

1.4.6 Thông tin tiến độ học

RegistrationId: Là id đăng ký khoá học.

Chapter: Là object chứa chapterId, Lessons (gồm có LessonId và status).

progress: là phần trăm hoàn thành bài học.

OverallProgress: Là phần trăm hoàn thành khoá học.

Timestamps: Gồm có CreatedAt(thời gian tạo) và UpdatedAt(thời gian sửa).

1.4.7 Thông tin lượt truy cập

Route: Là đường dẫn mà người dùng truy cập.

Count: Là số lượng lượt truy cập.

UserId: Là id người dùng truy cập vào.

1.5 Tương tác dữ liệu bằng NodeJs & Express và MongoDB

Tương tác dữ liệu giữa Node.Js & Express và MongoDB được thực hiện thông qua các câu lệnh truy vấn MongoDB. Các câu lệnh truy vấn MongoDB được sử dụng để truy vấn dữ liệu từ cơ sở dữ liệu và trả về một chuỗi json.

Ví dụ, để truy vấn tất cả các sinh viên từ bảng tblSinhViens, Node.js & Express sẽ sử dụng câu lệnh truy vấn MongoDB sau:

```
static layDanhSachSinhVien(req, res) {  
    SinhVien.find()  
        .then((sinhvien) => res.json(sinhvien))  
        .catch((err) => res.json('Lỗi /student: ' + err));  
}
```

Kết quả của câu lệnh truy vấn này sẽ là một mảng các đối tượng sinh viên theo kiểu json.

Để thêm một sinh viên mới vào cơ sở dữ liệu, NodeJs & Express sẽ sử dụng câu lệnh truy vấn MongoDB sau:

```
static themThongTinSinhVien(req, res) {  
    const newData = req.body;  
  
    SinhVien.create(newData)  
        .then((result) => {  
            res.status(201).json(result);  
        })  
        .catch((err) => {  
            res.status(500).json({ error: err });  
        });  
}
```

Câu lệnh truy vấn này sẽ thêm một sinh viên mới vào bảng tblSinhViens với các thông tin được cung cấp trong các giá trị của câu lệnh.

Tương tự, NodeJs & Express có thể sử dụng các câu lệnh truy vấn MongoDB khác để cập nhật, xóa hoặc sửa đổi dữ liệu trong cơ sở dữ liệu MongoDB.

Để thực hiện tương tác dữ liệu giữa NodeJs & Express và MongoDB, cần cài đặt các thư viện mongodb để giao tiếp với cơ sở dữ liệu MongoDB.

Sau khi cài đặt các thư viện này, có thể sử dụng các hàm sau để thực hiện tương tác dữ liệu:

Hàm connect() để kết nối với cơ sở dữ liệu MongoDB.

Hàm find() để truy vấn dữ liệu.

Hàm save() để thêm một bản ghi mới.

Hàm findByIdAndUpdate() để cập nhật một bản ghi.

Hàm findByIdAndDelete () để xóa một bản ghi.

CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT

2.1 Mô hình MVC

2.1.1 Giới thiệu

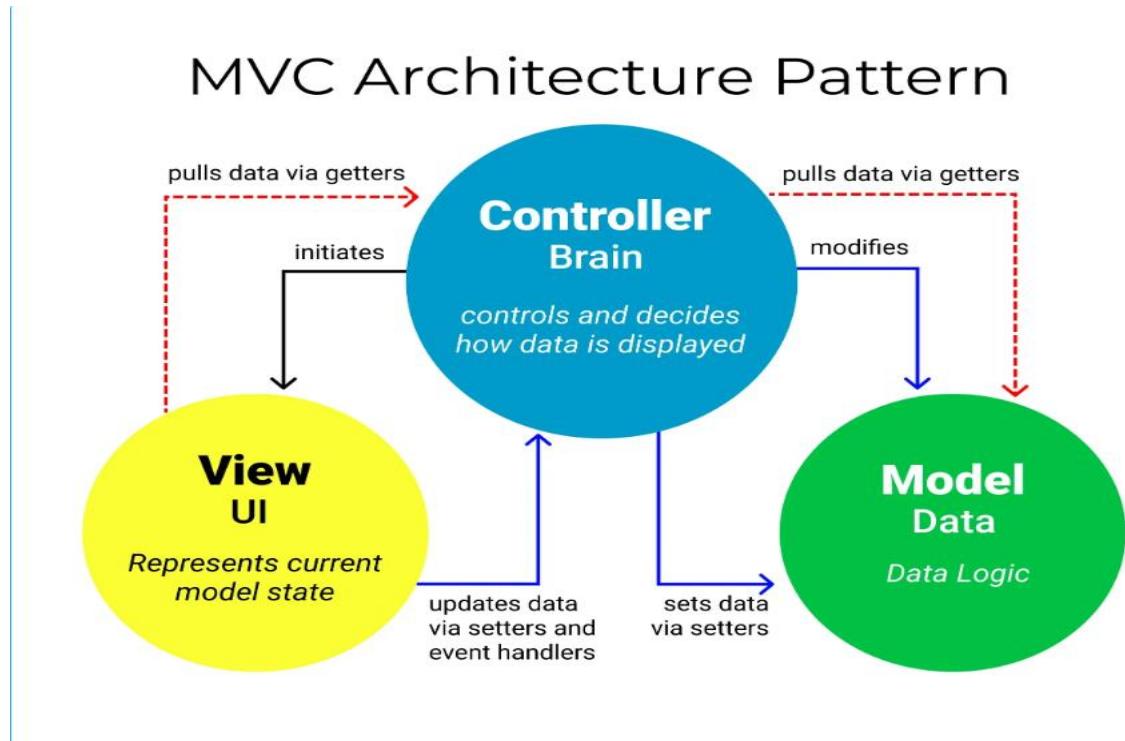
MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong việc phát triển ứng dụng để tạo ra các giao diện người dùng trực quan và có khả năng tương tác cao. MVC chia một ứng dụng thành 3 phần chính và mỗi phần có một vai trò riêng biệt:

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 một cách 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.

Mỗi 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

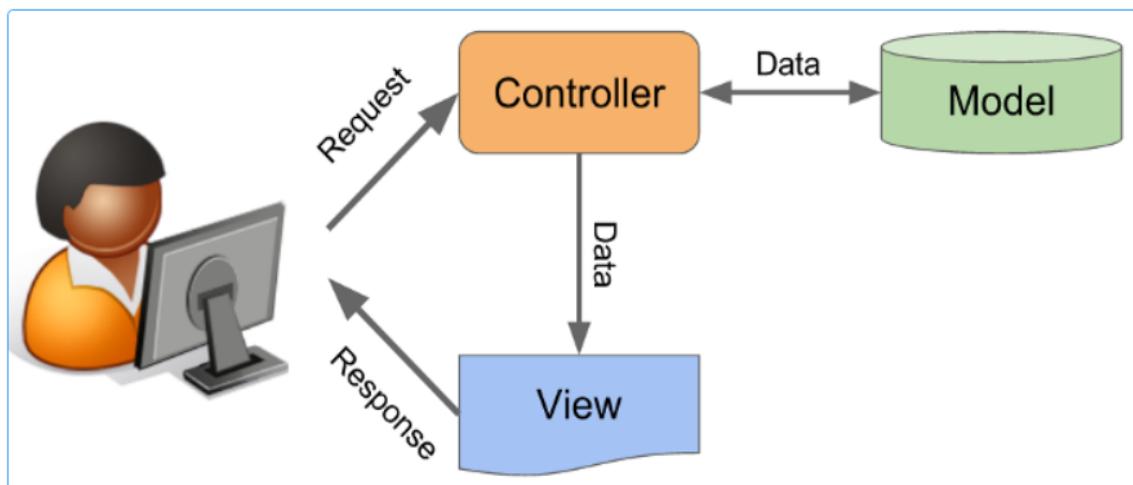
Luồng xử lý trong của mô hình MVC, có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

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.

Hơn nữa, 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.

Hỗ trợ kỹ thuật Asynchronous

Kiến trúc MVC có thể được tích hợp với cả JavaScript Framework. Có nghĩa là, các ứng dụng MVC có thể hoạt động ngay cả với các file PDF, trình duyệt riêng cho web hay các widget trên desktop.

Ngoài ra, MVC cũng hỗ trợ kỹ thuật Asynchronous, giúp các developer phát triển các ứng dụng có thể load rất nhanh.

Nền tảng MVC thân thiện với SEO

Nền tảng MVC hỗ trợ phát triển các trang web thân thiện với SEO. Bằng nền tảng này, có thể dễ dàng phát triển các URL thân thiện với SEO để 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.

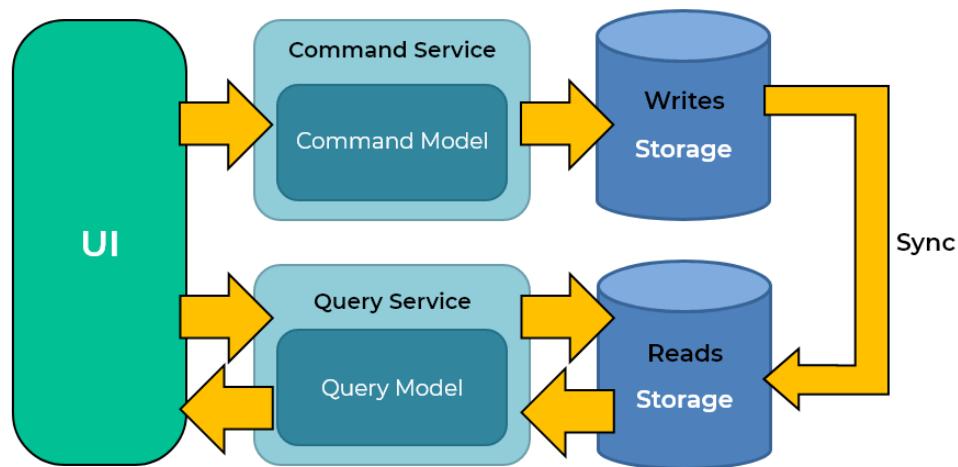
2.2 CQRS pattern

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 hằng 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

SON Web Mã (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 Tổng quan về 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.

2.4.1 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.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwpSpd3yD65VohhpucO0mLASjC"  
crossorigin="anonymous">  
  
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  
crossorigin="anonymous"></script>
```

2.4.2 Các thành phần chính của Bootstrap

2.4.2.1 Định dạng văn bản – Typography

Bootstrap cung cấp các lớp CSS để định dạng văn bản trên trang web.

2.4.2.2 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 **.button**

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>
```

Kết quả:

Đăng Nhập

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>
        <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>
```

Kết quả:

Student Table

#	Student ID	Name	Age
1	101	John Doe	20
2	102	Jane Smith	22

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>
```

Kết quả:

Trang chủ Thể loại liên hệ

2.4.2.3 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.

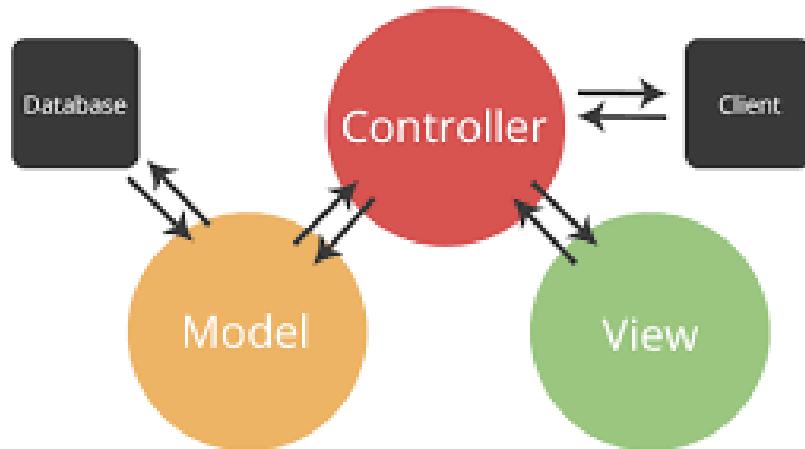
Tóm lại, Bootstrap là một framework mã nguồn mở mạnh mẽ và linh hoạt có thể giúp các nhà phát triển tạo các trang web và ứng dụng web đáp ứng nhanh chóng và dễ dàng.

2.5 Xây dựng chức năng bằng NodeJs & framework Express

2.5.1 Nền tảng NodeJs

Giới thiệu NodeJs

Node.js là một nền tảng JavaScript chạy trên máy chủ. Node.js được sử dụng để phát triển các ứng dụng web, mobile, IoT,...Node.js sử dụng mô hình event-driven, trong đó các sự kiện được xử lý một cách không đồng bộ. Mô hình này giúp Node.js có thể xử lý nhiều yêu cầu cùng lúc một cách hiệu quả.



Hình 2.4 Mô hình MVC

Cấu trúc của NodeJs

Kernel: Là lớp cơ bản nhất của Node.js, cung cấp các chức năng cơ bản như xử lý sự kiện, quản lý bộ nhớ,...

Modules: Là các thư viện chức năng được sử dụng để xây dựng ứng dụng Node.js.

API: Là các giao diện lập trình ứng dụng cung cấp cho các nhà phát triển khả năng truy cập vào các chức năng của Node.js.

Các tính năng của NodeJs

Ứng dụng web: Node.js được sử dụng để xây dựng các ứng dụng web động, hiệu quả và linh hoạt.

Ứng dụng mobile: Node.js được sử dụng để xây dựng các ứng dụng mobile với giao diện web.

Ứng dụng IoT: Node.js được sử dụng để xây dựng các ứng dụng IoT kết nối với các thiết bị vật lý.

Cài đặt NodeJs

Cách 1: Sử dụng trình quản lý gói:

Trình quản lý gói là một công cụ giúp cài đặt phần mềm trên máy tính. Có nhiều trình quản lý gói khác nhau, chẳng hạn như Chocolatey, Homebrew và Snap.

Để cài đặt Node.js bằng trình quản lý gói, có thể sử dụng các lệnh sau:

Chocolatey:

```
choco install node
```

Homebrew:

```
brew install node
```

Snap:

```
snap install node
```

Cách 2: Tải xuống và cài đặt từ trang web chính thức:

Có thể tải xuống và cài đặt Node.js từ trang web chính thức của Node.js. Để làm như vậy, có thể truy cập trang web <https://nodejs.org/en/> và nhấp vào nút "Download". Sau khi tải xuống tệp cài đặt, có thể cài đặt Node.js bằng cách chạy tệp cài đặt.

Cách 3: Sử dụng Docker

Docker là một nền tảng giúp tạo và chạy các ứng dụng trong các container.

Có thể sử dụng Docker để cài đặt Node.js bằng cách chạy lệnh sau:

```
docker run -it --rm node
```

Lệnh này sẽ khởi chạy một container Docker có Node.js cài đặt sẵn.

Yêu cầu hệ thống

Để cài đặt Node.js, cần có một máy tính có hệ điều hành Windows, macOS hoặc Linux. Cần có ít nhất 1GB RAM và 100MB dung lượng ổ cứng trống.

Kiểm tra cài đặt

Sau khi cài đặt Node.js, có thể kiểm tra xem Node.js đã được cài đặt thành công hay chưa bằng cách chạy lệnh sau:

```
node -v
```

Lệnh này sẽ hiển thị phiên bản Node.js hiện đang được cài đặt.

2.5.2 Cài đặt các thư viện hỗ trợ

Để cài đặt các thư viện hỗ trợ cho Node.js, có thể sử dụng trình quản lý gói NPM. Để làm như vậy có thể sử dụng lệnh sau:

```
npm install [tên-thư-viện]
```

Ví dụ: để cài đặt thư viện Express có thể sử dụng lệnh sau:

```
npm install express
```

Trình quản lý gói NPM sẽ tự động tải xuống và cài đặt thư viện vào thư mục node_modules của dự án hiện tại. Dưới đây là một số thư viện hỗ trợ phổ biến cho Node.js:

Express.js là một framework web cho Node.js, được sử dụng phổ biến để xây dựng ứng dụng web và API. Nó giúp đơn giản hóa việc xử lý yêu cầu HTTP, quản lý đường dẫn và tạo ra các ứng dụng web hiệu quả và dễ bảo trì.

Fast-csv là một thư viện Node.js được thiết kế để đọc và ghi các tệp CSV một cách nhanh chóng và hiệu quả. Nó cung cấp các phương thức và tính năng giúp xử lý dữ liệu CSV một cách thuận tiện trong ứng dụng Node.js.

Mongoose là một thư viện JavaScript cho Node.js và MongoDB, được sử

dụng để tương tác với cơ sở dữ liệu MongoDB thông qua JavaScript. Nó giúp đơn giản hóa quá trình thiết kế, xây dựng và duy trì ứng dụng Node.js sử dụng MongoDB làm cơ sở dữ liệu.

Path là một phần của Node.js và cung cấp các tiện ích để làm việc với các đường dẫn tệp và thư mục. Nó không yêu cầu cài đặt bổ sung vì là một phần của môi trường thực thi Node.js.

Fs (short for File System) là một phần của Node.js và cung cấp các chức năng để tương tác với hệ thống tệp và thư mục. fs cho phép đọc, ghi, xóa, di chuyển và thực hiện nhiều thao tác khác trên các tệp và thư mục trong ứng dụng Node.js.

2.5.3 Chạy ứng dụng Node.js

Để chạy ứng dụng Node.js sử dụng lệnh sau:

```
node [tên-tệp]
```

Ví dụ: để chạy ứng dụng có tên "server.js" sử dụng lệnh sau:

```
node server.js
```

Lệnh này sẽ khởi chạy ứng dụng Node.js tại cổng mặc định là 3000.

Có thể sử dụng lệnh sau để chạy ứng dụng Node.js:

```
nodemon server.js
```

Lệnh này sẽ khởi chạy ứng dụng Node.js và tự động khởi động lại ứng dụng khi có bất kỳ thay đổi nào được thực hiện đối với mã nguồn của ứng dụng.

2.5.4 Tài liệu Node.js

Có nhiều tài liệu Node.js có sẵn để giúp học cách sử dụng Node.js. Dưới đây là một số tài liệu phổ biến:

Tài liệu chính thức của Node.js: Tài liệu chính thức của Node.js là một nguồn tài nguyên tuyệt vời để bắt đầu học Node.js. Tài liệu này cung cấp hướng dẫn chi tiết về các khái niệm cơ bản của Node.js, chẳng hạn như module, event loop, và asynchronous programming.

Node Tutorial: Node Tutorial là một hướng dẫn từng bước về cách tạo ứng dụng Node.js. Hướng dẫn này bao gồm các khái niệm cơ bản của Node.js,

cũng như các khái niệm nâng cao hơn, chẳng hạn như routing và module loading.

Node Cheat Sheet: Node Cheat Sheet là một bản tóm tắt các lệnh và cú pháp Node.js. Bảng cheat sheet này là một tài nguyên hữu ích để tham khảo nhanh các lệnh và cú pháp Node.js.

Node Learning Resources: Node Learning Resources là một trang web tổng hợp các tài nguyên học Node.js. Trang web này bao gồm các hướng dẫn, bài viết, video và các tài nguyên khác.

Ngoài ra, cũng có thể tìm thấy nhiều tài liệu Node.js trên các trang web và diễn đàn trực tuyến.

Dưới đây là một số trang web và diễn đàn phổ biến có tài liệu Node.js:

Node.js Documentation: Trang web chính thức của Node.js cung cấp tài liệu về Node.js, bao gồm các hướng dẫn, bài viết, và API reference.

Node.js Tutorials: Trang web này cung cấp các hướng dẫn từng bước về cách tạo ứng dụng Node.js.

Node.js Cheat Sheets: Trang web này cung cấp các bảng cheat sheet về Node.js.

Node.js Learning Resources: Trang web này tổng hợp các tài nguyên học Node.js.

Stack Overflow: Stack Overflow là một diễn đàn hỏi đáp dành cho các nhà phát triển. Có thể tìm thấy nhiều câu hỏi và câu trả lời về Node.js trên Stack Overflow.

GitHub: GitHub là một nền tảng chia sẻ mã nguồn. Có thể tìm thấy nhiều ví dụ và tài liệu Node.js trên GitHub.

Reddit: Reddit là một trang web chia sẻ nội dung. Có thể tìm thấy nhiều bài viết và câu hỏi về Node.js trên Reddit.

Medium: Medium là một nền tảng xuất bản nội dung. Có thể tìm thấy nhiều bài viết về Node.js trên Medium.

2.5.5 Kiến thức về framework Express.js

Express là một framework Node.js giúp xây dựng ứng dụng web một cách nhanh chóng và dễ dàng. Express cung cấp nhiều tính năng hữu ích cho việc phát triển ứng dụng web, bao gồm routing, middleware, session management,...

Routing

Routing là tính năng giúp định tuyến các yêu cầu đến các hàm xử lý tương ứng. Express sử dụng routing để định tuyến các yêu cầu đến các hàm xử lý dựa trên URL của yêu cầu.

Middleware

Middleware là các hàm được thực thi trước và sau các hàm xử lý. Middleware thường được sử dụng để thực hiện các tác vụ chung, chẳng hạn như xác thực người dùng, kiểm tra bảo mật,...

Session management

Session management là tính năng giúp lưu trữ thông tin trạng thái của người dùng trong một phiên. Express sử dụng session management để lưu trữ thông tin trạng thái của người dùng, chẳng hạn như tên người dùng, ID người dùng,...

Các tính năng của Express.js

Nhanh chóng: Express giúp xây dựng ứng dụng web một cách nhanh chóng.

Dễ sử dụng: Express có cú pháp đơn giản và dễ học.

Linh hoạt: Express có thể được tùy chỉnh để đáp ứng nhu cầu của các ứng dụng khác nhau.

Các ứng dụng của Express.js

Ứng dụng web: Express được sử dụng để xây dựng các ứng dụng web động, hiệu quả và linh hoạt.

Ứng dụng API: Express được sử dụng để xây dựng các ứng dụng API cung cấp các dịch vụ cho các ứng dụng khác.

Ứng dụng trò chơi: Express được sử dụng để xây dựng các ứng dụng trò chơi web.

Cài đặt Express.js

Để cài đặt Express.js, Có thể sử dụng trình quản lý gói NPM. Để làm như vậy, có thể sử dụng lệnh sau:

```
npm install express
```

Trình quản lý gói NPM sẽ tự động tải xuống và cài đặt Express.js vào thư mục node_modules của dự án hiện tại.

Kiểm tra cài đặt

Sau khi cài đặt Express.js, có thể kiểm tra xem Express.js đã được cài đặt thành công hay chưa bằng cách sử dụng lệnh sau:

```
npm list | grep express
```

Lệnh này sẽ hiển thị danh sách các thư viện đã được cài đặt, bao gồm Express.js.

2.6 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 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 website

Website khoá học self-learning online được sử dụng bởi đối tượng sinh viên, các leader team.

Đối với sinh viên khi tham gia hệ thống sẽ thực hiện: Đăng ký tài khoản và đăng nhập vào hệ thống hoặc có thể đăng nhập bằng tài khoản google. Sinh viên có tra cứu các khoá học và đăng ký khoá học để học.

Đối với các leader: Các leader team sẽ đăng nhập vào trang quản trị với tài khoản được system admin cung cấp là sub admin. Sub admin có thể thay đổi mật khẩu, cập nhật thông tin cá nhân, quản lý tài khoản người dùng, thêm, cập nhật, vô hiệu hoá và khôi phục khoá học, chương học, bài học. Ngoài ra sub admin có thể xem thống kê về lượt truy cập, thời gian các bài học và số lượng khoá học, số lượng đăng ký để điều chỉnh cho phù hợp.

Đối với người quản trị website: Người quản trị đăng nhập vào trang quản trị với tài khoản mặc định, đây là tài khoản có quyền cao nhất của website.

3.2 Đặc tả các chức năng hệ thống

3.2.1 Chức năng của sinh viên

Xem và tra cứu khoá học: Sinh viên có thể đăng nhập hoặc không (tài khoản khách) để xem thông tin khoá học có trên website hoặc tra cứu khoá học theo cấp độ, tên khoá học, lĩnh vực.

Đăng ký khoá học: Sinh viên có thể đăng ký khoá học.

Đánh giá tiến độ học: Sinh viên có thể theo dõi phần trăm và trạng thái khoá học đã done hay chưa.

Xem lộ trình học: Sinh viên có thể theo học theo lộ trình được cung cấp trên khoá học.

3.2.2 Chức năng của leader

Đăng nhập: Đăng nhập bằng tài khoản được system admin cung cấp.

Thay đổi mật khẩu: Sau khi đăng nhập, có thể thay đổi mật khẩu để tăng tính bảo mật.

Quản lý người dùng: Có thể xem, tìm kiếm, thêm, sửa, xoá, vô hiệu hoá/ khôi phục tài khoản người dùng.

Quản lý khoá học: Có thể xem, tìm kiếm, thêm, sửa, xoá, vô hiệu hoá/ khôi phục khoá học, chương học thuộc khoá học, bài học thuộc chương học.

Thông kê: Có thể xem thông kê về lượt truy cập, số lượng bài học, thời gian học và số lượng đăng ký để điều chỉnh khoá học cho phù hợp.

3.2.3 Chức năng của quản trị viên

Đăng nhập: Đăng nhập bằng tài khoản mặc định, đây là tài khoản có quyền cao nhất.

Thay đổi mật khẩu: Sau khi đăng nhập, có thể thay đổi mật khẩu để tăng tính bảo mật.

Quản lý người dùng: Có thể xem, tìm kiếm, thêm, sửa, xoá, vô hiệu hoá/ khôi phục tài khoản người dùng và tài khoản sub admin.

Quản lý khoá học: Có thể xem, tìm kiếm, thêm, sửa, xoá, vô hiệu hoá/ khôi phục khoá học, chương học thuộc khoá học, bài học thuộc chương học.

Thông kê: Có thể xem thông kê về lượt truy cập, số lượng bài học, thời gian học và số lượng đăng ký để điều chỉnh khoá học cho phù hợp.

3.2.4 Quản lý người dùng

Đăng Nhập: Tài khoản được phân quyền user (người dùng), sub admin (tài khoản của các leader tham gia quản trị website), system admin (là tài khoản có quyền cao nhất).

Đăng Ký: Các leader không có quyền đăng ký tài khoản quản trị chỉ có thể đăng nhập bằng tài khoản được cấp nhưng có thể đăng ký tài khoản user. Sinh viên có thể đăng ký tài khoản của website hoặc đăng nhập bằng tài khoản google.

3.3 Thiết kế dữ liệu

3.3.1 Danh sách thực thể

Accounts(username, password, role, profile{ fullName, birthDate, specialty, avatar, address, phone }, googleId, isDeleted, timestamps)

Courses(name, description, benefits[], level, oldPrice, newPrice, author image, totalLessons, totalChapters, chapters[Chapter], category, duration, status, isDeleted, timestamps)

Chapters(courseId, title, lessons[Lesson], chapterOrder, status, isDeleted, timestamps)

Lessons(chapterId, name, description, videoUrl, duration, lessonOrder, isDeleted, status, timestamps)

RegistrationCourses(userId, courseId, registrationDate)

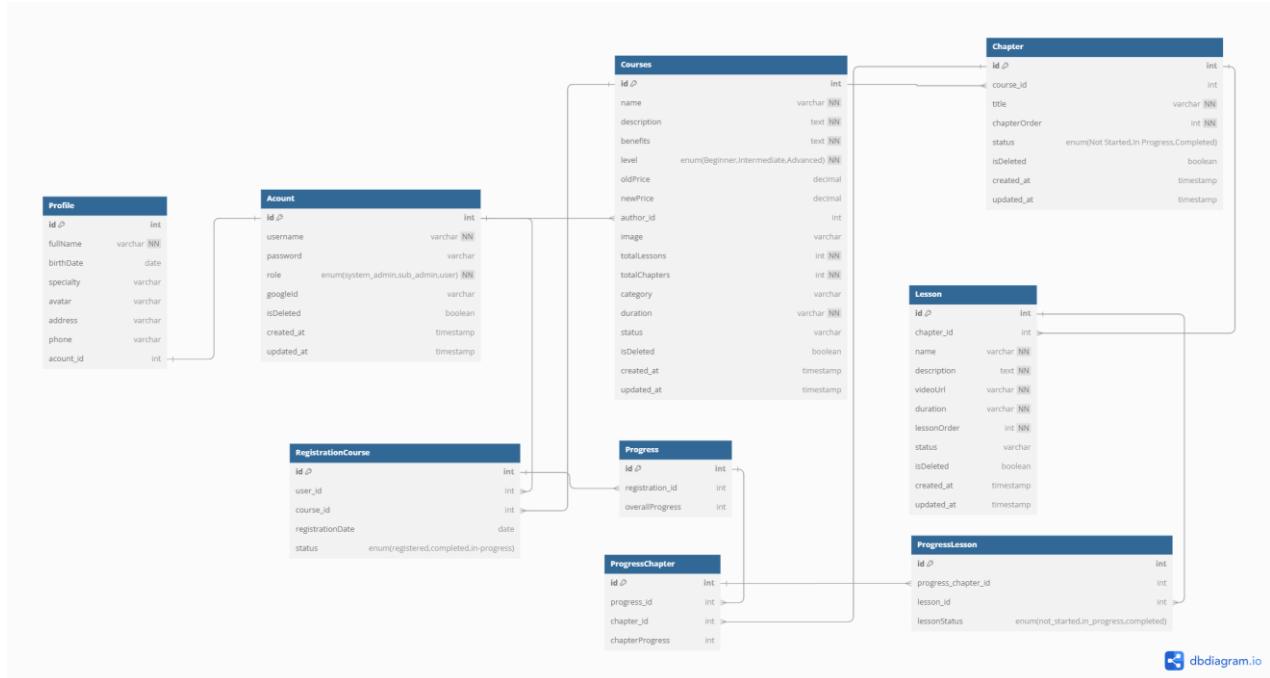
Progress(registrationId, chapters[{chapterId, lessons[lessonId, status], progress }], overallProgress)

Visits(route, count, userId[], timestamps)

Bảng 3.1 Danh sách thực thể

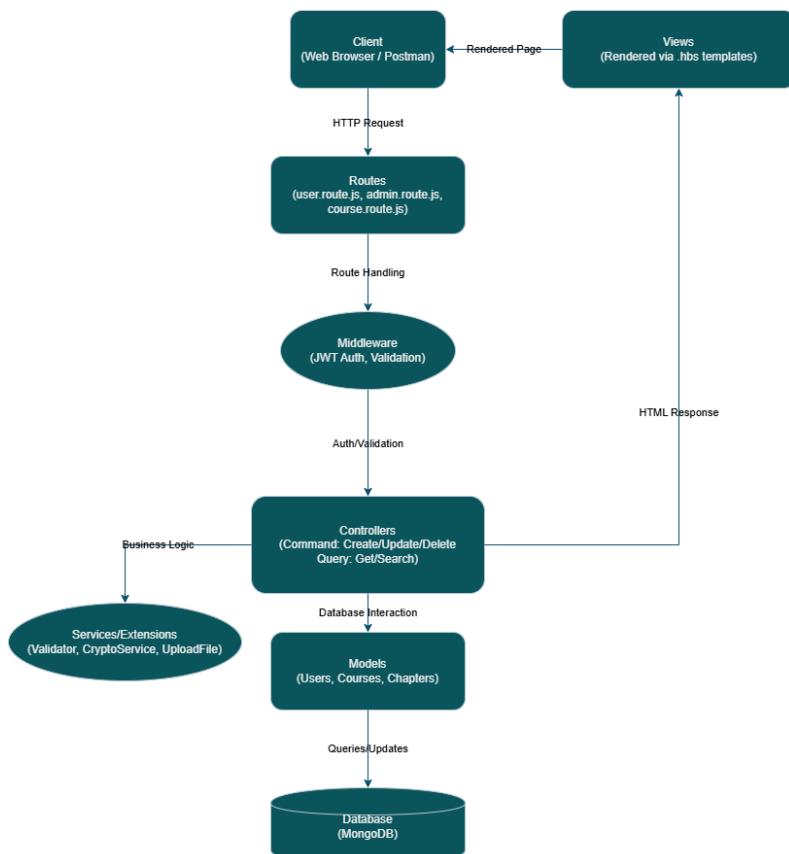
STT	Tên thực thể	Điễn giải
1	Accounts	Lưu trữ thông tin về tài khoản và người dùng
2	Courses	Lưu trữ thông tin về khoá học
3	Chapters	Lưu trữ thông tin về chương học
4	Lessons	Lưu trữ thông tin về bài học
5	RegistrationCourses	Lưu trữ thông tin đăng ký khoá học
6	Progress	Lưu trữ thông tin về tiến độ học của khoá học
7	Visits	Lưu trữ thông tin về lượt truy cập

3.3.2 Mô hình dữ liệu



Hình 3.1 Mô hình dữ liệu

3.4 Kiến trúc hệ thống

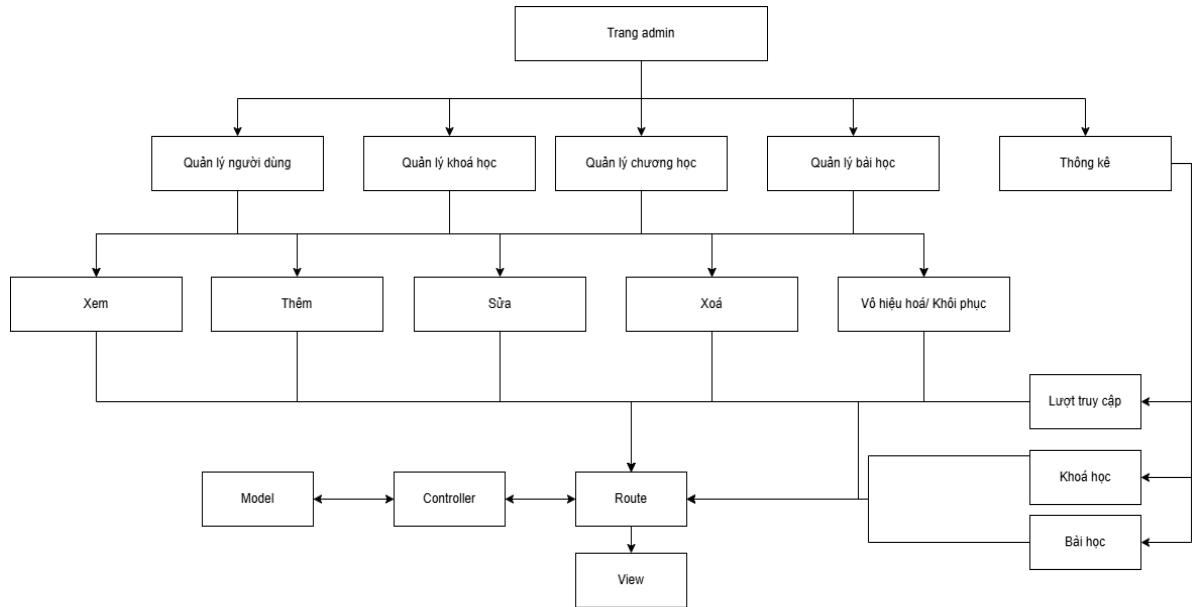


Hình 3.1 Kiến trúc hệ thống

3.5 Thiết kế giao diện

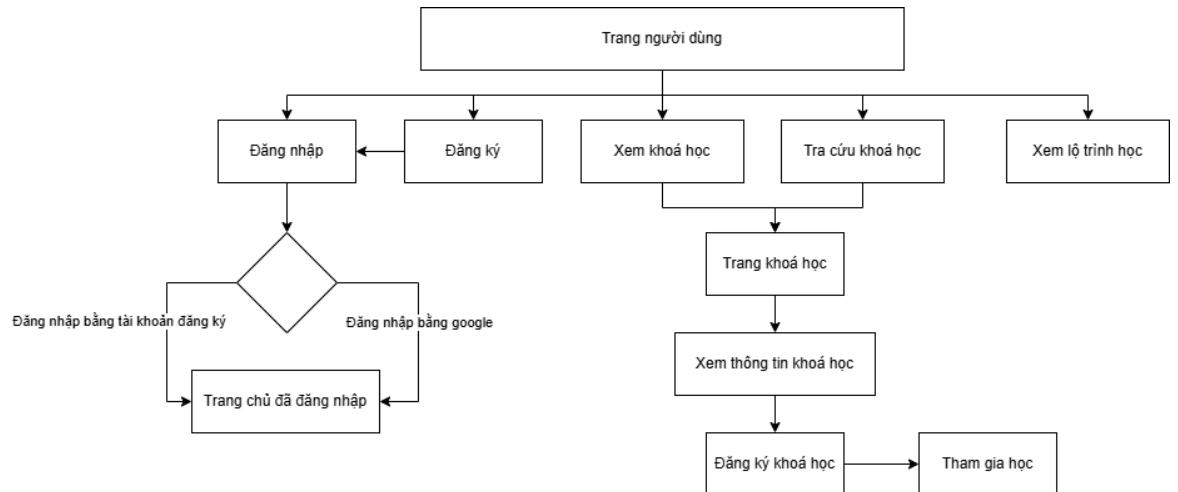
3.5.1 Sơ đồ website

Người quản trị:



Hình 3. 2 Sơ đồ website người quản trị

Người dùng:



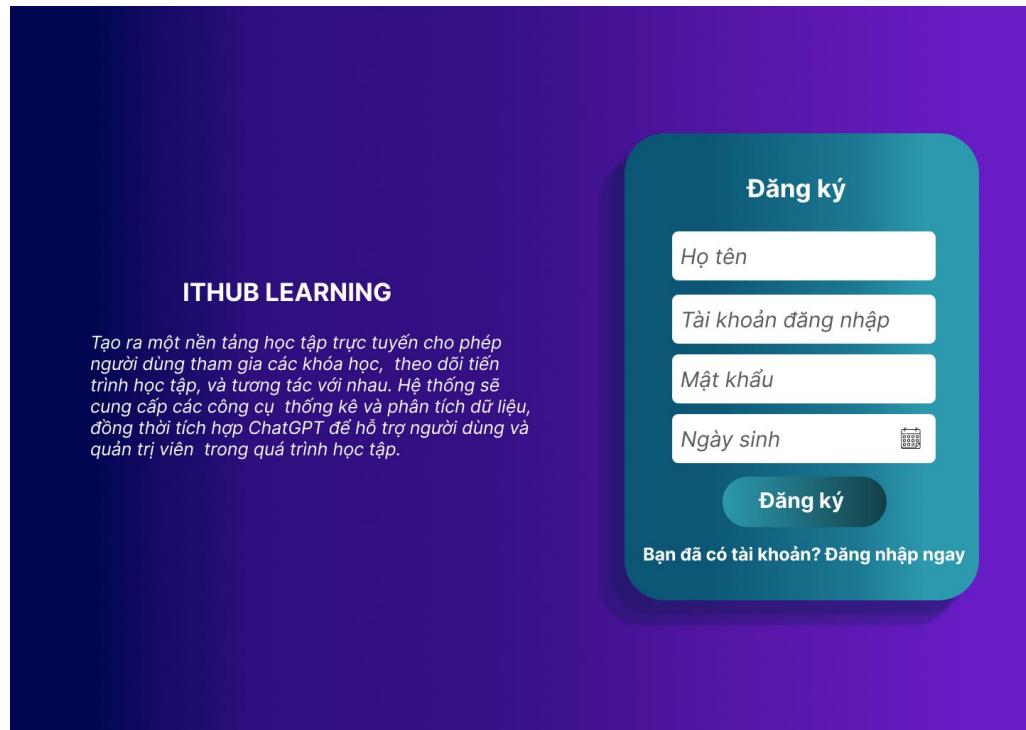
Hình 3. 3 Sơ đồ website người dùng

3.5.2 Giao diện trang

Chức năng người dùng:

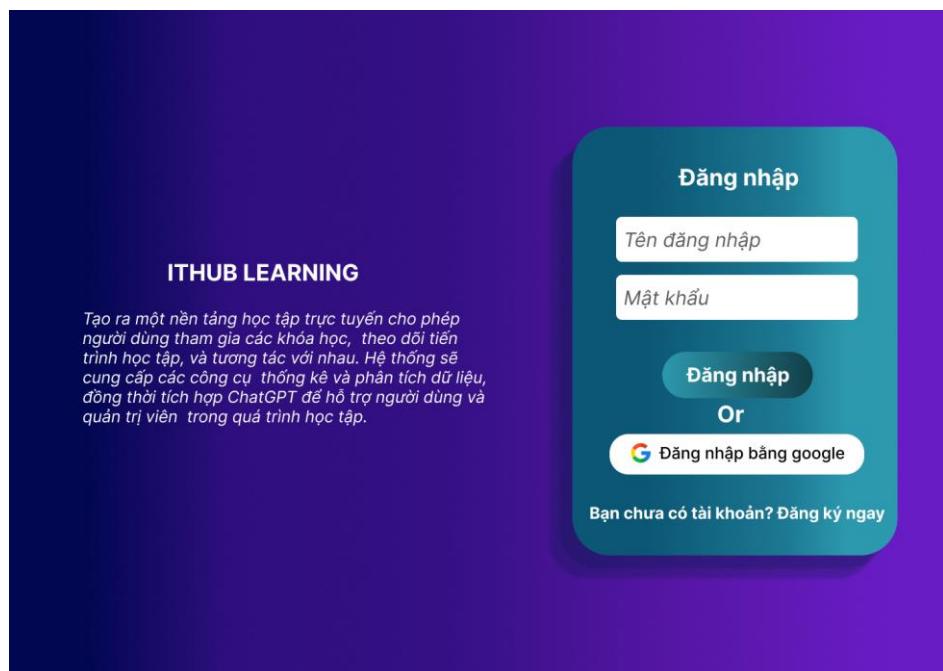
Đăng ký tài khoản: Người dùng có thể đăng ký tài khoản của website và

đăng nhập vào website với tài khoản đã đăng ký.



Hình 3. 4 Giao diện trang đăng ký

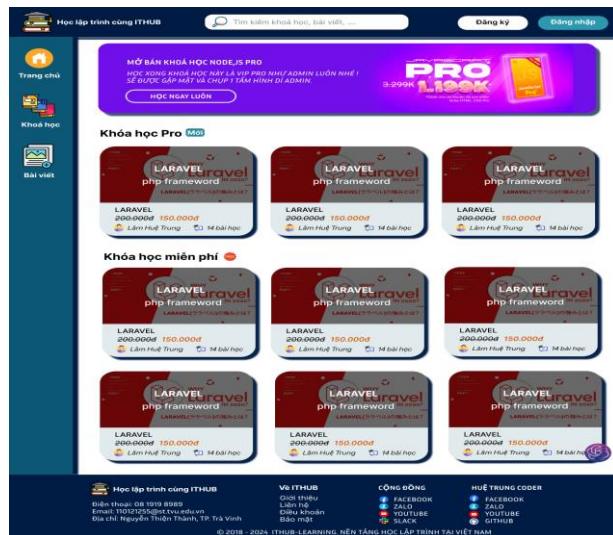
Đăng nhập: Người dùng có thể đăng nhập vào website với tài khoản đã đăng ký hoặc đăng nhập bằng tài khoản google.



Hình 3. 5 Giao diện trang đăng nhập của người dùng

Xem khoá học: Tại trang chủ người dùng có thể xem các khoá học có trên website mà không cần phải đăng nhập.

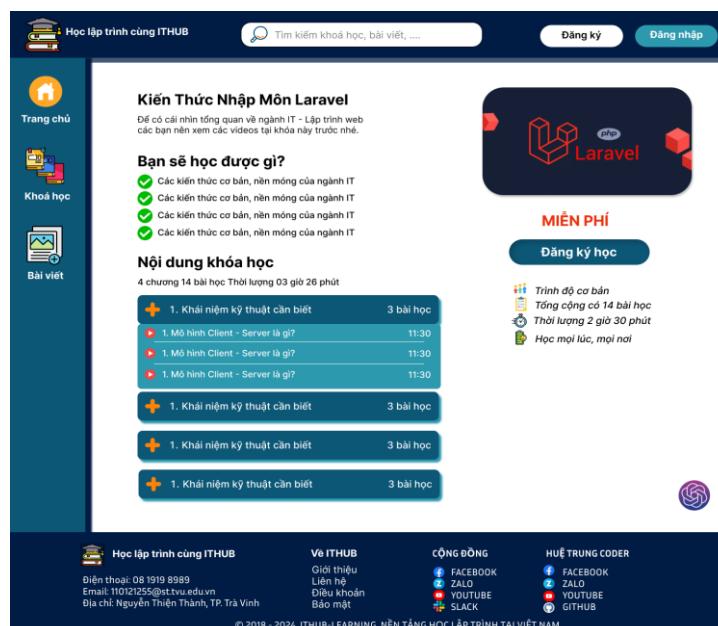
Tra cứu khoá học: Người dùng có thể tìm kiếm các khoá học bằng tên khoá học, giá, trình độ, lĩnh vực.



Hình 3. 6 Giao diện trang chủ người dùng

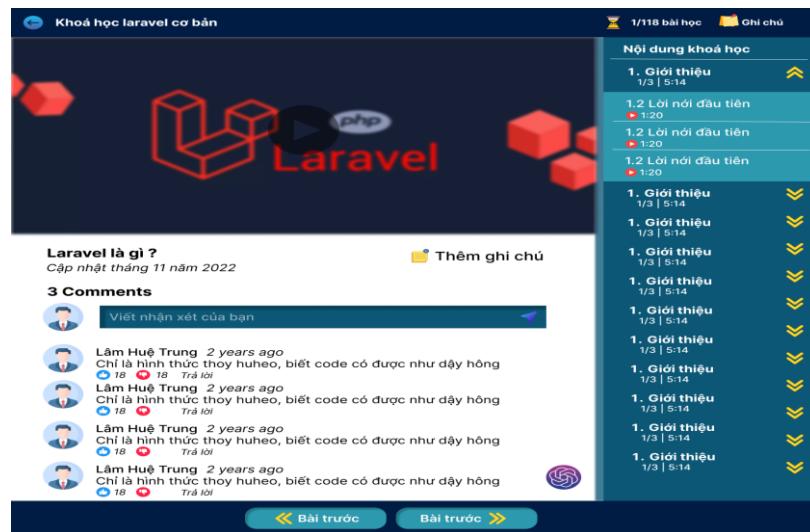
Xem thông tin chi tiết của khoá học: Khi người dùng nhấp chuột vào khoá học sẽ xem được các thông tin cơ bản của khoá học như tên khoá học, mô tả, v.v.

Đăng ký khoá học: Người dùng đăng ký khoá học tại giao diện xem chi tiết, nếu như chưa đăng nhập thì website sẽ yêu cầu đăng nhập.



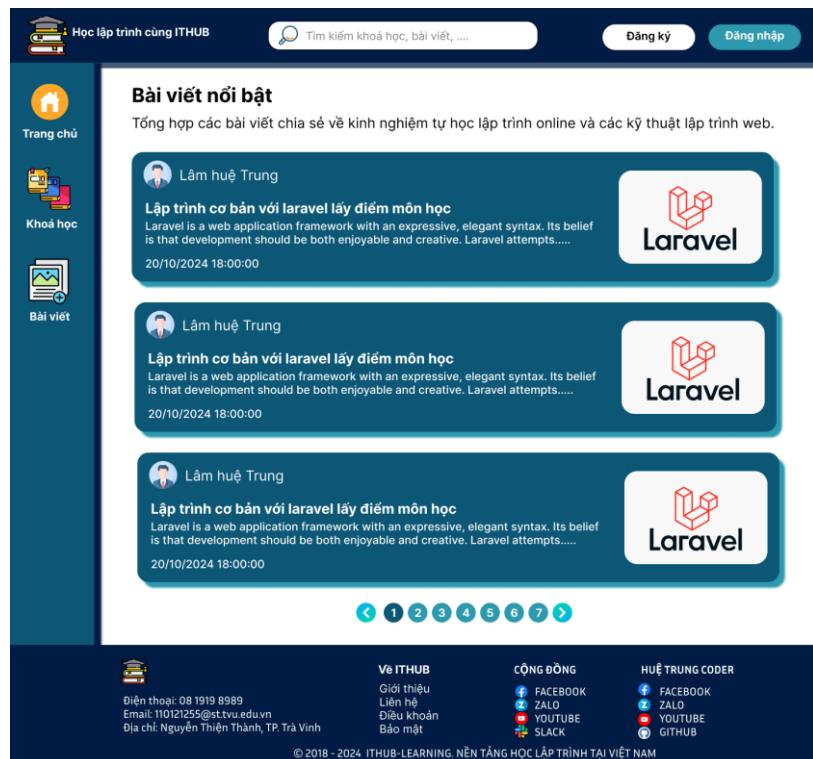
Hình 3. 7 Giao diện đăng ký khoá học

Tham gia học: Sau khi người dùng đã đăng ký khoá học, người dùng có thể tham gia học theo cấu trúc bài giảng và được đánh giá tiến độ tự động.



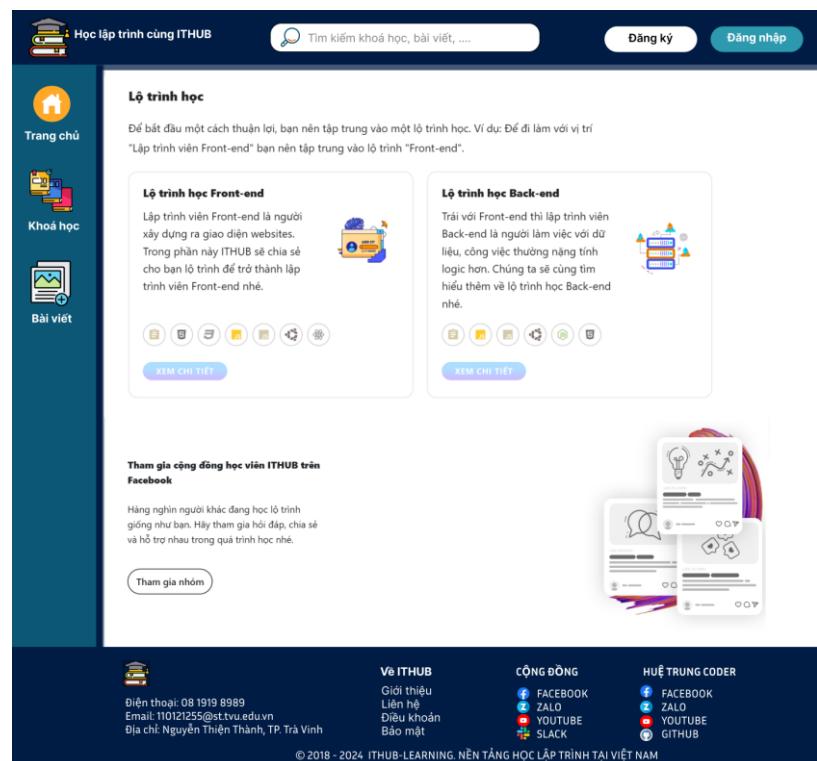
Hình 3. 8 Giao diện học khoá học

Ngoài ra, người dùng có thể xem các bài viết liên quan đến lập trình và lộ trình học của website đưa ra.



Hình 3. 9 Giao diện bài viết

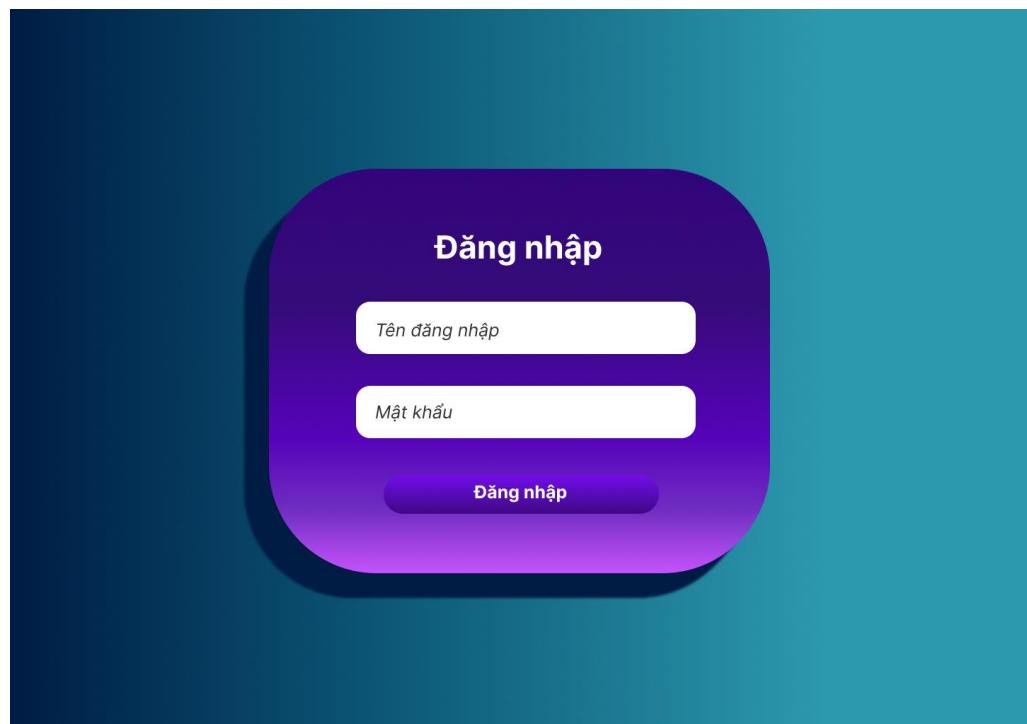
Xây dựng website khoá học online bằng NodeJs & Express



Hình 3. 10 Giao diện lộ trình học

Chức năng người quản trị:

Đăng nhập: Quản trị viên có thể đăng nhập vào trang admin với tài khoản mặc định và leader đăng nhập bằng tài khoản đã được cấp.



Hình 3. 11 Giao diện đăng nhập trang quản trị

Giao diện chính trang quản trị:



Hình 3. 12 Giao diện chính trang quản trị

Ngoài ra, còn có các giao diện như thêm, sửa, xoá, danh sách người dùng, khoá học, chương học, bài học và thống kê số lượng lượt truy cập, tỷ lệ bài học và thời gian, số lượng tài khoản, tiến trình học và số lượng đăng ký khoá học.

The figure shows the Admin Dashboard with a modal window titled "Add & Edit.....". The modal contains three input fields labeled "Name" and three "Save" and "Previous" buttons. The sidebar menu on the left is identical to the one in Figure 3.12.

The modal window has the following structure:

Add & Edit.....

Name

Name

Name

Save **Previous**

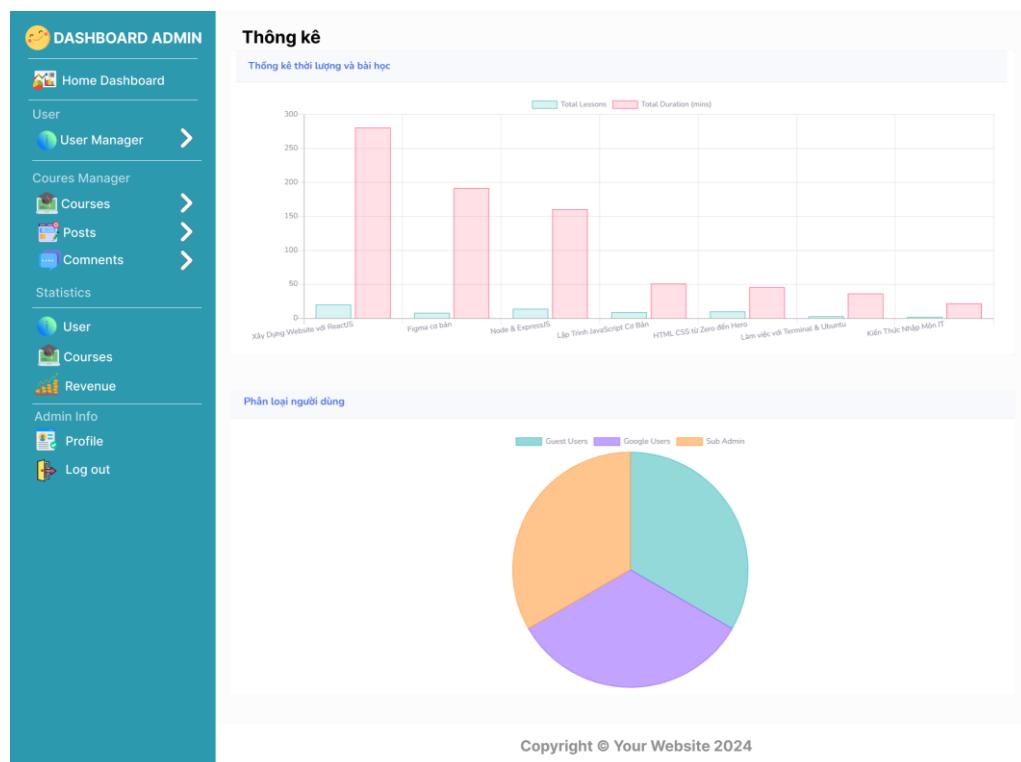
At the bottom right of the main area, it says "Copyright © Your Website 2024".

Hình 3. 13 Giao diện thêm và sửa thông tin

Xây dựng website khoá học online bằng NodeJs & Express

The screenshot shows the 'DASHBOARD ADMIN' interface. On the left, a sidebar menu includes 'Home Dashboard', 'User Manager', 'Courses Manager', 'Statistics', 'Revenue', 'Admin Info', 'Profile', and 'Log out'. The main area is titled 'List all' and displays a table with columns: Name, Position, Office, Age, Start date, and 'Thao tác' (Actions). The table has 57 entries. At the bottom, there is a search bar, a page number indicator (1 of 6), and navigation buttons for 'Previous' and 'Next'.

Hình 3. 14 Giao diện danh sách hiển thị



Hình 3. 15 Giao diện thống kê

CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU

4.1 Sản phẩm đạt được sau quá trình nghiên cứu

Sau quá trình thực hiện đồ án, website khoá học self-learning online đã được xây dựng thành công và đáp ứng được các yêu cầu đề ra.

Sản phẩm là sự kết hợp Node.js & Express.js, handlebar, sass, bootstrap kết hợp với hệ quản trị cơ sở dữ liệu MongoDB theo mô hình MVC.

Link sản phẩm: <https://ithub-learning.onrender.com/>

4.2 Về hiệu năng

Ứng dụng có khả năng xử lý nhanh chóng các yêu cầu của người dùng. Theo thử nghiệm, ứng dụng có thể xử lý các yêu cầu từ người dùng một cách nhanh chóng.

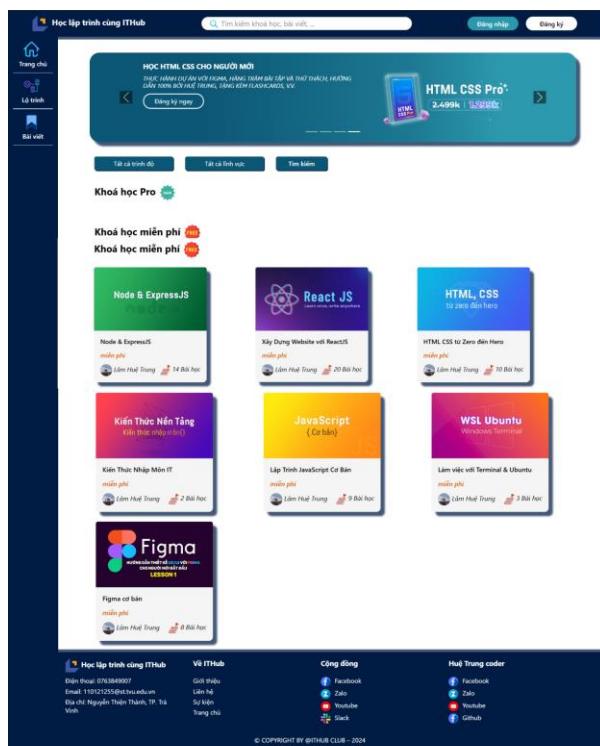
4.3 Về trải nghiệm người dùng

Giao diện của ứng dụng được thiết kế đơn giản, thân thiện và dễ sử dụng. Người dùng có thể dễ dàng tìm thấy các chức năng cần thiết. Các chức năng có mô tả cụ thể dễ hiểu giúp cho người dùng thực hiện dễ dàng.

4.4 Kết quả thử nghiệm

Xây dựng thành công giao diện điện trang chủ sau khi chỉnh sửa màu sắc và các thành phần trên trang giúp người dùng dễ dàng tìm thấy các tính năng cần thiết. Người dùng có thể thực hiện các thao tác đăng ký, đăng nhập xem thông tin khoá học, tra cứu khoá học, lọc khoá học, xem thông tin chi tiết của khoá học tao giao diện trang chủ.

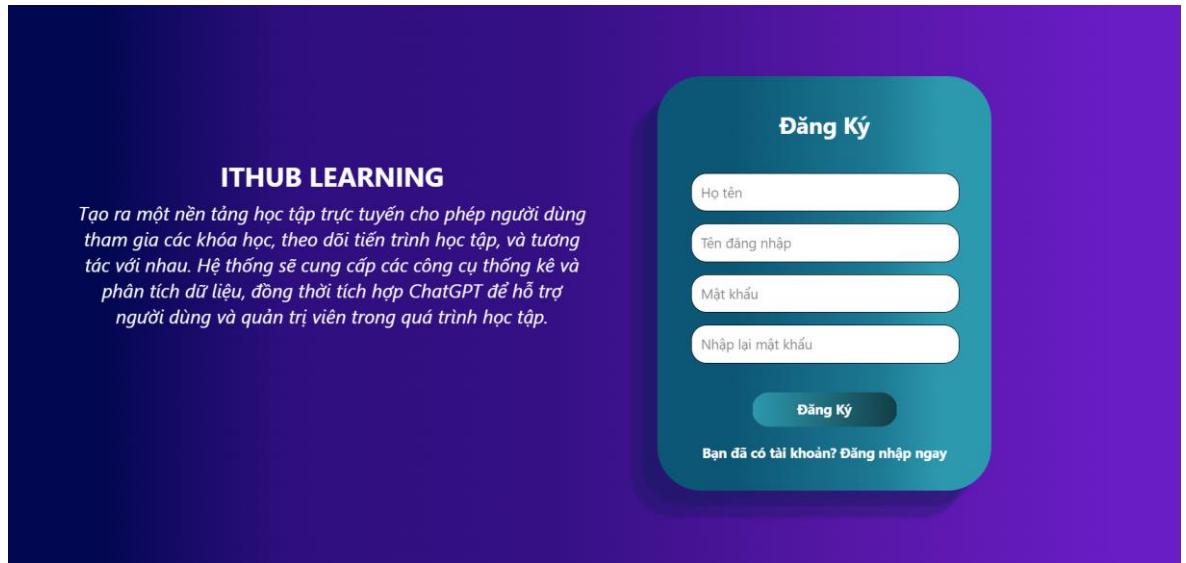
Xây dựng website khoá học online bằng NodeJs & Express



Hình 4. 1 Giao diện trang chủ người dùng

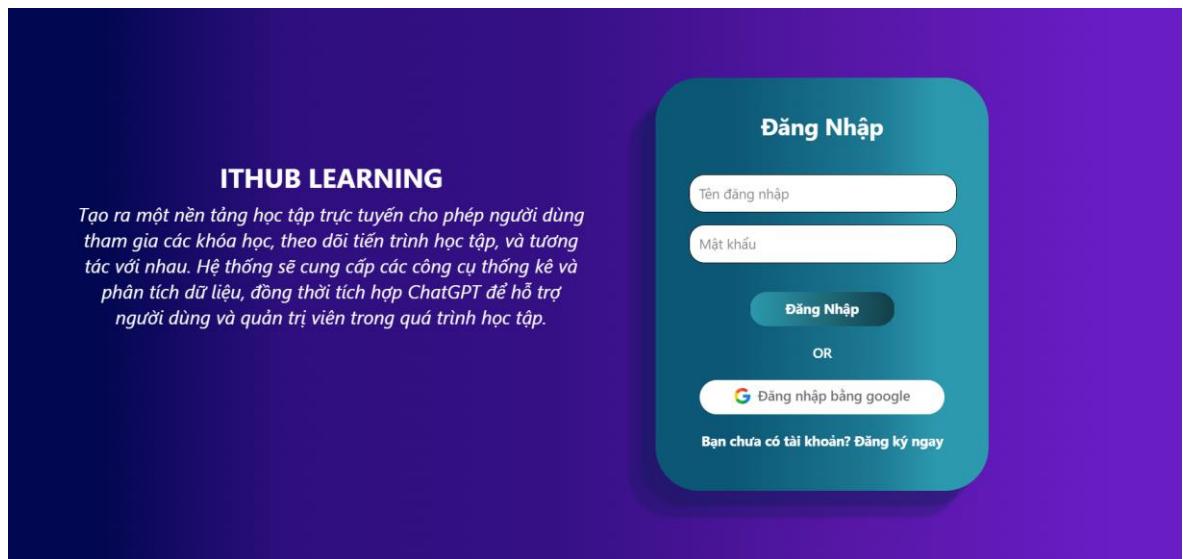
4.4.1 Chức năng người dùng

Đăng ký tài khoản: Người dùng có thể đăng ký tài khoản của website và đăng nhập vào website với tài khoản đó.



Hình 4. 2 Trang đăng ký

Đăng nhập: Người dùng có thể đăng nhập bằng tài khoản đã đăng ký hoặc đăng nhập bằng tài khoản google.



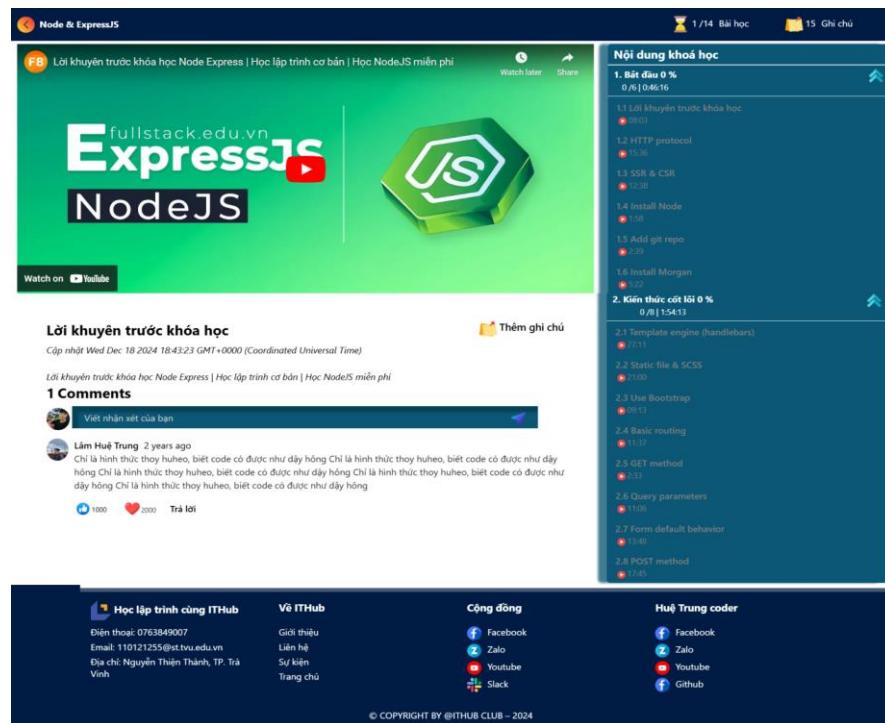
Hình 4. 3 Trang đăng nhập

Xem thông tin khoá học: Người dùng có thể nhấn vào khoá học ở trang chủ để xem thông tin khoá học.

Đăng ký khoá học: Người dùng có thể đăng ký khoá học, nếu người dùng vẫn chưa đăng nhập thì website sẽ yêu cầu đăng nhập.

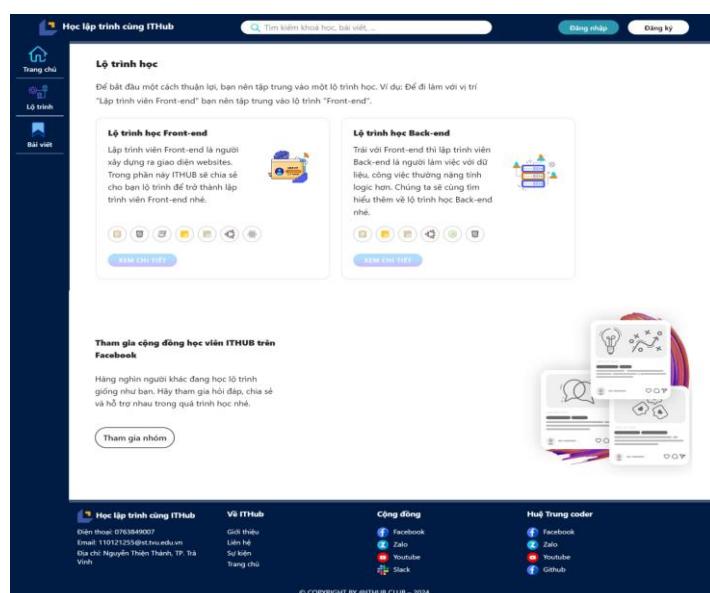
Hình 4. 4 Trang xem thông tin khoá học và đăng ký

Tham gia học: Người dùng sau khi đăng ký khoá học sẽ thực hiện học tập theo hệ thống bài học do website đưa ra và được đánh giá tiến độ tự động theo thời gian xem video hướng dẫn.



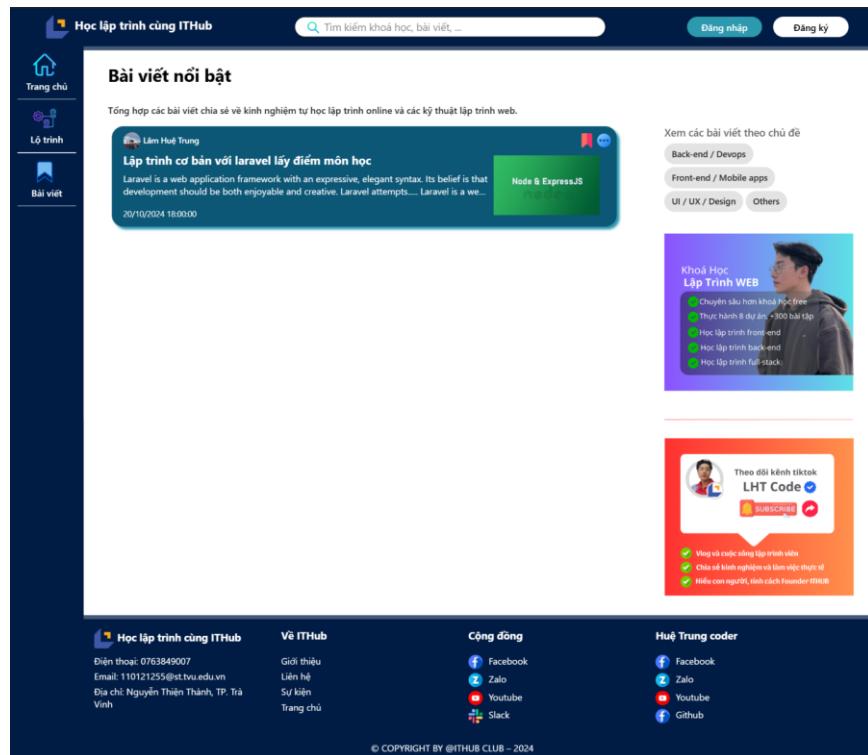
Hình 4. 5 Trang tham gia học

Xem lộ trình: Nếu người dùng không biết bắt đầu từ đâu thì có thể tham khảo lộ trình do website đưa ra và tham gia học các khoá học theo hướng dẫn của lộ trình.



Hình 4. 6 Trang lộ trình học tập

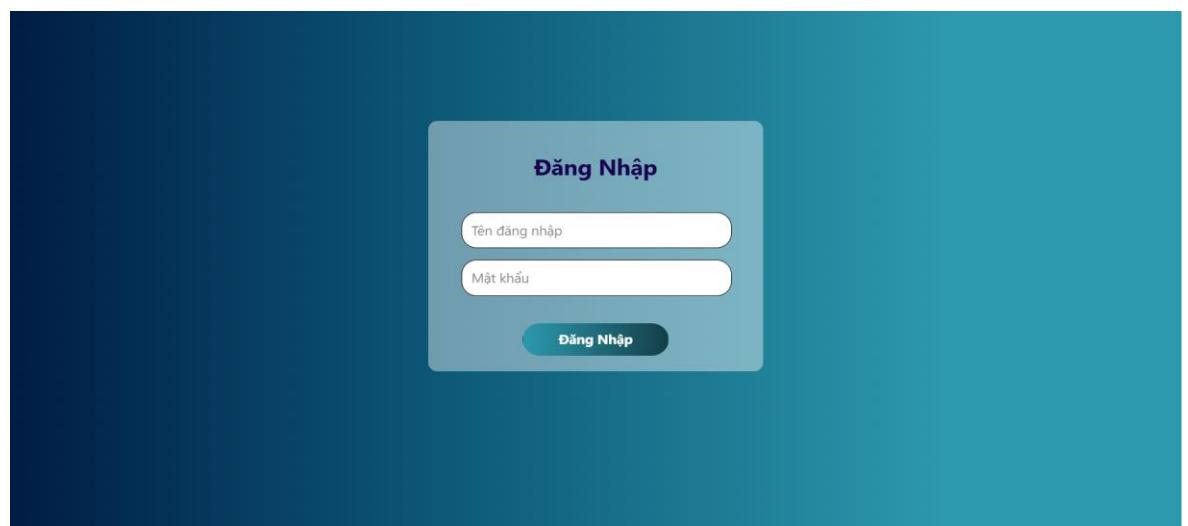
Xem bài viết: Người dùng có thể có thể tham khảo các bài viết về lập trình trên trang website để nắm được thêm nhiều kiến thức khác.



Hình 4. 7 Trang bài viết

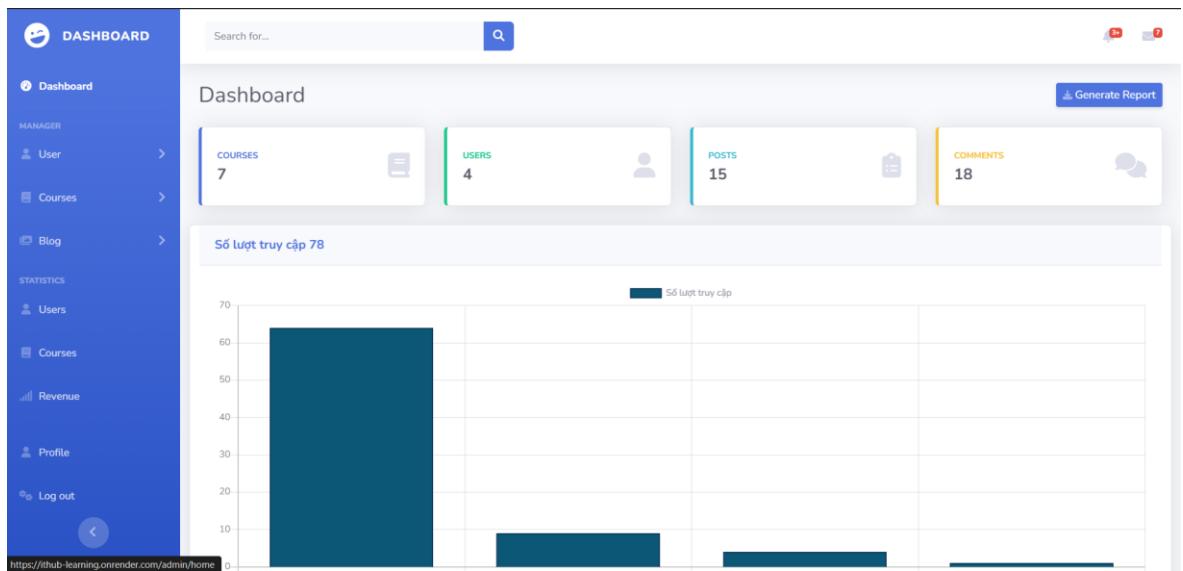
4.4.2 Chức năng người quản trị

Đăng nhập: Quản trị viên có quyền cao nhất và đăng nhập bằng tài khoản mặc định được tạo ra khi khởi chạy dự án. Các leader sẽ đăng nhập vào trang trị với tài khoản sub admin được cung cấp.



Hình 4. 8 Trang đăng nhập quản trị viên

Xây dựng website khoá học online bằng NodeJs & Express



Hình 4. 9 Trang quản trị

Thêm người dùng: Đối với tài khoản quản trị thì có thể tạo tài khoản cho người dùng và các leader, đối với các leader thì chỉ được tạo tài khoản cho người dùng.

The screenshot shows the 'Add User' form within the Admin Dashboard. The form fields include: User Name (Example: LamHueTrung), Full Name (Example: Lâm Huệ Trung), Birthday (dd/mm/yyyy), Specialty (Example: Front-end, Back-end, Design UI/UX, v.v.), Phone Number (Examples: 076384xxxx), Address (Examples: Trà Vinh, Sóc Trăng, Tp. HCM), Avatar (Choose file / Không có tệp nào được chọn), and Role (Sub Admin). At the bottom are 'Submit' and 'Back' buttons. The footer of the page includes the copyright notice: Copyright © Your Website 2024.

Hình 4. 10 Trang thêm người dùng

Danh sách người dùng: Bao gồm một số thông tin cơ bản của người dùng và các chức năng như xem chi tiết, sửa, xoá, vô hiệu hoá/ khôi phục.

Xây dựng website khoá học online bằng NodeJs & Express

The screenshot shows a user management interface. On the left is a sidebar with a blue header 'DASHBOARD' and a list of menu items: 'Dashboard', 'MANAGER' (with 'User', 'Courses', 'Blog'), 'STATISTICS' (with 'Users', 'Courses'), 'Revenue', 'Profile', and 'Log out'. The main content area has a search bar at the top. Below it is a table titled 'List all users' with columns: Active status, Role, Full name, User name, Password, and Actions. There are three entries:

Active status	Role	Full name	User name	Password	Actions
Đang hoạt động	USER	Nguyễn Hoàng Nhựt	NguyenHoangNhut	***** Xem mật khẩu	[Edit, Delete]
Đang hoạt động	SUB ADMIN	Mã Đại Phú	MaDaiPhu	***** Xem mật khẩu	[Edit, Delete]
Đang hoạt động	USER	huE TRUNG LÂM	namphongtctv@gmail.com	Bảo mật Google	[Edit, Delete]

At the bottom, there's a message 'Showing 1 to 3 of 3 entries', a 'Back' button, and navigation links 'Previous' and 'Next'. The URL in the address bar is 'https://ithub-learning.onrender.com/admin'.

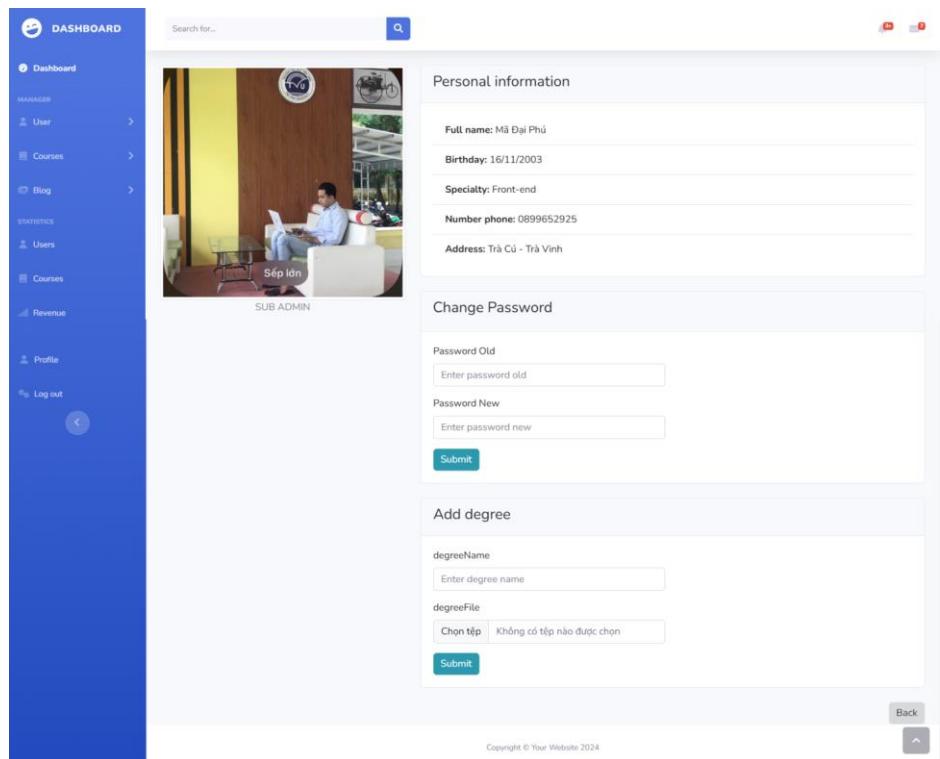
Hình 4. 11 Trang danh sách người dùng

The screenshot shows an 'Update User' form. The sidebar on the left is identical to the one in the previous screenshot. The main area has a title 'Update User'. It contains fields for 'User Name' (NguyenHoangNhut), 'Full Name' (Nguyễn Hoàng Nhựt), 'Birthday' (dd/mm/yyyy), 'Specialty' (empty), 'Phone Number' (empty), 'Address' (Examples: Trà Vinh, Sóc Trăng, Tp. HCM), 'Avatar' (Chọn tệp - Không có tệp nào được chọn), and 'Role' (Sub Admin). At the bottom are 'Submit' and 'Back' buttons. The URL in the address bar is 'https://ithub-learning.onrender.com/admin'.

Hình 4. 12 Trang cập nhật thông tin người dùng

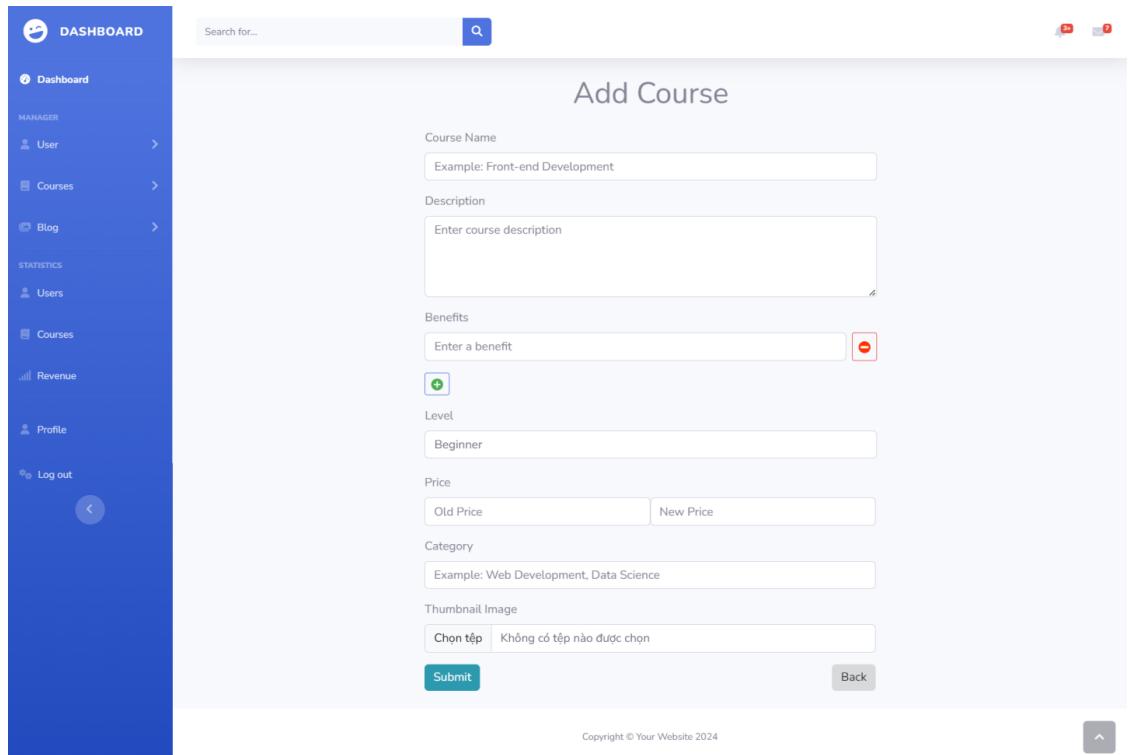
Thay đổi mật khẩu: Quản trị viên có thể thay đổi mật khẩu mặc định. Đặc biệt, khi tạo tài khoản sub admin và user thì mật khẩu sẽ được tạo mặc định là tài khoản + * nên khi nhận được tài khoản thì người dùng và các sub admin có thể vào thay đổi mật khẩu thành của chính mình để tăng tính bảo mật.

Xây dựng website khoá học online bằng NodeJs & Express



Hình 4. 13 Giao diện xem thông tin người dùng

Thêm khoá học: Quản trị viên và các leader có thể thêm khoá học mới với các thông tin cần thiết.



Hình 4. 14 Trang thêm khoá học

Trang danh sách khoá học: Hiển thị thông tin của tất cả khoá học có trên trang website với các chức năng xem chi tiết, sửa, xoá, vô hiệu hoá/ khôi phục, xem danh sách chương học thuộc khoá học.

Active status	Name course	Level	Total chapters	Total lessons	Author	Actions
Đang hoạt động	Node & ExpressJS	Intermediate	2	14	Lâm Huệ Trung	⋮⋮⋮
Đang hoạt động	Xây Dựng Website với ReactJS	Intermediate	4	20	Lâm Huệ Trung	⋮⋮⋮
Đang hoạt động	HTML CSS từ Zero đến Hero	Beginner	3	10	Lâm Huệ Trung	⋮⋮⋮
Đang hoạt động	Kiến Thức Nhập Môn IT	Beginner	1	2	Lâm Huệ Trung	⋮⋮⋮
Đang hoạt động	Lập Trình JavaScript Cơ Bản	Beginner	3	9	Lâm Huệ Trung	⋮⋮⋮
Đang hoạt động	Làm việc với Terminal & Ubuntu	Beginner	2	3	Lâm Huệ Trung	⋮⋮⋮
Đang hoạt động	Figma cơ bản	Beginner	2	8	Lâm Huệ Trung	⋮⋮⋮

Showing 1 to 7 of 7 entries

Back Copyright © Your Website 2024

Hình 4. 15 Trang danh sách khoá học

Xem chi tiết khoá học: Xem được thông tin chi tiết của khoá học.

Node & ExpressJS

Dang hoạt động

Course information

Category: Programming
Course name: Node & ExpressJS
Author: Lâm Huệ Trung

Description: Học Back-end với Node & ExpressJS framework, hiểu các khái niệm khi làm Back-end và xây dựng RESTful API cho trang web.

Benefits:

- + Nắm chắc lý thuyết chung trong việc xây dựng web
- + Xây dựng một ứng dụng Back-end thực tế
- + Nắm chắc kỹ thuật về API và RESTful API
- + Nắm chắc khái niệm về giao thức HTTP
- + Học được cách tổ chức code trong thực tế
- + Biết cách làm việc với Mongoose, MongoDB trong NodeJS
- + Biết cách xây dựng API theo chuẩn RESTful API
- + Được chia sẻ kinh nghiệm làm việc thực tế

Level: Intermediate

Old price: 0

New price: 0

Total lessons: 14

Total chapters: 2

Duration: 02:40:29

Created at: Wed Dec 18 2024 18:24:40 GMT+0000 (Coordinated Universal Time)

Updated at: Sat Dec 28 2024 15:38:09 GMT+0000 (Coordinated Universal Time)

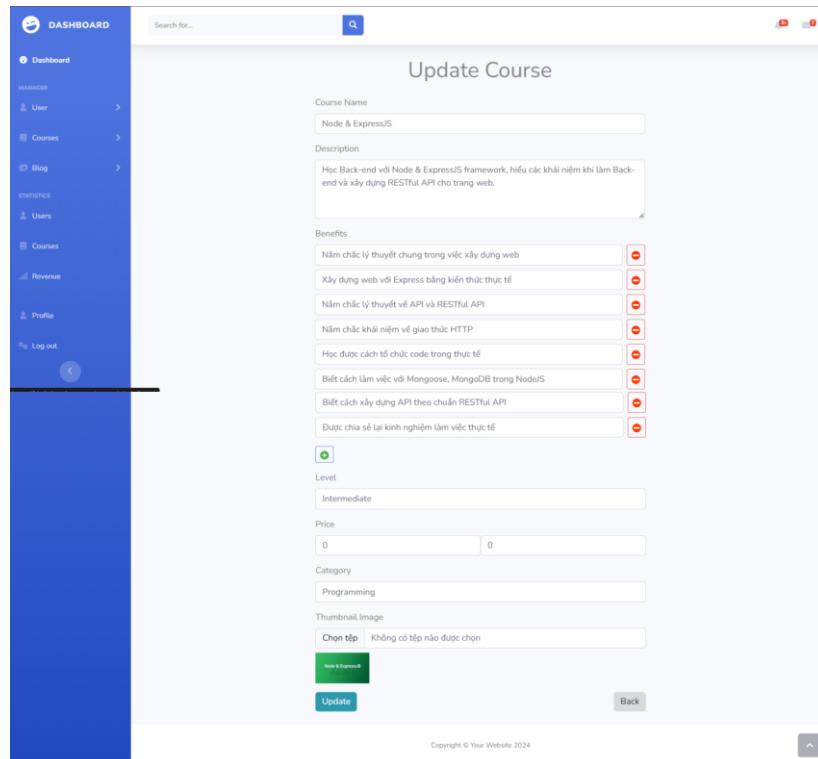
Statistical

Number of subscribers: 0

Back Copyright © Your Website 2024

Hình 4. 16 Trang thông tin chi tiết khoá học

Cập nhật thông tin khoá học: Cập nhật thông tin cho khoá học nếu có sai sót trong quá trình thêm hoặc có sự thay đổi.



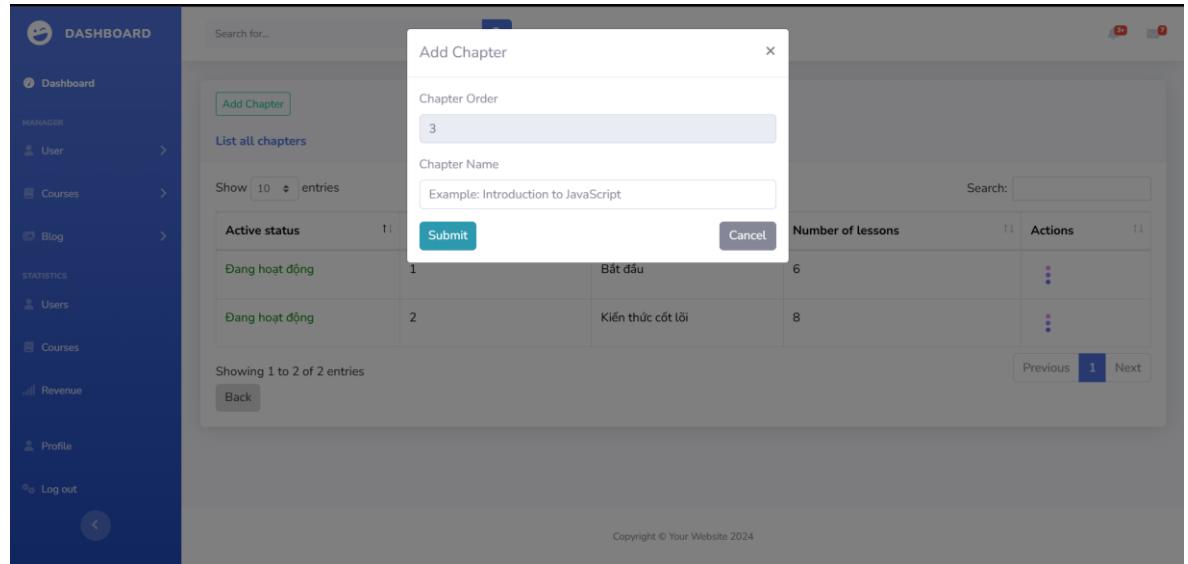
Hình 4. 17 Trang cập nhật thông tin khoá học

Xem tất cả chương học thuộc khoá học: Hiển thị danh sách tất cả các chương học thuộc khoá học, ngoài ra còn có các chức năng như thêm, sửa, xoá, vô hiệu hoá/ khôi phục chương học và xem khoá học thuộc chương.

Active status	Chapter order	Name chapter	Number of lessons	Actions
Dang hoạt động	1	Bắt đầu	6	⋮
Dang hoạt động	2	Kiến thức cốt lõi	8	⋮

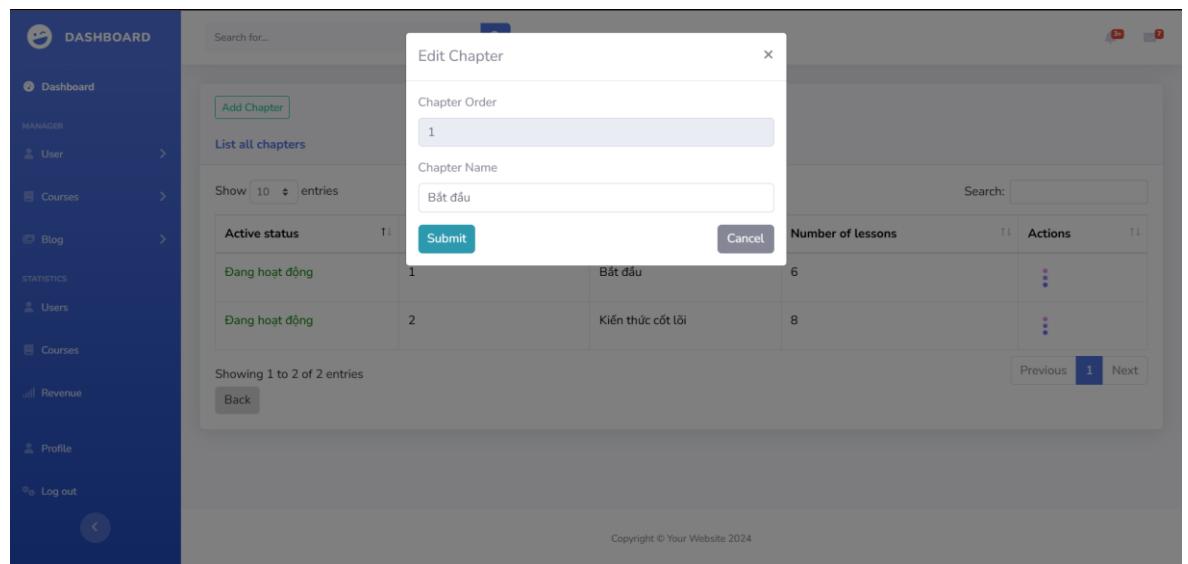
Hình 4. 18 Trang danh sách chương học

Thêm chương học: Thêm chương học mới cho khóa học.



Hình 4. 19 Thêm chương học

Cập nhật chương học: Cập nhật lại tên chương học nếu có sự sai sót trong quá trình tạo.



Hình 4. 20 Trang cập nhật chương học

Xem danh sách bài học: Hiển thị danh sách tất cả các bài học thuộc chương, các chức năng như thêm, sửa, xoá, vô hiệu hoá/ khôi phục và xem chi tiết.

Xây dựng website khoá học online bằng NodeJs & Express

The screenshot shows a dashboard interface for managing a course. On the left sidebar, there are several menu items: Dashboard, Manager (User, Courses, Blog), Statistics (Users, Courses, Revenue), Profile, and Log out. The main content area is titled 'List all lessons'. It features a search bar at the top right and a table below it. The table has columns for Active status, Lesson Order, Name Lesson, duration, and Actions. There are 6 entries listed:

Active status	Lesson Order	Name Lesson	duration	Actions
Đang hoạt động	1	Lời khuyên trước khóa học	08:03	[More]
Đang hoạt động	2	HTTP protocol	15:36	[More]
Đang hoạt động	3	SSR & CSR	12:38	[More]
Đang hoạt động	4	Install Node	1:58	[More]
Đang hoạt động	5	Add git repo	2:39	[More]
Đang hoạt động	6	Install Morgan	5:22	[More]

At the bottom of the table, it says 'Showing 1 to 6 of 6 entries' and has 'Previous' and 'Next' buttons. The footer of the page includes a copyright notice: 'Copyright © Your Website 2024'.

Hình 4. 21 Danh sách bài học thuộc chương

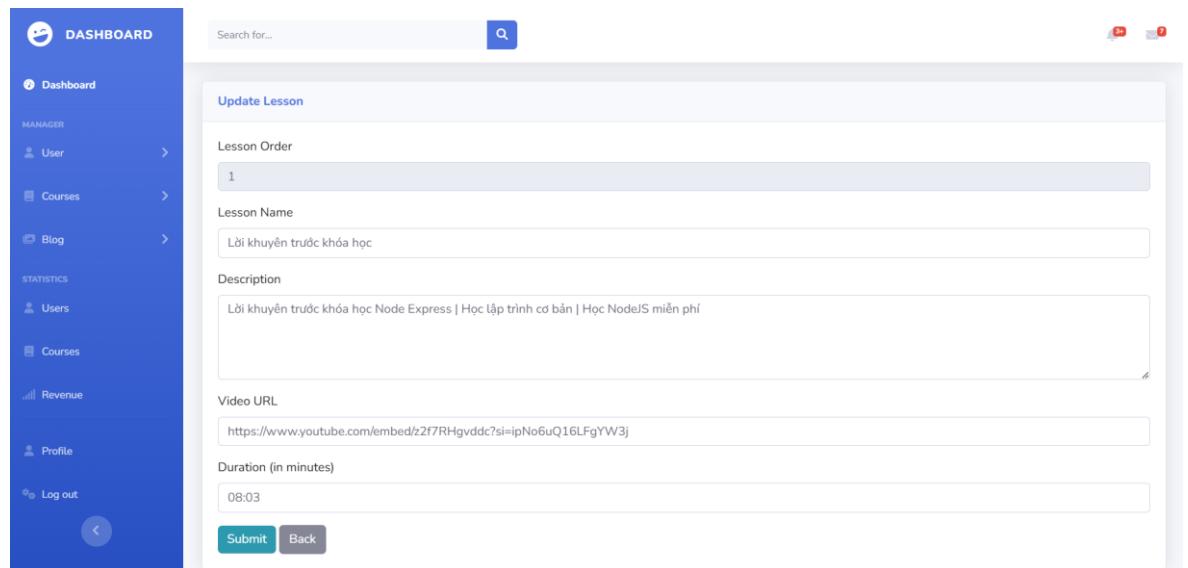
Thêm bài học: Thêm bài học mới thuộc chương với id được truyền vào từ url.

The screenshot shows a 'Add New Lesson' form. The sidebar on the left is identical to the one in the previous screenshot. The main form has fields for Lesson Order (containing '7'), Lesson Name (empty), Description (empty), Video URL (empty), and Duration (in minutes) (empty). At the bottom are 'Submit' and 'Back' buttons.

Hình 4. 22 Giao diện thêm bài học

Cập nhật bài học: Cập nhật thông tin bài học nếu có sai sót trong quá trình tạo hoặc nếu có sự thay đổi bài học.

Xây dựng website khoá học online bằng NodeJs & Express



Hình 4. 23 Trang cập nhật bài học

Xem thông tin chi tiết khoá học: Xem đầy đủ thông tin của bài học đã tạo.

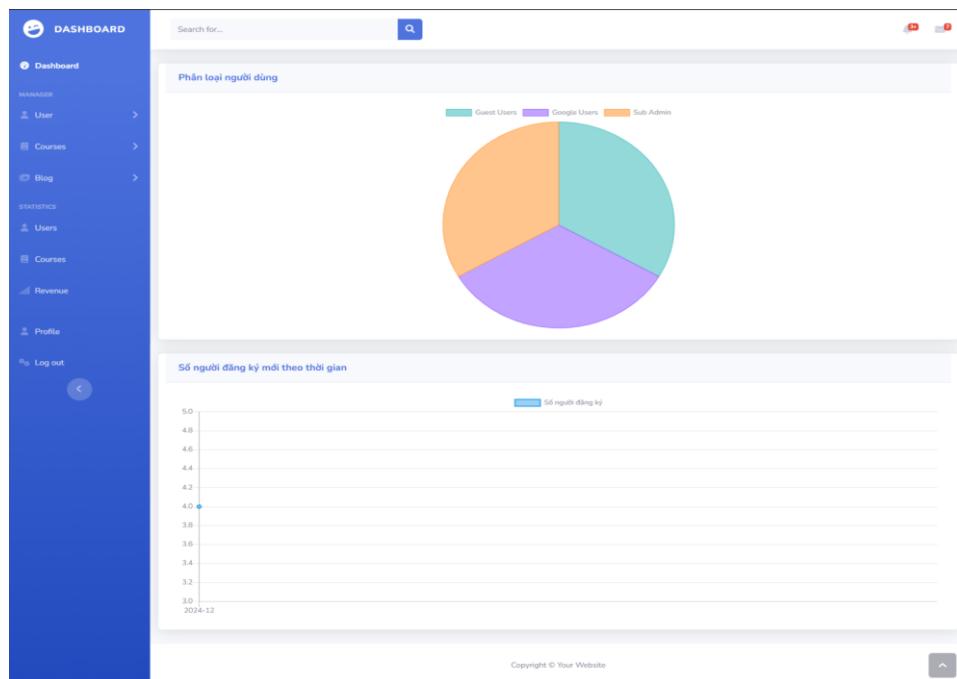
Hình 4. 24 Trang thông tin bài học

Chức năng xoá được xây dựng trong phần khoá học như sau: Nếu xoá khoá học thì sẽ thực hiện xoá tất cả chương và bài học thuộc khoá học đó. Nếu xoá chương học thì sẽ xoá tất cả bài học của chương đó.

Chức năng vô hiệu hoá/ khôi phục: Nếu vô hiệu hoá khoá học sẽ vô hiệu

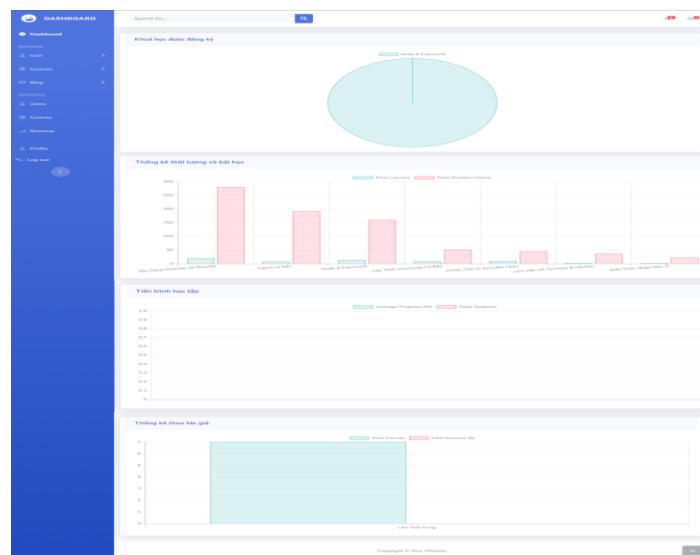
hoá tất cả chương học và bài học của khoá học. Nếu khoá học muốn khôi phục thì phải khôi phục chương học và khoá học trước.

Thông kê người dùng: Bao gồm thống kê về số lượng tài khoản người dùng theo loại tài khoản (user, google, sub admin) và thống kê về số lượng đăng ký theo thời gian. Sử dụng biểu đồ tròn và biểu đồ đường kết hợp với biểu đồ cột.



Hình 4. 25 Trang thống kê người dùng

Thông kê khoá học: Bao gồm thống kê số lượng đăng ký của khoá học, thống kê thời lượng và bài học, tiến trình học tập, thống kê theo tác giả.



Hình 4. 26 Trang thống kê khoá học

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1 Kết quả đạt được

Website khoá học self-learning online được xây dựng thành công và đáp ứng được các yêu cầu đề ra. Ứng dụng có khả năng xử lý nhanh chóng, giao diện thân thiện và dễ sử dụng.

5.1.2 Những đóng góp mới

Website khoá học self-learning online là một website hữu ích, góp phần nâng cao hiệu quả học tập và nâng cao kiến thức của sinh viên. Website có những đóng góp mới sau:

Sử dụng các công nghệ hiện đại như Node.js, Express.js, MongoDB và theo MVC kết hợp với nhiều công nghệ như JWT, CQRS, mã hoá Crypto, Bootstrap giúp ứng dụng có khả năng xử lý nhanh chóng và có khả năng mở rộng cao.

Thiết kế giao diện đơn giản, thân thiện và dễ sử dụng, giúp người dùng dễ dàng tìm thấy các chức năng cần thiết.

5.2 Hướng phát triển

Website khoá học self-learning online có thể được phát triển thêm theo các hướng sau:

Cải thiện các chức năng đã có mặc trên trang website và hoàn thiện một số chức năng còn gian dở như Quản lý bài viết, bình luận, thêm ghi chú, thống kê danh thu, thanh toán, v.v.

Cải thiện khả năng bảo mật của website.

Cải thiện chức năng đánh giá tiến độ và tích hợp thêm github action và vnoij để đánh giá tiến độ tốt hơn.

Tối ưu hóa hiệu năng của website để có thể xử lý được nhiều yêu cầu hơn.

Việc phát triển ứng dụng theo các hướng trên sẽ giúp ứng dụng đáp ứng được nhu cầu sử dụng ngày càng cao của người dùng.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] TopDev, "TopDev," CQRS pattern là gì? Ví dụ dễ hiểu về CQRS Pattern, 2024. [Online]. Available: <https://topdev.vn/>.
- [2] C. L. V. Tiên, "VIETNIX," Mô hình MVC là gì? Ví dụ và cách ứng dụng MVC vào lập trình, 2024. [Online]. Available: <https://vietnix.vn/>.
- [3] P. Thuan, "VIBLO," Authentication with Google OAuth using Nodejs, Passportjs, MongoDB, 2018. [Online]. Available: <https://viblo.asia/p/authentication-with-google-oauth-using-nodejs-passportjs-mongodb-gAm5yqAV5db>.
- [4] npmjs, "npmjs.com," sass, 2024. [Online]. Available: <https://www.npmjs.com/package/sass>.
- [5] npmjs, "npmjs.com," Express Handlebars, 2024. [Online]. Available: <https://www.npmjs.com/package/express-handlebars>.
- [6] npmjs, "https://www.npmjs.com/", crypto-js, 2024. [Online]. Available: <https://www.npmjs.com/package/crypto-js>.
- [7] Nodejs, "Nodejs.org," Node.js documentation, 2024. [Online]. Available: [https://nodejs.org/docs/latest/api/documentation.html..](https://nodejs.org/docs/latest/api/documentation.html)
- [8] MongoDB, "MongoDB.com," MongoDB Documentation, 2024. [Online]. Available: <https://www.mongodb.com/docs/..>
- [9] T. V. Minh, "VIBLO," Tìm hiểu về json web token (JWT), 2016. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-json-web-token-jwt-7rVRqp73v4bP>.
- [10] P. T. N. Mai, "VIBLO," Bootstrap là gì? Giới thiệu về Bootstrap, 2016. [Online]. Available: <https://viblo.asia/p/bai-1-bootstrap-la-gi-gioi-thieu-ve-bootstrap-DzVkpLbDknW>.
- [11] Expressjs, "Expressjs.com," Express.js API Reference, 2024. [Online]. Available: <https://expressjs.com/en/5x/api.html..>

