

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ 1 , NĂM HỌC 2023 – 2024**

XÂY DỰNG ỨNG DỤNG QUẢN LÝ

ĐỀ CƯƠNG HỌC PHẦN

CHO BỘ MÔN CÔNG NGHỆ THÔNG TIN

*Giáo viên hướng dẫn:*

TS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*

Họ tên: Trần Thái Hưng

MSSV: 110120031

Lớp: DA20TTB

***Trà Vinh, tháng 01 năm 2024***

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ 1 , NĂM HỌC 2023 – 2024**

XÂY DỰNG ỨNG DỤNG QUẢN LÝ

ĐỀ CƯƠNG HỌC PHẦN

CHO BỘ MÔN CÔNG NGHỆ THÔNG TIN

*Giáo viên hướng dẫn:*TS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*

Họ tên: Trần Thái Hưng

MSSV: 110120031

Lớp: DA20TTB

***Trà Vinh, tháng 01 năm 2024***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Trước hết, em xin gửi lời cảm ơn đến toàn thể quý thầy cô, giảng viên Trường Đại học Trà Vinh, đặc biệt là các thầy cô ở Khoa Kỹ thuật & Công nghệ, bộ môn Công nghệ thông tin, đã tạo điều kiện tốt nhất để em hoàn thành trọn vẹn bài báo cáo này.

Tiếp theo, tôi xin tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Bảo Ân – Giảng viên Khoa Kỹ thuật & Công nghệ, Trường Đại học Trà Vinh, trong quá trình giảng dạy đã vô cùng tâm huyết trong việc truyền đạt kiến thức đến em nói riêng và sinh viên Khoa Kỹ thuật & Công nghệ nói chung.

Trong bài báo cáo, do lượng kiến thức và kinh ngiệm còn khiêm tốn, thời gian nghiên cứu ngắn nên vẫn còn một số sai sót nhỏ không đáng kể. Do đó 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 khoẻ.

Em xin chân thành cảm ơn!

**MỤC LỤC**

Trà Vinh, ngày ….. tháng …… năm ……

**Sinh viên thực hiện**

(Ký tên và ghi rõ họ tên)

[DANH MỤC TỪ VIẾT TẮT 6](#_Toc155503253)

[DANH MỤC HÌNH ẢNH 6](#_Toc155503254)

[BẢNG BIỂU 9](#_Toc155503255)

[TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH 10](#_Toc155503256)

[MỞ ĐẦU 11](#_Toc155503257)

[CHƯƠNG 1: TỔNG QUAN 13](#_Toc155503258)

[1.1 Trình bày bài toán 13](#_Toc155503259)

[1.2 Đề xuất hướng giải quyết 13](#_Toc155503260)

[1.3 Kết quả đạt được 13](#_Toc155503261)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 14](#_Toc155503262)

[2.1 Cơ sở lý thuyết 14](#_Toc155503263)

[2.1.1 Tổng quan về Node.js 14](#_Toc155503264)

[2.1.2 Tổng quan về framework Express.js 16](#_Toc155503265)

[2.1.3 Tổng quan MongoDB 21](#_Toc155503266)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 25](#_Toc155503267)

[3.1 Đặc tả ứng dụng 25](#_Toc155503268)

[3.2 Phân tích và thiết kế cơ sở dữ liệu 27](#_Toc155503269)

[3.3 Thiết kế kiến trúc ứng dụng 33](#_Toc155503270)

[3.4 Thiết kế API 33](#_Toc155503271)

[3.4.1 Định tuyến API trong Express 33](#_Toc155503272)

[3.4.2 Định tuyến trong router 'admin.Routes' 34](#_Toc155503273)

[3.4.3 Định tuyến trong router 'project.Routes' 36](#_Toc155503274)

[3.4.4 Định tuyến trong router 'user.Routes' 38](#_Toc155503275)

[3.4.5 Các định tuyến router khác 38](#_Toc155503276)

[3.5 Thiết kế giao diện 39](#_Toc155503277)

[3.5.1 Thiết kế khung giao diện cho người bằng Handlebars.js 39](#_Toc155503278)

[3.5.2 Thiết kế khung giao diện cho admin bằng Handlebars.js 40](#_Toc155503279)

[3.6 Giao tiếp giữa Front-End và Back-End 40](#_Toc155503280)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 41](#_Toc155503281)

[4.1 Chức năng của người dùng 41](#_Toc155503282)

[4.1.1 Chức năng xem trang chủ 41](#_Toc155503283)

[4.1.2 Chức năng đăng nhập 41](#_Toc155503284)

[4.1.3 Chức năng xem chương trình của người dùng 42](#_Toc155503285)

[4.1.4 Chức năng tạo học phần 43](#_Toc155503286)

[4.1.5 Chức năng nhập liệu học phần 43](#_Toc155503287)

[4.1.6 Chức năng quản lý học phần 48](#_Toc155503288)

[4.1.7 Chức năng nhập liệu kéo thả 50](#_Toc155503289)

[4.2 Chức năng của admin 50](#_Toc155503290)

[4.2.1 Chức năng chỉnh sửa chương trình 50](#_Toc155503291)

[4.2.2 Chức năng quản lý PLO của admin 51](#_Toc155503292)

[4.2.3 Chức năng quản lý PO của admin 52](#_Toc155503293)

[4.2.4 Chức năng quản lý Matrix PLO-PO của admin 52](#_Toc155503294)

[4.2.5 Chức năng quản lý mẫu của admin 53](#_Toc155503295)

[4.2.6 Chức năng quản lý người dùng của admin 54](#_Toc155503296)

[4.2.7 Chức năng quản lý từ gợi ý của admin 54](#_Toc155503297)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 56](#_Toc155503298)

[5.1 Kết quả đạt được 56](#_Toc155503299)

[5.2 Hạn chế 56](#_Toc155503300)

[5.3 Hướng phát triển 57](#_Toc155503301)

[DANH MỤC TÀI LIỆU THAM KHẢO 58](#_Toc155503302)

# DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| API | Application Programming Interface (Giao Diện Lập Trình Ứng Dụng) |
| CPU | Central Processing Unit (Bộ Xử Lý Trung Tâm) |
| ECMAScript | European Computer Manufacturers Association Script |
| Event-driven | Kiểu Lập Trình Dựa Trên Sự Kiện |
| HTTP | Hypertext Transfer Protocol (Giao Thức Truyền Tải Siêu Văn Bản) |
| I/O | Input/Output (Nhập/Xuất) |
| MEAN | MongoDB, Express.js, AngularJS, Node.js |
| PLO | Chuẩn đầu ra chương trình |
| PO | Mục tiêu đào tạo |
| RDBMS | Relational Database Management System (Hệ Thống Quản Lý Cơ Sở Dữ Liệu Quan Hệ) |
| SEAN | SQL, Express.js, AngularJS, Node.js |
| SPA | Single-Page Application (Ứng Dụng Một Trang) |
| URL | Uniform Resource Locator |
| UI | User Interface |

# DANH MỤC HÌNH ẢNH

[Hình 1. Node.js là gì? [3] 14](#_Toc155507993)

[Hình 2. Express JS là gì ? [7] 17](#_Toc155507994)

[Hình 3. Sử dụng Middleware bắt lỗi 19](#_Toc155507995)

[Hình 4. Tạo thư mục tĩnh với Express.js 19](#_Toc155507996)

[Hình 5. MongoDB là gì? [10] 21](#_Toc155507997)

[Hình 6. Sơ đồ usecase 25](#_Toc155507998)

[Hình 7. Phân tích và thiết kế cơ sở dữ liệu 27](#_Toc155507999)

[Hình 8. Kiến trúc Client Server 33](#_Toc155508000)

[Hình 9. Định tuyến API trong Express 34](#_Toc155508001)

[Hình 10. Định tuyến đến trang chủ admin 34](#_Toc155508002)

[Hình 11. Các định tuyến quản lý chương trình học 35](#_Toc155508003)

[Hình 12. Các định tuyến cho quản lý người dùng. 36](#_Toc155508004)

[Hình 13. Các định tuyến quản lý khung và từ gợi ý 36](#_Toc155508005)

[Hình 14. Các định tuyến quản lý học phần 37](#_Toc155508006)

[Hình 15. Các định tuyến của router người dùng 38](#_Toc155508007)

[Hình 16. Định tuyến hiển thị chương trình cho người dùng 38](#_Toc155508008)

[Hình 17. Định tuyến hiển thị trang chủ cho người dùng 38](#_Toc155508009)

[Hình 18. Định tuyến thực hiện chức năng login 38](#_Toc155508010)

[Hình 19. Định tuyến thực hiện chức năng logout 39](#_Toc155508011)

[Hình 20. Thiết kế khung giao diện cho người. 39](#_Toc155508012)

[Hình 21. Thiết kế khung giao diện admin. 40](#_Toc155508013)

[Hình 22. Giao diện trang chủ 41](#_Toc155508014)

[Hình 23. Giao diện đăng nhập. 41](#_Toc155508015)

[Hình 24. Giao diện xem chương trình của người dùng 42](#_Toc155508016)

[Hình 25. Giao diện tạo học phần 43](#_Toc155508017)

[Hình 26. Giao diện nhập liệu học phần. 43](#_Toc155508018)

[Hình 27 Người dùng sử dụng chức năng đề xuất tên học phần. 44](#_Toc155508019)

[Hình 28. Mã số học phần được cập nhật. 44](#_Toc155508020)

[Hình 29. Học phần tiên quyết được cập nhật. 44](#_Toc155508021)

[Hình 30. Phần hai giao diện nhập liệu học phần. 44](#_Toc155508022)

[Hình 31. Người dùng thực hiện chức năng thêm tài liệu loại học liệu. 45](#_Toc155508023)

[Hình 32. Người dùng thực hiện chức năng xóa 1 giáo trình. 45](#_Toc155508024)

[Hình 33. Phần ba giao diện nhập liệu học phần . 45](#_Toc155508025)

[Hình 34. Người dùng sử dụng chức năng đề xuất. 45](#_Toc155508026)

[Hình 35. Phần bốn giao diện nhập liệu học phần 46](#_Toc155508027)

[Hình 36. Phần năm giao diện nhập liệu học phần 46](#_Toc155508028)

[Hình 37 Người dùng thực hiện chức năng thêm cấu trúc 46](#_Toc155508029)

[Hình 38. Người dùng sử dụng chức năng đề xuất 47](#_Toc155508030)

[Hình 39. Phần bảy giao diện nhập liệu học phần 47](#_Toc155508031)

[Hình 40. Phần tám giao diện nhập liệu học phần 47](#_Toc155508032)

[Hình 41. Chức năng đề xuất nhập liệu giáo viên 48](#_Toc155508033)

[Hình 42. Nút lưu và nút xuất file PDF 48](#_Toc155508034)

[Hình 43. Giao diện danh sách học phần 48](#_Toc155508035)

[Hình 44. Chức năng xuất PDF chuẩn đầu ra học phần 49](#_Toc155508036)

[Hình 45. Chức năng xuất PDF đề cương 49](#_Toc155508037)

[Hình 46. Chức năng nhập liệu kéo thả. 50](#_Toc155508038)

[Hình 47. Kết quả nhập liệu kéo thả. 50](#_Toc155508039)

[Hình 48. Giao diện chỉnh sửa chương trình của admin 50](#_Toc155508040)

[Hình 49. Giao diện quản lý PLO 51](#_Toc155508041)

[Hình 50. Thêm mới một PLO 51](#_Toc155508042)

[Hình 51. Giao diện quản lý mục tiêu đào tạo của admin 52](#_Toc155508043)

[Hình 52. Giao diện chỉnh sửa Matrix PLO-PO của admin 52](#_Toc155508044)

[Hình 53. Admin check vào PO4 và PO5,PO6 53](#_Toc155508045)

[Hình 54. Kết quả sau khi admin check PO 53](#_Toc155508046)

[Hình 55. Admin chỉnh sửa quy định về học vụ 53](#_Toc155508047)

[Hình 56. Kết quả mẫu học phần sau khi thay đổi 54](#_Toc155508048)

[Hình 57. Giao diện quản lý người dùng của admin 54](#_Toc155508049)

[Hình 58. Admin thực hiện chức năng thêm từ gợi ý là “tích phân” 55](#_Toc155508050)

[Hình 59. Kiểm tra kết quả thêm gợi ý 55](#_Toc155508051)

# BẢNG BIỂU

[Bảng 1. Mô tả bảng học phần (HocPhan) 27](#_Toc155508052)

[Bảng 2. Mô tả bảng chuẩn đầu ra học phần (CDR\_HocPhan) 28](#_Toc155508053)

[Bảng 3. Mô tả bảng khung (template) 29](#_Toc155508054)

[Bảng 4. Mô tả người dùng (users) 30](#_Toc155508055)

[Bảng 5. Mô tả bảng chuẩn đầu ra chương trình (ChuongTrinh) 30](#_Toc155508056)

[Bảng 6. Mô tả bảng chuẩn đầu ra chương trình (CDR\_CT) 31](#_Toc155508057)

[Bảng 7. Mô tả bảng mục tiêu đào tạo (MucTieuCTDT) 31](#_Toc155508058)

[Bảng 8. Mô tả bảng đáp ứng chương trình (Dapung\_CT) 32](#_Toc155508059)

[Bảng 9. Mô tả bảng đáp ứng chuẩn đầu ra (Dapung\_CDR) 32](#_Toc155508060)

[Bảng 10 Mô tả bảng tạo (Create) 32](#_Toc155508061)

# TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

**Về lý thuyết:**

* Tìm hiểu nghiệp vụ có liên quan: Nghiên cứu và phân tích các quy trình, thủ tục nghiệp vụ liên quan đến đề cương học phần.
* Tìm hiểu Node.js: Khám phá và học về Node.js, một nền tảng JavaScript server-side.
* Tìm hiểu Express.js: Nắm vững kiến thức về Express.js, một framework phổ biến của Node.js.
* Tìm hiểu Handlebars.js: Tìm hiểu về Handlebars.js, một thư viện template cho JavaScript.
* Tìm hiểu MongoDB: Học về MongoDB, một cơ sở dữ liệu NoSQL.

**Về thực nghiệm:**

* Thiết kế cơ sở dữ liệu: Xây dựng cấu trúc cơ sở dữ liệu phù hợp cho ứng dụng.
* Thiết kế schema tương tác với MongoDB: Tạo schema để quản lý và tương tác với dữ liệu trong MongoDB.
* Thiết kế view nhập liệu: Phát triển giao diện người dùng cho việc nhập liệu vào hệ thống.
* Nghiên cứu tạo đề xuất từ các nguồn trên mạng: Tìm hiểu và áp dụng các phương pháp để tạo ra các đề xuất hữu ích từ dữ liệu trực tuyến.
* Tạo đề xuất: Phát triển chức năng đề xuất trong ứng dụng.
* Xuất file PDF: Tích hợp tính năng xuất dữ liệu ra định dạng PDF.

# MỞ ĐẦU

1. **Lý do chọn đề tài**

Trong khuôn khổ giáo dục đại học, học phần được định nghĩa như là một đơn vị học thuật cụ thể, chứa đựng một khối lượng kiến thức và kỹ năng nhất định. Nó được cấu trúc để truyền đạt cho sinh viên trong một kỳ học hoặc trong một khoảng thời gian xác định. Học phần được thiết kế nhằm đạt được mục tiêu giáo dục đã được định trước, qua việc kết hợp giảng dạy lý thuyết, thực hành, thảo luận và các hoạt động học tập khác. Mỗi học phần thường đi kèm với một đề cương học phần mô tả chi tiết nội dung học, mục tiêu, phương pháp đánh giá và tài liệu tham khảo cần thiết.

Tuy nhiên, trong bối cảnh hiện tại của Bộ môn Công nghệ thông tin, quá trình thiết kế và quản lý đề cương học phần đang đối diện với thách thức. Sự không thống nhất trong định dạng và phương pháp nhập liệu của đề cương học phần giữa các giảng viên đã dẫn đến một sự không nhất quán. Điều này tạo ra khó khăn trong việc quản lý và cập nhật thông tin, cũng như ảnh hưởng tiêu cực đến trải nghiệm giáo dục của sinh viên. Vấn đề này đòi hỏi một giải pháp cấp thiết để bảo đảm chất lượng và hiệu quả của quá trình giáo dục.

Do đó, việc phát triển một ứng dụng quản lý đề cương học phần trở nên cần thiết. Mục đích của ứng dụng này là tạo ra một hệ thống thống nhất để chuẩn hóa quy trình soạn thảo và quản lý đề cương học phần. Ứng dụng sẽ cung cấp một giao diện người dùng thân thiện và dễ sử dụng, nhằm hỗ trợ giảng viên trong việc tạo mẫu đề cương chuẩn và nhập liệu một cách hiệu quả. Đó là lý do tôi quyết định chọn đề tài "Xây dựng ứng dụng quản lý đề cương học phần cho Bộ môn Công nghệ thông tin".

1. **Mục đích**

Mục đích chính của việc xây dựng ứng dụng quản lý đề cương học phần cho Bộ môn Công nghệ thông tin được định rõ như sau:

* Chuẩn hóa Mẫu Đề Cương: Trong quá trình biên soạn đề cương học phần, sự khác biệt về định dạng giữa các giảng viên là một vấn đề nổi bật. Ứng dụng sẽ giải quyết vấn đề này bằng cách cung cấp một giao diện người dùng cho phép tạo ra các mẫu đề cương chuẩn hóa, nhằm đạt được sự nhất quán trong toàn bộ bộ môn.
* Hỗ trợ Trong Quá Trình Nhập Liệu: Ứng dụng sẽ tận dụng cơ sở dữ liệu hiện có để đề xuất từ ngữ và cấu trúc ngữ pháp trong quá trình nhập liệu, qua đó giúp tối ưu hóa và tăng hiệu quả trong việc soạn thảo đề cương.
* Hỗ trợ in ấn: Ứng dụng sẽ cho phép xuất đề cương ra các định dạng như Word hoặc PDF, từ đó hỗ trợ dễ dàng trong việc in ấn và phân phối. Tính năng này sẽ cải thiện đáng kể khả năng tiếp cận và phân phối đề cương học phần cho sinh viên và giảng viên.

1. **Đối tượng**

Đối tượng nghiên cứu trong dự án này chủ yếu bao gồm:

* Giảng viên: Họ là người chịu trách nhiệm biên soạn và cập nhật đề cương học phần.
* Quản trị viên tại bộ môn: Họ sẽ sử dụng ứng dụng để thay đổi mẫu trong hệ thống.

1. **Phạm vi nghiên cứu**

Phạm vi nghiên cứu của dự án này tập trung vào Khoa Công nghệ thông tin tại Trường Đại học Trà Vinh.

# CHƯƠNG 1: TỔNG QUAN

## Trình bày bài toán

Dự án này tập trung vào việc giải quyết vấn đề không thống nhất trong thiết kế và quản lý đề cương học phần tại Bộ môn Công nghệ thông tin. Sự không nhất quán hiện tại trong quy trình làm việc này gây ra khó khăn trong việc cập nhật và quản lý thông tin một cách chính xác. Bên cạnh đó, nhu cầu về một phương pháp nhập liệu chuẩn hóa và hiệu quả cũng cần được đáp ứng.

## Đề xuất hướng giải quyết

Nhằm giải quyết những thách thức này, dự án sẽ áp dụng các công nghệ sau:

* **Node.js và Express.js:** Sử dụng Node.js làm nền tảng chính cho server và Express.js, một framework mạnh mẽ, để xây dựng ứng dụng web và API. Sự kết hợp này hứa hẹn tạo ra một hệ thống quản lý đề cương học phần hiệu quả và linh hoạt.
* **MongoDB:** Áp dụng MongoDB, một cơ sở dữ liệu NoSQL, cho việc lưu trữ dữ liệu đề cương, nhằm mục tiêu tối ưu hóa quản lý dữ liệu động và phức tạp.
* **Puppeteer:** Puppeteer sẽ được triển khai để chuyển đổi đề cương học phần sang định dạng PDF, hỗ trợ việc in ấn và phân phối tài liệu một cách tiện lợi.
* Tính năng 'Autocomplete' được đề xuất để tối ưu hóa nhập liệu. Hệ thống này tự động đề xuất từ và cụm từ.
* Sử dụng Handlebars.js giao diện người dùng (UI) trong các dự án.

## Kết quả đạt được

* Xây dựng giao diện nhập liệu
* PDF đề cương học phần.
* Đề xuất nhập liệu: tính năng 'Autocomplete' trong dự án này được nâng cấp dựa trên các nghiên cứu và phát triển từ nguồn. [1]

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

## Cơ sở lý thuyết

1. Tổng quan về Node.js

**A. Giới thiệu**

Node.js là một môi trường thực thi JavaScript, mở ra khả năng sử dụng ngôn ngữ này không chỉ trong môi trường trình duyệt mà còn trên máy chủ và môi trường máy tính khác. JavaScript, thường được áp dụng để tương tác với DOM trong trình duyệt, tìm thấy sự mở rộng này trong Node.js, tạo ra sự linh hoạt đáng kể cho việc phát triển ứng dụng web chạy trên máy chủ. [2]

A group of logos on a black background

Description automatically generated

Hình . Node.js là gì? [3]

Node.js, một nền tảng phát triển ứng dụng web, được xây dựng trên nền tảng của trình thực thi JavaScript V8 do Google phát triển. V8, viết hoàn toàn bằng ngôn ngữ C++, có khả năng biên dịch mã JavaScript sang mã máy, cho phép thực thi mã này trực tiếp trên phần cứng máy tính. Sự linh hoạt này cho phép Node.js chạy mã JavaScript không chỉ trong môi trường trình duyệt mà còn trên máy chủ và các thiết bị khác, mở ra cánh cửa cho việc phát triển ứng dụng đa nền tảng. [2]

Node.js, thông qua sự tích hợp với nhân V8, mở rộng khả năng của JavaScript bằng cách cung cấp các tính năng không thể thực hiện trong trình duyệt web thông thường do lý do bảo mật. Những tính năng này bao gồm quản lý hệ thống tệp địa phương, khả năng mở, đọc và xóa tệp. Do đó, Node.js không chỉ đóng vai trò như một môi trường chạy JavaScript trên máy chủ, mà còn mở rộng phạm vi ứng dụng của ngôn ngữ này ra ngoài môi trường trình duyệt. [2]

**B. Một số câu lệnh cơ bản**

* **node <filename>:** Thực thi một chương trình Node.js từ một tệp được chỉ định.
* **npm init:** Thiết lập một dự án Node.js mới và tạo một tệp package.json.
* **npm install <packageName>:** Cài đặt một gói từ kho lưu trữ npm trên máy cục bộ.
* **npm install -g <packageName>:** Cài đặt một gói toàn cầu, làm cho nó có sẵn cho tất cả các dự án.
* **require('<moduleName>'):** Nhập một mô-đun hoặc gói trong tệp Node.js của bạn.
* **console.log():** In ra màn hình console để gỡ lỗi.
* **process.env:** Truy cập biến môi trường, thường được sử dụng cho dữ liệu nhạy cảm.
* **module.exports:** Xuất một mô-đun để làm cho các chức năng của nó có sẵn để sử dụng trong các tệp khác.
* **\_\_dirname:** Lấy đường dẫn của thư mục hiện tại.
* **fs.readFile():** Đọc dữ liệu từ một tệp một cách không đồng bộ.

**C. Ưu nhược điểm của Node.js**

**Ưu điểm:**

* **Xử lý không đồng bộ (Asynchronous):** Ứng dụng Node.js hoạt động trong một quy trình duy nhất mà không tạo ra luồng mới cho mỗi yêu cầu. Nguyên tắc không đồng bộ I/O được tích hợp trong thư viện tiêu chuẩn, ngăn chặn mã JavaScript bị chặn, và thư viện thường được thiết kế theo mô hình không đồng bộ. [4]
* **Xử lý I/O mà không chặn (non-blocking I/O):** Trong quá trình thực hiện các thao tác I/O như đọc từ mạng, truy cập cơ sở dữ liệu, hoặc hệ thống tệp, Node.js không chặn luồng và tránh lãng phí CPU bằng cách tiếp tục thực hiện thao tác khi nhận được phản hồi. [4]
* **Xử lý hàng nghìn kết nối đồng thời:** Node.js có khả năng xử lý hàng nghìn kết nối đồng thời với một máy chủ duy nhất mà không cần quản lý đồng thời luồng, giảm nguy cơ lỗi và tăng cường hiệu suất. [4]
* **Chia sẻ ngôn ngữ với phía client:** Lập trình viên frontend sử dụng JavaScript cho trình duyệt được áp dụng kiến thức của mình vào việc phát triển mã phía máy chủ bằng Node.js mà không cần học một ngôn ngữ mới. [4]
* **Sử dụng các tiêu chuẩn ECMAScript mới:** Node.js hỗ trợ việc sử dụng các tiêu chuẩn ECMAScript mới mà không cần chờ đợi người dùng cập nhật trình duyệt, cho phép nhà phát triển lựa chọn phiên bản ECMAScript phù hợp với yêu cầu của họ. [4]
* **Hỗ trợ cả hai hệ thống module CommonJS và ES (từ Node.js v12):** Điều này có nghĩa là bạn sử dụng được cả require() và import trong Node.js, mang lại linh hoạt trong việc quản lý mô-đun. [5]

**Nhược điểm:**

Mặc dù có sẵn một lượng đáng kể các module và thư viện cho Node.js, tuy nhiên, không phải tất cả chúng đều được duy trì một cách đồng đều và chất lượng cao. Điều này dẫn đến tình trạng mà người phát triển phải đối mặt với vấn đề không tương thích và sự không đồng nhất trong quá trình cập nhật các module và thư viện. Điều này tạo ra những thách thức đối với tính ổn định và bảo trì của hệ thống.

1. Tổng quan về framework Express.js

**A. Giới thiệu**

Express, một framework được thiết kế cho môi trường Node.js, hướng tới việc đơn giản hóa quá trình phát triển API và cung cấp thêm các tính năng mới mang tính ứng dụng. Với khả năng tổ chức ứng dụng thông qua các lớp trung gian (Middleware) và hệ thống định tuyến (routing), Express đưa ra những tiện ích quan trọng cho việc xử lý đối tượng HTTP trong Node.js. Nó tối ưu trong việc hiển thị chế độ xem HTML động và thiết lập một tiêu chuẩn mở rộng để triển khai ứng dụng một cách thuận lợi. [6]



Hình . Express JS là gì ? [7]

Express là một framework của Node.js được ứng dụng rộng rãi trong xây dựng ứng dụng web. Nó đơn giản hóa quá trình xử lý yêu cầu và phản hồi, tạo điều kiện thuận lợi cho việc phát triển ứng dụng web và API. Thường kết hợp với Node.js, Express cùng nhau tạo thành stack MEAN, bao gồm MongoDB, Express, Angular, và Node.js. Điều này không chỉ thể hiện tính linh hoạt của Express mà còn cho phép sự tùy chọn các thành phần khác, tạo ra các stack như MEBN hoặc SEAN. [6] [8]

Express là sự lựa chọn lý tưởng cho việc xây dựng ứng dụng trang đơn (SPA), nơi nó xuất sắc trong việc cung cấp HTML, CSS, và JavaScript cũng như xây dựng các API. Nó có mức độ học tương đối thấp, giúp nhà phát triển frontend nhanh chóng triển khai server SPA mà không đòi hỏi kiến thức mới nhiều. Không chỉ vậy, Express còn tương thích mạnh mẽ với các tính năng thời gian thực như WebSocket và WebRTC, làm cho nó trở thành lựa chọn phổ biến trong việc xây dựng các ứng dụng yêu cầu tính thời gian thực. [6]

**B. Ưu nhược điểm của Express.js**

**Ưu điểm:**

* Sự đơn giản và dễ sử dụng: Express.js có cú pháp đơn giản và dễ hiểu, giúp lập trình viên dễ dàng nắm bắt và triển khai các tính năng. [6]
* Hỗ trợ Middleware: Express.js cung cấp hệ thống Middleware mạnh mẽ, cho phép thực hiện các chức năng như xác thực, ghi log, nén dữ liệu và xử lý lỗi một cách linh hoạt và dễ dàng. [8]
* Hiệu suất cao: Express.js được xây dựng trên Node.js, nền tảng có hiệu suất cao, giúp xử lý nhanh chóng các yêu cầu web đồng thời và có khả năng mở rộng tốt.
* Khả Năng Mở Rộng: Express.js hỗ trợ một lượng lớn kết nối đồng thời nhờ vào mô hình I/O không chặn, Event-driven, và được thiết kế để có khả năng mở rộng. [6]

**Nhược điểm:**

* Hạn chế trong các tính năng tích hợp: Express.js, mặc dù cung cấp các tính năng cơ bản, nhưng để đạt được những chức năng phức tạp và nâng cao hơn, nhà phát triển thường phải dựa vào các module bổ sung. [6]
* Khó Khăn trong việc học Middleware cao: Việc hiểu và triển khai Middleware một cách hiệu quả dẫn đến trở nên phức tạp, đặc biệt là đối với những người không quen với khái niệm này, đòi hỏi sự tập trung và nắm vững kiến thức về cách tích hợp và tương tác với các Middleware trong môi trường Express.js.[8]

**D. Middleware**

Express tận dụng "Middleware" để phân chia và quản lý các chức năng xử lý yêu cầu trong quá trình phát triển ứng dụng. Middleware có khả năng thực hiện đa dạng nhiệm vụ như ghi log, phục vụ tệp tĩnh, và xác thực người dùng. Mỗi hàm Middleware xử lý một nhiệm vụ cụ thể và chuyển giao đến Middleware tiếp theo trong chuỗi. Điều này giúp làm cho mã nguồn trở nên chuẩn hóa và dễ quản lý hơn. Express cung cấp nhiều Middleware được cộng đồng phát triển, mang đến các tính năng như biên dịch tài nguyên tĩnh, bảo mật, xác thực người dùng, và nhiều chức năng khác. [6]

Middleware trong Express là các hàm được thực thi trước khi đến router handler, thực hiện các chức năng như kiểm tra quyền truy cập, ghi log, xử lý lỗi, và nhiều nhiệm vụ khác. Chúng được truy cập vào các đối tượng yêu cầu (req), đối tượng phản hồi (res), và hàm Middleware tiếp theo trong chuỗi xử lý yêu cầu trả lời của ứng dụng, thường được biểu diễn bằng biến next.

**Một số ví dụ:**

* **Sử dụng Middleware bắt lỗi:**

Khi xảy ra một lỗi trong quá trình xử lý yêu cầu của ứng dụng Express, có một số phương pháp được áp dụng để giải quyết tình huống lỗi này. Trong bối cảnh của Middleware, quá trình định tuyến lỗi thường được thực hiện bằng cách chuyển quyền kiểm soát đến một Middleware đặc biệt chịu trách nhiệm xử lý lỗi (error-handling Middleware). Hành động này giúp phân tách mã xử lý lỗi từ mã chính xử lý yêu cầu, làm cho mã nguồn trở nên dễ đọc và quản lý hơn.

A computer screen shot of a program code

Description automatically generated

Hình . Sử dụng Middleware bắt lỗi

Middleware này được kích hoạt khi xảy ra lỗi trong quá trình xử lý yêu cầu. Chức năng chính của nó là in thông tin lỗi vào console để giám sát và ghi log. Đồng thời, trạng thái phản hồi được thiết lập là 500 (Internal Server Error), làm thông báo rằng có một lỗi nội bộ trên máy chủ. Sau đó, một phản hồi được gửi đi với nội dung "Something went wrong!", giúp thông báo lỗi đến người gửi yêu cầu một cách thích hợp.

* **Tạo thư mục tĩnh:**

A computer screen shot of a computer code

Description automatically generated

Hình . Tạo thư mục tĩnh với Express.js

Middleware “express.static” được triển khai để phục vụ các tài nguyên tĩnh từ thư mục (“public”). Khi có yêu cầu được gửi đến máy chủ với một đường dẫn tương ứng với các tài nguyên tĩnh như hình ảnh, tệp CSS, JavaScript trong thư mục ("public"), Middleware này sẽ cung cấp tài nguyên đó mà không cần thông qua các định tuyến khác của ứng dụng Express.

**C. Routing**

Trong Express, Routing là một cơ chế cho phép định nghĩa các định tuyến để xử lý yêu cầu từ người dùng. Mỗi router kết nối với một hoặc nhiều hàm xử lý, thực thi khi router tương ứng với một URL và một phương thức HTTP cụ thể (như GET, POST, PUT, DELETE, v.v.). [6]

Express sử dụng các phương thức của đối tượng ứng dụng (app) để định nghĩa các định tuyến, chẳng hạn như app.get(), app.post(), app.put(), và app.delete(). Đối số đầu tiên của mỗi phương thức này là đường dẫn, và đối số thứ hai là một hàm callback được gọi khi có một yêu cầu HTTP phù hợp với đường dẫn đó.

**D. Views và templates**

Express.js, một view engine là một công cụ để tạo giao diện người dùng (UI) động cho ứng dụng web. View engine cho phép tạo các trang web động bằng cách kết hợp dữ liệu từ máy chủ và các mẫu (templates) để tạo nội dung HTML hiển thị cho người dùng. Các view engines thường sử dụng cú pháp đặc biệt để chèn dữ liệu động vào các trang web. [6]

**Một số view engines thường được sử dụng trong Express.js:**

* EJS (Embedded JavaScript).
* Pug (trước đây là Jade).
* Handlebars.js.

Handlebars.js, được biết đến là một view engine phổ biến trong cộng đồng Express.js, nổi tiếng với sự đơn giản và khả năng tích hợp dễ dàng vào các dự án web. Trọng tâm chính của Handlebars.js là việc chèn dữ liệu động vào mẫu HTML, tạo ra sự linh hoạt và hiệu quả trong quá trình phát triển các trang web động.

**Ưu và Nhược Điểm của Việc Sử Dụng Handlebars.js trong Express.js:**

Trong môi trường phát triển web sử dụng Express.js, Handlebars.js được đánh giá cao về khả năng tích hợp nhanh chóng, cú pháp gần gũi với HTML, cũng như khả năng tạo các trang web động mà không đòi hỏi logic ứng dụng phức tạp. Hệ thống partials và layout của nó cải thiện khả năng tái sử dụng mã và hỗ trợ việc phát triển giao diện người dùng một cách hiệu quả. Tuy nhiên, Handlebars.js cũng có những hạn chế nhất định, bao gồm việc không hỗ trợ logic ứng dụng phức tạp trong mẫu.

1. Tổng quan MongoDB

**A. Giới thiệu**

MongoDB là một hệ thống cơ sở dữ liệu mạnh mẽ và linh hoạt, nổi bật với khả năng mở rộng và hỗ trợ nhiều tính năng quan trọng như chỉ mục phụ, truy vấn dải, sắp xếp, tổng hợp, và chỉ mục địa lý. Với việc được thiết kế dễ sử dụng và linh hoạt, MongoDB đem lại trải nghiệm phát triển độc đáo. [9]

MongoDB được xây dựng với sự dễ sử dụng là ưu tiên hàng đầu của nó. Sự linh hoạt của nó đến từ việc không có các schema cố định, cho phép nhà phát triển thực hiện các thay đổi trong cấu trúc dữ liệu một cách linh hoạt và nhanh chóng. Với cơ sở dữ liệu hướng tài liệu, MongoDB giúp biểu diễn mối quan hệ phức tạp thông qua các tài liệu, tạo ra một phương thức tự nhiên và linh hoạt cho việc xử lý dữ liệu. [9]



Hình . MongoDB là gì? [10]

**Một số khái niệm cơ bản của MongoDB:**

* Tài Liệu (Document): Trong MongoDB, mỗi đơn vị dữ liệu được gọi là "tài liệu," một khái niệm tương đương với một hàng trong hệ thống quản lý cơ sở dữ liệu quan hệ, tuy nhiên, với sự linh hoạt và độ đa dạng lớn hơn. [9]
* Bộ Sưu Tập (Collection): Bộ sưu tập được coi là một tập hợp các tài liệu có liên quan, đồng nghĩa với một bảng trong hệ thống quản lý cơ sở dữ liệu quan hệ, nhưng với khả năng linh hoạt và đa dạng dữ liệu. [9]
* Cơ Sở Dữ Liệu (Database): Mỗi phiên bản của MongoDB lưu trữ nhiều cơ sở dữ liệu độc lập, và mỗi cơ sở dữ liệu chứa các bộ sưu tập riêng biệt, tạo nên sự đơn lẻ và cô lập trong quản lý dữ liệu. [9]
* Khóa "\_id": Mỗi tài liệu trong MongoDB đều có một khóa đặc biệt là "\_id," đảm bảo tính duy nhất trong phạm vi mỗi bộ sưu tập. [9]
* Mongo Shell: MongoDB được trang bị "mongo shell," một công cụ đơn giản nhưng mạnh mẽ. Mongo shell không chỉ là nền tảng hỗ trợ quản lý phiên bản MongoDB và thao tác dữ liệu qua ngôn ngữ truy vấn MongoDB mà còn là một trình thông dịch JavaScript toàn diện, hỗ trợ người dùng tạo và thực thi các kịch bản đa dạng cho mục đích nghiên cứu và ứng dụng. [9]

**B. Một số tương tác Document**

**Inserting Documents:**

Lệnh insert là một phương pháp cơ bản trong MongoDB, dùng để thêm dữ liệu vào hệ thống.

Để chèn một document duy nhất, bạn sử dụng phương thức insertOne để chèn một document vào.

Để chèn nhiều documents, bạn sử dụng phương thức insertMany hoặc insert.

**Ví dụ minh họa:**

Dưới đây là ví dụ thêm 1 sinh viên bằng câu lệnh insertOne:

db.sinhviens.insertOne({ MaSV: "110120031", TenSV: "Trần Thái Hưng", lop: "DA20TTB", Khoa: "Công nghệ Thông tin" })

Dưới đây là ví dụ thêm 2 sinh viên bằng câu lệnh insertMany :

db.sinhviens.insertMany([{ MaSV: "110120032", TenSV: "Nguyễn Thị A", lop: "DA20TTB", Khoa: " Công nghệ Thông tin" },{ MaSV: "110120033", TenSV: "Lê Văn Bình", lop: "DA20TTA", Khoa: " Công nghệ Thông tin" }])

**Updating Documents:**

Sau khi tài liệu được lưu trữ trong cơ sở dữ liệu, để cập nhật nó bằng cách sử dụng các phương thức như updateOne, updateMany và replaceOne. Các phương thức này yêu cầu một tài liệu bộ lọc để xác định phạm vi và một tài liệu sửa đổi hoặc thay thế để mô tả các thay đổi cần thực hiện.

**Ví dụ minh họa:**

Để cập nhật thông tin của một sinh viên trong bộ sưu tập, chúng ta sử dụng updateOne. Dưới đây là một ví dụ cụ thể, trong đó chúng ta cập nhật thông tin của sinh viên có mã sinh viên là "110120031":

db.sinhviens.updateOne({ MaSV : "110120031" },{ $set : {TenSV : "Nguyễn Thị Bình" }})

Để cập nhật thông tin cho nhiều sinh viên cùng một lúc, chúng ta sử dụng updateMany. Ví dụ sau đây minh họa việc cập nhật lớp của tất cả sinh viên thuộc Khoa "Công nghệ Thông tin" thành "DA20TT":

db.sinhviens.updateMany({ Khoa : "Công nghệ Thông tin" },{ $set : { lop : "DA20TT" }})

Đối với việc thay thế toàn bộ thông tin của một sinh viên bằng một document mới, chúng ta sử dụng replaceOne. Dưới đây là một ví dụ thay thế thông tin của sinh viên có mã sinh viên là "110120032":

db.sinhviens.replaceOne({ MaSV : "110120032" },{ TenSV : "Phạm Văn Cường", lop: "DA20TT", Khoa: "Công nghệ Thông tin" })

**Removing Documents:**

Phương thức deleteOne và deleteMany trong MongoDB đều yêu cầu một tài liệu bộ lọc (filter document) như là tham số đầu tiên. Tài liệu bộ lọc này chứa các tiêu chí xác định, quyết định việc xóa các tài liệu cụ thể từ bộ sưu tập (collection).

Việc thực hiện phương thức mà không có tiêu chí xác định trong MongoDB sẽ dẫn đến việc xóa tất cả các tài liệu từ bộ sưu tập, điều này cần được thực hiện cẩn thận để tránh mất mát dữ liệu đáng kể.

**Ví dụ minh họa:**

Để xóa sinh viên có mã số sinh viên (MaSV) là "110120031" (Trần Thái Hưng), sử dụng lệnh deleteOne như sau:

db.sinhviens.deleteOne({ MaSV: "110120031" })

Để xóa tất cả sinh viên thuộc Khoa "Công nghệ Thông tin", sử dụng lệnh deleteMany như sau:

db.sinhviens.deleteMany({ Khoa: "Công nghệ Thông tin" })

**Find Documents:**

Để thực hiện truy vấn dữ liệu từ một bộ sưu tập trong MongoDB, phương thức chính được sử dụng là find. Dưới đây là một ví dụ:

db.sinhviens.find({ Khoa: "Công nghệ Thông tin" })

**C. Ưu và Nhược Điểm của MongoDB**

**Ưu điểm:**

* Linh hoạt trong cấu trúc dữ liệu: MongoDB không đòi hỏi một schema cố định, cho phép lưu trữ dữ liệu trong các tài liệu có cấu trúc linh hoạt.
* Mở rộng dễ dàng: Hệ thống có khả năng mở rộng theo chiều ngang một cách hiệu quả, giúp quản lý lượng dữ liệu lớn mà không gặp nhiều trở ngại.
* Truy vấn mạnh mẽ: Hỗ trợ truy vấn đa dạng và phức tạp, bao gồm truy vấn tài liệu, truy vấn dựa trên cấu trúc cây, và hơn thế nữa.
* Hiệu suất cao: Cung cấp hiệu suất truy vấn nhanh, đặc biệt là trong các hoạt động với đọc và ghi dữ liệu.

**Nhược điểm:**

* Quản lý giao dịch: Mặc dù MongoDB hiện đã hỗ trợ giao dịch, nhưng các giao dịch trên MongoDB không mạnh mẽ và dễ sử dụng như trong các hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS).
* Phức tạp với các truy vấn JOIN: So với RDBMS, MongoDB không hiệu quả khi thực hiện các truy vấn liên kết phức tạp.

# CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

## Đặc tả ứng dụng

A screenshot of a cell phone

Description automatically generated

Hình . Sơ đồ usecase

**Quản trị đăng nhập vào hệ thống và thực hiện các chức năng:**

* Quản lý người dùng: Thêm, sửa, xóa người dùng trong hệ thống.
* Quản lý chương trình: Sửa đổi thông tin chương trình một cách chính xác.
* Quản lý khung: Sửa, xóa khung, điều chỉnh vị trí và chỉnh sửa file PDF.
* Quản lý PLO và PO: Thêm, sửa, xóa PLO và PO.
* Quản lý ma trận PLO-PO: Kiểm tra và thực hiện thay đổi trong mối quan hệ.
* Quản lý từ gợi ý: Chỉnh sửa mảng gợi ý.
* Xem các giao diện cho quản trị viên**:** Trang chủ, Template, PLO, PO,...

**Giáo viên đăng nhập vào hệ thống và thực hiện các chức năng:**

* Quản lý học phần: Thêm, sửa, xóa về học phần.
* Xem chương trình: Xem chương trình của bộ môn hoặc chuyên ngành.
* PDF: Xuất file PDF chứa chuẩn đầu ra hoặc file PDF của học phần đã tạo.
* Các chức năng nhập liệu: Sử dụng chức năng nhập liệu đề xuất, nhập liệu kéo thả và các chức năng khác trong phần nhập liệu học phần.
* Xem các giao diện cho người dùng**:** Trang chủ, đăng nhập, nhập liệu,...

## Phân tích và thiết kế cơ sở dữ liệu

A black background with white rectangles

Description automatically generated

Hình . Phân tích và thiết kế cơ sở dữ liệu

**Một số bảng chính trong dự án bao gồm:**

Bảng . Mô tả bảng học phần (HocPhan)

| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| MaHP | Mã học phần | String | Primary key |  |
| MaMon | Mã môn | String |  |  |
| SoGioHoc | Số giờ tự học | String |  |  |
| TH | Số giờ thực hành | String |  |  |
| LT | Số giờ lý thuyết | String |  |  |
| LoaiHocPhan | Loại học phần | String |  |  |
| SoTC\_LT | Số tính chỉ lý thuyết | String |  |  |
| SoTC\_TH | Số tính chỉ thực hành | String |  |  |
| TrinhDo\_DT | Trình độ đào tạo | String |  |  |
| ChuyenNganh | Chuyên ngành | String |  |  |
| HocKy | Học kỳ | String |  |  |
| MoTa | Mô tả | String |  |  |
| TenMon | Tên môn | String |  |  |
| NamHoc | Năm học | String |  |  |
| PhuLuc | Phụ lục | String |  |  |

Bảng . Mô tả bảng chuẩn đầu ra học phần (CDR\_HocPhan)

| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| MaCDR\_MH | Mã chuẩn đầu ra học phần | String | Primary key |  |
| MaHP | Mã học phần | String | Foreign key |  |
| loai\_CDRMH | Loại chuẩn đầu ra học phần | String |  |  |
| Noidung\_CDRMH | Nội dung chuẩn đầu ra học phần | String |  |  |
| DapUng\_CDRMH | Đáp ứng chuẩn đầu ra học phần | String |  |  |
| TrinhDo | Trình độ | String |  |  |
| loaiTUA | Loại TUA | String |  |  |

Bảng . Mô tả bảng khung (template)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| order | Mã khung | Number | Primary key | Dùng mã khung để thay đổi vị trí của các phần trong học phần |
| compileMethod | Tiêu đề biên dịch | String |  |  |
| htmlContent | Chuỗi html | String |  | Chứa chuỗi html chưa biên dịch |

Bảng . Mô tả người dùng (users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| username | tài khoản người dùng | String | Primary key |  |
| password | Mật khẩu người dùng | String |  |  |
| role | Quyền của người dùng | Number |  |  |

Bảng . Mô tả bảng chuẩn đầu ra chương trình (ChuongTrinh)

| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| id\_CT | Mã chương trình | String | Primary key |  |
| TenChuongTrinh | Tên chương trình | String |  |  |
| LoaiHinh | Loại hình | String |  |  |
| STC\_YeuCau | Số tính chỉ yêu cầu | Number |  |  |
| Tg\_DaoTao | Thời gian đào tạo | Number |  |  |
| Bac | Bậc | String |  |  |

Bảng . Mô tả bảng chuẩn đầu ra chương trình (CDR\_CT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| id\_CDR | Mã chuẩn đầu ra  chương trình | String | Primary key |  |
| id\_CT | Mã chương trình | String | Foreign key |  |
| Ten\_CDR | Tên chuẩn đầu ra | String |  |  |
| LoaiCDR\_CT | Loại chuẩn đầu ra chương trình | String |  |  |
| NoiDung | Nội dung chuẩn đầu ra chương trình | String |  |  |

Bảng . Mô tả bảng mục tiêu đào tạo (MucTieuCTDT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| MaMT\_CTDT | Mã mục tiêu đào tạo | String | Primary key |  |
| id\_CT | Mã chương trình | String | Foreign key |  |
| TenMT\_CTDT | Tên chuẩn mục tiêu | String |  |  |
| NoiDung | Nội dung mục tiêu | String |  |  |

Bảng . Mô tả bảng đáp ứng chương trình (Dapung\_CT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| MaMT\_CTDT | Mã mục tiêu đào tạo | String | Primary key |  |
| id\_CDR | Mã chương trình | String | Primary key |  |

Bảng . Mô tả bảng đáp ứng chuẩn đầu ra (Dapung\_CDR)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| Ten\_CDR | Tên chuẩn đầu ra | String | Primary key |  |
| MaCDR\_MH | Mã chương trình | String | Primary key |  |

Bảng Mô tả bảng tạo (Create)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| MaHP | Mã học phần | String | Primary key |  |
| username | Tài khoản người dùng | String | Primary key |  |

## Thiết kế kiến trúc ứng dụng

A diagram of a software application

Description automatically generated

Hình . Kiến trúc Client Server

**Dòng dữ liệu và tương tác theo hướng sau:**

* Người dùng khởi tạo yêu cầu từ trình duyệt đến giao diện người dùng (UI) của ứng dụng.
* UI, sử dụng Handlebars.js, tạo và phản hồi nội dung HTML đến trình duyệt.
* Đối với các hoạt động đòi hỏi dữ liệu động (CRUD - Tạo, Đọc, Cập nhật, Xóa), UI gọi đến API.
* API, được xây dựng bằng Express.js, xử lý và chuyển yêu cầu tới máy chủ Node.js để thực hiện xử lý.
* Máy chủ Node.js xử lý logic nghiệp vụ và tương tác với MongoDB để truy xuất hoặc cập nhật dữ liệu.
* Dữ liệu được truyền từ máy chủ trở lại API, sau đó API trả lại dữ liệu cho UI.
* UI hiển thị dữ liệu cho người dùng qua trình duyệt.
* Mỗi mũi tên chỉ ra hướng chuyển giao dữ liệu hoặc yêu cầu giữa các thành phần.

## Thiết kế API

### Định tuyến API trong Express

Trong quá trình phát triển ứng dụng, việc tổ chức và cấu trúc mã nguồn một cách hợp lý rất quan trọng. Điều này bao gồm việc phân chia API vào trong các module riêng biệt, giúp nâng cao khả năng quản lý và bảo trì mã nguồn. Trong tệp app.js, việc định nghĩa các API được thực hiện thông qua phương thức app.use(). Cách tổ chức này giúp việc truy cập và sử dụng các API trở nên thuận tiện và hiệu quả hơn.

A screen shot of a computer program

Description automatically generated

Hình . Định tuyến API trong Express

Trong cấu trúc của ứng dụng, mỗi router được liên kết với một API tương ứng. Bên trong mỗi router, các endpoint phục vụ chức năng cụ thể được tổ chức, bao gồm cả việc định nghĩa đường dẫn, xử lý các yêu cầu HTTP, Middleware, và các logic xử lý liên quan khác.

**Một số lưu ý:** Các đường dẫn bên trong router sẽ được bắt đầu bằng API mà nó được gắn.

### Định tuyến trong router 'admin.Routes'

Mỗi định tuyến trong router 'admin' được áp dụng hai middleware là verifyToken để xác nhận JWT và hasRole(1) để kiểm tra quyền của admin.

**Định tuyến đến trang chủ admin:**

A screen shot of a computer

Description automatically generated

Hình . Định tuyến đến trang chủ admin

* + GET '/': Hiển thị trang chủ dành cho quản trị viên.

**Các định tuyến quản lý chương trình học:**

A screen shot of a computer screen

Description automatically generated

Hình . Các định tuyến quản lý chương trình học

Định tuyến cho 'program':

* + GET: Hiển thị thông tin chương trình.
  + GET '/edit': Mở giao diện chỉnh sửa.
  + PUT '/edit/:id': Cập nhật thông tin theo ID.

Định tuyến cho 'program/po':

* + GET: Liệt kê PO của chương trình.
  + POST: Thêm PO mới.
  + PUT '/update/:id': Cập nhật PO.
  + DELETE '/delete/:id': Xóa PO.

Định tuyến cho 'program/plo':

* + GET: Liệt kê PLO của chương trình.
  + POST: Thêm PLO mới.
  + PUT '/update/:id': Cập nhật PLO.
  + DELETE '/delete/:id': Xóa PLO.

Định tuyến cho 'program/Matrix':

* + GET: Xem ma trận.
  + POST: Cập nhật ma trận.

**Các định tuyến cho quản lý người dùng:**

A computer screen with text on it

Description automatically generated

Hình . Các định tuyến cho quản lý người dùng.

Định tuyến cho '/list-user':

* + GET: Hiển thị danh sách người dùng.
  + POST: Tạo người dùng mới.
  + DELETE '/delete/:id': Xóa người dùng theo ID.
  + DELETE '/user/delete/:id': Xóa học phần thuộc người dùng theo ID.

**Các định tuyến quản lý khung và từ gợi ý:**

A computer code on a black background

Description automatically generated

Hình . Các định tuyến quản lý khung và từ gợi ý

Định tuyến cho '/template':

* + GET: Xem danh sách khung.
  + PUT '/update/:id': Cập nhật khung theo ID.

Định tuyến cho '**/**autocomplete'**:**

* + GET: Xem từ gợi ý.
  + PUT '/update/:id': Cập nhật từ gợi ý theo ID.

### Định tuyến trong router 'project.Routes'

Mỗi định tuyến trong 'project' Router được bảo vệ bởi middleware verifyToken để xác thực JWT, đảm bảo tính an toàn và xác thực người dùng trước khi tiếp tục xử lý yêu cầu.

**Các định tuyến quản lý học phần:**

A screen shot of a computer program

Description automatically generated

Hình . Các định tuyến quản lý học phần

Định tuyến cho '/':

* + GET: Điều hướng đến view nhập liệu học phần.
  + POST: Lưu thông tin học phần.

Định tuyến cho '/pdf ':

* + POST: Xuất file PDF học phần.

Định tuyến cho '/pdf-CDR':

* + POST: Xuất file PDF chuẩn đầu ra học phần.

Định tuyến cho '/create:

* + GET: Điều hướng đến view tạo tên file mới cho học phần.

Định tuyến cho '/checkName ':

* + POST: Kiểm tra tên file học phần mới của user.

Định tuyến cho '/update ':

* + GET: Điều hướng đến view cập nhật học phần.

Định tuyến cho '/ delete/:MaHP ':

* + DELETE: Xóa thông tin mã học phần. Quá trình này bao gồm xóa thông tin cũ và gửi phản hồi cho client. Client sau đó gửi POST để lưu thông tin mới.

Định tuyến cho '/delete\_file/:fileName':

* + DELETE: Xóa file dựa trên tên file.

### Định tuyến trong router 'user.Routes'

**Các định tuyến quản lý danh sách học phần của người dùng:**

A screen shot of a computer program

Description automatically generated

Hình . Các định tuyến của router người dùng

Định tuyến cho '/ listProject':

* + GET: Danh sách trang danh sách học phần của người dùng.

Định tuyến cho ' /delete/:filename':

* + DELETE: Xử lý việc xóa học phần của người dùng dựa vào mã học phần cung cấp.

### Các định tuyến router khác

Hiển thị chương trình cho người dùng trong router 'chuongTrinh.Routes'.

A screen shot of a computer

Description automatically generated

Hình . Định tuyến hiển thị chương trình cho người dùng

Hiển thị trang chủ cho người dùng trong router 'home.Routes'.

A black and white screen with white text

Description automatically generated

Hình . Định tuyến hiển thị trang chủ cho người dùng

Định tuyến thực hiện chức năng login trong router 'login.Routes'.

A screen shot of a computer screen

Description automatically generated

Hình . Định tuyến thực hiện chức năng login

Định tuyến cho '/':

* + GET: Điều hướng đến trang đăng nhập.
  + POST: Tạo token JWT và di chuyển đến trang chủ.

Định tuyến thực hiện chức năng logout bằng cách xóa token và điều hướng người dùng đến trang login trong router 'logout.Routes'.

A computer screen shot of a black and white screen

Description automatically generated

Hình . Định tuyến thực hiện chức năng logout

## Thiết kế giao diện

### Thiết kế khung giao diện cho người bằng Handlebars.js

A computer screen shot of a program code

Description automatically generated

Hình . Thiết kế khung giao diện cho người.

Cấu trúc của trang web được chia thành ba thành phần chính, bao gồm header, body và footer. Header và footer được nhúng vào thông qua các mẫu con (partials) {{>header}} và {{>footer}}, tạo điều kiện cho việc tái sử dụng chúng trên nhiều trang khác nhau. Phần body được biểu diễn bởi {{{body}}}, là vị trí mà nội dung chính của trang web sẽ được hiển thị. Cấu trúc này không chỉ giúp tối ưu hóa việc quản lý trang web mà còn tạo ra một kiến trúc rõ ràng.

### Thiết kế khung giao diện cho admin bằng Handlebars.js

A computer screen shot of text

Description automatically generated

Hình . Thiết kế khung giao diện admin.

Ban đầu, khung được ẩn đi với thuộc tính style="display: none;" và được hiển thị lại khi quyền đăng nhập là quản trị (thường là thông qua JavaScript). Khung này gồm có phần đầu trang (Admin\_header), một thanh điều hướng (Admin\_Nav), và một khu vực nội dung (body) để chứa nội dung động. Việc sử dụng mẫu con (partials) cho header và thanh điều hướng giúp tái sử dụng và duy trì mã nguồn một cách gọn gàng.

## Giao tiếp giữa Front-End và Back-End

Trong cấu trúc ứng dụng web mà tôi phát triển, quá trình liên lạc giữa phần Front-End và Back-End chủ yếu diễn ra qua các yêu cầu HTTP. Tại Front-End, tôi sử dụng thư viện axios để phát đi các yêu cầu HTTP, với việc gửi dữ liệu đồng thời lên Back-End dưới dạng JSON hoặc FormData. Khi Back-End nhận được những yêu cầu này, nó bắt đầu quá trình xử lý. Sau khi yêu cầu đã được xử lý thành công, Back-End sẽ đáp lại bằng cách gửi dữ liệu về Front-End dưới dạng JSON hoặc thực hiện điều hướng tại Front-End nếu có yêu cầu.

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

1. Chức năng của người dùng
2. Chức năng **xem trang chủ**

A screenshot of a computer

Description automatically generated

Hình . Giao diện trang chủ

Khi người dùng truy cập trang web, họ sẽ được vào trang chủ của hệ thống. Trên trang này, họ được xem các mẫu và chuẩn đầu ra của chương trình được hiển thị, cũng như tận dụng các tính năng từ thanh điều hướng. Để truy cập vào các tính năng đầy đủ của hệ thống, cần phải đăng nhập.

1. Chức năng đăng nhập

A screenshot of a computer

Description automatically generated

Hình . Giao diện đăng nhập.

Trong quá trình truy cập, người dùng sử dụng đường dẫn trực tiếp hoặc nhấn vào nút đăng nhập trên thanh điều hướng. Khi người dùng truy cập, hệ thống hiển thị giao diện đăng nhập. Tại đây, họ nhập thông tin đăng nhập và nhấn nút đăng nhập. Hệ thống sẽ cấp token khi thông tin đăng nhập chính xác, khởi tạo phiên làm việc. Quyền hạn của tài khoản sẽ quyết định giao diện mà người dùng được chuyển tới sau khi đăng nhập thành công. Trong trường hợp thông tin không chính xác, hệ thống sẽ thông báo và yêu cầu người dùng thử lại.

1. Chức năng xem chương trình của người dùng

A screenshot of a computer

Description automatically generated

Hình . Giao diện xem chương trình của người dùng

Trong quá trình nhập liệu, để xem chi tiết chương trình học, người dùng sử dụng chức năng "Xem Chương Trình" trên thanh điều hướng hoặc truy cập đường dẫn trực tiếp. Khi người dùng truy cập, hệ thống hiển thị giao diện xem chương trình. Tại đây, người dùng quan sát được thông tin chi tiết về chương trình, bao gồm chuẩn đầu ra và mục tiêu, cùng mối quan hệ giữa chúng. Sau khi đã quan sát, người dùng muốn quay lại, họ cần nhấn nút "Quay Về Trang Chủ".

1. Chức năng tạo học phần

A screenshot of a computer

Description automatically generated

Hình . Giao diện tạo học phần

Trong quá trình xem mẫu trên hệ thống, người dùng muốn tạo học phần của mình, họ nhấn vào nút "Tạo dự án" trên thanh điều hướng. Hệ thống hiển thị giao diện cho phép người dùng nhập tên học phần. Sau khi người dùng nhập tên học phần, hệ thống sẽ kiểm tra xem tên học phần đã tồn tại hay chưa. Nếu tên học phần đã tồn tại, hệ thống thông báo cho người dùng và yêu cầu họ nhập một tên học phần mới. Ngược lại, nếu tên học phần không trùng, hệ thống sẽ tạo học phần với tên được nhập và chuyển hướng người dùng đến giao diện nhập liệu học phần.

1. Chức năng nhập liệu học phần

**A screenshot of a computer

Description automatically generated**

Hình . Giao diện nhập liệu học phần.

Sau khi học phần của người dùng được tạo thành công. Hệ thống sẽ hiển thị giao diện cho người dùng nhập liệu. Giao diện nhập liệu được chia thành các phần khác nhau, mỗi phần có chức năng tương ứng. Tại giao diện này, người dùng được nhập thông tin và sử dụng các chức năng có sẵn trong giao diện. Khi người dùng nhập tên vào học phần, hệ thống sẽ hiển thị đề xuất cho họ.

A screenshot of a computer

Description automatically generated

Hình Người dùng sử dụng chức năng đề xuất tên học phần.

Người dùng lựa chọn đề xuất ‘CÔNG NGHỆ PHẦN MỀM’, sau đó một số thông tin sẽ được cập nhật tương ứng dựa vào sự lựa chọn của họ.

A close up of a sign

Description automatically generated

Hình . Mã số học phần được cập nhật.



Hình . Học phần tiên quyết được cập nhật.

A white rectangular object with black text

Description automatically generated

Hình . Phần hai giao diện nhập liệu học phần.

Trong phần hai liên quan đến nguồn học liệu, người dùng bổ sung những nguồn mới thông qua các chức năng tương ứng trên ‘bảng chương phần 2’. Khi lựa chọn thêm, hệ thống sẽ tạo cấu trúc mới tại vị trí được chọn trong phần đó. Trong trường hợp người dùng muốn loại bỏ một nguồn học liệu sau khi đã thêm, họ lựa chọn nguồn cần xóa và sử dụng chức năng xóa trên bảng tương ứng.

Trong quá trình thêm hoặc xóa nguồn học liệu, số thứ tự của các nguồn trong danh sách sẽ tự động được cập nhật bởi hệ thống. Điều này đảm bảo tính nhất quán và thuận tiện cho quá trình theo dõi và quản lý.

A white text with black text

Description automatically generated

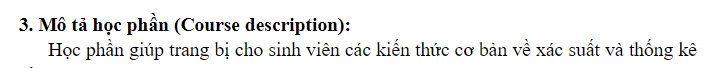
Hình . Người dùng thực hiện chức năng thêm tài liệu loại học liệu.

A screenshot of a computer

Description automatically generated

Hình . Người dùng thực hiện chức năng xóa 1 giáo trình.

Trong phần ba, khi người dùng thực hiện nhập liệu cho mô tả học phần, họ được nhập trực tiếp. Khi sử dụng phím "space", tính năng nhập liệu đề xuất sẽ được kích hoạt, giúp hỗ trợ người dùng trong quá trình nhập liệu mô tả học phần một cách hiệu quả và chính xác.



Hình . Phần ba giao diện nhập liệu học phần .

Hình minh họa đề xuất khi sử dụng phím 'space' và nhập chữ 'h'.

A screenshot of a computer

Description automatically generated

Hình . Người dùng sử dụng chức năng đề xuất.

Trong phần bốn, chuẩn đầu ra của học phần, chức năng hoạt động tương tự như trong phần hai, cho phép người dùng nhập và quản lý thông tin liên quan đến chuẩn đầu ra.

A screenshot of a computer

Description automatically generated

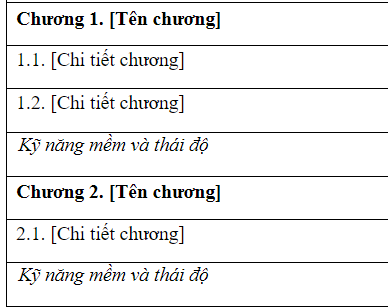
Hình . Phần bốn giao diện nhập liệu học phần

Trong phần năm nội dung học phần, chức năng xóa hoạt động tương tự như trong phần hai. Người dùng nhập tên và chi tiết của mỗi chương. Để thêm chương mới, người dùng chọn thêm chương, và hệ thống sẽ tạo cấu trúc chương mới. Thêm chi tiết chương được thực hiện thông qua ‘bảng chương phần 5’.

A white sheet with black text

Description automatically generated

Hình . Phần năm giao diện nhập liệu học phần



Hình Người dùng thực hiện chức năng thêm cấu trúc

Phần sáu hoạt động tương tự như phần ba trong hệ thống, người dùng thực hiện nhập liệu tương tự.

A white background with black text

Description automatically generated

Hình . Người dùng sử dụng chức năng đề xuất

Trong phần bảy, đánh giá học phần, người dùng thực hiện các chức năng tương tự như trong các phần trước, cho phép họ nhập liệu và quản lý đánh giá liên quan đến học phần.

A screenshot of a computer

Description automatically generated

Hình . Phần bảy giao diện nhập liệu học phần

Phần tám là phần mẫu chung. Trong phần này, người dùng trực tiếp xem và kiểm tra các quy định, định dạng, hoặc cấu trúc chung được thiết kế và quản lý bởi quản trị viên.

A screenshot of a computer

Description automatically generated

Hình . Phần tám giao diện nhập liệu học phần

A white background with black text

Description automatically generated

Hình . Chức năng đề xuất nhập liệu giáo viên

Kết thúc quá trình biên soạn học phần, người dùng ấn "Lưu" để ghi lại công việc. Để xuất nội dung ra PDF, họ sẽ chọn "Xuất PDF" và chờ hệ thống xử lý trong vòng 5 giây để hoàn thành yêu cầu xuất file.

A grey rectangular sign with white text

Description automatically generated

Hình . Nút lưu và nút xuất file PDF

1. Chức năng quản lý học phần

A screenshot of a computer

Description automatically generated

Hình . Giao diện danh sách học phần

Tại đây, người dùng xem lại danh sách học phần đã tạo, họ chọn "Chỉnh sửa" để sửa đổi học phần hoặc "Xóa" để loại bỏ học phần không cần thiết.

Để xuất học phần ra file PDF hoặc PDF chuẩn đầu ra của học phần, người dùng chỉ cần nhấn vào tùy chọn "PDF" tương ứng.

**A screenshot of a computer

Description automatically generated**

Hình . Chức năng xuất PDF chuẩn đầu ra học phần

**A screenshot of a computer

Description automatically generated**

Hình . Chức năng xuất PDF đề cương

1. Chức năng nhập liệu kéo thả

Để thực hiện chức năng này, người dùng cần nhấp và giữ chuột trái, sau đó di chuyển chuột để bôi đen vùng văn bản cần thực hiện, và cuối cùng, họ kéo văn bản đã bôi đen vào phần cho phép nhập liệu.

A screen shot of a computer

Description automatically generated

Hình . Chức năng nhập liệu kéo thả.

A white background with black text

Description automatically generated

Hình . Kết quả nhập liệu kéo thả.

1. Chức năng của admin
2. Chức năng chỉnh sửa chương trình

**A screenshot of a computer

Description automatically generated**

Hình . Giao diện chỉnh sửa chương trình của admin

Sau khi nhận thay đổi từ khoa, admin trong giao diện quản lý sẽ nhập các thay đổi cần thiết và sau đó lưu chúng bằng cách nhấn vào nút "Chỉnh sửa".

1. Chức năng quản lý PLO của admin

**A screenshot of a computer

Description automatically generated**

Hình . Giao diện quản lý PLO

Khi admin nhận được sự thay đổi từ khoa, admin vào giao diện quản lý chuẩn đầu ra. Tại đây, admin sử dụng nút "Xóa" để loại bỏ chuẩn đầu ra không cần thiết, và "Thêm" để bổ sung chuẩn đầu ra mới. Quá trình này giúp admin quản lý chương trình chính xác.

A screenshot of a computer

Description automatically generated

Hình . Thêm mới một PLO

1. Chức năng quản lý PO của admin

**A screenshot of a computer

Description automatically generated**

Hình . Giao diện quản lý mục tiêu đào tạo của admin

Giao diện quản lý mục tiêu đào tạo (PO) được thiết kế với mục đích hỗ trợ admin trong quá trình cập nhật một cách dễ dàng và hiệu quả. Trên giao diện này, danh sách các mục tiêu đào tạo được hiển thị một cách rõ ràng, đi kèm với các tùy chọn "Xóa" và "Thêm" để quản lý chúng. Tùy chọn "Xóa" cho phép admin loại bỏ những mục tiêu đào tạo không cần thiết, trong khi tùy chọn "Thêm" giúp họ thêm mới các Mục Tiêu một cách thuận tiện.

1. Chức năng quản lý Matrix PLO-PO của admin

**A screenshot of a computer

Description automatically generated**

Hình . Giao diện chỉnh sửa Matrix PLO-PO của admin

Trong giao diện này, admin thay đổi và đánh dấu các PLO và PO thông qua hộp kiểm. Khi hoàn tất việc lựa chọn, admin lưu lại bằng cách nhấn vào nút "Chỉnh sửa PLO-PO" để cập nhật các thay đổi vào hệ thống.

A screenshot of a computer

Description automatically generated

Hình . Admin check vào PO4 và PO5,PO6

A screenshot of a computer

Description automatically generated

Hình . Kết quả sau khi admin check PO

1. Chức năng quản lý mẫu của admin

Khi nhận thay đổi từ khoa, admin truy cập vào quản lý khung và thực hiện cập nhật mẫu chung. Sau khi hoàn thành chỉnh sửa, admin nhấn vào nút "Chỉnh sửa" để áp dụng các thay đổi. Hệ thống sau đó tiến hành cập nhật mẫu theo những thay đổi được xác nhận.

A screenshot of a computer

Description automatically generated

Hình . Admin chỉnh sửa quy định về học vụ

A screenshot of a text

Description automatically generated

Hình . Kết quả mẫu học phần sau khi thay đổi

1. Chức năng quản lý người dùng của admin

**A screenshot of a computer

Description automatically generated**

Hình . Giao diện quản lý người dùng của admin

Trong giao diện quản lý, admin thêm người dùng mới thông qua nút "Thêm mới". Admin quản lý người dùng và bài viết hiện có, với các tùy chọn để "Chỉnh sửa" thông tin hoặc "Xóa" người dùng hoặc bài viết không mong muốn khỏi hệ thống.

1. Chức năng quản lý từ gợi ý của admin

Tại đây admin thực hiện thêm chỉnh sửa gợi ý. Sau khi hoàn thành nhấn lưu gợi ý để cập nhật gợi ý.

A screenshot of a computer

Description automatically generated

Hình . Admin thực hiện chức năng thêm từ gợi ý là “tích phân”

A white background with black text

Description automatically generated

Hình . Kiểm tra kết quả thêm gợi ý

* Top of Form

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

**Đối với bản thân:**

Trong quá trình thực hiện dự án, tôi đã đạt được những bước tiến quan trọng trong việc nâng cao kiến thức và kỹ năng chuyên môn. Sự am hiểu sâu sắc về MongoDB, một cơ sở dữ liệu NoSQL, đã mở rộng hiểu biết của tôi về quản lý và lưu trữ dữ liệu trong môi trường không cấu trúc. Sử dụng API Express trong bối cảnh Node.js cũng đã củng cố khả năng của tôi trong việc phát triển ứng dụng web, đặc biệt là trong việc xây dựng các dịch vụ web linh hoạt và mạnh mẽ.

Quá trình này đã tạo điều kiện cho tôi nắm vững công nghệ Node.js, nền tảng JavaScript cho server, từ đó mở rộng kỹ năng lập trình JavaScript của bản thân, đặc biệt là trong lĩnh vực tương tác với DOM, yếu tố then chốt trong việc thiết kế giao diện người dùng web. Kỹ năng làm việc với Handlebars.js, thư viện template mạnh mẽ, cũng được nâng cao, góp phần vào việc tạo ra các trang web động một cách hiệu quả.

**Đối với người dùng:**

* Form nhập liệu theo quy định: Phát triển giao diện nhập liệu tuân theo quy định cụ thể.
* Xuất PDF đề cương học phần: Tích hợp chức năng xuất đề cương và chuẩn đầu ra dưới dạng PDF.
* Nhập liệu đề xuất.

**Đối với quản trị viên (admin):**

* Xây dựng giao diện quản trị cho phép thay đổi mẫu nhập liệu: Tạo điều kiện cho quản trị viên thay đổi và cập nhật mẫu đề cương học phần, đáp ứng nhu cầu thay đổi của chương trình đào tạo.
* Giao diện quản lý PO, PLO, quan hệ PO và PLO..

**Tạo đề xuất cho người dùng nhập liệu:**

* Xây dựng hàm đề xuất riêng: Phát triển hàm 'autocomplete' tùy chỉnh để hỗ trợ người dùng trong quá trình nhập liệu.

1. Hạn chế

* Chưa hỗ trợ xuất đề cương dưới dạng Word.
* Yêu cầu kiến thức lập trình để chỉnh sửa mẫu đề cương.
* Hiện tại ứng dụng chưa có trang tìm kiếm.
* Một số phần của ứng dụng chưa hiển thị tính năng đề xuất.
* Một số thông số và cài đặt gợi ý được thiết lập cứng, hạn chế khả năng tùy biến theo nhu cầu của người dùng.

1. Hướng phát triển

Phát triển giao diện mẫu kéo và thả: Nhằm tối ưu hóa quá trình tạo và chỉnh sửa đề cương học phần, việc phát triển một giao diện người dùng dựa trên mô hình kéo và thả sẽ được triển khai. Giao diện này sẽ cung cấp một phương thức trực quan và dễ dàng cho việc tùy chỉnh mẫu đề cương, giảm bớt nhu cầu về kỹ năng lập trình đối với người dùng cuối. Tích hợp tính năng xuất đề cương học phần sang định dạng Microsoft Word, sử dụng công cụ 'mammoth'. Cải thiện giao diện: bổ xung trang tìm kiếm, cải thiện đề xuất. Cập nhật tất cả các mảng đề xuất vào cơ sở dữ liệu.

# DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | “w3schools,” [Online]. Available: https://www.w3schools.com/howto/howto\_js\_autocomplete.asp. |
| [2] | M. Schwarzmüller, “oreilly,” 2019. [Online]. Available: https://learning.oreilly.com/course/node-js-the/9781838826864/. |
| [3] | “wikimedia,” [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Node.js\_logo.svg/langfr-330px-Node.js\_logo.svg.png. |
| [4] | J. Wilson, Node.js 8 the Right Way: Practical, Server-Side JavaScript That Scales, O'Reilly Media, 2018. |
| [5] | “nodejs,” [Online]. Available: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs. |
| [6] | E. Hahn, Express in Action, Manning Publications, 2016. |
| [7] | E. T. Coder, “dev.to,” [Online]. Available: https://res.cloudinary.com/practicaldev/image/fetch/s--KkScstnJ--/c\_imagga\_scale,f\_auto,fl\_progressive,h\_420,q\_auto,w\_1000/https://dev-to-uploads.s3.amazonaws.com/uploads/articles/zojuy79lo3fn3qdt7g6p.png. |
| [8] | S. H. Clive Harber, Getting MEAN with Mongo, Express, Angular, and Node, Second Edition, Manning Publications, 2019. |
| [9] | E. B. K. C. Shannon Bradshaw, MongoDB: The Definitive Guide, 3rd Edition, O'Reilly Media, Inc., 2019. |
| [10] | “bleepstatic,” [Online]. Available: https://www.bleepstatic.com/content/hl-images/2023/12/16/mongodb-header-o.jpg. |