

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



**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2025-2026
XÂY DỰNG WEBSITE ĐĂNG KÝ HỌC NGOẠI NGỮ
TÍCH HỢP TRỢ LÝ ẢO AI HỖ TRỢ TƯ VẤN KHÓA HỌC**

Giảng viên hướng dẫn:
ThS. Phạm Minh Dương

Sinh viên thực hiện:
Họ tên: Nguyễn Duy Tín
MSSV: 110122182
Lớp: DA22TTA

Vĩnh Long, tháng 12 năm 2025

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



**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2025-2026
XÂY DỰNG WEBSITE ĐĂNG KÝ HỌC NGOẠI NGỮ
TÍCH HỢP TRỢ LÝ ẢO AI HỖ TRỢ TƯ VẤN KHÓA HỌC**

Giảng viên hướng dẫn:
ThS. Phạm Minh Dương

Sinh viên thực hiện:
Họ tên: Nguyễn Duy Tín
MSSV: 110122182
Lớp: DA22TTA

Vĩnh Long, tháng 12 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

[illegible]

Vĩnh Long, ngày tháng năm

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

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

[illegible]

Vĩnh Long, ngày tháng năm

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết, tôi xin gửi lời cảm ơn chân thành đến Thầy Phạm Minh Đương, người đã trực tiếp hướng dẫn, giúp đỡ và giải đáp thắc mắc trong suốt quá trình thực hiện đề tài. Sự tận tình của Thầy đã giúp tôi hoàn thành đồ án này một cách tốt nhất.

Tôi cũng xin bày tỏ lòng biết ơn đến các Thầy (Cô) trong Khoa Công nghệ thông tin đã cung cấp những kiến thức nền tảng cần thiết trong suốt thời gian học tập và nghiên cứu.

Do lượng kiến thức và kinh nghiệm của tôi còn khiêm tốn nên vẫn có thể có những sai sót trong đồ án này. Kính mong quý Thầy (Cô) thông cảm và tôi mong nhận được những nhận xét, đóng góp từ Thầy (Cô) để tôi hoàn thiện hơn.

Tôi xin chân thành cảm ơn!

Vĩnh Long, ngày tháng năm

Sinh viên thực hiện

Nguyễn Duy Tín

MỤC LỤC

	Trang
CHƯƠNG 1: TỔNG QUAN.....	1
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	2
2.1 Tìm hiểu về RESTful API	2
2.1.1 Tổng quan về RESTful API	2
2.1.2 Các thành phần chính.....	2
2.1.3 Nguyên lý hoạt động.....	3
2.1.4 Tiêu chí của một API chuẩn RESTful	3
2.2 Tìm hiểu về NodeJS	3
2.2.1 Tổng quan về NodeJS	3
2.2.2 Cơ chế hoạt động và kiến trúc	4
2.2.3 Các thành phần quan trọng	4
2.3 Tìm hiểu về ExpressJS	4
2.3.1 Tổng quan về ExpressJS	4
2.3.2 Chức năng	5
2.4 Tìm hiểu về MongoDB.....	5
2.4.1 Tổng quan về MongoDB	5
2.4.2 Lịch sử hình thành.....	5
2.5 Tìm hiểu về ReactJS.....	6
2.5.1 Tổng quan về ReactJS.....	6
2.5.2 Một số tính năng	6
2.6 Tìm hiểu về Vite.....	6
2.6.1 Tổng quan về Vite.....	6
2.6.2 Giải pháp về hiệu suất của Vite	7
2.7 Tìm hiểu về Tailwind CSS	7
2.8 Khảo sát các website cùng lĩnh vực	8
2.8.1 Khảo sát website khoahocgiaioi.com.....	8
2.8.2 Khảo sát website khokhoahoc.org	9
2.8.3 Rút ra kinh nghiệm cho đề tài	9
2.9 Các nghiệp vụ liên quan đến đề tài	10
2.10 Các công trình nghiên cứu liên quan.....	11

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	14
3.1 Mô tả hệ thống.....	14
3.2 Xác định các yêu cầu chức năng của hệ thống.....	15
3.3 Thiết kế dữ liệu hệ thống.....	15
3.4 Thiết kế xử lý hệ thống.....	22
3.5 Mô hình hoạt động của hệ thống.....	27
3.5.1 Kiến trúc hệ thống.....	27
3.5.2 Cấu trúc thư mục.....	32
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU.....	36
4.1 Kiểm thử API với Postman	36
4.2 Giao diện người dùng và chức năng.....	40
4.3 Giao diện quản trị viên và chức năng.....	47
4.4 So sánh với 2 website đã khảo sát ở chương 2.....	48
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	51
5.1 Kết luận	51
5.2 Hướng phát triển.....	52
DANH MỤC TÀI LIỆU THAM KHẢO	53

DANH MỤC HÌNH ẢNH

Hình 1: Khảo sát website <i>khoahocgiaioi.com</i>	8
Hình 2: Khảo sát website <i>khokhoahoc.org</i>	9
Hình 3: Mô hình cơ sở dữ liệu	16
Hình 4: Use-case của actor Khách	22
Hình 5: Use-case của actor Người dùng	24
Hình 6: Use-case của actor Admin	25
Hình 7: Kiến trúc hệ thống.....	28
Hình 8: Luồng xử lý của Chatbot.....	30
Hình 9: Cấu trúc thư mục back-end.....	32
Hình 10: Cấu trúc thư mục front-end	34
Hình 11: Kiểm thử API đăng ký tài khoản.....	36
Hình 12: Kiểm thử API đăng nhập	36
Hình 13: Kiểm thử API tạo khóa học mới (admin)	37
Hình 14: Kiểm thử API lấy tất cả khóa học	38
Hình 15: Kiểm thử API lấy chi tiết 1 khóa học	39
Hình 16: Kiểm thử API đăng ký khóa học	39
Hình 17: Kiểm thử API chatbot tư vấn khóa học.....	40
Hình 18: Giao diện trang chủ (Chưa đăng nhập)	41
Hình 19: Giao diện trang chủ (Đã đăng nhập)	41
Hình 20: Giao diện trang khóa học	42
Hình 21: Giao diện chatbot.....	42
Hình 22: Giao diện trang chi tiết khóa học	43
Hình 23: Giao diện trang chọn buổi học	44
Hình 24: Giao diện xem lại thanh toán và áp dụng mã giảm giá.....	44
Hình 25: Giao diện trang khóa học của tôi	45
Hình 26: Giao diện trang chi tiết khóa học đã kết thúc.....	45
Hình 27: Giao diện trang đánh giá khóa học	46
Hình 28: Giao diện trang liên hệ	46
Hình 29: Giao diện trang tổng quan.....	47
Hình 30: Giao diện trang quản lý người dùng	48
Hình 31: Giao diện trang quản lý lớp học	48

DANH MỤC BẢNG BIỂU

<i>Bảng 1: Mô tả bảng USER.....</i>	<i>16</i>
<i>Bảng 2: Mô tả bảng LANGUAGE.....</i>	<i>17</i>
<i>Bảng 3: Mô tả bảng LANGUAGE_LEVEL.....</i>	<i>17</i>
<i>Bảng 4: Mô tả bảng CLASSSESSION.....</i>	<i>17</i>
<i>Bảng 5: Mô tả bảng TEACHER.....</i>	<i>18</i>
<i>Bảng 6: Mô tả bảng COURSE</i>	<i>18</i>
<i>Bảng 7: Mô tả bảng REGISTRATION_COURSE.....</i>	<i>19</i>
<i>Bảng 8: Mô tả bảng COUPON.....</i>	<i>20</i>
<i>Bảng 9: Mô tả bảng REVIEW.....</i>	<i>21</i>
<i>Bảng 10: Mô tả bảng CONTACT.....</i>	<i>21</i>
<i>Bảng 11: Mô tả bảng SLIDESHOW</i>	<i>21</i>
<i>Bảng 12: Mô tả các actor của dự án.....</i>	<i>22</i>
<i>Bảng 13: Mô tả use-case của actor Khách</i>	<i>23</i>
<i>Bảng 14: Mô tả use-case của actor Người dùng</i>	<i>24</i>
<i>Bảng 15: Mô tả use-case của actor Admin</i>	<i>26</i>

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ gốc
1	AI	Artificial Intelligence
2	API	Application Programming Interface
3	CSS	Cascading Style Sheets
4	DOM	Document Object Model
5	HTML	Hypertext Markup Language
6	HTTP	Hypertext Transfer Protocol
7	I/O	Input/Output
8	JS	JavaScript
9	JSON	JavaScript Object Notation
10	LLM	Large Language Model
11	MERN	MongoDB, Express, React, NodeJS
12	NoSQL	Non-Relational SQL
13	REST	Representational State Transfer
14	UI	User Interface
15	URL	Uniform Resource Locator

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Vấn đề nghiên cứu

Đồ án này nghiên cứu các công nghệ web như ReactJS, NodeJS, cơ sở dữ liệu NoSQL và mô hình ngôn ngữ lớn (OpenAI) để phát triển một website đăng ký khóa học ngoại ngữ. Đồ án hướng đến việc số hóa quy trình hoạt động của trung tâm ngoại ngữ, từ khâu tư vấn, tra cứu thông tin đến đăng ký khóa học.

Các hướng tiếp cận

Đồ án được thực hiện dựa trên việc nghiên cứu lý thuyết, tài liệu về các công nghệ và công cụ liên quan. Áp dụng các lý thuyết đã nghiên cứu để xây dựng website với đầy đủ các yêu cầu, đồng thời phát triển được tính năng trợ lý ảo AI hỗ trợ tư vấn khóa học. Bên cạnh đó, đồ án còn khảo sát website hiện có của các trung tâm ngoại ngữ để phân tích quy trình đăng ký khóa học, cách tổ chức thông tin và thiết kế giao diện. Từ đó rút ra bài học và định hướng cải tiến.

Cách giải quyết vấn đề

Đồ án tiến hành thiết kế cơ sở dữ liệu với các collection chính: users, courses, teachers, registrations,... Xác định và thiết kế các giao diện của website: Trang chủ, trang khóa học, chi tiết khóa học, khóa học đã đăng ký,... và trang quản trị. Xác định và lập trình các chức năng của website: Đăng nhập/Đăng xuất, xem/xóa/sửa người dùng, khóa học, đăng ký khóa học,... Đồng thời tìm hiểu cách tích hợp mô hình ngôn ngữ lớn (OpenAI) vào website, từ đó xây dựng trợ lý ảo AI.

Kết quả đạt được

Kết quả của đồ án là xây dựng thành công một website đăng ký học ngoại ngữ đáp ứng đầy đủ các yêu cầu chức năng đã đề ra từ chức năng của người dùng đến chức năng quản lý của quản trị viên. Hệ thống cho phép học viên dễ dàng tìm kiếm, đăng ký và quản lý các khóa học ngoại ngữ một cách thuận tiện.

Giao diện website được thiết kế trực quan, thân thiện với người dùng, giúp học viên và quản trị viên thao tác nhanh chóng. Bên cạnh đó, tính năng trợ lý ảo AI hoạt động ổn định, phản hồi nhanh và tương đối chính xác, góp phần nâng cao trải nghiệm người dùng và hỗ trợ công tác tư vấn khóa học cho trung tâm ngoại ngữ một cách hiệu quả hơn.

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, tất cả các ngành nghề đều phải thay đổi để thích nghi và ứng dụng công nghệ để rút ngắn thời gian làm việc, tăng năng suất công việc, từ đó tiết kiệm chi phí, nâng cao năng lực cạnh tranh. Vì thế, nhiều trung tâm ngoại ngữ đã chọn giải pháp thay thế quy trình đăng ký bằng giấy tờ tốn quá nhiều thời gian, nhân lực, đôi khi mắc sai sót sang sử dụng hệ thống hiện đại hơn, dễ sử dụng hơn, ít mắc sai sót và ít tốn kém hơn cụ thể là website đăng ký học ngoại ngữ.

Một website đăng ký học ngoại ngữ đáp ứng được yêu cầu cấp bách hiện nay là số hóa quy trình đăng ký nhưng giải pháp này chưa triệt để, khâu quan trọng nhất đó là tư vấn khóa học để người dùng quyết định có đăng ký học hay không vẫn còn phải thực hiện thủ công bởi các nhân viên tư vấn. Với nhu cầu học ngoại ngữ tăng cao như hiện nay, một website của một trung tâm ngoại ngữ lớn có thể phải phục vụ cho một số lượng người dùng lớn trong một khoảng thời gian nhất định, gây sức ép lên đội ngũ nhân viên. Do đó tôi quyết định chọn đề tài “Xây dựng website đăng ký học ngoại ngữ tích hợp trợ lý ảo AI hỗ trợ tư vấn khóa học” để tạo ra một website đáp ứng đầy đủ yêu cầu của quy trình đăng ký khóa học từ tư vấn tự động, đăng ký đến thanh toán. Không chỉ giải quyết vấn đề tồn đọng, đề tài còn ứng dụng được trí thông minh nhân tạo vào sản phẩm thực tế, làm tiền đề để ứng dụng vào các sản phẩm sau này.

2. Mục tiêu

Mục tiêu của đề tài là xây dựng một website đăng ký học ngoại ngữ nhằm số hóa quy trình đăng ký và tư vấn khóa học cho trung tâm ngoại ngữ. Hệ thống giúp học viên dễ dàng tìm hiểu thông tin, nhận tư vấn tự động và đăng ký khóa học trực tuyến một cách thuận tiện.

Đồng thời, đề tài hướng đến việc xây dựng một sản phẩm hoàn chỉnh, có tính ứng dụng cao, giao diện thân thiện, hoạt động ổn định và có khả năng mở rộng trong tương lai. Ngoài ra, việc tích hợp trợ lý ảo AI còn giúp ứng dụng trí tuệ nhân tạo vào sản phẩm thực tế, nâng cao chất lượng dịch vụ tư vấn và giảm tải cho đội ngũ nhân viên.

3. Nội dung

Nội dung của đề tài tập trung vào việc phân tích và thiết kế hệ thống website đăng ký học ngoại ngữ tích hợp trợ lý ảo AI. Cụ thể, đề tài tiến hành xây dựng giao diện người dùng cho website, thiết kế và triển khai hệ thống backend, cơ sở dữ liệu phục vụ quản lý người dùng, khóa học, đăng ký học,... Bên cạnh đó, đề tài nghiên cứu và tích hợp mô hình ngôn ngữ lớn OpenAI để xây dựng trợ lý ảo AI hỗ trợ tư vấn khóa học, giải đáp thắc mắc cho người dùng. Cuối cùng, hệ thống được kiểm thử, đánh giá nhằm đảm bảo đáp ứng đúng yêu cầu chức năng, bảo mật và trải nghiệm người dùng.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

Website hiện có của các trung tâm ngoại ngữ, quy trình đăng ký khóa học.

Nghiên cứu ReactJS + Vite, Tailwind CSS để thiết kế giao diện.

Nghiên cứu NodeJS + Express, MongoDB,... để lập trình chức năng.

Nghiên cứu mô hình ngôn ngữ lớn (OpenAI) để xây dựng tính năng trợ lý ảo.

Phạm vi nghiên cứu

Tập trung xây dựng các chức năng theo yêu cầu, các cơ chế bảo mật website, trải nghiệm người dùng.

5. Phương pháp nghiên cứu

Đề tài sử dụng phương pháp nghiên cứu tài liệu nhằm tìm hiểu các công nghệ, công cụ và mô hình liên quan đến phát triển website và trí tuệ nhân tạo. Các tài liệu tham khảo bao gồm bài viết khoa học, tài liệu kỹ thuật, nguồn trực tuyến,...

Phương pháp phân tích và tổng hợp được áp dụng để khảo sát các website đăng ký học ngoại ngữ hiện có, từ đó rút ra ưu điểm, hạn chế và định hướng cải tiến cho hệ thống.

Ngoài ra, phương pháp thực nghiệm được sử dụng trong quá trình xây dựng và kiểm thử website nhằm đánh giá hiệu quả hoạt động của hệ thống, đặc biệt là tính năng trợ lý ảo AI và khả năng đáp ứng nhu cầu của người dùng.

CHƯƠNG 1: TỔNG QUAN

Công nghệ ngày nay đang trở thành một phần không thể thiếu trong đời sống hiện đại. Việc nhấn mạnh vai trò của công nghệ không nhằm thể hiện sự phụ thuộc của con người, mà hướng đến việc ứng dụng công nghệ một cách hợp lý và hiệu quả trong cuộc sống cũng như trong công việc. Nhờ sự hỗ trợ của công nghệ, nhiều quy trình được tự động hóa, các công việc được thực hiện chính xác hơn, nhanh chóng hơn, đồng thời giảm thiểu chi phí vận hành và tối ưu hóa nguồn lực.

Trong bối cảnh đó, việc xây dựng một website quản lý đăng ký học ngoại ngữ là hết sức cần thiết và phù hợp với xu hướng phát triển chung. Website không chỉ giúp số hóa quy trình đăng ký học, thay thế cho các thủ tục thủ công rườm rà như đến trực tiếp trung tâm để tìm hiểu thông tin khóa học, học phí và làm hồ sơ đăng ký, mà còn góp phần giải quyết bài toán về nhân sự. Khi số lượng học viên đăng ký tăng cao, hệ thống có thể đáp ứng đồng thời nhiều người dùng mà không gây quá tải; ngược lại, khi nhu cầu giảm, trung tâm vẫn duy trì hoạt động ổn định mà không lãng phí nguồn lực. Bên cạnh đó, việc triển khai website còn giúp trung tâm dễ dàng quảng bá hình ảnh, tiếp cận nhiều đối tượng học viên hơn trong thời đại mà hầu hết mọi người đều sử dụng các thiết bị thông minh và Internet.

Bài toán đặt ra là xây dựng được một website vừa thân thiện với người dùng, vừa mang lại trải nghiệm thao tác và đăng ký học mượt mà, thuận tiện. Hệ thống cần có trang quản trị đầy đủ chức năng như quản lý dữ liệu, xem, sửa, xóa và thống kê nhằm hỗ trợ công tác quản lý hiệu quả. Đặc biệt, việc tích hợp chức năng trợ lý ảo AI có khả năng tư vấn chính xác, linh hoạt và thân thiện sẽ đóng vai trò quan trọng trong việc hỗ trợ học viên lựa chọn khóa học phù hợp, nâng cao chất lượng dịch vụ và tạo điểm khác biệt cho hệ thống.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Tìm hiểu về RESTful API

2.1.1 Tổng quan về RESTful API

REST (Representational State Transfer) là một kiểu kiến trúc phần mềm được sử dụng phổ biến trong các hệ thống phân tán, hướng đến mục tiêu dễ mở rộng, dễ bảo trì và tận dụng hiệu quả giao thức HTTP. API (Application Programming Interface) đóng vai trò là giao diện trung gian cho phép các phần mềm hoặc hệ thống khác nhau giao tiếp với nhau mà không cần quan tâm đến cách thức xử lý bên trong. Từ đó, RESTful API được hiểu là một chuẩn thiết kế API dành cho các ứng dụng web, tuân thủ các ràng buộc của kiến trúc REST và sử dụng các phương thức của giao thức HTTP để quản lý và thao tác trên các tài nguyên [1].

2.1.2 Các thành phần chính

Một hệ thống RESTful API bao gồm các yếu tố chính sau:

Tài nguyên: Mọi đối tượng dữ liệu (người dùng, bài viết...) được xem là tài nguyên và định danh bằng một URL duy nhất.

Phương thức HTTP: Các hành động thao tác với tài nguyên:

GET: Lấy dữ liệu.

POST: Tạo mới dữ liệu.

PUT: Cập nhật toàn bộ dữ liệu.

PATCH: Cập nhật một phần dữ liệu.

DELETE: Xóa dữ liệu.

Endpoint: Đường dẫn URL nơi client gửi yêu cầu đến (Ví dụ: /users).

Request và response:

Request: Gồm URL, phương thức HTTP, header (token xác thực) và body.

Response: Gồm mã trạng thái, dữ liệu (JSON/XML) và header.

Mã trạng thái: Thông báo kết quả xử lý (Ví dụ: 200 OK - Thành công, 404 Not Found - Không tìm thấy, 500 Internal Server Error - Lỗi server).

2.1.3 Nguyên lý hoạt động

Quy trình giao tiếp diễn ra theo mô hình Client-Server qua 3 bước:

1. Client gửi Request: Sử dụng phương thức HTTP (GET, POST,...) gửi đến một Endpoint cụ thể, kèm theo header (như Access Token). RESTful API thường dùng access_token thay vì session/cookie.

2. Server xử lý: Xác thực, truy vấn tài nguyên và xử lý logic.

3. Server phản hồi: Trả về dữ liệu (thường là định dạng JSON vì tính gọn nhẹ) kèm mã trạng thái HTTP.

2.1.4 Tiêu chí của một API chuẩn RESTful

Để được coi là RESTful, API cần tuân thủ các nguyên tắc:

Stateless: Server không lưu thông tin client giữa các request; mỗi request phải độc lập và chứa đầy đủ thông tin.

Client-Server: Tách biệt giao diện và xử lý logic.

Cacheable: Hỗ trợ lưu cache để tăng hiệu suất.

Uniform Interface: Giao diện thống nhất, tài nguyên được định danh chuẩn qua URL.

Layered System: Hệ thống phân lớp (bảo mật, cân bằng tải...).

2.2 Tìm hiểu về NodeJS

2.2.1 Tổng quan về NodeJS

NodeJS là một môi trường thực thi JavaScript mã nguồn mở và đa nền tảng, cho phép chạy mã JavaScript bên ngoài trình duyệt (ví dụ như trên máy chủ). NodeJS không phải là một ngôn ngữ lập trình (nó dùng ngôn ngữ JavaScript) và cũng không phải là một web framework (thường bị nhầm lẫn với ExpressJS hay NestJS).

NodeJS được xây dựng dựa trên V8 JavaScript Engine của Google Chrome. V8 có khả năng biên dịch mã JavaScript trực tiếp sang mã máy, giúp tốc độ thực thi rất nhanh.

NodeJS được phát triển bởi Ryan Lienhart Dahl vào năm 2009. Ban đầu, dự án có tên là “web.js” với mục tiêu thay thế các web server truyền thống như Apache. Tuy nhiên, khi dự án phát triển vượt xa vai trò của một web server đơn thuần và trở thành nền tảng đa mục đích, nó được đổi tên thành “Node.JS” [2].

2.2.2 Cơ chế hoạt động và kiến trúc

Sức mạnh của NodeJS đến từ kiến trúc đặc biệt giúp nó xử lý hiệu quả các tác vụ I/O:

Non-blocking I/O: Khác với mô hình truyền thống (như PHP hay ASP) phải chờ một yêu cầu hoàn tất mới xử lý yêu cầu tiếp theo, NodeJS gửi yêu cầu đi (như đọc file, truy vấn database) và tiếp tục xử lý việc khác ngay lập tức mà không cần chờ kết quả.

Single-Threaded và Event Loop: NodeJS hoạt động trên một luồng duy nhất cho logic ứng dụng. Tuy nhiên, nhờ cơ chế Event Loop, nó có thể xử lý đồng thời nhiều yêu cầu. Khi một tác vụ I/O hoàn tất, hệ thống sẽ gửi một sự kiện để kích hoạt hàm phản hồi tương ứng.

Lập trình hướng sự kiện: Chương trình không chạy theo trình tự cố định mà sẽ chờ các sự kiện (như click chuột, request từ client) xảy ra để gọi hàm xử lý.

2.2.3 Các thành phần quan trọng

NPM (Node Package Manager): Đây là trình quản lý gói của NodeJS và cũng là kho lưu trữ mã nguồn mở lớn nhất thế giới cho JavaScript, chứa hơn 50.000 package giúp lập trình viên tái sử dụng mã dễ dàng.

Modules: Các gói thư viện nhỏ chứa hàm/đối tượng cụ thể (ví dụ: ‘http’ để tạo server, ‘fs’ để làm việc với file).

Cluster: Module giúp tạo ra nhiều tiến trình con để tận dụng các hệ thống đa nhân, khắc phục hạn chế đơn luồng của NodeJS.

2.3 Tìm hiểu về ExpressJS

2.3.1 Tổng quan về ExpressJS

ExpressJS là một framework ứng dụng web được xây dựng trên nền tảng NodeJS. Đây là một công cụ phổ biến giúp tổ chức và sắp xếp mã nguồn hiệu quả,

thường được sử dụng để phát triển các ứng dụng web hoặc ứng dụng di động. ExpressJS cung cấp các tính năng mạnh mẽ để hỗ trợ các phương thức HTTP và tạo ra các API để sử dụng [3].

2.3.2 Chức năng

ExpressJS tập trung vào các nhiệm vụ chính sau trong phát triển backend:

Xử lý HTTP Request: Thiết lập các lớp trung gian để tiếp nhận và trả về các yêu cầu HTTP.

Định tuyến: Xác định các hành động cụ thể dựa trên phương thức HTTP (GET, POST,...) và đường dẫn URL.

Rendering: Hỗ trợ trả về các trang HTML động dựa trên các tham số đầu vào.

2.4 Tìm hiểu về MongoDB

2.4.1 Tổng quan về MongoDB

MongoDB là một cơ sở dữ liệu hướng tài liệu, thuộc nhóm cơ sở dữ liệu NoSQL. Khác với các cơ sở dữ liệu quan hệ truyền thống sử dụng cấu trúc bảng, MongoDB lưu trữ dữ liệu dưới dạng các tài liệu JSON với một lược đồ rất linh hoạt gọi là BSON.

Đặc điểm cốt lõi của MongoDB là mỗi “collection” (tương đương bảng trong SQL) có thể chứa các document với kích cỡ và cấu trúc khác nhau, cho phép truy vấn dữ liệu rất nhanh chóng [4].

2.4.2 Lịch sử hình thành

MongoDB được ra đời lần đầu tiên vào tháng 10 năm 2007 bởi MongoDB Inc. (lúc đó tên là 10gen). Ban đầu, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Google App Engine hay Windows Azure. Đến năm 2009, MongoDB chính thức được chuyển thành mã nguồn mở.

Hiện nay, MongoDB đã trở thành một trong những cơ sở dữ liệu NoSQL nổi bật nhất, được sử dụng làm backend cho nhiều hệ thống lớn như eBay, The New York Times và SourceForge.

2.5 Tìm hiểu về ReactJS

2.5.1 Tổng quan về ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển và vận hành bởi Facebook, dùng để xây dựng giao diện người dùng cho website.

ReactJS sử dụng kiến trúc component-based, nghĩa là giao diện được chia thành các thành phần nhỏ, độc lập và có thể tái sử dụng.

React áp dụng Virtual DOM là một bản sao nhẹ của DOM thực giúp tối ưu quá trình render. Khi dữ liệu thay đổi, React chỉ vẽ lại những phần cần thay đổi, không render lại toàn bộ trang [5].

2.5.2 Một số tính năng

React cho phép dùng JSX - một cú pháp mở rộng của JavaScript giúp code UI giống HTML, dễ đọc hơn.

One-way data binding: Dữ liệu đi theo một chiều từ component cha đến component con. Điều này giúp dữ liệu và trạng thái trong ứng dụng dễ kiểm soát hơn.

Hiệu năng cao: Nhờ Virtual DOM và chỉ cập nhật phần thay đổi, React giúp UI nhanh, mượt, giảm tải cho DOM thực.

Phù hợp xây SPA (Single-Page Applications): React rất được ưa chuộng để xây các ứng dụng một trang nơi nội dung thay đổi mà không cần tải lại trang.

Tái sử dụng code và dễ bảo trì: Component-based và modular giúp ứng dụng lớn dễ quản lý, code dễ đọc, dễ bảo trì và mở rộng hơn.

2.6 Tìm hiểu về Vite

2.6.1 Tổng quan về Vite

Vite là một công cụ build / dev-server dành cho phát triển web hiện đại, được tạo bởi Evan You, người tạo ra Vue.js.

Trong suốt quá trình phát triển, Vite phục vụ mã cục bộ bằng cách sử dụng ES modules native, tức là trình duyệt sẽ tải từng module riêng lẻ, không cần đóng gói toàn bộ trước.

Khi build để deploy (production), Vite sẽ đóng gói toàn bộ mã nguồn (JS, CSS,...) để tối ưu, thông thường sử dụng công cụ bundler nhanh là esbuild hoặc Rollup tùy cấu hình [6].

2.6.2 Giải pháp về hiệu suất của Vite

Vì không bundle toàn bộ ở dev, việc khởi động server và load trang lần đầu rất nhanh, ngay cả với dự án lớn.

Hot Module Replacement (HMR) nhanh và hiệu quả: Khi sửa code, Vite chỉ reload module bị thay đổi mà không reload toàn bộ giúp quá trình phát triển nhanh hơn.

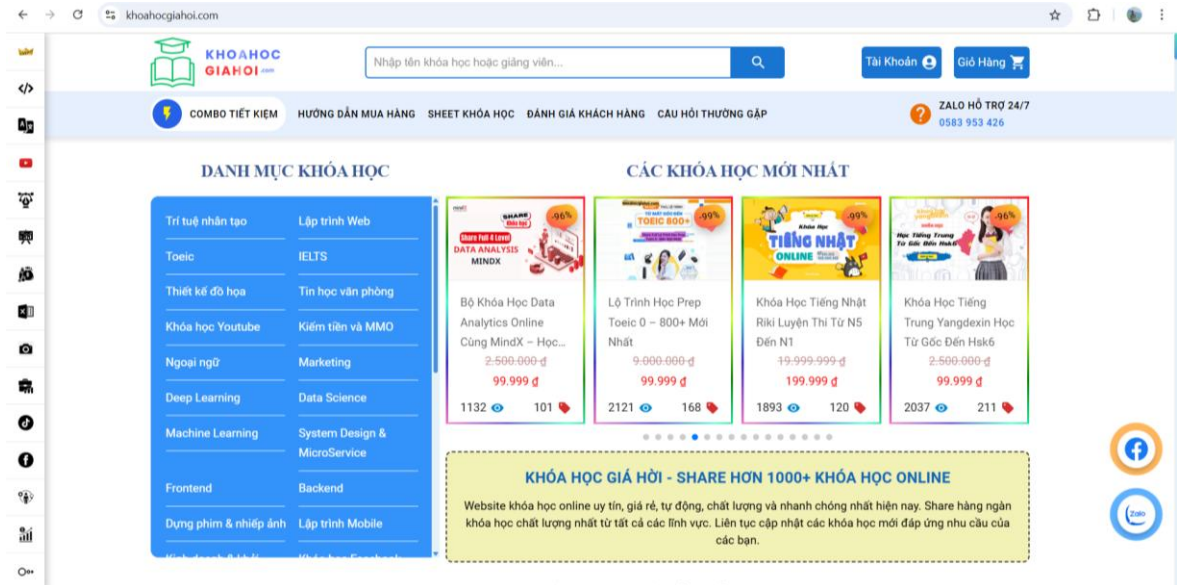
Khi build cho production, Vite tạo mã được tối ưu, hiệu quả nhờ bundler hiện đại, giúp tải và chạy web nhanh hơn.

2.7 Tìm hiểu về Tailwind CSS

Tailwind CSS là một framework CSS, thay vì cung cấp sẵn các component hoặc giao diện (button, card, form,...) như các framework truyền thống, Tailwind cung cấp một bộ rất lớn các class tiện ích (utility classes) nhỏ, mỗi class tương ứng một thuộc tính CSS. Khi xây dựng giao diện, lập trình viên kết hợp các class này trực tiếp trong HTML để đặt style từ margin, padding, màu sắc, font, layout, responsive,... giúp tăng tốc độ phát triển UI [7].

2.8 Khảo sát các website cùng lĩnh vực

2.8.1 Khảo sát website khoa hoc gia hoi.com



Hình 1: Khảo sát website khoa hoc gia hoi.com

Giao diện

Bố cục dạng cửa hàng bán khóa học, đơn giản, trực quan.

Màu sắc nhẹ nhàng cách trình bày khá rối mắt.

Các mục như Trang chủ, Danh mục khóa học, Mua khóa học, Giới thiệu được bố trí hợp lý.

Hình ảnh khóa học lớn, nội dung hiển thị gọn nhưng thông tin không nhiều.

Chức năng

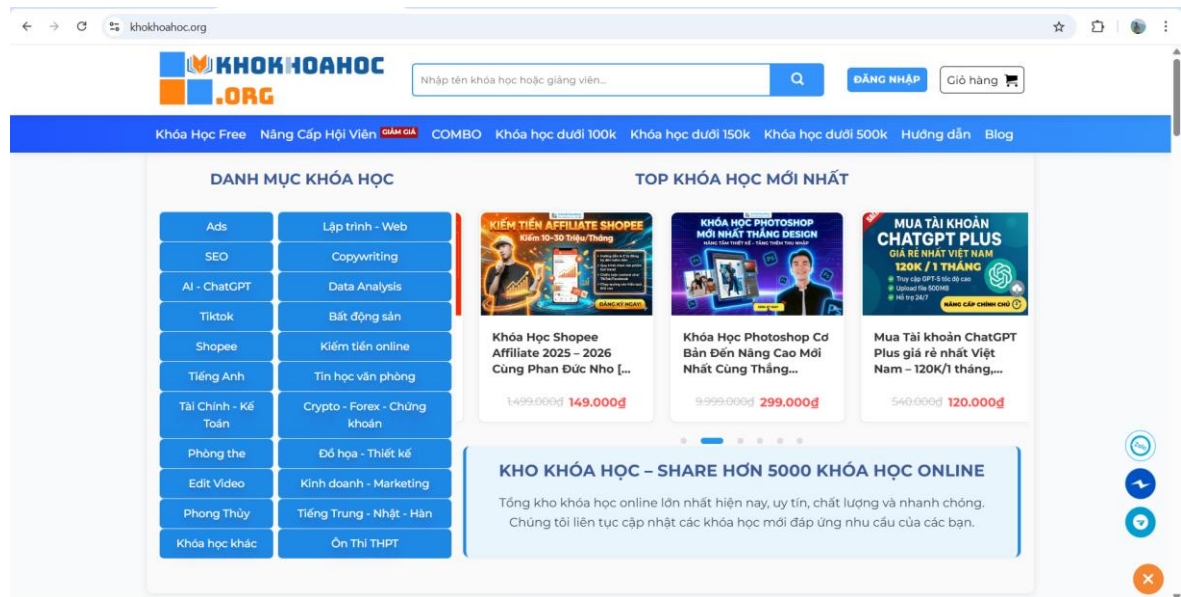
Danh mục khóa học chia theo chủ đề, người dùng chọn và xem khóa học nhanh.

Có giỏ hàng, đặt mua, thanh toán, thao tác tương tự một trang thương mại điện tử.

Mỗi khóa học có mô tả ngắn, giá, ưu đãi,...

Chức năng tìm kiếm đơn giản (giá, tác giả, chức năng sắp xếp,...).

2.8.2 Khảo sát website khokhoahoc.org



Hình 2: Khảo sát website khokhoahoc.org

Giao diện

Bố cục dạng marketplace khóa học, nhiều danh mục đặt bên trái.

Giao diện khá nhiều thông tin nhưng hơi rối, nhiều mục, nhiều banner.

Ảnh khóa học được hiển thị theo dạng danh sách, khá thuận tiện để duyệt theo số lượng lớn.

Có nhiều phân loại như:

Khóa học theo chủ đề

Combo khóa

Khóa học giá rẻ / miễn phí

Chức năng

Chức năng giỏ hàng, thêm vào yêu thích, mua khóa học đầy đủ.

Có phân theo danh mục và bộ lọc.

Hiển thị giá, mô tả khóa học, thời lượng khá rõ.

2.8.3 Rút ra kinh nghiệm cho đề tài

Nên kế thừa bố cục danh mục khóa học, cách hiển thị khóa học rõ ràng, các trang thành phần như chi tiết khóa học, liên hệ,...

Tập trung vào chức năng cho đăng ký học ngoại ngữ như:

Đăng ký khóa theo lịch, ngôn ngữ, trình độ, giảng viên,...

Xem chi tiết khóa học.

Chức năng lọc, tìm kiếm, sắp xếp khóa học.

Phát triển tính năng tư vấn tự động bằng AI.

2.9 Các nghiệp vụ liên quan đến đề tài

Nghiệp vụ quản lý người dùng

Đăng ký tài khoản học viên.

Đăng nhập, đăng xuất hệ thống.

Quản lý thông tin cá nhân học viên.

Phân quyền người dùng (học viên, quản trị viên).

Nghiệp vụ tư vấn khóa học

Tiếp nhận nhu cầu học tập của người dùng.

Phân tích thông tin và đề xuất khóa học phù hợp.

Trả lời câu hỏi, giải đáp thắc mắc về khóa học thông qua trợ lý ảo AI.

Hỗ trợ học viên ra quyết định đăng ký khóa học.

Nghiệp vụ đăng ký khóa học

Tìm kiếm khóa học phù hợp.

Đăng ký khóa học trực tuyến.

Lưu thông tin đăng ký của học viên.

Xem danh sách các khóa học đã đăng ký.

Hủy hoặc chỉnh sửa đăng ký.

Nghiệp vụ thanh toán

Thanh toán online (VNPay) và thanh toán tại trung tâm.

Tạo và áp dụng khuyến mãi.

Nghiệp vụ quản trị hệ thống

Quản lý người dùng, giảng viên, khóa học, đăng ký học,...

Thống kê số lượng học viên, khóa học, lượt đăng ký.

Theo dõi hoạt động của hệ thống.

Quản lý nội dung hiển thị trên website.

2.10 Các công trình nghiên cứu liên quan

“Chatbot hiệu quả cho tư vấn tuyển sinh đại học sử dụng mô hình ngôn ngữ lớn”

Nghiên cứu này giới thiệu một hệ thống chatbot thông minh hỗ trợ tư vấn tuyển sinh đại học dựa trên việc ứng dụng các Mô hình Ngôn ngữ Lớn (LLM) kết hợp với kỹ thuật Tạo sinh Tăng cường Truy xuất (RAG). Hệ thống được xây dựng bằng cách khai thác dữ liệu thực tế từ Trường Đại học Đông Á và áp dụng chiến lược truy xuất hỗn hợp (hybrid retrieval) giữa tìm kiếm theo độ tương đồng véc-tơ và so khớp từ khóa để đảm bảo câu trả lời chính xác, giàu ngữ cảnh. Nhóm tác giả đã tích hợp các mô hình ngôn ngữ tiên tiến như LLaMA 3, Gemini 1.5 và Grok, đồng thời lựa chọn mô hình nhúng nomic-embed-text-v1.5 để tối ưu hóa khả năng hiểu tiếng Việt cũng như hiệu quả sử dụng tài nguyên. Bên cạnh đó, việc áp dụng hệ thống quản lý bộ nhớ MemGPT và khung làm việc Chainlit giúp chatbot duy trì ngữ cảnh dài hạn, cá nhân hóa phản hồi và cung cấp giao diện tương tác thân thiện cho thí sinh. Kết quả thử nghiệm khẳng định hệ thống không chỉ cải thiện độ chính xác trong truy xuất thông tin mà còn giúp giảm đáng kể khối lượng công việc cho nhân viên tư vấn. Mặc dù hiện tại dữ liệu còn giới hạn ở một trường đại học, nhưng định hướng tương lai của nghiên cứu là mở rộng sang mô hình đa trường học, tích hợp hỗ trợ đa ngôn ngữ và tính năng tương tác bằng giọng nói [8].

“Development of an AI-Powered Faculty Support Chatbot”

Nghiên cứu này trình bày quá trình thiết kế và triển khai SOC Buddy, một chatbot hỗ trợ khoa dựa trên trí tuệ nhân tạo (AI) tại Trường Công nghệ thông tin thuộc Đại học Công nghệ Liên bang, Akure, Nigeria. Mục tiêu chính của dự án là giải quyết các thách thức mà sinh viên và ứng viên gặp phải khi truy cập thông tin học thuật, đồng thời giảm bớt áp lực công việc cho các cố vấn cấp lớp. Hệ thống được phát triển bằng cách sử dụng các mô hình ngôn ngữ lớn (LLM) như GPT-3.5

Turbo, kết hợp với khung làm việc LangChain và cơ sở dữ liệu vector Pinecone để tạo ra một kho kiến thức chuyên biệt. Kết quả đánh giá hiệu năng cho thấy SOC Buddy đạt độ chính xác 92%, thời gian phản hồi trung bình nhanh chóng trong khoảng 2,5 giây và có khả năng giải quyết thành công 88% truy vấn mà không cần sự can thiệp của con người. Với tỷ lệ hài lòng của người dùng đạt 85%, nghiên cứu kết luận rằng việc triển khai chatbot AI không chỉ cải thiện khả năng tiếp cận hỗ trợ 24/7 mà còn đảm bảo tính nhất quán trong việc cung cấp thông tin, đóng vai trò như một mô hình hiệu quả để hiện đại hóa hệ thống hỗ trợ giáo dục đại học [9].

“Nghiên cứu xây dựng Chatbot AI: Tinbot 10 và đề xuất một số biện pháp sư phạm sử dụng Tinbot 10 để hỗ trợ dạy học lập trình Python”

Nghiên cứu của Trương Thị Đào và Kiều Phương Thùy tập trung vào việc xây dựng chatbot AI mang tên TinBot 10 trên nền tảng Poe AI nhằm hỗ trợ giảng dạy lập trình Python cho học sinh lớp 10 (Chủ đề F), giúp giải quyết những khó khăn phổ biến như việc hiểu cú pháp, gỡ lỗi và vận dụng thuật toán vào thực tiễn. Dựa trên khảo sát thực trạng từ 120 giáo viên và 594 học sinh, nhóm tác giả đã đề xuất quy trình xây dựng TinBot và các biện pháp sư phạm đi kèm, bao gồm: đưa ra gợi ý theo cấp độ thay vì đáp án trực tiếp, khuyến khích học sinh “học từ lỗi sai”, kiểm soát việc sử dụng AI và áp dụng phương pháp dạy học theo tình huống. Kết quả thực nghiệm trên 91 học sinh cho thấy nhóm sử dụng TinBot 10 đạt điểm trung bình 8.04, cao hơn đáng kể so với nhóm đối chứng đạt 7.07 (với $p < 0.05$), khẳng định chatbot AI này là một trợ giảng ảo hiệu quả trong việc cá nhân hóa học tập và cải thiện rõ rệt kết quả học tập của học sinh. Nghiên cứu không chỉ cung cấp cơ sở khoa học cho việc ứng dụng AI trong giáo dục phổ thông mà còn mở ra hướng đi mới cho việc dạy học các ngôn ngữ lập trình khác trong thời đại số [10].

“Development Of Online Course Registration Management System”

Nghiên cứu này tập trung vào việc xây dựng một hệ thống quản lý đăng ký khóa học trực tuyến nhằm thay thế các phương thức đăng ký thủ công truyền thống vốn tốn thời gian, dễ gây sai sót và đòi hỏi nhiều không gian lưu trữ hồ sơ giấy. Hệ thống được phát triển bằng cách sử dụng ngôn ngữ PHP cho giao diện phía trước (frontend) và MySQL để quản trị cơ sở dữ liệu phía sau (backend), kết hợp cùng HTML, CSS và JavaScript để tạo ra một nền tảng phản hồi nhanh, bảo mật và thân

thiện với người dùng. Mô hình hoạt động dựa trên hai phân hệ riêng biệt: quản trị viên có quyền quản lý toàn bộ các hoạt động, từ việc tạo khóa học đến cấp mã định danh duy nhất cho sinh viên; trong khi đó, sinh viên có thể đăng nhập bằng thông tin được cấp để tìm kiếm, đăng ký khóa học mong muốn và in đơn đăng ký trực tiếp từ hệ thống. Kết quả nghiên cứu cho thấy việc áp dụng hệ thống này không chỉ giúp tự động hóa quy trình đăng ký và theo dõi thông tin theo thời gian thực, mà còn tăng cường tính bảo mật, giảm thiểu khối lượng công việc hành chính và nâng cao năng suất cho các cơ sở giáo dục hoặc tổ chức đào tạo [11].

“Online Course Registration Management System: A Comprehensive Analysis”

Nghiên cứu về Hệ thống quản lý đăng ký khóa học trực tuyến (OCRMS) tập trung vào việc chuyển đổi quy trình đăng ký môn học từ phương pháp thủ công, sử dụng giấy tờ truyền thống sang một nền tảng kỹ thuật số hiện đại và hiệu quả. Hệ thống này được xây dựng dựa trên các công nghệ như HTML, PHP, MySQL, Bootstrap và Xampp, cho phép sinh viên đăng ký khóa học từ bất cứ đâu thông qua internet, đồng thời giúp các tổ chức giáo dục quản lý dữ liệu một cách tập trung và chính xác. Cấu trúc của hệ thống bao gồm hai cổng chính: cổng sinh viên (để đăng ký, quản lý hồ sơ và xem lịch sử ghi danh) và cổng quản trị/giáo viên (để quản lý khóa học, sinh viên và các phiên học). Việc triển khai hệ thống không chỉ giúp giảm bớt các thủ tục hành chính rườm rà, hạn chế sai sót mà còn cung cấp khả năng phân tích dữ liệu về xu hướng ghi danh, hỗ trợ đắc lực cho việc ra quyết định chiến lược. Trong tương lai, nghiên cứu hướng tới việc tích hợp trí tuệ nhân tạo (AI) để gợi ý khóa học cá nhân hóa, bổ sung cổng thanh toán trực tuyến và phát triển ứng dụng di động để tăng tính tiếp cận [12].

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả hệ thống

Đề tài yêu cầu xây dựng một website quản lý và đăng ký khóa học ngoại ngữ cho trung tâm với các chức năng chính: đăng nhập, đăng ký tài khoản, xem danh sách khóa học, tìm kiếm, xem chi tiết khóa học, đăng ký khóa học trực tuyến, đánh giá khóa học và đặc biệt là tích hợp trợ lý ảo AI để hỗ trợ tư vấn tự động.

Hệ thống quản lý các khóa học với các thông tin chi tiết như: mã khóa học, mô tả, học phí, mức giảm giá, số tiết học, ngày bắt đầu, ngày kết thúc và ảnh bìa. Mỗi khóa học sẽ thuộc về một ngôn ngữ và một trình độ cụ thể để người học dễ dàng lựa chọn. Ngoài ra, khóa học còn được gắn với thông tin về ca học bao gồm ngày học trong tuần và giờ học cụ thể.

Mỗi khóa học do một giảng viên phụ trách giảng dạy. Thông tin giảng viên bao gồm mã giảng viên, họ tên, email, giới tính và ngôn ngữ chuyên môn mà giảng viên đó phụ trách.

Người dùng khi truy cập hệ thống có thể đăng ký tài khoản thành viên. Thông tin người dùng bao gồm mã người dùng, họ tên, email, mật khẩu, địa chỉ, giới tính và ảnh đại diện. Khi đã đăng nhập, người dùng có thể thực hiện đăng ký khóa học. Hệ thống sẽ lưu trữ thông tin đăng ký gồm ngày đăng ký, trạng thái thanh toán, hình thức thanh toán, buổi học, mã giảm giá (nếu có) và trạng thái lớp. Sau khi hoàn thành khóa học, người dùng có thể để lại đánh giá bao gồm điểm xếp hạng (số sao) và nội dung bình luận để góp ý chất lượng.

Ngoài các chức năng thông thường, hệ thống còn tích hợp Trợ lý ảo AI. Người dùng có thể trò chuyện với AI để nhận tư vấn về khóa học, gợi ý khóa học phù hợp dựa trên nhu cầu hoặc giải đáp các thắc mắc chung mà không cần đợi nhân viên tư vấn. Bên cạnh đó, người dùng vắng lai cũng có thể gửi thông tin liên hệ trực tiếp qua biểu mẫu nếu cần hỗ trợ riêng.

Người quản trị có toàn quyền trên hệ thống, thực hiện các chức năng quản lý như: quản lý danh sách người dùng, quản lý giảng viên, thêm/xóa/sửa thông tin khóa học, ngôn ngữ, trình độ, quản lý các ca học, phê duyệt mở lớp và xem thống kê doanh thu và các thông tin liên quan.

3.2 Xác định các yêu cầu chức năng của hệ thống

Hệ thống website đăng ký học ngoại ngữ tích hợp trợ lý ảo AI cần đáp ứng các yêu cầu chức năng chính sau:

Quản lý người dùng: Cho phép người dùng đăng ký, đăng nhập, đăng xuất; quản lý thông tin cá nhân và lịch sử đăng ký khóa học.

Quản lý khóa học: Hiển thị danh sách khóa học ngoại ngữ theo ngôn ngữ, trình độ, học phí, thời gian học và giảng viên.

Đăng ký khóa học: Cho phép học viên lựa chọn và đăng ký khóa học trực tuyến, xem trạng thái đăng ký.

Thanh toán và mã giảm giá: Cho phép học viên thanh toán qua 2 hình thức trực tuyến và trực tiếp tại trung tâm; với hình thức trực tuyến, cho phép áp dụng mã giảm giá (nếu có).

Đánh giá: Cho phép học viên quay lại đánh giá khóa học sau khi khóa học kết thúc.

Trợ lý ảo AI tư vấn khóa học: Hỗ trợ tư vấn, giải đáp thắc mắc và gợi ý khóa học phù hợp dựa trên nhu cầu của học viên.

Quản trị hệ thống: Thống kê hệ thống, quản lý khóa học, người dùng,...

Thông báo và hỗ trợ: Gửi thông báo xác nhận thanh toán thành công (kèm hóa đơn); cung cấp các kênh hỗ trợ trực tuyến cho người học như số điện thoại, zalo, email,... hoặc nhập thông tin liên hệ vào biểu mẫu có sẵn.

3.3 Thiết kế dữ liệu hệ thống

Mô hình cơ sở dữ liệu được xây dựng với 11 bảng như sau



Hình 3: Mô hình cơ sở dữ liệu

Bảng USER: Lưu trữ thông tin chi tiết của người dùng (học viên và quản trị viên).

Bảng 1: Mô tả bảng USER

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã định danh duy nhất của người dùng
2	userid	String	Mã người dùng hiển thị
3	username	String	Tên đăng nhập
4	password	String	Mật khẩu (đã mã hóa)
5	fullname	String	Họ và tên đầy đủ
6	email	String	Địa chỉ email
7	role	String	Vai trò (Admin/User)
8	address	String	Địa chỉ liên hệ
9	gender	String	Giới tính

10	avatar	String	Đường dẫn hình ảnh đại diện
11	googleId	String	Id do google cung cấp (nếu đăng nhập bằng google)
12	authType	String	Loại xác thực (local/google)

Bảng LANGUAGE: Danh mục các ngôn ngữ được giảng dạy (Tiếng Anh, Tiếng Nhật,...).

Bảng 2: Mô tả bảng LANGUAGE

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã định danh ngôn ngữ
2	languageid	String	Mã ngôn ngữ hiển thị
3	language	String	Tên ngôn ngữ

Bảng LANGUAGE_LEVEL: Danh mục các trình độ của khóa học.

Bảng 3: Mô tả bảng LANGUAGE_LEVEL

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã định danh trình độ
2	language_levelid	String	Mã trình độ hiển thị
3	language_level	String	Tên trình độ
4	language_id	FK- LANGUAGE	Mã ngôn ngữ

Bảng CLASSSESSION: Lưu trữ thông tin về các ca học (buổi học).

Bảng 4: Mô tả bảng CLASSSESSION

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
-----	----------------	--------------	---------

1	_id	ObjectId	Mã ca học
2	days	String	Các ngày học trong tuần (VD: 2-4-6)
3	time	String	Khung giờ học (VD: 17h30 - 19h00)

Bảng TEACHER: Lưu trữ thông tin hồ sơ của các giảng viên.

Bảng 5: Mô tả bảng TEACHER

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã định danh duy nhất của giảng viên
2	teacherid	String	Mã giảng viên hiển thị
3	full_name	String	Họ và tên giảng viên
4	email	String	Email liên hệ
5	gender	String	Giới tính
6	language_id	FK-LANGUAGE	Mã ngôn ngữ giảng dạy

Bảng COURSE: Lưu trữ thông tin về các khóa học ngoại ngữ.

Bảng 6: Mô tả bảng COURSE

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã định danh duy nhất của khóa học
2	courseid	String	Mã khóa học hiển thị
3	Description	String	Mô tả chi tiết nội dung khóa học
4	Tuition	Double	Học phí khóa học
5	Number_of_periods	Double	Số tiết học

6	discount_percent	Double	Phần trăm giảm giá (nếu có)
7	image	String	Đường dẫn ảnh đại diện khóa học
8	Start_Date	Date	Ngày bắt đầu khóa học
9	end_date	Date/String	Ngày kết thúc khóa học
10	languagelevel_id	FK- LANGUAGE _LEVEL	Mã trình độ
11	teacher_id	FK- TEACHER	Liên kết đến bảng giảng viên
12	views	Double	Số lượt xem khóa học

Bảng REGISTRATION_COURSE: Lưu trữ thông tin đăng ký khóa học của học viên.

Bảng 7: Mô tả bảng REGISTRATION_COURSE

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã đơn đăng ký
2	user_id	FK-USER	Mã người dùng đăng ký
3	course_id	FK-COURSE	Mã khóa học đăng ký
4	class_session_id	FK- CLASSESSION	Mã ca học đã chọn
5	coupon_id	FK-COUPON	Mã giảm giá áp dụng (nếu có)
6	discount_amount	Double	Số tiền giảm
7	final_amount	Double	Giá cuối cùng phải trả

8	payment_method	string	Phương thức thanh toán
9	isPaid	Boolean	Trạng thái thanh toán (True/False)
10	paymentDate	Date	Ngày thanh toán
11	status	String	Trạng thái ghi danh (Ví dụ: Chờ mở lớp/đã mở lớp/đã hủy)

Bảng COUPON: Lưu trữ các mã giảm giá đang có tại trung tâm.

Bảng 8: Mô tả bảng COUPON

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã quản lý mã giảm giá
2	code	String	Mã giảm giá cấp cho người dùng
3	description	String	Mô tả mã giảm giá
4	discount_type	String	Loại giảm giá (giảm %/tiền mặt)
5	discount_value	Double	Giá trị giảm
6	min_order_value	Double	Giá khóa học tối thiểu được áp dụng
7	max_discount_amount	Double	Số tiền giảm tối đa (cho loại %)
8	start_date	Date	Ngày bắt đầu
9	expiration_date	Date	Ngày hết hạn
10	usage_limit	Int	Giới hạn số lượt dùng
11	usage_count	Int	Số lượt đã dùng
12	isActive	Bool	Đang khóa/đang mở

Bảng REVIEW: Lưu trữ các đánh giá và bình luận của học viên về khóa học.

Bảng 9: Mô tả bảng REVIEW

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã đánh giá
2	user_id	FK-USER	Người thực hiện đánh giá
3	course_id	FK-COURSE	Khóa học được đánh giá
4	rating	Double	Điểm đánh giá (Ví dụ: 1-5 sao)
5	comment	String	Nội dung bình luận
6	review_date	Date	Ngày gửi đánh giá

Bảng CONTACT: Lưu trữ thông tin liên hệ từ người dùng.

Bảng 10: Mô tả bảng CONTACT

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã liên hệ
2	fullname	String	Tên người liên hệ
3	email	String	Email người liên hệ
4	phone	String	Số điện thoại
5	content	String	Nội dung lời nhắn
6	status	String	Trạng thái xử lý liên hệ

Bảng SLIDESHOW: Lưu trữ banner hiển thị trên trang chủ.

Bảng 11: Mô tả bảng SLIDESHOW

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	_id	ObjectId	Mã banner

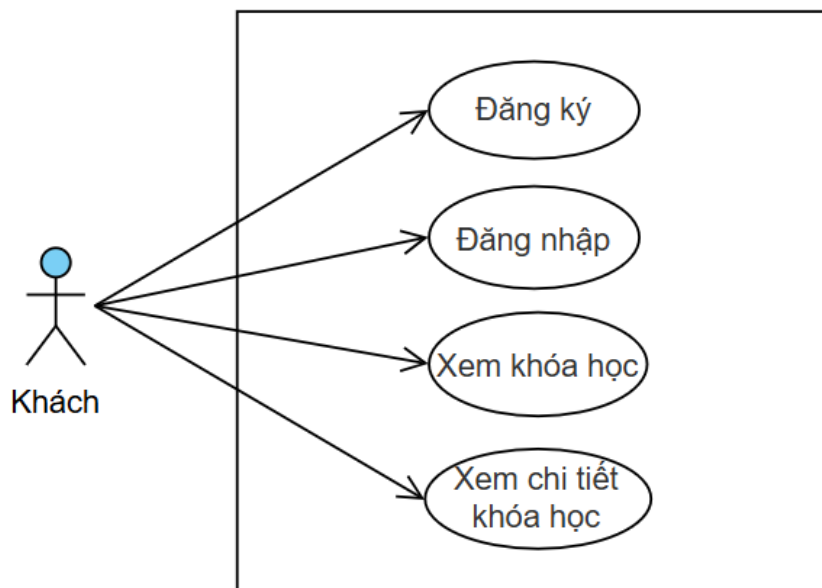
2	image	String	Link ảnh banner
3	title	String	Mô tả
4	order	Int	Thứ tự hiển thị
5	isActive	Bool	Đang khóa/đang mở

3.4 Thiết kế xử lý hệ thống

Hệ thống được xác định có 3 vai trò gồm: Khách truy cập (chưa thực hiện đăng nhập), người dùng đã đăng nhập và người quản trị (Admin). Mỗi người dùng sẽ có các quyền tương ứng.

Bảng 12: Mô tả các actor của dự án

STT	Tên Actor	Ý nghĩa
1	Khách	Người dùng thông thường (chưa đăng nhập)
2	Người dùng	Người dùng đã đăng nhập
3	Admin	Người quản trị toàn bộ hệ thống



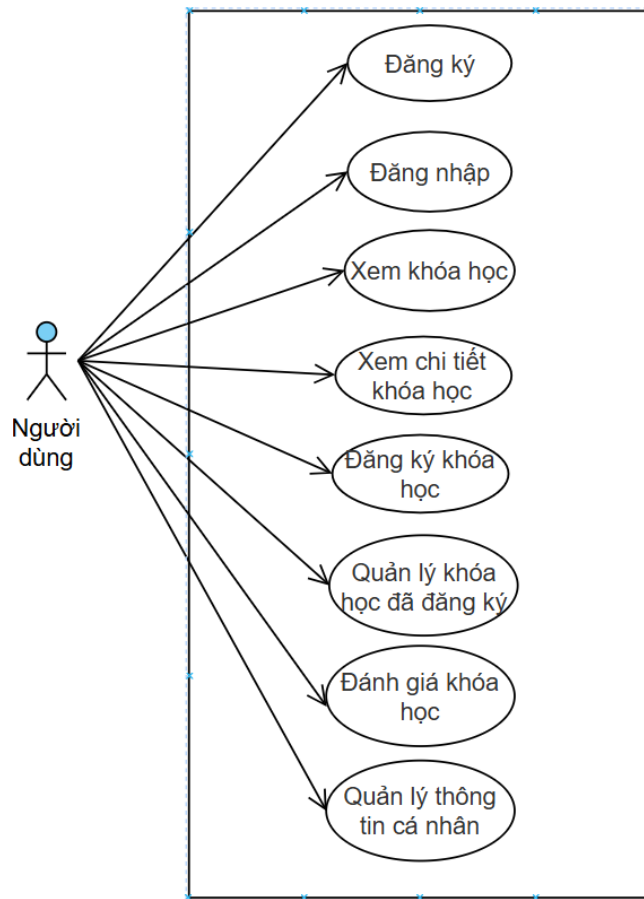
Hình 4: Use-case của actor Khách

Actor Khách có các Use-case được mô tả như sau:

Bảng 13: Mô tả use-case của actor Khách

STT	Tên Use-case	Ý nghĩa
1	Đăng ký	Khách có thể tạo tài khoản mới bằng cách cung cấp thông tin như tên đăng nhập, họ tên, mật khẩu,...
2	Đăng nhập	Khách có thể nhập thông tin tài khoản (tên đăng nhập/mật khẩu) hoặc đăng nhập bằng tài khoản Google để đăng nhập vào hệ thống.
3	Xem khóa học	Khách có thể xem các khóa học đang có ở trung tâm, lọc theo ngôn ngữ, trình độ,...
4	Xem chi tiết khóa học	Khách có thể xem chi tiết một khóa học của ngôn ngữ và trình độ cụ thể, chi tiết khóa học sẽ gồm các thông tin như mã khóa học, tên khóa học, giảng viên, số tiết, ngày bắt đầu,...

Ngoài các chức năng đăng ký, đăng nhập, xem khóa học, xem chi tiết khóa học tương tự của khách, người dùng (đã đăng nhập) sẽ có quyền thực hiện một số chức năng khác như sau:



Hình 5: Use-case của actor Người dùng

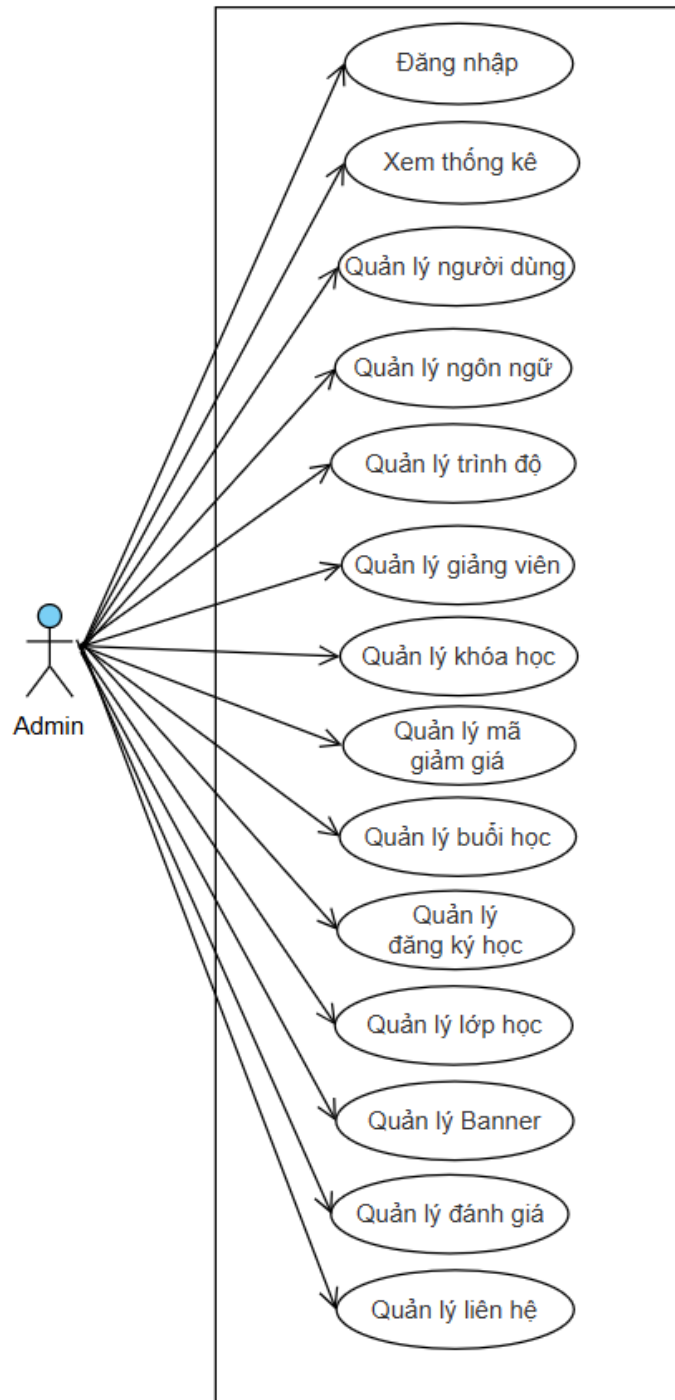
Actor Người dùng có các Use-case được mô tả như sau:

Bảng 14: Mô tả use-case của actor Người dùng

STT	Tên Use-case	Ý nghĩa
1	Đăng ký khóa học	Người dùng có thể vào trang chi tiết của một khóa học bất kỳ và đăng ký nếu khóa học đó đang ở trong thời gian đăng ký cho phép, người dùng sẽ chọn buổi học phù hợp và chọn đăng ký
2	Quản lý khóa học đã đăng ký	Người dùng có thể xem các khóa học đã đăng ký, hủy/thanh toán khóa học trong thời gian cho phép
3	Đánh giá khóa học	Sau khi hoàn thành khóa học, người dùng có thể trở lại trang chi tiết khóa học đã đăng ký để đánh

		giá
4	Quản lý thông tin cá nhân	Người dùng có thể cập nhật thông tin cá nhân như email, địa chỉ,...

Admin (người quản trị) sẽ có quyền cao nhất của hệ thống, có tất cả các chức năng của hai loại người dùng trên. Ngoài ra, admin còn có các quyền quản trị hệ thống sau:



Hình 6: Use-case của actor Admin

Actor Admin có các Use-case được mô tả như sau:

Bảng 15: Mô tả use-case của actor Admin

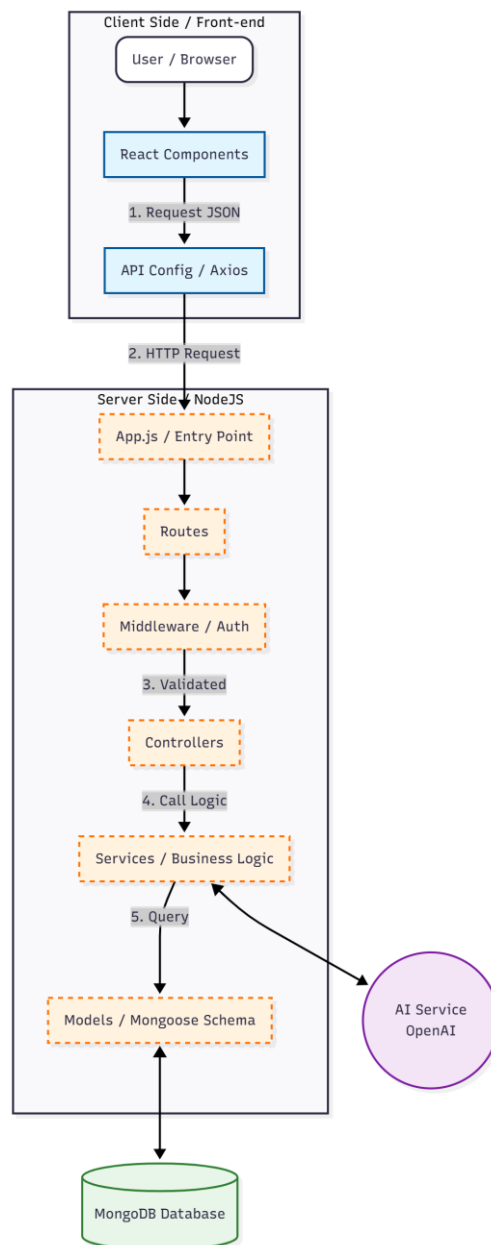
STT	Tên Use-case	Ý nghĩa
1	Xem thống kê	Admin có thể xem thống kê của toàn bộ hệ thống như tổng học viên, tổng khóa học, tổng giảng viên, tổng đăng ký mới, doanh thu,...
2	Quản lý người dùng	Admin có thể quản lý thông tin người dùng, tìm kiếm, lọc, các chức năng thêm/xóa/sửa
3	Quản lý ngôn ngữ	Admin có thể quản lý thông tin ngôn ngữ, tìm kiếm, lọc, các chức năng thêm/xóa/sửa
4	Quản lý trình độ	Admin có thể quản lý thông tin trình độ, các chức năng thêm/xóa/sửa
5	Quản lý giảng viên	Admin có thể quản lý thông tin giảng viên, tìm kiếm, lọc, các chức năng thêm/xóa/sửa
6	Quản lý khóa học	Admin có thể quản lý thông tin khóa học, tìm kiếm, lọc, các chức năng thêm/xóa/sửa
7	Quản lý buổi học	Admin có thể quản lý thông tin buổi học, các chức năng thêm/xóa/sửa
8	Quản lý mã giảm giá	Admin có thể quản lý thông tin, trạng thái mã giảm giá, các chức năng thêm/xóa/sửa
9	Quản lý đăng ký học	Admin có thể quản lý các đăng ký học của học viên, các thao tác thêm/xóa/sửa
10	Quản lý lớp học	Admin có thể quản lý lớp học theo buổi học, có thể ra quyết định mở hay hủy một lớp bất kỳ nếu không đủ số lượng học viên

11	Quản lý Banner	Admin có thể quản lý các Banner hiển thị trên trang chủ, các thao tác thêm/xóa/sửa
12	Quản lý đánh giá	Admin có thể quản lý các đánh giá khóa học của học viên
13	Quản lý liên hệ	Admin có thể quản lý thông tin liên hệ, cập nhật trạng thái hoặc xóa liên hệ

3.5 Mô hình hoạt động của hệ thống

3.5.1 Kiến trúc hệ thống

Hệ thống website đăng ký học ngoại ngữ được xây dựng theo kiến trúc Client-Server với tiêu chuẩn giao tiếp RESTful API trên nền tảng công nghệ MERN Stack (MongoDB, Express, React, NodeJS). Phía Server được tổ chức theo mô hình MVC (Model-View-Controller) kết hợp Services Layer (Tầng dịch vụ) nhằm đảm bảo sự tách biệt rõ ràng giữa logic điều hướng, logic nghiệp vụ và tích hợp linh hoạt các dịch vụ của bên thứ ba như OpenAI.



Hình 7: Kiến trúc hệ thống

Các thành phần chính của hệ thống

Client Side / ReactJS (Phía người dùng)

User / Browser: Người dùng cuối tương tác với hệ thống thông qua trình duyệt web.

React Components: Thành phần giao diện người dùng (UI), nơi hiển thị thông tin và tiếp nhận các thao tác của người dùng.

API Config / Axios: Bộ cấu hình và thư viện dùng để gửi các yêu cầu HTTP đến Server và xử lý phản hồi trả về.

Server Side / NodeJS (Phía máy chủ)

App.js / Entry Point: Điểm khởi chạy của ứng dụng, nơi cấu hình các thiết lập cơ bản và kết nối cơ sở dữ liệu.

Routes: Bộ định tuyến, có nhiệm vụ tiếp nhận các request từ Client và điều hướng chúng đến đúng nơi xử lý.

Middleware / Auth: Lớp trung gian thực hiện các chức năng xác thực người dùng, phân quyền và kiểm tra dữ liệu đầu vào trước khi cho phép đi tiếp.

Controllers: Bộ điều khiển, chịu trách nhiệm nhận dữ liệu đã được xác thực, điều phối luồng xử lý và trả kết quả cuối cùng cho Client.

Services / Business Logic: Lớp dịch vụ chứa toàn bộ logic nghiệp vụ cốt lõi của hệ thống. Đây là nơi xử lý các thuật toán phức tạp, giao tiếp với Database hoặc gọi API của bên thứ 3 (AI).

Models / Mongoose Schema: Lớp định nghĩa cấu trúc dữ liệu, đóng vai trò ánh xạ dữ liệu giữa ứng dụng và MongoDB.

Database và External Services (Cơ sở dữ liệu và dịch vụ ngoài)

MongoDB Database: Cơ sở dữ liệu NoSQL dùng để lưu trữ toàn bộ dữ liệu của hệ thống (users, courses, reviews...).

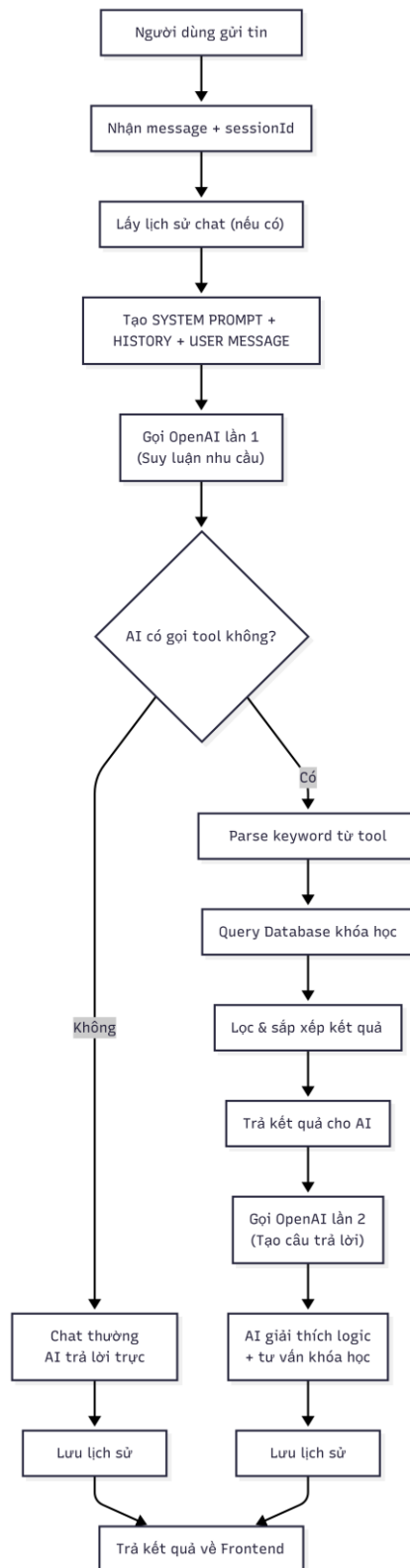
AI Service (OpenAI): Dịch vụ trí tuệ nhân tạo bên ngoài, được tích hợp để xử lý các tác vụ thông minh như Chatbot tư vấn, gợi ý khóa học.

Luồng xử lý của hệ thống

Khi người dùng thực hiện thao tác trên trình duyệt (User / Browser), giao diện (React Components) sẽ đóng gói dữ liệu thành Request JSON (1) và gửi đi thông qua cấu hình API Config / Axios. Yêu cầu HTTP Request (2) này được Server tiếp nhận tại App.js và chuyển đến Routes để định tuyến. Trước khi xử lý, yêu cầu phải đi qua lớp Middleware để xác thực (3). Nếu hợp lệ, yêu cầu được chuyển đến Controllers để điều phối. Tại đây, Controller thực hiện Call Logic (4) xuống tầng Services để xử lý nghiệp vụ. Tùy thuộc vào loại yêu cầu, Service sẽ thực hiện Query (5) thông qua Models để tương tác với MongoDB Database hoặc kết nối

tới AI Service (OpenAI). Cuối cùng, kết quả dữ liệu sẽ được trả ngược lại theo quy trình để hiển thị lên giao diện người dùng.

Luồng xử lý của Chatbot



Hình 8: Luồng xử lý của Chatbot

Mô tả luồng xử lý của Chatbot tư vấn khóa học

Trước hết, người dùng gửi tin nhắn thông qua giao diện frontend của hệ thống. Tin nhắn này cùng với sessionId được chuyển về backend để xử lý.

Tại backend, hệ thống tiếp nhận nội dung tin nhắn và sessionId, sau đó kiểm tra và lấy lịch sử hội thoại trước đó tương ứng với session (nếu tồn tại) nhằm đảm bảo tính liên tục của cuộc hội thoại.

Tiếp theo, hệ thống tạo nội dung hội thoại hoàn chỉnh, bao gồm:

System Prompt (định nghĩa vai trò và quy tắc cho AI).

Lịch sử chat đã lưu.

Tin nhắn hiện tại của người dùng.

Nội dung này được gửi đến OpenAI (lần gọi thứ nhất) với mục đích để AI phân tích và suy luận nhu cầu của người dùng, đồng thời quyết định có cần gọi công cụ tìm kiếm khóa học (tool) hay không.

Sau khi nhận phản hồi từ AI, hệ thống kiểm tra:

Trường hợp AI không gọi tool:

Điều này cho thấy người dùng chỉ đang trò chuyện thông thường. Khi đó, hệ thống sử dụng trực tiếp câu trả lời của AI, lưu lại lịch sử hội thoại và trả kết quả về frontend để hiển thị cho người dùng.

Trường hợp AI có gọi tool:

Hệ thống tiến hành phân tích từ khóa tìm kiếm (keyword) do AI suy luận. Dựa trên keyword này, backend truy vấn cơ sở dữ liệu để lấy danh sách các khóa học phù hợp.

Kết quả truy vấn tiếp tục được lọc, sắp xếp và giới hạn số lượng nhằm ưu tiên các khóa học phù hợp nhất (ví dụ: lớp sắp khai giảng).

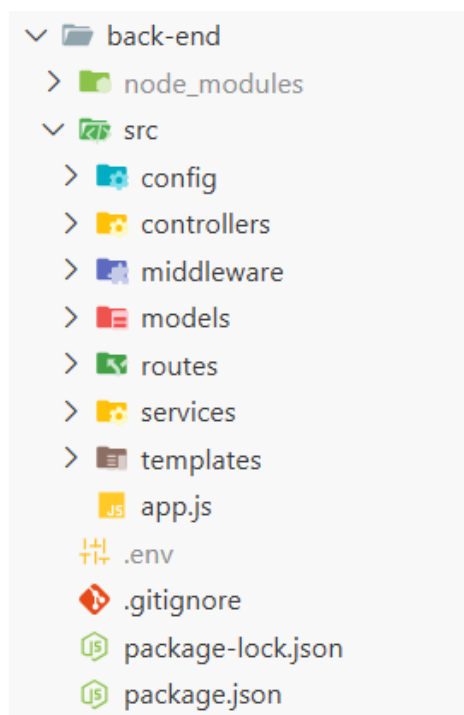
Sau đó, danh sách khóa học được trả ngược lại cho AI. Hệ thống tiếp tục gọi OpenAI lần thứ hai để AI sử dụng dữ liệu thực tế từ cơ sở dữ liệu, từ đó giải thích logic tư vấn và giới thiệu khóa học cho người dùng một cách tự nhiên.

Cuối cùng, hệ thống lưu lại lịch sử hội thoại của phiên làm việc và trả kết quả tư vấn về frontend để hiển thị cho người dùng.

3.5.2 Cấu trúc thư mục

Xây dựng Backend

Backend được xây dựng theo kiến trúc RESTful API trên nền tảng NodeJS và ExpressJS, kết nối với cơ sở dữ liệu MongoDB. Để đảm bảo tính tổ chức, dễ bảo trì và mở rộng, mã nguồn được triển khai theo mô hình phân lớp (Layered Architecture) với sự phân chia rõ ràng giữa Routes, Controllers, Services và Models.



Hình 9: Cấu trúc thư mục back-end

Mã nguồn backend được đặt trong thư mục src để quản lý tập trung, với các thành phần chi tiết như sau:

src/config/: Chứa các tệp cấu hình môi trường và kết nối dịch vụ như cấu hình thông tin định danh để upload ảnh lên Cloudinary, cấu hình JWT, dịch vụ gửi email,...

src/controllers/: Lớp điều phối, tiếp nhận các yêu cầu từ người dùng thông qua Router, gọi xuống lớp services để xử lý logic và trả về phản hồi cho Client.

src/services/: Lớp dịch vụ, chứa toàn bộ logic nghiệp vụ phức tạp của dự án (logic tính toán, xử lý dữ liệu thô từ OpenAI, xử lý transaction), tách biệt logic khỏi Controller giúp code sạch và dễ tái sử dụng.

src/models/: Lớp dữ liệu, định nghĩa các Schema Mongoose (User, Course, Registration,...), tương tác trực tiếp với cơ sở dữ liệu MongoDB.

src/routes/: Định nghĩa các điểm cuối API, nạp các middleware và điều hướng các request HTTP (GET, POST, PUT, PATCH, DELETE) tới các Controller tương ứng.

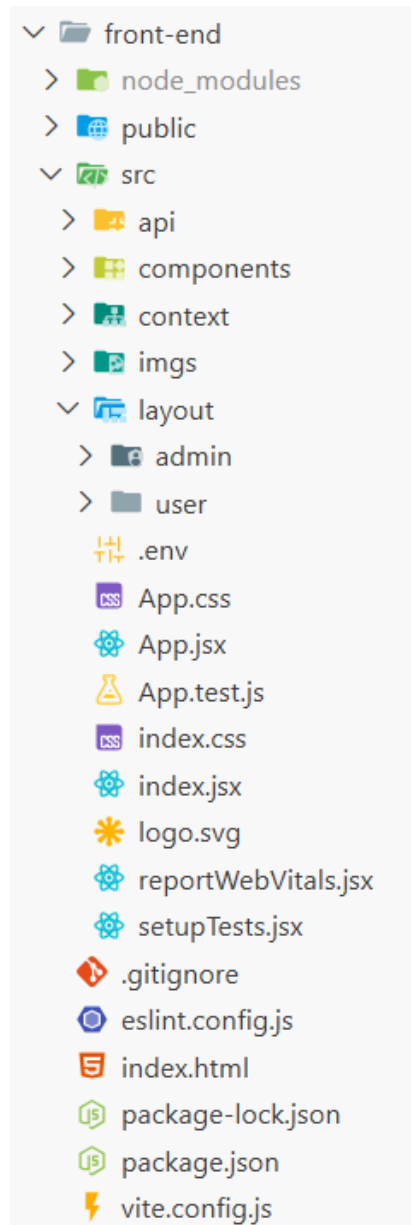
src/middleware/: Các hàm trung gian xử lý request, thực hiện xác thực, kiểm tra quyền trước khi request đến được Controller.

src/templates/: Lưu trữ các file HTML mẫu cho email (email đổi mật khẩu, hóa đơn,...).

src/app.js: Tập khởi tạo chính của ứng dụng, Thiết lập Express và kết nối các routes.

Xây dựng Frontend

Frontend được xây dựng theo kiến trúc Single Page Application (SPA) trên nền tảng ReactJS, sử dụng công cụ build Vite để tối ưu hóa tốc độ và hiệu năng. Giao diện người dùng được thiết kế theo hướng Component-Based (hướng thành phần) kết hợp với Tailwind CSS, giúp tối ưu hóa việc tái sử dụng mã nguồn, đảm bảo tính nhất quán và khả năng mở rộng dễ dàng.



Hình 10: Cấu trúc thư mục front-end

Mã nguồn frontend được tổ chức trong thư mục src, phân chia rõ ràng giữa các thành phần giao diện, logic gọi API và quản lý trạng thái như sau:

src/api/: Cấu hình HTTP Client trung tâm, tự động điều chỉnh baseURL theo môi trường phát triển và thiết lập bảo mật phiên làm việc.

src/components/: Chứa các thành phần giao diện tái sử dụng như header, footer, thẻ khóa học,... Các component này hoạt động độc lập và được lắp ghép để tạo nên các trang hoàn chỉnh.

src/context/: Quản lý trạng thái toàn cục, lưu trữ các dữ liệu dùng chung cho toàn ứng dụng như thông tin đăng nhập,...

`src/layout/`: Định nghĩa khung sườn cho trang web. Được chia tách cụ thể thành admin (giao diện quản trị viên) và user (giao diện người dùng) để quản lý riêng biệt cho từng đối tượng.

`src/imgs/`: Lưu trữ các tài nguyên hình ảnh tĩnh phục vụ cho giao diện.

`src/App.jsx`: Component gốc của ứng dụng, điều hướng người dùng giữa các trang.

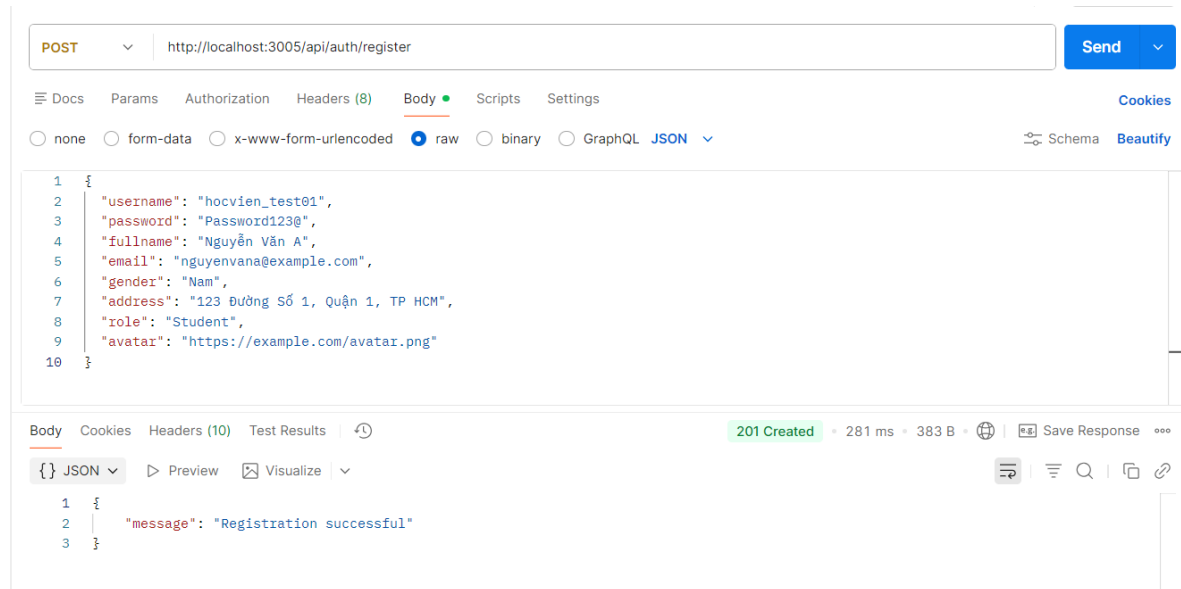
`vite.config.js`: Tập cấu hình các plugin cần thiết cho React và Tailwind CSS.

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Kiểm thử API với Postman

Các API liên quan đến xác thực

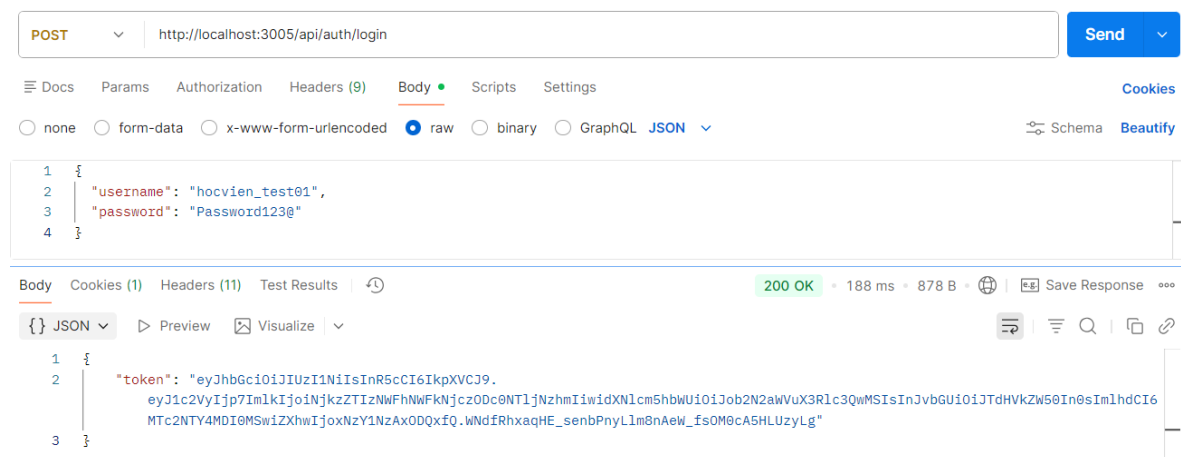
API dùng cho việc đăng ký tài khoản được xây dựng như sau:



Hình 11: Kiểm thử API đăng ký tài khoản

API thực hiện chức năng đăng ký tài khoản mới thông qua phương thức POST tại endpoint `http://localhost:3005/api/auth/register`, trong đó client gửi thông tin người dùng chi tiết (gồm username, password, fullname, email, gender, address, role và avatar) dưới dạng JSON; nếu tạo thành công, máy chủ sẽ phản hồi mã 201 Created kèm thông báo "Registration successful".

API dùng cho việc đăng nhập được xây dựng như sau:

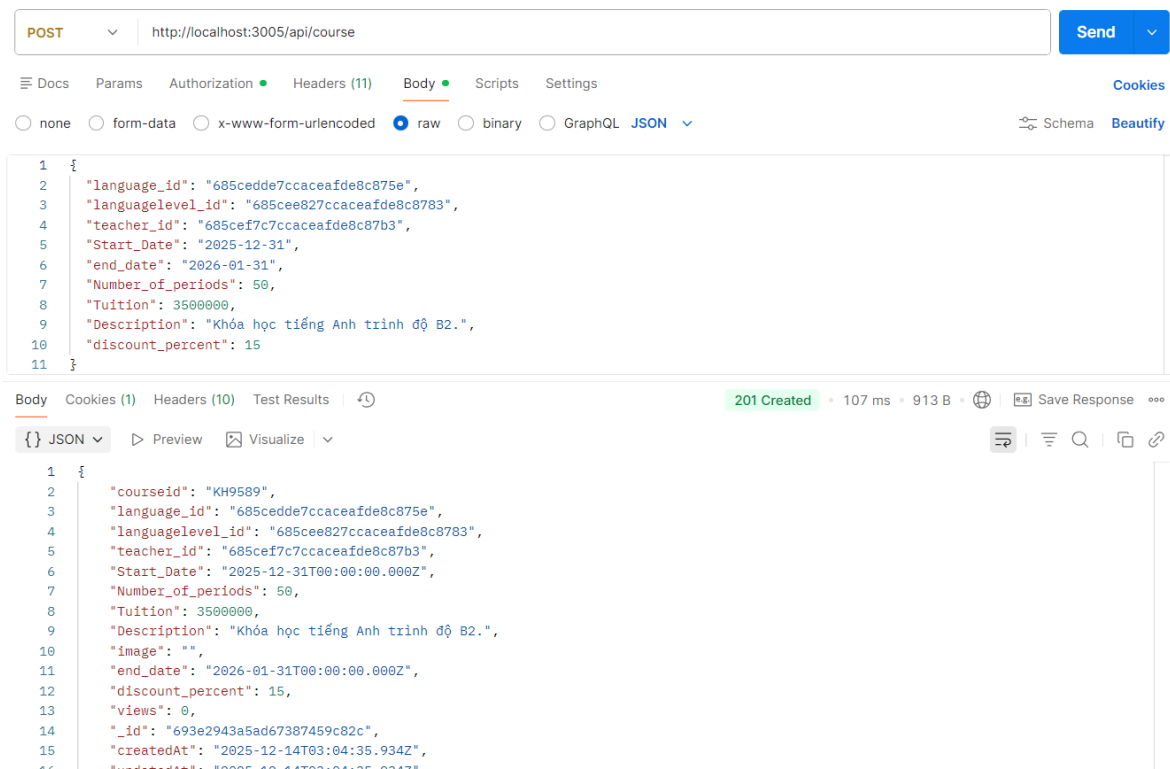


Hình 12: Kiểm thử API đăng nhập

API thực hiện chức năng đăng nhập thông qua phương thức POST tại endpoint `http://localhost:3005/api/auth/login`, trong đó client gửi thông tin xác thực gồm username và password dưới dạng JSON; nếu thông tin chính xác, máy chủ sẽ phản hồi mã 200 OK kèm theo một chuỗi token để cấp quyền truy cập cho các phiên làm việc tiếp theo.

Các API liên quan đến khóa học

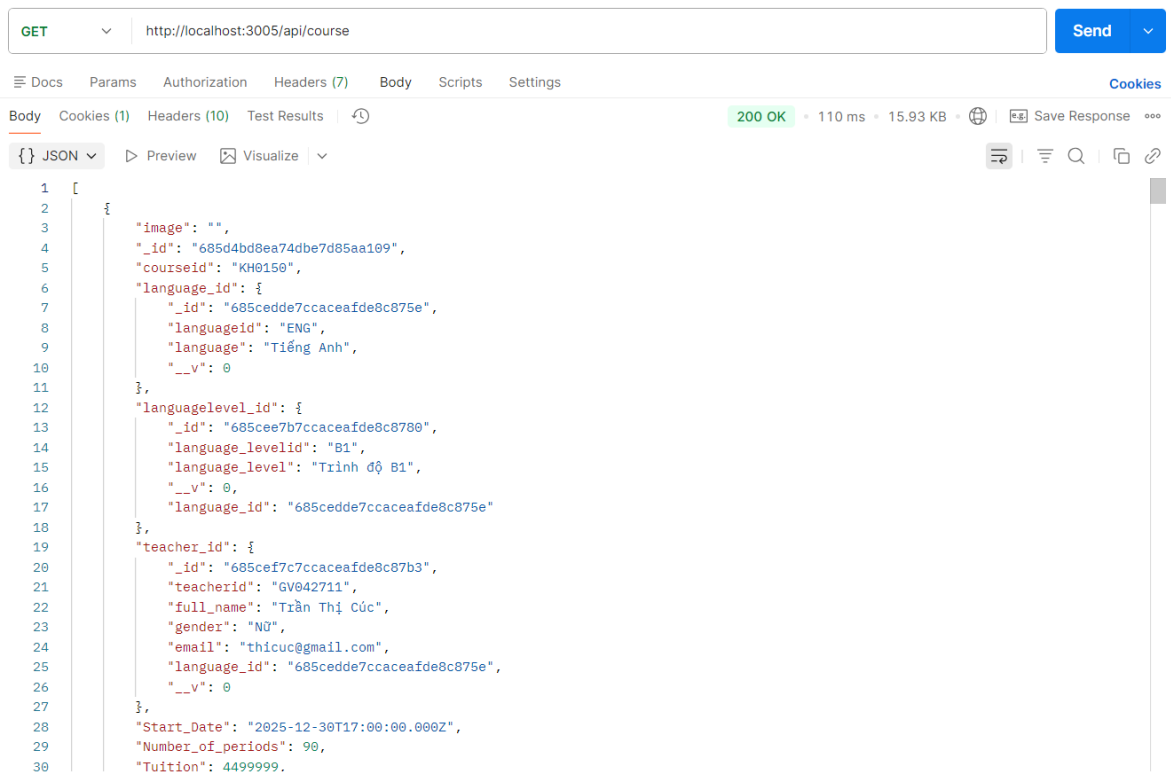
API tạo khóa học mới được xây dựng như sau:



Hình 13: Kiểm thử API tạo khóa học mới (admin)

API dành cho Quản trị viên (Admin) để tạo khóa học mới thông qua phương thức POST tại endpoint `http://localhost:3005/api/course`. Client gửi dữ liệu JSON chứa thông tin chi tiết khóa học như: các ID liên kết (ngôn ngữ, trình độ, giảng viên), thời gian (ngày bắt đầu/kết thúc), số tiết học, học phí và mô tả. Nếu thành công, hệ thống trả về mã 201 Created cùng đối tượng khóa học vừa tạo (bao gồm `courseid` được sinh tự động).

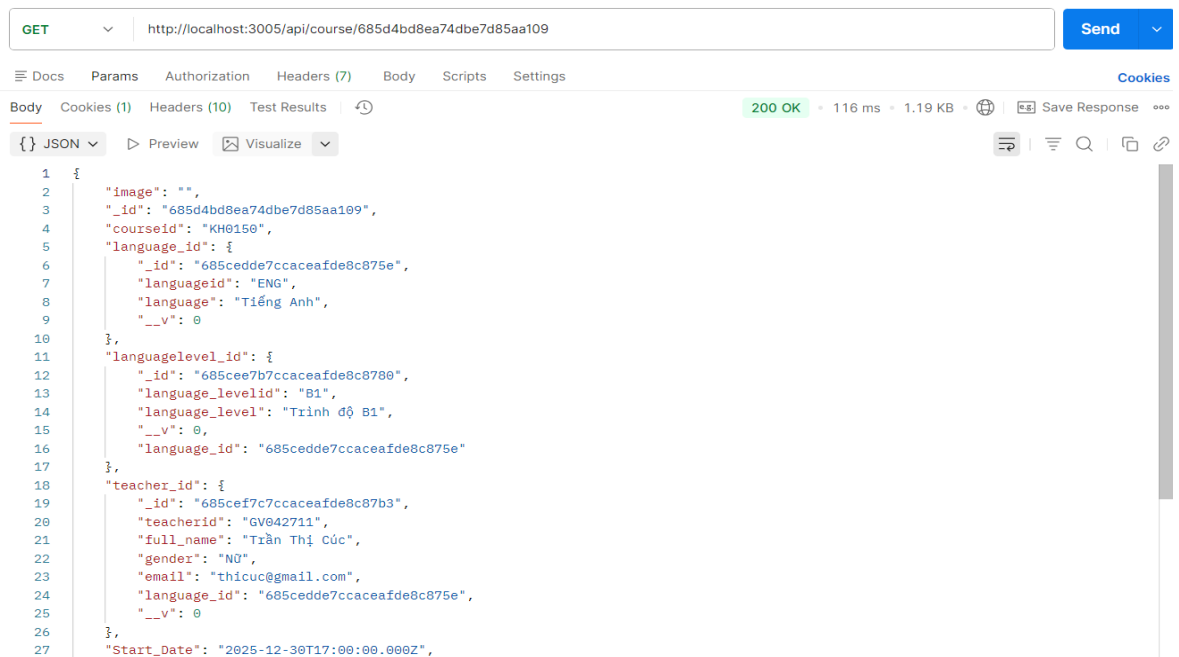
API lấy tất cả khóa học được xây dựng như sau:



Hình 14: Kiểm thử API lấy tất cả khóa học

API thực hiện chức năng lấy danh sách tất cả khóa học thông qua phương thức GET tại endpoint `http://localhost:3005/api/course`. Máy chủ trả về mã 200 OK cùng dữ liệu JSON là một mảng các khóa học; các trường liên kết ngoại như `language_id`, `languagelevel_id` và `teacher_id` đã được tham chiếu (lấy kèm dữ liệu chi tiết từ các bảng liên quan) để hiển thị đầy đủ tên ngôn ngữ, trình độ và thông tin giảng viên ngay trong response.

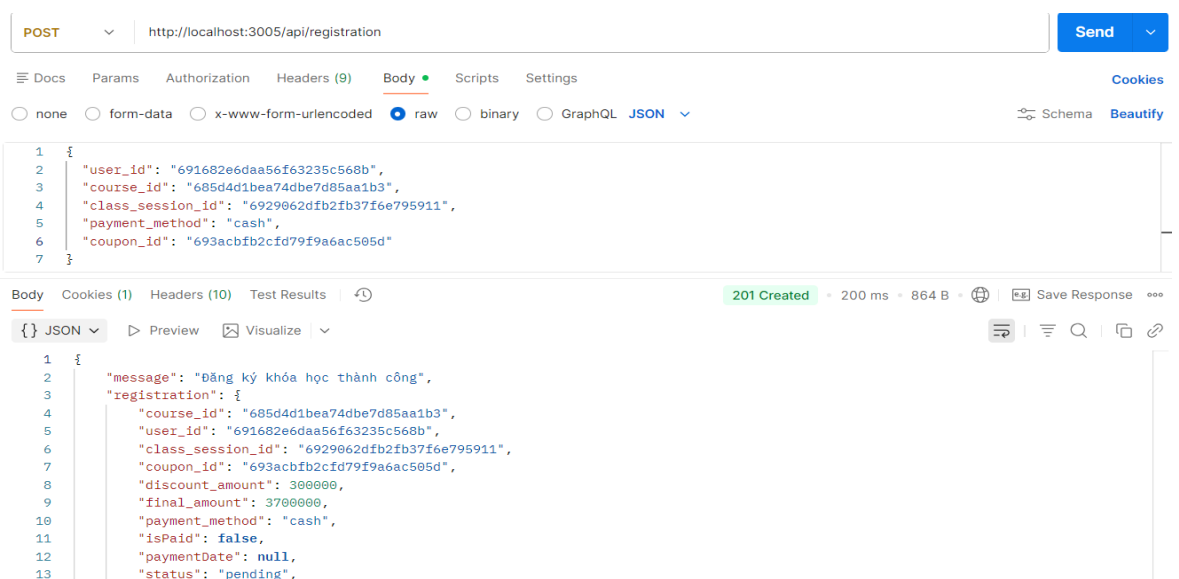
API lấy chi tiết một khóa học được xây dựng như sau:



Hình 15: Kiểm thử API lấy chi tiết 1 khóa học

API dùng để lấy thông tin chi tiết của một khóa học cụ thể thông qua phương thức GET tại endpoint `http://localhost:3005/api/course/:id` (với `:id` là `_id` của khóa học cần xem). Máy chủ trả về mã 200 OK cùng một đối tượng JSON chứa đầy đủ dữ liệu của khóa học đó; tương tự như API lấy danh sách, các trường liên kết (`language_id`, `languagelevel_id`, `teacher_id`) cũng được tham chiếu để hiển thị chi tiết tên ngôn ngữ, trình độ và thông tin giảng viên.

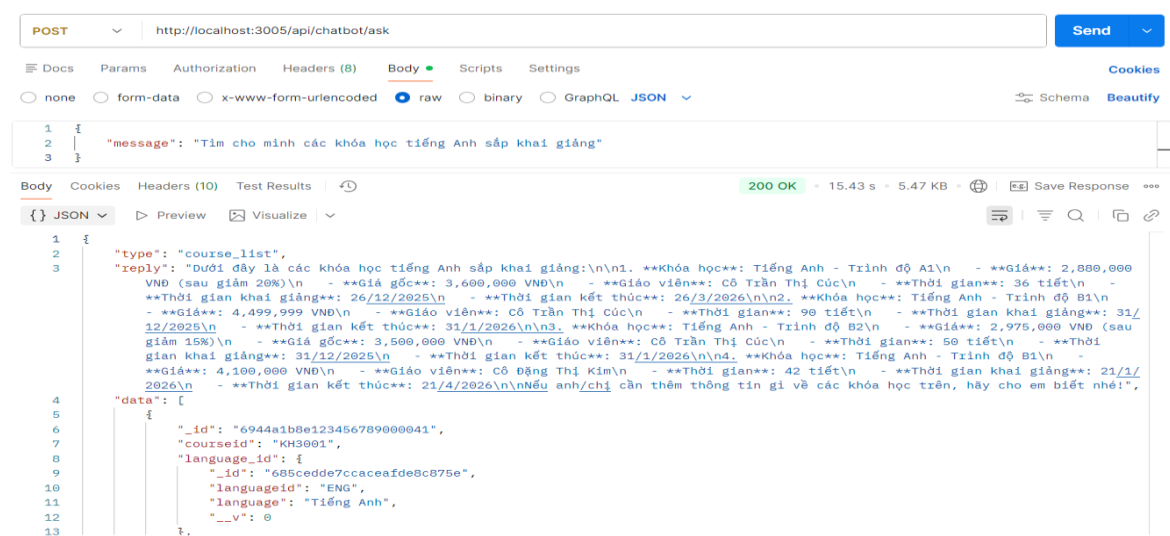
API đăng ký khóa học được xây dựng như sau:



Hình 16: Kiểm thử API đăng ký khóa học

API thực hiện chức năng đăng ký ghi danh vào khóa học thông qua phương thức POST tại endpoint `http://localhost:3005/api/registration`. Client gửi dữ liệu gồm các thông tin định danh (`user_id`, `course_id`, `class_session_id`), mã giảm giá (`coupon_id`) và phương thức thanh toán; hệ thống sẽ xử lý, tính toán số tiền giảm giá và tổng tiền phải trả, sau đó phản hồi mã 201 Created kèm đối tượng đăng ký mới.

API chatbot tư vấn khóa học được xây dựng như sau:



Hình 17: Kiểm thử API chatbot tư vấn khóa học

API tư vấn khóa học thông minh (Chatbot) thông qua phương thức POST tại endpoint `http://localhost:3005/api/chatbot/ask`. Client gửi câu hỏi dưới dạng ngôn ngữ tự nhiên (`message`); hệ thống sẽ xử lý (tích hợp LLM OpenAI) và phản hồi mã 200 OK kèm theo cấu trúc JSON gồm: `reply` (câu trả lời văn bản có định dạng Markdown), `type` (loại phản hồi `"course_list"`) và mảng `data` chứa thông tin chi tiết các khóa học thực tế tìm được từ database để client có thể hiển thị dưới dạng danh sách.

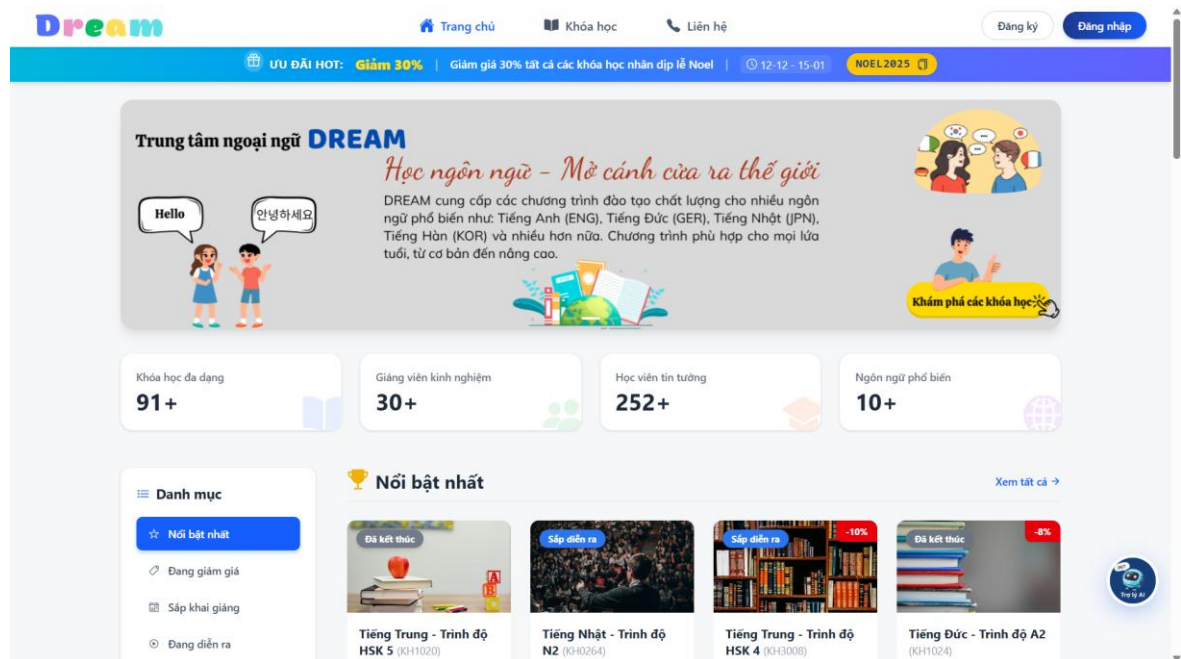
4.2 Giao diện người dùng và chức năng

Hệ thống website được xây dựng với các giao diện và chức năng như sau:

Trang chủ khi chưa đăng nhập cho phép người dùng xem banner, các số liệu thống kê tổng quan về trung tâm như tổng khóa học, giảng viên,... Các khóa học hiển thị bên dưới theo các tiêu chí như nổi bật nhất, đang giảm giá, sắp khai giảng và đang diễn ra. Thanh điều hướng bên trên sẽ bao gồm trang chủ, khóa học và liên

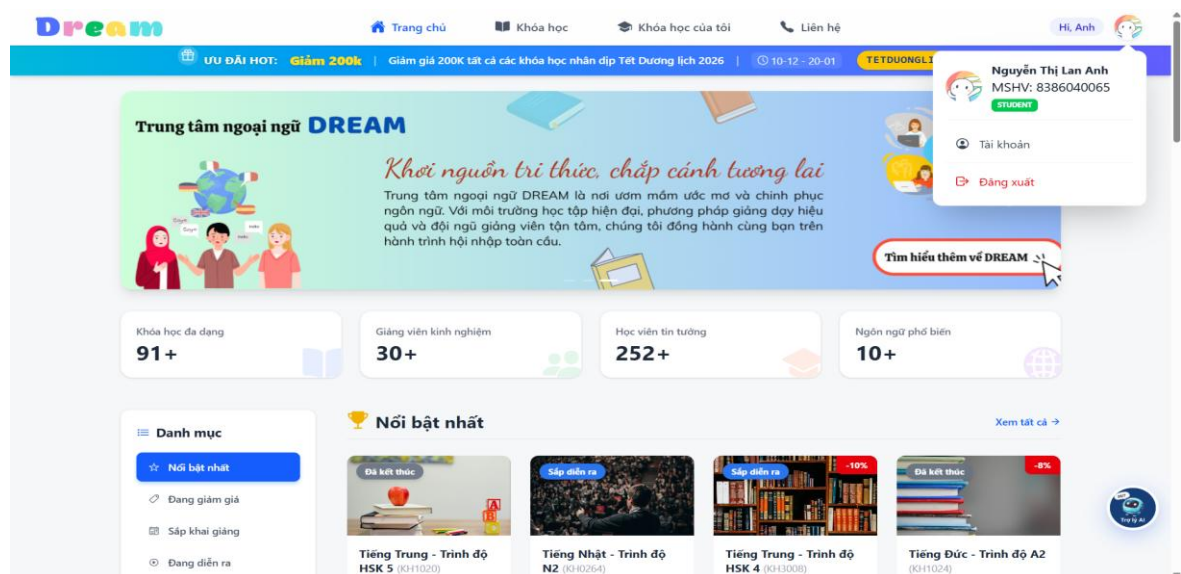
Xây dựng Website đăng ký học ngoại ngữ tích hợp trợ lý ảo AI

hệ, bên dưới thanh điều hướng sẽ chạy quảng cáo các mã giảm giá đang hoạt động. Người dùng có thể đăng ký tài khoản học viên, sau khi đăng ký có thể đăng nhập bằng tài khoản đã đăng ký. Ngoài ra, người dùng có thể tương tác với chatbot để nhận tư vấn về khóa học thông qua minichat góc phải bên dưới màn hình.



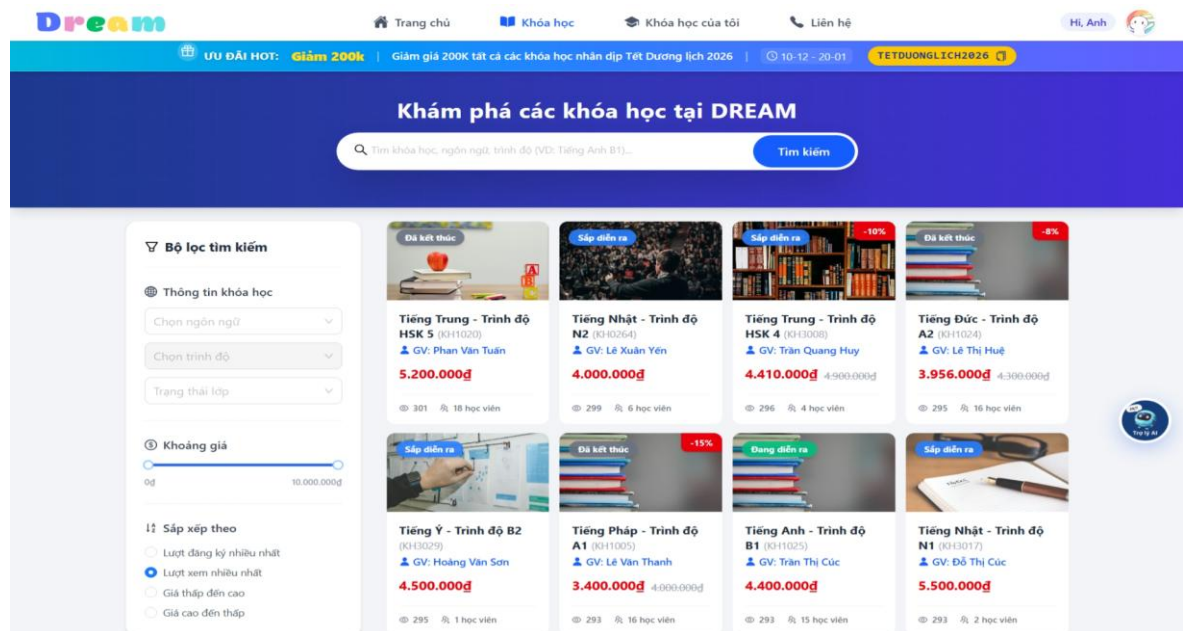
Hình 18: Giao diện trang chủ (Chưa đăng nhập)

Khi đăng nhập thành công, thanh điều hướng phía trên sẽ có thêm một tab nữa để quản lý khóa học đã đăng ký. Ngoài ra, người dùng sẽ được cấp một mã số học viên được dùng trong suốt quá trình học tập, chức năng quản lý tài khoản và đăng xuất.



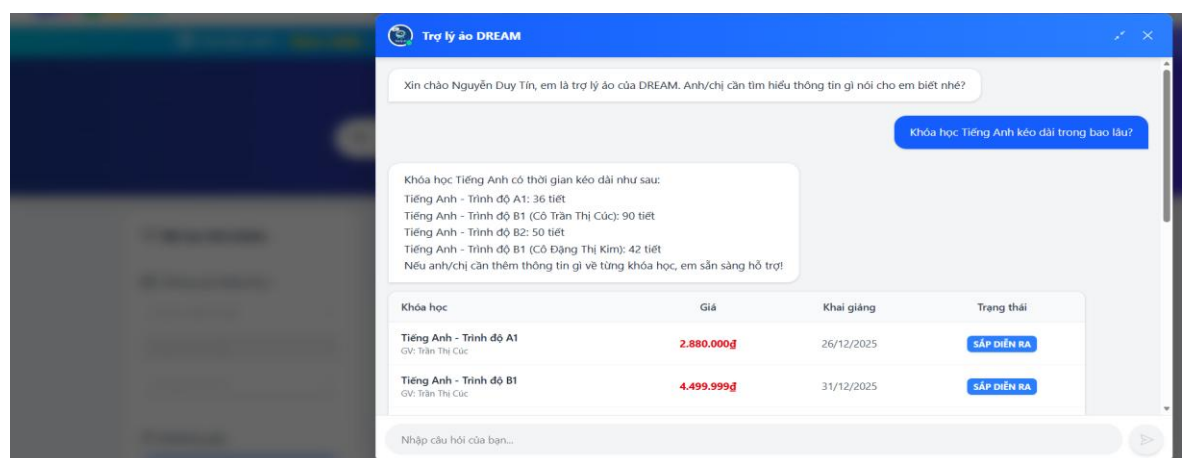
Hình 19: Giao diện trang chủ (Đã đăng nhập)

Trang khóa học cho phép người dùng tìm kiếm khóa học bất kỳ bằng thanh tìm kiếm, người dùng cũng có thể tìm kiếm bằng cách lọc theo ngôn ngữ, trình độ, trạng thái lớp (sắp diễn ra/đang diễn ra/đã kết thúc), chức năng sắp xếp theo lượt xem, lượt đăng ký, giá. Khi đã tìm được khóa học phù hợp, người dùng có thể chọn khóa học đó để xem chi tiết.



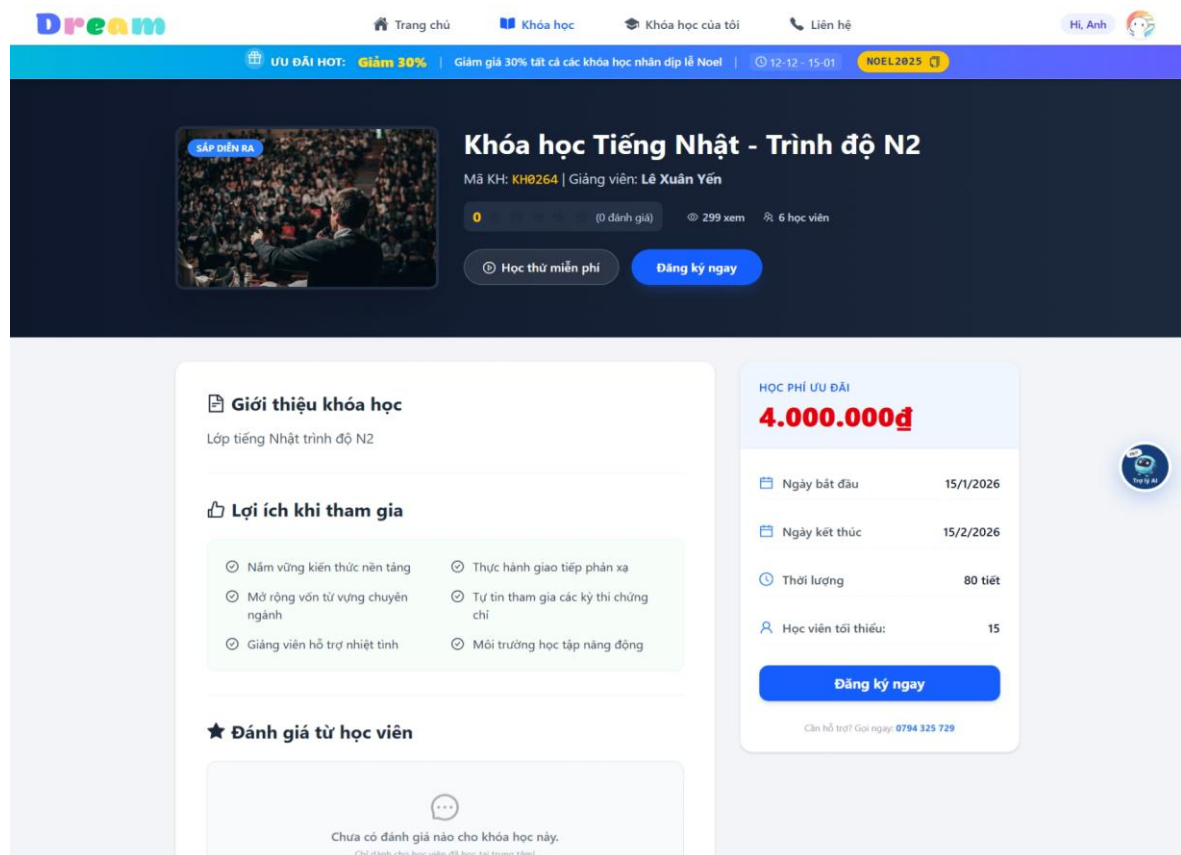
Hình 20: Giao diện trang khóa học

Ngoài các chức năng lọc và tìm kiếm thông thường, người dùng có thể nhờ sự tư vấn từ chatbot AI. Người dùng có thể hỏi bất cứ câu hỏi gì, nếu có liên quan đến khóa học, chatbot sẽ tự động tìm kiếm khóa học phù hợp và trả về kết quả là link tới khóa học.



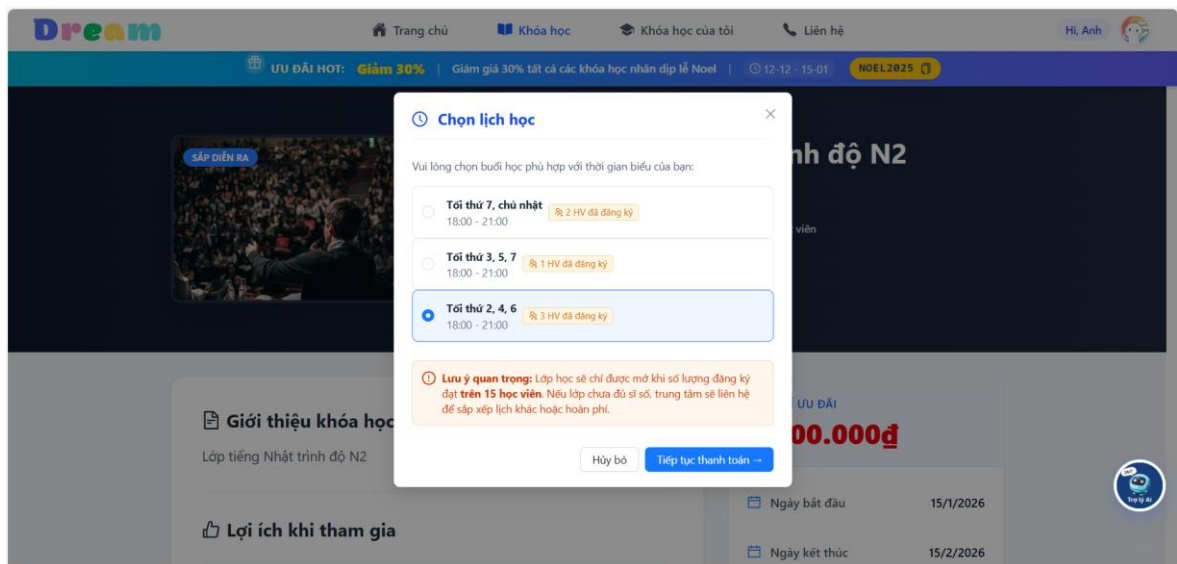
Hình 21: Giao diện chatbot

Trang chi tiết sẽ hiển thị chi tiết của khóa học như tên ngôn ngữ, trình độ, mã khóa học, số lượt đăng ký, lượt xem,... Người dùng có thể học thử online một video ngắn trên youtube bằng nút “Học thử miễn phí”. Nếu người dùng đã đăng nhập và trạng thái khóa học là sắp diễn ra, người dùng có thể đăng ký khóa học đó. Nếu khóa học là đã kết thúc, người dùng sẽ xem được đánh giá của những học viên đã tham gia khóa học (nếu có).



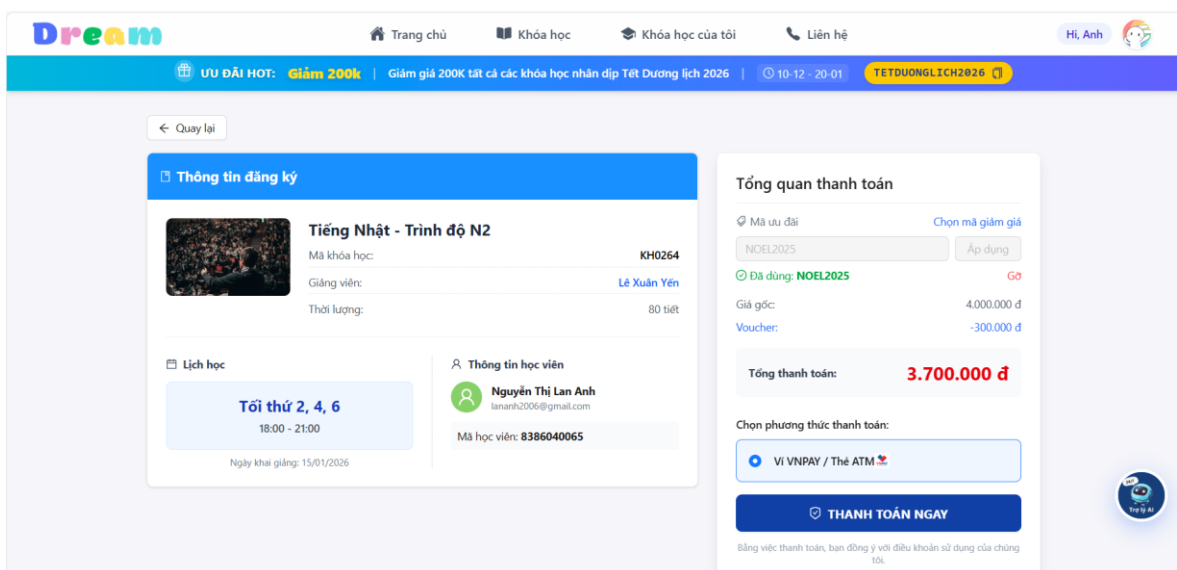
Hình 22: Giao diện trang chi tiết khóa học

Sau khi chọn đăng ký ngay, người dùng sẽ phải chọn tiếp buổi học phù hợp sau đó chọn tiếp tục thanh toán.



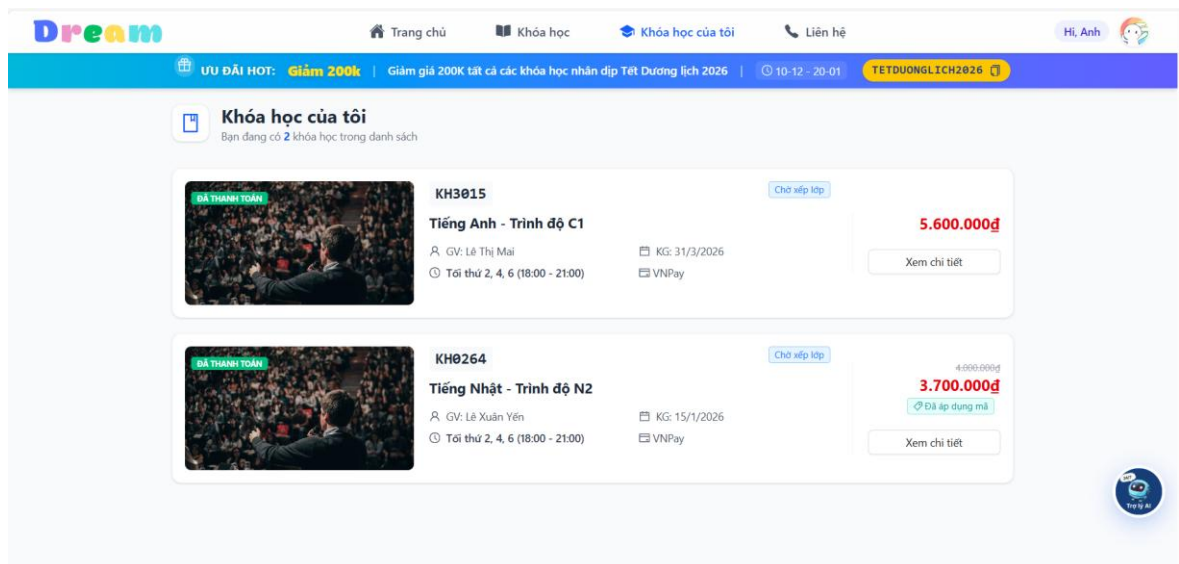
Hình 23: Giao diện trang chọn buổi học

Người dùng có thể xem lại khóa học và buổi học đã chọn, áp dụng mã giảm giá (nếu có). Mặc định chỉ có phương thức thanh toán là VNPay. Sau khi xem lại, người dùng có thể chọn thanh toán ngay để tiếp tục thanh toán hoặc quay lại trang trước.



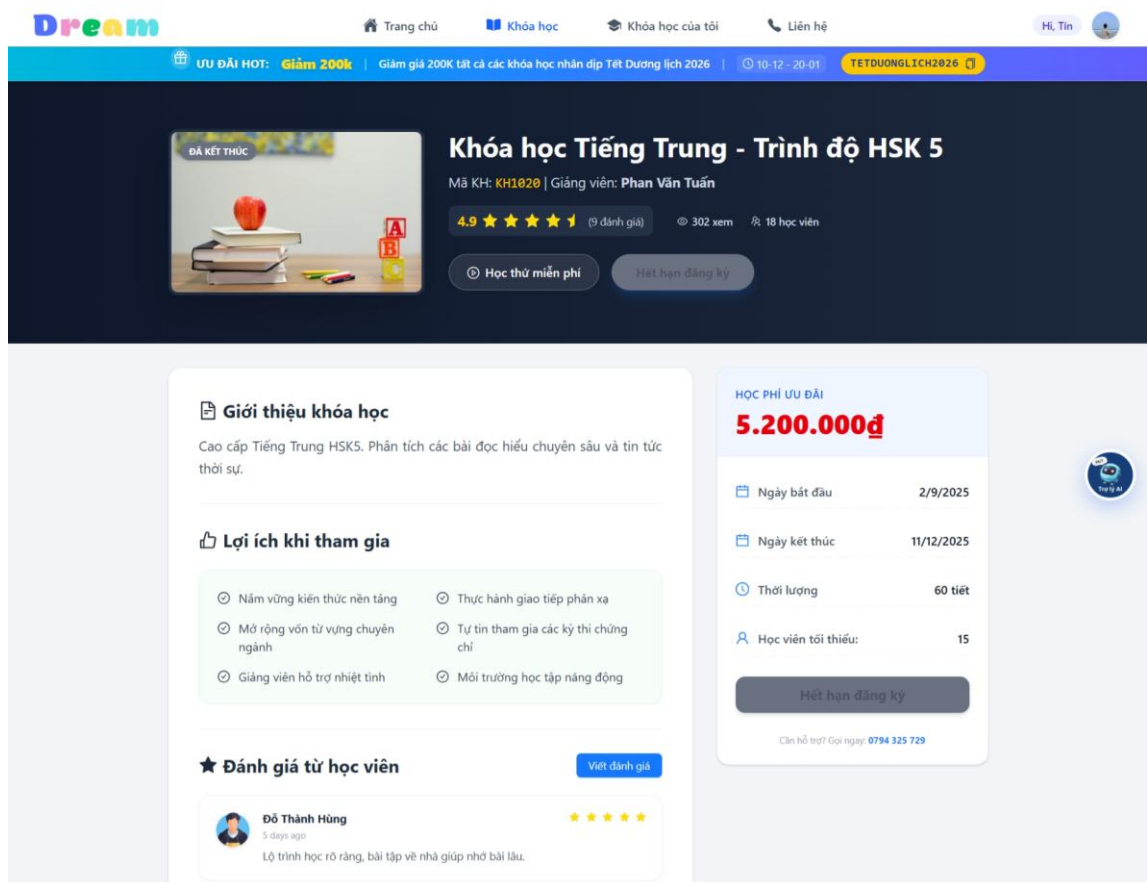
Hình 24: Giao diện xem lại thanh toán và áp dụng mã giảm giá

Nếu thanh toán thành công, người dùng sẽ được chuyển sang trang khóa học của tôi. Trang này sẽ hiển thị tất cả các khóa học mà người dùng đã đăng ký. Tại đây, người dùng sẽ theo dõi trạng thái khóa học là chờ xếp lớp/đã mở lớp/đã hủy do không đủ số lượng mở lớp.



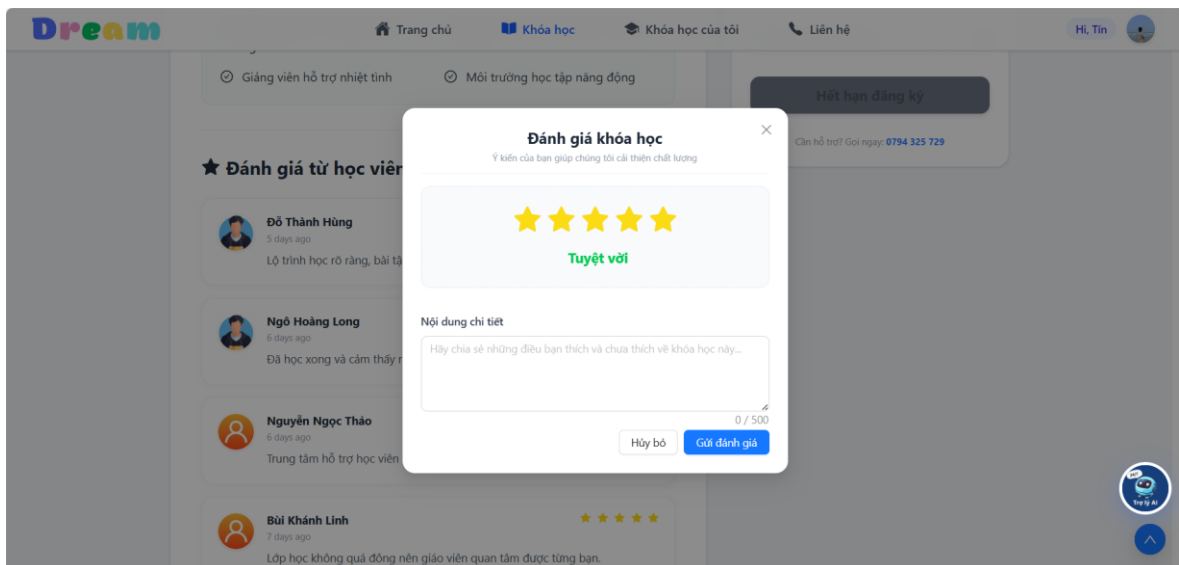
Hình 25: Giao diện trang khóa học của tôi

Nếu khóa học mà người dùng tham gia đã kết thúc, người dùng sẽ có thể quay lại trang chi tiết khóa học để đánh giá. Lúc này trang chi tiết sẽ hiển thị số sao, nội dung đánh giá của các học viên khác và nút đánh giá.



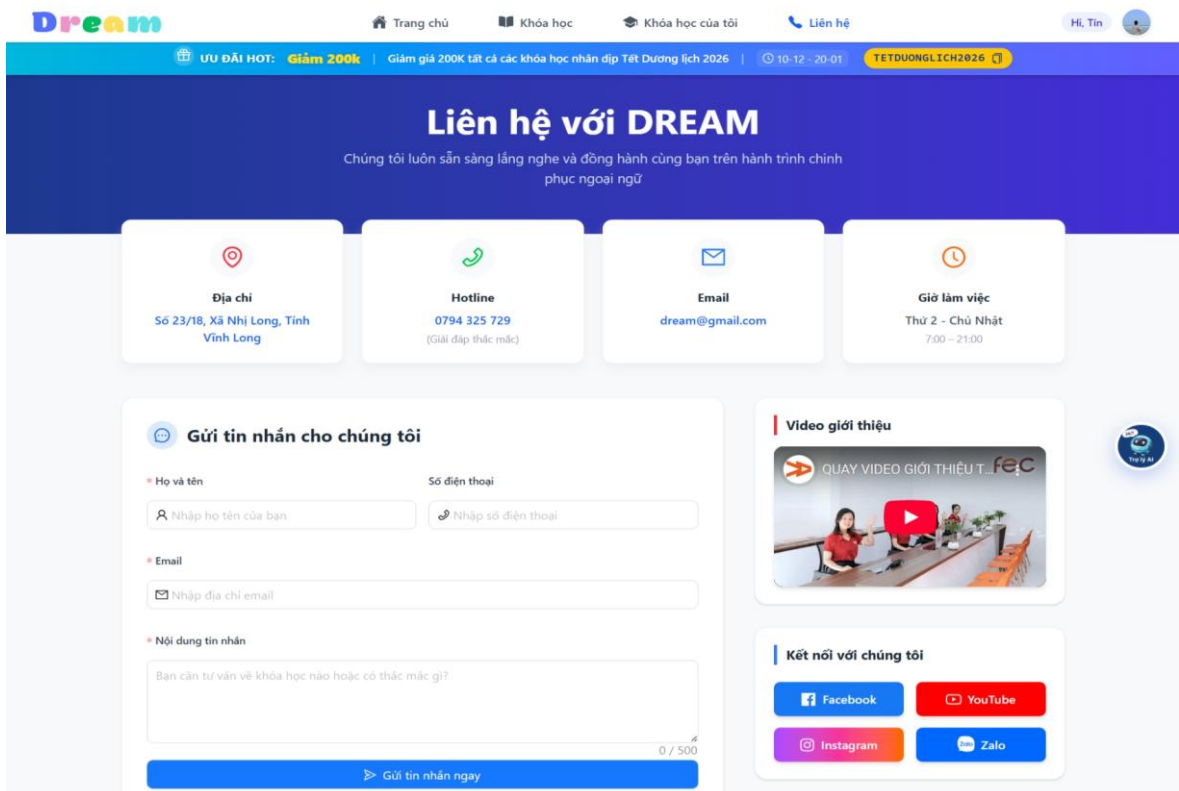
Hình 26: Giao diện trang chi tiết khóa học đã kết thúc

Khi chọn nút viết đánh giá, sẽ hiện lên một form cho phép người dùng chọn số sao và viết nội dung đánh giá. Người dùng chỉ được đánh giá một lần duy nhất, nút viết đánh giá sẽ bị khóa khi người dùng chọn gửi đánh giá.



Hình 27: Giao diện trang đánh giá khóa học

Nếu có bất kỳ thắc mắc gì, người dùng có thể liên hệ với trung tâm thông qua biểu mẫu ở trang liên hệ hoặc qua các hình thức khác như email, số điện thoại,...

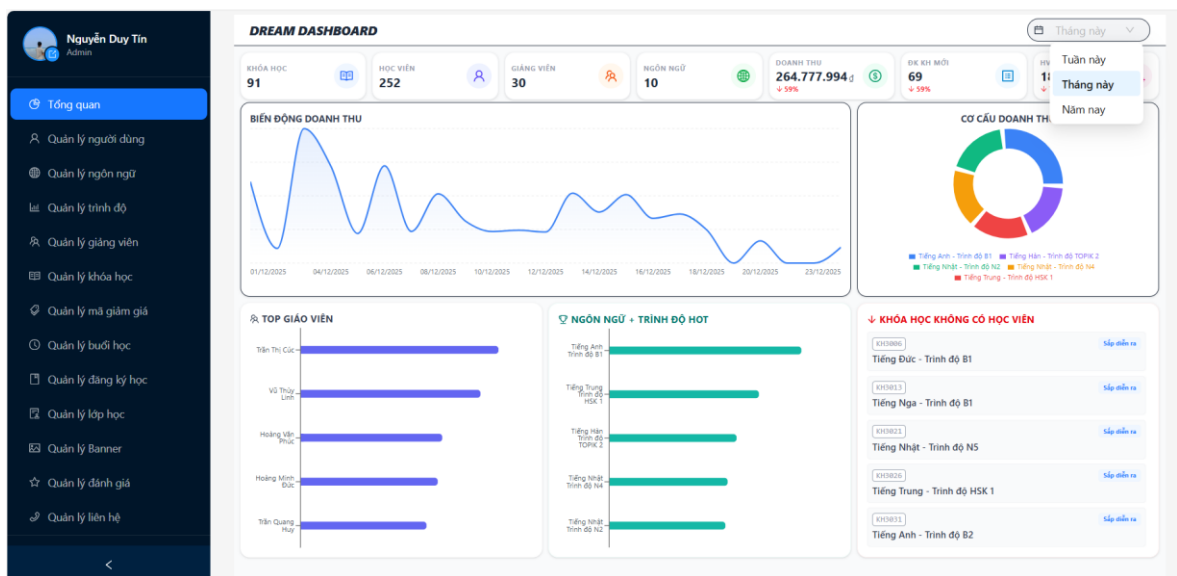


Hình 28: Giao diện trang liên hệ

4.3 Giao diện quản trị viên và chức năng

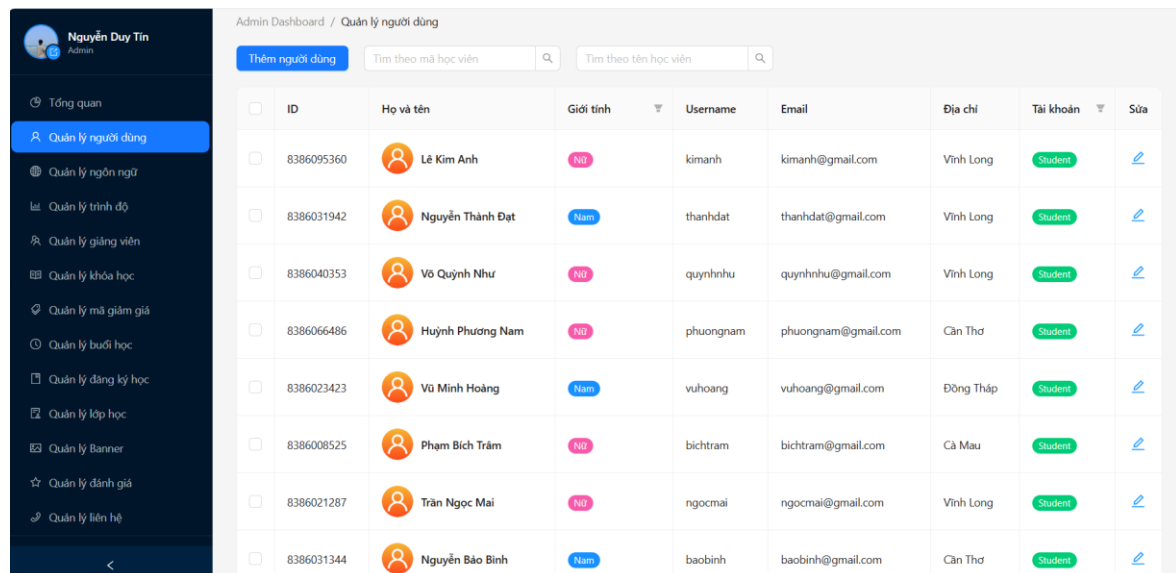
Giao diện trang quản trị được xây dựng với các chức năng quản lý như sau:

Trang tổng quan sẽ hiển thị số lượng thực tế các khóa học, học viên, giảng viên, doanh thu,... hiện có ở trung tâm. Biểu đồ biến động doanh thu, biểu đồ cơ cấu doanh thu theo ngôn ngữ - trình độ, các biểu đồ liên quan đến xếp hạng như top giảng viên, top ngôn ngữ - trình độ nhiều lượt đăng ký, top khóa học không có học viên. Admin có thể lựa chọn khoảng thời gian để theo dõi như tuần này/tháng này/năm nay.



Hình 29: Giao diện trang tổng quan

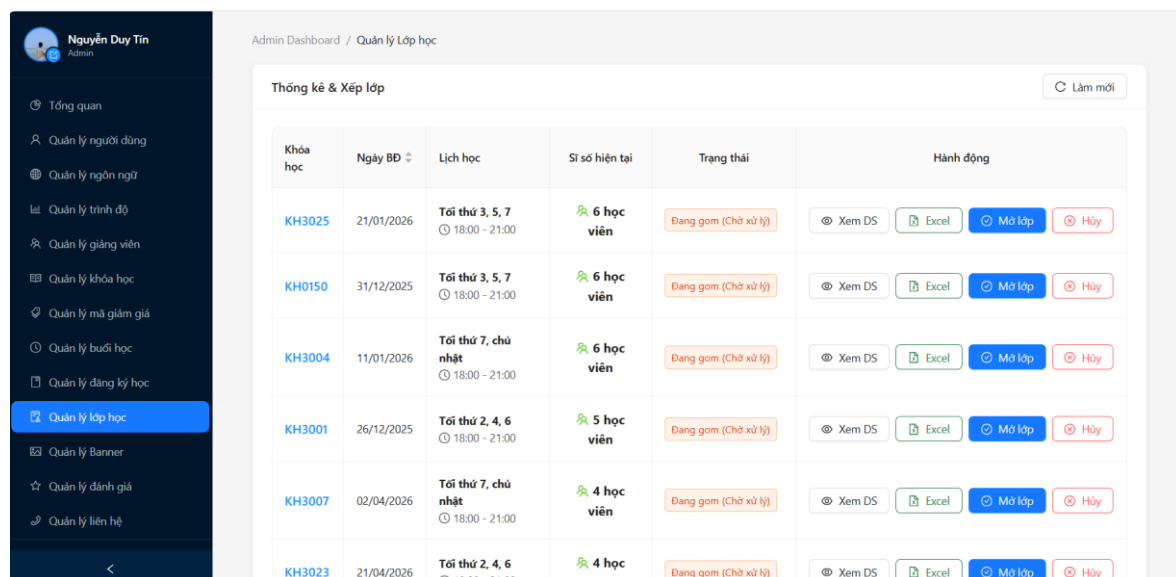
Trang quản lý người dùng hiển thị danh sách tất cả người dùng đang có trong hệ thống, cho phép admin tìm kiếm người dùng theo tên/mã số học viên, lọc theo giới tính, loại tài khoản, thêm mới/sửa/xóa người dùng bất kỳ.



Hình 30: Giao diện trang quản lý người dùng

Các trang như quản lý ngôn ngữ, trình độ,... cũng có chức năng tương tự.

Đối với trang quản lý lớp học, admin sẽ xem xét có nên mở lớp của khóa học và buổi cụ thể hay không dựa vào sĩ số đăng ký hiện tại. Admin sẽ có 4 lựa chọn: xem danh sách học viên đã đăng ký, xuất file excel danh sách học viên, mở lớp hoặc hủy. Nếu đáp ứng các điều kiện cần thiết, admin quyết định mở lớp, trạng thái sẽ chuyển sang đã mở lớp áp dụng cho tất cả học viên đã đăng ký khóa học và buổi học tương ứng. Nếu hủy, tất cả trạng thái đăng ký của học viên sẽ chuyển sang đã hủy. Admin có thể liên hệ với học viên để hoàn tiền (nếu có).



Hình 31: Giao diện trang quản lý lớp học

4.4 So sánh với 2 website đã khảo sát ở chương 2

Về mô hình hoạt động và nghiệp vụ

Các website được khảo sát (khoahocgiaioi.com và khokhoahoc.org) chủ yếu hoạt động theo mô hình thương mại điện tử bán tài liệu số. Đặc điểm chung của các hệ thống này là tập trung vào việc phân phối các khóa học dạng video ghi hình sẵn hoặc tài liệu học tập. Quy trình nghiệp vụ dừng lại ở các bước: xem danh mục, thêm vào giỏ hàng và thanh toán trực tuyến.

Ngược lại, hệ thống được xây dựng trong đề án hướng tới mô hình quản lý đào tạo và chuyển đổi số tại trung tâm. Không chỉ đơn thuần là cung cấp thông tin khóa học, hệ thống giải quyết các bài toán nghiệp vụ phức tạp hơn bao gồm:

Hệ thống quản lý chi tiết đến từng ca học và giảng viên phụ trách, thay vì chỉ quản lý nội dung video.

Quy trình đăng ký gắn liền với việc quản lý các lớp học.

Giao diện và trải nghiệm người dùng

Giao diện của khoahocgiaioi.com tuy đơn giản nhưng cách trình bày màu sắc và bố cục còn gây rối mắt, mang tính chất liệt kê sản phẩm. Trong khi đó, khokhoahoc.org gặp vấn đề về quá tải thông tin với hệ thống danh mục dày đặc, ảnh hưởng đến khả năng tìm kiếm và tiếp nhận thông tin của người dùng.

Khắc phục các nhược điểm trên, đề tài được xây dựng tối giản trên nền tảng MERN Stack kết hợp với công cụ build Vite. Điều này mang lại tốc độ phản hồi nhanh, giảm thiểu độ trễ khi chuyển trang và tạo cảm giác mượt mà. Việc ứng dụng Tailwind CSS cũng giúp giao diện đạt được sự thống nhất, hiện đại và tối ưu hóa hiển thị trên nhiều thiết bị.

Tính năng khác biệt: Tích hợp trí tuệ nhân tạo (Chatbot AI)

Điểm khác biệt lớn nhất và là đóng góp quan trọng của đề tài so với các hệ thống đã khảo sát nằm ở tính năng Trợ lý ảo AI.

Tại các website khảo sát, người dùng hoàn toàn phụ thuộc vào thanh tìm kiếm thủ công hoặc phải chờ đợi nhân viên tư vấn hỗ trợ qua các kênh chat truyền thống.

Hệ thống đã tích hợp thành công mô hình ngôn ngữ lớn (OpenAI) để xây dựng Chatbot tư vấn tự động. Tính năng này cho phép hệ thống tiếp nhận nhu cầu của

người học thông qua ngôn ngữ tự nhiên, phân tích và đưa ra các gợi ý khóa học phù hợp nhất theo nhu cầu. Đây là bước tiến quan trọng trong việc cá nhân hóa trải nghiệm người dùng và tự động hóa khâu tư vấn tuyển sinh - điều mà các website khóa học thương mại thông thường chưa thực hiện được.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau quá trình nghiên cứu và thực hiện, đồ án đã hoàn thành các mục tiêu đề ra ban đầu và đạt được những kết quả cụ thể như sau:

Về mặt sản phẩm và chức năng

Xây dựng thành công hệ thống website hoàn chỉnh, đáp ứng các yêu cầu chức năng từ phía người dùng (học viên) đến người quản trị (Admin).

Số hóa quy trình hoạt động của trung tâm ngoại ngữ từ khâu tìm kiếm thông tin, tư vấn, đăng ký khóa học đến thanh toán.

Hệ thống cho phép học viên thao tác dễ dàng, tìm kiếm khóa học theo nhiều tiêu chí (ngôn ngữ, trình độ), đăng ký trực tuyến, quản lý các khóa học đã đăng ký.

Đối với quản trị viên, hệ thống cung cấp các công cụ quản lý hiệu quả: quản lý người dùng, giảng viên, khóa học,... phê duyệt lớp học và xem các báo cáo thống kê số lượng học viên, doanh thu trực quan qua biểu đồ.

Về mặt công nghệ và kỹ thuật

Ứng dụng thành công các công nghệ web hiện đại (MongoDB, ExpressJS, ReactJS, NodeJS) để xây dựng một hệ thống hoạt động ổn định và bảo mật.

Giao diện người dùng (UI/UX) thân thiện, trực quan nhờ sử dụng ReactJS kết hợp Tailwind CSS, giúp tối ưu trải nghiệm trên các thiết bị.

Tích hợp thành công mô hình ngôn ngữ lớn (OpenAI) để xây dựng tính năng trợ lý ảo AI.

Hạn chế

Ngoài những kết quả đạt được đã nêu, đồ án vẫn còn một vài hạn chế:

Phương thức thanh toán còn hạn chế (VNPay và trực tiếp tại trung tâm).

Chưa phân quyền, chưa có giao diện cho giảng viên.

Chưa có thời khóa biểu cho học viên.

5.2 Hướng phát triển

Viết prompt cho AI chuyên sâu hơn với dữ liệu của trung tâm để tăng độ chính xác trong các câu trả lời.

Tích hợp thêm khả năng nhận diện giọng nói để học viên có thể tương tác với trợ lý ảo bằng lời nói thay vì chỉ nhập văn bản.

Tích hợp đa dạng các cổng thanh toán trực tuyến khác ngoài VNPay như Momo, ZaloPay hoặc thanh toán qua thẻ quốc tế để thuận tiện hơn cho người dùng.

Phân quyền cho giảng viên, xây dựng giao diện cho giảng viên quản lý thời khóa biểu giảng dạy.

Phát triển tính năng xem thời khóa biểu cho học viên.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Vietnix Team, “RESTful API là gì? Cách thiết kế RESTful API chuẩn, hiệu quả.” Accessed: Dec. 22, 2025. [Online]. Available: <https://vietnix.vn/restful-api-la-gi/>
- [2] Vietnix Team, “NodeJS là gì? Tổng hợp thông tin về Node.JS cho người mới.” Accessed: Dec. 22, 2025. [Online]. Available: <https://vietnix.vn/nodejs-la-gi/>
- [3] B. biên tập TopDev, “Expressjs là gì? Tất tần tật về khái niệm Express.js framework,” TopDev. Accessed: Dec. 22, 2025. [Online]. Available: <https://topdev.vn/blog/express-js-la-gi/>
- [4] TopDev, “MongoDB là gì? Định nghĩa và chi tiết về MongoDB,” TopDev. Accessed: Dec. 22, 2025. [Online]. Available: <https://topdev.vn/blog/mongodb-la-gi/>
- [5] “ReactJS là gì: Tính năng nổi bật, cách hoạt động và Lifecycle – ITviec Blog.” Accessed: Dec. 22, 2025. [Online]. Available: <https://itviec.com/blog/reactjs-la-gi/>
- [6] “Vite là gì? & Vite so với Webpack,” techmaster.vn. Accessed: Dec. 22, 2025. [Online]. Available: <https://techmaster.vn/posts/37679/vite-la-gi-vite-so-voi-webpack>
- [7] “Tìm hiểu về Tailwind CSS.” Accessed: Dec. 22, 2025. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-tailwind-css-924lJp6WKPM>
- [8] T. T. K. Nguyen, V. N. Hoang, P. N. Tinh, and N. N. H. Van, “Efficient chatbot for university admission consultation using large language models,” *Tạp Chí Khoa Học Và Công Nghệ - Đại Học Đà Nẵng*, pp. 80–85, Sept. 2025, doi: 10.31130/ud-jst.2025.23(9A).329E.
- [9] “Development of an AI-Powered Faculty Support Chatbot,” *International Journal of Research and Innovation in Applied Science (IJRIAS)*. Accessed: Dec. 24, 2025. [Online]. Available: <https://rsisinternational.org/journals/ijrias/articles/development-of-an-ai-powered-faculty-support-chatbot/>
- [10] Dao T. T., Thuy K. P., and Đào T. T., “NGHIÊN CỨU XÂY DỰNG CHATBOT AI: TINBOT 10 VÀ ĐỀ XUẤT MỘT SỐ BIỆN PHÁP SU

PHẠM SỬ DỤNG TINBOT 10 ĐỂ HỖ TRỢ DẠY HỌC LẬP TRÌNH PYTHON”.

- [11] “Development Of Online Course Registration Management System,” *Elem. Educ. Online*, vol. 20, no. 3, Jan. 2021, doi: 10.17051/ilkonline.2021.03.350.
- [12] A. R. Shrivastava, “Online Course Registration Management System: A Comprehensive Analysis,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 5, pp. 1682–1685, May 2024, doi: 10.22214/ijraset.2024.61929.