

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**Đồ án tốt nghiệp**  
**BÁO CÁO**

**XÂY DỰNG HỆ THỐNG GIÁO DỤC**  
**THÔNG MINH DỰA TRÊN MÔ HÌNH**  
**NGÔN NGỮ LỚN**

Ngành: Khoa học máy tính

**THESIS COMMITTEE:**

**GVHD 1:** Assoc. Prof. Võ Thị Ngọc Châu, PhD

**GVHD 2:** Assoc. Prof. Nguyễn Hứa Phùng, PhD

—o0o—

**Sinh viên 1:** Phan Phạm Thi (2114857)

**Sinh viên 2:** Nguyễn Trường Tuấn Anh (2112796)

**Sinh viên 3:** Đỗ Phương Nam (2114111)

Ho Chi Minh City, June 2024

## PROTESTATION

Báo cáo này được thực hiện với mục tiêu triển khai và phát triển hoàn chỉnh các tính năng của hệ thống học tập trực tuyến thông minh, nhằm đáp ứng nhu cầu học tập cá nhân hóa và tối ưu hóa quá trình học của sinh viên ngành Khoa học máy tính. Trong suốt quá trình thực hiện, nhóm đã gặp phải không ít thử thách, từ việc tối ưu hóa các thuật toán học máy cho đến việc triển khai các tính năng AI trong hệ thống. Tuy nhiên, với sự nỗ lực không ngừng và sự hỗ trợ từ các nguồn tài liệu cũng như người hướng dẫn, nhóm đã hoàn thành mục tiêu phát triển hệ thống như đã đề ra.

Mặc dù đã đạt được nhiều thành công ở giai đoạn 1, nhóm cũng nhận thức rằng một số tính năng trong hệ thống vẫn cần được cải tiến thêm, đặc biệt trong việc tối ưu hóa tốc độ xử lý và tính chính xác của các đề xuất học tập. Tuy nhiên, những khó khăn này sẽ là cơ sở để nhóm tiếp tục hoàn thiện hệ thống trong giai đoạn 2 này.

Chúng tôi cam kết rằng báo cáo này là công trình nghiên cứu độc lập của nhóm. Tất cả các kết quả nghiên cứu, dữ liệu thu thập và phân tích đều được thực hiện một cách trung thực và không sao chép từ bất kỳ công trình nghiên cứu nào trước đó. Mọi nguồn tài liệu tham khảo đều được trích dẫn đầy đủ và rõ ràng trong báo cáo này.

### Authors

*Phan Phạm Thi*

*Nguyễn Trường Tuấn Anh*

*Đỗ Phương Nam*

## **ACKNOWLEDGEMENTS**

Để hoàn thành bài báo cáo này, chúng tôi xin gửi lời cảm ơn chân thành đến các thầy cô, bạn bè và những người đã đồng hành cùng chúng tôi trong suốt quá trình nghiên cứu và phát triển hệ thống. Đặc biệt, chúng tôi xin gửi lời cảm ơn sâu sắc đến giáo viên hướng dẫn, người đã luôn sát cánh, cung cấp cho chúng tôi những kiến thức quý báu và giúp đỡ nhóm hoàn thiện công trình nghiên cứu này. Sự chỉ dẫn tận tình của thầy cô đã giúp chúng tôi nắm vững các phương pháp và công nghệ tiên tiến trong lĩnh vực học máy và trí tuệ nhân tạo, từ đó áp dụng vào việc phát triển phần mềm và hệ thống học tập thông minh.

Ngoài ra, chúng tôi cũng xin cảm ơn các thành viên trong nhóm vì sự hợp tác, chia sẻ và đóng góp không ngừng trong suốt quá trình thực hiện dự án.

### **Authors**

*Phan Phạm Thi*

*Nguyễn Trường Tuấn Anh*

*Đỗ Phương Nam*

## **Abstract**

Báo cáo này trình bày thiết kế và phát triển hệ thống học trực tuyến thông minh dành cho sinh viên ngành Khoa học máy tính, tập trung vào giáo dục lập trình. Mục tiêu chính của hệ thống là cung cấp trải nghiệm học tập được cá nhân hóa bằng cách tận dụng các mô hình ngôn ngữ lớn (LLM) và tối ưu hóa quy trình học tập để linh hoạt, hiệu quả và phù hợp với nhu cầu của từng cá nhân.

Hệ thống nhấn mạnh vào việc tăng cường tương tác giữa người học và trí tuệ nhân tạo (AI), tạo ra môi trường học tập thông minh. Các tính năng chính của hệ thống bao gồm quản lý khóa học, lộ trình học tập được cá nhân hóa, gia sư hỗ trợ AI để hỗ trợ lập trình, tạo bài tập tự động và theo dõi tiến độ. Hệ thống hỗ trợ giảng viên trong việc tạo và quản lý khóa học, đồng thời tự động tạo và đánh giá bài tập, điều chỉnh độ khó dựa trên tiến độ của người học.

Một trong những tính năng nổi bật là gia sư AI, giúp người học hiểu mã nguồn, đưa ra các đề xuất tối ưu hóa mã, hỗ trợ gỡ lỗi và cung cấp phản hồi theo thời gian thực về chất lượng mã. Ngoài ra, hệ thống tạo các bài kiểm tra và bài tập lập trình, phù hợp với tiến độ và nhu cầu của người học.

Hệ thống học trực tuyến thông minh này hướng đến mục tiêu tối ưu hóa trải nghiệm học tập, giúp quá trình học tập trở nên cá nhân hóa, thích ứng và hiệu quả hơn. Nó giúp người học nâng cao kỹ năng lập trình và tiếp cận kiến thức một cách linh hoạt và hiệu quả.

# Mục lục

<b>1</b>	<b>Giới thiệu tổng quan</b>	<b>10</b>
1.1	Nhắc lại mục tiêu đề tài . . . . .	10
1.2	Tóm tắt nội dung đã thực hiện ở Giai đoạn 1 . . . . .	10
<b>2</b>	<b>Tổng quan hệ thống sau khi hoàn thiện</b>	<b>12</b>
2.1	Các chức năng chính . . . . .	12
2.1.1	Quản lý tài khoản (Thành viên đảm nhiệm: Phạm Thi) . . . . .	12
2.1.2	Quản lý khóa học (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh) . . . . .	13
2.1.3	Quản lý bài học (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh) . . . . .	14
2.1.4	Quản lý bài tập (Thành viên đảm nhiệm: Tuấn Anh) . . . . .	14
2.1.5	Quản lý lộ trình học tập (Thành viên đảm nhiệm: Phạm Thi) . . . . .	15
2.1.6	Theo dõi tiến độ học tập (Thành viên đảm nhiệm: Phạm Thi) . . . . .	15
2.1.7	Quản lý phản hồi (Thành viên đảm nhiệm: Phạm Thi) . . . . .	15
2.1.8	Quản lý người dùng (Thành viên đảm nhiệm: Phạm Thi) . . . . .	16
2.1.9	Quản lý nhật ký đăng nhập (Thành viên đảm nhiệm: Phạm Thi) . . . . .	16
2.1.10	Quản lý dashboard (Thành viên đảm nhiệm: Phạm Thi) . . . . .	16
<b>3</b>	<b>Thiết kế kiến trúc và sơ đồ hệ thống</b>	<b>17</b>
3.1	Kiến trúc tổng thể hệ thống . . . . .	17
3.1.1	Frontend (Client-side): . . . . .	17
3.1.2	Backend (Server-side): . . . . .	17
3.1.3	Cơ sở dữ liệu (Database): . . . . .	17
3.1.4	AI Module (LLM – Large Language Model): . . . . .	17
3.1.5	Hệ thống lưu trữ tài liệu: . . . . .	18

3.2	Sơ đồ cơ sở dữ liệu . . . . .	18
3.2.1	Tổng quan cấu trúc cơ sở dữ liệu . . . . .	19
3.2.1.1	Nhóm người dùng (User Management) . . . . .	19
3.2.1.2	Nhóm khóa học và tài liệu (Course Management) . . . . .	19
3.2.1.3	Nhóm câu hỏi và bài kiểm tra (Assessment) . . . . .	19
3.2.1.4	Nhóm tương tác và phản hồi (Interaction) . . . . .	19
3.2.1.5	Nhóm bài tập Code Exercises . . . . .	19
3.2.2	Mối quan hệ giữa các bảng . . . . .	19
3.2.3	Kết luận . . . . .	20
<b>4</b>	<b>Giao diện người dùng</b>	<b>21</b>
4.1	Landing Page . . . . .	21
4.2	Đăng nhập . . . . .	22
4.3	Profile . . . . .	22
4.4	Sinh viên . . . . .	23
4.4.1	Giao diện chính . . . . .	23
4.4.2	Quản lý khóa học . . . . .	23
4.4.3	Chi tiết khóa học . . . . .	24
4.4.4	Chi tiết bài học . . . . .	24
4.4.5	Chi tiết module . . . . .	25
4.4.6	Chi tiết bài quiz . . . . .	26
4.5	Giảng viên . . . . .	27
4.6	Admin . . . . .	27
4.6.1	Giao diện chính . . . . .	27
4.6.2	Quản lý khóa học . . . . .	27
4.6.3	Quản lý người dùng . . . . .	28
4.6.4	Quản lý phản hồi . . . . .	28
4.6.5	Thêm người dùng . . . . .	29
4.6.6	Thêm khóa học . . . . .	29
<b>5</b>	<b>Chi tiết cách thức hoạt động của một số tính năng chính</b>	<b>30</b>
5.1	Giới thiệu . . . . .	30

5.2	Sinh ra lộ trình học tập . . . . .	30
5.3	Đánh giá kết quả học tập . . . . .	31
5.3.1	Đánh giá quá trình học tập (Progress Tracking Assessment) . . . . .	31
5.3.2	Đánh giá kết thúc học tập (Analysis Results) . . . . .	32
5.4	Các chức năng khác . . . . .	32
5.4.1	Tạo câu hỏi trắc nghiệm (Generate Quiz) . . . . .	32
5.4.2	Theo dõi thời gian học tập (Track Study Time) . . . . .	33
5.4.3	Tái tạo nội dung bài học (Regenerate Lesson Content) . . . . .	34
5.5	Kết luận . . . . .	34
<b>6</b>	<b>Kiểm thử và tối ưu hóa</b>	<b>35</b>
6.1	Các API cần kiểm thử . . . . .	35
6.1.1	Đăng nhập và Xác thực . . . . .	35
6.1.2	Quản lý khóa học . . . . .	35
6.1.3	Quản lý bài học . . . . .	35
6.1.4	Quản lý bài tập . . . . .	36
6.1.5	Quản lý phản hồi . . . . .	36
6.1.6	Quản lý người dùng . . . . .	36
<b>7</b>	<b>Kết quả đạt được</b>	<b>37</b>
7.1	Mức độ hoàn thiện hệ thống . . . . .	37
7.2	Các kết quả nổi bật . . . . .	37
7.3	Kết quả triển khai và kiểm thử . . . . .	38
<b>8</b>	<b>Đánh giá</b>	<b>39</b>
8.1	Tính đúng đắn và hiệu quả của AI (LLM) . . . . .	39
8.2	Tính đầy đủ và ổn định của chức năng . . . . .	39
8.3	Trải nghiệm người dùng (UX/UI) . . . . .	39
8.4	Tính mở rộng và bảo trì . . . . .	40
8.5	Tính ứng dụng thực tế . . . . .	40
<b>9</b>	<b>Hướng phát triển trong tương lai</b>	<b>41</b>
9.1	Cải thiện chất lượng AI . . . . .	41



9.2	Mở rộng loại bài tập và nội dung học . . . . .	41
9.3	Hỗ trợ đánh giá toàn diện và tương tác AI nâng cao . . . . .	41
9.4	Tối ưu giao diện và trải nghiệm người dùng . . . . .	41
<b>10</b>	<b>Kết luận</b>	<b>43</b>



# Danh sách hình vẽ

2.1	Flow Định danh tài khoản . . . . .	13
3.1	Sơ đồ ERD của hệ thống . . . . .	18
4.1	Giao diện trang chủ của hệ thống . . . . .	21
4.2	Giao diện đăng nhập của hệ thống . . . . .	22
4.3	Giao diện thông tin cá nhân của sinh viên . . . . .	22
4.4	Giao diện chính của sinh viên . . . . .	23
4.5	Giao diện quản lý khóa học của sinh viên . . . . .	23
4.6	Giao diện chi tiết khóa học của sinh viên . . . . .	24
4.7	Giao diện chi tiết bài học của sinh viên . . . . .	24
4.8	Danh sách các modules học tập của bài học . . . . .	25
4.9	Giao diện chi tiết module của sinh viên . . . . .	25
4.10	Giao diện chi tiết bài quiz của sinh viên . . . . .	26
4.11	Danh sách các bài quiz của sinh viên . . . . .	26
4.12	Giao diện chính của quản trị viên . . . . .	27
4.13	Giao diện quản lý khóa học của quản trị viên . . . . .	27
4.14	Giao diện quản lý người dùng của quản trị viên . . . . .	28
4.15	Giao diện quản lý phản hồi của quản trị viên . . . . .	28
4.16	Giao diện thêm người dùng của quản trị viên . . . . .	29
4.17	Giao diện thêm khóa học của quản trị viên . . . . .	29

# Danh sách bảng

7.1	So sánh với mục tiêu ban đầu . . . . .	38
-----	--	----

# Chương 1

## Giới thiệu tổng quan

### 1.1 Nhắc lại mục tiêu đề tài

Mục tiêu của giai đoạn 2 là triển khai hoàn chỉnh các tính năng đã được lên kế hoạch và thiết kế trong giai đoạn 1. Ở giai đoạn 1, mục tiêu của nhóm như sau:

- **Xây dựng hệ thống học tập cá nhân hóa:** Hệ thống sẽ tập trung vào việc cá nhân hóa trải nghiệm học cho từng học viên, phù hợp với khả năng, trình độ, và nhu cầu học tập riêng biệt. Mỗi học viên có một lộ trình học và cách giải thích riêng, giúp họ tiến bộ một cách hiệu quả nhất. Với khả năng phân tích và hiểu ngữ cảnh của LLM, hệ thống có thể đưa ra các phản hồi phù hợp với nhu cầu học tập của từng học viên.
- **Xác định tính khả thi của việc tích hợp LLM trong giáo dục:** Đánh giá khả năng tích hợp mô hình ngôn ngữ lớn vào hệ thống giáo dục, đặc biệt là trong bối cảnh giáo dục lập trình. Chúng tôi sẽ nghiên cứu tính hiệu quả của LLM trong việc cá nhân hóa và nâng cao khả năng tiếp thu kiến thức của học viên.
- **Giải quyết vấn đề lạm dụng LLM trong giải bài tập:** Hiện nay, sinh viên có thể lợi dụng LLM để giải bài tập lập trình mà không thực sự học. Mục tiêu của đề tài là nghiên cứu xem liệu có thể khiến LLM không trực tiếp đưa ra lời giải cho học sinh mà thay vào đó là cung cấp các câu hỏi gợi mở hoặc hướng dẫn giúp học viên tự giải quyết vấn đề. Điều này nhằm phát triển tư duy giải quyết vấn đề của học viên, thay vì chỉ đưa ra câu trả lời.

### 1.2 Tóm tắt nội dung đã thực hiện ở Giai đoạn 1

Giai đoạn 1 của dự án đã hoàn thành việc phát triển các tính năng cơ bản của hệ thống hỗ trợ học tập trực tuyến, tạo nền tảng vững chắc cho các bước triển khai tiếp theo. Các tính năng

chính như trang dashboard, course list, course detail, và hệ thống đề xuất bài học đã được triển khai đầy đủ, giúp sinh viên dễ dàng truy cập các khóa học, theo dõi tiến độ học tập và nhận được các gợi ý học tập cá nhân hóa.

Các mô-đun học tập như quiz, bài tập lập trình (code exercises), và tài liệu đọc (reading material) được phát triển để không chỉ cung cấp lý thuyết mà còn tạo cơ hội cho sinh viên thực hành và tự kiểm tra khả năng của mình. Bên cạnh đó, hệ thống cũng ghi nhận các thống kê về thời gian học và tiến độ học tập, từ đó cung cấp các báo cáo chi tiết và đánh giá hiệu quả học tập của sinh viên.

Mặc dù các tính năng cơ bản đã được triển khai, vẫn còn nhiều vấn đề cần giải quyết trong các giai đoạn tiếp theo, bao gồm việc hoàn thiện phần authentication, tối ưu hóa công nghệ và quy trình làm việc, cải thiện độ chính xác của các mô hình ngôn ngữ lớn (LLMs), và mở rộng các tính năng học tập cho giảng viên và quản trị viên. Những vấn đề này sẽ là trọng tâm của giai đoạn 2, với mục tiêu hoàn thiện hệ thống và tối ưu hóa trải nghiệm học tập của người dùng.

## Chương 2

# Tổng quan hệ thống sau khi hoàn thiện

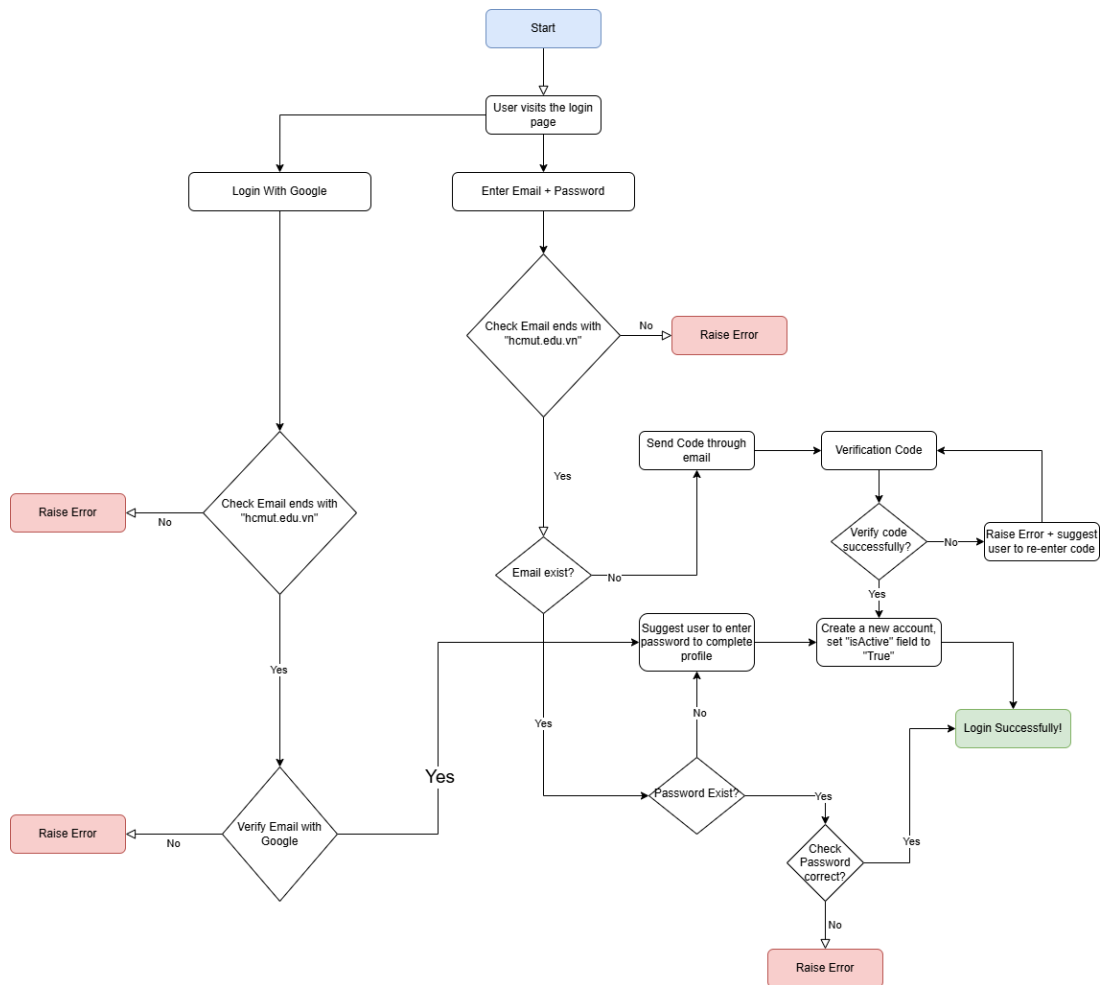
### 2.1 Các chức năng chính

Dưới đây là danh sách các yêu cầu chức năng của hệ thống, được phân loại theo từng module chức năng. Mỗi module đại diện cho một nhóm chức năng liên quan, hỗ trợ các hoạt động học tập, giảng dạy và quản lý hệ thống.

#### 2.1.1 Quản lý tài khoản (Thành viên đảm nhiệm: Phạm Thi)

Module này cung cấp các chức năng liên quan đến đăng nhập, xác thực và quản lý thông tin cá nhân của người dùng.

- Đăng nhập vào hệ thống bằng email/mật khẩu hoặc tài khoản Google (Role: Sinh viên, Giảng viên, Quản trị viên).
- Làm mới token xác thực khi phiên đăng nhập hết hạn (Role: Sinh viên, Giảng viên, Quản trị viên).
- Xác minh email và gửi lại mã xác minh nếu cần (Role: Sinh viên, Giảng viên, Quản trị viên).
- Đặt lại mật khẩu khi quên mật khẩu (Role: Sinh viên, Giảng viên, Quản trị viên).
- Cập nhật thông tin cá nhân (tên, ngày sinh, ảnh đại diện) (Role: Sinh viên, Giảng viên, Quản trị viên).
- Xem thông tin hồ sơ cá nhân (ID, tên, email, MSSV/MSCB, ngày sinh, vai trò) (Role: Sinh viên, Giảng viên, Quản trị viên).



**Hình 2.1:** Flow Định danh tài khoản

### 2.1.2 Quản lý khóa học (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh)

Module này hỗ trợ quản lý thông tin khóa học, bao gồm tạo, xem, cập nhật và xóa dữ liệu liên quan.

- Xem danh sách khóa học (đã đăng ký hoặc đang phụ trách, hỗ trợ phân trang và tìm kiếm) (Role: Sinh viên, Giảng viên).
- Xem chi tiết khóa học (số lượng sinh viên, bài học, bài tập, tài liệu, tiến độ học tập) (Role: Sinh viên, Giảng viên).
- Xem thông tin Giảng viên của khóa học (Role: Sinh viên).
- Cập nhật mục tiêu học tập và ảnh khóa học (Role: Giảng viên).
- Tạo khóa học mới (đơn lẻ hoặc nhiều khóa) (Role: Quản trị viên).
- Cập nhật thông tin khóa học (tên, tín chỉ, học kỳ) (Role: Quản trị viên).

- Xóa khóa học cùng dữ liệu liên quan (Role: Quản trị viên).
- Xem danh sách khóa học có sẵn của HCMUT (Role: Quản trị viên).
- Xem khóa học truy cập gần đây nhất qua dashboard (Role: Sinh viên).

### **2.1.3 Quản lý bài học (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh)**

Module này cho phép tạo, chỉnh sửa và quản lý bài học trong khóa học.

- Xem danh sách bài học trong khóa học (Role: Sinh viên).
- Tạo bài học mới (Role: Giảng viên).
- Cập nhật bài học (tiêu đề, mô tả, mục tiêu học tập) (Role: Giảng viên).
- Xóa bài học cùng tài liệu liên quan (Role: Giảng viên).
- Thêm tài liệu vào bài học (Role: Giảng viên).
- Xem chi tiết bài học và danh sách tài liệu liên quan (Role: Giảng viên).

### **2.1.4 Quản lý bài tập (Thành viên đảm nhiệm: Tuấn Anh)**

Module này hỗ trợ quản lý bài tập dạng quiz và lập trình, bao gồm tạo, xem và nộp bài.

- Xem danh sách bài tập trong khóa học (chỉ bài tập đã mở) (Role: Sinh viên).
- Xem chi tiết bài tập dạng quiz hoặc lập trình (câu hỏi, yêu cầu, test cases) (Role: Sinh viên, Giảng viên).
- Tạo bài tập dạng quiz hoặc lập trình (Role: Giảng viên).
- Cập nhật thông tin bài tập dạng quiz hoặc lập trình (Role: Giảng viên).
- Xóa bài tập dạng quiz hoặc lập trình (Role: Giảng viên).
- Gửi câu hỏi tới trợ lý lập trình và xem lịch sử hội thoại (Role: Sinh viên).
- Nộp bài quiz và xóa câu trả lời để làm lại (Role: Sinh viên).
- Xem danh sách sự kiện bài tập sắp tới qua lịch học (Role: Sinh viên, Giảng viên).

### 2.1.5 Quản lý lộ trình học tập (Thành viên đảm nhiệm: Phạm Thi)

Module này cung cấp các chức năng liên quan đến lộ trình học tập cá nhân hóa và bài học đề xuất.

- Xem lộ trình học tập cá nhân hóa và danh sách bài học đề xuất (Role: Sinh viên).
- Xóa lộ trình học tập cá nhân (Role: Sinh viên).
- Xem chi tiết bài học đề xuất (tiêu đề, mục tiêu, nội dung, tiến độ) (Role: Sinh viên).
- Đánh dấu hoặc bỏ đánh dấu bài học đề xuất (Role: Sinh viên).
- Yêu cầu tạo lộ trình học tập dựa trên mục tiêu và khóa học (Role: Sinh viên).
- Tái tạo nội dung bài học dựa trên vấn đề đã xác định (Role: Sinh viên).
- Nhận gợi ý mục tiêu học tập từ hệ thống (Role: Sinh viên).
- Xem tài liệu của module trong lộ trình học tập (Role: Sinh viên).
- Tạo bài quiz dựa trên nội dung module (Role: Sinh viên).

### 2.1.6 Theo dõi tiến độ học tập (Thành viên đảm nhiệm: Phạm Thi)

Module này hỗ trợ theo dõi và đánh giá tiến độ học tập của sinh viên.

- Nhận đánh giá tiến độ học tập theo chuẩn Rubric (Role: Sinh viên).
- Cập nhật thời gian học cho bài học đề xuất (Role: Sinh viên).
- Xem phân tích tiến độ học tập trong khóa học hoặc bài học cụ thể (Role: Sinh viên).
- Xem điểm số của sinh viên trong khóa học (tên, email, MSSV, điểm trung bình) (Role: Giảng viên).
- Xem điểm chi tiết của một bài tập cụ thể (Role: Giảng viên).

### 2.1.7 Quản lý phản hồi (Thành viên đảm nhiệm: Phạm Thi)

Module này cho phép gửi, xem và quản lý phản hồi trong hệ thống.

- Gửi phản hồi về hệ thống hoặc khóa học (tiêu đề, mô tả, đánh giá) (Role: Sinh viên, Giảng viên).



- Xem danh sách phản hồi (lọc theo tháng, năm, trạng thái, khóa học) (Role: Giảng viên, Quản trị viên).
- Cập nhật trạng thái phản hồi (pending, in\_progress, resolved) (Role: Quản trị viên).
- Xóa phản hồi khỏi hệ thống (Role: Quản trị viên).

### **2.1.8 Quản lý người dùng (Thành viên đảm nhiệm: Phạm Thi)**

Module này cung cấp các chức năng quản lý thông tin và trạng thái người dùng trong hệ thống.

- Tạo người dùng mới (sinh viên, Giảng viên, quản trị viên) (Role: Quản trị viên).
- Đếm số lượng người dùng (tổng hoặc theo vai trò) (Role: Quản trị viên).
- Xem danh sách tất cả người dùng (lọc theo vai trò, trạng thái, tìm kiếm) (Role: Quản trị viên).
- Xem thông tin chi tiết của một người dùng cụ thể (Role: Quản trị viên).
- Cập nhật trạng thái người dùng (bật/tắt) (Role: Quản trị viên).

### **2.1.9 Quản lý nhật ký đăng nhập (Thành viên đảm nhiệm: Phạm Thi)**

Module này hỗ trợ theo dõi và ghi nhận hoạt động đăng nhập của người dùng.

- Xem danh sách nhật ký đăng nhập (ID người dùng, vai trò, thời gian) (Role: Quản trị viên).
- Tạo bản ghi nhật ký đăng nhập mới (Role: Quản trị viên).

### **2.1.10 Quản lý dashboard (Thành viên đảm nhiệm: Phạm Thi)**

Module này cung cấp tổng quan hoạt động cho người dùng qua giao diện dashboard.

- Xem tổng quan hoạt động giảng dạy (số lượng khóa học, bài học, sinh viên, bài tập) (Role: Giảng viên).
- Xem và thêm các hoạt động gần đây (tối đa 5 hoạt động) (Role: Sinh viên).

## Chương 3

# Thiết kế kiến trúc và sơ đồ hệ thống

### 3.1 Kiến trúc tổng thể hệ thống

Hệ thống học tập trực tuyến thông minh được thiết kế theo mô hình Client – Server, với các thành phần chính bao gồm:

#### 3.1.1 Frontend (Client-side):

Được xây dựng bằng Vue 3, giao diện người dùng cung cấp các tính năng như đăng nhập, học bài, làm bài tập, theo dõi tiến độ học tập, tương tác với gia sư AI, gửi phản hồi, ...

Frontend giao tiếp với backend thông qua các API RESTful.

#### 3.1.2 Backend (Server-side):

Đảm nhiệm xử lý logic nghiệp vụ, quản lý dữ liệu người dùng, khóa học, bài học, bài tập, tiến độ học, phản hồi, ... Đồng thời xử lý yêu cầu từ AI (qua LLM API). Backend được xây dựng bằng FastAPI.

#### 3.1.3 Cơ sở dữ liệu (Database):

Sử dụng hệ quản trị cơ sở dữ liệu PostgreSQL để lưu trữ thông tin người dùng, khóa học, bài học, kết quả học tập, bài tập, phản hồi, ... Thiết kế cơ sở dữ liệu đảm bảo tính mở rộng, hỗ trợ việc theo dõi tiến độ và sinh báo cáo.

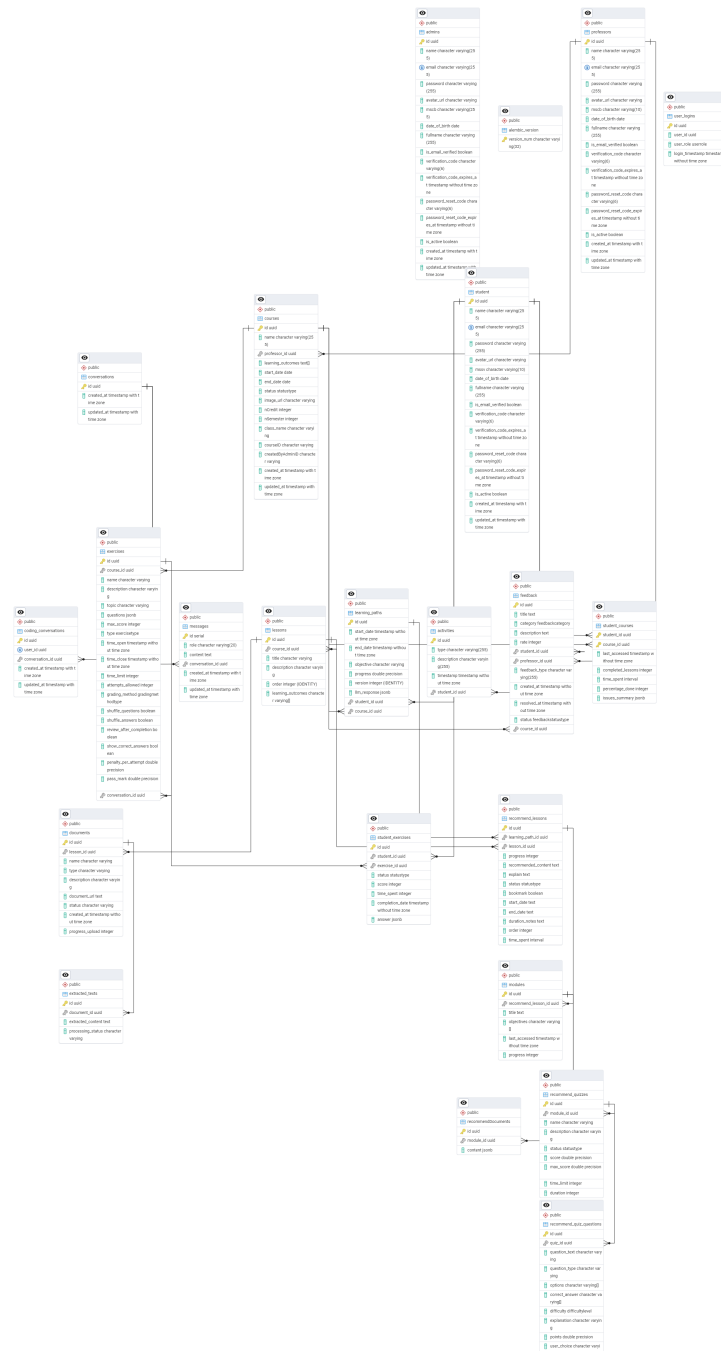
#### 3.1.4 AI Module (LLM – Large Language Model):

Các tác vụ như gợi ý lộ trình học, sinh bài tập, chấm điểm và hỗ trợ giải thích được thực hiện nhờ tích hợp LLM API (OpenAI, Gemini) Backend sẽ định nghĩa prompt phù hợp và gửi yêu cầu tới LLM để lấy kết quả.

### 3.1.5 Hệ thống lưu trữ tài liệu:

Các tài liệu PDF, slide bài giảng được lưu ở AWS S3 thư mục local server.

## 3.2 Sơ đồ cơ sở dữ liệu



**Hình 3.1: Sơ đồ ERD của hệ thống**

### 3.2.1 Tổng quan cấu trúc cơ sở dữ liệu

Hệ thống được xây dựng trên nhiều bảng chính, chia thành các nhóm chức năng như sau:

#### 3.2.1.1 Nhóm người dùng (User Management)

- **professors, student, admins**: chứa thông tin cơ bản của người dùng theo 3 role như username, email, password,...
- **user\_login** bảng mở rộng lưu thông tin đăng nhập người dùng.

#### 3.2.1.2 Nhóm khóa học và tài liệu (Course Management)

- **courses, student\_courses, extracted\_text**: quản lý thông tin khóa học, tham gia khóa học và tài liệu học tập liên quan do giảng viên đăng tải.
- **lessons, documents, modules**: quản lý bài học và các nội dung bài học liên quan.
- **learning\_paths, recommend\_lessons, recommend\_documents**: quản lý các bài kiểm tra, bài tập và câu hỏi liên quan đến khóa học.

#### 3.2.1.3 Nhóm câu hỏi và bài kiểm tra (Assessment)

- **recommend\_quizzes, exercises**: chứa thông tin về các câu hỏi, loại câu hỏi, câu trả lời và kết quả bài kiểm tra của sinh viên trong các bài quiz được sinh ra theo từng bài học đề xuất.

#### 3.2.1.4 Nhóm tương tác và phản hồi (Interaction)

- **feedbacks, activities**: quản lý tương tác giữa sinh viên và giảng viên, người dùng và hệ thống, và các hoạt động của người dùng.

#### 3.2.1.5 Nhóm bài tập Code Exercises

- **conversations, messages, coding\_conversations**: ghi nhận nhật ký hoạt động, cài đặt hệ thống và quản lý phiên làm việc.

### 3.2.2 Môi quan hệ giữa các bảng

Các bảng trong hệ thống có mối quan hệ nhiều-nhiều và một-nhiều, với các khóa ngoại được thể hiện rõ trong sơ đồ:

- Một người dùng có thể tham gia nhiều khóa học thông qua bảng **student\_courses**.

- Một khóa học có thể có nhiều bài học, bài kiểm tra và tài liệu.
- Bài kiểm tra có thể bao gồm nhiều câu hỏi và mỗi câu hỏi có nhiều đáp án.
- Người dùng có thể đưa ra phản hồi và nhận thông báo từ hệ thống.
- Một sinh viên đối với một khóa học có thể có nhiều lộ trình học khác nhau.
- Một lộ trình học có thể có nhiều bài học và tài liệu khác nhau.
- Một bài học có thể có nhiều module nhỏ và nội dung khác nhau.
- Một module có thể có nhiều bài quiz hoặc code exercises khác nhau (do sinh viên tùy chọn sinh ra)

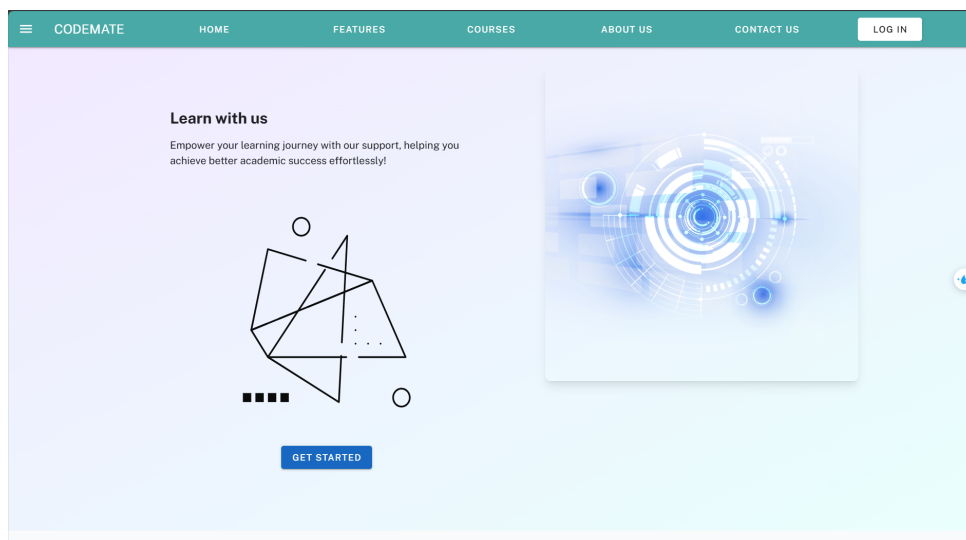
### **3.2.3 Kết luận**

Sơ đồ ERD này cung cấp một cái nhìn toàn diện về kiến trúc cơ sở dữ liệu của hệ thống. Cấu trúc được tổ chức hợp lý, cho phép dễ dàng mở rộng và tích hợp thêm các chức năng mới như AI hỗ trợ học tập, phân tích dữ liệu học viên, và hơn thế nữa.

# Chương 4

## Giao diện người dùng

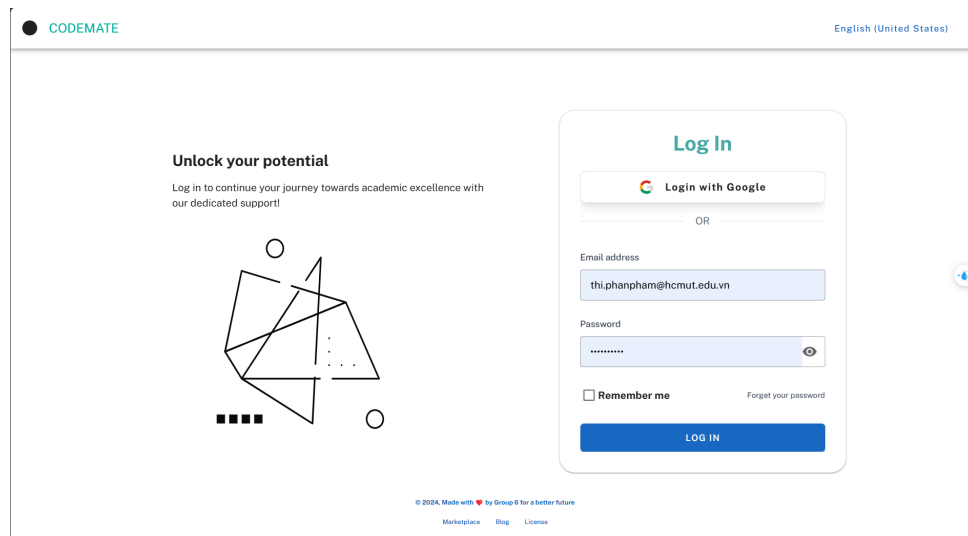
### 4.1 Landing Page



**Hình 4.1:** *Giao diện trang chủ của hệ thống*

Đây là giao diện trang chủ của hệ thống, nơi người dùng có thể tìm hiểu về các chức năng chính của hệ thống và đăng nhập vào tài khoản cá nhân. Giao diện được thiết kế đơn giản, dễ sử dụng và thân thiện với người dùng.

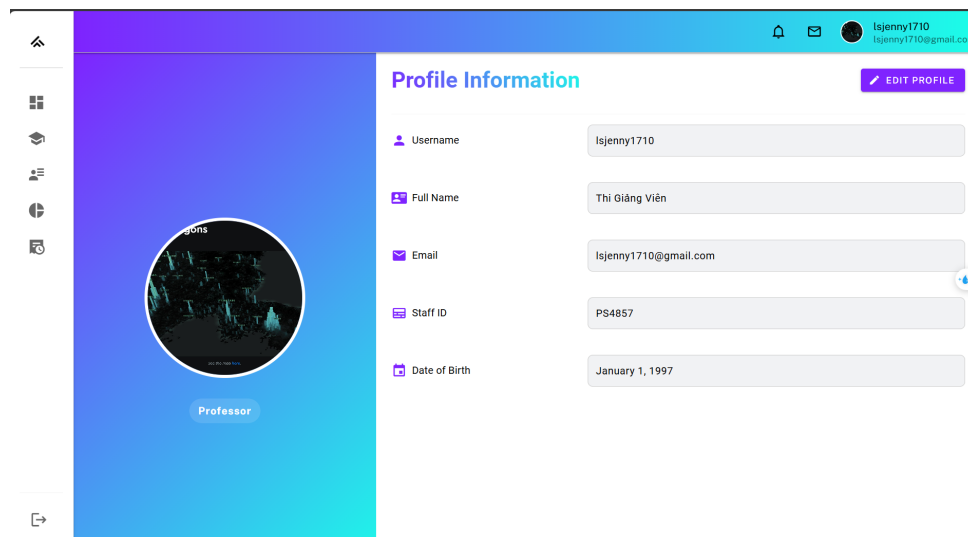
## 4.2 Đăng nhập



Hình 4.2: Giao diện đăng nhập của hệ thống

Giao diện đăng nhập cho phép người dùng đăng nhập vào tài khoản cá nhân bằng email và mật khẩu hoặc thông qua tài khoản Google. Hệ thống hỗ trợ xác thực người dùng và phân quyền theo vai trò (học viên, giảng viên, quản trị viên).

## 4.3 Profile

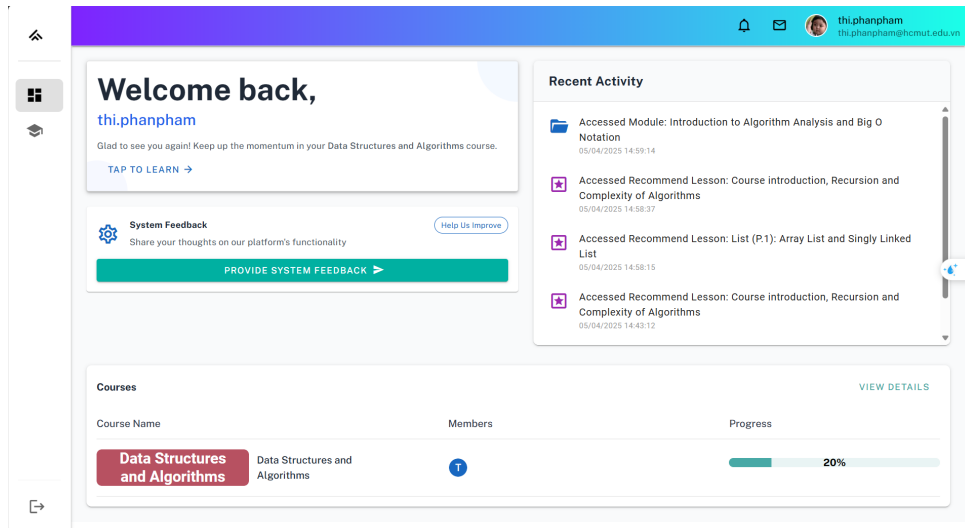


Hình 4.3: Giao diện thông tin cá nhân của sinh viên

Giao diện thông tin cá nhân cho phép người dùng xem và chỉnh sửa thông tin cá nhân của mình, bao gồm tên, email, số điện thoại và ảnh đại diện.

## 4.4 Sinh viên

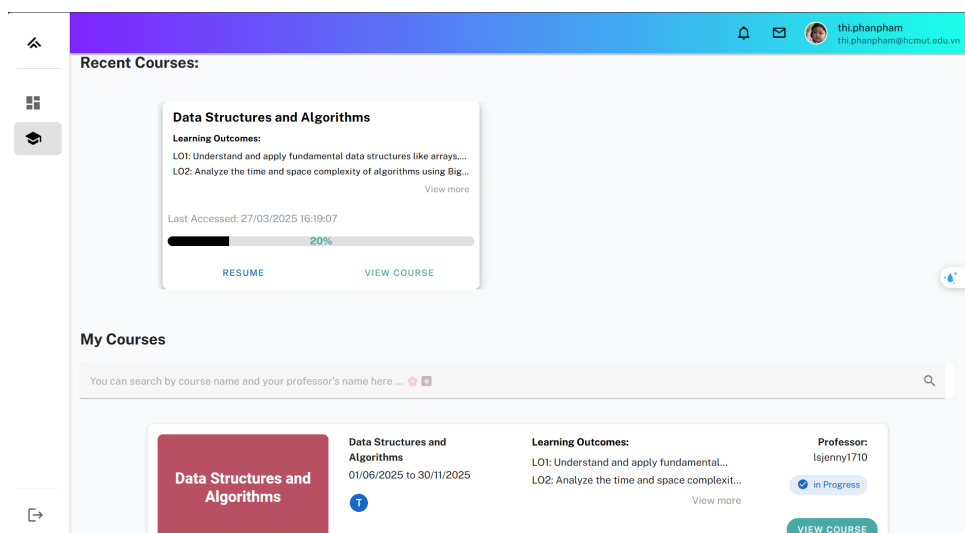
### 4.4.1 Giao diện chính



Hình 4.4: Giao diện chính của sinh viên

Giao diện chính của sinh viên hiển thị danh sách các khóa học mà sinh viên đang theo học. Tại đây, sinh viên có thể xem thông tin khóa học, bài học và bài tập cần hoàn thành. Ngoài ra sinh viên có thể xem các hoạt động gần đây của mình và gửi feedback cho hệ thống nếu có vấn đề gì.

### 4.4.2 Quản lý khóa học



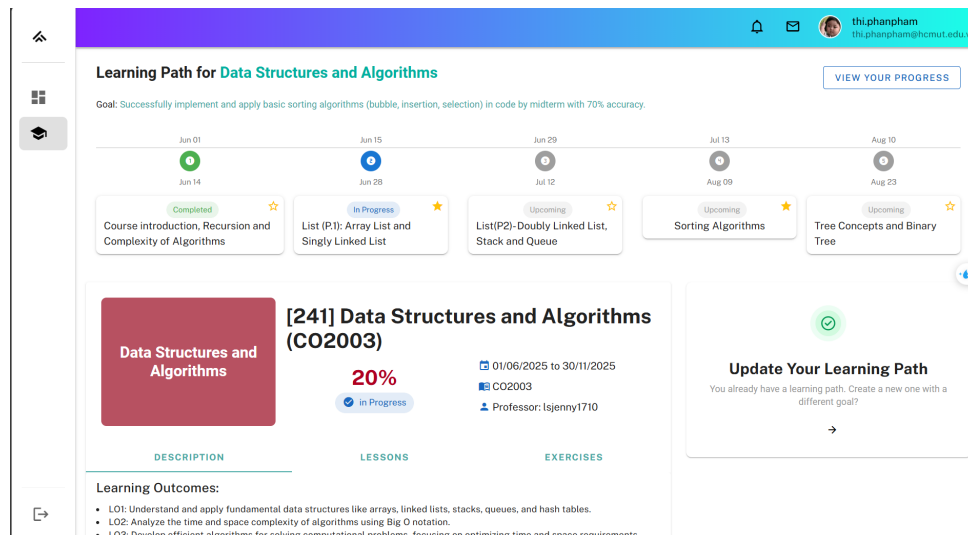
Hình 4.5: Giao diện quản lý khóa học của sinh viên

Giao diện quản lý khóa học cho phép sinh viên xem danh sách các khóa học mà mình đang theo học. Tại đây, sinh viên có thể xem thông tin chi tiết về khóa học, bao gồm tên khóa học, giảng



viên phụ trách, thời gian bắt đầu và kết thúc, Learning Outcomes và trạng thái hoàn thành của khóa học.

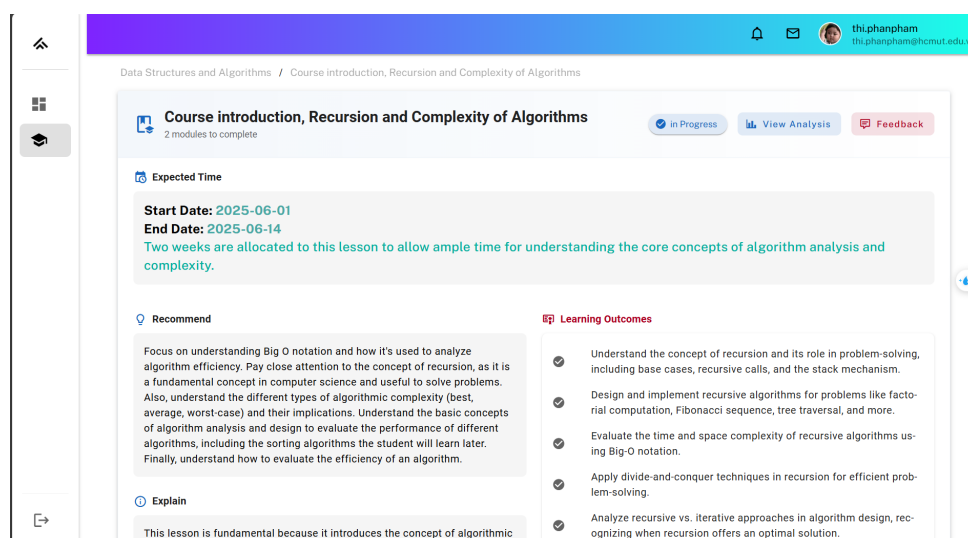
### 4.4.3 Chi tiết khóa học



**Hình 4.6:** Giao diện chi tiết khóa học của sinh viên

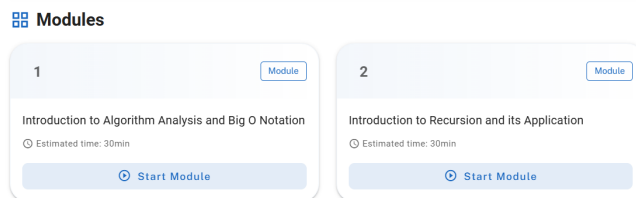
Giao diện chi tiết khóa học cho phép sinh viên xem thông tin chi tiết về khóa học, bao gồm danh sách bài học, bài tập và tiến độ học tập. Tại đây, sinh viên có thể xem các bài học đã hoàn thành và chưa hoàn thành, cũng như các bài tập cần làm. Ngoài ra, nếu sinh viên đã sinh lộ trình học thì có thể xem lộ trình học của mình tại đây.

### 4.4.4 Chi tiết bài học



**Hình 4.7:** Giao diện chi tiết bài học của sinh viên

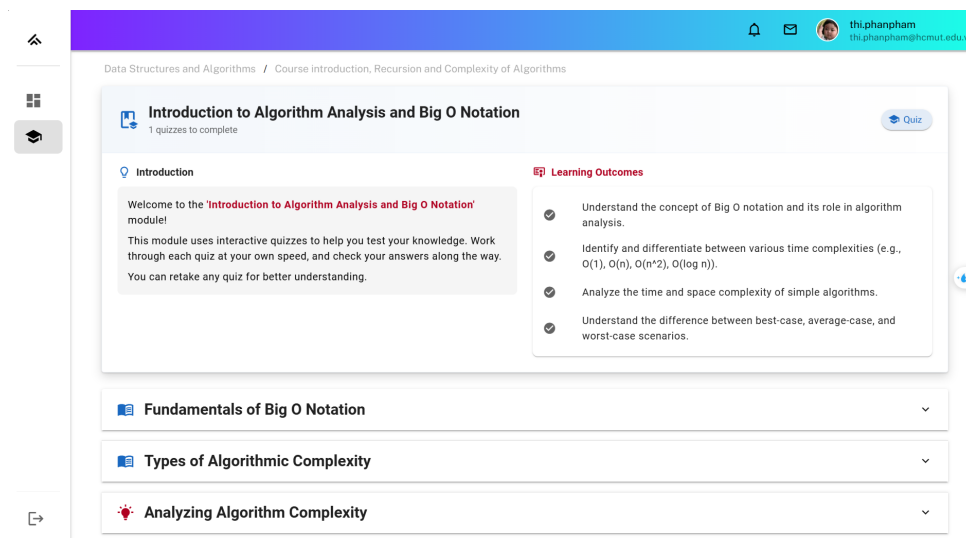
Giao diện chi tiết bài học được đề xuất nằm trong lộ trình học của sinh viên. Tại đây, sinh viên có thể xem thông tin chi tiết về bài học, bao gồm giải thích của AI về nội dung bài học, lý do tại sao bài học này được đề xuất và các modules học tập liên quan. Sinh viên có thể xem ngày bắt đầu và ngày kết thúc của bài học để theo dõi tiến độ học tập của mình. Các modules cần thiết cho bài học sẽ được liệt kê ở dưới cùng.



Hình 4.8: Danh sách các modules học tập của bài học

Ngoài ra, sinh viên có thể xem kết quả phân tích học tập của mình đối với bài học này để biết được mình có đủ kiến thức để qua bài học khác hay không hay cần ôn lại kiến thức ở bài học nào.

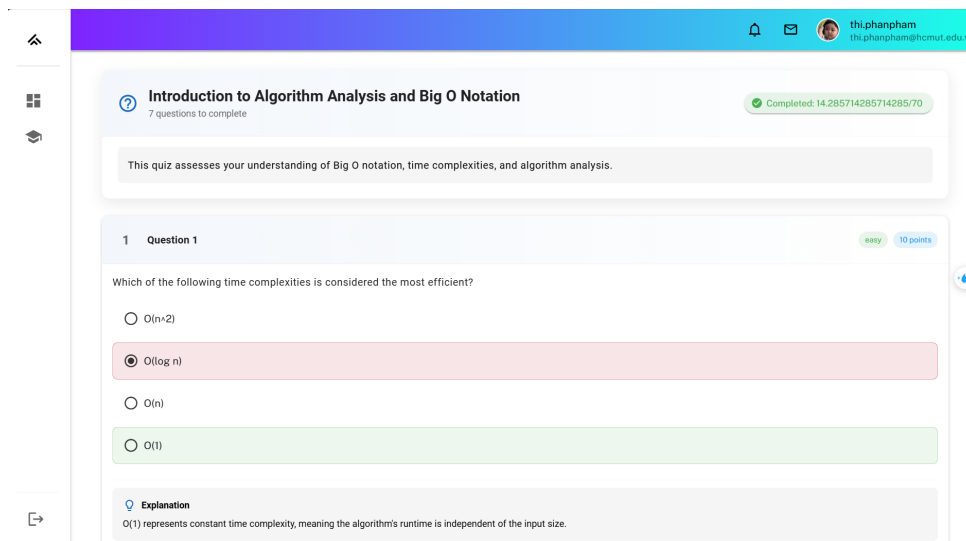
#### 4.4.5 Chi tiết module



Hình 4.9: Giao diện chi tiết module của sinh viên

Giao diện chi tiết module cho phép sinh viên xem chi tiết các kiến thức cần thiết cho bài học. Sinh viên có thể sinh ra các bài quiz tùy chọn số lượng để rèn luyện kiến thức của mình. Hệ thống sẽ tự động chấm điểm và đưa ra nhận xét cho sinh viên sau khi làm bài quiz.

#### 4.4.6 Chi tiết bài quiz



**Hình 4.10:** Giao diện chi tiết bài quiz của sinh viên

Giao diện chi tiết bài quiz cho phép sinh viên làm bài quiz và xem kết quả sau khi làm bài. Hệ thống sẽ tự động chấm điểm và đưa ra nhận xét cho sinh viên sau khi làm bài quiz. Sinh viên có thể làm lại bài quiz nếu chưa hài lòng với kết quả hoặc sinh ra bài quiz mới để làm. Danh sách các bài quiz được hiển thị cuối trang module



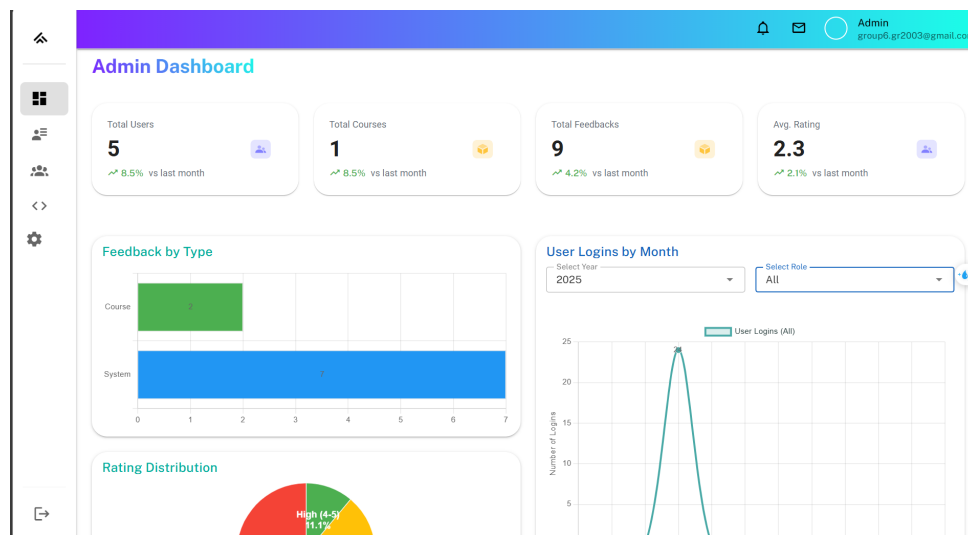
**Hình 4.11:** Danh sách các bài quiz của sinh viên

Danh sách các bài quiz cho phép sinh viên xem danh sách các bài quiz đã làm và kết quả của từng bài quiz. Sinh viên có thể xem lại các câu hỏi và đáp án của mình trong từng bài quiz.

## 4.5 Giảng viên

## 4.6 Admin

### 4.6.1 Giao diện chính



Hình 4.12: Giao diện chính của quản trị viên

Giao diện chính của quản trị viên hiển thị số lượng các khóa học và người dùng trong hệ thống. Tại đây, quản trị viên có thể xem thống kê về lượng người dùng đăng nhập, thống kê về feedback của người dùng.

### 4.6.2 Quản lý khóa học

The Course Management interface includes a search bar, filters for Credits, Semester, Start Date, and End Date, and a table of courses.

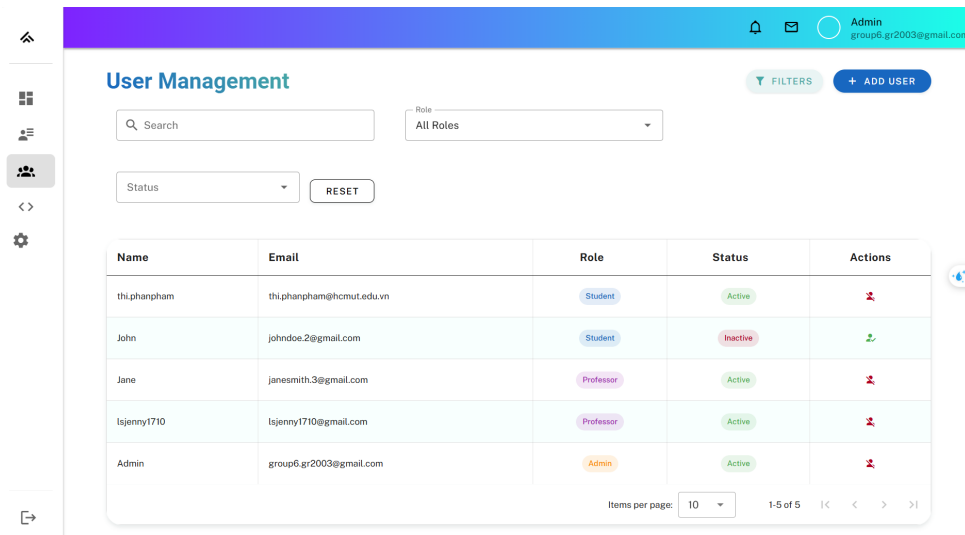
Course ID	Name	Class Name	Credits	Semester	Start Date	End Date	Status	Actions
CO2003	Data Structures and Algorithms	L02	4	241	2025-06-01	2025-11-30	In Progress	<a href="#">Edit</a> <a href="#">Delete</a>

Rows per page: 10

Hình 4.13: Giao diện quản lý khóa học của quản trị viên

Giao diện quản lý khóa học cho phép quản trị viên xem danh sách các khóa học trong hệ thống. Tại đây, quản trị viên có thể thêm, sửa, xóa khóa học và phân quyền cho giảng viên phụ trách khóa học.

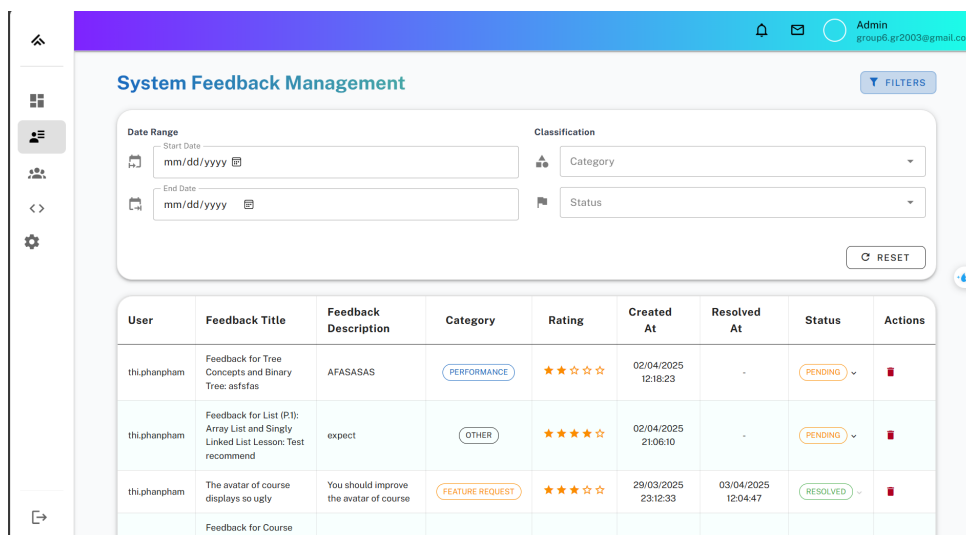
#### 4.6.3 Quản lý người dùng



Hình 4.14: Giao diện quản lý người dùng của quản trị viên

Giao diện quản lý người dùng cho phép quản trị viên xem danh sách các người dùng trong hệ thống. Tại đây, quản trị viên có thể thêm, sửa, xóa người dùng và phân quyền cho người dùng.

#### 4.6.4 Quản lý phản hồi

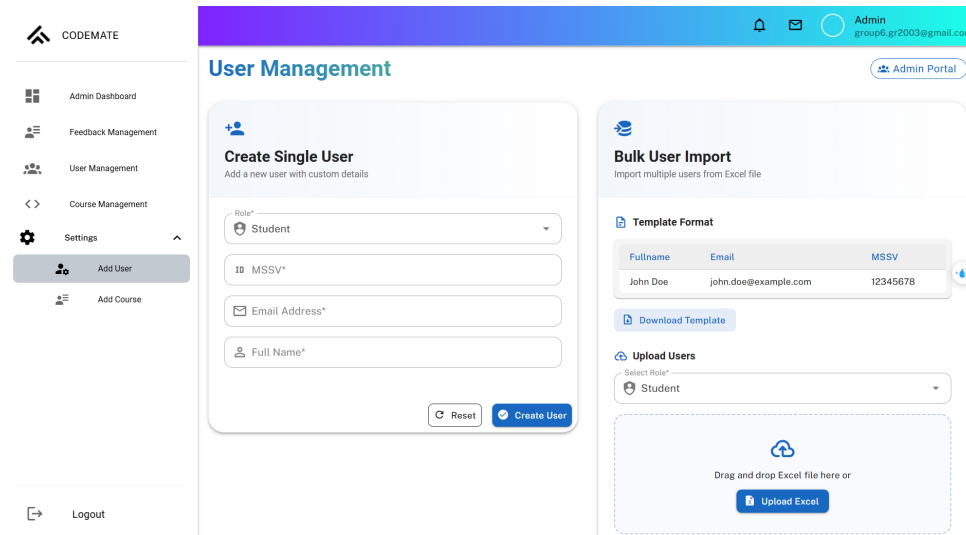


Hình 4.15: Giao diện quản lý phản hồi của quản trị viên

Giao diện quản lý phản hồi cho phép quản trị viên xem danh sách các phản hồi của người dùng trong hệ thống. Tại đây, quản trị viên có thể xem chi tiết phản hồi và trả lời phản hồi của người

dùng.

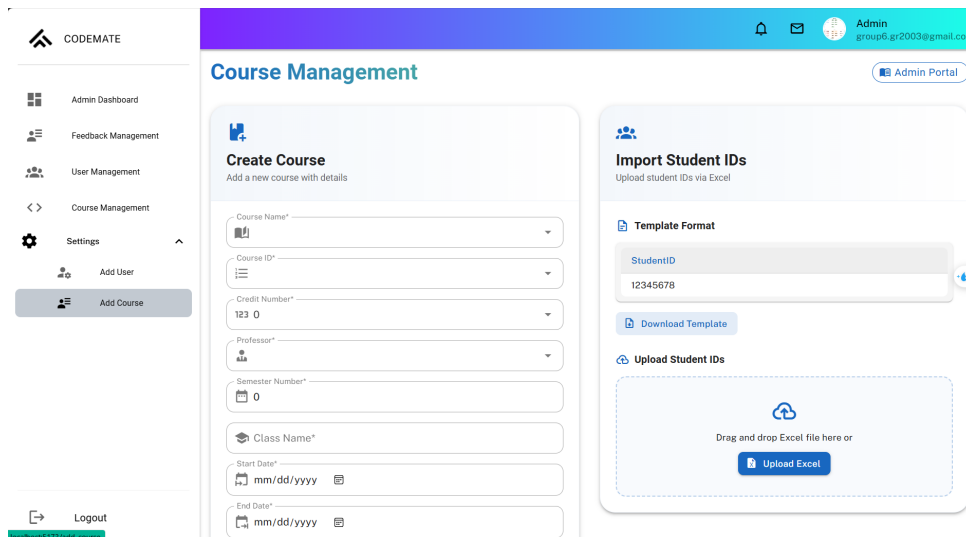
#### 4.6.5 Thêm người dùng



Hình 4.16: Giao diện thêm người dùng của quản trị viên

Giao diện thêm người dùng cho phép quản trị viên thêm người dùng mới vào hệ thống. Tại đây, quản trị viên có thể nhập thông tin người dùng và phân quyền cho người dùng.

#### 4.6.6 Thêm khóa học



Hình 4.17: Giao diện thêm khóa học của quản trị viên

Giao diện thêm khóa học cho phép quản trị viên thêm khóa học mới vào hệ thống. Tại đây, quản trị viên có thể nhập thông tin khóa học và phân quyền cho giảng viên phụ trách khóa học.

## Chương 5

# Chi tiết cách thức hoạt động của một số tính năng chính

### 5.1 Giới thiệu

Hệ thống học tập thông minh được thiết kế để hỗ trợ học sinh trong việc lập kế hoạch học tập và đánh giá tiến độ một cách hiệu quả. Với sự tích hợp của AI, hệ thống có khả năng phân tích dữ liệu học tập, tạo lộ trình cá nhân hóa và đưa ra các đánh giá chi tiết. Báo cáo này tập trung vào các chức năng chính như sinh ra lộ trình học tập, đánh giá kết quả học tập, và các tính năng bổ sung khác.

### 5.2 Sinh ra lộ trình học tập

Chức năng này chịu trách nhiệm tạo ra một lộ trình học tập cá nhân hóa dựa trên mục tiêu học tập của học sinh, nội dung khóa học và thời gian khả dụng. Quá trình được thực hiện qua ba giai đoạn chính:

1. **Phân tích mục tiêu và thời gian:** Hàm `analyze_goal_and_timeline` kiểm tra tính hợp lệ của mục tiêu (ví dụ: độ dài, tính liên quan) và xác định thời gian khả thi dựa trên ngày bắt đầu và kết thúc khóa học. Ví dụ mã nguồn:

```
1 async def analyze_goal_and_timeline(goal: str, course_start_date:
2     datetime, ...):
3     if len(goal) < 10:
4         raise ApplicationException(message="Goal is too short...")
5     prompt = f"... "
6     response = chunking_manager.call_llm_api(prompt, ...)
```

2. **Lựa chọn bài học liên quan:** Hàm `select_relevant_lessons` lọc các bài học phù hợp với mục tiêu, sắp xếp theo thứ tự logic và điều chỉnh theo thời gian. AI được sử dụng để ưu tiên các bài học nền tảng nếu thời gian ngắn.
3. **Tạo lộ trình chi tiết:** Hàm `generate_detailed_learning_path` xây dựng lộ trình đầy đủ, bao gồm các bài học được đề xuất và phân bổ thời gian cụ thể. Kết quả được lưu vào cơ sở dữ liệu qua endpoint `/generate-learning-path`.

Kết quả là một JSON chứa thông tin chi tiết về lộ trình, ví dụ:

```
1 {  
2   "learning_path_start_date": "2025-04-06",  
3   "recommend_lessons": [{"lesson_id": "123", "order": 1, ...}]  
4 }
```

## 5.3 Đánh giá kết quả học tập

Chức năng này bao gồm hai phần phụ: theo dõi quá trình học tập và phân tích kết quả cuối cùng.

### 5.3.1 Đánh giá quá trình học tập (Progress Tracking Assessment)

Phần này sử dụng endpoint `/student/{courseId}/assessment` để đánh giá tiến độ học tập dựa trên phương pháp STAR (Situation, Task, Action, Result). Các bước thực hiện:

- **Tạo prompt chuẩn:** Hàm `generate_standard_prompt` sinh ra một prompt chi tiết gửi đến AI, yêu cầu đánh giá dựa trên các tiêu chí: kiến thức lý thuyết, kỹ năng thực hành (hoặc kỹ năng thay thế), và sự nỗ lực.
- **Phân tích dữ liệu:** AI sử dụng dữ liệu bài học (`progress`, `time_spent`) để đưa ra đánh giá theo Rubric chuẩn, với các mức: Excellent, Good, Average, Poor.
- **Kết quả đầu ra:** Một JSON chứa thông tin đánh giá, ví dụ:

```
1 {  
2   "student_assessment": {  
3     "assessment_summary": {  
4       "situation": "Nguyen Van A is taking a Python course...",  
5       "action": {"theoretical_knowledge": "...", "coding_skills": "...", "effort": "..."},  
6       "result": "With 75% progress as of 2025-03-31, Nguyen Van A is on track..."  
7     },  
9   },  
10 }
```



```
8      "progress_review": {"strengths": "...", "areas_to_note": "..."}
9    }
10 }
```

### 5.3.2 Đánh giá kết thúc học tập (Analysis Results)

Phần này sử dụng hàm `analyze_issues` để phân tích kết quả cuối cùng sau khi học sinh hoàn thành một bài học được đề xuất. Các bước:

- **Phân tích vấn đề:** Dựa trên `issues_summary` từ cơ sở dữ liệu, AI xác định các vấn đề quan trọng (`significant_issues`) và mức độ nghiêm trọng (`severity`).
- **Đề xuất hành động:** Đưa ra một hoặc hai khuyến nghị (`proceed`, `repeat`, `review_prior`), ví dụ:

```
1 {
2   "can_proceed": false,
3   "needs_repeat": true,
4   "recommendations": [
5     {"action": "repeat", "reason": "Significant difficulties with
6       recursion...", "details": "Review 'Advanced Recursion'"}
7   ]
8 }
```

- **Cập nhật tiến độ:** Nếu cần lặp lại, hệ thống có thể tái tạo nội dung bài học qua `regenerate_lesson_content`.

## 5.4 Các chức năng khác

### 5.4.1 Tạo câu hỏi trắc nghiệm (Generate Quiz)

Chức năng này cho phép hệ thống tạo bài kiểm tra trắc nghiệm dựa trên nội dung của một bài học được đề xuất, sử dụng endpoint `/generate-quiz`. Các bước thực hiện:

- **Tập hợp dữ liệu:** Hệ thống lấy thông tin từ bài học (`lesson`), nội dung được đề xuất (`recommended content`), và tài liệu liên quan (`documents`).
- **Tạo câu hỏi:** AI sinh ra các câu hỏi với ba mức độ khó (`easy`, `medium`, `hard`) theo phân phối được yêu cầu, sử dụng mô hình Gemini. Ví dụ mã nguồn:

```
1 async def generate_quiz(request: GenerateQuizRequest, ...):  
2     prompt = f"Generate EXACTLY {questions_needed} {difficulty_name}  
3         difficulty questions..."  
     response = llm.invoke(prompt)
```

- **Lưu trữ:** Câu hỏi và bài kiểm tra được lưu vào cơ sở dữ liệu với thông tin như thời gian giới hạn và điểm tối đa.

Kết quả là một JSON chứa bài kiểm tra, ví dụ:

```
1 {  
2     "quiz_id": "abc123",  
3     "name": "Quiz: Python Basics",  
4     "questions": [{"question_text": "What is a list?", "difficulty": "easy"  
5         , "points": 5, ...}]  
}
```

#### 5.4.2 Theo dõi thời gian học tập (Track Study Time)

Chức năng này theo dõi và cập nhật thời gian học tập của học sinh cho từng bài học được đề xuất, sử dụng endpoint `/student/recommend_lessons/{recommend_lesson_id}/time_spent`. Các bước:

- **Cập nhật thời gian:** Hàm `add_time_spent` nhận thời gian mới (định dạng HH:MM:SS) và cộng dồn vào thời gian hiện có.

```
1 def add_time_spent(existing_time_spent, new_time_spent_str):  
2     new_time_delta = parse_time_spent(new_time_spent_str)  
3     return existing_time_spent + new_time_delta
```

- **Lưu trữ:** Thời gian được cập nhật vào cơ sở dữ liệu qua `recommend_lesson_controller`.

- **Kết quả:** Trả về thời gian đã cập nhật, ví dụ:

```
1 {  
2     "recommend_lesson_id": "xyz789",  
3     "updated_time_spent": "02:30:45"  
4 }
```

### 5.4.3 Tái tạo nội dung bài học (Regenerate Lesson Content)

Chức năng này điều chỉnh nội dung bài học dựa trên các vấn đề học sinh gặp phải, sử dụng endpoint `/regenerate-lesson-content/{recommend_lesson_id}`. Các bước:

- **Phân tích vấn đề:** Dựa trên `issues_summary`, hệ thống xác định các khía cạnh cần cải thiện (ví dụ: hiểu sai khái niệm, lỗi mã).
- **Tạo nội dung mới:** AI sinh nội dung cập nhật với các mô-đun chi tiết, ví dụ mã nguồn:

```
1 async def regenerate_lesson_content(recommend_lesson_id: UUID, issues_  
    summary: dict, ...):  
2     prompt = f"Generate updated content targeting issues ..."  
3     updated_content = chunking_manager.call_llm_api(prompt, ...)
```

- **Cập nhật:** Nội dung mới được lưu vào cơ sở dữ liệu và đánh dấu trạng thái "new".

Kết quả là một JSON chứa nội dung mới, ví dụ:

```
1 {  
2     "recommended_content": "Updated content targeting recursion issues ...",  
3     "modules": [{"title": "Recursion Basics", "objectives": [...], ...}]  
4 }
```

## 5.5 Kết luận

Hệ thống học tập thông minh cung cấp các công cụ mạnh mẽ để cá nhân hóa lộ trình học tập, đánh giá tiến độ, và hỗ trợ học sinh qua các tính năng như tạo bài kiểm tra, theo dõi thời gian, và tái tạo nội dung. Các chức năng này đảm bảo học sinh đạt mục tiêu đúng hạn, đồng thời xác định điểm mạnh, điểm yếu và định hướng cải thiện. Trong tương lai, hệ thống có thể mở rộng với các tính năng như phân tích dữ liệu học tập sâu hơn hoặc tích hợp công cụ học tập tương tác.

# Chương 6

## Kiểm thử và tối ưu hóa

### 6.1 Các API cần kiểm thử

#### 6.1.1 Đăng nhập và Xác thực

- **API đăng nhập** (Role: Sinh viên, Giảng viên, Quản trị viên): Đây là API quan trọng nhất trong mọi hệ thống, cần kiểm tra khả năng đăng nhập qua email/mật khẩu và Google.
- **Làm mới token xác thực** (Role: Sinh viên, Giảng viên, Quản trị viên): Kiểm tra khả năng gia hạn phiên làm việc và làm mới token xác thực sau khi hết hạn.

#### 6.1.2 Quản lý khóa học

- **Xem danh sách khóa học** (Role: Sinh viên, Giảng viên): API này kiểm tra việc hiển thị danh sách khóa học có phân trang và tìm kiếm. Đây là API thường xuyên được sử dụng.
- **Tạo khóa học mới** (Role: Quản trị viên): API này cho phép tạo khóa học mới và kiểm tra tính đúng đắn trong việc lưu trữ thông tin khóa học.
- **Cập nhật thông tin khóa học** (Role: Quản trị viên): Kiểm tra việc chỉnh sửa các thông tin của khóa học như tên, tín chỉ, học kỳ.

#### 6.1.3 Quản lý bài học

- **Tạo bài học mới** (Role: Giảng viên): API tạo bài học mới giúp đảm bảo rằng giảng viên có thể thêm nội dung bài học vào khóa học một cách dễ dàng.
- **Cập nhật bài học** (Role: Giảng viên): Kiểm tra khả năng chỉnh sửa thông tin bài học (tiêu đề, mô tả, mục tiêu học tập).

#### 6.1.4 Quản lý bài tập

- **Xem danh sách bài tập** (Role: Sinh viên): API này kiểm tra việc hiển thị danh sách bài tập (quiz/lập trình) trong khóa học, giúp sinh viên nắm được những bài tập cần hoàn thành.
- **Tạo bài tập mới** (Role: Giảng viên): API này kiểm tra khả năng tạo bài tập mới, bao gồm bài quiz và bài tập lập trình.
- **Nộp bài tập** (Role: Sinh viên): Kiểm tra API nộp bài quiz và khả năng xóa câu trả lời để làm lại.

#### 6.1.5 Quản lý phản hồi

- **Gửi phản hồi về khóa học hoặc hệ thống** (Role: Sinh viên, Giảng viên): Đây là một tính năng quan trọng giúp người dùng gửi ý kiến phản hồi về khóa học hoặc hệ thống. Cần kiểm tra khả năng gửi và lưu trữ phản hồi.
- **Xem và quản lý phản hồi** (Role: Quản trị viên): API này kiểm tra khả năng quản lý và lọc phản hồi để đảm bảo quản trị viên có thể xử lý và theo dõi phản hồi hiệu quả.

#### 6.1.6 Quản lý người dùng

- **Tạo người dùng mới** (Role: Quản trị viên): API này cho phép tạo người dùng mới với các vai trò khác nhau. Đây là một API quan trọng để kiểm tra quy trình tạo người dùng trong hệ thống.
- **Xem danh sách người dùng** (Role: Quản trị viên): API này giúp quản trị viên xem tất cả người dùng, lọc theo vai trò và trạng thái. Đây là tính năng thiết yếu trong việc quản lý người dùng.

# Chương 7

## Kết quả đạt được

### 7.1 Mức độ hoàn thiện hệ thống

Hệ thống đã được xây dựng hoàn chỉnh với các chức năng đã đặt ra trong Giai đoạn 1, bao gồm:

- Đăng ký, đăng nhập, phân quyền người dùng (học viên, giảng viên, quản trị viên).
- Quản lý khóa học và bài học (theo giáo viên hoặc hệ thống tự sinh từ tài liệu).
- Gợi ý lộ trình học tập cá nhân hóa bằng mô hình ngôn ngữ lớn (LLM).
- Chấm điểm tự động và hỗ trợ giải thích bài tập bằng AI.
- Theo dõi tiến độ học tập và sinh báo cáo kết quả học tập.
- Hệ thống phản hồi từ người học về nội dung và trải nghiệm hệ thống.

Tất cả các chức năng chính đã được kiểm thử và hoạt động ổn định trên môi trường triển khai (local).

### 7.2 Các kết quả nổi bật

Hệ thống có thể:

- Tự động đề xuất lộ trình học dựa trên thông tin đầu vào từ người học.
- Tạo bài tập lập trình phù hợp với trình độ và mục tiêu học tập.
- Chấm điểm và đưa ra nhận xét bằng AI giúp người học hiểu sâu hơn về sai sót.
- Ghi nhận tiến độ và phản hồi người học, hỗ trợ giảng viên cải thiện chất lượng giảng dạy.

- Hệ thống có giao diện thân thiện, dễ sử dụng, phù hợp với đối tượng sinh viên ngành CNTT.

## 7.3 Kết quả triển khai và kiểm thử

**Bảng 7.1:** So sánh với mục tiêu ban đầu

Mục tiêu đề ra	Kết quả đạt được
Gợi ý lộ trình học tập cá nhân bằng AI	✓ Hoàn thành
Sinh bài tập lập trình tùy theo bài học hoặc mục tiêu	✓ Hoàn thành
Chấm điểm và giải thích tự động bằng LLM	✓ Hoàn thành
Theo dõi tiến độ và báo cáo	✓ Hoàn thành
Giao diện thân thiện, dễ sử dụng	✓ Hoàn thành

# Chương 8

## Đánh giá

### 8.1 Tính đúng đắn và hiệu quả của AI (LLM)

Đánh giá đầu ra AI:

- 
- 
- 

Tồn tại:

- 
- 

Đánh giá tổng thể:

- ✓

### 8.2 Tính đầy đủ và ổn định của chức năng

### 8.3 Trải nghiệm người dùng (UX/UI)

- Giao diện rõ ràng, dễ sử dụng, đặc biệt cho đối tượng sinh viên CNTT.
- Có phản hồi từ người dùng thử nghiệm: tích cực, dễ nắm bắt chức năng.



**Hạn chế:**

- Một vài giao diện chưa responsive hoặc cần làm mượt hơn thao tác.

## **8.4 Tính mở rộng và bảo trì**

- Cấu trúc hệ thống modular, dễ bảo trì và mở rộng thêm bài tập, loại câu hỏi, hoặc mô-đun học mới.
- Có thể mở rộng tích hợp thêm loại AI khác như sinh trắc nghiệm, kiểm tra chính tả/lỗi logic.

## **8.5 Tính ứng dụng thực tế**

- Hệ thống có tiềm năng ứng dụng trong môi trường học tập đại học hoặc nền tảng học trực tuyến cho người học lập trình.
- Phù hợp với mô hình giáo dục cá nhân hóa đang phát triển.

## Chương 9

# Hướng phát triển trong tương lai

### 9.1 Cải thiện chất lượng AI

- Tối ưu prompt để sinh bài tập chính xác hơn, tránh tình trạng chung chung hoặc sai ngữ cảnh.
- Fine-tune mô hình AI với dữ liệu từ người dùng thực tế (bài làm, phản hồi, feedback) để tăng độ chính xác và cá nhân hóa hơn.

### 9.2 Mở rộng loại bài tập và nội dung học

- Thêm các dạng bài tập mới: trắc nghiệm, điền khuyết, câu hỏi lý thuyết, project-based, ...
- Tích hợp thêm các nguồn tài liệu học (slide, ebook, video) và tự động phân tích sinh lộ trình học từ đó.

### 9.3 Hỗ trợ đánh giá toàn diện và tương tác AI nâng cao

- Cho phép hệ thống đánh giá tiến độ học tập theo nhiều khía cạnh (kiến thức, kỹ năng giải quyết vấn đề, tư duy).
- Phát triển tính năng trợ giảng AI giao tiếp theo phong cách hội thoại tự nhiên, không chỉ trả lời theo prompt đơn lẻ.

### 9.4 Tối ưu giao diện và trải nghiệm người dùng

- Nâng cao trải nghiệm giao diện: hỗ trợ dark mode, responsive tốt hơn trên mobile.

- Thêm các tính năng tương tác như thông báo, nhắc nhở, hoặc gamification để tăng động lực học tập.
- Tinh chỉnh các chức năng theo phản hồi từ người dùng thử nghiệm để cải thiện tính thân thiện và dễ sử dụng.

# Chương 10

## Kết luận

Đồ án đã thực hiện thành công việc xây dựng một hệ thống học lập trình thông minh tích hợp AI (LLM), hỗ trợ người học qua các chức năng như gợi ý lộ trình cá nhân hóa, sinh bài tập phù hợp và tương tác giải thích bài làm. Hệ thống không chỉ đáp ứng đúng yêu cầu đề ra mà còn cho thấy tiềm năng ứng dụng thực tế trong việc nâng cao hiệu quả học tập lập trình.

Thông qua việc triển khai dự án, nhóm đã tích lũy được nhiều kinh nghiệm trong việc thiết kế hệ thống, xây dựng giao diện người dùng, cũng như tích hợp AI vào bài toán thực tế. Những giới hạn còn tồn tại sẽ là tiền đề để nhóm phát triển hệ thống ngày càng hoàn thiện và hữu ích hơn trong tương lai.

## Tài liệu tham khảo

- [1] North Carolina State University. *Rubric Best Practices, Examples, and Templates*. Truy cập ngày 06/04/2025. [https://teaching-resources.delta.ncsu.edu/rubric\\_best-practices-examples-templates/](https://teaching-resources.delta.ncsu.edu/rubric_best-practices-examples-templates/)