

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

HỘI ĐỒNG 4 ĐỒ ÁN TỐT NGHIỆP

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

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

Thư kí: Assoc. Prof. Trần Thị Quế Nguyệt, PhD

—oo—

Sinh viên 1: Phan Phạm Thi (2114857)

Sinh viên 2: Nguyễn Trường Tuấn Anh (2112796)

Sinh viên 3: Đỗ Phương Nam (2114111)



Thành phố Hồ Chí Minh, tháng 5 năm 2025



Lời cam kết

Bài báo cáo này được thực hiện với mục tiêu nghiên cứu và phát triển một hệ thống học tập trực tuyến thông minh, đáp ứng nhu cầu học tập cá nhân hóa và tối ưu hóa quá trình học của người dùng, đặc biệt là sinh viên ngành Khoa học máy tính thuộc khoa Khoa học & Kỹ thuật máy tính, trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh. Trong quá trình thực hiện, chúng tôi đã gặp không ít thử thách và khó khăn, từ việc nghiên cứu các mô hình ngôn ngữ lớn (LLM) cho đến việc triển khai các tính năng AI hỗ trợ trong hệ thống. Tuy nhiên, với sự nỗ lực và hỗ trợ từ nhiều phía, chúng tôi đã và đang hoàn thành được mục tiêu nghiên cứu và phát triển hệ thống như mong muốn.

Mặc dù vậy, chúng tôi cũng nhận thức được rằng trong quá trình thực hiện vẫn còn những hạn chế và khó khăn, đặc biệt trong việc tối ưu hóa một số tính năng AI và việc tạo ra các bài tập tự động. Tuy nhiên, những thử thách này sẽ là nền tảng để chúng tôi tiếp tục cải tiến hệ thống trong các giai đoạn phát triển tiếp theo. Chúng tôi hy vọng rằng những đóng góp của chúng tôi sẽ mang lại giá trị thực tế cho việc học tập và giảng dạy lập trình, đồng thời mở ra cơ hội ứng dụng rộng rãi hơn trong môi trường giáo dục.

Chúng tôi xin cam kết rằng báo cáo này là công trình nghiên cứu độc lập của nhóm tôi. 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. Nhóm tôi cam kết rằng những kết quả nghiên cứu trong báo cáo này phản ánh đúng những nỗ lực và công sức mà chúng tôi đã bỏ ra trong suốt quá trình thực hiện. Bất kỳ sự trợ giúp hay đóng góp nào từ bên ngoài đều được ghi nhận đầy đủ và không có sự gian lận hay sao chép.

Tác giả

Phan Phạm Thi

Nguyễn Trường Tuấn Anh

Đỗ Phương Nam



Lời cảm ơn

Để hoàn thành bài báo cáo này, chúng tôi xin gửi lời cảm ơn sâu sắc đến các thầy cô, bạn bè và những người đã đồng hành cùng tôi trong suốt quá trình nghiên cứu và phát triển hệ thống. Đặc biệt, chúng tôi xin gửi lời cảm ơn đến giáo viên hướng dẫn, người đã cung cấp cho chúng tôi những kiến thức quý báu, giúp chúng tôi hoàn thiện công trình nghiên cứu này. Sự chỉ dẫn tận tình của thầy cô đã giúp chúng tôi hiểu rõ hơn về các phương pháp và công nghệ tiên tiến trong lĩnh vực học máy và trí tuệ nhân tạo cũng như ứng dụng chúng vào trong lĩnh vực phát triển phần mềm - công nghệ phần mềm.

Tác giả

Phan Phạm Thị

Nguyễn Trường Tuấn Anh

Đỗ Phương Nam



Tóm tắt

Báo cáo này trình bày thiết kế và phát triển một hệ thống học tập trực tuyến thông minh dành cho sinh viên ngành Khoa học máy tính và những người quan tâm đến việc học lập trình. Mục tiêu chính của hệ thống là cung cấp trải nghiệm học tập cá nhân hóa thông qua việc ứng dụng các mô hình ngôn ngữ lớn (LLM), tối ưu hóa quá trình học tập để trở nên linh hoạt, hiệu quả và phù hợp với nhu cầu cụ thể của từng người học.

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

Một tính năng nổi bật của hệ thống là gia sư AI, hỗ trợ người học trong việc hiểu mã nguồn, đưa ra gợi ý tối ưu hóa mã, hỗ trợ sửa lỗi và cung cấp phản hồi thời gian thực về chất lượng mã lệnh. Ngoài ra, hệ thống còn tự động tạo bài tập, bao gồm các câu hỏi trắc nghiệm và bài tập lập trình, được điều chỉnh theo tiến độ học và nhu cầu của người dùng.

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

Mục lục

1	Giới thiệu về đề tài	13
1.1	Động lực	13
1.2	Mục tiêu	14
1.3	Phạm vi	14
1.4	Ý nghĩa của đề tài	15
1.4.1	Ý nghĩa của dự án dưới góc nhìn thực tiễn	15
1.4.2	Ý nghĩa của dự án theo góc nhìn khoa học	15
1.5	Cấu trúc báo cáo	15
2	Kiến thức nền tảng	17
2.1	Kiến thức nền tảng về giáo dục và giáo dục thông minh	17
2.1.1	Khái quát về giáo dục	17
2.1.2	Khái quát về giáo dục thông minh	18
2.2	Mô hình ngôn ngữ	18
2.3	Các thuật ngữ kỹ thuật liên quan đến giáo dục thông minh	19
2.3.1	Đồ thị tri thức (Knowledge graph)	19
2.3.2	Knowledge tracing	20
2.4	Kiến trúc công nghệ	21
2.4.1	Công cụ xử lý ngôn ngữ tự nhiên - Framework tích hợp LLMs	21
2.4.1.1	LangChain	21
2.4.1.2	LlamaIndex	22
2.4.1.3	Tổng quan	23
2.4.2	Framework xây dựng giao diện người dùng	24
2.4.2.1	React	24



2.4.2.2	Vue	24
2.4.2.3	Angular	25
2.4.2.4	Tổng quan	26
2.4.3	Framework phát triển API	26
2.4.3.1	FastAPI	27
2.4.3.2	Django	27
2.4.3.3	Tổng quan	27
2.5	Tổng kết	28
3	Công trình liên quan	29
3.1	Hệ thống đề xuất lộ trình học	29
3.2	Ứng dụng LLM vào việc giảng dạy khoa học máy tính, lập trình,...	30
3.3	Prompt engineering	32
3.4	Retrieval-Augmentation Generation	32
3.5	Knowledge tracing/Student Profiling	33
4	Hệ thống đề xuất	35
4.1	Mô tả hệ thống	35
4.2	Khảo sát yêu cầu	36
4.2.1	Người dùng	36
4.2.2	Yêu cầu chức năng	36
4.2.2.1	User Management	36
4.2.2.2	Learning Resource Recommendation System	37
4.2.2.3	Exercise Generator	37
4.2.2.4	AI Tutor	37
4.2.2.5	Progress Tracking	38
4.2.2.6	Course Management	38
4.2.2.7	Admin Management	39
4.2.3	Yêu cầu phi chức năng	39
4.2.4	Yêu cầu dữ liệu	41
4.2.4.1	Các thực thể và thuộc tính dữ liệu	41
4.2.4.2	Mối quan hệ dữ liệu	42



4.3	Kết luận	43
5	Phân tích và thiết kế	44
5.1	Phân tích	44
5.1.1	Business Process Modeling	44
5.1.2	Use case diagrams	45
5.1.2.1	Course Management	46
5.1.2.2	User Management	47
5.1.2.3	Learning Resource Recommendation System	48
5.1.2.4	Progress Tracking	49
5.1.2.5	Exercise Generator	51
5.1.2.6	AI Tutor	52
5.2	Thiết kế	54
5.2.1	Kiến trúc hệ thống	54
5.2.1.1	Phân tích kiến trúc hệ thống phổ biến hiện nay	54
5.2.1.2	Lựa chọn kiến trúc hệ thống	55
5.3	Database	59
5.3.1	EER diagram	59
5.3.2	Lược đồ CSDL quan hệ ánh xạ	61
5.3.3	Class diagram	62
5.3.4	Sitemap	63
5.3.4.1	Sinh viên	63
5.3.4.2	Giảng viên	65
5.3.4.3	Manager	66
5.4	Quy trình Đề xuất và Theo dõi Lộ trình Học tập Cá nhân hóa	67
5.4.1	Tổng quan	67
5.4.2	Giai đoạn 1: Phân tích Mục tiêu và Tạo Lộ trình Học tập	68
5.4.3	Giai đoạn 2: Tạo Bài Kiểm tra Tự động	69
5.4.4	Giai đoạn 3: Theo dõi và Đánh giá Tiến độ Học tập	70
5.4.5	Giai đoạn 4: Điều chỉnh Nội dung Học tập	71
5.4.6	Kết luận	72



6 Hiệu thực	73
6.1 Hiệu thực hệ thống	73
6.2 Kiến trúc Backend	77
6.3 Quá trình triển khai hệ thống	78
6.4 Các chức năng đã được thực hiện	81
6.4.1 Quản lý tài khoản (Thành viên đảm nhiệm: Phạm Thi)	81
6.4.2 Quản lý khóa học (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh)	82
6.4.3 Quản lý bài học (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh)	83
6.4.4 Quản lý bài tập (Thành viên đảm nhiệm: Tuấn Anh)	83
6.4.5 Quản lý lộ trình học tập (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh)	84
6.4.6 Theo dõi tiến độ học tập (Thành viên đảm nhiệm: Phạm Thi/Tuấn Anh)	84
6.4.7 Quản lý phản hồi (Thành viên đảm nhiệm: Phạm Thi)	85
6.4.8 Quản lý người dùng (Thành viên đảm nhiệm: Phạm Thi)	85
6.4.9 Quản lý nhật ký đăng nhập (Thành viên đảm nhiệm: Phạm Thi)	85
6.4.10 Quản lý dashboard (Thành viên đảm nhiệm: Phạm Thi)	85
6.5 Giới thiệu chi tiết về các chức năng chính của hệ thống	86
6.5.1 Nguyên tắc SMART trong việc thiết lập mục tiêu	86
6.5.2 Phương pháp Rubric-Based Assessment	86
6.5.3 Phương pháp đánh giá STAR	87
6.6 Sinh ra lộ trình học tập	88
6.7 Đề xuất mục tiêu học tập	89
6.8 Đánh giá kết quả học tập	90
6.8.1 Đánh giá quá trình học tập (Progress Tracking Assessment)	90
6.8.2 Theo dõi và phân tích tiến độ học tập	91
6.8.3 Đánh giá kết quả học tập (Monitor Study Progress)	92
6.9 Tạo bài kiểm tra (Generate Quiz)	93
6.10 Tái tạo nội dung bài học (Regenerate Lesson Content)	94
6.11 Kiến trúc tổng thể hệ thống	95
6.11.1 Frontend (Client-side):	95
6.11.2 Backend (Server-side):	95
6.11.3 Cơ sở dữ liệu (Database):	96



6.11.4	AI Module (LLM – Large Language Model):	96
6.11.5	Hệ thống lưu trữ tài liệu:	96
6.12	Sơ đồ cơ sở dữ liệu	97
6.12.1	Tổng quan cấu trúc cơ sở dữ liệu	97
6.12.1.1	Nhóm người dùng (User Management)	98
6.12.1.2	Nhóm khóa học và tài liệu (Course Management)	98
6.12.1.3	Nhóm câu hỏi và bài kiểm tra (Assessment)	98
6.12.1.4	Nhóm tương tác và phản hồi (Interaction)	98
6.12.1.5	Nhóm bài tập Code Exercises	98
6.12.2	Mối quan hệ giữa các bảng	98
6.12.3	Kết luận	99
6.13	Kết luận	99
7	Giao diện người dùng	100
7.1	Landing Page	100
7.2	Đăng nhập	101
7.3	Profile	101
7.4	Sinh viên	102
7.4.1	Giao diện chính	102
7.4.2	Quản lý khóa học	102
7.4.3	Chi tiết khóa học	103
7.4.4	Chi tiết bài học	103
7.4.5	Chi tiết module	104
7.4.6	Chi tiết bài quiz	105
7.5	Giảng viên	105
7.5.1	Giao diện chính	105
7.5.2	Quản lý khóa học	106
7.5.3	Quản lý phản hồi của sinh viên	107
7.5.4	Quản lý tiến độ sinh viên	107
7.6	Admin	108
7.6.1	Giao diện chính	108



7.6.2	Quản lý khóa học	109
7.6.3	Quản lý người dùng	109
7.6.4	Quản lý phản hồi	110
7.6.5	Thêm người dùng	110
7.6.6	Thêm khóa học	111
8	Kiểm thử hệ thống	112
8.0.1	Tổng quan về kiểm thử	112
8.0.2	Mô tả chung về các kịch bản kiểm thử	112
8.0.2.1	Cấu trúc của một kịch bản kiểm thử	113
8.0.2.2	Ví dụ về một kịch bản kiểm thử trong bảng Excel	113
8.0.2.3	Mục tiêu của các kịch bản kiểm thử	113
8.1	Các API cần kiểm thử	114
8.2	Kết quả kiểm thử API	114
8.2.1	Kiểm thử phương thức	114
8.2.2	Kết quả tổng hợp của các testcase	114
8.3	Hạn chế	114
9	Đánh giá hệ thống	115
9.1	Đánh giá tính năng và hiệu quả của hệ thống	115
9.2	Đánh giá hệ thống với các mô hình ngôn ngữ hiện có	117
9.3	Đánh giá các hệ thống liên quan	123
9.4	Các vấn đề và khó khăn trong quá trình phát triển	124
9.5	Kết chương	124
10	Tổng kết	125
10.1	Kết quả đạt được	125
10.2	Hướng phát triển	125

Danh mục hình ảnh

5.1	Activity diagram	44
5.2	Use case diagram	45
5.3	Detailed use case with id 1: Course Management	46
5.4	Detailed use case with id 2: User Management	47
5.5	Detailed use case with id 3: Adaptive Learning Path Management of Learner Actor	48
5.6	Detailed use case with id 3: Learning Resource Recommendation System of System Actor	48
5.7	Detailed use case with id 4: Progress Tracking of Learner Actor	49
5.8	Detailed use case with id 4: Progress Tracking of System Actor	49
5.9	Detailed use case with id 5: Exercise Generator	51
5.10	Detailed use case with id 6: AI Tutor of Learner Actor	52
5.11	Detailed use case with id 6: AI Tutor of System Actor	52
5.12	Component-based Diagram cho toàn bộ hệ thống	58
5.13	Component-based Diagram cho những công nghệ được chọn	58
5.14	Sơ đồ EER của hệ thống	59
5.15	Lược đồ CSDL quan hệ ánh xạ của hệ thống	61
5.16	Class diagram hiện thực của hệ thống	62
5.17	Sitemap cho đối tượng Sinh viên	63
5.18	Sitemap cho đối tượng Giảng viên	65
5.19	Sitemap cho đối tượng Quản trị viên	66
6.1	Cấu trúc Frontend	74
6.2	Hệ thống Typography sử dụng trong ứng dụng	75
6.3	Bảng màu được sử dụng trong ứng dụng	76



6.4	Cấu trúc Backend	77
6.5	Cấu trúc Cơ sở dữ liệu	79
6.6	Flow Định danh tài khoản	82
6.7	Sơ đồ ERD của hệ thống	97
7.1	Giao diện trang chủ của hệ thống	100
7.2	Giao diện đăng nhập của hệ thống	101
7.3	Giao diện thông tin cá nhân của sinh viên	101
7.4	Giao diện chính của sinh viên	102
7.5	Giao diện quản lý khóa học của sinh viên	102
7.6	Giao diện chi tiết khóa học của sinh viên	103
7.7	Giao diện chi tiết bài học của sinh viên	103
7.8	Danh sách các modules học tập của bài học	104
7.9	Giao diện chi tiết module của sinh viên	104
7.10	Giao diện chi tiết bài quiz của sinh viên	105
7.11	Danh sách các bài quiz của sinh viên	105
7.12	Dashboard Professor	106
7.13	Modal Feedback	106
7.14	Professor: My Course	107
7.15	Professor: Feedback Management	107
7.16	Professor: Progress Student	108
7.17	Giao diện chính của quản trị viên	108
7.18	Giao diện quản lý khóa học của quản trị viên	109
7.19	Giao diện quản lý người dùng của quản trị viên	109
7.20	Giao diện quản lý phản hồi của quản trị viên	110
7.21	Giao diện thêm người dùng của quản trị viên	110
7.22	Giao diện thêm khóa học của quản trị viên	111
8.1	Ví dụ về một kịch bản kiểm thử	113

Danh mục bảng biểu

2.1	So sánh giữa LangChain và LlamaIndex	23
2.2	So sánh giữa React, Vue và Angular	26
2.3	So sánh giữa FastAPI và Django	27
9.1	Đánh giá các hệ thống liên quan	123

Chương 1

Giới thiệu về đề tài

1.1 Động lực

Trong bối cảnh toàn cầu hóa và sự phát triển không ngừng của công nghệ, việc nâng cao chất lượng giáo dục, đặc biệt trong lĩnh vực lập trình, ngày càng trở nên quan trọng. Một trong những thách thức lớn hiện nay là làm thế nào để cá nhân hóa quá trình học lập trình, đáp ứng sự đa dạng trong nhu cầu và khả năng của từng học viên. Các hệ thống giáo dục truyền thống thường gặp khó khăn trong việc tạo ra lộ trình học phù hợp với từng học viên, đặc biệt là trong lập trình, nơi mà khả năng tư duy và phong cách học của mỗi học viên đều khác nhau.

Sự phát triển của công nghệ trí tuệ nhân tạo (AI) và mô hình ngôn ngữ lớn (LLM) đang mở ra cơ hội lớn để xây dựng các hệ thống dạy học lập trình hiệu quả hơn. LLM, với khả năng hiểu và sinh ngôn ngữ tự nhiên, không chỉ giúp học viên giải quyết các vấn đề lập trình nhanh chóng mà còn hỗ trợ cá nhân hóa học tập. LLM có thể cung cấp tài liệu học tập và bài giảng được tùy chỉnh, cũng như hỗ trợ giải quyết bài tập lập trình, từ đó cải thiện hiệu quả học tập.

Tuy nhiên, sự phát triển mạnh mẽ của AI cũng gây ra mối lo ngại về việc sinh viên sử dụng AI để làm bài tập mà không thực sự học hỏi. Điều này đặt ra câu hỏi liệu AI có thể giúp sinh viên tự tìm ra lời giải cho các vấn đề thay vì đưa ra câu trả lời trực tiếp, giúp họ phát triển kỹ năng tư duy giải quyết vấn đề một cách hiệu quả.

Việc chọn giáo dục lập trình làm lĩnh vực nghiên cứu và ứng dụng hệ thống có hai lý do chính:

- Thứ nhất, dạy học lập trình ngày càng trở nên phổ biến trong thời đại internet, giúp học viên phát triển tư duy logic và khả năng giải quyết vấn đề, vì nó yêu cầu phân tích và tư duy hệ thống.
- Thứ hai, lĩnh vực giáo dục lập trình rất gần gũi với ngành khoa học và kỹ thuật máy tính, nơi mà nhóm nghiên cứu đang theo học, tạo ra sự liên kết thực tiễn giữa nghiên cứu và



ứng dụng trong thực tế.

1.2 Mục tiêu

Từ động lực đã nêu, mục tiêu của đề tài này gồm các điểm chính 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.

1.3 Phạm vi

Đề tài này sẽ được thực hiện trong khoảng thời gian 7-8 tháng, bao gồm hai học kỳ tại trường đại học. Phạm vi nghiên cứu chủ yếu tập trung vào lĩnh vực giáo dục công nghệ thông tin, đặc biệt là trong việc giảng dạy lập trình. Cụ thể, hệ thống sẽ được xây dựng để hỗ trợ việc học lập trình cho sinh viên, thông qua việc cá nhân hóa các bài giảng, hướng dẫn và hỗ trợ học tập dựa trên các mô hình ngôn ngữ lớn.

Về mặt công nghệ, nhóm thực hiện đề tài sẽ không xây dựng lại các mô hình ngôn ngữ lớn từ đầu mà sẽ tận dụng các mô hình hiện có, chẳng hạn như GPT, Gemini, hoặc các mô hình tương tự, để khai thác và tùy chỉnh chúng nhằm đáp ứng các yêu cầu cụ thể của hệ thống. Mô hình ngôn ngữ này sẽ được sử dụng để cung cấp các phản hồi tự động, đề xuất bài học, tạo các bài kiểm tra và giúp cá nhân hóa trải nghiệm học lập trình của sinh viên.

Mặc dù hệ thống có thể áp dụng cho các lĩnh vực giáo dục khác, trong phạm vi của đề tài này, mục tiêu chính là tập trung vào việc ứng dụng trong lĩnh vực lập trình và công nghệ thông tin, nhằm hỗ trợ việc giảng dạy và học tập môn học này.



1.4 Ý nghĩa của đề tài

1.4.1 Ý nghĩa của dự án dưới góc nhìn thực tiễn

- Dự án này mang lại ý nghĩa thực tiễn quan trọng trong việc áp dụng công nghệ AI vào giáo dục, cụ thể là trong việc cá nhân hóa quá trình học tập của học viên. Nhờ vào AI và LLM, hệ thống học tập có thể tạo ra một trải nghiệm học tập hấp dẫn và hiệu quả hơn cho học viên, từ đó giúp tăng động lực học tập.
- AI có thể đóng vai trò như một trợ giảng 1-1, giúp giảm tải áp lực cho giáo viên. AI sẽ hỗ trợ trả lời các câu hỏi cơ bản, đánh giá và chấm điểm tự động cho các bài tập, giúp giáo viên có thể tập trung vào việc hỗ trợ học viên ở mức độ sâu hơn. Điều này không chỉ cải thiện hiệu quả giảng dạy mà còn tạo ra một môi trường học tập linh hoạt và phù hợp với từng cá nhân học viên.

1.4.2 Ý nghĩa của dự án theo góc nhìn khoa học

- Dự án này có ý nghĩa khoa học đáng kể trong việc ứng dụng AI và LLM vào giáo dục, đặc biệt là trong giáo dục thông minh. Việc tích hợp LLM sẽ không chỉ mở rộng khả năng của các hệ thống dạy học mà còn đánh giá được hiệu quả của chúng trong việc nâng cao khả năng giải quyết vấn đề và học lập trình.
- Dự án sẽ chứng minh tính ứng dụng của LLM trong việc thực hiện các tác vụ lập trình và giải quyết vấn đề, đồng thời khai thác các lý thuyết giáo dục trong khoa học nhận thức, như lý thuyết tải nhận thức (cognitive load theory) và học thuyết kiến tạo (constructivism). Việc áp dụng LLM vào giáo dục có thể giúp học viên không chỉ tiếp thu kiến thức mà còn phát triển khả năng tự học và tư duy phản biện.

1.5 Cấu trúc báo cáo

Báo cáo này được tổ chức thành 10 chương chính, mỗi chương trình bày một phần quan trọng trong quá trình thực hiện và phát triển hệ thống. Cấu trúc báo cáo như sau:

- Chương 1 – Giới thiệu về đề tài:** Cung cấp cái nhìn tổng quan về động lực, mục tiêu, phạm vi và ý nghĩa của đề tài cả trong thực tiễn và khoa học. Phần này cũng giới thiệu cấu trúc báo cáo để làm rõ cách thức bố trí và tiếp cận các nội dung.
- Chương 2 – Kiến thức nền tảng:** Trình bày các kiến thức cơ bản liên quan đến giáo dục, giáo dục thông minh, mô hình ngôn ngữ, và các công nghệ được sử dụng như LangChain, LlamaIndex, React, FastAPI, cùng các khái niệm kỹ thuật như đồ thị tri thức và knowledge tracing.



- **Chương 3 – Công trình liên quan:** Phân tích các nghiên cứu và hệ thống liên quan, bao gồm hệ thống đề xuất lộ trình học, ứng dụng mô hình ngôn ngữ lớn (LLM), prompt engineering, retrieval-augmentation generation và knowledge tracing, nhằm xác định khoảng trống nghiên cứu và cơ hội cải tiến.
- **Chương 4 – Hệ thống đề xuất:** Mô tả chi tiết hệ thống đề xuất, bao gồm khảo sát yêu cầu, người dùng mục tiêu, các chức năng như quản lý người dùng, đề xuất tài nguyên học tập, tạo bài tập, trợ lý AI, theo dõi tiến độ và quản lý khóa học, cùng với yêu cầu phi chức năng và dữ liệu.
- **Chương 5 – Phân tích và thiết kế:** Trình bày quá trình phân tích và thiết kế hệ thống, bao gồm mô hình hóa quy trình nghiệp vụ, biểu đồ use case, kiến trúc hệ thống, thiết kế cơ sở dữ liệu (EER diagram, lược đồ quan hệ), biểu đồ lớp, sitemap và quy trình đề xuất lộ trình học tập cá nhân hóa.
- **Chương 6 – Hiện thực:** Mô tả quá trình triển khai hệ thống, kiến trúc backend, các chức năng đã thực hiện như quản lý tài khoản, khóa học, bài học, bài tập, lộ trình học tập, theo dõi tiến độ, và các phương pháp như SMART, Rubric-Based Assessment, STAR trong thiết kế mục tiêu và đánh giá.
- **Chương 7 – Giao diện người dùng:** Giới thiệu các giao diện chính của hệ thống, bao gồm landing page, đăng nhập, hồ sơ cá nhân, giao diện cho sinh viên, giảng viên và admin, với các chức năng như quản lý khóa học, bài học, bài kiểm tra và phản hồi.
- **Chương 8 – Kiểm thử hệ thống:** Mô tả quá trình kiểm thử, các kịch bản kiểm thử, API được kiểm tra, kết quả kiểm thử và các hạn chế gặp phải trong quá trình kiểm thử hệ thống.
- **Chương 9 – Đánh giá hệ thống:** Đánh giá tính năng, hiệu quả của hệ thống, so sánh với các mô hình ngôn ngữ hiện có, phân tích các hệ thống liên quan, và thảo luận các vấn đề, khó khăn trong quá trình phát triển.
- **Chương 10 – Tổng kết:** Tóm tắt kết quả đạt được, thảo luận các hạn chế của hệ thống và đề xuất các hướng phát triển trong tương lai.

Chương 2

Kiến thức nền tảng

2.1 Kiến thức nền tảng về giáo dục và giáo dục thông minh

2.1.1 Khái quát về giáo dục

Trong phần này, ta sẽ giới thiệu khái niệm cơ bản về giáo dục. Giáo dục là quá trình truyền đạt và tiếp thu kiến thức, kỹ năng và giá trị, giúp con người phát triển toàn diện về trí tuệ, thể chất và đạo đức. Mục tiêu chính của giáo dục là trang bị cho con người những kiến thức cơ bản cần thiết để hòa nhập vào xã hội, đồng thời phát triển các kỹ năng tư duy, giải quyết vấn đề và sáng tạo.

Bên cạnh đó, giáo dục còn hướng đến việc bồi dưỡng phẩm chất đạo đức, khuyến khích mỗi cá nhân phát triển tiềm năng riêng biệt của mình. Đây chính là nền tảng để mỗi người có thể xây dựng sự nghiệp và cuộc sống thành công.

Trong bối cảnh thế giới không ngừng thay đổi, giáo dục ngày nay không chỉ cung cấp kiến thức mà còn giúp con người thích nghi với những thách thức mới, phát triển khả năng học tập suốt đời và duy trì sự phát triển liên tục qua mọi giai đoạn cuộc sống.

Về lý thuyết giáo dục, có nhiều học thuyết liên quan đến lĩnh vực này, kết hợp từ nhiều lĩnh vực khác nhau như tâm lý học và thần kinh học:

- *Học tập kiến tạo (Constructivism)*: Phương pháp học tập này cho rằng con người tự xây dựng sự hiểu biết và kiến thức của mình thông qua trải nghiệm và việc phản ánh lại các trải nghiệm đó. Thay vì tiếp thu thông tin một cách thụ động, người học chủ động tạo ra ý nghĩa thông qua việc liên kết thông tin mới với kiến thức đã có trước đó.
- *Thuyết tải nhận thức (Cognitive load theory)*: Được phát triển bởi John Sweller vào những năm 1980, lý thuyết này tập trung vào cách bộ não xử lý và lưu trữ thông tin trong quá



trình học tập, đặc biệt là những yếu tố ảnh hưởng đến hiệu quả học tập. Thuyết này nhấn mạnh rằng bộ não con người có giới hạn về lượng thông tin có thể xử lý cùng lúc trong trí nhớ ngắn hạn.

2.1.2 Khái quát về giáo dục thông minh

Tuy không có một định nghĩa chính thức và rõ ràng cho thuật ngữ “**giáo dục thông minh**”, trong báo cáo này, thuật ngữ “**thông minh**” được hiểu là cách thức thực hiện mọi hoạt động một cách hiệu quả và hữu ích. Mục tiêu của giáo dục thông minh là bồi dưỡng con người trên nhiều mặt, từ kiến thức đến đạo đức, với phương thức giúp học sinh tiếp thu kiến thức một cách hiệu quả nhất có thể.

Giáo dục thông minh không chỉ liên quan đến việc dạy và học mà còn phải thích ứng với nhu cầu, khả năng, sở thích và điểm mạnh, yếu của từng học sinh. Việc cá nhân hóa quá trình học tập đóng vai trò quan trọng, vì mỗi học sinh có cách tiếp thu khác nhau. Công nghệ trí tuệ nhân tạo (AI) hiện nay đã tạo ra nhiều cơ hội để phát triển giáo dục thông minh, đặc biệt là qua phương pháp học tập cá nhân hóa (personalized learning), nơi mà mỗi học sinh có thể học theo một tốc độ, phong cách và nội dung phù hợp với bản thân.

Học tập cá nhân hóa nhằm tối đa hóa hiệu quả học tập bằng cách điều chỉnh chương trình học và phương pháp giảng dạy để phù hợp với nhu cầu và mục tiêu riêng biệt của mỗi học sinh, thay vì áp dụng một chương trình chung cho tất cả.

2.2 Mô hình ngôn ngữ

Mô hình ngôn ngữ (*language model*) có thể được hiểu ở nhiều góc độ khác nhau, nhưng nhìn chung, một mô hình ngôn ngữ là một hệ thống được thiết kế để hiểu và tạo ra ngôn ngữ tự nhiên của con người. Trong lĩnh vực machine learning, mô hình ngôn ngữ được định nghĩa rõ ràng hơn là một mô hình xác suất của một ngôn ngữ tự nhiên, có khả năng dự đoán một từ hoặc một chuỗi từ dựa trên những từ đã xuất hiện trước đó.

Trong lịch sử phát triển của lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), đã có nhiều loại mô hình ngôn ngữ khác nhau. Dưới đây là các loại mô hình ngôn ngữ chính:

- **Mô hình ngôn ngữ xác suất:** Là loại mô hình ngôn ngữ là trong đó, mỗi một chuỗi từ ngữ nhất định sẽ được gán một xác suất cụ thể. Bài toán chính mà dạng mô hình này giải quyết đó là tính *khả năng xảy ra* (*likelihood*) của một chuỗi các từ vựng bất kì, câu hỏi đặt ra thường là "Liệu một chuỗi các từ w_1, w_2, \dots, w_n có khả năng xuất hiện là bao nhiêu?". Một ví dụ điển hình cho dạng mô hình ngôn ngữ xác suất đó là mô hình ngôn ngữ n-gram. Theo mô hình ngôn ngữ này, thì xác suất của một từ xuất hiện phụ thuộc vào các từ trước



đó:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_{i-(n-1)}, \dots, w_{i-1})$$

- **Mô hình ngôn ngữ dựa trên Neural network:** là một loại mô hình ngôn ngữ sử dụng neural network để hiểu và sinh ra ngôn ngữ tự nhiên của con người. Sức mạnh của neural network nằm ở chỗ là nó có thể nhận diện được các pattern phức tạp, do đó, mô hình dạng này giúp thực hiện nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP) một cách hiệu quả hơn so với các mô hình thống kê truyền thống đã nói ở trên.
- **Mô hình ngôn ngữ lớn:** Dựa trên khái niệm về mô hình ngôn ngữ, ta có khái niệm cho *Mô hình ngôn ngữ lớn (Large language model)*. Mô hình ngôn ngữ lớn là gì? Nó khác gì so với một mô hình ngôn ngữ? Làm sao để biết được một mô hình ngôn ngữ là 'lớn'? Một mô hình ngôn ngữ lớn là một mô hình ngôn ngữ dựa trên *Deep neural network* và được huấn luyện bằng một tập dữ liệu ngôn ngữ khổng lồ. Việc đánh giá một mô hình là to hay nhỏ có nhiều khía cạnh, có thể là quy mô của mô hình hoặc là khả năng. Về quy mô, mô hình ngôn ngữ lớn, đúng như tên gọi, có kích thước rất lớn, với số lượng tham số lên đến con số hàng tỉ, hoặc hàng trăm tỉ. Một ví dụ tiêu biểu có thể kể đến như mô hình GPT-3 của OpenAI có khoảng 175 tỉ tham số [1]. Một mô hình ngôn ngữ lớn có thể thực hiện được nhiều tác vụ hơn, liên quan đến suy luận, giải quyết vấn đề, những thứ mà các mô hình ngôn ngữ truyền thống gần như không thể thực hiện được.

2.3 Các thuật ngữ kỹ thuật liên quan đến giáo dục thông minh

2.3.1 Đồ thị tri thức (Knowledge graph)

Trước khi nói về thuật ngữ *Đồ thị tri thức*, ta hãy nhìn sơ qua về thuật ngữ *Đồ thị*. Ở đây, nghĩa của từ đồ thị được hiểu theo định nghĩa của nó ở trong *Lý thuyết đồ thị (graph theory)* là một nhánh nghiên cứu của *Toán rời rạc (Discrete mathematics)*: Một đồ thị (graph) là một cấu trúc toán học được sử dụng để mô tả mối quan hệ giữa các đối tượng (được gọi là đỉnh hoặc nút). Các đối tượng này được kết nối với nhau thông qua các cạnh (hay còn gọi là cung) biểu diễn mối quan hệ giữa chúng. Từ định nghĩa của đồ thị, Trong bài báo cáo này, nhóm xin đưa ra định nghĩa của *Đồ thị tri thức (knowledge graph)*: là một dạng thể hiện có cấu trúc, cụ thể hơn là thể hiện bằng đồ thị, trong đó các đỉnh là các thực thể trong thế giới thực như: một đối tượng, một khái niệm, một sự kiện, và các cạnh là mối quan hệ giữa chúng[2].



2.3.2 Knowledge tracing

Knowledge tracing là bài toán mô hình hóa kiến thức của học sinh theo thời gian để ta có thể dự đoán chính xác cách học sinh sẽ thực hiện trong các tương tác sau này [3]. Nhờ có *Knowledge tracing*, giảng viên/người giảng dạy/hệ thống có thể kiểm tra được tiến độ học tập cũng như năng lực của mỗi học sinh *một cách hiệu quả hơn* bằng cách theo dõi tương tác của học viên với các học liệu trực tuyến (tài liệu, các câu hỏi, bài kiểm tra,...)[4].



2.4 Kiến trúc công nghệ

2.4.1 Công cụ xử lý ngôn ngữ tự nhiên - Framework tích hợp LLMs

Trong quá trình phát triển một hệ thống học tập trực tuyến thông minh dựa trên mô hình ngôn ngữ lớn (LLMs), việc lựa chọn công cụ xử lý ngôn ngữ tự nhiên phù hợp là một yếu tố quyết định đến hiệu quả và tính linh hoạt của hệ thống. Với mục tiêu tối ưu hóa trải nghiệm học tập cá nhân hóa, hệ thống cần tích hợp một framework mạnh mẽ để khai thác và xử lý dữ liệu từ nhiều nguồn. Hai công cụ nổi bật trong lĩnh vực này là LangChain và LlamaIndex, đều được thiết kế để hỗ trợ xây dựng các ứng dụng sử dụng LLMs. Báo cáo này sẽ phân tích và so sánh hai framework trên, nhằm xác định công cụ phù hợp nhất cho hệ thống học tập thông minh, dựa trên tính năng, khả năng tích hợp, và yếu tố chi phí.

2.4.1.1 LangChain

Langchain là một thư viện Python được thiết kế để hỗ trợ phát triển các ứng dụng dựa trên mô hình ngôn ngữ lớn (LLMs).

Các tính năng chính:

- Tích hợp dễ dàng với nhiều LLMs
 - Dễ dàng tích hợp và chuyển đổi giữa các mô hình ngôn ngữ lớn khác nhau như **GPT-3**, **GPT-4**, **BERT**, hoặc các mô hình tùy chỉnh.
 - * VD: Sử dụng **GPT-4** cho việc tạo nội dung học tập phức tạp, trong khi sử dụng một mô hình nhẹ hơn như **BERT** cho việc phân tích nhanh câu trả lời của học viên
 - Chuỗi xử lý linh hoạt cho các tác vụ phức tạp
 - * Cho phép xây dựng các chuỗi xử lý (chains) phức tạp, kết hợp nhiều bước xử lý khác nhau.
 - * VD: Trong hệ thống gia sư AI, tạo một chuỗi xử lý để phân tích code của học viên, đánh giá, và đưa ra gợi ý cải thiện.
 - Hỗ trợ truy xuất thông tin và tìm kiếm ngữ nghĩa
 - Cung cấp các công cụ để truy xuất thông tin dựa trên ngữ nghĩa, giúp tìm kiếm thông tin liên quan một cách hiệu quả
 - * VD: Khi học viên đặt câu hỏi, hệ thống có thể tìm kiếm thông tin liên quan từ cơ sở dữ liệu bài giảng bằng cách dùng **FAISS (Facebook AI Similarity Search)** - Một thư viện hỗ trợ sử dụng vector search để tìm kiếm thông tin trong **Knowledge Base**.



- Khả năng xử lý đa dạng các loại dữ liệu
 - Có thể làm việc với nhiều loại dữ liệu khác nhau như văn bản, PDF, hình ảnh (qua tích hợp với các mô hình xử lý hình ảnh).
 - * VD: Xử lý các tài liệu PDF và chuyển đổi chúng thành nội dung học tập.

2.4.1.2 LlamaIndex

LlamaIndex là một framework chuyên biệt cho việc xây dựng các ứng dụng AI với khả năng truy xuất dữ liệu mạnh mẽ.

Các tính năng chính:

- Indexing và truy xuất dữ liệu hiệu quả
 - Chuyên về việc tạo index cho dữ liệu lớn và truy xuất thông tin một cách nhanh chóng và hiệu quả.
 - * VD: Indexing toàn bộ nội dung khóa học và truy xuất thông tin liên quan.
 - Quá trình Indexing trong LlamaIndex như sau:
 - * Data Connectors: Đây là các công cụ nhập dữ liệu từ nhiều nguồn khác nhau như API, PDF, cơ sở dữ liệu, hay các ứng dụng bên ngoài (Gmail, Notion, Airtable). Chúng chịu trách nhiệm thu thập dữ liệu và đưa vào hệ thống dưới dạng tài liệu thống nhất.
 - * Documents/Nodes: Documents là các container chứa dữ liệu từ các nguồn khác nhau, chẳng hạn như từ PDF, API hoặc cơ sở dữ liệu. Nodes là những phần nhỏ của tài liệu, được bổ sung metadata và các mối quan hệ, giúp tăng độ chính xác trong quá trình truy xuất dữ liệu.
 - * Data Indexes: Sau khi dữ liệu được nhập vào, LlamaIndex sẽ giúp tổ chức dữ liệu thành một định dạng có thể truy xuất. Quá trình này bao gồm việc phân tích, tạo các biểu diễn embedding, và suy luận metadata, tạo thành kho kiến thức để sử dụng sau này.
- Tích hợp với nhiều nguồn dữ liệu
 - LlamaIndex có thể làm việc với nhiều nguồn dữ liệu khác nhau như file local, API, cơ sở dữ liệu.
 - * VD: Tích hợp dữ liệu từ nhiều nguồn để tạo nội dung khóa học.
- Hỗ trợ cho các truy vấn phức tạp



- LlamaIndex cho phép thực hiện các truy vấn phức tạp, kết hợp nhiều điều kiện và lọc kết quả.
 - * VD: Tìm kiếm bài học phù hợp dựa trên nhiều tiêu chí.

2.4.1.3 Tổng quan

Tiêu chí	LangChain	LlamaIndex
Overall	Phát triển ứng dụng phức tạp linh hoạt, tích hợp LLM	Tập trung vào tìm kiếm và truy xuất thông tin từ tập dữ liệu lớn một cách nhanh chóng và chính xác
Prompts	Hỗ trợ giao diện tiêu chuẩn cho việc tạo và quản lý prompts, giúp tùy chỉnh và sử dụng lại dễ dàng hơn trên các mô hình khác	Không hỗ trợ chi tiết
Chains	Cung cấp giao diện mạnh mẽ để xây dựng và quản lý chuỗi, cùng với nhiều thành phần có thể tái sử dụng	Không hỗ trợ chuỗi xử lý phức tạp
Agents	Sử dụng LLM để xác định và thực hiện các hành động dựa trên input	Không có cơ chế agent
Data Indexing	Phương pháp indexing qua các chuỗi phức tạp	Lập chỉ mục nhanh chóng các dữ liệu không cấu trúc
Customization	Tùy chỉnh cao cho các workflow và ứng dụng phức tạp	Tùy chỉnh hạn chế, tập trung vào lập chỉ mục và tìm kiếm
Context Retention	Lưu giữ thông tin từ các tương tác trước đó để cho phép các cuộc hội thoại có nhận thức về ngữ cảnh và mạch lạc	Lưu giữ ngữ cảnh cơ bản, phù hợp với nhiệm vụ tìm kiếm
Use Cases	Phù hợp cho các ứng dụng như chatbot, hỗ trợ khách hàng, tạo nội dung phức tạp	Phù hợp cho hệ thống tìm kiếm nội bộ, quản lý kiến thức
Performance	Xử lý tốt các cấu trúc dữ liệu phức tạp	Tối ưu cho tốc độ và độ chính xác trong truy xuất thông tin
Lifecycle Management	Cung cấp bộ đánh giá LangSmith để kiểm tra và gỡ lỗi ứng dụng LLM	Tích hợp công cụ gỡ lỗi và giám sát cho hiệu suất và độ tin cậy

Table 2.1: So sánh giữa LangChain và LlamaIndex

Với những ưu điểm trên, **LangChain** là một lựa chọn tối ưu cho hệ thống vì nó hỗ trợ mạnh mẽ về việc quản lý prompts – một yếu tố quan trọng để tạo ra các hướng dẫn cho các mô hình ngôn ngữ lớn (LLM), cho phép tùy chỉnh và tối ưu chức năng Adaptive Learning Path Management. Ngoài ra, tính năng memory của LangChain giúp lưu giữ và quản lý ngữ cảnh của các cuộc hội thoại trước đó, giúp các tương tác trở nên liền mạch và cá nhân hóa hơn.



2.4.2 Framework xây dựng giao diện người dùng

Framework xây dựng giao diện người dùng (UI Framework) giúp phát triển các ứng dụng web với giao diện người dùng linh hoạt và dễ dàng tương tác. Việc lựa chọn framework xây dựng giao diện người dùng (UI) là một quyết định quan trọng đối với các ứng dụng web hiện đại. React, Vue, và Angular là ba framework phổ biến nhất hiện nay, mỗi framework có điểm mạnh và hạn chế riêng. Báo cáo này sẽ so sánh ba framework này dựa trên các tiêu chí như tính dễ học, hiệu suất, khả năng mở rộng và sự linh hoạt trong phát triển.

2.4.2.1 React

React là một thư viện JavaScript được phát triển bởi Facebook, chuyên về việc xây dựng UI dựa trên các component tái sử dụng. React nổi bật với khả năng làm việc nhanh và hiệu quả nhờ vào Virtual DOM và hỗ trợ mạnh mẽ từ cộng đồng.

Các tính năng chính:

- Virtual DOM giúp tối ưu hiệu suất render và cập nhật giao diện.
- Hệ sinh thái rộng lớn với các thư viện hỗ trợ như React Router, Redux, và React Query.
- Được tối ưu hóa cho các ứng dụng quy mô lớn với khả năng mở rộng mạnh mẽ.
- Yêu cầu kiến thức về JSX (JavaScript XML) và quản lý trạng thái với Redux hoặc Context API.

2.4.2.2 Vue

Vue là một framework JavaScript nhẹ, dễ học, và được thiết kế để xây dựng các ứng dụng web động. Vue đặc biệt phù hợp với các nhà phát triển mới, với cú pháp đơn giản và dễ sử dụng, nhưng vẫn mạnh mẽ cho các ứng dụng phức tạp.

Các tính năng chính:

- Cung cấp Vue CLI giúp khởi tạo dự án và xây dựng ứng dụng nhanh chóng.
- Hệ thống component mạnh mẽ và dễ hiểu.
- Vuex cho quản lý trạng thái đơn giản và dễ sử dụng.
- Tích hợp dễ dàng với các dự án hiện tại và khả năng mở rộng cho các ứng dụng lớn.
- Vue Router giúp quản lý routing trong các ứng dụng một cách dễ dàng.



2.4.2.3 Angular

Angular là một framework full-stack được phát triển bởi Google, được biết đến với khả năng xây dựng các ứng dụng web phức tạp và quy mô lớn. Angular đi kèm với tất cả các công cụ cần thiết để xây dựng một ứng dụng đầy đủ chức năng.

Các tính năng chính:

- Cung cấp một framework toàn diện, bao gồm công cụ để quản lý routing, form, HTTP requests, và các tính năng bảo mật.
- Tích hợp TypeScript giúp nâng cao tính an toàn của mã nguồn.
- Hệ thống dependency injection và mô hình MVC (Model-View-Controller) giúp tổ chức mã nguồn và quản lý các phần tử trong ứng dụng.
- Phức tạp và yêu cầu học hỏi nhiều hơn so với React và Vue, đặc biệt là khi sử dụng RxJS và các concepts như modules, services và directives.

2.4.2.4 Tổng quan

Tiêu chí	React	Vue	Angular
Dễ học	Dễ học với JSX và các concept như state và lifecycle nhưng yêu cầu làm quen với các thư viện như Redux	Dễ học với cú pháp đơn giản, dễ tiếp cận và tài liệu phong phú	Phức tạp hơn, yêu cầu học các khái niệm như Dependency Injection, RxJS và TypeScript
Hiệu suất	Rất tốt nhờ Virtual DOM và khả năng render nhanh chóng	Cũng sử dụng Virtual DOM nhưng nhẹ và nhanh hơn trong các ứng dụng nhỏ và trung bình	Hiệu suất ổn, nhưng thường chậm hơn trong các ứng dụng lớn do tính phức tạp của framework
Quản lý trạng thái	Quản lý trạng thái mạnh mẽ nhưng phức tạp với Redux hoặc Context API	Quản lý trạng thái đơn giản với Vuex, dễ sử dụng cho cả ứng dụng nhỏ và lớn	Quản lý trạng thái mạnh mẽ với NgRx, nhưng phức tạp hơn Vuex và Redux
Khả năng mở rộng	Phù hợp cho các ứng dụng quy mô lớn và có thể mở rộng dễ dàng nhờ vào hệ sinh thái thư viện mạnh mẽ	Rất linh hoạt và có thể mở rộng cho cả các ứng dụng nhỏ và lớn mà không gặp phải sự phức tạp như Angular	Tốt cho các ứng dụng quy mô lớn với tính cấu trúc chặt chẽ và tổ chức mã nguồn rõ ràng
Cộng đồng	Cộng đồng lớn với nhiều tài nguyên và thư viện hỗ trợ	Cộng đồng đang phát triển, với tài liệu dễ hiểu và hỗ trợ từ cộng đồng ngày càng mạnh mẽ	Cộng đồng lớn, nhưng tài liệu có thể phức tạp và khó tiếp cận hơn đối với người mới bắt đầu
Tính linh hoạt	Linh hoạt cao, nhưng yêu cầu sử dụng các thư viện bổ sung cho các tính năng như routing và quản lý trạng thái	Linh hoạt và dễ dàng tích hợp vào các dự án hiện tại, cung cấp tính năng đầy đủ mà không cần thư viện bên ngoài	Tính linh hoạt thấp hơn so với React và Vue, yêu cầu tuân theo một cấu trúc chặt chẽ và sử dụng các công cụ riêng biệt

Table 2.2: So sánh giữa React, Vue và Angular

Với cú pháp dễ sử dụng, khả năng mở rộng tốt và sự dễ dàng trong việc tích hợp vào dự án hiện tại, nhóm quyết định chọn Vue.js làm framework xây dựng giao diện người dùng cho hệ thống học tập thông minh.

2.4.3 Framework phát triển API

Framework phát triển API là công cụ quan trọng trong việc xây dựng các hệ thống web hiện đại. Việc lựa chọn framework phát triển API là rất quan trọng trong các ứng dụng web và dịch vụ backend. FastAPI và Django đều có những tính năng mạnh mẽ và phù hợp cho việc phát triển API, nhưng mỗi công cụ lại có những ưu điểm riêng. Báo cáo này sẽ so sánh FastAPI và Django dựa trên hiệu suất, tính dễ sử dụng, và khả năng mở rộng.



2.4.3.1 FastAPI

FastAPI là một framework web nhanh chóng được phát triển bằng Python, được tối ưu hóa cho việc xây dựng các API với tính bảo mật cao và dễ dàng mở rộng.

Các tính năng chính:

- Tốc độ cao nhờ vào Starlette và Pydantic
- Tự động tạo tài liệu API với OpenAPI
- Hỗ trợ đầy đủ các chuẩn như HTTP, WebSockets, GraphQL, và OAuth2

2.4.3.2 Django

Django là một framework web Python nổi tiếng, cung cấp đầy đủ các tính năng để phát triển ứng dụng web bao gồm API, hệ quản trị cơ sở dữ liệu và bảo mật.

Các tính năng chính:

- Cung cấp các công cụ tích hợp như Django REST Framework (DRF) để xây dựng API
- Quản lý cơ sở dữ liệu mạnh mẽ với ORM tích hợp sẵn
- Hệ sinh thái lớn với nhiều thư viện hỗ trợ

2.4.3.3 Tổng quan

Tiêu chí	FastAPI	Django
Hiệu suất	Rất nhanh nhờ vào Starlette và Pydantic	Hiệu suất tốt nhưng không thể so sánh với FastAPI trong các ứng dụng quy mô lớn
Dễ sử dụng	Dễ học, đặc biệt cho người đã quen Python	Được hỗ trợ tốt với tài liệu phong phú và cộng đồng lớn
Tính bảo mật	Tích hợp đầy đủ các công cụ bảo mật hiện đại như OAuth2, JWT	Cung cấp các công cụ bảo mật mạnh mẽ nhưng đòi hỏi cấu hình thêm
Khả năng mở rộng	Rất dễ mở rộng và thích hợp cho các ứng dụng hiện đại	Hỗ trợ mở rộng tốt với kiến trúc có sẵn

Table 2.3: So sánh giữa FastAPI và Django

Với hiệu suất cao, tính năng tự động tạo tài liệu API và khả năng dễ dàng tích hợp, nhóm quyết định chọn FastAPI làm framework phát triển API cho hệ thống học tập thông minh.



2.5 Tổng kết

Chương 2 đã cung cấp một cái nhìn tổng quan về các khái niệm và lý thuyết nền tảng trong lĩnh vực giáo dục và giáo dục thông minh. Chúng ta đã tìm hiểu về khái niệm giáo dục, vai trò quan trọng của nó trong việc phát triển toàn diện con người và sự thích nghi với môi trường xã hội. Đồng thời, khái niệm giáo dục thông minh đã được làm rõ, nhấn mạnh vào sự kết hợp giữa công nghệ và các phương pháp học tập cá nhân hóa, nhằm nâng cao hiệu quả giảng dạy và học tập.

Các khái niệm về mô hình ngôn ngữ, từ mô hình ngôn ngữ xác suất đến các mô hình ngôn ngữ lớn (LLMs), đã được trình bày chi tiết, làm nền tảng để hiểu rõ hơn về công nghệ ứng dụng trong giáo dục thông minh. Đặc biệt, những thuật ngữ quan trọng như đồ thị tri thức (Knowledge Graph), Knowledge Tracing, và các công nghệ hỗ trợ trong giáo dục như LangChain, LlamaIndex, Neo4J, và VueJS đã được giới thiệu, nhằm làm rõ các công cụ và nền tảng công nghệ cần thiết để phát triển hệ thống học tập thông minh hiện đại.

Kết thúc chương này, ta có thể thấy rằng việc áp dụng công nghệ vào giáo dục không chỉ là một xu hướng mà còn là một yêu cầu thiết yếu để nâng cao chất lượng và hiệu quả học tập. Trong chương tiếp theo, chúng ta sẽ đi sâu vào việc thiết kế và triển khai hệ thống giáo dục thông minh, đồng thời đánh giá các yếu tố ảnh hưởng đến sự thành công của hệ thống này.

Chương 3

Công trình liên quan

3.1 Hệ thống đề xuất lộ trình học

Trong những năm qua, đã có rất nhiều nghiên cứu và phát triển liên quan đến các hệ thống đề xuất lộ trình học tập. Các hệ thống này không chỉ nhằm cá nhân hóa trải nghiệm học tập mà còn tối ưu hóa quá trình học tập dựa trên dữ liệu từ người học và tài nguyên giáo dục. Từ việc xây dựng đồ thị tri thức để tổ chức các khái niệm học tập, cho đến ứng dụng các mô hình học máy tiên tiến, các công trình này đã đặt nền móng cho việc nâng cao hiệu quả và tính thích nghi của giáo dục thông minh.

- Hệ thống ở bài [5] sử dụng *Đồ thị tri thức* để biểu diễn mối liên hệ giữa các khái niệm (concepts) và các *kết quả học tập* (*learning outcome*). Dựa trên đồ thị tri thức, hệ thống sẽ dùng một giải thuật sinh đường để tạo sinh một lộ trình học tập phù hợp dựa trên đầu vào là *chủ đề bắt đầu* và *chủ đề mục tiêu* của học sinh. Lộ trình học cũng có thể được thay đổi dựa trên năng lực, điểm số,... (Đây là các tham số động - *dynamic parameters*) của học sinh. Đặc biệt, hệ thống liên tục cập nhật độ khó của các tài nguyên học tập (*learning objects – LOs*) cũng như khả năng học tập của học sinh thông qua dữ liệu được thu thập thời gian thực. Các thông số như thời gian học, điểm số đạt được, và số lần thử nghiệm (*number of attempts*) đều được phân tích để điều chỉnh lộ trình sao cho phù hợp nhất với tiến độ học của từng cá nhân. Ngoài ra, hệ thống còn áp dụng các kỹ thuật gợi ý để lựa chọn các tài liệu học tập có mức độ tương thích cao với năng lực hiện tại của học sinh, đảm bảo rằng các tài liệu được gợi ý không chỉ phù hợp về mặt nội dung mà còn tối ưu hóa trải nghiệm học tập.
- Hệ thống FOKE [6] hiện thực một tính năng đề xuất lộ trình học tập cá nhân hóa, tận dụng sức mạnh của các mô hình ngôn ngữ lớn, đồ thị tri thức, và kỹ thuật prompt engineering. Đầu vào của hệ thống bao gồm domain knowledge base, hồ sơ người học, và đặc tả nhiệm



vụ học tập (learning task specification). Các dữ liệu này được xử lý thông qua nhiều mô-đun để tạo ra các gợi ý học tập chính xác, tương tác và có khả năng giải thích. Thành phần cốt lõi của hệ thống là *Rừng tri thức* (*Knowledge Forest - KF*), một phiên bản mở rộng của đồ thị tri thức. Rừng tri thức đại diện cho tri thức miền dưới dạng các cây tri thức (knowledge trees), trong đó mỗi cây thể hiện cấu trúc phân cấp của các khái niệm và mối quan hệ tương ứng. Hồ sơ người học được xây dựng theo cơ chế lập hồ sơ đa chiều (multi-dimensional user profiling), mô hình hóa các đặc điểm tĩnh (thông tin cá nhân, phong cách học tập), hành vi (hoạt động học tập, hiệu suất), và hành trình thời gian (chuỗi kỹ năng đạt được). Dựa trên hai thành phần trên, FOKE kết hợp kỹ thuật *graph embedding* và các gợi ý cấu trúc (structured prompts) để tạo ra các đề xuất lộ trình học tập cá nhân hóa. Đầu ra của hệ thống bao gồm các khuyến nghị về trình tự học tập hoặc tài nguyên học tập, chẳng hạn như khóa học cần hoàn thành, các kỹ năng cần trau dồi, và các tài liệu hỗ trợ.

- Bài báo [7] giới thiệu một phương pháp mới để lập kế hoạch lộ trình học tập cá nhân hóa (*Personalized Learning Path Planning - PLPP*) bằng cách tích hợp các mô hình ngôn ngữ lớn và kỹ thuật thiết kế prompt engineering. Mục tiêu chính của nghiên cứu là cải thiện tính cá nhân hóa, sự thích nghi và tính minh bạch của các hệ thống giáo dục thông minh. Phương pháp đề xuất sử dụng các gợi ý được thiết kế đặc biệt để tích hợp thông tin cụ thể về người học, bao gồm kiến thức nền tảng, mục tiêu học tập và sở thích cá nhân. Các gợi ý này hướng dẫn LLMs, chẳng hạn như GPT-4 và LLama-2-70B, tạo ra các lộ trình học tập cá nhân hóa, phù hợp với nhu cầu và năng lực của từng người học. Ngoài ra, hệ thống hỗ trợ hội thoại đa lượt, cho phép mô hình đặt câu hỏi làm rõ và nhận phản hồi từ người học để chỉnh sửa các đề xuất. Ví dụ, một gợi ý có thể bao gồm: "Dựa trên hiểu biết hiện tại của học sinh về [Chủ đề], đề xuất ba khái niệm tiếp theo mà họ nên học để đạt được trình độ thành thạo trong [Môn học]. Đầu vào của hệ thống bao gồm hồ sơ người học, trạng thái kiến thức hiện tại, và mục tiêu học tập. Đầu ra là các lộ trình học tập được cá nhân hóa, có cấu trúc rõ ràng và được hỗ trợ bởi các giải thích chi tiết về thứ tự học tập được đề xuất. Ví dụ, hệ thống có thể cung cấp lời giải thích như: "Việc học khái niệm [A] trước khái niệm [B] là cần thiết để đảm bảo nền tảng vững chắc trong [Môn học]."

3.2 Ứng dụng LLM vào việc giảng dạy khoa học máy tính, lập trình,...

Trong lĩnh vực khoa học máy tính và lập trình, nơi yêu cầu người học không chỉ nắm vững lý thuyết mà còn thành thạo các kỹ năng thực hành, việc tích hợp LLMs đã mang lại nhiều lợi ích nổi bật. Các công cụ dựa trên LLMs không chỉ giúp người học giải quyết các vấn đề kỹ thuật mà còn cung cấp các phản hồi chi tiết, nâng cao khả năng tự học và giải quyết vấn đề. Một số



công trình liên quan đến việc ứng dụng LLM vào việc giảng dạy lập trình có thể kể đến:

- Trong bài báo [8], nhóm nghiên cứu đã tích hợp AI vào khóa học CS50 tại Đại học Harvard với mục tiêu tái hiện tỷ lệ hướng dẫn lý tưởng 1:1 giữa giảng viên và sinh viên. Hệ thống, được gọi là CS50.ai, cung cấp các tính năng hỗ trợ đa dạng như:
 - Giải thích mã nguồn: Sinh viên có thể làm nổi bật một đoạn mã và nhận được giải thích chi tiết bằng ngôn ngữ tự nhiên.
 - Cải thiện phong cách mã nguồn: Công cụ style50 gợi ý các cải tiến phong cách mã lập trình dựa trên hướng dẫn phong cách của khóa học.
 - Trợ lý học tập AI ("CS50 Duck"): Một chatbot tương tác dựa trên GPT-4, hỗ trợ trả lời các câu hỏi học thuật và hành chính liên quan đến khóa học.

Ngoài ra, Hệ thống được thiết kế với "*pedagogical guardrails*" (đường rào sư phạm), đảm bảo rằng AI hướng dẫn thay vì cung cấp trực tiếp đáp án. Kết quả thực nghiệm cho thấy hệ thống tăng cường đáng kể khả năng tự học của sinh viên, đồng thời giảm tải cho giảng viên trong các lớp học lớn và khóa học trực tuyến quy mô lớn (MOOCs).

- Trong bài báo [9], nhóm nghiên cứu triển khai một hệ thống đánh giá lập trình tự động (Automated Programming Assessment System - APAS) Artemis, tích hợp mô hình AI-Tutor dựa trên GPT-3.5 Turbo. Hệ thống được thiết kế để hỗ trợ sinh viên lập trình bằng cách cung cấp phản hồi tức thời và hướng dẫn cải thiện mã nguồn. Tính năng nổi bật của hệ thống bao gồm:
 - Phản hồi theo thời gian thực: Sinh viên có thể nhận phản hồi chi tiết về các lỗi logic, cú pháp và hiệu suất của mã nguồn.
 - Gợi ý cải tiến mã nguồn: AI-Tutor cung cấp các lời khuyên để cải thiện chất lượng mã nguồn, như quản lý bộ nhớ tốt hơn hoặc tối ưu hóa thuật toán.
 - Không tiết lộ đáp án: Hệ thống chỉ cung cấp hướng dẫn, tránh việc tiết lộ giải pháp trực tiếp để đảm bảo người học tự khám phá.

Đầu vào của hệ thống bao gồm bài tập lập trình, mã nguồn hiện tại của sinh viên, và một giải pháp mẫu do giảng viên cung cấp. Đầu ra là các phản hồi chi tiết từ AI về chất lượng mã nguồn của sinh viên và các gợi ý để cải thiện. Nghiên cứu chỉ ra rằng AI-Tutor không chỉ cải thiện chất lượng mã nguồn mà còn giúp sinh viên phát triển tư duy thuật toán. Tuy nhiên, nhóm tác giả cũng lưu ý rằng phản hồi đôi khi còn chung chung và cần cải thiện giao diện để hỗ trợ tương tác tốt hơn.



3.3 Prompt engineering

Prompt engineering là một hướng đi khá mới do những bước phát triển lớn gần đây của mô hình ngôn ngữ nói chung và mô hình ngôn ngữ lớn nói riêng. Về khái niệm, *Prompt engineering* bao gồm việc thiết kế một cách có chiến lược các hướng dẫn cụ thể cho nhiệm vụ, được gọi là *prompts*, để hướng dẫn đầu ra của mô hình mà không thay đổi các tham số[10]. Một số kĩ thuật tiêu biểu trong prompt engineering có thể kể đến như sau[10]:

- Zero-shot prompting: Mô hình được nhận một tả của công việc mà người dùng muốn nó thực hiện trong prompt nhưng không có dữ liệu được gán nhãn (labeled) để huấn luyện. Sau đó, mô hình tận dụng kiến thức sẵn có của nó để tạo ra các dự đoán dựa trên prompt đã cung cấp để thực hiện tác vụ.
- Few-shot prompting: Khác với Zero-shot prompting, để thực hiện tác vụ, ngoài đặc tả về công việc sẽ được thực hiện, mô hình còn nhận được một số ví dụ cụ thể về công việc mà người dùng muốn mô hình sẽ làm.
- Chain-of-thought prompting[11]: Kĩ thuật này được sử dụng nhằm cải thiện khả năng suy luận của mô hình. Thay vì chỉ đưa ra một câu trả lời trực tiếp, người dùng đưa cho mô hình cung cấp một chuỗi các bước suy nghĩ logic và có hệ thống trước khi đưa ra đáp án cuối cùng.

3.4 Retrieval-Augmentation Generation

Một trong những vấn đề của LLM đó là chúng có khả năng trả về những thông tin sai lệch, và Retrieval-Augmentation Generation (hay RAG) là kĩ thuật để hạn chế vấn đề đó, khi LLM sẽ dựa trên một nguồn dữ liệu gọi là *knowledge base* để đưa ra câu trả lời. Có ba giai đoạn chính trong một kĩ thuật RAG[12] thông thường:

- *Indexing*: Raw data từ các nguồn khác nhau (PDF, HTML, Word, Markdown) được làm sạch và trích xuất, sau đó sẽ được chuyển đổi thành định dạng văn bản thuần nhất. Văn bản sẽ được chia ra thành các *chunks* và các chunks này sẽ được biến đổi sang dạng vector và được lưu trong một vector database.
- *Retrieval*: Khi nhận được một câu truy vấn, hệ thống sẽ biến đổi truy vấn thành vector và so sánh độ tương đồng giữa vector biểu diễn cho câu truy vấn và các đoạn văn bản đã được lập chỉ mục. Những đoạn văn bản có độ tương đồng cao nhất sẽ được chọn làm ngữ cảnh mở rộng cho mô hình.

- *Generation:* Truy vấn và các đoạn văn bản được tổng hợp thành một prompt, sau đó mô hình ngôn ngữ lớn sẽ phản hồi cho câu prompt đó. Mô hình có thể dựa trên kiến thức sẵn có hoặc chỉ sử dụng thông tin từ các tài liệu được cung cấp.

Với sự phát triển mạnh mẽ của LLMs trong những năm gần đây, các công trình liên quan đến RAG xuất hiện nhiều. Đầu tiên, đánh chỉ mục (indexing). PDF là một trong những định dạng tài liệu phổ biến nhất trên Internet, và cũng là một trong những nguồn dữ liệu chủ yếu được sử dụng cho việc đánh chỉ mục. Một nghiên cứu được thực hiện ở bài [13] cho thấy, công đoạn trích xuất nội dung từ một tập tin PDF (PDF parsing) có ảnh hưởng rất đáng kể đến chất lượng của việc truy vấn trong RAG. Về nguồn dữ liệu, thay vì chỉ tập trung vào văn bản là chủ yếu, có một số nghiên cứu, kỹ thuật liên quan được phát triển cho các nguồn dữ liệu khác như: Đồ thị tri thức là một dạng dữ liệu có cấu trúc, các dạng dữ liệu giả cấu trúc như PDF[12].

3.5 Knowledge tracing/Student Profiling

Với sự phát triển của máy tính và trí tuệ nhân tạo, nhiều công trình liên quan đến knowledge tracing/student profiling cũng đã được đề xuất và đưa ra, với mục tiêu cuối cùng là mô hình được năng lực của người học, từ đó phục vụ cho các bài toán khác trong giáo dục thông minh, đặc biệt là giáo dục thông minh. Một trong những mô hình tiêu biểu và cổ điển nhất trong lĩnh vực này đó là *Bayesian knowledge tracing*, với ý tưởng đó là mô hình hóa kiến thức của học sinh bằng cách sử dụng các biến ngẫu nhiên I_j , trong đó:

$$I_j = \begin{cases} 1, & \text{Nếu học sinh thành thục kĩ năng thứ } j \\ 0, & \text{ngược lại} \end{cases} \quad (3.1)$$

Ngoài ra, ở trong [14], nhóm tác giả đã xây dựng một mô hình knowledge tracing dựa trên một giả định đó là các khái niệm, mảng kiến thức có liên quan đến nhau về mặt ngữ nghĩa, và nếu như mức độ thành thạo của người học ở một mảng kiến thức nhất định có thể đo được, và người học có một bước tiến ở một kĩ năng, hay một khái niệm nào đó, thì ở các mảng kiến thức liên quan, người học cũng có một bước tiến nhất định tương đối.

Với sự phát triển của Học sâu trong những năm trở lại đây, một nghiên cứu sử dụng một mô hình với tên gọi là *Deep knowledge tracing* [3] để giải quyết bài toán đã nêu, trong đó, ý tưởng của giải pháp đó là sử dụng mô hình *Recurrent Neural Network (RNN)*. Việc sử dụng RNN cho phép ta có thể nhận diện ra được những hình mẫu phức tạp hơn trong việc mô hình hóa kiến thức của học sinh.

Trong những năm trở lại đây, với sự xuất hiện của Mô hình ngôn ngữ lớn, có nhiều nghiên cứu đánh giá và tìm cách ứng dụng Mô hình ngôn ngữ lớn vào trong bài toán Knowledge tracing. Trong [15], (nhóm) tác giả đã đặt ra vấn đề rằng, do đặc tính *black-box* của các mô hình học sâu, nên kết quả của chúng *không thể giải thích được*, và (nhóm) tác giả đã tận dụng hai đặc tính



suy luận và tạo sinh của mô hình ngôn ngữ lớn để đưa ra một lời giải thích cho hoạt động đánh giá năng lực của học viên của mô hình.

Chương 4

Hệ thống đề xuất

4.1 Mô tả hệ thống

Hệ thống học tập trực tuyến thông minh được thiết kế để phục vụ cho sinh viên ngành công nghệ thông tin và những người có nhu cầu học lập trình. Mục tiêu chính của hệ thống là cung cấp trải nghiệm học tập cá nhân hóa thông qua việc ứng dụng các mô hình ngôn ngữ lớn (LLM – Large Language Model), nhằm tối ưu hóa quá trình học tập một cách linh hoạt, hiệu quả và phù hợp với nhu cầu riêng biệt của từng người học.

Hệ thống tập trung vào sự tương tác giữa người học và trí tuệ nhân tạo (AI), tạo ra một môi trường học tập thông minh. Cụ thể, hệ thống hỗ trợ giảng viên dễ dàng tạo, chỉnh sửa và quản lý nội dung học tập. Các chức năng chính của hệ thống bao gồm:

- **Quản lý khóa học:** Giảng viên có thể cung cấp thông tin chi tiết về khóa học, bao gồm tên khóa học, mô tả, mục tiêu học tập, và đối tượng học viên. Các tài liệu học tập có thể được tải lên hoặc liên kết đến các tài nguyên bổ sung.
- **Quản lý nội dung học:** Khóa học được chia thành các module hoặc bài học nhỏ với mục tiêu học tập cụ thể và thời gian hoàn thành dự kiến. Mỗi module hoặc bài học sẽ có mục tiêu học tập rõ ràng để sinh viên biết họ cần đạt được những kỹ năng hoặc kiến thức nào sau khi hoàn thành.
- **Lộ trình học tập cá nhân hóa:** Lộ trình học tập của mỗi sinh viên sẽ được cá nhân hóa thông qua các đề xuất learning item (bao gồm khóa học, module, bài học) dựa trên mục tiêu học tập, nền tảng kiến thức, và hành vi học tập của người dùng (như điểm số, thời gian hoàn thành bài học). Hệ thống sẽ linh hoạt đề xuất các item phù hợp với trình độ hiện tại của người học. Các bài học và module sẽ được điều chỉnh dần theo độ khó tăng lên, giúp người học tiếp cận kiến thức mới một cách hợp lý và tiến bộ liên tục.



- **Gia sư AI hỗ trợ lập trình:** Một trong những tính năng nổi bật của hệ thống là gia sư AI, đóng vai trò như một trợ lý thông minh trong việc giám sát quá trình học của người học. Gia sư AI có khả năng cung cấp các giải thích chi tiết về mã nguồn, gợi ý cách viết mã tối ưu, hỗ trợ sửa lỗi (debug), và đưa ra phản hồi về chất lượng mã lệnh. Tính năng này không chỉ giúp người học nâng cao kỹ năng lập trình mà còn giúp cải thiện hiệu quả học tập thông qua việc tiếp thu kiến thức một cách nhanh chóng và dễ hiểu.
- **Hệ thống tạo bài tập tự động:** Hệ thống có khả năng tự động tạo ra các bài tập lập trình theo chủ đề mà người học mong muốn. Bộ test case cho bài tập sẽ được sinh ra tự động tùy thuộc vào mức độ khó và yêu cầu của bài tập. Người học có thể yêu cầu hệ thống tạo bài tập ở các cấp độ khác nhau (từ cơ bản đến nâng cao) hoặc nhập vào một câu hỏi lập trình cụ thể để hệ thống tạo bộ test case phù hợp.
- **Bài tập sẽ được điều chỉnh linh hoạt:** Bài tập sẽ được điều chỉnh linh hoạt để phù hợp với năng lực và tiến độ học tập của người học. Hệ thống sẽ theo dõi và đánh giá kết quả bài tập dựa trên thời gian giải quyết, độ chính xác, và điểm số, từ đó điều chỉnh lộ trình học tập của người học.
- **Theo dõi tiến độ và báo cáo học tập:** Hệ thống cung cấp công cụ theo dõi tiến độ học tập chi tiết. Sinh viên có thể xem lại thời gian học, số lượng bài tập đã hoàn thành và kết quả của các bài kiểm tra. Hệ thống sẽ cung cấp báo cáo chi tiết về quá trình học tập của sinh viên, bao gồm các bài học đã hoàn thành, kết quả đạt được và các gợi ý từ hệ thống cho bước học tiếp theo. Dựa trên các chỉ số này, hệ thống sẽ đánh giá mức độ tiến bộ của người học và điều chỉnh lộ trình học tập một cách linh hoạt.

4.2 Khảo sát yêu cầu

4.2.1 Người dùng

Các người dùng liên quan trong hệ thống:

- Học viên.
- Quản trị viên.
- Giảng viên

4.2.2 Yêu cầu chức năng

4.2.2.1 User Management

1. **Đăng ký và đăng nhập:** Người dùng có thể tạo tài khoản và đăng nhập bằng thông tin cá nhân hoặc tài khoản mạng xã hội.



2. **Hồ sơ học tập:** Mỗi người dùng có một hồ sơ học tập, lưu trữ thông tin cá nhân, tiến độ học tập, và kết quả các bài kiểm tra/bài tập.
3. **Quản lý lịch sử học tập:** Hệ thống lưu trữ và cho phép người dùng xem lại các bài học, bài tập đã hoàn thành và kết quả đạt được.
4. **Cấu hình cá nhân:** Người dùng có thể cập nhật thông tin cá nhân, độ tuổi.

4.2.2.2 Learning Resource Recommendation System

1. **Cá nhân hóa đề xuất học tập:** Hệ thống tự động đề xuất các tài nguyên học tập (khóa học, module, bài học, bài tập, project) cho mỗi người dùng dựa trên mục tiêu học tập, độ tuổi, trình độ học vấn, phong cách học tập, và hành vi học tập, lịch sử học tập, mức độ thông thạo của người học trên một khái niệm hoặc một đầu ra học tập cụ thể.
2. **Lựa chọn tài nguyên học tập:** Hệ thống cho phép người dùng tự chọn các learning item dựa trên sở thích hoặc mục tiêu cá nhân, cung cấp sự linh hoạt trong việc tiếp cận nội dung học tập.
3. **Theo dõi phản hồi người dùng:** Hệ thống ghi nhận phản hồi của người dùng về các tài nguyên học tập để cải thiện đề xuất và tối ưu hóa trải nghiệm người học.

4.2.2.3 Exercise Generator

1. **Tạo bài tập theo mức độ:** Hệ thống tự động sinh ra các bài tập từ cơ bản đến nâng cao phù hợp với khả năng của người dùng.
2. **Tạo bài tập theo yêu cầu:** Người dùng có thể yêu cầu hệ thống tạo bài tập ở các cấp độ khác nhau hoặc nhập vào một câu hỏi lập trình cụ thể.
3. **Tự động điều chỉnh bài tập:** Dựa trên tiến độ học tập, các bài tập sẽ được nâng cấp để phù hợp với mức độ phát triển của người dùng.
4. **Phân tích kết quả bài tập:** Hệ thống đánh giá bài tập dựa trên thời gian giải, độ chính xác, và kết quả điểm số để điều chỉnh lộ trình học.
5. **Tạo bộ test case tự động:** Hệ thống tự động tạo bộ test case phù hợp với yêu cầu của bài tập mà người dùng upload.

4.2.2.4 AI Tutor

1. Hệ thống có tích hợp một tính năng là Smart Assistant. Khi người dùng tương tác với tài nguyên học tập, sẽ có một hệ thống chatbot hỗ trợ trả lời nội dung liên quan đến tài nguyên học tập mà người dùng đang xem.



2. Ứng với các tài nguyên học tập đòi hỏi người dùng phải thực hành như là các bài tập lập trình, thì hệ thống cũng có tích hợp một số tính năng đi kèm như sau:

- **Hỗ trợ giải thích code:** Người dùng có thể yêu cầu gia sư AI giải thích code hoặc hướng dẫn cách giải quyết các bài tập lập trình.
- **Gợi ý hướng giải:** Khi gặp khó khăn, hệ thống sẽ đưa ra các gợi ý từ gia sư AI nhằm hỗ trợ người dùng hoàn thành bài tập.
- **Đưa ra gợi ý:** Hệ thống có thể hỏi người dùng liệu họ có muốn thay đổi bài tập sang một bài dễ hơn với nội dung tương tự khi gặp vấn đề với bài hiện tại.
- **Tối ưu hóa code:** Gia sư AI cung cấp gợi ý cách viết mã tối ưu hơn.
- **Hỗ trợ sửa lỗi:** Gia sư AI cung cấp hỗ trợ debug khi người dùng gặp lỗi trong code.
- **Đánh giá chất lượng code:** Gia sư AI cung cấp phản hồi kịp thời về chất lượng mã lệnh của người dùng.

4.2.2.5 Progress Tracking

1. **Theo dõi tiến độ học tập:** Hệ thống ghi nhận thời gian học, số lượng bài tập hoàn thành và kết quả các bài kiểm tra.
2. **Đánh giá tiến độ:** Dựa trên các chỉ số như tốc độ giải quyết bài tập và điểm số, hệ thống đánh giá sự tiến bộ của người dùng và điều chỉnh lộ trình học tập.
3. **Báo cáo tiến độ:** Người dùng có thể xem báo cáo chi tiết về quá trình học tập, bao gồm các bài học đã hoàn thành, kết quả đạt được, và gợi ý từ hệ thống cho bước tiếp theo.
4. **Phân tích dữ liệu học tập:** Hệ thống phân tích hành vi học tập của người dùng, bao gồm điểm số, thời gian hoàn thành bài tập, và tần suất yêu cầu hỗ trợ để cải thiện trải nghiệm học tập.

4.2.2.6 Course Management

1. **Tạo khóa học/module:** Giảng viên có thể tạo và thiết kế các khóa học hoặc module học tập mới. Ứng với mỗi một khóa học, giảng viên sẽ phải cung cấp các thông tin về đầu ra học tập (learning outcome), về mô tả của khóa học. Ứng với mỗi khóa học sẽ bao gồm các modules/bài học/projects tương ứng.
2. **Quản lý nội dung:** Giảng viên có thể cập nhật, chỉnh sửa nội dung của khóa học cũng như tài nguyên học tập. Hệ thống cho phép tải lên các tài liệu học tập hoặc liên kết đến các tài nguyên bổ sung.



4.2.2.7 Admin Management

1. Quản lý người dùng:

- Quản trị viên có thể xem danh sách người dùng, tìm kiếm, lọc, và sắp xếp theo các tiêu chí như tên, email, trạng thái tài khoản, và tiến độ học tập.
- Quản trị viên có thể khóa/mở khóa tài khoản người dùng hoặc thay đổi quyền truy cập của người dùng (như chuyển từ người dùng bình thường sang giảng viên).

2. Quản lý phản hồi:

- Xem và xử lý các phản hồi, đánh giá của người dùng về khóa học, giảng viên, và hệ thống.
- Gửi phản hồi cho người dùng về các vấn đề đã được xử lý.

4.2.3 Yêu cầu phi chức năng

• Hiệu năng

- Hệ thống phải xử lý yêu cầu truy cập và tương tác của người dùng một cách nhanh chóng và hiệu quả, với thời gian phản hồi không vượt quá 2 giây cho các thao tác thông thường như đăng nhập, xem khóa học, và tải trang.
- Hệ thống phải có khả năng hỗ trợ ít nhất 1000 người dùng đồng thời mà không làm giảm hiệu năng.
- Thời gian phản hồi của các công cụ AI không được quá 7-8s.

• Khả năng mở rộng

- Hệ thống có khả năng mở rộng khi số lượng người dùng, khóa học, và bài tập tăng lên mà không ảnh hưởng đến chất lượng dịch vụ và thay đổi hệ thống quá nhiều.
- Có thể thêm các chức năng phụ kết hợp với chức năng sẵn có nếu có yêu cầu mở rộng.

• Bảo mật

- Dữ liệu cá nhân của người dùng và kết quả học tập phải được mã hóa trong quá trình truyền tải và lưu trữ.

• Tính khả dụng

- Người dùng có thể dễ dàng điều hướng giữa các khóa học, theo dõi tiến trình học tập và thực hiện các bài kiểm tra mà không gặp khó khăn.



- Giao diện người dùng cần trực quan, dễ sử dụng, màu sắc cơ bản.
- Tính tương thích
 - Hệ thống phải tương thích với nhiều trình duyệt phổ biến (Chrome, Firefox, Safari, Edge) và các hệ điều hành khác nhau (Windows, macOS, Linux).



4.2.4 Yêu cầu dữ liệu

4.2.4.1 Các thực thể và thuộc tính dữ liệu

1. User

- Thực thể: User
- Thuộc tính: ID, username, email, password, role (student, admin, teacher)
 - Student: Goal, Birthyear, Background.
 - Teacher: Organizaiton
 - Admin

2. Course

- Thực thể: Course
- Thuộc tính: ID, Name, Description, Type, Learning Outcomes, nCredits.

3. LearningPath

- Thực thể: LearningPath.
- Thuộc tính: ID, Progress, Objective, StartDate, EndDate.

4. Lesson

- Thực thể: Lesson.
- Thuộc tính: ID, Lesson Outcomes, Description, Name, Documents.

5. Recommend Lesson

- Thực thể: Recommend Lesson.
- Thuộc tính: Expalin, Content, Progress.

6. Exercise

- Thực thể: Exercise
- Thuộc tính: ID, Name, Description, Type (Quiz, Code).

7. Module

- Thực thể: Module
- Thuộc tính: ID, Objective, Title, Access.

8. Quiz



- Thực thể: Quiz
- Thuộc tính: ID, Status, Score, MaxScore, LLM Response(Name, Difficulty, Question, Answer, Options, Image).

9. Document

- Thực thể: Document
- Thuộc tính: ID, LLM Response.

10. Activity

- Thực thể: Activity
- Thuộc tính: ID, Type, Description, Timestamp.

11. Feedback

- Thực thể: Feedback
- Thuộc tính: ID, Content, Time.

4.2.4.2 Mối quan hệ dữ liệu

- User:
 - Specializes: Teacher, Manager, Student
 - Mối quan hệ với các thực thể:
 - * Student:
 - Include: 1-N Activity
 - Send: 1-N Feedback
 - Submission: N-N Exercise
 - Join: N-N Course
 - Bookmark: N-N Lesson
 - * Teacher:
 - Upload: 1-N Course
 - * Manager:
 - Manage: N-N Course
- Course:
 - Include: 1-N Lesson
 - 1 Student - 1 Course: 1 LearningPath



- Mối quan hệ với các thực thể:
 - * Include: 1-N Lesson
 - * 1 Student 1 Course Have 1 LearningPath: Mối quan hệ giữa Student và Course, mỗi học viên chỉ có một LearningPath cho mỗi khóa học.
- Lesson:
 - Specializes: Recommend Lesson
 - Mối quan hệ với các thực thể:
 - * Include: 1-N Module
- LearningPath:
 - Have: 1-N Recommend Lesson
- Recommend Lesson:
 - Include: 1-N Module
- Module:
 - Generate: 1-N Quiz
 - Generate: 1-1 Document

4.3 Kết luận

Chương 4 đã trình bày chi tiết về hệ thống đề xuất trong môi trường học tập trực tuyến thông minh. Hệ thống cung cấp các công cụ mạnh mẽ giúp người học có trải nghiệm học tập hiệu quả và tối ưu. Các tính năng như tạo bài tập tự động, theo dõi tiến độ học tập, và hỗ trợ từ gia sư AI giúp người học nâng cao khả năng lập trình và tiếp cận kiến thức một cách linh hoạt, hiệu quả.

Chương 5

Phân tích và thiết kế

5.1 Phân tích

5.1.1 Business Process Modeling

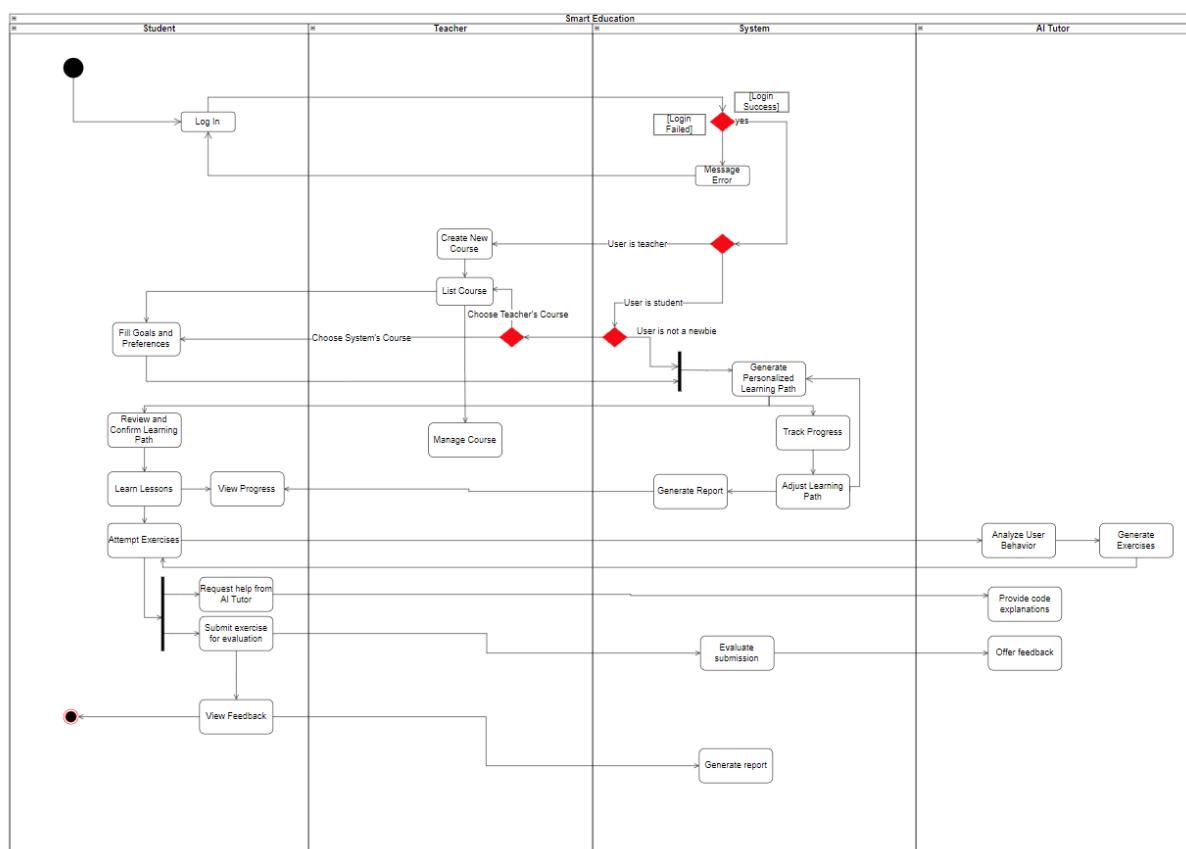


Figure 5.1: Activity diagram

Activity diagram mô tả quy trình tương tác giữa bốn Actors: **Student (Học sinh)**, **Teacher (Giáo viên)**, **System (Hệ thống)**, và **AI Tutor (Gia sư AI)** trong hệ thống giáo dục thông minh.

Học sinh bắt đầu bằng việc đăng nhập và tùy thuộc vào vai trò của họ (giáo viên hoặc học sinh), các hành động khác nhau sẽ diễn ra. Giáo viên có thể tạo khóa học mới hoặc chọn khóa học hiện có để quản lý, trong khi học sinh sẽ xác định mục tiêu học tập và xác nhận lộ trình học do hệ thống đề xuất. Sau đó, học sinh có thể học bài, thực hiện bài tập, và nếu cần có thể yêu cầu trợ giúp từ AI Tutor. Hệ thống cũng tự động theo dõi tiến độ học tập, điều chỉnh lộ trình và đánh giá bài nộp của học sinh. AI Tutor hỗ trợ bằng cách phân tích hành vi, giải thích mã code, và tạo bài tập phù hợp.

5.1.2 Use case diagrams

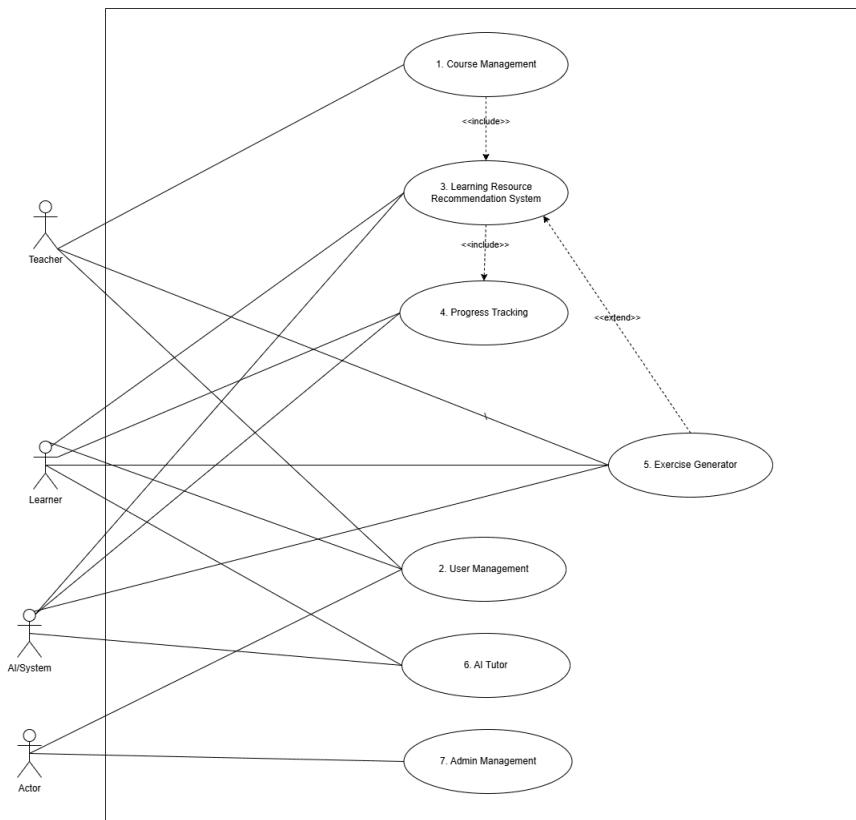


Figure 5.2: Use case diagram

Mô hình use case tổng thể cho hệ thống học tập thông minh bao gồm ba tác nhân chính: Giáo viên, Người học, Quản trị viên và Hệ thống/AI. Các use case bao gồm Quản lý khóa học, Quản lý người dùng, Quản lý lộ trình học tập thích ứng, Theo dõi tiến độ, Tạo bài tập, Quản lý hệ thống cho Quản trị viên và Gia sư AI.

5.1.2.1 Course Management

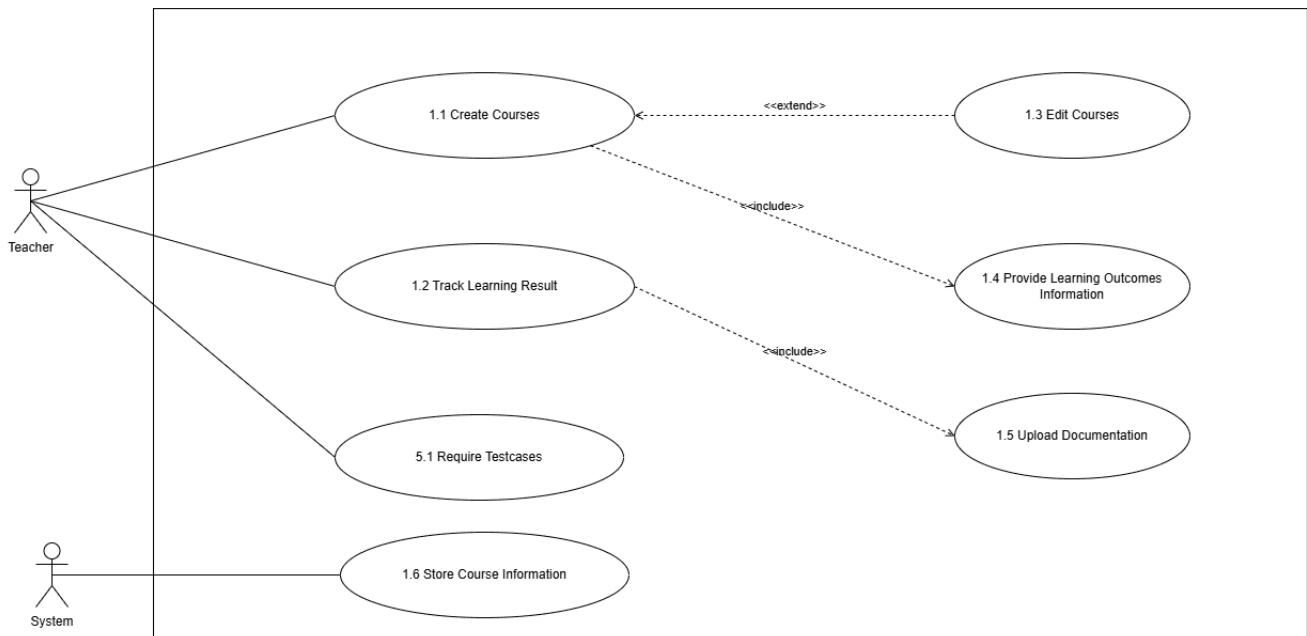


Figure 5.3: Detailed use case with id 1: Course Management

Use case **Quản lý khóa học** bao gồm ba chức năng chính: Tạo khóa học, Theo dõi kết quả học tập và Yêu cầu bài kiểm tra. Use case "**Chỉnh sửa khóa học**" được mở rộng từ "**Tạo khóa học**", cho thấy việc chỉnh sửa là một phần mở rộng tùy chọn của quá trình tạo khóa học. Use case diagram thể hiện các trách nhiệm chính của giáo viên trong việc quản lý khóa học trong hệ thống.

Ngoài ra, ở module này, Hệ thống sẽ có nhiệm vụ lưu trữ thông tin về khóa học mà giáo viên đã tạo ra.

5.1.2.2 User Management

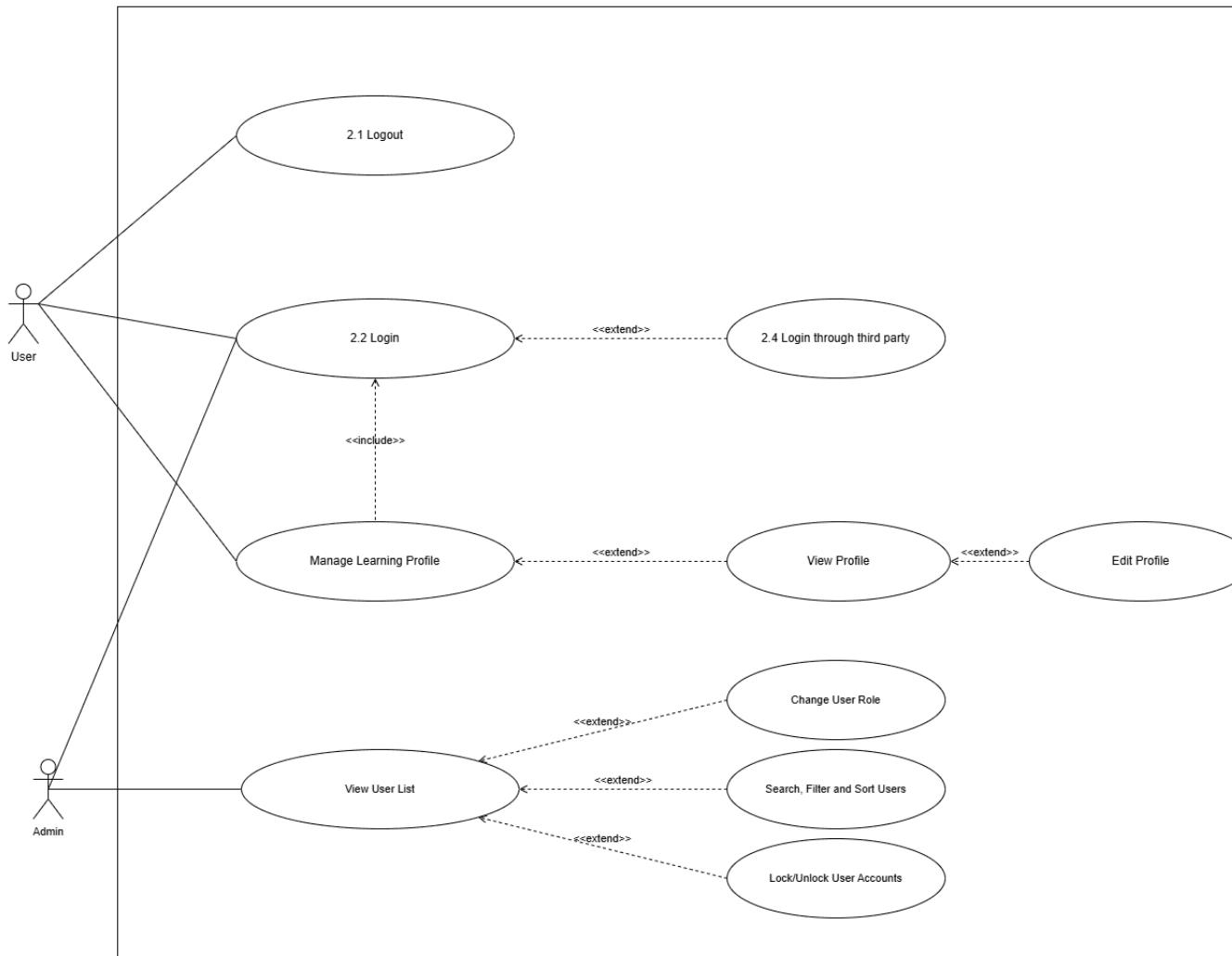


Figure 5.4: Detailed use case with id 2: User Management

Use case **Quản lý người dùng**, tập trung vào tác nhân Người dùng bao gồm các chức năng như Đăng nhập, Đăng xuất, Quản lý hồ sơ học tập và Xem hồ sơ. Use case diagram cho thấy "**Đăng nhập qua third-party**" mở rộng từ use case "**Đăng nhập**" cơ bản, cung cấp một phương thức đăng nhập thay thế. Ngoài ra, "**Xem hồ sơ**" được thể hiện như một phần mở rộng của "**Quản lý hồ sơ học tập**", cho thấy đây là một hành động liên quan nhưng riêng biệt.

Quản trị viên có quyền đăng nhập bằng tài khoản của mình, từ đó "**Xem danh sách người dùng**" sử dụng hệ thống. Ngoài ra, quản trị viên có thể "**Tìm kiếm, Lọc và Sắp xếp theo thông tin người dùng**", "**Vô hiệu hóa tài khoản/Kích hoạt tài khoản**" và "**Thay đổi quyền truy cập của người dùng**". Ví dụ: Thay đổi từ người học sang giảng viên.

5.1.2.3 Learning Resource Recommendation System

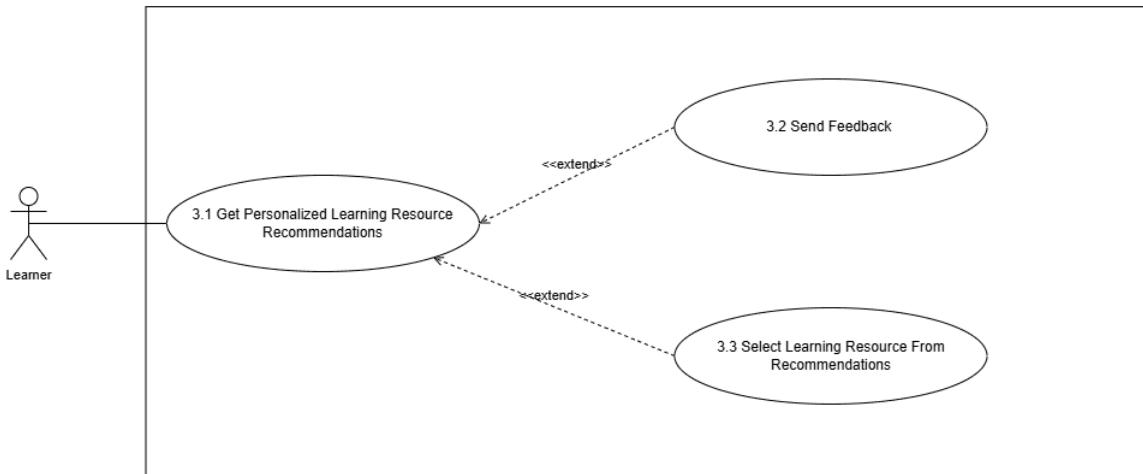


Figure 5.5: Detailed use case with id 3: Adaptive Learning Path Management of Learner Actor

Tác nhân chính trong biểu đồ trên là người học (Learner), có thể tương tác với hệ thống để nhận các đề xuất tài nguyên học tập. Use case "**Get Personalized Learning Resource Recommendations**" cung cấp cho người học các tài nguyên học tập phù hợp với nhu cầu và lịch sử học tập của họ. Use case "**Send Feedback**" có mối quan hệ "<extend>" với "**Get Personalized Learning Resource Recommendations**", vì người học có thể gửi phản hồi về các tài nguyên đã được đề xuất để cải thiện hệ thống. Use case "**Select Learning Resource From Recommendations**" cũng mở rộng từ "**Get Personalized Learning Resource Recommendations**", cho phép người học tự chọn các tài nguyên mà họ thấy phù hợp với mục tiêu cá nhân.

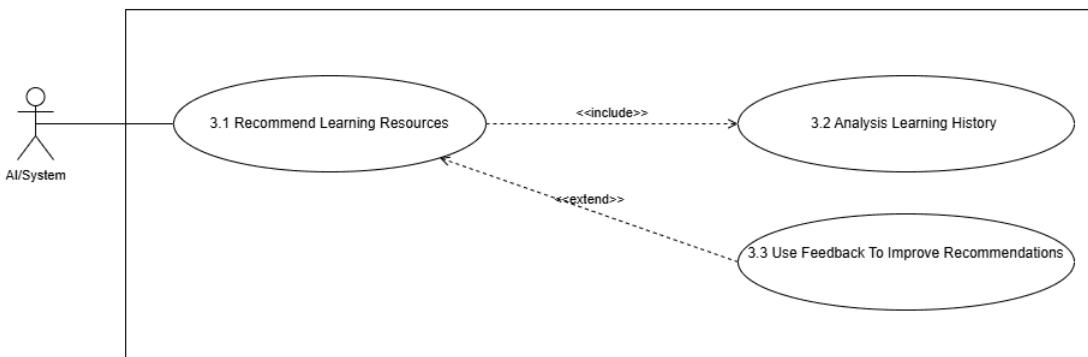


Figure 5.6: Detailed use case with id 3: Learning Resource Recommendation System of System Actor

Tác nhân chính là hệ thống AI, có nhiệm vụ chính là đề xuất tài nguyên học tập cá nhân hóa cho người dùng. Use case "**Recommend Learning Resources**" bao gồm hoạt động "**Analysis Learning History**" (phân tích lịch sử học tập), do quá trình phân tích lịch sử học tập là một phần không thể thiếu khi hệ thống tạo ra các đề xuất. Hệ thống có thể mở rộng thêm use case "**Use**

Feedback To Improve Recommendations" (sử dụng phản hồi để cải thiện đề xuất) dựa trên các phản hồi người dùng, giúp tinh chỉnh và nâng cao chất lượng đề xuất. Mỗi quan hệ "**<include>**" được sử dụng giữa "**Recommend Learning Resources**" và "**Analysis Learning History**" vì quá trình phân tích là cần thiết. Mỗi quan hệ "**<extend>**" giữa "**Recommend Learning Resources**" và "**Use Feedback To Improve Recommendations**" vì cải thiện đề xuất dựa trên phản hồi là một phần mở rộng có thể xảy ra tùy theo ngữ cảnh.

5.1.2.4 Progress Tracking

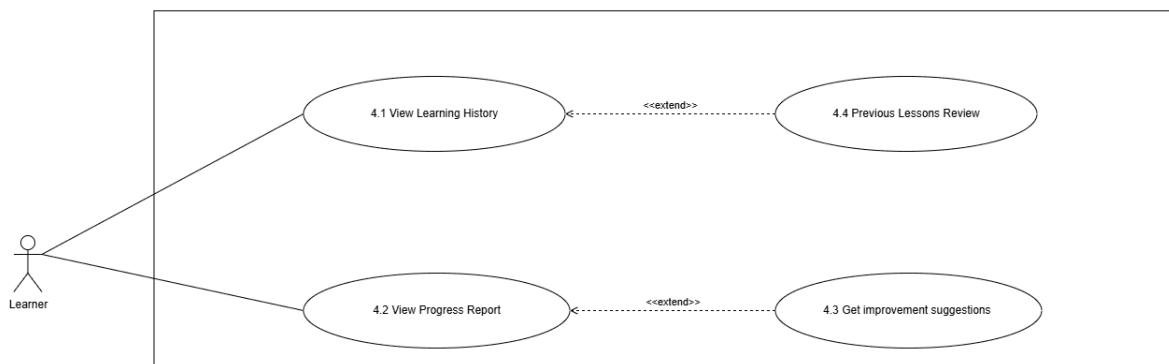


Figure 5.7: Detailed use case with id 4: Progress Tracking of Learner Actor

Use case diagram trên thể hiện các chức năng **Theo dõi tiến độ dành cho Người học**. Có hai use case chính là "**Xem lịch sử học tập**" và "**Xem báo cáo tiến độ**". "**Xem lịch sử học tập**" mở rộng đến "**Xem lại các bài học trước đó**", cho phép người học ôn tập. "**Xem báo cáo tiến độ**" mở rộng đến "**Nhận đề xuất cải thiện**", cung cấp hướng dẫn cá nhân hóa, nhấn mạnh việc theo dõi và phản hồi liên tục để hỗ trợ quá trình học tập của người dùng.

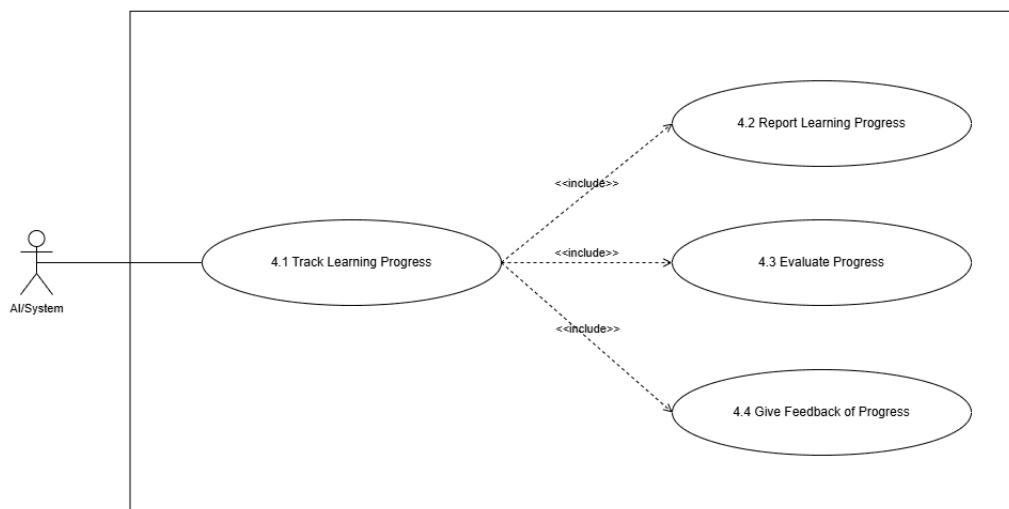


Figure 5.8: Detailed use case with id 4: Progress Tracking of System Actor

Sơ đồ use case của hệ thống **Exercise Generator** mô tả các tương tác giữa người dùng và



hệ thống. Người học có thể yêu cầu hệ thống tạo tự động các bài tập thông qua chức năng "**Request Auto Exercise Generation**", và sau đó thực hiện các bài tập đó để cải thiện kỹ năng của mình thông qua "**Do Exercise**". Bên cạnh đó, giảng viên có thể sử dụng chức năng "**Upload Exercise**" để tải lên các đề bài hoặc bài tập, từ đó hệ thống sẽ tự động xử lý và tạo ra các test case nhằm kiểm tra độ chính xác của bài giải thông qua "**Generate Test Cases**". Sau khi các test case được tạo, giảng viên có thể sử dụng chức năng "**Review Test Cases**" để xem xét và đánh giá các test case nhằm đảm bảo chúng phù hợp với nội dung bài tập và đáp ứng được yêu cầu giảng dạy. Các chức năng này giúp hệ thống tạo ra môi trường học tập linh hoạt, hỗ trợ hiệu quả cho người học và giảng viên.

5.1.2.5 Exercise Generator

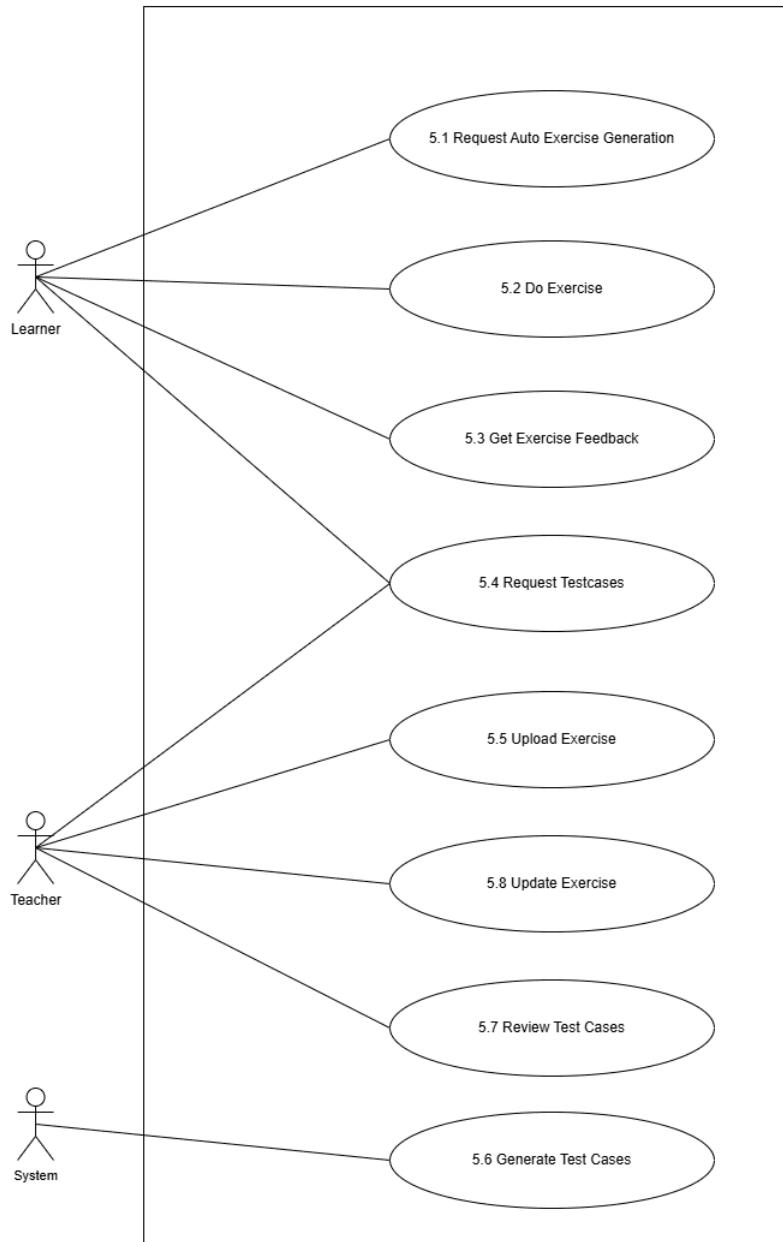


Figure 5.9: Detailed use case with id 5: Exercise Generator

Use case diagram thể hiện các chức năng liên quan đến bài tập dành cho Người học. Có bốn use case chính: "*Yêu cầu tạo bài tập tự động*", "*Làm bài tập*", "*Nhận phản hồi về bài tập*", và "*Yêu cầu bài kiểm tra*". Người học có thể yêu cầu hệ thống tạo bài tập tự động phù hợp với trình độ của họ. Sau khi làm bài tập, người học có thể nhận phản hồi để hiểu rõ hơn về kết quả của mình. Khả năng yêu cầu bài kiểm tra cho phép người học tự đánh giá kiến thức của mình khi cần.

5.1.2.6 AI Tutor

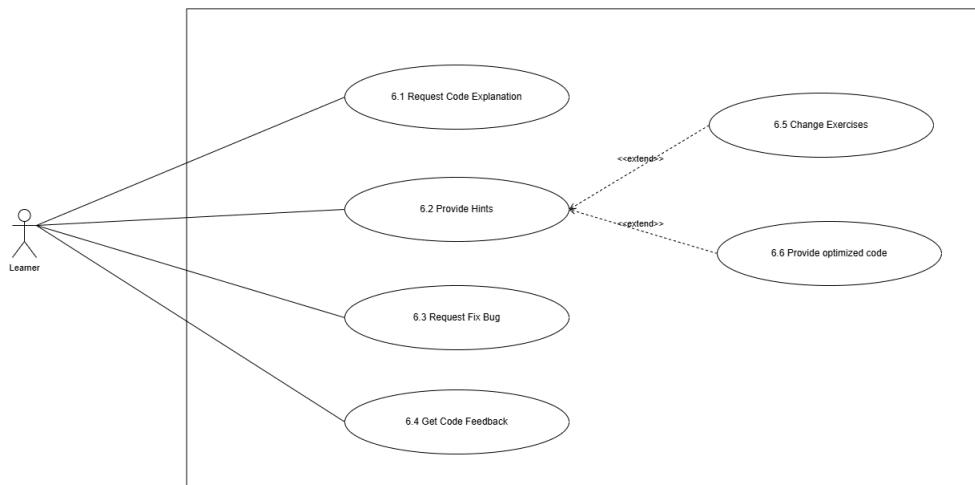


Figure 5.10: Detailed use case with id 6: AI Tutor of Learner Actor

Use case diagram thể hiện các chức năng của Gia sư AI dành cho Người học. Có bốn use case chính: "*Yêu cầu giải thích mã*", "*Yêu cầu gợi ý*", "*Yêu cầu sửa lỗi*", và "*Nhận phản hồi về mã*". Người học có thể yêu cầu giải thích về mã nguồn để hiểu rõ hơn về cấu trúc và logic của code. Chức năng yêu cầu gợi ý có hai chức năng quan hệ "*extend*" là "*Hỗ trợ chuyển đổi bài tập*" dựa trên nhu cầu, học lực của người dùng và "*Cung cấp mã tối ưu hóa*" giúp người học vượt qua các khó khăn trong quá trình học lập trình cũng như cải thiện tối đa kỹ thuật code của người dùng. Việc nhận phản hồi về mã giúp người học cải thiện kỹ năng lập trình của mình một cách liên tục.

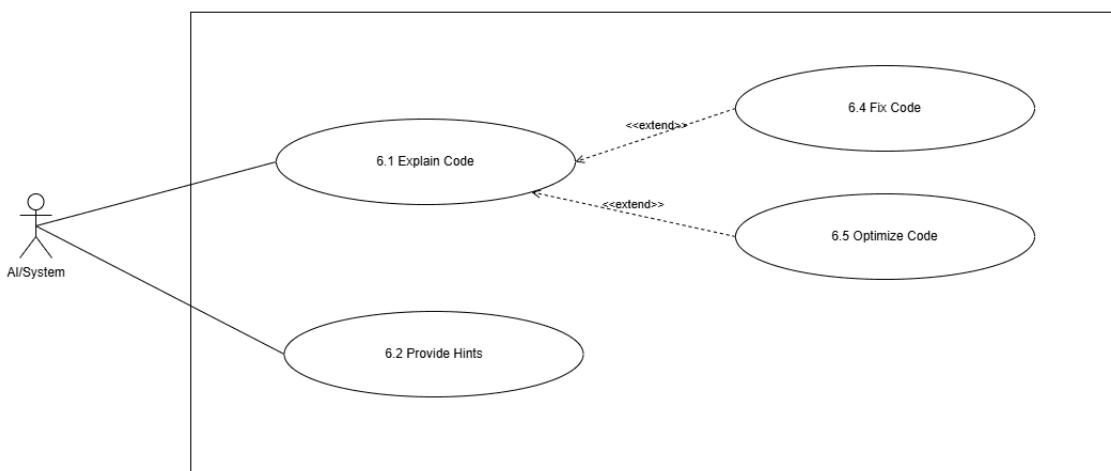


Figure 5.11: Detailed use case with id 6: AI Tutor of System Actor

Use case diagram trên mô tả **hệ thống AI Tutor** với vai trò hỗ trợ người dùng trong việc xử lý và hiểu mã lập trình. Hệ thống có khả năng *giải thích mã lệnh* (*Use case 6.1*) và *cung cấp gợi ý khi người dùng gặp khó khăn* (*Use case 6.2*). Từ chức năng giải thích mã, hệ thống có



thể mở rộng thêm các chức năng khác như *sửa lỗi* (*Use case 6.4*) và *tối ưu mã lệnh* (*Use case 6.5*). Hệ thống nhằm mục tiêu hỗ trợ người dùng học lập trình hiệu quả hơn thông qua các công cụ hỗ trợ trực tiếp từ AI.



5.2 Thiết kế

5.2.1 Kiến trúc hệ thống

5.2.1.1 Phân tích kiến trúc hệ thống phổ biến hiện nay

Hiện nay, có nhiều kiến trúc phần mềm khác nhau, mỗi loại có ưu và nhược điểm riêng, phù hợp với các loại dự án và yêu cầu cụ thể. Dưới đây là một số kiến trúc phần mềm phổ biến và các điểm chính của chúng:

1. Kiến trúc Monolithic

- **Định nghĩa:** Kiến trúc monolithic là một mô hình truyền thống trong đó toàn bộ ứng dụng được xây dựng như một đơn vị duy nhất, tích hợp chặt chẽ.
- **Điểm chính:**
 - Đơn giản để phát triển, triển khai và kiểm thử
 - Hiệu suất tốt do không có overhead của network calls
 - Khó mở rộng và bảo trì khi ứng dụng phức tạp
 - Khó áp dụng công nghệ mới cho từng phần của ứng dụng
- **Ví dụ:** Một ứng dụng web PHP truyền thống, nơi tất cả chức năng (UI, business logic, data access) đều nằm trong cùng một codebase.

2. Kiến trúc Microservices

- **Định nghĩa:** Kiến trúc microservices chia nhỏ ứng dụng thành các dịch vụ độc lập, mỗi dịch vụ chịu trách nhiệm cho một chức năng cụ thể và có thể được phát triển, triển khai độc lập.
- **Điểm chính:**
 - Dễ dàng mở rộng và bảo trì từng service riêng biệt
 - Cho phép sử dụng công nghệ phù hợp nhất cho mỗi service
 - Tăng cường khả năng chịu lỗi của hệ thống
 - Phức tạp hơn trong việc quản lý và điều phối các service
- **Ví dụ:** Netflix sử dụng kiến trúc microservices, với các service riêng biệt cho việc xử lý video, quản lý người dùng, đề xuất nội dung, v.v.

3. Kiến trúc Client-Server

- **Định nghĩa:** Kiến trúc client-server chia ứng dụng thành hai phần chính: client (trình bày và tương tác người dùng) và server (xử lý logic nghiệp vụ và lưu trữ dữ liệu).



• **Điểm chính:**

- Phân tách rõ ràng giữa client và server
- Dễ dàng mở rộng server để đáp ứng nhu cầu người dùng
- Cải thiện bảo mật thông qua kiểm soát truy cập tại server
- Phụ thuộc vào kết nối mạng giữa client và server
- **Ví dụ:** Một ứng dụng web với frontend (React) và backend (Node.js) giao tiếp qua RESTful API.

4. Kiến trúc Event-Driven

- **Định nghĩa:** Kiến trúc event-driven là một mô hình trong đó việc tạo ra, phát hiện, tiêu thụ và phản ứng với các sự kiện là cốt lõi của hệ thống.
- **Điểm chính:**
 - Tách rời các thành phần hệ thống, giảm sự phụ thuộc trực tiếp
 - Khả năng mở rộng và linh hoạt cao
 - Dễ dàng thêm chức năng mới mà không ảnh hưởng đến các thành phần hiện có
 - Có thể phức tạp trong việc theo dõi luồng xử lý và debug
- **Ví dụ:** Hệ thống thương mại điện tử, khi một đơn hàng được đặt, một sự kiện "OrderPlaced" được phát ra, kích hoạt các quy trình xử lý khác như cập nhật kho, gửi email xác nhận, v.v.

5. Kiến trúc Serverless

- **Định nghĩa:** Kiến trúc serverless cho phép phát triển và chạy ứng dụng mà không cần quản lý trực tiếp máy chủ, tập trung vào việc viết code cho các chức năng (functions) cụ thể.
- **Điểm chính:**
 - Giảm chi phí vận hành và bảo trì hạ tầng
 - Tự động mở rộng theo nhu cầu sử dụng
 - Chỉ trả tiền cho tài nguyên thực sự sử dụng
 - Có thể gặp vấn đề về cold start và vendor lock-in
- **Ví dụ:** Một ứng dụng xử lý ảnh sử dụng AWS Lambda để thực hiện các tác vụ như resize, filter, và lưu trữ ảnh khi người dùng tải lên.

5.2.1.2 Lựa chọn kiến trúc hệ thống

Dựa trên yêu cầu của hệ thống, nhóm quyết định sử dụng kiến trúc Client-Server. Cụ thể như sau:



1. Client Tier:

- **Frontend:** Ứng dụng Vue.js cung cấp giao diện người dùng cho sinh viên, giảng viên và admin.
- **Chức năng:** Hiển thị nội dung học tập, quản lý khóa học, bài tập, lộ trình học tập và tương tác với người dùng.
- **Tương tác:** Giao tiếp với server thông qua RESTful API.

2. Server Tier:

(a) API Layer:

- **Framework:** FastAPI xử lý các yêu cầu từ client.
- **Chức năng:**
 - *User Management:* Quản lý thông tin người dùng, xác thực và phân quyền.
 - *Course Management:* CRUD cho khóa học, module, bài học và nội dung học tập.
 - *Recommendation System:* Đề xuất tài nguyên học tập dựa trên dữ liệu người dùng.
 - *AI Tutor:* Tích hợp LangChain để hỗ trợ giải đáp thắc mắc và giải thích mã nguồn.
 - *Exercise Generation:* Tạo bài tập và test cases tự động bằng LangChain.
 - *Progress Tracking:* Theo dõi tiến độ học tập và tạo báo cáo.

(b) Persistence Layer:

- **PostgreSQL:** Lưu trữ dữ liệu có cấu trúc như thông tin người dùng, khóa học, bài học và tiến độ học tập.
- **AWS S3:** Lưu trữ tài liệu học tập, video bài giảng và các tài nguyên khác.

(c) Caching Layer:

- **Redis:** Lưu trữ tạm thời dữ liệu thường xuyên truy cập để cải thiện hiệu suất và giảm tải cho cơ sở dữ liệu.

Phân tích nguyên nhân:

- **Phân tách rõ ràng:** Kiến trúc Client-Server tách biệt giữa giao diện người dùng (client) và logic nghiệp vụ/dữ liệu (server), giúp dễ dàng phát triển và bảo trì.
- **Dễ mở rộng:** Server có thể được mở rộng bằng cách tăng tài nguyên hoặc thêm các instance FastAPI khi số lượng người dùng tăng.



- **Bảo mật:** Server kiểm soát truy cập và xác thực, đảm bảo an toàn dữ liệu.
- **Hiệu quả:** Sử dụng Redis để tối ưu hiệu suất truy vấn và AWS S3 để quản lý tài nguyên lớn như video, tài liệu.

Không dùng:

- *Monolithic:* Khó mở rộng, bảo trì và tích hợp công nghệ mới khi hệ thống phát triển.
- *Microservices:* Phức tạp trong quản lý nhiều dịch vụ, không cần thiết cho quy mô hiện tại của hệ thống.
- *Event-Driven:* Phức tạp trong việc đồng bộ và theo dõi trạng thái, không phù hợp với yêu cầu tương tác trực tiếp.
- *Serverless:* Phụ thuộc nhà cung cấp, chi phí cao khi tải tăng và hạn chế tùy chỉnh hiệu suất.

Luồng hoạt động:

- Người dùng tương tác với giao diện Vue.js (Client Tier).
- Client gửi các request đến server thông qua RESTful API.
- FastAPI (Server Tier) xử lý logic nghiệp vụ, truy vấn PostgreSQL hoặc AWS S3 để lấy/lưu dữ liệu.
- Redis được sử dụng để cache dữ liệu, giảm thời gian phản hồi.
- Kết quả được trả về client để hiển thị cho người dùng.

Tóm lại, kiến trúc Client-Server được chọn vì sự đơn giản, khả năng mở rộng và tính bảo mật cao, phù hợp với yêu cầu của hệ thống học tập trực tuyến thông minh. Việc sử dụng Vue.js, FastAPI, PostgreSQL, AWS S3 và Redis đảm bảo hiệu suất, linh hoạt và dễ dàng bảo trì trong tương lai.

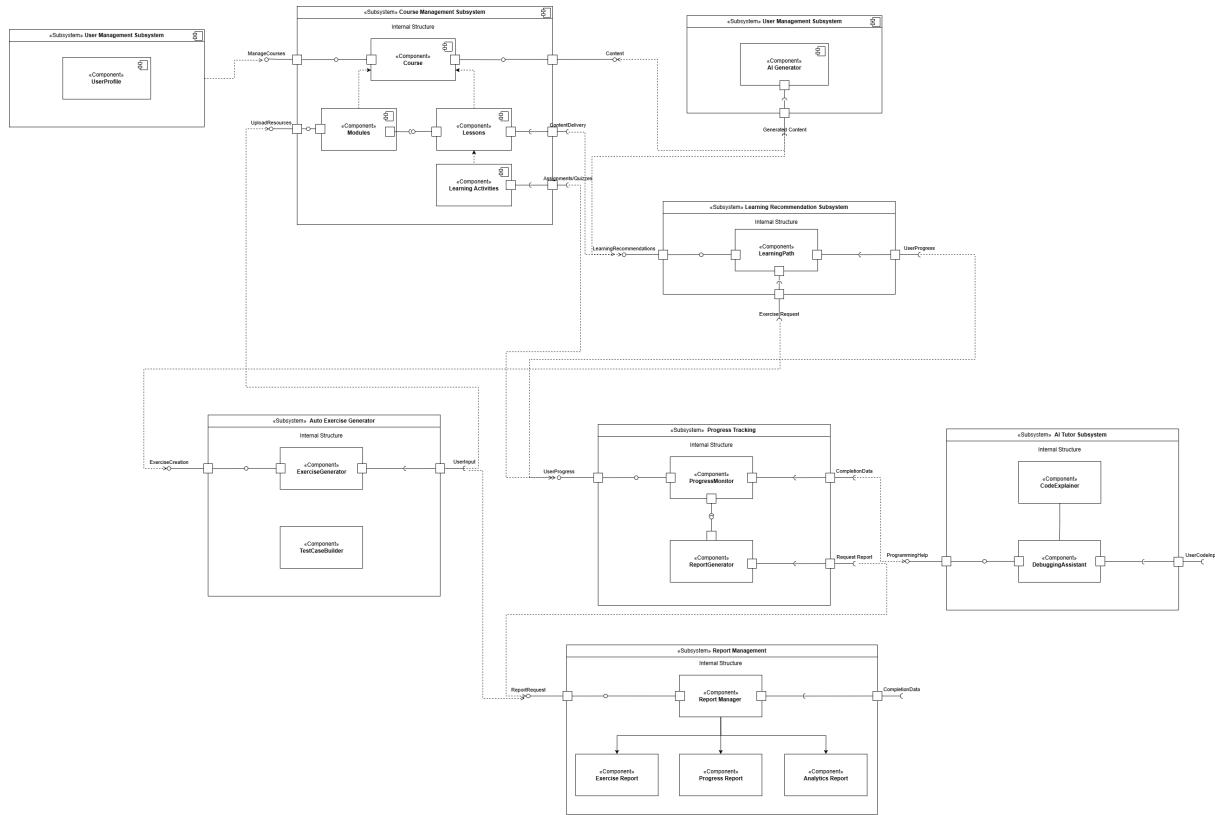


Figure 5.12: Component-based Diagram cho toàn bộ hệ thống

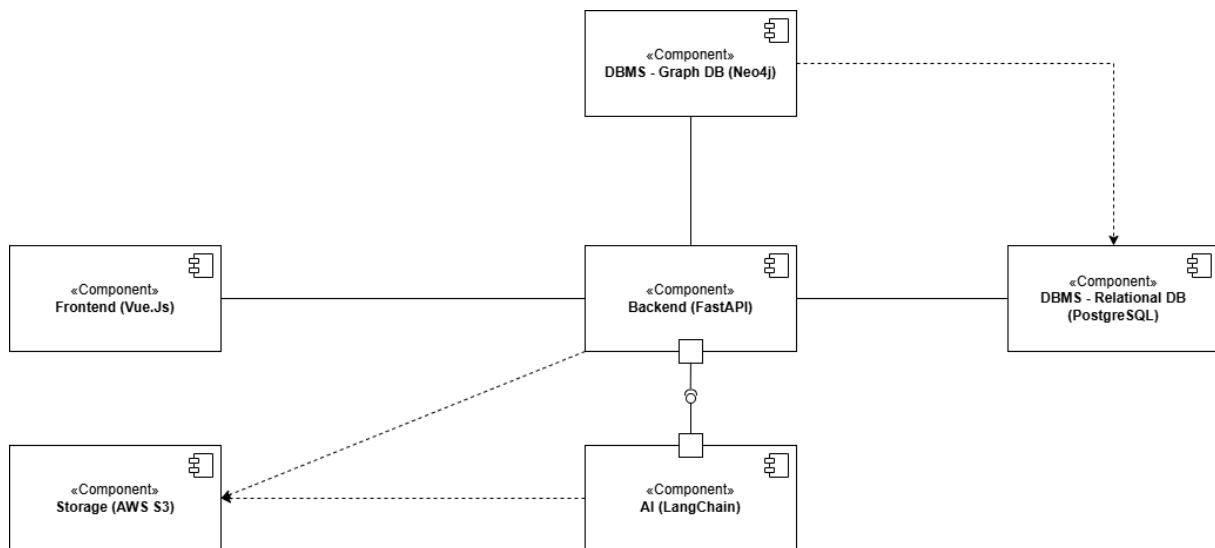


Figure 5.13: Component-based Diagram cho những công nghệ được chọn

5.3 Database

5.3.1 EER diagram

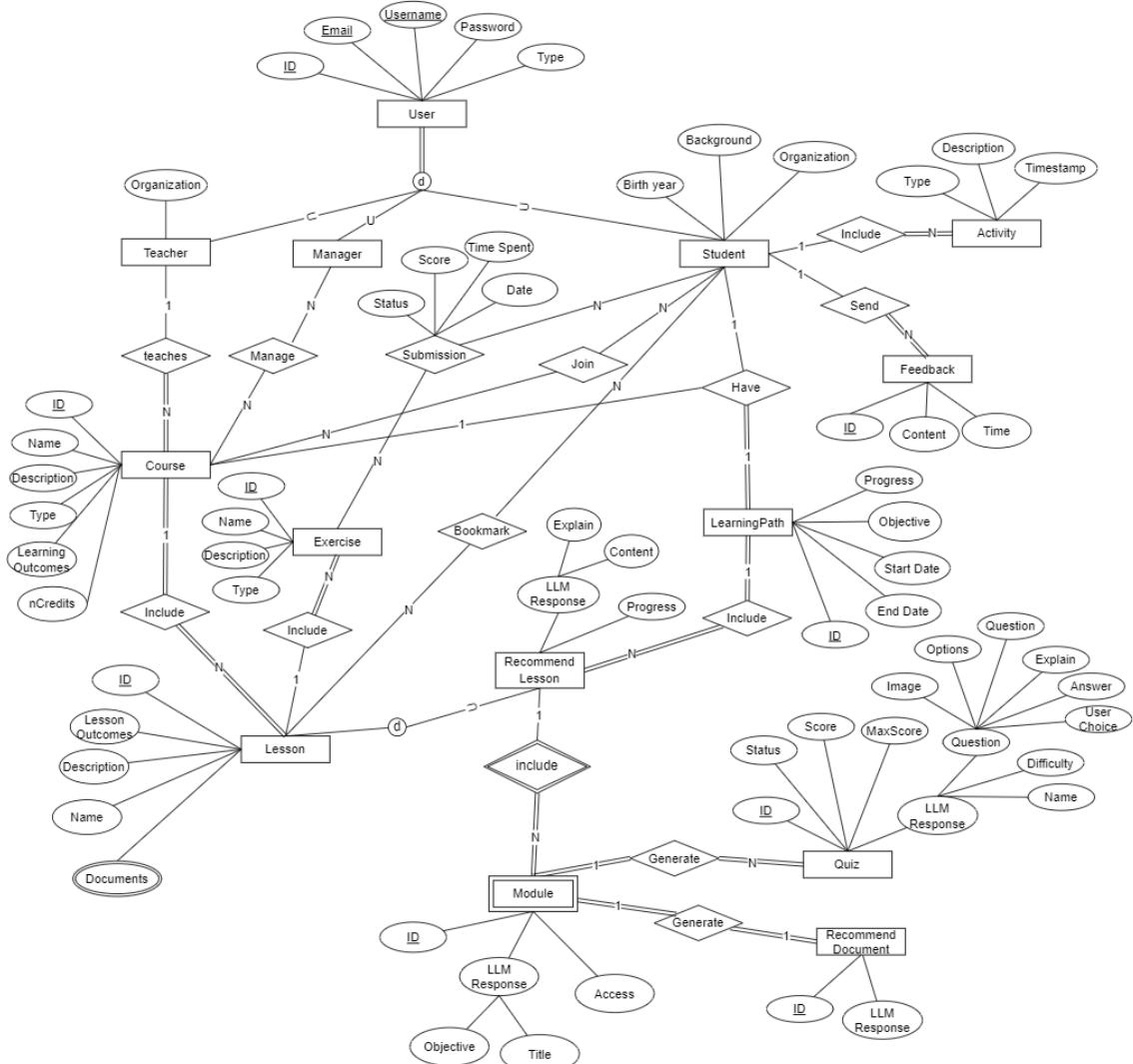


Figure 5.14: Sơ đồ EER của hệ thống

1. User có ba loại con: Teacher, Manager, và Student. Mỗi loại người dùng có các mối quan hệ khác nhau với các thực thể khác trong hệ thống:

- Student có các mối quan hệ sau:
 - Include một hoặc nhiều Activity.
 - Send nhiều Feedback cho hệ thống.
 - Submit nhiều Exercise để thực hành.
 - Join nhiều Course để tham gia các khóa học.
 - Bookmark nhiều Lesson để lưu trữ các bài học quan trọng.



- Teacher có thể Upload nhiều Course vào hệ thống để giảng dạy.
 - Manager có thể Manage nhiều Course trong hệ thống, tức là có quyền quản lý các khóa học.
2. Course bao gồm nhiều Lesson, và mỗi khóa học có thể Include nhiều bài học khác nhau. Mỗi học viên (Student) có thể tham gia một khóa học và cho mỗi khóa học, họ sẽ có một LearningPath riêng biệt. Mỗi quan hệ này thể hiện rằng mỗi học viên có một lộ trình học tập cho mỗi khóa học mà họ tham gia.
3. Lesson có thể Specialize thành Recommend Lesson, tức là các bài học có thể bao gồm các đề xuất học tập cho học viên. Mỗi bài học cũng có thể Include nhiều Module.
4. LearningPath của mỗi học viên sẽ Have nhiều Recommend Lesson. Các lộ trình học tập này sẽ dẫn dắt học viên thông qua các bài học được đề xuất trong suốt quá trình học.
5. Recommend Lesson có thể Include nhiều Module, tức là mỗi bài học được đề xuất có thể bao gồm nhiều mô-đun khác nhau để hỗ trợ học viên học tập một cách chi tiết hơn.
6. Module có thể Generate nhiều Quiz để đánh giá kết quả học tập của học viên. Mỗi mô-đun cũng có thể Generate một tài liệu (Document) duy nhất liên quan đến nội dung học.
7. Quiz chứa các câu hỏi và bài kiểm tra học viên, có thể bao gồm các phản hồi từ LLM Response, cung cấp thông tin về tên, độ khó, câu hỏi, câu trả lời, các lựa chọn, và hình ảnh liên quan đến bài kiểm tra.
8. Document có thể được tạo ra từ các mô-đun và cũng có phản hồi từ LLM Response, giúp cung cấp thêm tài liệu học tập cho học viên.
9. Activity bao gồm các hành động của người dùng trong hệ thống, và mỗi hoạt động có thể được Include bởi một hoặc nhiều User, đặc biệt là học viên.
10. Feedback là thông tin phản hồi từ học viên gửi đến hệ thống và có thể được Send từ nhiều User, đặc biệt là từ học viên để cải thiện chất lượng giảng dạy và học tập.

5.3.2 Lược đồ CSDL quan hệ ánh xạ

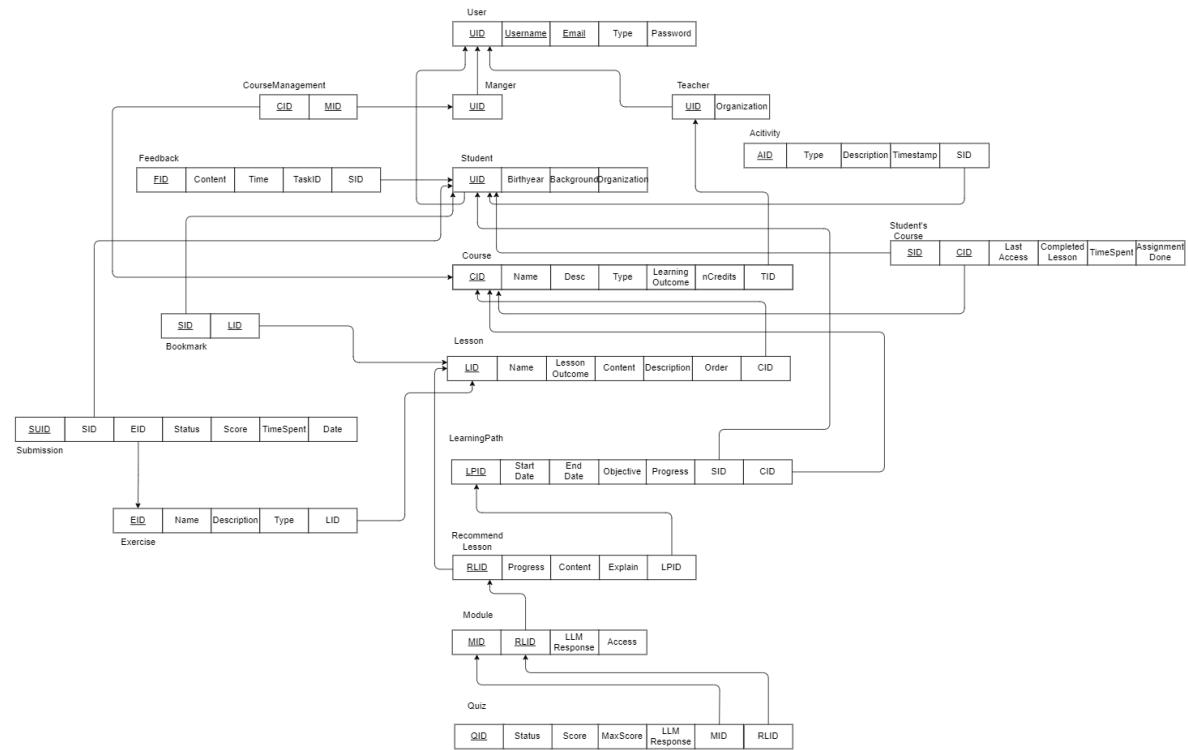


Figure 5.15: Lược đồ CSDL quan hệ ánh xạ của hệ thống

Lược đồ cơ sở dữ liệu ánh xạ biểu diễn các mối quan hệ giữa các thực thể chính trong hệ thống quản lý học tập thông minh. Các thực thể như User (bao gồm Student, Teacher, và Manager), Course, LearningPath, Lesson, Module, và Quiz được tổ chức thành các bảng cơ sở dữ liệu, thể hiện mối quan hệ và tương tác của chúng.

Lược đồ cơ sở dữ liệu này không chỉ giúp tổ chức dữ liệu theo một cấu trúc rõ ràng mà còn giúp xác định các ràng buộc và quan hệ giữa các thực thể khác nhau trong hệ thống.

5.3.3 Class diagram

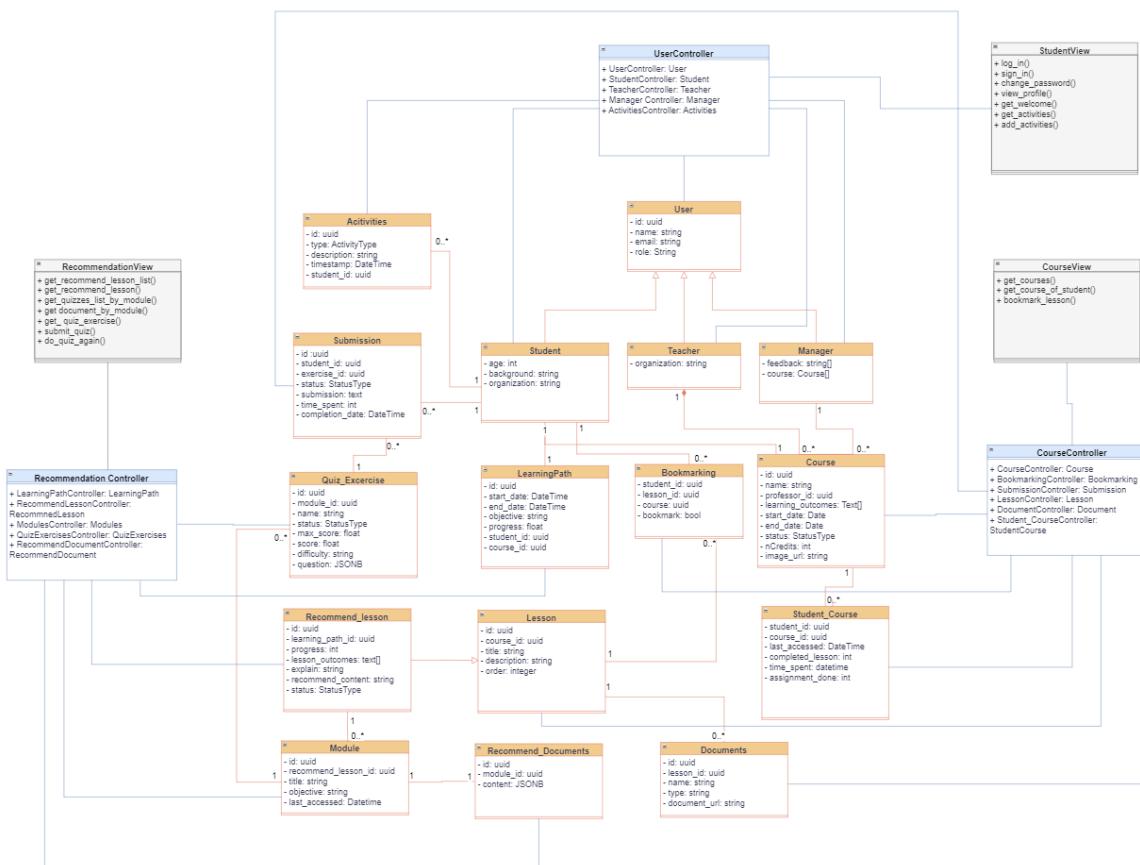


Figure 5.16: Class diagram hiện thực của hệ thống

Class diagram được thiết kế theo mô hình MVC mô tả một hệ thống học tập thông minh với ba vai trò chính là sinh viên, giáo viên và quản trị viên, tất cả đều kế thừa từ lớp người dùng (User). Mỗi người dùng có các chức năng cơ bản như đăng nhập, đăng xuất và cập nhật hồ sơ. Sinh viên chọn theo học khóa học của giảng viên, thực hiện bài tập, và theo dõi tiến độ học tập của mình. Lộ trình học tập bao gồm bài tập được cá nhân hóa cho từng sinh viên. Giáo viên chịu trách nhiệm tạo và quản lý các khóa học, bao gồm việc thêm bài tập, theo dõi kết quả học tập của sinh viên, và tạo ra các bộ câu hỏi kiểm tra. Quản trị viên có quyền quản lý toàn bộ hệ thống, bao gồm khóa học và người dùng, cũng như giải quyết các thắc mắc từ người dùng. AI được sử dụng để hỗ trợ quá trình học tập thông qua việc tự động tạo lộ trình học tập và bài tập. AI dựa vào dữ liệu từ các tài liệu học tập, câu hỏi và đáp án có sẵn, cũng như các bài tập lập trình để cá nhân hóa trải nghiệm học cho người dùng.

5.3.4 Sitemap

5.3.4.1 Sinh viên

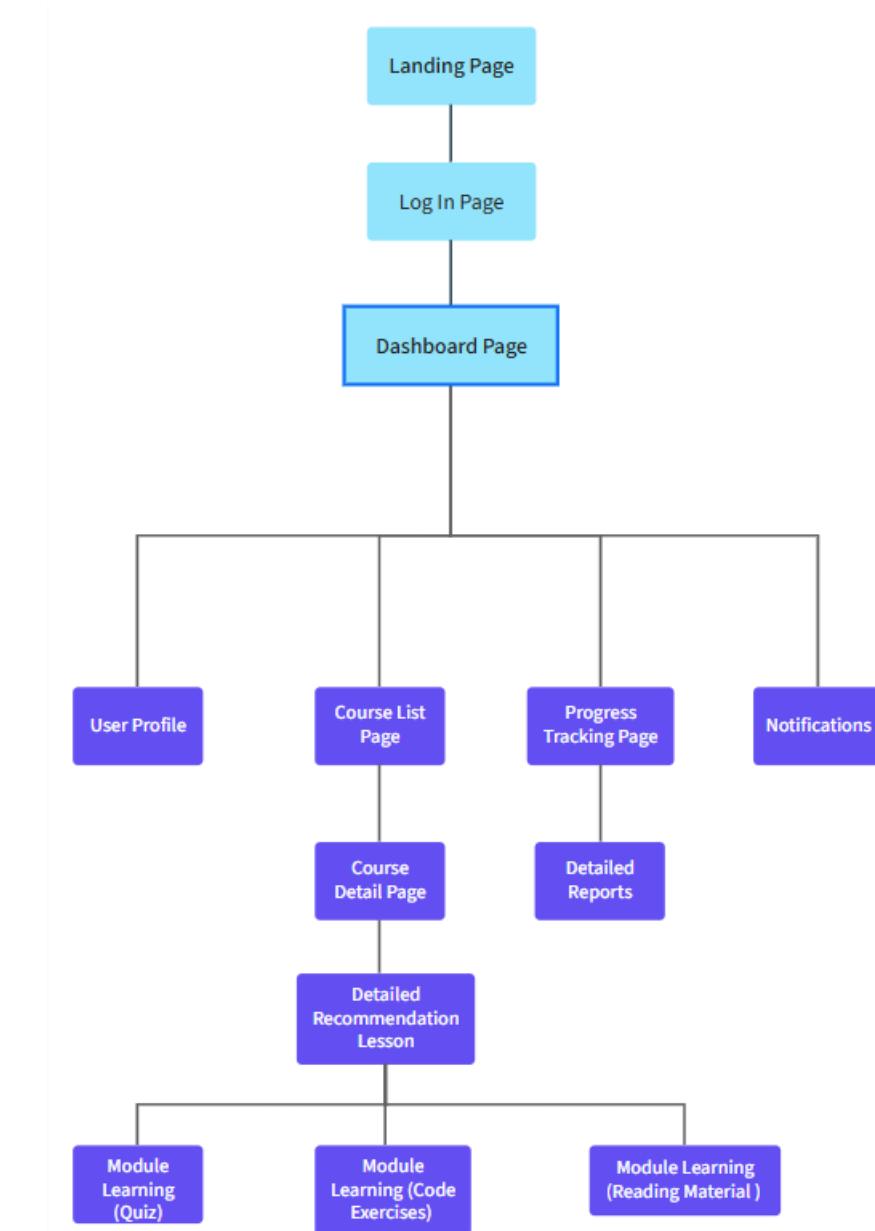


Figure 5.17: Sitemap cho đối tượng Sinh viên

Sơ đồ trang web này thể hiện cấu trúc tổng thể của ứng dụng web, hiển thị các trang chính và mối quan hệ giữa chúng, đặc biệt là đối với đối tượng người dùng **Sinh viên**. Ở phía trên, có "Landing Page", "Log In Page" và "Dashboard Page". Đây là các trang điều hướng chính cho người dùng.

Bên dưới, có một số phần khác:



- **User Profile:** Phần này chứa các trang liên quan đến tài khoản, thông tin cá nhân và cài đặt của sinh viên.
- **Course List Page:** Trang này hiển thị danh sách các khóa học mà sinh viên có thể truy cập.
- **Progress Tracking Page:** Trang này cho phép sinh viên theo dõi tiến độ và hiệu suất của họ trong các khóa học.
- **Notifications:** Phần này bao gồm trang chứa các thông báo cho sinh viên các cập nhật, tin nhắn hoặc cảnh báo quan trọng.
- **Course Detail Page:** Trang này cung cấp thông tin chi tiết về một khóa học cụ thể, như nội dung, mục tiêu và các tài nguyên khác.
- **Detailed Recommendation Lesson:** Trang này cung cấp các khóa học đề xuất học tập dựa trên tiến độ và mong muốn của người dùng.
- **Module Learning (Quiz), Module Learning (Code Exercises), and Module Learning (Reading Material):** Những phần này đại diện cho nội dung module của các bài học được đề xuất và chia thành 3 dạng bài tập như bài kiểm tra dạng quiz (multiple choices), bài tập lập trình thực hành và tài liệu đọc.

Nhìn chung, sơ đồ trang web này hiển thị các trang của hệ thống của đối tượng người dùng **Sinh viên** và cung cấp một cái nhìn tổng quan về cách các trang và chức năng liên quan được tổ chức và tương tác với nhau.

5.3.4.2 Giảng viên

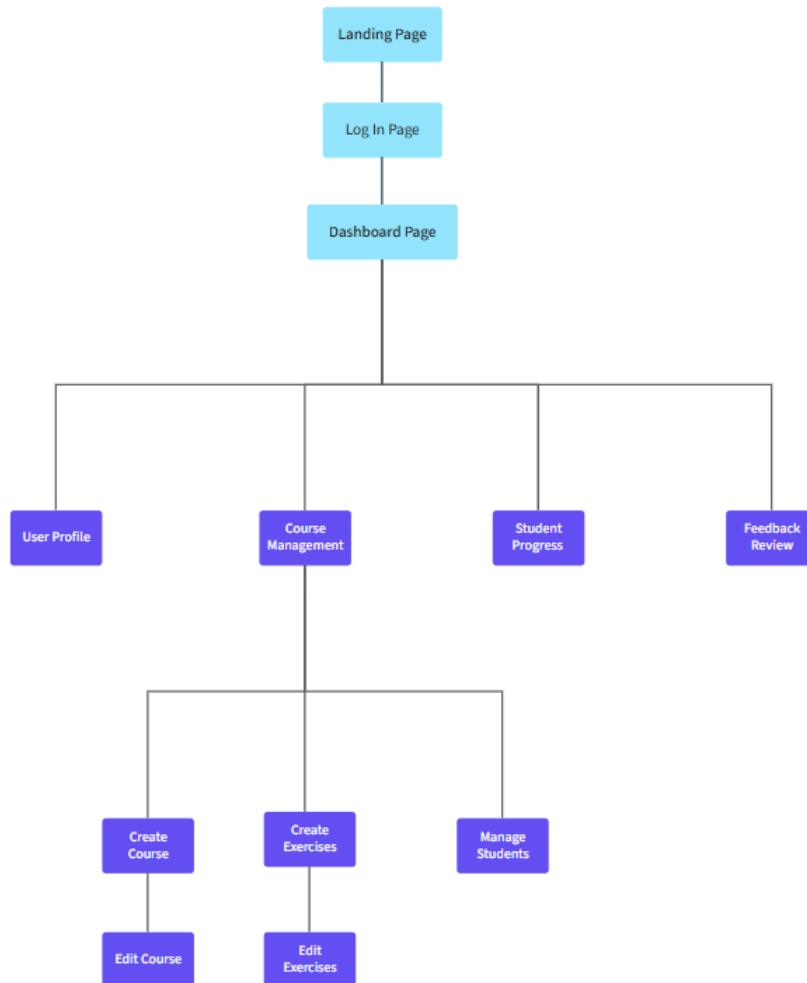


Figure 5.18: Sitemap cho đối tượng Giảng viên

Sơ đồ trang web này đại diện cho cấu trúc của ứng dụng web, tập trung vào các chức năng và trang dành riêng cho giảng viên.

Ở phía trên, vẫn có các trang điều hướng chính như "Landing Page", "Log In Page" và "Dashboard Page", giúp giảng viên dễ dàng truy cập vào các phần quan trọng của hệ thống.

Bên dưới, sơ đồ được chia thành bốn phần chính:

- **User Profile (Hồ sơ người dùng):** Phần này chứa các trang liên quan đến thông tin cá nhân, tài khoản và cài đặt của giảng viên. Giảng viên có thể cập nhật thông tin cá nhân và quản lý các tùy chọn hệ thống của mình.
- **Course Management (Quản lý khóa học):** Đây là phần quan trọng nhất với giảng viên, bao gồm các trang để tạo, chỉnh sửa, và quản lý các khóa học mà họ giảng dạy. Giảng viên có thể thêm các bài giảng, tài liệu học, và bài kiểm tra cho sinh viên.

- **Student Progress (Tiến độ học sinh):** Phần này giúp giảng viên theo dõi và giám sát tiến độ học tập của sinh viên trong các khóa học mà họ giảng dạy. Giảng viên có thể xem thông tin về kết quả học tập của sinh viên, bao gồm điểm kiểm tra, bài tập và các hoạt động học tập khác.
- **Feedback Management (Quản lý phản hồi):** Phần này cho phép giảng viên thu thập, quản lý và phân tích phản hồi từ sinh viên về khóa học, bài giảng, hoặc các hoạt động học tập. Đây là công cụ hữu ích để giảng viên đánh giá mức độ hiệu quả của phương pháp giảng dạy và cải thiện chất lượng khóa học.

Các trang con trong mỗi phần cung cấp các chức năng cụ thể hơn, chẳng hạn như chỉnh sửa nội dung khóa học, xem bảng điểm và thông tin phản hồi từ sinh viên.

Tổng quan: Sơ đồ trang web này cung cấp cái nhìn tổng quát về các chức năng chính mà giảng viên cần để quản lý khóa học, theo dõi tiến độ sinh viên và thu thập phản hồi. Cấu trúc đơn giản, dễ sử dụng, giúp giảng viên dễ dàng thực hiện các công việc quản lý và giảng dạy hiệu quả.

5.3.4.3 Manager

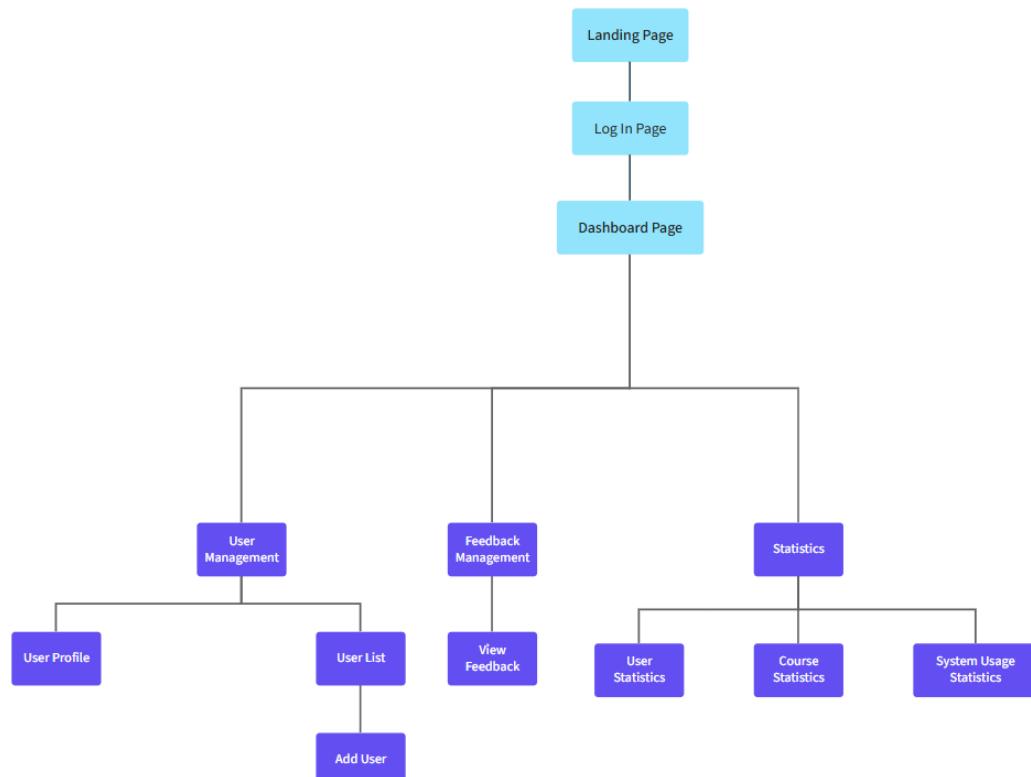


Figure 5.19: Sitemap cho đối tượng Quản trị viên

Sơ đồ trang web này cung cấp cái nhìn tổng quát về các chức năng chính mà Admin cần để quản lý các chức năng người dùng.



Ở phía trên, các trang điều hướng chính vẫn là "Landing Page", "Log In Page", và "Dashboard Page".

Các phần chính trong sơ đồ trang web này là:

- **User Management:** Phần này bao gồm các trang để quản lý hồ sơ cá nhân, cài đặt và thông tin tài khoản của người dùng.
- **Feedback Management:** Phần này tập trung vào các trang để cung cấp phản hồi, đánh giá và nhận xét liên quan đến trải nghiệm học tập của người dùng.
- **Statistics:** Phần này cung cấp các trang hiển thị các số liệu thống kê, chỉ số và dữ liệu liên quan đến tiến độ, hiệu suất và mức độ sử dụng hệ thống của người dùng.

Các trang cấp thấp hơn trong sơ đồ trang web này hướng đến học sinh, chẳng hạn như các trang để xem chi tiết khóa học, truy cập tài liệu học tập và theo dõi tiến độ cá nhân.

Sơ đồ trang web này gợi ý về một ứng dụng web được thiết kế để cung cấp cho học sinh một trải nghiệm học tập cá nhân hóa, với sự tập trung vào quản lý người dùng, phản hồi và các thông tin chi tiết dựa trên dữ liệu để hỗ trợ hành trình giáo dục của họ.

Tổng kết

Nhìn chung, ba sơ đồ trang web này thể hiện các góc nhìn và ưu tiên khác nhau trong thiết kế và chức năng của ứng dụng web, đáp ứng nhu cầu của các loại người dùng khác nhau, chẳng hạn như học sinh, giảng viên và quản trị viên.

5.4 Quy trình Đề xuất và Theo dõi Lộ trình Học tập Cá nhân hóa

5.4.1 Tổng quan

Hệ thống học tập trực tuyến thông minh được thiết kế để cung cấp trải nghiệm học tập cá nhân hóa, hỗ trợ học viên đạt được mục tiêu học tập một cách hiệu quả. Quy trình tận dụng trí tuệ nhân tạo (AI) để phân tích nhu cầu học viên, đề xuất nội dung phù hợp, đánh giá tiến độ, và điều chỉnh lộ trình học tập khi cần. Các chức năng chính bao gồm xây dựng lộ trình học tập, tạo bài kiểm tra tự động, theo dõi hiệu quả học tập, và tái tạo nội dung để giải quyết khó khăn.

Quy trình được chia thành bốn giai đoạn chính:

- Phân tích mục tiêu và tạo lộ trình học tập.



- Tạo bài kiểm tra tự động.
- Theo dõi và đánh giá tiến độ học tập.
- Điều chỉnh nội dung học tập dựa trên kết quả học tập.

5.4.2 Giai đoạn 1: Phân tích Mục tiêu và Tạo Lộ trình Học tập

Giai đoạn này tập trung vào việc hiểu mục tiêu học tập của học viên và xây dựng một kế hoạch học tập phù hợp, sử dụng AI để phân tích và đề xuất nội dung.

1. **Hiểu mục tiêu học tập:** Hệ thống thu thập mục tiêu của học viên, ví dụ: “nắm vững lập trình Python cơ bản” hoặc “cải thiện kỹ năng giải thuật”. AI phân tích mục tiêu để đảm bảo nó rõ ràng, phù hợp với khóa học, và khả thi trong thời gian quy định. Nếu mục tiêu có thời hạn cụ thể (như trước kỳ thi giữa kỳ), hệ thống xác định khoảng thời gian phù hợp.

2. **Prompt sử dụng:**

Phân tích mục tiêu học tập: "[goal]". Xác định
↳ tính hoàn chỉnh, phù hợp với khóa học
↳ "[course_name]", và khả thi dựa trên kết
↳ quả học tập mong đợi. Đề xuất dòng thời
↳ gian từ [start_date] đến [end_date]. Trả
↳ về JSON với start_date, end_date, và ghi
↳ chú xác thực.

Prompt này yêu cầu AI kiểm tra tính hợp lệ của mục tiêu và đề xuất thời gian thực hiện, đảm bảo lộ trình học tập phù hợp với lịch trình khóa học.

3. **Lựa chọn bài học phù hợp:** AI xem xét nội dung khóa học và chọn các bài học liên quan nhất đến mục tiêu. Các bài học được sắp xếp từ cơ bản đến nâng cao, kèm theo giải thích về lý do lựa chọn, giúp học viên dễ dàng tiếp cận kiến thức.

4. **Prompt sử dụng:**

Dựa trên mục tiêu "[goal]" và khóa học
↳ "[course_name]", chọn các bài học từ danh
↳ sách [lessons_data]. Sắp xếp theo thứ tự
↳ logic, ưu tiên bài học cơ bản nếu thời
↳ gian ngắn. Trả về danh sách bài học với
↳ ID, thứ tự, tiêu đề, và lý do chọn.



Prompt này hướng dẫn AI lọc và sắp xếp bài học dựa trên mục tiêu, đảm bảo tính liên quan và trình tự hợp lý.

5. **Xây dựng lộ trình chi tiết:** Hệ thống tạo lộ trình học tập bao gồm các bài học được chia thành các phần nhỏ (module), mỗi phần có nội dung lý thuyết, hướng dẫn thực hành, và tài liệu tham khảo. AI thiết kế lộ trình phù hợp với thời gian và tốc độ học của học viên.

6. Prompt sử dụng:

Tạo lộ trình học tập cho mục tiêu "[goal]" từ
↪ [start_date] đến [end_date]. Dựa trên bài
↪ học [selected_lessons], tạo danh sách bài
↪ học đề xuất với nội dung trọng tâm, giải
↪ thích lý do, và 2-3 module mỗi bài học.
↪ Trả về JSON với bài học và module chi
↪ tiết.

Prompt này yêu cầu AI tạo lộ trình chi tiết, bao gồm nội dung và module, để hỗ trợ học viên đạt mục tiêu một cách có hệ thống.

5.4.3 Giai đoạn 2: Tạo Bài Kiểm tra Tự động

Hệ thống sử dụng AI để tạo bài kiểm tra nhằm đánh giá sự hiểu biết của học viên dựa trên nội dung học tập.

- Xác định nội dung kiểm tra:** Hệ thống chọn một phần nội dung cụ thể (module) để tạo bài kiểm tra, tập trung vào các khái niệm và kỹ năng quan trọng.
- Tạo câu hỏi đa dạng:** AI tạo các câu hỏi với nhiều mức độ khó (dễ, trung bình, khó) và loại hình (trắc nghiệm đơn, đa lựa chọn, đúng/sai). Mỗi câu hỏi đi kèm giải thích chi tiết để học viên hiểu rõ đáp án.

3. Prompt sử dụng:

Tạo [count] câu hỏi [difficulty] cho module
↪ "[module_title]". Câu hỏi dựa trên mục
↪ tiêu [objectives] và nội dung
↪ "[recommended_content]". Bao gồm loại câu
↪ hỏi đa dạng (single_choice,
↪ multiple_choice, true_false), với giải
↪ thích chi tiết. Trả về JSON với câu hỏi,
↪ lựa chọn, đáp án, và điểm.



Prompt này chỉ đạo AI tạo câu hỏi phù hợp với nội dung module, đảm bảo đa dạng và có giá trị đánh giá cao.

4. **Thiết lập thông tin bài kiểm tra:** Hệ thống đặt thời gian làm bài và tổng điểm dựa trên số lượng câu hỏi, đảm bảo bài kiểm tra cân đối và khả thi.

5. Prompt sử dụng:

Thiết lập bài kiểm tra cho module

- ↪ "[module_title]". Dựa trên
- ↪ [questions_count] câu hỏi, tính thời gian
- ↪ làm bài (2 phút/câu, tối đa 60 phút) và
- ↪ tổng điểm. Trả về JSON với tên, mô tả,
- ↪ thời gian, và điểm tối đa.

Prompt này giúp AI cấu hình bài kiểm tra phù hợp với khối lượng nội dung, đảm bảo trải nghiệm học tập công bằng.

5.4.4 Giai đoạn 3: Theo dõi và Đánh giá Tiến độ Học tập

Hệ thống liên tục theo dõi quá trình học tập của học viên, sử dụng AI để đánh giá và đưa ra gợi ý cải thiện.

1. **Thu thập dữ liệu học tập:** Hệ thống ghi nhận thời gian học, tiến độ hoàn thành bài học, và kết quả bài kiểm tra, từ đó đánh giá mức độ nỗ lực và hiểu biết của học viên.

2. **Đánh giá theo tiêu chí:** AI phân tích tiến độ dựa trên ba yếu tố:

- *Kiến thức lý thuyết:* Mức độ hiểu các khái niệm quan trọng.
- *Kỹ năng thực hành:* Khả năng áp dụng kiến thức, như viết mã lập trình.
- *Mức độ nỗ lực:* Sự chăm chỉ, dựa trên thời gian học và số bài học đã bắt đầu.

3. Prompt sử dụng:

Đánh giá tiến độ học viên "[student_name]" với

- ↪ mục tiêu "[goal]". Dựa trên dữ liệu bài
- ↪ học [lessons_data], phân tích theo kiến
- ↪ thức lý thuyết, kỹ năng thực hành, và nỗ
- ↪ lực. Trả về JSON với báo cáo STAR, điểm
- ↪ mạnh, điểm cần cải thiện, và gợi ý chi
- ↪ tiết.



Prompt này yêu cầu AI tạo báo cáo toàn diện, giúp học viên hiểu rõ tình trạng học tập và nhận gợi ý cụ thể.

4. **Báo cáo chi tiết:** AI tạo báo cáo nêu rõ điểm mạnh (như nắm vững lý thuyết), điểm cần cải thiện (như kỹ năng thực hành còn yếu), và gợi ý (như tập trung vào bài tập thực hành). Báo cáo được trình bày rõ ràng để học viên dễ áp dụng.

5.4.5 Giai đoạn 4: Điều chỉnh Nội dung Học tập

Khi học viên gặp khó khăn, hệ thống sử dụng AI để phân tích và tái tạo nội dung học tập, giúp giải quyết vấn đề hiệu quả.

1. **Phân tích vấn đề:** AI xem xét các khó khăn, như hiểu sai khái niệm hoặc mắc lỗi thực hành, và xác định liệu học viên cần ôn lại nội dung trước đó.

2. Prompt sử dụng:

Phân tích vấn đề từ dữ liệu [issues_summary]
↪ cho bài học "[lesson_title]". Xác định các
↪ khó khăn chính (hiểu sai, lỗi thực hành)
↪ và đề xuất hành động (lặp lại, xem lại bài
↪ trước). Trả về JSON với danh sách vấn đề,
↪ mức độ nghiêm trọng, và hành động đề xuất.

Prompt này giúp AI xác định nguyên nhân khó khăn và đề xuất cách khắc phục phù hợp.

3. **Tái tạo nội dung:** AI tạo lại nội dung bài học, tập trung vào giải quyết các vấn đề đã xác định. Nội dung mới bao gồm ví dụ thực tế, hướng dẫn chi tiết, và tài liệu tham khảo phù hợp.

4. Prompt sử dụng:

Tái tạo nội dung cho bài học "[lesson_title]"
↪ dựa trên vấn đề [issues_analysis]. Cung
↪ cấp nội dung mới với giải thích chi tiết,
↪ ví dụ thực tế, và 2-3 module tập trung vào
↪ khó khăn. Trả về JSON với nội dung đề
↪ xuất, giải thích, và module.

Prompt này chỉ đạo AI thiết kế nội dung mới, đảm bảo phù hợp với nhu cầu cụ thể của học viên.

5. **Cập nhật lộ trình:** Nội dung mới được tích hợp vào lộ trình học tập, giúp học viên tiếp tục học mà không bị gián đoạn.



5.4.6 Kết luận

Quy trình đề xuất và theo dõi lộ trình học tập cá nhân hóa tận dụng trí tuệ nhân tạo để mang lại trải nghiệm học tập linh hoạt và hiệu quả. Các câu lệnh AI (prompt) được thiết kế cẩn thận để phân tích mục tiêu, tạo nội dung, đánh giá tiến độ, và điều chỉnh lộ trình, giúp học viên vượt qua khó khăn và đạt được mục tiêu học tập. Quy trình này không chỉ nâng cao chất lượng học tập mà còn đảm bảo sự hỗ trợ liên tục, phù hợp với từng cá nhân.

Chương 6

Hiện thực

6.1 Hiện thực hệ thống

Hệ thống học tập được triển khai nhằm hỗ trợ sinh viên và giảng viên trong việc tổ chức và quản lý việc học tập hiệu quả hơn. Với sự hỗ trợ của Mô hình Ngôn ngữ Lớn (LLMs), hệ thống có thể tự động tạo ra các bài học đề xuất dựa trên mục tiêu học tập của sinh viên và nội dung được cung cấp bởi giảng viên.

Hệ thống được thiết kế với kiến trúc phân tách rõ ràng giữa frontend và backend, sử dụng các công nghệ hiện đại để đảm bảo tính linh hoạt, dễ mở rộng và bảo trì.

Frontend

Frontend được xây dựng bằng VueJS, tích hợp Vuetify và Tailwind CSS để cung cấp giao diện người dùng trực quan, hiện đại và dễ sử dụng. Các chức năng chính bao gồm:

- Quản lý dashboard của sinh viên
- Hiển thị danh sách khóa học, chi tiết khóa học
- Gửi phản hồi và nhập mục tiêu học tập

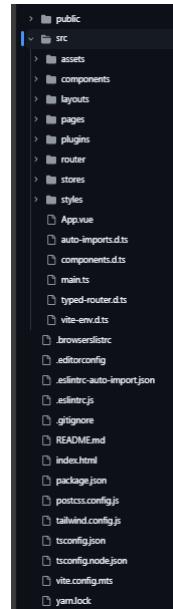


Figure 6.1: Cấu trúc Frontend

Hệ thống thiết kế (Design System)

Hệ thống thiết kế được triển khai với mục tiêu cung cấp một giao diện người dùng nhất quán, dễ sử dụng và dễ mở rộng. Hệ thống này bao gồm các thành phần chính sau:

Typography

Typography được xây dựng trên cơ sở font chữ Public Sans, với nhiều kích cỡ và độ đậm khác nhau để phục vụ các nhu cầu hiển thị khác nhau. Font chữ này được chọn vì dễ đọc, phù hợp với các thiết kế hiện đại và hỗ trợ đa ngôn ngữ. Các cấp độ bao gồm:

- **Heading:** Từ Heading 1 (48px, Bold) đến Heading 4 (24px, Semibold).
- **Body Text:** Các cấp độ từ Large đến Small, sử dụng độ đậm từ Regular đến Bold, kích thước từ 10px đến 20px.



Typography		
Public Sans		
Font Public Sans dễ đọc, dễ nhìn, phù hợp với thiết kế hiện đại và sạch sẽ, phù hợp với nhiều loại dự án, ngoài ra font cũng hỗ trợ nhiều ngôn ngữ		
Heading 1	Bold	48px
Heading 2	Bold	40px
Heading 3	Bold	32px
Heading 4	Semibold	24px
Body/Large 1	Bold	20px
Body/Large 2	Semibold	20px
Body/Large 3	Medium	20px
Body/Large 4	Regular	20px
Body/Base 1	Bold	16px
Body/Base 2	Semibold	16px
Body/Base 3	Medium	16px
Body/Base 4	Regular	16px
Body/Small 1	Bold	12px
Body/Small 2	Semibold	12px
Body/Small 3	Medium	12px
Body/Small 4	Regular	12px
Body/Ex-Small 1	Bold	10px
Body/Ex-Small 2	Semibold	10px
Body/Ex-Small 3	Medium	10px
Body/Ex-Small 4	Regular	10px

Figure 6.2: Hệ thống Typography sử dụng trong ứng dụng

Bảng màu (Color Palette)

Bảng màu được thiết kế để mang lại sự nhất quán và dễ nhận biết trong giao diện người dùng. Bảng màu chính bao gồm các màu **Primary**, **Secondary**, và các trạng thái như **Error**, **Warning**, **Neutral**, **Success**. Mỗi màu được chia thành các sắc độ từ nhạt (25, 50) đến đậm (900, 950), nhằm đảm bảo khả năng hiển thị tốt trong các trường hợp sử dụng khác nhau.

- **Primary:** Dùng cho các thành phần chính như nút bấm và tiêu đề.
- **Secondary:** Dùng cho các chi tiết phụ hoặc nhấn mạnh nội dung.
- **Error, Warning, Success:** Dùng để phản hồi trạng thái của hệ thống.

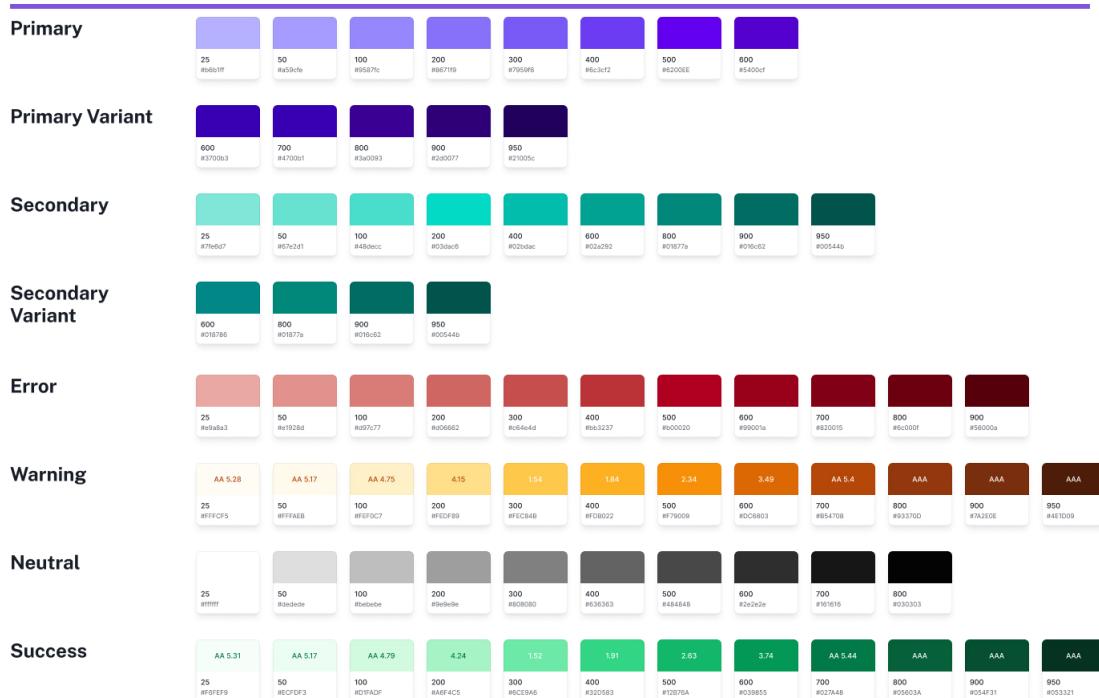


Figure 6.3: Bảng màu được sử dụng trong ứng dụng

Hệ thống này được tích hợp với Tailwind CSS, giúp tự động hóa việc áp dụng các quy tắc thiết kế (design tokens) vào mã nguồn. Điều này làm giảm thiểu lỗi thiết kế, đồng thời đảm bảo giao diện nhất quán trong toàn bộ ứng dụng.

Backend

Backend được phát triển với FastAPI, kết hợp với PostgreSQL để lưu trữ dữ liệu. Kiến trúc hệ thống được tổ chức theo các thành phần chính:

- Repository
- Provider
- Controller

Mỗi thành phần đảm nhiệm một vai trò cụ thể, giúp hệ thống trở nên dễ hiểu, có tính tổ chức cao và dễ bảo trì.

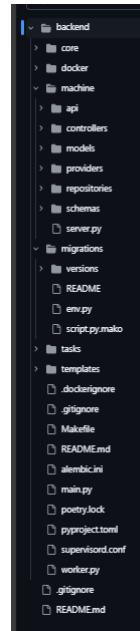


Figure 6.4: Cấu trúc Backend

6.2 Kiến trúc Backend

1. Repository

Repository chịu trách nhiệm quản lý các thao tác liên quan đến dữ liệu trong hệ thống. Đây là nơi thực hiện các logic liên quan đến việc truy vấn, lưu trữ và cập nhật dữ liệu từ cơ sở dữ liệu PostgreSQL.

Đặc điểm chính:

- Tách biệt logic truy xuất dữ liệu khỏi các thành phần khác
- Hỗ trợ các thao tác CRUD (Create, Read, Update, Delete)

Ví dụ: Repository sẽ xử lý các tác vụ như:

- Truy vấn thông tin sinh viên và giảng viên
- Cập nhật tiến độ học tập
- Xóa thông tin khóa học khi cần thiết

2. Provider

Provider chịu trách nhiệm xử lý các logic nghiệp vụ (business logic). Đây là nơi các phép toán và xử lý phức tạp được thực hiện trước khi trả kết quả cho Controller hoặc người dùng.



Đặc điểm chính:

- Không phụ thuộc vào giao diện hoặc nguồn dữ liệu cụ thể
- Tập trung vào logic nghiệp vụ và các phép toán quan trọng

Ví dụ: Provider sẽ:

- Phân tích mục tiêu học tập của sinh viên để đề xuất bài học
- Tự động tạo bài học dựa trên nội dung được giảng viên cung cấp

3. Controller

Controller là nơi tiếp nhận các yêu cầu từ người dùng thông qua API, sau đó phối hợp với Provider và Repository để thực hiện các chức năng tương ứng.

Đặc điểm chính:

- Giao tiếp trực tiếp với frontend thông qua các endpoint API
- Chuyển đổi dữ liệu giữa frontend và các thành phần backend

Ví dụ: Controller xử lý:

- Yêu cầu đăng nhập và xác thực người dùng
- Truy xuất danh sách khóa học và chi tiết khóa học
- Nhận mục tiêu học tập từ sinh viên và gửi đến Provider để xử lý

6.3 Quá trình triển khai hệ thống

1. Tạo cấu trúc cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế để lưu trữ các thông tin liên quan đến sinh viên, giảng viên, khóa học và bài học. Sơ đồ dưới đây minh họa cấu trúc cơ sở dữ liệu:

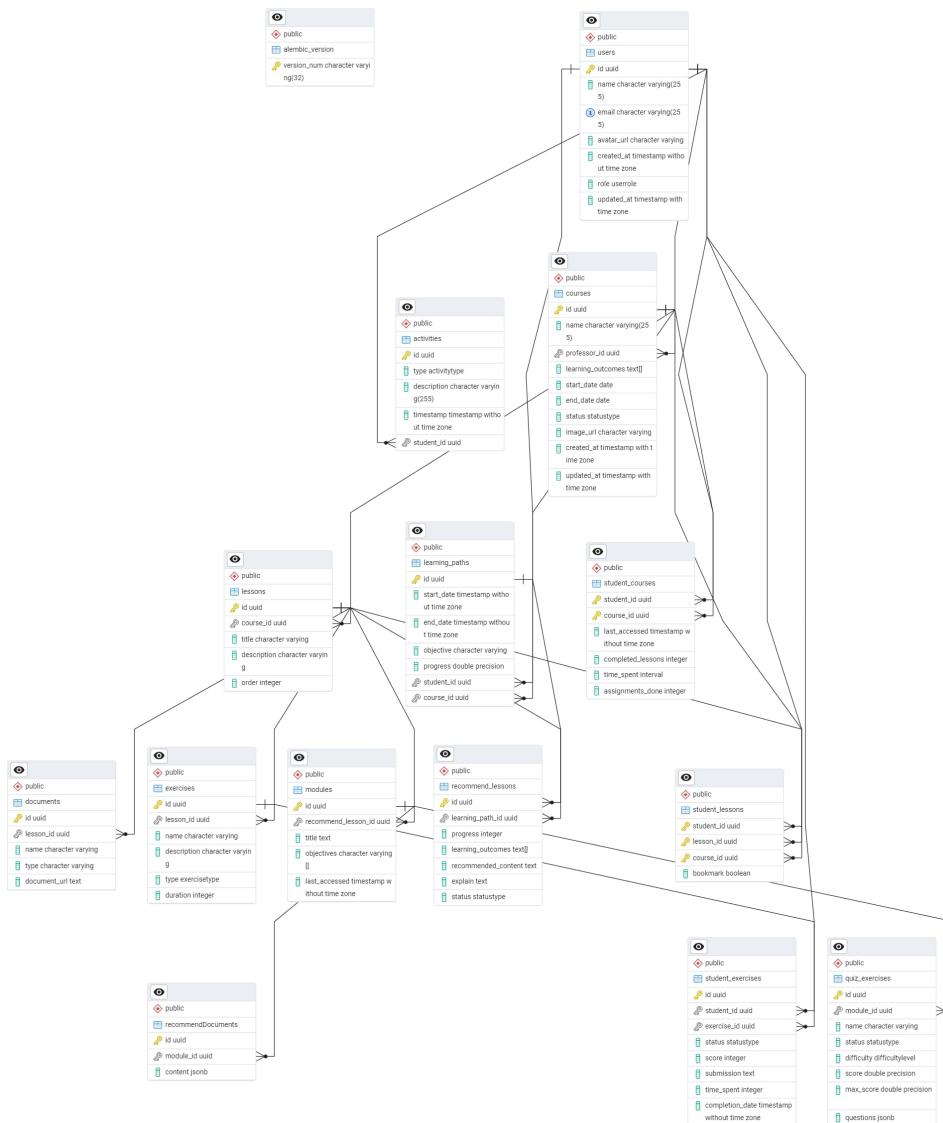


Figure 6.5: Cấu trúc Cơ sở dữ liệu

2. Xây dựng API

Các API hỗ trợ giao tiếp giữa frontend và backend, đảm bảo rằng người dùng có thể tương tác với hệ thống một cách hiệu quả. Các chức năng chính của API bao gồm:

- Quản lý khóa học
- Quản lý bài học
- Gửi và nhận phản hồi từ sinh viên

FastAPI cung cấp cơ chế xử lý yêu cầu nhanh chóng và an toàn, giúp cải thiện hiệu năng hệ thống.



3. Quản lý bài học và đề xuất học tập

Hệ thống sử dụng LLMs để phân tích dữ liệu từ sinh viên và giảng viên, từ đó đưa ra các bài học phù hợp. Điều này giúp cá nhân hóa việc học tập và tăng hiệu quả của hệ thống.

Cụ thể:

- Giảng viên cung cấp nội dung khóa học và mục tiêu học
- Sinh viên nhập mục tiêu học tập cá nhân
- LLMs xử lý và đề xuất các bài học tương thích

4. Triển khai hệ thống

Frontend: Frontend của hệ thống được triển khai trên nền tảng **Vercel**, mang đến một môi trường ổn định, tối ưu cho việc triển khai các ứng dụng web. Vercel cung cấp khả năng tự động build và deploy mỗi khi có thay đổi trong mã nguồn, giúp quá trình cập nhật trở nên nhanh chóng và mượt mà. Nhờ vào quy trình này, việc triển khai và duy trì hệ thống frontend trở nên đơn giản và hiệu quả.

Vercel không chỉ đảm bảo hiệu suất cao mà còn hỗ trợ các tính năng tối ưu hóa tự động, giúp giảm thời gian tải trang và nâng cao trải nghiệm người dùng. Với khả năng mở rộng linh hoạt, hệ thống có thể dễ dàng đáp ứng với nhu cầu tăng trưởng trong tương lai.

Link deploy frontend: <https://codemate-fe-murex.vercel.app/>

Backend: Hệ thống sử dụng nền tảng **Render** để triển khai backend. Render cung cấp một môi trường chạy ổn định, dễ mở rộng và bảo mật cao, giúp tối ưu hóa việc quản lý và triển khai ứng dụng. Với quy trình build và deploy tự động, việc cập nhật hệ thống trở nên mượt mà và hiệu quả, giảm thiểu các vấn đề phát sinh trong quá trình triển khai.

Để triển khai hệ thống, chúng tôi sử dụng công cụ **Poetry** để quản lý các gói phụ thuộc (dependencies) của Python. Điều này giúp đảm bảo rằng các thư viện và công cụ cần thiết luôn được cài đặt đúng phiên bản, đồng thời giảm thiểu các lỗi liên quan đến môi trường chạy.

Hệ thống được cấu hình để tự động khởi chạy khi có yêu cầu, đảm bảo hiệu suất cao và khả năng mở rộng dễ dàng khi cần thiết. Ngoài ra, Render cũng hỗ trợ giám sát và quản lý môi trường ứng dụng, giúp dễ dàng phát hiện và xử lý các sự cố.

Link deploy backend: <https://edumind.onrender.com>



6.4 Các chức năng đã được thực hiện

Dưới đây là các chức năng đã được thực hiện trong hệ thống học tập trực tuyến thông minh. Các chức năng này được phân chia theo từng module và vai trò của người dùng trong hệ thống.

6.4.1 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).

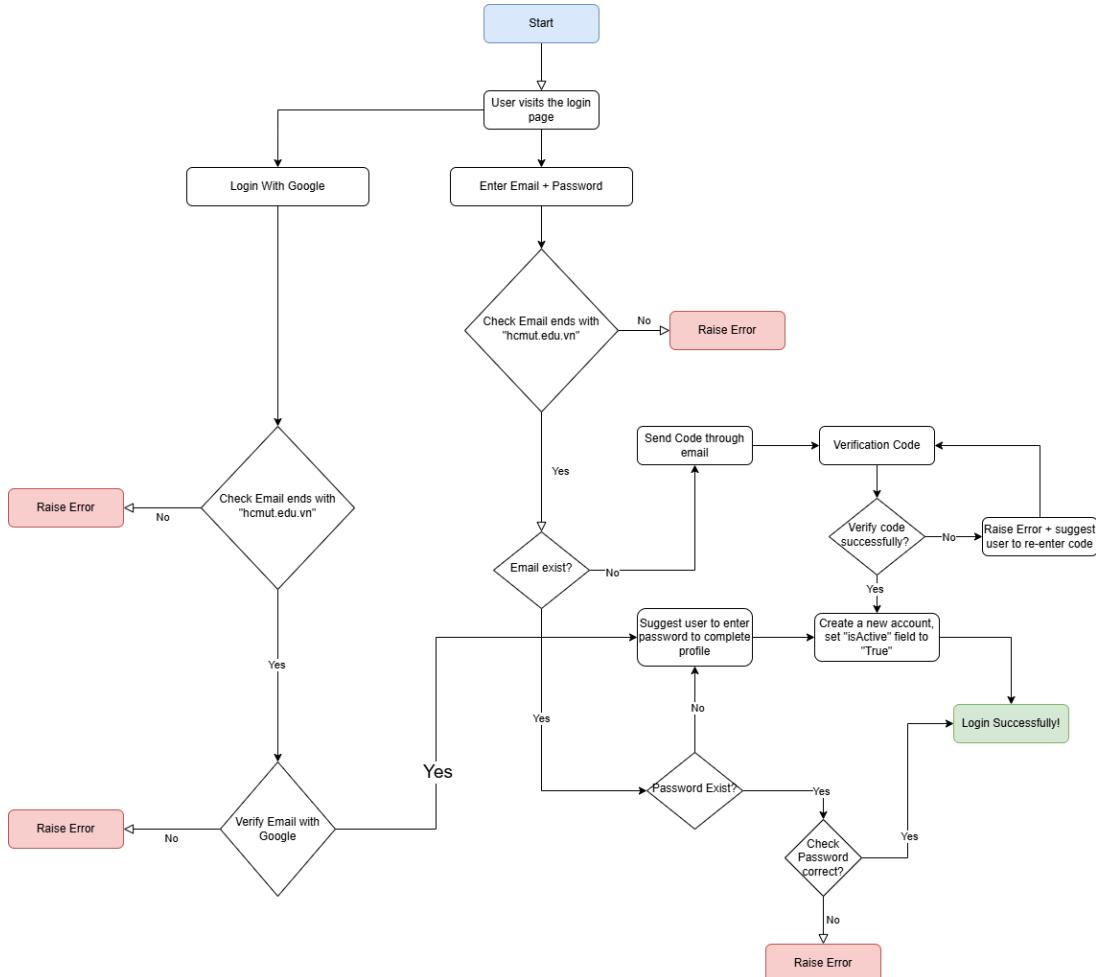


Figure 6.6: Flow Định danh tài khoản

6.4.2 Quản lý khóa học (Thành viên đảm nhiệm: Phạm Thị/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).

6.4.3 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).

6.4.4 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).



6.4.5 Quản lý lộ trình học tập (Thành viên đảm nhiệm: Phạm Thị/Tuấn Anh)

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).

6.4.6 Theo dõi tiến độ học tập (Thành viên đảm nhiệm: Phạm Thị/Tuấn Anh)

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).



6.4.7 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).

6.4.8 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).

6.4.9 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).

6.4.10 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).



6.5 Giới thiệu chi tiết về các chức năng chính của hệ thống

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. Mục nội dung này sẽ tập trung vào các chức năng chính như sinh ra lộ trình học tập, đề xuất mục tiêu học tập, đánh giá kết quả học tập, tạo bài kiểm tra, và các tính năng bổ sung khác.

Sau đây là các nguyên tắc cơ bản trong việc thiết lập mục tiêu học tập và phương pháp đánh giá mà hệ thống áp dụng:

6.5.1 Nguyên tắc SMART trong việc thiết lập mục tiêu

Nguyên tắc SMART là một khung tiêu chuẩn được áp dụng rộng rãi trong việc thiết lập mục tiêu hiệu quả. Trong hệ thống của chúng tôi, mỗi mục tiêu học tập được đề xuất đều tuân theo năm tiêu chí của nguyên tắc SMART:

- Specific (Cụ thể):** Mục tiêu phải rõ ràng và cụ thể, không mơ hồ. Ví dụ: “Hiểu và áp dụng được các thuật toán sắp xếp” thay vì “Học về thuật toán”.
- Measurable (Đo lường được):** Mục tiêu phải có thể đo lường được để đánh giá tiến độ. Ví dụ: “Đạt điểm 8/10 trong bài kiểm tra về cấu trúc dữ liệu” hoặc “Hoàn thành 5 bài tập lập trình về đệ quy”.
- Achievable (Khả thi):** Mục tiêu phải thực tế và có thể đạt được với các nguồn lực và thời gian hiện có. Mục tiêu quá tham vọng có thể dẫn đến nản chí.
- Relevant (Liên quan):** Mục tiêu phải phù hợp với nội dung khóa học và kết quả học tập mong đợi. Ví dụ, đặt mục tiêu học sâu về machine learning sẽ không phù hợp cho một khóa học cơ bản về Python.
- Time-bound (Giới hạn thời gian):** Mục tiêu phải có thời hạn cụ thể để tạo động lực và tính cấp bách. Ví dụ: “Hoàn thành trước kỳ thi giữa kỳ” hoặc “Trong vòng 3 tuần”.

Trong API /generate-student-goals, AI sẽ tạo ra các mục tiêu học tập tuân thủ nguyên tắc SMART cho ba cấp độ học tập khác nhau: Struggling (khó khăn), Average (trung bình), và Advanced (nâng cao). Mỗi mục tiêu đều được giới hạn dưới 200 ký tự để đảm bảo tính súc tích và dễ nhớ.

6.5.2 Phương pháp Rubric-Based Assessment

Rubric-Based Assessment (Đánh giá dựa trên Rubric) là phương pháp đánh giá sử dụng các tiêu chí cụ thể và mức độ đạt được cho mỗi tiêu chí. Phương pháp này mang lại nhiều lợi ích:



- **Tính khách quan:** Đánh giá dựa trên các tiêu chí rõ ràng, giảm thiểu tính chủ quan.
- **Tính nhất quán:** Đảm bảo các học sinh được đánh giá theo cùng một tiêu chuẩn.
- **Tính minh bạch:** Học sinh hiểu rõ họ được đánh giá như thế nào và cần cải thiện điều gì.
- **Phản hồi có cấu trúc:** Cung cấp phản hồi chi tiết theo từng tiêu chí.

Trong hệ thống của chúng tôi, Rubric được áp dụng để đánh giá học sinh theo ba tiêu chí chính:

1. **Kiến thức lý thuyết (Theoretical Knowledge):** Đánh giá mức độ hiểu biết về lý thuyết dựa trên kiến thức trong các bài học liên quan đến khái niệm lý thuyết.
2. **Kỹ năng thực hành (Coding Skills):** Đánh giá kỹ năng lập trình dựa trên việc hoàn thành các bài tập mã hóa. Nếu khóa học không bao gồm lập trình, tiêu chí này sẽ được thay thế bằng một kỹ năng thực hành khác phù hợp.
3. **Nỗ lực (Effort):** Đánh giá sự cố gắng dựa trên thời gian học tập (time_spent), số lượng bài học đã bắt đầu (status), và tiến độ tổng thể.

Mỗi tiêu chí được đánh giá theo bốn mức độ: Excellent (Xuất sắc), Good (Tốt), Average (Trung bình), và Poor (Kém), với các miêu tả cụ thể cho mỗi mức độ.

6.5.3 Phương pháp đánh giá STAR

STAR là một phương pháp đánh giá cấu trúc được sử dụng rộng rãi trong giáo dục và phát triển nghề nghiệp. Trong hệ thống của chúng tôi, phương pháp STAR được áp dụng để đánh giá tiến độ học tập của học sinh một cách toàn diện. STAR bao gồm bốn thành phần:

- **Situation (Tình huống):** Mô tả bối cảnh học tập của học sinh, bao gồm tên, khóa học, và mối liên hệ với mục tiêu học tập.
- **Task (Nhiệm vụ):** Nêu rõ mục tiêu học tập của học sinh, nhấn mạnh tầm quan trọng của việc đạt được mục tiêu trước thời hạn (ví dụ: trước kỳ thi giữa kỳ).
- **Action (Hành động):** Liệt kê chi tiết các hành động học sinh đã thực hiện cho mỗi tiêu chí đánh giá (Kiến thức lý thuyết, Kỹ năng thực hành, Nỗ lực), dựa trên dữ liệu bài học (progress, time_spent, objectives).
- **Result (Kết quả):** Đánh giá tiến độ tổng thể so với mục tiêu, phân tích theo từng tiêu chí, với nhận xét rõ ràng (on track/needs effort/at risk) và tham chiếu đến ngày hiện tại.



API /student/{courseId}/assessment sử dụng phương pháp STAR kết hợp với Rubric để tạo ra báo cáo đánh giá toàn diện cho học sinh, giúp họ hiểu rõ tiến độ học tập và những điểm cần cải thiện.

6.6 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:  
    datetime, ...):  
2     if len(goal) < 10:  
3         raise ApplicationException(message="Goal is too short...")  
4     prompt = f"..."  
5     chunking_manager = ChunkingManager(  
6         provider="gemini",  
7         gemini_model_name="gemini-2.0-flash-lite",  
8         max_tokens_per_chunk=15000,  
9         temperature=0.7,  
10        max_output_tokens=8000  
11    )  
12    response = chunking_manager.call_llm_api(prompt, "You are an  
        expert in goal validation and timeline analysis.")
```

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     "learning_path_end_date": "2025-05-20",
```



```
4     "learning_path_objective": "Achieve goal...",
5     "recommend_lessons": [
6         {
7             "lesson_id": "123",
8             "recommended_content": "What to focus on in this lesson...",
9             "explain": "Why this lesson supports the goal...",
10            "order": 1,
11            "number_of_modules": 2
12        }
13    ],
14    "modules": [
15        {
16            "title": "Module Title",
17            "objectives": ["Objective 1", "Objective 2"],
18            "reading_material": {...}
19        }
20    ]
21 }
```

6.7 Đề xuất mục tiêu học tập

Đây là tính năng mới được thêm vào để giúp học sinh xác định mục tiêu học tập phù hợp, sử dụng endpoint /generate-student-goals/{course_id}. Các bước thực hiện:

- Thu thập thông tin khóa học:** Hệ thống lấy dữ liệu về tên khóa học, mục tiêu học tập, và tổng quan các bài học.
- Sinh mục tiêu theo trình độ:** AI tạo ra các mục tiêu học tập phù hợp cho ba cấp độ khác nhau: Struggling (khó khăn), Average (trung bình), và Advanced (nâng cao), mỗi cấp độ có 2-3 mục tiêu cụ thể.
- Đảm bảo chất lượng mục tiêu:** Mỗi mục tiêu đều tuân theo nguyên tắc SMART (Specific, Measurable, Achievable, Relevant, Time-bound) và không vượt quá 200 ký tự.

Kết quả là một JSON chứa các mục tiêu học tập được đề xuất, ví dụ:

```
1 {
2     "suggested_goals": [
3         {
4             "proficiency_level": "Struggling",
5             "goal": "Master basic Python syntax and write simple programs
with 70% accuracy by midterm",
6         }
7     ]
8 }
```



```
6         "explanation": "This goal focuses on building fundamental
7             programming skills...",
8         "key_lessons": ["Introduction to Python", "Basic Data Types"]
9     },
10    {
11        "proficiency_level": "Average",
12        "goal": "Implement and optimize sorting algorithms with 80%
13             efficiency by midterm",
14        "explanation": "This goal challenges the student to apply
15             theoretical concepts...",
16        "key_lessons": ["Algorithm Complexity", "Sorting Algorithms"]
17    }
18 ]
19 }
```

6.8 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.

6.8.1 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.
- Xử lý văn bản dài:** Sử dụng ChunkingManager để xử lý các prompt và phản hồi dài, đảm bảo không bị lỗi do giới hạn token.
- Kết quả đầu ra:** Một JSON chứa thông tin đánh giá, ví dụ:

```
1 {
2     "student_assessment": {
3         "student_info": {
4             "name": "Nguyen Van A",
5             "email": "nguyenvana@example.com"
6         },
7         "learning_goal": "Master Python programming",
```



```
8     "assessment_date": "2025-03-31",
9     "assessment_summary": {
10        "situation": "Nguyen Van A is taking a Python course...",
11        "task": "The task of Nguyen Van A is to master Python
12          programming...",
13        "action": {
14            "theoretical_knowledge": "...",
15            "coding_skills": "...",
16            "effort": "..."
17        },
18        "result": "With 75% progress as of 2025-03-31, Nguyen Van
19          A is on track..."
20    },
21    "progress_review": {
22        "strengths": "...",
23        "areas_to_note": "..."
24    },
25    "advice": {
26        "theoretical_knowledge": "...",
27        "coding_skills": "...",
28        "effort": "..."
29    }
}
```

6.8.2 Theo dõi và phân tích tiến độ học tập

Hệ thống cung cấp API /progress_tracking/student/recommend_lessons/{recommend_lesson_id} để cập nhật thời gian học tập và theo dõi tiến độ của học sinh:

- **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     hours, minutes, seconds = map(int, new_time_spent_str.split(':'))
3     new_time_delta = timedelta(hours=hours, minutes=minutes, seconds=
4         seconds)
5     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 }
```

6.8.3 Đánh giá kết quả học tập (Monitor Study Progress)

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

- **Kiểm tra tiến độ:** Hệ thống kiểm tra xem học sinh đã đạt đủ tiến độ cần thiết (ít nhất 80%) để được đánh giá.
- **Phân tích vấn đề:** Dựa trên issues_summary từ cơ sở dữ liệu, hàm analyze_issues sử dụng AI để 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     "needs_review_prior": false,
5     "issues_analysis": {
6         "significant_issues": [
7             {
8                 "type": "concept_misunderstanding",
9                 "frequency": 8,
10                "description": "Struggling to grasp recursion base
11                           cases.",
12                "severity": "high",
13                "impact_on_goals": "This significantly hinders..."
14            }
15        ],
16        "total_issues_count": 13,
17        "increasing_issues": ["recursion application"],
18        "most_frequent_type": "concept_misunderstanding"
19    },
20    "recommendations": [
21    ]
```



```
21         "action": "repeat",
22         "reason": "Significant difficulties with recursion ...",
23         "details": "Review 'Advanced Recursion'"
24     }
25 ]
26 }
```

6.9 Tạo bài kiểm tra (Generate Quiz)

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

- **Tập hợp dữ liệu:** Hệ thống lấy thông tin từ module, bài học (lesson), nội dung được đề xuất (recommended content), và tài liệu liên quan (documents).
- **Tạo câu hỏi theo phân phối độ khó:** 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 qua AIToolProvider. Ví dụ mã nguồn:

```
1 async def generate_quiz(request: GenerateQuizRequest, ...):
2     # Fetch module details
3     module = await modules_controller.modules_repository.first(
4         where_= [Modules.id == request.module_id]
5     )
6
7     # Generate questions for each difficulty level
8     for difficulty in difficulties:
9         prompt = f"Generate EXACTLY {questions_needed} {difficulty_"
10            name} difficulty questions..."
11
12         response = llm.invoke(question_request.content)
```

- **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     "description": "Assessment covering key concepts from Introduction to
```

Python",



```
5     "time_limit": 30,
6     "max_score": 100,
7     "module_id": "def456",
8     "questions": [
9         {
10             "id": "q123",
11             "question_text": "What is a list in Python?",
12             "question_type": "single_choice",
13             "options": ["A mutable sequence type", "An immutable sequence
14                         type", "A mapping type", "A set type"],
15             "correct_answer": ["A mutable sequence type"],
16             "difficulty": "easy",
17             "explanation": "Lists in Python are mutable sequences...",
18             "points": 5
19         }
20     ]
21 }
```

6.10 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. Các bước:

- Phân tích vấn đề:** Dựa trên kết quả phân tích từ monitor_study_progress, hệ thống xác định các khía cạnh cần cải thiện (ví dụ: hiểu sai khái niệm, lỗi mã).
- Đánh giá tình huống:** Kiểm tra xem học sinh có cần xem lại bài học trước đó không (needs_review_prior) và thu thập thông tin liên quan nếu cần.
- 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(
2     recommend_lesson_id: UUID,
3     analysis_result: dict,
4     prior_recommend_lessons_data: List[dict],
5     ...
6 ):
7     prompt = f"""
8         ## Lesson Content Regeneration Task
9         ### Context
10        - Lesson Title: "{lesson.title}"
11        - Analysis Result: {json.dumps(analysis_result, indent=2)}
12    """
13 
```



```
13     updated_content = chunking_manager.call_llm_api(prompt, ...)
```

- **Cập nhật dữ liệu:** Xóa mô-đun cũ, tạo mô-đun mới và đánh dấu trạng thái bài học là "new" với tiến độ 0

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

```
1 {
2     "recommended_content": "Updated content addressing recursion issues...",
3         ,
4     "explain": "Explanation of how this content resolves the issues...",
5     "start_date": "2025-04-06",
6     "end_date": "2025-04-20",
7     "duration_notes": "Updated notes on timeline...",
8     "modules": [
9         {
10             "title": "Recursion Basics Revisited",
11             "objectives": ["Understand base cases", "Apply recursive patterns"],
12             "reading_material": { ... }
13         }
14     ]
15 }
```

6.11 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:

6.11.1 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.

6.11.2 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.



6.11.3 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.

6.11.4 AI Module (LLM – Large Language Model):

Các tác vụ như gọi 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.

6.11.5 Hệ thống lưu trữ tài liệu:

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

6.12 Sơ đồ cơ sở dữ liệu

Figure 6.7: Sơ đồ ERD của hệ thống

6.12.1 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:



6.12.1.1 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.

6.12.1.2 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.

6.12.1.3 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.

6.12.1.4 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.

6.12.1.5 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.

6.12.2 Mối 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)

6.12.3 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.

6.13 Kết luận

Hệ thống học tập trực tuyến thông minh đã được phát triển với mục tiêu mang lại trải nghiệm học tập cá nhân hóa, hiệu quả và linh hoạt cho sinh viên ngành công nghệ thông tin và những người có nhu cầu học lập trình. Thông qua việc tích hợp các mô hình ngôn ngữ lớn (LLM) và các tính năng tiên tiến như gia sư AI, hệ thống tạo bài tập tự động, và lộ trình học tập cá nhân hóa, hệ thống không chỉ hỗ trợ người học trong việc tiếp thu kiến thức mà còn giúp họ phát triển kỹ năng lập trình một cách thực tế và sâu sắc.

Các module đã được triển khai, bao gồm quản lý tài khoản, khóa học, bài học, bài tập, lộ trình học tập, tiến độ học tập, phản hồi, người dùng, nhật ký đăng nhập và dashboard, đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng được đề ra. Những tính năng này cho phép sinh viên, giảng viên và quản trị viên tương tác một cách hiệu quả, đồng thời đảm bảo tính bảo mật, khả năng mở rộng và hiệu năng cao của hệ thống.

Trong tương lai, hệ thống có thể được cải tiến thêm bằng cách tích hợp các công nghệ mới, mở rộng các tính năng hỗ trợ đa ngôn ngữ, hoặc tối ưu hóa thuật toán để xuất để nâng cao độ chính xác và phù hợp của các tài nguyên học tập. Với nền tảng vững chắc đã được xây dựng, hệ thống hứa hẹn sẽ tiếp tục là một công cụ hữu ích, đóng góp tích cực vào quá trình học tập và giảng dạy trong lĩnh vực công nghệ thông tin.

Chương 7

Giao diện người dùng

7.1 Landing Page

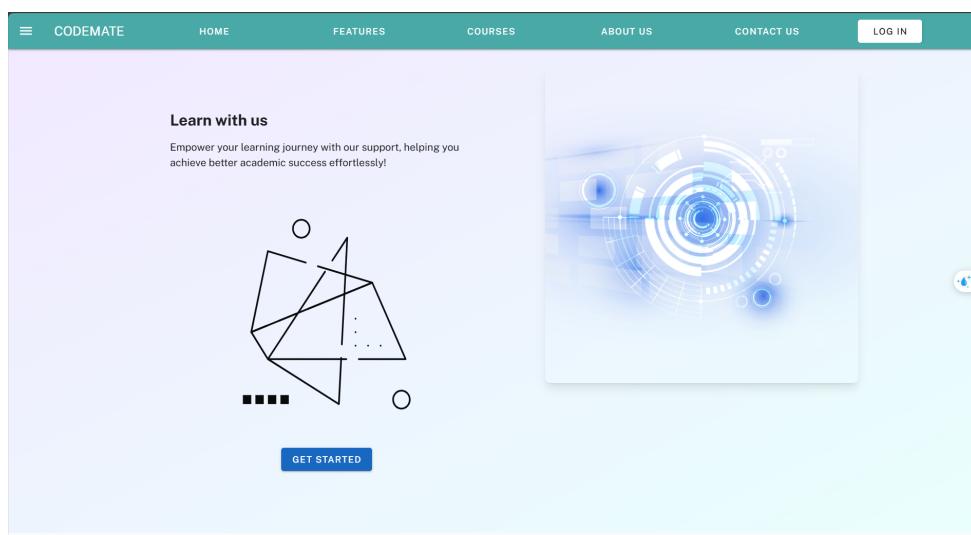


Figure 7.1: Giao diện trang chủ của hệ thống

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.

7.2 Đăng nhập

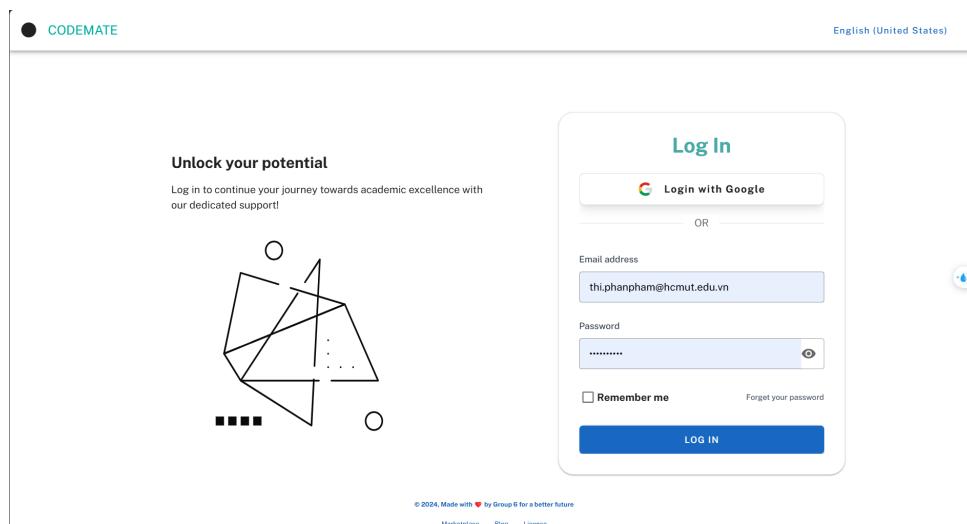


Figure 7.2: Giao diện đăng nhập của hệ thống

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).

7.3 Profile

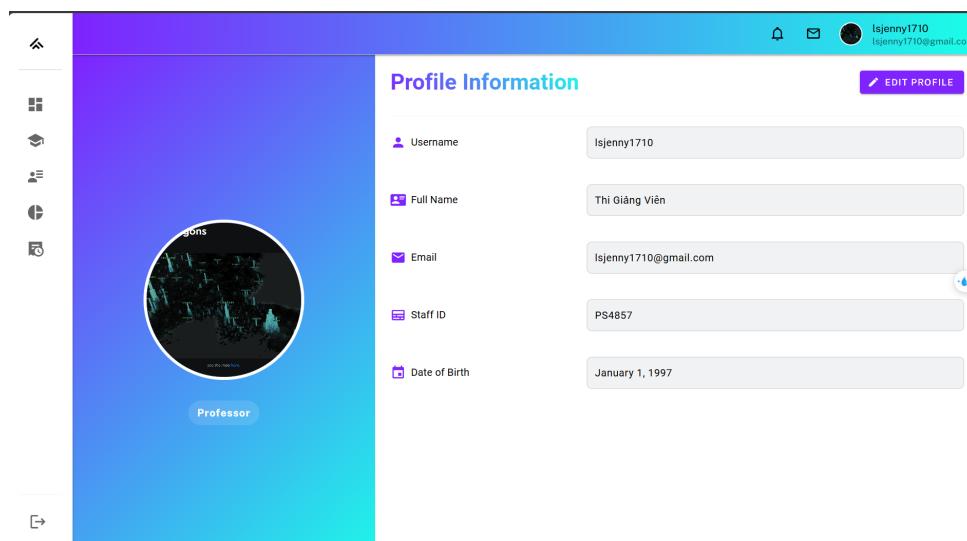


Figure 7.3: Giao diện thông tin cá nhân của sinh viên

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.



7.4 Sinh viên

7.4.1 Giao diện chính

The screenshot shows the student's main dashboard. At the top, it says "Welcome back, thi.phanphan". Below that is a "System Feedback" section with a "Provide System Feedback" button. On the right, there's a "Recent Activity" sidebar listing several recent interactions. The main area displays a "Courses" section with a table showing course details like name, members, and progress. One course, "Data Structures and Algorithms", is highlighted.

Figure 7.4: 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.

7.4.2 Quản lý khóa học

The screenshot shows the student's course management interface. It features a "Recent Courses" section with a card for "Data Structures and Algorithms" showing learning outcomes and a progress bar at 20%. Below that is a "My Courses" section with a search bar and a list of courses. One course, "Data Structures and Algorithms", is shown in more detail with its learning outcomes, professor information, and a "View Course" button.

Figure 7.5: 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.

7.4.3 Chi tiết khóa học

The screenshot shows a learning path for 'Data Structures and Algorithms' (CO2003). The goal is 'Successfully implement and apply basic sorting algorithms (bubble, insertion, selection) in code by midterm with 70% accuracy.' The timeline includes:

- Jun 01: Course introduction, Recursion and Complexity of Algorithms (Completed)
- Jun 15: List (P1): Array List and Singly Linked List (In Progress)
- Jun 29: List(P2)-Doubly Linked List, Stack and Queue (Upcoming)
- Jul 13: Sorting Algorithms (Upcoming)
- Aug 10: Tree Concepts and Binary Tree (Upcoming)

The course details for [241] Data Structures and Algorithms (CO2003) are shown: 20% completion, in progress from 01/06/2025 to 30/11/2025, taught by Professor: lsjenny1710. The 'DESCRIPTION', 'LESSONS', and 'EXERCISES' tabs are visible. A sidebar offers to 'Update Your Learning Path'.

Figure 7.6: 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 lô trình học thì có thể xem lô trình học của mình tại đây.

7.4.4 Chi tiết bài học

The screenshot shows the 'Course introduction, Recursion and Complexity of Algorithms' lesson. It includes:

- Start Date: 2025-06-01
- End Date: 2025-06-14
- Two weeks are allocated to this lesson to allow ample time for understanding the core concepts of algorithm analysis and complexity.

Under 'Recommend' and 'Explain' sections, there is descriptive text about the lesson's purpose and content. The 'Learning Outcomes' section lists:

- Understand the concept of recursion and its role in problem-solving, including base cases, recursive calls, and the stack mechanism.
- Design and implement recursive algorithms for problems like factorial computation, Fibonacci sequence, tree traversal, and more.
- Evaluate the time and space complexity of recursive algorithms using Big-O notation.
- Apply divide-and-conquer techniques in recursion for efficient problem-solving.
- Analyze recursive vs. iterative approaches in algorithm design, recognizing when recursion offers an optimal solution.

Figure 7.7: 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.

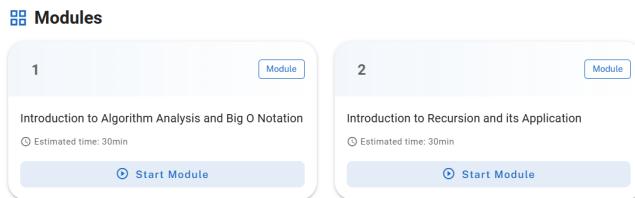


Figure 7.8: Danh sách các modules học tập của bài học

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.

7.4.5 Chi tiết module

Figure 7.9: 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.



7.4.6 Chi tiết bài quiz

The screenshot shows a quiz titled "Introduction to Algorithm Analysis and Big O Notation" with 7 questions to complete. The student has completed 14.285714285714285/70 points. Question 1 asks: "Which of the following time complexities is considered the most efficient?" with options: $O(n^2)$, $O(\log n)$ (selected), $O(n)$, and $O(1)$. The explanation states: " $O(1)$ represents constant time complexity, meaning the algorithm's runtime is independent of the input size."

Figure 7.10: 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

The screenshot shows a list of quizzes. One quiz titled "Introduction to Algorithm Analysis and Big O Notation" is listed with a status of "completed". The score is 14.285714285714285. Buttons for "View Again" and "Start Quiz" are visible.

Figure 7.11: 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.

7.5 Giảng viên

7.5.1 Giao diện chính

Trang Dashboard giảng viên cung cấp một giao diện trực quan giúp điều hướng đến những trang khác một cách dễ dàng gồm các thành phần chính như sau:

- Thông kê giảng viên:
 - Hiển thị tổng số khóa học mà giảng viên quản lý: chuyển đến trang danh sách khóa



học

- Tổng số sinh viên đang theo học trong các khóa học: chuyển đến trang tiến độ sinh viên trong khóa học
- Hiển thị tổng số bài giảng mà giảng viên đã thiết lập.
- Hiển thị tổng số bài tập đã được giao cho sinh viên: chuyển đến trang tạo bài tập.

The screenshot shows the dashboard for a professor. At the top, it says "Welcome back, nguyentruongtuananh03". Below this, there's a "System Feedback" section with a "Provide System Feedback" button. The main dashboard area has four cards: "Total Courses" (3), "Total Lessons" (1), "Total Exercises" (16), and "Total Students" (1). To the right, there's a sidebar titled "Upcoming Events" listing four events with due dates:

- Data Structure: Stack & Queue (Due: 12/02/2025 14:44:00)
- Introduction to AI: Search Algorithms (Due: 22/02/2025 00:00:00)
- Introduction to AI: Machine Learning Basics (Due: 27/02/2025 00:00:00)
- Introduction to AI: Neural Networks & Deep Learning (Due: 28/02/2025 00:00:00)

Figure 7.12: Dashboard Professor

- Upcoming Events: hiển thị danh sách các sự kiện quan trọng sắp diễn ra mà giảng viên cần quan tâm (Hạn nộp bài tập, kỳ thi, sự kiện liên quan đến học viên)
- Gửi Feedback cho hệ thống: Giảng viên có thể gửi feedback về hệ thống qua một modal.

The modal window is titled "System Feedback". It contains fields for "Feedback Title", "Feedback Category", and "Your Feedback". Below these is a 5-star rating scale with three stars highlighted. There's a checkbox for "Allow follow-up" and a note about it. At the bottom are "CANCEL" and "SUBMIT FEEDBACK" buttons.

Figure 7.13: Modal Feedback

7.5.2 Quản lý khóa học

Trang My Course cung cấp danh sách các khóa học mà giảng viên phụ trách. Mỗi khóa học gồm các thông tin chi tiết như: tên khóa học, mã khóa học, sinh viên, learning outcomes, trạng thái khóa học, thời gian khóa học.



The screenshot shows a list of three courses under the 'My Course' section:

- Data Structures**: 01/02/2025 to 15/06/2025. Professor: nguyentruongtuananh03. Learning Outcomes: Master data structures, Implement algorithms. Status: new. Buttons: VIEW COURSE.
- Introduction to AI**: 20/01/2025 to 30/05/2025. Professor: nguyentruongtuananh03. Learning Outcomes: Understand AI basics, Learn about machine learning. Status: new. Buttons: VIEW COURSE.
- Web Development**: 01/03/2025 to 15/07/2025. Professor: nguyentruongtuananh03. Learning Outcomes: Understand frontend technologies. Status: new. Buttons: VIEW COURSE.

Figure 7.14: Professor: My Course

7.5.3 Quản lý phản hồi của sinh viên

Tính năng quản lý phản hồi giúp giảng viên theo dõi và phân tích ý kiến của sinh viên về khóa học.

- Giảng viên có thể chọn một khóa học cụ thể từ danh sách các khóa học mà họ đang giảng dạy.
- Hệ thống sẽ hiển thị ra danh sách feedback gồm các thông tin: người gửi, tiêu đề phản hồi, mô tả, phân loại, rating.

The screenshot shows a table of feedback for the 'Introduction to AI' course:

Student	Feedback Title	Feedback Content	Category	Rating	Created At	Resolved At	Status
anh.nguyenbk70 anh.nguyenbk70@hcmut.edu.vn	Great course structure	The course organization is excellent.	DATA STRUCTURES	★★★★★	26/01/2025 12:00:00	26/01/2025 12:00:00	PENDING
anh.nguyenbk70 anh.nguyenbk70@hcmut.edu.vn	Needs better examples	The examples provided are not very relatable.	DATA STRUCTURES	★★★★☆	26/01/2025 13:00:00	26/01/2025 13:00:00	PENDING
anh.nguyenbk70 anh.nguyenbk70@hcmut.edu.vn	Excellent instructor	The instructor explains concepts clearly and keeps the course engaging.	DATA STRUCTURES	★★★★★	26/01/2025 12:00:00	26/01/2025 12:00:00	PENDING
anh.nguyenbk70 anh.nguyenbk70@hcmut.edu.vn	More practical exercises needed	The course could benefit from more hands-on exercises to reinforce learning.	DATA STRUCTURES	★★★★☆	26/01/2025 13:00:00	26/01/2025 13:00:00	PENDING
anh.nguyenbk70 anh.nguyenbk70@hcmut.edu.vn	Outdated content	Some of the course materials seem outdated and need updates.	DATA STRUCTURES	★★★☆☆	26/01/2025 12:00:00	26/01/2025 12:00:00	PENDING
anh.nguyenbk70 anh.nguyenbk70@hcmut.edu.vn	Helpful resources	The additional resources and reading materials are very useful.	DATA STRUCTURES	★★★★★	26/01/2025 13:00:00	26/01/2025 13:00:00	PENDING

Figure 7.15: Professor: Feedback Management

7.5.4 Quản lý tiến độ sinh viên

Trang tiến độ này theo dõi tiến độ sinh viên giúp giảng viên đánh giá hiệu suất học tập của từng sinh viên trong khóa học

- Tổng quan lớp học: điểm cao nhất, thấp nhất, trung bình điểm

- Tiết độ cá nhân:
 - Learning path cá nhân của sinh viên trong khóa học.
 - Điểm số của từng bài kiểm tra và bài tập.
- Chi tiết bài tập: Điểm của sinh viên trong bài tập, lời giải, lịch sử nộp bài và số lần thử lại.

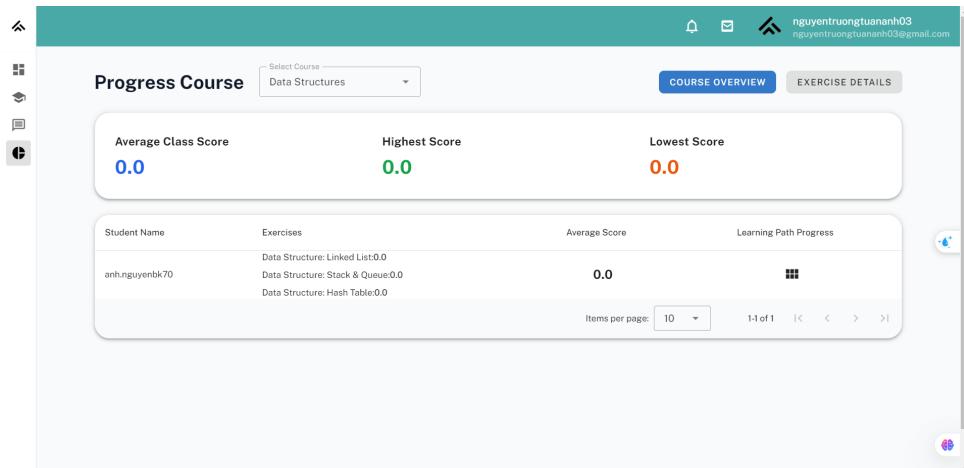


Figure 7.16: Professor: Progress Student

7.6 Admin

7.6.1 Giao diện chính

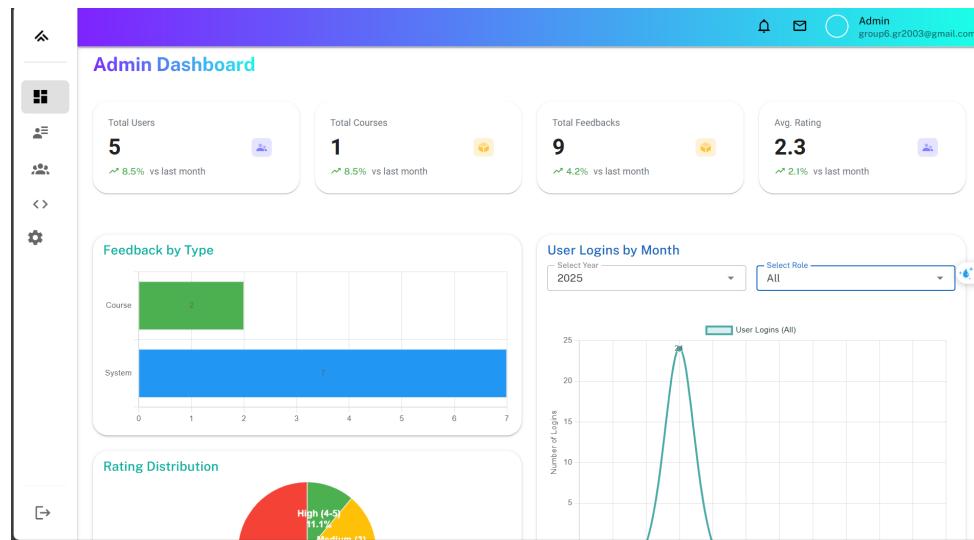


Figure 7.17: Giao diện chính của quản trị viên

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.



7.6.2 Quản lý khóa học

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	

Figure 7.18: 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.

7.6.3 Quản lý người dùng

Name	Email	Role	Status	Actions
thi.phanpham	thi.phanpham@hcmut.edu.vn	Student	Active	
John	johndoe.2@gmail.com	Student	Inactive	
Jane	janesmith.3@gmail.com	Professor	Active	
lsjenny1710	lsjenny1710@gmail.com	Professor	Active	
Admin	group6.gr2003@gmail.com	Admin	Active	

Figure 7.19: 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.

7.6.4 Quản lý phản hồi

The screenshot shows the 'System Feedback Management' page. At the top, there are filters for 'Date Range' (Start Date and End Date) and 'Classification' (Category and Status). Below the filters is a table with columns: User, Feedback Title, Feedback Description, Category, Rating, Created At, Resolved At, Status, and Actions. There are four entries in the table:

User	Feedback Title	Feedback Description	Category	Rating	Created At	Resolved At	Status	Actions
thi.phanphan	Feedback for Tree Concepts and Binary Tree. sefsfas	AFASASAS	PERFORMANCE	★★★★★	02/04/2025 12:18:23	-	PENDING	█
thi.phanphan	Feedback for List (P1): Array List and Singly Linked List Lesson: Test recommend	expect	OTHER	★★★★★	02/04/2025 21:06:10	-	PENDING	█
thi.phanphan	The avatar of course displays so ugly	You should improve the avatar of course	FEATURE REQUEST	★★★★★	29/03/2025 23:12:33	03/04/2025 12:04:47	RESOLVED	█
	Feedback for Course introduction Revision	ran wo coe tho						

Figure 7.20: 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.

7.6.5 Thêm người dùng

The screenshot shows the 'User Management' page. On the left sidebar, there are links for Admin Dashboard, Feedback Management, User Management, Course Management, Settings, Add User, and Add Course. At the bottom left is a Logout link. At the top right, it shows 'Admin group6.gr2003@gmail.com' and a 'Admin Portal' button. The main area has two sections: 'Create Single User' and 'Bulk User Import'. The 'Create Single User' section has fields for Role (Student), ID (MSSV), Email Address, and Full Name, with 'Create User' and 'Reset' buttons. The 'Bulk User Import' section has a 'Template Format' table (Fullname, Email, MSSV) with sample data (John Doe, john.doe@example.com, 12345678), a 'Download Template' button, an 'Upload Users' section (Select Role: Student), and a file upload area ('Drag and drop Excel file here or Upload Excel').

Figure 7.21: 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.



7.6.6 Thêm khóa học

The screenshot shows the 'Course Management' interface. On the left, a sidebar menu includes 'Admin Dashboard', 'Feedback Management', 'User Management', 'Course Management' (which is selected), 'Settings', 'Add User', and 'Add Course' (highlighted in grey). Below the sidebar is a 'Logout' button and the URL 'localhost:5173/add-course'. The main area has a purple header bar with the title 'Course Management'. The left panel contains a 'Create Course' form with fields for 'Course Name*', 'Course ID*', 'Credit Number*', 'Professor*', 'Semester Number*', 'Class Name*', 'Start Date*', and 'End Date*'. The right panel contains an 'Import Student IDs' section with a 'Template Format' example ('StudentID: 12345678'), a 'Download Template' button, and a 'Upload Student IDs' section with a file upload input field ('Drag and drop Excel file here or Upload Excel'). The top right corner shows the user 'Admin group6.gr/2003@gmail.com'.

Figure 7.22: 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 8

Kiểm thử hệ thống

8.0.1 Tổng quan về kiểm thử

Mục đích của kiểm thử trong hệ thống: Đảm bảo rằng các tính năng hoạt động đúng, hệ thống không có lỗi và có thể chịu được tải trong quá trình sử dụng thực tế.

Loại kiểm thử thực hiện:

- **Kiểm thử chức năng:** Đảm bảo các chức năng của hệ thống hoạt động đúng
- **Kiểm thử hiệu suất:** Đảm bảo hệ thống có thể xử lý nhiều yêu cầu đồng thời
- **Kiểm thử bảo mật:** Đảm bảo hệ thống bảo vệ tốt các dữ liệu người dùng
- **Kiểm thử tích hợp:** Kiểm tra sự hoạt động đồng bộ của các module
- **Kiểm thử hồi quy:** Đảm bảo những thay đổi mới không làm hỏng các tính năng cũ

8.0.2 Mô tả chung về các kịch bản kiểm thử

Mỗi kịch bản kiểm thử được tổ chức và theo dõi theo bảng sau đây. Các mục chính trong mỗi kịch bản kiểm thử bao gồm:

- **TEST SCENARIO:** Tên kịch bản kiểm thử
- **Screen name/Function name:** Tên màn hình hoặc tên chức năng cần kiểm thử
- **Test case code:** Mã số của test case
- **Number of passed test cases (P):** Số lượng test case đã thành công
- **Number of failed test cases (F):** Số lượng test case không thành công

- **Number of test cases under pending (PE):** Số lượng test case đang chờ xử lý
- **Number of unexecuted test cases:** Số lượng test case chưa được thực hiện
- **Total number of test cases:** Tổng số test case cần kiểm thử

8.0.2.1 Cấu trúc của một kịch bản kiểm thử

- **Test case code:** Mã của test case, giúp phân biệt các test case khác nhau
- **Testing Purpose:** Mục đích của việc kiểm thử
- **Steps:** Các bước thực hiện kiểm thử
- **Expected outcome:** Kết quả mong đợi sau khi thực hiện các bước kiểm thử
- **Browser compatibility testing:** Kiểm tra tính tương thích với các trình duyệt phổ biến
- **Test status report:** Báo cáo tình trạng của các lần kiểm thử

8.0.2.2 Ví dụ về một kịch bản kiểm thử trong bảng Excel

TEST SCENARIO *												
Test case code	Testing Purpose	Steps	Expected outcome	Google Chrome	Firefox	Microsoft Edge	Cốc Cốc	Curent	Error code	Note		
TEST STATUS REPORT												
		1. Function description: POST /activities/ 2. Test executor: Phan Phien Thi 3. Test execution day: 27/11/2024										
AA_Add_Activity_1	Verify adding a valid activity	1. Create a valid AddActivityRequest with required fields (student_id, type, description). 2. Send a POST request to /activities/ with the valid data.	The activity is added successfully, and the response status is 200 with message "Successfully added the activity."									
AA_Add_Activity_2	Verify error when missing student_id	1. Create an AddActivityRequest without student_id. 2. Send a POST request to /activities/ with the invalid data.	The API returns a 400 BadRequest error with message "Student ID, activity type, and activity description are required."	P F PE								

Figure 8.1: Ví dụ về một kịch bản kiểm thử

8.0.2.3 Mục tiêu của các kịch bản kiểm thử

- **Kiểm thử chức năng:** Đảm bảo các API hoạt động đúng với các yêu cầu và dữ liệu đầu vào hợp lệ hoặc không hợp lệ.
- **Kiểm thử tương thích trình duyệt:** Đảm bảo hệ thống hoạt động ổn định trên các trình duyệt phổ biến.
- **Kiểm thử độ tin cậy:** Kiểm tra xem các API có thể xử lý được các tình huống thực tế và đưa ra kết quả chính xác.



8.1 Các API cần kiểm thử

Các APIs được kiểm thử sẽ là các APIs có liên quan đến các chức năng chính của hệ thống như:

- Sinh ra lộ trình học tập cho sinh viên
- Gợi ý mục tiêu học tập cho sinh viên đối với một khóa học cụ thể
- Sinh ra bài tập quiz tự động cho sinh viên
- Đánh giá kết quả học tập của sinh viên
- Tái tạo nội dung bài học cho sinh viên

8.2 Kết quả kiểm thử API

8.2.1 Kiểm thử phương thức

8.2.2 Kết quả tổng hợp của các testcase

8.3 Hạn chế

Chương 9

Đánh giá hệ thống

9.1 Đánh giá tính năng và hiệu quả của hệ thống

Hệ thống hỗ trợ học tập bằng cách sử dụng LLMs (Language Models) để tự động tạo ra các bài học đề xuất cho sinh viên dựa trên các bài học có sẵn từ giảng viên. Các tính năng đã triển khai, như trang dashboard, course list, course detail, và các trang liên quan đến việc đề xuất lessons đã bước đầu cho phép sinh viên tương tác hiệu quả với hệ thống học tập.

Trong quá trình phát triển, nhóm đã hoàn thành việc xây dựng giao diện và chức năng cơ bản, bao gồm:

- **Dashboard:** Cung cấp một cái nhìn tổng quan về các hoạt động học tập gần đây, tình trạng tiến độ học tập của sinh viên và các khóa học tham gia.
- **Course List và Course Detail:** Cho phép sinh viên theo dõi các khóa học, xem chi tiết các bài học, và tiếp cận với các tài liệu học tập.
- **Đề xuất Lessons:** Dựa trên mục tiêu học tập mà sinh viên đặt ra, hệ thống sẽ tự động tìm kiếm các bài học phù hợp từ các khóa học đã đăng tải, giúp sinh viên tiếp cận đúng bài học cần thiết.
- **Tạo Quiz và Code Exercises:** Hệ thống có khả năng tự động tạo ra các bài quiz và bài tập lập trình dựa trên nội dung bài học, giúp sinh viên thực hành và củng cố kiến thức.
- **Tạo tài liệu đọc:** Hệ thống có khả năng tự động tạo ra các tài liệu đọc dựa trên nội dung bài học, giúp sinh viên có thêm tài liệu tham khảo.
- **Giảng viên:** Hệ thống cho phép giảng viên cập nhật learning outcomes cho các khóa học, giúp sinh viên có thể theo dõi được các mục tiêu học tập mà họ cần đạt được. Ngoài



ra, giảng viên sẽ là người tải lên các tài liệu học tập để hệ thống có thể sử dụng như 1 knowledge base. Giảng viên cũng có thể theo dõi tiến độ học tập của sinh viên và đánh giá kết quả học tập của họ.

- **Quản trị viên:** Hệ thống cho phép quản trị viên theo dõi và quản lý các hoạt động của sinh viên và giảng viên trong hệ thống. Quản trị viên có thể xem báo cáo về các phản hồi của người dùng đối với hệ thống và thêm tài khoản cho sinh viên hoặc giảng viên. Quản trị viên cũng có thể tạo các khóa học cho giảng viên theo từng học kì.

Các mô-đun học tập (quiz, code exercises, reading material) đã được phát triển cho phép sinh viên tham gia vào việc học theo dạng quiz, code exercises, và tài liệu đọc. Hệ thống cũng đã ghi lại quá trình học tập của sinh viên, tạo điều kiện để tiến hành báo cáo và đánh giá kết quả học tập.



9.2 Đánh giá hệ thống với các mô hình ngôn ngữ hiện có

Mục đích của cuộc thử nghiệm đó là đánh giá chất lượng của lời giải thích của từng mô hình ngôn ngữ hồn dành cho người học. Mô hình tiến hành thử nghiệm:

- GPT 4o
- GPT 4o Mini
- Google Gemini Pro 1.5

Dữ liệu đầu vào cho mỗi mô hình là một yêu cầu từ người dùng về việc cung cấp các tài nguyên học tập hữu ích cho việc phát triển kỹ năng. Dưới đây là kết quả trả về từ mỗi mô hình:

- GPT 4o mini:

```
1      GPT 4o mini:  
2      Response body  
3      {  
4          "final_response": [  
5              {  
6                  "id": "b0f66328-862d-4e4d-9f5a-2bdee93490b8",  
7                  "code": "COD001",  
8                  "description": "A classic beginner coding challenge.",  
9                  "explanation": "This coding challenge is a great  
10                 starting point for you as it will help you practice  
11                 fundamental programming skills. Engaging with such  
12                 challenges will enhance your problem-solving  
13                 abilities, which are essential for backend  
14                 development."  
15             },  
16             {  
17                 "id": "d3f325a9-8103-4a7b-bb26-7de45b607f9c",  
18                 "code": "COD002",  
19                 "description": "A simple exercise to reverse a given  
20                     string.",  
21                 "explanation": "This exercise will allow you to work on  
22                     string manipulation, a critical skill in backend  
23                     development. By mastering this concept, you will be  
24                     better prepared to handle data processing tasks that  
25                     are common in backend applications."  
26             },  
27             {  
28                 "id": "59427089-44c4-4916-8886-52a9f6f488b1",  
29                 "code": "CRS003",  
30             }  
31         }  
32     }  
33 }
```



```
20      "description": "A comprehensive introduction to web
21          development.",
22      "explanation": "This course will provide you with a
23          solid foundation in web development, covering
24          essential concepts that are crucial for backend
25          development. Understanding how the web works will
26          empower you to create effective backend solutions."
27  },
28  {
29      "id": "1144a6d7-a357-4e4c-b4f5-315afde72a67",
30      "code": "PA001",
31      "description": "A project to build a calculator using
32          Python.",
33      "explanation": "Working on this project will give you
34          hands-on experience with Python, a popular backend
35          language. It will help you apply your knowledge in a
36          practical setting, enhancing your coding skills and
            preparing you for real-world backend tasks."
37  },
38  {
39      "id": "7a32d80f-5313-425f-9bf2-c271084384d8",
40      "code": "PA003",
41      "description": "A project to create and deploy a
42          personal portfolio website.",
43      "explanation": "This project will not only allow you to
44          practice your web development skills but also give
45          you experience in deploying a web application.
46          Understanding deployment is vital for backend
47          developers, as it will help you manage and maintain
48          applications effectively."
49  }
50 ]
51 }
```

- GPT 4o

```
1  {
2      GPT 4o:
3      Response body
4      "final_response": [
5          {
6              "id": "59427089-44c4-4916-8886-52a9f6f488b1",
7              "code": "CRS003",
8              "description": "A comprehensive introduction to web
```



```
    development.",
9 "explanation": "Since you're starting your journey towards
  backend development, having a solid understanding of web
  development is essential. This course will provide you
  with the foundational knowledge necessary to understand
  how web applications are structured, which is crucial
  before diving into backend specifics."
10 },
11 {
12   "id": "CRS001",
13   "code": "6e906d7e-8f93-4dd8-8890-fa4b42cfcc8c",
14   "description": "A beginner-friendly course to learn Python
     programming.",
15   "explanation": "Python is a popular language for backend
     development due to its simplicity and effectiveness. This
     course is perfect for you as it will introduce you to
     programming concepts using Python, which is often used
     for backend tasks such as server scripting and data
     management."
16 },
17 {
18   "id": "PA003",
19   "code": "7a32d80f-5313-425f-9bf2-c271084384d8",
20   "description": "A project to create and deploy a personal
     portfolio website.",
21   "explanation": "Engaging in a project like this will give
     you hands-on experience with web hosting and deployment,
     skills that are vital for a backend developer. You'll
     learn how to manage servers and deploy applications,
     which are key components of backend development."
22 },
23 {
24   "id": "LES002",
25   "code": "1bb2500f-4275-4672-9089-fd9b382809a6",
26   "description": "A lesson focused on understanding binary
     search.",
27   "explanation": "Understanding algorithms is a critical skill
     for backend development, especially for optimizing data
     retrieval processes. This lesson will help you grasp the
     concept of binary search, enhancing your problem-solving
     skills and efficiency in handling data."
28 },
29 {
30   "id": "PA001",
31   "code": "1144a6d7-a357-4e4c-b4f5-315afde72a67",
32   "description": "A project to build a calculator using Python
```



```
33     .",
34     "explanation": "This project will reinforce your Python
35     programming skills by applying them to a practical task.
36     It will help you understand how to implement logic and
         functionality , which is essential for developing backend
         applications ."
34     }
35   ]
36 }
```

- Gemini Pro 1.5

```
1  {
2    Gemini Pro 1.5:
3    Response body
4    "final_response": [
5      {
6        "id": "b0f66328-862d-4e4d-9f5a-2bdee93490b8",
7        "code": "COD001",
8        "description": "A classic beginner coding challenge.",
9        "explanation": "This coding challenge will help you build
          foundational programming skills. It's a great way to
          start practicing problem-solving , a crucial skill for any
          programmer, especially in backend development where you'
          ll constantly be tackling new challenges."
10      },
11      {
12        "id": "d3f325a9-8103-4a7b-bb26-7de45b607f9c",
13        "code": "COD002",
14        "description": "A simple exercise to reverse a given string .
          ",
15        "explanation": "This exercise focuses on string manipulation
          , a core skill for backend development. Practicing this
          will improve your ability to process and handle data
          effectively , a common task in backend applications."
16      },
17      {
18        "id": "CRS001",
19        "code": "CRS001",
20        "description": "A beginner-friendly course to learn Python
          programming .",
21        "explanation": "Since you're just starting , this course
          provides a foundational understanding of Python , a widely
          -used language for backend development. Learning the
```



```
        basics will lay a solid groundwork for more advanced
        concepts and projects later on."
22    },
23    {
24        "id": "LES001",
25        "code": "LES001",
26        "description": "An in-depth lesson on Python lists.",
27        "explanation": "This lesson will build your proficiency in
        working with Python lists, a fundamental data structure.
        Understanding list manipulation is essential for many
        backend programming tasks, improving your overall coding
        efficiency."
28    },
29    {
30        "id": "PA001",
31        "code": "PA001",
32        "description": "A project to build a calculator using Python
        .",
33        "explanation": "This project allows you to apply what you
        learn in a practical way. Building a calculator will
        solidify your understanding of fundamental programming
        concepts like user input, arithmetic operations, and
        function use within the context of a complete application
        ."
34    }
35 ]
36 }
```

Đánh giá về lời giải thích các tài nguyên:

- GPT-4o-mini: Các lời giải thích cho mỗi tài nguyên đều rõ ràng, nhấn mạnh lợi ích mà người học sẽ nhận được, chẳng hạn như việc nâng cao kỹ năng giải quyết vấn đề hoặc làm quen với các công cụ phát triển backend. Tuy nhiên, các lời giải thích có phần chung chung và ít tập trung vào ứng dụng thực tế của các kỹ năng học được.
- GPT-4o: Cung cấp những lời giải thích chi tiết hơn về lý do tại sao mỗi tài nguyên lại hữu ích cho người học, ví dụ như việc hiểu cách hoạt động của web và tầm quan trọng của các kiến thức về máy chủ và triển khai. Những lời giải thích này giúp người học hiểu rõ hơn về mục đích và ứng dụng của các tài nguyên trong việc phát triển kỹ năng backend.
- Gemini Pro: Cũng cung cấp những lời giải thích chi tiết, nhưng có phần thiên về việc khuyến khích người học tham gia vào các bài học thực tế. Mô hình này đưa ra những lý



do cụ thể như việc học cách xử lý chuỗi (COD002) hay hiểu cách danh sách hoạt động trong Python (LES001), từ đó giúp người học nâng cao kỹ năng giải quyết vấn đề và lập trình thực tế.

Về chất lượng và chi tiết giải thích:

- GPT-4o-mini: Lời giải thích trong GPT-4o-mini thường ngắn gọn và tập trung vào lý do chung chung tại sao một tài nguyên lại hữu ích, nhưng thiếu sâu sắc về cách tài nguyên đó giúp người học phát triển các kỹ năng cụ thể.
- GPT-4o: Các lời giải thích chi tiết hơn và cung cấp cái nhìn rõ ràng về các ứng dụng thực tế của từng tài nguyên. Chúng giúp người học nhận thức được lý do tại sao mỗi tài nguyên lại quan trọng trong hành trình học tập của họ.
- Gemini Pro: Cung cấp các lời giải thích rõ ràng, tuy nhiên có xu hướng tập trung vào các kỹ năng cụ thể mà người học sẽ phát triển thông qua các bài học và dự án. Gemini Pro khuyến khích người học áp dụng ngay các kỹ năng vào thực tế thông qua các dự án và bài học.

Kết luận: Tất cả ba mô hình GPT-4o-mini, GPT-4o và Gemini Pro đều cung cấp các tài nguyên học tập có giá trị, nhưng mức độ chi tiết và sự rõ ràng trong lời giải thích có sự khác biệt. GPT-4o là mô hình có giải thích chi tiết và nhấn mạnh đến tầm quan trọng của việc hiểu sâu về lý thuyết, trong khi Gemini Pro có xu hướng tập trung vào thực hành nhiều hơn. GPT-4o-mini, mặc dù cung cấp các tài nguyên tương tự, nhưng các lời giải thích có phần đơn giản và ít sâu sắc hơn.



9.3 Đánh giá các hệ thống liên quan

Tiêu chí	Hệ thống Adaptive LearningPath [5]	Hệ thống FOKE [6]	CS50.ai[8]	APAS[9]	LeetCode
Cá nhân hóa	Đề xuất lộ trình dựa trên năng lực học sinh	Dựa trên hồ sơ người học và kỹ thuật prompt engineering	Gợi ý mã nguồn, trợ lý học tập AI	Phản hồi tức thời, gợi ý cải thiện mã	Tạo bài tập nhưng ít cá nhân hóa
Tính tương thích với người dùng	Thông qua đồ thị tri thức và tài nguyên phù hợp	Tùy chỉnh theo phong cách học tập và hành vi	Trợ lý học tập dựa trên GPT-4, trả lời câu hỏi học thuật	Phản hồi chi tiết về lỗi và cải tiến mã	Tạo bài tập dựa trên ngữ cảnh nhưng ít tùy chỉnh
Khả năng tương tác	Chỉ có gợi ý tài liệu học tập	Tương tác qua gợi ý cấu trúc và phản hồi liên tục	Chatbot tương tác trực tiếp với sinh viên	AI-Tutor cung cấp phản hồi và gợi ý cải thiện mã	Tương tác chủ yếu thông qua bài tập và câu hỏi
Độ chính xác của đề xuất	Tùy chỉnh theo dữ liệu thực tế	Sử dụng graph embedding và gợi ý cấu trúc	AI đưa ra phản hồi chính xác về mã nguồn	Phản hồi tức thời về mã nguồn và gợi ý cải thiện	Tạo bài tập nhưng không có giải thích chi tiết
Sự minh bạch và giải thích	Đưa ra lý do thay đổi lộ trình học	Cung cấp giải thích rõ ràng về lộ trình học	Giải thích chi tiết về mã nguồn và phong cách lập trình	Cung cấp lý do cài đặt mã, không tiết lộ đáp án	Ít giải thích về cách tạo bài tập hoặc câu hỏi
Tính mở rộng	Có thể mở rộng với nhiều tài nguyên học tập	Có thể mở rộng với nhiều đối tượng người học và lĩnh vực	Có thể áp dụng cho các khóa học online lớn	Có thể mở rộng cho nhiều khóa học và bài tập	Hệ thống có sẵn các bài tập cho nhiều cấp độ
Ứng dụng vào thực tế	Có thể áp dụng cho lớp học truyền thống và trực tuyến	Ứng dụng vào nhiều môi trường học tập khác nhau	Ứng dụng trong các khóa học lớn như CS50	Hệ thống đánh giá tự động có thể áp dụng cho lớp học và MOOCs	Ứng dụng cho các bài tập lập trình cá nhân và nhóm
Tiết kiệm thời gian cho giảng viên	Giảm tải công việc của giảng viên	Giảm tải cho giảng viên trong việc thiết kế lộ trình học	Giảm tải cho giảng viên trong việc giải thích mã	Giảm tải cho giảng viên trong việc kiểm tra và đánh giá bài tập	Giảm tải cho giảng viên trong việc tạo bài tập
Hiệu quả học tập	Cải thiện khả năng học của học sinh qua lộ trình cá nhân hóa	Cải thiện hiệu quả học tập nhờ cá nhân hóa sâu	Tăng cường khả năng tự học của sinh viên	Giúp sinh viên phát triển tư duy thuật toán	Hiệu quả học tập phụ thuộc vào việc tự học của người dùng

Table 9.1: Đánh giá các hệ thống liên quan

- Các tiêu chí đánh giá hệ thống

1. Cá nhân hóa: Đánh giá mức độ cá nhân hóa của lộ trình học tập, phản hồi và tài nguyên học tập.
2. Tính tương thích với người dùng: Đánh giá khả năng tương thích của hệ thống với các đặc điểm người dùng như phong cách học tập, mục tiêu học tập, và năng lực hiện tại.
3. Khả năng tương tác: Đánh giá mức độ và tính hiệu quả của sự tương tác giữa người học và hệ thống (ví dụ: chatbot, phản hồi tức thời).
4. Độ chính xác của đề xuất: Đánh giá khả năng của hệ thống trong việc tạo ra các đề xuất chính xác, phù hợp với người học.
5. Sự minh bạch và giải thích: Đánh giá mức độ giải thích rõ ràng của hệ thống về các đề xuất và lý do chọn các tài liệu học tập.
6. Tính mở rộng: Đánh giá khả năng mở rộng của hệ thống khi có nhiều người học và tài nguyên giáo dục hơn.



7. Ứng dụng vào thực tế: Đánh giá khả năng áp dụng hệ thống vào môi trường học tập thực tế, như lớp học truyền thống, lớp học trực tuyến, hoặc hệ thống MOOCs.
8. Tiết kiệm thời gian cho giảng viên: Đánh giá mức độ mà hệ thống giúp giảng viên giảm tải công việc như đánh giá bài tập, phản hồi học sinh, v.v.
9. Hiệu quả học tập: Đánh giá mức độ cải thiện kết quả học tập của sinh viên khi sử dụng hệ thống.

9.4 Các vấn đề và khó khăn trong quá trình phát triển

9.5 Kết chương

Chương 10

Tổng kết

10.1 Kết quả đạt được

10.2 Hướng phát triển

Bibliography

- [1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. <https://arxiv.org/abs/2005.14165>
- [2] IBM. (n.d.). What is a knowledge graph? [Accessed: 2024-09-22]. <https://www.ibm.com/topics/knowledge-graph>
- [3] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 505–513.
- [4] Abdelrahman, G., Wang, Q., & Nunes, B. (2023). Knowledge tracing: A survey. *ACM Comput. Surv.*, 55(11). <https://doi.org/10.1145/3569576>
- [5] Raj, N. S., & Renumol, V. G. (2022). An improved adaptive learning path recommendation model driven by real-time learning analytics. *Journal of Computers in Education*, 11, 121–148. <https://doi.org/10.1007/s40692-022-00250-y>
- [6] Hu, S., & Wang, X. (2024). Foke: A personalized and explainable education framework integrating foundation models, knowledge graphs, and prompt engineering. <https://arxiv.org/abs/2405.03734>
- [7] Ng, C., & Fung, Y. (2024). Educational personalized learning path planning with large language models. <https://arxiv.org/abs/2407.11773>
- [8] Liu, R., Zenke, C., Liu, C., Holmes, A., Thornton, P., & Malan, D. J. (2024). Teaching cs50 with ai: Leveraging generative artificial intelligence in computer science education. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V*, 1, 750–756. <https://doi.org/10.1145/3626252.3630938>
- [9] Frankford, E., Sauerwein, C., Bassner, P., Krusche, S., & Breu, R. (2024). Ai-tutoring in software engineering education. *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*, 309–319. <https://doi.org/10.1145/3639474.3640061>
- [10] Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications.



- [11] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Le, Q., & Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *CoRR*, *abs/2201.11903*. <https://arxiv.org/abs/2201.11903>
- [12] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. <https://arxiv.org/abs/2312.10997>
- [13] Lin, D. (2024). Revolutionizing retrieval-augmented generation with enhanced pdf structure recognition. <https://arxiv.org/abs/2401.12599>
- [14] Bulathwela, S., Perez-Ortiz, M., Ranawaka, E. S. V., Siriwardana, R. I. P. B. B., Ganepola, G. A. K. Y., Ravikumar, T., Pussegoda, S., Novak, E., Yilmaz, E., & Shawe-Taylor, J. (2021). Cross modal, cross cultural, cross lingual, cross domain, and cross site global oer network.
- [15] Li, H., Yu, J., Ouyang, Y., Liu, Z., Rong, W., Li, J., & Xiong, Z. (2024). Explainable few-shot knowledge tracing. *arXiv preprint arXiv:2405.14391*.
- [16] Alkhatlan, A., & Kalita, J. (2018). Intelligent tutoring systems: A comprehensive historical survey with recent developments. *CoRR*, *abs/1812.09628*. <http://arxiv.org/abs/1812.09628>
- [17] GeeksforGeeks. (2024). React vs angular vs vue: Which framework is the best? [Accessed: 2024-12-12]. <https://www.geeksforgeeks.org/react-vs-angular-vs-vue-which-framework-is-the-best/>
- [18] HackerNoon. (2024). Python web frameworks compared: Django vs fastapi [Accessed: 2024-12-12]. <https://hackernoon.com/python-web-frameworks-compared-django-vs-fastapi#:~:text=Python%20Web%20Frameworks%20Compared%3A%20Django%20vs%20FastAPI%201,...%208%20Community%20and%20Support%20...%20More%20items>
- [19] DB-Engines. (2024). Arangodb vs neo4j vs orientdb [Accessed: 2024-12-12]. <https://db-engines.com/en/system/ArangoDB%3bNeo4j%3bOrientDB>
- [20] GeeksforGeeks. (n.d.). *React vs angular vs vue: Which framework is the best?* Retrieved December 12, 2024, from <https://www.geeksforgeeks.org/react-vs-angular-vs-vue-which-framework-is-the-best/>
- [21] HackerNoon. (n.d.). *Python web frameworks compared: Django vs fastapi*. Retrieved December 12, 2024, from <https://hackernoon.com/python-web-frameworks-compared-django-vs-fastapi#:~:text=Python%20Web%20Frameworks%20Compared%3A%20Django%20vs%20FastAPI%201,...%208%20Community%20and%20Support%20...%20More%20items>
- [22] DB-Engines. (n.d.). *Arangodb vs neo4j vs orientdb*. Retrieved December 12, 2024, from <https://db-engines.com/en/system/ArangoDB%3bNeo4j%3bOrientDB>