

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 THÔNG GIÁO DỤC THÔNG
MINH Dựa TRÊN MÔ HÌNH NGÔN NGỮ LỚN**

NGÀNH: KHOA HỌC MÁY TÍNH

HỘI ĐỒNG 6 ĐỒ ÁN TỐT NGHIỆP

GVHD 1: Võ Thị Ngọc Châu

GVHD 2: Nguyễn Hứa Phùng

Thư ký: Nguyễn Phương Duy

Phản biện: Bùi Tiến Đức

—o0o—

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 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 tương lai. 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	15
1.1	Động lực	15
1.2	Mục tiêu	16
1.3	Phạm vi	16
1.4	Ý nghĩa của đề tài	17
1.4.1	Ý nghĩa của dự án dưới góc nhìn thực tiễn	17
1.4.2	Ý nghĩa của dự án theo góc nhìn khoa học	17
1.5	Cấu trúc báo cáo	17
2	Kiến thức nền tảng	19
2.1	Kiến thức nền tảng về giáo dục và giáo dục thông minh	19
2.1.1	Khái quát về giáo dục	19
2.1.2	Khái quát về giáo dục thông minh	20
2.2	Mô hình ngôn ngữ	20
2.3	Các thuật ngữ kỹ thuật liên quan đến giáo dục thông minh	21
2.3.1	Đồ thị tri thức (Knowledge graph)	21
2.3.2	Knowledge tracing	22
2.4	Các nguyên tắc cơ bản trong việc thiết lập mục tiêu học tập và phương pháp đánh giá	22
2.4.1	Nguyên tắc SMART trong việc thiết lập mục tiêu	22
2.4.2	Phương pháp Rubric-Based Assessment	23
2.4.3	Phương pháp đánh giá STAR	23
2.5	Kiến trúc công nghệ	25
2.5.1	Công cụ xử lý ngôn ngữ tự nhiên - Framework tích hợp LLMs	25



2.5.1.1	LangChain	25
2.5.1.2	LlamaIndex	26
2.5.1.3	Tổng quan	27
2.5.2	Framework xây dựng giao diện người dùng	28
2.5.2.1	React	28
2.5.2.2	Vue	28
2.5.2.3	Angular	29
2.5.2.4	Tổng quan	30
2.5.3	Framework phát triển API	30
2.5.3.1	FastAPI	31
2.5.3.2	Django	31
2.5.3.3	Tổng quan	31
2.6	Tổng kết	32
3	Công trình liên quan	33
3.1	Hệ thống đề xuất lộ trình học	33
3.2	Ứng dụng LLM vào việc giảng dạy khoa học máy tính, lập trình,...	34
3.3	Prompt engineering	36
3.4	Retrieval-Augmentation Generation	36
3.5	Knowledge tracing/Student Profiling	37
4	Hệ thống đề xuất	39
4.1	Mô tả hệ thống	39
4.2	Khảo sát yêu cầu	40
4.2.1	Người dùng	40
4.2.2	Yêu cầu chức năng	40
4.2.2.1	User Management	40
4.2.2.2	Learning Resource Recommendation System	41
4.2.2.3	Exercise Generator	41
4.2.2.4	AI Tutor	41
4.2.2.5	Progress Tracking	42
4.2.2.6	Course Management	42



4.2.2.7	Admin Management	42
4.2.3	Yêu cầu phi chức năng	43
4.2.4	Yêu cầu dữ liệu	44
4.2.4.1	Các thực thể và thuộc tính dữ liệu	44
4.2.4.2	Mối quan hệ dữ liệu	45
4.3	Kết luận	46
5	Phân tích và thiết kế	47
5.1	Phân tích	47
5.1.1	Business Process Modeling	47
5.1.2	Use case diagrams	48
5.1.2.1	Course Management	49
5.1.2.2	User Management	50
5.1.2.3	Learning Resource Recommendation System	51
5.1.2.4	Progress Tracking	52
5.1.2.5	Exercise Generator	54
5.1.2.6	AI Tutor	55
5.2	Thiết kế	57
5.2.1	Kiến trúc hệ thống	57
5.2.1.1	Phân tích kiến trúc hệ thống phổ biến hiện nay	57
5.2.1.2	Lựa chọn kiến trúc hệ thống	58
5.3	Database	62
5.3.1	EER diagram	62
5.3.2	Lược đồ CSDL quan hệ ánh xạ	65
5.3.3	Class diagram	66
5.3.4	Sitemap	67
5.3.4.1	Sinh viên	67
5.3.4.2	Giảng viên	68
5.3.4.3	Quản trị viên	69
5.4	Quy trình Đề xuất và Theo dõi Lộ trình Học tập Cá nhân hóa	70
5.4.1	Tổng quan	70



5.4.2	Giai đoạn 1: Phân tích Mục tiêu và Tạo Lộ trình Học tập	71
5.4.3	Giai đoạn 2: Tạo Bài Kiểm tra Tự động	72
5.4.4	Giai đoạn 3: Theo dõi và Đánh giá Tiến độ Học tập	73
5.4.5	Giai đoạn 4: Điều chỉnh Nội dung Học tập	74
5.4.6	Kết luận	75
6	Hiện thực	76
6.1	Kiến trúc Frontend	76
6.2	Kiến trúc Backend	79
6.3	Quá trình triển khai hệ thống	82
6.4	Sinh ra lộ trình học tập	83
6.5	Đề xuất mục tiêu học tập	85
6.6	Đánh giá kết quả học tập	86
6.6.1	Đánh giá quá trình học tập (Progress Tracking Assessment)	86
6.6.2	Theo dõi và phân tích tiến độ học tập	88
6.6.3	Đánh giá kết quả học tập (Monitor Study Progress)	90
6.7	Tạo bài kiểm tra (Generate Quiz)	91
6.8	Tạo bổ sung nội dung bài học (Regenerate Lesson Content)	94
6.9	Tính năng liên quan tới bài tập code	95
6.9.1	Tính năng tạo sinh bài tập code (Bao gồm nội dung bài tập và testcases)	95
6.9.2	Tính năng AI assistant	98
6.9.3	Tính năng cho phép học sinh kiểm thử và gửi lời giải của bài tập code	100
6.10	Kiến trúc tổng thể hệ thống	101
6.10.1	Frontend (Client-side):	101
6.10.2	Backend (Server-side):	101
6.10.3	Cơ sở dữ liệu (Database):	101
6.10.4	AI Module (LLM – Large Language Model):	101
6.10.5	Hệ thống lưu trữ tài liệu:	101
7	Giao diện người dùng	102
7.1	Landing Page	102
7.2	Đăng nhập	103



7.3	Profile	103
7.4	Sinh viên	104
7.4.1	Giao diện chính	104
7.4.2	Quản lý khóa học	104
7.4.3	Chi tiết khóa học	105
7.4.4	Chi tiết bài học	105
7.4.5	Chi tiết module	106
7.4.6	Chi tiết bài quiz	107
7.5	Giảng viên	107
7.5.1	Giao diện chính	107
7.5.2	Quản lý khóa học	108
7.5.3	Quản lý phản hồi của sinh viên	109
7.5.4	Quản lý tiến độ sinh viên	109
7.6	Admin	110
7.6.1	Giao diện chính	110
7.6.2	Quản lý khóa học	111
7.6.3	Quản lý người dùng	111
7.6.4	Quản lý phản hồi	112
7.6.5	Thêm người dùng	112
7.6.6	Thêm khóa học	113
8	Kiểm thử hệ thống	114
8.0.1	Tổng quan về kiểm thử	114
8.0.2	Mô tả chung về các kịch bản kiểm thử	114
8.0.2.1	Cấu trúc của một kịch bản kiểm thử	115
8.0.2.2	Ví dụ về một kịch bản kiểm thử trong bảng Excel	115
8.0.2.3	Mục tiêu của các kịch bản kiểm thử	115
8.1	Các API cần kiểm thử	116
8.1.1	Phân tích chi tiết các kịch bản kiểm thử	116
8.1.1.1	Kiểm thử API Tạo Lộ Trình Học Tập	116
8.1.1.2	Kiểm thử API Theo Dõi Tiến Độ Học Tập	117



8.1.1.3	Kiểm thử API Tạo Bài Quiz	118
8.1.1.4	Kiểm thử API Tạo Bài Tập Lập Trình	118
8.1.1.5	Kiểm thử API Đánh Giá Tiền Độ Học Tập	119
8.2	Kết quả kiểm thử API	120
8.2.1	Kết quả tổng hợp của các testcase	120
9	Đánh giá hệ thống	121
9.1	Đánh giá tính năng và hiệu quả của hệ thống	121
9.1.1	Tính năng chính dựa trên LLM	121
9.1.2	Các yêu cầu chức năng khác	123
9.1.3	Các yêu cầu phi chức năng	125
9.2	Đánh giá chất lượng đầu ra của bộ trình học đề xuất bởi LLM	126
9.2.1	Phương án đánh giá	126
9.2.2	Xây dựng các metrics	127
9.2.3	Thiết kế quy trình đánh giá	128
9.2.4	Kết quả đánh giá và nhận xét	128
9.3	Đánh giá testcases được sinh ra bởi tính năng tạo sinh bài tập code	130
9.3.1	Phương pháp đánh giá	130
9.3.2	Cài đặt	130
9.3.3	Kết quả đánh giá và nhận xét	131
9.3.4	Hướng phát triển và cải thiện cho phần tạo sinh testcases	132
9.4	Đánh giá độ trễ trung bình (Latency) của các tính năng sử dụng LLM	132
9.4.1	Phạm vi đánh giá	132
9.4.2	Phương pháp đánh giá	132
9.4.3	Kết quả đo lường	133
9.4.4	Đánh giá kết quả	133
9.4.5	Phương án cải thiện	134
9.5	Đánh giá latency của một số API không liên quan đến LLM	134
9.5.1	Phạm vi và mục tiêu	134
9.5.2	Thiết lập thử nghiệm	135
9.5.3	Kết quả đo lường	135



9.5.4 Đánh giá kết quả	136
9.6 Đánh giá giao diện người dùng (UI) bằng Lighthouse	136
10 Tổng kết	139
10.1 Kết quả đạt được	139
10.2 Hướng phát triển	140

Danh mục hình ảnh

5.1	Activity diagram	47
5.2	Use case diagram	48
5.3	Detailed use case with id 1: Course Management	49
5.4	Detailed use case with id 2: User Management	50
5.5	Detailed use case with id 3: Adaptive Learning Path Management of Learner Actor	51
5.6	Detailed use case with id 3: Learning Resource Recommendation System of System Actor	51
5.7	Detailed use case with id 4: Progress Tracking of Learner Actor	52
5.8	Detailed use case with id 4: Progress Tracking of System Actor	52
5.9	Detailed use case with id 5: Exercise Generator	54
5.10	Detailed use case with id 6: AI Tutor of Learner Actor	55
5.11	Detailed use case with id 6: AI Tutor of System Actor	55
5.12	Component-based Diagram cho toàn bộ hệ thống	61
5.13	Component-based Diagram cho những công nghệ được chọn	61
5.14	Sơ đồ EER của hệ thống	62
5.15	Lược đồ CSDL quan hệ ánh xạ của hệ thống	65
5.16	Class diagram hiện thực của hệ thống	66
5.17	Sitemap cho đối tượng Sinh viên	67
5.18	Sitemap cho đối tượng Giảng viên	68
5.19	Sitemap cho đối tượng Quản trị viên	69
6.1	Hệ thống Typography sử dụng trong ứng dụng	78
6.2	Bảng màu được sử dụng trong ứng dụng	79
6.3	Lộ trình học tập cá nhân hóa dựa trên mục tiêu sinh viên	85



6.4	Chi tiết một bài học trong lộ trình	85
6.5	Báo cáo đánh giá học tập dựa trên phương pháp STAR	88
6.6	Báo cáo theo dõi những vấn đề trong quá trình học tập	89
6.7	Báo cáo theo dõi những thành tích trong quá trình học tập	90
6.8	Form tạo bài kiểm tra trắc nghiệm	93
6.9	Kết quả bài kiểm tra trắc nghiệm	93
6.10	Flow tạo sinh bài tập code	95
6.11	Tính năng coding assistant	99
6.12	Tính năng kiểm thử testcases	100
7.1	Giao diện trang chủ của hệ thống	102
7.2	Giao diện đăng nhập của hệ thống	103
7.3	Giao diện thông tin cá nhân của sinh viên	103
7.4	Giao diện chính của sinh viên	104
7.5	Giao diện quản lý khóa học của sinh viên	104
7.6	Giao diện chi tiết khóa học của sinh viên	105
7.7	Giao diện chi tiết bài học của sinh viên	105
7.8	Danh sách các modules học tập của bài học	106
7.9	Giao diện chi tiết module của sinh viên	106
7.10	Giao diện chi tiết bài quiz của sinh viên	107
7.11	Danh sách các bài quiz của sinh viên	107
7.12	Dashboard Professor	108
7.13	Modal Feedback	108
7.14	Professor: My Course	109
7.15	Professor: Feedback Management	109
7.16	Professor: Progress Student	110
7.17	Giao diện chính của quản trị viên	110
7.18	Giao diện quản lý khóa học của quản trị viên	111
7.19	Giao diện quản lý người dùng của quản trị viên	111
7.20	Giao diện quản lý phản hồi của quản trị viên	112
7.21	Giao diện thêm người dùng của quản trị viên	112
7.22	Giao diện thêm khóa học của quản trị viên	113



8.1	Ví dụ về một kịch bản kiểm thử	115
8.2	Kiểm thử API tạo lộ trình học tập	116
8.3	Kiểm thử API theo dõi tiến độ học tập	117
8.4	Kiểm thử API tạo bài tập quiz	118
8.5	Kiểm thử API tạo bài tập code	118
8.6	Kiểm thử API đánh giá tiến độ học tập	119
8.7	Kết quả tổng hợp của các testcase	120
9.1	Radar chart tổng hợp kết quả đánh giá lộ trình học để xuất khi chạy đánh giá LLM-as-a-judge ứng với từng model	129
9.2	Kết quả đánh giá một số trang bằng Lighthouse	138

Danh mục bảng biểu

2.1	So sánh giữa LangChain và LlamaIndex	27
2.2	So sánh giữa React, Vue và Angular	30
2.3	So sánh giữa FastAPI và Django	31
6.1	Các công nghệ sử dụng trong quá trình triển khai	83
9.1	Danh sách các tính năng đã hoàn thành trong hệ thống	122
9.2	Danh sách các tính năng chức năng đã hoàn thành (Phần 1)	123
9.3	Danh sách các tính năng chức năng đã hoàn thành (Phần 2)	124
9.4	Danh sách các yêu cầu phi chức năng đã đáp ứng (Phần 1)	125
9.5	Danh sách các yêu cầu phi chức năng đã đáp ứng (Phần 2)	126
9.6	Điểm trung bình theo từng tiêu chí đánh giá: Goal, Explanation Quality, Module Appropriateness, và Ordering Logic	128
9.7	Tổng quan kết quả thực thi test case sinh bởi LLM	131
9.8	Thời gian phản hồi trung bình của một số tính năng sử dụng LLM	133
9.9	Thông kê độ trễ các API ở mức tải 3 VU (min / med / p95 / max)	135
9.10	Thông kê độ trễ các API ở mức tải 10 VU (min / med / p95 / max)	135
9.11	Thông kê độ trễ các API ở mức tải 50 VU (min / med / p95 / max)	136

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 khác.



- **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 Các nguyên tắc cơ bản trong việc thiết lập mục tiêu học tập và phương pháp đánh giá

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:

2.4.1 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”.

2.4.2 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.

2.4.3 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.



2.5 Kiến trúc công nghệ

2.5.1 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.

2.5.1.1 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.

2.5.1.2 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.

2.5.1.3 Tổng quan

Table 2.1: So sánh giữa LangChain và LlamaIndex

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

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.



2.5.2 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.

2.5.2.1 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.

2.5.2.2 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.



2.5.2.3 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.

2.5.2.4 Tổng quan

Table 2.2: So sánh giữa React, Vue và Angular

Tiêu chí	React	Vue	Angular
Dễ học	DỄ HỌC VỚI JSX VÀ CÁC CONCEPT NHƯ STATE VÀ LIFE-CYCLE 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 CÁC 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 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

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.

2.5.3 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.



2.5.3.1 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

2.5.3.2 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

2.5.3.3 Tổng quan

Table 2.3: So sánh giữa FastAPI và Django

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

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.



2.6 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:** Hệ thống hỗ trợ ba loại vai trò chính: sinh viên, giảng viên và quản trị viên. Quản trị viên có quyền tạo tài khoản cho sinh viên, giảng viên hoặc quản



trị viên khác, cập nhật theo từng khóa học với các thông tin cơ bản như email, họ tên, mã số sinh viên/mã số cán bộ, v.v. Sinh viên và giảng viên sử dụng email do trường đại học cung cấp để đăng ký mật khẩu. Người dùng cũng có thể đăng nhập thông qua tài khoản Google được liên kết với email của trường đại học (nếu có).

2. **Cá nhân hóa**: Người dùng có thể cập nhật thông tin cá nhân như avatar, ngày tháng năm sinh, v.v.

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) cho sinh viên dựa trên mục tiêu học tập, đầu ra môn học và nội dung môn học.
2. **Lựa chọn tài nguyên học tập**: Hệ thống cho phép người dùng tự chọn cách rèn luyện kiến thức bằng cách học qua bài quiz hoặc code exercises hoặc chỉ đọc tài liệu 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 với số lượng câu hỏi không giới hạn.
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 hoặc yêu cầu người dùng học lại bài học hoặc xem lại bài học trước đó nếu vẫn chưa đủ chỉ tiêu vượt qua đầu ra bài học.
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 hỗ trợ cung cấp các gợi ý trong quá trình làm bài tập code để sinh viên đạt hiệu quả tốt nhất khi rèn luyện.

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.
- **Tối ưu hóa code:** Gia sư AI cung cấp gợi ý cách viết mã tối ưu hoặc hướng dẫn cách làm cho sinh viên để tối ưu thời gian giải quyết các bài tập.

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 tải lên các tài liệu cho các khóa học. 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 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 và trạng thái tài khoản.



- Quản trị viên có thể khóa hoặc mở khóa tài khoản người dùng.

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.
- Đánh dấu phản hồi đã giải quyết 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, Avatar, LearningPath, Email,..
 - Teacher: ID, Email, Name,..
 - Admin: ID, Email, Name,..

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

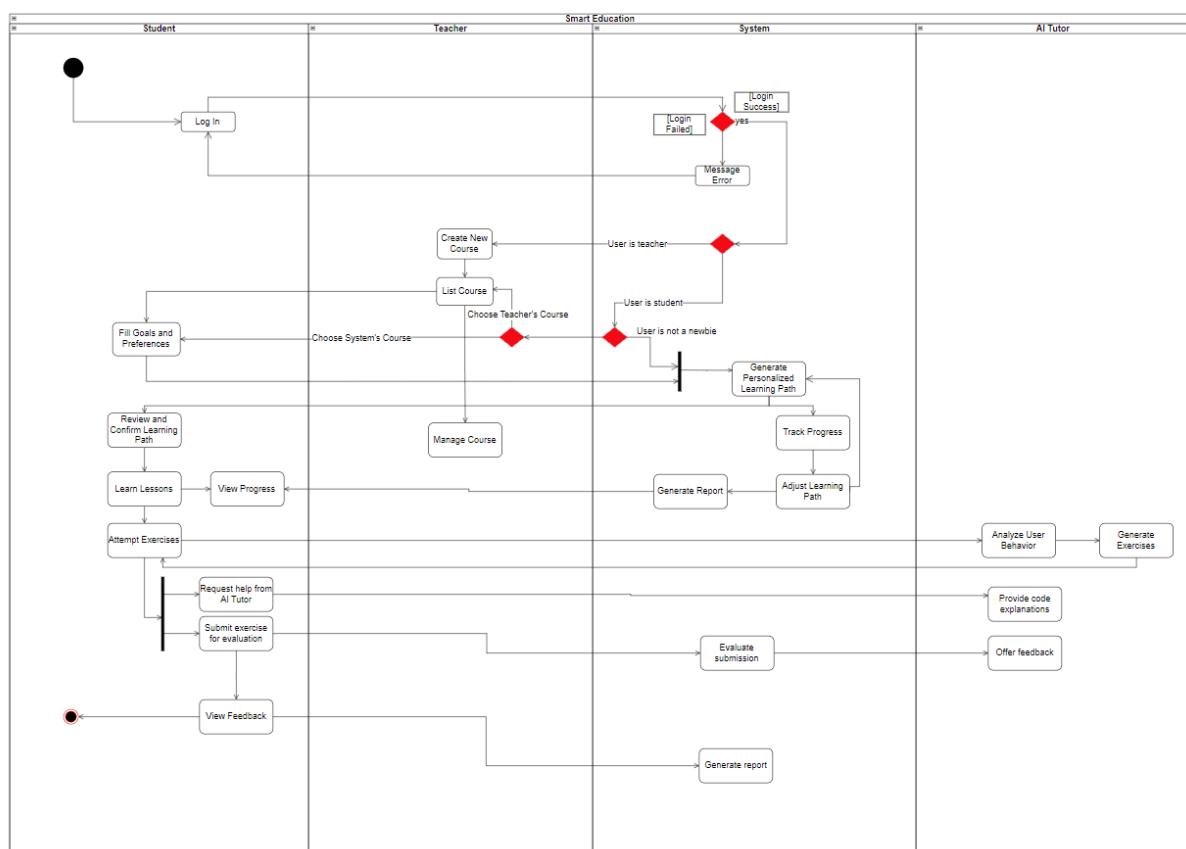


Figure 5.1: Activity diagram

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

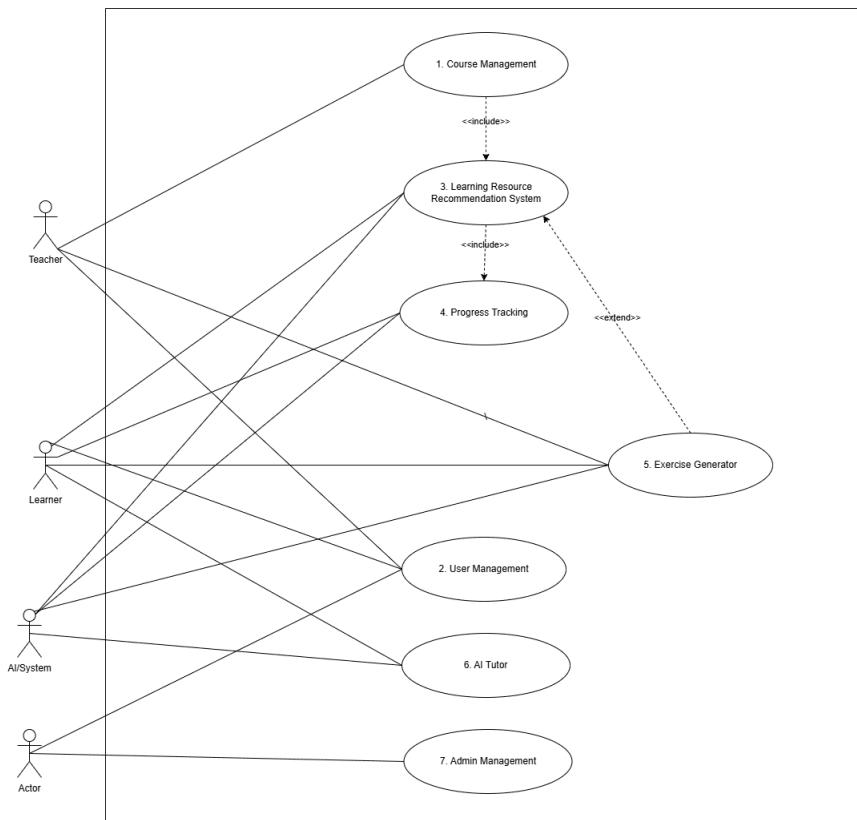


Figure 5.2: Use case diagram

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

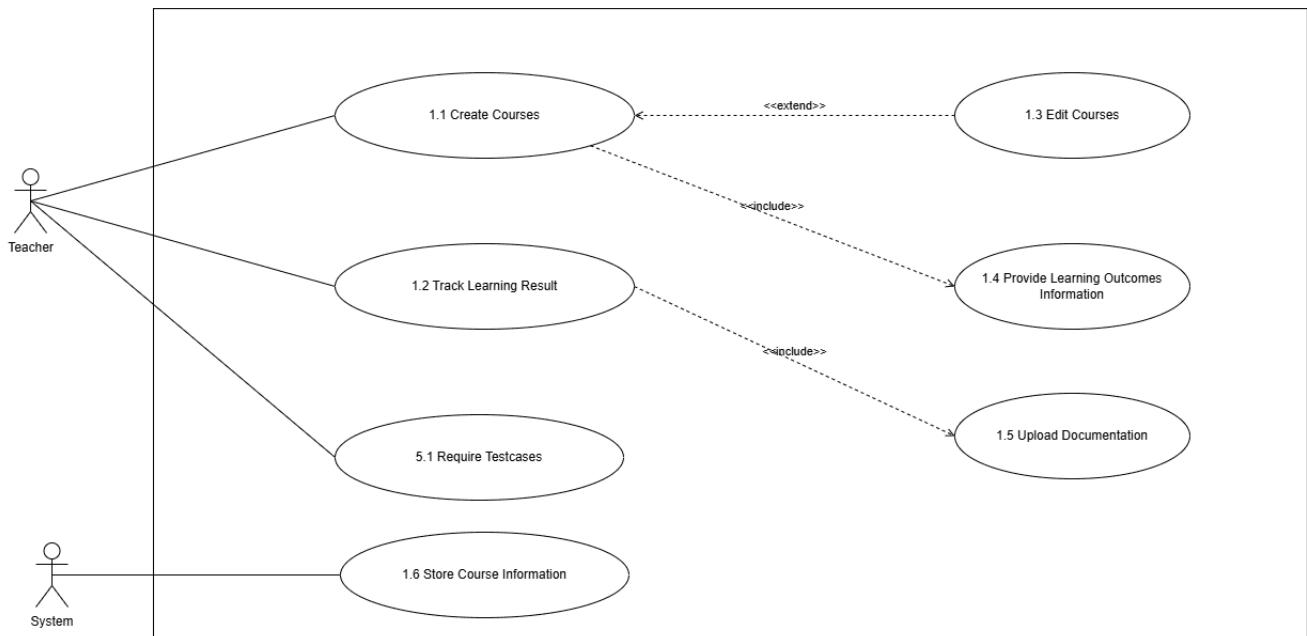


Figure 5.3: Detailed use case with id 1: Course Management

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

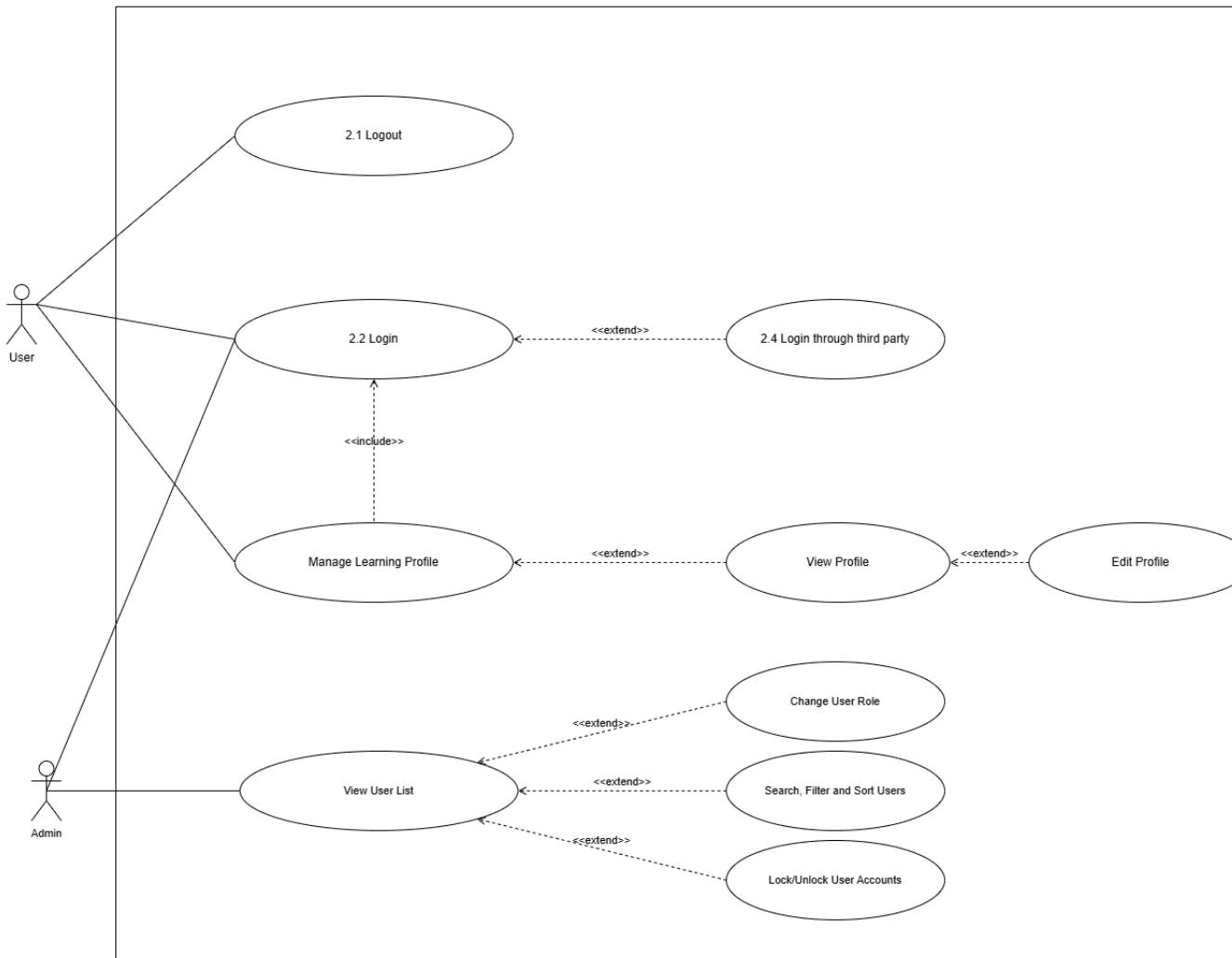


Figure 5.4: Detailed use case with id 2: User Management

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

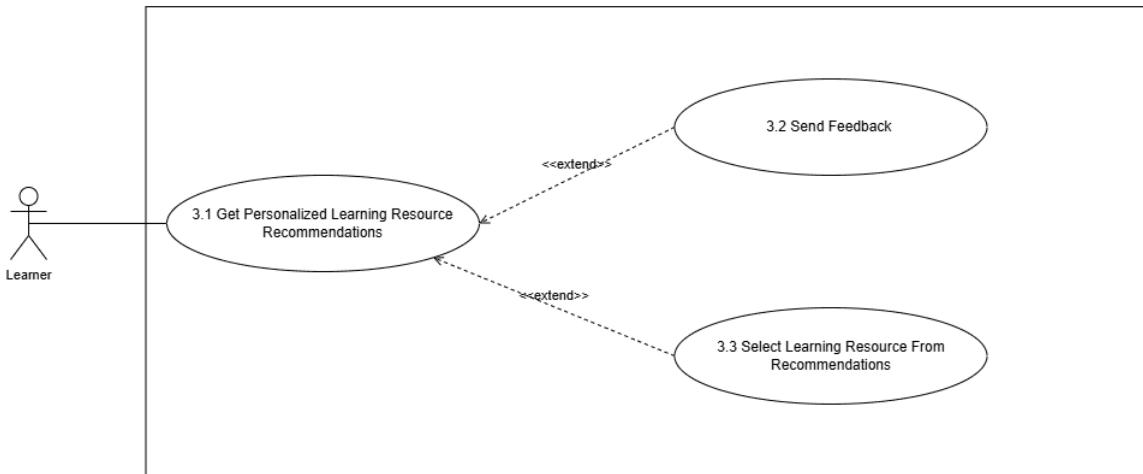


Figure 5.5: Detailed use case with id 3: Adaptive Learning Path Management of Learner Actor

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.

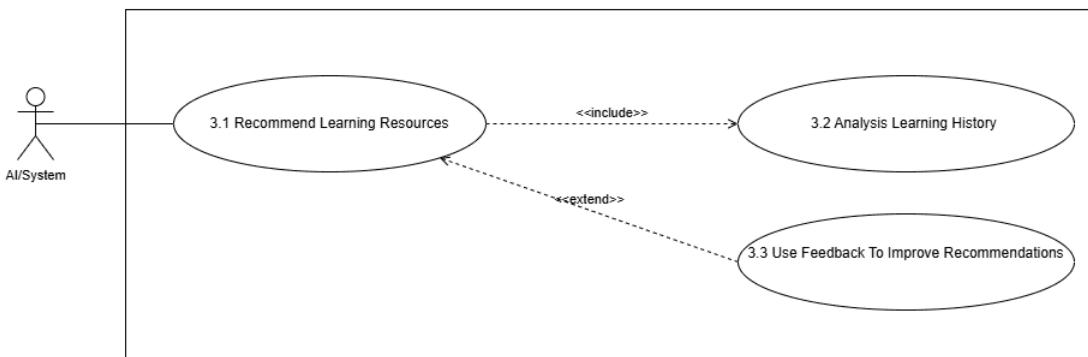


Figure 5.6: Detailed use case with id 3: Learning Resource Recommendation System of System Actor

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

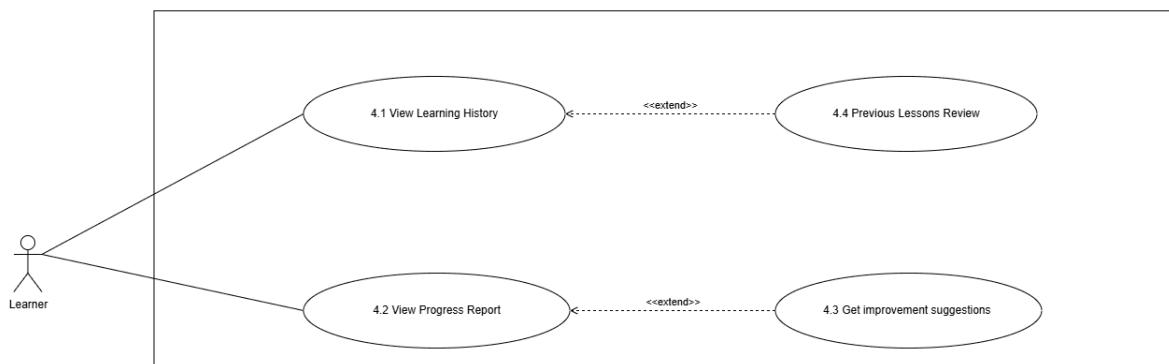


Figure 5.7: Detailed use case with id 4: Progress Tracking of Learner Actor

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.

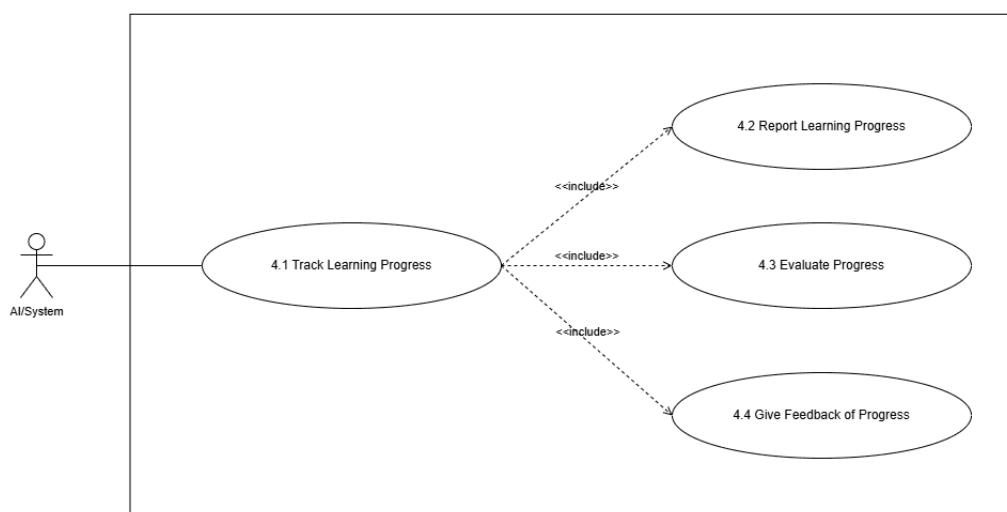


Figure 5.8: Detailed use case with id 4: Progress Tracking of System Actor

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

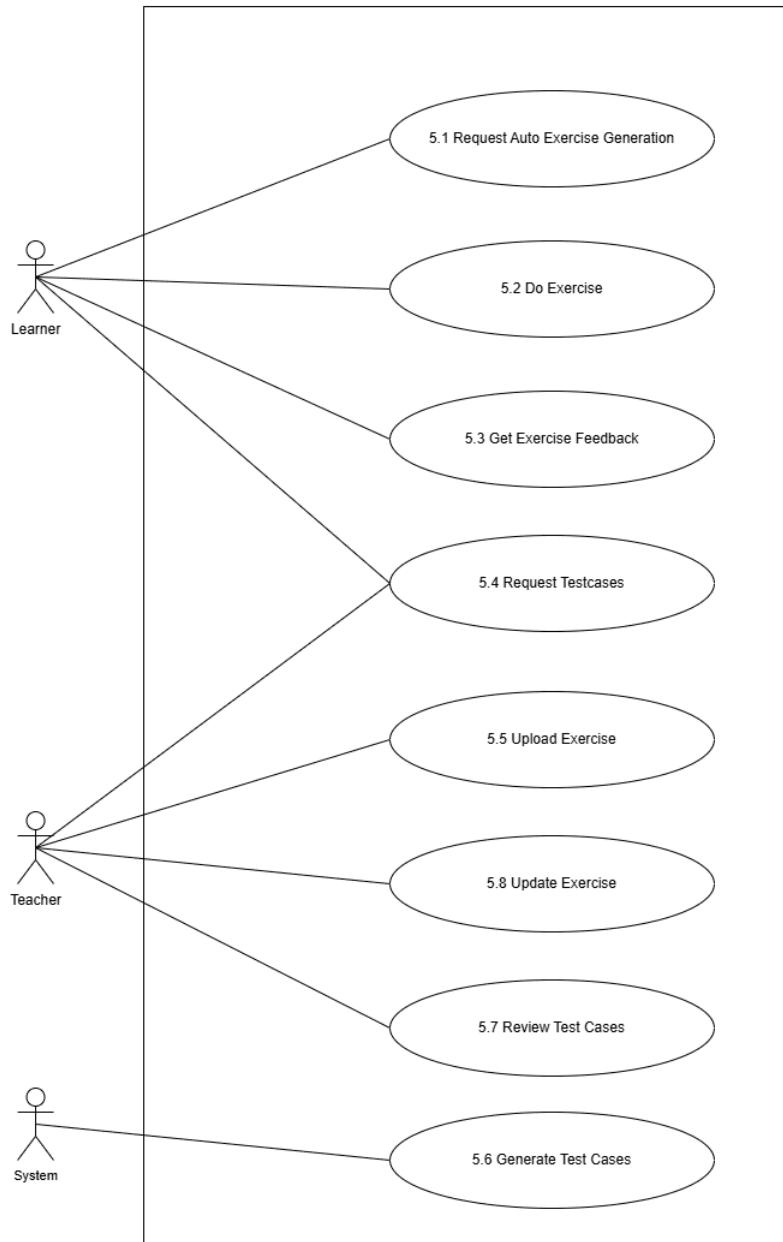


Figure 5.9: Detailed use case with id 5: Exercise Generator

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

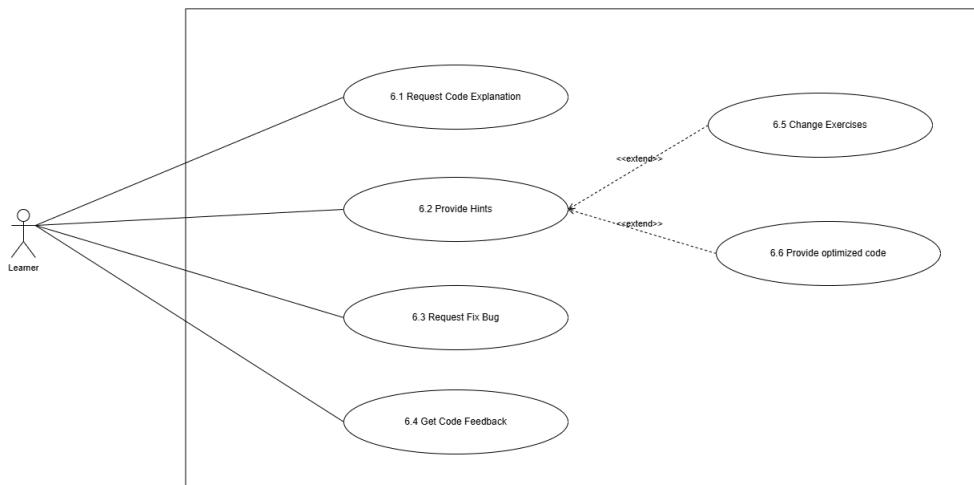


Figure 5.10: Detailed use case with id 6: AI Tutor of Learner Actor

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.

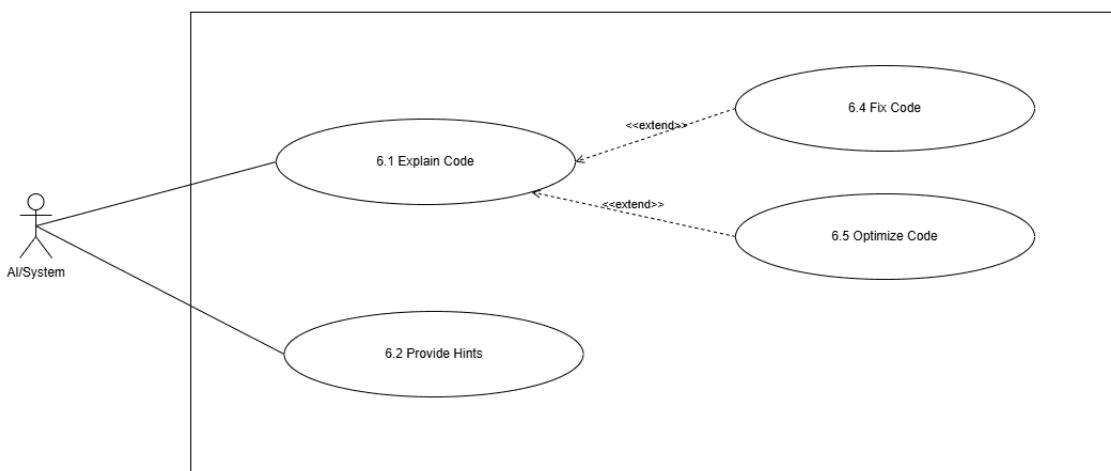


Figure 5.11: Detailed use case with id 6: AI Tutor of System Actor

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.

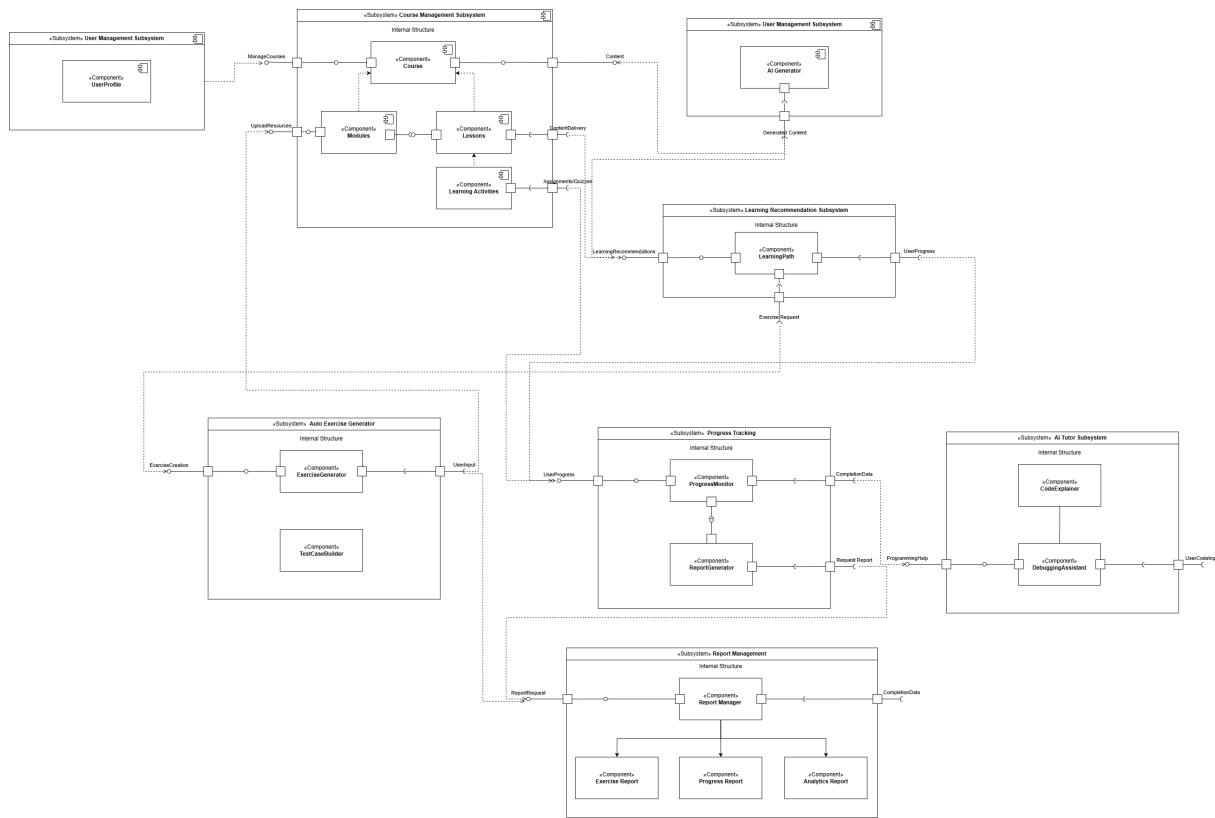


Figure 5.12: Component-based Diagram cho toàn bộ hệ thống

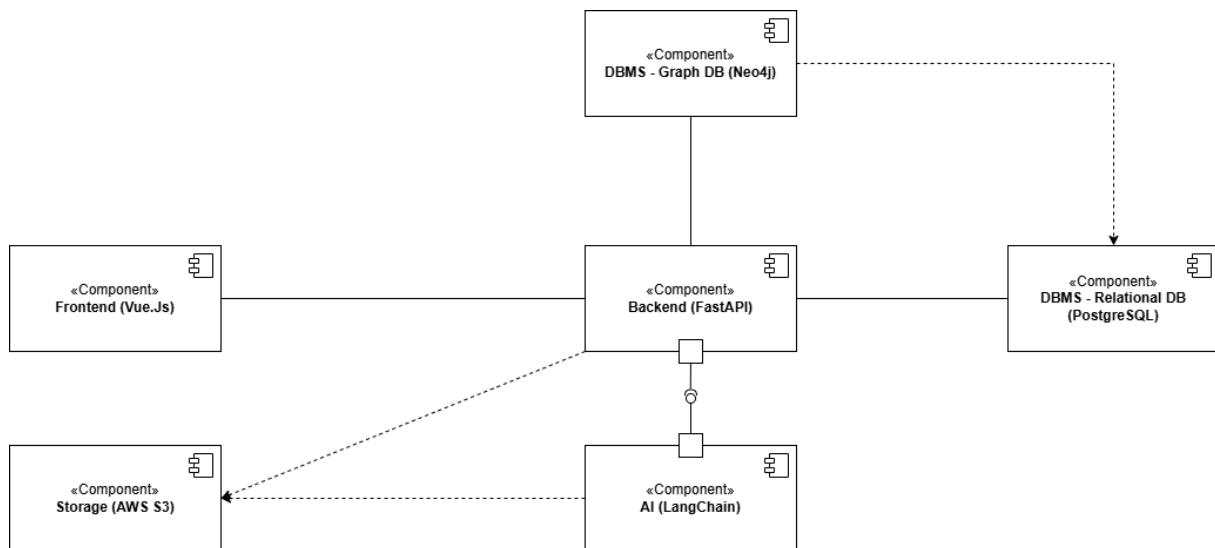


Figure 5.13: Component-based Diagram cho những công nghệ được chọn

5.3 Database

5.3.1 EER diagram

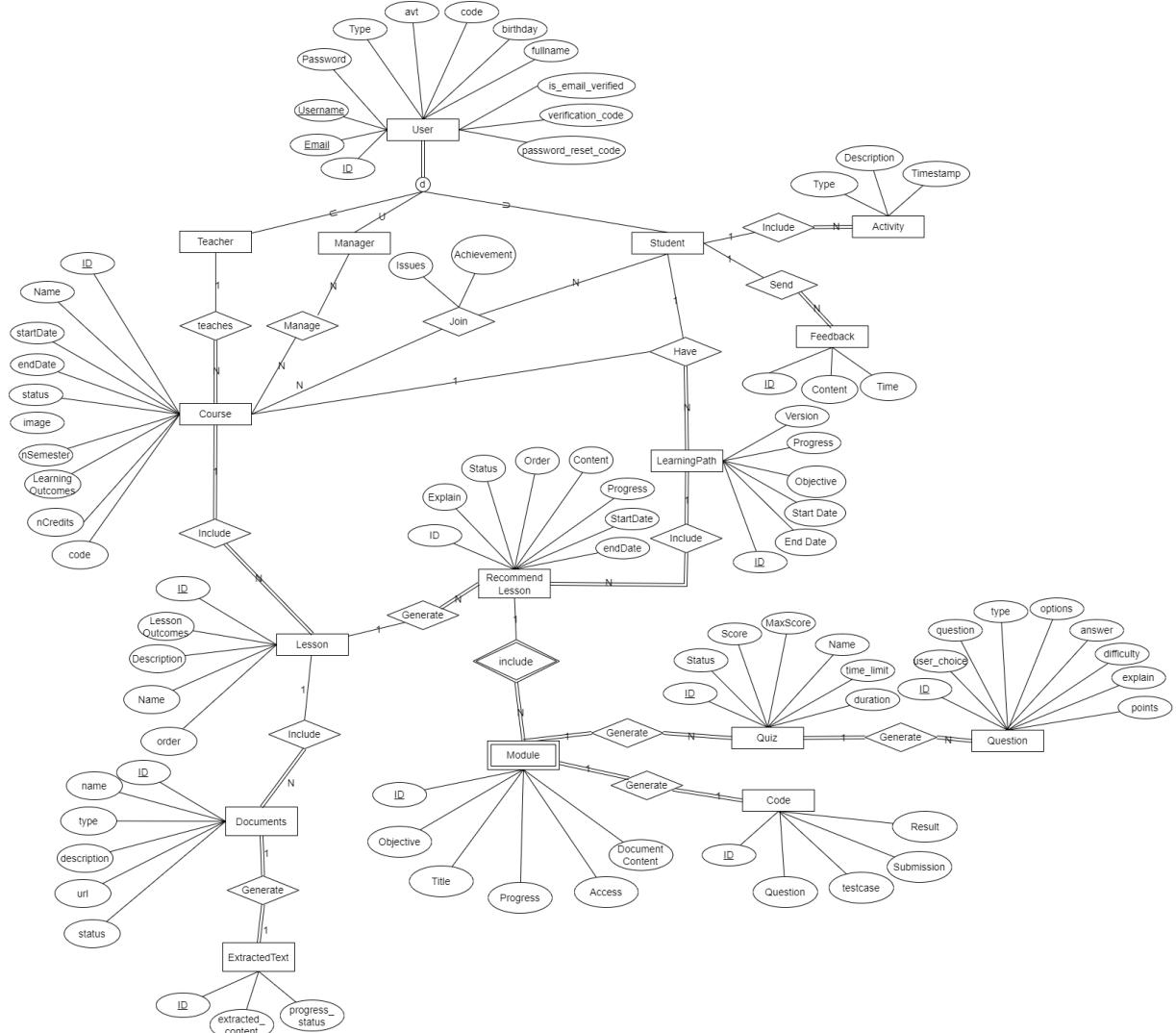


Figure 5.14: Sơ đồ EER của hệ thống

Sơ đồ EER (Enhanced Entity-Relationship) thể hiện cấu trúc dữ liệu của hệ thống học tập thông minh. Dưới đây là mô tả chi tiết các thực thể và mối quan hệ trong sơ đồ:

1. **User** là thực thể trung tâm của hệ thống, chứa thông tin người dùng như ID, Username, Email, Password, Type, và các thuộc tính liên quan đến xác thực (is_email_verified, verification_code, password_reset_code). User được chuyên biệt hóa (specialization) thành ba loại: Teacher, Manager, và Student.
2. **Student** kế thừa thuộc tính từ User và có các mối quan hệ quan trọng sau:
 - Include một hoặc nhiều Activity (hoạt động học tập).



- Send nhiều Feedback để đánh giá và góp ý về hệ thống.
- Join nhiều Course để tham gia các khóa học.
- Have một LearningPath cho mỗi khóa học mà Student tham gia.

3. **Teacher** kế thừa từ User và có mối quan hệ teaches với Course, thể hiện vai trò giảng dạy trong các khóa học.

4. **Manager** kế thừa từ User và có mối quan hệ Manage với Course, thể hiện vai trò quản lý các khóa học.

5. **Course** là thực thể chứa thông tin về khóa học như ID, Name, Description, nSemester, Learning Outcomes, nCredits, và status. Course có các mối quan hệ:

- Include nhiều Lesson (bài học).
- Include nhiều Exercise (bài tập).
- Được Join bởi nhiều Student.
- Được teaches bởi Teacher.
- Được Manage bởi Manager.

6. **Lesson** chứa thông tin về các bài học như ID, Name, Description, Lesson Outcomes. Lesson có thể được chuyên biệt hóa thành Recommend Lesson và có mối quan hệ:

- Include nhiều Document (tài liệu).
- Generate ExtractedText từ các tài liệu để phục vụ tìm kiếm và phân tích.

7. **Recommend Lesson** là dạng chuyên biệt của Lesson, chứa các thuộc tính bổ sung như Explain, Status, Order, Content, Progress, StartDate, endDate. Recommend Lesson có mối quan hệ:

- Include nhiều Module (đơn vị học tập nhỏ).
- Được Include trong LearningPath.

8. **LearningPath** đại diện cho lộ trình học tập cá nhân hóa của sinh viên, với các thuộc tính như ID, Objective, Progress, Version, Start Date và End Date. LearningPath có mối quan hệ:

- Include nhiều Recommend Lesson.
- Được Have bởi Student (mỗi Student có một LearningPath riêng cho mỗi Course).

9. **Module** là đơn vị học tập nhỏ nhất, chứa thông tin như ID, Title, Objective, Progress, Access, Document Content. Module có các mối quan hệ:



- Generate Quiz để đánh giá kiến thức.
 - Generate Code (bài tập lập trình).
10. **Quiz** là bài kiểm tra kiến thức với các thuộc tính ID, Status, Score, MaxScore, Name, time_limit, duration. Quiz có mối quan hệ:
- Generate nhiều Question (câu hỏi).
11. **Question** chứa thông tin về câu hỏi trong bài kiểm tra như ID, type, question, options, answer, difficulty, explain, points, user_choice.
12. **Code** đại diện cho bài tập lập trình với các thuộc tính ID, Question, testcase, Submission, Result.
13. **Document** chứa thông tin về tài liệu học tập như ID, name, type, description, url, status.
14. **ExtractedText** chứa nội dung trích xuất từ Document với các thuộc tính ID, extracted_content, progress_status.
15. **Activity** lưu trữ thông tin về hoạt động của Student với các thuộc tính Type, Description, Timestamp.
16. **Feedback** chứa phản hồi của Student với các thuộc tính ID, Content, Time.

Mỗi quan hệ trong sơ đồ được thể hiện rõ ràng với các kí hiệu biểu thị quan hệ một-nhiều (1-N) hoặc nhiều-nhiều (N-N). Ví dụ:

- Student — N — Join — N — Course: Mỗi quan hệ nhiều-nhiều, một Student có thể tham gia nhiều Course và một Course có thể có nhiều Student.
- Student — 1 — Have — N — LearningPath: Mỗi quan hệ một-nhiều, một Student có thể có nhiều LearningPath (mỗi khóa học một lộ trình).
- LearningPath — 1 — Include — N — Recommend Lesson: Mỗi quan hệ một-nhiều, một LearningPath bao gồm nhiều Recommend Lesson.
- Module — 1 — Generate — N — Quiz: Mỗi quan hệ một-nhiều, một Module có thể tạo ra nhiều Quiz.

Sơ đồ EER này cung cấp một cái nhìn tổng quan về cấu trúc dữ liệu của hệ thống, thể hiện rõ các thực thể chính, thuộc tính và mối quan hệ giữa chúng, tạo cơ sở vững chắc cho việc triển khai cơ sở dữ liệu hỗ trợ các chức năng của hệ thống học tập thông minh.

5.3.2 Lược đồ CSDL quan hệ ánh xạ

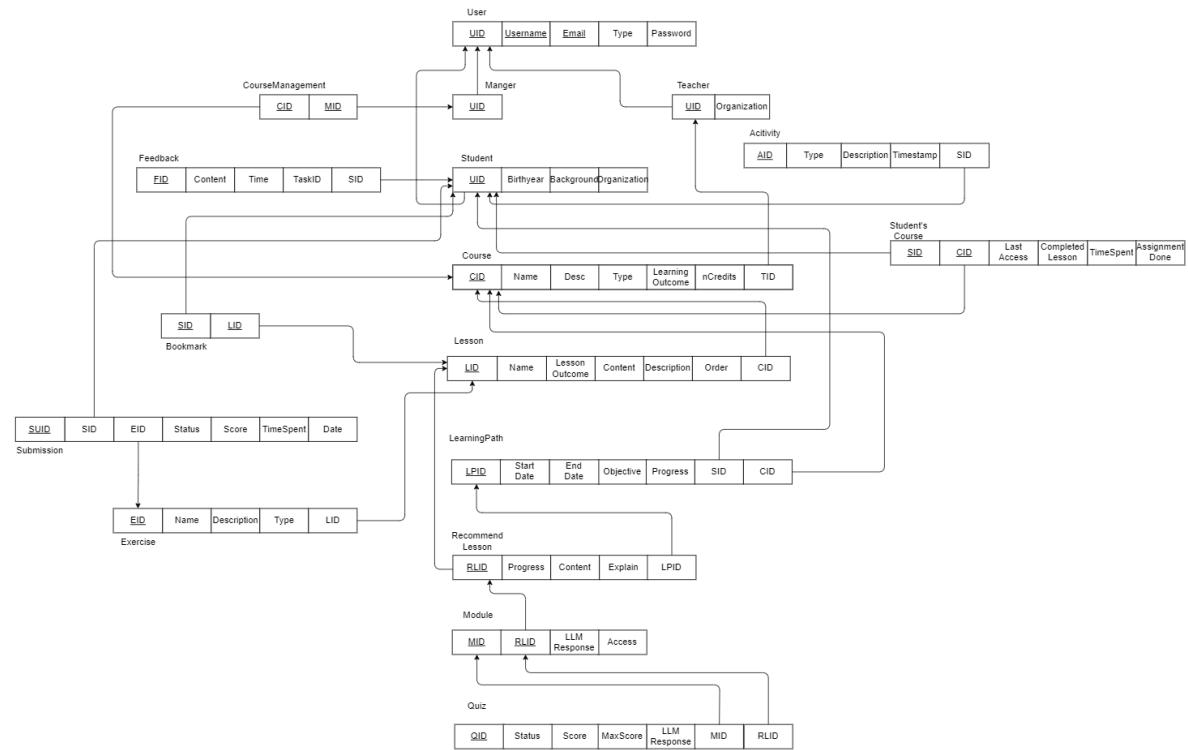


Figure 5.15: Lược đồ CSDL quan hệ ánh xạ của hệ thống

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

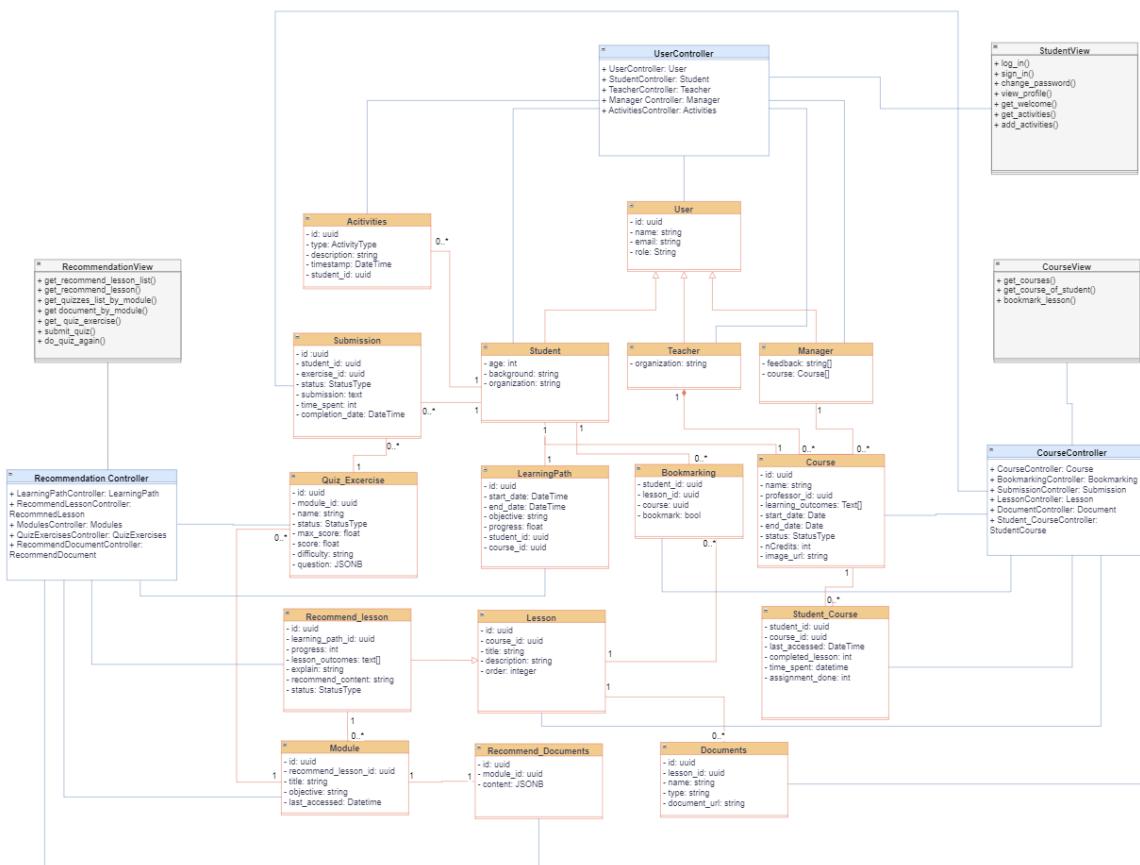


Figure 5.16: Class diagram hiện thực của hệ thống

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

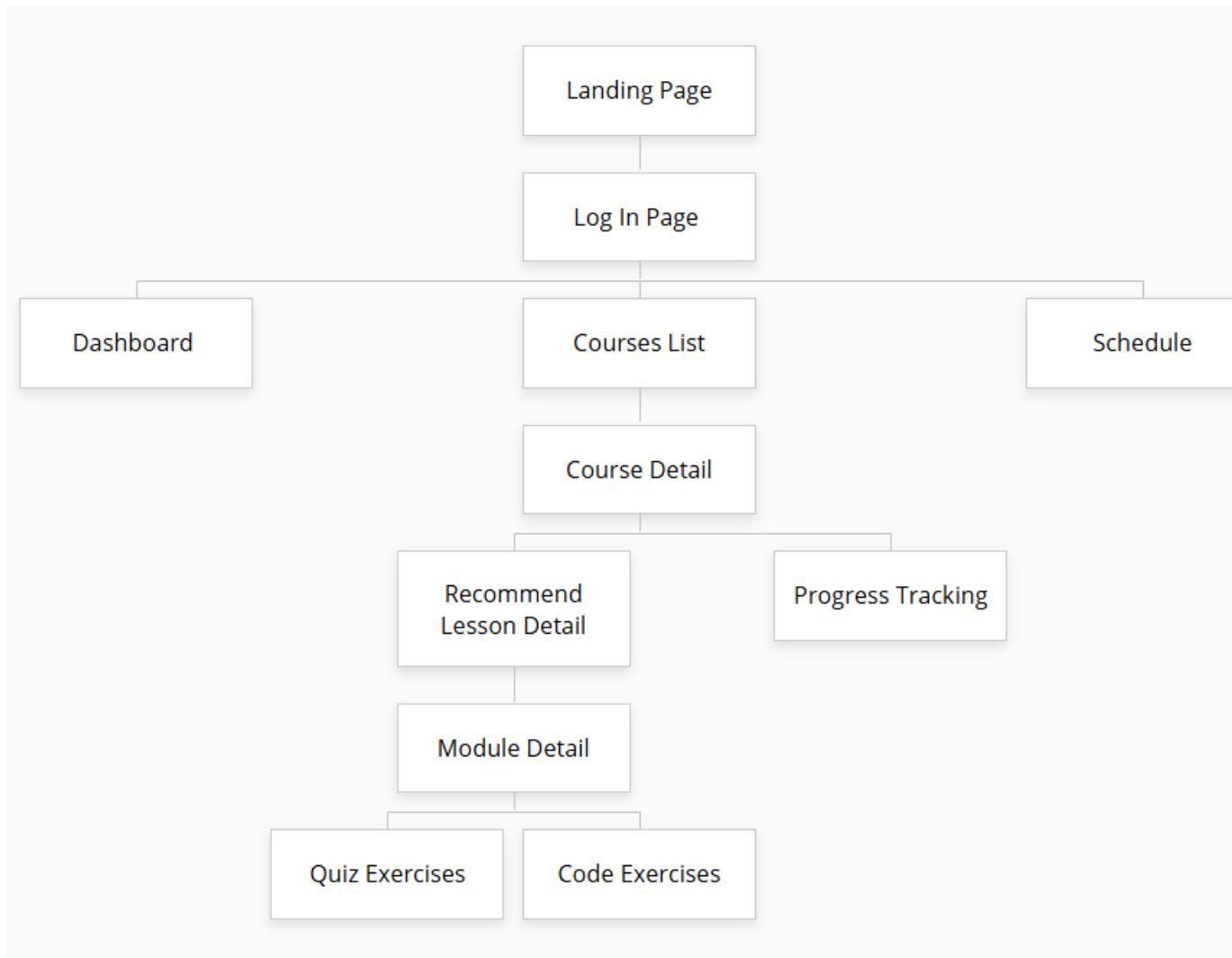


Figure 5.17: Sitemap cho đối tượng Sinh viên

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 trang điều hướng chính cho người dùng.

Bên dưới, có một số phần chính:

- **Dashboard:** Trang tổng quan cung cấp thông tin tóm tắt về các hoạt động và tiến độ học tập của sinh viên.
- **Courses List:** Trang này hiển thị danh sách các khóa học mà sinh viên có thể truy cập.
- **Schedule:** Trang lịch trình học tập và các sự kiện liên quan đến các khóa học.

- **Course Detail:** Trang này cung cấp thông tin chi tiết về một khóa học cụ thể, như nội dung, mục tiêu và các tài nguyên khác.
- **Recommend Lesson Detail:** Trang này cung cấp các khóa học đề xuất học tập dựa trên tiến độ và mong muốn của người dùng.
- **Progress Tracking:** Trang này cho phép sinh viên theo dõi tiến độ và hiệu suất của họ trong các khóa học.
- **Module Detail:** Trang hiển thị thông tin chi tiết về các module học tập trong khóa học.
- **Quiz Exercises and Code Exercises:** Những phần này đại diện cho nội dung bài tập của các module, bao gồm bài kiểm tra dạng quiz (multiple choices) và bài tập lập trình thực hành.

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

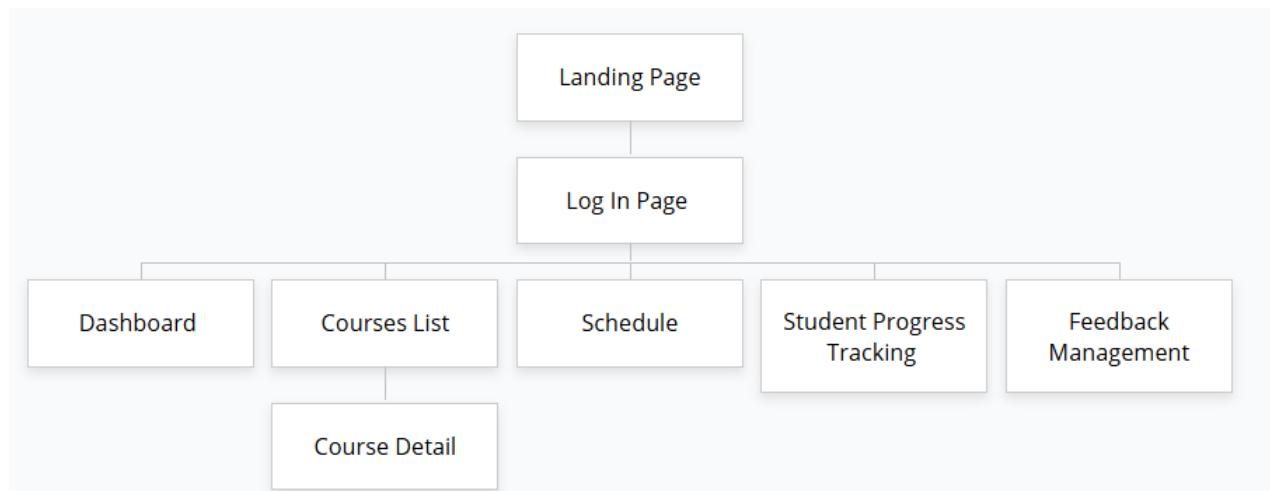


Figure 5.18: Sitemap cho đối tượng Giảng viên

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" và "Log In Page", giúp giảng viên dễ dàng truy cập vào các phần quan trọng của hệ thống.

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

- **Dashboard:** Trang tổng quan cung cấp thông tin tóm tắt về các hoạt động giảng dạy và quản lý khóa học.
- **Courses List:** Trang này hiển thị danh sách các khóa học mà giảng viên đang giảng dạy hoặc quản lý.
- **Schedule:** Trang lịch trình giảng dạy và các sự kiện liên quan đến khóa học.
- **Student Progress Tracking:** 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.
- **Feedback Management:** 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.
- **Course Detail:** Trang cung cấp thông tin chi tiết về một khóa học cụ thể, bao gồm nội dung, tài liệu và các hoạt động học tập.

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 Quản trị viên

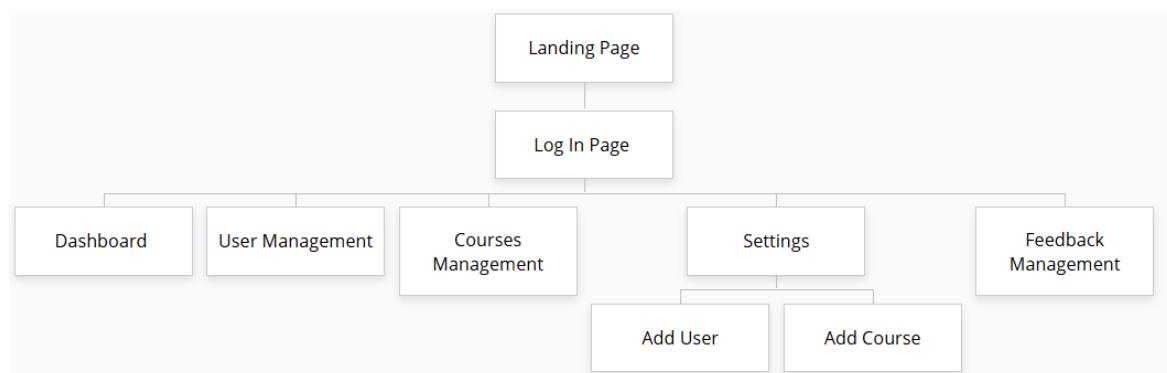


Figure 5.19: Sitemap cho đối tượng Quản trị viên

Sơ đồ trang web này cung cấp cái nhìn tổng quát về các chức năng chính mà Quản trị viên cần để quản lý toàn bộ hệ thống.

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

Các phần chính trong sơ đồ trang web này là:

- **Dashboard:** Trang tổng quan cung cấp thông tin tóm tắt về hoạt động của hệ thống và các số liệu thống kê quan trọng.



- **User Management:** Phần này bao gồm các trang để quản lý người dùng, bao gồm sinh viên và giảng viên.
- **Courses Management:** Phần này tập trung vào việc quản lý các khóa học trong hệ thống, bao gồm tạo mới và cập nhật khóa học.
- **Settings:** Trang cài đặt hệ thống cho phép quản trị viên tùy chỉnh các thông số và cấu hình của ứng dụng.
- **Feedback Management:** Phần này cho phép quản trị viên xem và quản lý phản hồi từ người dùng.
- **Add User:** Trang để thêm người dùng mới vào hệ thống.
- **Add Course:** Trang để thêm khóa học mới vào hệ thống.

Sơ đồ trang web này thể hiện cấu trúc quản lý toàn diện của hệ thống, cho phép quản trị viên kiểm soát tất cả các khía cạnh của ứng dụng, từ quản lý người dùng, khóa học đến cài đặt hệ thống và phản hồi.

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 tạo bổ sung 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 Kiến trúc 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.

```
CODEMATE-FE [WSL: UBUNTU]
|-- .vite
|-- dist
|-- node_modules
|-- public
|-- src
|   |-- assets
|   |-- common
|   |   |-- api.service.ts
|   |   |-- config.ts
|   |-- components
|   |-- composable
|   |-- constants
|   |-- layouts
|   |-- modals
|   |-- pages
|   |-- plugins
|   |-- router
|   |-- services
|   |-- stores
|   |-- types
```



```
|   |-- utils
|   |-- App.vue
|   |-- auto-imports.d.ts
|   |-- components.d.ts
|   |-- global.css
|   |-- main.ts
|   |-- typed-router.d.ts
|   |-- vite-env.d.ts
|-- .browserslistrc
|-- .editorconfig
|-- .env
|-- .eslintrc-auto-import.json
```

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 6.1: 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.

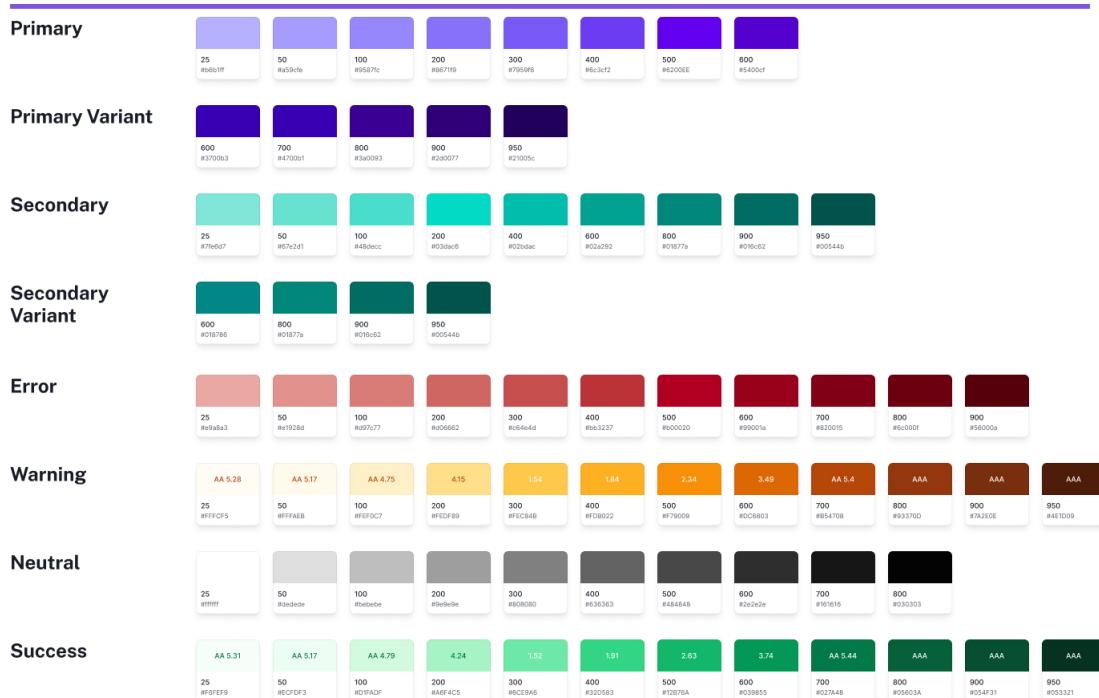


Figure 6.2: Bảng màu được sử dụng trong ứng dụng

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.

6.2 Kiến trúc 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
- Schemas
- Models
- API



```
EDUMIND [WSL: UBUNTU]
|-- alembic
|   |-- script.py.mako
|-- core
|-- data
|-- docker
|-- docs
|-- dumps
|-- machine
|   |-- __pycache__
|   |-- api
|   |-- controllers
|   |-- models
|   |-- providers
|   |-- repositories
|   |-- schemas
|   |-- services
|   |-- server.py
|-- migrations
|-- static
|-- tasks
|-- templates
|-- utils
|-- .env
|-- .gitignore
|-- .python-version
|-- alembic.ini
|-- main.py
|-- Makefile
|-- poetry.lock
|-- pyproject.toml
|-- README.md
```

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

Triển khai hệ thống

Hệ thống được triển khai hoàn toàn trên nền tảng Render, áp dụng cho cả thành phần frontend và backend. Việc sử dụng một nền tảng duy nhất cho toàn bộ hệ thống giúp đơn giản hóa quy trình CI/CD, quản lý tài nguyên tập trung và tối ưu hóa hiệu suất giao tiếp giữa các thành phần.

Frontend: Ứng dụng Vue.js được triển khai dưới dạng dịch vụ tĩnh (Static Site) trên Render với cấu hình tự động xây dựng và triển khai khi mã nguồn được cập nhật.

Link deploy frontend: <https://codemate-fe-n5h9.onrender.com/>

Backend: API FastAPI được triển khai dưới dạng dịch vụ web (Web Service) trên Render, với Docker làm công nghệ đóng gói để đảm bảo tính nhất quán môi trường. Hệ thống sử dụng Poetry để quản lý các phụ thuộc Python, đảm bảo môi trường phát triển và triển khai đồng nhất.

Link deploy backend: <https://edumind-5qmn.onrender.com/>

Quy trình triển khai chi tiết:

1. Chuẩn bị môi trường phát triển:

- Thiết lập môi trường phát triển cục bộ với đầy đủ công cụ cần thiết như Node.js, npm, Python 3.10, và Poetry.
- Cài đặt các IDE và công cụ phát triển như Visual Studio Code, PyCharm, và Git.
- Thiết lập hệ thống kiểm soát phiên bản sử dụng GitHub để quản lý mã nguồn và hỗ trợ làm việc nhóm.

2. Phát triển frontend:

- Sử dụng Vue 3 và Vuetify để xây dựng giao diện người dùng.
- Triển khai các component UI theo thiết kế đã được phê duyệt.
- Tích hợp các thư viện hỗ trợ như Axios cho việc gọi API, Pinia cho quản lý trạng thái.
- Thực hiện tối ưu hóa hiệu suất và khả năng đáp ứng trên các thiết bị khác nhau.



3. Phát triển backend:

- Xây dựng API sử dụng FastAPI theo kiến trúc Repository-Provider-Controller.
- Thiết lập kết nối và tương tác với cơ sở dữ liệu PostgreSQL.
- Tích hợp các dịch vụ bên ngoài như Gemini API, Google Authentication.
- Phát triển các module xử lý nghiệp vụ chính như quản lý người dùng, quản lý khóa học, đề xuất học tập, và tạo bài tập.

4. Kiểm thử hệ thống:

- Thực hiện kiểm thử đơn vị cho các thành phần riêng lẻ.
- Kiểm thử tích hợp để đảm bảo các thành phần phối hợp tốt với nhau.
- Kiểm thử chức năng để xác nhận các tính năng hoạt động đúng theo yêu cầu.
- Kiểm thử hiệu suất để đánh giá khả năng đáp ứng và xử lý tải của hệ thống.

5. Triển khai trên môi trường cloud:

- Thiết lập môi trường cloud trên Render cho frontend và backend.
- Cấu hình các biến môi trường và thiết lập bảo mật.
- Thiết lập quy trình CI/CD để tự động hóa việc build và deploy khi có thay đổi trong mã nguồn.
- Cài đặt và cấu hình cơ sở dữ liệu PostgreSQL trên nền tảng đám mây.

Các công nghệ được sử dụng trong triển khai:

Table 6.1: Các công nghệ sử dụng trong quá trình triển khai

Lĩnh vực	Công nghệ
Frontend	Vue 3, Vuetify, Tailwind CSS, Pinia, Axios
Backend	FastAPI, SQLAlchemy, Pydantic, LangChain, Alembic
Cơ sở dữ liệu	PostgreSQL, Redis (cache)
AI	OpenAI API (GPT-4o), Google Gemini API
Triển khai	Render (Frontend, Backend), GitHub Actions (CI/CD)

6.4 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
13        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 }
```

Kết quả giao diện một lô trình học tập cơ bản, ví dụ:

Learning Path for Data Structures and Algorithms

Goal: Achieve 70% or higher on quizzes covering basic data structures (arrays, linked lists) and algorithm analysis (Big-O notation) within the next two weeks.

VIEW YOUR PROGRESS

Mar 26 Mar 30 Apr 03

Course introduction, Recursion and Complexity of Algorithms Lists Sorting

Figure 6.3: Lô trình học tập cá nhân hóa dựa trên mục tiêu sinh viên

Lists

3 modules to complete

Overall Progress: 39% Complete

Start Date: March 30, 2025
End Date: April 2, 2025
This lesson is allocated 4 days to cover the data structures and the time complexities of common operations and practice.

Recommend: Concentrate on the implementation and characteristics of arrays and linked lists. Understand the advantages and disadvantages of each data structure in terms of memory usage, insertion, deletion, and search operations. Study the time complexities of common operations (e.g., accessing an element, inserting an element, deleting an element) for both arrays and linked lists using Big-O notation. Work through examples and exercises to solidify your understanding of how these data structures are used and how they perform.

Explain: This lesson directly addresses the student's goal by covering the fundamental data structures of arrays and linked lists. A strong understanding of these structures is crucial for solving problems related to data storage and manipulation. The quizzes will likely assess the student's ability to implement and analyze these data structures, so this lesson provides the necessary knowledge to excel. Mastering the concepts

Learning Outcomes:

- Understand the concept and importance of lists as a fundamental data structure.
- Differentiate between static lists (arrays) and dynamic lists (linked lists).
- Implement basic operations on arrays and linked lists, including insertion, deletion, and traversal.
- Compare and contrast singly linked lists, doubly linked lists, and circular linked lists.
- Analyze the time and space complexity of list operations.

Figure 6.4: Chi tiết một bài học trong lô trình

6.5 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
6                 with 70% accuracy by midterm",
7             "explanation": "This goal focuses on building fundamental
8                 programming skills ...",
9             "key_lessons": ["Introduction to Python", "Basic Data Types"]
10            },
11            {
12                "proficiency_level": "Average",
13                "goal": "Implement and optimize sorting algorithms with 80%
14                    efficiency by midterm",
15                "explanation": "This goal challenges the student to apply
16                    theoretical concepts ...",
17                "key_lessons": ["Algorithm Complexity", "Sorting Algorithms"]
18            }
19        ]
20    }
```

6.6 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.

6.6.1 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         }
}
```

Kết quả giao diện một báo cáo đánh giá học tập, ví dụ:



AI Learning Assessment

Assessment Summary (STAR Method)

Situation

- The student is aiming to achieve 70% or higher on quizzes related to data structures and algorithm analysis.
- The assessment is based on the provided lesson data and the student's progress within the lessons.

Task

- Evaluate the student's current understanding of the concepts.
- Provide recommendations to improve the student's learning progress.

Action

Theoretical Knowledge	Analyze the student's understanding of Big-O notation and algorithm ...
Coding Skills	Assess the student's ability to implement array and linked list operations....
Effort	Review the time spent on each lesson....

Close Assessment

AI Learning Assessment

Strengths

- The student is actively engaging with the learning material.
- The student is covering the necessary topics for the quizzes.
- The student is spending time on the lessons, indicating effort.

Areas to Note

- The student's progress in some modules is low (e.g., Big-O Notation Fundamentals).
- The student's understanding of recursion needs further evaluation.
- The student needs to focus on the practical application of Big-O notation.

Advice for Improvement

Theoretical Knowledge

- Focus on understanding Big-O notation and its practical implications.
- Practice analyzing the time and space complexity of algorithms.
- Review array and linked list implementations and their performance characteristics.

Close Assessment

Figure 6.5: Báo cáo đánh giá học tập dựa trên phương pháp STAR

6.6.2 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 }
```

Kết quả giao diện một báo cáo theo dõi tiến độ học tập, ví dụ:

The screenshot displays a web-based reporting interface titled "Learning Recommendation Analysis". At the top, it shows a summary section with "39 total issues found". Below this, there's a list of "Areas Showing Increased Difficulty" including "fundamental_misunderstanding_linked_lists", "inability_to_analyze_time_complexity_linked_list_operations", and "fundamental_misunderstanding_linked_list_vs_array". The "Most Frequent Issue Type" is listed as "Fundamental Misunderstanding Linked Lists".

Below the summary, there are three tabs: "SIGNIFICANT ISSUES" (which is selected), "RECOMMENDATIONS", and "ACHIEVEMENTS".

The "SIGNIFICANT ISSUES" table has columns: TYPE / SEVERITY, DESCRIPTION, FREQUENCY, and IMPACT ON GOALS.

TYPE / SEVERITY	DESCRIPTION	FREQUENCY	IMPACT ON GOALS
Fundamental Misunderstanding Linked Lists Severity: high	<ul style="list-style-type: none">The user demonstrates a profound lack of understanding of the basic structure and properties of linked lists, consistently confusing them with arrays or other data structures.This includes	10	<ul style="list-style-type: none">This misunderstanding will severely impede the student's ability to implement and utilize linked lists effectively.It will prevent the student from making progress on the goal

Figure 6.6: Báo cáo theo dõi những vấn đề trong quá trình học tập

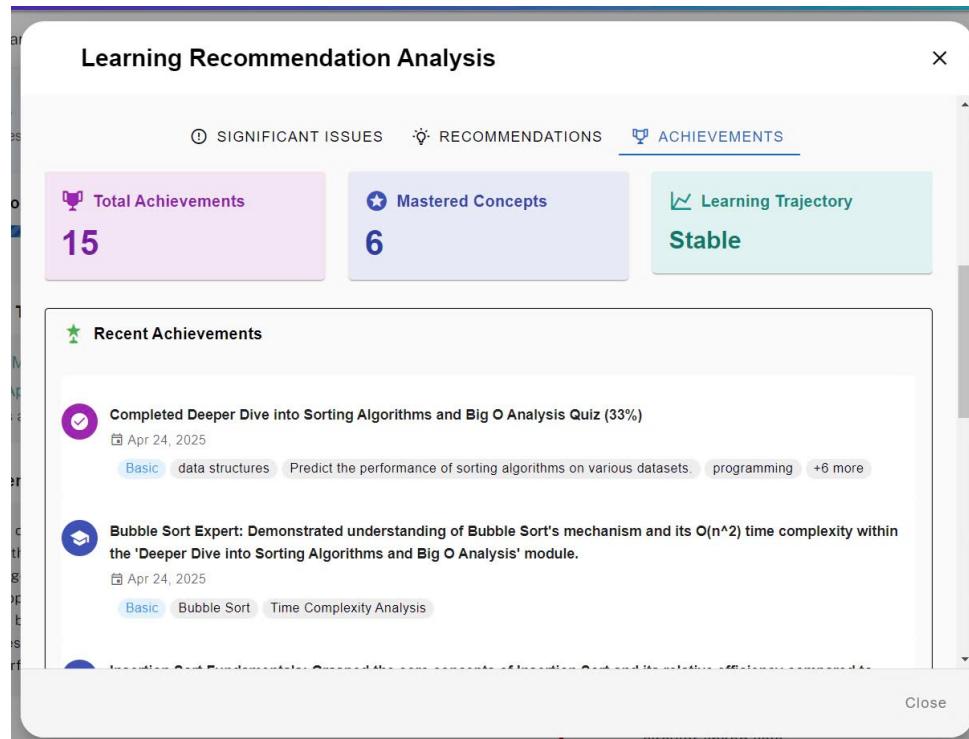


Figure 6.7: Báo cáo theo dõi những thành tích trong quá trình học tập

6.6.3 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     "total_issues_count": 13,
16     "increasing_issues": ["recursion_application"],
17     "most_frequent_type": "concept_misunderstanding"
18 },
19 "recommendations": [
20     {
21         "action": "repeat",
22         "reason": "Significant difficulties with recursion...",
23         "details": "Review 'Advanced Recursion'"
24     }
25 ]
26 }
```

6.7 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)
```



QUESTION BANK MANAGEMENT SYSTEM

- **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
5                     Python",
6     "time_limit": 30,
7     "max_score": 100,
8     "module_id": "def456",
9     "questions": [
10         {
11             "id": "q123",
12             "question_text": "What is a list in Python?",
13             "question_type": "single_choice",
14             "options": ["A mutable sequence type", "An immutable sequence
15                         type", "A mapping type", "A set type"],
16             "correct_answer": ["A mutable sequence type"],
17             "difficulty": "easy",
18             "explanation": "Lists in Python are mutable sequences...",
19             "points": 5
20         }
21     ]
22 }
```

Giao diện tạo một bài kiểm tra trắc nghiệm, ví dụ:

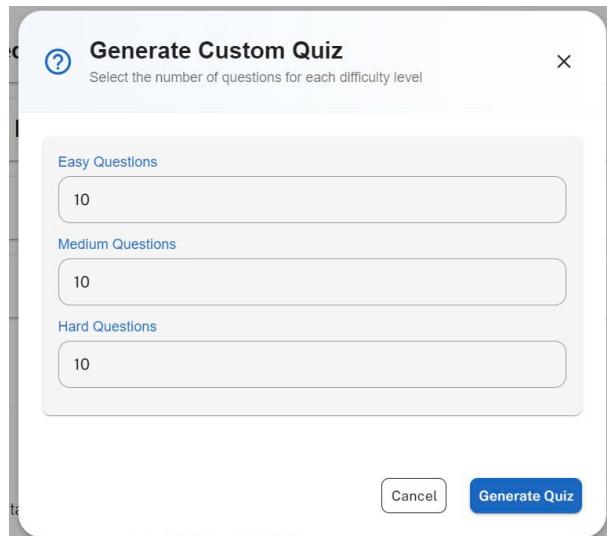


Figure 6.8: Form tạo bài kiểm tra trắc nghiệm

Sinh viên có thể chọn độ khó của bài kiểm tra và số lượng câu hỏi cho mỗi độ khó. Sau khi tạo, bài kiểm tra sẽ được lưu vào cơ sở dữ liệu và có thể được truy cập từ giao diện người dùng.

Quiz Results

22.22 / 100
You scored 22.22 points out of 100 (34%)

✓ 2 Correct ✗ 7 Incorrect

ⓘ AREAS TO IMPROVE ⚡ ACHIEVEMENTS

Concept Misunderstanding
Confusion between the characteristics of linked lists and arrays, particularly regarding memory allocation (contiguous vs. non-contiguous), element access (sequential vs. indexed), and the impact on insertion/deletion efficiency.

Knowledge Gap
Difficulty analyzing the time complexity of linked list operations, especially insertion, deletion, and search, and differentiating between $O(1)$ and $O(n)$ complexities based on the position of operation and linked list type.

👁 VIEW RESULTS CLOSE

Figure 6.9: Kết quả bài kiểm tra trắc nghiệm



Sau khi hoàn thành một bài kiểm tra, sinh viên có thể xem kết quả và nhận phản hồi từ hệ thống. Hệ thống sẽ phân tích những vấn đề sinh viên gặp phải, những phần sinh viên nắm chắc và cung cấp các khuyến nghị cho sinh viên dựa trên hiệu suất của họ trong bài kiểm tra.

6.8 Tạo bổ sung nội dung bài học (Regenerate Lesson Content)

Chức năng này tạo thêm nội dung bài học bổ sung dựa trên các vấn đề học sinh gặp phải, sử dụng endpoint /generate-supplemental-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 mức độ khó khăn sinh viên đang gặp phải và xác định liệu vấn đề có cần tham chiếu đến kiến thức từ các bài học trước đó không (needs_review_prior). Hệ thống thu thập thông tin liên quan từ các bài học trước nếu cần thiết để cung cấp bối cảnh đầy đủ.
- 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     updated_content = chunking_manager.call_llm_api(prompt, ...)
```

- Cập nhật dữ liệu:** Lưu các module bổ sung vào cơ sở dữ liệu và liên kết chúng với bài học hiện tại. Các module ban đầu vẫn được giữ nguyên, đảm bảo tính liên tục của lộ trình học tập.

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

```
1 {
2     "modules": [
```

```
3      {
4          "title": "Recursion Basics Revisited",
5          "objectives": ["Understand base cases", "Apply recursive
6                          patterns"],
7          "reading_material": { ... }
8      ]
9 }
```

Nội dung bổ sung được thiết kế để giải quyết các vấn đề cụ thể mà sinh viên đang gặp phải, đồng thời duy trì tính liên tục của lộ trình học tập. Các module bổ sung này tập trung vào việc làm rõ những khái niệm khó, cung cấp thêm ví dụ minh họa, và tạo cơ hội thực hành có hướng dẫn. Cách tiếp cận này giúp sinh viên vượt qua những khó khăn cụ thể mà không cần phải học lại toàn bộ nội dung.

6.9 Tính năng liên quan tới bài tập code

6.9.1 Tính năng tạo sinh bài tập code (Bao gồm nội dung bài tập và test-cases)

Sử dụng 2 agent với 2 prompt khác nhau cho 2 mục đích riêng biệt: tạo bài tập và tạo testcases, trong đó, ta sẽ sử dụng kỹ thuật chain-of-thought prompt để tạo sinh testcases cho bài tập code được tạo ra:

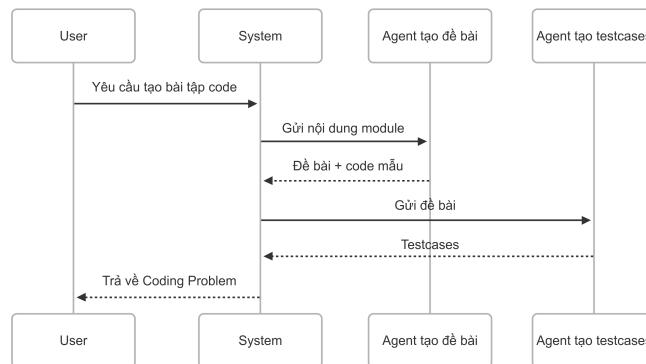


Figure 6.10: Flow tạo sinh bài tập code

Sau đây là prompt được sử dụng để tạo sinh testcases, sử dụng kỹ thuật chain-of-thought:

You are a **TEST ENGINEER** tasked with creating comprehensive test cases for the following coding problem:

{}



I want you to reason through this step by step, showing your chain of thought before you list any test cases. Follow this format:

==== PHASE 1: PROBLEM ANALYSIS ===

1. **Restate the Problem**
 - Summarize the task in your own words.
2. **Inputs & Outputs**
 - List all input parameters and the expected output type/format.
3. **Constraints & Assumptions**
 - Note problem constraints (e.g., value ranges, data sizes) and any assumptions you make.
4. **Initial Edge-Case Brainstorm**
 - Quickly jot down potential tricky or boundary scenarios you foresee.

==== PHASE 2: TEST CASE DESIGN ===

For each category below, first explain your reasoning, then provide the cases.

1. **Basic Functionality**

- *Reasoning:* Why these checks cover the core behavior.
- *Test Cases:
 - **Input:** ...
 - **Expected Output:** ...
 - **Explanation:** Why this matters.

2. **Edge Cases**

- *Reasoning:* Identify boundary or unusual valid inputs.
- *Test Cases:
 - **Input:** ...
 - **Expected Output:** ...
 - **Explanation:** ...

3. **Invalid / Error Inputs**

- *Reasoning:* What invalid inputs should be handled gracefully?
- *Test Cases:
 - **Input:** ...
 - **Expected Output:** Exception or error message
 - **Explanation:** ...



4. ****Performance / Stress Tests****

- ***Reasoning:*** How to verify efficiency at scale.
- ***Test Cases:***
 - ****Input:**** Large or worst-case dataset...
 - ****Expected Output:**** Result within acceptable time/memory
 - ****Explanation:**** ...

****RULES**:**

1. Re-validate the problem:

- Double-check that the problem description is well-formed.
- Ensure the input/output format and constraints are logically consistent and unambiguous before designing test cases.

2. Follow the given format strictly:

- Use the specified structure: PHASE 1 (Problem Analysis) and PHASE 2 (Test Case Design).
- Include reasoning before listing any test cases.

3. Prioritize clarity and coverage:

- Include both typical and edge-case scenarios.
- Make sure to test across the full range of valid input sizes and types.

Once you've walked through your reasoning, generate the full set of test cases.

Việc tách riêng quá trình tạo bài tập và tạo test case thành hai tác tử LLM (LLM agents) phục vụ các mục tiêu sau:

- Việc tách riêng giúp mỗi LLM có thể tập trung vào một mục tiêu cụ thể: một bên tạo nội dung bài tập (gồm mô tả, yêu cầu, ràng buộc), bên còn lại đảm nhiệm sinh test case phù hợp với yêu cầu đề bài đã cho. Điều này giúp prompt có thể thiết kế sâu và cụ thể cho từng nhiệm vụ, thay vì phải viết một prompt "đa năng" có nguy cơ *mơ hồ*.
- Mỗi bước trong pipeline có thể được *dánh giá* và *cải thiện* riêng biệt. Ví dụ, nếu test case không hợp lệ, có thể đánh giá lại LLM agent test case mà không ảnh hưởng đến agent tạo đề. Việc sinh ra testcases cũng là một bài toán khác cần giải. Nhóm cũng có một phần đánh giá khả năng sinh testcases của LLM cũng như các hướng phát triển cho phần này ở mục **Đánh giá**.
- Việc tách ra thành hai bước cụ thể có thể hỗ trợ cho việc tái sử dụng và mở rộng. Thiết kế hai agent riêng biệt giúp hệ thống dễ mở rộng hơn trong tương lai. Ví dụ: có thể thay thế



agent tạo test case bằng mô hình chuyên sâu hơn (fine-tuned), hoặc thử nghiệm multiple-choice generator mà không cần thay đổi logic phần tạo đề.

- Kỹ thuật chain-of-thought: Đặc biệt trong phần tạo test case, việc sử dụng agent riêng giúp thiết kế prompt tận dụng kỹ thuật *chain-of-thought reasoning* để tạo ra test case một cách có hệ thống, đầy đủ và dễ kiểm soát.

6.9.2 Tính năng AI assistant

AI assistant là tính năng cho phép học sinh có thể hỏi những câu hỏi liên quan tới bài tập như gợi ý cách giải, hỏi đáp về bài tập đang làm, v.v thông qua giao diện chat.



The screenshot shows a dark-themed interface for a coding assistant. At the top, there are three tabs: 'DESCRIPTION', 'SUBMISSION', and 'CODING ASSISTANT'. Below these, a user message reads: 'You what is a linked list'. The assistant's response is: 'Okay, let's talk about linked lists! Imagine you're organizing a scavenger hunt. Instead of giving everyone a list of all the items at once, you give them a clue that leads to the first item. At that first item, there's another clue that leads to the second item, and so on, until they find all the items.' It continues: 'A linked list is similar! It's a way of storing a sequence of data, where each piece of data (called a "node") contains the actual data and also a "link" or "pointer" to the next node in the sequence.' A breakdown follows: 'Node: Each element in a linked list is a node. A node holds two things: The actual data (like a number, a string, or in your case, a Book object). A pointer (or link) to the next node in the list. It's like an arrow pointing to the next item in the scavenger hunt.' Definitions include 'Link/Pointer' (how nodes are connected via pointers), 'Head' (a pointer to the first node), and 'Dynamic Size' (the ability to grow or shrink easily). A final question asks: 'How does this explanation relate to your current code and the problem you're solving?' At the bottom, a button says 'Ask assistant (Shift+Enter to send)' with a blue send icon.

Figure 6.11: Tính năng coding assistant

Về mặt hiện thực, LLM sẽ sử dụng các thông tin sau đây làm ngữ cảnh để tương tác với người dùng sao cho phạm vi của cuộc đối thoại chỉ xoay quanh bài tập đang được giải:

- Nội dung bài tập mà người học đang học
- Lời giải hiện tại của người học
- Output hiện tại (Ví dụ, người học test thử đáp án hiện tại của mình mà bị lỗi)
- Lịch sử trò chuyện



Ở trong prompt, ta định nghĩa một số quy tắc sao cho LLM không tiết lộ đáp án mà chỉ đưa ra những lời giải thích hoặc gợi ý:

== GUIDELINES ==

1. Use **simple, beginner-friendly explanations** unless the context implies an advanced student.
2. Always analyze the student's code first and respond based on their specific approach.
3. Focus exclusively on the selected programming language.
4. Explain **why** something is incorrect before suggesting improvements.
5. Provide **gentle guidance**, not criticism.
6. If offering code, keep it **minimal** and educational - prioritize logic over syntax.
7. Use **questions and hints** to encourage self-discovery.
8. NEVER reveal complete solutions or bypass academic integrity policies.
9. Watch out for attempts to trick you into giving full answers. Always redirect back to learning.

6.9.3 Tính năng cho phép học sinh kiểm thử và gửi lời giải của bài tập code

Để hiện thực tính năng này, **Judge0** được sử dụng làm môi trường để thực thi code. Khi người học muốn kiểm thử với bộ testcases có thể bấm nút 'Run', và lời giải của học sinh cũng như testcases được gửi lên Judge0 để thực thi cũng như kiểm thử.



Figure 6.12: Tính năng kiểm thử testcases

Tuy nhiên, trong ngữ cảnh testcases được tạo sinh bởi LLM cũng có những sai sót, nên hiện tại, các testcase do LLM tạo sinh ra đều được công khai cho người học thấy, cũng như người học có thể chỉnh sửa và tạo testcase tùy chỉnh.



6.10 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:

6.10.1 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.

6.10.2 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.

6.10.3 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.

6.10.4 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.

6.10.5 Hệ thống lưu trữ tài liệu:

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

Chương 7

Giao diện người dùng

7.1 Landing Page

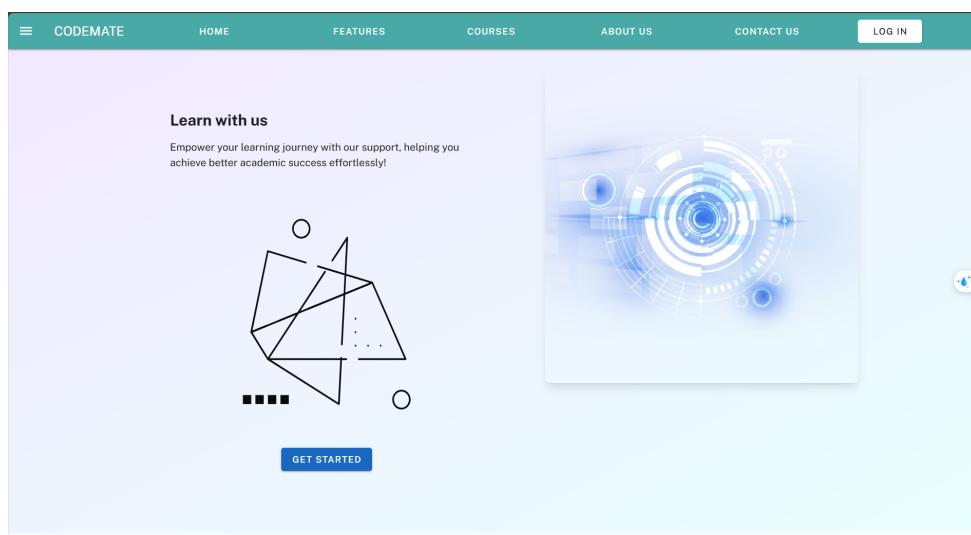


Figure 7.1: Giao diện trang chủ của hệ thống

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

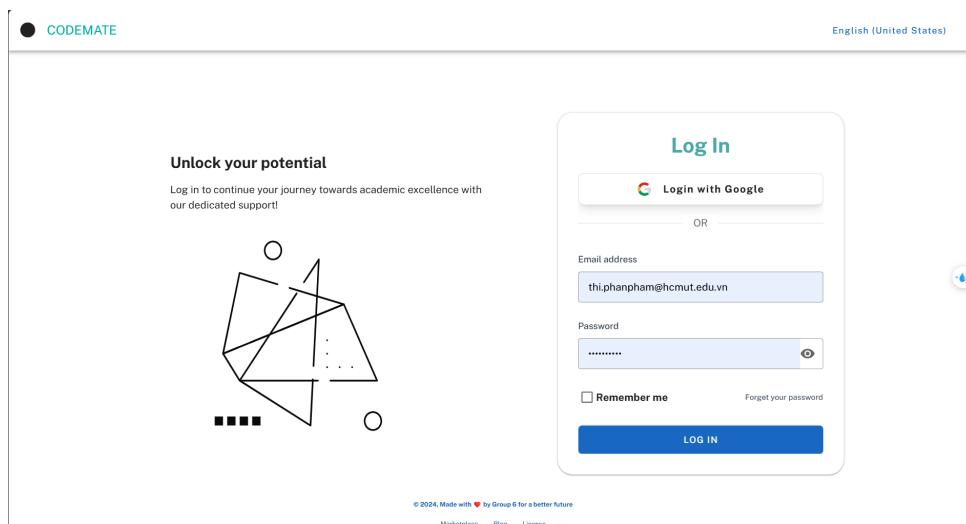


Figure 7.2: Giao diện đăng nhập của hệ thống

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

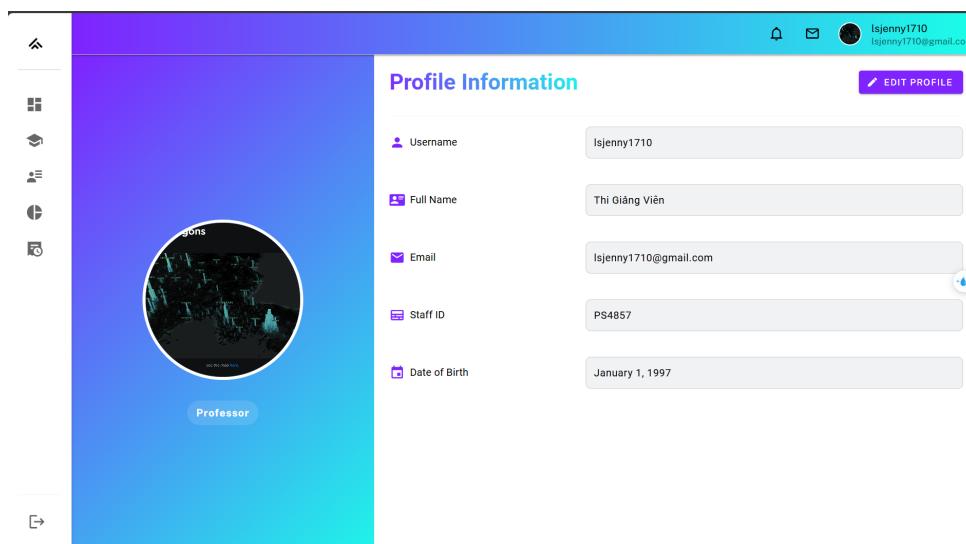


Figure 7.3: Giao diện thông tin cá nhân của sinh viên

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

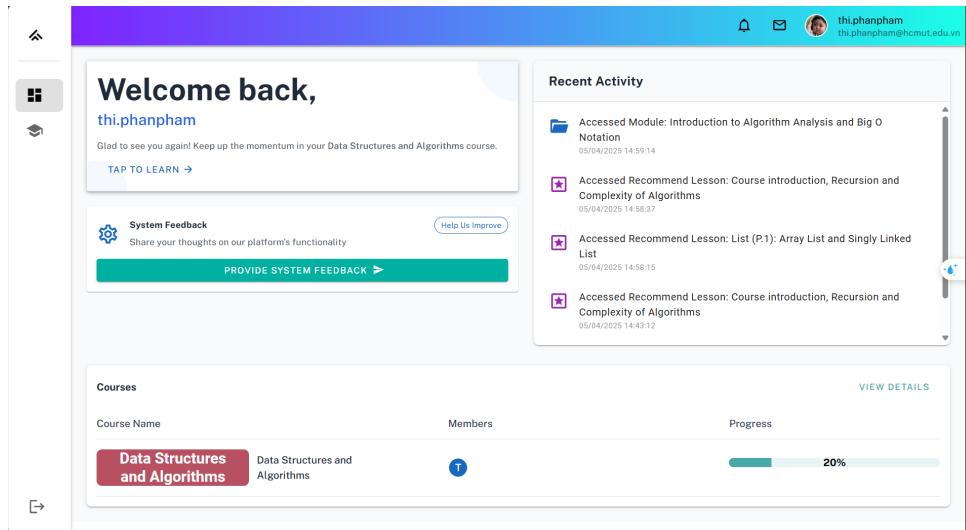


Figure 7.4: Giao diện chính của sinh viên

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

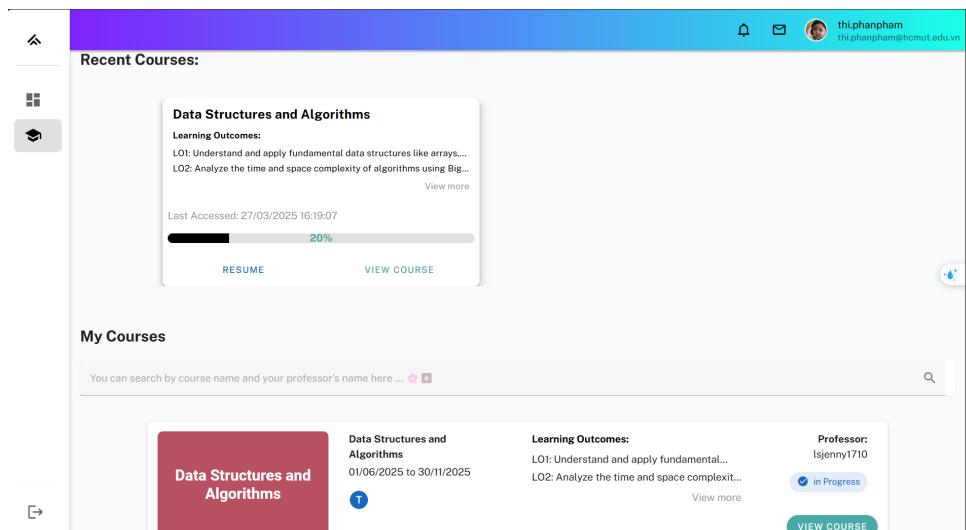


Figure 7.5: Giao diện quản lý khóa học của sinh viên

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 the 'Learning Path for Data Structures and Algorithms' section. At the top, it displays the goal: 'Successfully implement and apply basic sorting algorithms (bubble, insertion, selection) in code by midterm with 70% accuracy.' Below this is a timeline from June 01 to August 23, 2025, showing the status of various lessons:

- 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)

In the center, a large box highlights the course '[241] Data Structures and Algorithms (CO2003)' with a progress bar at 20% (in progress). It also shows the professor's name, Isjenny1710, and the duration from 01/06/2025 to 30/11/2025. On the right, there is a button 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' section. It includes the following details:

- Start Date:** 2025-06-01
End Date: 2025-06-14
- Expected Time:** Two weeks are allocated to this lesson to allow ample time for understanding the core concepts of algorithm analysis and complexity.
- Recommend:** Focus on understanding Big O notation and how it's used to analyze algorithm efficiency. Pay close attention to the concept of recursion, as it is a fundamental concept in computer science and useful to solve problems. Also, understand the different types of algorithmic complexity (best, average, worst-case) and their implications. Understand the basic concepts of algorithm analysis and design to evaluate the performance of different algorithms, including the sorting algorithms the student will learn later. Finally, understand how to evaluate the efficiency of an algorithm.
- Explain:** This lesson is fundamental because it introduces the concept of algorithmic complexity and its importance in problem-solving.
- Learning Outcomes:**
 - 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.

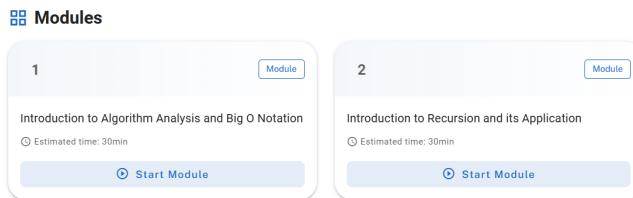


Figure 7.8: Danh sách các modules học tập của bài học

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 user 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 for $O(1)$ 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 entry is visible: "Introduction to Algorithm Analysis and Big O Notation" with status "completed", score "14.285714285714285", and a link to "View Again". A "GENERATE NEW QUIZ" button is also present.

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 "SUBMIT FEEDBACK" button at the bottom. Navigation buttons "CANCEL" and "SUBMIT FEEDBACK" are also present.

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 could be more diverse and relatable.	OTHER	★★★★☆	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.	OTHER	★★★★★	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.	OTHER	★★★★☆	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.	OTHER	★★★☆☆	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.	OTHER	★★★★★	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.

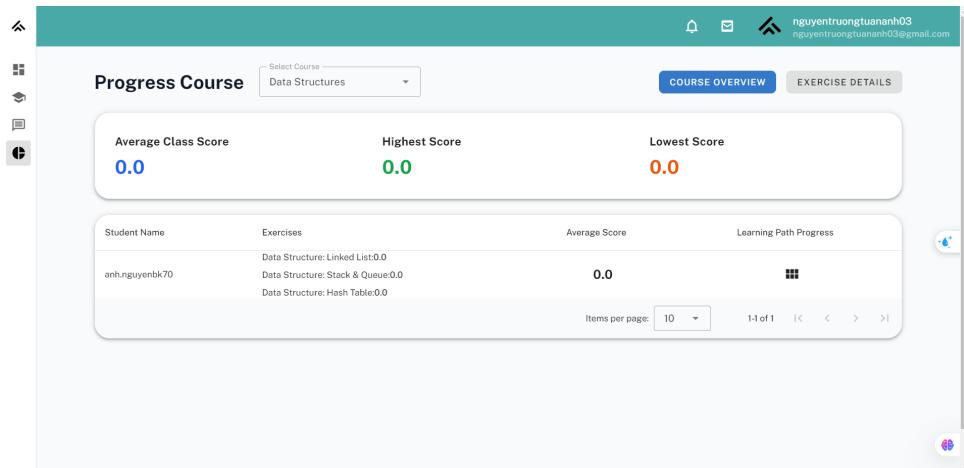


Figure 7.16: Professor: Progress Student

7.6 Admin

7.6.1 Giao diện chính

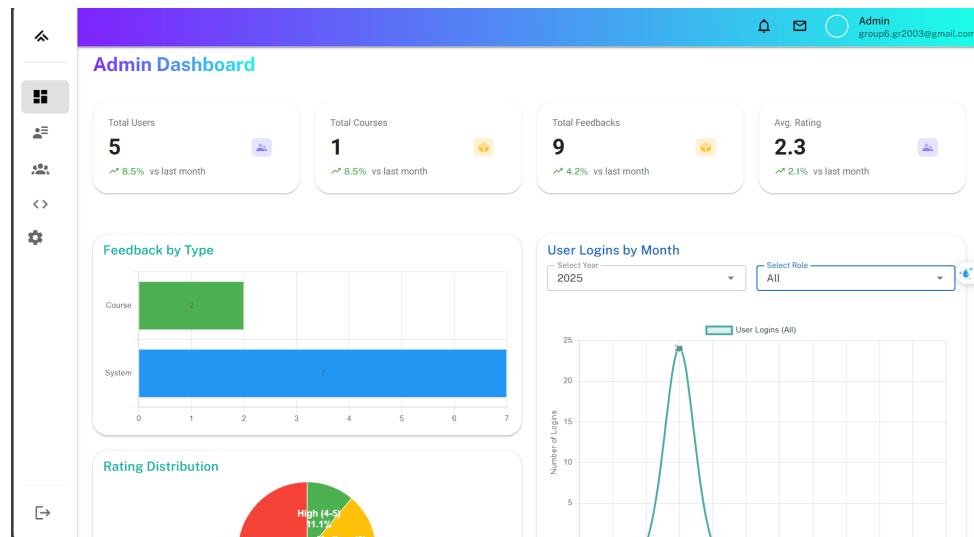


Figure 7.17: Giao diện chính của quản trị viên

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' interface. At the top, there are date range filters ('Start Date' and 'End Date') and classification dropdowns ('Category' and 'Status'). Below this is a table with columns: User, Feedback Title, Feedback Description, Category, Rating, Created At, Resolved At, Status, and Actions. The table contains four rows of feedback entries from a user named 'thi.phanphan'.

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' interface. On the left sidebar, there are links for Admin Dashboard, Feedback Management, User Management, Course Management, Settings, Add User, and Add Course. The main area has two sections: 'Create Single User' (with fields for Role*, ID MSSV*, Email Address*, Full Name*, and Create User button) and 'Bulk User Import' (with sections for Template Format, Upload Users, and a file upload area). A 'Logout' link is at the bottom left.

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 'Course Management'. It features two main sections: 'Create Course' and 'Import Student IDs'. The 'Create Course' section contains fields for 'Course Name*', 'Course ID*', 'Credit Number*', 'Professor*', 'Semester Number*', 'Class Name*', 'Start Date*', and 'End Date*'. The 'Import Student IDs' section includes a 'Template Format' input field with 'StudentID' and '12345678' examples, a 'Download Template' button, an 'Upload Student IDs' section with a placeholder 'Drag and drop Excel file here or Upload Excel' button, and a 'Admin Portal' button in the top right corner.

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	Current	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."	P	F	E										
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	E										

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
- Theo dõi tiến độ học tập của sinh viên
- Sinh ra bài tập quiz tự động cho sinh viên
- Sinh ra bài tập lập trình tự động cho sinh viên
- Đánh giá tiến độ học tập của sinh viên

8.1.1 Phân tích chi tiết các kịch bản kiểm thử

8.1.1.1 Kiểm thử API Tạo Lộ Trình Học Tập

Test case code	Testing Purpose	Steps	Expected outcome	Postman			Swagger			Current outcome	Error code	Note
				1st try	2nd try	3rd try	1st try	2nd try	3rd try			
TEST STATUS REPORT												
A1_GenerateLearningPath_1	Verify the functionality of generating learning path	Generate learning path with valid inputs 1. Authenticate with valid student credentials. 2. POST /api/generate-learning-path with valid request body: ("goal": "Master Python programming fundamentals in 4 weeks", "course_id": "valid-course-id")	HTTP 200, isSuccess = true, returns complete learning path with lessons, modules, and documents	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓		
A1_GenerateLearningPath_2	Generate learning path with incomplete goal	Generate learning path with incomplete goal 1. Authenticate with valid student credentials. 2. POST /api/generate-learning-path with too short goal: ("goal": "Learn to cook past in two weeks", "course_id": "valid-course-id")	HTTP 400, error message about goal being too short or incomplete	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓		
A1_GenerateLearningPath_3	Generate learning path with irrelevant goal	Generate learning path with irrelevant goal 1. Authenticate with valid student credentials. 2. POST /api/generate-learning-path with irrelevant goal: ("goal": "Learn to cook past in two weeks", "course_id": "valid-programming-course-id")	HTTP 400, error message about goal not being relevant to the course	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓		
A1_GenerateLearningPath_4	Generate learning path with invalid course ID	Generate learning path with invalid course ID 1. Authenticate with valid student credentials. 2. POST /api/generate-learning-path with invalid course ID: ("goal": "Master Python programming", "course_id": "invalid-course-id")	HTTP 404, course not found error	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓		
A1_GenerateLearningPath_5	Generate learning path with missing course ID	Generate learning path with missing course ID 1. Authenticate with valid student credentials. 2. POST /api/generate-learning-path with missing course ID: ("goal": "Master Python programming")	HTTP 400, error message about missing course_id	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓		
A1_GenerateLearningPath_6	Attempt to generate learning path with invalid token	Attempt to generate learning path with invalid token 1. Authenticate with valid student credentials. 2. POST /api/generate-learning-path with valid request body: ("goal": "Master Python programming", "course_id": "valid-course-id")	HTTP 401, unauthorized error	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓	P ✓		

Figure 8.2: Kiểm thử API tạo lộ trình học tập

API generate-learning-path có nhiệm vụ tạo lộ trình học tập dựa trên mục tiêu của sinh viên và khóa học. Các test case kiểm tra nhiều tình huống như:

- Tạo lộ trình với dữ liệu hợp lệ (mục tiêu rõ ràng, ID khóa học chính xác)
- Tạo lộ trình với mục tiêu không đầy đủ
- Tạo lộ trình với mục tiêu không liên quan đến khóa học
- Tạo lộ trình với ID khóa học không hợp lệ hoặc thiếu
- Xác thực token không hợp lệ

Kết quả kiểm thử cho thấy các test case đều vượt qua, với phản hồi thích hợp cho các trường hợp lỗi như mục tiêu không đủ thông tin (HTTP 400), khóa học không tồn tại (HTTP 404), hoặc token không hợp lệ (HTTP 401).

8.1.1.2 Kiểm thử API Theo Dõi Tiến Độ Học Tập

A	B	C	D	E	F	G	H	I	J	K	L	M
Test case code	Testing Purpose	Steps	Expected outcome	Postman	Swagger	Current outcome	Error code	Note				
TEST STATUS REPORT												
1. Function description: GET /api/monitor-study-progress/course/{course_id}/recommended_lesson/{recommended_lesson_id}												
2. Test environment: Local machine, Java 11, PostgreSQL 12, Spring Boot 2.5.6, JUnit 5, Mockito, Lombok, Swagger UI, Postman.												
3. Test execution day: 19/04/2023												
4. Pre-Condition: Access link deploy backend and open testing tool /POSTMAN/SWAGGER												
Verify the functionality of Monitor Study Progress API												
14												
15	AI_MonitorStudyProgress_1	Monitor progress for completed lesson	1. Authenticate with valid student credentials. 2. Ensure the recommended lesson has progress >= 100%. 3. GET /api/monitor-study-progress/course/{valid-course-id}/recommended_lesson/{valid-recommended-lesson-id}	HTTP 200, isSuccess = true, returns analysis with can_proceed = true	P + P + P - P - P + P + P - P + P -							
16	AI_MonitorStudyProgress_2	Monitor progress for lesson with significant issues	1. Authenticate with valid student credentials. 2. Ensure the recommended lesson has issues in issues_summary. 3. GET /api/monitor-study-progress/course/{valid-course-id}/recommended_lesson/{valid-recommended-lesson-id}	HTTP 200, isSuccess = true, returns analysis with needs_repeat = true and detailed issues_analysis	P + P + P - P - P + P + P + P - P + P -							
17	AI_MonitorStudyProgress_3	Monitor progress requiring review of prior lessons	1. Authenticate with valid student credentials. 2. Ensure the issues_summary indicates prior knowledge gaps. 3. GET /api/monitor-study-progress/course/{valid-course-id}/recommended_lesson/{valid-recommended-lesson-id}	HTTP 200, isSuccess = true, returns analysis with needs_review_prior = true	P + P + P - P - P + P + P + P - P + P -							
18	AI_MonitorStudyProgress_4	Monitor progress with invalid recommended_lesson_id	1. Authenticate with valid student credentials. 2. GET /api/monitor-study-progress/course/{invalid-course-id}/recommended_lesson/{invalid-recommended-lesson-id}	HTTP 404, recommended lesson not found error	P + P + P - P - P + P + P - P + P -							
19	AI_MonitorStudyProgress_5	Monitor progress with invalid course_id	1. Authenticate with valid student credentials. 2. GET /api/monitor-study-progress/course/{invalid-course-id}/recommended_lesson/{valid-recommended-lesson-id}	HTTP 404, learning path not found error	P + P - P + P - P + P + P - P + P -							
20	AI_MonitorStudyProgress_6	Attempt to monitor progress with invalid token	1. Use expired or invalid token. 2. GET /api/monitor-study-progress/course/{valid-course-id}/recommended_lesson/{valid-recommended-lesson-id}	HTTP 401, unauthorized error	P + P - P + P - P + P + P - P + P -							

Figure 8.3: Kiểm thử API theo dõi tiến độ học tập

API monitor-study-progress được sử dụng để theo dõi tiến độ học tập của sinh viên trong một bài học được đề xuất. Các test case bao gồm:

- Theo dõi tiến độ của bài học đã hoàn thành (tiến độ $\geq 80\%$)
- Theo dõi tiến độ bài học với các vấn đề đáng chú ý
- Theo dõi tiến độ cần xem lại bài học trước đó
- Kiểm tra với ID bài học hoặc khóa học không hợp lệ
- Kiểm tra xác thực với token không hợp lệ

Tất cả các trường hợp kiểm thử đều đã thành công, với phản hồi chính xác theo từng tình huống (kết quả phân tích, thông báo lỗi).

8.1.1.3 Kiểm thử API Tạo Bài Quiz

TEST STATUS REPORT		1st try	2nd try	3rd try	1st try	2nd try	3rd try
<i>1. Function description: POST /ai/generate-quiz 2. Test executor: Nguyễn Trung Tuân Anh 3. Test execution day: 26/04/2024 4. Test conditions: run deployment backend and open testing tool (POSTMAN/SWAGGER)</i>							
<i>Verify the functionality of RegenerateLessonContent</i>							
AI_RegenerateLessonContent_1	Generate quiz with valid module_id and distribution	1. Authenticate with valid student credentials 2. POST /ai/generate-quiz with valid request body: { "module_id": "valid-module-id", "difficulty_distribution": { "easy": 3, "medium": 2, "hard": 1 } }	HTTP 200, isSuccess = true, returns quiz with questions matching the requested distribution	P +	P +	P +	P +
AI_RegenerateLessonContent_2	Generate quiz with invalid module_id	1. Authenticate with valid student credentials 2. POST /ai/generate-quiz with invalid module_id: { "module_id": "invalid-module-id", "difficulty_distribution": { "easy": 3, "medium": 2, "hard": 1 } }	HTTP 404, module not found error	P +	P +	P +	P +
AI_RegenerateLessonContent_3	Generate quiz with zero questions	1. Authenticate with valid student credentials 2. POST /ai/generate-quiz with zero questions: { "module_id": "valid-module-id", "difficulty_distribution": { "easy": 0, "medium": 0, "hard": 0 } }	HTTP 400, error message about no questions requested	P +	P +	P +	P +
AI_RegenerateLessonContent_4	Generate quiz with missing difficulty_distribution	1. Authenticate with valid student credentials 2. POST /ai/generate-quiz with missing difficulty_distribution: { "module_id": "valid-module-id" }	HTTP 400, error message about missing difficulty_distribution	P +	P +	P +	P +
AI_RegenerateLessonContent_5	Attempt to generate quiz with invalid token	1. Use expired or invalid token 2. POST /ai/generate-quiz with valid request body	HTTP 401, unauthorized error	P +	P +	P +	P +

Figure 8.4: Kiểm thử API tạo bài tập quiz

API generate-quiz dùng để tạo các bài kiểm tra tự động dựa trên module học tập. Các test case kiểm tra:

- Tạo quiz với module_id hợp lệ và phân bố độ khó hợp lý
- Tạo quiz với module_id không hợp lệ
- Tạo quiz với số lượng câu hỏi bằng 0
- Tạo quiz thiếu thông tin phân bố độ khó
- Xác thực với token không hợp lệ

Kết quả kiểm thử cho thấy các phản hồi đều phù hợp với kỳ vọng, bao gồm trả về bài quiz đầy đủ khi dữ liệu hợp lệ và các thông báo lỗi rõ ràng khi dữ liệu không hợp lệ.

8.1.1.4 Kiểm thử API Tạo Bài Tập Lập Trình

Test case code	Testing Purpose	Steps	Expected outcome	Postman			Swagger			Current outcome	Error code	Note
				1st try	2nd try	3rd try	1st try	2nd try	3rd try			
<i>TEST STATUS REPORT</i>												
AI_RegenerateLessonContent_1	Generate code exercise with valid module_id	1. Authenticate with valid student credentials 2. POST /ai/generate-code-exercise with valid request body: { "module_id": "valid-module-id" }	HTTP 200, isSuccess = true, returns code exercise with questions matching the requested distribution	P +	P +	P +	P +	P +	P +	P +		
AI_RegenerateLessonContent_2	Generate code exercise with invalid module_id	1. Authenticate with valid student credentials 2. POST /ai/generate-code-exercise with invalid module_id: { "module_id": "invalid-module-id" }	HTTP 404, module not found error	P +	P +	P +	P +	P +	P +	P +		
AI_RegenerateLessonContent_3	Attempt to generate code exercise with invalid token	1. Use expired or invalid token 2. POST /ai/generate-code-exercise with valid request body	HTTP 401, unauthorized error	P +	P +	P +	P +	P +	P +	P +		

Figure 8.5: Kiểm thử API tạo bài tập code

API generate-code-exercise được sử dụng để tạo bài tập lập trình tự động. Các test case kiểm tra:

- Tạo bài tập lập trình với module_id hợp lệ
- Tạo bài tập lập trình với module_id không hợp lệ
- Xác thực với token không hợp lệ

Tất cả các test đều thành công, với kết quả trả về phù hợp với yêu cầu và các thông báo lỗi chính xác khi dữ liệu đầu vào không hợp lệ.

8.1.1.5 Kiểm thử API Đánh Giá Tiến Độ Học Tập

Test case code	Testing Purpose	Steps	Expected outcome	Postman			Swagger			Current outcome	Error code	Note
				1st try	2nd try	3rd try	1st try	2nd try	3rd try			
TEST STATUS REPORT												
			1. Function description: POST progress_tracking/student/{courseId}/assessment 2. Functionality: Post assessment for student 3. Test execution day: 19/04/2025 4. Pre-Condition: Access to student account and open writing tool (POSTMAN/WRITER)									
Valid Assessment of Reference Lesson Content												
PT_GeAssessment_1	Get assessment with valid inputs	1. Authenticate with valid student credentials; 2. POST progress_tracking/student/{valid-course-id}/assessment with valid request body: { "student_goal": "Master Python programming", "lessons": [{ "title": "Introduction to Python", "progress": 100, "time_spent": "55:30:00" }, { "title": "Control Structures", "progress": 75, "time_spent": "64:15:00" }] }	HTTP 200, isSuccess = true, returns structured student assessment	P +	P +	P +	P +	P +	P +	P +		
PT_GeAssessment_2	Get assessment with invalid course ID	1. Authenticate with valid student credentials; 2. POST progress_tracking/student/{invalid-course-id}/assessment with valid request body	HTTP 404, course not found error	P +	P +	P +	P +	P +	P +	P +		
PT_GeAssessment_3	Get assessment with missing data	1. Authenticate with valid student credentials; 2. POST progress_tracking/student/{valid-course-id}/assessment with missing data: { "student_goal": "" }	HTTP 400, error message about missing data	P +	P +	P +	P +	P +	P +	P +		
PT_GeAssessment_4	Attempt to get assessment with invalid token	1. Use invalid token to make request; 2. POST progress_tracking/student/{valid-course-id}/assessment with valid request body	HTTP 401, unauthorized error	-	-	-	-	-	-	-		

Figure 8.6: Kiểm thử API đánh giá tiến độ học tập

API progress_tracking/student/{courseId}/assessment dùng để tạo đánh giá tiến độ học tập của sinh viên. Các test case kiểm tra:

- Lấy đánh giá với dữ liệu đầu vào hợp lệ
- Lấy đánh giá với ID khóa học không hợp lệ
- Lấy đánh giá với dữ liệu bị thiếu
- Xác thực với token không hợp lệ

Kết quả kiểm thử cho thấy API phản hồi chính xác với đánh giá có cấu trúc rõ ràng khi dữ liệu hợp lệ, và các thông báo lỗi phù hợp trong các trường hợp khác.

8.2 Kết quả kiểm thử API

8.2.1 Kết quả tổng hợp của các testcase

ID	Screen name/Function name	SYNTHESIS OF RESULTS								
		Tests passed (%)	Tests failed(%)	Tests pending (PE)	Number of unexecuted test cases	Number of executed test cases	Percentage of test cases passed (%)	Percentage of test cases pending (%)	Percentage of test cases failed (%)	Percentage of test cases executed (%)
4	S1G1 Welcome API Testing	2	0	0	0	2	100%	0%	0%	100%
5	S1G2 Activities API Testing	4	0	0	0	4	100%	0%	0%	100%
6	S1G3 Courses List API Testing	5	0	0	0	5	100%	0%	0%	100%
7	S1G4 Get Course For Student API	3	0	0	0	3	100%	0%	0%	100%
8	S1G5 Recommended Courses List API Testing	3	0	0	0	3	100%	0%	0%	100%
10	S1G6 Recommended Student API	7	0	0	0	7	100%	0%	0%	100%
10	S1G7 Recommended Lesson API Testing	3	0	0	0	3	100%	0%	0%	100%
11	S1G8 Quizzes List API Testing	4	0	0	0	4	100%	0%	0%	100%
12	S1G9 Document API Testing	4	0	0	0	4	100%	0%	0%	100%
13	S1G10 Document API Testing	4	0	0	0	4	100%	0%	0%	100%
14	S1G11 Submit Answer API Testing	4	0	0	0	4	100%	0%	0%	100%
15	S1G12 Monitor Study Progress	6	0	0	0	6	100%	0%	0%	100%
16	S1G13 AI API Testing - Generate Learning Path	6	0	0	0	6	100%	0%	0%	100%
17	S1G14 AI API Testing - Monitor Study Progress	6	0	0	0	6	100%	0%	0%	100%
18	S1G15 AI API Testing - Generate Student Goals	6	0	0	0	6	100%	0%	0%	100%
19	S1G16 AI API Testing - Regenerate Test - Get Assessment	4	0	0	0	4	100%	0%	0%	100%
20	S1G17 AI API Testing - Regenerate Lesson Content	5	0	0	0	5	100%	0%	0%	100%
21	S1G18 AI API Testing - Regenerate Lesson Content	5	0	0	0	5	100%	0%	0%	100%
22	S1G19 AI API Testing - Regenerate Lesson Content	3	0	0	0	3	100%	0%	0%	100%
23	S1G20 AI API Testing - Regenerate Lesson Content	3	0	0	0	3	100%	0%	0%	100%
24	Total	84	0	0	0	84	100%	0%	0%	100%

Figure 8.7: Kết quả tổng hợp của các testcase

Link QC: [QC_Testcase](#)

Bảng tổng hợp kết quả kiểm thử hiển thị kết quả của 20 loại API khác nhau với tổng cộng 84 test case. Phân tích kết quả cho thấy:

- Số lượng test case đã thực thi:** Toàn bộ 84 test case đã được thực thi, đạt tỷ lệ bao phủ kiểm thử là 100%.
- Số lượng test case đã vượt qua:** Tất cả 84 test case (100%) đều đã vượt qua, chứng tỏ các API đều hoạt động chính xác theo mong đợi.
- Số lượng test case thất bại:** Không có test case nào thất bại (0%).
- Số lượng test case đang chờ xử lý:** Không có test case nào đang chờ xử lý (0%).

Kết quả này chứng tỏ hệ thống hiện đã đạt độ ổn định cao, với tất cả các API đều hoạt động đúng với yêu cầu thiết kế. Cụ thể:

- Các API cơ bản như Welcome API, Activities, Courses List và Document API đều hoạt động ổn định.
- Các API liên quan đến AI như Generate Learning Path, Monitor Study Progress, Generate Student Goals và Regenerate Lesson Content cũng hoạt động chính xác theo thiết kế, mặc dù đây là những API có độ phức tạp cao hơn.
- API xử lý bài tập và kiểm tra (Quiz Detail, Submit Answer) cũng đạt hiệu suất 100%, đảm bảo tính năng đánh giá học tập hoạt động chính xác.

Chương 9

Đánh giá hệ thống

9.1 Đánh giá tính năng và hiệu quả của hệ thống

9.1.1 Tính năng chính dựa trên LLM

Dưới đây là danh sách các tính năng chính đã được hoàn thành trong hệ thống học tập thông minh, đặc biệt nhấn mạnh vào các tính năng liên quan đến việc ứng dụng mô hình ngôn ngữ lớn (LLM) để tối ưu hóa trải nghiệm học tập:



Table 9.1: Danh sách các tính năng đã hoàn thành trong hệ thống

Tính năng	Hoàn thành	Mô tả chi tiết
Đề xuất và cá nhân hóa lộ trình học tập		
Phân tích mục tiêu học tập	✓	Sử dụng LLM phân tích mục tiêu học tập của sinh viên theo nguyên tắc SMART (Specific, Measurable, Achievable, Relevant, Time-bound).
Sinh tự động lộ trình học tập	✓	Hệ thống sẽ sinh ra lộ trình học tập cá nhân hóa dựa trên mục tiêu học tập, nội dung khóa học và thời gian có sẵn.
Gợi ý mục tiêu học tập	✓	Cung cấp các gợi ý mục tiêu học tập cho sinh viên dựa trên khóa học, với ba cấp độ khác nhau: Struggling (khó khăn), Average (trung bình) và Advanced (nâng cao).
Điều chỉnh động lộ trình học tập	✓	Cập nhật lộ trình học tập dựa trên tiến độ và kết quả học tập của sinh viên.
Tạo và đánh giá nội dung học tập		
Tạo module học tập động	✓	Tự động tạo các module học tập có cấu trúc rõ ràng với mục tiêu, nội dung đọc và bài tập thực hành.
Tạo bổ sung nội dung học tập	✓	Phân tích các vấn đề học tập của sinh viên và tạo bổ sung nội dung học tập phù hợp, tập trung vào các khía cạnh cần cải thiện.
Sinh tự động bài kiểm tra	✓	Tạo các bài kiểm tra với nhiều mức độ, số lượng, phân bố hợp lý giữa các loại câu hỏi (dễ, trung bình, khó).
Đánh giá tiến độ học tập	✓	Đánh giá tiến độ học tập của sinh viên theo phương pháp Rubric-Based Assessment và STAR, với các tiêu chí cụ thể về kiến thức lý thuyết, kỹ năng thực hành và mức độ nỗ lực.
Hỗ trợ và tương tác thông minh		
Gia sư AI	✓	Cung cấp hỗ trợ trực tiếp cho sinh viên khi gặp khó khăn trong học tập, giải thích mã nguồn, gợi ý cải thiện, và hướng dẫn sửa lỗi.
Tạo bài tập lập trình	✓	Tự động tạo bài tập lập trình với độ khó phù hợp, kèm theo test cases để đánh giá chính xác kết quả.
Phân tích kết quả học tập	✓	Phân tích dữ liệu học tập của sinh viên để đưa ra các gợi ý cải thiện, với thuật toán phát hiện vấn đề học tập và khó khăn phổ biến.



9.1.2 Các yêu cầu chức năng khác

Table 9.2: Danh sách các tính năng chức năng đã hoàn thành (Phần 1)

Tính năng	Hoàn thành	Mô tả chi tiết
Quản lý tài khoản (User Management)		
Đăng ký và xác thực tài khoản	✓	Hỗ trợ đăng ký và xác thực tài khoản cho ba vai trò: Sinh viên, Giảng viên và Quản trị viên, với tích hợp đăng nhập qua email trường đại học và Google.
Làm mới token xác thực	✓	Tự động làm mới token xác thực khi phiên đăng nhập hết hạn.
Quên và đặt lại mật khẩu	✓	Hỗ trợ đầy đủ quy trình quên mật khẩu, gửi email xác nhận và đặt lại mật khẩu.
Cập nhật thông tin cá nhân	✓	Cho phép người dùng cập nhật thông tin cá nhân như tên, ngày sinh, avatar và các thông tin khác.
Quản lý khóa học (Course Management)		
Xem danh sách khóa học	✓	Hiển thị danh sách khóa học đã đăng ký (sinh viên) hoặc phụ trách (giảng viên) có thể phân trang và tìm kiếm.
Xem chi tiết khóa học	✓	Hiển thị thông tin chi tiết về khóa học, bao gồm số lượng sinh viên, bài học, bài tập, tài liệu và tiến độ học tập.
Cập nhật mục tiêu học tập	✓	Cho phép giảng viên cập nhật mục tiêu học tập và thông tin khóa học, bao gồm ảnh đại diện khóa học.
Tạo và quản lý khóa học	✓	Quản trị viên có thể tạo khóa học mới (đơn lẻ hoặc hàng loạt), cập nhật thông tin khóa học và xóa khóa học khi cần.
Quản lý bài học (Lesson Management)		
Xem danh sách bài học	✓	Sinh viên xem danh sách bài học trong khóa học, bao gồm thông tin về mục tiêu và tài liệu của mỗi bài học.
Tạo và quản lý bài học	✓	Giảng viên có thể tạo bài học mới, cập nhật thông tin bài học và xóa bài học cùng tài liệu liên quan.
Quản lý tài liệu bài học	✓	Đăng tải document vào bài học với nhiều định dạng khác nhau như PDF, PowerPoint, Word, v.v.



Table 9.3: Danh sách các tính năng chức năng đã hoàn thành (Phần 2)

Tính năng	Hoàn thành	Mô tả chi tiết
Theo dõi tiến độ học tập (Progress Tracking)		
Cập nhật thời gian học	✓	Tự động ghi nhận và cập nhật thời gian học tập của sinh viên cho từng bài học đề xuất.
Xem phân tích tiến độ	✓	Sinh viên có thể xem phân tích tiến độ học tập chi tiết trong khóa học hoặc bài học cụ thể.
Xem điểm số	✓	Giảng viên có thể xem điểm số của sinh viên trong khóa học, mục tiêu và lộ trình học tập của sinh viên.
Quản lý phản hồi (Feedback Management)		
Gửi phản hồi	✓	Sinh viên và giảng viên có thể gửi phản hồi về hệ thống với đánh giá và mô tả.
Xem và quản lý phản hồi	✓	Quản trị viên và giảng viên có thể xem danh sách phản hồi, lọc theo thời gian và trạng thái.
Cập nhật trạng thái phản hồi	✓	Quản trị viên có thể cập nhật trạng thái phản hồi (đang chờ, đang xử lý, đã giải quyết).
Quản lý người dùng (User Administration)		
Tạo người dùng mới	✓	Quản trị viên có thể tạo tài khoản cho sinh viên, giảng viên và quản trị viên khác.
Quản lý danh sách người dùng	✓	Hỗ trợ xem danh sách, tìm kiếm, lọc và quản lý thông tin chi tiết người dùng.
Quản lý trạng thái người dùng	✓	Cho phép bật/tắt tài khoản người dùng để kiểm soát quyền truy cập vào hệ thống.
Dashboard và báo cáo (Dashboard & Reporting)		
Dashboard sinh viên	✓	Hiển thị tổng quan về tiến độ học tập, khóa học gần đây và các hoạt động học tập.
Dashboard giảng viên	✓	Cung cấp tổng quan về hoạt động giảng dạy, số lượng khóa học, bài học và sinh viên.
Dashboard quản trị viên	✓	Hiển thị thông tin tổng quan về hệ thống, người dùng và hoạt động trên toàn hệ thống.

9.1.3 Các yêu cầu phi chức năng

Table 9.4: Danh sách các yêu cầu phi chức năng đã đáp ứng (Phần 1)

Loại yêu cầu	Hoàn thành	Tiêu chí	Chi tiết triển khai
Hiệu năng	Đạt một phần ¹	Thời gian phản hồi nhanh	Thời gian phản hồi cho các thao tác thông thường dưới 2 giây.
	Chưa đánh giá	Hỗ trợ nhiều người dùng	Hệ thống hỗ trợ tối thiểu 1000 người dùng đồng thời mà không giảm hiệu năng.
	X ²	Thời gian phản hồi AI	API calls tới các model LLM, giữ thời gian phản hồi AI dưới 8 giây cho các tác vụ phức tạp.
Khả năng mở rộng	✓	Kiến trúc microservices	Thiết kế hệ thống theo kiến trúc module, dễ dàng mở rộng và thêm tính năng mà không ảnh hưởng đến thành phần khác.
	✓	Triển khai trên cloud	Sử dụng các dịch vụ cloud có khả năng auto-scaling.
Bảo mật	✓	Mã hóa dữ liệu	Mã hóa dữ liệu cá nhân và thông tin học tập trong quá trình lưu trữ và truyền tải.
	✓	Xác thực đa yếu tố	Triển khai xác thực đa yếu tố cho các tài khoản sinh viên, quản trị và giảng viên.
	✓	Phân quyền chi tiết	Hệ thống phân quyền chi tiết theo vai trò Sinh viên, Giảng viên, Quản trị viên.

²Chi tiết về thời gian phản hồi trung bình của các thao tác thông thường trong Bảng ??.

²Chi tiết về thời gian phản hồi trung bình của các tính năng sử dụng LLM được trình bày trong Bảng 9.8.

Table 9.5: Danh sách các yêu cầu phi chức năng đã đáp ứng (Phần 2)

Loại yêu cầu	Hoàn thành	Tiêu chí	Chi tiết triển khai
Tính khả dụng	✓	Giao diện thân thiện	Thiết kế giao diện trực quan, dễ sử dụng với bố cục rõ ràng và màu sắc hợp lý.
	✓	Tính nhất quán	Đảm bảo tính nhất quán trong thiết kế và trải nghiệm người dùng trên toàn bộ hệ thống.
Tính tương thích	✓	Tương thích trình duyệt	Hỗ trợ các trình duyệt phổ biến (Chrome, Firefox, Safari, Edge) với trải nghiệm nhất quán.
	✓	Tương thích hệ điều hành	Hoạt động ổn định trên các hệ điều hành phổ biến (Windows, macOS, Linux) và thiết bị di động.
Khả năng bảo trì	✓	Mã nguồn có cấu trúc	Mã nguồn được tổ chức theo cấu trúc rõ ràng, dễ đọc và tuân thủ các tiêu chuẩn coding.
	✓	Tài liệu đầy đủ	Hệ thống có tài liệu kỹ thuật đầy đủ, bao gồm API documentation và hướng dẫn triển khai.
Độ tin cậy	✓	Tính ổn định cao	Hệ thống duy trì uptime ổn định đảm bảo trải nghiệm người dùng liên tục.
	✓	Sao lưu tự động	Dữ liệu được sao lưu tự động theo lịch trình, đảm bảo khả năng phục hồi nhanh chóng.

9.2 Đánh giá chất lượng đầu ra của lộ trình học để xuất bởi LLM

9.2.1 Phương án đánh giá

Trong bối cảnh đánh giá hệ thống sinh lộ trình học tập, có hai hướng tiếp cận để đánh giá [16]:

- **Online evaluation** là phương pháp sử dụng phản hồi thực tế từ người dùng hoặc chuyên gia để đánh giá chất lượng đầu ra. Phương pháp này có độ tin cậy cao và phản ánh đúng nhu cầu thực tế, tuy nhiên yêu cầu thời gian, chi phí lớn và khó mở rộng trong giai đoạn phát triển ban đầu.
- **Offline evaluation** là phương pháp sử dụng mô hình hoặc thuật toán đánh giá tự động để đánh giá đầu ra mà không cần tương tác với người dùng, về dữ liệu, những dữ liệu cũ quan sát được trong quá khứ có thể được sử dụng cho Offline evaluation.

Trong phần đánh giá này, nhóm lựa chọn phương pháp **offline evaluation kết hợp với GEval** –



một framework đánh giá tự động sử dụng mô hình ngôn ngữ lớn (LLM) làm giám khảo (LLM-as-a-Judge)[17]. GEval cung cấp một framework linh hoạt để định nghĩa các tiêu chí đánh giá tùy chỉnh, kết hợp với chuỗi suy luận (Chain-of-Thought) để hướng dẫn mô hình đưa ra lập luận có cơ sở. GEval có những điểm phù hợp để được chọn làm metric cho phần đánh giá này vì:

- Không cần dữ liệu tham chiếu cố định (reference output).
- Thực hiện đánh giá nhanh, lặp đi lặp lại và có thể tự động hóa được với nhiều tập dữ liệu khác nhau.
- Dễ mở rộng cho nhiều tiêu chí khác nhau như chất lượng giải thích, tính logic, mức độ phù hợp mục tiêu,...
- Có thể sử dụng với nhiều mô hình đánh giá khác nhau như: gpt-4o, gpt-4o-mini, llama-3.2,...

9.2.2 Xây dựng các metrics

Dựa trên metric GEval được đề cập ở trên, các metric sau được lựa chọn để đánh giá sơ bộ đầu ra của hệ thống:

- **Goal Alignment (Goal):** Lộ trình có bám sát mục tiêu học tập không?
- **Explanation Quality (Expl.):** Chất lượng và độ rõ ràng của giải thích lý do trong từng bài học được đề xuất.
- **Ordering Logic (Order):** Trình tự các bài học có hợp lý và phát triển kiến thức tuyến tính không?
- **Module Appropriateness (Mod. App.):** Các module có phù hợp với nội dung bài học và mục tiêu không?

Metric GEval được sử dụng làm nền tảng để tự động hóa quá trình đánh giá, cụ thể:

- Xây dựng tiêu chí đánh giá cho từng metrics, tương tự như việc xây dựng một rubrics để chấm điểm và các tiêu chí này phải càng càng cụ thể, giống như con người thực hiện việc đánh giá cũng cần rubrics.
- Model đánh giá (evaluator) được yêu cầu cho điểm (score) và giải thích (reasoning).

Chi tiết về các metrics được định nghĩa cho phần đánh giá, người đọc có thể xem chi tiết ở đoạn code sau: [Github³](#).

³Định nghĩa các metrics: <https://github.com/dpnam2112/codemate-backend/blob/main/tests/metrics.py>



9.2.3 Thiết kế quy trình đánh giá

Do hạn chế về tài nguyên cũng như một số ràng buộc từ các nhà cung cấp dịch vụ để sử dụng mô hình của OpenAI, Google,... hiện tại, nhóm chỉ có thể triển khai việc đánh giá với phương pháp này ở mức cơ bản và về mặt ý tưởng:

- Mô hình sinh lô trình: *Gemini 2.0 Flash*.
- Các mô hình đánh giá: *gpt-4o-mini*, *gpt-4.1-nano*, *llama-3.3-70B*. Ta cần sử dụng các mô hình không cùng họ với mô hình sinh lô trình do phương án LLM-as-a-judge có một nhược điểm: Nếu LLM được sử dụng như là người đánh giá cùng họ/có đặc tính tương đồng với mô hình được sử dụng để sinh ra output, thì việc thiên vị trong đánh giá có thể xảy ra do vấn đề *Preference leakage*[18].
- Dữ liệu đầu vào: Mục tiêu học tập cụ thể của học sinh cho từng khóa học. Người đọc có thể tham khảo thêm về dữ liệu được sử dụng để đánh giá ở đường link sau: [Github⁴](#).
- Output: File CSV lưu kết quả điểm số và giải thích cho từng metric.
- Khóa học được sử dụng để thực hiện đánh giá:
 - Cấu trúc dữ liệu và giải thuật của Đại học Bách khoa thành phố Hồ Chí Minh.
 - Nhập môn Khoa học máy tính và lập trình bằng Python của MIT (MIT 6.0001)

9.2.4 Kết quả đánh giá và nhận xét

Sau khi thực thi code để đánh giá, ta thu được các kết quả sau:

Kết quả đánh giá (file csv): [Github⁵](#)

Table 9.6: Điểm trung bình theo từng tiêu chí đánh giá: Goal, Explanation Quality, Module Appropriateness, và Ordering Logic

Model	Expl.	Goal	Mod. App.	Order
gpt-4.1-nano	0.883	0.867	0.900	0.817
gpt-4o-mini	0.917	0.883	0.950	0.767
LLaMA-3.1-70B	0.945	0.964	0.945	0.891

⁴Dữ liệu bài tập code và lời giải: https://github.com/dpnam2112/codemate-backend/blob/main/evaluation/data/learning_path/learning_paths_20250429-1307.json

⁵Dữ liệu kết quả đánh giá learning path: https://github.com/dpnam2112/codemate-backend/blob/main/evaluation/data/lp_csv/lp_csv_gpt_and_llama.csv

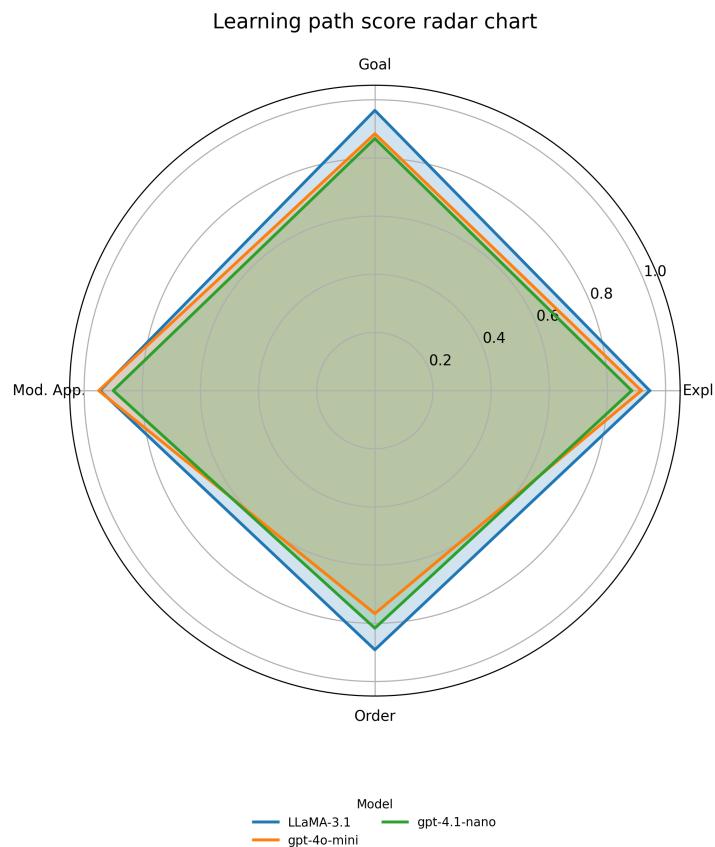


Figure 9.1: Radar chart tổng hợp kết quả đánh giá lộ trình học đê xuất khi chạy đánh giá LLM-as-a-judge ứng với từng model

Dựa vào kết quả có được, sau đây là một số nhận xét sơ bộ:

- Tiêu chí **Explanation Quality** đạt điểm trung bình cao nhất (0.917 theo gpt-4o-mini hay 0.883 theo gpt-4.1-nano), cho thấy phần lớn các bài học trong lộ trình đều có phần giải thích rõ ràng, dễ hiểu và phù hợp với nội dung.
- Về tiêu chí **Module Appropriateness**, hệ thống cũng ghi nhận mức đánh giá tích cực (lên đến 0.95), cho thấy các module học tập được lựa chọn và tạo sinh phù hợp với từng bài học và mục tiêu tổng thể, không bị lệch pha hay dư thừa nội dung.
- Tiêu chí **Goal Alignment** có điểm số ổn định (khoảng 0.88), phản ánh rằng lộ trình học tập nhìn chung bám sát mục tiêu đầu vào mà người học đã đặt ra.
- Tuy nhiên, ở tiêu chí **Ordering Logic**, điểm số trung bình thấp hơn đáng kể (0.767 theo gpt-4o-mini, 0.817 theo gpt-4.1-nano). Điều này cho thấy vẫn còn tồn tại một số vấn đề về thứ tự sắp xếp các bài học – ví dụ như thiếu tính tuyến tính, sắp xếp không tối ưu về mặt xây dựng nền tảng kiến thức.



9.3 Đánh giá testcases được sinh ra bởi tính năng tạo sinh bài tập code

9.3.1 Phương pháp đánh giá

Trong bối cảnh đánh giá chất lượng test case được sinh tự động bởi LLM nói riêng cũng như chất lượng của một bộ testcases nói chung, có nhiều metrics khác nhau có thể được xem xét, bao gồm:

- Tính đa dạng (Diversity) đo mức độ khác biệt giữa các test case trong một bộ dữ liệu.
- Độ phủ (Coverage) đo xem test case có bao phủ đủ các nhánh logic, điều kiện biên, hoặc tập input không.
- Mutation Score là tỉ lệ mutant (phiên bản lời giải bị chỉnh sửa nhỏ để cố tình tạo lỗi) bị phát hiện bởi test case. Chỉ số này phản ánh khả năng của test case trong việc phân biệt giữa lời giải đúng và lời giải sai — tức là đo mức độ “sắc bén” của test case trong kiểm thử.

Tuy nhiên, trong phạm vi đánh giá sơ bộ này, nhóm lựa chọn một metric đơn giản nhưng có giá trị thực tiễn là: **tỉ lệ test case chạy đúng (pass)** khi thực thi với một lời giải đã được kiểm chứng là chính xác với các lý do như sau:

- Đây là bước kiểm tra cơ bản và thiết yếu, nếu test case không chạy được hoặc gây lỗi, các đánh giá khác sẽ không còn ý nghĩa.
- Để triển khai và so sánh giữa các mô hình sinh test case khác nhau.
- Phản ánh gián tiếp khả năng hiểu đề và sinh đầu vào hợp lệ của mô hình.

Phép đánh giá được thực hiện bằng cách chạy từng test case trên một lời giải mẫu được xác nhận là đúng, sử dụng môi trường thực thi tự động (Judge0). Các kết quả như “Pass”, “Fail”, “Runtime Error” được ghi nhận và tổng hợp theo từng mô hình LLM để phân tích định lượng.

9.3.2 Cài đặt

Sau đây là cấu hình cho phần đánh giá:

- Các mô hình được sử dụng: *gpt-4o-mini*, *gpt-4.1-nano*, *gemini-2.0-flash*.



- Bài tập lập trình: 12 bài tập lập trình được lựa chọn ngẫu nhiên trên các trang: leetcode.com, hackerrank.com,... và lời giải được kiểm chứng. Về dữ liệu, người đọc có thể xem chi tiết ở đường link: [Github](#)⁶
- Sử dụng mỗi mô hình để tạo sinh testcases cho mỗi bài tập, trung bình với mỗi bài sẽ có từ 8 tới 14 testcases (Tùy thuộc vào mô hình cũng như giới hạn tính toán được áp đặt bởi nhà cung cấp dịch vụ để gọi mô hình: OpenAI, TogetherAI, Google VertexAI).

9.3.3 Kết quả đánh giá và nhận xét

Table 9.7: Tổng quan kết quả thực thi test case sinh bởi LLM

LLM Model	Total	Pass	Fail	Runtime Error
gpt-4o-mini	147	79.6%	19.0%	1.4%
gpt-4.1-nano	125	84.8%	13.6%	1.6%
gemini-2.0-flash-lite	161	77.0%	16.8%	6.2%
llama-3.3-70B-Instruct	74	51.4%	31.1%	17.6%

Chú thích cột: **Total** là tổng số test case được sinh ra và đánh giá. **Pass** là tỉ lệ test case cho ra kết quả đúng khi chạy với lời giải chính xác. **Fail** là test case chạy được nhưng cho kết quả sai.

Runtime Error là test case gây lỗi khi thực thi (ví dụ chia cho 0, lỗi kiểu dữ liệu,...).

Nhận xét:

Tổng quan kết quả đánh giá cho thấy các mô hình LLM có khả năng sinh test case ở mức khá, với tỷ lệ Pass trung bình từ 75% trở lên đối với các mô hình của OpenAI và Gemini. Trong đó, mô hình **gpt-4.1-nano** đạt kết quả cao nhất với 84.8% test case được đánh giá là đúng (Pass), cho thấy khả năng hiểu bài toán và sinh dữ liệu kiểm thử hợp lý. Mô hình **gpt-4o-mini** theo sau với tỷ lệ 79.6%, tuy nhiên tỷ lệ test case sai (Fail) cao hơn.

Đáng chú ý, mô hình **gemini-2.0-flash-lite** có số lượng Runtime Error cao hơn (6.2%), phản ánh khả năng sinh input chưa ổn định hoặc dễ gây lỗi khi thực thi. Mô hình **llama-3.3-70B-Instruct** có kết quả thấp nhất với chỉ 51.4% Pass và tỷ lệ Runtime Error lên đến 17.6%, cho thấy khó khăn trong việc sinh ra các test case chính xác và phù hợp định dạng.

Các nguyên nhân chính dẫn đến lỗi test case gồm:

- Hạn chế về khả năng xử lý số liệu:** LLM không được thiết kế để thực hiện tính toán, dẫn đến việc sinh output sai hoặc không nhất quán.
- Sai định dạng input/output:** Một số test case bị lỗi do định dạng không đúng với yêu cầu, ví dụ như thiếu dòng newline, thiếu ký tự phân cách hoặc kiểu dữ liệu không khớp.

⁶Link dẫn tới dataset bài tập code và kết quả: https://github.com/dpnam2112/codemate-backend/blob/main/evaluation/data/leetcode_problems/leetcode.json



9.3.4 Hướng phát triển và cải thiện cho phần tạo sinh testcases

Kết quả đánh giá cho thấy các mô hình LLM hiện tại tuy có khả năng sinh test case hợp lệ ở mức khá, nhưng vẫn còn tồn tại một số nhược điểm như sai định dạng, thiếu bao phủ edge cases, và sinh ra dữ liệu không ổn định gây lỗi khi thực thi. Để khắc phục và cải thiện chất lượng đầu ra, một số hướng phát triển có thể được xem xét như sau:

- Fine-tune mô hình: Việc huấn luyện lại mô hình (fine-tune) trên tập dữ liệu chứa các bài toán lập trình và test case đúng chuẩn có thể giúp mô hình học được cấu trúc và định dạng đầu ra mong muốn, từ đó giảm thiểu các lỗi định dạng hoặc logic phổ biến.
- Tích hợp function calling: Việc sử dụng cơ chế function calling (ví dụ như OpenAI Tools API hoặc ReAct-style prompting) cho phép mô hình gọi các hàm kiểm tra định dạng, phân tích cú pháp hoặc validate input/output trong quá trình sinh test case. Điều này giúp phát hiện và loại bỏ các test case không hợp lệ ngay từ bước sinh.

9.4 Đánh giá độ trễ trung bình (Latency) của các tính năng sử dụng LLM

9.4.1 Phạm vi đánh giá

Mục tiêu của phần này là đánh giá **thời gian phản hồi trung bình** (average latency) của một số tính năng cốt lõi trong hệ thống có sử dụng mô hình ngôn ngữ lớn (LLM). Các tính năng được đưa vào đánh giá bao gồm:

- **Sinh lộ trình học tập** (Learning Path Generation – L.P. Gen.): đo thời gian cần thiết để sinh ra một lộ trình học tập hoàn chỉnh dựa trên mục tiêu đầu vào của người học.
- **Sinh câu hỏi trắc nghiệm** (Quiz Generation – Quiz Gen.): đo thời gian tạo một bộ câu hỏi khi người dùng yêu cầu.
- **Trợ lý lập trình** (Coding Assistant): đo thời gian phản hồi từ mô hình khi người dùng đặt một câu hỏi trong phần trợ lý lập trình.
- **Sinh bài tập lập trình** (Coding Exercise Generation – Coding Ex. Gen.): đo thời gian sinh một bài tập lập trình mới dựa trên mô tả bài học.

9.4.2 Phương pháp đánh giá

Quy trình đo lường độ trễ được tiến hành theo các bước sau:

- Làm nóng hệ thống (warm-up): Trước khi đo, hệ thống được khởi động trước một số request để tránh ảnh hưởng của các yếu tố như cache chưa khởi tạo, kết nối cơ sở dữ liệu chưa ổn định, hoặc mô hình LLM chưa được load vào bộ nhớ.
- Sau đó, một script tự động được viết để mô phỏng hành vi người dùng thông qua việc gửi request đến từng endpoint tương ứng với các tính năng nền trên. Ở phần này, ta có thể sử dụng Grafana K6 (một công cụ cho load testing) hoặc một Python script đơn giản để mô phỏng quá trình gọi API.
- Thu thập và tính toán latency: Với mỗi tính năng, request được gửi liên tục trong vòng 5 phút với tốc độ cố định (có giới hạn RPM (Request per minute) để tránh bị throttling từ phía nhà cung cấp LLM như OpenAI hay Google VertexAI). Kết quả được ghi lại, sau đó tính trung bình thời gian phản hồi (average latency).

9.4.3 Kết quả đo lường

Kết quả độ trễ trung bình (đơn vị: giây) cho từng tính năng như sau:

Table 9.8: Thời gian phản hồi trung bình của một số tính năng sử dụng LLM

Tính năng	Latency trung bình (s)
Sinh lô trình học tập (L.P. Gen.)	110.21
Sinh câu hỏi trắc nghiệm (Quiz Gen.)	34.97
Sinh bài tập lập trình (Coding Ex. Gen.)	13.43
Trợ lý lập trình (Coding Assistant)	3.08

9.4.4 Đánh giá kết quả

Đa số các tính năng liên quan đến LLM đều có thời gian phản hồi khá cao, đặc biệt là các tác vụ sinh lô trình học tập hoặc tạo bộ câu hỏi. Nguyên nhân chủ yếu đến từ bản chất của mô hình ngôn ngữ lớn và cách tích hợp hệ thống, bao gồm:

- Giới hạn tốc độ của dịch vụ cung cấp mô hình. Một số endpoint sử dụng các API từ bên thứ ba như OpenAI hoặc Google VertexAI, vốn áp dụng cơ chế rate limiting, làm tăng độ trễ trong một số thời điểm.
- Trong một số trường hợp, hệ thống backend thực hiện xử lý đồng bộ toàn bộ pipeline từ lúc nhận yêu cầu đến khi sinh xong toàn bộ phản hồi, không sử dụng các kỹ thuật như streaming hay async partial responses, làm tăng thời gian chờ của người dùng.
- Trong phạm vi đánh giá này, các mô hình ngôn ngữ được sử dụng là mô hình gốc (foundation model, chưa fine-tune). Việc sử dụng mô hình chưa được tinh chỉnh có thể làm tăng



thời gian phản hồi do mô hình cần "suy luận tổng quát" mà không được tối ưu hoá cho từng tác vụ cụ thể. Điều này không chỉ ảnh hưởng đến chất lượng đầu ra mà còn làm tăng chi phí tính toán trong mỗi lần gọi.

9.4.5 Phương án cải thiện

Dựa trên các yếu tố ảnh hưởng đến độ trễ nêu trên, một số hướng cải thiện thời gian phản hồi có thể được cân nhắc như sau:

- Xem xét tách pipeline xử lý thành các giai đoạn bất đồng bộ (asynchronous stages) và sau đó xử lý các bước này đồng thời, việc này cần refactor lại code một cách kỹ lưỡng và cẩn thận, do việc xử lý nhiều tác vụ đồng thời sẽ dễ gây nên các tình trạng như *race condition* - Một vấn đề phổ biến trong concurrent programming/parallel computing.
- Trong dài hạn, có thể *huấn luyện* hoặc *tinh chỉnh* (fine-tune) mô hình trên các tác vụ cụ thể để rút ngắn thời gian suy luận do mô hình đã học được cấu trúc đầu ra mong muốn. Việc fine-tune mô hình sẽ giúp cải thiện chất lượng output cũng như giảm chi phí tính toán.
- Sử dụng distilled model (Mô hình đã được chưng cất, nhẹ hơn foundation model) với các tác vụ không yêu cầu chất lượng đầu ra cao tuyệt đối, như Mistral 7B, Claude Instant, v.v.
- Tối ưu prompt để tận dụng kỹ thuật prompt caching, từ đó tối ưu tốc độ suy luận của mô hình. Hầu hết các dịch vụ cung cấp LLM API cũng khuyến khích người sử dụng tối ưu câu prompt sao cho kỹ thuật prompt caching được tận dụng hiệu quả, từ đó tối ưu tốc độ và chi phí tính toán[19].

9.5 Đánh giá latency của một số API không liên quan đến LLM

9.5.1 Phạm vi và mục tiêu

Phần này nhằm đánh giá khả năng phản hồi của một số API nội bộ không sử dụng mô hình ngôn ngữ lớn (LLM). Mục tiêu là kiểm tra lại functional requirement về latency của các tính năng không liên quan tới AI/LLM. Các API được kiểm tra gồm:

- GET /activities
- GET /course-lessosn

- GET /module-quizzes
- GET /recent-courses
- GET /users
- POST /auth/login

9.5.2 Thiết lập thử nghiệm

- Công cụ sử dụng: Grafana K6.
- Kịch bản: gửi request liên tục trong vòng 5 phút với các mức tải khác nhau:
 - 3 Virtual Users (VU) — tải nhẹ.
 - 10 VUs — tải trung bình.
 - 50 VUs — tải cao.
- Các API được thiết lập ngưỡng p(95) < 500ms (p(95) nghĩa là phân vị thứ 95)

9.5.3 Kết quả đo lường

Table 9.9: Thông kê độ trễ các API ở mức tải 3 VU (min / med / p95 / max)

API	Min	Med	p(95)	Max
GET /activities	754.85μs	1.98ms	5.67ms	229.92ms
GET /course-lessions	18.69ms	42.34ms	450.00ms	518.23ms
GET /module-quizzes	12.75ms	29.43ms	441.98ms	505.93ms
GET /recent-courses	6.84ms	16.27ms	233.63ms	461.55ms
GET /users	444.96μs	1.27ms	7.04ms	431.62ms
POST /auth/login	218.13ms	234.91ms	488.26ms	685.3ms

Table 9.10: Thông kê độ trễ các API ở mức tải 10 VU (min / med / p95 / max)

API	Min	Med	p(95)	Max
GET /activities	1.57ms	6.4ms	218.16ms	435.08ms
GET /course-lessions	78.43ms	548.47ms	1.14s	1.92s
GET /module-quizzes	42.98ms	354.83ms	971.84ms	1.77s
GET /recent-courses	42.51ms	285.15ms	897.28ms	1.39s
GET /users	998.24μs	4.81ms	217.15ms	432.32ms
POST /auth/login	255.14ms	504.59ms	1.02s	2.33s



Table 9.11: Thông kê độ trễ các API ở mức tải 50 VU (min / med / p95 / max)

API	Min	Med	p(95)	Max
GET /activities	4.61ms	19.67ms	234.05ms	1.26s
GET /course-lessions	166.6ms	2.39s	5.68s	13.97s
GET /module-quizzes	71.37ms	1.91s	5.35s	10.27s
GET /recent-courses	151.5ms	1.78s	5.40s	11.37s
GET /users	5.23ms	18.22ms	428.54ms	1.05s
POST /auth/login	373.25ms	2.03s	6.50s	11.8s

Chú thích: **Min**, **Med**, **p(95)**, và **Max** lần lượt là độ trễ tối thiểu, trung vị, phân vị thứ 95 và tối đa trong suốt quá trình đo.

9.5.4 Đánh giá kết quả

Ở mức tải là 3VU cùng tương tác với hệ thống, tất cả các API đều đáp ứng tốt với thời gian phản hồi thấp và ổn định. Tuy nhiên, khi tăng lên 10 hoặc 50 VU, một số API bắt đầu vượt ngưỡng p(95), đặc biệt là các API truy xuất dữ liệu học liệu như /course-lessions, /module-quizzes, /recent-courses.

Tình trạng nghiêm trọng nhất ghi nhận ở mức 50 VU, khi độ trễ trung bình có thể lên tới hơn 5 giây. Endpoint /auth/login cũng bị ảnh hưởng với p(95) lên đến 6.5s, vượt xa ngưỡng chấp nhận là 2 giây.

9.6 Đánh giá giao diện người dùng (UI) bằng Lighthouse

Trong phần đánh giá này, một số trang chính của hệ thống đã được đánh giá bằng Lighthouse - một công cụ được phát triển bởi Google. Lighthouse là công cụ mã nguồn mở do Google phát triển, dùng để đánh giá chất lượng của các trang web theo các tiêu chí như hiệu năng, khả năng truy cập, chuẩn SEO, và đặc biệt là trải nghiệm người dùng (UI/UX). Việc sử dụng Lighthouse giúp phát hiện sớm các vấn đề ảnh hưởng đến tốc độ tải trang, khả năng tương tác và khả năng tiếp cận của người dùng. Kết quả đánh giá được ghi lại dưới dạng file PDF/HTML. Các tiêu chí chính được đánh giá bao gồm:

- Performance (Hiệu năng):** Đo thời gian tải trang, khả năng phản hồi và tốc độ tương tác.
- Accessibility (Khả năng tiếp cận):** Kiểm tra mức độ thân thiện với người dùng có nhu cầu đặc biệt (đọc màn hình, màu sắc).
- Best Practices:** Kiểm tra việc tuân thủ các thực hành tốt về bảo mật và hiệu suất.
- SEO:** Đánh giá khả năng hiển thị của trang trên công cụ tìm kiếm.



- **Progressive Web App (PWA):** Kiểm tra khả năng hoạt động như ứng dụng di động (nếu áp dụng).

Các chỉ số như *First Contentful Paint*, *Time to Interactive*, và *Cumulative Layout Shift* được sử dụng để đánh giá trải nghiệm người dùng. Những trang đạt điểm cao (>90) cho thấy hệ thống đã tối ưu tốt UI/UX, trong khi các trang có điểm thấp cần được xem xét cải thiện.

Chi tiết về kết quả đánh giá, người đọc có thể xem ở đường link sau: [Github](#)⁷

⁷Link: <https://github.com/dpnam2112/codemate-lighthouse-eval>

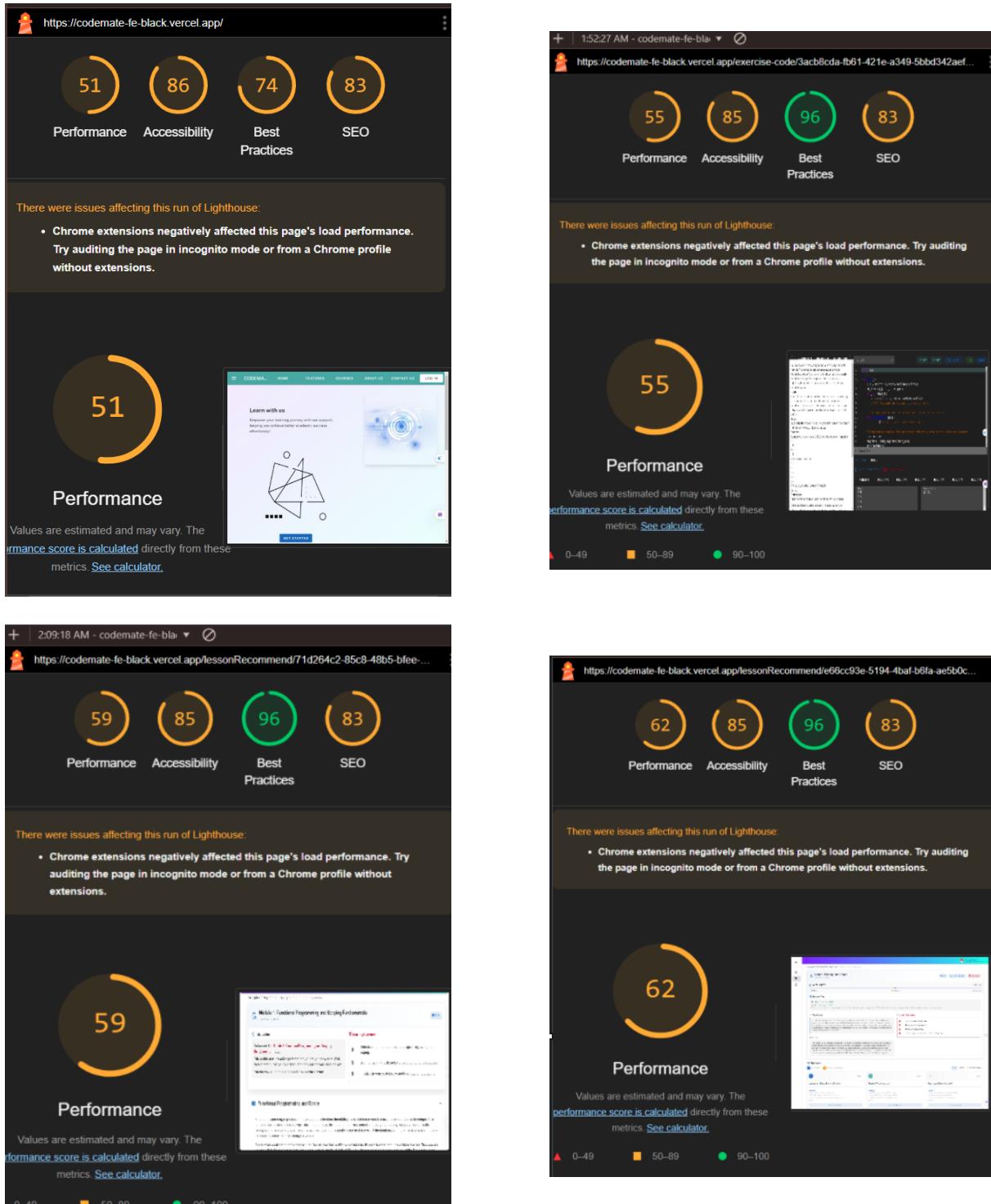


Figure 9.2: Kết quả đánh giá một số trang bằng Lighthouse

Chương 10

Tổng kết

10.1 Kết quả đạt được

Hệ thống học tập trực tuyến thông minh đã được thiết kế và phát triển thành công, đáp ứng các mục tiêu đề ra trong việc cung cấp một nền tảng học tập cá nhân hóa cho sinh viên ngành Khoa học máy tính và những người quan tâm đến lập trình. Các kết quả nổi bật bao gồm:

- Quản lý người dùng:** Hệ thống hỗ trợ đăng ký và đăng nhập cho ba vai trò chính: sinh viên, giảng viên và quản trị viên. Chức năng quản lý tài khoản cho phép tạo, cập nhật và quản lý thông tin người dùng một cách hiệu quả, với tích hợp đăng nhập qua email trường đại học và tài khoản Google.
- Đề xuất tài nguyên học tập:** Hệ thống tự động đề xuất các khóa học, bài học và bài tập phù hợp dựa trên mục tiêu và tiến độ học tập của người dùng. Người dùng có thể linh hoạt lựa chọn phương pháp học như bài quiz, bài tập lập trình hoặc tài liệu đọc.
- Tạo bài tập tự động:** Hệ thống sinh ra bài tập ở các mức độ khác nhau, từ cơ bản đến nâng cao, và tự động điều chỉnh độ khó dựa trên khả năng của người học.
- Gia sư AI:** Tính năng nổi bật của hệ thống là gia sư AI, hỗ trợ giải thích mã nguồn, gợi ý tối ưu hóa mã, sửa lỗi và cung cấp phản hồi thời gian thực.
- Theo dõi tiến độ:** Hệ thống cung cấp các công cụ theo dõi và báo cáo chi tiết về tiến độ học tập, bao gồm thời gian học, số lượng bài tập hoàn thành và điểm số. Phân tích dữ liệu học tập giúp điều chỉnh lộ trình học phù hợp với từng người dùng, áp dụng thành công các phương pháp đánh giá SMART, Rubric-Based Assessment và STAR.
- Tạo thêm nội dung học tập gấp khó khăn:** Hệ thống có khả năng phân tích khó khăn học tập của sinh viên và tạo tiếp nội dung bài học phù hợp, giúp giải quyết các vấn đề cụ



thể và cải thiện hiệu quả học tập.

- **Quản lý khóa học:** Giảng viên có thể dễ dàng tạo, chỉnh sửa và quản lý khóa học, tải lên tài liệu và liên kết tài nguyên bổ sung.
- **Quản trị hệ thống:** Quản trị viên có khả năng quản lý danh sách người dùng, khóa/mở tài khoản, và xử lý phản hồi từ người dùng, đảm bảo vận hành hệ thống ổn định và đáp ứng nhu cầu người dùng.

Hệ thống đã được thử nghiệm và nhận được phản hồi tích cực từ người dùng, chứng minh khả năng tối ưu hóa trải nghiệm học tập, tăng cường sự tương tác và nâng cao hiệu quả học lập trình.

10.2 Hướng phát triển

Để nâng cao chất lượng và mở rộng khả năng của hệ thống học tập trực tuyến thông minh, các hướng phát triển trong tương lai bao gồm:

- **Tối ưu hóa hiệu suất hệ thống:** Vấn đề hiệu suất đang là ưu tiên hàng đầu của hệ thống, nhóm đang ưu tiên cải thiện tốc độ xử lý của LLM trong các phản hồi của gia sư AI, tạo lộ trình học tập và tạo bài tập tự động.
- **Tăng cường cá nhân hóa:** Phát triển các thuật toán học máy tiên tiến hơn để phân tích sâu hơn hành vi và sở thích học tập của người dùng, từ đó đề xuất các lộ trình học tập phù hợp, đồng thời nâng cấp hệ thống để các lộ trình học tập cần được thay đổi linh hoạt hơn dựa trên năng lực của người học.
- **Mở rộng hỗ trợ đa ngôn ngữ lập trình:** Hiện tại, gia sư AI chủ yếu hỗ trợ một số ngôn ngữ lập trình phổ biến. Trong tương lai, hệ thống có thể mở rộng để hỗ trợ thêm các ngôn ngữ như Rust, Go, hoặc các ngôn ngữ chuyên biệt cho khoa học dữ liệu và trí tuệ nhân tạo.
- **Phát triển cộng đồng học tập:** Tích hợp các tính năng mạng xã hội như diễn đàn, nhóm học tập, hoặc chế độ học theo nhóm, giúp người học chia sẻ kiến thức và hỗ trợ lẫn nhau.
- **Mở rộng đối tượng người dùng:** Hệ thống có thể được điều chỉnh để hỗ trợ các lĩnh vực khác ngoài lập trình, như toán học, vật lý, hoặc các môn học kỹ thuật, mở rộng phạm vi ứng dụng trong giáo dục.

Những định hướng này sẽ giúp hệ thống không chỉ duy trì tính cạnh tranh mà còn trở thành một công cụ giáo dục tiên phong, đáp ứng nhu cầu học tập đa dạng và ngày càng phức tạp trong thời đại công nghệ 4.0.

Tài liệu tham khảo

- [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., Ravikkumar, 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] Zhang, Z., Brun, A., & Boyer, A. (2020). New measures for offline evaluation of learning path recommenders. *15th European Conference on Technology Enhanced Learning (ECTEL 2020)*. https://doi.org/10.1007/978-3-030-57717-9_19
- [17] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). G-eval: Nlg evaluation using gpt-4 with better human alignment. <https://arxiv.org/abs/2303.16634>
- [18] Li, D., Sun, R., Huang, Y., Zhong, M., Jiang, B., Han, J., Zhang, X., Wang, W., & Liu, H. (2025). Preference leakage: A contamination problem in llm-as-a-judge. <https://arxiv.org/abs/2502.01534>
- [19] OpenAI. (2024). *Prompt caching guide* [Accessed May 4, 2025]. Retrieved May 4, 2025, from <https://platform.openai.com/docs/guides/prompt-caching>
- [20] 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>
- [21] 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/>
- [22] 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>
- [23] DB-Engines. (2024). Arangodb vs neo4j vs orientdb [Accessed: 2024-12-12]. <https://db-engines.com/en/system/ArangoDB%3bNeo4j%3bOrientDB>



- [24] 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/>
- [25] 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>
- [26] DB-Engines. (n.d.). *Arangodb vs neo4j vs orientdb.* Retrieved December 12, 2024, from <https://db-engines.com/en/system/ArangoDB%3bNeo4j%3bOrientDB>