

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2025-2026

**XÂY DỰNG HỆ THỐNG BACKEND CHO
NỀN TẢNG HỌC TẬP TRỰC TUYẾN
CÓ TÍCH HỢP YẾU TỐ TRÒ CHƠI**

Giảng viên hướng dẫn:

ThS. Nguyễn Ngọc Đan Thanh

Sinh viên thực hiện:

Họ tên: Nguyễn Văn Tổng

MSSV: 110122188

Lớp: DA22TTC

Vĩnh Long, tháng 12 năm 2025

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2025-2026

**XÂY DỰNG HỆ THỐNG BACKEND CHO
NỀN TẢNG HỌC TẬP TRỰC TUYẾN
CÓ TÍCH HỢP YẾU TỐ TRÒ CHƠI**

Giảng viên hướng dẫn:

ThS. Nguyễn Ngọc Đan Thanh

Sinh viên thực hiện:

Họ tên: Nguyễn Văn Tổng

MSSV: 110122188

Lớp: DA22TTC

Vĩnh Long, tháng 12 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Vĩnh Long, ngày tháng 12 năm 2025

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

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

Vĩnh Long, ngày tháng 12 năm 2025

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

LỜI CẢM ƠN

Em muốn cảm ơn sâu sắc cho sự giúp đỡ và hướng dẫn mà em đã nhận được từ giảng viên hướng dẫn trong quá trình hoàn thành đề án thực tập chuyên ngành lĩnh vực Công nghệ Thông tin.

Thời gian qua, em đã học được nhiều kinh nghiệm quý giá và kiến thức từ giảng viên hướng dẫn và em rất biết ơn sự hỗ trợ và động viên mà giảng viên hướng dẫn đã cung cấp cho em trong suốt quá trình nghiên cứu.

Thông qua báo cáo này, em xin gửi lời cảm ơn đến cô Nguyễn Ngọc Đan Thanh- giảng viên Khoa Công nghệ Thông tin-đã trang bị cho em những kiến thức quý báu thông qua đề án chuyên ngành với đề tài “Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi”, tạo cơ sở để em thực hiện hoàn thành đề án.

Trong quá trình nghiên cứu đề tài, do kiến thức còn hạn chế nên em vẫn còn nhiều thiếu sót trong quá trình tìm hiểu, đánh giá và trình bày đề tài. Em rất mong nhận được những đóng góp ý kiến của thầy/cô để em khắc phục và phát triển để có thể từng bước hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	i
NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG	ii
LỜI CẢM ƠN.....	iii
MỤC LỤC	iv
DANH MỤC HÌNH ẢNH.....	vii
DANH MỤC BẢNG BIỂU.....	viii
DANH MỤC TỪ VIẾT TẮT	1
TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH.....	2
MỞ ĐẦU	5
CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU	8
1.1 Giới thiệu đề tài	8
1.1.1 Đặt vấn đề	9
1.1.2 Phạm vi và giới hạn.....	9
1.2 Mục đích và yêu cầu	10
1.2.1 Mục đích.....	10
1.2.2 Yêu cầu.....	10
1.3 Ý nghĩa của đề tài	10
1.3.1 Đối với sinh viên.....	10
1.3.2 Đối với giảng viên.....	11
1.4 Đối tượng sử dụng và các kịch bản chính.....	12
1.5 Các quy trình nghiệp vụ trọng yếu.....	12
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT	14
2.1 Tổng quan về Hệ thống Quản lý học tập	14
2.2 Lý thuyết về Trò chơi hóa trong giáo dục	15
2.3 Các công nghệ và nền tảng phát triển.....	17

2.4 Kết chương.....	19
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	20
3.1 Đặc tả yêu cầu.....	20
3.2 Thiết kế kiến trúc hệ thống	24
3.3 Thiết kế cơ sở dữ liệu.....	26
3.3.1 Mô hình ERD	26
3.3.2 Đặc tả các bảng dữ liệu	32
3.4 Thiết kế Backend API.....	50
3.4.1 Kiến trúc API tổng thể	50
3.4.2 Cấu trúc Response chuẩn	51
3.4.3 Chức năng thời gian thực với Socket.IO và Redis.....	51
3.4.4 Kiến trúc phân lớp.....	53
3.4.5 Trò chơi hóa toàn diện	53
3.5 Thiết kế các luồng xử lý chính.....	54
3.5.1 Cơ chế bảo mật và sơ đồ xử lý xác thực	54
3.5.2 Luồng kiểm tra thời gian thực.....	55
3.5.3 Luồng cập nhật Bảng xếp hạng và Trao thưởng	59
3.5.4 Luồng nhập Câu hỏi hàng loạt file.....	60
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	65
4.1 Giới thiệu chung	65
4.2 Kết quả về kiến trúc và cơ sở dữ liệu	65
4.3 Kết quả về Hệ thống API Backend.....	66
4.4 Đánh giá hiệu năng hệ thống	71
4.4.1 Kiểm thử khả năng chịu tải với tác vụ thời gian thực.....	72
4.4.2 Kiểm thử độ ổn định với tác vụ xử lý dữ liệu nặng	73
CHƯƠNG 5. KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN.....	75

5.1 Giới thiệu chung	75
5.2 Kết luận	75
5.3 Hạn chế của đề tài	76
5.4 Hướng phát triển trong tương lai	77
DANH MỤC TÀI LIỆU THAM KHẢO	79

DANH MỤC HÌNH ẢNH

Hình 1 Sơ đồ một kiến trúc monolithic theo hướng mô-đun hóa.....	3
Hình 3.1 Sơ đồ usecase tổng quát.....	20
Hình 3.2 Sơ đồ usecase phân hệ quản lý các tài nguyên đào tạo	21
Hình 3.3 Sơ đồ phân hệ quản lý tiến trình học tập	22
Hình 3.4 Sơ đồ usecae phân hệ quản lý kết quả học tập	23
Hình 3.5 Sơ đồ kiến trúc tổng thể hệ thống.....	24
Hình 3.6 Sơ đồ kiến trúc phân lớp Backend.....	25
Hình 3.7 Mô hình ERD tổng quát	27
Hình 3.8 Sơ đồ xử lý xác thực	54
Hình 3.9 Sơ đồ tham gia bài kiểm tra.....	56
Hình 3.10 Sơ đồ làm bài và tương tác	57
Hình 3.11 Sơ đồ kết thúc và tổng kết kết quả	58
Hình 3.12 Sơ đồ luồng dữ liệu cập nhật Bảng xếp hạng	59
Hình 3.13 Sơ đồ tiếp nhận và tiền xử lý yêu cầu.....	61
Hình 3.14 Sơ đồ phân tích, xác thực và ánh xạ dữ liệu.....	62
Hình 3.15 Sơ đồ xử lý hình ảnh và ghi vào cơ sở dữ liệu	63
Hình 4.1 Kết quả kiểm thử khả năng chịu tải thời gian thực	72
Hình 4.2 Kết quả kiểm thử ổn định với tác vụ xử lý dữ liệu nặng	73

DANH MỤC BẢNG BIỂU

Bảng 2.1 So sánh các loại hình hệ thống LMS	15
Bảng 2.2 So sánh các Framework Backend cho Node.js	18
Bảng 3.1 Danh sách các thực thể dùng chung	26
Bảng 3.2 Danh sách thực thể trong phân hệ Quản lý các tài nguyên đào tạo	28
Bảng 3.3 Danh sách thực thể chính trong phân hệ Quản lý tiến trình học tập	30
Bảng 3.4 Danh sách thực thể chính trong phân hệ Quản lý kết quả học tập	31
Bảng 3.5 Đặc tả bảng Roles	32
Bảng 3.6 Đặc tả bảng Users	32
Bảng 3.7 Đặc tả bảng Programs	32
Bảng 3.8 Đặc tả bảng Subjects	32
Bảng 3.9 Đặc tả bảng TypeSubjects	33
Bảng 3.10 Đặc tả bảng TypeOfKnowledges	33
Bảng 3.11 Đặc tả bảng TienQuyet	33
Bảng 3.12 Đặc tả bảng Pos	33
Bảng 3.13 Đặc tả bảng PLOs	34
Bảng 3.14 Đặc tả bảng POsPLOs	34
Bảng 3.15 Đặc tả bảng Los	34
Bảng 3.16 Đặc tả bảng Semesters	34
Bảng 3.17 Đặc tả bảng Courses	35
Bảng 3.18 Đặc tả bảng TeacherSubjectAssignments	35
Bảng 3.19 Đặc tả bảng StudentCourses	36
Bảng 3.20 Đặc tả bảng CourseGradeColumns	36
Bảng 3.21 Đặc tả bảng CourseGradeColumnQuizzes	36
Bảng 3.22 Đặc tả bảng CourseGradeResults	37
Bảng 3.23 Đặc tả bảng CourseResults	37
Bảng 3.24 Đặc tả bảng CourseGradeResultHistories	37
Bảng 3.25 Đặc tả bảng Chapters	38
Bảng 3.26 Đặc tả bảng ChapterSections	38
Bảng 3.27 Đặc tả bảng chapter_lo	38
Bảng 3.28 Đặc tả bảng LearningAnalytics	38
Bảng 3.29 Đặc tả bảng UserLearningPaths	40

Bảng 3.30	Đặc tả bảng UserPerformanceStats.....	40
Bảng 3.31	Đặc tả bảng ProgramOutcomeTracking.....	41
Bảng 3.32	Đặc tả bảng UserLOTrackings.....	41
Bảng 3.33	Đặc tả bảng StudentProgramProgress.....	42
Bảng 3.34	Đặc tả bảng SubjectOutcomeAnalysis.....	42
Bảng 3.35	Đặc tả bảng UserQuizTrackings	43
Bảng 3.36	Đặc tả bảng UserQuestionHistories	43
Bảng 3.37	Đặc tả bảng QuizAnalytics	43
Bảng 3.38	Đặc tả bảng QuizScoringStats.....	44
Bảng 3.39	Đặc tả bảng PracticeRecommendations.....	44
Bảng 3.40	Đặc tả bảng LeaderboardEntries	45
Bảng 3.41	Đặc tả bảng RankingHistory	45
Bảng 3.42	Đặc tả bảng LeaderboardAchievements	46
Bảng 3.43	Đặc tả bảng SocialInteractions.....	46
Bảng 3.44	Đặc tả bảng UserSocialStats	46
Bảng 3.45	Đặc tả bảng EmojiUsageHistory.....	47
Bảng 3.46	Đặc tả bảng Quizzes.....	47
Bảng 3.47	Đặc tả bảng Questions.....	48
Bảng 3.48	Đặc tả bảng Answers.....	48
Bảng 3.49	Đặc tả bảng QuizQuestions.....	48
Bảng 3.50	Đặc tả bảng Currencies	48
Bảng 3.51	Đặc tả bảng UserCurrencies.....	49
Bảng 3.52	Đặc tả bảng Transactions	49
Bảng 3.53	Đặc tả bảng UserInventories	49
Bảng 3.54	Đặc tả bảng Level	50
Bảng 3.55	Đặc tả bảng UserLevelProgress	50
Bảng 3.56	Đặc tả bảng Badges.....	50

DANH MỤC TỪ VIẾT TẮT

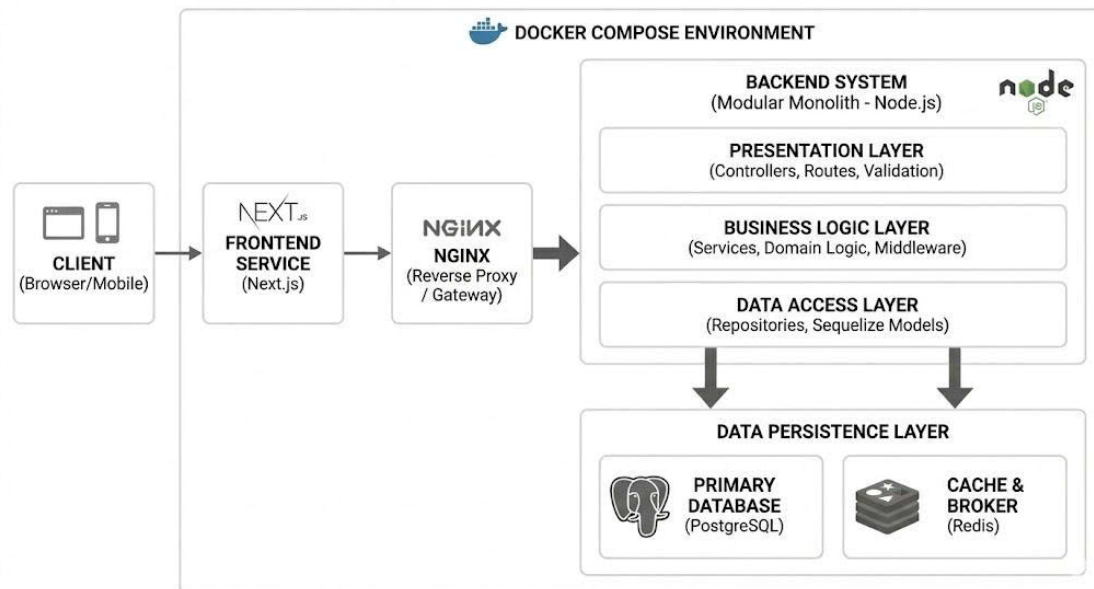
STT	Tên từ viết tắt	Diễn giải
1	AI	Trí tuệ nhân tạo (Artificial Intelligence)
2	API	Giao diện lập trình ứng dụng (Application Programming Interface)
3	AUN-QA	Đảm bảo chất lượng mạng lưới các trường đại học Đông Nam Á (ASEAN University Network - Quality Assurance)
4	CI/CD	Tích hợp liên tục và Triển khai liên tục (Continuous Integration / Continuous Deployment)
5	CNTT	Công nghệ thông tin
6	CSDL	Cơ sở dữ liệu
7	CTĐT	Chương trình đào tạo
8	HTTP	Giao thức truyền tải siêu văn bản (Hypertext Transfer Protocol)
9	JSON	Định dạng trao đổi dữ liệu đối tượng JavaScript (JavaScript Object Notation)
10	JWT	Mã thông báo web JSON dùng để xác thực (JSON Web Token)
11	LMS	Hệ thống quản lý học tập (Learning Management System)
12	LO	Chuẩn đầu ra môn học (Learning Outcomes)
13	ORM	Kỹ thuật ánh xạ quan hệ đối tượng (Object-Relational Mapping)
14	PLO	Chuẩn đầu ra chương trình đào tạo (Program Learning Outcomes)
15	PO	Mục tiêu chương trình đào tạo (Program Objectives)
16	RBAC	Kiểm soát truy cập dựa trên vai trò (Role-Based Access Control)
17	REST	Kiến trúc chuyển giao trạng thái đại diện (Representational State Transfer)
18	SQL	Ngôn ngữ truy vấn có cấu trúc (Structured Query Language)
19	SSL	Lớp công bảo mật (Secure Sockets Layer)
20	XP	Điểm kinh nghiệm (Experience Points)

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đồ án này trình bày quá trình nghiên cứu, thiết kế và xây dựng một hệ thống Backend hoàn chỉnh và hiệu năng cao cho một nền tảng học tập trực tuyến thế hệ mới. Trọng tâm của dự án là giải quyết những hạn chế hiện tại của các hệ thống quản lý học tập (LMS) truyền thống bằng cách tích hợp sâu các yếu tố trò chơi vào lõi của hệ thống, nhằm mục đích tăng cường động lực học tập và sự tương tác của sinh viên.

Vấn đề nghiên cứu cốt lõi mà đề tài tập trung giải quyết là làm thế nào để xây dựng một hạ tầng phía máy chủ (server-side) có thể đáp ứng đồng thời hai yêu cầu phức tạp: một bên là khả năng quản lý chặt chẽ, chi tiết các nghiệp vụ học thuật theo chuẩn đầu ra (PO, PLO, LO) phục vụ cho công tác kiểm định chất lượng; một bên là việc cung cấp một hệ sinh thái trò chơi hóa phong phú, linh hoạt và các tính năng tương tác thời gian thực để tạo ra một trải nghiệm học tập hấp dẫn.

Hướng tiếp cận được lựa chọn trong nghiên cứu này là xây dựng một kiến trúc monolithic theo hướng mô-đun hóa (monolithic modular architecture), nhằm cân bằng giữa khả năng mở rộng, tính dễ bảo trì và sự đơn giản trong triển khai. Cách tiếp cận này cho phép tổ chức hệ thống thành các mô-đun chức năng độc lập về mặt logic, đồng thời tránh được độ phức tạp không cần thiết của các kiến trúc phân tán trong giai đoạn đầu phát triển. Tầng ứng dụng được triển khai trên nền tảng Node.js kết hợp với Express.js, tận dụng mô hình xử lý I/O không đồng bộ để hỗ trợ hiệu quả các kịch bản có nhiều kết nối đồng thời. Hệ quản trị cơ sở dữ liệu PostgreSQL được sử dụng làm kho dữ liệu quan hệ chính, nhờ khả năng đảm bảo toàn vẹn dữ liệu và hỗ trợ các truy vấn phức tạp. Để cải thiện hiệu năng truy cập và hỗ trợ các tương tác gần thời gian thực, hệ thống tích hợp Redis như một lớp lưu trữ bộ nhớ đệm và quản lý trạng thái tạm thời, đồng thời sử dụng Socket.IO để xây dựng các kênh giao tiếp hai chiều giữa máy khách và máy chủ. Trong quá trình phát triển API, nghiên cứu áp dụng phương pháp tiếp cận Design-First, trong đó Đặc tả giao tiếp giữa frontend và Backend được xác định ngay từ đầu dưới dạng các hợp đồng API rõ ràng. Các hợp đồng này được hiện thực hóa và kiểm thử thông qua các công cụ hỗ trợ thiết kế API, qua đó tạo điều kiện cho việc phát triển song song và giảm thiểu rủi ro tích hợp giữa các thành phần của hệ thống.



Hình 1 Sơ đồ một kiến trúc monolithic theo hướng mô-đun hóa

Cách giải quyết vấn đề được cụ thể hóa thông qua việc hiện thực hóa các thành phần kỹ thuật cốt lõi theo định hướng kiến trúc đã lựa chọn. Trước hết, hệ thống xây dựng một lược đồ cơ sở dữ liệu toàn diện, được thiết kế và tổ chức theo ba mô-đun nghiệp vụ chính, bao gồm: (i) quản lý học thuật, (ii) kiểm tra và đánh giá và (iii) trò chơi hóa và phân tích học tập. Việc phân tách này nhằm phản ánh cấu trúc miền nghiệp vụ, đồng thời tạo điều kiện cho việc kiểm soát tính toàn vẹn dữ liệu và khả năng mở rộng hệ thống. Trên nền tảng dữ liệu đó, các chức năng của hệ thống được cung cấp thông qua các dịch vụ API theo phong cách RESTful, phục vụ cho các tác vụ quản trị và xử lý nghiệp vụ, kết hợp với hệ thống sự kiện WebSocket để hỗ trợ các kịch bản yêu cầu tương tác và cập nhật theo thời gian thực. Cách tiếp cận này cho phép hệ thống đáp ứng hiệu quả các luồng nghiệp vụ có mức độ đồng thời cao, đặc biệt trong bối cảnh tổ chức các hoạt động kiểm tra và đánh giá trực tuyến. Các luồng nghiệp vụ được thiết kế bao phủ từ những tác vụ quản trị cơ bản, như quản lý chương trình đào tạo, học phần và ngân hàng câu hỏi, đến các chức năng có độ phức tạp cao hơn, bao gồm: nhập câu hỏi hàng loạt từ tệp nén (ZIP) với cơ chế kiểm tra hợp lệ và chuẩn hóa dữ liệu, tổ chức các phiên làm bài kiểm tra theo thời gian thực với quản lý trạng thái và xử lý tái kết nối, cũng như cập nhật bảng xếp hạng và tiến trình học tập tức thì nhằm hỗ trợ các yếu tố trò chơi hóa và phân tích hành vi người học.

Một số kết quả chính đạt được trong quá trình triển khai hệ thống bao gồm việc xây dựng thành công một hệ thống Backend có cấu trúc rõ ràng và khả năng mở rộng ở mức triển khai, đáp ứng yêu cầu vận hành và phát triển trong bối cảnh ứng dụng giáo

dục trực tuyến. Bên cạnh đó, mô hình dữ liệu được thiết kế chi tiết và có tính hệ thống, không chỉ phục vụ cho nhu cầu lưu trữ và xử lý nghiệp vụ hiện tại mà còn tạo nền tảng cho việc mở rộng các hoạt động phân tích học tập trong các giai đoạn nghiên cứu tiếp theo. Hệ thống cũng đã hiện thực hóa các cơ chế tương tác theo thời gian thực và các luồng nghiệp vụ có mức độ phức tạp cao, qua đó góp phần cải thiện trải nghiệm người dùng so với các hệ thống quản lý học tập truyền thống vốn chủ yếu dựa trên tương tác bất đồng bộ. Về mặt triển khai, toàn bộ hệ thống được container hóa bằng Docker và được quản lý thông qua Docker Compose, giúp đảm bảo tính nhất quán của môi trường thực thi, tăng tính di động của ứng dụng và hỗ trợ quá trình triển khai, thử nghiệm trên nhiều môi trường khác nhau một cách thuận tiện.

MỞ ĐẦU

Lý do chọn đề tài

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong giáo dục đại học, các cơ sở đào tạo tại Việt Nam hiện nay phải đối mặt với một thách thức kép: vừa cần hiện đại hóa công tác quản lý đào tạo nhằm đáp ứng các yêu cầu kiểm định chất lượng và tiếp cận đào tạo theo chuẩn đầu ra (outcome-based education), vừa phải đổi mới phương pháp tổ chức dạy học để tăng cường mức độ tương tác và duy trì động lực học tập của người học. Mặc dù trên thị trường đã tồn tại nhiều hệ thống quản lý học tập (LMS), bao gồm cả các giải pháp mã nguồn mở và thương mại, phần lớn các hệ thống này vẫn bộc lộ những hạn chế nhất định. Cụ thể, chúng thường thiếu khả năng tùy biến sâu để quản lý chi tiết chương trình đào tạo theo đặc thù từng cơ sở hoặc được thiết kế với trải nghiệm người dùng thiên về quản lý hành chính, chưa chú trọng đầy đủ đến các yếu tố tạo động lực và sự tham gia tích cực của người học.

Xuất phát từ khoảng trống đó, đề tài “Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi” được lựa chọn thực hiện. Mục tiêu của đề tài là đề xuất và hiện thực hóa một giải pháp công nghệ có khả năng đáp ứng đồng thời hai yêu cầu: (i) hỗ trợ hiệu quả công tác quản lý học thuật cho nhà trường và giảng viên và (ii) góp phần xây dựng một môi trường học tập trực tuyến có tính tương tác cao, khuyến khích sự tham gia và ghi nhận nỗ lực của người học thông qua các yếu tố trò chơi hóa.

Trong bối cảnh đó, việc phát triển một hệ thống Backend tùy chỉnh được xem là hướng tiếp cận phù hợp, cho phép tích hợp chặt chẽ và linh hoạt các yêu cầu quản lý học thuật và trải nghiệm học tập, qua đó khắc phục những hạn chế vốn có của các giải pháp LMS thương mại hiện nay.

Mục đích nghiên cứu

Mục đích của đề tài là nghiên cứu, thiết kế và hiện thực hóa một hệ thống Backend, đóng vai trò là nền tảng hạ tầng kỹ thuật cho một ứng dụng học tập trực tuyến. Các mục tiêu cụ thể bao gồm:

(i) Thiết kế mô hình dữ liệu có khả năng mở rộng, cho phép biểu diễn và quản lý các mối quan hệ nghiệp vụ phức tạp trong giáo dục đại học, đặc biệt là hệ thống chuẩn đầu ra đa cấp (PO, PLO, LO)

(ii) Xây dựng hệ thống API theo kiến trúc RESTful, đảm bảo hiệu quả, bảo mật và khả năng tài liệu hóa, nhằm hỗ trợ các ứng dụng phía client tương tác với hệ thống.

(iii) Hiện thực hóa các cơ chế tương tác thời gian thực, tiêu biểu là luồng làm bài kiểm tra và cập nhật bảng xếp hạng, nhằm hỗ trợ các kịch bản yêu cầu phản hồi tức thì.

(iv) Tích hợp các cơ chế trò chơi hóa ở mức Backend thông qua mô hình dữ liệu, logic nghiệp vụ và sự kiện hệ thống, qua đó hình thành các vòng lặp tương tác tích cực hỗ trợ động lực học tập của người học

Đối tượng và Phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài được xác định một cách toàn diện, bao quát các khía cạnh cần thiết để xây dựng một hệ thống Backend hoàn chỉnh:

(i) Về nghiệp vụ: Nghiên cứu các quy trình quản lý chương trình đào tạo theo chuẩn đầu ra; cấu trúc và quản lý ngân hàng câu hỏi điện tử; các cơ chế tổ chức kiểm tra, đánh giá trực tuyến; và các nguyên tắc thiết kế trò chơi hóa ở mức hệ thống.

(ii) Về kỹ thuật: Nghiên cứu và ứng dụng ngăn xếp công nghệ theo hướng PERN, tập trung vào Backend với Node.js, Express.js và PostgreSQL; cùng các công nghệ hỗ trợ như Socket.IO, Redis, ORM, Docker, các mẫu thiết kế phần mềm và tiêu chuẩn bảo mật web.

Phạm vi nghiên cứu được xác định nhằm đảm bảo tính tập trung và khả thi trong thời gian thực hiện:

(i) Đề tài tập trung hoàn toàn vào phát triển Backend, bao gồm thiết kế cơ sở dữ liệu, hiện thực logic nghiệp vụ và xây dựng API; không đi sâu vào phát triển giao diện người dùng.

(ii) Đề tài sẽ tập trung vào các luồng nghiệp vụ cốt lõi đã xác định như quản lý học thuật, hệ thống kiểm tra đa chế độ và yếu tố trò chơi. Các chức năng phụ trợ như diễn đàn, hệ thống nhắn tin nội bộ, hoặc cổng thanh toán sẽ không được triển khai.

(iii) Đề tài tập trung vào các luồng nghiệp vụ cốt lõi như quản lý học thuật, hệ thống kiểm tra và đánh giá trực tuyến và các cơ chế trò chơi hóa; các chức năng mở rộng không thuộc phạm vi nghiên cứu.

(iv) Hệ thống được thiết kế cho kịch bản sử dụng ở quy mô người dùng đồng thời mức trung bình; các hướng tiếp cận nâng cao như kiến trúc microservices hoặc tích hợp trí tuệ nhân tạo không nằm trong phạm vi nghiên cứu.

CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU

1.1 Giới thiệu đề tài

Trong bối cảnh cuộc cách mạng chuyển đổi số đang định hình lại nền giáo dục đại học tại Việt Nam và trên thế giới, việc ứng dụng công nghệ để hiện đại hóa quy trình dạy và học đã trở thành một yêu cầu tất yếu. Tuy nhiên, một khoảng trống đáng kể vẫn tồn tại giữa các chức năng phổ thông của những hệ thống quản lý học tập (LMS) hiện có và các yêu cầu nghiệp vụ đặc thù, ngày càng phức tạp của giáo dục đại học. Các giải pháp truyền thống thường gặp khó khăn trong việc chuẩn hóa chương trình đào tạo theo các quy định nghiêm ngặt và thiếu các cơ chế hiệu quả để tạo động lực học tập cho sinh viên trong quá trình học tập.

Để giải quyết bài toán này, đề tài “Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi” ra đời nhằm xây dựng một nền tảng hạ tầng dữ liệu và API vững chắc, tập trung vào hai trụ cột chính:

(i) Quản lý học thuật theo chuẩn đầu ra chuyên sâu: Các trường đại học hiện nay phải tuân thủ các quy định của Bộ Giáo dục và các chuẩn kiểm định quốc tế như AUN-QA. Do đó, hệ thống phải có khả năng quản lý và ánh xạ chặt chẽ hệ thống chuẩn đầu ra đa cấp, từ Chuẩn đầu ra chương trình (PO, PLO) đến Chuẩn đầu ra môn học (LO). Việc số hóa toàn diện công tác quản lý chương trình đào tạo, môn học, phân công giảng dạy và theo dõi tiến độ theo từng chuẩn là một nhu cầu cấp thiết mà các hệ thống hiện tại chưa đáp ứng triệt để.

(ii) Hiện đại hóa hệ thống kiểm tra - đánh giá tích hợp Gamification: Thay vì các hình thức kiểm tra đơn điệu, hệ thống hướng tới cung cấp một nền tảng đánh giá đa chế độ, linh hoạt, bao gồm:

- Assessment Mode: Kiểm tra đánh giá chính thức.
- Practice Mode: Luyện tập không giới hạn với phản hồi tức thời.
- Adaptive Mode: Tự động điều chỉnh độ khó theo trình độ người học.

Các chế độ này không hoạt động độc lập mà được tích hợp sâu với hệ thống phân tích học tập và các yếu tố trò chơi hóa như điểm thưởng, huy hiệu, cấp độ nhằm tăng cường động lực và mang lại trải nghiệm học tập hấp dẫn hơn.

1.1.1 Đặt vấn đề

Mặc dù có nhiều hệ thống quản lý học tập (LMS - Learning Management System) như Moodle, Canvas, Blackboard, nhưng các hệ thống này thường tồn tại một số hạn chế, cụ thể:

- **Thiếu tính tùy biến:** Khó điều chỉnh để phù hợp với yêu cầu quản lý chương trình đào tạo theo chuẩn của từng trường khác nhau.
- **Giao diện phức tạp:** Khó sử dụng, đặc biệt với giảng viên và sinh viên không am hiểu công nghệ.
- **Thiếu yếu tố game hóa:** Chưa tích hợp các cơ chế tăng động lực học tập như điểm thưởng, cấp độ, huy hiệu
- **Hạn chế về phân tích:** Chưa cung cấp công cụ phân tích học tập chi tiết theo từng chuẩn đầu ra
- **Chi phí cao:** Giấy phép sử dụng thường tốn kém, không phù hợp với ngân sách của nhiều trường

Do đó, việc xây dựng một hệ thống Backend tùy chỉnh cho nền tảng học tập trực tuyến là cần thiết để đáp ứng đúng nhu cầu quản lý CTĐT theo quy định Việt Nam đồng thời tích hợp sâu các tính năng game hóa để tăng động lực học tập qua đó cung cấp phân tích học tập chi tiết dựa trên dữ liệu giúp tối ưu hiệu năng cho nhu cầu sử dụng thực tế và kiểm soát chi phí và quyền sở hữu mã nguồn.

1.1.2 Phạm vi và giới hạn

Phạm vi thực hiện đề tài tập trung chuyên biệt vào xây dựng hệ thống Backend cho nền tảng học tập trực tuyến. Toàn bộ công việc thiết kế, tối ưu hóa schema database, phát triển các logic nghiệp vụ (business logic), xây dựng API, đảm bảo bảo mật, cũng như tối ưu hóa hiệu năng – đều do Backend đảm nhận. Ngoài ra, dự án chú trọng phát triển các chức năng như bài kiểm tra trắc nghiệm thời gian thực, bảng điều khiển (dashboard) phân tích kết quả học tập, trò chơi hóa, quản lý bảng xếp hạng và cá nhân hóa lộ trình học tập cho từng sinh viên.

Giới hạn của đề tài là không đi sâu vào các module giao diện người dùng cuối, các chức năng diễn đàn (forum), hệ thống nhắn tin nội bộ, hoặc module thanh toán và

tích hợp bên ngoài. Backend được xây dựng nhằm quản lý dữ liệu, API và các Logic nghiệp vụ phía máy chủ, phục vụ cho quá trình phát triển frontend về sau. Số lượng người dùng đồng thời phục vụ tối đa là 500 – 1.000, phù hợp với quy mô lớp học hoặc trường đại học cỡ vừa. Hệ thống chưa triển khai các tính năng nâng cao như AI chấm điểm tự động toàn diện, hoặc các microservice xử lý phân tán độc lập ngoài phạm vi demo thử nghiệm.

1.2 Mục đích và yêu cầu

1.2.1 Mục đích

Đề tài là phát triển một Backend chuẩn hóa theo đúng quy trình phát triển phần mềm – từ thiết kế cơ sở dữ liệu, API, logic nghiệp vụ cho tới triển khai Docker hóa. Hệ thống cần đảm bảo hoạt động ổn định khi lưu trữ và xử lý nhiều loại dữ liệu đào tạo, sẵn sàng mở rộng thêm dịch vụ thông minh giúp sinh viên/giảng viên nâng cao hiệu quả học tập. Tối ưu hóa tốc độ truy xuất, đảm bảo bảo mật, khả năng backup dữ liệu, phân quyền quản lý rõ ràng và đáp ứng tất cả các chuẩn nhiệm vụ được đưa ra trong đề cương.

1.2.2 Yêu cầu

Về chức năng: Hệ thống Backend phải có module quản lý chương trình đào tạo, quản lý môn học, bài kiểm tra trắc nghiệm thời gian thực, bảng điều khiển (dashboard) phân tích kết quả học tập cho giảng viên/sinh viên, hệ thống trò chơi hóa vinh danh và xử lý kiểm tra hoàn thành từng chuẩn đầu ra. Mỗi API phải trả về dữ liệu nhanh, đúng định dạng, có khả năng kiểm tra log và xử lý lỗi chi tiết.

Về phi chức năng: Hệ thống phải có bảo mật dữ liệu cá nhân (JWT, phân quyền), tối ưu hóa truy vấn, khả năng mở rộng, tài liệu hướng dẫn tích hợp, backup tự động, kiểm thử đầy đủ chức năng và lỗi nghiệp vụ. Phải đảm bảo khả năng triển khai thực tế, sẵn sàng tích hợp với frontend hoặc service ngoài nếu có nhu cầu mở rộng.

1.3 Ý nghĩa của đề tài

1.3.1 Đối với sinh viên

Hệ thống Backend mới giúp tăng động lực học tập và sự gắn kết của sinh viên. Khi tham gia vào môi trường học trực tuyến, người học sẽ được trải nghiệm nhiều tính năng trò chơi hóa như điểm thưởng, huy hiệu và thăng cấp, tạo cảm giác tiến bộ rõ ràng sau mỗi lần hoàn thành nhiệm vụ hoặc vượt qua thử thách. Bảng xếp hạng liên tục cập

nhất giúp sinh viên có thêm động lực cạnh tranh lành mạnh với bạn bè, từ đó nâng cao sự cố gắng mỗi ngày. Những phản hồi tức thời sau bài kiểm tra hoặc hoạt động luyện tập làm cho quá trình học luôn được duy trì liên tục, đồng thời hệ thống streak tracking khuyến khích sinh viên duy trì thói quen học đều đặn. Ngoài ra, các phần thưởng như trứng (egg rewards) hay hệ thống shop tạo cảm giác bất ngờ và đặt ra những mục tiêu ngắn hạn giúp sinh viên chủ động hơn trong quá trình học tập.

Không chỉ dừng lại ở sự gắn kết và động lực, hệ thống còn hỗ trợ sinh viên rèn luyện chủ động, cá nhân hóa lộ trình học. Các chế độ luyện tập không giới hạn cùng với feedback tức thời cho phép sinh viên làm đi làm lại đến khi nắm chắc kiến thức. Chế độ thích ứng tự động điều chỉnh độ khó phù hợp với trình độ hiện tại của mỗi cá nhân, giảm cảm giác quá tải hoặc nhàm chán. Dữ liệu phân tích cá nhân sẽ giúp sinh viên nhìn thấy điểm mạnh – yếu của mình, từ đó nhận được các gợi ý về nội dung cần cải thiện, kiểm soát tiến độ học tập rõ ràng và hiệu quả hơn.

Đặc biệt, với bảng điều khiển (dashboard) cá nhân, mọi điểm số, xếp hạng, tiến độ học tập đều hiển thị trực quan theo thời gian thực. Hệ thống cũng cung cấp phân tích chi tiết theo từng cụm chuẩn đầu ra (LO), giúp mỗi sinh viên hiểu rõ năng lực cá nhân và lịch sử làm bài cũng được lưu lại, dễ dàng truy xuất để ôn luyện hoặc so sánh với kết quả trung bình của lớp. Việc theo dõi chuỗi ngày học liên tục càng củng cố thói quen chủ động và tự kiểm tra quá trình rèn luyện của bản thân.

1.3.2 Đối với giảng viên

Giảng viên được hỗ trợ bởi một bảng điều khiển (dashboard) trực quan tổng kết toàn bộ hiệu suất lớp học. Những dữ liệu phân tích sâu về từng câu hỏi, nhất là phân tích phương án trả lời sai hay bị chọn nhầm, giúp thầy cô phát hiện sớm nhóm sinh viên có nguy cơ học yếu hoặc những chủ đề học chưa đạt yêu cầu. Việc nhận diện sinh viên có nguy cơ tụt lại phía sau trở nên dễ dàng hơn thông qua các báo cáo dựa trên kết quả đầu ra, bám sát từng chuẩn đầu ra (PO, PLO, LO) và thậm chí có thể theo dõi tiến bộ theo thời gian thực.

Không chỉ vậy, hệ thống còn đem lại lợi ích rõ rệt về mặt tiết kiệm thời gian và công sức. Chức năng tự động tính điểm cho các bài trắc nghiệm giảm tải cho giảng viên khâu chấm điểm, ngân hàng câu hỏi tự động lưu trữ, phân loại và tái sử dụng tiện lợi. Chỉ với vài thao tác, giảng viên có thể tạo ra các đề kiểm tra mới bám sát phân bố chuẩn

đầu ra nhờ tính năng Bản thiết kế bài kiểm tra. Toàn bộ quá trình tổng hợp kết quả, tính điểm đều được hệ thống xử lý tự động và cung cấp các báo cáo có thể xuất ngay, hỗ trợ công việc quản lý học tập dễ dàng.

Đặc biệt với hệ thống phân tích dữ liệu thông minh, giảng viên nhận biết ngay lập tức các chủ đề sinh viên còn gặp khó khăn, từ đó có thể điều chỉnh nội dung giảng dạy hoặc phương pháp truyền đạt cho phù hợp. Chức năng thử nghiệm A/B với các cấu hình bài kiểm tra khác nhau cũng giúp xác minh hiệu quả của tài liệu giảng dạy hiện tại, liên tục cải tiến phương pháp theo hướng lấy dữ liệu thực tế làm trọng tâm. Nhờ vậy, chất lượng giảng dạy được nâng cao, người học nhận được hỗ trợ kịp thời và chương trình đào tạo ngày càng tối ưu hơn.

1.4 Đối tượng sử dụng và các kịch bản chính

Để hiểu rõ hơn bối cảnh ứng dụng của hệ thống, phần này sẽ mô tả các nhóm đối tượng người dùng chính tương ứng các kịch bản sử dụng tiêu biểu mà hệ thống Backend được thiết kế để phục vụ.

Hệ thống hướng tới ba nhóm đối tượng người dùng chính trong môi trường giáo dục đại học:

(i) Sinh viên: Là người dùng cuối, trực tiếp tham gia vào các hoạt động học tập, làm bài kiểm tra, luyện tập và tương tác với các tính năng trò chơi hóa.

(ii) Giảng viên: Là người tạo và quản lý nội dung khóa học, thiết kế các bài kiểm tra, theo dõi tiến độ của sinh viên và sử dụng các công cụ phân tích để cải thiện chất lượng giảng dạy.

(iii) Quản trị viên học vụ (Admin): Là người có quyền cao nhất, chịu trách nhiệm quản lý toàn bộ hệ thống, từ việc thiết lập chương trình đào tạo, quản lý tài khoản người dùng, đến việc phân công giảng dạy và giám sát hoạt động chung.

1.5 Các quy trình nghiệp vụ trọng yếu

Để khắc phục các hạn chế của hệ thống cũ và đáp ứng mục tiêu đề ra, hệ thống Backend được thiết kế để vận hành ba quy trình nghiệp vụ cốt lõi. Đây là những bài toán phức tạp đòi hỏi sự xử lý chính xác và hiệu năng cao từ phía máy chủ:

(i) Quy trình số hóa và chuẩn hóa đánh giá năng lực

Hệ thống chuyển đổi phương pháp ra đề thi từ thủ công sang tự động hóa hoàn toàn. Thay vì chọn từng câu hỏi, giảng viên chỉ cần định nghĩa ma trận đề thi dựa trên các chuẩn đầu ra của học phần (LO).

Vai trò của Backend: Hệ thống thực thi các thuật toán truy vấn ngẫu nhiên từ ngân hàng câu hỏi để sinh ra các mã đề khác nhau nhưng đảm bảo độ cân bằng về độ khó và tỷ lệ kiến thức (ví dụ: 40% câu hỏi thuộc LO1, 60% thuộc LO2). Quy trình này đảm bảo tính khách quan trong đánh giá và tiết kiệm tối đa thời gian cho giảng viên.

(ii) Quy trình xử lý tương tác thời gian thực và hệ thống trò chơi

Khác với các hệ thống LMS tĩnh, nền tảng này tập trung vào trải nghiệm tức thì. Trong các phiên làm bài kiểm tra (Quiz) hoặc chế độ đua tranh giải, hệ thống phải phục vụ hàng trăm sinh viên thao tác cùng một lúc.

Vai trò của Backend: Backend duy trì các kênh kết nối thời gian thực (Real-time sockets) để tiếp nhận câu trả lời, chấm điểm ngay lập tức (Instant Feedback) và kích hoạt song song các dịch vụ trò chơi hóa. Điểm kinh nghiệm (XP), vật phẩm ảo và thứ hạng trên Bảng xếp hạng (Leaderboard) được tính toán và cập nhật trong tích tắc, tạo ra vòng lặp phản hồi tích cực kích thích người học.

(iii) Quy trình quản trị và xử lý dữ liệu hàng loạt (Batch Processing)

Để phục vụ quy mô đào tạo lớn, hệ thống cung cấp khả năng xử lý dữ liệu đầu vào khối lượng lớn, điển hình là việc nhập danh sách sinh viên đầu khóa hoặc ngân hàng câu hỏi từ các tệp tin bên ngoài (Excel, ZIP).

Vai trò của Backend: Hệ thống thực hiện các cơ chế phân tích, xác thực dữ liệu và sử dụng Giao tác chặt chẽ. Điều này đảm bảo tính toàn vẹn cao của dữ liệu: một tệp tin nhập vào sẽ được xử lý thành công toàn bộ hoặc hủy bỏ toàn bộ (rollback) nếu có lỗi, ngăn chặn triệt để tình trạng dữ liệu rác hoặc không đồng bộ.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

Chương này trình bày các lý thuyết nền tảng được nghiên cứu để chuẩn bị cho việc phát triển dự án. Việc xây dựng một hệ thống Backend ổn định đòi hỏi sự hiểu biết vững chắc về các công nghệ và kiến trúc được lựa chọn. Nội dung chương sẽ đi từ các khái niệm tổng quan như hệ thống phân tán, cơ sở dữ liệu, đến các công nghệ cụ thể như WebSocket, Docker, bộ đệm, cùng với các nguyên tắc về bảo mật và kiểm thử. Mục tiêu của chương này là cung cấp một cơ sở kiến thức đủ mạnh để các quyết định thiết kế trong những chương sau được đưa ra một cách hợp lý và có cơ sở.

2.1 Tổng quan về Hệ thống Quản lý học tập

Hệ thống Quản lý Học tập (Learning Management System – LMS) là một loại ứng dụng phần mềm được thiết kế nhằm hỗ trợ quản trị, lưu trữ, theo dõi, báo cáo và phân phối các nội dung giáo dục hoặc chương trình đào tạo trong môi trường số. Trong bối cảnh giáo dục hiện đại, LMS đóng vai trò là hạ tầng trung tâm cho các hoạt động dạy và học trực tuyến, cho phép tổ chức, quản lý và đánh giá quá trình học tập của người học một cách có hệ thống. Trên thế giới, nhiều nền tảng LMS đã được triển khai rộng rãi tại các cơ sở giáo dục, tiêu biểu như Moodle, Canvas và Blackboard. Các hệ thống này cung cấp những chức năng nền tảng như quản lý khóa học, phân phối tài liệu học tập, tổ chức diễn đàn thảo luận và hỗ trợ các hình thức kiểm tra, đánh giá trực tuyến, qua đó đáp ứng các nhu cầu cơ bản trong triển khai đào tạo trực tuyến.

Tuy nhiên, qua quá trình khảo sát và phân tích, có thể nhận thấy rằng trong nhiều trường hợp, các hệ thống LMS thương mại hoặc mã nguồn mở phổ biến hiện nay vẫn bộc lộ những hạn chế nhất định khi áp dụng vào bối cảnh giáo dục đại học tại Việt Nam. Thứ nhất, khả năng tùy biến sâu để đáp ứng các yêu cầu quản lý chương trình đào tạo theo chuẩn đầu ra (PO, PLO, LO) cũng như các tiêu chí kiểm định chất lượng đặc thù (ví dụ AUN-QA) thường bị giới hạn, hoặc đòi hỏi chi phí và nỗ lực kỹ thuật đáng kể để mở rộng. Thứ hai, các yếu tố trò chơi hóa trong nhiều hệ thống LMS hiện hữu chủ yếu được triển khai ở mức giao diện hoặc tính năng bổ trợ, chưa được tích hợp một cách có hệ thống vào mô hình dữ liệu và logic nghiệp vụ, dẫn đến mức độ hỗ trợ động lực học tập của người học còn hạn chế.

Từ những nhận định trên, có thể thấy rằng việc nghiên cứu và phát triển một hệ thống Backend tùy chỉnh là cần thiết nhằm khắc phục các hạn chế nêu trên. Hướng tiếp

cần này cho phép can thiệp sâu vào cấu trúc dữ liệu, luồng nghiệp vụ và cơ chế tích hợp, qua đó xây dựng một nền tảng vừa đáp ứng hiệu quả yêu cầu quản lý học thuật, vừa hỗ trợ các cơ chế tương tác và trò chơi hóa một cách linh hoạt, phù hợp với đặc thù của từng cơ sở giáo dục.

Trên thị trường hiện nay, các hệ thống LMS có thể được phân loại thành ba nhóm chính, mỗi nhóm có những ưu điểm và hạn chế riêng khi xem xét khả năng triển khai trong môi trường giáo dục đại học.

Bảng 2.1 So sánh các loại hình hệ thống LMS

Loại hình	Đại diện tiêu biểu	Ưu điểm	Nhược điểm
Mã nguồn mở	Moodle, Sakai	Miễn phí bản quyền; Cộng đồng lớn; Khả năng tùy biến cao nếu có đủ nguồn lực kỹ thuật.	Đòi hỏi chi phí hạ tầng và nhân sự vận hành; Giao diện mặc định thường kém hấp dẫn; Khó khăn trong việc nâng cấp.
Thương mại	Blackboard, Canvas	Giải pháp trọn gói, ổn định; Hỗ trợ kỹ thuật chuyên nghiệp; Giao diện thân thiện.	Chi phí bản quyền rất cao; Bị phụ thuộc vào nhà cung cấp; Khả năng tùy biến hạn chế.
Tùy chỉnh	Dự án này	Đáp ứng chính xác nghiệp vụ đặc thù; Toàn quyền sở hữu mã nguồn; Tích hợp sâu các tính năng độc đáo.	Chi phí và thời gian phát triển ban đầu lớn; Đòi hỏi đội ngũ phát triển để bảo trì và nâng cấp.

Việc lựa chọn xây dựng một hệ thống tùy chỉnh, dù tốn nhiều công sức ban đầu, lại là hướng đi chiến lược cho phép giải quyết triệt để các bài toán nghiệp vụ phức tạp và tạo ra lợi thế cạnh tranh bằng những trải nghiệm học tập độc đáo mà các giải pháp có sẵn không thể cung cấp.

2.2 Lý thuyết về Trò chơi hóa trong giáo dục

Trò chơi hóa (Gamification) được định nghĩa là việc ứng dụng có chủ đích các yếu tố và nguyên tắc thiết kế của trò chơi vào những bối cảnh không phải trò chơi, nhằm tăng cường mức độ tham gia, động lực và hiệu quả của các hoạt động mục tiêu [1]. Trong lĩnh vực giáo dục, gamification không đồng nghĩa với việc biến hoạt động học tập thành một trò chơi hoàn chỉnh, mà tập trung vào việc chắt lọc và vận dụng các cơ chế tâm lý cốt lõi của trò chơi để hỗ trợ và thúc đẩy hành vi học tập tích cực.

Nền tảng lý thuyết quan trọng của gamification trong giáo dục thường được đặt trên Thuyết Tự quyết (Self-Determination Theory – SDT), theo đó động lực nội tại của con người chịu ảnh hưởng bởi ba nhu cầu tâm lý cơ bản [1]:

(i) Tự chủ (Autonomy): nhu cầu được lựa chọn, kiểm soát và ra quyết định trong quá trình học tập;

(ii) Năng lực (Competence): nhu cầu được cảm nhận sự tiến bộ, thành thạo và đạt được kết quả thông qua nỗ lực cá nhân;

(iii) Kết nối xã hội (Relatedness): nhu cầu được tương tác, công nhận và kết nối với những người khác trong môi trường học tập.

Một hệ thống học tập được trò chơi hóa hiệu quả cần được thiết kế sao cho đồng thời hỗ trợ cả ba nhu cầu trên, từ đó duy trì động lực học tập bền vững của người học. Trên cơ sở các lý thuyết động lực học, gamification trong giáo dục thường được triển khai thông qua một tập hợp các cơ chế và thành phần trò chơi, khi được áp dụng hợp lý sẽ hình thành các vòng lặp tương tác tích cực cho người học [1]. Các cơ chế phổ biến bao gồm:

(i) Điểm số (Points / Experience Points – XP): đóng vai trò là thước đo định lượng cho sự nỗ lực và tiến bộ của người học; mỗi hành động học tập tích cực như hoàn thành bài học hoặc trả lời đúng câu hỏi đều có thể được ghi nhận bằng điểm số, tạo phản hồi tức thì.

(ii) Cấp độ (Levels): biểu diễn mức độ phát triển và sự thành thạo của người học trong hệ thống; việc đạt được cấp độ mới thường gắn với các cột mốc tiến bộ hoặc mở khóa nội dung, qua đó hình thành mục tiêu dài hạn.

(iii) Huy hiệu (Badges) và Thành tựu (Achievements): các phần thưởng mang tính biểu trưng, dùng để ghi nhận những thành tích hoặc hành vi học tập cụ thể, góp phần thỏa mãn nhu cầu thể hiện năng lực.

(iv) Bảng xếp hạng (Leaderboards): hỗ trợ yếu tố cạnh tranh có kiểm soát bằng cách hiển thị thứ hạng của người học trong một phạm vi nhất định; cơ chế này cần được thiết kế thận trọng để tránh gây áp lực hoặc làm giảm động lực của những người học ở nhóm dưới.

(v) Tiền tệ ảo (Virtual Currency) và Phần thưởng (Rewards): cho phép người học tích lũy và sử dụng các phần thưởng ảo để đổi lấy những quyền lợi hoặc yếu tố cá nhân hóa trong hệ thống, từ đó tăng cảm giác tự chủ và giá trị của nỗ lực học tập.

Hiệu quả của các cơ chế trò chơi hóa đã được minh họa thông qua nhiều nền tảng giáo dục trực tuyến thành công. Chẳng hạn:

(i) Duolingo áp dụng các cơ chế như chuỗi học tập liên tục (streak), điểm kinh nghiệm và bảng xếp hạng theo chu kỳ nhằm hình thành thói quen học tập đều đặn và duy trì mức độ tham gia cao của người dùng.

(ii) Khan Academy sử dụng hệ thống điểm năng lượng, huy hiệu và avatar để ghi nhận sự tiến bộ và mức độ thành thạo, phù hợp với định hướng học thuật và tự học.

Từ việc phân tích các mô hình trên, đề tài “Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi” chất lọc và kế thừa các nguyên tắc cốt lõi của gamification trong giáo dục, bao gồm:

(i) Thiết kế các vòng lặp phản hồi ngắn, trong đó hành động học tập của người học nhanh chóng được ghi nhận và phản hồi;

(ii) Xây dựng mục tiêu học tập ở cả ngắn hạn và dài hạn, thông qua các cơ chế cấp độ và thành tựu;

(iii) Ghi nhận và so sánh tiến trình học tập bằng các cơ chế xếp hạng và huy hiệu theo phạm vi phù hợp.

Các nguyên tắc này được xem xét và tích hợp chủ yếu ở mức hệ thống Backend, thông qua mô hình dữ liệu, logic nghiệp vụ và cơ chế sự kiện, nhằm đảm bảo tính nhất quán, khả năng mở rộng và hỗ trợ lâu dài cho trải nghiệm học tập.

2.3 Các công nghệ và nền tảng phát triển

Việc lựa chọn công nghệ phù hợp đóng vai trò quan trọng trong quá trình thiết kế và triển khai hệ thống, do ảnh hưởng trực tiếp đến hiệu năng, khả năng mở rộng và khả năng bảo trì. Phần này trình bày các nền tảng và công nghệ được sử dụng trong đề tài, đồng thời phân tích cơ sở của các quyết định lựa chọn.

Node.js được lựa chọn làm môi trường runtime cho hệ thống Backend nhờ vào mô hình xử lý I/O bất đồng bộ và kiến trúc hướng sự kiện, vốn đặc biệt phù hợp với các

ứng dụng yêu cầu xử lý nhiều kết nối đồng thời và hỗ trợ các kịch bản thời gian thực, chẳng hạn như nền tảng học tập trực tuyến của đề tài [2]. Tuy nhiên, trên nền tảng Node.js tồn tại nhiều framework khác nhau, do đó việc lựa chọn framework phù hợp cần được cân nhắc cẩn trọng.

Để làm rõ quyết định này, Bảng 2.2 trình bày so sánh ba framework Backend phổ biến trên Node.js, bao gồm Express.js, NestJS và Koa.js.

Bảng 2.2 So sánh các Framework Backend cho Node.js

Tiêu chí	Express.js	NestJS	Koa.js
Triết lý	Tối giản, linh hoạt, không áp đặt (Unopinionated)	Kiến trúc Module hóa, có cấu trúc chặt chẽ (Opinionated)	Tối giản, hiện đại, kế thừa từ Express
Kiến trúc	Dựa trên Middleware	Lấy cảm hứng từ Angular (Modules, Controllers, Providers)	Dựa trên Middleware, sử dụng Async/Await
Hỗ trợ TypeScript	Hỗ trợ tốt thông qua các tệp định nghĩa kiểu (@types)	Hỗ trợ gốc (Native), là ngôn ngữ chính	Hỗ trợ tốt
Đường cong học tập	Thấp, dễ bắt đầu	Cao, đòi hỏi hiểu biết về các khái niệm như DI, Decorators	Trung bình
Hệ sinh thái	Cực kỳ lớn và trưởng thành	Đang phát triển mạnh, có nhiều module sẵn có	Nhỏ hơn so với Express

Dựa trên kết quả so sánh, Express.js được lựa chọn làm framework chính cho dự án. Express.js là một framework ứng dụng web tối giản, cung cấp các chức năng nền tảng như định tuyến và quản lý middleware mà không áp đặt cấu trúc kiến trúc cứng nhắc [3]. Đặc điểm này cho phép hệ thống được thiết kế theo kiến trúc tùy chỉnh (MVC kết hợp Service Layer), phù hợp với các yêu cầu nghiệp vụ đặc thù của đề tài. Bên cạnh đó, hệ sinh thái phong phú của Express.js hỗ trợ việc mở rộng và xử lý các vấn đề kỹ thuật trong quá trình phát triển.

Cơ sở dữ liệu quan hệ PostgreSQL

PostgreSQL được lựa chọn làm cơ sở dữ liệu chính nhờ khả năng tuân thủ chặt chẽ các tiêu chuẩn SQL, hỗ trợ tốt các ràng buộc toàn vẹn, quan hệ phức tạp và giao tác

an toàn. Những đặc điểm này đặc biệt quan trọng đối với hệ thống quản lý học thuật, nơi dữ liệu có cấu trúc và tính nhất quán cao.

Để tương tác với PostgreSQL, hệ thống sử dụng Sequelize – một Object-Relational Mapper dựa trên Promise cho Node.js. Sequelize cho phép ánh xạ các thực thể trong mã nguồn sang các bảng dữ liệu quan hệ, qua đó hỗ trợ phát triển theo hướng hướng đối tượng, cải thiện khả năng bảo trì và góp phần giảm thiểu các rủi ro bảo mật thường gặp khi thao tác trực tiếp với SQL.

Giao tiếp thời gian thực

Socket.IO được sử dụng để hiện thực cơ chế giao tiếp hai chiều giữa máy chủ và máy khách. Thư viện này xây dựng trên nền WebSocket và có khả năng tự động chuyển sang các phương thức thay thế khi cần thiết, giúp đảm bảo tính tương thích và độ tin cậy cao trong các kịch bản thời gian thực [4]. Trong hệ thống, Socket.IO đóng vai trò trung tâm trong việc điều phối các luồng làm bài kiểm tra và cập nhật các trạng thái theo thời gian thực.

Bên cạnh đó, Redis được sử dụng như một kho lưu trữ dữ liệu trong bộ nhớ nhằm hỗ trợ các yêu cầu về hiệu năng. Redis đảm nhiệm hai vai trò chính: (i) bộ nhớ đệm (caching) cho các dữ liệu được truy cập thường xuyên nhằm giảm tải cho PostgreSQL; và (ii) quản lý trạng thái tạm thời của các phiên làm bài kiểm tra thời gian thực, cho phép hệ thống khôi phục và đồng bộ trạng thái một cách nhanh chóng.

2.4 Kết chương

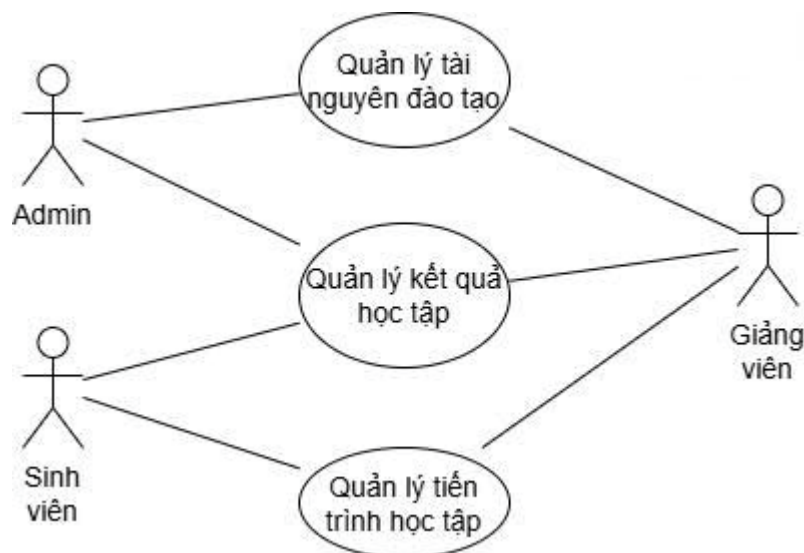
Chương 2 đã trình bày một cách hệ thống các cơ sở lý thuyết làm nền tảng cho dự án. Từ việc phân tích bối cảnh và hạn chế của các hệ thống LMS hiện tại, chương đã làm rõ sự cần thiết của việc tích hợp các lý thuyết về Gamification để tăng cường trải nghiệm học tập. Các công nghệ và nền tảng cốt lõi như Node.js, Express.js, PostgreSQL, Sequelize, Socket.IO và Redis đã được phân tích chi tiết về vai trò và ưu điểm của chúng trong kiến trúc hệ thống. Những kiến thức này sẽ là cơ sở vững chắc cho việc trình bày quá trình thiết kế và hiện thực hóa hệ thống trong chương tiếp theo.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

Chương 3 trình bày về quá trình thiết kế và hiện thực hóa hệ thống Backend cho nền tảng học tập trực tuyến. Dựa trên các cơ sở lý thuyết đã được phân tích ở Chương 2, chương này sẽ đi sâu vào các quyết định kiến trúc, mô hình hóa dữ liệu và phương pháp cài đặt các logic nghiệp vụ phức tạp. Nội dung chính bao gồm việc phân tích kiến trúc hệ thống tổng thể, thiết kế chi tiết cơ sở dữ liệu theo các module chức năng và mô tả quy trình hiện thực hóa các luồng hoạt động cốt lõi thông qua hệ thống API. Mục tiêu của chương là cung cấp một cái nhìn toàn diện về cấu trúc kỹ thuật và cách thức vận hành của hệ thống Backend, tạo tiền đề cho việc đánh giá các kết quả đạt được ở chương tiếp theo.

3.1 Đặc tả yêu cầu

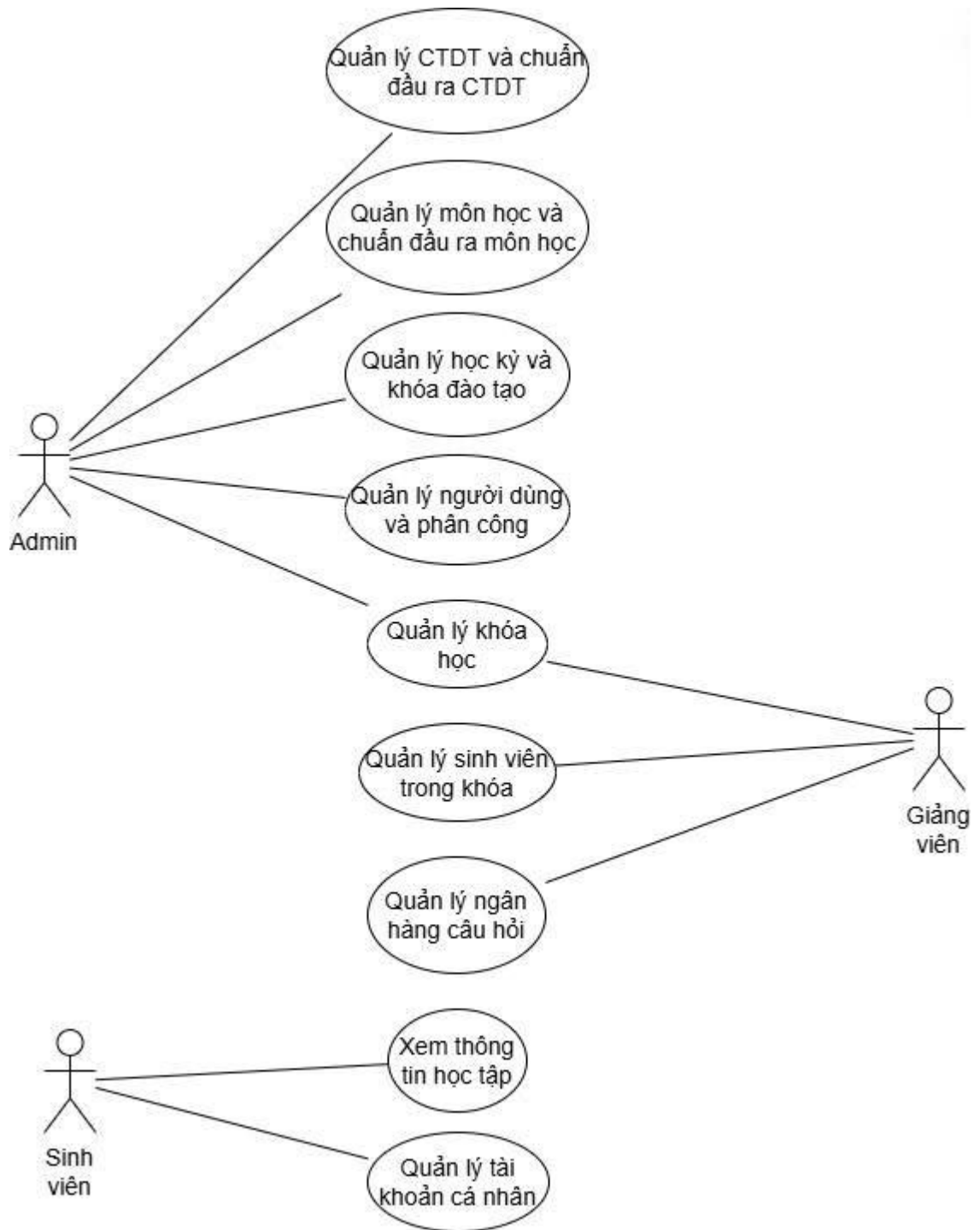
Để hiện thực hóa các mục tiêu nghiên cứu đã đề ra, hệ thống không được xây dựng như một khối chức năng rời rạc, mà được thiết kế như một tổ hợp các phân hệ nghiệp vụ chuyên biệt, có mối liên kết chặt chẽ với nhau nhằm hình thành một quy trình đào tạo khép kín trong môi trường số. Kiến trúc nghiệp vụ tổng thể của hệ thống được Đặc tả thông qua ba phân hệ chính, tương ứng với ba giai đoạn cốt lõi trong chuỗi giá trị của hoạt động đào tạo: (i) tài nguyên đào tạo, (ii) tiến trình học tập và (iii) kết quả học tập.



Hình 3.1 Sơ đồ usecase tổng quát

(i) Phân hệ 1: Quản lý các tài nguyên đào tạo

Phân hệ Quản lý các tài nguyên đào tạo chịu trách nhiệm số hóa và tổ chức toàn bộ cấu trúc đào tạo của cơ sở giáo dục. Yêu cầu cốt lõi của phân hệ này là khả năng định nghĩa, lưu trữ và quản lý mối quan hệ phân cấp phức tạp giữa các thực thể học thuật như chương trình đào tạo, môn học, học phần, lớp học và các nội dung giảng dạy liên quan.

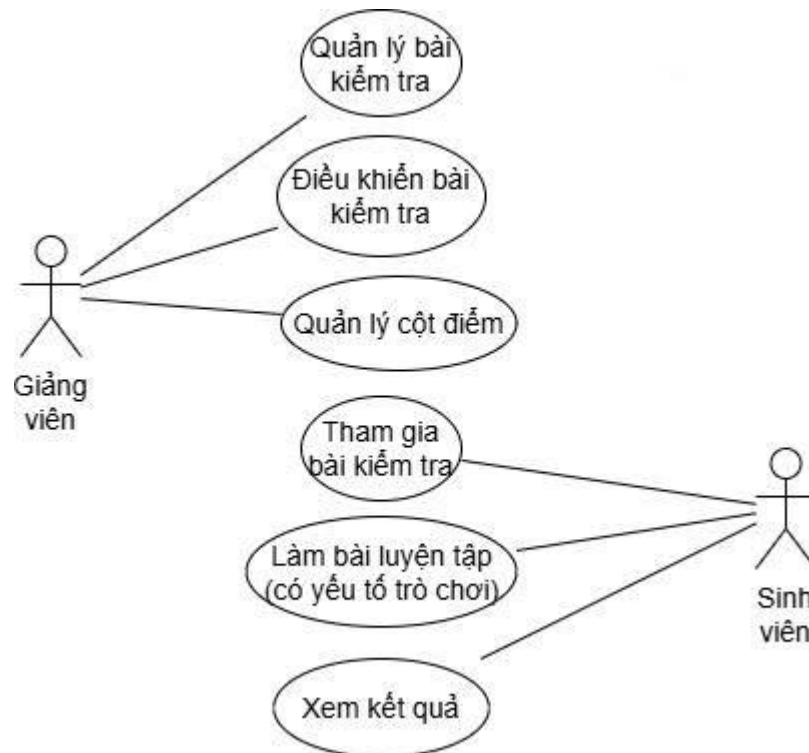


Hình 3.2 Sơ đồ usecase phân hệ quản lý các tài nguyên đào tạo

Đặc biệt, hệ thống phải hỗ trợ ánh xạ đa chiều giữa các chuẩn đầu ra ở nhiều cấp độ (chương trình, môn học, bài học), nhằm đảm bảo rằng mọi hoạt động giảng dạy và học tập đều có thể truy nguyên về các chuẩn đầu ra tương ứng. Đây là yêu cầu then chốt phục vụ cho công tác đảm bảo và kiểm định chất lượng đào tạo, đồng thời tạo nền tảng dữ liệu đầu vào cho các phân hệ tiếp theo.

(ii) Phân hệ 2: Quản lý tiến trình học tập

Phân hệ Quản lý tiến trình học tập tập trung vào việc theo dõi và điều phối toàn bộ quá trình học tập của người học, đặc biệt thông qua các hoạt động luyện tập và đánh giá quá trình. Yêu cầu trọng tâm của phân hệ này là tính linh hoạt và khả năng phản hồi tức thời, đáp ứng các kịch bản học tập đa dạng.



Hình 3.3 Sơ đồ phân hệ quản lý tiến trình học tập

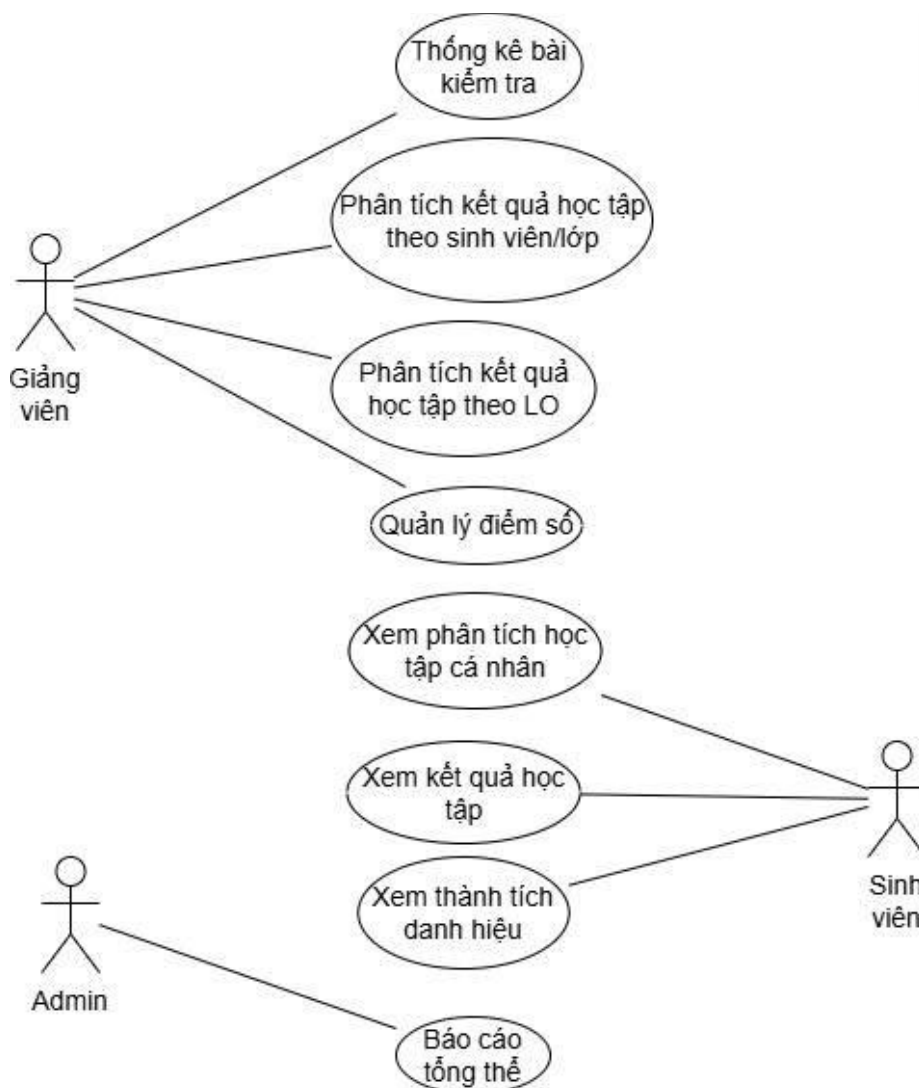
Hệ thống phải hỗ trợ nhiều hình thức đánh giá, bao gồm: các bài kiểm tra chính thức có yêu cầu bảo mật cao; các hoạt động luyện tập cho phép thực hiện nhiều lần nhằm củng cố kiến thức; và các hình thức đánh giá mang tính tương tác cao như trắc nghiệm theo thời gian thực. Phân hệ này có nhiệm vụ ghi nhận chi tiết tiến trình học tập, bao gồm thời gian tham gia, số lần thực hiện, mức độ hoàn thành và kết quả trung gian của người học.

Dữ liệu phát sinh từ phân hệ Quản lý tiến trình học tập đóng vai trò là nguồn dữ liệu động, vừa phản ánh hành vi học tập thực tế, vừa làm đầu vào cho việc tổng hợp và đánh giá kết quả ở cấp độ cao hơn.

(iii) Phân hệ 3: Quản lý kết quả học tập

Phân hệ Quản lý kết quả học tập đảm nhiệm vai trò tổng hợp, phân tích và trình bày các kết quả học tập thu được từ toàn bộ quá trình đào tạo. Phân hệ này tập trung vào việc chuyển đổi dữ liệu tiến trình thành thông tin có ý nghĩa, phục vụ cho cả người học, giảng viên và nhà quản lý.

Các yêu cầu chính bao gồm khả năng thống kê và phân tích kết quả học tập theo nhiều chiều khác nhau, chẳng hạn như theo cá nhân sinh viên, theo lớp học, theo môn học hoặc theo chuẩn đầu ra. Hệ thống cần cung cấp các báo cáo trực quan về mức độ hoàn thành học tập, tiến độ đạt chuẩn đầu ra và xu hướng học tập theo thời gian.



Hình 3.4 Sơ đồ usecase phân hệ quản lý kết quả học tập

Kết quả phân tích từ phân hệ này không chỉ phục vụ cho đánh giá kết quả học tập, mà còn đóng vai trò phản hồi ngược, hỗ trợ việc điều chỉnh nội dung đào tạo và phương thức tổ chức học tập trong các chu kỳ tiếp theo.

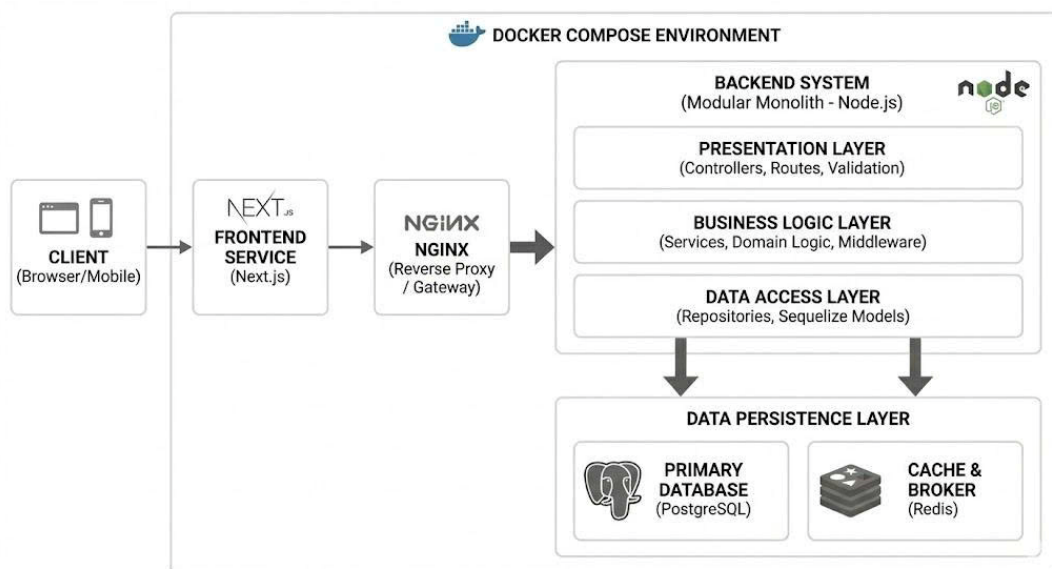
3.2 Thiết kế kiến trúc hệ thống

Để đảm bảo tính linh hoạt, khả năng mở rộng và dễ bảo trì, hệ thống được thiết kế theo kiến trúc 3 tầng, bao gồm: tầng máy khách, tầng ứng dụng và tầng cơ sở dữ liệu [4]. Một thành phần cân bằng tải (Load Balancer) sử dụng Nginx Reverse Proxy được đặt giữa máy khách và tầng ứng dụng để điều phối lưu lượng truy cập, giúp tăng cường hiệu suất và độ ổn định.

Tầng máy khách: Bao gồm các ứng dụng phía người dùng như trình duyệt web hoặc ứng dụng di động, là điểm truy cập chính vào hệ thống.

Tầng ứng dụng: Chứa toàn bộ logic nghiệp vụ. Tầng này bao gồm máy chủ Express.js API và máy chủ Socket.IO xử lý các tác vụ thời gian thực.

Tầng cơ sở dữ liệu: Chịu trách nhiệm lưu trữ và quản lý toàn bộ dữ liệu của hệ thống, bao gồm nhiều hệ quản trị CSDL khác nhau để phục vụ cho các mục đích riêng biệt của hệ thống.



Hình 3.5 Sơ đồ kiến trúc tổng thể hệ thống

Luồng xử lý một yêu cầu HTTP tiêu chuẩn sẽ bắt đầu từ máy khách, đi qua Nginx để phân phối tải, sau đó được máy chủ Express.js tiếp nhận, xử lý và tương tác với tầng cơ sở dữ liệu trước khi trả kết quả về cho người dùng.

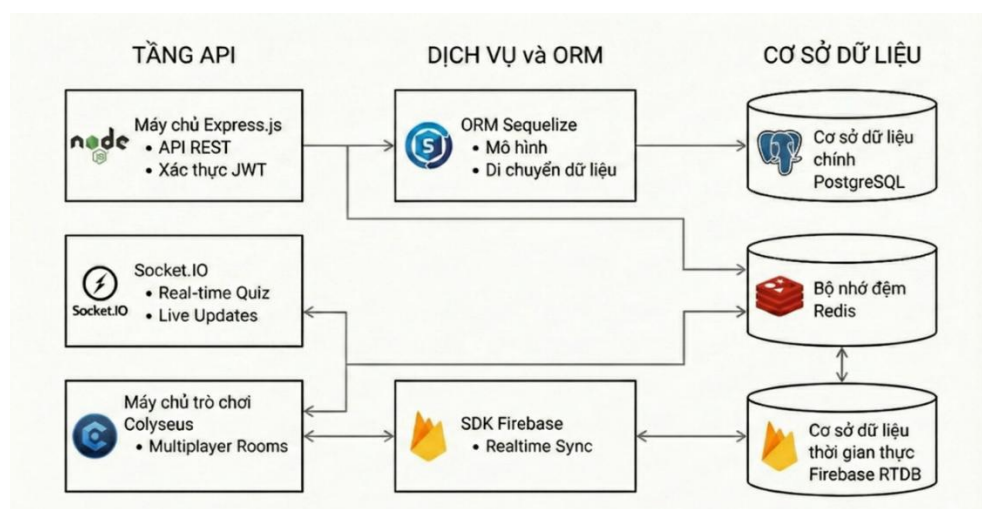
Trong kiến trúc này, Nginx không chỉ đơn thuần đóng vai trò là một bộ cân bằng tải để phân phối các yêu cầu đến các container Backend. Nó được cấu hình như một Reverse Proxy đa chức năng, đóng vai trò là cổng vào duy nhất của toàn bộ hệ thống và đảm nhiệm nhiều tác vụ quan trọng khác nhằm tối ưu hóa hiệu năng và tăng cường bảo mật.

Bên trong Application Layer, phần Express.js Backend được tổ chức theo kiến trúc phân lớp (Layered Architecture) rõ ràng, kế thừa và mở rộng từ mô hình MVC, nhằm tách biệt các mối quan tâm (Separation of Concerns):

Presentation Layer (Routes và Controllers): Tầng này chịu trách nhiệm tiếp nhận các yêu cầu HTTP từ client, xác thực dữ liệu đầu vào và điều hướng chúng đến các Controller tương ứng. Controller không chứa logic nghiệp vụ mà chỉ đóng vai trò điều phối, gọi các Service phù hợp và định dạng dữ liệu trả về.

Business Logic Layer (Services và Middleware): Đây là nơi chứa đựng toàn bộ logic nghiệp vụ phức tạp của ứng dụng. Các Service (ví dụ: QuizThời gian thựcService, GamificationService) thực hiện các tác vụ chuyên biệt. Middleware được sử dụng cho các tác vụ xuyên suốt như xác thực JWT, phân quyền và ghi log.

Data Access Layer (Models và Repositories): Tầng này chịu trách nhiệm duy nhất cho việc tương tác với cơ sở dữ liệu. Sequelize Models định nghĩa cấu trúc của các đối tượng dữ liệu, trong khi Repository Pattern (tùy chọn) có thể được áp dụng để đóng gói các truy vấn phức tạp, giúp tầng Business Logic không cần biết về chi tiết cách dữ liệu được lưu trữ hay truy xuất.



Hình 3.6 Sơ đồ kiến trúc phân lớp Backend

Toàn bộ hệ thống được thiết kế để vận hành trong môi trường Docker và được điều phối bởi Docker Compose. Mô hình triển khai bao gồm 5 dịch vụ chính:

PostgreSQL: Cơ sở dữ liệu chính, lưu trữ bền vững toàn bộ dữ liệu học thuật và người dùng.

Redis: Tầng bộ nhớ đệm tốc độ cao và là trung gian pub/sub cho các sự kiện thời gian thực.

Backend (Node.js/Express): Máy chủ API chính, xử lý logic nghiệp vụ. Có thể nhân bản container này để xử lý tải cao.

Frontend (Next.js): Ứng dụng client, phục vụ giao diện người dùng.

Nginx: Đóng vai trò là Reverse Proxy, cổng vào duy nhất của hệ thống, thực hiện cân bằng tải, SSL termination và hỗ trợ kết nối WebSocket.

Việc sử dụng Docker Compose giúp đơn giản hóa quá trình thiết lập và triển khai toàn bộ hạ tầng chỉ bằng một lệnh duy nhất.

3.3 Thiết kế cơ sở dữ liệu

3.3.1 Mô hình ERD

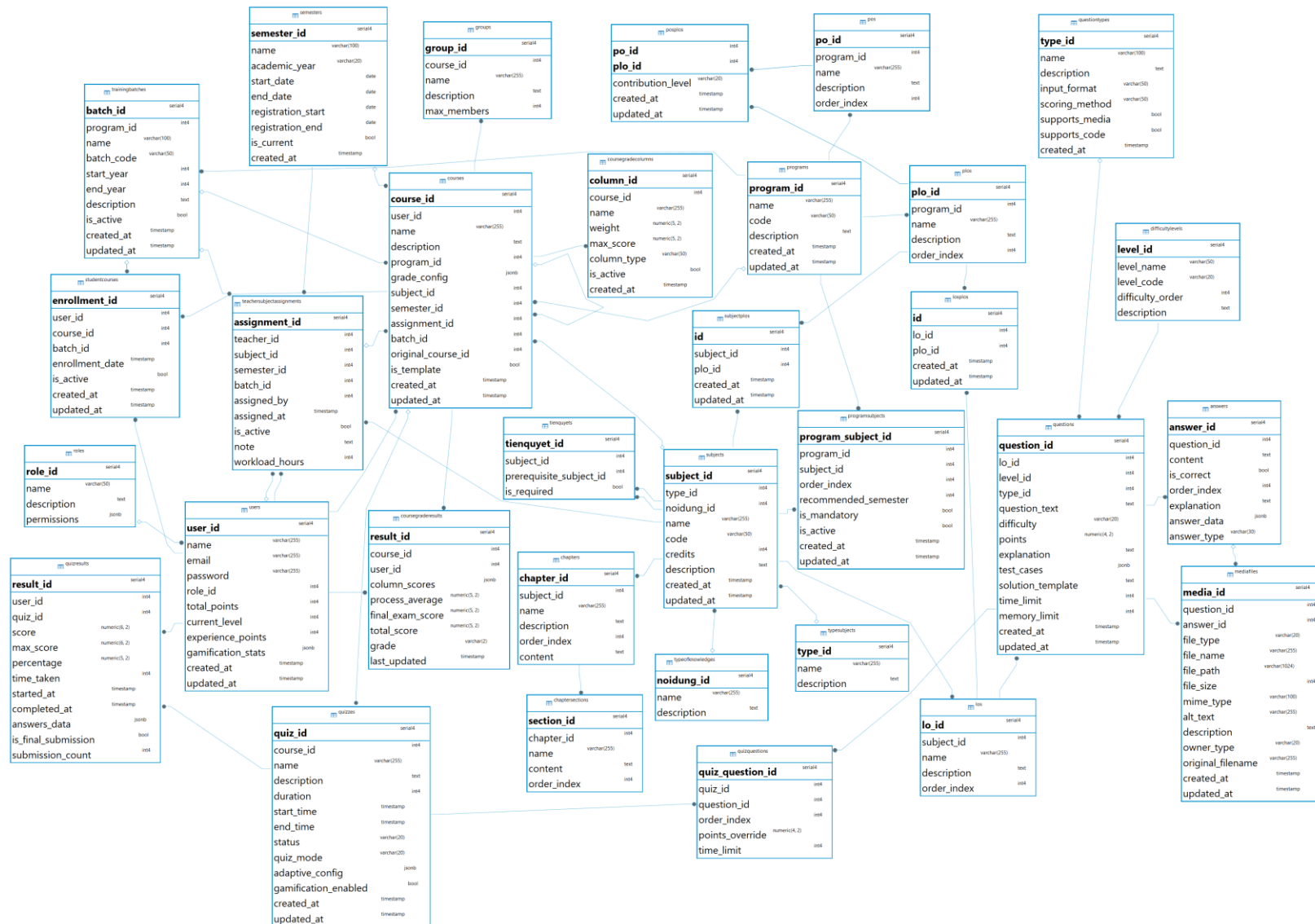
Lược đồ dữ liệu được chuẩn hóa theo dạng chuẩn 3NF để giảm thiểu dư thừa và được tổ chức thành 3 phân hệ logic (modules) gắn kết chặt chẽ: (i) Phân hệ Quản lý các tài nguyên đào tạo; (ii) Phân hệ Quản lý tiến trình học tập; (iii) Phân hệ Quản lý kết quả học tập và được trình bày cụ thể ở mỗi phân hệ. Riêng các thực thể có dữ liệu dùng chung sẽ được trình bày ở bảng 3.1.

Bảng 3.1 Danh sách các thực thể dùng chung

STT	Tên thực thể	Diễn giải	Đặc tả (vai trò dữ liệu)	Ghi chú
1	Users	Người dùng hệ thống	Lưu trữ thông tin định danh, xác thực và trạng thái tài khoản; được tham chiếu xuyên suốt các phân hệ để quản lý truy cập, truy vết hoạt động và tổng hợp dữ liệu học tập.	Dữ liệu dùng chung
2	Roles	Vai trò người dùng	Chuẩn hóa mô hình phân quyền, xác định tập quyền hạn tương ứng với từng loại người dùng. Được liên kết với <i>Users</i> nhằm hỗ trợ kiểm soát truy cập, bảo mật và phân tách nghiệp vụ ở tầng ứng dụng.	Dữ liệu dùng chung

Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi

Mô hình ERD tổng quát:



Hình 3.7 Mô hình ERD tổng quát

3.3.1.1 Quản lý các tài nguyên đào tạo

Phân hệ Quản lý các tài nguyên đào tạo đóng vai trò là nền tảng dữ liệu của hệ thống, chịu trách nhiệm mô hình hóa cấu trúc đào tạo theo định hướng chuẩn đầu ra. Dữ liệu do phân hệ này quản lý là cơ sở để triển khai các chức năng học tập, đánh giá và phân tích ở các phân hệ tiếp theo. Thiết kế dữ liệu áp dụng cách tiếp cận phân tách giữa cấu trúc học thuật và thực thể triển khai. Trong đó, bảng Subjects lưu trữ thông tin khung môn học, còn bảng Courses đại diện cho các lớp học phần được mở theo từng học kỳ (Semesters). Cách tổ chức này cho phép một môn học được triển khai qua nhiều lớp với các cấu hình giảng viên và tổ chức khác nhau, đồng thời vẫn đảm bảo tính nhất quán về mặt học thuật.

Bên cạnh đó, hệ thống hỗ trợ ánh xạ đa chiều giữa các cấp độ chuẩn đầu ra thông qua các bảng liên kết, cho phép biểu diễn mối quan hệ n–n giữa chuẩn chương trình (POs), chuẩn đầu ra chương trình (PLOs) và chuẩn đầu ra môn học (LOs). Thiết kế này tạo điều kiện cho việc truy vết học thuật và phục vụ công tác đảm bảo chất lượng đào tạo. Để đáp ứng sự đa dạng trong cách tổ chức đánh giá, bảng Courses sử dụng trường dữ liệu JSONB nhằm cấu hình linh hoạt các thành phần và trọng số điểm, hạn chế việc thay đổi lược đồ cơ sở dữ liệu khi có sự khác biệt giữa các lớp học phần.

Bảng 3.2 Danh sách thực thể trong phân hệ Quản lý các tài nguyên đào tạo

STT	Tên thực thể	Diễn giải	Đặc tả	Ghi chú
1	Programs	Chương trình đào tạo	Thông tin CTĐT	Dùng chung
2	POs	Chuẩn chương trình (PO)	Danh mục PO thuộc CTĐT	
3	PLOs	Chuẩn đầu ra CTĐT (PLO)	Danh mục PLO thuộc CTĐT	
4	PO_PLOs	Ánh xạ PO–PLO	Bảng liên kết n–n giữa PO và PLO	
5	Subjects	Môn học (khung)	Thông tin môn học (mã, tín chỉ, mô tả), không gắn theo học kỳ	
6	Chapters	Chương/Chủ đề	Phân rã nội dung theo môn học	
7	ChapterSections	Mục/Bài học	Cấp chi tiết hơn của chương; gắn với chapter	

STT	Tên thực thể	Diễn giải	Đặc tả	Ghi chú
8	LOs	Chuẩn đầu ra môn học (LO)	Danh mục LO theo môn học (subject)	
9	LO_PLOs	Ánh xạ LO–PLO	Liên kết LO đóng góp cho PLO (truy vết chuẩn đầu ra)	
10	ProgramSubjects	Môn học trong CTĐT	Liên kết CTĐT–môn học, có thể kèm học kỳ	
11	Semesters	Học kỳ/Năm học	Danh mục học kỳ, mốc thời gian	Dùng chung
12	Courses	Lớp học phần	1 khóa của môn học theo học kỳ	
13	TeacherSubjectAssignments	Phân công giảng dạy	Gán giảng viên phụ trách môn/khóa/học kỳ	Dùng chung

3.3.1.2 Quản lý tiến trình học tập

Phân hệ Quản lý tiến trình học tập là phân hệ có tần suất truy cập cao và mức độ biến động dữ liệu lớn, do trực tiếp xử lý các hoạt động luyện tập và đánh giá quá trình của người học. Vì vậy, thiết kế dữ liệu của phân hệ này được tối ưu theo hướng hiệu năng, khả năng đồng bộ và tính linh hoạt. Hệ thống áp dụng chiến lược lưu trữ lai nhằm cân bằng giữa hiệu năng truy xuất và độ bền dữ liệu. Các dữ liệu mang tính bền vững, chẳng hạn như lịch sử làm bài, kết quả đánh giá, hoặc thông tin thành tựu đã đạt được, được lưu trữ trong cơ sở dữ liệu quan hệ PostgreSQL để đảm bảo tính toàn vẹn và khả năng phục hồi. Trong khi đó, các dữ liệu có mức độ biến động cao và yêu cầu phản hồi tức thời, như điểm số thời gian thực hoặc bảng xếp hạng tạm thời, được đồng bộ với Redis – hệ quản trị dữ liệu trong bộ nhớ. PostgreSQL trong trường hợp này đóng vai trò lưu trữ trạng thái cuối cùng của quá trình học tập.

Để hỗ trợ sự thay đổi linh hoạt của các hoạt động và luật học tập, một số bảng dữ liệu của phân hệ sử dụng thuộc tính kiểu JSONB để lưu trữ các chỉ số mở rộng. Cách tiếp cận này cho phép bổ sung hoặc điều chỉnh các yếu tố theo dõi tiến trình học tập mà không cần thay đổi lược đồ cơ sở dữ liệu, qua đó nâng cao khả năng thích ứng của hệ thống.

Bên cạnh đó, nhằm phục vụ hiệu quả cho các nhu cầu tổng hợp và theo dõi tiến trình học tập, các bảng dữ liệu phục vụ báo cáo được thiết kế theo hướng bảng tổng

hợp. Thay vì tính toán lại từ dữ liệu gốc mỗi khi truy vấn, hệ thống sử dụng các cơ chế kích hoạt tự động để cập nhật số liệu ngay khi có sự kiện phát sinh. Thiết kế này giúp giảm tải truy vấn và duy trì hiệu năng ổn định khi khối lượng dữ liệu tăng lên trên CSDL.

Bảng 3.3 Danh sách thực thể chính trong phân hệ Quản lý tiến trình học tập

STT	Tên thực thể	Diễn giải	Đặc tả (vai trò dữ liệu)	Ghi chú
1	StudentCourses	Ghi danh lớp học phần	Liên kết người học với course; trạng thái enrolled/is_active	
2	Questions	Câu hỏi	Nội dung câu hỏi, metadata (lo_id, level_id, type_id,...)	Dùng chung với Phân hệ 3 (ngân hàng câu hỏi)
3	Answers	Phương án trả lời	Danh sách đáp án cho câu hỏi; đánh dấu đúng/sai; giải thích	Dùng chung với Phân hệ 3
4	MediaFiles	Tệp đính kèm	Media cho câu hỏi (ảnh/tệp), phục vụ hiển thị/khai thác nội dung cho câu hỏi	Dùng chung với Phân hệ 3
5	QuestionTypes	Loại câu hỏi	Danh mục loại câu hỏi	Dùng chung
6	DifficultyLevels	Mức độ khó	Danh mục level, phục vụ phân tầng/thiết kế luyện tập	Dùng chung

3.3.1.3 Quản lý kết quả học tập

Phân hệ Quản lý kết quả học tập tập trung giải quyết một bài toán cốt lõi của hệ thống đào tạo trực tuyến, đó là quản lý hiệu quả một ngân hàng câu hỏi quy mô lớn và tổ chức các hoạt động kiểm tra, đánh giá với yêu cầu hiệu năng cao. Thiết kế dữ liệu của phân hệ này hướng đến khả năng tái sử dụng, mở rộng linh hoạt và hỗ trợ phân tích kết quả học tập ở mức chi tiết. Hệ thống áp dụng nguyên tắc tách biệt giữa ngân hàng câu hỏi và bài kiểm tra, nhằm tăng khả năng tái sử dụng dữ liệu. Các câu hỏi được lưu trữ độc lập trong ngân hàng câu hỏi chung, không gắn cố định với một bài kiểm tra cụ thể. Trong khi đó, bảng Quizzes chỉ đóng vai trò là tập hợp tham chiếu đến các câu hỏi này thông qua các mối quan hệ liên kết. Thiết kế này cho phép giảng viên xây dựng nhiều bài kiểm tra khác nhau từ cùng một tập câu hỏi gốc, đồng thời tránh dư thừa dữ liệu và giảm chi phí bảo trì.

Để hỗ trợ các hình thức đánh giá đa dạng, bảng Quizzes được thiết kế theo hướng đa chế độ thông qua thuộc tính mode. Thuộc tính này xác định hành vi của bài kiểm tra, chẳng hạn như đánh giá chính thức, luyện tập hoặc thi theo thời gian thực. Trên cơ sở đó, hệ thống áp dụng các ràng buộc nghiệp vụ khác nhau tại tầng ứng dụng, trong khi vẫn duy trì một cấu trúc lưu trữ thống nhất ở tầng dữ liệu. Cách tiếp cận này giúp đơn giản hóa thiết kế cơ sở dữ liệu và tăng khả năng mở rộng trong tương lai.

Bên cạnh việc lưu trữ kết quả tổng hợp, hệ thống chú trọng ghi nhận chi tiết quá trình tương tác của người học với từng câu hỏi. Bảng UserQuestionHistories không chỉ lưu trạng thái đúng/sai, mà còn ghi nhận các thông tin như thời gian phản hồi và số lần thử. Dữ liệu chi tiết này tạo thành nguồn dữ liệu thô quan trọng cho các hoạt động phân tích kết quả học tập, đồng thời đặt nền tảng cho các nghiên cứu và ứng dụng phân tích nâng cao trong các giai đoạn phát triển tiếp theo của hệ thống.

Bảng 3.4 Danh sách thực thể chính trong phân hệ Quản lý kết quả học tập

STT	Tên thực thể	Diễn giải	Đặc tả (vai trò dữ liệu)	Ghi chú
1	StudentCourses	Ghi danh lớp học phần	Nền tảng tổng hợp kết quả theo SV–lớp học phần	Dùng chung (liên quan Phân hệ 2)
2	Courses	Lớp học phần	Đơn vị tổng hợp kết quả theo lớp/học kỳ/môn	Dùng chung
3	Subjects	Môn học	Tổng hợp theo môn học/khung chương trình	Dùng chung
4	Semesters	Học kỳ	Tổng hợp theo học kỳ/năm học	Dùng chung
5	TrainingBatches	Khóa/niên khóa	Tổng hợp theo khóa	Dùng chung
6	Programs	CTĐT	Tổng hợp theo CTĐT; báo cáo gắn kiểm định	Dùng chung
7	POs / PLOs / LOs + các bảng mapping	Chuẩn đầu ra và truy vết	Cơ sở để lập báo cáo mức độ đạt chuẩn đầu ra theo nhiều cấp	Dùng chung (đặc biệt quan trọng cho báo cáo kiểm định)

3.3.2 Đặc tả các bảng dữ liệu

Bảng: Roles (Vai trò)

Bảng 3.5 Đặc tả bảng Roles

STT	Tên cột	Mô tả
1	role_id	Khóa chính, mã định danh duy nhất cho vai trò (VD: Admin, Giảng viên, Sinh viên).
2	name	Tên hiển thị của vai trò.

Bảng: Users (Người dùng)

Bảng 3.6 Đặc tả bảng Users

STT	Tên cột	Mô tả
1	user_id	Khóa chính, mã định danh người dùng.
2	name	Họ và tên đầy đủ của người dùng.
3	email	Địa chỉ email (dùng để đăng nhập hoặc liên lạc).
4	password	Mật khẩu đăng nhập (thường được mã hóa).
5	role_id	Khóa ngoại liên kết với bảng Roles, xác định vai trò người dùng.
6	total_points	Tổng điểm tích lũy (cho tính năng gamification).
7	current_level	Cấp độ hiện tại của người dùng.
8	experience_points	Điểm kinh nghiệm hiện tại.
9	gamification_stats	Thống kê chi tiết về gamification (dạng văn bản/JSON).

Bảng: Programs (Chương trình đào tạo)

Bảng 3.7 Đặc tả bảng Programs

STT	Tên cột	Mô tả
1	program_id	Khóa chính, mã chương trình đào tạo.
2	name	Tên chương trình đào tạo (VD: Kỹ thuật phần mềm).
3	description	Mô tả chương trình.

Bảng: Subjects (Môn học - Danh mục)

Bảng 3.8 Đặc tả bảng Subjects

STT	Tên cột	Mô tả
1	subject_id	Khóa chính, mã môn học.
2	type_id	Khóa ngoại liên kết bảng TypeSubjects (Loại môn).

STT	Tên cột	Mô tả
3	noidung_id	Khóa ngoại liên kết bảng TypeOfKnowledges (Khối kiến thức).
4	name	Tên môn học.
5	description	Mô tả môn học.
6	credits	Số tín chỉ

Bảng: TypeSubjects (Loại môn học)

Bảng 3.9 Đặc tả bảng TypeSubjects

STT	Tên cột	Mô tả
1	type_id	Khóa chính, mã loại môn.
2	name	Tên loại môn (VD: Bắt buộc, Tự chọn).
3	description	Mô tả chi tiết.

Bảng: TypeOfKnowledges (Loại khối kiến thức)

Bảng 3.10 Đặc tả bảng TypeOfKnowledges

STT	Tên cột	Mô tả
1	noidung_id	Khóa chính, mã khối kiến thức.
2	name	Tên khối kiến thức (VD: Đại cương, Chuyên ngành).
3	description	Mô tả chi tiết.

Bảng: TienQuyets (Môn tiên quyết)

Bảng 3.11 Đặc tả bảng TienQuyets

STT	Tên cột	Mô tả
1	tienquyet_id	Khóa chính.
2	subject_id	Mã môn học hiện tại.
3	prerequisite_subject_id	Mã môn học cần phải học trước (môn tiên quyết).
4	is_required	Đánh dấu là tiên quyết bắt buộc hay chỉ là khuyến nghị.

Bảng: POs (Program Objectives - Mục tiêu chương trình)

Bảng 3.12 Đặc tả bảng Pos

STT	Tên cột	Mô tả
1	po_id	Khóa chính, mã mục tiêu.
2	program_id	Thuộc chương trình đào tạo nào.
3	name	Tên mục tiêu.
4	description	Mô tả mục tiêu.

Bảng: PLOs (Program Learning Outcomes - Chuẩn đầu ra chương trình)

Bảng 3.13 Đặc tả bảng PLOs

STT	Tên cột	Mô tả
1	plo_id	Khóa chính.
2	program_id	Thuộc chương trình nào.
3	name	Tên chuẩn đầu ra.
4	description	Mô tả chi tiết.

Bảng: POsPLOs (Ma trận liên kết PO và PLO)

Bảng 3.14 Đặc tả bảng POsPLOs

STT	Tên cột	Mô tả
1	po_id	Mã mục tiêu chương trình.
2	plo_id	Mã chuẩn đầu ra chương trình.
3	created_at	Ngày tạo

Bảng: LOs (Learning Outcomes - Chuẩn đầu ra môn học)

Bảng 3.15 Đặc tả bảng Los

STT	Tên cột	Mô tả
1	lo_id	Khóa chính.
2	subject_id	Thuộc môn học nào.
3	name	Tên chuẩn đầu ra môn học.
4	description	Mô tả chi tiết.

Bảng: Semesters (Học kỳ)

Bảng 3.16 Đặc tả bảng Semesters

STT	Tên cột	Mô tả
1	semester_id	Khóa chính, mã học kỳ.
2	name	Tên học kỳ (VD: Học kỳ 1 năm 2024).
3	start_date	Ngày bắt đầu học kỳ.
4	end_date	Ngày kết thúc học kỳ.
5	is_current	Đánh dấu đây là học kỳ hiện tại.
6	academic_year	Năm học (VD: 2023-2024).

Bảng: Courses (Lớp học phần/Khóa học cụ thể)

Bảng 3.17 Đặc tả bảng Courses

STT	Tên cột	Mô tả
1	course_id	Khóa chính.
2	user_id	Giảng viên phụ trách (hoặc người tạo).
3	name	Tên lớp học phần.
4	description	Mô tả lớp học.
5	program_id	Thuộc chương trình nào.
6	grade_config	Cấu hình thang điểm (lưu dạng văn bản/JSON).
7	subject_id	Thuộc môn học nào (trong bảng Subjects).
8	semester_id	Diễn ra trong học kỳ nào.
9	assignment_id	Liên kết đến bảng phân công giảng dạy.
10	original_course_id	ID khóa học gốc (nếu lớp này được sao chép từ lớp khác).
11	is_template	Đánh dấu nếu đây là khóa học mẫu.
12	created_at	Thời gian tạo.
13	updated_at	Thời gian cập nhật.

Bảng: TeacherSubjectAssignments (Phân công giảng dạy)

Bảng 3.18 Đặc tả bảng TeacherSubjectAssignments

STT	Tên cột	Mô tả
1	assignment_id	Khóa chính.
2	teacher_id	Mã giảng viên được phân công.
3	subject_id	Mã môn học.
4	semester_id	Mã học kỳ.
5	assigned_by	Người thực hiện phân công (Admin).
6	assigned_at	Thời gian phân công.
7	is_active	Trạng thái phân công (đang dạy hay đã hủy).
8	note	Ghi chú thêm.
9	workload_hours	Số giờ làm việc/tín chỉ giảng dạy.

Bảng: StudentCourses (Sinh viên đăng ký lớp học)

Bảng 3.19 Đặc tả bảng StudentCourses

STT	Tên cột	Mô tả
1	enrollment_id	Khóa chính.
2	user_id	Mã sinh viên.
3	course_id	Mã lớp học phần tham gia.
4	enrollment_date	Ngày tham gia lớp.
5	created_at	Thời gian tạo bản ghi.
6	updated_at	Thời gian cập nhật.

Bảng: CourseGradeColumns (Cột điểm trong sổ điểm)

Bảng 3.20 Đặc tả bảng CourseGradeColumns

STT	Tên cột	Mô tả
1	column_id	Khóa chính.
2	course_id	Thuộc lớp học phần nào.
3	name	Tên cột điểm (VD: Giữa kỳ, Bài tập 1).
4	weight	Trọng số điểm (VD: 20%).
5	max_score	Điểm tối đa của cột này.
6	column_type	Loại cột (VD: Bài tập, Quiz, Chuyên cần).
7	is_active	Trạng thái kích hoạt.
8	created_at	Thời gian tạo.

Bảng: CourseGradeColumnQuizzes (Liên kết Cột điểm với Quiz)

Bảng 3.21 Đặc tả bảng CourseGradeColumnQuizzes

STT	Tên cột	Mô tả
1	link_id	Khóa chính.
2	column_id	Mã cột điểm.
3	quiz_id	Mã bài kiểm tra (từ module Quiz - không hiển thị chi tiết ở đây).
4	weight	Trọng số riêng của quiz trong cột này (nếu có).
5	created_at	Thời gian tạo.

Bảng: CourseGradeResults (Kết quả điểm chi tiết)

Bảng 3.22 Đặc tả bảng CourseGradeResults

STT	Tên cột	Mô tả
1	result_id	Khóa chính.
2	course_id	Mã lớp học.
3	user_id	Mã sinh viên.
4	column_scores	Lưu chi tiết điểm các cột (dạng văn bản/JSON).
5	process_average	Điểm quá trình trung bình.
6	final_exam_score	Điểm thi cuối kỳ.
7	total_score	Tổng điểm kết thúc môn (số).
8	grade	Điểm chữ (A, B, C...).
9	last_updated	Thời gian cập nhật điểm gần nhất.

Bảng: CourseResults (Kết quả tổng kết khóa học)

Bảng 3.23 Đặc tả bảng CourseResults

STT	Tên cột	Mô tả
1	course_result_id	Khóa chính.
2	user_id	Mã sinh viên.
3	course_id	Mã lớp học.
4	final_grade	Điểm tổng kết (Chữ).
5	total_score	Điểm tổng kết (Số).
6	completion_status	Trạng thái hoàn thành (VD: Đạt, Không đạt).
7	completed_at	Thời gian hoàn thành.
8	created_at	Thời gian tạo bản ghi.

Bảng: CourseGradeResultHistories (Lịch sử sửa điểm)

Bảng 3.24 Đặc tả bảng CourseGradeResultHistories

STT	Tên cột	Mô tả
1	history_id	Khóa chính.
2	result_id	Liên kết tới bảng kết quả điểm.
3	old_scores	Điểm cũ trước khi sửa.
4	new_scores	Điểm mới sau khi sửa.
5	changed_by	Người thực hiện sửa điểm.
6	change_reason	Lý do sửa điểm.
7	changed_at	Thời gian sửa.

Bảng: Chapters (Chương bài học)

Bảng 3.25 Đặc tả bảng Chapters

STT	Tên cột	Mô tả
1	chapter_id	Khóa chính.
2	subject_id	Thuộc môn học nào.
3	name	Tên chương.
4	description	Mô tả chương.
5	order_index	Thứ tự hiển thị.
6	content	Nội dung chính của chương.

Bảng: ChapterSections (Mục nhỏ trong chương)

Bảng 3.26 Đặc tả bảng ChapterSections

STT	Tên cột	Mô tả
1	section_id	Khóa chính.
2	chapter_id	Thuộc chương nào.
3	name	Tên mục.
4	content	Nội dung chi tiết của mục.
5	order_index	Thứ tự hiển thị.

Bảng: chapter_lo (Liên kết Chương và Chuẩn đầu ra môn học)

Bảng 3.27 Đặc tả bảng chapter_lo

STT	Tên cột	Mô tả
1	chapter_id	Mã chương.
2	lo_id	Mã chuẩn đầu ra môn học (để biết chương này đáp ứng chuẩn nào).

Bảng: LearningAnalytics (Phân tích hành vi học tập chi tiết)

Bảng 3.28 Đặc tả bảng LearningAnalytics

STT	Tên cột	Mô tả
1	analytics_id	Khóa chính, mã định danh duy nhất cho mỗi bản ghi phân tích.
2	program_id	Mã chương trình đào tạo liên quan đến đối tượng phân tích.
3	subject_id	Mã môn học (có thể để trống nếu phân tích ở cấp độ chương trình).
4	analysis_type	Loại hình phân tích (Ví dụ: tổng quan chương trình, phân tích môn học, nhóm sinh viên...).

STT	Tên cột	Mô tả
5	time_period	Lưu trữ khoảng thời gian thực hiện phân tích (dưới dạng JSONB).
6	overview_metrics	Các chỉ số đo lường tổng quát như tổng số sinh viên, tỷ lệ hoàn thành, điểm tương tác.
7	outcome_analysis	Dữ liệu phân tích chi tiết về kết quả đạt được của người học.
8	lo_performance	Hiệu suất đạt được theo từng chuẩn đầu ra môn học (Learning Outcomes).
9	difficulty_distribution	Phân bố tỷ lệ câu hỏi và kết quả làm bài theo các mức độ khó (Dễ, Trung bình, Khó).
10	temporal_trends	Dữ liệu về xu hướng thay đổi của các chỉ số học tập theo thời gian.
11	student_segmentation	Phân loại các nhóm sinh viên (sinh viên giỏi, sinh viên có nguy cơ tụt hậu...).
12	correlation_analysis	Phân tích mối tương quan giữa các chỉ số hành vi và kết quả học tập.
13	predictive_insights	Các dự báo về kết quả và đề xuất can thiệp sớm từ hệ thống AI.
14	benchmark_comparisons	So sánh kết quả hiện tại với tiêu chuẩn quốc gia hoặc các kỳ trước.
15	data_quality	Đánh giá độ tin cậy và mức độ hoàn thiện của dữ liệu đầu vào.
16	visualization_config	Cấu hình định dạng xuất dữ liệu và các loại biểu đồ hiển thị trên dashboard.
17	analysis_status	Trạng thái của tiến trình phân tích (đang xử lý, hoàn thành, lỗi...).
18	created_by	Mã định danh của người dùng (giảng viên/admin) thực hiện phân tích.
19	processing_time	Thời gian hệ thống cần để hoàn thành việc xử lý dữ liệu (tính bằng ms).
20	data_snapshot_date	Thời điểm lấy dữ liệu thô (snapshot) để bắt đầu quá trình phân tích.
21	analysis_config	Lưu trữ các tham số cấu hình riêng cho thuật toán phân tích.

Bảng: UserLearningPaths (Lộ trình học tập cá nhân hóa)

Bảng 3.29 Đặc tả bảng UserLearningPaths

STT	Tên cột	Mô tả
1	path_id	Khóa chính.
2	user_id	Mã sinh viên.
3	subject_id	Mã môn học.
4	current_chapter_id	Chương hiện tại đang học.
5	current_lo_id	Chuẩn đầu ra đang cố gắng đạt được.
6	progress_percentage	Phần trăm hoàn thành lộ trình.
7	estimated_completion_date	Ngày dự kiến hoàn thành.
8	learning_pace	Tốc độ học (Nhanh/Chậm/Trung bình).
9	preferred_difficulty	Độ khó ưa thích/phù hợp.
10	path_data	Dữ liệu chi tiết về lộ trình (các bước tiếp theo).
11	created_at	Thời gian tạo.
12	updated_at	Thời gian cập nhật.

Bảng: UserPerformanceStats (Thống kê hiệu suất tổng hợp)

Bảng 3.30 Đặc tả bảng UserPerformanceStats

STT	Tên cột	Mô tả
1	stats_id	Khóa chính.
2	user_id	Mã người dùng.
3	period_type	Loại kỳ (VD: Weekly, Monthly).
4	period_start	Ngày bắt đầu kỳ thống kê.
5	period_end	Ngày kết thúc kỳ thống kê.
6	total_questions_attempted	Tổng số câu hỏi đã làm trong kỳ.
7	total_questions_correct	Tổng số câu đúng.
8	accuracy_rate	Tỷ lệ chính xác.
9	average_response_time	Thời gian phản hồi trung bình.
10	study_time_minutes	Tổng thời gian học (phút).
11	subjects_studied	Số môn đã học.
12	chapters_completed	Số chương hoàn thành.
13	los_mastered	Số chuẩn đầu ra đã thành thạo.
14	improvement_rate	Tỷ lệ cải thiện so với kỳ trước.

STT	Tên cột	Mô tả
15	created_at	Thời gian tạo.

Bảng: ProgramOutcomeTracking (Theo dõi chuẩn đầu ra chương trình - PO)

Bảng 3.31 Đặc tả bảng ProgramOutcomeTracking

STT	Tên cột	Mô tả
1	tracking_id	Khóa chính.
2	user_id	Mã sinh viên.
3	program_id	Mã chương trình đào tạo.
4	po_id	Mã mục tiêu chương trình (PO).
5	achievement_level	Mức độ đạt được (VD: Đạt, Xuất sắc).
6	evidence_count	Số lượng bằng chứng chứng minh (bài tập, đồ án).
7	last_assessment_date	Ngày đánh giá gần nhất.
8	improvement_plan	Kế hoạch cải thiện (nếu chưa đạt).
9	created_at	Thời gian tạo.
10	updated_at	Thời gian cập nhật.

Bảng: UserLOTrackings (Theo dõi chuẩn đầu ra môn học - LO)

Bảng 3.32 Đặc tả bảng UserLOTrackings

STT	Tên cột	Mô tả
1	tracking_id	Khóa chính.
2	user_id	Mã sinh viên.
3	lo_id	Mã chuẩn đầu ra môn học (LO).
4	mastery_level	Mức độ thành thạo (điểm số hoặc phần trăm).
5	practice_count	Số lần luyện tập liên quan đến LO này.
6	correct_count	Số lần trả lời đúng liên quan đến LO này.
7	last_practiced	Lần luyện tập gần nhất.
8	first_attempt	Lần thử đầu tiên.
9	improvement_rate	Tỷ lệ cải thiện.
10	difficulty_preference	Mức độ khó ưa thích cho LO này.
11	created_at	Thời gian tạo.
12	updated_at	Thời gian cập nhật.

Bảng: StudentProgramProgress (Tiến độ chương trình học)

Bảng 3.33 Đặc tả bảng StudentProgramProgress

STT	Tên cột	Mô tả
1	progress_id	Khóa chính.
2	user_id	Mã sinh viên.
3	program_id	Mã chương trình.
4	total_subjects	Tổng số môn phải học.
5	completed_subjects	Số môn đã hoàn thành.
6	in_progress_subjects	Số môn đang học.
7	total_credits	Tổng số tín chỉ chương trình.
8	earned_credits	Số tín chỉ đã tích lũy.
9	overall_gpa	Điểm trung bình tích lũy (GPA).
10	completion_percentage	Phần trăm hoàn thành chương trình.
11	estimated_graduation_date	Dự kiến ngày tốt nghiệp.
12	academic_standing	Xếp loại học lực hiện tại (VD: Giỏi, Khá).
13	last_updated	Thời gian cập nhật gần nhất.

Bảng: SubjectOutcomeAnalysis (Phân tích kết quả môn học)

Bảng 3.34 Đặc tả bảng SubjectOutcomeAnalysis

STT	Tên cột	Mô tả
1	analysis_id	Khóa chính.
2	subject_id	Mã môn học.
3	semester_id	Mã học kỳ.
4	total_students	Tổng số sinh viên.
5	lo_achievement_rates	Tỷ lệ đạt chuẩn đầu ra (Text/JSON).
6	plo_achievement_rates	Tỷ lệ đóng góp vào chuẩn chương trình (Text/JSON).
7	improvement_suggestions	Các đề xuất cải tiến môn học.
8	average_grade	Điểm trung bình của lớp.
9	pass_rate	Tỷ lệ qua môn.
10	excellence_rate	Tỷ lệ đạt loại giỏi/xuất sắc.
11	weak_areas	Các vùng kiến thức sinh viên hay yếu.
12	strong_areas	Các vùng kiến thức sinh viên làm tốt.
13	analysis_date	Ngày thực hiện phân tích.

Bảng: UserQuizTrackings (Lịch sử làm bài Quiz)

Bảng 3.35 Đặc tả bảng UserQuizTrackings

STT	Tên cột	Mô tả
1	tracking_id	Khóa chính.
2	user_id	Mã sinh viên.
3	quiz_id	Mã bài kiểm tra.
4	total_attempts	Tổng số lần đã thử làm bài này.
5	best_score	Điểm cao nhất đạt được.
6	latest_score	Điểm lần làm gần nhất.
7	average_score	Điểm trung bình các lần làm.
8	total_time_spent	Tổng thời gian đã dành cho quiz này.
9	last_attempt	Thời gian lần làm cuối.
10	first_attempt	Thời gian lần làm đầu.
11	created_at	Thời gian tạo bản ghi.

Bảng: UserQuestionHistories (Lịch sử trả lời từng câu hỏi)

Bảng 3.36 Đặc tả bảng UserQuestionHistories

STT	Tên cột	Mô tả
1	history_id	Khóa chính.
2	user_id	Mã sinh viên.
3	question_id	Mã câu hỏi.
4	quiz_id	Mã bài kiểm tra chứa câu hỏi này.
5	lo_id	Mã chuẩn đầu ra của câu hỏi.
6	user_answer	Câu trả lời của người dùng.
7	is_correct	Đúng hay sai.
8	response_time	Thời gian suy nghĩ để trả lời câu này (giây).
9	attempt_number	Lần thử thứ mấy.

Bảng: QuizAnalytics (Phân tích bài kiểm tra)

Bảng 3.37 Đặc tả bảng QuizAnalytics

STT	Tên cột	Mô tả
1	analytics_id	Khóa chính.
2	quiz_id	Mã bài kiểm tra.
3	total_participants	Tổng số người tham gia.
4	average_score	Điểm trung bình chung.

STT	Tên cột	Mô tả
5	highest_score	Điểm cao nhất.
6	lowest_score	Điểm thấp nhất.
7	average_completion_time	Thời gian hoàn thành trung bình.
8	question_statistics	Thống kê chi tiết từng câu (Text/JSON).
9	performance_distribution	Phân phối điểm số (biểu đồ phổ điểm).
10	created_at	Thời gian tạo.
11	updated_at	Thời gian cập nhật.

Bảng: QuizScoringStats (Thống kê độ khó/phân loại câu hỏi)

Bảng 3.38 Đặc tả bảng QuizScoringStats

STT	Tên cột	Mô tả
1	stats_id	Khóa chính.
2	quiz_id	Mã bài kiểm tra.
3	question_id	Mã câu hỏi.
4	total_attempts	Số lần câu hỏi này được trả lời.
5	correct_attempts	Số lần trả lời đúng.
6	incorrect_attempts	Số lần trả lời sai.
7	accuracy_rate	Tỷ lệ trả lời đúng.
8	average_response_time	Thời gian phản hồi trung bình.
9	difficulty_rating	Đánh giá độ khó (tự động tính).
10	discrimination_index	Chỉ số phân loại (độ phân biệt học sinh giỏi/yếu).
11	created_at	Thời gian tạo.

Bảng: PracticeRecommendations (Gợi ý luyện tập)

Bảng 3.39 Đặc tả bảng PracticeRecommendations

STT	Tên cột	Mô tả
1	recommendation_id	Khóa chính.
2	user_id	Mã sinh viên.
3	recommendation_type	Loại gợi ý (VD: Ôn tập, Nâng cao).
4	target_lo_id	Mục tiêu vào chuẩn đầu ra nào.
5	target_subject_id	Mục tiêu vào môn nào.
6	target_chapter_id	Mục tiêu vào chương nào.
7	priority_level	Mức độ ưu tiên.

STT	Tên cột	Mô tả
8	recommendation_content	Nội dung gợi ý (VD: "Làm bài quiz chương 3").
9	reasoning	Lý do gợi ý (VD: "Bạn làm sai câu này nhiều lần").
10	estimated_impact	Dự đoán tác động đến điểm số.
11	estimated_time_minutes	Thời gian dự kiến để hoàn thành.
12	status	Trạng thái (Chưa làm, Đã làm).
13	effectiveness_score	Điểm hiệu quả của gợi ý sau khi thực hiện.
14	created_at	Thời gian tạo.
15	expires_at	Thời gian hết hạn gợi ý.

Bảng: LeaderboardEntries (Bảng xếp hạng)

Bảng 3.40 Đặc tả bảng LeaderboardEntries

STT	Tên cột	Mô tả
1	entry_id	Khóa chính.
2	leaderboard_type	Loại bảng xếp hạng (Tuần, Tháng, Tổng).
3	user_id	Mã người dùng.
4	score	Điểm số xếp hạng.
5	rank_position	Vị trí xếp hạng hiện tại.
6	additional_metrics	Các chỉ số phụ để xét hạng.
7	period_start	Ngày bắt đầu kỳ xếp hạng.
8	period_end	Ngày kết thúc kỳ xếp hạng.
9	created_at	Thời gian tạo.

Bảng: RankingHistory (Lịch sử thay đổi hạng)

Bảng 3.41 Đặc tả bảng RankingHistory

STT	Tên cột	Mô tả
1	history_id	Khóa chính.
2	user_id	Mã người dùng.
3	leaderboard_type	Loại bảng xếp hạng.
4	old_rank	Hạng cũ.
5	old_score	Điểm cũ.
6	new_score	Điểm mới.
7	new_rank	Hạng mới.

STT	Tên cột	Mô tả
8	rank_change	Mức thay đổi (VD: +2, -1).
9	change_date	Ngày thay đổi.

Bảng: LeaderboardAchievements (Thành tựu đạt được)

Bảng 3.42 Đặc tả bảng LeaderboardAchievements

STT	Tên cột	Mô tả
1	achievement_id	Khóa chính.
2	user_id	Mã người dùng.
3	leaderboard_type	Loại bảng xếp hạng.
4	achievement_type	Loại thành tựu (VD: Top 1 tuần).
5	achieved_date	Ngày đạt được.
6	rank_achieved	Hạng đã đạt được.
7	score_achieved	Điểm số tại thời điểm đạt.
8	period_description	Mô tả kỳ (VD: Tuần 40 năm 2024).
9	created_at	Thời gian ghi nhận.

Bảng: SocialInteractions (Tương tác mạng xã hội học tập)

Bảng 3.43 Đặc tả bảng SocialInteractions

STT	Tên cột	Mô tả
1	interaction_id	Khóa chính.
2	user_id	Người thực hiện tương tác.
3	target_user_id	Người nhận tương tác.
4	interaction_type	Loại tương tác (Like, Comment, Challenge).
5	content	Nội dung tương tác.
6	related_quiz_id	Liên quan đến quiz nào (nếu có).
7	related_subject_id	Liên quan đến môn nào (nếu có).
8	status	Trạng thái.
9	created_at	Thời gian tạo.

Bảng: UserSocialStats (Thống kê xã hội của người dùng)

Bảng 3.44 Đặc tả bảng UserSocialStats

STT	Tên cột	Mô tả
1	stats_id	Khóa chính.
2	user_id	Mã người dùng.

STT	Tên cột	Mô tả
3	friends_count	Số lượng bạn bè.
4	study_groups_count	Số nhóm học tập tham gia.
5	messages_sent	Số tin nhắn đã gửi.
6	messages_received	Số tin nhắn đã nhận.
7	collaborative_sessions	Số phiên học nhóm.
8	help_requests_sent	Số yêu cầu trợ giúp đã gửi.
9	help_responses_given	Số lần giúp đỡ người khác.
10	social_score	Điểm số xã hội/uy tín.
11	last_updated	Thời gian cập nhật gần nhất.

Bảng: EmojiUsageHistory (Lịch sử dùng biểu cảm)

Bảng 3.45 Đặc tả bảng EmojiUsageHistory

STT	Tên cột	Mô tả
1	usage_id	Khóa chính.
2	user_id	Mã người dùng.
3	emoji_id	Mã biểu tượng cảm xúc (từ bảng Emojis_REF).
4	usage_context	Ngữ cảnh sử dụng (Chat, Forum, Feedback).
5	related_quiz_id	Liên quan quiz nào.
6	related_question_id	Liên quan câu hỏi nào.
7	used_at	Thời gian sử dụng.

Bảng: Quizzes (Cấu hình Bài kiểm chi tiết)

Bảng 3.46 Đặc tả bảng Quizzes

STT	Tên cột	Mô tả
1	quiz_id	Khóa chính.
2	course_id	Thuộc khóa học nào
3	name	Tên bài kiểm tra
4	duration	Thời gian làm bài
5	start_time	Thời gian bắt đầu
6	end_time	Thời gian kết thúc
7	status	Trạng thái của bài kiểm tra
8	quiz_mode	Trạng thái sử dụng.
9	gamification_enabled	Có bật chế độ trò chơi không
10	pin	Mã pin để tham gia bài kiểm tra

Bảng: Questions (Ngân hàng Câu hỏi)

Bảng 3.47 Đặc tả bảng Questions

STT	Tên cột	Mô tả
1	question_id	Khóa chính.
2	level_id	Thuộc cấp độ nào.
3	lo_id	Thuộc chuẩn đầu ra nào.
4	question_text	Nội dung câu hỏi (đề bài).
5	explanation	Giải thích đáp án chi tiết.
6	difficulty	Độ khó (Dễ, Trung bình, Khó).
7	question_type_id	Loại câu hỏi (Trắc nghiệm, Tự luận, Điền từ...).
8	is_active	Trạng thái sử dụng.

Bảng: Answers (Câu hỏi)

Bảng 3.48 Đặc tả bảng Answers

STT	Tên cột	Mô tả
1	answer_id	Khóa chính.
2	question_id	Thuộc câu hỏi nào.
3	answer_text	Nội dung đáp án (VD: A. 1945).
4	incorrect	Đánh dấu đây là đáp án đúng (True/False).
5	answer_explanation	Giải thích riêng cho lựa chọn này (tại sao sai/đúng).

Bảng: QuizQuestions (Liên kết Bài thi - Câu hỏi)

Bảng 3.49 Đặc tả bảng QuizQuestions

STT	Tên cột	Mô tả
2	quiz_id	Mã bài thi, khóa chính.
3	question_id	Mã câu hỏi, khóa chính.
4	createdAt	Thời gian tạo.
5	updatedAt	Thời gian chỉnh sửa.

Bảng: Currencies (Các loại tiền tệ)

Bảng 3.50 Đặc tả bảng Currencies

STT	Tên cột	Mô tả
1	currency_id	Khóa chính.
2	name	Tên loại tiền (VD: Vàng, Kim cương, Xu học tập).
3	symbol	Ký hiệu (VD: \$, G).

STT	Tên cột	Mô tả
4	description	Mô tả.

Bảng: UserCurrencies (Ví tiền người dùng)

Bảng 3.51 Đặc tả bảng UserCurrencies

STT	Tên cột	Mô tả
1	user_currency_id	Khóa chính.
2	user_id	Mã người dùng.
3	currency_id	Mã loại tiền.
4	balance	Số dư hiện tại.
5	total_earned	Tổng số tiền đã kiếm được từ trước tới nay.
6	total_spent	Tổng số tiền đã tiêu.

Bảng: Transactions (Lịch sử giao dịch)

Bảng 3.52 Đặc tả bảng Transactions

STT	Tên cột	Mô tả
1	transaction_id	Khóa chính.
2	user_id	Người thực hiện.
3	currency_id	Loại tiền giao dịch.
4	amount	Số lượng (Dương là nhận, Âm là tiêu).
5	description	Mô tả giao dịch (VD: Mua gợi ý, Thưởng điểm danh).
6	reference_type	Loại đối tượng liên quan (VD: Mua Skill, Mua Badge).
7	balance_before	Số dư trước giao dịch.
8	balance_after	Số dư sau giao dịch.

Bảng: UserInventories (Kho đồ người dùng)

Bảng 3.53 Đặc tả bảng UserInventories

STT	Tên cột	Mô tả
1	inventory_id	Khóa chính.
2	user_id	Mã người dùng.
3	item_id	Mã vật phẩm.
4	item_type	Loại vật phẩm (Skill, PowerUp, Skin...).
5	quantity	Số lượng đang có.
6	is_equipped	Đang trang bị/sử dụng hay không.

Bảng: Level (Bảng cấu hình cấp độ)

Bảng 3.54 Đặc tả bảng Level

STT	Tên cột	Mô tả
1	level_id	Khóa chính
2	Description	Mô tả

Bảng: UserLevelProgress (Tiến độ cấp độ người dùng)

Bảng 3.55 Đặc tả bảng UserLevelProgress

STT	Tên cột	Mô tả
1	user_id	Mã người dùng.
2	current_level	Cấp hiện tại.
3	current_experience	Số XP hiện có.
4	experience_to_next	Số XP cần để lên cấp tiếp theo.
5	total_levels_gained	Tổng số cấp đã leo được.

Bảng: Badges (Huy hiệu/Thành tích)

Bảng 3.56 Đặc tả bảng Badges

STT	Tên cột	Mô tả
1	badge_id	Khóa chính.
2	name	Tên huy hiệu.
3	description	Điều kiện nhận huy hiệu.
4	icon_path	Hình ảnh huy hiệu.
5	rarity	Độ hiếm (Thường, Hiếm, Huyền thoại).
6	unlock_criteria	Tiêu chí mở khóa (dạng mã lệnh hoặc JSON).

3.4 Thiết kế Backend API

3.4.1 Kiến trúc API tổng thể

Cấu trúc URL thực tế [6]:

```
https://<domain>/api/{resource}[/ {id}][/{subresource}][/{action}]
```

Đặc điểm kỹ thuật:

- **Content-Type:** application/json cho tất cả request/response
- **Xác thực:** Bearer Token (JWT) trong header Authorization

- **Middleware:** authenticateToken và authorize với role-based access control
- **CORS:** Hỗ trợ multiple origins cho development và production

3.4.2 Cấu trúc Response chuẩn

Response thành công:

```
{
  "success": true,
  "data": {
    "totalItems": 150,
    "totalPages": 8,
    "currentPage": 1,
    "users": [...]
  }
}
```

Response lỗi:

```
{
  "success": false,
  "message": "Mô tả lỗi bằng tiếng Việt",
  "error": "Chi tiết lỗi kỹ thuật"
}
```

3.4.3 Chức năng thời gian thực với Socket.IO và Redis

Hệ thống Backend là khả năng vận hành thời gian thực – đặc biệt trong các tính năng quiz racing, gamification leaderboard, notification và các luồng dữ liệu tương tác trực tiếp với sinh viên/giảng viên.

Để hiện thực hóa điều này, hệ thống sử dụng Socket.IO kết nối client-server hai chiều, cho phép truyền dữ liệu trạng thái phòng thi, kết quả làm bài, leaderboard, thông báo điểm số hoặc phần thưởng tới người dùng ngay khi có thay đổi. Tất cả sự

kiện thời gian thực đều được Backend ghi nhận, xử lý phân tán và phân phối lại cho đúng user, nhóm hoặc lớp học liên quan.

Redis đóng vai trò là pub/sub trung gian giữa các container Backend, đồng bộ và phân phối các kênh thời gian thực trên toàn hệ thống. Nhờ đó, kể cả khi Backend mở rộng thành nhiều node, mọi trạng thái thời gian thực vẫn được duy trì chuẩn xác, không bị mất dữ liệu khi kết nối. Redis còn hỗ trợ cache các leaderboard, lịch sử sự kiện hoặc trạng thái session thi, giúp tăng tốc truy xuất và giảm tải cho database.

Các thuật toán phân bổ event, quản lý phòng/ máy khách Socket.IO được tối ưu để hạn chế nghẽn kết nối, tự động detect mất kết nối và khôi phục trạng thái. Nhờ hệ thống thời gian thực này, các chức năng như quiz racing, thi đua, nhận thưởng, bảng xếp hạng... đều vận hành mượt mà kể cả khi số lượng người dùng đồng thời tăng cao.

Sự kiện bài kiểm tra trực tuyến:

Sự kiện phía máy khách

```
socket.emit('join-quiz', { quizId, userId, pin })

socket.emit('submit-answer', { questionId, answer,
timestamp })

socket.emit('leave-quiz', { quizId, userId })
```

Sự kiện phía máy chủ

```
socket.on('quiz-started', { currentQuestion,
participants })

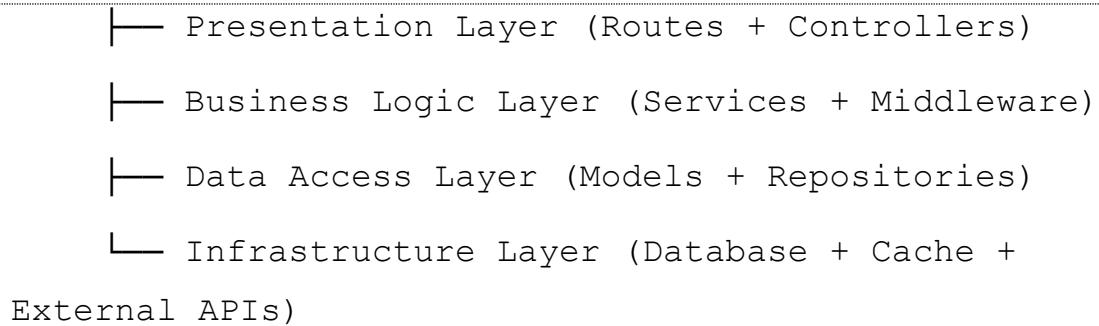
socket.on('question-changed', { question,
timeRemaining })

socket.on('participant-joined', { user, count })

socket.on('leaderboard-updated', { rankings })

socket.on('quiz-ended', { results })
```

3.4.4 Kiến trúc phân lớp



Điểm vào và khởi tạo:

- src/server.js: Khởi động máy chủ HTTP, lập lịch công việc cho quiz expiration
- src/app.js: Thiết lập ứng dụng Express, cấu hình CORS, tích hợp Socket.IO
- Xử lý lỗi toàn cục và middleware ghi log yêu cầu

3.4.5 Trò chơi hóa toàn diện

Yếu tố trò chơi là một thành tố then chốt trong kiến trúc Backend, mang đến động lực học tập hiệu quả và tạo không khí cạnh tranh lành mạnh cho sinh viên. Hệ thống này được xây dựng hoàn toàn phía Backend, vận hành bằng các logic game hóa như tích điểm kinh nghiệm (XP), coin, badge thành tích, cấp độ level, vật phẩm và bảng xếp hạng.

Mỗi hành vi học tập của sinh viên – như hoàn thành bài kiểm tra, đạt chuỗi luyện tập, vượt qua thử thách mới hoặc giúp đỡ các thành viên khác – đều được Backend tự động ghi nhận và tính toán điểm thưởng tương ứng. Thuật toán gamification vận hành liên tục, cập nhật trạng thái: tăng XP, lên level, trao badge, cộng coin và thông báo thành tích theo thời gian thực. Các phần thưởng này không chỉ là dữ liệu mà còn trở thành động lực thúc đẩy sinh viên học tập nghiêm túc, giúp họ “thăng hạng” và ghi điểm cá nhân cũng như nhóm.

Backend sử dụng Redis Sorted Set để quản lý bảng xếp hạng (leaderboard) cho nhiều đối tượng: cá nhân, nhóm, lớp hoặc toàn trường. Khi sinh viên tham gia quiz, trả lời đúng, hoàn thành nhiệm vụ hoặc nhận vật phẩm, số điểm sẽ được update ngay lập tức vào leaderboard thời gian thực, tạo hiệu ứng cạnh tranh trực tiếp và truyền cảm hứng học hỏi cho tất cả người dùng.

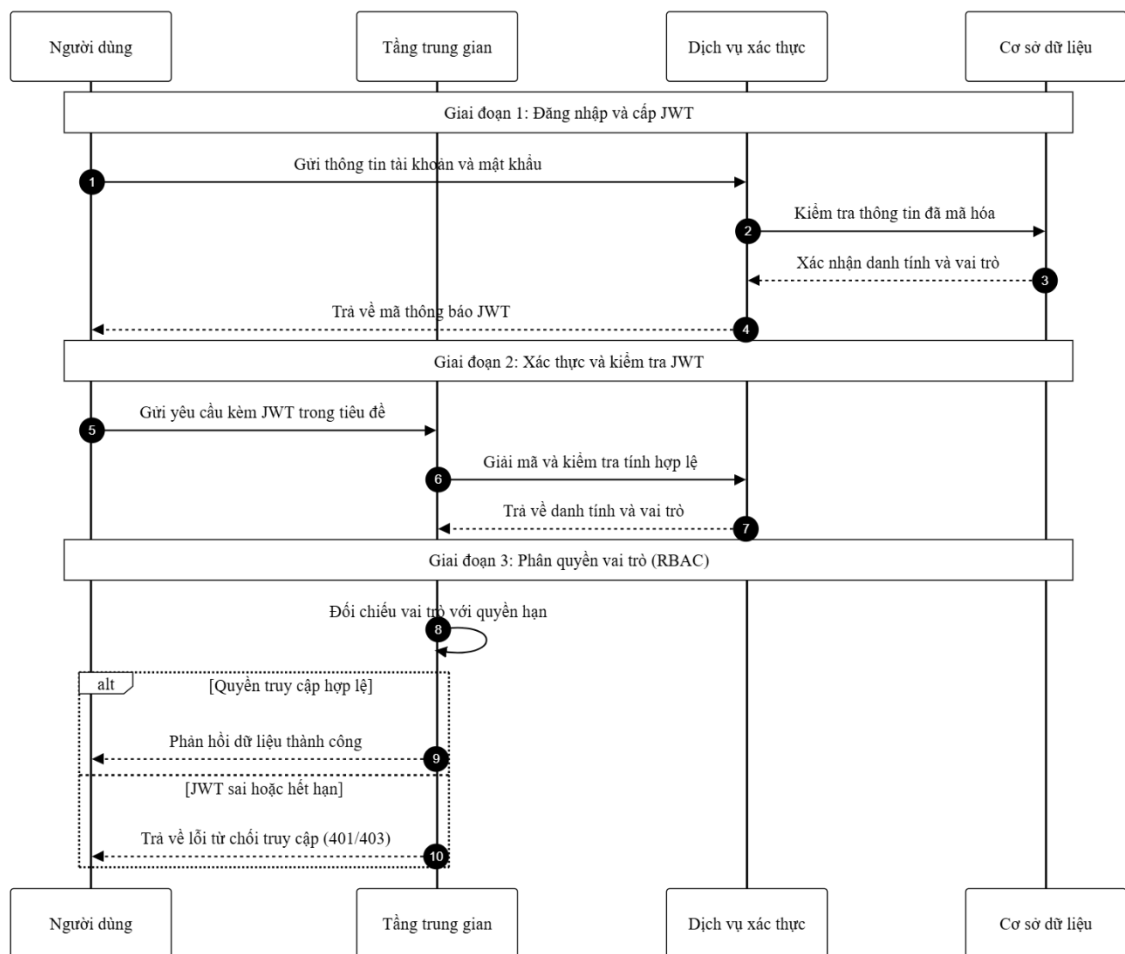
Các vật phẩm như biểu tượng cảm xúc, “trứng” đều được quản trị qua các module Backend độc lập, quy trình trao thưởng – mua bán – sử dụng vật phẩm diễn ra mạch lạc, dễ dàng mở rộng thêm các loại phần thưởng hoặc chế độ trò chơi mới trong tương lai.

Hệ thống cũng hỗ trợ logic đổi thưởng, cập nhật kho đồ, cộng/trừ “tiền” cho từng hành động, đảm bảo dữ liệu luôn đồng bộ và an toàn ở máy chủ.

3.5 Thiết kế các luồng xử lý chính

3.5.1 Cơ chế bảo mật và sơ đồ xử lý xác thực

Để đảm bảo an toàn cho toàn bộ dữ liệu học thuật và thông tin cá nhân hệ thống vận hành cơ chế bảo mật nghiêm ngặt dựa trên JWT và phân quyền theo vai trò. Quy trình này được chia thành các giai đoạn cụ thể nhằm kiểm soát chặt chẽ mọi truy cập từ phía người dùng đảm bảo tính toàn vẹn của dữ liệu.



Hình 3.8 Sơ đồ xử lý xác thực

Giai đoạn cấp phát: Khi người dùng thực hiện đăng nhập hệ thống sẽ kiểm tra tính chính xác của mật khẩu đã được bảo vệ bằng thuật toán băm trong cơ sở dữ liệu.

Nếu thông tin hợp lệ máy chủ sẽ tạo ra một chuỗi JWT chứa các thông tin định danh và vai trò của người dùng đồng thời gửi trả lời về phía máy khách để lưu trữ.

Giai đoạn kiểm tra: Trong mọi lần tương tác tiếp theo máy khách bắt buộc phải gửi kèm chuỗi JWT này trong phần tiêu đề của yêu cầu để hệ thống xác minh. Tầng trung gian của máy chủ sẽ tiếp nhận và tiến hành giải mã JWT để kiểm tra nguồn gốc cũng như thời hạn sử dụng nhằm đảm bảo yêu cầu không bị thay đổi bất hợp pháp.

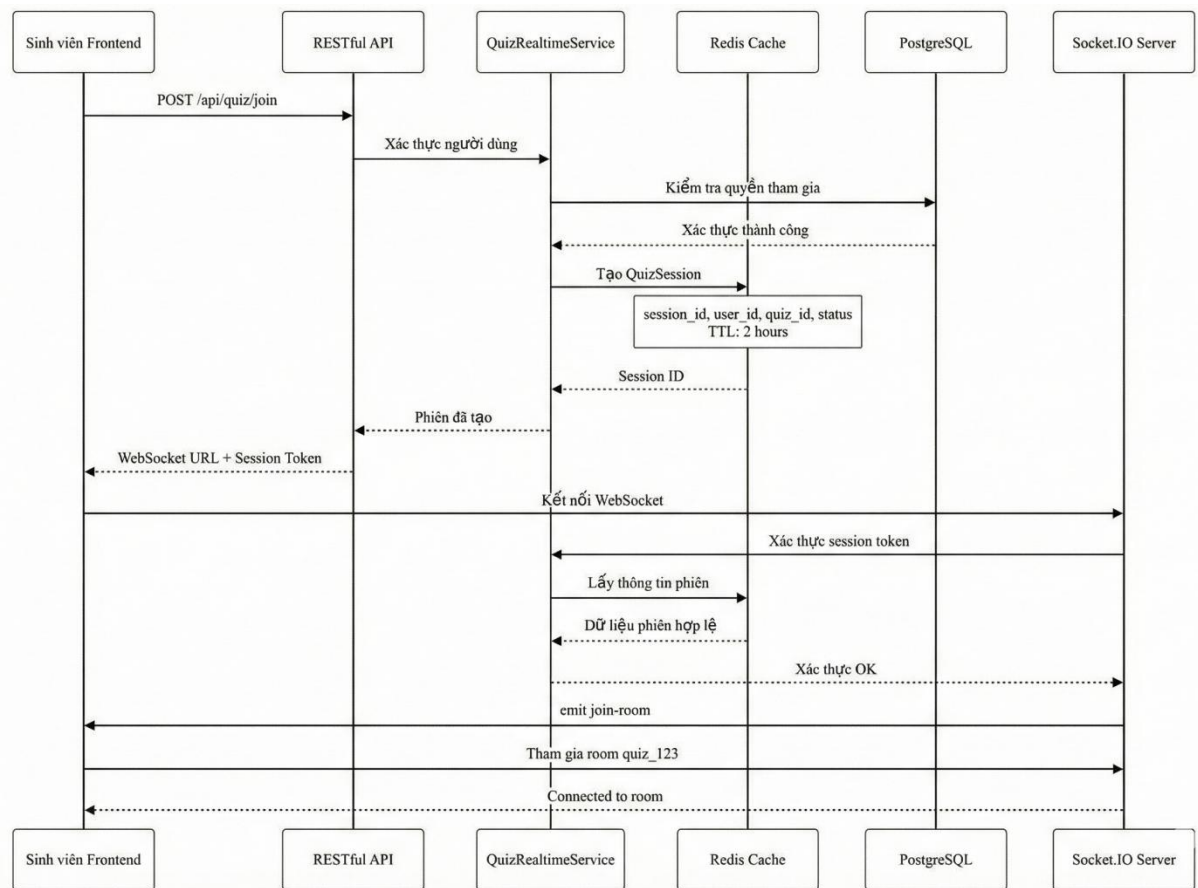
Giai đoạn phân quyền: Sau khi JWT được xác nhận là hợp lệ hệ thống sẽ trích xuất thông tin vai trò để tiến hành phân quyền dựa trên các quy tắc đã thiết lập. Ví dụ như sinh viên chỉ có quyền tham gia bài kiểm tra trong khi giảng viên mới có quyền thiết lập ma trận đề thi hoặc quản lý ngân hàng câu hỏi. Mọi hành vi truy cập sai quyền hạn hoặc sử dụng JWT giả mạo đều sẽ bị hệ thống từ chối và ghi lại thông báo lỗi để đảm bảo tính an toàn cho nền tảng học tập.

3.5.2 Luồng kiểm tra thời gian thực

Luồng hoạt động này là một trong những tính năng cốt lõi, thể hiện sự kết hợp nhuần nhuyễn giữa RESTful API cho các tác vụ khởi tạo và WebSocket (thông qua Socket.IO) cho các tương tác hai chiều trong suốt quá trình làm bài. Sơ đồ tuần tự dưới đây mô tả chi tiết các bước và sự tương tác giữa các thành phần.

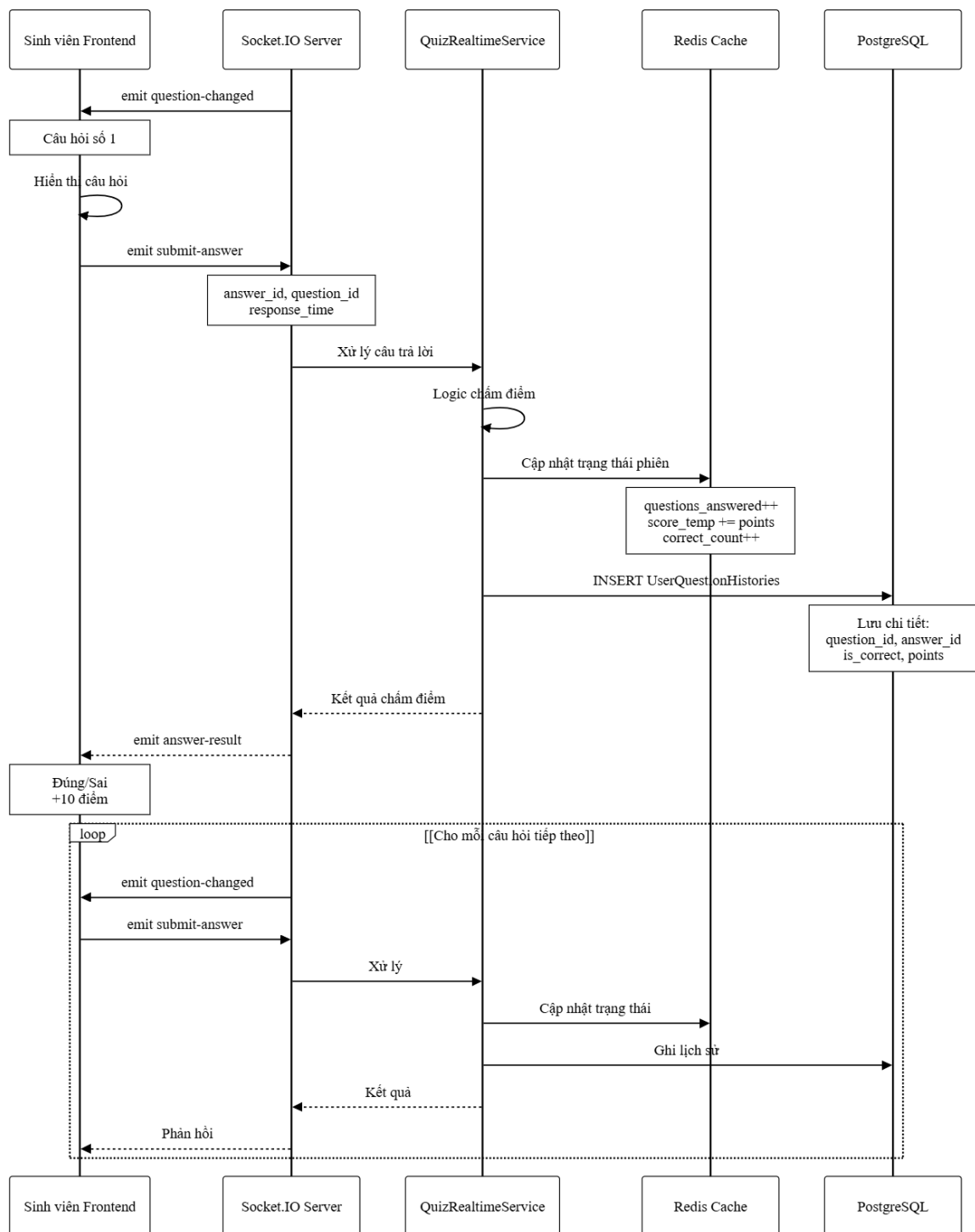
Luồng hoạt động có thể được chia thành ba giai đoạn chính:

Giai đoạn 1 - Tham gia bài kiểm tra: Sinh viên (Frontend User) gửi yêu cầu tham gia vào một phòng thi thông qua một API RESTful. Phía Backend (QuizThời gian thựcService) sẽ xác thực và tạo một phiên làm bài (QuizSession), lưu trạng thái ban đầu vào Redis. Sau khi xác thực thành công, client được hướng dẫn thiết lập một kết nối WebSocket bền vững tới Socket.IO Server và tham gia vào "phòng" (room) tương ứng của quiz đó.



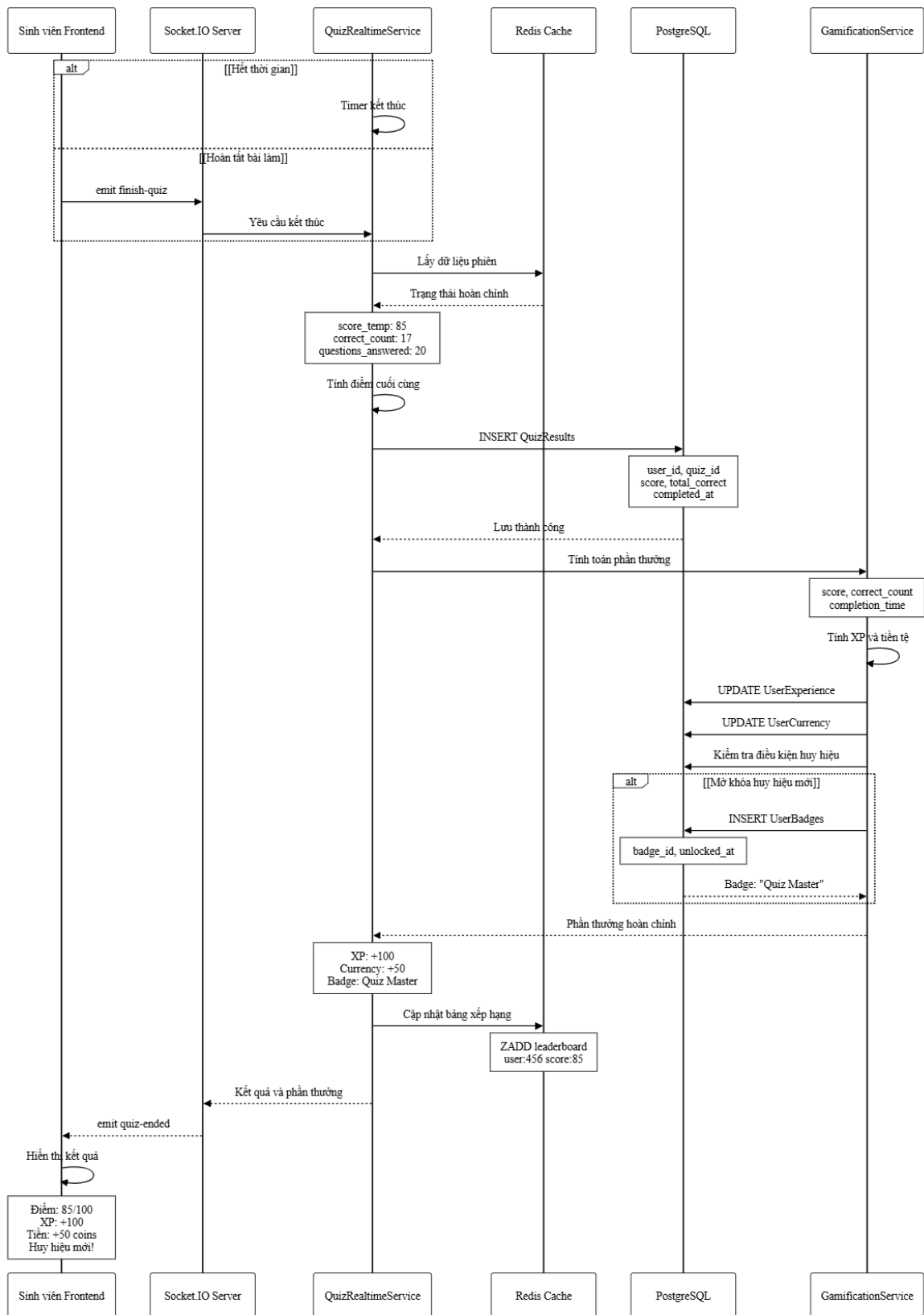
Hình 3.9 Sơ đồ tham gia bài kiểm tra

Giai đoạn 2 - Làm bài và Tương tác: Toàn bộ quá trình làm bài diễn ra qua kết nối WebSocket. Socket.IO Server sẽ đẩy (emit) các sự kiện như question-changed để gửi câu hỏi mới. Khi sinh viên nộp câu trả lời qua sự kiện submit-answer, QuizThời gian thựcService sẽ nhận dữ liệu, xử lý logic chấm điểm, cập nhật trạng thái phiên trong Redis (ví dụ: số câu đã trả lời, điểm tạm thời) và đồng thời ghi lại lịch sử chi tiết vào PostgreSQL (UserQuestionHistories) để đảm bảo dữ liệu được lưu trữ bền vững.



Hình 3.10 Sơ đồ làm bài và tương tác

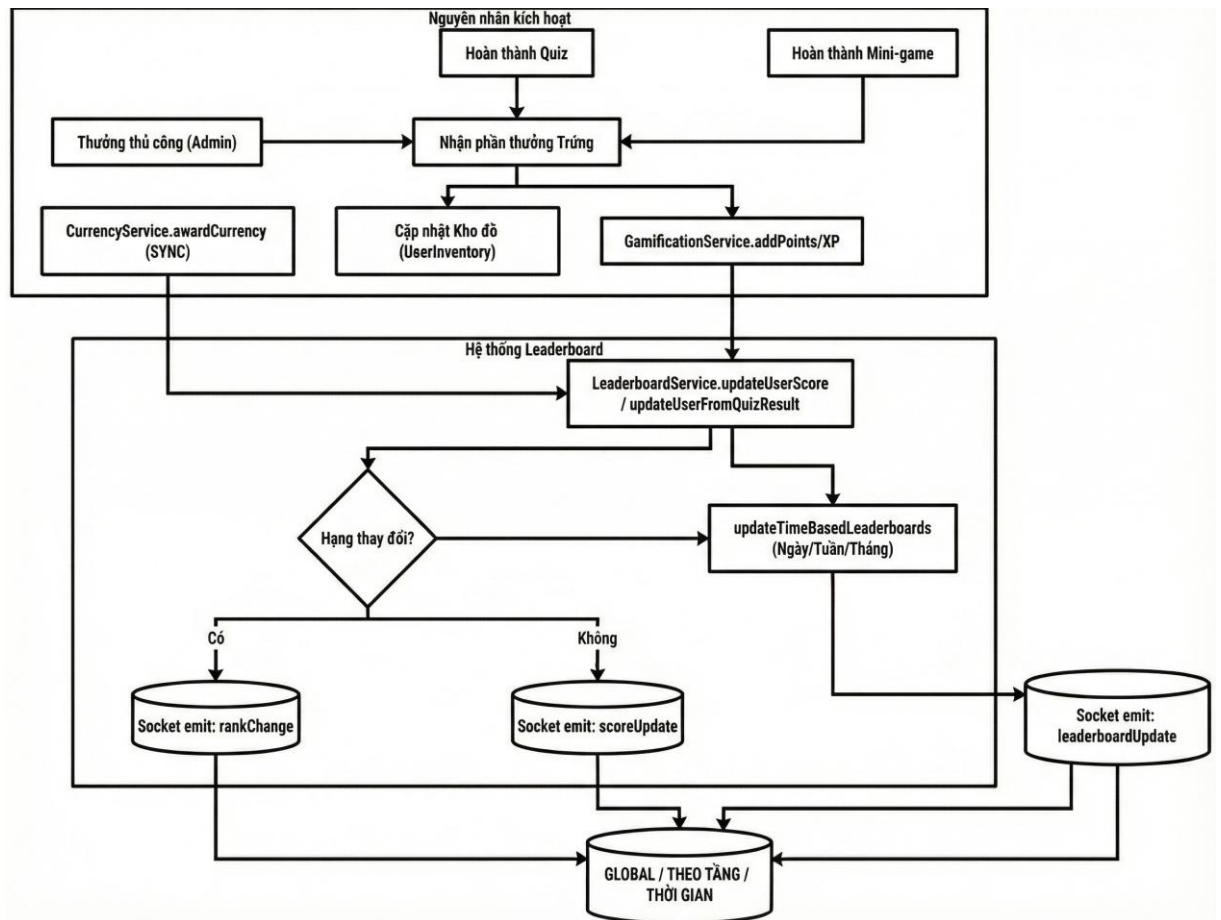
Giai đoạn 3 - Kết thúc và Tổng hợp kết quả: Khi hết thời gian hoặc sinh viên hoàn tất bài làm, QuizThời gian thựcService sẽ thực hiện quá trình tổng kết. Dữ liệu từ Redis được sử dụng để tính toán điểm số cuối cùng, sau đó được lưu vào bảng QuizResults trong PostgreSQL. Ngay sau đó, hệ thống sẽ gọi đến GamificationService để tính toán và trao các phần thưởng tương ứng như điểm kinh nghiệm (XP), tiền tệ và kiểm tra điều kiện mở khóa huy hiệu. Toàn bộ kết quả và phần thưởng sẽ được gửi về cho client thông qua sự kiện quiz-ended.



Hình 3.11 Sơ đồ kết thúc và tổng kết kết quả

Sau khi một hoạt động học tập có tính điểm (như hoàn thành quiz, mini-game) kết thúc, hệ thống sẽ kích hoạt luồng cập nhật điểm và bảng xếp hạng. Luồng này được thiết kế để đảm bảo hiệu năng cao và phản hồi tức thì, sử dụng Redis làm trung tâm cho việc quản lý bảng xếp hạng.

3.5.3 Luồng cập nhật Bảng xếp hạng và Trao thưởng



Hình 3.12 Sơ đồ luồng dữ liệu cập nhật Bảng xếp hạng

Như được minh họa trong Hình 3.7, quá trình cập nhật diễn ra như sau:

Kích hoạt trao thưởng: Các nguyên nhân kích hoạt (ví dụ: hoàn thành Quiz, được Admin thưởng thủ công) sẽ gọi đến các service tương ứng trong Backend. GamificationService sẽ xử lý việc cộng điểm kinh nghiệm (XP), trong khi các service khác có thể cập nhật kho đồ (UserInventory) hoặc trao tiền tệ.

Cập nhật điểm số vào Leaderboard: Mọi sự thay đổi về điểm số sẽ được tổng hợp tại LeaderboardService. Service này thực hiện một tác vụ kép: vừa cập nhật điểm số tổng vào PostgreSQL để lưu trữ lâu dài, vừa sử dụng lệnh ZADD để cập nhật điểm của người dùng vào cấu trúc dữ liệu **Sorted Set** của **Redis**. Việc sử dụng Sorted Set đảm bảo rằng bảng xếp hạng luôn được sắp xếp một cách tự động và có thể truy xuất thứ hạng với tốc độ cực nhanh, độ phức tạp $O(\log(N))$.

Phân phối sự kiện thay đổi: Sau khi cập nhật, LeaderboardService sẽ kiểm tra xem thứ hạng của người dùng có thay đổi hay không.

- Nếu có sự thay đổi thứ hạng, hệ thống sẽ phát sự kiện rankChange qua Socket.IO.
- Nếu chỉ có điểm số thay đổi mà không đổi thứ hạng, sự kiện scoreUpdate sẽ được phát đi.
- Đồng thời, một sự kiện tổng quát leaderboardUpdate cũng được gửi đi để các client khác có thể làm mới toàn bộ bảng xếp hạng.

Luồng hoạt động này không chỉ đảm bảo tính chính xác và đồng bộ của dữ liệu mà còn tạo ra hiệu ứng tức thì, giúp người dùng cảm nhận được sự thay đổi và cạnh tranh một cách tức thì.

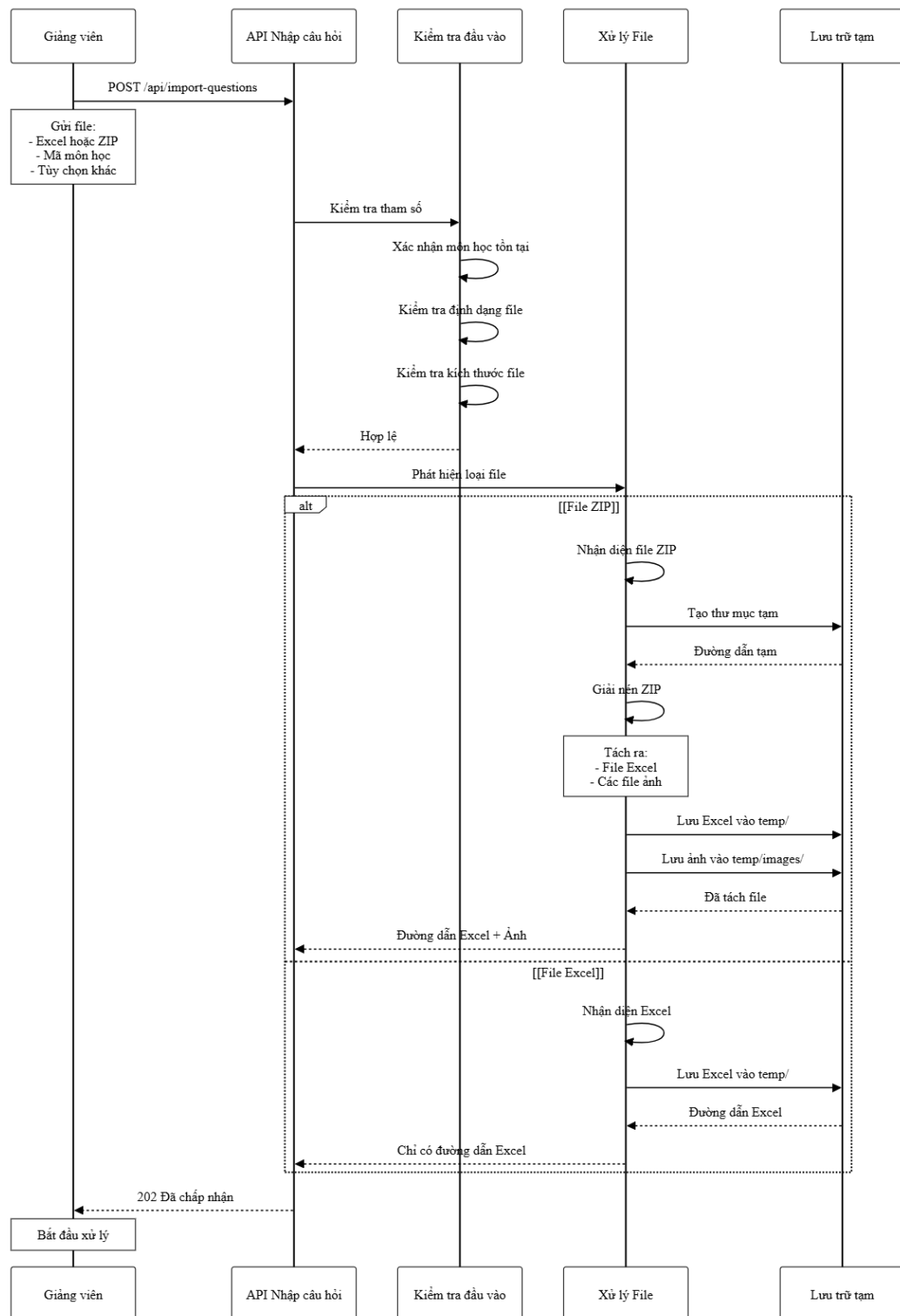
3.5.4 Luồng nhập Câu hỏi hàng loạt file

Để giải quyết bài toán xây dựng ngân hàng câu hỏi một cách hiệu quả cho giảng viên, hệ thống đã phát triển một chức năng nhập câu hỏi hàng loạt hiệu quả và linh hoạt. Chức năng này cho phép giảng viên tải lên một file Excel chứa hàng trăm câu hỏi, hoặc một file ZIP bao gồm cả file Excel và các tệp hình ảnh (hình ảnh, video) đi kèm, chỉ trong một thao tác duy nhất. Luồng xử lý này thể hiện khả năng xử lý file, xác thực dữ liệu phức tạp và đảm bảo tính toàn vẹn dữ liệu thông qua cơ chế giao tác.

Dưới đây là các giai đoạn mô tả chi tiết các bước xử lý của hệ thống khi nhận được yêu cầu nhập.

Giai đoạn 1: Tiếp nhận và Tiền xử lý yêu cầu

Giảng viên gửi một request POST đến endpoint /api/import-questions từ giao diện người dùng. Yêu cầu này chứa file (Excel hoặc ZIP), subject_id và các tham số tùy chọn khác. Backend nhận yêu cầu, kiểm tra tính hợp lệ của các tham số đầu vào. Hệ thống sẽ phát hiện loại gói tin được gửi lên: nếu là file ZIP, nó sẽ tiến hành giải nén để tách file Excel và các tệp hình ảnh vào một thư mục tạm; nếu chỉ là file Excel, hệ thống sẽ bỏ qua bước giải nén.

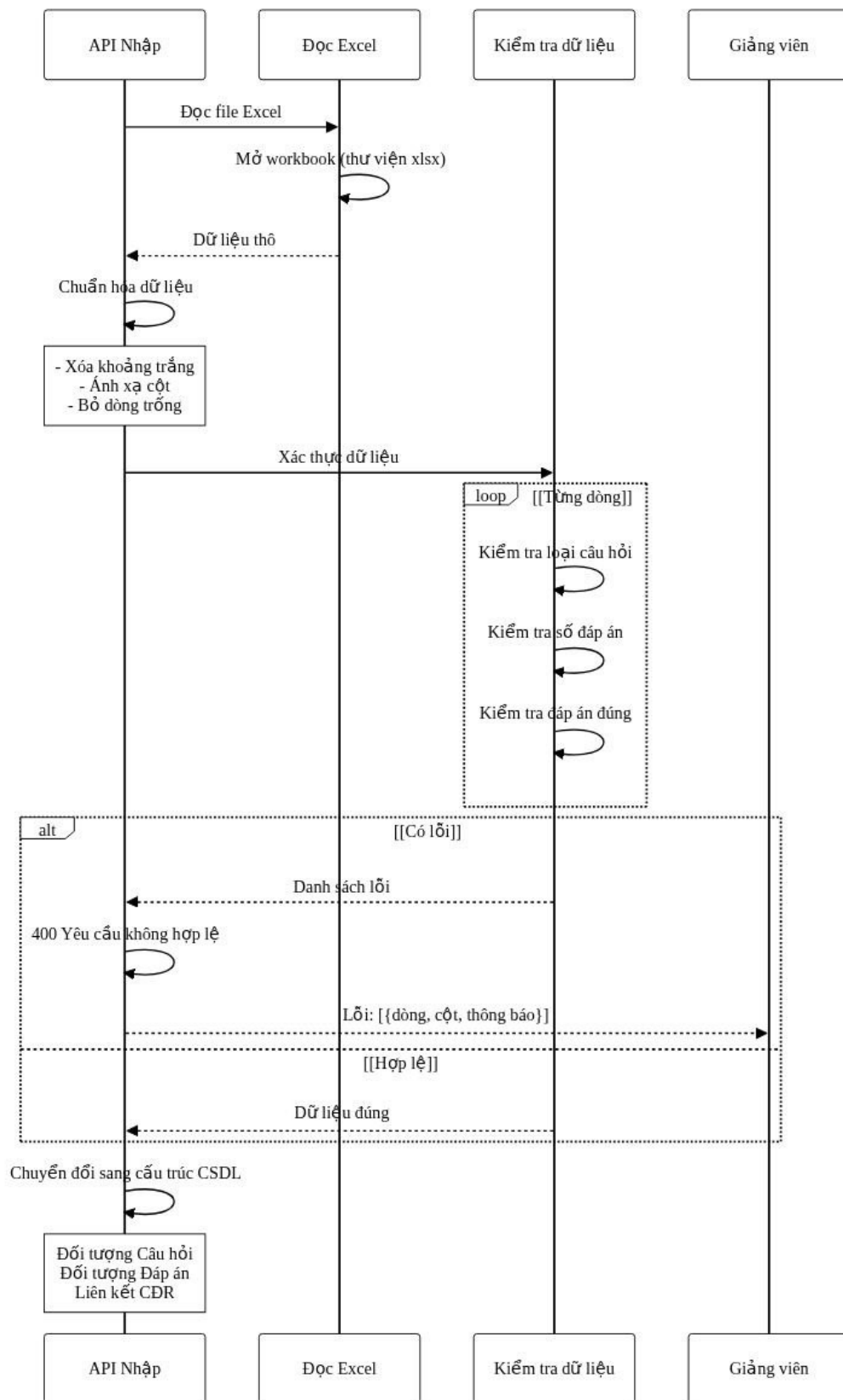


Hình 3.13 Sơ đồ tiếp nhận và tiền xử lý yêu cầu

Giai đoạn 2: Phân tích, Xác thực và Ánh xạ dữ liệu:

Đây là giai đoạn xử lý logic chính. Hệ thống sử dụng các thư viện chuyên dụng (ví dụ: xlsx trong Node.js) để đọc và phân tích nội dung từ file Excel.

Dữ liệu thô sau khi đọc sẽ trải qua các bước chuẩn hóa: loại bỏ khoảng trắng thừa, map các cột trong file Excel với các trường dữ liệu tương ứng của hệ thống và loại bỏ các dòng trống.



Hình 3.14 Sơ đồ phân tích, xác thực và ánh xạ dữ liệu

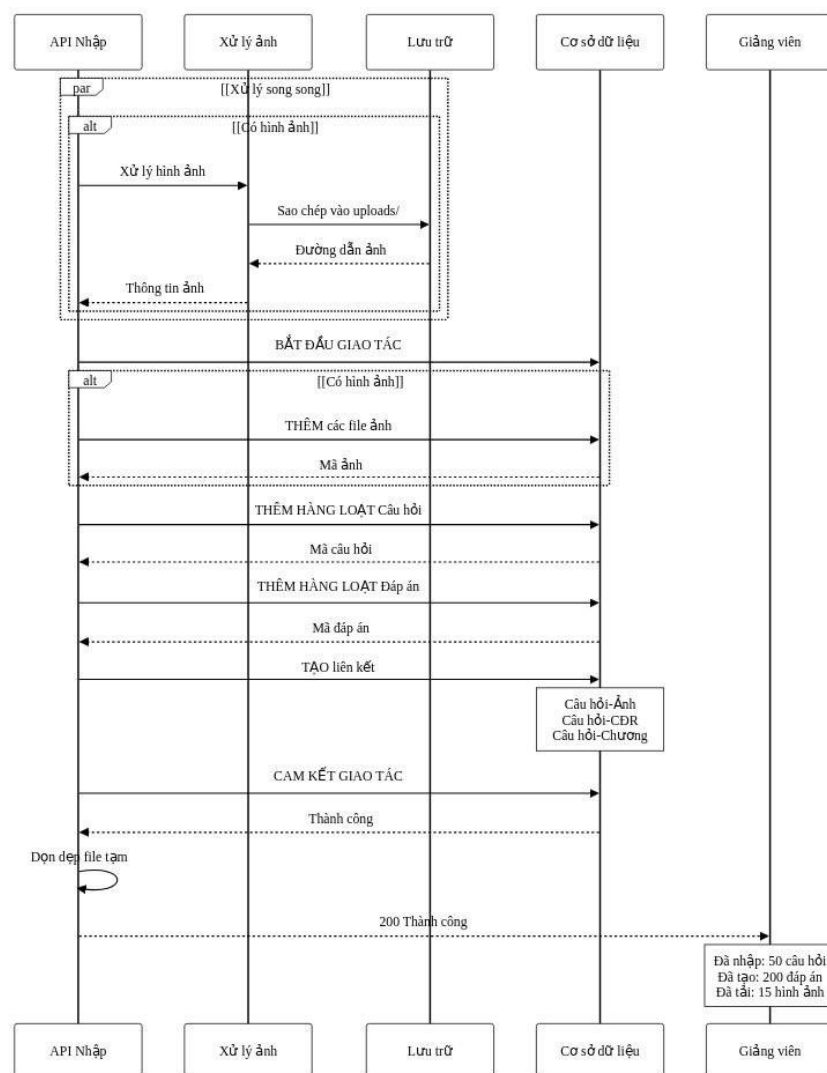
Một bước xác thực (validation) nghiêm ngặt được thực hiện để kiểm tra tính hợp lệ của từng dòng dữ liệu: loại câu hỏi có hợp lệ không, số lượng đáp án có khớp với loại câu hỏi không, đáp án đúng đã được chỉ định hay chưa.

Nếu bất kỳ dòng nào có lỗi, hệ thống sẽ ghi nhận lỗi chi tiết (vị trí dòng/cột, thông điệp lỗi) và ngay lập tức trả về một response 400 Bad Request với file JSON mô tả chính xác các lỗi, giúp giảng viên dễ dàng sửa lại file Excel và thử lại.

Nếu không có lỗi, dữ liệu hợp lệ sẽ được ánh xạ (map) sang schema của các đối tượng trong CSDL (Question, Answers, LO, Hình ảnh).

Giai đoạn 3: Xử lý Hình ảnh và Ghi vào Cơ sở dữ liệu

Song song với việc xử lý Excel, nếu có các tệp hình ảnh, hệ thống sẽ xử lý chúng. Các tệp hình ảnh trong file ZIP sẽ được sao chép vào thư mục lưu trữ (uploads/) của hệ thống.



Hình 3.15 Sơ đồ xử lý hình ảnh và ghi vào cơ sở dữ liệu

Toàn bộ quá trình ghi dữ liệu vào CSDL được bao bọc trong một giao tác duy nhất để đảm bảo tính toàn vẹn.

Hệ thống thực hiện ghi dữ liệu theo thứ tự: tạo các bản ghi Hình ảnhFile (nếu có), sau đó thực hiện ghi hàng loạt (bulk create) cho Questions và Answers. Các liên kết giữa câu hỏi/đáp án và hình ảnh, cũng như liên kết với các thực thể khác như LO (Chuẩn đầu ra) hoặc Chapter (Chương học) cũng được thiết lập trong giao tác này.

Nếu toàn bộ quá trình thành công, giao tác sẽ được commit và hệ thống trả về response 201 Created kèm theo thông báo về số lượng câu hỏi đã được nhập thành công. Nếu có bất kỳ lỗi nào xảy ra trong quá trình ghi CSDL, toàn bộ giao tác sẽ được rollback, đảm bảo không có dữ liệu rác nào được lưu lại.

Luồng xử lý này không chỉ mang lại sự tiện lợi tối đa cho giảng viên mà còn thể hiện một thiết kế Backend vững chắc, có khả năng xử lý các tác vụ phức tạp và đảm bảo tính nhất quán của dữ liệu.

Chương 3 đã trình bày một cách toàn diện quá trình hiện thực hóa hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi. Bắt đầu từ việc lựa chọn kiến trúc tổng thể 3 tầng và mô hình triển khai container hóa, chương đã đi sâu vào việc thiết kế chi tiết kiến trúc phân lớp của Backend và lược đồ cơ sở dữ liệu được phân chia theo từng module chức năng. Các quy chuẩn thiết kế API cũng đã được định nghĩa rõ ràng. Đặc biệt, chương đã mô tả và phân tích chi tiết các luồng xử lý nghiệp vụ phức tạp, qua đó thể hiện sự kết hợp chặt chẽ giữa các công nghệ đã chọn. Những thiết kế và phương pháp đã trình bày trong chương này là nền tảng vững chắc cho các kết quả chức năng sẽ được báo cáo trong Chương 4.

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1 Giới thiệu chung

Chương 4 trình bày các kết quả thực nghiệm đạt được sau quá trình thiết kế và hiện thực hóa hệ thống Backend đã được mô tả chi tiết ở Chương 3. Nội dung của chương sẽ tập trung vào việc mô tả và minh họa các thành phần cốt lõi đã được xây dựng hoàn chỉnh, từ cấu trúc mô hình dữ liệu, hệ thống API cho đến việc triển khai thành công các chức năng nghiệp vụ nổi bật. Thông qua việc trình bày các kết quả này, chương 4 nhằm mục đích chứng minh rằng các mục tiêu và yêu cầu của đề tài đã được đáp ứng một cách đầy đủ và hiệu quả.

Hệ thống đã được triển khai thành công trên môi trường máy chủ thử nghiệm, sử dụng công nghệ ảo hóa Docker để đảm bảo tính nhất quán và dễ dàng nhân rộng. Hạ tầng hệ thống bao gồm các container độc lập cho từng thành phần dịch vụ:

API Server: Chạy trên nền tảng Node.js v18, đóng gói trong container Linux Alpine tối giản.

Cơ sở dữ liệu: PostgreSQL 15 cấu hình tối ưu cho các truy vấn quan hệ phức tạp.

Cache và Message Broker: Redis v7 hoạt động ở chế độ lưu trữ trong bộ nhớ (in-memory) để xử lý các tác vụ thời gian thực.

Web Server: Nginx đóng vai trò Reverse Proxy, quản lý SSL và cân bằng tải cho các yêu cầu đến.

Toàn bộ quy trình khởi tạo hệ thống được tự động hóa thông qua Docker Compose, cho phép triển khai toàn bộ kiến trúc phức tạp chỉ với một câu lệnh duy nhất, giảm thiểu tối đa sai sót trong quá trình cấu hình thủ công.

4.2 Kết quả về kiến trúc và cơ sở dữ liệu

Dựa trên bản thiết kế hệ thống đã trình bày ở Chương 3, một hệ thống cơ sở dữ liệu quan hệ đã được triển khai trên nền tảng PostgreSQL 15. Lược đồ cơ sở dữ liệu được xây dựng theo các nguyên tắc chuẩn hóa, tuân thủ dạng chuẩn 3NF nhằm hạn chế dư thừa dữ liệu và đảm bảo tính nhất quán trong quá trình lưu trữ và khai thác thông tin.

Toàn bộ các bảng dữ liệu được tổ chức thành ba nhóm chức năng chính, tương ứng với các phân hệ nghiệp vụ cốt lõi của nền tảng học tập, bao gồm:

Nhóm Quản lý tài nguyên đào tạo: bao gồm các thực thể liên quan đến chương trình đào tạo, môn học, học kỳ, lớp học phân, chuẩn đầu ra (PO, PLO, LO), người dùng và phân công giảng dạy. Nhóm dữ liệu này đóng vai trò nền tảng, định hình cấu trúc học thuật và làm cơ sở cho các hoạt động học tập và đánh giá.

Nhóm Quản lý tiến trình học tập: tập trung vào các dữ liệu phục vụ cho hoạt động luyện tập và đánh giá quá trình, bao gồm ngân hàng câu hỏi, đáp án, mức độ khó, loại câu hỏi và thông tin ghi nhận sự tham gia của sinh viên trong các lớp học phân. Nhóm này hỗ trợ theo dõi hành vi học tập và cung cấp dữ liệu đầu vào cho việc đánh giá kết quả.

Nhóm Quản lý kết quả học tập và phân tích: bao gồm các thực thể phục vụ tổng hợp, thống kê và phân tích kết quả học tập theo sinh viên, lớp học phân, môn học và chương trình đào tạo. Nhóm dữ liệu này tạo điều kiện cho việc đánh giá mức độ hoàn thành chuẩn đầu ra và hỗ trợ công tác quản lý, kiểm định chất lượng đào tạo.

Cách tổ chức này không chỉ giúp lược đồ cơ sở dữ liệu có cấu trúc rõ ràng, mà còn phản ánh nhất quán các phân hệ nghiệp vụ đã được xác định trong giai đoạn phân tích và thiết kế hệ thống.

Bên cạnh đó, hệ thống đã triển khai các cơ chế tự động hóa tại tầng cơ sở dữ liệu thông qua việc sử dụng trigger và stored procedure. Các cơ chế này được thiết kế để tự động thực hiện các tác vụ nghiệp vụ quan trọng, chẳng hạn như cập nhật điểm tổng hợp khi có thay đổi ở các thành phần điểm, ghi nhận và cập nhật tiến trình học tập của sinh viên, cũng như hỗ trợ phát hiện các trạng thái dữ liệu bất thường. Việc áp dụng tự động hóa ở tầng dữ liệu góp phần giảm thiểu các thao tác thủ công, đồng thời tăng cường tính chính xác, nhất quán và toàn vẹn của dữ liệu trong quá trình vận hành hệ thống.

4.3 Kết quả về Hệ thống API Backend

Kết quả cốt lõi của đề tài là một hệ thống Backend ổn định, cung cấp hơn đầy đủ endpoints RESTful để phục vụ cho toàn bộ các nghiệp vụ của nền tảng. Hệ thống API tuân thủ nghiêm ngặt các tiêu chuẩn thiết kế đã đề ra: cấu trúc URL tường minh,

Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi

sử dụng đúng ngữ nghĩa của các phương thức HTTP và định dạng response JSON đồng nhất. Về mặt bảo mật, tất cả các endpoints đều được bảo vệ bởi cơ chế xác thực JWT (JSON Web Token) và hệ thống phân quyền chi tiết dựa trên vai trò (Role-Based Access Control - RBAC), đảm bảo chỉ những người dùng có thẩm quyền mới có thể truy cập và thao tác trên các tài nguyên tương ứng.

Thay vì liệt kê toàn bộ, phần này sẽ minh họa hoạt động của một số nhóm API tiêu biểu, đại diện cho các tính năng nổi bật nhất của hệ thống.

Nhóm API xác thực và đăng nhập

Đây là nhóm API nền tảng, đóng vai trò là cửa ngõ vào hệ thống và quản lý vòng đời của người dùng.

Endpoint tiêu biểu: POST /api/users/login

Mô tả: Endpoint này cho phép người dùng (sinh viên, giảng viên, admin) đăng nhập vào hệ thống bằng email và mật khẩu. Nếu xác thực thành công, hệ thống sẽ trả về một Access Token (thời hạn ngắn), là nền tảng cho cơ chế xác thực an toàn và liền mạch cho các yêu cầu sau này. Dưới đây là dữ liệu đầu vào (request) và dữ liệu trả về của endpoint:

```
{
  "email": "admin@example.com",
  "password": "admin123"
}
```

```
{
  "token": "eyJhbGciOi..... (đã rút gọn)",
  "user": {
    "user_id": 8,
    "name": "teacher",
    "email": "teacher@example.com",
    "role": "teacher"
  }
}
```

```
}  
  
}
```



Nhóm API Quản lý Câu hỏi và nhập dữ liệu:

Đây là nhóm chức năng nền tảng, cho phép giảng viên xây dựng ngân hàng câu hỏi một cách hiệu quả.

Endpoint nổi bật nhất là `POST /api/questions/import-from-zip`

Mô tả: cho phép nhập hàng loạt câu hỏi kèm theo các tệp hình ảnh (hình ảnh, video) chỉ trong một thao tác duy nhất. Hệ thống sẽ tự động giải nén, xử lý file Excel, liên kết các tệp hình ảnh và tạo mới các bản ghi trong cơ sở dữ liệu.

Cấu trúc body cần thiết để gọi endpoint

	Key		Value		Description	--- Bulk Edit
<input checked="" type="checkbox"/>	zip_file	File	 sample_questions_package.zip		File ZIP chứa Excel và các file ảnh	
<input checked="" type="checkbox"/>	subject_id	Text	10		ID của môn học	

Cấu trúc kết quả trả về:

```
{  
  
  "success": true,  
  
  "message": "Import 4 câu hỏi thành công từ ZIP  
với 3 hình ảnh files",  
  
  "data": {  
  
    "totalImported": 4,  
  
    "totalMediaInZip": 9,  
  
    "totalMediaLinked": 3,  
  
    "linkedMedia": [  
  
      {  
  
        "questionId": 454,  
  
        "answerId": null,  
  
        "MediaFile": "question1.png",  
  
        "type": "question"  
  
      }  
  
    ]  
  
  }  
}
```

```
    },  
    ... // các bản ghi hình ảnh khác tương tự  
  ],  
  "loMapping": {  
    "KQHT1": 28,  
    "KQHT2": 29  
  }  
}  
}
```

Nhóm API Kiểm tra Thời gian thực:

Đây là nhóm chức năng phức tạp, kết hợp cả RESTful API và WebSocket. Để bắt đầu một phiên làm bài, sinh viên gọi API POST `/api/quizzes/{id}/join`. Sau khi xác thực thành công, toàn bộ quá trình làm bài sẽ được điều phối qua các sự kiện Socket.IO như `question-changed` và `submit-answer`. Kết quả thực nghiệm cho thấy hệ thống có thể xử lý các tương tác này với độ trễ thấp, mang lại trải nghiệm làm bài mượt mà và tức thì.

Khi gọi endpoint với body là mã pin của bài kiểm tra do giảng viên cung cấp, sinh viên sẽ tham gia vào bài kiểm tra đó, dưới đây là cấu trúc khi thành công:

```
{  
  "message": "Tham gia bài kiểm tra thành công",  
  "quiz": {  
    "quiz_id": 239,  
    "name": "test",  
    "status": "pending",  
    "current_question_index": 0,  
    "total_questions": 2,  
    "start_time": null,  
  }  
}
```



```
        "end_time": null
    },
    "progress": null,
    "session": {
        "session_id": "quiz_239_122_1764140709510",
        "start_time": "2025-11-26T07:05:09.510Z",
        "end_time": "2025-11-26T07:35:09.510Z",
        "remaining_time": 1799995
    }
}
```

Nhóm API Gamification và Bảng xếp hạng:

Để tạo động lực cho người học, hệ thống cung cấp các API quản lý toàn bộ vòng đời gamification. Một trong những API được sử dụng thường xuyên nhất là GET /api/gamification/leaderboard, cho phép truy xuất bảng xếp hạng theo nhiều tiêu chí (khóa học, toàn trường). Nhờ việc sử dụng Redis Sorted Set để lưu trữ và sắp xếp bảng xếp hạng, thời gian phản hồi của API này luôn được đảm bảo ở mức có độ trễ thấp, ngay cả khi số lượng người dùng tăng cao.

```
{
    "success": true,
    "message": "Lấy bảng xếp hạng thành công",
    "data": {
        "leaderboard": [
            {
                "position": 1,
                "user_id": 122,
                "name": "Thạch Nguyễn Quế Anh",
                "total_points": 4648,
            }
        ]
    }
}
```

```
        "current_level": 33,  
        ... // các chỉ số phụ khác  
    },  
    "avatar_url": "/avatar-animal-  
pack/duck.png"  
    },  
    ... // Danh sách các người dùng tiếp theo  
}
```

Hệ thống cũng đã đáp ứng tốt các yêu cầu phi chức năng quan trọng như Bảo mật, toàn bộ hệ thống API được bảo vệ bởi cơ chế xác thực JWT và phân quyền RBAC. Mật khẩu người dùng được băm an toàn bằng bcrypt. Các biện pháp chống SQL Injection được đảm bảo bởi việc sử dụng Sequelize ORM. Hiệu năng việc áp dụng Redis làm bộ đệm và tối ưu hóa các truy vấn CSDL đã giúp các API quan trọng có thời gian phản hồi nhanh. Quá trình kiểm thử cho thấy hệ thống có thể đáp ứng tốt trong quy mô nhiều người dùng đồng thời. Khả năng mở rộng kiến trúc container hóa với Docker và Docker Compose cho phép hệ thống có thể dễ dàng đóng gói và triển khai, tạo tiền đề để chuyển đổi sang các nền tảng điều phối container như Kubernetes khi cần mở rộng quy mô lớn đáp ứng lưu lượng truy cập tăng cao trong tương lai.

Chương 4 đã trình bày và minh họa một cách chi tiết các kết quả nghiên cứu thực nghiệm của đề tài. Hệ thống Backend đã được xây dựng hoàn chỉnh với một cơ sở dữ liệu có cấu trúc logic gồm nhiều bảng và một hệ thống API phong phú. Các chức năng cốt lõi, từ quản lý học thuật, kiểm tra thời gian thực cho đến gamification, đã được chứng minh là hoạt động ổn định và hiệu quả thông qua các ví dụ minh họa cụ thể. Các kết quả này không chỉ đáp ứng đầy đủ các mục tiêu chức năng và phi chức năng đã đề ra ban đầu mà còn tạo ra một nền tảng vững chắc, sẵn sàng cho việc đưa vào triển khai và phát triển các tính năng nâng cao trong tương lai.

4.4 Đánh giá hiệu năng hệ thống

Để đảm bảo hệ thống không chỉ vận hành đúng về mặt chức năng mà còn đáp ứng được các yêu cầu khắt khe về tốc độ và độ ổn định trong môi trường thực tế, quá trình kiểm thử hiệu năng đã được thực hiện bằng công cụ Postman Performance

Runner. Thay vì kiểm thử dần trải, nhóm nghiên cứu tập trung vào chiến lược kiểm thử cô lập (Isolation Testing) đối với hai "nút thắt" quan trọng nhất của hệ thống: khả năng phản hồi thời gian thực khi sinh viên tham gia thi và khả năng chịu tải khi xử lý dữ liệu nhập liệu phức tạp từ giảng viên.

4.4.1 Kiểm thử khả năng chịu tải với tác vụ thời gian thực

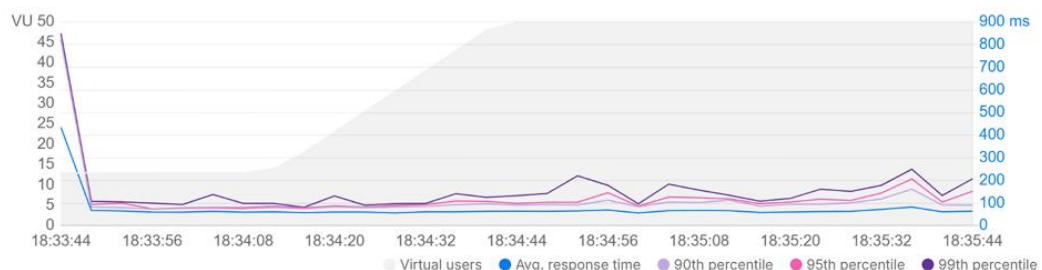
Kịch bản đầu tiên mô phỏng tình huống có mật độ truy cập cao nhất trong thực tế: thời điểm bắt đầu một bài kiểm tra khi hàng loạt sinh viên cùng lúc gửi yêu cầu tham gia phòng thi. Hệ thống được thiết lập để chịu tải từ 50 người dùng ảo truy cập đồng thời vào API tham gia bài kiểm tra (`POST /api/quizzes/{id}/join`) trong khoảng thời gian 2 phút với thời gian tăng tải (ramp-up) là 30 giây.

1. Summary

Total requests sent	Throughput	Average response time	Error rate
3,891	30.77 requests/second	67 ms	0.00 %

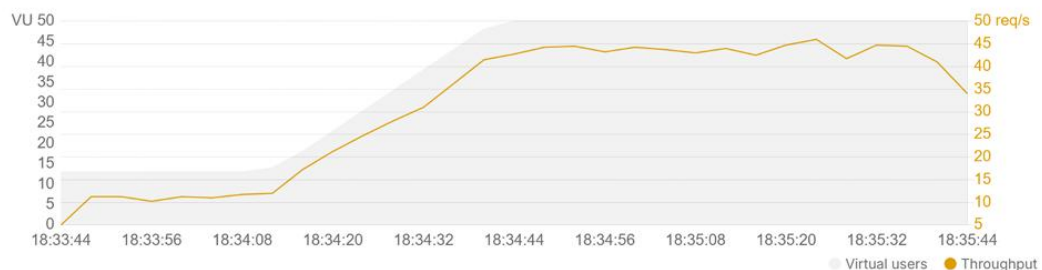
1.1 Response time

Response time trends during the test duration.



1.2 Throughput

Rate of requests sent per second during the test duration.



Hình 4.1 Kết quả kiểm thử khả năng chịu tải thời gian thực

Kết quả thực nghiệm cho thấy kiến trúc sử dụng Node.js kết hợp với Redis Cache đã phát huy hiệu quả vượt trội. Hệ thống đã xử lý thành công tổng cộng 3.891 yêu cầu với tốc độ xử lý trung bình đạt 30,77 yêu cầu/giây. Đáng chú ý, thời gian phản hồi trung bình (Average Response Time) được duy trì ở mức rất thấp là 67 mili-

Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi

giây (ms). Ngay cả đối với nhóm 1% các yêu cầu chậm nhất (99th percentile), độ trễ vẫn nằm trong ngưỡng chấp nhận được là 179 ms. Quan trọng nhất, tỷ lệ lỗi (Error Rate) là 0.00%, khẳng định hệ thống hoàn toàn đảm bảo trải nghiệm mượt mà, không gián đoạn cho lớp học quy mô trung bình đến lớn ngay cả trong những khoảnh khắc cao điểm.

4.4.2 Kiểm thử độ ổn định với tác vụ xử lý dữ liệu nặng

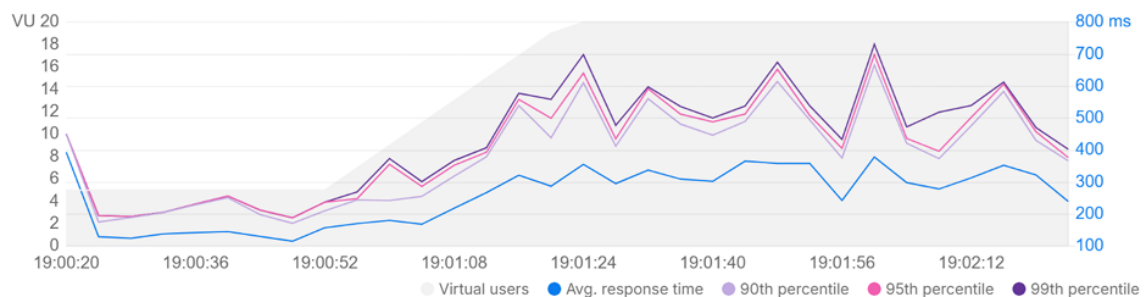
Kịch bản thứ hai tập trung đánh giá sức bền của hệ thống đối với các tác vụ tiêu tốn nhiều tài nguyên CPU và I/O.

1. Summary

Total requests sent	Throughput	Average response time	Error rate
1,315	10.26 requests/second	285 ms	0.00 %

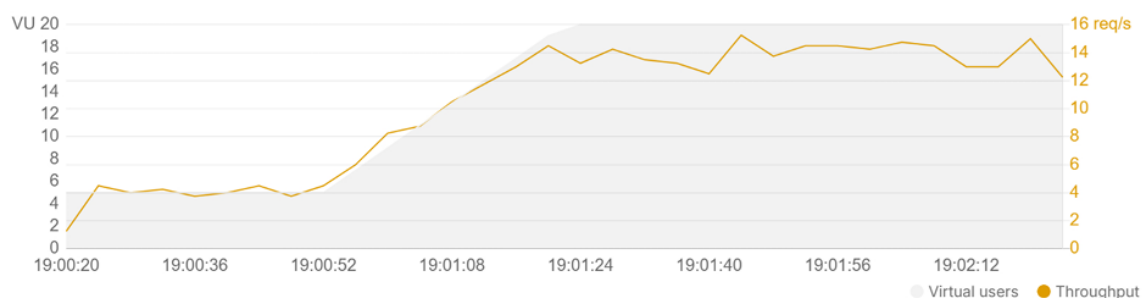
1.1 Response time

Response time trends during the test duration.



1.2 Throughput

Rate of requests sent per second during the test duration.



Hình 4.2 Kết quả kiểm thử ổn định với tác vụ xử lý dữ liệu nặng

Chức năng được lựa chọn kiểm thử là nhập câu hỏi hàng loạt từ file ZIP (POST /api/questions/import-from-zip), bao gồm các quy trình phức tạp gồm giải nén file, đọc dữ liệu Excel, xử lý hình ảnh và thực hiện giao tác cơ sở dữ liệu.

Kịch bản này giả lập 20 giảng viên thực hiện tải file cùng lúc, một cường độ áp lực cao so với nhu cầu sử dụng thông thường tại các trường học.

Dữ liệu thu được cho thấy khả năng xử lý giao dịch mạnh mẽ của Backend. Mặc dù phải thực hiện chuỗi tác vụ nặng, thời gian phản hồi trung bình vẫn đạt con số ấn tượng là 285 ms , với khả năng xử lý khoảng 10,26 yêu cầu/giây. Trong suốt quá trình kiểm thử với 1.315 yêu cầu gửi đi, không có bất kỳ yêu cầu nào bị thất bại, tỷ lệ lỗi duy trì tuyệt đối ở mức 0.00%. Điều này chứng minh cơ chế quản lý giao dịch và xử lý file của hệ thống hoạt động chính xác, đảm bảo tính toàn vẹn dữ liệu ngay cả khi hệ thống đang chịu tải trọng lớn từ nhiều nguồn nhập liệu khác nhau.

CHƯƠNG 5. KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

5.1 Giới thiệu chung

Chương 5 là chương cuối cùng của báo cáo, có nhiệm vụ tổng kết lại toàn bộ quá trình nghiên cứu và xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi. Nội dung của chương sẽ tóm tắt các kết quả chính đã đạt được, từ đó rút ra những kết luận quan trọng về những đóng góp của đề tài cả về mặt học thuật lẫn thực tiễn. Đồng thời, chương cũng sẽ nhìn nhận một cách khách quan những hạn chế còn tồn tại và dựa trên đó đề xuất các hướng phát triển, nâng cấp tiềm năng trong tương lai, nhằm mở ra một tầm nhìn dài hạn cho dự án.

5.2 Kết luận

Sau bốn tuần tập trung nghiên cứu và phát triển, đề tài đã hoàn thành các mục tiêu đã đề ra, triển khai thành công một hệ thống Backend làm nền tảng cho ứng dụng học tập trực tuyến theo định hướng hiện đại. Kết quả nghiên cứu và triển khai của đề tài thể hiện qua ba thành phần kỹ thuật cốt lõi.

Thứ nhất, đề tài đã xây dựng một mô hình cơ sở dữ liệu quan hệ trên nền tảng PostgreSQL, được thiết kế nhằm quản lý hiệu quả các nghiệp vụ đặc thù của giáo dục đại học, bao gồm quản lý chương trình đào tạo theo chuẩn đầu ra, tổ chức ngân hàng câu hỏi và hỗ trợ các cơ chế mở rộng phục vụ tương tác học tập. Lược đồ dữ liệu được tổ chức có cấu trúc, tạo nền tảng cho việc mở rộng và khai thác dữ liệu trong các giai đoạn phát triển tiếp theo.

Thứ hai, một hệ thống API theo kiến trúc RESTful đã được thiết kế và triển khai, cung cấp giao diện lập trình ứng dụng cho các chức năng cốt lõi của hệ thống. Các API được bảo vệ thông qua cơ chế xác thực bằng JSON Web Token (JWT) kết hợp với phân quyền dựa trên vai trò (Role-Based Access Control – RBAC), qua đó đảm bảo yêu cầu về an toàn và toàn vẹn dữ liệu trong quá trình truy cập và vận hành.

Thứ ba, hệ thống đã tích hợp các cơ chế tương tác thời gian thực thông qua việc kết hợp Socket.IO và Redis, cho phép triển khai các chức năng yêu cầu phản hồi tức thì, điển hình là luồng làm bài kiểm tra đồng bộ và cơ chế cập nhật trạng thái theo thời gian thực. Các chức năng này góp phần nâng cao mức độ tương tác và khả năng phản hồi của hệ thống đối với người dùng.

Bên cạnh đó, toàn bộ hệ thống được container hóa bằng Docker, giúp đảm bảo tính nhất quán của môi trường triển khai và hỗ trợ khả năng vận hành linh hoạt trên nhiều nền tảng khác nhau. Từ các kết quả đạt được, đề tài mang lại những đóng góp đáng kể cả về mặt học thuật và thực tiễn.

Về phương diện học thuật, đề tài đề xuất một mô hình kiến trúc tham khảo cho việc thiết kế hệ thống Backend tích hợp quản lý học thuật theo chuẩn đầu ra với các cơ chế tương tác nâng cao. Việc áp dụng cách tiếp cận design-first trong xây dựng API cùng với mô hình phân lớp rõ ràng ở tầng Backend cung cấp những kinh nghiệm có giá trị cho các nghiên cứu và dự án phát triển phần mềm giáo dục tương tự.

Về phương diện thực tiễn, đề tài cung cấp một giải pháp Backend khả thi, có khả năng mở rộng và thích ứng với các yêu cầu đặc thù của môi trường giáo dục đại học. Hệ thống được thiết kế để dễ dàng tích hợp với các ứng dụng phía người dùng (web hoặc thiết bị di động) và có tiềm năng triển khai thử nghiệm tại các khoa hoặc cơ sở đào tạo quy mô vừa, qua đó góp phần cải thiện hiệu quả quản lý và hỗ trợ hoạt động dạy – học.

5.3 Hạn chế của đề tài

Bên cạnh những kết quả đã đạt được, đề tài vẫn tồn tại một số hạn chế nhất định cần được nhìn nhận một cách khách quan nhằm định hướng cho các cải tiến và mở rộng trong tương lai.

Về kiểm thử hệ thống

Mặc dù hệ thống đã được kiểm thử chức năng và kiểm thử tích hợp trong quá trình phát triển, các hoạt động kiểm thử phi chức năng vẫn chưa được triển khai một cách toàn diện. Cụ thể, các kịch bản kiểm thử hiệu năng và kiểm thử tải với số lượng lớn người dùng giả lập chưa được thực hiện. Do đó, khả năng đáp ứng và mức độ ổn định của hệ thống trong điều kiện tải cao hoặc môi trường triển khai thực tế vẫn cần được đánh giá và xác nhận thông qua các thí nghiệm kiểm thử chuyên sâu hơn.

Về kiến trúc hệ thống

Hệ thống hiện tại được thiết kế theo kiến trúc monolithic modular, trong đó các phân hệ nghiệp vụ được tổ chức thành các module logic trong cùng một khối triển khai. Cách tiếp cận này phù hợp với quy mô và nguồn lực của giai đoạn phát triển

ban đầu, giúp đơn giản hóa việc triển khai và kiểm soát hệ thống. Tuy nhiên, trong dài hạn, kiến trúc này có thể gặp hạn chế về khả năng mở rộng và bảo trì độc lập từng module khi quy mô hệ thống và số lượng người dùng tăng lên, đặc biệt khi so sánh với các kiến trúc phân tán như microservices.

5.4 Hướng phát triển trong tương lai

Dựa trên những kết quả đã đạt được và các hạn chế đã được nhận diện, đề tài có thể được tiếp tục mở rộng và nâng cấp theo một số định hướng nghiên cứu và phát triển sau đây.

Nâng cấp kiến trúc hệ thống theo hướng microservices

Trong các giai đoạn phát triển tiếp theo, hệ thống có thể được tái cấu trúc từ kiến trúc monolithic modular sang kiến trúc microservices bằng cách tách các module có nghiệp vụ tương đối độc lập, chẳng hạn như phân hệ trò chơi hóa, phân hệ phân tích dữ liệu và phân hệ quản lý bài kiểm tra, thành các dịch vụ riêng biệt. Cách tiếp cận này cho phép từng dịch vụ được phát triển, triển khai và mở rộng độc lập, đồng thời tăng khả năng chịu tải và tính linh hoạt trong việc lựa chọn công nghệ phù hợp cho từng thành phần.

Phát triển hệ thống phân tích và dự đoán dựa trên trí tuệ nhân tạo

Một hướng mở rộng quan trọng là khai thác dữ liệu học tập đã được thu thập để xây dựng các mô hình học máy (Machine Learning) nhằm phân tích hành vi học tập và dự đoán kết quả học tập của sinh viên. Trên cơ sở đó, hệ thống có thể hỗ trợ đưa ra các cảnh báo sớm cho giảng viên đối với những trường hợp có nguy cơ không đạt yêu cầu học tập. Ngoài ra, các kỹ thuật trí tuệ nhân tạo còn có thể được ứng dụng để cá nhân hóa lộ trình học tập và gợi ý nội dung phù hợp với năng lực và điểm hạn chế của từng người học.

Mở rộng các cơ chế tương tác và yếu tố cộng đồng

Để duy trì mức độ tham gia và tương tác lâu dài của người học, hệ sinh thái trò chơi hóa của hệ thống có thể được mở rộng thông qua việc bổ sung các tính năng mang tính cộng đồng và cạnh tranh cao hơn, chẳng hạn như cơ chế nhóm học tập (guilds), các sự kiện hoặc giải đấu học thuật định kỳ, cũng như hệ thống nhiệm vụ

Xây dựng hệ thống Backend cho nền tảng học tập trực tuyến có tích hợp yếu tố trò chơi theo ngày hoặc theo tuần. Những tính năng này có thể góp phần tăng cường động lực học tập và thúc đẩy sự gắn kết giữa người học trong môi trường trực tuyến.

Tích hợp với các hệ thống và nền tảng bên ngoài

Hệ thống có thể được mở rộng khả năng liên thông thông qua việc xây dựng các cổng API công khai theo chuẩn xác thực hiện đại như OAuth 2.0. Điều này cho phép tích hợp với các hệ thống quản lý sinh viên của các cơ sở đào tạo, cũng như kết nối với các nền tảng cung cấp học liệu trực tuyến khác. Việc tích hợp này giúp hệ thống trở thành một thành phần trong hệ sinh thái số tổng thể của nhà trường, thay vì hoạt động như một ứng dụng độc lập.

Hoàn thiện quy trình DevOps và vận hành hệ thống

Một hướng phát triển quan trọng khác là xây dựng và hoàn thiện quy trình Tích hợp liên tục và Triển khai liên tục (Continuous Integration/Continuous Deployment – CI/CD) bằng các công cụ tự động hóa phù hợp. Song song đó, hệ thống cần được bổ sung các cơ chế giám sát và ghi log tập trung nhằm theo dõi trạng thái vận hành, phát hiện sớm các sự cố và đánh giá hiệu năng trong môi trường sản xuất. Những cải tiến này sẽ góp phần nâng cao độ ổn định và khả năng duy trì lâu dài của hệ thống khi được triển khai ở quy mô thực tế.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Y. Chou and E. von Mechelen, *Actionable gamification: beyond points, badges, and leaderboards*. Fremont, CA: Octalysis Media, 2016.
- [2] M. Casciaro and L. Mammino, *Node.js design patterns: design and implement production-grade Node.js applications using proven patterns and techniques, third edition*, 3rd ed. Place of publication not identified: Packt Publishing, 2020.
- [3] “Express - Node.js web application framework.” Accessed: Nov. 09, 2025.
[Online]. Available: <https://expressjs.com/>
- [4] “Socket.IO.” Accessed: Nov. 09, 2025. [Online]. Available: <https://socket.io/>
- [5] E.-H. El Bachari, M. El Adnani, and S. El Hajjaji, “An architecture for a smart learning management system,” presented at the 2018 International Conference on Information Technology and e-Services (ICITeS), Tangier, Morocco: 2018 International Conference on Information Technology and e-Services (ICITeS), Apr. 2018. doi: 10.1109/ICITeS.2018.8373307.
- [6] B. Jin, S. Sahni, and A. Shevat, *Designing Web APIs: Building APIs That Developers Love*. O’Reilly Media, 2018.