

ĐỒ ÁN TỐT NGHIỆP

**HỆ THỐNG KIỂM TRA ĐẠO VĂN CHO
TRƯỜNG ĐẠI HỌC**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : ThS. Nguyễn Đình Ánh

Sinh viên thực hiện : Trương Trung Nghĩa

MSSV: 2180607790 Lớp: 21DTHD5

TP. Hồ Chí Minh, 2025

ĐỒ ÁN TỐT NGHIỆP

HỆ THỐNG KIỂM TRA ĐẠO VĂN CHO TRƯỜNG ĐẠI HỌC

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : ThS. Nguyễn Đình Ánh

Sinh viên thực hiện : Trương Trung Nghĩa

MSSV: 2180607790 Lớp: 21DTHD5

Giảng viên hướng dẫn	Giảng viên phản biện	Chủ tịch hội đồng

TP. Hồ Chí Minh, 2025

LỜI CẢM ƠN

Đầu tiên, tôi xin phép gửi lời cảm ơn chân thành đến Trường Đại học Công Nghệ thành phố Hồ Chí Minh đã tạo điều kiện cơ sở vật chất và hệ thống tri thức đa dạng để tôi có thể tiếp cận, khai thác và thực hiện quá trình nghiên cứu khoa học. Đặc biệt, tôi xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn – ThS. Nguyễn Đình Ánh đã truyền đạt những kiến thức cũng như những kinh nghiệm quý báu trong xuyên suốt thời gian thực hiện nghiên cứu vừa qua của bản thân tôi.

Trong quá trình thực hiện nghiên cứu và khảo sát, tôi đã gặt hái được những kiến thức bổ ích và những giá trị tri thức quý giá. Những thành quả này sẽ là kinh nghiệm quý báu và là hành trang để tôi phát triển hơn trong tương lai. Bên cạnh đó, do vốn kiến thức thực tế còn nhiều hạn chế và khả năng trình bày còn chưa cao. Vì vậy, dù tôi đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót. Kính mong thầy xem xét và góp ý để bài báo cáo của tôi được hoàn thiện hơn.

Kính chúc thầy gặt hái nhiều thành công trên con đường giảng dạy.

Tôi xin chân thành cảm ơn!

Tp.HCM, ngày tháng năm

Người thực hiện

Trương Trung Nghĩa

LỜI CAM KẾT

Tôi xin cam kết báo cáo thực tập này được hoàn thành dựa trên các kết quả thực tập của tôi và các kết quả nghiên cứu này chưa được dùng cho bất cứ Báo cáo (báo cáo, khóa luận tốt nghiệp) cùng cấp nào khác.

Tp.HCM, ngày tháng năm

Người thực hiện

Trương Trung Nghĩa

MỤC LỤC

TRANG PHỤ BÌA

LỜI CẢM ƠN

LỜI CAM KẾT

MỤC LỤC

DANH MỤC BẢNG

DANH MỤC HÌNH

DANH MỤC TỪ VIẾT TẮT

CHƯƠNG 1: TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU THUỘC LĨNH VỰC ĐỀ TÀI.....1

1.1 Cơ sở nghiên cứu1

1.1.1 Lý do chọn đề tài1

1.1.2 Mục tiêu nghiên cứu2

1.1.3 Câu hỏi nghiên cứu.....2

1.1.4 Phương pháp nghiên cứu3

1.1.5 Đối tượng và phạm vi nghiên cứu5

1.1.6 Cấu trúc báo cáo7

1.2 Tổng quan nghiên cứu7

1.2.1 Các công trình nghiên cứu trong nước7

1.2.2 Các công trình nghiên cứu nước ngoài.....8

1.2.3 Tổng quan một số hệ thống kiểm tra đạo văn phổ biến trên thế giới.....9

1.2.4 Đánh giá ưu – nhược điểm của các hệ thống hiện tại.....10

1.2.5 Khoảng trống nghiên cứu và cơ hội phát triển10

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ MÔ HÌNH NGHIÊN CỨU12

2.1 Các khái niệm liên quan12

2.1.1 Khái niệm đạo văn và kiểm tra đạo văn12

2.1.3 Xử lý ngôn ngữ tự nhiên (NLP)13

2.1.4 Mô hình nhúng văn bản (Embedding Models).....13

2.1.5 Cơ sở dữ liệu và lưu trữ.....13

2.2 Yêu cầu và mục tiêu thiết kế hệ thống16

2.3 Kiến trúc tổng thể của hệ thống.....18

2.4 Mô hình xử lý văn bản.....20

2.4.1 Đọc và lưu trữ tài liệu đầu vào21

2.4.2 Tiền xử lý văn bản (Preprocessing).....21

2.4.3 Chuyển đổi văn bản thành embedding	23
2.4.4 Lưu trữ dữ liệu embedding	23
2.5 Các thuật toán so sánh văn bản.....	24
2.6 Bảo mật và quyền riêng tư dữ liệu người dùng	26
2.7 Giao diện và trải nghiệm người dùng (UX/UI)	27
CHƯƠNG 3: TRIỂN KHAI THỬ NGHIỆM VÀ ĐÁNH GIÁ HIỆU QUẢ.....	29
3.1 Quy trình triển khai hệ thống.....	29
3.2 Xây dựng cơ sở dữ liệu.....	31
3.3 Giao diện chương trình	41
3.4 Thiết lập bộ dữ liệu thử nghiệm	49
3.5 Thử nghiệm và đánh giá độ chính xác của hệ thống	50
3.6 Đánh giá kết quả và cải tiến	52
CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ	54
4.1 Kết luận về kết quả đạt được	54
4.2 Những đóng góp của đề tài.....	54
4.3 Đề xuất hướng nghiên cứu tiếp theo.....	54
TÀI LIỆU THAM KHẢO	56
PHỤ LỤC	58

DANH MỤC BẢNG

Bảng 1.1: So sánh các hệ thống đạo văn nổi tiếng trên thế giới hiện nay	10
Bảng 2.1: Thông tin các tham số quan trọng trong việc cấu hình index và tìm kiếm sử dụng thuật toán HNSW.....	25
Bảng 3.1: Diễn giải các thực thể.....	32
Bảng 3.2: Thực thể assignments.....	35
Bảng 3.3: Thực thể classes	35
Bảng 3.4: Thực thể document_batches	36
Bảng 3.5: Thực thể documents	36
Bảng 3.6: Thực thể enrollments	37
Bảng 3.7: Thực thể failed_jobs.....	37
Bảng 3.8: Thực thể job_batches	38
Bảng 3.9: Thực thể jobs.....	38
Bảng 3.10: Thực thể media.....	39
Bảng 3.11: Thực thể migrations	40
Bảng 3.12: Thực thể model_has_permissions.....	40
Bảng 3.13: Thực thể model_has_roles	40
Bảng 3.14: Thực thể password_reset_tokens	40

DANH MỤC HÌNH

Hình 2.1: Sơ đồ tổng quan kiến trúc hệ thống.....	18
Hình 2.2: Quá trình tiếp nhận và lưu trữ dữ liệu đầu vào.....	21
Hình 2.3: Quá trình xử lý văn bản.....	22
Hình 3.1: Sơ đồ phân cấp 3 lớp tính năng chính	29
Hình 3.2: Sơ đồ công việc thực hiện	30
Hình 3.3: Cơ sở dữ liệu hệ thống.....	34
Hình 3.4: Cơ sở dữ liệu vector embedding.....	35
Hình 3.5: Giao diện trang đăng nhập cho Giảng Viên và Sinh Viên	41
Hình 3.6: Giao diện trang đăng nhập cho Quản Trị Viên	42
Hình 3.7: Giao diện trang cá nhân.....	42
Hình 3.8: Giao diện trang chủ cho Giáo Viên và Sinh Viên	43
Hình 3.9: Giao diện trang chủ cho Quản Trị Viên	43
Hình 3.10: Giao diện trang quản lý phương tiện truyền thông	44
Hình 3.11: Giao diện trang quản lý quyền truy cập	44
Hình 3.12: Giao diện trang quản lý người dùng.....	45
Hình 3.13: Giao diện trang quản lý chuyên ngành.....	45
Hình 3.14: Giao diện trang quản lý lớp học	46
Hình 3.15: Giao diện trang quản lý và theo dõi tài liệu tải lên	46
Hình 3.16: Giao diện trang tải lên và lưu trữ tài liệu	47
Hình 3.17: Giao diện trang kiểm tra đạo văn bằng văn bản.....	47
Hình 3.18: Giao diện trang kiểm tra đạo văn bằng file	48
Hình 3.19: Giao diện trang tổng quan kết quả.....	48
Hình 3.20: Giao diện trang hiển thị kết quả kiểm tra	49
Hình 3.21: Kiểm thử đơn vị và tính năng.....	50

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Tiếng Việt	Tiếng Anh
UX/UI	Thiết kế trải nghiệm người dùng / Giao diện người dùng	User Experience / User Interface
AI	Trí tuệ nhân tạo	Artificial Intelligence
OOP	Lập trình hướng đối tượng	Object-Oriented Programming
UML	Ngôn ngữ mô hình hóa thống nhất	Unified Modeling Language
LMS	Hệ thống quản lý học tập	Learning Management System
API	Giao diện lập trình ứng dụng	Application Programming Interface
HTTP	Giao thức truyền tải siêu văn bản	Hypertext Transfer Protocol
NLP	Xử lý ngôn ngữ tự nhiên	Natural Language Processing
CPU	Bộ vi xử lý	Central Processing Unit
JSON	Dữ liệu đối tượng JavaScript	JavaScript Object Notation
ANN	Tìm kiếm láng giềng gần nhất xấp xỉ	Approximate Nearest Neighbor
SSL	Lớp cổng bảo mật	Secure Sockets Layer

CHƯƠNG 1: TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU THUỘC LĨNH VỰC ĐỀ TÀI

1.1 Cơ sở nghiên cứu

1.1.1 Lý do chọn đề tài

Trong bối cảnh giáo dục đại học ngày càng phát triển, vấn đề đạo văn đã trở thành một thách thức lớn đối với các trường đại học và cao đẳng. Đạo văn không chỉ ảnh hưởng đến chất lượng học tập và nghiên cứu mà còn làm giảm uy tín của các cơ sở giáo dục. Việc phát hiện và ngăn chặn đạo văn một cách hiệu quả là một yêu cầu cấp thiết để đảm bảo tính công bằng và minh bạch trong học thuật.

Hiện nay, việc kiểm tra đạo văn ở các trường đại học chủ yếu được thực hiện thủ công hoặc sử dụng các công cụ thương mại nước ngoài. Kiểm tra thủ công tốn rất nhiều thời gian, công sức của giảng viên và kết quả thường thiếu nhất quán. Các phần mềm thương mại như Turnitin tuy cung cấp kết quả đáng tin cậy nhưng chi phí cao và chưa chắc phù hợp hoàn toàn với ngôn ngữ, tài liệu tiếng Việt. Hơn nữa, việc phụ thuộc vào hệ thống bên ngoài có thể gây khó khăn về mặt bảo mật dữ liệu và khả năng tùy biến theo nhu cầu đặc thù của từng trường.

Hệ thống kiểm tra đạo văn được đề xuất trong dự án này không chỉ giúp các trường đại học quản lý và phát hiện đạo văn một cách tự động và thủ công mà còn cung cấp các công cụ hỗ trợ quản lý lớp học, thống kê báo cáo, và quản lý tài liệu một cách hiệu quả. Điều này sẽ giúp giảng viên và người quản lý tiết kiệm thời gian và công sức trong việc kiểm tra và đánh giá bài làm của sinh viên.

Ngoài ra, hệ thống còn tích hợp các tính năng quản lý người dùng, quản lý quyền truy cập, và quản lý thông báo, giúp tạo ra một môi trường làm việc và học tập hiệu quả hơn. Việc hỗ trợ các định dạng khác nhau như Word, PDF và TXT cũng giúp cho việc phân tích và báo cáo trở nên dễ dàng và thuận tiện hơn.

Với những lý do trên, việc phát triển một hệ thống kiểm tra đạo văn toàn diện và hiệu quả là một nhu cầu cấp thiết và có ý nghĩa quan trọng trong việc nâng cao chất lượng giáo dục đại học. Đề tài này không chỉ mang lại lợi ích thiết thực cho các trường

đại học mà còn góp phần vào việc xây dựng một môi trường học thuật công bằng và minh bạch, từ đó mang lại một môi trường học tập và nghiên cứu chất lượng hơn.

1.1.2 Mục tiêu nghiên cứu

Đề tài hướng tới các mục tiêu chính sau:

- **Xây dựng hệ thống phần mềm kiểm tra đạo văn toàn diện:** Phát triển thành công một hệ thống web cho phép phát hiện đạo văn trong các bài viết học thuật của sinh viên, tích hợp đầy đủ các chức năng tự động và hỗ trợ người dùng (giảng viên, sinh viên) một cách thuận tiện.
- **Hỗ trợ đa trường và đa ngôn ngữ:** Thiết kế hệ thống có khả năng phục vụ nhiều trường đại học khác nhau trên cùng một nền tảng. Hệ thống hỗ trợ kiểm tra văn bản tiếng Việt và tiếng Anh, qua đó phát hiện cả trường hợp đạo văn dịch (sao chép ý tưởng từ tài liệu tiếng Anh rồi dịch sang tiếng Việt hoặc ngược lại).
- **Độ chính xác và hiệu quả cao:** Ứng dụng thuật toán và mô hình tiên tiến nhằm nâng cao độ chính xác trong việc phát hiện đạo văn, giảm thiểu bỏ sót (false negative) cũng như hạn chế việc đánh dấu nhầm (false positive). Đồng thời, hệ thống phải xử lý trong thời gian hợp lý, có thể kiểm tra nhanh ngay cả khi cơ sở dữ liệu chứa hàng chục nghìn tài liệu.
- **Chức năng báo cáo trực quan:** Cung cấp kết quả kiểm tra đạo văn dưới dạng trực quan, dễ hiểu, chẳng hạn như làm nổi bật phần văn bản trùng lặp ngay trên file bài viết gốc (Word/PDF), kèm theo tỷ lệ % nội dung trùng lặp, nguồn gốc của đoạn trùng lặp, v.v. Ngoài ra, tạo các báo cáo thống kê tổng hợp cho người quản trị.
- **Phân quyền và quản lý hệ thống hiệu quả:** Đảm bảo hệ thống có cơ chế phân quyền rõ ràng cho các vai trò: quản trị hệ thống, giảng viên, sinh viên. Mỗi vai trò có những chức năng và quyền hạn phù hợp. Hệ thống cũng cần cung cấp các công cụ quản lý dữ liệu (cơ sở dữ liệu tài liệu, thông tin người dùng, lớp học, khóa học) và cơ chế thông báo, phản hồi để vận hành trơn tru trong môi trường đa người dùng.

1.1.3 Câu hỏi nghiên cứu

Trong quá trình phát triển và triển khai hệ thống kiểm tra đạo văn toàn diện cho trường đại học, các câu hỏi nghiên cứu chủ đạo được đề ra nhằm định hướng phương pháp luận, đánh giá hiệu quả và phát triển giải pháp tối ưu. Các câu hỏi chính bao gồm:

1. Làm thế nào để xây dựng một hệ thống kiểm tra đạo văn phù hợp với đặc thù ngôn ngữ và văn hóa học thuật tại Việt Nam, đồng thời đảm bảo độ chính xác và hiệu quả cao?
2. Mô hình embedding đa ngôn ngữ MiniLM có khả năng phát hiện các dạng đạo văn phức tạp như paraphrase và dịch thuật trong tiếng Việt và tiếng Anh như thế nào?
3. Làm sao để kết hợp hiệu quả cơ sở dữ liệu vector embedding (Milvus) với hệ thống quản lý dữ liệu truyền thống nhằm tăng tốc độ và độ chính xác trong việc so sánh và phát hiện đạo văn?
4. Các thuật toán tìm kiếm vector gần nhất, đặc biệt là HNSW, ảnh hưởng ra sao đến hiệu suất và khả năng mở rộng của hệ thống kiểm tra đạo văn khi xử lý khối lượng dữ liệu lớn?
5. Các phương pháp tự động hóa quy trình kiểm tra đạo văn, bao gồm kiểm tra tự động và thủ công, có thể được tích hợp hiệu quả thế nào để nâng cao trải nghiệm người dùng và giảm thiểu sai sót?
6. Làm thế nào để thiết kế hệ thống giao diện và trải nghiệm người dùng (UX/UI) đáp ứng đa dạng đối tượng người dùng (sinh viên, giảng viên, quản trị viên) trong môi trường giáo dục đại học?
7. Những thách thức và hạn chế trong việc triển khai mô hình kiểm tra đạo văn nội địa dựa trên công nghệ AI hiện đại, và các hướng khắc phục tiềm năng là gì?

1.1.4 Phương pháp nghiên cứu

Đề tài kết hợp nhiều phương pháp nghiên cứu và triển khai nhằm đảm bảo tính khoa học và tính thực tiễn của kết quả:

- **Phương pháp phân tích hệ thống:** Áp dụng phương pháp phân tích hướng đối tượng (OOP) trong việc xác định yêu cầu và thiết kế hệ thống. Sử dụng UML để mô tả các mô hình use case, biểu đồ lớp, biểu đồ hoạt động cho các chức năng

chính, giúp hình dung rõ cấu trúc và hành vi của hệ thống trước khi tiến hành xây dựng.

- **Phương pháp nghiên cứu tài liệu, lý thuyết:** Tìm hiểu các thuật toán, mô hình liên quan thông qua sách, bài báo khoa học, tài liệu trực tuyến về phát hiện đạo văn. Đặc biệt tập trung vào các công trình nghiên cứu sử dụng kỹ thuật học máy cho so sánh văn bản, các mô hình embedding ngôn ngữ và hệ quản trị cơ sở dữ liệu chuyên biệt cho tìm kiếm vector.
- **Phương pháp thực nghiệm, so sánh:** Thiết lập các thực nghiệm để đánh giá hệ thống. Ví dụ, so sánh kết quả phát hiện đạo văn của hệ thống đề xuất với kết quả kiểm tra thủ công hoặc một công cụ phổ biến (nếu có điều kiện) trên cùng một tập dữ liệu để định lượng độ chính xác (precision, recall). So sánh thời gian xử lý của hệ thống khi sử dụng mô hình embedding so với khi sử dụng phương pháp truyền thống (chẳng hạn dựa trên từ khóa) trên cùng một bộ dữ liệu, nhằm đánh giá hiệu quả của việc ứng dụng công nghệ mới.
- **Phương pháp thống kê:** Thu thập kết quả thử nghiệm và dùng phương pháp thống kê mô tả để rút ra nhận xét. Chẳng hạn, thống kê tỷ lệ văn bản bị đánh dấu đạo văn, độ dài trung bình của đoạn văn trùng lặp, v.v. để hiểu rõ hơn đặc điểm của tập vi phạm.
- **Phương pháp đánh giá hiệu suất:** Sử dụng các công cụ đo đạc (profiling) để ghi nhận thời gian tính toán cho các thành phần (thời gian tạo vector bằng mô hình MiniLM cho mỗi đoạn văn, thời gian truy vấn Milvus cho mỗi lần kiểm tra, thời gian tạo báo cáo đánh dấu trên file PDF/Word, v.v.). Từ các số liệu này, đánh giá khả năng mở rộng: ví dụ, dự đoán thời gian kiểm tra khi số lượng tài liệu trong cơ sở dữ liệu tăng gấp đôi, gấp ba.
- **Phương pháp chuyên gia và phản hồi người dùng:** Thu thập ý kiến đánh giá từ giảng viên và sinh viên (những người dùng thử hệ thống) về tính hữu ích, tính thân thiện của giao diện, cũng như đóng góp về các trường hợp hệ thống bỏ sót hoặc nhầm lẫn. Những ý kiến này giúp định hướng điều chỉnh ngưỡng thuật toán và cải thiện trải nghiệm người dùng.

Tất cả những phương pháp trên được kết hợp một cách phù hợp trong quá trình triển khai đề tài. Các kết quả thu được sẽ được trình bày chi tiết trong phần Kết quả nghiên cứu và thảo luận dưới đây.

1.1.5 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm:

- **Các văn bản học thuật của sinh viên ở bậc đại học và sau đại học:** chủ yếu là các bài báo cáo môn học, tiểu luận, luận văn tốt nghiệp hoặc các bài nghiên cứu khoa học sinh viên. Những văn bản này thường ở định dạng Microsoft Word (.docx) hoặc PDF.
- **Hành vi đạo văn trong văn bản:** các hình thức sao chép nội dung có thể gặp như sao chép y nguyên (copy & paste) từ nguồn khác, sao chép nhưng thay đổi một vài từ hoặc trật tự câu, dịch thuật từ ngôn ngữ khác, tự đạo văn (sinh viên nộp lại bài của chính mình ở môn khác), v.v. Đề tài tập trung vào việc phát hiện các dạng đạo văn này thông qua phân tích nội dung văn bản.
- **Phương pháp và công cụ phát hiện đạo văn:** nghiên cứu việc áp dụng mô hình ngôn ngữ MiniLM để biểu diễn văn bản, sử dụng cơ sở dữ liệu vector Milvus để tìm kiếm tương đồng, kết hợp với các kỹ thuật xử lý văn bản truyền thống. Đồng thời, nghiên cứu cách tích hợp những công nghệ này trong một hệ thống web phục vụ nhiều người dùng. Đối tượng sử dụng (người dùng mục tiêu) của hệ thống gồm:
 - **Sinh viên:** người nộp bài viết để kiểm tra đạo văn. Sinh viên có thể sử dụng hệ thống để tự kiểm tra bài của mình trước khi nộp chính thức (nếu được phép) hoặc nộp bài theo yêu cầu và nhận phản hồi kết quả từ giảng viên.
 - **Giảng viên:** người sử dụng hệ thống để kiểm tra đạo văn các bài nộp của sinh viên trong lớp mình giảng dạy. Giảng viên có quyền xem báo cáo chi tiết, đưa ra phản hồi hoặc hình thức xử lý nếu phát hiện vi phạm. Giảng viên cũng có thể sử dụng chức năng kiểm tra thủ công cho một đoạn văn cụ thể khi nghi ngờ.
- **Quản trị hệ thống:** bộ phận kỹ thuật của nhà trường hoặc nhóm phát triển, quản lý vận hành hệ thống. Quản trị viên có quyền quản lý người dùng (tài khoản giảng

viên, sinh viên), quản lý cơ sở dữ liệu tài liệu (thêm/xóa tài liệu nguồn vào kho đối chiếu nếu cần), cấu hình các thông số hệ thống (ví dụ: ngưỡng % trùng lặp để xác định đạo văn, cài đặt thông báo), và theo dõi thống kê toàn hệ thống.

Phạm vi nghiên cứu:

- **Về loại hình tài liệu:** Đề tài tập trung vào văn bản ở định dạng số (text), không xem xét phạm vi đạo văn trên các loại nội dung khác như mã nguồn phần mềm, nội dung đa phương tiện (hình ảnh, âm thanh) hay công thức toán học. Chỉ những nội dung chữ viết trong file văn bản được xử lý.
- **Về ngôn ngữ:** Hệ thống hỗ trợ tiếng Việt và tiếng Anh. Các ngôn ngữ khác (Pháp, Đức, Trung, v.v.) chưa nằm trong phạm vi triển khai, nhưng kiến trúc mở cho phép bổ sung mô hình ngôn ngữ tương ứng trong tương lai nếu cần. Trong giai đoạn nghiên cứu này, mô hình MiniLM đa ngôn ngữ được sử dụng chủ yếu cho hai ngôn ngữ nói trên.
- **Về phạm vi địa lý và tổ chức:** Hệ thống được thiết kế cho môi trường nhiều trường đại học cùng sử dụng. Tuy nhiên, trong khuôn khổ nghiên cứu và thử nghiệm, hệ thống được cài đặt và chạy thử tại một hoặc một vài trường cụ thể (ví dụ: trường của tác giả và có thể mở rộng thêm một trường đối tác để thử nghiệm liên trường). Việc triển khai trên diện rộng cho tất cả các trường sẽ được xem xét sau khi hoàn thiện mô hình thử nghiệm.
- **Về quy mô dữ liệu:** Do giới hạn thời gian và nguồn lực, cơ sở dữ liệu tài liệu dùng để đối chiếu đạo văn trong quá trình thử nghiệm có quy mô vừa phải (khoảng vài nghìn tài liệu, bao gồm khóa luận, báo cáo thu thập từ thư viện số của trường và các bài mẫu do giảng viên cung cấp). Hệ thống được thiết kế hướng đến khả năng mở rộng lên hàng trăm nghìn tài liệu, nhưng việc thử nghiệm hiệu suất thực tế ở quy mô rất lớn nằm ngoài phạm vi của đề tài này. Thay vào đó, đề tài thực hiện phân tích dự đoán dựa trên các kết quả thử nghiệm nhỏ hơn.
- **Về thời gian:** Đề tài được thực hiện trong khoảng thời gian dự kiến là 6-8 tháng. Trong thời gian này, tập trung vào phát triển các tính năng cốt lõi và chứng minh tính hiệu quả. Các tính năng nâng cao khác (như giao diện tối ưu cho di động, tích hợp với hệ thống quản lý học tập LMS, v.v.) nếu không kịp thực hiện sẽ ghi

nhận như hướng phát triển sau này.

Tóm lại, đối tượng nghiên cứu và phạm vi đã được xác định rõ để đảm bảo đề tài tập trung vào giải quyết đúng vấn đề đặt ra trong bối cảnh thực tế, đồng thời không sa đà vào những khía cạnh ngoài tầm kiểm soát. Hệ thống hướng tới phục vụ trực tiếp cho môi trường đại học, với quy mô và loại hình dữ liệu đặc thù, từ đó tạo ra sản phẩm có tính ứng dụng cao.

1.1.6 Cấu trúc báo cáo

Kết cấu đề tài trong bài nghiên cứu được trình bày gồm 5 chương chính:

- Chương 1: Tổng quan về vấn đề đạo văn và các hệ thống kiểm tra hiện nay
- Chương 2: Cơ sở lý thuyết và mô hình nghiên cứu
- Chương 3: Phương pháp nghiên cứu
- Chương 4: Triển khai thử nghiệm và đánh giá hiệu quả
- Chương 5: Kết luận và kiến nghị

1.2 Tổng quan nghiên cứu

1.2.1 Các công trình nghiên cứu trong nước

Tại Việt Nam, việc nghiên cứu các giải pháp kỹ thuật nhằm phát hiện hành vi sao chép trong văn bản học thuật đã được đề cập trong nhiều công trình học thuật và luận văn tốt nghiệp. Một trong những nghiên cứu đáng chú ý là của **Nguyễn Thị Mai và Trần Văn Dũng** (2016), với đề tài “Phát hiện đạo văn trong tài liệu tiếng Việt sử dụng kỹ thuật fingerprinting”, được công bố tại *Hội thảo Quốc gia về Nghiên cứu Ứng dụng CNTT*. Tác giả sử dụng thuật toán **w-shingling** kết hợp **băm (hashing)** để xây dựng chuỗi đặc trưng của văn bản, từ đó so sánh với kho dữ liệu hiện có. Hệ thống thử nghiệm cho thấy khả năng nhận diện tốt đối với các đoạn văn bị sao chép nguyên bản, nhưng hiệu quả giảm rõ rệt khi văn bản bị viết lại với từ đồng nghĩa.

Một hướng tiếp cận khác được trình bày trong nghiên cứu của **Phạm Văn Hòa** (2018), “Sử dụng chỉ mục nghịch đảo trong phát hiện trùng lặp nội dung trong bài luận sinh viên”, được triển khai tại Trường Đại học Bách Khoa TP.HCM. Tác giả áp dụng

kỹ thuật inverted index tương tự như công cụ tìm kiếm, cho phép phát hiện các đoạn trùng lặp dài giữa văn bản đầu vào và tài liệu trong cơ sở dữ liệu. Dù đơn giản và dễ triển khai, nghiên cứu chỉ ra rằng phương pháp này gặp khó khăn với các hành vi paraphrase – hình thức đạo văn phổ biến trong môi trường học thuật.

Gần đây hơn, **Ngô Đức Hùng** (2021) trong luận văn cao học tại Đại học Quốc gia Hà Nội đã tiến hành thử nghiệm một hệ thống phát hiện đạo văn sử dụng kết hợp **n-gram từ** và thống kê tần suất từ vựng. Mặc dù chưa áp dụng mô hình học sâu, hệ thống đạt độ chính xác 84% trong việc phát hiện đạo văn trực tiếp, và là cơ sở cho các bước nâng cấp sau này với mô hình ngôn ngữ hiện đại hơn.

1.2.2 Các công trình nghiên cứu nước ngoài

Trên thế giới, lĩnh vực phát hiện đạo văn đã được nghiên cứu từ nhiều góc độ, từ thuật toán truyền thống đến phương pháp học sâu hiện đại. Một trong những công trình nền tảng là của **Stein, Potthast và Barrón-Cedeño** (2011), “Intrinsic Plagiarism Detection”, công bố tại *CLEF Workshop*, trong đó nhóm tác giả giới thiệu kỹ thuật phân tích phong cách viết để nhận diện đoạn văn không đồng nhất với văn bản gốc – một hình thức đạo văn tinh vi không phụ thuộc vào đối sánh từ ngữ. Phương pháp này phù hợp trong các trường hợp không có tài liệu tham chiếu, nhưng đòi hỏi mô hình ngôn ngữ mạnh và tập dữ liệu huấn luyện lớn.

Một nghiên cứu quan trọng khác là của **Kolak và Resnik** (2002), “Detection of Text Reuse in Large Corpora”, tập trung vào áp dụng thuật toán fingerprinting dựa trên **hashing với Winnowing**. Đây là một trong những giải pháp phổ biến để phát hiện đạo văn trong các hệ thống quy mô lớn, như hệ thống kiểm tra luận văn đầu ra của đại học. Bằng cách băm các đoạn văn bản thành chuỗi nhỏ và chọn lọc các điểm đặc trưng, hệ thống cho phép phát hiện sao chép ngay cả khi thứ tự câu chữ bị đảo lộn ở mức nhẹ.

Bên cạnh đó, công trình của **Clough và Stevenson** (2009), “Developing a Corpus of Plagiarised Short Answers”, cung cấp bộ dữ liệu thực tế để thử nghiệm các phương pháp như **n-gram matching** và **vector space model**. Đây là nền tảng để đánh giá định lượng hiệu quả của các thuật toán phát hiện đạo văn ở cấp độ câu và đoạn ngắn.

Trong thời gian gần đây, nhiều nghiên cứu quốc tế đã nhấn mạnh sự kết hợp giữa các kỹ thuật truyền thống như **inverted index**, **shingling** với các công cụ kiểm tra

semantic như **word embedding** để nâng cao hiệu quả, đặc biệt trong việc phát hiện đạo văn nguy trang bằng cách viết lại. Tuy nhiên, ở các kịch bản yêu cầu tốc độ cao và độ chính xác lớn như môi trường giáo dục đại học, việc tối ưu hoá chỉ số n-gram, chiến lược băm dữ liệu, và tổ chức chỉ mục văn bản vẫn giữ vai trò quan trọng trong thiết kế hệ thống.

1.2.3 Tổng quan một số hệ thống kiểm tra đạo văn phổ biến trên thế giới

Dưới đây là Top 3 các hệ thống kiểm tra đạo văn lớn và phổ biến nhất hiện nay trên thế giới:

Turnitin

- Xuất xứ: Mỹ, 1998. Phát triển cho môi trường đại học.
- Cơ sở dữ liệu: Lớn nhất thế giới, bao gồm tài liệu sinh viên, bài báo khoa học, trang web.
- Tính năng chính: Kiểm tra trùng lặp, báo cáo tương đồng, tích hợp LMS.
- Đối tượng: Trường đại học, giáo viên.
- Ghi chú: Chi phí cao, không hỗ trợ người dùng cá nhân.

Copyscape

- Xuất xứ: Israel, 2004. Tập trung vào nội dung web.
- Cơ sở dữ liệu: Dựa trên các công cụ tìm kiếm (Google).
- Tính năng chính: Tìm bài sao chép trên Internet, theo dõi vi phạm, API.
- Đối tượng: Blogger, doanh nghiệp, quản trị web.
- Ghi chú: Nhanh, chi phí hợp lý, hạn chế với tài liệu học thuật.

Grammarly (Premium)

- Xuất xứ: Ukraina/Mỹ, 2009. Hỗ trợ viết tiếng Anh.
- Cơ sở dữ liệu: Web + kho ProQuest.
- Tính năng chính: Kiểm tra ngữ pháp + đạo văn, đánh giá Originality.
- Đối tượng: Sinh viên, freelancer, nhà viết.

- Ghi chú: Dễ dùng, chi phí cá nhân, không hỗ trợ tiếng Việt.

1.2.4 Đánh giá ưu – nhược điểm của các hệ thống hiện tại

Bảng 1.1: So sánh các hệ thống đạo văn nổi tiếng trên thế giới hiện nay

Tiêu chí	Turnitin	Copyscape	Grammarly
Độ chính xác	Rất cao (học thuật), tốt cho đạo văn ngữ gốc	Tốt cho sao chép web, yếu với paraphrase	Tốt với tiếng Anh, yếu với ngôn ngữ khác
Tốc độ xử lý	Trung bình, mất vài phút	Rất nhanh, tính giây	Nhanh, tích hợp ngay khi soạn thảo
Nhận biết diễn giải	Trung bình, có hỗ trợ cross-language	Rất hạn chế	Kém, chỉ so khớp câu gần giống
Ngôn ngữ hỗ trợ	>170 ngôn ngữ, tự dịch Anh-Việt	Bất kỳ ngôn ngữ nào Google hỗ trợ	Chỉ tiếng Anh
Chi phí	Rất cao, theo trường	Linh hoạt, rẻ dùng lẻ	Trung bình, trả theo tháng
Tích hợp hệ thống	Tốt nhất với LMS	API riêng, không có cho LMS	Không có cho học đường, chỉ plugin soạn thảo
Giao diện UI/UX	Chi tiết, chuyên nghiệp, có học Chi phí cao, giao diện Anh	Tối giản, thiếu trực quan Hạn chế tài liệu học thuật, giao diện Anh	Thân thiện, phù hợp người mới
Nhược điểm	Chi phí cao, giao diện Anh	Hạn chế tài liệu học thuật, giao diện Anh	Không hỗ trợ tiếng Việt

Nguồn: Tác giả tự tổng hợp

1.2.5 Khoảng trống nghiên cứu và cơ hội phát triển

Từ tổng quan trên có thể thấy, mặc dù đã có những giải pháp phát hiện đạo văn, nhưng vẫn tồn tại khoảng trống trong việc có một hệ thống **toàn diện** và **tích hợp** đáp ứng nhu cầu của nhiều trường đại học Việt Nam:

- Các giải pháp nước ngoài hiệu quả nhưng chi phí cao, khó tùy biến, và chưa chắc phù hợp ngôn ngữ tiếng Việt.
- Các giải pháp nội bộ thì phạm vi hẹp, chưa hỗ trợ đa ngôn ngữ, và công nghệ còn hạn chế dẫn đến độ chính xác chưa cao.
- Chưa có nhiều hệ thống có thể hỗ trợ tự động hóa quy trình kiểm tra, lưu trữ kết quả, báo cáo và thống kê tình trạng và phạm vi đạo văn.
- Chưa có hệ thống nào đóng vai trò nền tảng chung cho nhiều trường đại học cùng sử dụng, nhằm chia sẻ dữ liệu và kinh nghiệm, giúp phát hiện cả những trường hợp sinh viên sao chép chéo giữa các trường với nhau.

Những hạn chế đó là động lực để thực hiện đề tài này, ứng dụng công nghệ mới (mô hình ngôn ngữ MiniLM và cơ sở dữ liệu vector Milvus) nhằm xây dựng một hệ thống kiểm tra đạo văn có tính **toàn diện**: hỗ trợ nhiều trường, nhiều ngôn ngữ, giao diện thân thiện, chức năng phong phú và khả năng mở rộng trong tương lai.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ MÔ HÌNH NGHIÊN CỨU

2.1 Các khái niệm liên quan

2.1.1 Khái niệm đạo văn và kiểm tra đạo văn

Đạo văn là hành vi sao chép, sử dụng ý tưởng, câu chữ hoặc dữ liệu của người khác mà không trích dẫn nguồn gốc một cách hợp lệ, gây hiểu lầm về quyền sở hữu trí tuệ.

Phân loại đạo văn:

- Đạo văn nguyên văn: sao chép nguyên đoạn văn bản mà không trích dẫn.
- Đạo văn gián tiếp (paraphrase): diễn đạt lại ý tưởng của người khác nhưng không trích dẫn.
- Đạo văn ý tưởng: sử dụng ý tưởng hoặc cấu trúc logic mà không tham khảo nguồn.
- Đạo văn dịch thuật: dịch nguyên văn từ ngôn ngữ này sang ngôn ngữ khác mà không ghi nguồn.

Kiểm tra đạo văn là quá trình so sánh văn bản cần kiểm tra với các nguồn dữ liệu để xác định mức độ trùng lặp, nhằm đánh giá tính nguyên bản và trung thực học thuật. Việc kiểm tra thường sử dụng công nghệ so sánh chuỗi ký tự, thuật toán phát hiện tương đồng ngữ nghĩa và các mô hình trí tuệ nhân tạo để phát hiện paraphrase.

2.1.2 Ngôn ngữ và Framework lập trình

Hệ thống chính được phát triển sử dụng PHP với framework Laravel nhằm xây dựng nền tảng web ổn định, dễ bảo trì và mở rộng. Laravel cung cấp các tiện ích quản lý người dùng, phân quyền và kết nối cơ sở dữ liệu MySQL hiệu quả.

Dịch vụ kiểm tra đạo văn riêng biệt dùng Python với Flask – framework nhẹ, đơn giản cho phát triển API và dịch vụ microservice. Python được lựa chọn vì sự phong phú của thư viện xử lý ngôn ngữ tự nhiên và tích hợp thuận tiện với các mô hình AI.

Các thành phần hệ thống giao tiếp qua REST API hoặc các giao thức HTTP chuẩn, được cân bằng tải bởi Nginx để đảm bảo khả năng mở rộng và ổn định.

2.1.3 Xử lý ngôn ngữ tự nhiên (NLP)

NLP là lĩnh vực nghiên cứu và ứng dụng các kỹ thuật để máy tính có thể hiểu, phân tích và tạo ra ngôn ngữ tự nhiên của con người.

Trong hệ thống kiểm tra đạo văn, NLP được ứng dụng để phân đoạn văn bản thành các câu hoặc đoạn nhỏ, chuẩn hóa nội dung (loại bỏ dấu câu, chuyển chữ hoa thành chữ thường), và đặc biệt là để tạo các biểu diễn số (vector embedding) của câu văn nhằm phát hiện đạo văn theo nghĩa rộng, bao gồm paraphrase.

Các kỹ thuật NLP được sử dụng gồm tokenization, stop word removal, stemming/lemmatization, và embedding từ các mô hình ngôn ngữ hiện đại.

2.1.4 Mô hình nhúng văn bản (Embedding Models)

Mô hình embedding là các mạng nơ-ron được huấn luyện để ánh xạ các câu, đoạn văn thành các vector số trong không gian đa chiều sao cho các câu có nghĩa tương tự sẽ có vector gần nhau về mặt hình học.

Hệ thống của bạn sử dụng mô hình MiniLM – một mô hình đa ngôn ngữ có khả năng nhúng câu hiệu quả, hỗ trợ cả tiếng Anh và tiếng Việt, giúp tăng độ chính xác trong việc phát hiện các đoạn văn có nội dung tương đồng dù khác cách diễn đạt.

MiniLM giúp giảm thời gian so sánh và tăng khả năng nhận diện các trường hợp paraphrase hoặc đạo văn phức tạp.

2.1.5 Cơ sở dữ liệu và lưu trữ

Cơ sở dữ liệu vector Milvus

Milvus được phát triển bởi Zilliz và hiện là một trong những dự án cơ sở dữ liệu vector hàng đầu thế giới, được cấp phép theo chuẩn Apache 2.0. Milvus nổi bật nhờ khả năng mở rộng linh hoạt từ môi trường máy cá nhân đến các hệ thống phân tán quy mô lớn chạy trên Kubernetes. Kiến trúc của Milvus hướng đến hiệu năng cao và khả năng tối ưu hóa trên nhiều nền tảng phần cứng khác nhau như CPU đa lõi, GPU, và ổ cứng NVMe.

Milvus chuyên quản lý các dữ liệu phi cấu trúc như văn bản, hình ảnh, âm thanh bằng cách chuyển đổi chúng thành các vector embedding đa chiều. Cơ sở dữ liệu này hỗ trợ lưu trữ nhiều kiểu dữ liệu khác nhau, bao gồm vector thưa, vector nhị phân, JSON,

và các kiểu dữ liệu phức tạp khác, giúp giảm thiểu việc phải duy trì nhiều hệ thống cơ sở dữ liệu riêng biệt.

Milvus cung cấp ba hình thức triển khai linh hoạt:

- **Milvus Lite:** thư viện Python nhẹ dùng cho phát triển nhanh và môi trường tài nguyên hạn chế.
- **Milvus Standalone:** phiên bản chạy trên một máy chủ đơn, đóng gói trong Docker, phù hợp thử nghiệm và môi trường vừa phải.
- **Milvus Distributed:** kiến trúc đám mây gốc, triển khai trên Kubernetes, hỗ trợ quy mô hàng chục tỷ vector với khả năng chịu lỗi cao.

Hiệu suất Milvus vượt trội nhờ các điểm sau:

- Tối ưu hóa phần cứng dựa trên các công nghệ như AVX512, SIMD, GPU và NVMe SSD.
- Hỗ trợ đa dạng thuật toán tìm kiếm và lập chỉ mục như IVF, HNSW, DiskANN được tối ưu sâu về tốc độ và bộ nhớ.
- Thành phần tìm kiếm cốt lõi được viết bằng C++ với các kỹ thuật tối ưu cấp thấp, đa luồng, tận dụng triệt để tài nguyên phần cứng.
- Thiết kế hệ quản trị dữ liệu theo hướng cột, giúp giảm thiểu lượng dữ liệu đọc khi truy vấn, tăng tốc xử lý hàng loạt các trường dữ liệu.

Milvus có kiến trúc tách rời các thành phần chức năng như tìm kiếm, chèn dữ liệu, và lập chỉ mục, cho phép mở rộng quy mô linh hoạt theo chiều ngang hoặc chiều dọc. Nhờ đó, hệ thống có thể mở rộng từ vài triệu đến hàng chục tỷ vector một cách ổn định và hiệu quả, đáp ứng nhu cầu của nhiều doanh nghiệp lớn trong các ngành công nghiệp khác nhau.

Milvus cung cấp đa dạng các loại truy vấn nhằm đáp ứng nhu cầu khác nhau của ứng dụng:

- Tìm kiếm ANN (Approximate Nearest Neighbor) để lấy các vector gần nhất với truy vấn.
- Tìm kiếm có điều kiện lọc (Filtering Search) và tìm kiếm theo khoảng cách

(Range Search).

- Tìm kiếm lai kết hợp nhiều vector (Hybrid Search).
- Tìm kiếm toàn văn bản dựa trên thuật toán BM25.
- Chức năng tái xếp hạng kết quả (Reranking) để tinh chỉnh thứ tự trả về.

Cơ sở dữ liệu quan hệ MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến nhất thế giới, được phát triển và duy trì bởi Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn chuẩn SQL để quản lý và thao tác dữ liệu, giúp tổ chức dữ liệu có cấu trúc một cách hiệu quả.

Ưu điểm chính của MySQL:

- **Phổ biến rộng rãi và cộng đồng lớn:** MySQL được sử dụng rộng rãi trong nhiều ứng dụng web và doanh nghiệp, với tài liệu phong phú và nhiều công cụ hỗ trợ.
- **Hiệu năng cao và ổn định:** MySQL tối ưu cho các ứng dụng cần xử lý truy vấn nhanh, khả năng mở rộng linh hoạt, đáp ứng tốt các hệ thống từ nhỏ đến lớn.
- **Dễ dàng tích hợp:** MySQL tương thích với nhiều ngôn ngữ lập trình và nền tảng, hỗ trợ đa dạng công cụ quản lý và giao diện.
- **Bảo mật và sao lưu:** Cung cấp nhiều cơ chế bảo mật dữ liệu, kiểm soát truy cập và hỗ trợ sao lưu – phục hồi dữ liệu hiệu quả.
- **Tính năng phong phú:** Hỗ trợ các kiểu dữ liệu đa dạng, giao dịch ACID, khóa ngoại, chỉ mục, và nhiều loại lưu trữ (storage engines) phù hợp các mục đích sử dụng khác nhau.

MySQL là lựa chọn lý tưởng cho việc lưu trữ và quản lý dữ liệu có cấu trúc trong các hệ thống thông tin, đặc biệt trong môi trường web và các ứng dụng doanh nghiệp.

2.2 Yêu cầu và mục tiêu thiết kế hệ thống

Hệ thống kiểm tra đạo văn toàn diện được xây dựng nhằm đáp ứng các yêu cầu chức năng và phi chức năng thiết yếu trong bối cảnh môi trường giáo dục đại học. Các yêu cầu cụ thể bao gồm:

- **Quản lý người dùng và phân quyền:** Hệ thống phải hỗ trợ đầy đủ chức năng đăng ký, đăng nhập, đăng xuất, cập nhật thông tin cá nhân, thay đổi và khôi phục mật khẩu, cùng với việc phân quyền rõ ràng cho các nhóm người dùng như sinh viên, giảng viên, quản trị viên.
- **Tự động hóa quy trình kiểm tra đạo văn:** Hệ thống cần cung cấp cơ chế kiểm tra tự động đối với các tài liệu được nộp, giảm thiểu thao tác thủ công, tăng tính chính xác và hiệu quả trong việc phát hiện trùng lặp nội dung. Ngoài ra, cần có khả năng kiểm tra thủ công khi cần thiết.
- **Tích hợp mô hình embedding hiện đại:** Sử dụng các mô hình nhúng văn bản đa ngôn ngữ (MiniLM) để chuyển đổi nội dung văn bản thành vector biểu diễn, giúp phát hiện các trường hợp đạo văn phức tạp như paraphrase hoặc dịch thuật.
- **Cơ sở dữ liệu đa dạng:** Hệ thống phải lưu trữ hiệu quả dữ liệu truyền thống (MySQL) cùng với dữ liệu vector embedding (Milvus), đảm bảo tốc độ truy vấn và mở rộng quy mô dữ liệu lớn.
- **Báo cáo và thống kê:** Cung cấp các báo cáo chi tiết theo lớp học, giảng viên, môn học và thời gian; thống kê tổng hợp mức độ đạo văn nhằm hỗ trợ quản lý và đánh giá tình hình đạo văn tại trường.
- **Quản lý tài liệu và dữ liệu tham khảo:** Cho phép quản lý tài liệu đã nộp, dữ liệu tham khảo, loại bỏ trùng lặp, kiểm tra định dạng file và phân loại tài liệu theo nhãn.
- **Quản lý thông báo và phản hồi:** Hệ thống cần hỗ trợ gửi thông báo, sự kiện, nhắc nhở, cùng với khả năng tiếp nhận phản hồi và khiếu nại từ người dùng.
- **Khả năng mở rộng và tích hợp:** Hệ thống được thiết kế để dễ dàng tích hợp với các nền tảng quản lý học tập (LMS) và các dịch vụ liên quan khác, đồng

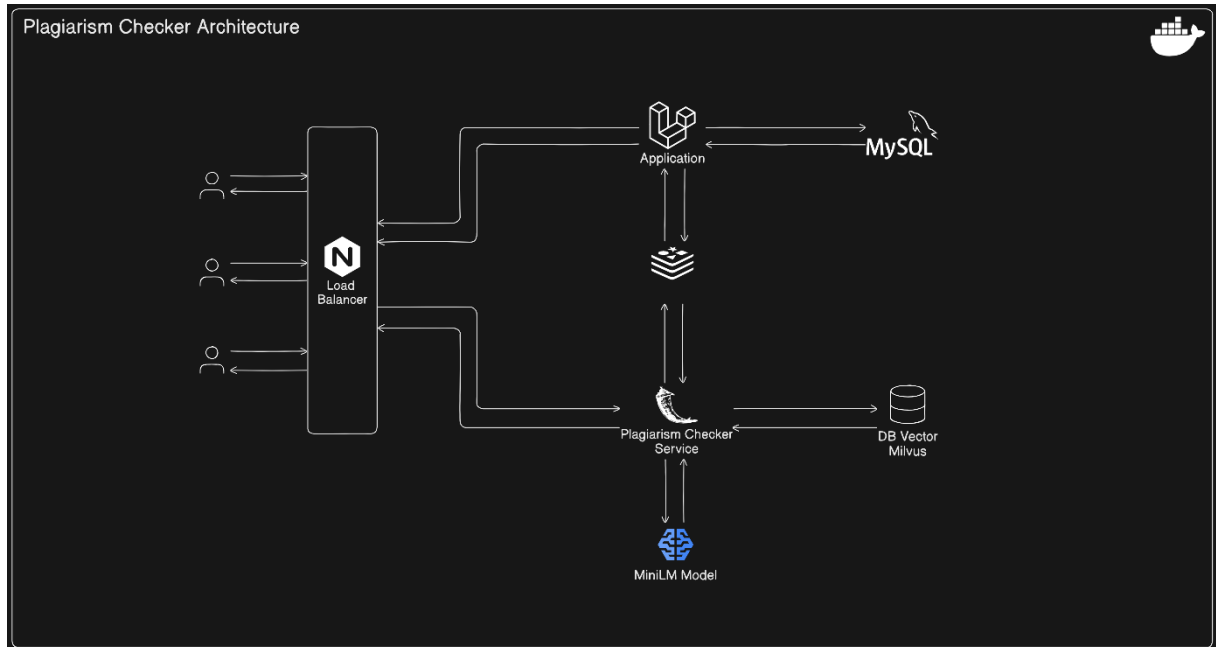
thời cho phép mở rộng về mặt công nghệ như cập nhật mô hình nhúng mới hoặc bổ sung các chức năng kiểm tra nâng cao.

- **Bảo mật và ổn định:** Đảm bảo an toàn thông tin người dùng, bảo vệ dữ liệu khỏi truy cập trái phép, đồng thời duy trì độ ổn định cao dưới tải lớn và trong quá trình vận hành liên tục.

Trên cơ sở các yêu cầu, hệ thống hướng đến các mục tiêu thiết kế sau nhằm đáp ứng nhu cầu thực tiễn trong kiểm tra đạo văn tại các trường đại học:

- **Tăng tốc độ và độ chính xác kiểm tra đạo văn:** Áp dụng mô hình embedding đa ngôn ngữ MiniLM phối hợp với dịch vụ Milvus nhằm giảm thời gian xử lý, đồng thời nâng cao khả năng phát hiện các dạng đạo văn tinh vi như paraphrase hoặc đạo văn dịch thuật.
- **Tự động hóa toàn diện quy trình kiểm tra:** Giảm thiểu sự can thiệp thủ công trong việc xử lý, kiểm tra và báo cáo đạo văn, giúp giảng viên và cán bộ quản lý tiết kiệm thời gian, nâng cao hiệu suất làm việc.
- **Hỗ trợ quản lý dữ liệu đa dạng và hiệu quả:** Quản lý hiệu quả các tài liệu học thuật, dữ liệu tham khảo, lịch sử kiểm tra và quyền truy cập người dùng, đảm bảo tính minh bạch và chính xác trong quy trình kiểm tra.
- **Cung cấp giao diện thân thiện và đa năng:** Tạo ra môi trường tương tác trực quan cho người dùng với các chức năng đăng ký, quản lý, kiểm tra và báo cáo dễ sử dụng, hỗ trợ người dùng từ nhiều đối tượng khác nhau trong trường đại học.
- **Tối ưu khả năng mở rộng và tích hợp:** Thiết kế kiến trúc microservice và áp dụng các design pattern (Factory, Strategy) để linh hoạt chuyển đổi và nâng cấp mô hình kiểm tra, dễ dàng tích hợp với các hệ thống khác, phù hợp với chiến lược phát triển lâu dài.
- **Bảo đảm an toàn và ổn định:** Hệ thống được xây dựng trên nền tảng công nghệ tin cậy, có cơ chế bảo mật nghiêm ngặt và khả năng chịu tải cao nhằm vận hành ổn định trong môi trường thực tế với lượng người dùng lớn.

2.3 Kiến trúc tổng thể của hệ thống



Hình 2.1: Sơ đồ tổng quan kiến trúc hệ thống

Hệ thống kiểm tra đạo văn được thiết kế theo kiến trúc microservice, phân tách thành các thành phần độc lập nhằm đảm bảo tính mở rộng, bảo trì dễ dàng và khả năng vận hành hiệu quả trong môi trường sản xuất. Hai thành phần chính của hệ thống bao gồm:

- **Ứng dụng chính (Application Layer):** Được phát triển bằng PHP với framework Laravel, chịu trách nhiệm giao diện người dùng, quản lý người dùng, phân quyền, xử lý dữ liệu nhập xuất, lưu trữ thông tin cấu hình và kết quả kiểm tra. Ứng dụng này đóng vai trò trung tâm trong việc tiếp nhận yêu cầu từ người dùng (người quản trị hệ thống, giảng viên, sinh viên), gửi dữ liệu sang dịch vụ kiểm tra đạo văn, đồng thời hiển thị kết quả trả về và báo cáo tổng hợp.
- **Dịch vụ kiểm tra đạo văn (Plagiarism Checker Service):** Xây dựng bằng Python sử dụng Flask để cung cấp API REST xử lý chuyên sâu nhiệm vụ phân tích và so sánh nội dung văn bản. Thành phần này tích hợp mô hình nhúng văn bản (embedding model) MiniLM và sử dụng Milvus để lưu trữ,

truy vấn vector embedding nhằm phát hiện sự tương đồng văn bản nhanh và chính xác. Việc tách riêng dịch vụ này giúp tận dụng sức mạnh của hệ sinh thái Python và các thư viện xử lý ngôn ngữ tự nhiên.

Kiến trúc microservice giúp hệ thống vận hành với khả năng **cân bằng tải** và **phân tán yêu cầu** hiệu quả, qua đó đảm bảo độ ổn định và khả năng phục hồi cao. Thành phần **Nginx** được sử dụng làm **load balancer**, phân phối các yêu cầu từ người dùng đến các instance của ứng dụng Laravel và dịch vụ kiểm tra, góp phần cải thiện khả năng chịu tải và giảm thiểu điểm nghẽn trong hệ thống. Mô hình này cũng cho phép triển khai độc lập từng thành phần, dễ dàng cập nhật, mở rộng hoặc thay thế các module mà không ảnh hưởng đến toàn bộ hệ thống.

Về mặt lưu trữ, hệ thống kết hợp **MySQL** làm cơ sở dữ liệu quan hệ cho các dữ liệu truyền thống như thông tin tài khoản, lịch sử kiểm tra, cấu hình quyền truy cập, và **Milvus** cho việc lưu trữ và truy vấn dữ liệu vector embedding để rút ngắn đáng kể thời gian xử lý các truy vấn so sánh phức tạp với khối lượng dữ liệu lớn.

Để tăng tính linh hoạt và khả năng mở rộng cho hệ thống, đặc biệt trong bối cảnh công nghệ AI và NLP phát triển nhanh chóng, kiến trúc phần mềm được thiết kế ứng dụng các **design pattern** phổ biến trong phát triển phần mềm:

- **Singleton Pattern:** Hệ thống áp dụng Singleton Pattern để quản lý kết nối tới cơ sở dữ liệu, đảm bảo chỉ có một phiên bản kết nối duy nhất tồn tại trong toàn bộ vòng đời ứng dụng. Thiết kế này giúp tránh lãng phí tài nguyên do việc tạo nhiều kết nối không cần thiết, đồng thời đảm bảo tính nhất quán và đồng bộ khi truy cập dữ liệu.
- **Factory Pattern:** Hệ thống sử dụng Factory Pattern để khởi tạo các mô hình embedding khác nhau một cách linh hoạt. Khi cần chuyển đổi hoặc cập nhật mô hình nhúng (ví dụ từ MiniLM sang một mô hình mới có hiệu quả cao hơn). Thiết kế này giúp cô lập phần khởi tạo mô hình khỏi phần xử lý logic chính, giảm thiểu sự phụ thuộc giữa các module, đồng thời hỗ trợ việc thử nghiệm các mô hình embedding khác nhau mà không cần thay đổi mã nguồn rộng rãi.
- **Strategy Pattern:** Hệ thống sử dụng Strategy Pattern được áp dụng để quản

lý và lựa chọn chiến lược nhúng văn bản trong quá trình xử lý kiểm tra đạo văn. Từ đó, hệ thống có thể thay đổi chiến lược sử dụng mô hình embedding dựa trên yêu cầu thực tế, chẳng hạn chọn mô hình phù hợp với ngôn ngữ hoặc mục tiêu kiểm tra cụ thể. Điều này tạo ra sự linh hoạt trong vận hành, dễ dàng mở rộng hỗ trợ nhiều loại mô hình khác nhau mà không làm gián đoạn các thành phần khác của hệ thống.

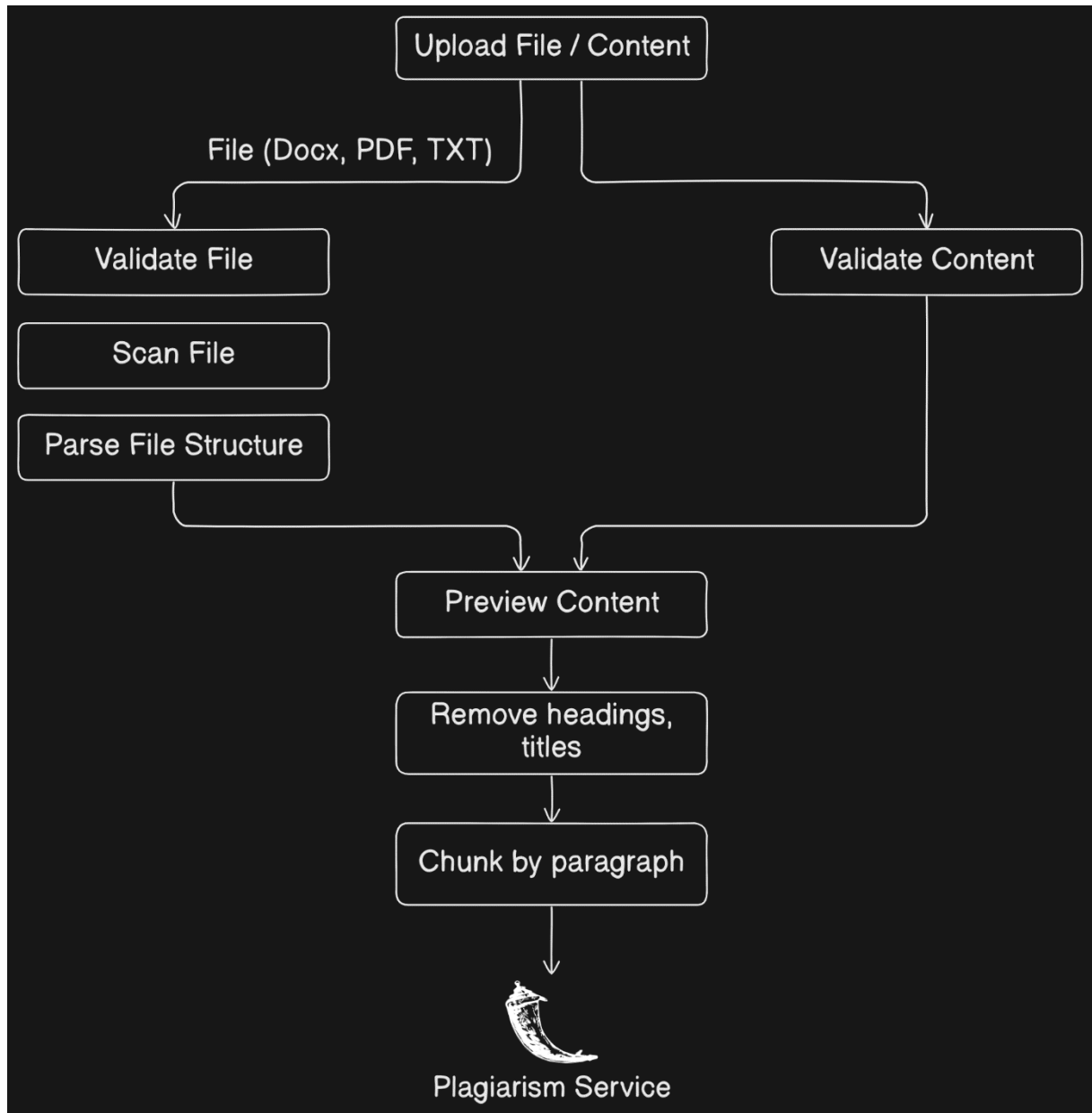
Việc kết hợp hai design pattern này nâng cao tính **modular**, **mở rộng** và **bảo trì** của dịch vụ kiểm tra đạo văn, đặc biệt trong bối cảnh công nghệ NLP và mô hình AI liên tục được cải tiến. Khi có các mô hình embedding mới xuất hiện, việc tích hợp hoặc chuyển đổi có thể thực hiện nhanh chóng và an toàn mà không cần thay đổi kiến trúc tổng thể.

Tổng kết lại, kiến trúc hệ thống kiểm tra đạo văn của đề tài là sự kết hợp hài hòa giữa các thành phần chuyên biệt, cơ sở dữ liệu hiệu năng cao và các mẫu thiết kế phần mềm hiện đại nhằm tạo nên một hệ thống có độ tin cậy cao, dễ mở rộng và phù hợp với yêu cầu thực tế của môi trường giáo dục Việt Nam.

2.4 Mô hình xử lý văn bản

Mô hình xử lý văn bản trong hệ thống kiểm tra đạo văn được thiết kế nhằm chuẩn hóa, tiền xử lý và chuyển đổi các văn bản đầu vào thành các biểu diễn số (embedding vectors) phục vụ cho việc phát hiện sự tương đồng. Quá trình này bao gồm nhiều bước liên tiếp, được minh họa trong sơ đồ kiến trúc xử lý văn bản của hệ thống.

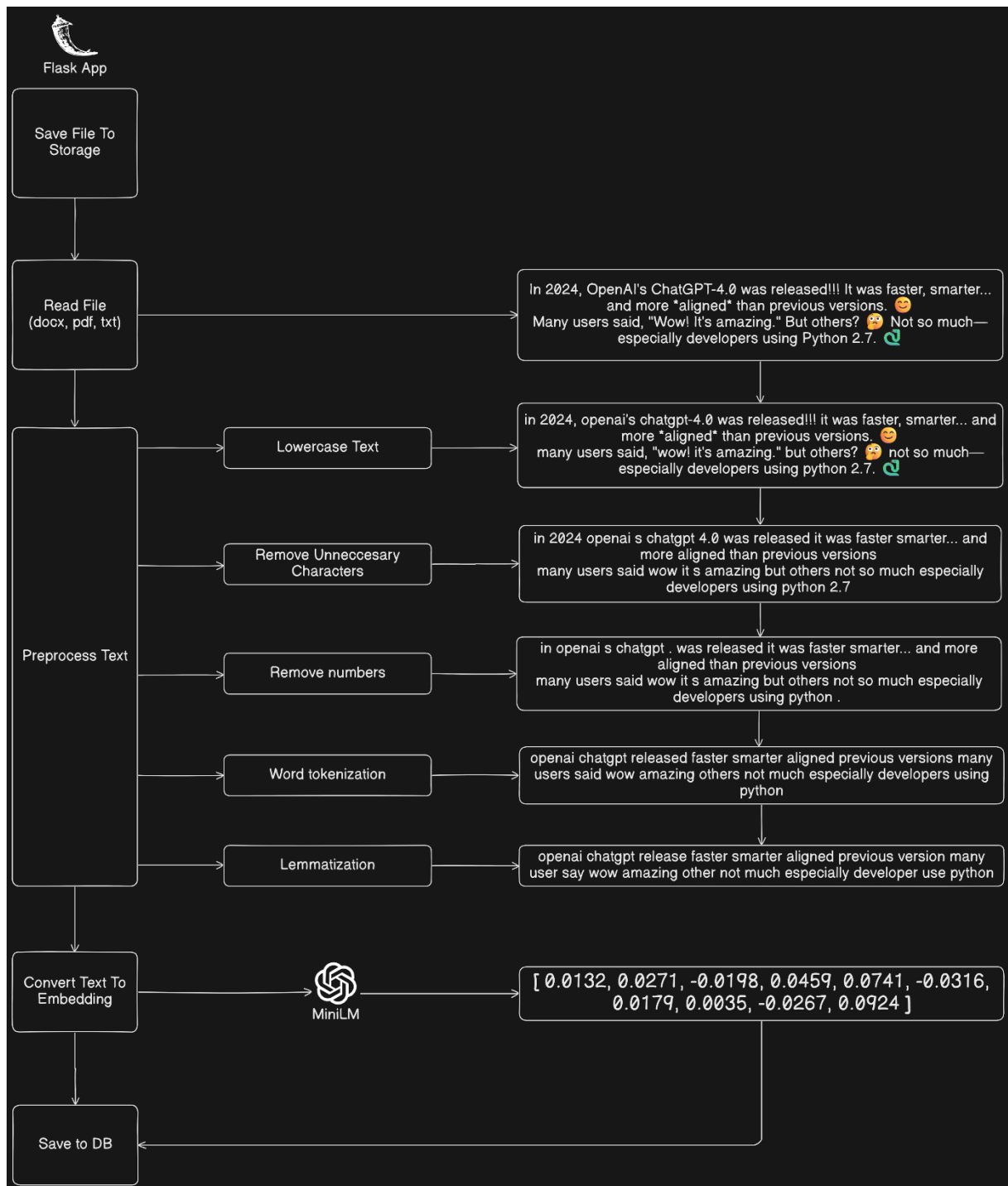
2.4.1 Đọc và lưu trữ tài liệu đầu vào



Hình 2.2: Quá trình tiếp nhận và lưu trữ dữ liệu đầu vào

Hệ thống hỗ trợ đa dạng các định dạng file phổ biến bao gồm **DOCX, PDF, TXT**. Các tài liệu này được nhận từ người dùng qua giao diện ứng dụng chính, sau đó được gửi tới dịch vụ xử lý văn bản viết bằng Flask. Trước tiên, các file được lưu trữ tạm thời trên hệ thống để phục vụ cho bước tiền xử lý. Quá trình đọc file đảm bảo trích xuất nội dung văn bản chính xác, loại bỏ các thành phần không cần thiết như hình ảnh, định dạng font chữ để tập trung vào phân tích ngữ nghĩa.

2.4.2 Tiền xử lý văn bản (Preprocessing)



Hình 2.3: Quá trình xử lý văn bản

Tiền xử lý là bước quan trọng nhằm chuẩn hóa nội dung văn bản, làm sạch dữ liệu trước khi chuyển sang bước nhúng. Các bước tiền xử lý trong mô hình bao gồm:

- **Chuyển về chữ thường (Lowercase):** Toàn bộ văn bản được chuẩn hóa sang dạng chữ thường để tránh phân biệt không cần thiết giữa chữ hoa và chữ thường trong quá trình phân tích.
- **Loại bỏ ký tự không cần thiết (Remove unnecessary characters):** Hệ

thông tự động lọc bỏ các ký tự đặc biệt, dấu câu không ảnh hưởng đến ngữ nghĩa, giúp giảm nhiễu cho mô hình.

- **Xóa số (Remove numbers):** Các con số trong văn bản được loại bỏ để tránh ảnh hưởng không mong muốn đến kết quả so sánh, trừ những trường hợp số có ý nghĩa ngữ cảnh đặc biệt được xử lý riêng (nếu có).
- **Phân tách từ (Word tokenization):** Văn bản được tách thành các từ đơn lẻ hoặc cụm từ để chuẩn bị cho bước xử lý tiếp theo.
- **Rút gọn từ (Lemmatization):** Các từ được chuyển về dạng gốc (lemma) nhằm giảm sự đa dạng về hình thức từ, hỗ trợ mô hình nhận diện các biến thể từ cùng gốc ngữ nghĩa.

Mỗi bước tiền xử lý đều nhằm đảm bảo đầu vào sạch, đồng nhất, giúp tăng độ chính xác và hiệu suất của mô hình nhúng văn bản.

2.4.3 Chuyển đổi văn bản thành embedding

Sau khi hoàn thành tiền xử lý, văn bản được đưa vào mô hình nhúng văn bản **MiniLM** – một mô hình đa ngôn ngữ tối ưu cho việc biểu diễn ngữ nghĩa câu dưới dạng vector số trong không gian đa chiều. MiniLM cho phép hệ thống:

- Biểu diễn các đoạn văn bản dưới dạng vector có kích thước cố định, giúp so sánh hiệu quả dựa trên khoảng cách hình học.
- Nhận diện được sự tương đồng ngữ nghĩa ở mức cao, đặc biệt hữu ích trong việc phát hiện các trường hợp đạo văn có paraphrase hoặc dịch thuật.
- Giữ hiệu suất cao trong xử lý đa ngôn ngữ, bao gồm tiếng Việt và tiếng Anh.

Kết quả đầu ra của mô hình là các vector embedding số được lưu trữ trong cơ sở dữ liệu vector Milvus để phục vụ các truy vấn tìm kiếm tương đồng nhanh chóng và chính xác.

2.4.4 Lưu trữ dữ liệu embedding

Các vector embedding được lưu trữ trong Milvus – một cơ sở dữ liệu vector chuyên dụng, tối ưu cho các thao tác truy vấn khoảng cách và tương đồng trong không gian đa chiều. Sự kết hợp giữa Milvus và MySQL (dữ liệu truyền thống) cho phép hệ thống

quản lý đồng bộ cả dữ liệu số và dữ liệu meta, hỗ trợ toàn diện cho chức năng kiểm tra đạo văn.

Mô hình xử lý văn bản được thiết kế theo luồng tuần tự, kết hợp các kỹ thuật tiên xử lý truyền thống với công nghệ AI hiện đại, nhằm tạo ra một hệ thống kiểm tra đạo văn vừa chính xác, vừa hiệu quả và phù hợp với thực tế ứng dụng tại các trường đại học.

2.5 Các thuật toán so sánh văn bản

Để phát hiện đạo văn hiệu quả trong hệ thống, việc so sánh các đoạn văn bản được thực hiện trên không gian vector embedding đa chiều. Thuật toán chính được ứng dụng là HNSW (Hierarchical Navigable Small World graph), một cấu trúc dữ liệu đồ thị được thiết kế nhằm tăng tốc độ tìm kiếm các vector gần nhất (nearest neighbor search) trong không gian lớn. HNSW xây dựng một đồ thị nhiều lớp, trong đó mỗi lớp là một mạng lưới nhỏ thế giới (small-world graph).

Mạng lưới này có đặc điểm là:

- Các đỉnh tương ứng với các vector embedding của văn bản.
- Mỗi đỉnh được kết nối với một số lượng giới hạn các đỉnh lân cận theo khoảng cách địa lý trong không gian vector (thường là khoảng cách cosine hoặc Euclidean).
- Đồ thị có nhiều lớp, với lớp trên cùng chứa ít đỉnh hơn, giúp tạo ra một cấu trúc phân cấp để tìm kiếm nhanh hơn.
- Khi cần tìm kiếm các văn bản tương tự, thuật toán thực hiện như sau:
- Bắt đầu từ lớp trên cùng của đồ thị, tiến hành duyệt theo hướng giảm dần khoảng cách đến vector truy vấn (query vector).
- Ở mỗi lớp, thuật toán tìm các đỉnh gần nhất so với vector truy vấn bằng cách duyệt các cạnh trong đồ thị nhỏ thế giới.
- Khi đạt đến lớp dưới cùng (lớp cơ sở), thuật toán tiến hành tìm kiếm chi tiết hơn để xác định các vector gần nhất trong toàn bộ tập dữ liệu.
- Kết quả trả về là tập các vector có khoảng cách nhỏ nhất so với vector truy vấn, tương ứng với các đoạn văn bản có nội dung tương đồng cao.

Ưu điểm của HNSW trong so sánh văn bản:

- Tốc độ cao: HNSW có độ phức tạp tìm kiếm gần như logarit theo số lượng điểm dữ liệu, giúp tăng tốc đáng kể so với các phương pháp tìm kiếm tuần tự.
- Độ chính xác gần tối ưu: Do cấu trúc phân cấp và khả năng điều hướng trong đồ thị nhỏ thế giới, HNSW cho phép tìm kiếm gần đúng với độ chính xác rất cao so với phép tìm kiếm chính xác toàn bộ (exact nearest neighbor).
- Khả năng mở rộng: HNSW phù hợp với các tập dữ liệu lớn, hỗ trợ hiệu quả cho các hệ thống kiểm tra đạo văn có khối lượng tài liệu và vector embedding lớn.
- Tích hợp dễ dàng: Thuật toán được hỗ trợ trực tiếp trong Milvus, giúp hệ thống của bạn tận dụng công nghệ lưu trữ và truy vấn vector tối ưu mà không cần phát triển lại thuật toán từ đầu.

Bảng 2.1: Thông tin các tham số quan trọng trong việc cấu hình index và tìm kiếm sử dụng thuật toán HNSW

Tham số	Mô tả	Phạm vi giá trị	Gợi ý điều chỉnh
M	Số kết nối tối đa (cả vào và ra) mỗi nút trong đồ thị, ảnh hưởng đến việc xây dựng và tìm kiếm.	2 – 2048 (mặc định 30)	<ul style="list-style-type: none">- Tăng M giúp tăng độ chính xác nhưng tốn bộ nhớ và chậm hơn.- Giảm M giúp tiết kiệm bộ nhớ, tăng tốc độ.- Khuyến nghị: 5 – 100.
efConstruction	Số lượng nút ứng viên được xem xét khi xây dựng index, ảnh hưởng chất lượng index.	1 – số nguyên lớn (mặc định 360)	<ul style="list-style-type: none">- Tăng efConstruction cải thiện độ chính xác nhưng tăng thời gian và bộ nhớ xây dựng.- Giảm để tiết kiệm tài nguyên.

			- Khuyến nghị: 50 – 500.
ef (tìm kiếm)	Kiểm soát phạm vi tìm kiếm khi truy vấn, xác định số nút được duyệt để tìm láng giềng gần nhất.	1 – số nguyên lớn (mặc định = số lượng K trả về)	<ul style="list-style-type: none"> - Tăng ef giúp tìm kiếm chính xác hơn nhưng chậm hơn. - Giảm ef tăng tốc tìm kiếm nhưng giảm độ chính xác. - Khuyến nghị: từ K đến 10K.

Nguồn: Tài liệu hệ thống cơ sở dữ liệu Milvus

Trong hệ thống kiểm tra đạo văn toàn diện, HNSW được sử dụng để so sánh vector embedding của đoạn văn bản người dùng gửi lên với tập các vector đã lưu trong cơ sở dữ liệu Milvus. Việc này cho phép phát hiện nhanh các đoạn văn bản có nội dung tương tự hoặc trùng lặp, từ đó đưa ra kết quả đánh giá mức độ đạo văn. Thuật toán HNSW không chỉ hỗ trợ phát hiện các đoạn trùng lặp nguyên văn mà còn giúp nhận diện các trường hợp paraphrase hoặc diễn đạt lại ý tưởng với mức độ tương đồng cao thông qua các embedding đã được chuẩn hóa từ MiniLM embedding model.

2.6 Bảo mật và quyền riêng tư dữ liệu người dùng

Việc bảo mật thông tin và bảo vệ quyền riêng tư của người dùng là một trong những yêu cầu thiết yếu trong thiết kế hệ thống kiểm tra đạo văn, đặc biệt trong môi trường giáo dục đại học, nơi dữ liệu cá nhân và học thuật có tính nhạy cảm cao.

- **Bảo vệ dữ liệu cá nhân:** Hệ thống tuân thủ các nguyên tắc bảo mật thông tin cá nhân bằng cách mã hóa dữ liệu nhạy cảm như mật khẩu (sử dụng thuật toán băm an toàn như bcrypt), dữ liệu liên quan đến thông tin cá nhân sinh viên và giảng viên được lưu trữ trong cơ sở dữ liệu có kiểm soát truy cập chặt chẽ.
- **Kiểm soát truy cập và phân quyền:** Mỗi người dùng được cấp quyền truy cập theo vai trò (role-based access control) như sinh viên, giảng viên, quản trị viên, đảm bảo chỉ những cá nhân có thẩm quyền mới có thể xem hoặc chỉnh sửa dữ liệu tương ứng.

- **Bảo mật truyền tải:** Các giao tiếp giữa client và server được bảo vệ bằng giao thức HTTPS, đảm bảo dữ liệu trao đổi không bị đánh cắp hoặc giả mạo.
- **Bảo vệ dữ liệu kiểm tra đạo văn:** Các tài liệu được người dùng gửi lên, cùng với kết quả kiểm tra, được lưu trữ an toàn, có cơ chế sao lưu và bảo mật để tránh rò rỉ hoặc truy cập trái phép.
- **Tuân thủ pháp luật về dữ liệu:** Hệ thống được thiết kế phù hợp với các quy định pháp lý hiện hành về bảo vệ dữ liệu cá nhân và sở hữu trí tuệ, tạo sự tin tưởng và tuân thủ chuẩn mực đạo đức trong môi trường học thuật.
- **Giám sát và phát hiện xâm nhập:** Ứng dụng có cơ chế ghi nhận nhật ký hoạt động (logging) và giám sát sự kiện bảo mật để phát hiện sớm các hành vi bất thường hoặc tấn công, từ đó kịp thời xử lý.

Những biện pháp này góp phần xây dựng môi trường kiểm tra đạo văn an toàn, minh bạch và bảo vệ quyền lợi hợp pháp của người dùng.

2.7 Giao diện và trải nghiệm người dùng (UX/UI)

Giao diện và trải nghiệm người dùng đóng vai trò quan trọng trong việc nâng cao hiệu quả sử dụng và sự hài lòng của người dùng hệ thống kiểm tra đạo văn. Hệ thống được thiết kế với các tính năng đa dạng, hỗ trợ đầy đủ quy trình quản lý và kiểm tra trong môi trường đại học, đồng thời đảm bảo sự trực quan và tiện lợi.

- **Quản lý người dùng:** Các chức năng đăng ký tài khoản, đăng nhập/đăng xuất, cập nhật và chỉnh sửa thông tin cá nhân, thay đổi và lấy lại mật khẩu được thiết kế đơn giản, dễ sử dụng, hỗ trợ đa ngôn ngữ và có các biện pháp xác thực mạnh mẽ để bảo vệ tài khoản.
- **Kiểm tra đạo văn:** Giao diện cho phép người dùng thực hiện kiểm tra tự động và thủ công, cung cấp hướng dẫn chi tiết từng bước, cùng với chức năng nộp tài liệu linh hoạt, hỗ trợ nhiều định dạng file phổ biến.
- **Báo cáo và thống kê:** Các báo cáo kiểm tra đạo văn được trình bày dưới dạng bảng và biểu đồ dễ đọc, phân loại theo lớp học, giảng viên, môn học và thời gian. Tính năng xuất dữ liệu sang các định dạng PDF, Excel và CSV đáp ứng nhu cầu lưu trữ và phân tích.

- **Quản lý thông báo:** Giao diện cho phép gửi, xem và quản lý thông báo, đánh dấu đã đọc, cấu hình nhắc nhở và sự kiện, giúp người dùng chủ động theo dõi các cập nhật quan trọng.
- **Quản lý lớp học và tài liệu:** Tính năng xem danh sách lớp học, học sinh, giảng viên; tìm kiếm, lọc và quản lý tài liệu với các thao tác thêm, sửa, xóa; xếp hạng và đánh giá tài liệu giúp tổ chức công việc thuận tiện và khoa học.
- **Phản hồi và khiếu nại:** Người dùng dễ dàng gửi phản hồi và khiếu nại trực tiếp qua giao diện, hỗ trợ cải tiến dịch vụ và xử lý các vấn đề phát sinh kịp thời.
- **Phân quyền truy cập:** Quản lý chi tiết quyền và vai trò người dùng với giao diện trực quan, đảm bảo kiểm soát truy cập linh hoạt theo nhu cầu tổ chức.
- **Thiết kế UX/UI hiện đại:** Giao diện được xây dựng dựa trên các nguyên tắc thiết kế thân thiện với người dùng, đảm bảo tính tương tác, dễ hiểu và dễ thao tác, phù hợp với nhiều đối tượng sử dụng từ sinh viên đến giảng viên và cán bộ quản lý.
- **Tương thích đa thiết bị:** Hệ thống hỗ trợ truy cập từ máy tính để bàn, laptop, máy tính bảng và điện thoại thông minh, tạo điều kiện thuận lợi cho người dùng làm việc mọi lúc mọi nơi.

CHƯƠNG 3: TRIỂN KHAI THỬ NGHIỆM VÀ ĐÁNH GIÁ HIỆU QUẢ

3.1 Quy trình triển khai hệ thống



Hình 3.1: Sơ đồ phân cấp 3 lớp tính năng chính

Việc triển khai hệ thống kiểm tra đạo văn toàn diện cho trường đại học được thực hiện qua các bước tuần tự và có tổ chức, nhằm đảm bảo hệ thống vận hành ổn định, đáp ứng đầy đủ các chức năng thiết kế và yêu cầu người dùng.

The screenshot shows a project management interface with a table of tasks. The tasks are organized into a hierarchy under 'Plagiarism Checker'. The first section, 'User Management', includes tasks like 'Sign up', 'Login/Logout', 'Edit personal information', and 'Forgot/Change password', all marked as 'Done'. The second section, 'Access Management', includes 'Access Management' (Done), 'Class Management' (Done), 'Plagiarism Checker' (Open), 'Feedback' (Open), 'Document Management' (Open), 'Data Management' (Open), 'Notification Management' (Open), 'Reports And Statistics' (Open), 'Deploy System' (Open), and 'Prepare Documents' (Open). The tasks are assigned to '7790_Trường Trung Nghĩa' and have various due dates and priorities.

Task name	Status	Assignee	Due	Priority	Tags	Blocked By
▼ User Management	Done	7790_Trường Trung Nghĩa	January 15, 2025	High		
Sign up	Done	7790_Trường Trung Nghĩa	January 15, 2025	High		
Login/Logout	Done	7790_Trường Trung Nghĩa	January 15, 2025	High		
Edit personal information	Done	7790_Trường Trung Nghĩa	January 15, 2025	High		
Forgot/Change password	Done	7790_Trường Trung Nghĩa	January 15, 2025	High		
+ New sub-item						
Access Management	Done	7790_Trường Trung Nghĩa	January 19, 2025	High		
Class Management	Done	7790_Trường Trung Nghĩa	February 1, 2025	High		
Plagiarism Checker	Open	7790_Trường Trung Nghĩa		High		
Feedback	Open	7790_Trường Trung Nghĩa		Medium		
Document Management	Open	7790_Trường Trung Nghĩa		Medium		
Data Management	Open	7790_Trường Trung Nghĩa		Medium		
Notification Management	Open	7790_Trường Trung Nghĩa		Low		
Reports And Statistics	Open	7790_Trường Trung Nghĩa		Low		
Deploy System	Open	7790_Trường Trung Nghĩa		Low		
Prepare Documents	Open	7790_Trường Trung Nghĩa		Low		
+ New task						

Hình 3.2: Sơ đồ công việc thực hiện

Quy trình triển khai bao gồm các giai đoạn chính sau đây:

Thiết lập môi trường và cấu hình hạ tầng:

- Chuẩn bị môi trường phát triển và vận hành: Thiết lập máy chủ, cấu hình các thành phần cần thiết như web server (Nginx), cơ sở dữ liệu MySQL, dịch vụ vector Milvus, và môi trường chạy ứng dụng PHP/Laravel cùng Python/Flask.
- Cài đặt cân bằng tải (Load Balancer): Triển khai Nginx để phân phối tải đều cho các thành phần ứng dụng, đảm bảo hệ thống có khả năng chịu tải và vận hành ổn định khi có lượng người dùng lớn.
- Thiết lập hệ thống bảo mật: Cấu hình các chứng chỉ SSL, kiểm soát truy cập, phân quyền và bảo vệ dữ liệu nhằm đáp ứng tiêu chuẩn an toàn thông tin.

Triển khai các module chức năng chính:

- Quản lý người dùng: Phát triển và triển khai các chức năng đăng ký tài khoản, đăng nhập/đăng xuất, cập nhật thông tin cá nhân và quản lý mật khẩu, đảm bảo trải nghiệm người dùng mượt mà và bảo mật cao.
- Quản lý lớp học và phân quyền: Xây dựng các công cụ quản lý danh sách lớp học, giảng viên, sinh viên, cùng chức năng phân quyền truy cập theo vai trò, đáp ứng yêu cầu tổ chức và vận hành hệ thống.

- Xây dựng dịch vụ kiểm tra đạo văn: Phát triển dịch vụ Python/Flask thực hiện tiền xử lý văn bản, chuyển đổi embedding, và truy vấn Milvus nhằm phát hiện nội dung trùng lặp hoặc đạo văn. Cấu hình các thuật toán so sánh văn bản và tích hợp với ứng dụng chính qua API.
- Báo cáo và thống kê: Tích hợp các công cụ tổng hợp, phân tích và xuất báo cáo kiểm tra đạo văn theo lớp, giảng viên, môn học và thời gian dưới nhiều định dạng (PDF, Excel, CSV).
- Quản lý thông báo và phản hồi: Thiết lập hệ thống gửi, xem và quản lý thông báo, sự kiện, nhắc nhở cùng tính năng phản hồi và khiếu nại từ người dùng nhằm nâng cao tính tương tác và hỗ trợ.
- Quản lý tài liệu và dữ liệu tham khảo: Phát triển các chức năng thêm, sửa, xóa tài liệu, phân loại, kiểm tra file trùng lặp, hỗ trợ tìm kiếm hiệu quả và đánh giá tài liệu nộp.

Kiểm thử và tối ưu:

- Kiểm thử chức năng: Thực hiện các bài kiểm tra đơn vị, tích hợp và hệ thống nhằm đảm bảo tất cả các tính năng hoạt động đúng theo yêu cầu thiết kế, không phát sinh lỗi nghiêm trọng.
- Kiểm thử hiệu năng: Đánh giá khả năng chịu tải, thời gian phản hồi và độ ổn định của hệ thống dưới các mức tải khác nhau, đồng thời tối ưu cấu hình server, cân bằng tải và truy vấn cơ sở dữ liệu.
- Kiểm thử bảo mật: Thực hiện đánh giá bảo mật, rà soát các lỗ hổng tiềm ẩn, kiểm tra phân quyền và bảo vệ dữ liệu, đảm bảo hệ thống tuân thủ các chuẩn an toàn thông tin.

Quy trình triển khai được tổ chức khoa học nhằm đảm bảo hệ thống kiểm tra đạo văn hoạt động hiệu quả, đáp ứng toàn diện các yêu cầu kỹ thuật và nghiệp vụ, tạo nền tảng vững chắc cho việc duy trì và phát triển lâu dài.

3.2 Xây dựng cơ sở dữ liệu

Bảng 3.1: Diễn giải các thực thể

STT	Tên bảng	Diễn giải
1	users	Quản lý thông tin tài khoản người dùng gồm email, họ tên, ngày sinh, vai trò admin, đăng nhập...
2	students	Đại diện cho sinh viên, liên kết với users, có mã sinh viên và ngày nhập học.
3	teachers	Đại diện cho giảng viên, liên kết với users, có ngày bắt đầu công tác.
4	subjects	Danh mục môn học với mã và tên môn, mô tả chi tiết.
5	classes	Thông tin lớp học (môn học, giảng viên, tên lớp, phòng, thời gian bắt đầu/kết thúc).
6	assignments	Gán giảng viên cho lớp học vào một ngày cụ thể.
7	enrollments	Quản lý việc sinh viên ghi danh vào lớp học cùng ngày ghi danh.
8	documents	Tài liệu được tải lên hệ thống, liên kết với lớp học, môn học, người tải lên.
9	document_batches	Nhóm các tài liệu theo đợt tải lên, lưu thông tin trạng thái và metadata.
10	media	Quản lý file đa phương tiện như ảnh, tài liệu kèm theo metadata (kích thước, loại, mô tả...).
11	plagiarism_checks	Kết quả kiểm tra đạo văn, liên kết đến người dùng và tài liệu, lưu điểm giống nhau, metadata.
12	permissions	Các quyền hệ thống, dùng cho phân quyền

		truy cập.
13	roles	Vai trò người dùng (Admin, Giáo viên...), dùng cho phân quyền.
14	model_has_roles	Bảng trung gian ánh xạ người dùng với vai trò.
15	model_has_permissions	Bảng trung gian ánh xạ người dùng với quyền cụ thể.
16	role_has_permissions	Gán quyền cho vai trò nhất định.
17	personal_access_tokens	Quản lý token truy cập cho người dùng (ví dụ API token).
18	password_reset_tokens	Lưu token khôi phục mật khẩu qua email.
19	migrations	Lưu trạng thái các migration trong quá trình cập nhật DB.
20	jobs	Hàng đợi xử lý công việc bất đồng bộ.
21	job_batches	Nhóm các jobs, theo batch cho xử lý hàng loạt.
22	failed_jobs	Lưu lại các job lỗi trong quá trình xử lý hàng đợi.

Nguồn: Tác giả tự tổng hợp

Sơ đồ cơ sở dữ liệu:



Hình 3.3: Cơ sở dữ liệu hệ thống

sentence_id	document_id	subject_code	original_name	embedding	raw_text
458668179868799621	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.11395452171564182, 0.8102183663...	Clin Case Rep. 2024;12:e9
458668179868799622	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.07596778836258386, 0.0023863248...	wileyonlinelibrary.com/jo
458668179868799623	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.832889846712358845, 0.812698874...	Received: 28 April 2024
458668179868799624	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.08534861980721924, 0.832679115...	DOI: 10.1002/ccr3.9194
458668179868799625	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.11883842945898877, 0.8482987351...	C A S E R E P O R T
458668179868799626	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.02521231397986412, 0.11212294548...	A 64-year-old male with p
458668179868799627	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.83962433440889226, 0.89083380883...	Hasan Haydar1 . Mouhamm
458668179868799628	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.0155897177583267, 0.0037253071...	This is an open access ar
458668179868799629	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.057492323219776154, 0.038872846...	Clinical Case Reports pub
458668179868799630	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.854628808608767136, 0.0175470281...	Hasan Haydar, Mouhammed S
458668179868799631	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.058049930213025226, 0.058665585...	All contributed equally i
458668179868799632	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.029466206301612854, 0.1133820222...	1Faculty of Medicine, Ham
458668179868799633	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.000912338493338956, 0.1156697777...	2Urology Department, Hama
458668179868799634	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.85647830472755432, 0.07895476371...	3Pathology Department, Ha
458668179868799635	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[0.054851993918418084, 0.0295487474...	4Rheumatology Department,
458668179868799636	1	CNTT	02/CCR3-12-e9194.PMC11259510.pdf	[-0.0007571366732888755, 0.13927284...	Correspondence Hasan Hayd

Hình 3.4: Cơ sở dữ liệu vector embedding

Chi tiết các bảng dữ liệu:

Bảng 3.2: Thực thể assignments

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	bigint	Khóa chính, tự tăng.
<u>class_id</u>	bigint	Liên kết đến bảng classes.
<u>teacher_id</u>	bigint	Liên kết đến bảng teachers.
assignment_date	date	Ngày phân công giảng viên cho lớp học.
created_at	timestamp	Ngày tạo bản ghi.
updated_at	timestamp	Ngày cập nhật bản ghi.
deleted_at	timestamp	Ngày xóa mềm bản ghi (nếu có).

Nguồn: Tác giả tự tổng hợp

Bảng 3.3: Thực thể classes

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	bigint	Khóa chính, tự tăng.
<u>subject_id</u>	bigint	Liên kết đến bảng subjects.

<u>teacher_id</u>	bigint	Liên kết đến bảng teachers.
name	varchar(100)	Tên lớp học.
room_number	varchar(20)	Số phòng học.
start_date	date	Ngày bắt đầu lớp học.
end_date	date	Ngày kết thúc lớp học.
created_at	timestamp	Ngày tạo bản ghi.
updated_at	timestamp	Ngày cập nhật bản ghi.
deleted_at	timestamp	Ngày xóa mềm bản ghi (nếu có).

Nguồn: Tác giả tự tổng hợp

Bảng 3.4: Thực thể document_batches

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	bigint	Khóa chính, tự tăng.
<u>media_id</u>	bigint	Liên kết đến bảng media.
<u>media_path_id</u>	bigint	Đường dẫn media liên quan.
status	varchar(255)	Trạng thái xử lý, mặc định là pending.
metadata	json	Dữ liệu bổ sung.
created_at	timestamp	Ngày tạo bản ghi.
updated_at	timestamp	Ngày cập nhật bản ghi.

Nguồn: Tác giả tự tổng hợp

Bảng 3.5: Thực thể documents

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	bigint	Khóa chính, tự tăng.
<u>class_id</u>	bigint	Liên kết đến bảng classes, có thể NULL.
<u>subject_id</u>	bigint	Liên kết đến bảng subjects.
<u>uploaded_by</u>	bigint	Người tải lên, liên kết users.

<u>media_id</u>	bigint	Liên kết đến bảng media, có thể NULL.
<u>batch_id</u>	bigint	Liên kết đến bảng document_batches, có thể NULL.
status	varchar(255)	Trạng thái tài liệu, mặc định pending.
original_name	varchar(255)	Tên gốc của tài liệu.
description	text	Mô tả tài liệu.
metadata	json	Dữ liệu bổ sung.
created_at	timestamp	Ngày tạo bản ghi.
updated_at	timestamp	Ngày cập nhật bản ghi.
deleted_at	timestamp	Ngày xóa mềm bản ghi (nếu có).

Nguồn: Tác giả tự tổng hợp

Bảng 3.6: Thực thể enrollments

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
id	bigint	Khóa chính, tự tăng.
student_id	bigint	Liên kết đến bảng students.
class_id	bigint	Liên kết đến bảng classes.
enrollment_date	date	Ngày ghi danh lớp học.
created_at	timestamp	Ngày tạo bản ghi.
updated_at	timestamp	Ngày cập nhật bản ghi.
deleted_at	timestamp	Ngày xóa mềm bản ghi (nếu có).

Nguồn: Tác giả tự tổng hợp

Bảng 3.7: Thực thể failed_jobs

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
id	bigint	Khóa chính, tự tăng.
uuid	varchar(255)	Mã định danh duy nhất của job.

connection	text	Chuỗi kết nối đến hàng đợi.
queue	text	Tên hàng đợi xử lý job.
payload	longtext	Nội dung công việc cần xử lý.
exception	longtext	Nội dung lỗi khi job thất bại.
failed_at	timestamp	Thời điểm xảy ra lỗi, mặc định là thời gian hiện tại.

Nguồn: Tác giả tự tổng hợp

Bảng 3.8: Thực thể *job_batches*

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	varchar(255)	Khóa chính, định danh duy nhất cho mỗi batch.
name	varchar(255)	Tên của batch.
total_jobs	int	Tổng số job trong batch.
pending_jobs	int	Số job đang chờ xử lý.
failed_jobs	int	Số job bị lỗi.
failed_job_ids	longtext	Danh sách ID của các job lỗi.
options	mediumtext	Tuỳ chọn bổ sung.
cancelled_at	int	Thời điểm hủy batch (timestamp, có thể NULL).
created_at	int	Thời điểm tạo batch.
finished_at	int	Thời điểm hoàn thành batch (có thể NULL).

Nguồn: Tác giả tự tổng hợp

Bảng 3.9: Thực thể *jobs*

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	bigint	Khóa chính, tự tăng.

queue	varchar(255)	Tên hàng đợi xử lý.
payload	longtext	Nội dung xử lý của job.
attempts	tinyint	Số lần thử thực hiện job.
reserved_at	int	Thời điểm job được giữ (có thể NULL).
available_at	int	Thời điểm job sẵn sàng thực thi.
created_at	int	Thời điểm tạo job.

Nguồn: Tác giả tự tổng hợp

Bảng 3.10: Thực thể media

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	bigint	Khóa chính, tự tăng.
disk	varchar(255)	Tên ổ đĩa lưu trữ (mặc định: public).
directory	varchar(255)	Thư mục chứa file (mặc định: media).
visibility	varchar(255)	Quyền truy cập (mặc định: public).
name	varchar(255)	Tên file.
path	varchar(255)	Đường dẫn tới file.
width	int	Chiều rộng (nếu là ảnh).
height	int	Chiều cao (nếu là ảnh).
size	int	Kích thước file.
type	varchar(255)	Loại file (mặc định: image).
ext	varchar(255)	Phần mở rộng của file.
alt	varchar(255)	Văn bản thay thế cho hình ảnh.
title	varchar(255)	Tiêu đề của file.
description	text	Mô tả file.
caption	text	Chú thích hình ảnh.

exif	text	Dữ liệu EXIF cho ảnh.
curations	longtext	Dữ liệu chọn lọc xử lý (nếu có).
created_at	timestamp	Thời điểm tạo file.
updated_at	timestamp	Thời điểm cập nhật file.

Nguồn: Tác giả tự tổng hợp

Bảng 3.11: Thực thể migrations

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>id</u>	int	Khóa chính, tự tăng.
migration	varchar(255)	Tên migration đã chạy.
batch	int	Số hiệu batch của migration.

Nguồn: Tác giả tự tổng hợp

Bảng 3.12: Thực thể model_has_permissions

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>permission_id</u>	bigint	Liên kết đến bảng permissions.
model_type	varchar(255)	Loại mô hình (User, Admin, etc.).
<u>model_id</u>	bigint	ID của mô hình được gán quyền.

Nguồn: Tác giả tự tổng hợp

Bảng 3.13: Thực thể model_has_roles

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
<u>role_id</u>	bigint	Liên kết đến bảng roles.
model_type	varchar(255)	Loại mô hình (User, Admin, etc.).
<u>model_id</u>	bigint	ID của mô hình được gán vai trò.

Nguồn: Tác giả tự tổng hợp

Bảng 3.14: Thực thể password_reset_tokens

Tên thuộc tính	Kiểu dữ liệu	Ghi chú
----------------	--------------	---------

email	varchar(255)	Email người dùng (Khóa chính).
token	varchar(255)	Mã token reset mật khẩu.
created_at	timestamp	Thời điểm tạo token.

Nguồn: Tác giả tự tổng hợp

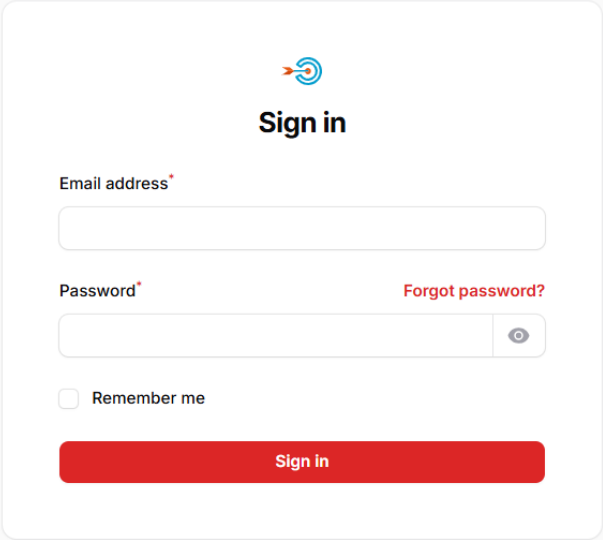
3.3 Giao diện chương trình

Giao diện trang đăng kí, đăng nhập, trang cá nhân:

The image shows a 'Sign in' form with the following elements:

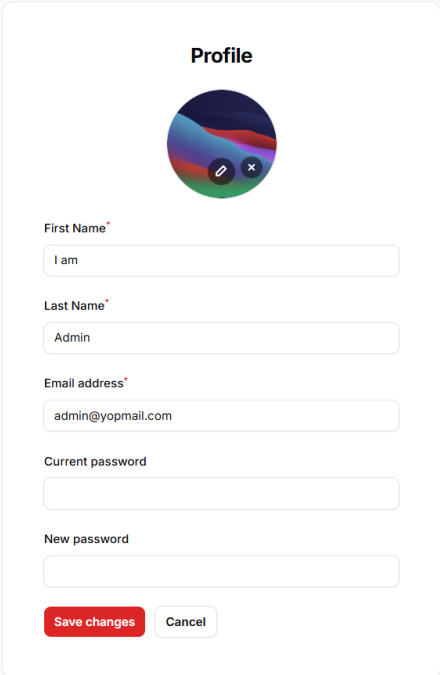
- Logo: A blue and orange circular icon with a stylized 'e'.
- Text: 'Sign in' in bold black font, followed by 'or [sign up for an account](#)' in smaller blue font.
- Form Fields:
 - 'Email address*' with a red asterisk, followed by a white input box.
 - 'Password*' with a red asterisk, followed by a white input box and a toggle icon (an eye inside a circle).
- Link: '[Forgot password?](#)' in blue text to the right of the password field.
- Checkbox: '☐ Remember me' below the password field.
- Button: A solid blue button labeled 'Sign in' at the bottom.

Hình 3.5: Giao diện trang đăng nhập cho Giảng Viên và Sinh Viên



The image shows a 'Sign in' form for an Admin user. At the top, there is a logo consisting of a blue circle with a white arrow pointing right. Below the logo, the text 'Sign in' is displayed in a bold, black font. The form contains two input fields: 'Email address*' and 'Password*'. The 'Email address*' field is a simple text box. The 'Password*' field is a text box with a small eye icon on the right side to toggle visibility. To the right of the password field, there is a link that says 'Forgot password?'. Below the password field, there is a checkbox labeled 'Remember me'. At the bottom of the form, there is a red button with the text 'Sign in' in white.

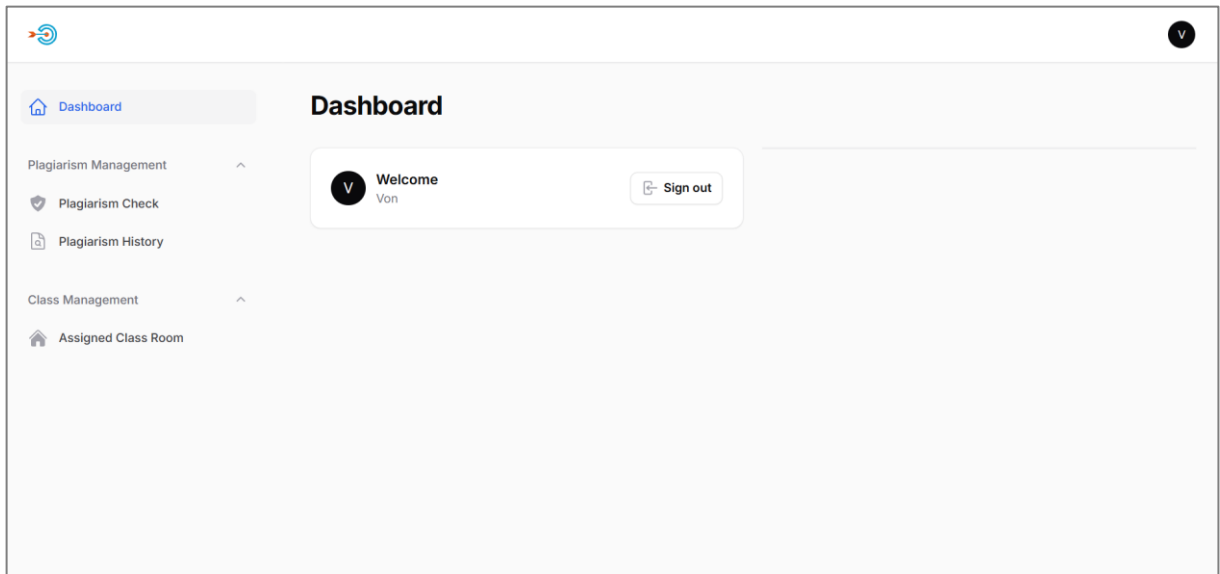
Hình 3.6: Giao diện trang đăng nhập cho Quản Trị Viên



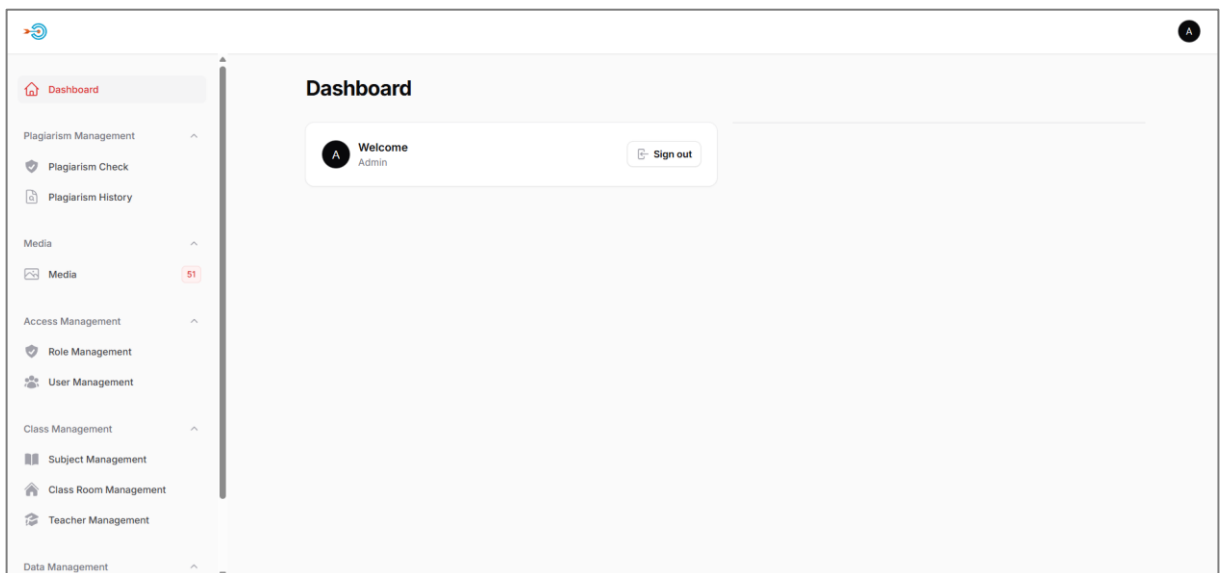
The image shows a 'Profile' form for an Admin user. At the top, there is a circular profile picture placeholder with a blue and green background and a small 'x' icon. Below the profile picture, the text 'Profile' is displayed in a bold, black font. The form contains five input fields: 'First Name*', 'Last Name*', 'Email address*', 'Current password', and 'New password'. The 'First Name*' field contains the text 'I am'. The 'Last Name*' field contains the text 'Admin'. The 'Email address*' field contains the text 'admin@yopmail.com'. Below the 'Current password' field, there is a 'New password' field. At the bottom of the form, there are two buttons: a red button with the text 'Save changes' and a white button with the text 'Cancel'.

Hình 3.7: Giao diện trang cá nhân

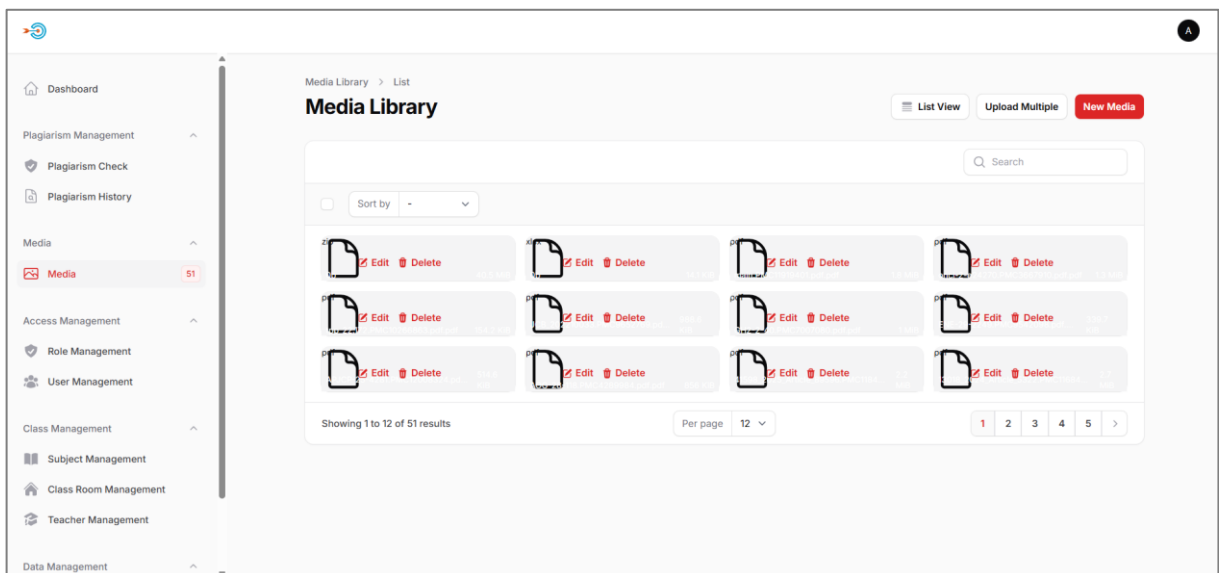
Giao diện trang chủ và quản trị:



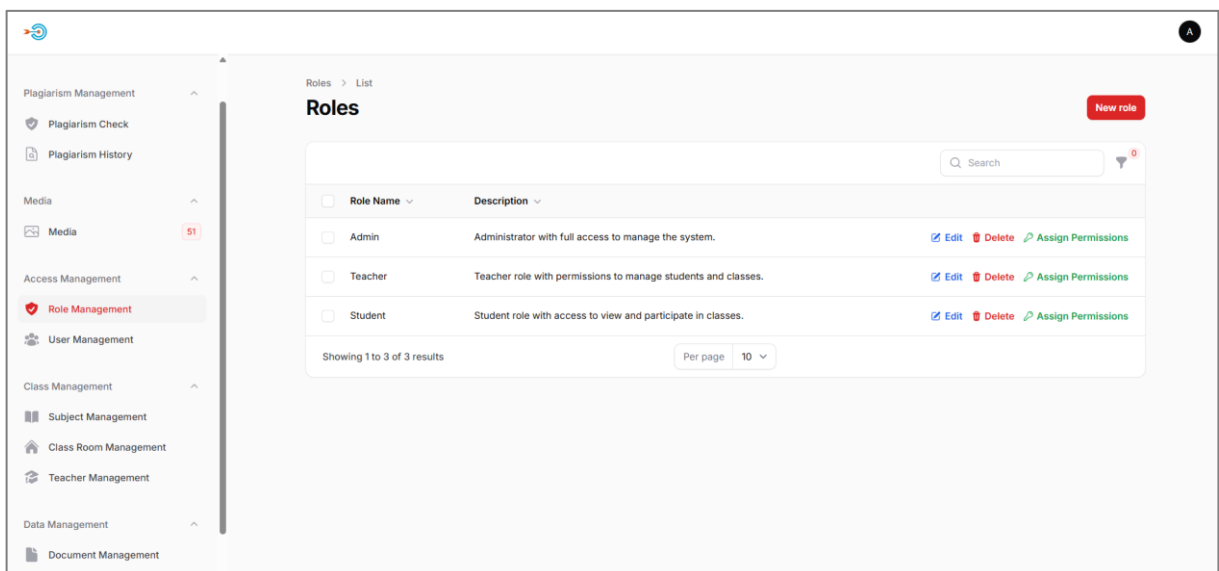
Hình 3.8: Giao diện trang chủ cho Giáo Viên và Sinh Viên



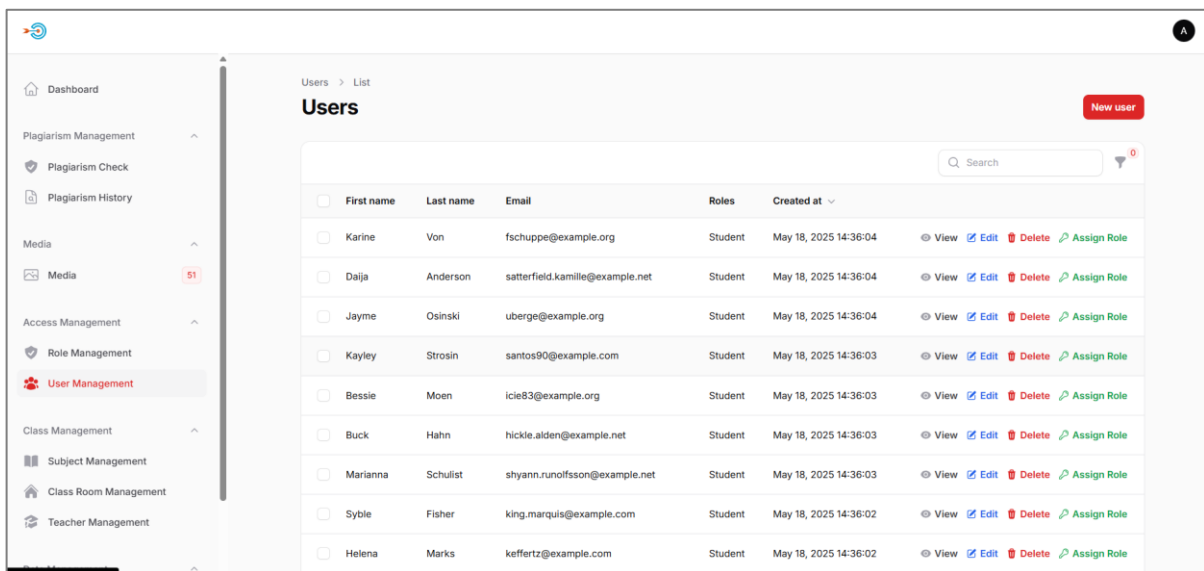
Hình 3.9: Giao diện trang chủ cho Quản Trị Viên



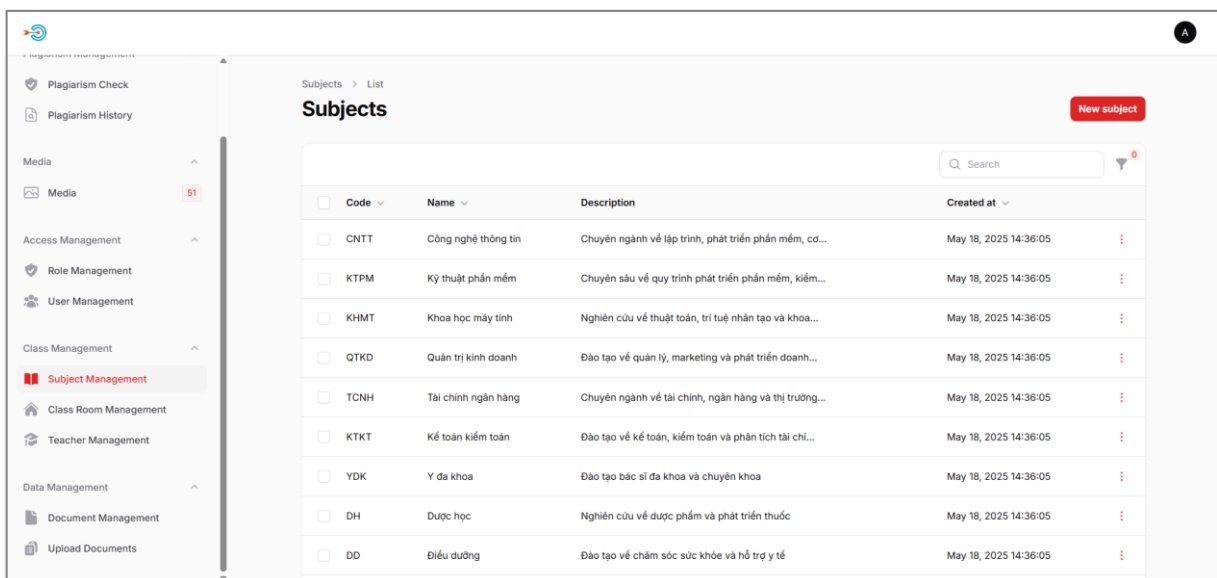
Hình 3.10: Giao diện trang quản lý phương tiện truyền thông



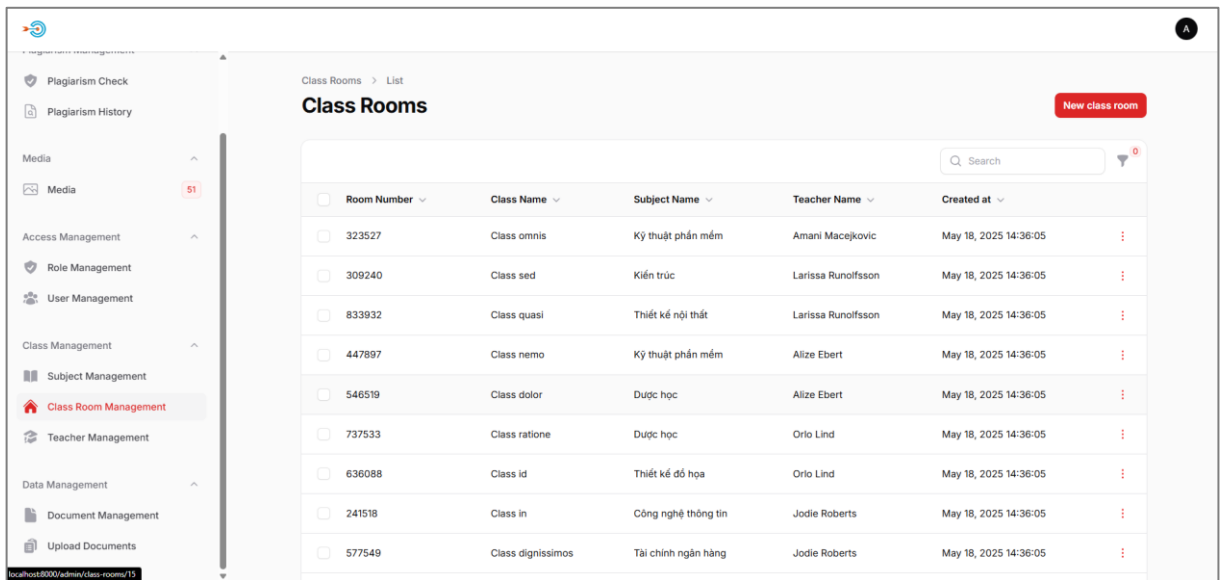
Hình 3.11: Giao diện trang quản lý quyền truy cập



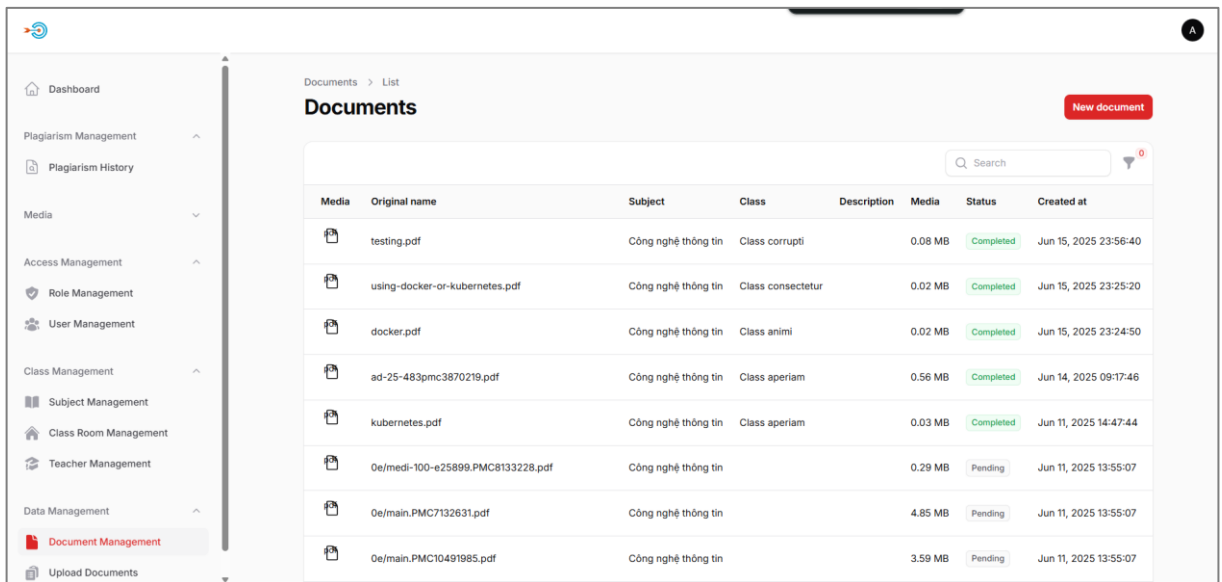
Hình 3.12: Giao diện trang quản lí người dùng



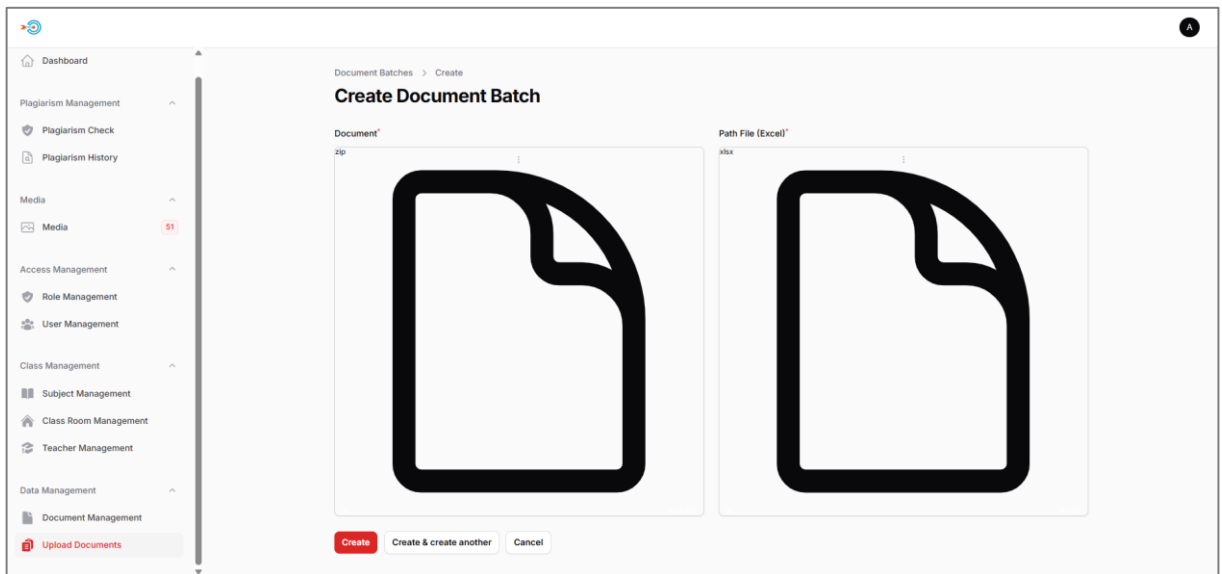
Hình 3.13: Giao diện trang quản lí chuyên ngành



Hình 3.14: Giao diện trang quản lý lớp học

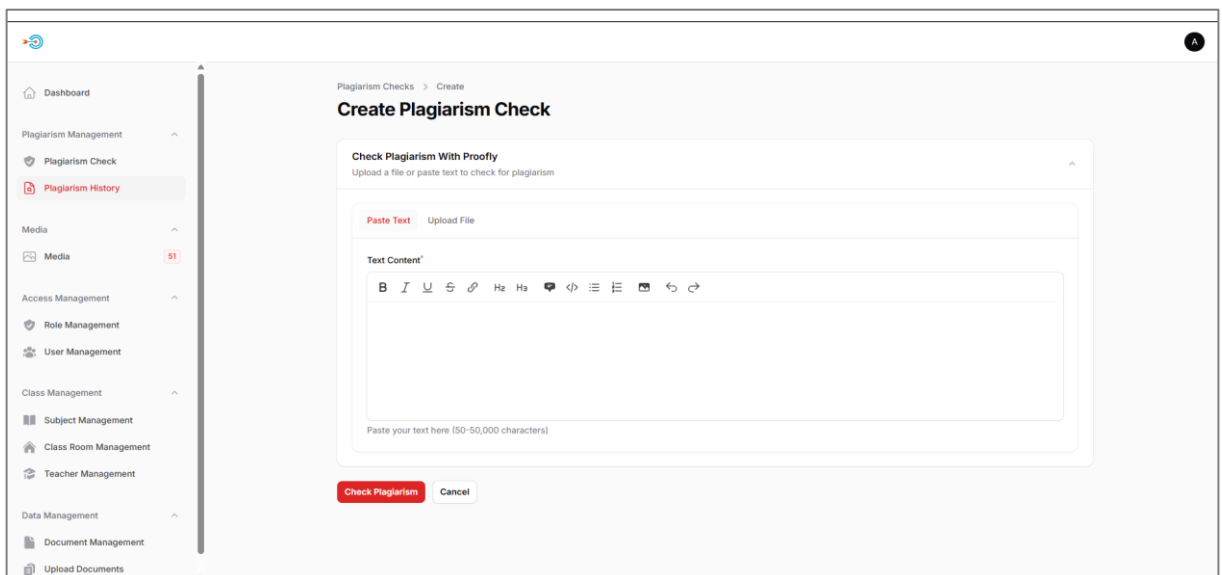


Hình 3.15: Giao diện trang quản lý và theo dõi tài liệu tải lên

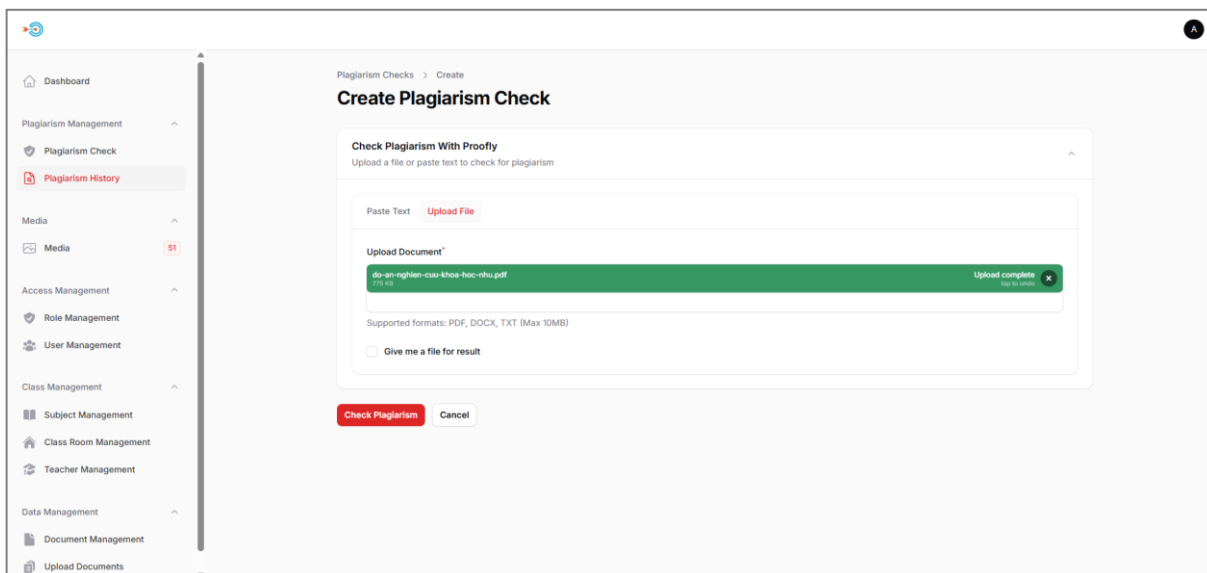


Hình 3.16: Giao diện trang tải lên và lưu trữ tài liệu

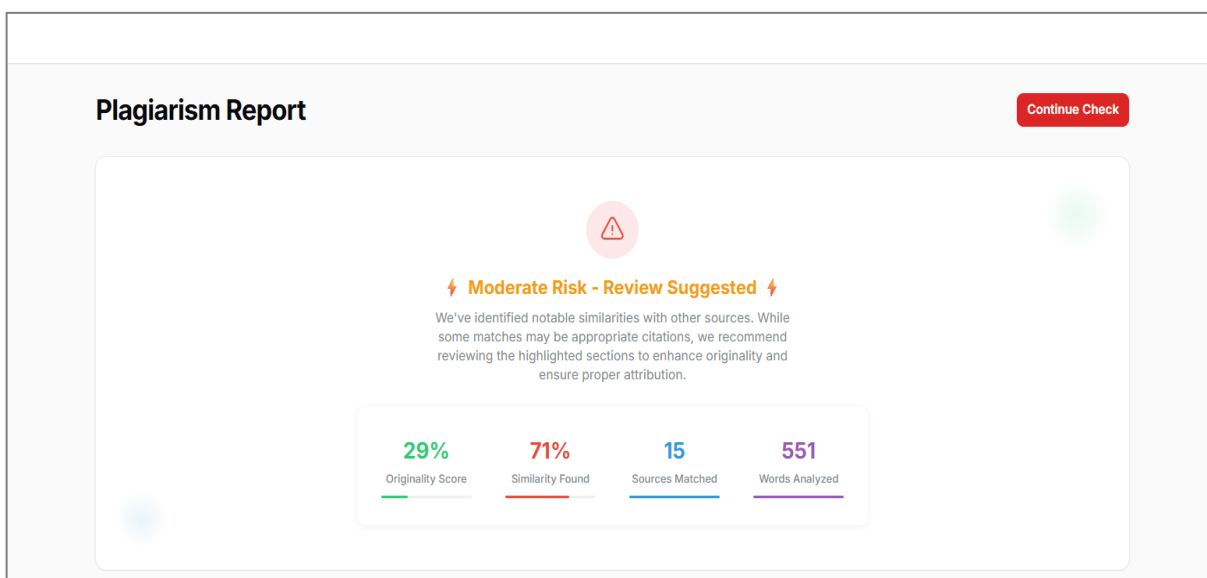
Giao diện trang kiểm tra đạo văn:



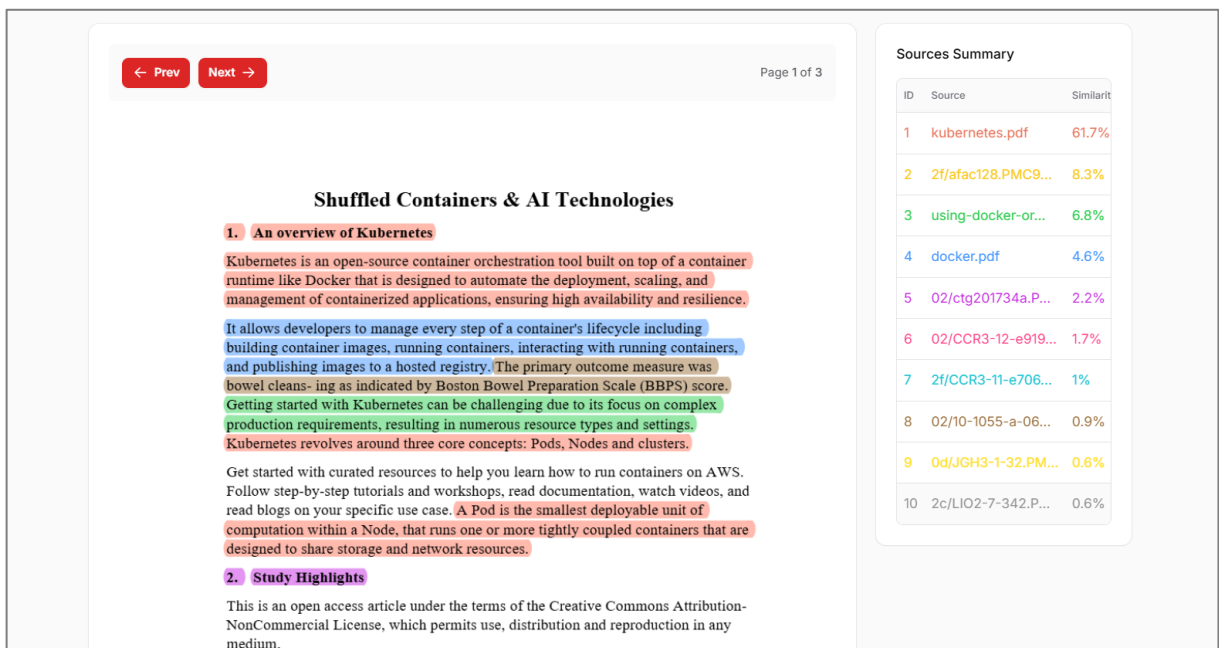
Hình 3.17: Giao diện trang kiểm tra đạo văn bằng văn bản



Hình 3.18: Giao diện trang kiểm tra đạo văn bằng file



Hình 3.19: Giao diện trang tổng quan kết quả



Hình 3.20: Giao diện trang hiển thị kết quả kiểm tra

3.4 Thiết lập bộ dữ liệu thử nghiệm

Để kiểm tra và đánh giá hiệu quả của hệ thống phát hiện đạo văn, tôi đã tải xuống và triển khai bộ dữ liệu từ PubMed Central (PMC) Open Access Subset với kích thước dung lượng lên đến hơn 12GB.

PMC Open Access Subset là một tập hợp các bài báo khoa học trong lĩnh vực y sinh và khoa học sự sống, được cung cấp miễn phí bởi Thư viện Y học Quốc gia Hoa Kỳ (NLM). Các bài báo trong tập hợp này được phát hành dưới các giấy phép Creative Commons hoặc tương tự, cho phép sử dụng lại và phân phối rộng rãi.

Thư mục chứa các tệp PDF của các bài báo khoa học, mỗi tệp tương ứng với một bài báo riêng biệt. Các bài báo này bao gồm đầy đủ nội dung như tiêu đề, tóm tắt, nội dung chính, hình ảnh, bảng biểu và tài liệu tham khảo.

Bộ dữ liệu này được sử dụng để:

- Đánh giá khả năng của hệ thống trong việc phát hiện đạo văn trong các tài liệu khoa học thực tế.
- Kiểm tra độ chính xác của hệ thống khi xử lý các tài liệu có cấu trúc và định dạng khác nhau.
- Huấn luyện và tinh chỉnh các mô hình học máy để cải thiện hiệu suất phát hiện

đạo văn.

- Kiểm tra và đánh giá tốc độ và độ chính xác trên tập dữ liệu lớn.

Quy trình triển khai

- Tải xuống dữ liệu: Sử dụng công cụ dòng lệnh hoặc trình duyệt để tải xuống tất cả các tệp PDF.
- Tiền xử lý dữ liệu: Chuyển đổi các tệp PDF thành văn bản thuần túy, loại bỏ các yếu tố không cần thiết như hình ảnh, bảng biểu để tập trung vào nội dung văn bản.
- Phân tích và đánh giá: Sử dụng hệ thống phát hiện đạo văn để phân tích các văn bản đã xử lý, xác định các đoạn văn có khả năng bị sao chép hoặc tương tự với các nguồn khác.
- Ghi nhận kết quả: Lưu trữ và phân tích các kết quả thu được để đánh giá hiệu suất của hệ thống, bao gồm các chỉ số như độ chính xác, độ nhạy và độ đặc hiệu.

3.5 Thử nghiệm và đánh giá độ chính xác của hệ thống

Hệ thống đã được xây dựng và vượt qua các bài kiểm thử cấp đơn vị và tính năng một cách chính xác, nhằm đảm bảo mọi tính năng của hệ thống hoạt động trơn tru và mượt mà.

```
PASS Tests\Unit\Models\SubjectTest
✓ subject can be created 0.22s
✓ subject has required attributes 0.04s
✓ subject attributes are mass assignable 0.04s

PASS Tests\Feature\ApplicationTest
✓ the application returns a successful response 0.43s

PASS Tests\Feature\Filament\Resources\SubjectResourceTest
✓ form schema defines required name field with max length 0.30s
✓ form validation rejects empty name field 0.24s
✓ can create subject 0.25s
✓ can view subject list 0.76s
✓ can edit subject 0.38s
✓ can delete subject 1.69s
✓ form validation rejects too long name 0.26s
✓ table shows correct columns 0.61s

Tests: 12 passed (52 assertions)
Duration: 5.32s
```

Hình 3.21: Kiểm thử đơn vị và tính năng

Kiểm thử đơn vị (Unit Test):

- Kiểm thử các tính năng xác thực người dùng.
- Kiểm thử các tính năng CRUD của các modules.

- Kiểm thử các tính năng khả năng upload, xử lý và lưu trữ dữ liệu.
- Kiểm thử tính năng chuyển đổi văn bản thông qua Embedding model.

Kiểm thử thành phần (Component Test):

- Kiểm thử endpoint của hệ thống chính và dịch vụ kiểm tra đạo văn.
- Kiểm thử tương tác giữa các microservices trong hệ thống.
- Kiểm thử toàn bộ quy trình tải lên và thực hiện kiểm tra đạo văn.

Kiểm thử hiệu suất (Performance Test):

- Kiểm thử tính năng chuyển đổi sang embedding trên dữ liệu lớn.
- Kiểm thử tính năng upload tài liệu với kích thước lớn.
- Kiểm thử tính năng tìm kiếm và kiểm tra đạo văn trên lượng lớn vector embedding.

Qua ba giai đoạn kiểm thử chính, hệ thống cho thấy mức độ ổn định và tính sẵn sàng cao cho việc triển khai thực tế. Trong quá trình **kiểm thử đơn vị**, các chức năng cốt lõi như xác thực người dùng, xử lý CRUD, upload và lưu trữ tài liệu, cũng như quá trình chuyển đổi văn bản bằng mô hình embedding đã hoạt động đúng như thiết kế, không phát sinh lỗi nghiêm trọng. Bước sang **kiểm thử thành phần**, hệ thống thể hiện khả năng vận hành liền mạch giữa các dịch vụ con, đảm bảo toàn bộ chuỗi xử lý kiểm tra đạo văn – từ tiếp nhận tài liệu cho tới phản hồi kết quả – diễn ra mạch lạc và chính xác. Cuối cùng, kết quả **kiểm thử hiệu suất** cho thấy hệ thống giữ vững tốc độ và tính ổn định khi xử lý các tài liệu lớn, đặc biệt là trong các thao tác embedding và truy vấn dữ liệu vector với khối lượng cao. Tổng thể, các bài thử nghiệm đã phản ánh khả năng vận hành tin cậy và nhất quán của hệ thống trong nhiều tình huống khác nhau.

Kiểm thử khả năng phát hiện đạo văn:

- Kiểm thử trên văn bản đã được **diễn giải lại** nội dung.
- Kiểm thử trên văn bản đã được **rút gọn** nội dung.
- Kiểm thử trên văn bản đã được **mở rộng** nội dung.
- Kiểm thử trên văn bản đã được **đảo cấu trúc câu** nội dung.

- Kiểm thử trên văn bản đã được **đồng nghĩa** nội dung.
- Kiểm thử trên văn bản đã được **thay đổi thứ tự** nội dung.

Do hạn chế về công cụ, bộ nhớ và thời gian triển khai, hiện tại hệ thống đã được kiểm thử trên một tập dữ liệu mẫu tương đối giới hạn khoảng 12GB, bao gồm các bài báo khoa học từ bộ dữ liệu PubMed Central Open Access Subset. Qua các thử nghiệm ban đầu, hệ thống đã cho thấy khả năng phát hiện các đoạn văn bản trùng lặp và tương đồng ở mức độ ổn định, đặc biệt với các trường hợp đạo văn nguyên văn hoặc gần giống về mặt ngôn ngữ.

Tuy nhiên, độ chính xác tổng thể chưa đạt mức cao tối ưu do một số yếu tố sau:

- Đa dạng về cấu trúc và định dạng tài liệu: Văn bản khoa học có cấu trúc phức tạp với nhiều phần như hình ảnh, bảng biểu, chú thích, gây khó khăn trong quá trình trích xuất và tiền xử lý.
- Khả năng xử lý trên các đoạn văn nhỏ: Mặc dù mô hình embedding MiniLM và cơ sở dữ liệu vector Milvus hỗ trợ kiểm tra trên các đoạn văn nhỏ nhưng cần nhiều thời gian thử nghiệm trên các tham số truy vấn khác nhau để có thể đánh giá được tham số phù hợp.
- Giới hạn kích thước và chất lượng dữ liệu thử nghiệm: Bộ dữ liệu thử nghiệm hiện chưa đa dạng và phong phú đủ để đánh giá toàn diện hiệu suất của hệ thống trong nhiều kịch bản thực tế.

Tốc độ xử lý văn bản và truy vấn vector embedding trong các thử nghiệm sơ bộ đạt hiệu quả ổn định với tốc độ xử lý tính khá tốt, đảm bảo khả năng vận hành liên tục và đáp ứng được yêu cầu của môi trường giáo dục.

3.6 Đánh giá kết quả và cải tiến

Dựa trên kết quả thử nghiệm ban đầu, hệ thống đã xác định được một số điểm mạnh và hạn chế cần tập trung cải tiến:

Điểm mạnh:

- Khả năng phát hiện đạo văn nguyên văn và các đoạn văn bản có nội dung tương đồng đơn giản đạt hiệu quả tốt.

- Kiến trúc microservice cho phép mở rộng và nâng cấp linh hoạt, thuận tiện tích hợp các mô hình mới trong tương lai.
- Hệ thống quản lý và báo cáo chi tiết giúp người dùng dễ dàng theo dõi và đánh giá kết quả kiểm tra.
- Hệ thống cho ra kết quả với tốc độ xử lý và phản hồi khá tốt.

Hạn chế và thách thức:

- Hiệu quả phát hiện paraphrase và đạo văn dạng dịch thuật còn hạn chế trên các trường hợp đạo văn tinh vi về ý tưởng.
- Độ chính xác của văn bản chưa đạt giá trị tối đa vì cần thực hiện đánh giá để xác định tham số phù hợp nhất.
- Cần mở rộng bộ dữ liệu thử nghiệm đa dạng hơn nhằm đánh giá toàn diện và cải thiện độ chính xác mô hình.

Kế hoạch cải tiến:

- Thực hiện xây dựng một mô hình kết hợp giữ kiểm tra tài liệu nội bộ và kiểm tra tài liệu trên internet.
- Thu thập và xây dựng thêm bộ dữ liệu thử nghiệm phong phú, đa dạng về ngôn ngữ và cấu trúc, đặc biệt tập trung vào tài liệu tiếng Việt và các trường hợp paraphrase.
- Nâng cấp bước tiền xử lý văn bản, bao gồm cải thiện khả năng trích xuất, làm sạch và chuẩn hóa dữ liệu đầu vào.
- Cải thiện giao diện báo cáo, bổ sung các chỉ số và trực quan hóa giúp người dùng đánh giá kết quả chính xác và nhanh chóng hơn.
- Đẩy mạnh tích hợp và tự động hóa trong quy trình kiểm tra để giảm thiểu thao tác thủ công và tăng độ tin cậy hệ thống.

CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ

4.1 Kết luận về kết quả đạt được

Đề tài đã xây dựng thành công một hệ thống kiểm tra đạo văn toàn diện dành cho trường đại học, đáp ứng đầy đủ các yêu cầu quản lý người dùng, kiểm tra tự động và thủ công, báo cáo và thống kê, quản lý tài liệu và thông báo. Hệ thống sử dụng mô hình embedding MiniLM phối hợp với cơ sở dữ liệu vector Milvus để tăng hiệu quả phát hiện các trường hợp đạo văn, bao gồm cả các dạng paraphrase và dịch thuật văn bản.

Qua thử nghiệm sơ bộ với bộ dữ liệu từ PubMed Central, hệ thống đã thể hiện khả năng phát hiện nội dung trùng lặp với độ chính xác ở mức khá ổn định, đồng thời duy trì hiệu năng xử lý phù hợp với quy mô triển khai trong môi trường giáo dục đại học. Kiến trúc microservice linh hoạt và thiết kế giao diện thân thiện góp phần nâng cao trải nghiệm người dùng và thuận tiện trong vận hành, bảo trì.

4.2 Những đóng góp của đề tài

Phát triển hệ thống nội địa: Tạo ra giải pháp kiểm tra đạo văn phù hợp với ngôn ngữ tiếng Việt và yêu cầu thực tiễn của các trường đại học Việt Nam, giảm phụ thuộc vào các phần mềm nước ngoài có chi phí cao và hạn chế về ngôn ngữ.

Áp dụng công nghệ vector embedding đa ngôn ngữ: Sử dụng mô hình MiniLM và hệ quản trị vector Milvus giúp cải thiện khả năng phát hiện đạo văn phức tạp vượt trội hơn các phương pháp dựa trên so sánh chuỗi truyền thống.

Kiến trúc phần mềm hiện đại: Thiết kế microservice kết hợp các mẫu thiết kế Factory và Strategy nhằm nâng cao khả năng mở rộng, bảo trì và dễ dàng tích hợp các công nghệ mới trong tương lai.

Tích hợp chức năng quản lý toàn diện: Hệ thống không chỉ hỗ trợ kiểm tra đạo văn mà còn tích hợp các module quản lý người dùng, lớp học, thông báo, phản hồi và báo cáo thống kê, tạo nên nền tảng dịch vụ hoàn chỉnh.

4.3 Đề xuất hướng nghiên cứu tiếp theo

Phát triển mô hình hybrid: Kết hợp dữ liệu nội bộ (luận văn, đề tài nghiên cứu, bài tập của sinh viên) với nguồn dữ liệu Internet và kho tài liệu học thuật quốc tế nhằm tăng khả năng phát hiện đạo văn toàn diện hơn.

Ứng dụng trí tuệ nhân tạo kiểm tra ngữ nghĩa sâu: Nghiên cứu và tích hợp các mô hình AI nâng cao, như transformer đa ngôn ngữ, để nhận diện paraphrase tinh vi, đạo văn ý tưởng và dịch thuật không minh bạch.

Tối ưu hóa xử lý đa ngôn ngữ: Mở rộng hỗ trợ và nâng cao hiệu quả xử lý tiếng Việt và các ngôn ngữ khác trong khu vực nhằm phục vụ môi trường đào tạo đa dạng.

Nâng cao trải nghiệm người dùng và tự động hóa: Phát triển các chức năng phản hồi thông minh, phân tích hành vi người dùng, tự động cảnh báo và đề xuất biện pháp xử lý vi phạm.

Xây dựng hệ sinh thái kiểm tra đạo văn: Liên kết hệ thống với các công cụ hỗ trợ viết bài, tham khảo tài liệu, giúp sinh viên và giảng viên chủ động trong việc nâng cao chất lượng nghiên cứu và học tập.

TÀI LIỆU THAM KHẢO

- [1]. What is Milvus,
<https://milvus.io/docs/overview.md>, truy cập ngày 12/05/2025.
- [2]. HNSW,
<https://milvus.io/docs/hnsw.md#HNSW>, truy cập ngày 12/05/2025.
- [3]. all-MiniLM-L12-v2 | HuggingFace,
<https://docs.pinecone.io/models/all-MiniLM-L12-v2>, truy cập ngày 13/05/2025.
- [4]. Turnitin,
<https://en.wikipedia.org/wiki/Turnitin#:~:text=Founded%20in%201998%2C%20it%20sells,76>, truy cập ngày 12/05/2025.
- [5]. About Copyscape,
<https://www.copyscape.com/about.php#:~:text=Copyscape%20is%20provided%20by%20Indigo,terms%20and%20conditions%20of%20use>, truy cập ngày 12/05/2025.
- [6]. Copyscape Plagiarism Checker Review: Is It Any Good,
<https://www.tinylevermarketing.com/blog/what-is-copyscape-a-complete-guide-to-the-tool#:~:text=What%20is%20Copyscape%3F>, truy cập ngày 12/05/2025.
- [7]. What is Copyscape? A Complete Guide to the Tool,
<https://www.copyscape.com/about.php#:~:text=Copyscape%20is%20provided%20by%20Indigo,terms%20and%20conditions%20of%20use>, truy cập ngày 12/05/2025.
- [8]. Gharavi, E., Veisi, H., Bijari, K., & Zahirnia, K. (2020). A Fast Multi-level Plagiarism Detection Method Based on Document Embedding Representation. *Neural Computing and Applications*, 32(12), 9033–9046.
- [9]. Moravvej, S. V., Mousavirad, S. J., Oliva, D., & Mohammadi, F. (2023). A Novel Plagiarism Detection Approach Combining BERT-based Word Embedding, Attention-based LSTMs and an Improved Differential Evolution Algorithm. *arXiv preprint May*, 1–20.

- [10]. Ferrero, J., Agnès, F., Besacier, L., & Schwab, D. (2017). Using Word Embedding for Cross-Language Plagiarism Detection. *International Journal of Advanced Computer Science and Applications*, 11(2), 232–243.
- [11]. Mazumder Setu, D., Islam, T., Dey, S. K., Al Asif, M. R., & Samsuddoha, M. (2025). A comprehensive strategy for identifying plagiarism in academic submissions. *Journal of Umm Al-Qura University for Engineering and Architecture*, 16, 310–325.

PHỤ LỤC

Phụ lục 1: Hướng dẫn sử dụng

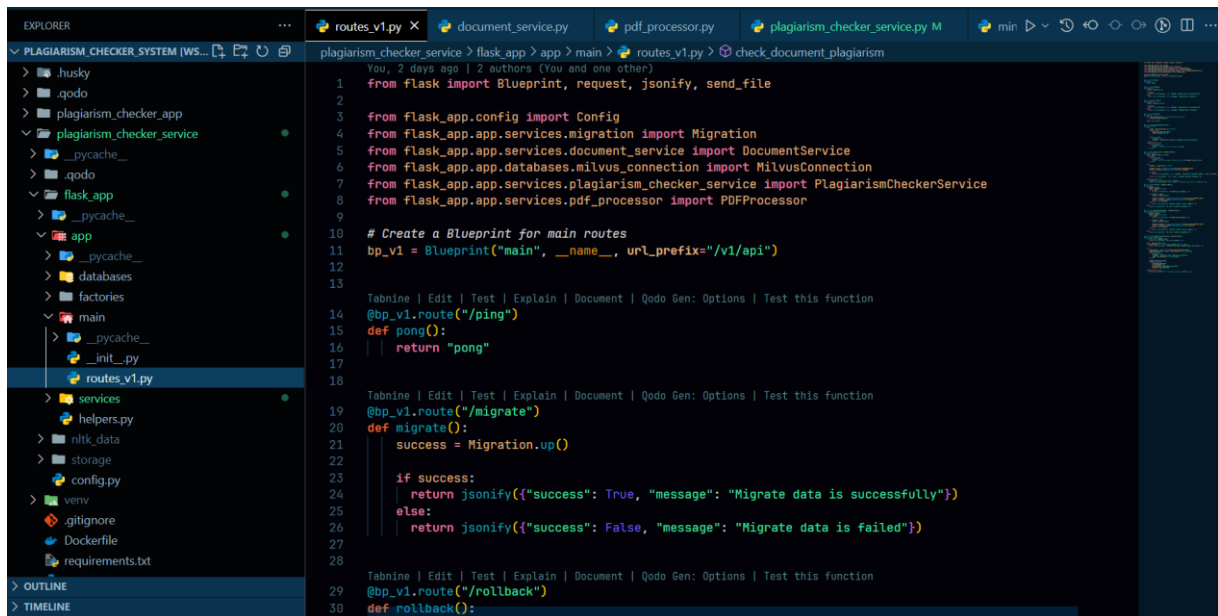
Ứng dụng gồm các chức năng:

- **Đăng ký tài khoản:** Người dùng truy cập trang đăng ký, điền đầy đủ thông tin cá nhân theo mẫu yêu cầu và xác nhận đăng ký qua email (nếu có).
- **Đăng nhập/Đăng xuất:** Sau khi đăng ký thành công, người dùng đăng nhập với tên tài khoản và mật khẩu được cấp. Chức năng đăng xuất cho phép kết thúc phiên làm việc an toàn.
- **Cập nhật và chỉnh sửa thông tin cá nhân:** Người dùng có thể truy cập trang quản lý hồ sơ để cập nhật thông tin cá nhân như tên, email, số điện thoại, và các dữ liệu liên quan.
- **Thay đổi/Lấy lại mật khẩu:** Hệ thống hỗ trợ thay đổi mật khẩu định kỳ và cung cấp tính năng lấy lại mật khẩu khi quên qua email hoặc câu hỏi bảo mật.
- **Kiểm tra tự động:** Người dùng tải lên tài liệu (hỗ trợ các định dạng DOCX, PDF, TXT) thông qua giao diện kiểm tra tự động. Hệ thống sẽ tiến hành xử lý, so sánh và trả về kết quả đạo văn dưới dạng báo cáo chi tiết.
- **Kiểm tra thủ công:** Giảng viên hoặc cán bộ quản lý có thể thực hiện kiểm tra thủ công với các đoạn văn bản hoặc tài liệu cụ thể nhằm đánh giá chi tiết hơn.
- **Báo cáo kiểm tra đạo văn:** Hệ thống cung cấp báo cáo theo các phân nhóm như lớp học, giảng viên, môn học, và thời gian, giúp đánh giá tổng quan tình trạng đạo văn.
- **Thống kê đạo văn:** Tổng hợp các chỉ số tổng số vi phạm, phân bố theo môn học và thời gian cụ thể để hỗ trợ công tác quản lý.
- **Xuất dữ liệu:** Người dùng có thể xuất báo cáo và thống kê dưới định dạng PDF, Excel, hoặc CSV để lưu trữ hoặc phục vụ phân tích sâu hơn.
- **Gửi và xem thông báo:** Quản trị viên gửi thông báo cho người dùng; người dùng có thể xem và đánh dấu đã đọc toàn bộ thông báo.
- **Cấu hình thông báo và nhắc nhở:** Thiết lập sự kiện, lịch nhắc nhở giúp người

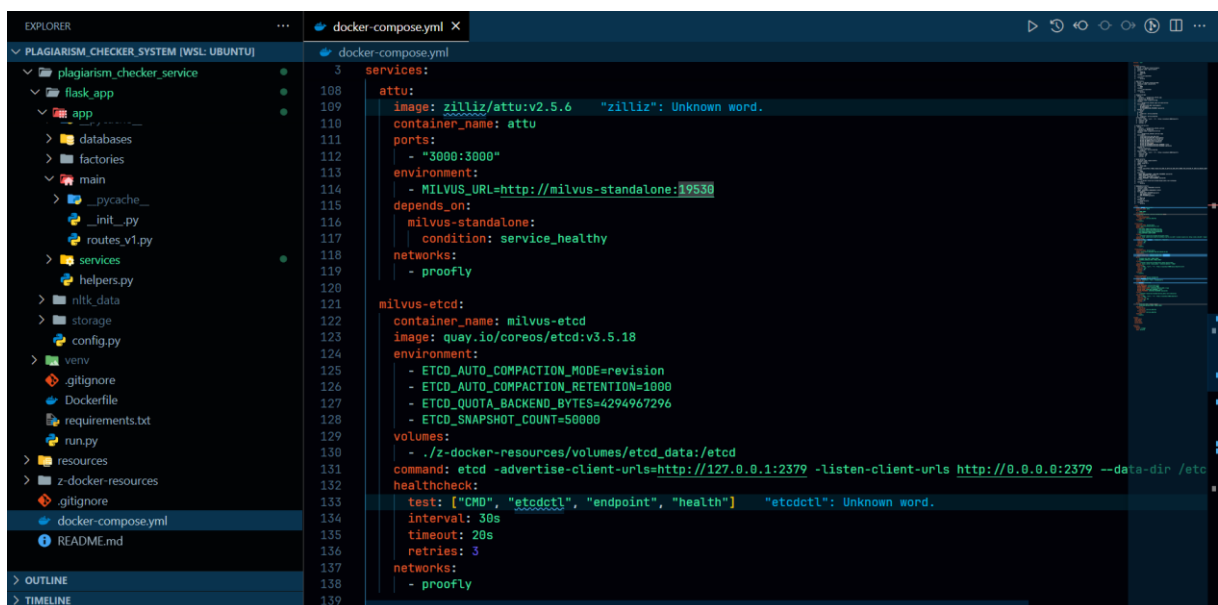
dùng không bỏ lỡ các cập nhật quan trọng.

- **Quản lý lớp học:** Xem danh sách lớp, học sinh, giảng viên; thực hiện tìm kiếm, lọc; đăng ký học sinh và giáo viên vào lớp.
- **Quản lý tài liệu:** Thêm, sửa, xóa tài liệu; tìm kiếm, lọc tài liệu; nộp tài liệu mới và xem danh sách tài liệu đã nộp. Hỗ trợ xếp hạng và đánh giá tài liệu theo tiêu chí chất lượng.
- **Phản hồi:** Người dùng có thể gửi phản hồi, khiếu nại trực tiếp qua giao diện hệ thống, giúp cải thiện chất lượng dịch vụ và hỗ trợ kịp thời.

Phụ lục 2: Mã nguồn hệ thống



```
1 from flask import Blueprint, request, jsonify, send_file
2
3 from flask_app.config import Config
4 from flask_app.app.services.migration import Migration
5 from flask_app.app.services.document_service import DocumentService
6 from flask_app.app.databases.milvus_connection import MilvusConnection
7 from flask_app.app.services.plagiarism_checker_service import PlagiarismCheckerService
8 from flask_app.app.services.pdf_processor import PDFProcessor
9
10 # Create a Blueprint for main routes
11 bp_v1 = Blueprint("main", __name__, url_prefix="/v1/api")
12
13
14 @bp_v1.route("/ping")
15 def pong():
16     return "pong"
17
18
19 @bp_v1.route("/migrate")
20 def migrate():
21     success = Migration.up()
22
23     if success:
24         return jsonify({"success": True, "message": "Migrate data is successfully"})
25     else:
26         return jsonify({"success": False, "message": "Migrate data is failed"})
27
28
29 @bp_v1.route("/rollback")
30 def rollback():
```



```
3 services:
108   attu:
109     image: zilliz/attu:v2.5.6 "zilliz: Unknown word."
110     container_name: attu
111     ports:
112       - "3000:3000"
113     environment:
114       - MILVUS_URL=http://milvus-standalone:19530
115     depends_on:
116       - milvus-standalone:
117         condition: service_healthy
118     networks:
119       - proofly
120
121   milvus-etcd:
122     container_name: milvus-etcd
123     image: quay.io/coreos/etcd:v3.5.18
124     environment:
125       - ETCD_AUTO_COMPACTION_MODE=revision
126       - ETCD_AUTO_COMPACTION_RETENTION=1000
127       - ETCD_QUOTA_BACKEND_BYTES=4294967296
128       - ETCD_SNAPSHOT_COUNT=50000
129     volumes:
130       - ./z-docker-resources/volumes/etcd_data:/etcd
131     command: etcd -advertise-client-urls=http://127.0.0.1:2379 -listen-client-urls http://0.0.0.0:2379 --data-dir /etcd
132     healthcheck:
133       test: ["CMD", "etcdctl", "endpoint", "health"] "etcdctl: Unknown word."
134       interval: 30s
135       timeout: 20s
136       retries: 3
137     networks:
138       - proofly
139
```