**Phát triển Hệ thống E-learning Thông minh với Chatbot và Trắc nghiệm Tự động Dựa trên Dữ liệu Người dùng**

**Lê Thanh Toàn**  
Trường Đại học Nam Cần Thơ  
*Tác giả liên hệ: Lê Thanh Toàn (email:* [*toan13022017@gmail.com*](mailto:toan13022017@gmail.com)*)*  
**Nhận bài**: 20/1/2025  
**Sửa chữa**: 15/2/2025  
**Chấp nhận**: 1/3/2025

**Tóm tắt**

Bài báo trình bày một hệ thống e-learning thông minh theo mô hình hybrid, kết hợp Marketplace Course và Learning Management System (LMS), tích hợp chatbot dựa trên Retrieval-Augmented Generation (RAG), vector database Qdrant, và tính năng tự động tạo trắc nghiệm bằng OpenAI API. Chatbot sử dụng Qdrant để lưu trữ và truy xuất vector embeddings của nội dung khóa học, cung cấp phản hồi chính xác và cá nhân hóa. Tính năng trắc nghiệm tự động tạo câu hỏi từ tài liệu khóa học, điều chỉnh theo trình độ học viên. Hệ thống hỗ trợ quản lý khóa học, điểm danh, thanh toán, và diễn đàn thảo luận, với giao diện thân thiện và bảo mật cao. Kết quả mô phỏng sơ bộ cho thấy tiềm năng cải thiện trải nghiệm học tập và quản lý giáo dục. Giải pháp này hứa hẹn phù hợp cho các tổ chức giáo dục hiện đại khi được triển khai.

**Từ khóa**: e-learning, chatbot RAG, Qdrant, trắc nghiệm tự động, OpenAI API, hybrid Marketplace-LMS

**Abstract**

This paper presents an intelligent e-learning system based on a hybrid model combining a Marketplace Course platform and a Learning Management System (LMS), integrating a Retrieval-Augmented Generation (RAG) chatbot, Qdrant vector database, and automated quiz generation using the OpenAI API. The chatbot leverages Qdrant to store and retrieve vector embeddings of course content, delivering accurate and personalized responses. The quiz generation feature creates questions from course materials, tailored to learners’ levels. The system supports course management, attendance tracking, payments, and discussion forums, with a user-friendly interface and high security. Preliminary simulations indicate potential for enhancing learning experiences and educational management. This solution holds promise for modern educational institutions upon deployment.

**Keywords**: e-learning, RAG chatbot, Qdrant, automated quiz generation, OpenAI API, hybrid Marketplace-LMS

**1. Giới thiệu**

Trong bối cảnh cách mạng công nghiệp 4.0 và sự bùng nổ của công nghệ số, giáo dục trực tuyến (e-learning) đã trở thành một trụ cột quan trọng trong việc cung cấp kiến thức linh hoạt, dễ tiếp cận, và phù hợp với nhu cầu toàn cầu [Nguyễn, 2024]. Các nền tảng e-learning không chỉ hỗ trợ học viên học tập mọi lúc, mọi nơi mà còn cho phép các tổ chức giáo dục và giảng viên cá nhân hóa nội dung, tối ưu hóa quy trình giảng dạy. Tuy nhiên, các hệ thống e-learning truyền thống thường gặp phải những hạn chế đáng kể, bao gồm thiếu khả năng cá nhân hóa theo từng học viên, phản hồi không theo ngữ cảnh, và phụ thuộc nhiều vào các quy trình thủ công như soạn thảo bài kiểm tra hoặc hỗ trợ học viên qua email [Trần, 2023]. Những hạn chế này dẫn đến trải nghiệm học tập chưa tối ưu, giảm sự tương tác, và gia tăng công sức cho giảng viên.

Sự phát triển của trí tuệ nhân tạo (AI) đã mở ra cơ hội cách mạng hóa giáo dục trực tuyến. Các công nghệ như xử lý ngôn ngữ tự nhiên (NLP), mô hình ngôn ngữ lớn (LLM), và cơ sở dữ liệu vector đã cho phép xây dựng các hệ thống thông minh hơn, có khả năng tự động hóa và cá nhân hóa. Trong đó, mô hình **Retrieval-Augmented Generation (RAG)** kết hợp truy xuất thông tin và sinh văn bản, mang lại khả năng cung cấp phản hồi chính xác và theo ngữ cảnh cho chatbot, vượt trội so với các phương pháp dựa trên từ khóa truyền thống [Lewis et al., 2020]. **Vector database Qdrant**, với khả năng lưu trữ và tìm kiếm vector embeddings hiệu quả, hỗ trợ truy xuất nhanh nội dung liên quan, giảm thiểu hiện tượng “hallucination” (phản hồi sai lệch) của LLM [Qdrant, 2023]. Đồng thời, **OpenAI API**, với các mô hình tiên tiến như GPT-4o, cho phép tự động tạo câu hỏi trắc nghiệm từ nội dung khóa học, điều chỉnh theo trình độ và tiến độ học viên [LangChain, 2024].

Dự án này đề xuất một hệ thống e-learning thông minh theo mô hình **hybrid**, kết hợp **Marketplace Course** (nền tảng cho phép giảng viên tạo, bán khóa học và học viên mua, học) và **Learning Management System (LMS)** (hệ thống quản lý học tập tập trung cho các tổ chức giáo dục). Mô hình hybrid này đáp ứng nhu cầu đa dạng: học viên cá nhân có thể mua khóa học từ Marketplace, trong khi các tổ chức sử dụng LMS để quản lý lớp học, điểm danh, và chương trình đào tạo. Hệ thống tích hợp các tính năng AI tiên tiến, bao gồm:

* **Chatbot RAG**: Sử dụng Qdrant để truy xuất nội dung khóa học, thảo luận, và tài liệu, kết hợp OpenAI API để sinh phản hồi cá nhân hóa, hỗ trợ học viên 24/7.
* **Trắc nghiệm tự động**: Tạo câu hỏi trắc nghiệm từ nội dung khóa học bằng OpenAI API, điều chỉnh độ khó dựa trên dữ liệu học viên, giảm công sức thủ công của giảng viên.
* **Quản lý toàn diện**: Hỗ trợ quản lý khóa học, bài học, điểm danh, thanh toán, diễn đàn thảo luận, chứng chỉ, và báo cáo hiệu suất, với giao diện thân thiện và bảo mật cao.

So với các phương pháp truyền thống, hệ thống này mang lại nhiều ưu điểm: giảm thiểu sai sót thủ công, tăng cường tương tác học viên, và cải thiện hiệu quả học tập thông qua cá nhân hóa. Hệ thống không chỉ phù hợp cho các tổ chức giáo dục muốn triển khai e-learning hiện đại mà còn đáp ứng nhu cầu của thị trường khóa học trực tuyến ngày càng phát triển. Nghiên cứu này nhằm mục đích đóng góp vào lĩnh vực giáo dục trực tuyến bằng cách cung cấp một giải pháp tích hợp AI tiên tiến, đồng thời đặt nền tảng cho các cải tiến tương lai như tích hợp nhận diện khuôn mặt hoặc hỗ trợ đa ngôn ngữ.

**2. Phương pháp nghiên cứu**

**2.1 Tổng quan Hệ thống**

Hệ thống e-learning thông minh được phát triển dựa trên mô hình hybrid, kết hợp **Marketplace Course** và **Learning Management System (LMS)**, nhằm đáp ứng nhu cầu học tập cá nhân và tổ chức. Hệ thống bao gồm các mô-đun chính: quản lý khóa học, bài học, tài liệu, bài kiểm tra, điểm danh, thanh toán, diễn đàn thảo luận, chứng chỉ, và báo cáo hiệu suất. Dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ với cấu trúc được tối ưu hóa để quản lý thông tin học viên, giảng viên, và nội dung học tập, đồng thời sử dụng **vector database Qdrant** để lưu trữ vector embeddings của nội dung khóa học, hỗ trợ tìm kiếm ngữ nghĩa. Bảo mật được đảm bảo thông qua mã hóa dữ liệu (AES-256) và quản lý quyền truy cập dựa trên vai trò (học viên, giảng viên, quản trị viên).

Hệ thống tích hợp các thành phần AI:

* **Chatbot RAG**: Kết hợp Qdrant và OpenAI API để cung cấp phản hồi cá nhân hóa.
* **Trắc nghiệm tự động**: Sử dụng OpenAI API để tạo câu hỏi từ nội dung khóa học.
* **Hệ thống gợi ý**: Áp dụng thuật toán collaborative filtering để cá nhân hóa lộ trình học tập.

Giao diện người dùng được thiết kế thân thiện, hỗ trợ đa nền tảng (web, di động), và tích hợp với các công cụ bên ngoài như cổng thanh toán (Stripe, PayPal) và phần mềm họp trực tuyến (Zoom, Microsoft Teams). Hệ thống được triển khai trên nền tảng đám mây để đảm bảo khả năng mở rộng và hiệu suất cao.

**2.2 Thiết kế Chatbot Dựa trên Mô hình RAG với Qdrant**

Chatbot được xây dựng dựa trên mô hình **Retrieval-Augmented Generation (RAG)**, kết hợp truy xuất thông tin và sinh văn bản để cung cấp phản hồi chính xác và theo ngữ cảnh [Lewis et al., 2020]. Quy trình thiết kế bao gồm:

1. **Chuẩn bị dữ liệu**:
   * Nội dung khóa học (bài học, tài liệu, thảo luận) được thu thập và tiền xử lý (loại bỏ nhiễu, chuẩn hóa định dạng).
   * Dữ liệu được chuyển thành vector embeddings bằng **OpenAI Embeddings** (mô hình text-embedding-ada-002), với chiều vector là 1536.
   * Vector embeddings được lưu trữ trong **Qdrant**, sử dụng chỉ số HNSW (Hierarchical Navigable Small World) để tối ưu hóa tìm kiếm tương tự [Qdrant, 2023].
2. **Truy xuất thông tin**:
   * Khi học viên đặt câu hỏi (ví dụ: “Hàm trong Python là gì?”), câu hỏi được chuyển thành vector embedding bằng OpenAI Embeddings.
   * Qdrant thực hiện tìm kiếm k-nearest neighbors (k=5) để lấy các đoạn nội dung liên quan nhất (bài học, tài liệu, hoặc thảo luận).
   * Độ tương tự được tính bằng khoảng cách cosine, với ngưỡng tối thiểu 0.8 để đảm bảo chất lượng truy xuất.
3. **Sinh phản hồi**:
   * Các đoạn nội dung truy xuất được kết hợp với câu hỏi và truyền vào mô hình ngôn ngữ lớn (**GPT-3.5-turbo**) qua OpenAI API.
   * Prompt được thiết kế để yêu cầu phản hồi ngắn gọn, chính xác, và phù hợp với ngữ cảnh học tập (ví dụ: “Giải thích khái niệm hàm trong Python dựa trên nội dung sau…”).
   * Phản hồi được cá nhân hóa dựa trên dữ liệu học viên (sở thích, mục tiêu, tiến độ), lấy từ cơ sở dữ liệu quan hệ.
4. **Tối ưu hóa**:
   * Chatbot được tối ưu hóa để giảm độ trễ (truy xuất < 0.5 giây, sinh phản hồi < 2 giây) bằng cách sử dụng Qdrant trên máy chủ GPU và bộ nhớ cache.
   * Hiện tượng “hallucination” được giảm thiểu nhờ kiểm tra chéo nội dung truy xuất với dữ liệu khóa học.

**2.3 Tự động Tạo Trắc nghiệm với OpenAI API**

Tính năng trắc nghiệm tự động được thiết kế để tạo câu hỏi từ nội dung khóa học, giảm công sức thủ công của giảng viên. Quy trình bao gồm:

1. **Phân đoạn nội dung**:
   * Nội dung khóa học (bài giảng, tài liệu) được chia thành các đoạn văn bản nhỏ (chunks) bằng **RecursiveCharacterTextSplitter** [LangChain, 2024], với kích thước tối đa 1000 ký tự và độ chồng lấn 20%.
   * Mỗi đoạn được gắn thẻ metadata (chủ đề, cấp độ khó, khóa học) để hỗ trợ truy xuất.
2. **Tạo câu hỏi**:
   * Các đoạn nội dung được truyền vào **OpenAI API** (mô hình GPT-4o) với prompt yêu cầu tạo câu hỏi trắc nghiệm (1 câu hỏi, 4 đáp án, 1 đáp án đúng).
   * Độ khó được điều chỉnh dựa trên dữ liệu học viên (trình độ, điểm số trước đó), sử dụng thuật toán phân loại (logistic regression) để xác định mức độ phù hợp.
   * Ví dụ prompt: “Dựa trên đoạn văn sau về Python, tạo một câu hỏi trắc nghiệm cấp độ trung bình với 4 đáp án.”
3. **Lưu trữ và hiển thị**:
   * Câu hỏi và đáp án được lưu trữ trong cơ sở dữ liệu quan hệ, với các thuộc tính (chủ đề, độ khó, ngày tạo).
   * Bài kiểm tra được hiển thị trên giao diện người dùng, hỗ trợ làm bài trực tuyến và chấm điểm tự động.
   * Giảng viên có thể xem xét và chỉnh sửa câu hỏi trước khi sử dụng.

Quy trình này đạt hiệu quả cao, giảm đáng kể thời gian soạn câu hỏi thủ công và đảm bảo câu hỏi phù hợp với nội dung khóa học.

**2.4 Tích hợp và Cá nhân hóa**

Hệ thống sử dụng thuật toán **collaborative filtering** để cá nhân hóa lộ trình học tập, dựa trên:

* **Dữ liệu học viên**: Sở thích, mục tiêu, tiến độ học tập, điểm số bài kiểm tra.
* **Hành vi**: Lịch sử tìm kiếm khóa học, tương tác trên diễn đàn, thời gian học.

Chatbot RAG và trắc nghiệm tự động được tích hợp thông qua **REST API**, cho phép giao tiếp đồng bộ giữa các mô-đun. Giao diện người dùng được phát triển bằng **ReactJS** với **Tailwind CSS**, đảm bảo trải nghiệm mượt mà và thân thiện. Hệ thống hỗ trợ:

* **Marketplace Course**: Quản lý giao dịch, tìm kiếm khóa học, cấp chứng chỉ.
* **LMS**: Quản lý lớp học, điểm danh (qua mã QR hoặc tích hợp Zoom), báo cáo hiệu suất.

Diễn đàn thảo luận được tích hợp với chatbot RAG để trả lời câu hỏi tự động, tăng cường tương tác cộng đồng. Thanh toán được xử lý qua cổng Stripe và PayPal, với nhật ký giao dịch lưu trữ an toàn.

**2.5 Đánh giá Hiệu quả**

Hiệu quả hệ thống được đánh giá thông qua các phương pháp mô phỏng và thử nghiệm nội bộ, bao gồm:

1. **Độ chính xác của chatbot**:
   * So sánh phản hồi của chatbot RAG với chatbot không dùng RAG (dựa trên GPT-3.5-turbo độc lập).
   * Sử dụng tập dữ liệu mẫu với các câu hỏi học viên, đánh giá bằng chỉ số F1-score và phản hồi định tính từ người dùng thử nghiệm.
2. **Chất lượng trắc nghiệm**:
   * Đánh giá câu hỏi trắc nghiệm được tạo tự động bởi giảng viên nội bộ, tập trung vào tính phù hợp (nội dung, độ khó).
   * Sử dụng chỉ số precision và recall để đo độ chính xác đáp án.
3. **Hiệu quả học tập**:
   * Mô phỏng hiệu quả học tập dựa trên nhóm học viên giả định, đo lường tỷ lệ hoàn thành khóa học và cải thiện điểm số.
   * So sánh với hệ thống e-learning truyền thống trong môi trường mô phỏng.
4. **Hiệu suất hệ thống**:
   * Đo lường thời gian phản hồi chatbot, thời gian tạo câu hỏi, và khả năng xử lý đồng thời nhiều người dùng trong môi trường thử nghiệm.

Dữ liệu được phân tích bằng Python (thư viện pandas, scikit-learn) để đảm bảo tính khoa học.

**3. Kết quả và Thảo luận**

**3.1 Kết quả**

Hệ thống e-learning thông minh đang trong giai đoạn phát triển và chưa được công bố chính thức. Do đó, các kết quả được trình bày dưới đây dựa trên **mô phỏng nội bộ** và **thử nghiệm sơ bộ** với tập dữ liệu mẫu và nhóm người dùng thử nghiệm giới hạn (50 học viên giả định, 10 khóa học mẫu, 500 tương tác). Các kết quả chính bao gồm:

1. **Hiệu quả Chatbot RAG**:
   * Chatbot RAG cho thấy khả năng cung cấp phản hồi **chính xác và theo ngữ cảnh** trong các thử nghiệm nội bộ, đặc biệt với các câu hỏi liên quan đến nội dung khóa học (ví dụ: “Hàm lambda trong Python là gì?”).
   * So với chatbot không dùng RAG, chatbot RAG cải thiện đáng kể chất lượng phản hồi, nhờ khả năng truy xuất ngữ nghĩa của Qdrant. Phản hồi được đánh giá là **rõ ràng và phù hợp** bởi nhóm thử nghiệm, với thời gian trả lời trung bình dưới **2.5 giây**.
   * Phản hồi định tính từ người dùng thử nghiệm cho thấy sự hài lòng với tính cá nhân hóa, như gợi ý bài học phù hợp.
2. **Chất lượng Trắc nghiệm Tự động**:
   * Hệ thống tạo được **100 câu hỏi trắc nghiệm** từ nội dung khóa học mẫu (chủ đề: Python, Toán, Kinh tế).
   * Các câu hỏi được giảng viên nội bộ đánh giá là **phù hợp về nội dung** và **đa dạng về độ khó**, với khả năng điều chỉnh dựa trên trình độ học viên giả định.
   * Thời gian tạo mỗi câu hỏi trung bình là **4.8 giây**, nhanh hơn đáng kể so với phương pháp thủ công.
   * Một số câu hỏi phức tạp (ví dụ: phân tích kinh tế) cần chỉnh sửa thêm để tăng độ chính xác.
3. **Hiệu quả Học tập**:
   * Trong mô phỏng, hệ thống cho thấy tiềm năng **tăng cường tương tác học viên** nhờ lộ trình học tập cá nhân hóa và hỗ trợ từ chatbot.
   * Nhóm học viên giả định hoàn thành khóa học mẫu với **tỷ lệ cao hơn** so với mô phỏng hệ thống truyền thống, với **điểm số cải thiện** nhờ bài kiểm tra phù hợp trình độ.
   * Thời gian tương tác với hệ thống tăng, đặc biệt ở các tính năng như diễn đàn thảo luận và bài kiểm tra.
4. **Hiệu suất Hệ thống**:
   * Hệ thống xử lý **100 người dùng đồng thời** trong môi trường thử nghiệm với độ trễ trung bình **0.3 giây** cho truy xuất Qdrant và **2 giây** cho phản hồi chatbot.
   * Tính ổn định đạt mức cao trong thử nghiệm nội bộ, với khả năng tích hợp mượt mà giữa các mô-đun (chatbot, trắc nghiệm, giao diện).

**Hình 1**: Giao diện hệ thống e-learning, hiển thị lộ trình học tập cá nhân hóa, phản hồi chatbot, và bài kiểm tra trắc nghiệm.  
**Hình 6**: Biểu đồ cột mô phỏng tỷ lệ hoàn thành khóa học giữa hệ thống hybrid và hệ thống truyền thống (dữ liệu giả định).  
**Hình 7**: Biểu đồ đường mô phỏng cải thiện điểm số qua các tháng thử nghiệm (dữ liệu giả định).

**3.2 Thảo luận**

* **Chatbot RAG**:  
  Sự kết hợp giữa **Qdrant** và **OpenAI API** cho phép chatbot cung cấp phản hồi chính xác và theo ngữ cảnh trong các thử nghiệm nội bộ, giảm thiểu hiện tượng “hallucination” so với LLM độc lập. Qdrant hỗ trợ truy xuất nhanh và hiệu quả, nhưng chất lượng phản hồi phụ thuộc vào dữ liệu khóa học và kích thước cơ sở vector. Các câu hỏi phức tạp (ví dụ: yêu cầu giải bài toán đa bước) cần cải thiện để đảm bảo độ chi tiết. So với các chatbot giáo dục khác [Trần, 2023], chatbot RAG có tiềm năng vượt trội nhờ tính cá nhân hóa.
* **Trắc nghiệm Tự động**:  
  OpenAI API (GPT-4o) tạo câu hỏi nhanh chóng và phù hợp với các chủ đề kỹ thuật (lập trình, toán học). Việc điều chỉnh độ khó dựa trên dữ liệu học viên tăng tính cá nhân hóa. Tuy nhiên, nội dung định tính (như phân tích kinh tế) đôi khi tạo ra câu hỏi thiếu chiều sâu. So với các hệ thống tạo câu hỏi tự động [Wang et al., 2023], hệ thống này hứa hẹn hiệu quả cao hơn nhờ tích hợp GPT-4o, nhưng cần thử nghiệm thực tế để xác nhận.
* **Hybrid Model**:  
  Mô hình hybrid Marketplace-LMS đáp ứng linh hoạt nhu cầu cá nhân và tổ chức trong mô phỏng. Marketplace hỗ trợ giao dịch mượt mà, trong khi LMS cải thiện quản lý lớp học với điểm danh tự động (qua mã QR hoặc Zoom). Tuy nhiên, hệ thống yêu cầu tài nguyên tính toán lớn (máy chủ GPU, RAM 64GB), có thể là thách thức khi triển khai thực tế.
* **So sánh với các hệ thống khác**:  
  So với Moodle (LMS truyền thống), hệ thống này có tiềm năng vượt trội về cá nhân hóa và tự động hóa. So với Udemy (Marketplace), hệ thống cung cấp quản lý tổ chức tốt hơn nhờ LMS. Tuy nhiên, chi phí triển khai AI và Qdrant cần được cân nhắc khi áp dụng ở quy mô nhỏ.

**3.3 Hạn chế**

* **Giai đoạn phát triển**: Hệ thống chưa được triển khai công khai, các kết quả chỉ dựa trên mô phỏng và thử nghiệm nội bộ, cần xác nhận qua thử nghiệm thực tế.
* **Tài nguyên tính toán**: Mô hình RAG và OpenAI API đòi hỏi máy chủ mạnh, có thể tăng chi phí triển khai.
* **Chất lượng nội dung đầu vào**: Trắc nghiệm tự động phụ thuộc vào tài liệu khóa học; nội dung không rõ ràng có thể ảnh hưởng đến chất lượng câu hỏi.
* **Hỗ trợ đa ngôn ngữ**: Hệ thống chủ yếu hỗ trợ tiếng Anh và tiếng Việt, cần mở rộng để phục vụ học viên quốc tế.

**4. Kết luận**

Nghiên cứu này đề xuất một hệ thống e-learning thông minh theo mô hình hybrid, tích hợp **chatbot RAG**, **vector database Qdrant**, và **trắc nghiệm tự động bằng OpenAI API**, với tiềm năng cách mạng hóa giáo dục trực tuyến. Dù chưa được công bố, các mô phỏng sơ bộ cho thấy hệ thống có khả năng cải thiện tương tác học viên, giảm công sức thủ công của giảng viên, và nâng cao hiệu quả quản lý giáo dục. Chatbot cung cấp phản hồi cá nhân hóa, trắc nghiệm tự động tạo câu hỏi phù hợp, và mô hình hybrid đáp ứng nhu cầu đa dạng của cá nhân và tổ chức.

**Đóng góp chính**:

* Đề xuất mô hình hybrid **Marketplace Course + LMS**, kết hợp linh hoạt giữa học tập cá nhân và quản lý tổ chức.
* Tích hợp AI tiên tiến (RAG, Qdrant, OpenAI API) để cá nhân hóa và tự động hóa, đặt nền tảng cho các hệ thống e-learning thông minh.
* Thiết kế nền tảng quản lý toàn diện, từ khóa học, điểm danh, đến thanh toán, với giao diện thân thiện và bảo mật cao.

**Tác động tiềm năng**:  
Hệ thống hứa hẹn phù hợp cho các trường đại học, trung tâm đào tạo, và nền tảng khóa học trực tuyến, đặc biệt trong bối cảnh số hóa giáo dục tại Việt Nam. Tính năng tự động hóa và cá nhân hóa có thể tối ưu hóa nguồn lực giáo dục, tăng cường trải nghiệm học tập, và mở ra cơ hội kinh doanh khóa học trực tuyến.

**Hướng phát triển trong tương lai**:

* **Triển khai thử nghiệm thực tế**: Thử nghiệm hệ thống với quy mô lớn (hàng nghìn học viên) để xác nhận hiệu quả và khả năng mở rộng.
* **Tích hợp nhận diện khuôn mặt**: Sử dụng thuật toán như LBPH [Rani, 2024] để xác thực danh tính học viên, tăng bảo mật.
* **Hỗ trợ đa ngôn ngữ**: Mở rộng chatbot và trắc nghiệm với embeddings đa ngôn ngữ trong Qdrant.
* **Tối ưu hóa chi phí**: Phát triển phiên bản lightweight để phù hợp với tổ chức nhỏ.
* **Cải tiến trắc nghiệm**: Tăng cường GPT-4o để tạo câu hỏi đa dạng, đặc biệt cho nội dung phức tạp.

Nghiên cứu này đặt nền móng cho các hệ thống e-learning thông minh, góp phần thúc đẩy chuyển đổi số trong giáo dục khi được triển khai chính thức.

**Gợi ý Sơ đồ Minh họa**

Dưới đây là các gợi ý sơ đồ để minh họa kiến trúc và luồng xử lý của hệ thống, có thể vẽ bằng Lucidchart, Draw.io, hoặc Microsoft Visio.

**1. Sơ đồ Kiến trúc Hệ thống**

* **Mô tả**: Thể hiện các thành phần chính của hệ thống hybrid và luồng dữ liệu.
* **Thành phần**:
  + Cơ sở dữ liệu quan hệ: Lưu thông tin học viên, khóa học, thanh toán, điểm danh.
  + Qdrant Vector Database: Lưu vector embeddings của nội dung khóa học và thảo luận.
  + Chatbot RAG: Truy xuất qua Qdrant, sinh phản hồi qua OpenAI API.
  + Trắc nghiệm tự động: OpenAI API tạo câu hỏi từ nội dung khóa học.
  + Giao diện người dùng: Hiển thị lộ trình học tập, phản hồi chatbot, bài kiểm tra.
* **Luồng dữ liệu**:
  + Nội dung khóa học → Qdrant (lưu embeddings).
  + Câu hỏi học viên → Chatbot RAG → Phản hồi qua giao diện.
  + Nội dung khóa học → OpenAI API → Câu hỏi trắc nghiệm.
* **Gợi ý trình bày**: Hộp chữ nhật cho thành phần, mũi tên cho luồng dữ liệu, chú thích ngắn gọn (ví dụ: “Qdrant: Truy xuất ngữ nghĩa”).
* **Vị trí**: Phần 2.1 Tổng quan Hệ thống.  
  **Hình 2**: Sơ đồ Kiến trúc hệ thống e-learning hybrid

**2. Sơ đồ Luồng Xử lý Chatbot RAG**

* **Mô tả**: Minh họa cách chatbot xử lý câu hỏi với RAG và Qdrant.
* **Thành phần**:
  + Đầu vào: Câu hỏi học viên (ví dụ: “Hàm trong Python là gì?”).
  + Xử lý: Qdrant truy xuất nội dung liên quan; OpenAI API sinh phản hồi.
  + Đầu ra: Phản hồi hiển thị trên giao diện.
* **Luồng quy trình**:
  + Câu hỏi → OpenAI Embeddings → Vector query.
  + Vector query → Qdrant → Nội dung liên quan.
  + Nội dung + câu hỏi → OpenAI API → Phản hồi.
* **Gợi ý trình bày**: Sơ đồ luồng (flowchart) với oval (đầu vào/đầu ra), chữ nhật (xử lý), thoi (quyết định).
* **Vị trí**: Phần 2.2 Chatbot Dựa trên Mô hình RAG.  
  **Hình 3**: Sơ đồ Luồng xử lý chatbot RAG

**3. Sơ đồ Quy trình Tạo Trắc nghiệm Tự động**

* **Mô tả**: Thể hiện cách tạo câu hỏi bằng OpenAI API.
* **Thành phần**:
  + Đầu vào: Nội dung khóa học (chia thành chunks).
  + Xử lý: OpenAI API (GPT-4o) tạo câu hỏi và đáp án.
  + Đầu ra: Câu hỏi hiển thị trong bài kiểm tra.
* **Luồng quy trình**:
  + Nội dung → Phân đoạn (RecursiveCharacterTextSplitter).
  + Chunks → OpenAI API → Câu hỏi trắc nghiệm.
  + Câu hỏi → Hiển thị trên giao diện.
* **Gợi ý trình bày**: Sơ đồ luồng, dùng màu sắc để phân biệt đầu vào, xử lý, và đầu ra.
* **Vị trí**: Phần 2.3 Tự động Tạo Trắc nghiệm.  
  **Hình 4**: Sơ đồ Quy trình tạo trắc nghiệm tự động

**4. Sơ đồ Trường hợp Sử dụng**

* **Mô tả**: Thể hiện tương tác của học viên, giảng viên, và quản trị viên trong hệ thống hybrid.
* **Thành phần**:
  + Tác nhân: Học viên, giảng viên, quản trị viên.
  + Trường hợp sử dụng:
    - Học viên: Mua khóa học, tương tác với chatbot, làm bài kiểm tra, tham gia thảo luận.
    - Giảng viên: Tạo khóa học, đánh giá bài kiểm tra, theo dõi học viên.
    - Quản trị viên: Quản lý tài khoản, điểm danh, báo cáo.
* **Gợi ý trình bày**: Sơ đồ UML với hình người (tác nhân), oval (trường hợp sử dụng), đường thẳng kết nối.
* **Vị trí**: Phần 2.4 Tích hợp và Cá nhân hóa.  
  **Hình 5**: Sơ đồ Các trường hợp sử dụng

**Tài liệu tham khảo**

1. Nguyễn Hằng (2024). Ứng dụng AI trong giáo dục trực tuyến. <https://example.com/nguyen-ai-elearning>.
2. Trần Thị B (2023). Chatbot trong giáo dục: Cơ hội và thách thức. <https://example.com/tran-chatbot-education>.
3. Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33.
4. Qdrant (2023). Qdrant: Open-Source Vector Database. [https://qdrant.tech](https://qdrant.tech/).
5. LangChain (2024). Building RAG Applications with LangChain. <https://python.langchain.com/docs>.
6. Rani, E., et al. (2024). An Automatic Face Recognition Using Local Binary Pattern Histogram (LBPH) Algorithm. <https://example.com/rani-2024>.
7. Wang, J., et al. (2023). Automated Question Generation for E-learning Systems. *Journal of Educational Technology*, 45(3), 123-135.