

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN

NGUYỄN LONG VŨ - 3121411228
LA HIẾU PHONG - 3121411162

XÂY DỰNG HỆ THỐNG
E-LEARNING BẰNG NEXTJS

NGÀNH: CÔNG NGHỆ THÔNG TIN
TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2025

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN

NGUYỄN LONG VŨ - 3121411228
LA HIẾU PHONG - 3121411162

XÂY DỰNG HỆ THỐNG
E-LEARNING BẰNG NEXTJS

NGÀNH: CÔNG NGHỆ THÔNG TIN
TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

NGƯỜI HƯỚNG DẪN: TS. CAO THÁI PHƯƠNG THANH

TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2025

MỞ ĐẦU

Trong bối cảnh công nghệ số đang phát triển với tốc độ vũ bão, đặc biệt là sau những biến động sâu sắc do đại dịch COVID-19 gây ra, giáo dục trực tuyến (E-Learning) đã khẳng định vị thế là một trụ cột không thể thiếu đối với nhiều tổ chức và cá nhân trên toàn cầu. Việc tiếp cận và sở hữu một nền tảng E-Learning tiên tiến, hiệu quả không chỉ là một lợi thế cạnh tranh mà còn là một yêu cầu cấp thiết. Đối với các cơ sở giáo dục, E-Learning mở ra cánh cửa tiếp cận học viên rộng lớn hơn, vượt qua những rào cản về địa lý và thời gian, đồng thời cho phép tối ưu hóa quy trình giảng dạy và quản lý. Đối với các doanh nghiệp, đây là công cụ đắc lực trong việc đào tạo và phát triển nguồn nhân lực, giúp nhanh chóng cập nhật kiến thức và kỹ năng cho đội ngũ nhân viên một cách linh hoạt và tiết kiệm chi phí. Hơn thế nữa, người học cá nhân cũng được hưởng lợi từ sự phong phú của nguồn tài liệu, khả năng tự chủ động về thời gian và không gian học tập, cũng như cơ hội tiếp cận các phương pháp giáo dục hiện đại.

Tại Việt Nam, với tỷ lệ người dùng Internet ngày càng gia tăng và sự nhận thức sâu sắc hơn về tầm quan trọng của chuyên đổi số trong giáo dục, nhu cầu về một hệ thống E-Learning được thiết kế tối ưu, thân thiện với người dùng và có khả năng tùy biến cao để đáp ứng các yêu cầu đặc thù của từng tổ chức là vô cùng lớn. Một hệ thống như vậy không chỉ cần đảm bảo các chức năng cơ bản mà còn phải tích hợp những công nghệ mới nhất để nâng cao trải nghiệm người học và hiệu quả giảng dạy.

Xuất phát từ những yêu cầu thực tiễn đó, khóa luận này đặt ra mục tiêu nghiên cứu sâu rộng và triển khai một quy trình toàn diện để xây dựng một hệ thống E-Learning hiện đại, sử dụng framework NextJS cho frontend. Hệ thống này không chỉ tập trung vào việc cung cấp các tính năng cốt lõi như quản lý khóa học một cách khoa học, quy trình đăng ký học viên thuận tiện, khả năng phân phối nội dung học tập đa dạng (video, văn bản, bài tập tương tác), và các công cụ đánh giá kết quả học tập hiệu quả, mà còn giới thiệu một tính năng đặc biệt và mang tính đột phá: một chatbot thông minh. Chatbot này được phát triển dựa trên công nghệ mạng nơ-ron và tích hợp thông qua một API nằm trong backend của hệ thống, với khả năng cung cấp thông tin chi tiết về hệ thống, giải đáp thắc mắc và hỗ trợ người dùng trong

suốt quá trình tìm hiểu và tương tác với nền tảng.

Khóa luận này được kỳ vọng sẽ mang lại giá trị thực tiễn cao, không chỉ là một bài tập ứng dụng kiến thức và kỹ năng đã được tích lũy trong lĩnh vực phát triển web vào một bài toán cụ thể, mà còn là một nỗ lực đóng góp vào việc phát triển các giải pháp giáo dục trực tuyến tiên tiến tại Việt Nam. Việc lựa chọn đề tài “Xây dựng hệ thống E-Learning bằng NextJS” cho khóa luận tốt nghiệp thể hiện mong muốn khám phá và làm chủ các công nghệ mới, đồng thời giải quyết một nhu cầu cấp thiết của xã hội trong kỷ nguyên số.

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1 Lý do lựa chọn đề tài:

Sự phát triển không ngừng của công nghệ thông tin và Internet đã định hình lại cách chúng ta tiếp cận tri thức, khiến nhu cầu học tập trực tuyến (E-learning) và phát triển kỹ năng cá nhân trở thành một xu thế tất yếu. Các nền tảng E-learning không chỉ là kênh cung cấp kiến thức mà còn là môi trường học tập tương tác, cá nhân hóa, giúp người học tối ưu hóa thời gian, chi phí và tiếp cận nguồn tài liệu phong phú toàn cầu. Trong kỷ nguyên số và cuộc cách mạng công nghiệp 4.0, việc học tập suốt đời và liên tục cập nhật kỹ năng là chìa khóa thành công, làm gia tăng giá trị của các sản phẩm giáo dục số như khóa học trực tuyến, chương trình đào tạo chuyên sâu và các nền tảng chia sẻ tri thức.

Thị trường E-learning ngày càng cạnh tranh, đòi hỏi các nhà phát triển phải liên tục đổi mới, ứng dụng công nghệ tiên tiến để mang lại trải nghiệm tốt nhất cho người dùng. Một website cung cấp khóa học được xây dựng trên nền tảng công nghệ hiện đại, với kiến trúc vững chắc, giao diện người dùng (UI) trực quan, trải nghiệm người dùng (UX) mượt mà, và hiệu năng cao là yếu tố then chốt để thu hút và giữ chân người học.

Người học hiện đại thường mong muốn tìm kiếm thông tin khóa học một cách nhanh chóng, so sánh chất lượng, và nhận được sự hỗ trợ kịp thời. Việc xây dựng một website thương mại điện tử chuyên biệt cho các khóa học và sản phẩm giáo dục số như "ucademy-app", với các công nghệ tiên tiến, sẽ đáp ứng hiệu quả những nhu cầu này. Đề tài "Xây dựng website thương mại điện tử bán khóa học có tích hợp chatbot", tập trung vào việc phát triển một nền tảng như vậy, là một mục tiêu có tính thời sự và thực tiễn cao.

Việc tích hợp chatbot vào nền tảng "ucademy-app" mang lại giá trị gia tăng đáng kể. Chatbot đóng vai trò trợ lý ảo, sẵn sàng 24/7 để hỗ trợ người dùng tìm kiếm khóa học, giải đáp thắc mắc về nội dung, thủ tục đăng ký, thanh toán, chính sách, hoặc các vấn đề kỹ thuật cơ bản. Điều này không chỉ cải thiện trải nghiệm người dùng mà còn tối ưu hóa nguồn lực hỗ trợ. Việc lựa chọn mô hình phù hợp cho chatbot (ví dụ: Bag of Words cho khởi đầu, hoặc các giải pháp AI tiên tiến hơn) sẽ được cân nhắc để đảm bảo

hiệu quả.

Đề tài này có tính ứng dụng cao, là cơ hội để kết hợp và làm chủ các công nghệ full-stack JavaScript/TypeScript hiện đại: Next.js (React) và TypeScript cho việc xây dựng giao diện người dùng năng động, tối ưu cho SEO và hiệu suất; Node.js (ví dụ: sử dụng Express.js hoặc NestJS) cùng với TypeScript và Mongoose ODM cho việc phát triển backend API mạnh mẽ, linh hoạt, tương tác với cơ sở dữ liệu NoSQL MongoDB. Việc làm chủ các công nghệ này không chỉ giúp hoàn thành khóa luận mà còn mở ra nhiều cơ hội phát triển sự nghiệp trong ngành công nghệ phần mềm và EdTech.

1.2. Mục tiêu nghiên cứu:

- Nghiên cứu, phân tích yêu cầu để xây dựng một nền tảng E-learning ("ucademy-app") hoàn chỉnh, chuyên cung cấp các khóa học trực tuyến, đáp ứng nhu cầu đa dạng của người học và người dạy (nếu có).
- Thiết kế và phát triển một giao diện người dùng (Frontend) hiện đại, đáp ứng (responsive) bằng Next.js (React) và TypeScript. Tận dụng các tính năng ưu việt của Next.js như Server-Side Rendering (SSR), Static Site Generation (SSG), và API Routes để tối ưu hóa trải nghiệm người dùng, SEO và tốc độ tải trang. Các chức năng bao gồm: duyệt và tìm kiếm khóa học thông minh, xem chi tiết khóa học, quản lý giỏ hàng, quy trình thanh toán an toàn, quản lý hồ sơ cá nhân, theo dõi tiến độ học tập, và hệ thống tương tác/đánh giá.
- Xây dựng một hệ thống backend (Backend API) mạnh mẽ và có khả năng mở rộng bằng Node.js (ví dụ: Express.js hoặc NestJS) và TypeScript. Cung cấp các RESTful APIs hoặc GraphQL endpoints (tùy chọn) an toàn và hiệu quả cho frontend, xử lý logic nghiệp vụ phức tạp.
- Thiết kế và triển khai cơ sở dữ liệu NoSQL (MongoDB) sử dụng Mongoose ODM (Object Data Modeling) để quản lý schema và tương tác dữ liệu một cách hiệu quả, phù hợp với tính chất linh hoạt của dữ liệu khóa học, người dùng và các nội dung liên quan.

- Tích hợp và triển khai chatbot (ví dụ: sử dụng các thư viện NLP cơ bản, hoặc các dịch vụ chatbot AI) để cung cấp hỗ trợ tự động cho người dùng, giải đáp thắc mắc và hướng dẫn sử dụng nền tảng.
- Tích hợp cổng thanh toán trực tuyến an toàn và tiện lợi, hỗ trợ các phương thức thanh toán phổ biến.
- Xây dựng trang quản trị (Admin Dashboard) cho phép quản lý viên (và giảng viên nếu có) quản lý nội dung khóa học, người dùng, đơn hàng, và theo dõi các số liệu thống kê quan trọng.
- Thực hiện kiểm thử (Unit, Integration, End-to-End), đánh giá và tối ưu hóa hiệu năng, độ ổn định của website "ucademy-app" trước khi triển khai.

1.3 Phạm vi nghiên cứu:

- Tập trung nghiên cứu và phát triển các module chính của website thương mại điện tử "ucademy-app" trên nền tảng Next.js, Node.js, TypeScript, và MongoDB.
- Phát triển hệ thống quản lý nội dung học tập (LMS) cơ bản: tải lên và tổ chức bài giảng (video, text, tài liệu).
- Xây dựng hệ thống quản lý người dùng (học viên, giảng viên – nếu có, admin) với cơ chế xác thực và phân quyền sử dụng TypeScript xuyên suốt.
- Nghiên cứu cách xây dựng chatbot: từ việc thiết kế luồng hội thoại, xây dựng cơ sở tri thức (ví dụ: intents, entities cho mô hình Bag of Words hoặc các mô hình khác), đến tích hợp vào giao diện Next.js.
- Tích hợp cổng thanh toán: tập trung vào API của một nhà cung cấp cụ thể, đảm bảo luồng thanh toán an toàn và ghi nhận giao dịch vào MongoDB thông qua backend Node.js.
- Phạm vi đối tượng người dùng: những người có nhu cầu học tập và giảng dạy trực tuyến.
- Hoàn thành dự án theo quy trình phát triển phần mềm Agile (hoặc Waterfall tùy lựa chọn): từ lên ý tưởng, thiết kế, phát triển, kiểm thử đến triển khai.

1.4 Đối tượng nghiên cứu:

- Người dùng cuối (End-users):
 - Học viên: Nhu cầu học tập, hành vi tìm kiếm khóa học, trải nghiệm tương tác trên nền tảng Next.js.
 - Giảng viên (nếu có vai trò riêng): Nhu cầu về công cụ tạo và quản lý khóa học.
 - Quản trị viên: Yêu cầu về công cụ quản lý tổng thể nền tảng.
- Hệ thống (System):
 - Frontend (Next.js, TypeScript, React): Nghiên cứu các tính năng của Next.js (SSR, SSG, API Routes, Image Optimization), kiến trúc component trong React, quản lý state (ví dụ: Redux Toolkit, Zustand, Context API), routing, lazy loading, và các kỹ thuật tối ưu hiệu năng cho ứng dụng Next.js sử dụng TypeScript.
 - Backend (Node.js, TypeScript, Mongoose, ví dụ: Express.js/NestJS): Nghiên cứu cách xây dựng API hiệu quả và bảo mật với Node.js (và các framework phổ biến), ứng dụng TypeScript để tăng cường chất lượng code, sử dụng Mongoose để định nghĩa schema, thực hiện các thao tác CRUD, và quản lý quan hệ dữ liệu với MongoDB. Nghiên cứu các pattern thiết kế, middleware, xác thực (ví dụ: JWT).
 - Cơ sở dữ liệu (MongoDB, Mongoose): Nghiên cứu về mô hình dữ liệu document-based của MongoDB, cách thiết kế schema linh hoạt với Mongoose, các kỹ thuật đánh index, aggregation framework, và tối ưu hóa truy vấn cho MongoDB.
 - Chatbot: Nghiên cứu các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) cơ bản hoặc các API dịch vụ chatbot, cách thu thập và chuẩn bị dữ liệu huấn luyện, tích hợp chatbot với backend Node.js và hiển thị trên frontend Next.js.
 - Công thanh toán: Nghiên cứu API và quy trình tích hợp của một hoặc nhiều nhà cung cấp dịch vụ thanh toán.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 HTML, CSS và JavaScript:

HTML, CSS và JavaScript là những khái niệm không hề xa lạ với những người lập trình website chuyên nghiệp. Chúng chính là nền tảng không thể thiếu để xây dựng, tô vẽ và làm sống động cho chính những trang web. Và vì lẽ đó mà HTML, CSS, JavaScript là những khái niệm mà bắt buộc ai cũng phải có khi muốn bước chân vào con đường phát triển website.

2.1.1 HTML là gì?

2.1.2 CSS là gì?

2.1.3 JavaScript là gì?

2.2 TypeScript và NextJS:

Giới thiệu về việc ứng dụng hai công nghệ TypeScript và Next.js trong dự án, bao gồm các khái niệm và tính năng then chốt như TypeScript (một superset của JavaScript giúp bổ sung hệ thống kiểu tĩnh, hỗ trợ phát hiện lỗi sớm và lập trình hướng đối tượng), Next.js (một framework React mạnh mẽ cho phép rendering phía máy chủ (SSR) và tạo trang tĩnh (SSG), tối ưu hóa SEO và hiệu suất với App Router và API Routes), cùng với những lợi ích tổng hợp khi kết hợp chúng (nhằm xây dựng ứng dụng web hiện đại, an toàn và dễ bảo trì). Đồng thời, phần này cũng trình bày chi tiết các lý do cụ thể lựa chọn TypeScript để tăng tính an toàn dữ liệu và cải thiện khả năng bảo trì cho hệ thống bán khóa học, những ưu điểm vượt trội của Next.js trong việc phát triển ứng dụng full-stack, tối ưu hóa tốc độ tải trang và khả năng được tìm kiếm (SEO), và đánh giá sự phù hợp của bộ đôi công nghệ này khi tích hợp với các công cụ khác như Clerk (xác thực), Mongoose (cơ sở dữ liệu), và Vercel (triển khai) để đáp ứng các yêu cầu của dự án.

2.2.1 TypeScript là gì?

2.2.2 NextJS là gì?

2.2.3 Tại sao sử dụng TypeScript và NextJS trong dự án?

2.3 Mongoose:

Giới thiệu về Mongoose, một thư viện Object Data Modeling (ODM) mã nguồn mở phổ biến được thiết kế cho MongoDB trong môi trường Node.js, bao gồm các khái niệm và tính năng cốt lõi như khả năng định nghĩa Schema và Model để

áp đặt cấu trúc lên dữ liệu MongoDB, việc cung cấp các cơ chế xác thực dữ liệu (Validation) và Middleware (hooks) mạnh mẽ, cùng với sự hỗ trợ xây dựng truy vấn (Query Building) và tham chiếu dữ liệu liên collection (Population) một cách linh hoạt. Đồng thời, phần này cũng trình bày chi tiết những lý do chính lựa chọn Mongoose để mang lại cấu trúc rõ ràng cho cơ sở dữ liệu NoSQL và đơn giản hóa các thao tác CRUD trong dự án bán khóa học, vai trò quan trọng của Mongoose trong việc đảm bảo tính toàn vẹn dữ liệu thông qua các quy tắc xác thực và sự tích hợp chặt chẽ với TypeScript để phát triển an toàn, hiệu quả hơn, cùng với đánh giá về lợi ích của các tính năng nâng cao như populate và middleware trong ngữ cảnh ứng dụng, sự tương thích tốt khi xây dựng API routes trong Next.js, và sự hỗ trợ đặc lực từ cộng đồng lớn mạnh cũng như tài liệu phong phú của Mongoose.

2.3.1 Mongoose là gì?

2.3.2 Tại sao sử dụng Mongoose trong dự án?

2.4 Các module, thư viện, kiến thức cần thiết để xây dựng mạng nơ-ron, chatbot cho sản phẩm khóa luận:

Giới thiệu về các module, thư viện và kiến thức cần thiết để xây dựng chatbot cho sản phẩm khóa luận, bao gồm các công nghệ và công cụ cốt lõi như việc sử dụng mô hình ngôn ngữ lớn Gemini (cụ thể là phiên bản gemini-1.5-flash-latest) để xử lý ngôn ngữ tự nhiên và tạo phản hồi tương tác, ứng dụng framework Next.js cho việc phát triển toàn diện cả giao diện người dùng (frontend) và logic phía máy chủ (backend, đặc biệt qua Server Actions), cùng với vai trò của các thư viện và SDK thiết yếu như @google/genai (để giao tiếp với Gemini API), React (xây dựng UI), và Mongoose (nếu sử dụng MongoDB) cho việc tích hợp và vận hành hệ thống. Đồng thời, phần này cũng trình bày những lý do chiến lược đằng sau việc lựa chọn tổ hợp Gemini 1.5 Flash và Next.js nhằm tối ưu hóa tốc độ phản hồi, hiệu quả chi phí và đẩy nhanh tiến độ phát triển trong khuôn khổ một dự án học thuật, vai trò cụ thể của Next.js trong việc xây dựng kiến trúc full-stack cho chatbot và cách thức thư viện chủ chốt như @google/genai hỗ trợ tương tác với các dịch vụ AI bên ngoài, cũng như nhấn mạnh lợi ích của việc tận dụng hệ sinh thái các thư viện sẵn có để tăng tốc độ phát triển, giảm thiểu lỗi và

cho phép tập trung vào các yêu cầu nghiệp vụ chính của chatbot.

2.4.1 Giới thiệu về Kiến trúc Công nghệ cho Chatbot:

2.4.2 Công nghệ và Framework Cốt lõi:

2.4.2.1 Mô hình Ngôn ngữ Lớn Gemini:

2.4.2.2 Framework Web Next.js:

2.4.3 Các Module, Thư viện và SDK Thiết yếu:

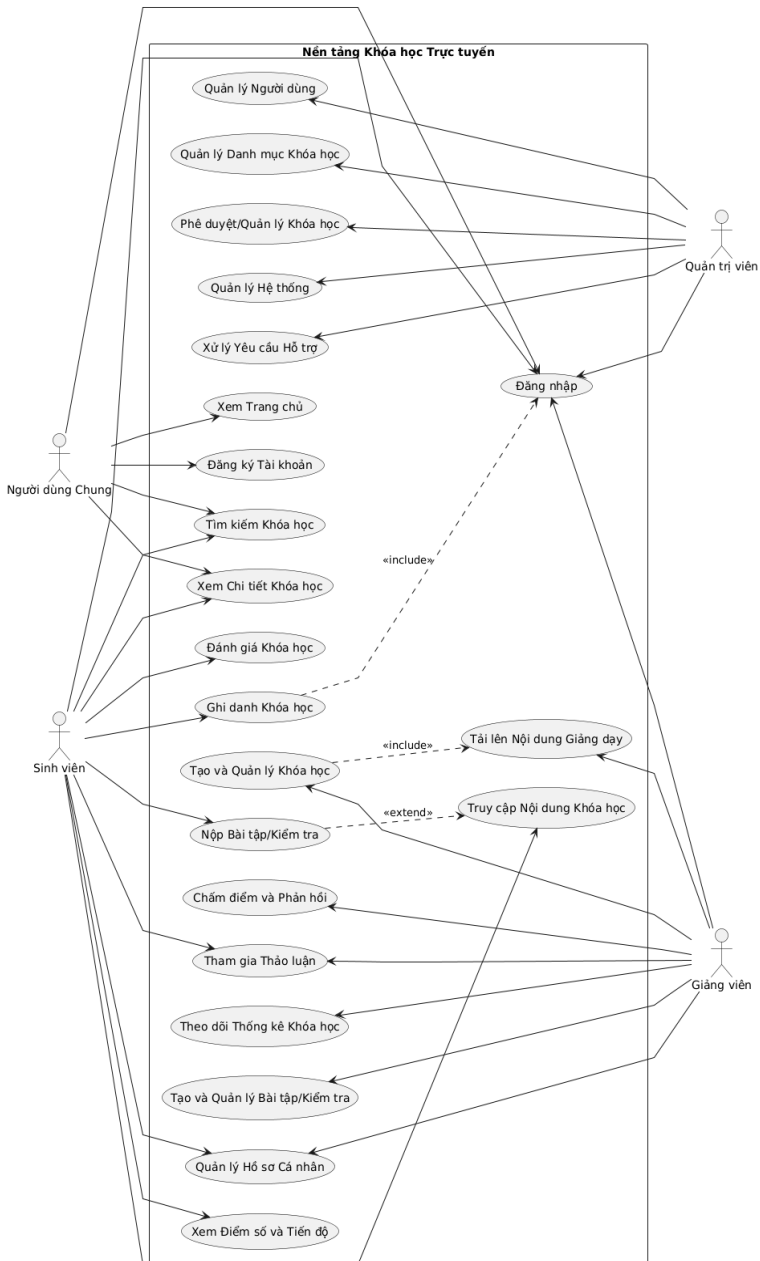
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Phương pháp phân tích hệ thống:

Giới thiệu về Ngôn ngữ Mô hình hóa Thống nhất (UML), một công cụ chuẩn được đề tài sử dụng để phân tích và thiết kế hệ thống quản lý khóa học trực tuyến, bao gồm các khía cạnh nền tảng như định nghĩa UML là một ngôn ngữ trực quan, đã được tiêu chuẩn hóa (ISO/IEC 19501), dùng để mô tả, minh họa, xây dựng và tài liệu hóa các hệ thống phần mềm, mục tiêu chính là cung cấp một phương tiện chung để phát triển, trao đổi mô hình và hỗ trợ mạnh mẽ cho các khái niệm hướng đối tượng cũng như các mẫu thiết kế phức tạp, cùng với các chức năng chủ yếu trong việc mô tả thiết kế khái niệm, quy trình, cấu trúc cơ sở dữ liệu và các thành phần phần mềm có khả năng tái sử dụng. Đồng thời, phần này cũng trình bày sự ứng dụng của các loại biểu đồ UML phổ biến để trực quan hóa hệ thống, bao gồm các biểu đồ cấu trúc như Biểu đồ Lớp (mô tả cấu trúc tĩnh) và Biểu đồ Thành phần (tổ chức các module phần mềm), các biểu đồ hành vi then chốt như Biểu đồ Use Case (thể hiện chức năng từ góc độ người dùng), Biểu đồ Tuần tự (mô tả tương tác theo thời gian) và Biểu đồ Hoạt động (minh họa luồng quy trình nghiệp vụ), cùng với vai trò quan trọng của chúng trong việc cung cấp một cái nhìn toàn diện, dễ hiểu và là cơ sở để trao đổi hiệu quả về thiết kế hệ thống phần mềm.

3.2 Mô tả yêu cầu bài toán:

3.3.1 Biểu đồ Usecase tổng quát:



3.3.2 Phân rã và đặc tả Usecase theo hướng chức năng:

Bao gồm các tiêu mục chứa các sơ đồ phân rã Usecase và đặc tả Usecase cho các chức năng tương ứng:

3.3.2.1 Chức năng đăng nhập:

3.3.2.2 Chức năng đăng ký tài khoản:

3.3.2.3 Chức năng tìm kiếm sản phẩm:

3.3.2.4 Chức năng thêm sản phẩm vào giỏ hàng:

3.3.2.5 Chức năng thanh toán:

3.3.2.6 Chức năng đánh giá sản phẩm:

3.3.2.7 Chức năng chỉnh sửa hồ sơ cá nhân:

3.3.2.8 Chức năng quản lý sản phẩm:

3.3.2.9 Chức năng quản lý đơn hàng:

3.3.2.10 Chức năng quản lý khách hàng:

3.3.2.11 Chức năng quản lý đánh giá:

3.3.2.12 Chức năng xem báo cáo thống kê:

3.3.2.13 Chức năng đổi mật khẩu:

3.3.3 Mô tả nghiệp vụ hệ thống bằng biểu đồ hoạt động:

Bao gồm các tiêu mục chứa các biểu đồ mô tả nghiệp vụ cho các chức năng tương ứng:

3.3.3.1 Chức năng đăng nhập:

3.3.3.2 Chức năng đăng ký tài khoản:

3.3.3.3 Chức năng tìm kiếm sản phẩm:

3.3.3.4 Chức năng thêm sản phẩm vào giỏ hàng:

3.3.3.5 Chức năng đặt hàng:

3.3.3.6 Chức năng thanh toán trực tuyến:

3.3.3.7 Chức năng cập nhật hồ sơ cá nhân:

3.3.3.8 Chức năng đánh giá sản phẩm:

3.3.3.9 Chức năng tạo sản phẩm mới:

3.3.3.10 Chức năng nhập hàng:

3.3.3.11 Chức năng chỉnh sửa thông tin sản phẩm:

3.3.3.12 Chức năng đổi mật khẩu:

3.3.4 Phân tích hành vi bằng biểu đồ tuần tự:

Bao gồm các tiêu mục chứa phân tích hành vi bằng biểu đồ tuần tự cho các chức năng tương ứng:

3.3.4.1 Chức năng đăng nhập:

3.3.4.2 Chức năng đăng ký tài khoản:

3.3.4.3 Chức năng tìm kiếm sản phẩm:

3.3.4.4 Chức năng thêm sản phẩm vào giỏ hàng:

3.3.4.5 Chức năng đặt hàng:

3.3.4.6 Chức năng thanh toán trực tuyến:

3.3.4.7 Chức năng đánh giá sản phẩm:

3.3.4.8 Chức năng chỉnh sửa hồ sơ cá nhân:

3.3.4.9 Chức năng thêm mới sản phẩm:

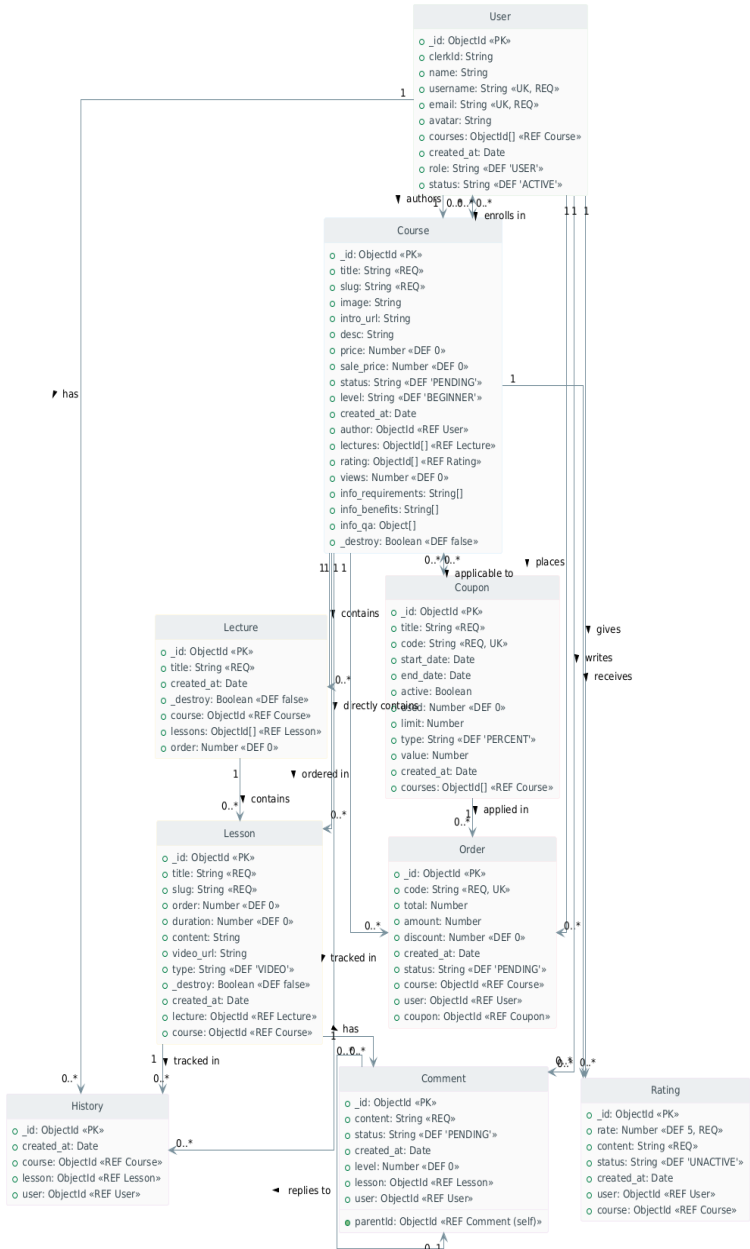
3.3.4.10 Chức năng chỉnh sửa thông tin sản phẩm:

3.3.4.11 Chức năng xem báo cáo thống kê:

3.3.4.12 Chức năng đổi mật khẩu:

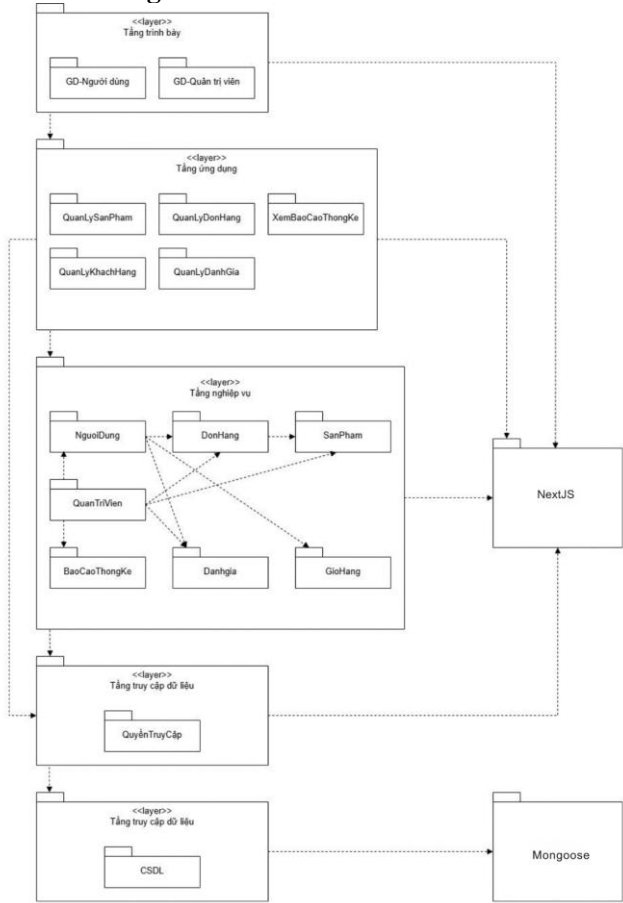
3.3.5. Phân tích cấu trúc bằng biểu đồ lớp

Cấu trúc của hệ thống ứng dụng bao gồm các đối tượng được mô tả qua biểu đồ lớp lĩnh vực như sau:

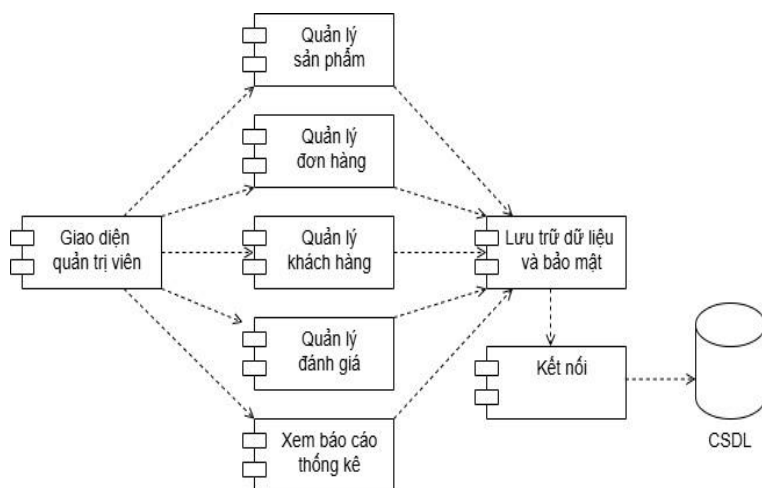


3.4 Thiết kế kiến trúc tổng thể:

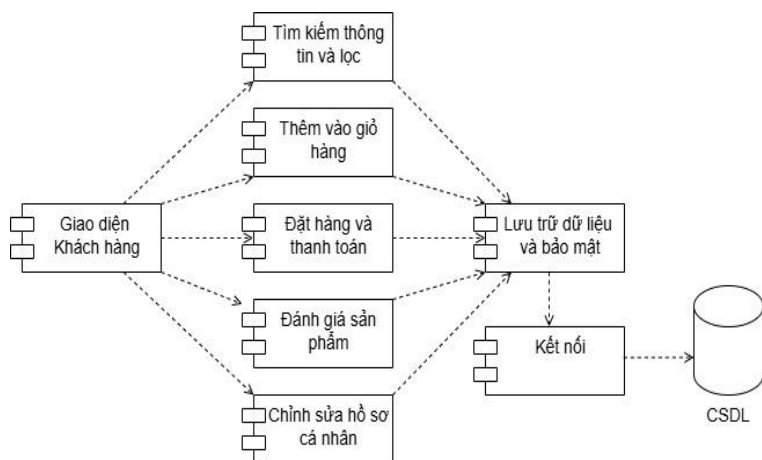
3.4.1 Biểu đồ gói:



3.4.2 Biểu đồ thành phần:

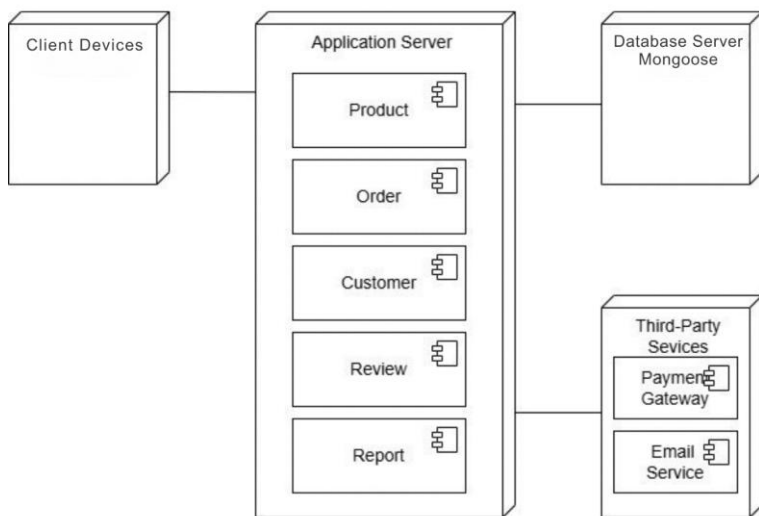


Biểu đồ thành phần của quản trị viên.



Biểu đồ thành phần của khách hàng.

3.4.3 Biểu đồ triển khai:



Biểu đồ triển khai hệ thống.

3.5 Thiết kế cơ sở dữ liệu:

Để thiết kế cơ sở dữ liệu cho hệ thống quản lý e-learning trực tuyến, đề tài sử dụng hệ thống cơ sở dữ liệu Mongoose là một hệ thống cơ sở dữ liệu NoSQL dựa trên cặp khóa (key) và giá trị (value). Trong Mongoose, mỗi tài liệu (document) được đặt trong một bộ sưu tập (collection) và được xác định bằng một khóa duy nhất. Dữ liệu trong mỗi tài liệu được tổ chức dưới dạng các cặp key-value, với mỗi key là một trường và giá trị tương ứng là nội dung của trường đó.

Cơ sở dữ liệu của hệ thống được thể hiện qua các bảng sau đây:

Bảng User: Lưu thông tin khách hàng của hệ thống.

Bảng Coupon: Lưu thông tin các mã giảm giá trong hệ thống.

Bảng Course: Lưu thông tin các khoá học trong hệ thống.

Bảng History: Lưu lịch sử khoá học của người dùng.

Bảng Lecture: Lưu thông tin các chương học trong khóa học.

Bảng Lesson: Lưu thông tin các bài học thuộc chương học trong hệ thống.

Bảng Order: Lưu thông tin các đơn hàng trong hệ thống.

Bảng Comment: Lưu thông tin các bình luận của người dùng.

Bảng Rating: Lưu thông tin các đánh giá của người dùng.

CHƯƠNG 4. TRIỂN KHAI THỬ NGHIỆM VÀ ĐÁNH GIÁ

4.1 Ứng dụng dành cho quản lý và người dùng:

Bao gồm các tiêu mục chứa các giao diện dành cho quản lý và người dùng cho các chức năng tương ứng:

- Giao diện trang đăng ký tài khoản:
- Giao diện trang đăng nhập tài khoản:
- Giao diện trang chủ khi chưa đăng nhập:
- Giao diện trang chủ người dùng chưa đăng nhập:
- Giao diện trang chủ đăng nhập Admin:
- Giao diện trang tìm kiếm khu vực khám phá:
- Giao diện trang chi tiết khóa học:
- Giao diện trang học tập:
- Giao diện trang tiếp tục học:
- Giao diện trang săn mã giảm giá:
- Giao diện trang thông tin cá nhân người dùng:
- Giao diện trang quản lý khóa học:
- Giao diện trang tìm kiếm và lọc khóa học:
- Giao diện trang tạo khóa học mới:
- Giao diện trang cập nhật khóa học:
- Giao diện trang quản lý thành viên:
- Giao diện trang cập nhật thông tin người dùng:
- Giao diện trang quản lý đơn hàng:
- Giao diện trang quản lý mã giảm giá:
- Giao diện trang tạo mã giảm giá:
- Giao diện trang quản lý đánh giá:
- Giao diện trang quản lý bình luận:
- Giao diện chatbot:

4.2 Đánh giá kết quả đạt được và phương hướng phát triển:

4.2.1 Kết quả đạt được:

Qua quá trình nghiên cứu và định hướng xây dựng nền tảng e-learning Ucademy, chúng ta đã phác thảo và có thể đạt được các kết quả quan trọng sau khi triển khai:

- Nền tảng công nghệ hiện đại và hiệu suất cao:

- Hệ thống Ucademy được xây dựng với Next.js, mang lại khả năng rendering linh hoạt (SSR, SSG, ISR) giúp tối ưu tốc độ tải trang và cải thiện SEO.
- Sử dụng Mongoose với MongoDB cho phép lưu trữ dữ liệu phi cấu trúc một cách linh hoạt, phù hợp với nội dung khóa học đa dạng và dữ liệu người dùng.
- Tích hợp TanStack Query (React Query) giúp đơn giản hóa việc tìm nạp, caching và đồng bộ hóa dữ liệu từ máy chủ, giảm thiểu các yêu cầu mạng không cần thiết và cải thiện đáng kể hiệu suất ứng dụng.
- Trải nghiệm người dùng (UI/UX) tiên tiến và nhất quán:
 - Giao diện Ucademy được thiết kế theo các xu hướng UI/UX mới nhất, bao gồm khả năng cá nhân hóa, chế độ tối, và các tương tác vi mô (micro-interactions) để tăng cường sự hấp dẫn và dễ sử dụng.
 - Ứng dụng Storybook được sử dụng để phát triển các thành phần UI một cách độc lập, đảm bảo tính nhất quán, khả năng tái sử dụng cao và tạo điều kiện thuận lợi cho việc kiểm thử cũng như tài liệu hóa.
 - Chú trọng đến thiết kế hòa nhập và khả năng tiếp cận, đảm bảo Ucademy có thể được sử dụng bởi đa dạng người học.
- Tích hợp Trí tuệ Nhân tạo (AI) bước đầu:
 - Triển khai Chatbot AI thông minh, sử dụng công nghệ xử lý ngôn ngữ tự nhiên, để hỗ trợ người học 24/7, trả lời các câu hỏi thường gặp và cung cấp hướng dẫn cơ bản.
- Bảo mật và Thanh toán:
 - Áp dụng các biện pháp bảo mật nền tảng, với khuyến nghị mạnh mẽ về việc triển khai Xác thực Đa Yếu tố (MFA) để bảo vệ tài khoản người dùng và dữ liệu của Ucademy.
 - Định hướng tích hợp các cổng thanh toán phổ biến tại thị trường Việt Nam như MoMo, VNPay, ZaloPay và các tùy chọn ngân hàng trực tuyến, thẻ quốc tế, nhằm mang lại sự tiện lợi tối đa cho người học.

- Khả năng mở rộng và bảo trì:
 - Kiến trúc hệ thống được thiết kế với các cân nhắc về khả năng mở rộng quy mô (cả chiều dọc và chiều ngang), bao gồm clustering, cân bằng tải, và tiềm năng sử dụng containerization (Docker, Kubernetes) trong tương lai.
 - Cơ sở dữ liệu MongoDB được tối ưu hóa thông qua việc đánh chỉ mục (indexing) và các truy vấn hiệu quả.

4.2.2 Hạn chế và phương hướng phát triển:

Mặc dù Ucademy được định hướng với nhiều ưu điểm vượt trội, vẫn có những khía cạnh cần tiếp tục phát triển và hoàn thiện:

- Hạn chế hiện tại (dựa trên kế hoạch):
 - Cá nhân hóa sâu bằng AI: Hệ thống hiện tại mới chỉ dừng ở chatbot AI cơ bản. Các tính năng cá nhân hóa sâu sắc hơn như lộ trình học tập thích ứng, tuyển chọn nội dung siêu cá nhân hóa, hay tạo học phần vi mô bằng AI chưa được triển khai.
 - Tối ưu hóa hiệu suất ở quy mô lớn: Khi số lượng người học, khóa học và dữ liệu tăng đột biến, cần liên tục theo dõi và tối ưu hóa hiệu suất hệ thống, bao gồm tốc độ xử lý và truy xuất thông tin.
 - Mở rộng tích hợp: Khả năng tích hợp với các nền tảng bên thứ ba khác (ví dụ: các hệ thống quản lý học tập doanh nghiệp (LMS), các công cụ phân tích dữ liệu học tập chuyên sâu) còn hạn chế.
 - Bảo mật nâng cao: Mặc dù các biện pháp bảo mật cơ bản và MFA được khuyến nghị, việc đối phó với các mối đe dọa an ninh mạng ngày càng tinh vi đòi hỏi sự cập nhật và nâng cấp liên tục.
- Phương hướng phát triển trong tương lai:
 - Nâng cao Trí tuệ Nhân tạo (AI):
 - Tích hợp công nghệ AI để cung cấp lộ trình học tập thích ứng, gợi ý khóa học/bài học dựa trên sở thích cá nhân, lịch sử học tập và hiệu suất của người học.

- Phát triển hoặc tích hợp các công cụ AI tạo nội dung (ví dụ: tạo câu hỏi ôn tập, tóm tắt bài học) để hỗ trợ người tạo khóa học.
- Nâng cấp chatbot AI để có khả năng hiểu sâu hơn về ngữ cảnh, hỗ trợ giải đáp các vấn đề phức tạp hơn và cung cấp phản hồi mang tính xây dựng.
- Tối ưu hóa và Mở rộng Quy mô:
 - Triển khai các chiến lược caching nâng cao hơn nữa (ví dụ: Redis cho caching toàn trang, fragment caching) để giảm tải cho cơ sở dữ liệu và tăng tốc độ phản hồi.
 - Xem xét áp dụng kiến trúc microservices khi Ucademy phát triển đủ lớn để tăng tính linh hoạt và khả năng mở rộng độc lập của từng thành phần.
- Mở rộng Hệ sinh thái Tích hợp:
 - Tích hợp thêm các phương thức thanh toán quốc tế và các đối tác vận chuyển (nếu có sản phẩm vật lý đi kèm).
 - Liên kết với các mạng xã hội học thuật hoặc các diễn đàn chuyên ngành để mở rộng cộng đồng người học và quảng bá khóa học.
 - Cung cấp API để các đối tác hoặc doanh nghiệp có thể tích hợp Ucademy vào hệ thống của họ.
- Tăng cường Bảo mật Toàn diện:
 - Áp dụng các công nghệ bảo mật tiên tiến hơn như MFA thích ứng (adaptive MFA), giám sát hành vi người dùng bất thường.
 - Thực hiện kiểm tra lỗ hổng bảo mật (penetration testing) định kỳ và cập nhật các bản vá lỗi kịp thời.
- Đổi mới Trải nghiệm Người dùng:
 - Khám phá và thử nghiệm các xu hướng UI/UX mới như Giao diện Người dùng bằng Giọng nói (VUI) cho một số tính năng nhất định, hoặc tích hợp các yếu tố AR/MR cho các nội dung học tập

phù hợp (ví dụ: mô phỏng 3D) trong tương lai xa hơn.

Bằng cách tập trung vào những phương hướng phát triển này, Ucademy không chỉ khắc phục được những hạn chế hiện tại mà còn có thể khẳng định vị thế là một nền tảng e-learning tiên tiến, cá nhân hóa và hiệu quả cao.

CHƯƠNG 5. KIỂM THỬ HỆ THỐNG

5.1 Mục tiêu Kiểm thử:

Kiểm thử là một giai đoạn quan trọng không thể thiếu trong quy trình phát triển phần mềm, đặc biệt đối với một nền tảng e-learning phức tạp như Ucademy. Mục tiêu chính của giai đoạn kiểm thử bao gồm:

- **Đảm bảo chất lượng và độ tin cậy:** Xác minh rằng tất cả các chức năng của Ucademy hoạt động đúng như thiết kế và yêu cầu, từ việc đăng ký người dùng, tạo và quản lý khóa học, đến các tính năng tương tác như chatbot AI và thanh toán.
- **Nâng cao trải nghiệm người dùng (UX):** Đảm bảo Ucademy dễ sử dụng, trực quan, có hiệu suất tốt và tương thích trên nhiều thiết bị và trình duyệt khác nhau.
- **Phát hiện và khắc phục lỗi sớm:** Tìm kiếm và sửa lỗi ở giai đoạn sớm nhất có thể để giảm thiểu chi phí và rủi ro khi triển khai.
- **Đảm bảo an toàn và bảo mật:** Xác định và vá các lỗ hổng bảo mật tiềm ẩn, bảo vệ dữ liệu người dùng và tính toàn vẹn của hệ thống.
- **Xác minh hiệu năng hệ thống:** Đánh giá khả năng chịu tải, tốc độ phản hồi và sự ổn định của Ucademy dưới các điều kiện hoạt động khác nhau.
- **Kiểm tra tính tuân thủ (nếu có):** Đối với các nền tảng e-learning, cần đảm bảo tuân thủ các tiêu chuẩn như SCORM, AICC nếu có yêu cầu về việc chia sẻ nội dung hoặc tích hợp với các Hệ thống Quản lý Học tập (LMS) khác.

5.2 Chiến lược Kiểm thử:

Chiến lược kiểm thử cho Ucademy được xây dựng dựa trên mô hình kim tự tháp kiểm thử (Testing Pyramid), tập trung vào các cấp độ khác nhau để đảm bảo phạm vi bao phủ toàn diện:

1. **Kiểm thử Đơn vị (Unit Testing):** Tập trung vào việc kiểm thử các thành phần nhỏ nhất, độc lập của mã nguồn (ví dụ: các hàm, components React trong Next.js).
2. **Kiểm thử Tích hợp (Integration Testing):** Kiểm tra sự tương tác giữa các thành phần, module khác nhau của hệ

thông (ví dụ: tương tác giữa frontend Next.js và API backend, kết nối với cơ sở dữ liệu MongoDB).

3. Kiểm thử Hệ thống/Đầu cuối (End-to-End Testing): Kiểm thử toàn bộ luồng hoạt động của ứng dụng từ góc độ người dùng, mô phỏng các kịch bản sử dụng thực tế.
4. Kiểm thử Phi chức năng (Non-functional Testing): Bao gồm kiểm thử hiệu năng, bảo mật, tính khả dụng, và tính tương thích.

5.3 Các Loại Kiểm thử Được Thực Hiện:

5.3.1. Kiểm thử Đơn vị (Unit Testing):

- Mục tiêu: Đảm bảo từng thành phần (component React, hàm JavaScript, module backend) của Ucademy hoạt động chính xác và độc lập.
- Phạm vi:
 - Các components giao diện người dùng (UI) được xây dựng bằng Next.js và React (ví dụ: CourseCard, LessonPlayer, ChatBubble).
 - Các hàm tiện ích (utility functions).
 - Các logic xử lý nghiệp vụ đơn lẻ trong API routes của Next.js.
- Công cụ:
 - Jest: Framework kiểm thử JavaScript phổ biến, được Next.js hỗ trợ tích hợp sẵn.
 - React Testing Library: Thư viện giúp kiểm thử các React component theo cách người dùng tương tác với chúng.
- Ví dụ kịch bản:
 - Kiểm tra component Button hiển thị đúng nhãn và thực thi đúng hàm onClick khi được nhấn.
 - Kiểm tra một hàm tính toán điểm số trả về kết quả chính xác với các đầu vào khác nhau.
 - Kiểm tra một Server Component đồng bộ render đúng nội dung. (Lưu ý: Async Server Components hiện tại được khuyến nghị kiểm thử bằng E2E).

5.3.2 Kiểm thử Tích hợp (Integration Testing):

- Mục tiêu: Xác minh sự tương tác chính xác giữa các module, thành phần khác nhau của Ucademy.

- Phạm vi:
 - Tương tác giữa các UI components (ví dụ: form nhập liệu và nút submit).
 - Tương tác giữa frontend (Next.js) và backend API routes (ví dụ: gửi yêu cầu tạo khóa học và nhận phản hồi).
 - Tương tác giữa API routes và cơ sở dữ liệu Mongoose (ví dụ: lưu trữ, truy vấn dữ liệu người dùng, khóa học).
- Công cụ:
 - Jest: Có thể sử dụng để kiểm thử API routes.
 - Supertest (hoặc tương tự): Thư viện giúp kiểm thử các HTTP endpoint.
 - Mocking Libraries (ví dụ: Jest Mocks): Để giả lập các phụ thuộc bên ngoài như cơ sở dữ liệu hoặc dịch vụ bên thứ ba trong quá trình kiểm thử API. (Lưu ý: Mongoose khuyến nghị Mocha hơn Jest cho kiểm thử phía server, nhưng Jest vẫn có thể sử dụng với các cấu hình phù hợp).
- Ví dụ kịch bản:
 - Kiểm tra khi người dùng điền form đăng ký và nhấn nút "Đăng ký", dữ liệu được gửi thành công lên API và một tài khoản mới được tạo trong cơ sở dữ liệu Mongoose.
 - Kiểm tra API route /api/courses trả về danh sách khóa học chính xác từ MongoDB.
 - Kiểm tra việc mock API call khi một component cần fetch dữ liệu.

5.3.3 Kiểm thử Đầu cuối (End-to-End - E2E Testing):

- Mục tiêu: Mô phỏng các luồng người dùng thực tế trên Ucademy từ đầu đến cuối, đảm bảo toàn bộ hệ thống hoạt động trơn tru.
- Phạm vi: Các kịch bản người dùng quan trọng như:
 - Luồng đăng ký, đăng nhập, quên mật khẩu.
 - Luồng tìm kiếm và xem chi tiết khóa học.
 - Luồng ghi danh vào khóa học và thanh toán.
 - Luồng học bài, hoàn thành bài học, xem tiến độ.

- Tương tác với chatbot AI.
- Công cụ:
 - Cypress: Framework E2E testing phổ biến, mạnh mẽ với khả năng tương tác trực quan.
 - Playwright: Framework E2E testing hiện đại từ Microsoft, hỗ trợ đa trình duyệt tốt.
- Ví dụ kịch bản:
 - Người dùng truy cập trang chủ, tìm kiếm khóa học "Next.js cơ bản", xem chi tiết, thêm vào giỏ hàng, tiến hành thanh toán qua VNPAY, và được ghi danh thành công vào khóa học.
 - Người dùng đăng nhập, vào một khóa học đã ghi danh, xem một video bài giảng, đánh dấu hoàn thành bài học, và thấy tiến độ được cập nhật.

5.3.4 Kiểm thử Hiệu năng (Performance Testing):

- Mục tiêu: Đánh giá tốc độ phản hồi, sự ổn định và khả năng mở rộng của Ucademy dưới các mức tải khác nhau.
- Phạm vi:
 - Thời gian tải trang (trang chủ, trang khóa học, trang bài học).
 - Thời gian phản hồi của API khi có nhiều yêu cầu đồng thời.
 - Khả năng chịu tải của hệ thống khi có lượng lớn người dùng truy cập cùng lúc (Stress Testing).
 - Khả năng mở rộng của cơ sở dữ liệu Mongoose.
- Công cụ:
 - k6: Công cụ kiểm thử tải hiện đại, mã nguồn mở, sử dụng JavaScript để viết kịch bản, phù hợp cho CI/CD.
 - Apache JMeter: Công cụ kiểm thử tải mạnh mẽ, hỗ trợ nhiều giao thức.
 - Next.js Analytics / Vercel Analytics: Để theo dõi hiệu suất thực tế sau triển khai.
- Ví dụ kịch bản:
 - Mô phỏng 1000 người dùng đồng thời truy cập và duyệt các khóa học trên Ucademy, đo thời gian phản hồi trung bình và tỷ lệ lỗi.

- Kiểm tra thời gian API `/api/lessons/{id}` trả về nội dung bài học khi có 500 yêu cầu/giây.

5.4 Môi trường và Công cụ Kiểm thử Tổng hợp:

- Môi trường Phát triển (Development): Kiểm thử viên và lập trình viên thực hiện kiểm thử đơn vị và tích hợp sớm.
- Môi trường Staging/Kiểm thử Chuyên dụng: Một môi trường gần giống Production nhất để thực hiện kiểm thử hệ thống, E2E, hiệu năng và UAT.
- Công cụ Quản lý Test Case và Báo cáo Lỗi: Jira, TestRail, hoặc các công cụ tương tự.
- Hệ thống Quản lý Phiên bản: Git (GitHub, GitLab, Bitbucket).
- CI/CD Pipeline: Jenkins, GitLab CI, GitHub Actions để tự động hóa việc chạy test sau mỗi lần commit code.

5.5 Quy trình Kiểm thử:

1. Lập kế hoạch kiểm thử (Test Planning): Xác định mục tiêu, phạm vi, nguồn lực, lịch trình, và các loại kiểm thử cần thực hiện.
2. Thiết kế Test Case (Test Case Design): Viết các kịch bản kiểm thử chi tiết dựa trên yêu cầu chức năng và phi chức năng. Các test case cần rõ ràng, cụ thể, bao gồm các bước thực hiện, dữ liệu đầu vào, và kết quả mong đợi.
3. Chuẩn bị Môi trường Kiểm thử (Test Environment Setup): Cài đặt và cấu hình môi trường, dữ liệu kiểm thử.
4. Thực thi Kiểm thử (Test Execution): Chạy các test case đã thiết kế (cả thủ công và tự động).
5. Ghi nhận và Báo cáo Kết quả (Result Logging and Reporting): Ghi lại kết quả thực tế, so sánh với kết quả mong đợi. Báo cáo các lỗi phát hiện được lên hệ thống quản lý lỗi.
6. Theo dõi và Sửa lỗi (Bug Tracking and Fixing): Lập trình viên sửa lỗi, sau đó kiểm thử viên kiểm tra lại (Regression Testing).
7. Kiểm thử Hồi quy (Regression Testing): Đảm bảo các thay đổi hoặc sửa lỗi không ảnh hưởng đến các chức năng đã hoạt động đúng trước đó.

8. Kiểm thử Chấp nhận Người dùng (User Acceptance Testing - UAT) (nếu có): Người dùng cuối hoặc đại diện của họ kiểm thử hệ thống để đảm bảo đáp ứng nhu cầu.
9. Tổng kết và Đánh giá (Test Summary and Evaluation): Đánh giá toàn bộ quá trình kiểm thử, rút kinh nghiệm.

5.6 Kết quả Kiểm thử và Phân tích (Dự kiến):

Phần này trong khóa luận thực tế sẽ trình bày các kết quả cụ thể thu được từ quá trình kiểm thử. Tuy nhiên, ở giai đoạn đề xuất, chúng ta có thể phác thảo những gì mong đợi.

Dự kiến, quá trình kiểm thử toàn diện sẽ giúp:

- Phát hiện và khắc phục một số lượng đáng kể các lỗi ở các mức độ khác nhau (ví dụ: lỗi giao diện, lỗi logic nghiệp vụ, lỗi hiệu năng, lỗ hổng bảo mật).
- Xác nhận rằng các chức năng cốt lõi của Ucademy như quản lý khóa học, ghi danh, học tập, và thanh toán hoạt động ổn định và chính xác.
- Đảm bảo chatbot AI có khả năng hiểu và phản hồi các truy vấn phổ biến của người dùng một cách hiệu quả.
- Cung cấp các số liệu cụ thể về hiệu năng hệ thống (ví dụ: thời gian tải trang trung bình, số lượng người dùng đồng thời tối đa mà hệ thống có thể xử lý).
- Đưa ra các đề xuất cải thiện về giao diện người dùng và trải nghiệm tổng thể dựa trên phản hồi từ usability testing.
- Xác nhận mức độ bảo mật của ứng dụng trước các mối đe dọa phổ biến.

Các kết quả sẽ được trình bày một cách khách quan, có thể sử dụng bảng biểu, đồ thị để minh họa (ví dụ: số lượng test case pass/fail, biểu đồ thời gian phản hồi).

5.7 Hạn chế của Quá trình Kiểm thử:

Mặc dù đã nỗ lực thực hiện kiểm thử một cách toàn diện, quá trình này vẫn có thể có một số hạn chế nhất định:

- Giới hạn về thời gian và nguồn lực: Không thể thực hiện kiểm thử toàn bộ (exhaustive testing) mọi kịch bản và mọi tổ hợp dữ liệu đầu vào.

- Độ phức tạp của hệ thống: Với các tính năng AI và tích hợp đa dạng, một số trường hợp lỗi đặc biệt hoặc hiếm gặp có thể chưa được phát hiện.
- Môi trường kiểm thử: Môi trường staging dù cố gắng mô phỏng production nhưng vẫn có thể có những khác biệt nhỏ, dẫn đến một số lỗi chỉ xuất hiện trên môi trường thực tế.
- Sự thay đổi liên tục: Nếu dự án vẫn đang trong quá trình phát triển song song với kiểm thử, các thay đổi mới có thể ảnh hưởng đến kết quả kiểm thử hoặc yêu cầu kiểm thử lại.
- Kiểm thử AI: Việc kiểm thử các mô hình AI, đặc biệt là NLP cho chatbot, là một thách thức và khó có thể bao phủ hết mọi biến thể trong ngôn ngữ tự nhiên của người dùng.

KẾT LUẬN

Trong suốt quá trình nghiên cứu, khảo sát và phân tích thiết kế, em đã nhận thấy được tầm quan trọng của đề tài khi ứng dụng thực tế vào việc xây dựng các nền tảng học tập trực tuyến. Em xin gửi lời cảm ơn chân thành tới các thầy, các cô trong khoa Công nghệ thông tin, trường Đại học Sài Gòn, đặc biệt là giảng viên hướng dẫn, thầy TS. Cao Thái Phương Thanh, đã giúp đỡ em rất nhiều trong quá trình tìm hiểu, tiếp cận và hoàn thành đề tài: “*Xây dựng hệ thống E-Learning bằng NextJS*”.

Qua quá trình thực hiện, em đã có cơ hội học hỏi thêm, tiếp xúc và ứng dụng các công nghệ, công cụ hiện đại như NextJS, Mongoose,... qua đó hiểu rõ hơn tầm quan trọng của chúng trong việc xây dựng một hệ thống E-Learning hiệu quả và đáp ứng nhu cầu người dùng. Đề tài không chỉ tập trung vào việc triển khai một hệ thống E-Learning cơ bản mà còn mở ra những hướng phát triển tiềm năng cho tương lai, với khả năng mở rộng và tích hợp thêm nhiều tính năng học tập tiên tiến.

Tuy nhiên, vì lý do giới hạn về thời gian và phạm vi triển khai nên trong quá trình xây dựng và phát triển, đề tài không thể tránh khỏi những thiếu sót. Vậy nên em rất mong nhận được sự thông cảm và góp ý từ thầy cô. Với những kiến thức đã được học thông qua các bài giảng tại trường, các kinh nghiệm thực tế thông qua dự án này, em sẽ tiếp tục trau dồi, nâng cao kiến thức và kỹ năng của bản thân để có thể hoàn thiện đề tài này một cách trọn vẹn hơn trong tương lai.