**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC QUY NHƠN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC HÀNH ĐỒ ÁN**

Đề tài: Quản Lý Khoá Học

Sinh viên thực hiện: Đặng Thành Công

MSSV: 4451190096

Lớp: Kỹ Thuật Phần Mền K44

Giảng viên hướng dẫn: ThS.Trần Hoàng Việt

Mục lục

[1.1 Giới thiệu tổng quát 1](#_Toc29006)

[1.2 Mục đích đề tài 1](#_Toc21595)

[1.3 Đối tượng và phạm vi nguyên cứu 2](#_Toc12564)

[1.4 Phương pháp nguyên cứu 3](#_Toc5049)

[1.5 Nội dụng thực hành đề tài 4](#_Toc8714)

[1.5.1 Phân tích yêu cầu 4](#_Toc12875)

[1.5.2 Thiết kế hệ thống 4](#_Toc30235)

[1.5.3 Phát triển ứng dụng 5](#_Toc6749)

[1.5.4 Tối ưu và cải tiến 5](#_Toc13873)

[1.5.5 Hướng dẫn sử dụng và triển khai 5](#_Toc474)

[1.6 Môi trường ứng dụng 5](#_Toc22180)

[1.6.1 Môi trường phát triển 5](#_Toc12646)

[1.6.2 Môi trường triển khai 6](#_Toc21617)

[1.6.3 Môi trường phát triển phần mền 6](#_Toc21839)

[CHƯƠNG 2: Cơ Sở Lý Thuyết 7](#_Toc18662)

[2 Cơ sở lý thuyết 7](#_Toc12026)

[2.1 Ngôn ngữ lập trình C# 7](#_Toc4065)

[2.2 Giới thiệu về ASP.NET Core MVC 7](#_Toc29364)

[2.2.1 ASP.NET Core MVC là gì? 7](#_Toc15574)

[2.2.2 Ưu điểm của ASP.NET Core MVC 8](#_Toc15630)

[2.2.3 Các đặc tính cơ bản của ASP.NET Core MVC 8](#_Toc15672)

[2.3 Giới thiệu về ASP.NET Core Identity và Bảo mật 9](#_Toc9271)

[2.3.1 ASP.NET Core Identity là gì? 9](#_Toc15097)

[2.3.2 Cơ chế hoạt động của ASP.NET Core Identity 9](#_Toc19685)

[2.3.3 Các thành phần cơ bản của ASP.NET Core Identity 9](#_Toc10301)

[2.3.4 Ưu và nhược điểm của ASP.NET Core Identity 10](#_Toc26576)

[2.4 Hệ quản trị cơ sở dữ liệu SQL Server 10](#_Toc25974)

[2.4.1 SQL Server là gì? 10](#_Toc25555)

[2.4.2 Các ưu điểm của công cụ SQL Server 10](#_Toc15742)

[CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 11](#_Toc16932)

[3 Mô tả bài toán 11](#_Toc3951)

[3.1 Sơ đồ use case 11](#_Toc9543)

[3.2 Đặc tả use case 11](#_Toc31813)

[3.2.1 Use case đăng nhập 11](#_Toc14506)

[US\_01 11](#_Toc24209)

[Đăng Nhập 11](#_Toc15881)

[3.2.2 Use case Đăng ký 12](#_Toc10986)

[US\_02 12](#_Toc5311)

[Đăng ký 12](#_Toc22176)

[User 12](#_Toc25552)

[3.2.3 Use case quên mật khẩu 13](#_Toc4347)

[US\_03 13](#_Toc25526)

[User 13](#_Toc4988)

[3.2.4 Use case thay đổi mật khẩu 14](#_Toc10295)

[US\_04 14](#_Toc27151)

[User 14](#_Toc8159)

[3.2.5 Use case tìm kiếm kháo học 15](#_Toc23063)

[US\_05 15](#_Toc28517)

[User 15](#_Toc7883)

[3.2.6 Use case xem danh sách khoá học 16](#_Toc20760)

[US\_06 16](#_Toc3431)

[Đăng ký 16](#_Toc24771)

[User 16](#_Toc2053)

[3.2.7 Use case tham gia khoá học 17](#_Toc16301)

[US\_07 17](#_Toc7192)

[Đăng ký 17](#_Toc6562)

[User 17](#_Toc20954)

[3.2.8 Use case tạo khoá học 18](#_Toc6943)

[US\_08 18](#_Toc7224)

[Đăng ký 18](#_Toc10785)

[User 18](#_Toc5370)

[3.2.9 Use case sửa thông tin khoá học 19](#_Toc32292)

[US\_09 19](#_Toc20276)

[Đăng ký 19](#_Toc22864)

[User 19](#_Toc6727)

[3.2.10 Use case xoá khoá học 20](#_Toc29678)

[US\_010 20](#_Toc8707)

[Đăng ký 20](#_Toc29845)

[User 20](#_Toc31903)

[3.2.11 Use case xem bài tập 21](#_Toc25264)

[US\_011 21](#_Toc32161)

[Đăng ký 21](#_Toc11855)

[User 21](#_Toc7413)

[3.2.12 Use case thêm bài tập 22](#_Toc32607)

[US\_012 22](#_Toc31679)

[Đăng ký 22](#_Toc20410)

[User 22](#_Toc7335)

[3.2.13 Use case xoá bài tập 23](#_Toc11958)

[US\_013 23](#_Toc21433)

[Đăng ký 23](#_Toc16915)

[User 23](#_Toc21003)

[3.3 Sơ đồ hoạt động 24](#_Toc30588)

[3.3.1 Đăng nhập 24](#_Toc5064)

[3.3.2 Đăng ký 24](#_Toc13052)

[3.3.3 Quên mật khẩu 25](#_Toc27467)

[3.3.4 Thay đổi mật khẩu 25](#_Toc27145)

[3.3.5 Tìm kiếm khoá học 26](#_Toc31353)

[3.3.6 Xem danh sách khoá học 26](#_Toc22561)

[3.3.7 Tham gia khoá học 27](#_Toc13172)

[3.3.8 Sửa thông tin khoá học 27](#_Toc8455)

[3.3.9 Xoá khoá học 28](#_Toc1117)

[3.3.10 Thêm khoá học 28](#_Toc5660)

[3.3.11 Xem bài tập 29](#_Toc11721)

[3.3.12 Thêm bài tập 29](#_Toc16246)

[3.3.13 Xoá bài tập 30](#_Toc514)

[3.4 Sơ đồ tuần tự 31](#_Toc3734)

[3.4.1 Đăng nhập 31](#_Toc31581)

[3.4.2 Đăng ký 31](#_Toc4082)

[3.4.3 Quên mật khẩu 32](#_Toc18979)

[3.4.4 Thay đổi mật khẩu 32](#_Toc22959)

[3.4.5 Tìm kiếm khoá học 32](#_Toc16097)

[3.4.6 Xem danh sách khoá học 33](#_Toc10553)

[3.4.7 Tham gia khoá học 33](#_Toc23580)

[3.4.8 Tạo khoá học 33](#_Toc28320)

[3.4.9 Sửa thông tin khoá học 34](#_Toc29370)

[3.4.10 Xoá khoá học 34](#_Toc15336)

[3.4.11 Xem bài tập 35](#_Toc1696)

[3.4.12 Xoá bài tâp 35](#_Toc32671)

[3.4.13 Thêm bài tập 35](#_Toc25943)

[3.5 Thiết kế cơ sở dữ liệu 36](#_Toc28037)

[3.5.1 Cơ sở dữ liệu tổng quát 36](#_Toc26286)

[3.5.2 Bảng User 37](#_Toc19716)

[3.5.3 Bảng Course 38](#_Toc20417)

[3.5.4 Bảng Reviews 38](#_Toc25794)

[3.5.5 Bảng Enrollment 39](#_Toc21185)

[3.5.6 Bảng Payment 39](#_Toc13225)

[3.5.7 Bảng Exercise 40](#_Toc14963)

[3.5.8 Bảng Lesson 40](#_Toc27392)

[3.5.9 Bảng Message 40](#_Toc1608)

[CHƯƠNG 4. GIAO DIỆN CHƯƠNG TRÌNH 41](#_Toc31382)

[I. Giao diện đăng nhập 41](#_Toc9325)

[II. Giao diện đăng ký 41](#_Toc18195)

[III. Giao diện trang chủ 42](#_Toc8533)

[IV. Giao diện chi tiết khoá học 42](#_Toc9182)

[V. Giao diện bài tập 43](#_Toc8324)

[VI. Giao diện nhắn tin 44](#_Toc13637)

[KẾT LUẬN 45](#_Toc20633)

* 2. Giới thiệu tổng quát

Ứng dụng "Hệ thống quản lý khóa học trực tuyến" được phát triển nhằm cung cấp một nền tảng học tập hiệu quả và tiện lợi cho người dùng, bao gồm học viên, giáo viên và quản trị viên. Hệ thống được xây dựng trên nền tảng Web với kiến trúc hướng dịch vụ, tận dụng RESTful API để cung cấp khả năng tương tác và quản lý dữ liệu khóa học một cách mạnh mẽ và linh hoạt.

Các chức năng chính của hệ thống bao gồm quản lý khóa học, quản lý học viên, quản lý giáo viên và các hình thức thanh toán trực tuyến. Người dùng có thể dễ dàng tìm kiếm, đăng ký khóa học, tham gia vào các lớp học, cũng như theo dõi tiến độ học tập. Giáo viên có thể tạo và quản lý nội dung khóa học, theo dõi quá trình học tập của học viên, và cung cấp hỗ trợ khi cần thiết. Hệ thống cũng hỗ trợ quản trị viên trong việc giám sát toàn bộ hoạt động của nền tảng.

Về mặt công nghệ, giao diện người dùng được phát triển với các framework hiện đại như React.js hoặc Angular, mang lại trải nghiệm mượt mà và tương tác tốt. Bên cạnh đó, hệ thống có thể dễ dàng mở rộng hoặc tích hợp với các nền tảng khác như di động hoặc ứng dụng trên Windows trong các giai đoạn phát triển tiếp theo, sử dụng các framework đa nền tảng như React Native hoặc Flutter.

Ứng dụng này không chỉ tập trung vào việc cung cấp trải nghiệm học tập trực tuyến chất lượng mà còn hướng đến việc nâng cao khả năng mở rộng, khả năng tương thích đa nền tảng, và đáp ứng nhu cầu của người dùng ở mọi cấp độ.

* 1. Mục đích đề tài

Mục đích của đề tài "Xây dựng hệ thống quản lý khóa học trực tuyến" là tạo ra một nền tảng giúp tối ưu hóa quy trình dạy và học trong môi trường số, đáp ứng nhu cầu ngày càng cao của người học và người dạy trong việc tiếp cận các khóa học trực tuyến. Hệ thống không chỉ hỗ trợ các chức năng cơ bản như quản lý khóa học, học viên và giáo viên mà còn cung cấp các tiện ích hiện đại như thanh toán trực tuyến, theo dõi tiến độ học tập, và tương tác giữa học viên và giáo viên một cách dễ dàng.

Ngoài ra, đề tài còn nhằm mục đích:

**Nâng cao khả năng quản lý học tập**: Hệ thống giúp giáo viên và nhà quản lý dễ dàng theo dõi và quản lý học viên, nội dung khóa học, cũng như các hoạt động liên quan đến quá trình dạy và học.

**Mở rộng khả năng học tập**: Ứng dụng cho phép học viên học tập mọi lúc, mọi nơi thông qua môi trường trực tuyến, tạo điều kiện thuận lợi cho việc học tập từ xa.

**Cải thiện tính linh hoạt và mở rộng**: Được xây dựng theo kiến trúc hướng dịch vụ, hệ thống có khả năng dễ dàng tích hợp thêm các tính năng mới hoặc mở rộng sang các nền tảng khác (di động, Windows,...) trong tương lai, phù hợp với nhu cầu phát triển của các tổ chức giáo dục.

**Đảm bảo tính ổn định và bảo mật**: Hệ thống được phát triển với mục tiêu đảm bảo hiệu suất, tính ổn định, và bảo mật dữ liệu, đặc biệt là trong việc bảo vệ thông tin cá nhân và các giao dịch trực tuyến của học viên.

**Nâng cao trải nghiệm người dùng**: Giao diện người dùng được thiết kế đơn giản, dễ sử dụng nhưng vẫn đảm bảo tính thẩm mỹ và tương tác cao, giúp học viên và giáo viên có trải nghiệm tốt khi sử dụng hệ thống.

Như vậy, mục đích của đề tài không chỉ là cung cấp một giải pháp học tập trực tuyến chất lượng mà còn hướng đến việc tạo ra một hệ thống có khả năng mở rộng, linh hoạt và đáp ứng được các yêu cầu phát triển của thị trường giáo dục số trong tương lai.

* 1. Đối tượng và phạm vi nguyên cứu

Đề tài "Xây dựng hệ thống quản lý khóa học trực tuyến" tập trung nghiên cứu các yếu tố kỹ thuật và chức năng liên quan đến việc thiết kế và phát triển một hệ thống quản lý khóa học trên nền tảng Web. Các đối tượng nghiên cứu chính bao gồm:

* **Người dùng cuối**: Học viên, giáo viên và quản trị viên sử dụng hệ thống để đăng ký, quản lý và giảng dạy các khóa học trực tuyến.
* **Quy trình quản lý khóa học**: Nghiên cứu các mô hình quản lý khóa học, đăng ký học viên, tổ chức và phân phối tài nguyên học tập.
* **Công nghệ phát triển**: Nghiên cứu các framework và công nghệ Web hiện đại như React.js, Angular, .NET Core, Spring cho việc xây dựng giao diện người dùng và cung cấp dịch vụ qua RESTful API.
* **Bảo mật và thanh toán**: Nghiên cứu cách tích hợp các phương thức thanh toán trực tuyến, bảo mật dữ liệu người dùng và an toàn giao dịch.

**Phạm vi nghiên cứu:**

Phạm vi nghiên cứu của đề tài tập trung vào các khía cạnh sau:

**Khảo sát và phân tích**: Nghiên cứu các hệ thống quản lý khóa học trực tuyến hiện có, từ đó phân tích nhu cầu của người dùng để thiết kế hệ thống phù hợp.

**Thiết kế hệ thống**: Xây dựng mô hình kiến trúc hướng dịch vụ (SOA) để đảm bảo khả năng mở rộng và tích hợp sau này, bao gồm cả thiết kế cơ sở dữ liệu và các thành phần chính của hệ thống như quản lý khóa học, người dùng, và các tính năng hỗ trợ giảng dạy trực tuyến.

**Phát triển ứng dụng Web**: Sử dụng các framework phát triển ứng dụng Web để xây dựng giao diện người dùng và backend của hệ thống, cụ thể là với React.js hoặc Angular cho frontend và .NET Core, Spring hoặc Laravel cho backend.

**Tích hợp thanh toán trực tuyến**: Nghiên cứu và triển khai các giải pháp thanh toán trực tuyến an toàn và hiệu quả để học viên có thể thanh toán phí khóa học qua các cổng thanh toán.

**Kiểm thử và triển khai**: Đảm bảo hệ thống hoạt động ổn định và hiệu quả thông qua các kỹ thuật kiểm thử phần mềm, bao gồm kiểm thử chức năng, kiểm thử bảo mật, và kiểm thử hiệu năng.

**Mở rộng nền tảng**: Đối với Đồ án Công nghệ phần mềm 2, phạm vi sẽ mở rộng nghiên cứu sang phát triển hệ thống trên các nền tảng khác như di động (React Native, Flutter) hoặc desktop (Windows).

Như vậy, đối tượng và phạm vi nghiên cứu tập trung vào việc xây dựng và hoàn thiện một hệ thống quản lý khóa học trực tuyến từ bước phân tích nhu cầu, thiết kế, phát triển cho đến kiểm thử và triển khai, đồng thời hướng đến khả năng mở rộng sang các nền tảng khác trong tương lai.

* 1. Phương pháp nguyên cứu

Để hoàn thành đề tài "Xây dựng hệ thống quản lý khóa học trực tuyến", các phương pháp nghiên cứu chính được sử dụng bao gồm:

**Phương pháp nghiên cứu tài liệu**:

* 1. Tìm hiểu và tham khảo các tài liệu, sách vở, và các nghiên cứu trước đây về hệ thống quản lý khóa học trực tuyến.
  2. Nghiên cứu các công nghệ, framework phát triển ứng dụng Web như React.js, Angular, .NET Core, Spring, cũng như các phương pháp phát triển phần mềm hướng dịch vụ (SOA).
  3. Tham khảo tài liệu về bảo mật trong các hệ thống trực tuyến, đặc biệt là bảo mật thông tin người dùng và các giao dịch tài chính.

**Phương pháp khảo sát và phân tích**:

* 1. Khảo sát và đánh giá các hệ thống quản lý khóa học trực tuyến hiện có, cả về chức năng lẫn trải nghiệm người dùng, từ đó xác định điểm mạnh và yếu.
  2. Phân tích yêu cầu người dùng để hiểu rõ các nhu cầu thực tế của học viên, giáo viên và quản trị viên trong quá trình sử dụng hệ thống.
  3. Dựa trên kết quả khảo sát, phân tích các vấn đề hiện có và đề xuất giải pháp cải tiến, nâng cao tính tiện dụng và hiệu quả của hệ thống.

**Phương pháp phân tích và thiết kế hệ thống**:

* 1. Sử dụng phương pháp phân tích thiết kế hệ thống theo mô hình hướng dịch vụ (Service-Oriented Architecture - SOA) để đảm bảo tính linh hoạt và khả năng mở rộng của hệ thống.
  2. Thiết kế mô hình cơ sở dữ liệu để quản lý hiệu quả thông tin về khóa học, học viên, giáo viên, và các thành phần khác liên quan.
  3. Thiết kế giao diện người dùng (UI/UX) trực quan, thân thiện, và dễ sử dụng nhằm đảm bảo trải nghiệm tốt nhất cho người dùng.

**Phương pháp phát triển phần mềm**:

* 1. Áp dụng mô hình phát triển phần mềm linh hoạt (Agile) để quản lý tiến độ và chất lượng trong suốt quá trình xây dựng hệ thống.
  2. Xây dựng hệ thống theo từng giai đoạn nhỏ, mỗi giai đoạn hoàn thiện một phần chức năng cụ thể để dễ dàng kiểm thử và tinh chỉnh.
  3. Sử dụng các công cụ và framework hiện đại như React.js/Angular cho frontend, và .NET Core/Spring cho backend, cùng với RESTful API để phát triển các dịch vụ.

**Phương pháp kiểm thử**:

* 1. Áp dụng kiểm thử chức năng (functional testing) để đảm bảo hệ thống đáp ứng đầy đủ các yêu cầu đề ra.
  2. Kiểm thử bảo mật (security testing) nhằm bảo vệ dữ liệu người dùng và các giao dịch tài chính khỏi các nguy cơ tiềm ẩn.
  3. Kiểm thử hiệu năng (performance testing) để đảm bảo hệ thống hoạt động ổn định khi có số lượng lớn người dùng truy cập đồng thời.

**Phương pháp đánh giá và cải tiến**:

* 1. Dựa trên kết quả kiểm thử, tiến hành đánh giá hệ thống về hiệu suất, tính khả dụng và bảo mật, từ đó đề xuất các cải tiến cần thiết.
  2. Thu thập phản hồi từ người dùng thử nghiệm để điều chỉnh và nâng cao chất lượng của hệ thống trước khi triển khai chính thức.

Nhờ sự kết hợp giữa các phương pháp nghiên cứu tài liệu, khảo sát, thiết kế, phát triển và kiểm thử, đề tài hướng đến việc xây dựng một hệ thống quản lý khóa học trực tuyến hoàn chỉnh, đảm bảo tính linh hoạt, bảo mật, và đáp ứng được nhu cầu của người dùng.

* 1. Nội dụng thực hành đề tài
     1. Phân tích yêu cầu

· Tiến hành khảo sát các hệ thống quản lý khóa học trực tuyến hiện có nhằm thu thập các yêu cầu từ người dùng.

· Phân tích các yêu cầu từ phía học viên, giáo viên và quản trị viên về chức năng quản lý khóa học, quản lý học viên, giáo viên, lịch học và thanh toán.

· Xác định yêu cầu phi chức năng, bao gồm yêu cầu về hiệu năng, bảo mật, và khả năng mở rộng của hệ thống.

· Phân tích yêu cầu giao tiếp giữa các thành phần, trong đó Web API sẽ được sử dụng để cung cấp các dịch vụ và giao diện người dùng được xây dựng bằng các công nghệ hiện đại.

* + 1. Thiết kế hệ thống

· **Thiết kế kiến trúc hệ thống**: Hệ thống sẽ được thiết kế theo kiến trúc hướng dịch vụ (SOA). .NET Core sẽ được sử dụng để xây dựng các dịch vụ Web API, cung cấp dữ liệu cho giao diện người dùng. Frontend có thể được phát triển bằng các công nghệ như React.js hoặc Angular.

· **Thiết kế cơ sở dữ liệu**: Sử dụng SQL Server để lưu trữ thông tin về khóa học, học viên, giáo viên, giao dịch thanh toán và các thông tin liên quan khác. Các bảng dữ liệu sẽ được tối ưu hóa để đảm bảo truy vấn dữ liệu nhanh chóng và hiệu quả.

· **Thiết kế API**: Sử dụng .NET Core để xây dựng các API theo chuẩn RESTful, đảm bảo tính nhất quán và khả năng mở rộng. Các API sẽ bao gồm chức năng như quản lý khóa học, quản lý người dùng, thanh toán và xử lý dữ liệu.

* + 1. Phát triển ứng dụng

· **Phát triển Web API**: Sử dụng .NET Core để phát triển các dịch vụ API hỗ trợ cho các chức năng quản lý khóa học, người dùng, và thanh toán. Sử dụng các mô hình và bộ điều khiển (Controllers) để quản lý luồng dữ liệu và logic nghiệp vụ.

· **Phát triển giao diện người dùng**: Sử dụng React.js hoặc Angular để xây dựng giao diện người dùng. Giao diện sẽ thân thiện với người dùng, đáp ứng đầy đủ yêu cầu về tính năng và hiệu năng.

· **Tích hợp API**: Tương tác giữa frontend và backend được thực hiện thông qua các API. Các yêu cầu từ phía người dùng sẽ được gửi qua các API và nhận kết quả phản hồi từ hệ thống backend.

* + 1. Tối ưu và cải tiến

· **Tối ưu hệ thống**: Sau khi phát triển các chức năng cơ bản, tiến hành tối ưu hóa các phần liên quan đến hiệu suất hệ thống, bao gồm cả việc cải tiến thời gian phản hồi của API và tốc độ tải trang.

· **Cải tiến trải nghiệm người dùng**: Dựa trên phản hồi của người dùng thử nghiệm, cải tiến giao diện để mang lại trải nghiệm tốt hơn. Chú trọng đến khả năng tương tác mượt mà giữa người dùng và hệ thống.

* + 1. Hướng dẫn sử dụng và triển khai

· **Hướng dẫn sử dụng**: Viết tài liệu hướng dẫn chi tiết cách sử dụng hệ thống cho các đối tượng học viên, giáo viên, và quản trị viên. Tài liệu sẽ bao gồm các bước cơ bản để quản lý khóa học, đăng ký học viên và thực hiện thanh toán.

· **Triển khai hệ thống**: Cung cấp hướng dẫn chi tiết về quy trình triển khai hệ thống lên môi trường sản xuất, bao gồm các bước cấu hình máy chủ, cài đặt phần mềm cần thiết, và thiết lập cơ sở dữ liệu.

* 1. Môi trường ứng dụng
     1. Môi trường phát triển

· **Công cụ phát triển backend**: Sử dụng Visual Studio 2022 với .NET Core SDK để phát triển Web API.

· **Công cụ phát triển frontend**: Sử dụng Visual Studio Code để phát triển giao diện người dùng với các công nghệ ASP .NET MVC.

· **Cơ sở dữ liệu**: Sử dụng SQL Server để lưu trữ dữ liệu.

* + 1. Môi trường triển khai

· **Máy chủ ứng dụng**: Triển khai hệ thống trên máy chủ có hỗ trợ Windows hoặc Linux với IIS hoặc Nginx.

· **Cơ sở dữ liệu**: Sử dụng SQL Server hoặc các hệ thống cơ sở dữ liệu khác tương thích để lưu trữ dữ liệu.

· **Công cụ CI/CD**: Sử dụng các công cụ như GitHub Actions để triển khai và duy trì hệ thống tự động.

* + 1. Môi trường phát triển phần mền

· **Frontend**: Phát triển giao diện người dùng với ASP .NET MVC , sử dụng CSS/Boostrap để tạo giao diện thân thiện và dễ dùng.

· **Backend**: Phát triển các dịch vụ API bằng .NET Core, triển khai các tính năng quản lý dữ liệu và logic nghiệp vụ.

· **Công cụ kiểm thử**: Sử dụng Postman để kiểm tra các API và các công cụ như Selenium để kiểm tra tự động hóa cho giao diện người dùng.

CHƯƠNG 2: Cơ Sở Lý Thuyết

1. Cơ sở lý thuyết
   1. Ngôn ngữ lập trình C#

· **Giới thiệu**: C# là một ngôn ngữ lập trình hiện đại, hướng đối tượng, được Microsoft phát triển trong môi trường .NET. C# kết hợp sức mạnh của lập trình C/C++ với tính đơn giản và hiệu quả của các ngôn ngữ hiện đại như Java.

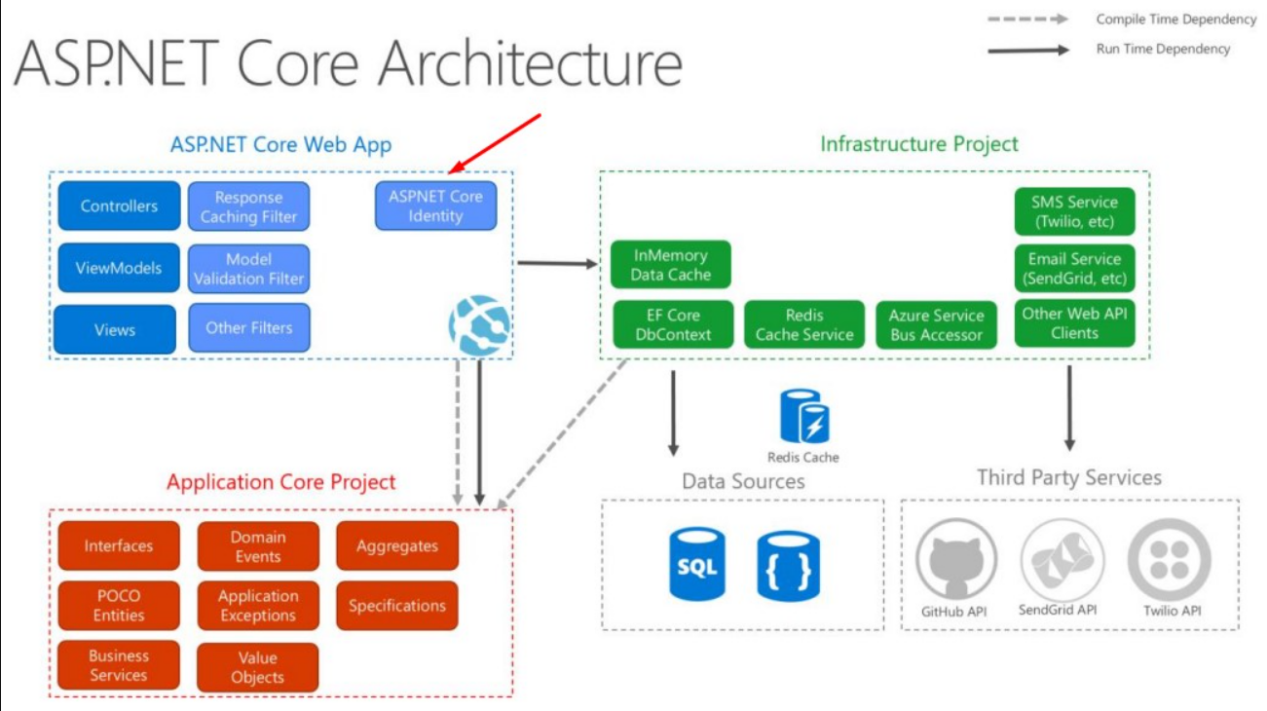
· **Ưu điểm**:

* Hỗ trợ mạnh mẽ lập trình hướng đối tượng (OOP).
* Khả năng tích hợp tốt với các công nghệ của Microsoft như .NET Core, ASP.NET Core.
* Hỗ trợ LINQ giúp lập trình viên dễ dàng làm việc với các tập hợp dữ liệu.
* Tính bảo mật và hiệu suất cao, phù hợp cho các ứng dụng web lớn.
  1. Giới thiệu về ASP.NET Core MVC
     1. ASP.NET Core MVC là gì?

· **ASP.NET Core MVC** là một framework để xây dựng các ứng dụng web động, cho phép lập trình viên phát triển ứng dụng dựa trên mô hình MVC (Model-View-Controller).

· Mô hình **MVC** tách biệt các thành phần trong ứng dụng web:

* **Model**: Quản lý dữ liệu và logic nghiệp vụ.
* **View**: Quản lý giao diện người dùng, hiển thị dữ liệu.
* **Controller**: Điều khiển luồng dữ liệu giữa Model và View.



* + 1. Ưu điểm của ASP.NET Core MVC

· **Hiệu suất cao**: ASP.NET Core được tối ưu hóa cho hiệu năng, giúp giảm thời gian phản hồi của các ứng dụng web.

· **Cross-platform**: Hỗ trợ phát triển trên nhiều hệ điều hành như Windows, macOS và Linux.

· **Kiến trúc module hóa**: Dễ dàng mở rộng và bảo trì.

· **Tích hợp mạnh mẽ với các công nghệ bảo mật**: Hỗ trợ tích hợp dễ dàng với các dịch vụ bảo mật như OAuth2, JWT và các chuẩn bảo mật khác.

* + 1. Các đặc tính cơ bản của ASP.NET Core MVC

· **Routing**: Hệ thống định tuyến mạnh mẽ, giúp lập trình viên dễ dàng quản lý các URL của ứng dụng.

· **Dependency Injection (DI)**: Hỗ trợ tiêm phụ thuộc mặc định, giúp quản lý các dịch vụ và phụ thuộc trong ứng dụng một cách hiệu quả.

· **Middleware**: Cơ chế xử lý các yêu cầu HTTP thông qua chuỗi middleware linh hoạt.

· **View Engine**: Sử dụng Razor view engine để tạo các giao diện động, giúp tách biệt logic xử lý và giao diện người dùng.

* 1. Giới thiệu về ASP.NET Core Identity và Bảo mật
     1. ASP.NET Core Identity là gì?

**ASP.NET Core Identity** là một hệ thống quản lý danh tính, cung cấp các chức năng như xác thực và phân quyền người dùng trong các ứng dụng web. Nó được tích hợp sâu trong ASP.NET Core và hỗ trợ dễ dàng quản lý tài khoản, mật khẩu, đăng nhập xã hội, và xác thực đa yếu tố (2FA).



* + 1. Cơ chế hoạt động của ASP.NET Core Identity

ASP.NET Core Identity hoạt động dựa trên các **cookie** để lưu trữ thông tin đăng nhập và **token** để xác thực và phân quyền người dùng. Hệ thống cho phép lập trình viên cấu hình các phương thức xác thực khác nhau như **OAuth2**, **JWT**, và **Cookie-based Authentication**.

* + 1. Các thành phần cơ bản của ASP.NET Core Identity

· **Authentication**: Xác thực người dùng, kiểm tra tính hợp lệ của thông tin đăng nhập.

· **Authorization**: Phân quyền người dùng dựa trên vai trò và quyền hạn, kiểm soát truy cập vào các tài nguyên.

· **User Management**: Quản lý người dùng bao gồm đăng ký, đăng nhập, thay đổi mật khẩu, và quản lý vai trò.

* + 1. Ưu và nhược điểm của ASP.NET Core Identity
       1. Ưu điểm của ASP.NET Core Identity

· **Tích hợp sẵn**: ASP.NET Core Identity được tích hợp sẵn trong ASP.NET Core, giúp lập trình viên dễ dàng triển khai các chức năng bảo mật.

· **Bảo mật cao**: Hỗ trợ các chuẩn bảo mật hiện đại như **OAuth2**, **JWT**, và **SAML**.

· **Mở rộng**: Dễ dàng mở rộng để tích hợp với các dịch vụ xác thực bên ngoài như Google, Facebook, và Microsoft.

* + - 1. Nhược điểm của ASP.NET Core Identity

· **Độ phức tạp**: Mặc dù tích hợp sẵn nhiều tính năng, việc cấu hình và mở rộng ASP.NET Core Identity có thể phức tạp đối với những người mới bắt đầu.

· **Tài liệu phong phú nhưng phân tán**: Tài liệu chính thức và các hướng dẫn sử dụng đôi khi thiếu sự nhất quán, làm cho việc triển khai có thể gặp khó khăn.

* 1. Hệ quản trị cơ sở dữ liệu SQL Server
     1. SQL Server là gì?

**SQL Server** là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó hỗ trợ lưu trữ và quản lý dữ liệu với khả năng bảo mật cao và hiệu năng tốt, thường được sử dụng trong các ứng dụng doanh nghiệp và hệ thống lớn.

* + 1. Các ưu điểm của công cụ SQL Server

· **Hiệu suất cao**: SQL Server được tối ưu hóa cho các tác vụ xử lý dữ liệu phức tạp với khả năng quản lý tài nguyên tốt.

· **Tích hợp mạnh mẽ với các sản phẩm của Microsoft**: Tích hợp chặt chẽ với hệ sinh thái Microsoft, như Visual Studio, .NET Core, và các dịch vụ đám mây Azure.

· **Bảo mật cao**: Hỗ trợ các cơ chế bảo mật tiên tiến như mã hóa dữ liệu, quản lý truy cập, và auditing.

· **Khả năng mở rộng**: SQL Server hỗ trợ quản lý cơ sở dữ liệu từ các hệ thống nhỏ đến các hệ thống lớn với khả năng mở rộng tốt.

**CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ HỆ THỐNG**

1. Mô tả bài toán

Bạn đang thiết kế một hệ thống quản lý khóa học trực tuyến sử dụng **ASP.NET Core MVC** và **SQL Server**. Dữ liệu trong hệ thống được tổ chức theo các bảng quan hệ để quản lý thông tin người dùng, khóa học, bài học, bài kiểm tra, đánh giá, và thanh toán. Hệ thống này sẽ thực hiện các chức năng quản lý đăng ký khóa học, thanh toán, cũng như lưu trữ thông tin và theo dõi tiến trình học tập của người dùng.

* 1. Sơ đồ use case
  2. Đặc tả use case
     1. Use case đăng nhập

|  |  |
| --- | --- |
| Mã Use Case | US\_01 |
| Tên Use Case | Đăng Nhập |
| Tác Nhân | Admin, User |
| Mô tả ngắn | Người dùng đăng nhập vào hệ thống để sử dụng các chức năng có trong hệ thống. |
| Tiền điều kiện | Người dùng đã khởi động ứng dụng |
| Kết quả | Quyền Admin sẽ chuyển tới trang quản trị viên Quyền User sẽ chuyển đến trang chủ |
| Điều kiện kích hoạt | Nhập đầy đủ các thông tin |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng nhấn vào chức năng "Đăng Nhập":  - Người dùng nhập tên và mật khẩu.  - Sau khi nhập xong, người dùng nhấn nút Đăng Nhập. Hệ thống thông báo "Đăng nhập thành công." và hiển thị các chức năng được phép dùng. |
| Luồng sự kiện phụ | Nếu người dùng nhập sai hoặc để trống tài khoản hoặc mật khẩu thì hệ thống sẽ xuất thông báo "Thông tin tài khoản hoặc mật khẩu không chính xác. Vui lòng nhập lại". Người dùng nhập lại tài khoản và mật khẩu hoặc nhấn nút Thoát để kết thúc. |

* + 1. Use case Đăng ký

|  |  |
| --- | --- |
| Mã Use Case | US\_02 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

* + 1. Use case quên mật khẩu

|  |  |
| --- | --- |
| Mã Use Case | US\_03 |
| Tên Use Case | Quên mật khẩu |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng yêu cầu đặt lại mật khẩu |
| Tiền điều kiện | Người dùng chọn chức năng Quên mật khẩu từ màn hình đăng nhập. |
| Kết quả | Hệ thống gửi email đặt lại mật khẩu. |
| Điều kiện kích hoạt | Nhập email đã đăng ký |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn "Quên mật khẩu":  - Nhập địa chỉ email đã đăng ký.  - Nhấn Gửi yêu cầu. Hệ thống kiểm tra và gửi email hướng dẫn. |
| Luồng sự kiện phụ | Nếu email không hợp lệ, hệ thống hiển thị thông báo lỗi. |

* + 1. Use case thay đổi mật khẩu

|  |  |
| --- | --- |
| Mã Use Case | US\_04 |
| Tên Use Case | Thay đổi mật khẩu |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng thay đổi mật khẩu cá nhân |
| Tiền điều kiện | Người dùng đã đăng nhập thành công. |
| Kết quả | Mật khẩu được thay đổi và cập nhật trong hệ thống. |
| Điều kiện kích hoạt | Nhập đúng thông tin tài khoản và mật khẩu hiện tại. |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Thay đổi mật khẩu":  - Nhập mật khẩu cũ và mật khẩu mới.  - Nhấn nút Xác nhận. Hệ thống kiểm tra và cập nhật mật khẩu mới. |
| Luồng sự kiện phụ | Nếu mật khẩu cũ không đúng hoặc mật khẩu mới không hợp lệ, hệ thống sẽ thông báo lỗi. |

* + 1. Use case tìm kiếm kháo học

|  |  |
| --- | --- |
| Mã Use Case | US\_05 |
| Tên Use Case | Tìm kiếm khoá học |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tìm kiếm các khóa học. |
| Tiền điều kiện | Người dùng đã truy cập trang tìm kiếm khóa học. |
| Kết quả | Danh sách khóa học phù hợp được hiển thị. |
| Điều kiện kích hoạt | Nhập từ khóa hoặc bộ lọc tìm kiếm. |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng nhập từ khóa vào thanh tìm kiếm và nhấn nút Tìm kiếm:  - Hệ thống hiển thị danh sách khóa học theo từ khóa |
| Luồng sự kiện phụ | Nếu không tìm thấy khóa học nào, hệ thống hiển thị thông báo "Không tìm thấy kết quả phù hợp". |

* + 1. Use case xem danh sách khoá học

|  |  |
| --- | --- |
| Mã Use Case | US\_06 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

* + 1. Use case tham gia khoá học

|  |  |
| --- | --- |
| Mã Use Case | US\_07 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

* + 1. Use case tạo khoá học

|  |  |
| --- | --- |
| Mã Use Case | US\_08 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

* + 1. Use case sửa thông tin khoá học

|  |  |
| --- | --- |
| Mã Use Case | US\_09 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

* + 1. Use case xoá khoá học

|  |  |
| --- | --- |
| Mã Use Case | US\_010 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

* + 1. Use case xem bài tập

|  |  |
| --- | --- |
| Mã Use Case | US\_011 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

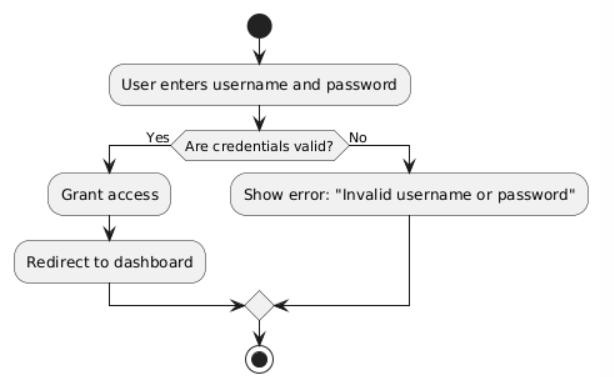
* + 1. Use case thêm bài tập

|  |  |
| --- | --- |
| Mã Use Case | US\_012 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

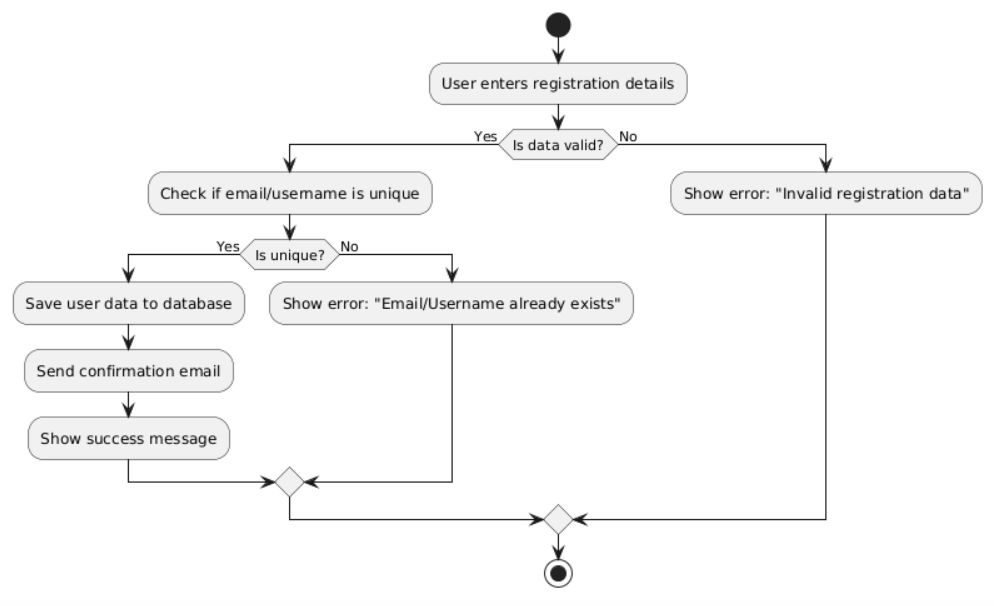
* + 1. Use case xoá bài tập

|  |  |
| --- | --- |
| Mã Use Case | US\_013 |
| Tên Use Case | Đăng ký |
| Tác Nhân | User |
| Mô tả ngắn | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| Tiền điều kiện | Người dùng khởi động ứng dụng và chọn chức năng Đăng ký |
| Kết quả | Người dùng có tài khoản mới trong hệ thống và có thể đăng nhập. |
| Điều kiện kích hoạt | Nhập các thông tin đăng ký bắt buộc |
| Luồng sự kiện chính | Use case bắt đầu khi người dùng chọn chức năng "Đăng ký":  - Nhập thông tin tài khoản (tên, email, mật khẩu,...).  - Nhấn nút Đăng ký. Hệ thống kiểm tra thông tin và tạo tài khoản. |
| Luồng sự kiện phụ | Nếu thông tin bị thiếu hoặc không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. |

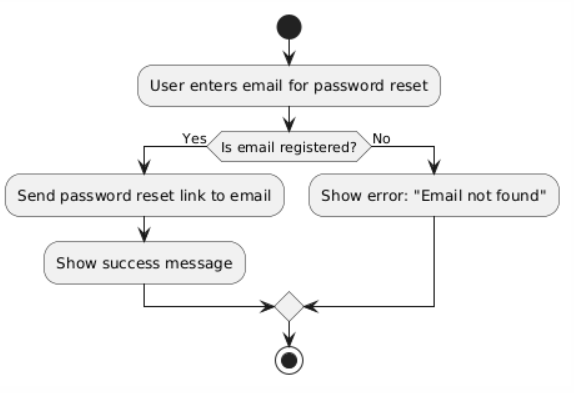
* 1. Sơ đồ hoạt động
     1. Đăng nhập



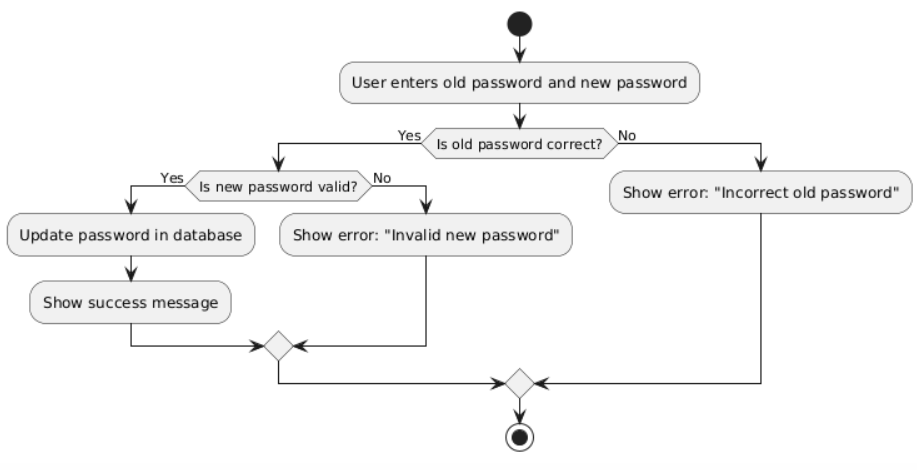
* + 1. Đăng ký



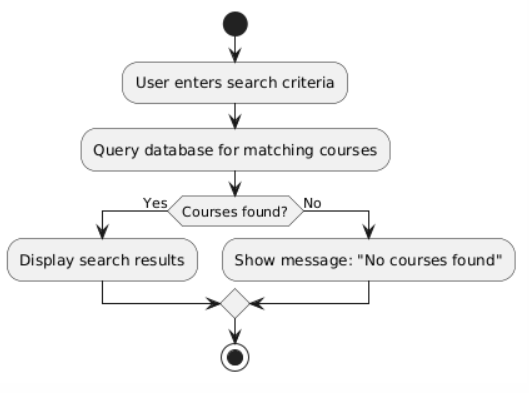
* + 1. Quên mật khẩu



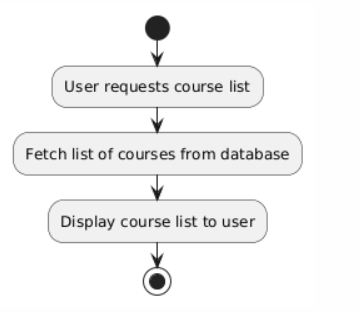
* + 1. Thay đổi mật khẩu



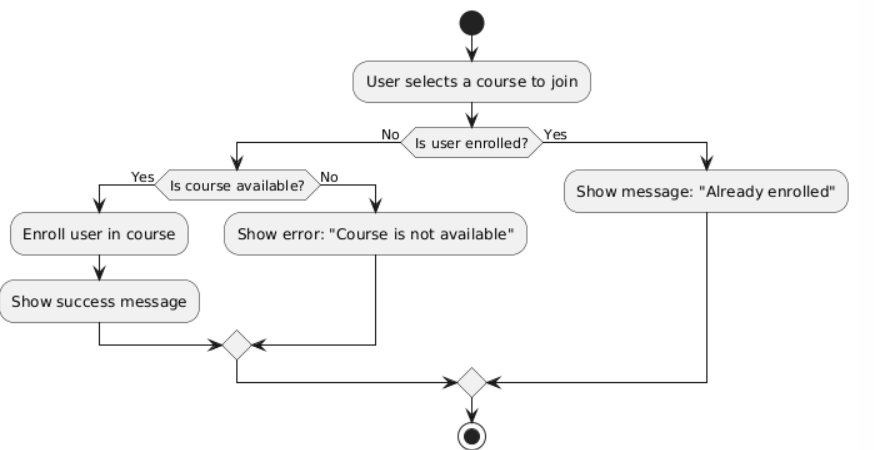
* + 1. Tìm kiếm khoá học



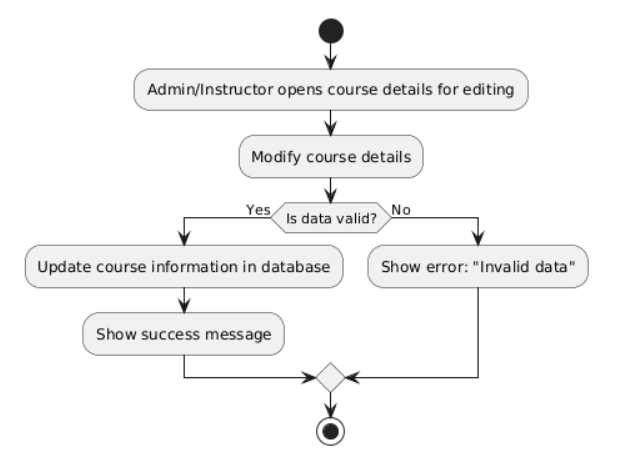
* + 1. Xem danh sách khoá học



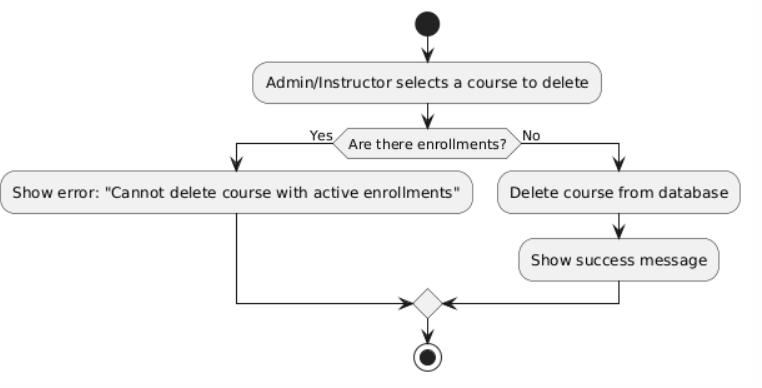
* + 1. Tham gia khoá học



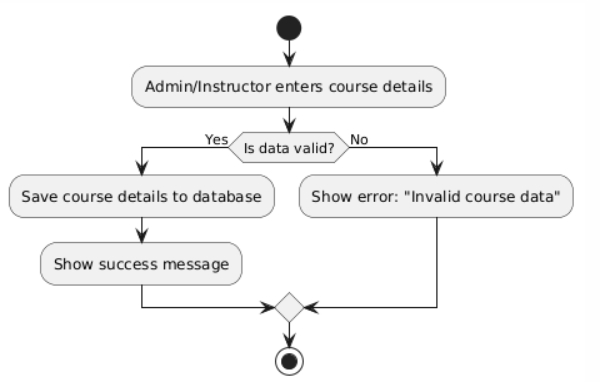
* + 1. Sửa thông tin khoá học



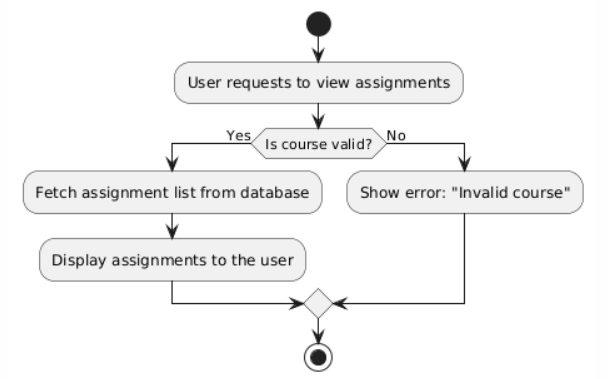
* + 1. Xoá khoá học



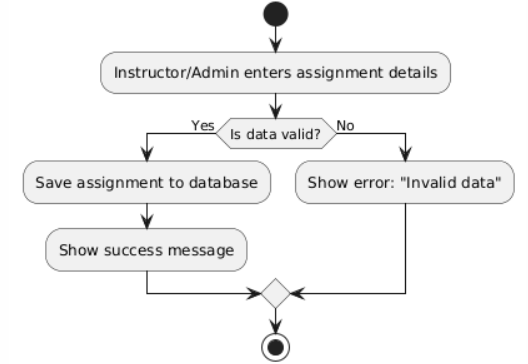
* + 1. Thêm khoá học



* + 1. Xem bài tập



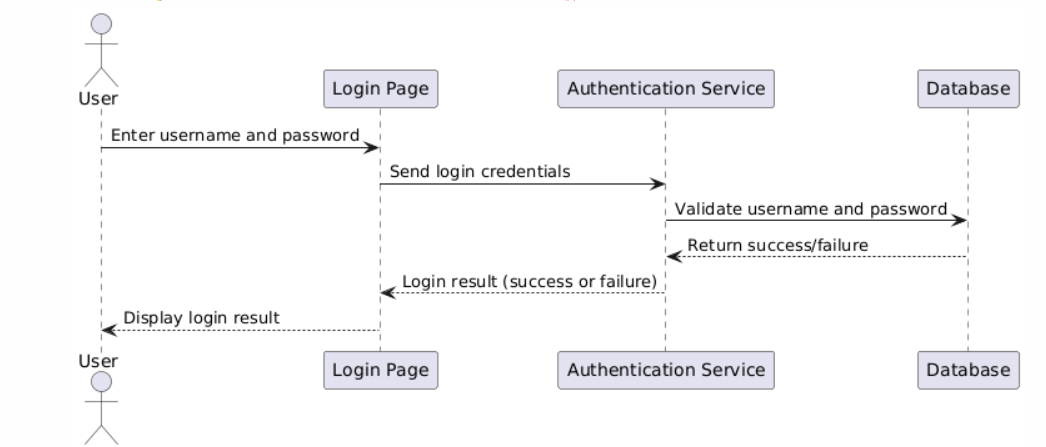
* + 1. Thêm bài tập



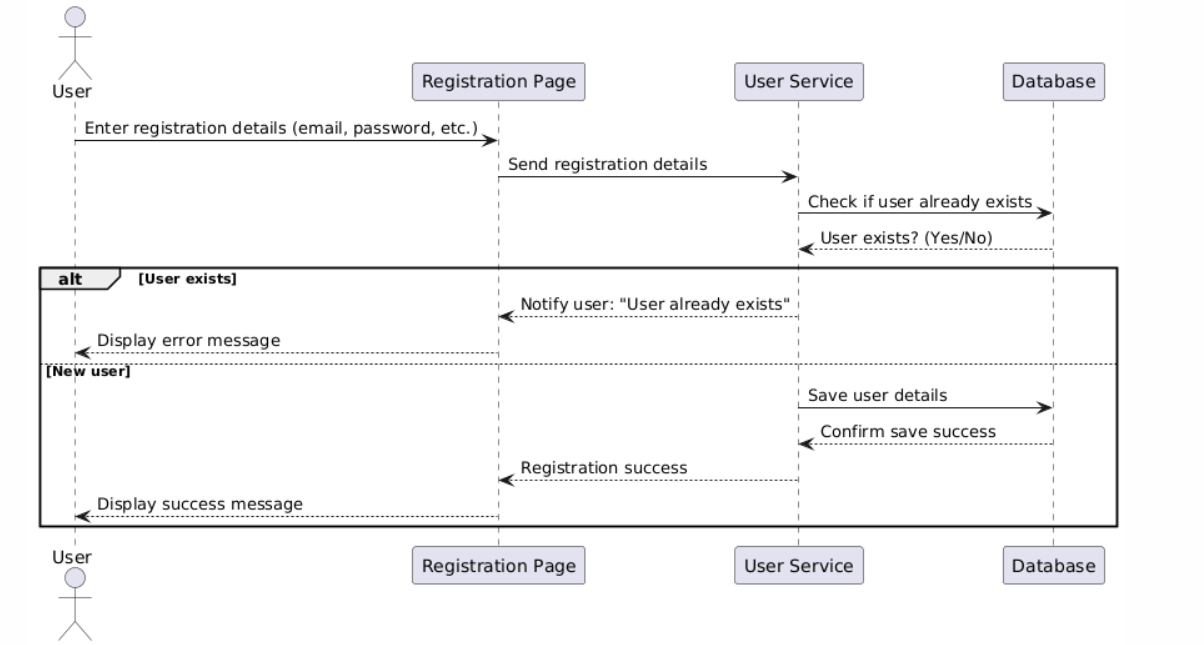
* + 1. Xoá bài tập



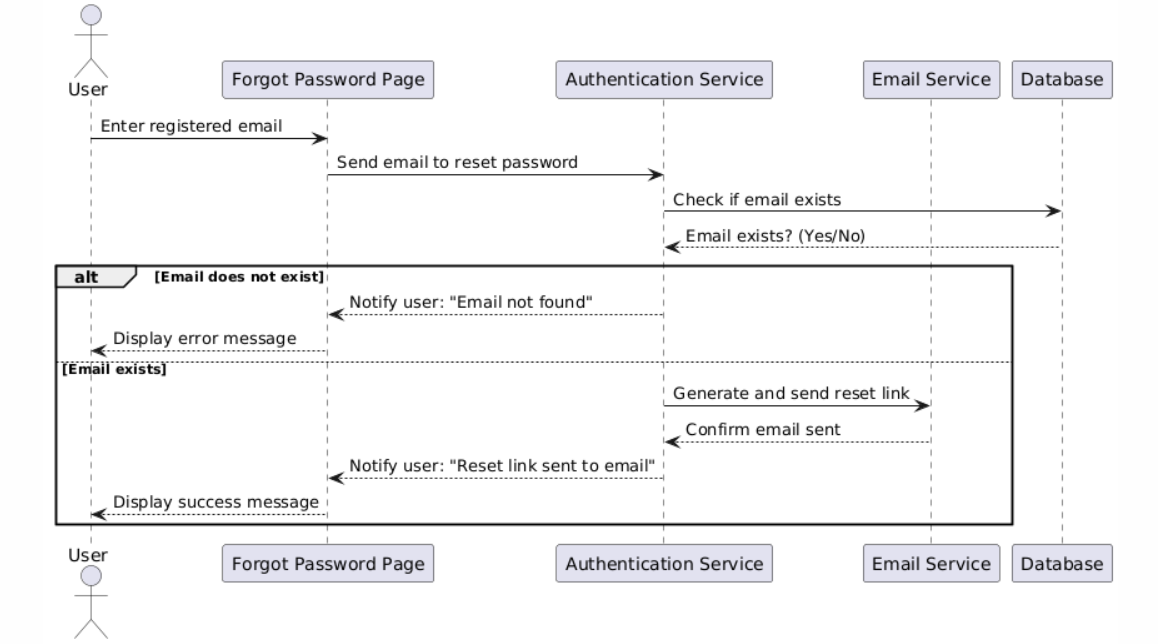
* 1. Sơ đồ tuần tự
     1. Đăng nhập



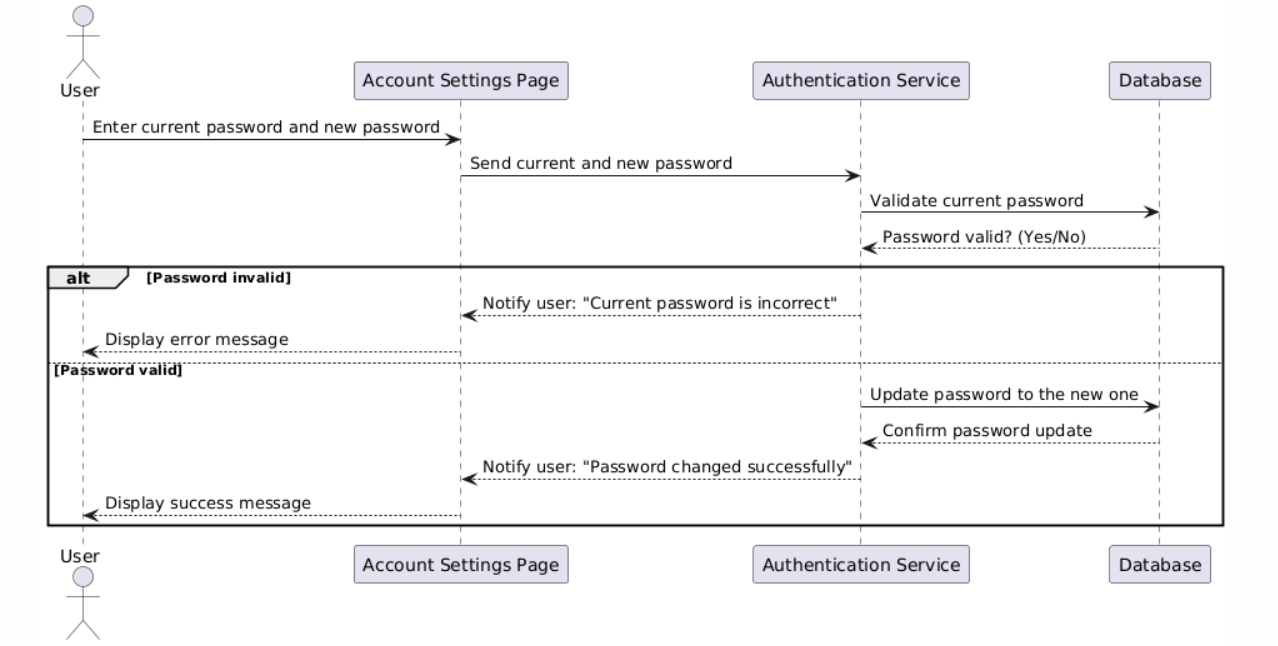
* + 1. Đăng ký



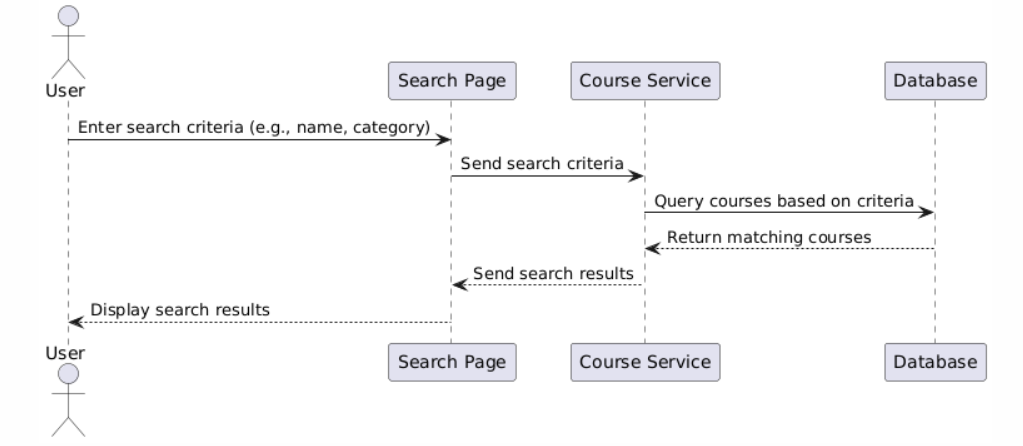
* + 1. Quên mật khẩu



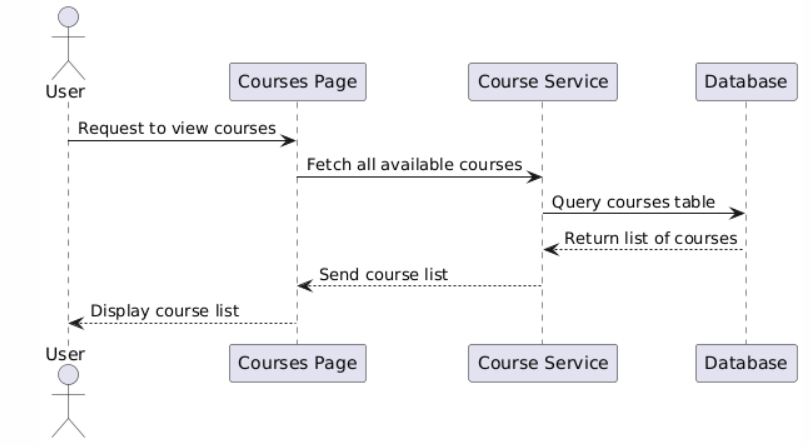
* + 1. Thay đổi mật khẩu



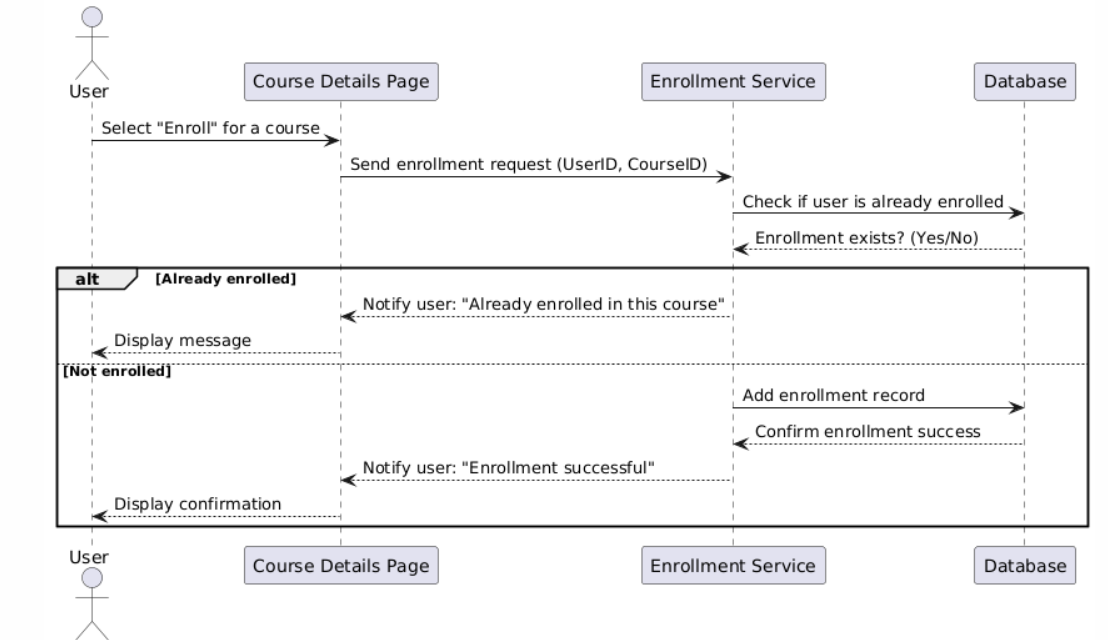
* + 1. Tìm kiếm khoá học



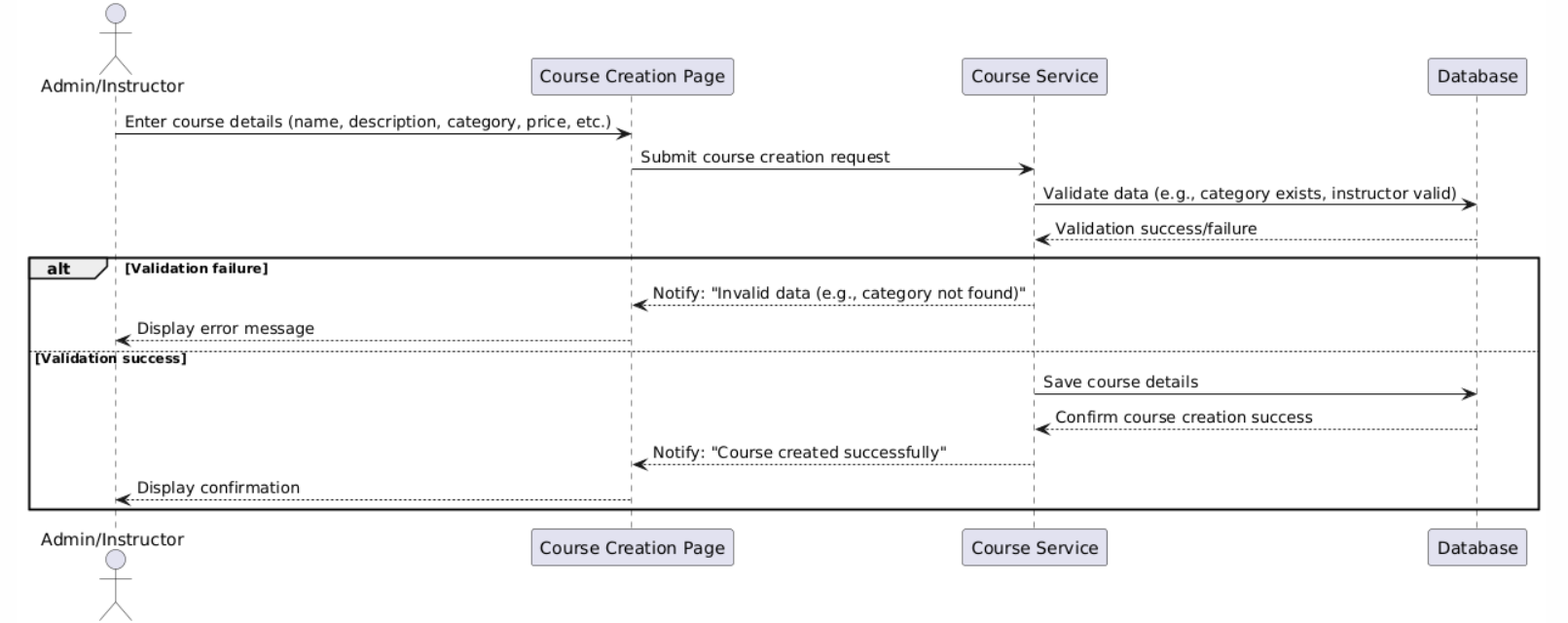
* + 1. Xem danh sách khoá học



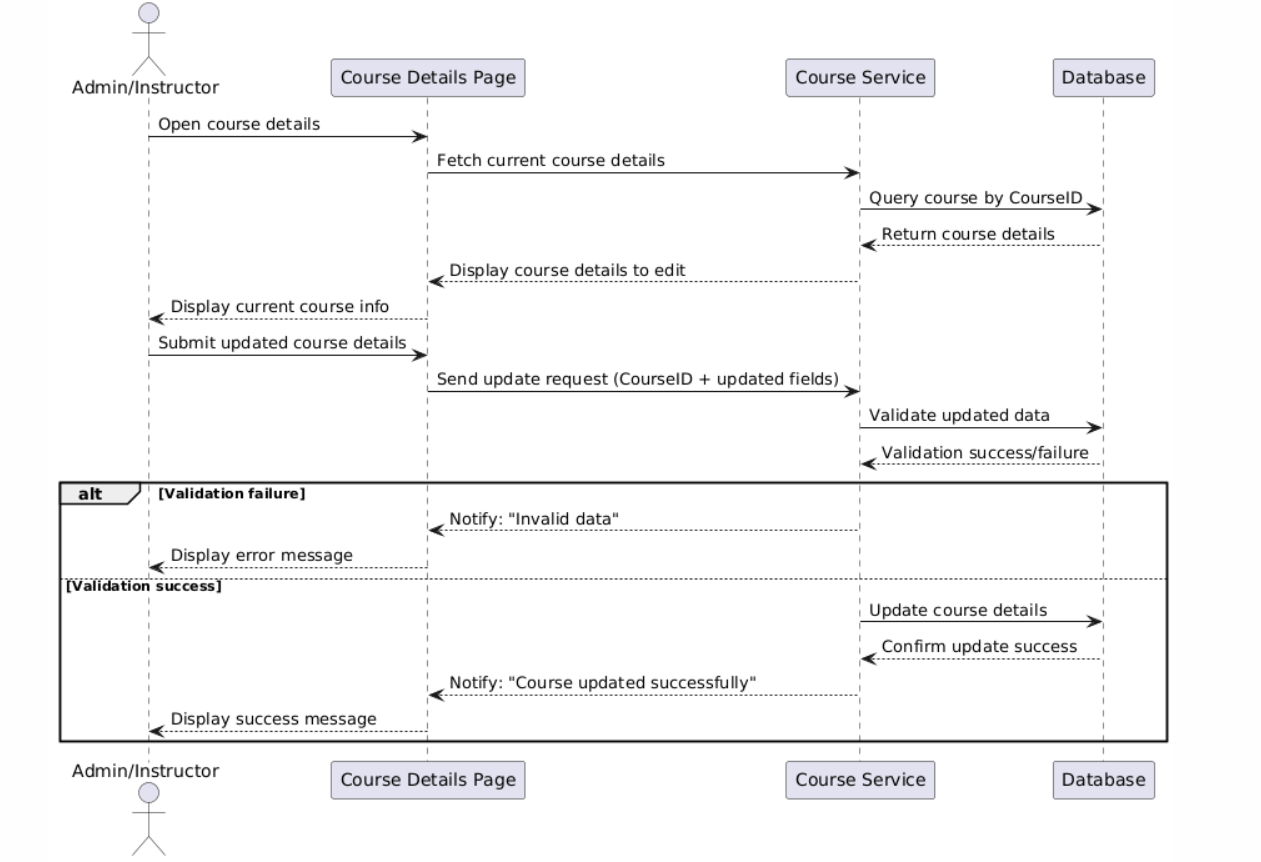
* + 1. Tham gia khoá học



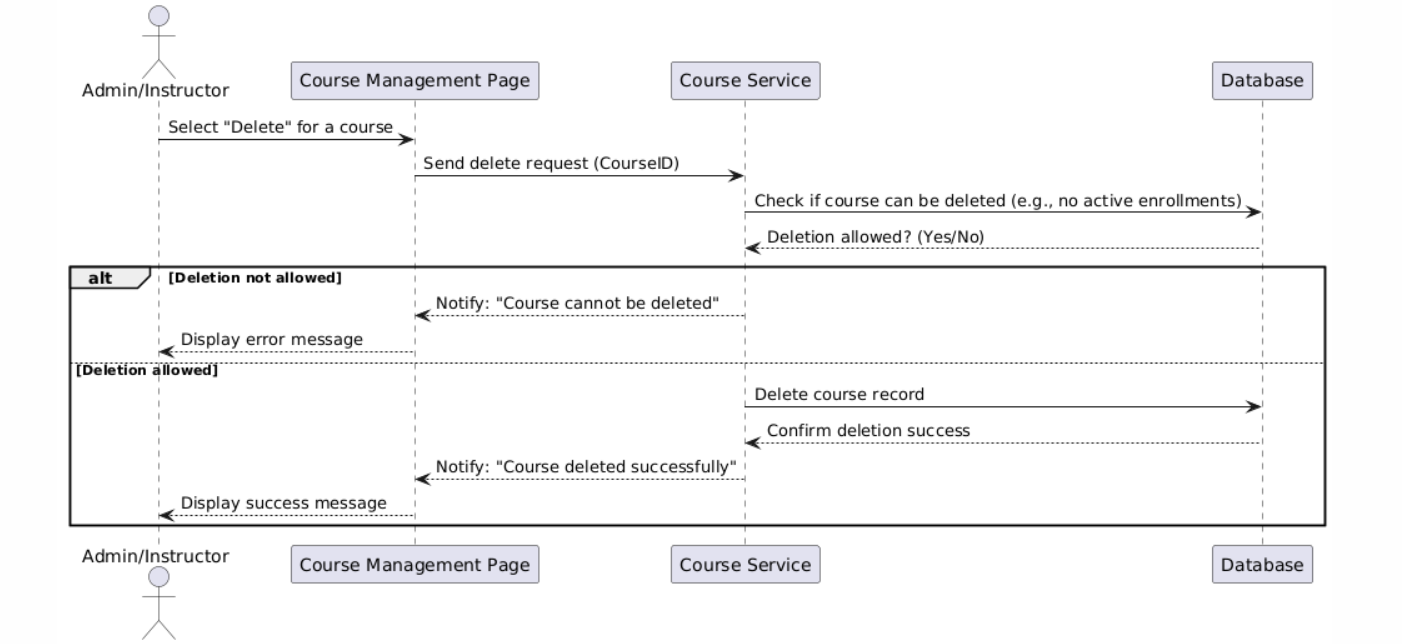
* + 1. Tạo khoá học



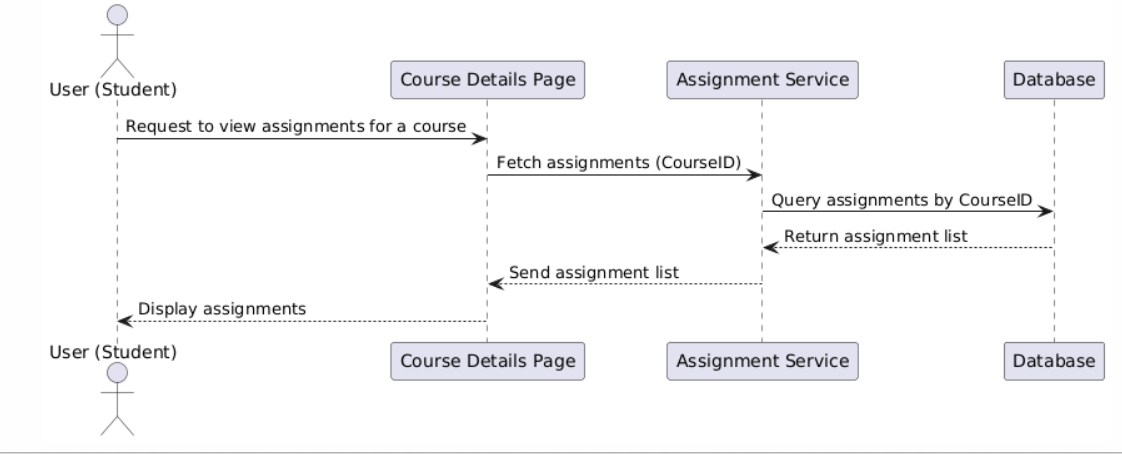
* + 1. Sửa thông tin khoá học



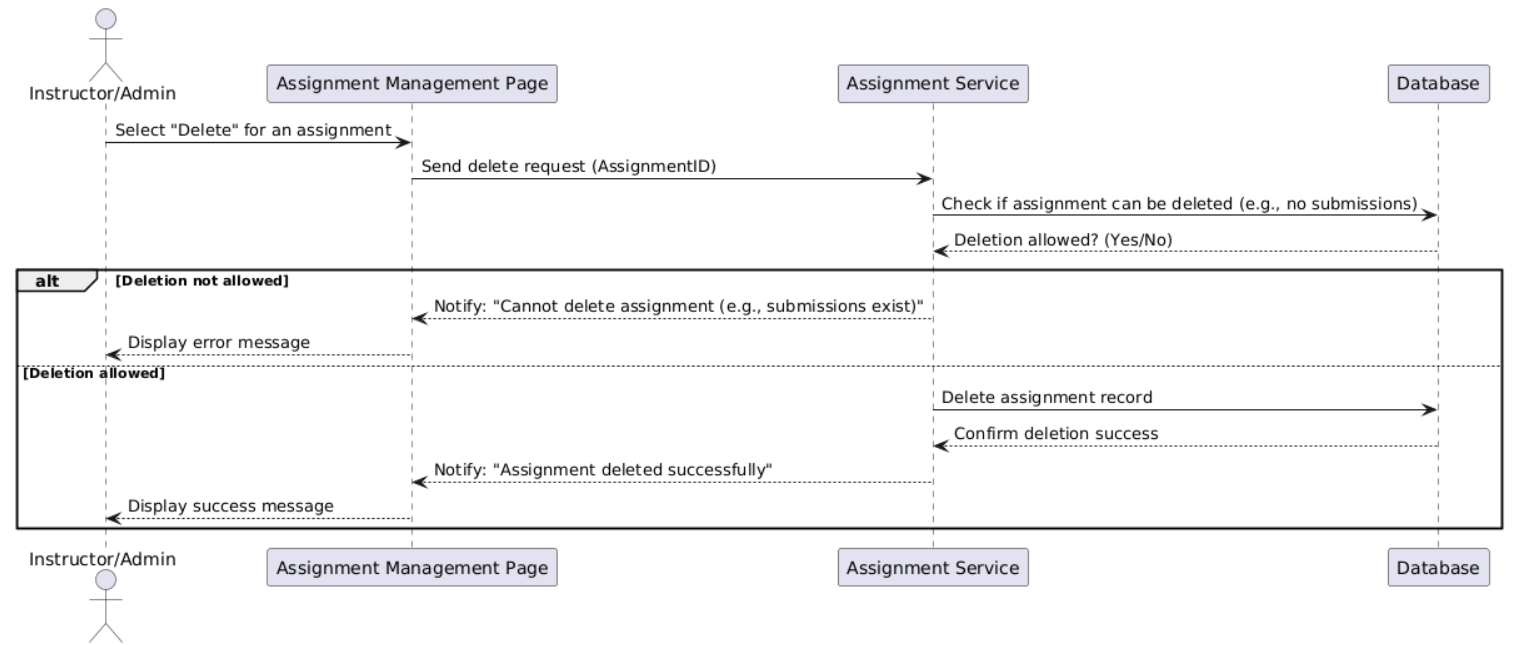
* + 1. Xoá khoá học



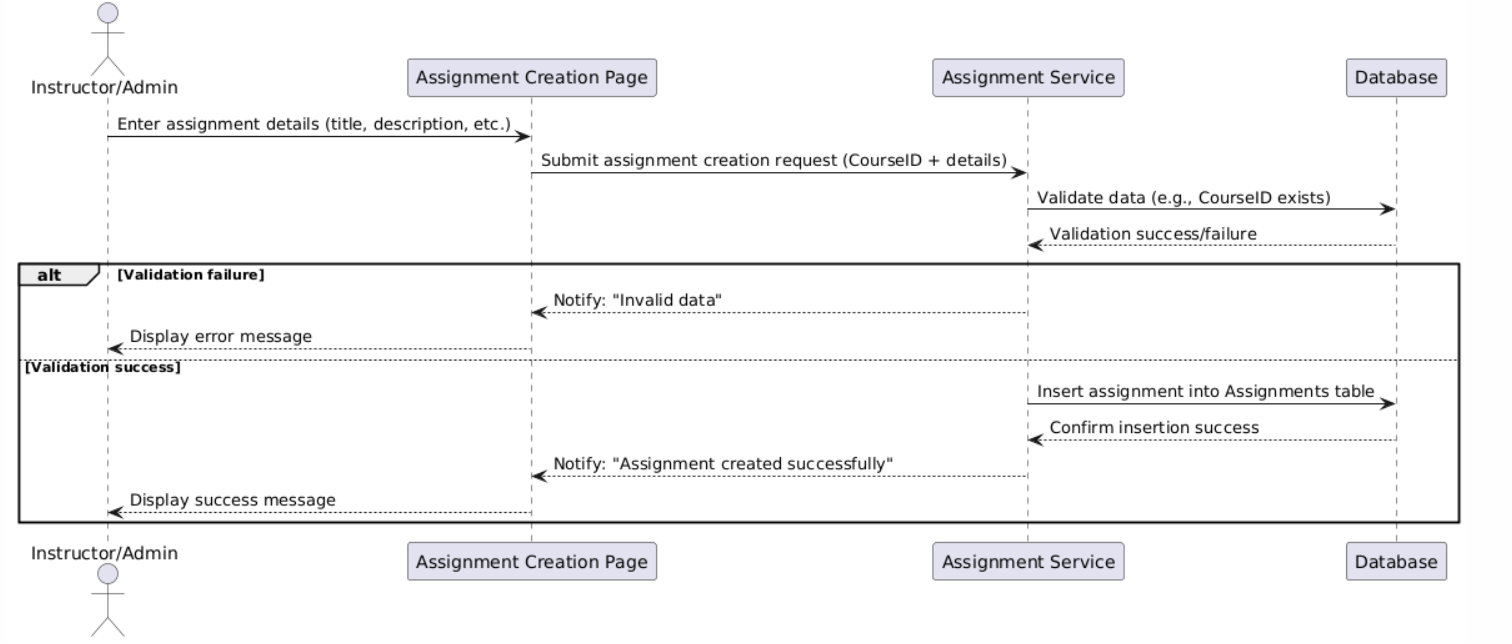
* + 1. Xem bài tập



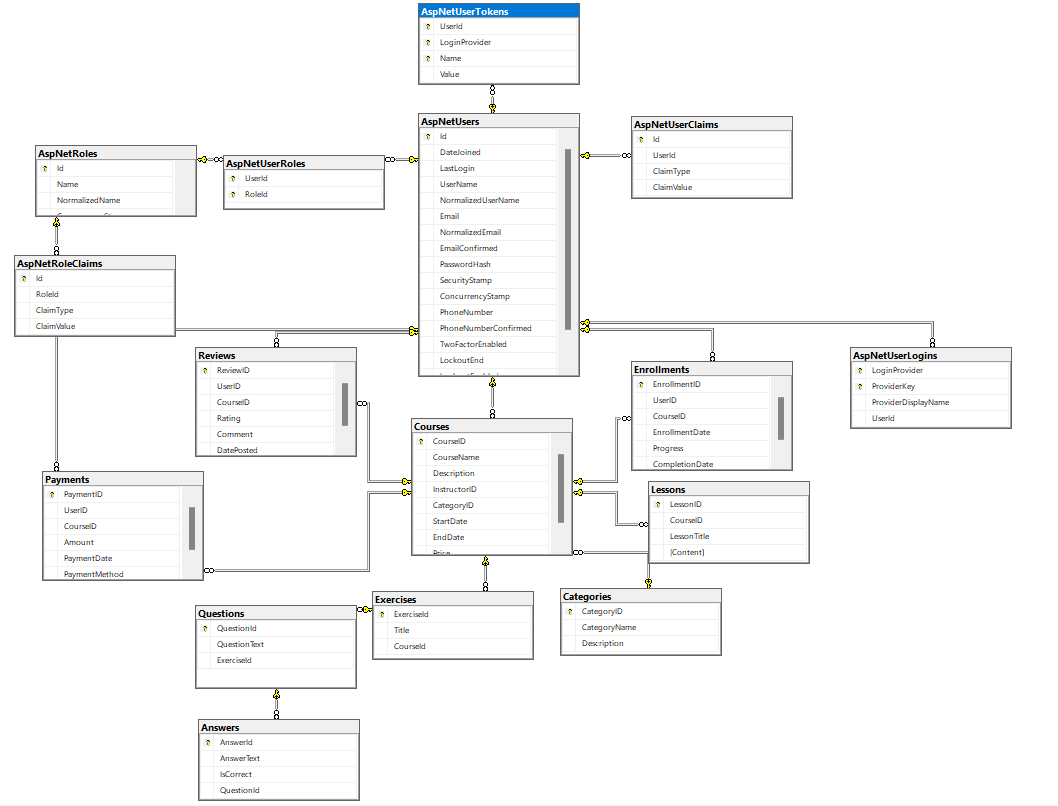
* + 1. Xoá bài tâp



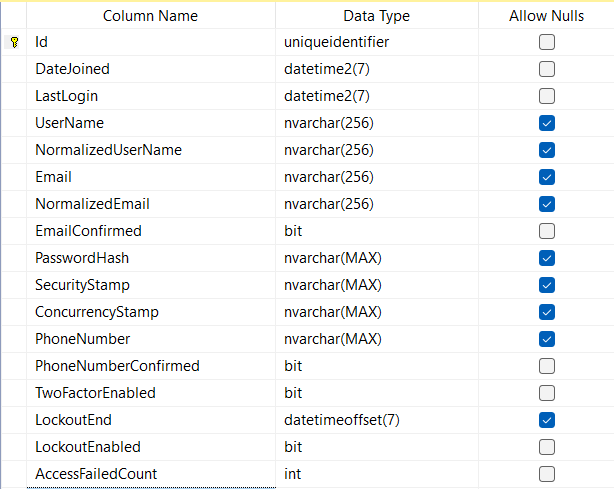
* + 1. Thêm bài tập



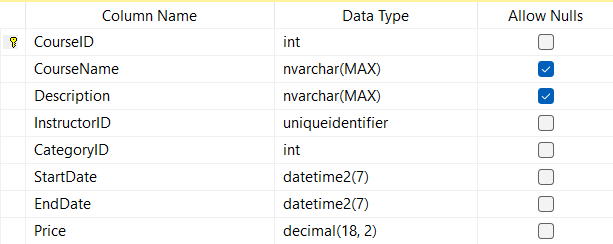
* 1. Thiết kế cơ sở dữ liệu
     1. Cơ sở dữ liệu tổng quát



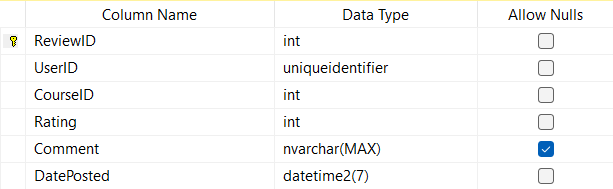
* + 1. Bảng User



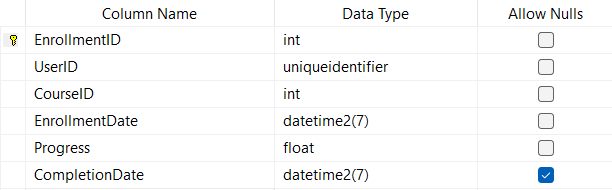
* Bảng AspNetUsers là trung tâm để quản lý và theo dõi thông tin người dùng trong hệ thống. Bảng này lưu trữ các thông tin cá nhân quan trọng như tên, email, số điện thoại và mật khẩu đã được mã hóa, đảm bảo bảo mật dữ liệu người dùng. Ngoài ra, bảng cũng quản lý các vai trò (roles) của người dùng nhằm phân quyền truy cập và kiểm soát các chức năng mà họ được phép thực hiện. Bảng AspNetUsers cũng ghi nhận thời gian người dùng tham gia hệ thống (DateJoined) và lần đăng nhập gần nhất (LastLogin), giúp theo dõi lịch sử hoạt động của họ. Đây là bảng nền tảng kết nối với nhiều bảng khác như Enrollments, Payments, và Messages.
  + 1. Bảng Course



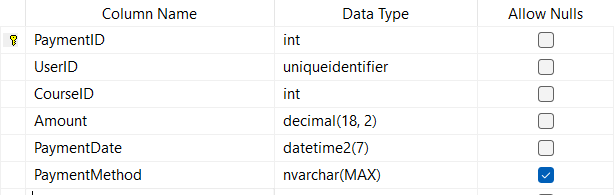
* Bảng Courses lưu trữ thông tin chi tiết về các khóa học trong hệ thống. Mỗi khóa học có tiêu đề, mô tả, và thuộc về một danh mục (category). Ngoài ra, thông tin về giáo viên (teacher) phụ trách khóa học cũng được liên kết từ bảng người dùng. Bảng này hỗ trợ quản lý hiệu quả các khóa học và cung cấp dữ liệu để xây dựng các chức năng liên quan như đánh giá khóa học, thanh toán, và quản lý bài học. Đây là một thành phần quan trọng để tổ chức nội dung học tập.
  + 1. Bảng Reviews



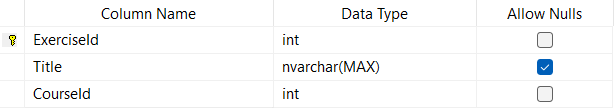
* Bảng Reviews chịu trách nhiệm lưu trữ các đánh giá của người dùng về khóa học. Người dùng có thể cung cấp nhận xét và chấm điểm (rating) cho các khóa học mà họ đã tham gia. Điều này giúp cải thiện chất lượng khóa học thông qua phản hồi từ người học. Bảng này kết nối chặt chẽ với bảng AspNetUsers để xác định người đánh giá và bảng Courses để xác định khóa học được đánh giá.
  + 1. Bảng Enrollment



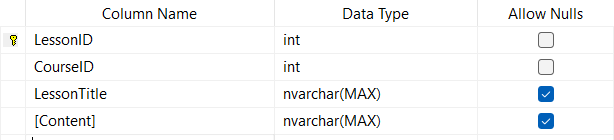
* Bảng Enrollments theo dõi thông tin về việc đăng ký khóa học của người dùng. Nó ghi nhận người dùng nào đã đăng ký khóa học nào và vào thời điểm nào. Thông tin này rất quan trọng trong việc quản lý học viên, kiểm tra trạng thái đăng ký và xác định các khóa học mà một người dùng đã tham gia. Đây là cầu nối giữa bảng AspNetUsers và Courses.
  + 1. Bảng Payment



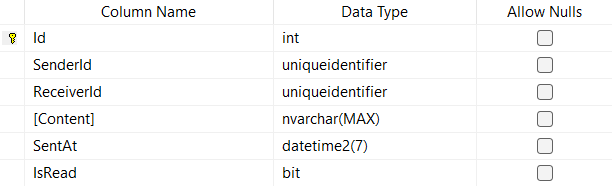
* Bảng Payments lưu trữ thông tin về các giao dịch thanh toán trong hệ thống. Nó bao gồm các chi tiết như số tiền đã thanh toán, người thực hiện giao dịch và khóa học mà giao dịch liên quan. Bảng này hỗ trợ quản lý tài chính, cung cấp dữ liệu cho các báo cáo về doanh thu và đảm bảo minh bạch trong việc xử lý các khoản thanh toán.
  + 1. Bảng Exercise



* Bảng Exercises quản lý thông tin về các bài tập trong hệ thống. Mỗi bài tập có tiêu đề và liên kết với các câu hỏi cụ thể (questions). Bảng này phục vụ cho mục đích học tập và đánh giá, giúp người học thực hành và cải thiện kiến thức thông qua các bài tập được cung cấp.
  + 1. Bảng Lesson



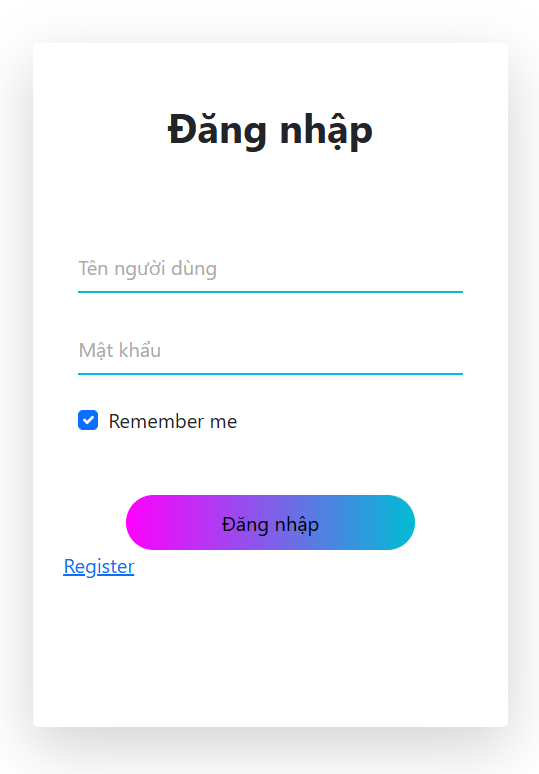
* Bảng Lessons lưu trữ thông tin về các bài học thuộc từng khóa học. Nó bao gồm tiêu đề bài học và liên kết với khóa học tương ứng. Đây là thành phần quan trọng để tổ chức và triển khai nội dung giảng dạy trong hệ thống, giúp người học dễ dàng tiếp cận tài liệu học tập.
  + 1. Bảng Message



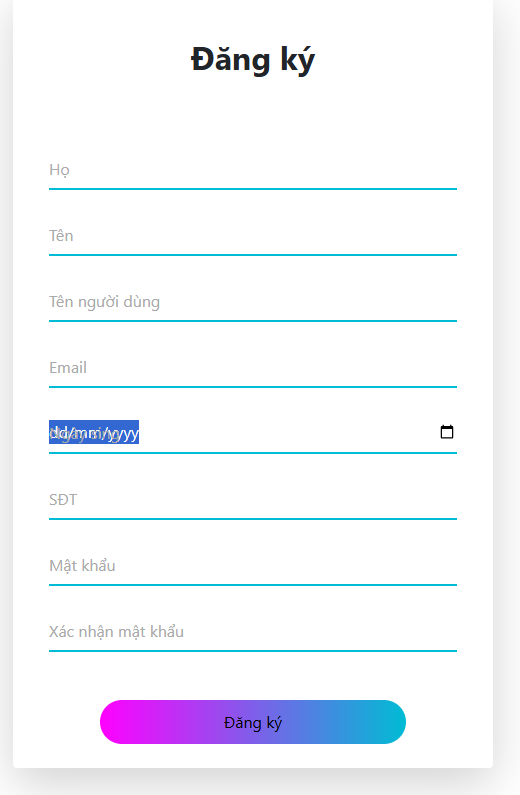
* Bảng Message chịu trách nhiệm lưu trữ thông tin về các tin nhắn được trao đổi giữa người dùng trong hệ thống. Mỗi tin nhắn bao gồm thông tin người gửi, người nhận, nội dung và thời gian gửi. Bảng này hỗ trợ việc giao tiếp hiệu quả giữa các người dùng, đồng thời theo dõi trạng thái tin nhắn (đã đọc hay chưa).

**CHƯƠNG 4. GIAO DIỆN CHƯƠNG TRÌNH**

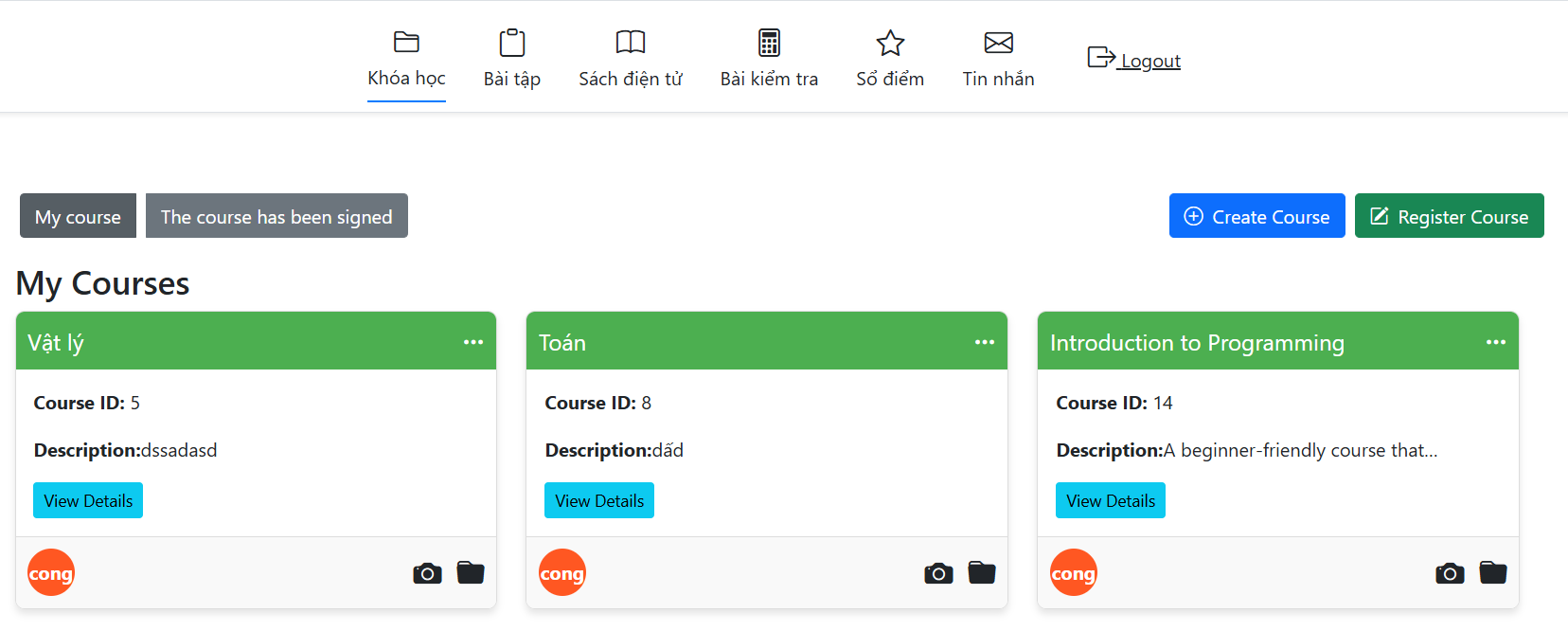
1. Giao diện đăng nhập



1. Giao diện đăng ký



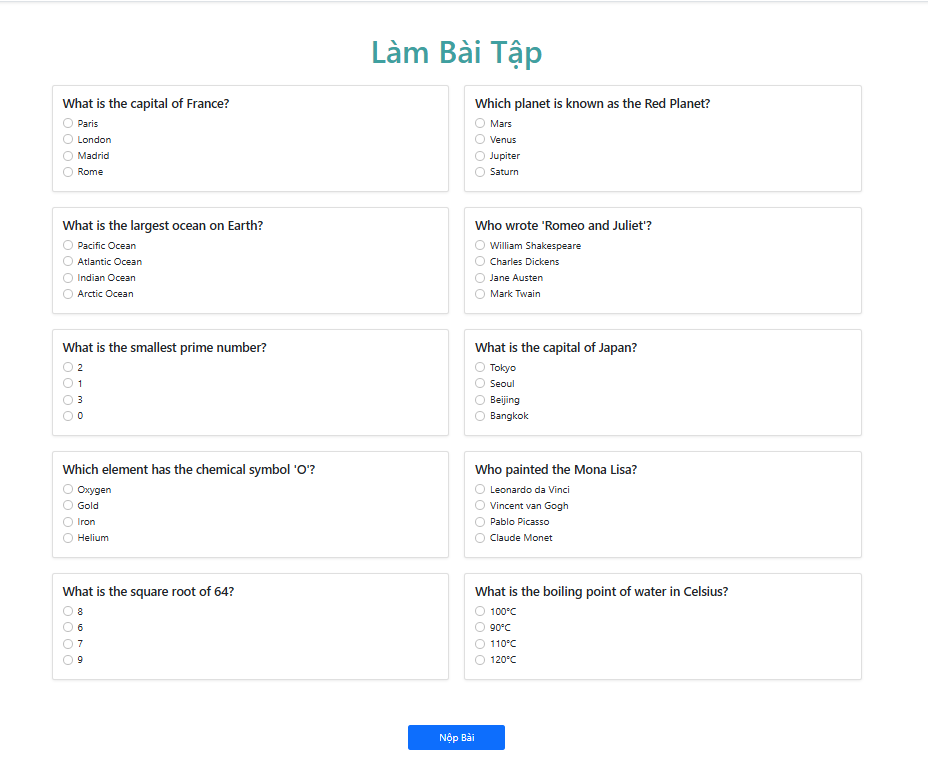
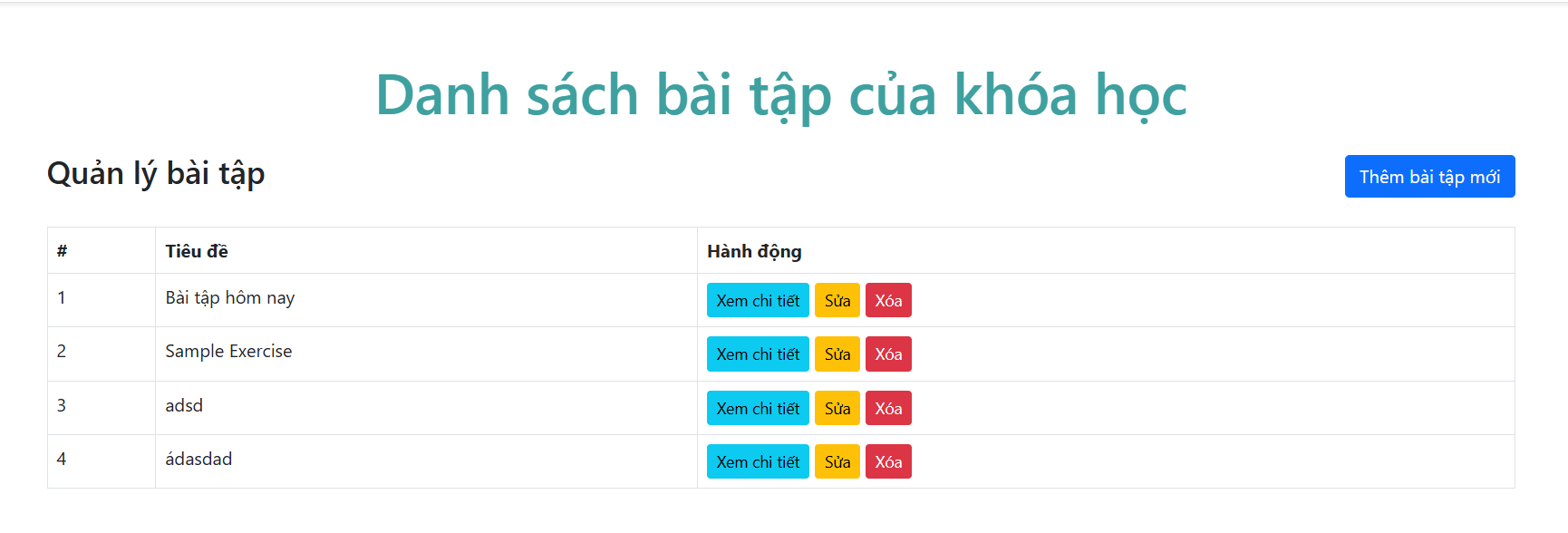
1. Giao diện trang chủ



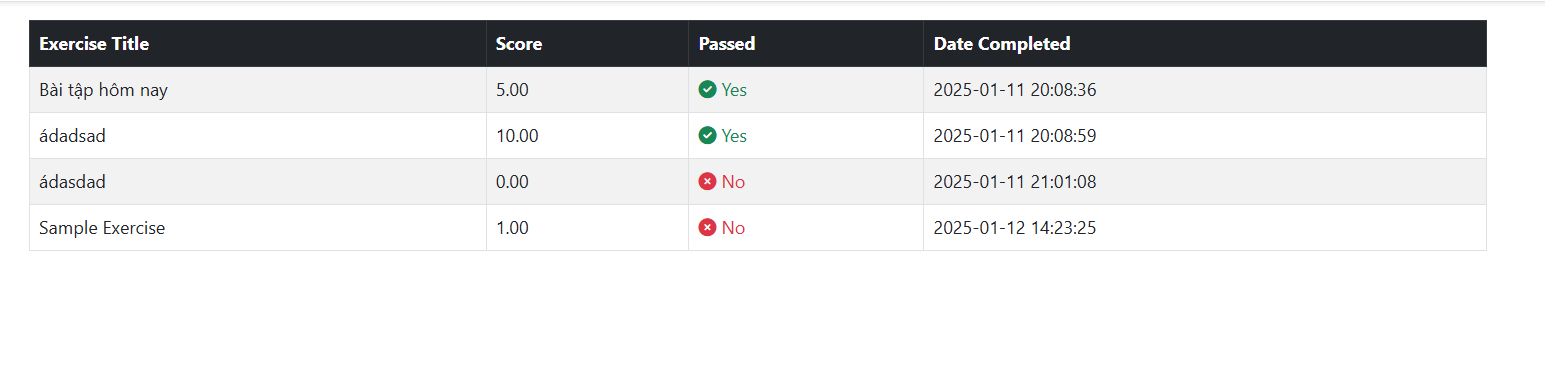
1. Giao diện chi tiết khoá học



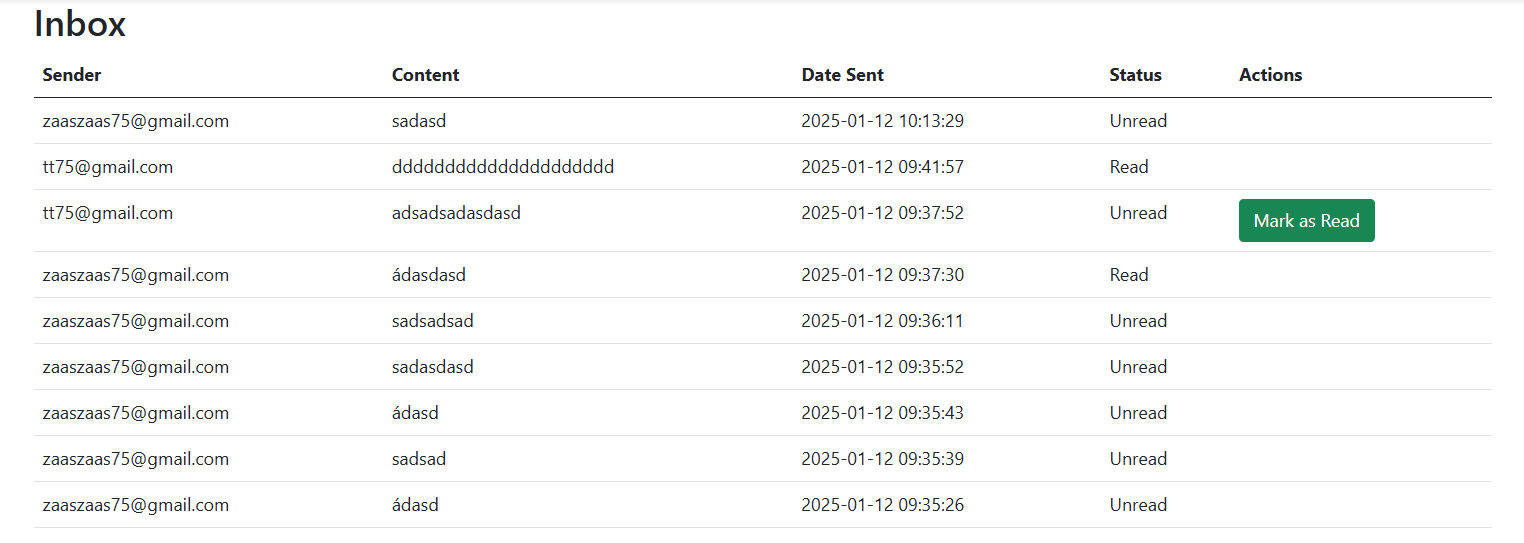
1. Giao diện bài tập







1. Giao diện nhắn tin



**KẾT LUẬN**

Hệ thống quản lý khóa học trực tuyến được xây dựng trên nền tảng .NET đã mang lại nhiều lợi ích quan trọng trong việc hỗ trợ quản lý và tổ chức giáo dục. Cụ thể, các kết quả đạt được có thể kể đến như:

* **Tăng hiệu quả quản lý:** Hệ thống giúp rút ngắn thời gian và công sức trong việc quản lý thông tin khóa học, học viên, giảng viên, và các hoạt động liên quan.
* **Trải nghiệm người dùng tốt hơn:** Giao diện thân thiện và dễ sử dụng giúp giảng viên, học viên, và quản trị viên dễ dàng thao tác và tra cứu thông tin.
* **Tối ưu hóa quy trình:** Các chức năng như đăng ký khóa học, đánh giá, thanh toán, và gửi tin nhắn giúp tự động hóa và tối ưu hóa quy trình quản lý, giảm thiểu lỗi và tăng tính minh bạch.
* **Khả năng mở rộng:** Hệ thống được thiết kế theo hướng module hóa, cho phép mở rộng thêm các tính năng mới trong tương lai một cách linh hoạt.

### ****Hạn chế****

Tuy nhiên, do giới hạn về thời gian và phạm vi phát triển, hệ thống vẫn còn một số hạn chế cần khắc phục:

* **Giao diện chưa tùy chỉnh linh hoạt:** Giao diện chưa thể đáp ứng hết các nhu cầu đa dạng của từng đối tượng người dùng.
* **Chức năng thống kê và báo cáo:** Các báo cáo hiện tại vẫn còn hạn chế, chưa đáp ứng đủ yêu cầu chi tiết như phân tích theo khóa học, giảng viên, hay học viên.
* **Liên kết dữ liệu:** Một số bảng trong cơ sở dữ liệu chưa có sự liên kết chặt chẽ, ảnh hưởng đến việc tối ưu hóa quản lý và đảm bảo toàn vẹn dữ liệu.
* **Chưa hỗ trợ các tính năng nâng cao:** Một số chức năng như quản lý tài nguyên giảng dạy hay hỗ trợ học viên đặc biệt chưa được tích hợp.

### ****Hướng phát triển trong tương lai****

Hệ thống có tiềm năng phát triển thêm nhiều tính năng nhằm nâng cao trải nghiệm người dùng và hỗ trợ quản lý toàn diện hơn, bao gồm:

1. **Phát triển báo cáo chi tiết hơn:** Tăng cường khả năng thống kê và báo cáo theo các tiêu chí như khóa học, học kỳ, giảng viên, học viên.
2. **Cải thiện giao diện:** Tích hợp khả năng tùy chỉnh giao diện để đáp ứng nhu cầu cụ thể của từng đối tượng người dùng.
3. **Mở rộng tính năng quản lý:** Thêm các chức năng như quản lý khen thưởng, kỷ luật học viên, và quản lý tài nguyên giảng dạy.
4. **Tăng cường liên kết dữ liệu:** Cải thiện cấu trúc cơ sở dữ liệu để đảm bảo tính chính xác và toàn vẹn của dữ liệu trong toàn bộ hệ thống.
5. **Hỗ trợ công nghệ tiên tiến:** Tích hợp các công nghệ như trí tuệ nhân tạo (AI) hoặc máy học (ML) để phân tích hành vi học viên và đề xuất lộ trình học tập.

### ****Tài liệu tham khảo****

1. Documentation | .NET Framework
2. ASP.NET Core Documentation
3. Entity Framework Core | Microsoft Docs
4. SQL Server Documentation
5. ChatGPT