

BÁO CÁO ĐỒ ÁN CƠ SỞ

WEBSITE HỖ TRỢ HỌC TẬP

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : ThS. Trịnh Công Nhựt

Sinh viên thực hiện : Nguyễn Thị Như Quỳnh 2280602695

: Trần Mai Ngọc Hân 2280600881

: Nguyễn Phan Thanh Hiệp 2280600930

Lớp : 22DTHC5

BÁO CÁO ĐỒ ÁN CƠ SỞ

WEBSITE HỖ TRỢ HỌC TẬP

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : ThS. Trịnh Công Nhựt

Sinh viên thực hiện : Nguyễn Thị Như Quỳnh 2280602695

: Trần Mai Ngọc Hân 2280600881

: Nguyễn Phan Thanh Hiệp 2280600930

Lớp : 22DTHC5

TP. Hồ Chí Minh, 2025

LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Trịnh Công Nhựt – người đã tận tình hướng dẫn, lắng nghe, góp ý và đồng hành cùng chúng em trong suốt quá trình thực hiện đồ án.

Những nhận xét, đánh giá và góp ý chuyên môn từ Thầy không chỉ giúp chúng em nhận ra những điểm còn thiếu sót, mà còn là nguồn động lực lớn để chúng em không ngừng hoàn thiện kiến thức và kỹ năng của bản thân. Sự tận tâm và nhiệt huyết trong giảng dạy cũng như tinh thần trách nhiệm của Thầy đã giúp chúng em vượt qua nhiều khó khăn, hoàn thành đồ án một cách tốt nhất có thể.

Chúng em vô cùng biết ơn sự hỗ trợ quý báu của Thầy và luôn trân trọng những kiến thức, kinh nghiệm quý giá mà Thầy đã truyền đạt. Thầy không chỉ là người hướng dẫn mà còn là người truyền cảm hứng, giúp chúng em thêm tự tin trên hành trình học tập và phát triển.

Kính chúc Thầy luôn mạnh khỏe, hạnh phúc và tiếp tục thành công trên con đường giảng dạy và nghiên cứu khoa học. Mong rằng trong tương lai, chúng em vẫn có cơ hội được học hỏi và nhận được sự chỉ dẫn từ Thầy.

Một lần nữa, chúng em xin chân thành cảm ơn Thầy!

TP.HCM, ngày 14 tháng 05 năm 2025

Nhóm sinh viên thực hiện

NGUYỄN THỊ NHƯ QUỲNH

TRẦN MAI NGỌC HÂN

NGUYỄN PHAN THANH HIỆP

LỜI CAM ĐOAN

Nhóm em xin cam đoan tất cả số liệu và kết quả nghiên cứu từ báo cáo “*Website hỗ trợ học tập*” dựa trên cơ sở tìm tòi và nghiêm cứu của các thành viên trong nhóm, không có bất kỳ sự sao chép hay gian lận nào. Mọi tài liệu tham khảo đều được trích dẫn nguồn gốc đầy đủ và rõ ràng.

TP.HCM, ngày 14 tháng 05 năm 2025

Người cam đoan

NGUYỄN THỊ NHƯ QUỲNH

TRẦN MAI NGỌC HÂN

NGUYỄN PHAN THANH HIỆP

MỤC LỤC

LỜI CẢM ƠN.....	i
LỜI CAM ĐOAN	ii
MỤC LỤC	iii
DANH MỤC HÌNH ẢNH.....	v
LỜI MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN	3
1.1. Khảo sát hiện trạng	3
1.2. Giới thiệu hệ thống website hỗ trợ học tập.....	5
1.2.1 Mục tiêu của hệ thống	5
1.2.2 Các chức năng chính	5
1.3. Lợi ích của hệ thống	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	7
2.1. Tổng quan các công nghệ đã nghiên cứu.....	7
2.1.1. ASP.NET Core MVC	7
2.1.2. Front-end (HTML/CSS/JavaScript)	7
2.1.3. Entity Framework Core	7
2.1.4. Docker	8
2.1.5. SignalR	8
2.1.6. ML.NET	8
2.2. Lựa chọn công nghệ áp dụng.....	9
CHƯƠNG 3: THỰC NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC.....	10
3.1. Cơ sở dữ liệu.....	10
3.1.1. Mô hình thực thể - quan hệ (ERD).....	10
3.1.2. Lược đồ quan hệ:.....	10

3.1.3. Cơ sở dữ liệu trên SQL Server:	10
3.2. Xây dựng các chức năng chính cho hệ thống	13
3.3. Kết quả thực nghiệm chương trình	26
CHƯƠNG 4: KẾT LUẬN	48
4.1. Kết quả đạt được	48
4.2. Hạn chế của hệ thống.....	48
4.3. Hướng phát triển trong tương lai	49
TÀI LIỆU THAM KHẢO	50

DANH MỤC HÌNH ẢNH

Hình 3.1: Hình ảnh sơ đồ ERD.....	10
Hình 3.2: Phương thức Index xử lý hiển thị danh sách các khóa học	13
Hình 3.3: Phương thức Details xử lý hiển thị thông tin chi tiết của khoá học	14
Hình 3.4: Phương thức Create xử lý chức năng thêm khoá học mới.	14
Hình 3.5: Phương thức Update xử lý chức năng chỉnh sửa khoá học.	14
Hình 3.6: Phương thức Delete và DeleteConfirmed xử lý chức năng xoá khoá học ...	15
Hình 3.7: Phương thức Index xử lý hiển thị danh sách các khóa học	15
Hình 3.8: Phương thức Create xử lý chức năng thêm bài học.	15
Hình 3.9: Phương thức Update xử lý chức năng chỉnh sửa bài học.	16
Hình 3.10: Phương thức DeleteConfirmed xử lý chức năng xoá bài học.	16
Hình 3.11: Phương thức ByCourse xử lý hiển thị danh sách các bài học theo khóa học.	
.....	16
Hình 3.12: Phương thức Index xử lý hiển thị danh sách nội dung các bài học và tìm kiếm nội dung theo tên.	17
Hình 3.13: Phương thức Create xử lý chức năng thêm nội dung bài học(1).....	17
Hình 3.14: Phương thức Create xử lý chức năng thêm nội dung bài học(2).....	18
Hình 3.15: Phương thức Create xử lý chức năng thêm nội dung bài học(3).....	18
Hình 3.16: Phương thức SaveImageAsync xử lý lưu hình ảnh.....	18
Hình 3.17: Phương thức SaveVideoAsync xử lý lưu video.	19
Hình 3.18: Phương thức Details xử lý hiển thị thông tin chi tiết của nội dung bài học.	
.....	19
Hình 3.19: Phương thức Update xử lý chức năng chỉnh sửa nội dung bài học.....	19
Hình 3.20: Phương thức DeleteConfirmed xử lý chức năng xoá nội dung bài học.	20
Hình 3.21: Phương thức ByLesson xử lý hiển thị nội dung bài học theo mã bài học...20	
Hình 3.22: Phương thức Index xử lý hiển thị danh sách các bài tập.....	20
Hình 3.23: Phương thức Create xử lý chức năng thêm bài tập.	21
Hình 3.24: Phương thức Update xử lý chức năng chỉnh sửa bài tập(1).	21
Hình 3.25: Phương thức Update xử lý chức năng chỉnh sửa bài tập(2).....	21
Hình 3.26: Phương thức Delete và DeleteConfirmed xử lý chức năng xoá bài tập....	22
Hình 3.27: Phương thức Index xử lý hiển thị danh sách các câu hỏi.	22

Hình 3.28: Phương thức Create xử lý chức năng thêm nội dung bài tập(1)	22
Hình 3.29: Phương thức Create xử lý chức năng thêm nội dung bài tập(2)	23
Hình 3.30: Phương thức Details xử lý hiển thị thông tin chi tiết của câu hỏi	23
Hình 3.31: Phương thức Update xử lý chức năng chỉnh sửa câu hỏi(1)	24
Hình 3.32: Phương thức Update xử lý chức năng chỉnh sửa câu hỏi(2)	24
Hình 3.33: Phương thức Delete và DeleteConfirmed xử lý chức năng xoá câu hỏi.....	25
Hình 3.34: Phương thức Search xử lý chức năng tìm kiếm bài tập theo tên.....	25
Hình 3.35: Phương thức Index xử lý chức năng hiển thị lịch sử nộp bài.....	26
Hình 3.36: Phương thức Index xử lý hiển thị danh sách các tin nhắn lên diễn đàn	26
Hình 3.37: Phương thức SendMessage xử lý chức năng gửi tin nhắn	26
Hình 3.38: Hình ảnh giao diện đăng ký.....	27
Hình 3.39: Hình ảnh giao diện đăng nhập.....	27
Hình 3.40: Hình ảnh giao diện quản lý người dùng.....	28
Hình 3.41: Hình ảnh giao diện danh sách bài học.....	28
Hình 3.42: Hình ảnh giao diện thêm bài học.....	29
Hình 3.43: Hình ảnh giao diện sửa bài học.	29
Hình 3.44: Hình ảnh giao diện xóa bài học.	30
Hình 3.45: Hình ảnh giao diện quản lý danh sách nội dung bài học.....	30
Hình 3.46: Hình ảnh giao diện quản lý danh sách nội dung lọc theo bài học.	31
Hình 3.47: Hình ảnh giao diện thêm nội dung bài học dạng văn bản.	31
Hình 3.48: Hình ảnh giao diện thêm nội dung bài học dạng hình ảnh.....	32
Hình 3.49: Hình ảnh giao diện thêm nội dung bài học dạng video.....	32
Hình 3.50: Hình ảnh giao diện cập nhật nội dung bài học.	33
Hình 3.51: Hình ảnh giao diện chi tiết nội dung bài học.....	33
Hình 3.52: Hình ảnh giao diện xóa nội dung bài học.....	34
Hình 3.53: Hình ảnh giao diện bài học.....	34
Hình 3.54: Hình ảnh giao diện danh sách khóa học.....	35
Hình 3.55: Hình ảnh giao diện thêm khóa học mới.	35
Hình 3.56: Hình ảnh giao diện chỉnh sửa khóa học.	36
Hình 3.57: Hình ảnh giao diện xóa khóa học.	36
Hình 3.58: Hình ảnh giao diện quản lý danh sách các bài tập.	37
Hình 3.59: Hình ảnh giao diện cập nhật thông tin bài tập.....	37

Hình 3.60: Hình ảnh giao diện xóa bài tập.....	38
Hình 3.61: Hình ảnh giao diện danh sách các bài tập.	38
Hình 3.62: Hình ảnh giao diện tìm kiếm bài tập theo tên.	39
Hình 3.63: Hình ảnh giao diện danh sách câu hỏi.....	39
Hình 3.64: Hình ảnh giao diện thêm câu hỏi dạng trắc nghiệm(1).	40
Hình 3.65: Hình ảnh giao diện thêm câu hỏi dạng trắc nghiệm(2).	40
Hình 3.66: Hình ảnh giao diện thêm câu hỏi dạng viết mã lệnh(1).	41
Hình 3.67: Hình ảnh giao diện thêm câu hỏi dạng viết mã lệnh(2).	41
Hình 3.68: Hình ảnh giao diện cập nhật câu hỏi dạng trắc nghiệm(1).....	42
Hình 3.69: Hình ảnh giao diện cập nhật câu hỏi dạng trắc nghiệm(2).....	42
Hình 3. 70: Hình ảnh giao diện chi tiết câu hỏi.....	43
Hình 3.71: Hình ảnh giao diện xóa câu hỏi.	43
Hình 3.72: Hình ảnh giao diện chạy code.	44
Hình 3.73: Hình ảnh giao diện chạy code thành công.....	44
Hình 3.74: Hình ảnh giao diện thông báo lỗi khi lỗi code.	45
Hình 3.75: Hình ảnh giao diện nộp bài thành công.....	45
Hình 3.76: Hình ảnh giao diện lịch sử nộp bài.....	46
Hình 3.77: Hình ảnh giao diện xem code đã nộp.	46
Hình 3.78: Hình ảnh giao diện diễn đàn chat cộng đồng.	47

LỜI MỞ ĐẦU

1. Lý do chọn đề tài

Trong chương trình đào tạo ngành Công nghệ Thông tin, môn Cấu trúc dữ liệu và Giải thuật là một môn học nền tảng quan trọng, giúp sinh viên rèn luyện tư duy logic, khả năng phân tích và giải quyết vấn đề. Tuy nhiên, nội dung của môn học thường mang tính trừu tượng, nhiều khái niệm phức tạp, đòi hỏi người học phải có sự kết hợp giữa lý thuyết và thực hành để nắm bắt và vận dụng hiệu quả.

Hiện nay có khá nhiều tài liệu và nền tảng học tập trực tuyến hỗ trợ môn học này, tuy nhiên phần lớn các tài liệu vẫn mang tính hàn lâm, sử dụng tiếng Anh, hoặc chưa thực sự phù hợp với sinh viên Việt Nam. Điều này gây ra không ít khó khăn trong việc tiếp cận kiến thức, đặc biệt là đối với những sinh viên còn hạn chế về ngoại ngữ hoặc cần sự minh họa trực quan, dễ hiểu.

Xuất phát từ thực tế đó, nhóm chúng tôi xây dựng một website hỗ trợ học tập môn Cấu trúc dữ liệu và Giải thuật với giao diện đơn giản, sử dụng tiếng Việt, và tập trung cung cấp các nội dung dễ tiếp cận hơn cho sinh viên. Website nhằm mục tiêu hỗ trợ người học trong việc ôn tập kiến thức, thực hành cơ bản và có thêm tài liệu tham khảo phù hợp với chương trình học trong nước. Dù không nhằm mục tiêu thay thế hay cạnh tranh với các nền tảng lớn, hệ thống của nhóm hướng đến sự tiện lợi, dễ sử dụng và thân thiện với sinh viên Việt Nam.

2. Mục đích nghiên cứu

Mục tiêu chính của đề tài là xây dựng một website hỗ trợ học tập với các chức năng cơ bản và dễ sử dụng, bao gồm:

- Quản lý thông tin người dùng, khóa học, bài học và bài tập .
- Cho phép người dùng học tập và thực hành qua các bài học, bài tập trên hệ thống.
- Quản trị viên quản lý toàn bộ hệ thống.
- Ngoài ra còn hỗ trợ người dùng trao đổi trực tiếp với nhau qua diễn đàn trên website.

3. Ý nghĩa của việc nghiên cứu

Kết quả của nghiên cứu mang ý nghĩa:

- Giúp tìm hiểu thêm về framework ASP.NET
- Giúp tìm hiểu về ngôn ngữ C#
- Giúp tìm hiểu sử dụng thư viện System.Web.Mvc
- Giúp tìm hiểu sử dụng Visual Studio

4. Kết cấu đề tài:

- LỜI MỞ ĐẦU
- CHƯƠNG 1: TỔNG QUAN
- CHƯƠNG 2: CƠ SỞ LÝ THUYẾT
- CHƯƠNG 3: THỰC NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC
- CHƯƠNG 4: KẾT LUẬN

CHƯƠNG 1: TỔNG QUAN

1.1. Khảo sát hiện trạng

Website hỗ trợ học tập nhằm mục đích tạo ra một nền tảng trực tuyến giúp người dùng học lập trình một cách hiệu quả. Để hiểu rõ hơn về thị trường hiện có, nhóm đã tiến hành khảo sát các hệ thống tương tự. Dưới đây là kết quả khảo sát:

a. *Codecademy*

- Học lập trình thông qua các bài tập.
- Cung cấp các khóa học về nhiều ngôn ngữ lập trình như Python, JavaScript, HTML/CSS, SQL, v.v.
- Có hệ thống theo dõi tiến độ học tập.
- Không hỗ trợ tiếng Việt.

b. *freeCodeCamp*

- Cung cấp các khóa học lập trình miễn phí, học thông qua các bài tập và dự án thực tế.
- Nhận chứng chỉ sau khi hoàn thành khóa học.
- Có cộng đồng hỗ trợ và chia sẻ kiến thức.
- Không hỗ trợ tiếng Việt.

c. *W3Schools*

- Cung cấp tài liệu học lập trình trực tuyến, hỗ trợ nhiều ngôn ngữ lập trình như HTML, CSS, JavaScript, Python, v.v.
- Có các bài tập và ví dụ trực quan.
- Không hỗ trợ tiếng Việt.

d. *Khan Academy*

- Cung cấp các khóa học lập trình cơ bản, học thông qua video và bài tập tương tác.
- Có hỗ trợ tiếng Việt cho một số khóa học.

e. *Udemy*

- Cung cấp các khóa học lập trình đa dạng, học thông qua video bài giảng và bài tập thực hành.
- Có khóa học từ nhiều giảng viên khác nhau.
- Có một số khóa học hỗ trợ tiếng Việt.

f. *Coursera*

- Cung cấp các khóa học thông qua video bài giảng, bài tập và dự án.
- Có chứng chỉ sau khi hoàn thành khóa học.
- Có phụ đề tiếng Việt cho một số khóa học.

g. *SoloLearn*

- Học lập trình thông qua các bài học ngắn và bài tập tương tác.
- Hỗ trợ nhiều ngôn ngữ lập trình như Python, Java, C++, v.v.
- Có cộng đồng hỗ trợ và chia sẻ kiến thức.
- Không hỗ trợ tiếng Việt.

h. *LeetCode*

- Trang web nổi tiếng với các bài tập về cơ sở dữ liệu và giải thuật gồm đa dạng các bài tập từ dễ đến nâng cao
- Giúp người dùng luyện khả năng tư duy và thuật toán cho quá trình thực tập và phỏng vấn
- Thiếu bài giảng cụ thể, khá khó khăn cho người mới sử dụng
- Không hỗ trợ tiếng Việt

Hầu hết các hệ thống đều có phiên bản miễn phí, nhưng thường đi kèm với các hạn chế về tính năng hoặc quảng cáo. Phiên bản trả phí thường cung cấp thêm các tính năng nâng cao và loại bỏ quảng cáo. Đa số các hệ thống không hỗ trợ tiếng Việt, chỉ có một số ít như Khan Academy và Udemy có hỗ trợ tiếng Việt nhưng chỉ cho một số khóa học.

Kết luận: Các hệ thống hiện có đã cung cấp nhiều tính năng hữu ích cho việc học lập trình, tuy nhiên vẫn còn thiếu sự hỗ trợ tiếng Việt và các tính năng đặc thù cho người dùng Việt Nam.

1.2. Giới thiệu hệ thống website hỗ trợ học tập

1.2.1 Mục tiêu của hệ thống

Hệ thống được xây dựng với mục đích hỗ trợ sinh viên học tập hiệu quả hơn môn Cấu trúc dữ liệu và Giải thuật, thông qua việc cung cấp nội dung học tập bằng tiếng Việt, giao diện thân thiện và dễ sử dụng. Hệ thống đóng vai trò như một công cụ hỗ trợ bổ sung, cung cấp kiến thức lý thuyết, thực hành, và các bài tập, giúp người học tiếp cận chủ đề một cách trực quan và hiệu quả hơn.

1.2.2 Các chức năng chính

- Bài giảng lý thuyết dễ hiểu, với các hình ảnh mô phỏng trực quan

- Cung cấp nội dung học tập chi tiết về cơ sở dữ liệu và giải thuật.
- Hệ thống phân chia bài giảng theo từng chủ đề, giúp người học dễ dàng tiếp cận.
- Các bài giảng được trình bày trực quan, có hình ảnh minh họa và ví dụ thực tế.

- Hệ thống bài tập từ dễ đến khó

- Bộ bài tập phong phú, sắp xếp theo cấp độ từ cơ bản đến nâng cao.
- Cung cấp môi trường viết và chạy mã trực tiếp trên website, người học có thể làm bài trực tiếp trên hệ thống và nhận phản hồi ngay lập tức.
- Cung cấp giải thích chi tiết cho từng bài tập để hỗ trợ quá trình học tập.

- Chấm điểm tự động và gợi ý sửa lỗi

- Hệ thống chấm bài tự động, kiểm tra tính đúng sai của lời giải.
- Nếu có lỗi, website sẽ đưa ra gợi ý để người học sửa đổi và hoàn thiện bài làm.
- Bài tập sẽ có video hướng dẫn giải chi tiết giúp người học hiểu rõ phương pháp tiếp cận.

- Tích hợp cộng đồng học tập

- Hỗ trợ tính năng thảo luận nhóm, giúp sinh viên trao đổi kiến thức và giải quyết bài toán cùng nhau.

- Cho phép người dùng đặt câu hỏi và nhận phản hồi từ cộng đồng.

1.3. Lợi ích của hệ thống

- Giúp sinh viên tự học và luyện tập hiệu quả hơn.
- Giao diện gần gũi, dễ sử dụng.
- Xây dựng cộng đồng học tập nơi sinh viên có thể chia sẻ và giúp đỡ lẫn nhau.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan các công nghệ đã nghiên cứu

2.1.1. *ASP.NET Core MVC*

- ASP.NET Core MVC là một Framework phát triển web đa nền tảng, hỗ trợ xây dựng ứng dụng web theo mô hình MVC (Model-View-Controller).

- **Ưu điểm:**
 - Hiệu suất cao, hỗ trợ đa nền tảng (Windows, Linux, macOS).
 - Tích hợp sẵn các tính năng bảo mật như xác thực, phân quyền.
 - Hỗ trợ Entity Framework Core để quản lý cơ sở dữ liệu.
 - Dễ dàng tích hợp với các dịch vụ đám mây như Azure.
- **Nhược điểm:**
 - Cần kiến thức về C# và .NET.
 - Cộng đồng hỗ trợ nhỏ hơn so với các framework như Node.js hoặc Django.

2.1.2. *Front-end (HTML/CSS/JavaScript)*

- HTML/CSS/JavaScript là các công nghệ cơ bản để xây dựng giao diện người dùng.

- **Ưu điểm:**
 - Dễ học và sử dụng.
 - Hỗ trợ nhiều thư viện và framework như Bootstrap, React.js, Vue.js.
- **Nhược điểm:**
 - Cần kết hợp với back-end để xử lý logic nghiệp vụ.

2.1.3. *Entity Framework Core*

- ORM (Object-Relational Mapping) giúp quản lý cơ sở dữ liệu bằng C#.

- **Ưu điểm:**
 - Hỗ trợ nhiều loại cơ sở dữ liệu (SQL Server, MySQL, PostgreSQL).

- Giảm thiểu việc viết mã SQL thủ công.
- Nhược điểm:
- Cần hiểu biết về LINQ và cách thức hoạt động của ORM.

2.1.4. Docker

- Docker là công cụ để đóng gói ứng dụng vào container, giúp triển khai và chạy ứng dụng trên nhiều môi trường khác nhau.

- Ưu điểm:
- Đảm bảo tính nhất quán giữa các môi trường phát triển và sản xuất.
- Dễ dàng tích hợp với CI/CD.
- Nhược điểm:
- Cần thời gian để học và cấu hình.

2.1.5. SignalR

- SignalR là thư viện hỗ trợ real-time communication trong ứng dụng web.

- Ưu điểm:
- Dễ dàng tích hợp với ASP.NET Core.
- Hỗ trợ WebSocket, Server-Sent Events, và Long Polling.
- Nhược điểm:
- Cần hiểu biết về real-time communication.

2.1.6. ML.NET

- ML.NET là thư viện Machine Learning của .NET.

- Ưu điểm:
- Dễ dàng tích hợp với ứng dụng ASP.NET Core.
- Hỗ trợ các bài toán như phân loại, hồi quy, và NLP.
- Nhược điểm:

- Cần kiến thức về Machine Learning.

2.2. Lựa chọn công nghệ áp dụng

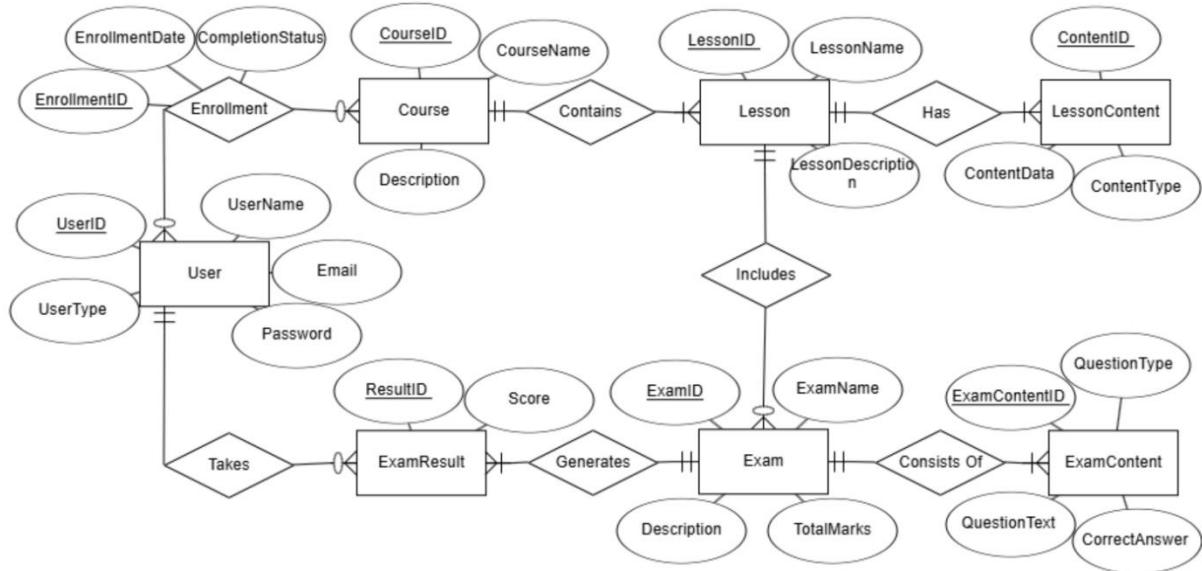
Dựa trên kết quả nghiên cứu, nhóm quyết định sử dụng các công nghệ sau:

- Back-end: ASP.NET Core MVC.
- Front-end: HTML/CSS/JavaScript, Bootstrap.
- Cơ sở dữ liệu: Entity Framework Core với SQL Server.
- Real-time communication: SignalR.
- Chạy mã trực tiếp: Docker và JDoodle API.
- Machine Learning: ML.NET (nếu cần).

CHƯƠNG 3: THỰC NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC

3.1. Cơ sở dữ liệu

3.1.1. Mô hình thực thể - quan hệ (ERD)



Hình 3.1: Hình ảnh sơ đồ ERD.

3.1.2. Lược đồ quan hệ:

User(UserID, UserName, Email, Password, UserType)

Course(CourseID, CourseName, Description)

Lesson(LessonID, CourseID, LessonName, LessonDescription)

LessonContent(ContentID, LessonID, ContentType, ContentData)

Enrollment(EnrollmentID, UserID, CourseID, EnrollmentDate, CompletionStatus)

Exam(ExamID, LessonID, ExamName, Description, TotalMarks)

ExamContent(ExamContentID, ExamID, QuestionType, QuestionText, CorrectAnswer)

ExamResult(ResultID, UserID, ExamID, Score)

3.1.3. Cơ sở dữ liệu trên SQL Server:

CREATE TABLE User (

 UserID INT PRIMARY KEY,

```
    UserName VARCHAR(255),  
    Email VARCHAR(255),  
    Password VARCHAR(255),  
    UserType VARCHAR(50)  
);
```

```
CREATE TABLE Course (  
    CourseID INT PRIMARY KEY,  
    CourseName VARCHAR(255),  
    Description TEXT  
);
```

```
CREATE TABLE Lesson (  
    LessonID INT PRIMARY KEY,  
    CourseID INT,  
    LessonName VARCHAR(255),  
    LessonDescription TEXT,  
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)  
);
```

```
CREATE TABLE LessonContent (  
    ContentID INT PRIMARY KEY,  
    LessonID INT,  
    ContentType VARCHAR(50),
```

```
ContentData TEXT,  
FOREIGN KEY (LessonID) REFERENCES Lesson(LessonID)  
);
```

```
CREATE TABLE Enrollment (  
    EnrollmentID INT PRIMARY KEY,  
    UserID INT,  
    CourseID INT,  
    EnrollmentDate DATE,  
    CompletionStatus VARCHAR(50),  
    FOREIGN KEY (UserID) REFERENCES [User](UserID),  
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)  
);
```

```
CREATE TABLE Exam (  
    ExamID INT PRIMARY KEY,  
    LessonID INT,  
    ExamName VARCHAR(255),  
    Description TEXT,  
    TotalMarks INT,  
    FOREIGN KEY (LessonID) REFERENCES Lesson(LessonID)  
);
```

```
CREATE TABLE ExamContent (
```

```

ExamContentID INT PRIMARY KEY,
ExamID INT,
QuestionType VARCHAR(50),
QuestionText TEXT,
CorrectAnswer TEXT,
FOREIGN KEY (ExamID) REFERENCES Exam(ExamID)
);

```

```

CREATE TABLE ExamResult (
ResultID INT PRIMARY KEY,
UserID INT,
ExamID INT,
Score INT,
FOREIGN KEY (UserID) REFERENCES [User](UserID),
FOREIGN KEY (ExamID) REFERENCES Exam(ExamID)
);

```

3.2. Xây dựng các chức năng chính cho hệ thống

Quản lý khóa học:

- **Hiển thị các khóa học:** Hành động Index hiển thị danh sách các khóa học có trong hệ thống

```

public async Task<IActionResult> Index()
{
    var course = await _courseRepository.GetAllAsync();
    return View(course);
}

```

Hình 3.2: Phương thức Index xử lý hiển thị danh sách các khóa học.

- **Xem chi tiết khóa học:** Hành động Details hiển thị thông tin chi tiết của một khóa học theo ID

```

public async Task<IActionResult> Details(int id)
{
    var course = await _courseRepository.GetByIdAsync(id);
    if (course == null)
    {
        return NotFound();
    }

    return View(course);
}

```

Hình 3.3: Phương thức Details xử lý hiển thị thông tin chi tiết của khóa học.

- **Thêm khóa học mới:** Hành động Create cho phép tạo môn học mới.

```

public IActionResult Create()
{
    return View();
}

[HttpPost]
0 references
public async Task<IActionResult> Create(Course course)
{
    if (ModelState.IsValid)
    {
        await _courseRepository.AddAsync(course);
        return RedirectToAction(nameof(Index));
    }
    return View(course);
}

```

Hình 3.4: Phương thức Create xử lý chức năng thêm khóa học mới.

- **Chỉnh sửa khóa học:** Hành động Update cung cấp form chỉnh sửa thông tin khóa học.

```

public async Task<IActionResult> Update(int id)
{
    var course = await _courseRepository.GetByIdAsync(id);
    if (course == null)
    {
        return NotFound();
    }
    return View(course);
}

[HttpPost]
0 references
public async Task<IActionResult> Update(int id, Course course)
{
    if (id != course.CourseID)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        await _courseRepository.UpdateAsync(course);
        return RedirectToAction(nameof(Index));
    }
    return View(course);
}

```

Hình 3.5: Phương thức Update xử lý chức năng chỉnh sửa khóa học.

- **Xóa khóa học:** Hành động Delete và DeleteConfirmed cho phép xóa một khóa học sau khi xác nhận.

```

public async Task<IActionResult> Delete(int id)
{
    var course = await _courseRepository.GetByIdAsync(id);
    if (course == null)
    {
        return NotFound();
    }
    return View(course);
}
[HttpPost]
0 references
public async Task<IActionResult> DeleteConfirmed(int CourseID)
{
    var course = await _courseRepository.GetByIdAsync(CourseID);
    if (course != null)
    {
        await _courseRepository.DeleteAsync(CourseID);
    }

    return RedirectToAction(nameof(Index));
}

```

Hình 3.6: Phương thức Delete và DeleteConfirmed xử lý chức năng xoá khóa học.

Quản lý bài học

- **Hiển thị danh sách các bài học:** Hiển thị danh sách các bài học có trong hệ thống.

```

public async Task<IActionResult> Index()
{
    var lessons = await _lessonRepository.GetAllAsync();
    return View(lessons);
}

```

Hình 3.7: Phương thức Index xử lý hiển thị danh sách các khóa học.

- **Thêm bài học:** Cho phép thêm một bài học với một trong các khóa học đã có.

```

[HttpPost]
0 references
public async Task<IActionResult> Create(Lesson lesson)
{
    if (!ModelState.IsValid)
    {
        foreach (var error in ModelState.Values.SelectMany(v => v.Errors))
        {
            Console.WriteLine(error.ErrorMessage);
        }
    }

    if (ModelState.IsValid)
    {
        await _lessonRepository.AddAsync(lesson);
        return RedirectToAction(nameof(Index));
    }

    var courses = await _courseRepository.GetAllAsync();
    ViewBag.Courses = new SelectList(courses, "CourseID", "CourseName");
    return View(lesson);
}

```

Hình 3.8: Phương thức Create xử lý chức năng thêm bài học.

- **Chỉnh sửa bài học:** Cho phép chỉnh sửa tên bài học, mô tả bài học và thay đổi khóa học chứa bài học đó.

```
[HttpPost]
0 references
public async Task<IActionResult> Update(int id, Lesson lesson)
{
    if (id != lesson.LessonID)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        var existingLesson = await _lessonRepository.GetByIdAsync(id);
        existingLesson.LessonName = lesson.LessonName;
        existingLesson.LessonDescription = lesson.LessonDescription;
        existingLesson.CourseID = lesson.CourseID;
        await _lessonRepository.UpdateAsync(existingLesson);
        return RedirectToAction(nameof(Index));
    }
    var courses = await _courseRepository.GetAllAsync();
    ViewBag.Courses = new SelectList(courses, "CourseID", "CourseName");
    return View(lesson);
}
```

Hình 3.9: Phương thức Update xử lý chức năng chỉnh sửa bài học.

- **Xóa bài học:** Cho phép xóa một bài học trên hệ thống sau khi người dùng xác nhận.

```
[HttpPost, ActionName("DeleteConfirmed")]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    await _lessonRepository.DeleteAsync(id);
    return RedirectToAction(nameof(Index));
```

Hình 3.10: Phương thức DeleteConfirmed xử lý chức năng xoá bài học.

- **Hiển thị danh sách bài học theo khóa học:** Hiển thị danh sách các bài học thuộc một khóa học cụ thể thông qua mã khóa học (courseId)

```
public async Task<IActionResult> ByCourse(int courseId)
{
    var lessons = await _lessonRepository.GetLessonsByCourseIdAsync(courseId);
    ViewBag.Course = await _courseRepository.GetByIdAsync(courseId);
    return View(lessons);
```

Hình 3.11: Phương thức ByCourse xử lý hiển thị danh sách các bài học theo khóa học.

Quản lý nội dung bài học

- **Hiển thị danh sách và tìm kiếm nội dung bài học:** Hiển thị danh sách nội dung các bài học có trên hệ thống và tìm kiếm nội dung theo tên.

```

public async Task<IActionResult> Index(int? lessonId)
{
    var allLessons = await _lessonRepository.GetAllAsync();
    ViewBag.Lessons = allLessons;
    ViewBag.SelectedLessonId = lessonId;

    IEnumerable<LessonContent> contents;

    if (lessonId.HasValue)
    {
        contents = await _lessoncontentRepository.GetByLessonIdAsync(lessonId.Value);
    }
    else
    {
        contents = await _lessoncontentRepository.GetAllAsync();
    }

    return View(contents);
}

```

Hình 3.12: Phương thức Index xử lý hiển thị danh sách nội dung các bài học và tìm kiếm nội dung theo tên.

- **Thêm nội dung bài học:** Thêm nội dung bài học với các dạng văn bản, hình ảnh và video.

```

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create(LessonContent lessoncontent, IFormFile? ImageFile, IFormFile? VideoFile,
    string? VideoUrl, string? TextContent)
{
    if (lessoncontent.ContentType == "text")
    {
        if (string.IsNullOrWhiteSpace(TextContent))
        {
            ModelState.AddModelError("TextContent", "Nhập nội dung văn bản.");
        }
        else
        {
            lessoncontent.ContentData = TextContent;
        }
    }
    else if (lessoncontent.ContentType == "image")
    {
        if (ImageFile != null && ImageFile.Length > 0)
        {
            string fileName = Guid.NewGuid().ToString() + Path.GetExtension(ImageFile.FileName);
            string filePath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/images", fileName);
            Directory.CreateDirectory(Path.GetDirectoryName(filePath)!);

            using (var stream = new FileStream(filePath, FileMode.Create))
            {
                await ImageFile.CopyToAsync(stream);
            }
        }
    }
}

```

Hình 3.13: Phương thức Create xử lý chức năng thêm nội dung bài học(1).

```

        lessoncontent.ContentData = "/images/" + fileName;
    }
    else
    {
        ModelState.AddModelError("ImageFile", "Chọn một file ảnh.");
    }
}
else if (lessoncontent.ContentType == "video")
{
    if (VideoFile != null && VideoFile.Length > 0)
    {
        string fileName = Guid.NewGuid().ToString() + Path.GetExtension(VideoFile.FileName);
        string filePath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/videos", fileName);
        using (var stream = new FileStream(filePath, FileMode.Create))
        {
            await VideoFile.CopyToAsync(stream);
        }
        lessoncontent.ContentData = "/videos/" + fileName;
    }
    else if (!string.IsNullOrWhiteSpace(VideoUrl))
    {
        lessoncontent.ContentData = VideoUrl;
    }
    else
    {
        ModelState.AddModelError("VideoFile", "Chọn file hoặc nhập link video.");
    }
}

```

Hình 3.14: Phương thức Create xử lý chức năng thêm nội dung bài học(2).

```

if (!ModelState.IsValid)
{
    ViewBag.Lessons = new SelectList(await _lessonRepository.GetAllAsync(), "LessonID", "LessonName");
    return View(lessoncontent);
}

await _lessoncontentRepository.AddAsync(lessoncontent);
foreach (var ModelState in ModelState)
{
    foreach (var error in ModelState.Value.Errors)
    {
        Console.WriteLine($"Lỗi ở {ModelState.Key}: {error.ErrorMessage}");
    }
}

return RedirectToAction(nameof(Index));
}

```

Hình 3.15: Phương thức Create xử lý chức năng thêm nội dung bài học(3).

- Lưu hình ảnh nội dung bài học: Cho phép lưu nội dung bài học với dạng hình ảnh vào thư mục “wwwroot/images”

```

private async Task<string> SaveImageAsync(IFormFile image)
{
    var fileName = Guid.NewGuid().ToString() + Path.GetExtension(image.FileName);
    var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/images", fileName);

    using (var stream = new FileStream(path, FileMode.Create))
    {
        await image.CopyToAsync(stream);
    }

    return "/images/" + fileName;
}

```

Hình 3.16: Phương thức SaveImageAsync xử lý lưu hình ảnh.

- Lưu video nội dung bài học: Cho phép lưu nội dung bài học với dạng video vào thư mục “wwwroot/videos”

```

private async Task<string> SaveVideoAsync(IFormFile video)
{
    var fileName = Guid.NewGuid().ToString() + Path.GetExtension(video.FileName);
    var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/videos", fileName);

    using (var stream = new FileStream(path, FileMode.Create))
    {
        await video.CopyToAsync(stream);
    }

    return "/videos/" + fileName;
}

```

Hình 3.17: Phương thức SaveVideoAsync xử lý lưu video.

- **Hiển thị chi tiết nội dung bài học:** Cho phép xem chi tiết nội dung bài học.

```

public async Task<IActionResult> Details(int id)
{
    if (id <= 0)
    {
        return BadRequest("ID bài học không hợp lệ.");
    }
    string filePath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/json", $"lesson_{id}.json");

    if (!System.IO.File.Exists(filePath))
    {
        return NotFound($"Không tìm thấy bài học với ID {id}.");
    }
    // Đọc nội dung JSON
    string jsonData = System.IO.File.ReadAllText(filePath);
    // Chuyển đổi JSON thành đối tượng Lesson
    Lesson lesson = JsonConvert.DeserializeObject<Lesson>(jsonData);

    if (lesson == null || lesson.LessonContents == null)
    {
        return NotFound("Dữ liệu bài học không hợp lệ.");
    }

    if (lesson.CourseID == 0)
    {
        lesson.CourseID = 1;
    }
    ViewBag.Lesson = lesson;
    return View("ByLesson", lesson.LessonContents);
}

```

Hình 3.18: Phương thức Details xử lý hiển thị thông tin chi tiết của nội dung bài học.

- **Chỉnh sửa nội dung bài học:** Cho phép thay đổi các thông tin trong nội dung bài học.

```

[HttpPost]
0 references
public async Task<IActionResult> Update(int id, LessonContent lessoncontent)
{
    if (id != lessoncontent.ContentID)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        var existingLessonContent = await _lessoncontentRepository.GetByIdAsync(id);
        existingLessonContent.ContentType = lessoncontent.ContentType;
        existingLessonContent.ContentData = lessoncontent.ContentData;
        existingLessonContent.LessonID = lessoncontent.LessonID;
        await _lessoncontentRepository.UpdateAsync(existingLessonContent);
        return RedirectToAction(nameof(Index));
    }
    var lessons = await _lessonRepository.GetAllAsync();
    ViewBag.Lessons = new SelectList(lessons, "LessonID", "LessonName");
    return View(lessoncontent);
}

```

Hình 3.19: Phương thức Update xử lý chức năng chỉnh sửa nội dung bài học.

- **Xóa nội dung bài học:** Cho phép xóa nội dung bài học sau khi xác nhận.

```
[HttpPost, ActionName("DeleteConfirmed")]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    await _lessoncontentRepository.DeleteAsync(id);
    return RedirectToAction(nameof(Index));
}
```

Hình 3.20: Phương thức DeleteConfirmed xử lý chức năng xoá nội dung bài học.

- **Hiển thị nội dung bài học theo bài học:** Hiển thị các nội dung bài học theo mã bài học (lessonId).

```
public IActionResult ByLesson(int lessonId)
{
    if (lessonId <= 0)
    {
        return BadRequest("ID bài học không hợp lệ.");
    }
    string filePath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/json", $"lesson_{lessonId}.json");

    if (!System.IO.File.Exists(filePath))
    {
        return NotFound($"Không tìm thấy bài học với ID {lessonId}.");
    }

    // Đọc nội dung JSON
    string jsonData = System.IO.File.ReadAllText(filePath);

    // Chuyển đổi JSON thành đối tượng Lesson
    Lesson lesson = JsonConvert.DeserializeObject<Lesson>(jsonData);

    if (lesson == null)
    {
        return NotFound($"Không có nội dung bài học với ID {lessonId}.");
    }

    return View(lesson);
}
```

Hình 3.21: Phương thức ByLesson xử lý hiển thị nội dung bài học theo mã bài học.

Quản lý bài tập

- **Hiển thị danh sách bài tập:** Hiển thị tất cả các bài tập có trong hệ thống.

```
public async Task<IActionResult> Index()
{
    var exams = await _examRepository.GetAllAsync();
    return View(exams);
}
```

Hình 3.22: Phương thức Index xử lý hiển thị danh sách các bài tập.

- **Thêm bài tập:** Cho phép thêm bài tập thuộc bài học cụ thể.

```

public async Task<IActionResult> Create()
{
    var lessons = await _lessonRepository.GetAllAsync();
    ViewBag.Lessons = new SelectList(lessons, "LessonID", "LessonName");

    return View();
}
[HttpPost]
0 references
public async Task<IActionResult> Create(Exam exam)
{
    if (ModelState.IsValid)
    {
        await _examRepository.AddExamAsync(exam);
        return RedirectToAction(nameof(Index));
    }
    var lessons = await _lessonRepository.GetAllAsync();
    ViewBag.Lessons = new SelectList(lessons, "LessonID", "LessonName");
    return View(exam);
}

```

Hình 3.23: Phương thức Create xử lý chức năng thêm bài tập.

- **Chỉnh sửa bài tập:** Cho phép thay đổi các thông tin bài tập: tên bài tập, mô tả, tổng điểm và bài học.

```

public async Task<IActionResult> Update(int id)
{
    var exams = await _examRepository.GetByIdAsync(id);
    if (exams == null)
    {
        return NotFound();
    }
    var lessons = await _lessonRepository.GetAllAsync();
    ViewBag.Lessons = new SelectList(lessons, "LessonID", "LessonName", exams.LessonID);
    return View(exams);
}

```

Hình 3.24: Phương thức Update xử lý chức năng chỉnh sửa bài tập(1).

```

[HttpPost]
0 references
public async Task<IActionResult> Update(int id, Exam exam)
{
    if (id != exam.ExamID)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        var existingExam = await _examRepository.GetByIdAsync(id);
        existingExam.ExamName = exam.ExamName;
        existingExam.Description = exam.Description;
        existingExam.TotalMarks = exam.TotalMarks;
        existingExam.LessonID = exam.LessonID;
        await _examRepository.UpdateAsync(existingExam);
        return RedirectToAction(nameof(Index));
    }
    var lessons = await _lessonRepository.GetAllAsync();
    ViewBag.Lessons = new SelectList(lessons, "LessonID", "LessonName");
    return View(exam);
}

```

Hình 3.25: Phương thức Update xử lý chức năng chỉnh sửa bài tập(2).

- **Xóa bài tập:** Cho phép xóa bài tập sau khi xác nhận.

```
public async Task<IActionResult> Delete(int id)
{
    var exams = await _examRepository.GetByIdAsync(id);
    if (exams == null)
    {
        return NotFound();
    }
    return View(exams);
}
[HttpPost, ActionName("DeleteConfirmed")]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    await _examRepository.DeleteAsync(id);
    return RedirectToAction(nameof(Index));
}
```

Hình 3.26: Phương thức Delete và DeleteConfirmed xử lý chức năng xoá bài tập.

Quản lý nội dung bài tập

- **Hiển thị danh sách nội dung bài tập:** Hiển thị toàn bộ các câu hỏi trong bài tập.

```
public async Task<IActionResult> Index()
{
    var examcontents = await _examcontentRepository.GetAllAsync();
    return View(examcontents);
}
```

Hình 3.27: Phương thức Index xử lý hiển thị danh sách các câu hỏi.

- **Thêm nội dung bài tập:** Thêm nội dung bài tập gồm câu hỏi dưới dạng trắc nghiệm hoặc viết mã lệnh.

```
public async Task<IActionResult> Create()
{
    await LoadExamsAsync();

    var model = new ExamContent
    {
        Options = new List<string> { "", "", "", "" }
    };

    return View(model);
}
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create(ExamContent examcontent)
{
    if (ModelState.IsValid)
    {
        if (examcontent.QuestionType == "MultipleChoice")
        {
            if (examcontent.Options == null || examcontent.Options.Count < 4 ||
                examcontent.Options.Any(opt => string.IsNullOrWhiteSpace(opt)))
            {
                ModelState.AddModelError(string.Empty, "Vui lòng nhập đầy đủ 4 đáp án.");
                await LoadExamsAsync();
                return View(examcontent);
            }
            examcontent.CorrectAnswer = examcontent.SelectedAnswer;
        }
    }
}
```

Hình 3.28: Phương thức Create xử lý chức năng thêm nội dung bài tập(1).

```

        }
        else if (examcontent.QuestionType == "Code")
        {
            if (string.IsNullOrWhiteSpace(examcontent.StarterCode))
            {
                ModelState.AddModelError(string.Empty, "Vui lòng nhập mã nguồn cho câu hỏi.");
                await LoadExamsAsync();
                return View(examcontent);
            }
            if (string.IsNullOrWhiteSpace(examcontent.SampleInput) || string.IsNullOrWhiteSpace(examcontent.ExpectedOutput))
            {
                ModelState.AddModelError(string.Empty, "Vui lòng nhập input mẫu và output mẫu.");
                await LoadExamsAsync();
                return View(examcontent);
            }
            examcontent.CorrectAnswer = null;
        }
        await _examcontentRepository.AddAsync(examcontent);
        return RedirectToAction(nameof(Index));
    }
    await LoadExamsAsync();
    return View(examcontent);
}

```

Hình 3.29: Phương thức Create xử lý chức năng thêm nội dung bài tập(2).

- **Hiển thị chi tiết câu hỏi:** Hiển thị chi tiết nội dung và đáp án của câu hỏi theo ID.

```

public async Task<IActionResult> Details(int id)
{
    var examcontent = await _examcontentRepository.GetByIdAsync(id);
    if (examcontent == null)
    {
        return NotFound();
    }
    return View(examcontent);
}

```

Hình 3.30: Phương thức Details xử lý hiển thị thông tin chi tiết của câu hỏi.

- **Chỉnh sửa câu hỏi:** Cho phép thay đổi các thông tin của câu hỏi và đáp án.

```

public async Task<IActionResult> Update(int id)
{
    var examcontent = await _examcontentRepository.GetByIdAsync(id);
    if (examcontent == null)
    {
        return NotFound();
    }

    await LoadExamsAsync();
    return View(examcontent);
}
[HttpPost]
0 references
public async Task<IActionResult> Update(int id, ExamContent examcontent)
{
    if (id != examcontent.ExamContentID)
    {
        return NotFound();
    }
    if (ModelState.IsValid)
    {
        var existingExamContent = await _examcontentRepository.GetByIdAsync(id);
        existingExamContent.QuestionType = examcontent.QuestionType;
        existingExamContent.QuestionText = examcontent.QuestionText;
        existingExamContent.Options = examcontent.Options;
        existingExamContent.SelectedAnswer = examcontent.SelectedAnswer;
        existingExamContent.StarterCode = examcontent.StarterCode;
        existingExamContent.SampleInput = examcontent.SampleInput;
        existingExamContent.ExpectedOutput = examcontent.ExpectedOutput;
    }
}

```

Hình 3.31: Phương thức Update xử lý chức năng chỉnh sửa câu hỏi(1).

```

existingExamContent.Language = examcontent.Language;
existingExamContent.ExamID = examcontent.ExamID;
if (examcontent.QuestionType == "MultipleChoice")
{
    existingExamContent.CorrectAnswer = examcontent.SelectedAnswer;
}
else if (examcontent.QuestionType == "Code")
{
    existingExamContent.CorrectAnswer = null;
}
await _examcontentRepository.UpdateAsync(existingExamContent);
return RedirectToAction(nameof(Index));
}
await LoadExamsAsync();
return View(examcontent);
}

```

Hình 3.32: Phương thức Update xử lý chức năng chỉnh sửa câu hỏi(2).

- **Xóa câu hỏi:** Cho phép xóa câu hỏi sau khi xác nhận.

```

public async Task<IActionResult> Delete(int id)
{
    var examcontent = await _examcontentRepository.GetByIdAsync(id);
    if (examcontent == null)
    {
        return NotFound();
    }
    return View(examcontent);
}
[HttpPost, ActionName("Delete")]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    try
    {
        await _examcontentRepository.DeleteAsync(id);
        return RedirectToAction(nameof(Index));
    }
    catch (ArgumentNullException ex)
    {
        ModelState.AddModelError(string.Empty, ex.Message);
        return View();
    }
}

```

Hình 3.33: Phương thức Delete và DeleteConfirmed xử lý chức năng xoá câu hỏi.

- **Tìm kiếm bài tập:** Hành động Search thực hiện chức năng tìm kiếm bài tập theo tên.

```

//Tìm kiếm bài tập theo tên
0 references
public async Task<IActionResult> Search(string searchString)
{
    var exams = await _examRepository.GetAllAsync();
    if (!string.IsNullOrEmpty(searchString))
    {
        exams = exams.Where(e => e.ExamName.Contains(searchString, StringComparison.OrdinalIgnoreCase)).ToList();
    }
    return View("Index", exams);
}

```

Hình 3.34: Phương thức Search xử lý chức năng tìm kiếm bài tập theo tên.

Quản lý bài tập người dùng

- **Hiển thị lịch sử nộp bài:** Hiển thị danh sách thông tin về các bài tập mà người dùng đã nộp.

```

0 references
public async Task<IActionResult> Index()
{
    var userId = _userManager.GetUserId(User);
    if (string.IsNullOrEmpty(userId))
    {
        return RedirectToAction("Login", "Account");
    }

    var viewModel = new UserProfileViewModel
    {
        ExamSubmissions = await _examRepository.GetExamResultsByUserAsync(userId)
    };

    return View(viewModel);
}

```

Hình 3.35: Phương thức Index xử lý chức năng hiển thị lịch sử nộp bài.

Quản lý diễn đàn chat cộng đồng

- **Hiển thị diễn đàn chat cộng đồng:** Hiển thị tất cả nội dung trong diễn đàn bao gồm tin nhắn và thông tin người gửi lên.

```

public async Task<IActionResult> Index()
{
    var messages = await _messageRepository.GetMessagesAsync();
    return View(messages);
}

```

Hình 3.36: Phương thức Index xử lý hiển thị danh sách các tin nhắn lên diễn đàn.

- **Gửi tin nhắn lên diễn đàn:** Cho phép người dùng đã đăng nhập gửi tin nhắn lên diễn đàn.

```

[Authorize]
[HttpPost]
0 references
public async Task<IActionResult> SendMessage(string content)
{
    if (string.IsNullOrEmpty(content))
    {
        return RedirectToAction("Index");
    }

    var user = await _userManager.GetUserAsync(User);
    var message = new Message
    {
        UserID = user.Id,
        Content = content,
        Timestamp = DateTime.Now
    };

    await _messageRepository.AddMessageAsync(message);
    return RedirectToAction("Index");
}

```

Hình 3.37: Phương thức SendMessage xử lý chức năng gửi tin nhắn.

3.3. Kết quả thực nghiệm chương trình

- Giao diện đăng ký

The screenshot shows a web browser window with the URL `localhost:5041/identity/Account/Register?returnUrl=%2F`. The page has a dark blue header with the website name "WebsiteHotrohoctap" and navigation links "Trang chủ", "Khoa Học", "Luyện Tập", "Diễn Đàn", and "Đăng nhập". The main content area is a white card titled "Đăng ký tài khoản mới". It contains three input fields: "Địa chỉ Email", "Mật khẩu", and "Xác nhận mật khẩu". Below these is a large blue "Đăng ký" button. At the bottom of the card, there is a link "Đã có tài khoản? [Đăng nhập](#)".

Hình 3.38: Hình ảnh giao diện đăng ký.

- Giao diện đăng nhập

The screenshot shows a web browser window with the URL `localhost:5041/identity/Account/Login`. The page has a dark blue header with the website name "WebsiteHotrohoctap" and navigation links "Trang chủ", "Khoa Học", "Luyện Tập", "Diễn Đàn", and "Đăng nhập". The main content area is a white card titled "Đăng nhập". It contains two input fields: "Email" and "Mật khẩu". Below the fields is a checkbox labeled "Ghi nhớ đăng nhập". Below the checkbox is a large blue "Đăng nhập" button. At the bottom of the card, there are two links: "Quên mật khẩu?" and "Đăng ký tài khoản mới".

Hình 3.39: Hình ảnh giao diện đăng nhập.

- Quản lý người dùng

Email	Vai trò hiện tại	Thay đổi vai trò
customer2@gmail.com	Customer	Customer Cập nhật
customer1@gmail.com	Customer	Customer Cập nhật
admin@gmail.com	Admin	Customer Cập nhật

Hình 3.40: Hình ảnh giao diện quản lý người dùng.

- Danh sách bài học

LessonName	LessonDescription	Thuộc khóa học	Thao tác
Mảng (Array)	Giới thiệu về mảng, cách sử dụng và các thao tác cơ bản.	Cơ bản	Sửa Chi tiết Xóa
Ngắn xếp & Hàng đợi (Stack & Queue)	Tìm hiểu về cấu trúc dữ liệu ngắn xếp, hàng đợi và ứng dụng của nó.	Cơ bản	Sửa Chi tiết Xóa
Danh sách liên kết (Linked List)	Giới thiệu danh sách liên kết và các loại danh sách.	Nâng cao	Sửa Chi tiết Xóa
Cây (Tree)	Khái niệm cây, cây nhị phân và các thuật toán liên quan.	Nâng cao	Sửa Chi tiết Xóa
Đồ thị (Graph)	Tổng quan về đồ thị, biểu diễn và thuật toán duyệt đồ thị.	Nâng cao	Sửa Chi tiết Xóa

Hình 3.41: Hình ảnh giao diện danh sách bài học.

- Thêm bài học mới

Thêm bài học

Tên bài học
Nhập tên bài học

Mô tả
Nhập mô tả bài học

Thuộc khóa học
-- Chọn khóa học --

[Quay lại danh sách bài học](#)

Hình 3.42: Hình ảnh giao diện thêm bài học.

- Sửa bài học

Cập nhật bài học

Tên bài học
Mảng (Array)

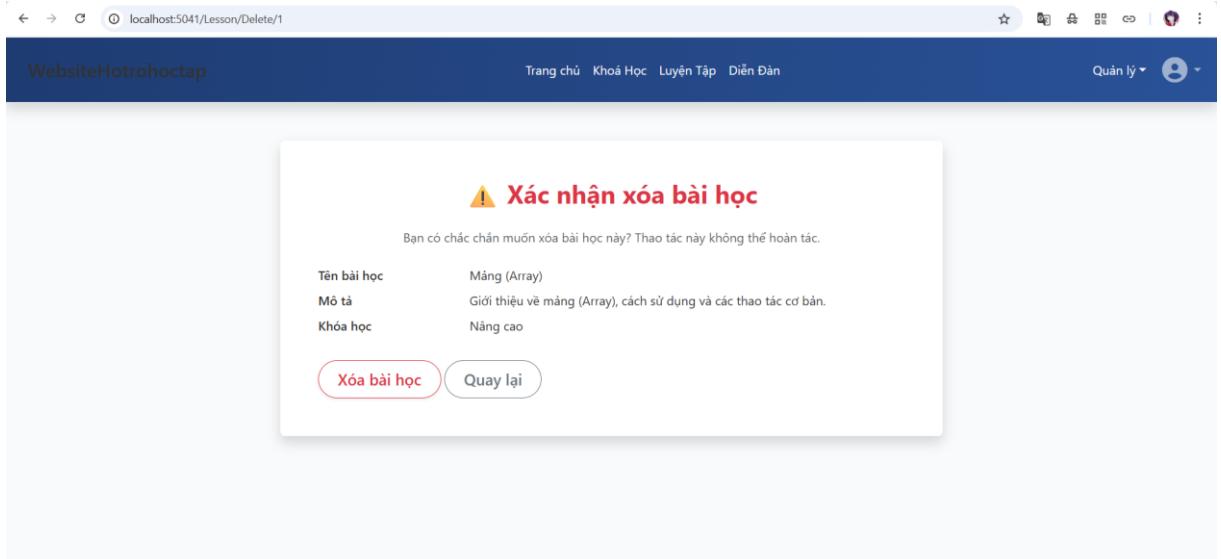
Mô tả bài học
Giới thiệu về mảng (Array), cách sử dụng và các thao tác cơ bản.

Khóa học
Nâng cao

[Quay lại danh sách bài học](#)

Hình 3.43: Hình ảnh giao diện sửa bài học.

- Xóa bài học



Hình 3.44: Hình ảnh giao diện xóa bài học.

- Quản lý danh sách nội dung bài học

Loại nội dung	Dữ liệu	Bài học	Thao tác
text	Khái niệm: Mảng là tập hợp các phần tử có cùng kiểu dữ liệu được lưu trữ liên tiếp trong bộ nhớ. Mỗi phần tử có thể được truy cập bằng chỉ số.	Mảng (Array)	Sửa Chi tiết Xóa
video		Mảng (Array)	Sửa Chi tiết Xóa
text	Khái niệm: Cây là một cấu trúc dữ liệu dạng phân cấp. Mỗi cây có một nút gốc (root), từ đó các nút khác được liên kết theo cấu trúc cha-con.	Cây (Tree)	Sửa Chi tiết Xóa

Hình 3.45: Hình ảnh giao diện quản lý danh sách nội dung bài học.

- Lọc nội dung theo bài học

localhost:5041/LessonContent?lessonId=4

WebsiteHotrohoctap		Trang chủ	Khoa Học	Luyện Tập	Diễn Đàn	Quản lý
Danh sách nội dung bài học						
+ Thêm nội dung						
<input type="button" value="Cây (Tree)"/> <input type="button" value="Lọc nội dung"/>						
Loại nội dung	Dữ liệu	Bài học	Thao tác			
text	Khái niệm: Cây là một cấu trúc dữ liệu dạng phân cấp. Mỗi cây có một nút gốc (root), từ đó các nút khác được liên kết theo cấu trúc cha-con.	Cây (Tree)	Sửa Chi tiết Xóa			
video		Cây (Tree)	Sửa Chi tiết Xóa			

Hình 3.46: Hình ảnh giao diện quản lý danh sách nội dung lọc theo bài học.

- Thêm nội dung bài học dạng văn bản

WebsiteHotrohoctap

Trang chủ Khoa Học Luyện Tập Diễn Đàn

Quản lý

Thêm Nội Dung Bài Học

Loại nội dung	Văn bản
Nội dung văn bản	<input type="text"/>
Chọn bài học	Mảng (Array)
Tạo mới	
← Quay lại danh sách	

Hình 3.47: Hình ảnh giao diện thêm nội dung bài học dạng văn bản.

- Thêm nội dung bài học dạng hình ảnh và video

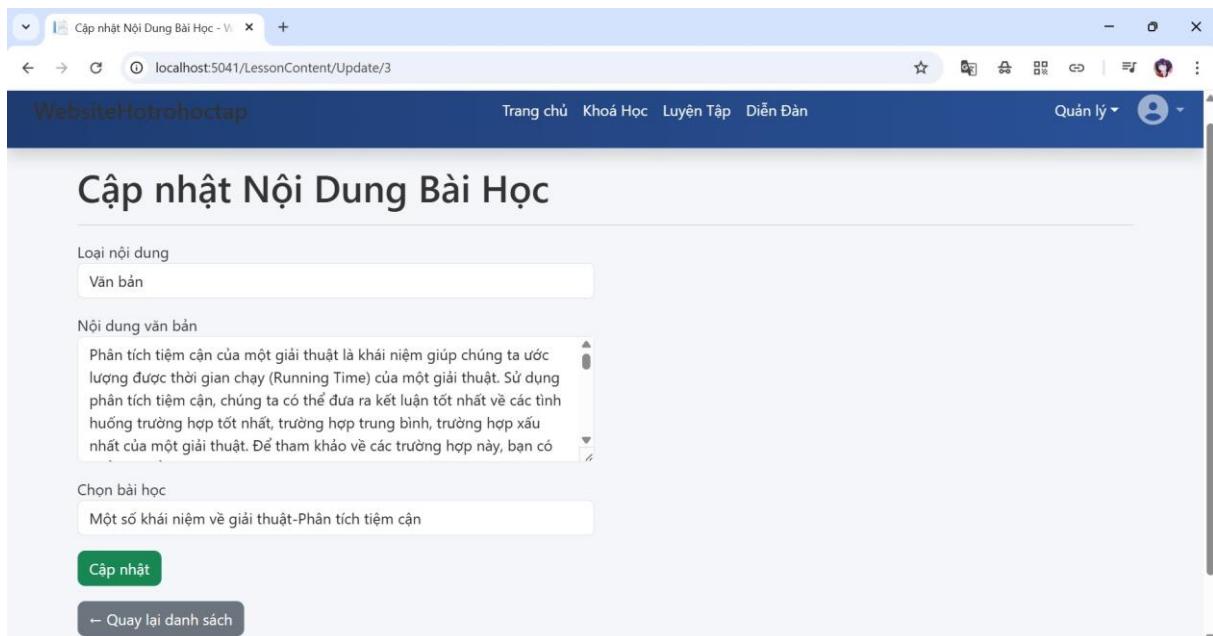
The screenshot shows a web browser window with the URL <http://localhost:5041/LessonContent/Create>. The page title is "WebsiteHotrohoctap". The main content area is titled "Thêm Nội Dung Bài Học". There is a dropdown menu "Loại nội dung" set to "Hình ảnh". A file input field "Chọn ảnh" contains "Choose File image.png". Below it is a dropdown menu "Chọn bài học" set to "Mảng (Array)". At the bottom are two buttons: a green "Tạo mới" button and a grey "← Quay lại danh sách" button.

Hình 3.48: Hình ảnh giao diện thêm nội dung bài học dạng hình ảnh.

The screenshot shows a web browser window with the URL <http://localhost:5041/LessonContent/Create>. The page title is "WebsiteHotrohoctap". The main content area is titled "Thêm Nội Dung Bài Học". There is a dropdown menu "Loại nội dung" set to "Video". A file input field "Chọn video hoặc nhập link" contains "Choose File No file chosen" and a text input field with the URL "https://youtube.com/...". Below it is a dropdown menu "Chọn bài học" set to "Mảng (Array)". At the bottom are two buttons: a green "Tạo mới" button and a grey "← Quay lại danh sách" button.

Hình 3.49: Hình ảnh giao diện thêm nội dung bài học dạng video.

- Cập nhật nội dung bài học:

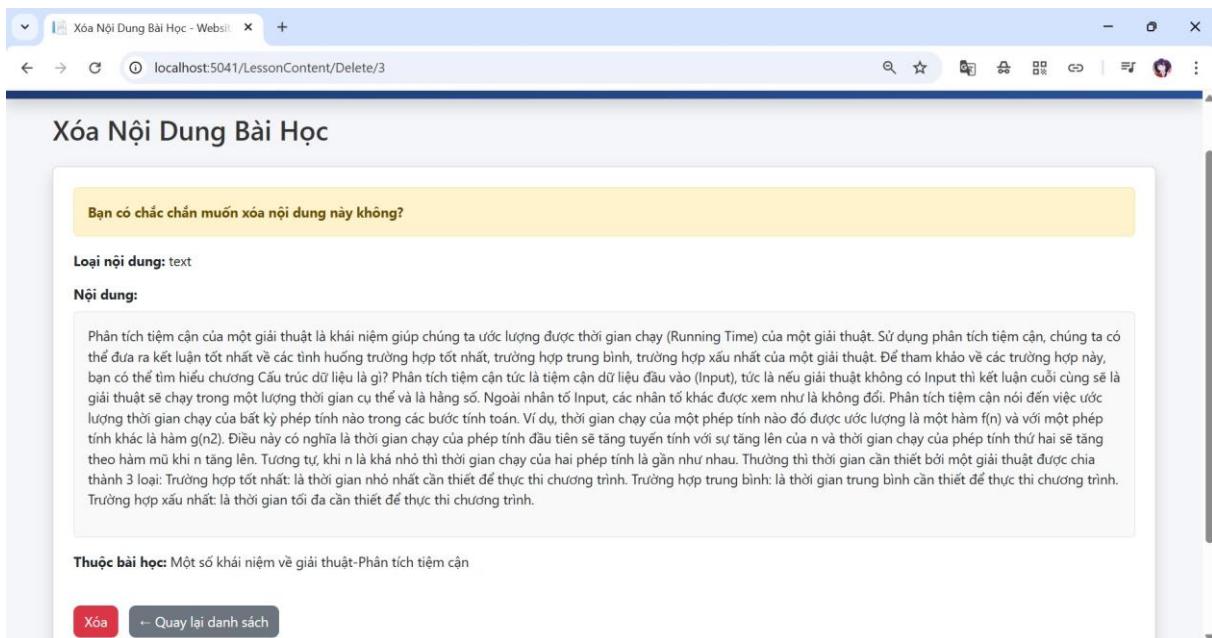


Hình 3.50: Hình ảnh giao diện cập nhật nội dung bài học.

- Xem chi tiết nội dung bài học:

Hình 3.51: Hình ảnh giao diện chi tiết nội dung bài học.

- Xóa nội dung bài học

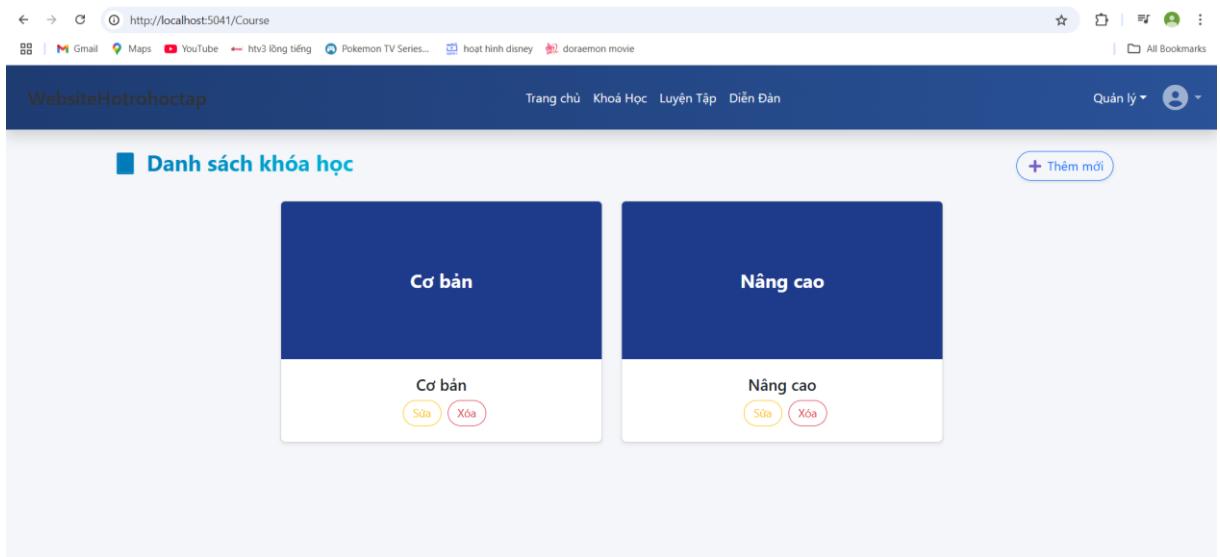


Hình 3.52: Hình ảnh giao diện xóa nội dung bài học.

- Giao diện bài học

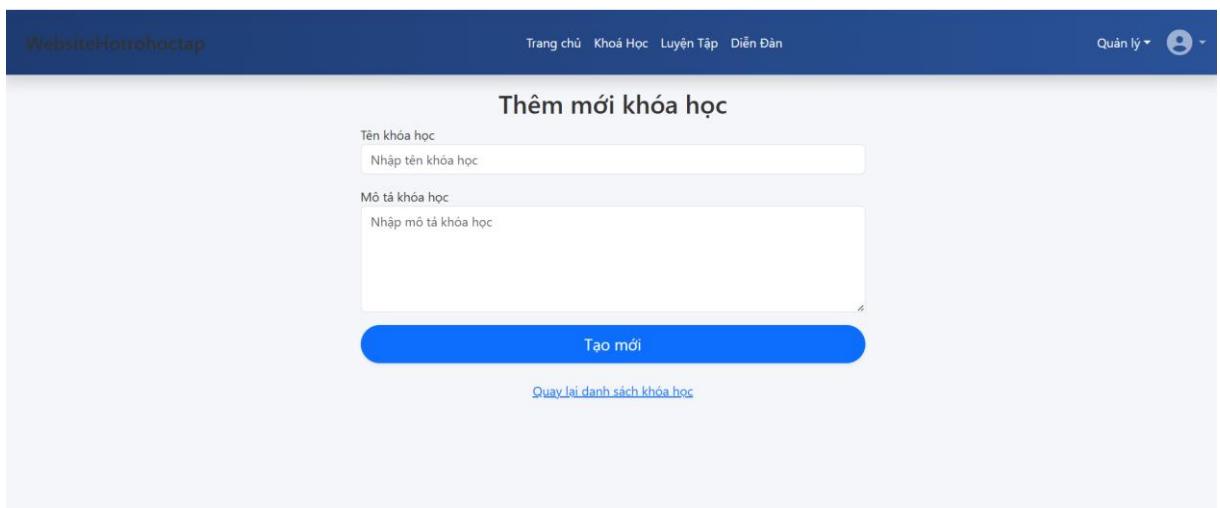
Hình 3.53: Hình ảnh giao diện bài học.

- Danh sách khóa học



Hình 3.54: Hình ảnh giao diện danh sách khóa học.

- Tạo khóa học mới



Hình 3.55: Hình ảnh giao diện thêm khóa học mới.

- Chính sửa khóa học

Cập nhật khóa học

CourseName
Cơ bản

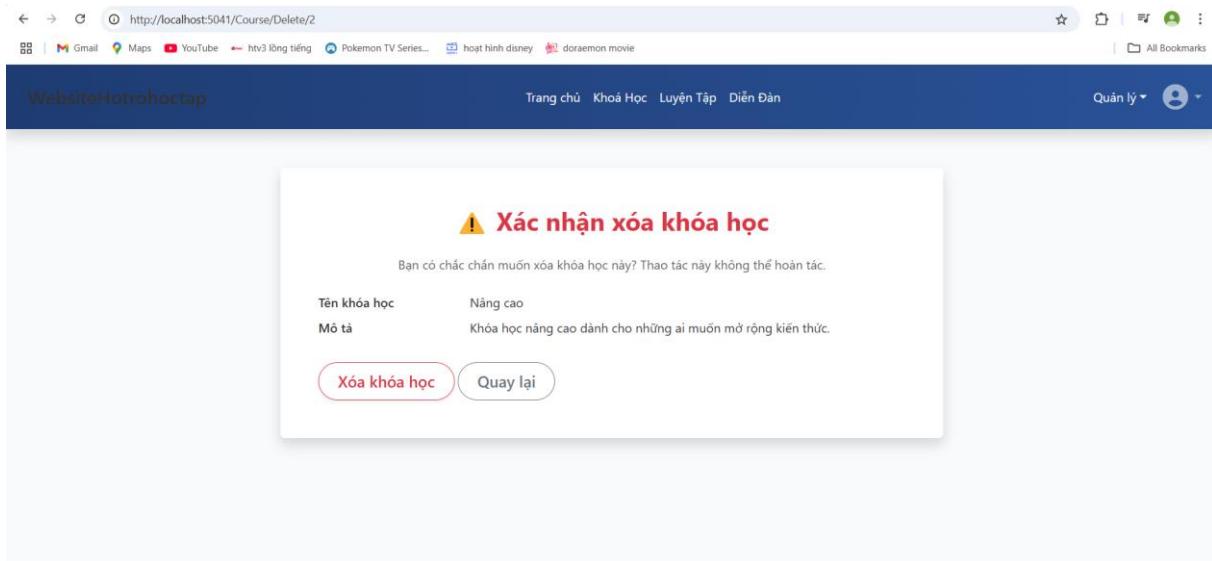
Description
Khóa học cơ bản về các kiến thức nền tảng.

Lưu thay đổi

[Trở lại danh sách khóa học](#)

Hình 3.56: Hình ảnh giao diện chỉnh sửa khóa học.

- Xóa khóa học



Hình 3.57: Hình ảnh giao diện xóa khóa học.

- Quản lý danh sách các bài tập

Tên bài tập	Mô tả	Tổng điểm	Thuộc bài học	Thao tác
test1	test1	100	Mảng (Array)	<button>Sửa</button> <button>Câu hỏi</button> <button>Xóa</button>
test2	test2	100	Stack & Queue	<button>Sửa</button> <button>Câu hỏi</button> <button>Xóa</button>

Hình 3.58: Hình ảnh giao diện quản lý danh sách các bài tập.

- Cập nhật thông tin bài tập

Cập nhật thông tin bài tập

Tên bài
test1

Mô tả
test1

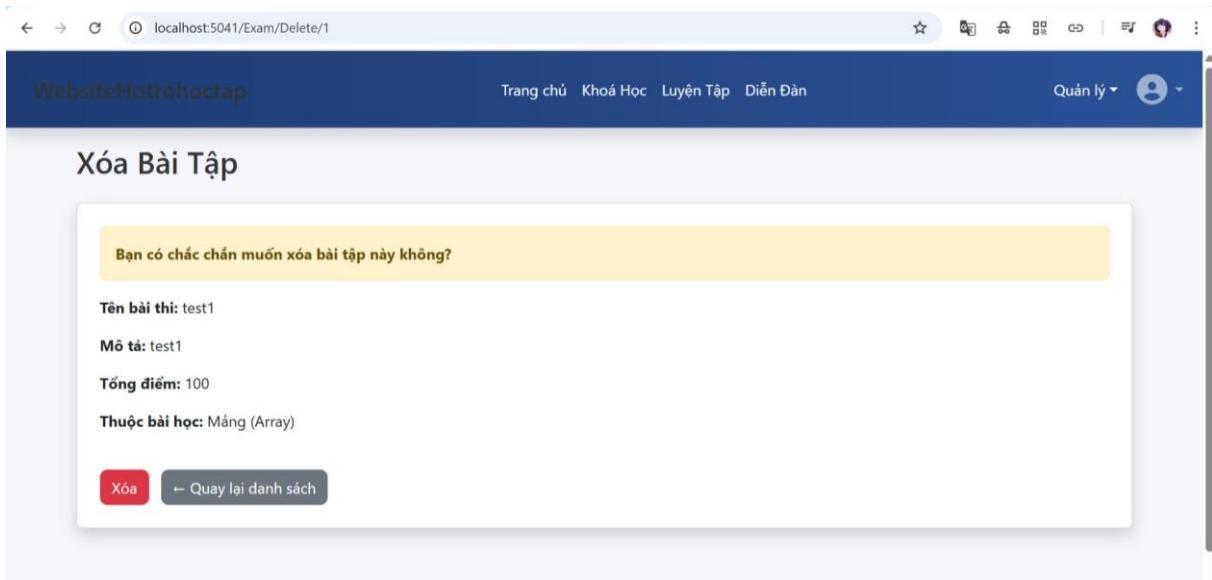
Tổng điểm
100

Thuộc bài học
Mảng (Array)

[Quay lại danh sách bài tập](#)

Hình 3.59: Hình ảnh giao diện cập nhật thông tin bài tập.

-Xóa bài tập



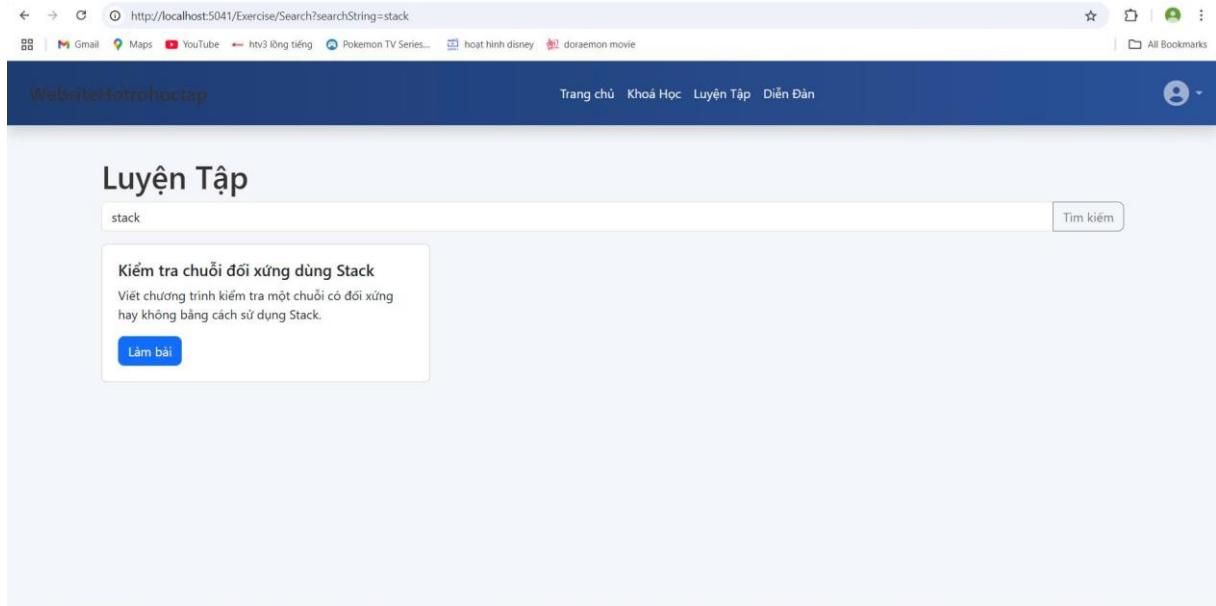
Hình 3.60: Hình ảnh giao diện xóa bài tập.

- Danh sách các bài tập

Tên bài tập	Mô tả	Độ khó	Thao tác
Tính tổng mảng	Viết chương trình tính tổng tất cả các phần tử trong một mảng số nguyên.	Đơn giản	Làm bài Chính sửa
Tìm số lớn nhất trong mảng	Viết chương trình tìm số lớn nhất trong một mảng số nguyên.	Đơn giản	Làm bài Chính sửa
Kiểm tra chuỗi đối xứng dùng Stack	Viết chương trình kiểm tra một chuỗi có đối xứng hay không bằng cách sử dụng Stack.	Trung bình	Làm bài Chính sửa
Quản lý hàng đợi	Viết chương trình mô phỏng hàng đợi (Queue): thêm phần tử vào hàng đợi và lấy phần tử ra.	Trung bình	Làm bài Chính sửa
Đảo ngược danh sách liên kết	Viết chương trình đảo ngược một danh sách liên kết đơn.	Trung bình	Làm bài Chính sửa
Chèn phần tử vào danh sách liên kết	Viết chương trình chèn một phần tử vào vị trí xác định trong danh sách liên kết đơn.	Trung bình	Làm bài Chính sửa
Đếm số node trong cây nhị phân	Viết chương trình đếm số node trong một cây nhị phân.	Trung bình	Làm bài Chính sửa
Duyệt cây theo thứ tự trước (Preorder)	Viết chương trình duyệt cây nhị phân theo thứ tự trước (Preorder).	Trung bình	Làm bài Chính sửa
Duyệt đồ thị bằng DFS	Viết chương trình duyệt đồ thị bằng thuật toán DFS.	Trung bình	Làm bài Chính sửa

Hình 3.61: Hình ảnh giao diện danh sách các bài tập.

- Tìm kiếm bài tập theo tên



Hình 3.62: Hình ảnh giao diện tìm kiếm bài tập theo tên.

- Danh sách câu hỏi

Danh sách câu hỏi				
Loại câu hỏi	Nội dung	Đáp án/Code mẫu	Thuộc bài kiểm tra	Thao tác
Trắc nghiệm	Câu hỏi 1	A	test1	<button>Sửa</button> <button>Chi tiết</button> <button>Xóa</button>
Trắc nghiệm	Câu hỏi 2	B	test1	<button>Sửa</button> <button>Chi tiết</button> <button>Xóa</button>
Viết mã	Cho xâu s được nhập từ bàn phím. Bạn hãy viết chương trình chuyển các ký tự '3' trong xâu s sang ký tự 'e' và hiển thị xâu s ra màn hình. Ví dụ nếu bạn nhập s =	#include<iostream> using namespace std; int main() { string s; cin >> s; for (int i = 0; i < s.size(); i++) { if (s[i] == '3') { s[i] = 'e'; } } cout << s; }	test1	<button>Sửa</button> <button>Chi tiết</button> <button>Xóa</button>

Hình 3.63: Hình ảnh giao diện danh sách câu hỏi.

- Thêm câu hỏi dạng trắc nghiệm

localhost:5041/ExamContent/Create

WebsiteHotrohoclap Trang chủ Khoa Học Luyện Tập Diễn Đàn Quản lý

Thêm câu hỏi

Loại câu hỏi
Trắc nghiệm

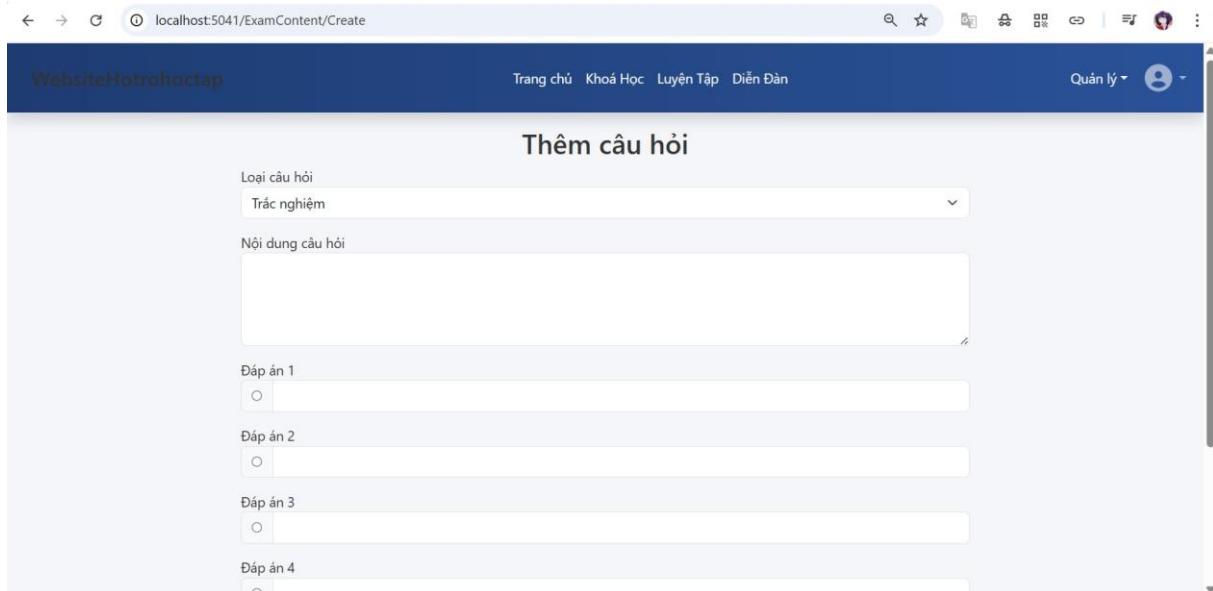
Nội dung câu hỏi

Đáp án 1

Đáp án 2

Đáp án 3

Đáp án 4



Hình 3.64: Hình ảnh giao diện thêm câu hỏi dạng trắc nghiệm(1).

localhost:5041/ExamContent/Create

Nội dung câu hỏi

Đáp án 1

Đáp án 2

Đáp án 3

Đáp án 4

Thuộc bài kiểm tra
-- Chọn bài kiểm tra --

Tạo câu hỏi



Hình 3.65: Hình ảnh giao diện thêm câu hỏi dạng trắc nghiệm(2).

- Thêm câu hỏi dạng viết mã lệnh

The screenshot shows a web application interface for adding a question. At the top, there's a header bar with the website name 'WebsiteHotrohoclap', navigation links 'Trang chủ', 'Khoa Học', 'Luyện Tập', 'Diễn Đàn', and a 'Quản lý' dropdown menu. Below the header, the main title is 'Thêm câu hỏi'. The form fields include:

- 'Loại câu hỏi': A dropdown menu set to 'Viết mã lệnh'.
- 'Nội dung câu hỏi': A large text area for entering the question content.
- 'Ngôn ngữ lập trình': A dropdown menu set to 'Java'.
- 'Code mẫu (nếu có)': A text area for providing sample code.

Hình 3.66: Hình ảnh giao diện thêm câu hỏi dạng viết mã lệnh(1).

This screenshot shows the same 'Thêm câu hỏi' form, but with more fields visible. The additional fields are:

- 'Input mẫu (nếu có)': A text area for sample input.
- 'Output mong đợi': A text area for expected output.
- 'Thuộc bài kiểm tra': A dropdown menu set to '-- Chọn bài kiểm tra --'.
- A blue 'Tạo câu hỏi' (Create Question) button at the bottom.

Hình 3.67: Hình ảnh giao diện thêm câu hỏi dạng viết mã lệnh(2).

- Cập nhật (chỉnh sửa) câu hỏi dạng trắc nghiệm

The screenshot shows a web browser window with the URL `localhost:5041/ExamContent/Update/2`. The page title is "WebsiteHutrohoctap". The top navigation bar includes links for "Trang chủ", "Khoa Học", "Luyện Tập", "Diễn Đàn", and "Quản lý" with a user icon. The main content area has a title "Cập nhật câu hỏi" with a pencil icon. It contains fields for "Loại câu hỏi" (selected as "Trắc nghiệm"), "Nội dung câu hỏi" (text "Câu hỏi 1"), and three answer options: "Đáp án 1" (radio button A selected), "Đáp án 2" (radio button B), and "Đáp án 3" (radio button C). A dropdown menu for "Thuộc bài kiểm tra" shows "test1". At the bottom are buttons for "Lưu thay đổi" (Save changes) and "Quay lại" (Back).

Hình 3.68: Hình ảnh giao diện cập nhật câu hỏi dạng trắc nghiệm(1).

This screenshot shows the same web application interface as Figure 3.68, but with four answer options instead of three. The "Đáp án 1" field now has radio button A selected. The "Đáp án 2" field has radio button B. The "Đáp án 3" field has radio button C. The "Đáp án 4" field has radio button D. The rest of the interface remains the same, including the dropdown for "Thuộc bài kiểm tra" set to "test1" and the bottom buttons for saving changes.

Hình 3.69: Hình ảnh giao diện cập nhật câu hỏi dạng trắc nghiệm(2).

- Xem chi tiết câu hỏi

WebsiteHotrohoctap

Trang chủ Khoa Học Luyện Tập Diễn Đàn

Quản lý

Chi tiết câu hỏi

Loại câu hỏi: Trắc nghiệm
Nội dung câu hỏi: Câu hỏi 1
Các lựa chọn:

- 1. A **Đáp án đúng**
- 2. B
- 3. C
- 4. D

Thuộc bài kiểm tra: test1

Sửa Quay lại danh sách

Hình 3. 70: Hình ảnh giao diện chi tiết câu hỏi.

- Xóa câu hỏi

⚠ Xác nhận xóa câu hỏi

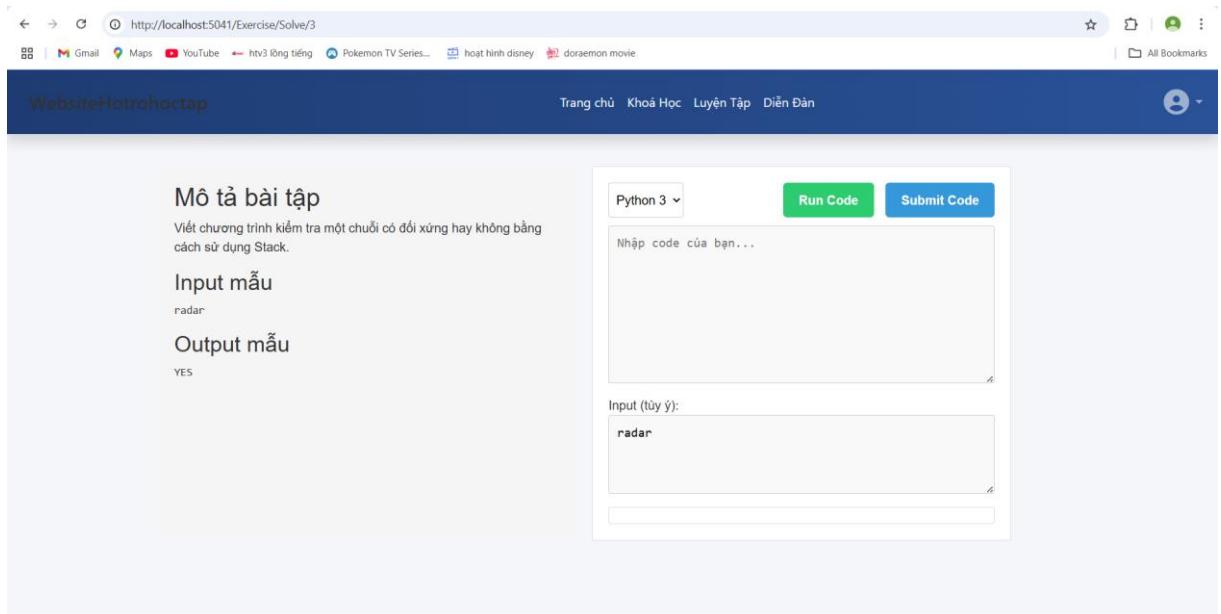
Bạn có chắc chắn muốn xóa câu hỏi này? Thao tác này không thể hoàn tác.

Loại câu hỏi: Trắc nghiệm
Nội dung câu hỏi: Câu hỏi 1
Thuộc bài kiểm tra: test1

Xóa câu hỏi Quay lại

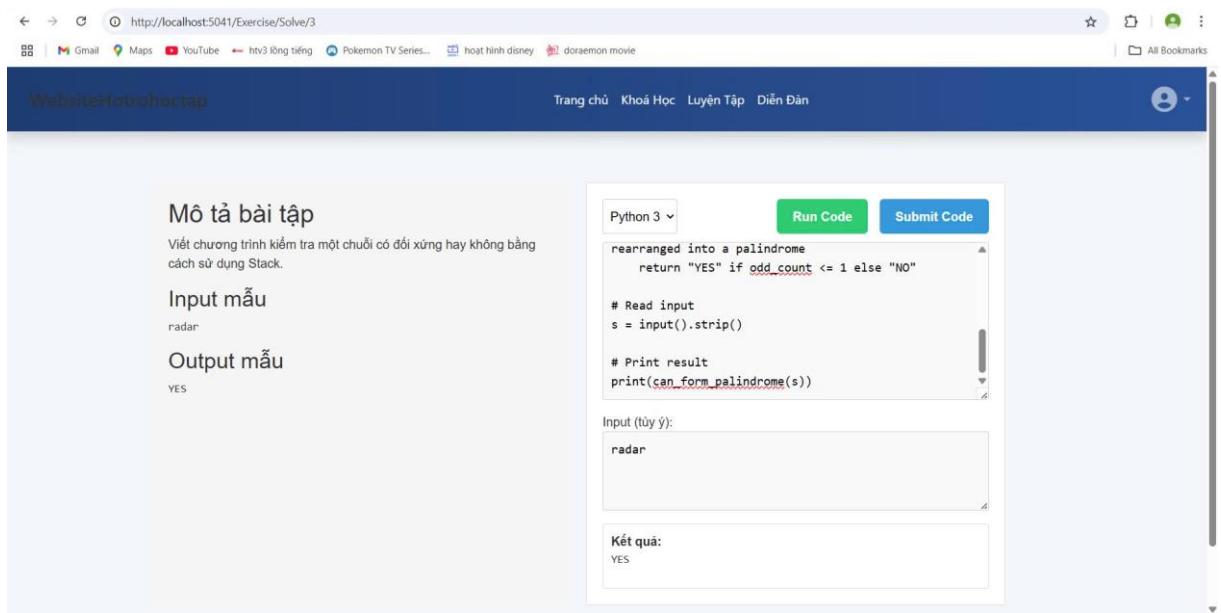
Hình 3.71: Hình ảnh giao diện xóa câu hỏi.

- Giao diện chạy code



Hình 3.72: Hình ảnh giao diện chạy code.

- Giao diện khi code chạy thành công



Hình 3.73: Hình ảnh giao diện chạy code thành công.

- Giao diện thông báo lỗi khi lỗi code

Mô tả bài tập

Viết chương trình kiểm tra một chuỗi có đối xứng hay không bằng cách sử dụng Stack.

Input mẫu

```
radar
```

Output mẫu

```
YES
```

C++17 Run Code Submit Code

```
s.push(c);
}

string reversedStr = "";
while (!s.empty()) {
    reversedStr += s.top();
    s.pop();
}
```

Input (tùy ý):

```
radar
```

Kết quả:

```
jdoode.cpp: In function 'bool isPalindrome(std::string)':  
jdoode.cpp:15:16: error: expected ';' before ')' token  
    15 |         s.pop()  
        |         ^  
        |         ;  
    16 |     )  
        |     ~
```

Hình 3.74: Hình ảnh giao diện thông báo lỗi khi lỗi code.

- Giao diện nộp bài thành công

WebsiteHotrohoctap Trang chủ Khoa Học Luyện Tập Diễn Đàn

Mô tả bài tập

Viết chương trình đảo ngược một danh sách liên kết đơn.

Input mẫu

```
1 2 3 4 5
```

Output mẫu

```
5 4 3 2 1
```

C++ Run Code Submit Code

```
#include <iostream>
using namespace std;
struct ListNode {
    int val;
    ListNode* next;
    ListNode(int x) : val(x),
    next(NULL) {}
};
```

Input (tùy ý):

```
1 2 3 4 5
```

Kết quả nộp bài:

Trang thái: Accepted
Số test case vượt qua: 1/1
Điểm: 100

Hình 3.75: Hình ảnh giao diện nộp bài thành công.

- Lịch sử nộp bài

Tên bài tập	Tổng điểm	Ngày nộp bài	Trạng thái	Thao tác
3	100	5/20/2025 12:10 AM	Accepted	View Code
5	100	5/20/2025 12:04 AM	Accepted	View Code
2	0	5/19/2025 11:00 PM	Wrong Answer	View Code
2	0	5/19/2025 11:00 PM	Wrong Answer	View Code

Hình 3.76: Hình ảnh giao diện lịch sử nộp bài.

- Xem code đã nộp

```

Code
# Nhập chuỗi
s = input()

# Tạo một Stack
stack = []

# Đẩy tất cả ký tự vào Stack
for char in s:
    stack.append(char)

# Tạo chuỗi đảo ngược bằng cách lấy ra từ Stack
reversed_s = ""
while stack:
    reversed_s += stack.pop()

# So sánh chuỗi ban đầu với chuỗi đảo ngược
if s == reversed_s:
    print("YES")
else:
    print("NO")

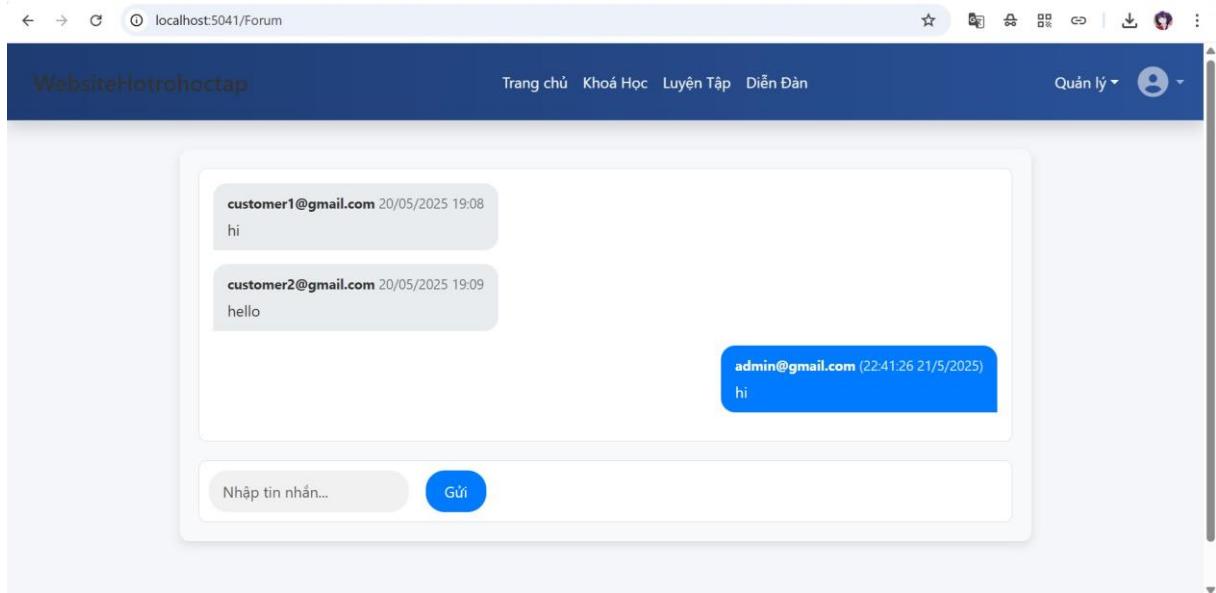
Input
radar

Output
YES

```

Hình 3.77: Hình ảnh giao diện xem code đã nộp.

- Diễn đàn chat cộng đồng



Hình 3.78: Hình ảnh giao diện diễn đàn chat cộng đồng.

CHƯƠNG 4: KẾT LUẬN

4.1. Kết quả đạt được

Sau quá trình tìm hiểu, thiết kế và xây dựng hệ thống, nhóm đã hoàn thiện website hỗ trợ học tập cho sinh viên trong lĩnh vực Cấu trúc dữ liệu và Giải thuật. Hệ thống được xây dựng với các chức năng chính đáp ứng nhu cầu học tập cơ bản như: xem bài giảng, thực hành bài tập, làm bài kiểm tra, và trao đổi qua diễn đàn học tập. Cụ thể:

- Chức năng quản lý khóa học: Cho phép thêm, sửa, xóa và xem chi tiết các khóa học hiện có.
- Chức năng quản lý bài học: Hỗ trợ xây dựng hệ thống bài học cho từng khóa học, giúp phân chia nội dung học một cách logic, dễ theo dõi.
- Chức năng quản lý nội dung bài học: Cung cấp nội dung bài học dưới nhiều định dạng như văn bản, hình ảnh, video; phù hợp với nhu cầu minh họa trực quan trong giảng dạy.
- Chức năng quản lý bài tập: Tạo và quản lý các bài tập theo từng bài học, giúp sinh viên luyện tập kiến thức đã học.
- Chức năng quản lý nội dung bài tập: Cho phép xây dựng bộ câu hỏi cho từng bài tập, bao gồm cả câu hỏi trắc nghiệm và viết mã lệnh.
- Chức năng diễn đàn học tập: Hỗ trợ sinh viên trao đổi, thảo luận, chia sẻ kinh nghiệm học tập, từ đó tạo nên môi trường học tập cộng đồng tích cực.

Toàn bộ chức năng được xây dựng với tiêu chí thân thiện, dễ sử dụng, giúp người học – đặc biệt là người mới bắt đầu có thể tiếp cận và sử dụng hiệu quả hơn trong quá trình học tập.

4.2. Hạn chế của hệ thống

Dù hệ thống đã cơ bản đáp ứng được mục tiêu đề ra, nhưng vẫn còn một số hạn chế nhất định:

- Chưa tích hợp hệ thống chấm bài lệnh lập trình phức tạp theo thời gian thực (ví dụ như hệ thống judge online).
- Tính năng giao tiếp trong diễn đàn còn đơn giản.

- Giao diện còn đơn giản, chưa tối ưu cho thiết bị di động.
- Chưa có chức năng theo dõi tiến độ học tập của từng người dùng.

4.3. Hướng phát triển trong tương lai

Để nâng cao hiệu quả của hệ thống, nhóm đề xuất một số hướng phát triển tiếp theo như:

- Tích hợp hệ thống chấm mã tự động nâng cao, hỗ trợ nhiều ngôn ngữ lập trình.
- Phát triển thêm chức năng đánh giá tiến độ học tập, thống kê kết quả và đề xuất lộ trình học cá nhân hóa.
- Tối ưu hóa giao diện người dùng, bổ sung phiên bản cho thiết bị di động.
- Phát triển tính năng học theo nhóm, tạo phòng học hoặc nhóm thảo luận nhỏ.

TÀI LIỆU THAM KHẢO

[1] Microsoft, “ASP.NET Core MVC overview,” Microsoft Learn.

<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview>

[2] Microsoft, “Entity Framework Core,” Microsoft Learn.

<https://learn.microsoft.com/en-us/ef/core/>

[3] Microsoft, “Razor Pages vs MVC,” Microsoft Learn.

<https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-8.0>

[4] W3Schools, “Bootstrap 5 Tutorial,” W3Schools.

<https://www.w3schools.com/bootstrap5/>