

ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

HỆ THỐNG QUẢN LÝ CA THI VÀ THI TRẮC NGHIỆM TRỰC TUYẾN

Ngành: **Công Nghệ Thông Tin**

Chuyên ngành: **Công Nghệ Phần Mềm**

Giảng viên hướng dẫn : ThS.Nguyễn Huy Cường

Sinh viên thực hiện : Cao Hiến Đạt

MSSV: 2180608276 Lớp: 21DTHD2

TP. Hồ Chí Minh, ngày 25 tháng 2 năm 2025

MỤC LỤC

| | |
|--|----|
| CHƯƠNG 1: TỔNG QUAN | 2 |
| 1.1 Lý do thực hiện đề tài | 2 |
| 1.2 Mục tiêu nghiên cứu | 3 |
| 1.2.1 Mục tiêu trực tiếp | 3 |
| 1.2.2 Mục tiêu gián tiếp | 3 |
| 1.3 Đối tượng nghiên cứu | 4 |
| 1.4 Nhiệm vụ nghiên cứu | 4 |
| 1.4.1 Nghiên cứu cơ sở lý luận | 5 |
| 1.4.2 Nghiên cứu thực trạng và yêu cầu hệ thống | 5 |
| 1.4.3 Xây dựng và triển khai hệ thống | 5 |
| 1.4.4 Kiểm thử và đánh giá hệ thống | 5 |
| 1.4.5 Đề xuất và hoàn thiện | 5 |
| 1.5 Phạm vi nghiên cứu | 6 |
| 1.5.1 Phạm vi nội dung | 6 |
| 1.5.2 Phạm vi không gian | 6 |
| 1.5.3 Phạm vi thời gian | 6 |
| 1.6 Phương pháp nghiên cứu | 6 |
| 1.6.1 Nhóm phương pháp nghiên cứu lý luận | 6 |
| 1.6.2 Nhóm phương pháp nghiên cứu thực tiễn | 7 |
| 1.6.3 Nhóm phương pháp xử lý số liệu | 7 |
| CHƯƠNG 2: CƠ SỞ LÝ LUẬN | 9 |
| 2.1 Các khái niệm nghiên cứu có liên quan | 9 |
| 2.2 Các đề tài, kết quả nghiên cứu có liên quan | 10 |
| 2.2.1. Nghiên cứu về hệ thống thi trắc nghiệm trực tuyến | 10 |
| 2.2.2. Các nghiên cứu về công nghệ liên quan | 10 |
| 2.3 Các hạn chế về mặt lý luận hiện tại | 10 |
| CHƯƠNG 3: KẾ HOẠCH VÀ DỰ KIẾN KẾT QUẢ | 12 |
| 3.1 Dự kiến kết quả | 12 |
| 3.2 Tiến độ thực hiện | 13 |
| 3.3 Kế hoạch thực hiện | 13 |
| TÀI LIỆU THAM KHẢO | 15 |

CHƯƠNG 1: TỔNG QUAN

1.1 Lý do thực hiện đề tài

Trong bối cảnh công nghệ thông tin ngày càng phát triển, nhu cầu về các hệ thống thi trắc nghiệm trực tuyến ngày càng gia tăng, đặc biệt trong lĩnh vực giáo dục và đào tạo. Hình thức thi trắc nghiệm trên máy tính tại phòng thi có giám thị giám sát giúp giảm thiểu sai sót trong khâu chấm điểm, tiết kiệm thời gian so với thi trên giấy, đồng thời nâng cao tính minh bạch và chính xác trong đánh giá kết quả. Tuy nhiên, nhiều hệ thống hiện tại vẫn gặp hạn chế về hiệu suất, khả năng mở rộng và tính bảo mật, đặc biệt khi phải phục vụ số lượng lớn thí sinh trong cùng một thời điểm. Bên cạnh đó, nhiều hệ thống thi trực tuyến hiện nay vẫn còn một số hạn chế như tốc độ tải đề chậm, dễ bị gian lận, chưa hỗ trợ đầy đủ các dạng câu hỏi nâng cao (câu hỏi nhóm, điền khuyết, âm thanh, LaTeX) hay chưa có cơ chế giám sát chặt chẽ trong quá trình thi. Điều này đặt ra nhu cầu cấp thiết về một hệ thống có khả năng xử lý tốt lượng lớn thí sinh đồng thời đảm bảo công bằng và bảo mật trong thi cử.

Về mặt cá nhân, tôi có niềm đam mê với công nghệ web, đặc biệt là các giải pháp tối ưu hiệu suất và bảo mật trong các hệ thống phân tán. Việc thực hiện đề tài này không chỉ giúp tôi vận dụng kiến thức đã học mà còn là cơ hội để nghiên cứu và áp dụng các công nghệ hiện đại như Blazor WebAssembly, microservices, Redis Cache, SignalR, RabbitMQ, JWT Authentication, và gRPC vào thực tế. Bên cạnh đó, dự án cũng là bước đệm quan trọng để tôi phát triển các hệ thống có quy mô lớn và độ tin cậy cao trong tương lai.

Từ những lý do trên, tôi quyết định lựa chọn đề tài “Hệ thống quản lý ca thi và thi trắc nghiệm trực tuyến sử dụng Blazor WebAssembly theo kiến trúc microservices” nhằm tạo ra một hệ thống thi cử hiệu quả, an toàn và đáp ứng tốt nhu cầu thực tế.

1.2 Mục tiêu nghiên cứu

1.2.1 Mục tiêu trực tiếp

- Xây dựng một hệ thống thi trắc nghiệm trực tuyến tại phòng máy có giám thị giám sát, đảm bảo hiệu suất cao, bảo mật tốt và dễ dàng quản lý.
- Ứng dụng Blazor WebAssembly kết hợp với kiến trúc microservices để tối ưu hóa tốc độ tải đề, xử lý bài thi và nâng cao trải nghiệm người dùng.
- Tích hợp các công nghệ JWT Authentication, Bcrypt, DTO để bảo vệ thông tin đăng nhập, ngăn chặn truy cập trái phép và đảm bảo an toàn dữ liệu.
- Sử dụng MSSQL với store procedures để truy vấn dữ liệu hiệu quả, tối ưu thời gian xử lý bài thi.
- Áp dụng Redis Cache để giảm tải truy vấn database, tăng tốc độ tải đề và chấm điểm.
- Xây dựng hệ thống giám sát quá trình thi theo thời gian thực bằng SignalR, giúp quản trị viên theo dõi, can thiệp kịp thời khi cần thiết.
- Thiết lập RabbitMQ để quản lý hàng đợi khi lưu bài làm định kỳ, tránh tình trạng nghẽn hệ thống khi có số lượng lớn thí sinh thi cùng lúc.
- Hỗ trợ các dạng câu hỏi nâng cao: câu hỏi nhóm, điền khuyết, âm thanh (số lần nghe giới hạn), LaTeX (công thức toán học).
- Tích hợp tính năng chống gian lận: cấm mở tab khác, tắt inspect trang, hiển thị điểm qua Canvas ảnh để tránh sinh viên chỉnh sửa điểm số.
- Hỗ trợ nhập sinh viên vào ca thi bằng Excel (EPPlus) giúp quản trị viên dễ dàng thêm nhiều sinh viên cùng lúc.
- Xử lý lỗi hệ thống bằng Middleware để hiển thị thông báo lỗi thân thiện thay vì lỗi API thô, đồng thời có cơ chế tự động reload khi xảy ra lỗi nặng.

1.2.2 Mục tiêu gián tiếp

- Góp phần nâng cao chất lượng thi cử bằng cách tạo ra một hệ thống thi trắc nghiệm tin cậy, minh bạch và dễ sử dụng.
- Cung cấp một giải pháp hiệu quả, tối ưu chi phí, giúp các tổ chức giáo dục dễ dàng triển khai thi trực tuyến mà không cần đầu tư hệ thống phần cứng phức tạp.

- Nâng cao tính bảo mật và giám sát, giảm thiểu rủi ro gian lận trong thi cử.
- Góp phần mở rộng hiểu biết về việc ứng dụng Blazor WebAssembly, microservices và các công nghệ tối ưu hiệu suất vào thực tiễn.
- Tạo tiền đề cho việc nghiên cứu và phát triển các hệ thống thi trực tuyến có quy mô lớn hơn trong tương lai.

1.3 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm:

- Mô hình thi trắc nghiệm trực tuyến tại phòng máy: Nghiên cứu cách tổ chức và vận hành hệ thống thi cử trên máy tính có giám thị giám sát.
- Kiến trúc microservices trong ứng dụng thi trực tuyến: Phân tích cách phân tách các dịch vụ, giao tiếp giữa chúng bằng Restful API và gRPC, đảm bảo tính mở rộng và hiệu suất.
- Công nghệ Blazor WebAssembly: Tìm hiểu ưu, nhược điểm của Blazor WebAssembly so với các công nghệ frontend khác trong việc xây dựng hệ thống thi trực tuyến.
- Các cơ chế tối ưu hiệu suất hệ thống:
 - Ứng dụng Redis Cache để giảm tải truy vấn database.
 - RabbitMQ để quản lý hàng đợi khi lưu bài làm.
 - SignalR để theo dõi trạng thái thí sinh theo thời gian thực.
- Các giải pháp bảo mật và chống gian lận trong thi trực tuyến:
 - JWT Authentication & Bcrypt để bảo vệ tài khoản người dùng.
 - Cơ chế kiểm soát hành vi thí sinh như chống mở tab khác, tắt inspect trang, sử dụng Canvas ảnh để hiển thị điểm.
- Hệ thống nhập liệu và quản lý kỳ thi: Cách tích hợp EPPlus để nhập sinh viên từ file Excel, hỗ trợ quản trị viên quản lý kỳ thi hiệu quả hơn.

1.4 Nhiệm vụ nghiên cứu

Để hoàn thành đề tài, các nhiệm vụ nghiên cứu cần thực hiện bao gồm:

1.4.1 Nghiên cứu cơ sở lý luận

- Tìm hiểu về các phương pháp thi trắc nghiệm truyền thống và các hệ thống thi trực tuyến hiện có.
- Phân tích ưu điểm và hạn chế của các công nghệ hiện nay trong việc xây dựng hệ thống thi trắc nghiệm.
- Nghiên cứu về kiến trúc microservices và cách áp dụng vào hệ thống Blazor WebAssembly.
- Tìm hiểu các giải pháp bảo mật, tối ưu hiệu suất và chống gian lận trong thi trực tuyến.

1.4.2 Nghiên cứu thực trạng và yêu cầu hệ thống

- Khảo sát các yêu cầu thực tế của một hệ thống thi trắc nghiệm trực tuyến tại phòng máy.
- Xác định các chức năng cần có cho hệ thống, bao gồm quản lý đề thi, sinh viên, giám thị, bài làm và giám sát thi.
- Đề xuất mô hình tổng thể của hệ thống, bao gồm các thành phần chính, cách giao tiếp giữa client và server, cũng như cơ chế xử lý dữ liệu.

1.4.3 Xây dựng và triển khai hệ thống

- Thiết kế giao diện và trải nghiệm người dùng.
- Xây dựng backend theo mô hình microservices.
- Tích hợp các cơ chế bảo mật.
- Phát triển tính năng chống gian lận.
- Xây dựng cơ chế nhập liệu nhanh bằng các file Excel.

1.4.4 Kiểm thử và đánh giá hệ thống

- Kiểm tra hiệu suất hệ thống với số lượng sinh viên lớn.
- Kiểm thử các tính năng bảo mật và chống gian lận.
- Đánh giá khả năng mở rộng và tính ổn định của hệ thống.

1.4.5 Đề xuất và hoàn thiện

- Đưa ra các đề xuất để cải tiến hệ thống dựa trên kết quả thử nghiệm.
- Hoàn thiện tài liệu và hướng dẫn sử dụng hệ thống.

- Đề xuất hướng phát triển trong tương lai, như hỗ trợ thi trên nhiều thiết bị hoặc tích hợp AI vào giám sát thi.

1.5 Phạm vi nghiên cứu

1.5.1 Phạm vi nội dung

- Đề tài tập trung nghiên cứu và phát triển hệ thống thi trắc nghiệm trực tuyến hoạt động tại phòng máy có giám thị giám sát, không áp dụng cho hình thức thi từ xa.
- Hệ thống được xây dựng theo mô hình microservices, sử dụng Blazor WebAssembly làm frontend và MSSQL, Redis Cache, RabbitMQ, SignalR để tối ưu hiệu suất.
- Hỗ trợ các dạng câu hỏi nâng cao: câu hỏi nhóm, điền khuyết, câu hỏi âm thanh, LaTeX công thức toán học.
- Tập trung vào các cơ chế chống gian lận, giám sát sinh viên theo thời gian thực, và tối ưu hiệu suất khi có số lượng người truy cập lớn.

1.5.2 Phạm vi không gian

- Hệ thống được nghiên cứu và triển khai thử nghiệm trong môi trường phòng máy vi tính tại các cơ sở giáo dục có tổ chức thi trắc nghiệm trực tuyến.
- Hệ thống áp dụng trong phạm vi một tổ chức cụ thể (trường học, trung tâm đào tạo, doanh nghiệp...), chưa triển khai diện rộng hoặc theo mô hình mở.

1.5.3 Phạm vi thời gian

- Thời gian nghiên cứu và phát triển hệ thống dự kiến trong khoảng 2 năm.
- Quá trình kiểm thử, đánh giá và hoàn thiện diễn ra sau khi hệ thống được triển khai thử nghiệm.

1.6 Phương pháp nghiên cứu

Trong quá trình thực hiện đề tài, các phương pháp nghiên cứu được áp dụng bao gồm:

1.6.1 Nhóm phương pháp nghiên cứu lý luận

- Nghiên cứu tài liệu chuyên ngành:

- Thu thập và tổng hợp lý thuyết từ sách chuyên khảo, tài liệu khoa học liên quan đến thi trắc nghiệm trực tuyến, bảo mật hệ thống, tối ưu hiệu suất và microservices.
- Tìm hiểu các công nghệ như Blazor WebAssembly, Redis Cache, RabbitMQ, SignalR, JWT Authentication, DTO, Middleware... qua tài liệu chính thống, bài báo khoa học, và hướng dẫn từ nhà phát triển.
- Phân tích các hệ thống thi trắc nghiệm hiện có:
 - So sánh mô hình hoạt động, các tính năng, ưu điểm và hạn chế của các hệ thống phổ biến.
 - Đánh giá mức độ phù hợp của các giải pháp bảo mật và tối ưu hiệu suất đang được sử dụng.

1.6.2 Nhóm phương pháp nghiên cứu thực tiễn

- Khảo sát thực trạng:
 - Thu thập thông tin về nhu cầu tổ chức thi trắc nghiệm tại các trường học, trung tâm đào tạo có tổ chức thi trực tiếp tại phòng máy.
 - Khảo sát yêu cầu từ giảng viên, quản trị viên và sinh viên để hiểu rõ nhu cầu thực tế về hệ thống thi trắc nghiệm.
- Nghiên cứu thử nghiệm và triển khai hệ thống:
 - Thiết kế và phát triển hệ thống dựa trên các yêu cầu thực tế.
 - Triển khai thử nghiệm hệ thống trên một nhóm người dùng nhỏ để đánh giá tính ổn định, hiệu suất và bảo mật.
 - Điều chỉnh, tối ưu dựa trên phản hồi của người dùng và kết quả thử nghiệm.

1.6.3 Nhóm phương pháp xử lý số liệu

- Phân tích định lượng:
 - Đánh giá hiệu suất hệ thống dựa trên các chỉ số như tốc độ truy xuất dữ liệu, khả năng chịu tải, độ trễ của hệ thống....

- Kiểm tra mức độ tối ưu của Redis Cache, RabbitMQ, SignalR bằng cách đo lường số lượng truy vấn và tốc độ phản hồi.
- Phân tích định tính:
 - Tổng hợp ý kiến từ người dùng thử nghiệm để đánh giá mức độ dễ sử dụng, trải nghiệm người dùng, hiệu quả giám sát thi.
 - Xác định các vấn đề tồn đọng và đề xuất giải pháp cải thiện.

CHƯƠNG 2: CƠ SỞ LÝ LUẬN

2.1 Các khái niệm nghiên cứu có liên quan

- **Hệ thống thi trắc nghiệm trực tuyến:** Hệ thống thi trắc nghiệm trực tuyến là một nền tảng phần mềm hỗ trợ tổ chức các kỳ thi bằng hình thức trắc nghiệm trên máy tính, giúp tự động hóa quá trình kiểm tra, đánh giá và chấm điểm. Hệ thống này có thể triển khai trong môi trường phòng máy có giám thị giám sát hoặc thi từ xa. Trong đề tài này, hệ thống chỉ áp dụng cho thi trực tiếp tại phòng máy.
- **Blazor WebAssembly:** Blazor WebAssembly là một framework của Microsoft dùng để xây dựng ứng dụng web chạy trên trình duyệt bằng ngôn ngữ C#. Đây là công nghệ Single Page Application (SPA) giúp tạo giao diện tương tác mượt mà mà không cần tải lại trang.
- **Microservices:** Microservices là kiến trúc phần mềm chia một hệ thống lớn thành các dịch vụ nhỏ, độc lập, giao tiếp với nhau qua API. Việc sử dụng microservices giúp hệ thống dễ mở rộng, dễ bảo trì và nâng cao hiệu suất.
- **JWT Authentication:** JWT (JSON Web Token) là một phương thức xác thực hiện đại, bảo mật, không cần lưu trạng thái trên server. Trong hệ thống, JWT được lưu vào SessionStorage để duy trì phiên đăng nhập.
- **Redis Cache:** Redis là hệ thống lưu trữ dữ liệu dạng key - value giúp giảm tải truy vấn trực tiếp vào database. Hệ thống thi trắc nghiệm sử dụng Redis Cache để tăng tốc độ tải đề thi, câu hỏi và lưu trạng thái làm bài của thí sinh.
- **RabbitMQ:** RabbitMQ là một hệ thống hàng đợi tin nhắn (message queue) giúp xử lý dữ liệu theo cơ chế bất đồng bộ. Trong hệ thống thi, RabbitMQ được dùng để lưu bài làm tự động sau mỗi 2 phút mà không gây quá tải hệ thống.
- **SignalR:** SignalR là một thư viện giúp giao tiếp thời gian thực giữa client và server. Trong hệ thống thi trắc nghiệm, SignalR hỗ trợ:
 - Giám sát trạng thái thí sinh (đăng nhập, bắt đầu thi, hoàn thành bài, bị out...).
 - Cơ chế khóa bài thi và dừng ca thi.

- Thông báo sự kiện thời gian thực từ quản trị viên đến thí sinh.
- **Cơ chế chống gian lận:** Hệ thống thi trắc nghiệm áp dụng nhiều cơ chế chống gian lận như:
 - Giám sát hành vi thí sinh trong phòng máy bằng giám thị.
 - Chặn mở tab mới và tắt Inspect Element trên trình duyệt.
 - Điểm số hiển thị bằng Canvas ảnh để tránh chỉnh sửa bằng công cụ dev tools.

2.2 Các đề tài, kết quả nghiên cứu có liên quan

2.2.1. Nghiên cứu về hệ thống thi trắc nghiệm trực tuyến

- Hệ thống thi trực tuyến LMS (Learning Management System): Các nền tảng LMS như Moodle, Edmodo hỗ trợ thi trắc nghiệm nhưng thường áp dụng cho thi từ xa, không tối ưu cho thi trong phòng máy có giám thị.
- Các nghiên cứu về tối ưu hiệu suất hệ thống thi trực tuyến: Các nghiên cứu trước đây tập trung vào tăng khả năng chịu tải, nhưng chưa khai thác sâu vào kiến trúc microservices kết hợp Redis Cache, RabbitMQ, SignalR như đề tài này.

2.2.2. Các nghiên cứu về công nghệ liên quan

- Blazor WebAssembly: Công nghệ này được Microsoft phát triển nhưng chưa được ứng dụng rộng rãi trong các hệ thống thi trắc nghiệm quy mô lớn.
- Redis Cache trong hệ thống thời gian thực: Một số nghiên cứu cho thấy Redis giúp giảm tải truy vấn đến database, cải thiện tốc độ phản hồi hệ thống.
- RabbitMQ trong xử lý dữ liệu lớn: RabbitMQ được chứng minh là giải pháp tốt để xử lý hàng đợi bài làm của hàng nghìn thí sinh mà không gây nghẽn hệ thống.

2.3 Các hạn chế về mặt lý luận hiện tại

Dù có nhiều nghiên cứu và ứng dụng về hệ thống thi trắc nghiệm trực tuyến, vẫn tồn tại một số hạn chế về mặt lý luận và công nghệ mà đề tài này cần giải quyết:

- Chưa có nhiều nghiên cứu về hệ thống thi trắc nghiệm dựa trên Blazor WebAssembly: Phần lớn các hệ thống thi trực tuyến hiện nay được phát triển bằng React, Angular hoặc ASP.NET MVC, trong khi Blazor WebAssembly vẫn còn khá mới. Và chưa có nhiều tài liệu chính thức về tối ưu hiệu suất của Blazor WebAssembly trong môi trường thi trắc nghiệm có số lượng thí sinh lớn.
- Hạn chế trong việc tối ưu hiệu suất hệ thống thi trắc nghiệm: Redis Cache và RabbitMQ đã được nghiên cứu trong các hệ thống lớn, nhưng chưa có nhiều nghiên cứu cụ thể về cách tích hợp hai công nghệ này vào hệ thống thi trực tuyến để tăng hiệu suất và đảm bảo tính ổn định. Và các phương pháp xử lý lưu bài làm tự động sao cho không gây nghẽn mạng vẫn còn là một vấn đề cần nghiên cứu thêm.
- Chưa có nhiều nghiên cứu chuyên sâu về cơ chế chống gian lận trong phòng máy: Các hệ thống thi trực tuyến thường tập trung vào thi từ xa, sử dụng AI giám sát hoặc camera để phát hiện gian lận. Và chưa có nhiều nghiên cứu chi tiết về việc kết hợp phần mềm và giám sát viên trong phòng máy để kiểm soát hành vi thí sinh.
- Chưa có giải pháp toàn diện để ngăn chặn chỉnh sửa điểm số trên giao diện web: Các hệ thống hiện tại chủ yếu hiển thị điểm số trực tiếp trên HTML, dễ bị thay đổi bằng công cụ dev tools. Và việc sử dụng Canvas ảnh để hiển thị điểm số chưa được nghiên cứu sâu về mức độ bảo mật và hiệu quả so với các giải pháp truyền thống.
- Hạn chế trong việc bảo vệ dữ liệu cá nhân và bảo mật hệ thống: Hầu hết các nghiên cứu về thi trực tuyến chưa tập trung nhiều vào bảo mật dữ liệu người dùng, đặc biệt là các rủi ro tiềm ẩn khi sử dụng JWT Token trong Blazor WebAssembly. Và chưa có nhiều nghiên cứu về cách kết hợp DTO và Middleware để bảo vệ dữ liệu trong các hệ thống thi trắc nghiệm.

CHƯƠNG 3: KẾ HOẠCH VÀ DỰ KIẾN KẾT QUẢ

3.1 Dự kiến kết quả

Sau khi hoàn thành nghiên cứu và triển khai hệ thống thi trắc nghiệm trực tuyến, đề tài dự kiến đạt được các kết quả sau:

- Xây dựng hệ thống thi trắc nghiệm trực tuyến tối ưu về hiệu suất: Phát triển một nền tảng thi trắc nghiệm trực tuyến dựa trên Blazor WebAssembly theo kiến trúc microservices, đảm bảo khả năng mở rộng. Ngoài ra, tối ưu hóa hiệu suất thông qua Redis Cache (giảm truy vấn đến database) và RabbitMQ (quản lý hàng đợi bài làm).
- Đảm bảo tính ổn định và bảo mật của hệ thống: Xây dựng cơ chế xác thực an toàn bằng JWT Authentication (lưu SessionStorage) và mã hóa mật khẩu bằng Bcrypt. Và bảo vệ dữ liệu thông qua Middleware xử lý lỗi và DTO để tránh rò rỉ thông tin nhạy cảm.
- Hỗ trợ đa dạng hình thức câu hỏi và tối ưu trải nghiệm người dùng: Xây dựng giao diện thân thiện, dễ sử dụng, tương thích với nhiều thiết bị cũng như hỗ trợ các dạng câu hỏi:
 - Câu hỏi trắc nghiệm đơn lựa chọn, đa lựa chọn.
 - Câu hỏi nhóm.
 - Câu hỏi điền khuyết.
 - Câu hỏi âm thanh (hạn chế số lần nghe).
 - Hỗ trợ hiển thị công thức toán học bằng LaTeX.
- Cơ chế giám sát và chống gian lận trong phòng thi: Tích hợp SignalR giúp giám thị theo dõi trạng thái thí sinh (đăng nhập, vào thi, hoàn tất bài thi, điểm số), sử dụng cơ chế kiểm soát tab và tắt inspect trang, giúp giảm thiểu hành vi gian lận. Ngoài ra, chặn chỉnh sửa điểm số bằng cách hiển thị điểm thông qua Canvas ảnh thay vì HTML thông thường.
- Hỗ trợ quản trị viên dễ dàng quản lý và theo dõi kỳ thi:
 - Tính năng đóng băng ca thi, cộng giờ, dừng ca thi và log out sinh viên.

- Hỗ trợ nhập danh sách sinh viên từ file Excel bằng EPPlus.
 - Cơ chế lưu bài làm tự động mỗi 2 phút, đảm bảo bài thi không bị mất dữ liệu khi gặp sự cố.
 - Khi hệ thống bị lỗi nặng, có cơ chế tự động reload để tránh gián đoạn thi cử.
- Đánh giá hiệu quả của hệ thống:
- Thử nghiệm thực tế với số lượng sinh viên tham gia lớn, đo lường hiệu suất hệ thống.
 - So sánh tốc độ tải đề, lưu bài và xử lý kết quả trước và sau khi áp dụng Redis Cache và RabbitMQ.
 - Đánh giá mức độ bảo mật thông qua kiểm thử xâm nhập và phản hồi từ người dùng.

3.2 Tiến độ thực hiện

Hiện tại, tiến độ thực hiện đã đạt gần một nửa so với kế hoạch tổng thể. Các chức năng quan trọng như đăng nhập, làm bài thi, lưu bài làm, cơ chế tải đề thi cũng như công cụ quản lý ca thi dành cho quản trị viên đều đang hoạt động ổn định và hiệu quả. Trong giai đoạn tiếp theo, dự án sẽ tập trung vào các kế hoạch chi tiết được trình bày trong bảng 3.3 bên dưới.

3.3 Kế hoạch thực hiện

| STT | Công việc | Tuần 1 - 2 | Tuần 3 - 4 | Tuần 5 - 6 | Tuần 7 - 8 | Tuần 9 - 10 | Tuần 11 - 12 |
|-----|---|---------------|---------------|---------------|---------------|----------------|--------------------|
| 1 | Nghiên cứu & Triển khai kiến trúc Microservices | x | | | | | |
| 2 | Phân tách các dịch vụ độc lập | x | x | | | | |

| | | | | | | | |
|----|--|---|---|---|---|---|---|
| 3 | Tích hợp Docker, Kubernetes & Docker Hub | x | x | x | x | x | x |
| 4 | Báo cáo & trao đổi tiến độ với GVHD | x | x | x | x | x | x |
| 5 | Triển khai giao thức gRPC | | x | | | | |
| 6 | Hoàn thiện giao diện MudBlazor | | x | | | | |
| 7 | Xây dựng trang quản lý đề thi | | x | x | | | |
| 8 | Xây dựng các tính năng bổ sung (trao đổi với GVHD) | | | x | x | x | |
| 9 | Kiểm tra các thuật toán & tối ưu code | | | x | x | x | x |
| 10 | Kiểm thử chức năng và cơ sở dữ liệu | | x | x | x | x | x |
| 11 | Chuẩn bị slide & tài liệu báo cáo giữa kì | | | x | | | |
| 12 | Xử lý Middleware & API lỗi | | | | x | | |
| 13 | Hoàn thiện các nhiệm vụ chưa xong | | | | x | x | |
| 14 | Tối ưu hệ thống và kiểm tra hiệu năng RabbitMQ | | | | x | x | |
| 15 | Viết tài liệu hướng dẫn & báo cáo | | | | | | x |
| 16 | Chuẩn bị demo sản phẩm | | | | | | x |

TÀI LIỆU THAM KHẢO

- [1] Ahmad Mozaffar (2023). Mastering Blazor WebAssembly. Packt Publishing Ltd.
- [2] Taurius Litvinavicius (2023). Exploring Blazor. APress Media, LLC.
- [3] Jimmy Engström (2024). Web Development with Blazor. Packt Publishing Ltd.
- [4] Lê Xuân Mạnh (2012). Hệ thống thi trắc nghiệm và ngân hàng câu hỏi trắc nghiệm ExamSuite 2012. Đồ án đại học, Đại học Công nghiệp Thực phẩm TP.HCM.
- [5] Cao Tùng Anh. (2019). Cơ sở dữ liệu và quản trị cơ sở dữ liệu.
- [6] Văn Như Bích , Võ Hoàng Khang (2019) Phân tích thiết kế hệ thống thông tin.

Tp.HCM, ngày 25 tháng 2 năm 2025

SINH VIÊN THỰC HIỆN

(Ký và ghi rõ họ tên)

XÁC NHẬN CỦA BỘ MÔN

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)