

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



**HỒ PHƯỚC NHÂN
LÂM LƯƠNG KHÊ**

**XÂY DỰNG WEBSITE THI TRẮC NGHIỆM ONLINE
BẰNG REACTJS VÀ .NET CORE API**

**ĐỒ ÁN NGÀNH
NGÀNH KHOA HỌC MÁY TÍNH**

TP. HỒ CHÍ MINH, 2023

GVHD: TS. Phan Trần Minh Khuê

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



HỒ PHƯỚC NHÂN
LÂM LƯƠNG KHÊ

XÂY DỰNG WEBSITE THI TRẮC NGHIỆM ONLINE
BẰNG REACTJS VÀ .NET CORE API

Mã số sinh viên: 2051010214

Mã số sinh viên: 2051010144

ĐỒ ÁN NGÀNH
NGÀNH KHOA HỌC MÁY TÍNH

Giảng viên hướng dẫn: TS. Phan Trần Minh Khuê

TP. HỒ CHÍ MINH, 2023

LỜI CẢM ƠN

Trong quá trình học tập nhóm chúng em xin cảm ơn chân thành đến tất cả quý thầy, cô đã rèn luyện lối tư duy logic, chỉ dạy, khuyên bảo để từ đó chúng em có được những kiến thức quý báu và có thể hoàn thành được môn đồ án ngành Khoa học máy tính.

Đặc biệt, nhóm chúng em xin gửi lời biết ơn sâu sắc đến thầy Phan Trần Minh Khuê giảng viên hướng dẫn bộ môn đồ án ngành khoa Công Nghệ Thông Tin trường Đại Học Mở thành phố Hồ Chí Minh. Với sự hướng dẫn giúp đỡ, chỉ dạy và đóng góp ý kiến tận tâm nhiệt tình và chu đáo của thầy cùng với lời động viên thì nhóm chúng em đã vượt qua những khó khăn khuất mắt khi làm môn đồ án này.

Đồng thời chúng em xin cảm ơn đến trường Đại Học Mở thành phố Hồ Chí Minh đã tạo ra một môi trường có cơ sở vật chất đầy đủ, tiện nghi, giúp cho các sinh viên có thể học tập, rèn luyện, trường còn tổ chức các hoạt động theo khoa về chủ đề yêu thích giúp cho sinh viên có thể sáng tạo, tự tin khi tham gia.

Cuối cùng chúng em xin gửi lời cảm ơn đến gia đình đã tạo những điều kiện thuận lợi giúp chúng em có thể đồng hành cùng trường Đại Học Mở được gặp được bạn bè, thầy cô luôn động viên, hỗ trợ tinh thần giúp chúng em cải thiện hơn trong cuộc sống.

Do kiến thức ngành khá rộng, còn chúng em thì bị hạn chế nhiều về kiến thức và kinh nghiệm thực tế nên trong quá trình nghiên cứu và phát triển môn đồ án có nhiều sai sót khi làm. Chúng em kính mong thầy cô đóng góp ý kiến, truyền đạt những kinh nghiệm quý báu để chúng em đúc kết và phát triển nhiều hơn trong con đường sự nghiệp sau này.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TÓM TẮT ĐỒ ÁN NGÀNH

Với những kiến thức đã học ở trường và sự nghiên cứu, tìm tòi, thông qua những lần làm bài tập lớn. Chúng em đã có một nền tảng cơ bản để xây dựng môn đồ án ngành này nhằm mục đích tạo ra một phần mềm Web thi trắc nghiệm trực tuyến bằng ngôn ngữ C# và ReactJS cùng với lưu trữ dữ liệu bằng MS SQL Server. Sau những thời gian nghiên cứu, và tìm hiểu về cách hoạt động qua lại của hai ngôn ngữ cùng với những lời góp ý, xây dựng hệ thống của thầy Phan Trần Minh Khuê, thì nhóm chúng em đã xác định được tính năng quan trọng cũng như giao diện người dùng với phần làm bài thi một cách trực quan. Đồng thời chúng em thấy được việc quản lý bài thi và những câu hỏi của giáo viên cho các học sinh, sinh viên là hết sức quan trọng. Nên chúng em đặc biệt quan tâm đến chức năng thêm và chỉnh sửa các câu hỏi bài thi cũng như quản lý kết quả của từng học sinh, sinh viên để dàng hơn.

Nhằm mang đến các giải pháp quản lý hiệu quả cho việc tổ chức bài thi trắc nghiệm chúng em tin rằng ứng dụng này phục vụ tốt cho nhu cầu về kiểm tra và đánh giá năng lực trong lĩnh vực giáo dục. Với giao diện dễ dùng việc sử dụng ứng dụng để quản lý sẽ tốn ít thời gian hơn vì những chức năng đơn giản và được miêu tả dễ hiểu.

Tóm lại đồ án ngành của chúng em giúp đơn giản hóa các quy trình của cấu trúc đề thi trắc nghiệm cho các giáo viên, giảng viên cũng như là học sinh, sinh viên. Chúng em hi vọng sẽ mang lại những trải nghiệm tích cực cho người quản lý cũng như làm bài thi góp phần nâng cao chất lượng giáo dục và đạt được mục tiêu học tập.

ABSTRACT

With the knowlegde learned at school and the research and exploration, through doing large homework. We have a basic foundation to build this industry project with the aim of creating an online multiple-choice exam Web software using C# and ReactJS languages along with data storage using MS SQL Server. After a period of research and learning about how the two languages interact, along with comments and system building from teacher Phan Tran Minh Khue, our team has identified an important feature. As well as a user interface with intuitive test-taking. At the same time, we see that the management of tests and questions from teachers to students is extremely important. So, we are especially intersted in the function of addong and editing exam questions as well as managing the results of each studen more easily.

In order to provide effective management solutions for organizing multiple-choice tests, we believe that this application servers well the needs of testing and assessing capaccity in the field of education. With an easy-to-use interface, using the application to manage will take less time because the function are simple and clearly described.

In short, our industry projects help simplify the process of multiple-choice exam structure for teachers, lectures as well and students. We hope to bring positive experiences to managers as well as test-taking, contributing to improving the qulity of educacion and achieving learning goals

MỤC LỤC

| | |
|--|-----------|
| LỜI CẢM ƠN | 1 |
| NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN..... | 2 |
| TÓM TẮT ĐỒ ÁN NGÀNH..... | 3 |
| ABSTRACT | 4 |
| DANH MỤC TỪ VIẾT TẮT | 8 |
| DANH MỤC HÌNH VẼ | 9 |
| DANH MỤC BẢNG | 11 |
| MỞ ĐẦU..... | 12 |
| Chương 1. GIỚI THIỆU ĐỀ TÀI..... | 13 |
| 1.1. Giới thiệu..... | 13 |
| 1.2. Mục tiêu đề tài..... | 13 |
| 1.2.1. Mục tiêu..... | 13 |
| 1.2.2. Đối tượng nghiên cứu..... | 13 |
| 1.3. Lý do chọn đề tài..... | 13 |
| 1.4. Bố cục báo cáo | 14 |
| Chương 2. CƠ SỞ LÝ THUYẾT | 15 |
| 2.1. C#.NET CORE API | 15 |
| 2.1.1. Tổng quan về C# | 15 |
| 2.1.2. Tổng quan về .NET | 15 |
| 2.1.3. RESTful API | 15 |
| 2.1.4. Cấu hình và cài đặt kết nối Database | 18 |
| 2.2. HTML + CSS | 20 |
| 2.2.1. Tổng quan về HTML..... | 20 |
| 2.2.2. Tổng quan về CSS | 20 |
| 2.3. Javascript..... | 20 |

| | | |
|-----------|---|----|
| 2.4. | ReactJS | 21 |
| 2.4.1. | Tổng quan về ReactJS | 21 |
| 2.4.2. | Cấu hình và cách hoạt động ReactJS | 23 |
| 2.4.3. | Kết nối Api giữa ReactJS và C#..... | 24 |
| Chương 3. | XÂY DỰNG WEBSITE THI TRẮC NGHIỆM ONLINE BẰNG REACTJS VÀ .NET CORE API..... | 25 |
| 3.1. | Giới thiệu hệ thống thi trắc nghiệm | 25 |
| 3.2. | Phân tích và thiết kế hệ thống | 26 |
| 3.2.1. | Lược đồ UseCase..... | 26 |
| 3.2.2. | Đặc tả UseCase..... | 26 |
| 3.2.3. | Sơ đồ hoạt động..... | 31 |
| 3.2.4. | Database | 32 |
| 3.3. | Các nghiệp vụ đã phát triển..... | 34 |
| 3.3.1. | Làm bài thi..... | 34 |
| 3.3.2. | Xuất kết quả..... | 35 |
| 3.3.3. | Xem các câu hỏi | 37 |
| 3.3.4. | Thêm câu hỏi | 38 |
| 3.3.5. | Sửa, xóa câu hỏi | 39 |
| 3.3.6. | Import file..... | 40 |
| 3.3.7. | Export file..... | 42 |
| 3.3.8. | Minh họa API | 43 |
| Chương 4. | KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 44 |
| 4.1. | Kết luận | 44 |
| 4.2. | Hướng phát triển | 44 |
| 4.2.1. | Ưu điểm..... | 44 |
| 4.2.2. | Nhược điểm | 44 |

| | |
|---------------------------------|-----------|
| TÀI LIỆU THAM KHẢO | 45 |
|---------------------------------|-----------|

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

| STT | Ký hiệu viết tắt và thuật ngữ | Tên đầy đủ |
|------------|--------------------------------------|-----------------------------------|
| 1 | CSDL | Cơ sở dữ liệu |
| 2 | HTTP | Hypertext Transfer Protocol |
| 3 | API | Application Programming Interface |
| 4 | URL | Uniform Resource Locator |
| 5 | Website | Trang mạng/ trang web |
| 6 | Request | Yêu cầu |
| 7 | HTML | HyperText Markup Language |
| 8 | CSS | Cascading Style Sheets |
| 9 | REST | REpresentational State Transfer |
| 10 | GET | Phương thức yêu cầu dữ liệu |
| 11 | POST | Phương thức tạo mới dữ liệu |
| 12 | PUT | Phương thức cập nhật dữ liệu |
| 13 | DELETE | Phương thức xóa dữ liệu |

DANH MỤC HÌNH VẼ

| | |
|---|----|
| Hình 2.1.1: Hình ảnh minh họa Mô hình MVC | 15 |
| Hình 2.1.2: Tạo thư mục Models..... | 16 |
| Hình 2.1.3: Lớp Participant.cs trong thư mục Models | 16 |
| Hình 2.1.4: Tạo lớp TracNghiemDbContext.cs cho thư mục Models | 16 |
| Hình 2.1.5: Giao diện View được thiết kế từ ReactJS | 17 |
| Hình 2.1.6: Tạo thư mục Controllers..... | 17 |
| Hình 2.1.7: Hình ảnh minh họa về Route và API..... | 17 |
| Hình 2.1.8: Cài đặt thư viện Entity cho Packages | 18 |
| Hình 2.1.9: Hình ảnh minh họa kết nối Database | 18 |
| Hình 2.1.10: Hình ảnh minh họa cấu hình Database..... | 19 |
| Hình 2.1.11: Tạo thư mục tự động Migrations..... | 19 |
| Hình 2.1.12: Dùng lệnh Add-Migration | 19 |
| Hình 2.1.13: Dùng lệnh Update-Database..... | 19 |
| Hình 2.1.14: Database trong MS SQL Server | 20 |
| Hình 2.4.1: Tạo thư mục Components | 21 |
| Hình 2.4.2: Hình ảnh minh họa cây DOM | 21 |
| Hình 2.4.3: Định tuyến trong ReactJS..... | 22 |
| Hình 2.4.4: Thư mục Hooks chứa các Context | 22 |
| Hình 2.4.5: Cách hoạt động Index.js | 23 |
| Hình 2.4.6: Cách hoạt động App.js sau Index.js | 23 |
| Hình 2.4.7: Cấu trúc axios cho API..... | 24 |
| Hình 3.2.1: Kiến trúc hệ thống quản lý thi trắc nghiệm..... | 25 |
| Hình 3.3.1: Sơ đồ hoạt động của Người dùng | 31 |
| Hình 3.3.2: Sơ đồ hoạt động của Admin | 32 |
| Hình 3.3.3: Cơ sở dữ liệu “QuizDB” | 33 |
| Hình 3.4.1: Giao diện làm bài thi | 34 |
| Hình 3.4.2: Xử lý thời đếm lùi | 34 |
| Hình 3.4.3: Phương thức GET cho bảng “Questions” | 35 |
| Hình 3.4.4: Giao diện xuất kết quả sau khi làm bài | 35 |
| Hình 3.4.5: Hàm so sánh kết quả trả lời | 36 |
| Hình 3.4.6: Hàm tính điểm câu trả lời đúng..... | 36 |

| | |
|---|----|
| Hình 3.4.7: Giao diện hiện các câu hỏi bên Admin | 37 |
| Hình 3.4.8: Dùng phương GET để hiển thị các câu hỏi | 37 |
| Hình 3.4.9: Giao diện thêm câu hỏi..... | 38 |
| Hình 3.4.10: Dùng phương thức POST để thêm dữ liệu | 38 |
| Hình 3.4.11: Giao diện chỉnh sửa câu hỏi | 39 |
| Hình 3.4.12: Dùng phương thức PUT để cập nhật câu hỏi | 39 |
| Hình 3.4.13: Dùng phương thức Delete để xóa câu hỏi | 40 |
| Hình 3.4.14: Giao diện Import file | 40 |
| Hình 3.4.15: Xử lý Import file..... | 40 |
| Hình 3.4.16: Dùng phương thức POST để thêm dữ liệu từ file | 41 |
| Hình 3.4.17: Giao diện xem kết quả từ người dùng | 42 |
| Hình 3.4.18: Xử lý xuất dữ liệu ra file | 42 |
| Hình 3.4.19: Giao diện các API bên C# | 43 |
| Hình 3.4.20: Minh họa dữ liệu sau khi gọi API của “Questions” | 43 |

DANH MỤC BẢNG

| | |
|--|----|
| Bảng 3.2.2-1: Bảng UseCase “Đăng ký” | 27 |
| Bảng 3.2.2-2: Bảng UseCase “Đăng nhập” | 28 |
| Bảng 3.2.2-3: Bảng UseCase “Làm bài thi” | 28 |
| Bảng 3.2.2-4: Bảng UseCase “Xem kết quả thi” | 29 |
| Bảng 3.2.2-5: Bảng UseCase “Quản lý câu hỏi” | 29 |
| Bảng 3.2.2-6: Bảng UseCase “Xem điểm” | 30 |
| Bảng 3.2.2-7: Bảng UseCase “Xuất bảng điểm” | 30 |
| Bảng 3.2.4-1: Bảng “Questions” | 33 |
| Bảng 3.2.4-2: Bảng “Participants” | 33 |

MỞ ĐẦU

Từ những năm 2015 đến nay, thi theo kiểu trắc nghiệm về các chủ đề môn học cho học sinh, sinh viên hầu như đã trở thành phổ biến ở khắp nơi trên thế giới. Qua đó Bộ giáo dục đào tạo của nước ta đã nhìn thấy được tầm quan trọng của hình thức thi này nên đã đưa ra hội họp ý kiến và đưa thi trắc nghiệm này vào thi chính thức cho các học sinh, sinh viên trong các kỳ thi cuối cấp và thi lên đại học. Vào những năm 2008, trắc nghiệm đã được thử trên các môn ở một số trường nhằm cho học sinh, sinh viên làm quen dần với hình thức mới này, nhưng các yêu cầu bài tập được soạn quá ít khi qua bút viết của thầy cô khiến cho các học sinh không trải nghiệm được nhiều bài tập. Vì thế, vào những năm gần đây mạng Internet phát triển mạnh cùng với các công nghệ hiện đại, đã hỗ trợ nhu cầu của con người sử dụng mạng ngày càng nhiều. Từ đó những website làm trắc nghiệm trực tuyến ra đời góp phần giúp cho các giáo viên, giảng viên có thể soạn nhanh chóng và tiết kiệm được thời gian công sức bỏ ra, cùng với học sinh, sinh viên thì có thể làm được nhiều dạng bài tập với nhiều độ khó khác nhau.

Qua những lý do trên thì nhóm em đã chọn đề tài website thi trắc nghiệm, tụi em đã tìm hiểu về đề tài và được thầy Phan Trần Minh Khuê hướng dẫn và phát triển một số tính năng để làm ra một sản phẩm. Với những phát triển này tụi em mong có thể cải thiện thêm tính năng khác để có thể đóng góp một phần cho các học sinh, sinh viên hiện nay. Vì vậy đề tài website thi trắc nghiệm của tụi em đã ra đời.

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu

Ngày nay website thi trắc nghiệm phổ biến rộng rãi trên các mạng Internet như kahoot.it, quizizz.com ... với hình thức thi trắc nghiệm trên các website này giúp cho các giáo viên, giảng viên có thể soạn câu hỏi nhanh chóng, tiết kiệm, còn về học sinh, sinh viên thì có thể làm được nhiều bài tập để có thể học hỏi và rà soát lại những kiến thức đã học. Nó còn có các chức năng thêm những câu hỏi phù hợp với bộ môn thi, sửa hoặc xóa những câu hỏi không phù hợp. Chỉ cần có mạng Internet thì chúng ta dù ở đâu cũng có thể làm những câu hỏi trắc nghiệm.

1.2. Mục tiêu đề tài

1.2.1. Mục tiêu

Mục tiêu của website thi trắc nghiệm tạo ra một giao diện thuận mắt người dùng để có thể quản lý các câu hỏi do người dùng là các giảng viên lựa chọn, có thể đáp ứng và đánh giá năng lực của từng học sinh, sinh viên khi làm bài.

1.2.2. Đối tượng nghiên cứu

Đối tượng nghiên cứu website thi trắc nghiệm này là các giáo viên, giảng viên, học sinh, sinh viên hoặc bất kì người nào thích thú và đã bỏ nhiều thời gian vào việc làm bài thi trắc nghiệm ở những website sử dụng công nghệ C# và ReactJS. Qua đó chúng em đã tổng kết những trải nghiệm và cho ra một số tính năng tiện dụng để có thể cải thiện được quá trình học tập, rà soát lại các kiến thức mà mình chưa có nhằm mang lại nhiều giá trị cho các đối tượng nghiên cứu.

1.3. Lý do chọn đề tài

Tại nước ta trong những năm gần đây, do phương pháp dạy học và hình thức thi cử là điều thiết yếu của các học sinh, sinh viên. Hình thức thi trắc nghiệm trên giấy vào những năm trước đó, tuy nó được cải tiến hơn khi làm bài trên giấy nhưng để cho ra được kết quả từng bài thi trắc nghiệm vẫn còn nhiều việc không được thuận tiện gây tốn thời gian và tài nguyên của thầy cô khi chấm bài, thu bài.

Song song đó việc áp dụng công nghệ hiện đại vào giáo dục ngày càng cấp thiết cũng như hình thức làm trắc nghiệm quá phổ biến, nên chúng em đã chọn đề tài website thi

trắc nghiệm để giúp cho người dùng có thể tạo những câu hỏi và câu trả một cách dễ dàng nhưng vẫn đảm bảo kiến thức. Đồng thời việc làm trắc nghiệm đối với các sinh viên, học sinh sẽ dễ dàng hơn so với lúc trước, khi hoàn thành một bài thi trắc nghiệm học sinh, sinh viên có thể biết thêm nhiều kiến thức mới, trải nghiệm mới, học hỏi và nâng cao tầm hiểu biết để cải thiện bản thân mình. Vì thế đó là lý do chúng em lựa chọn đề tài website thi trắc nghiệm này để có thể góp một ít cho xã hội.

1.4. Bộ cục báo cáo

Bộ cục báo cáo được tổng hợp lại và chia thành các chương như sau:

Chương 1: Giới thiệu về đề tài, mục tiêu đề tài, đối tượng nghiên cứu và lý do chọn đề tài.

Chương 2: Trình bày về lý thuyết của các ngôn ngữ, cấu hình và cài đặt các dữ liệu.

Chương 3: Xây dựng hệ thống Website thi trắc nghiệm, giới thiệu kiến trúc cũng như phân tích, thiết kế hệ thống và các nghiệp vụ phát triển của hệ thống.

Chương 4: Kết luận đề tài và tìm ra nhiều ưu điểm, nhược điểm theo hướng phát triển.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. C#.NET CORE API

2.1.1. Tổng quan về C#

C# (hay là C sharp) là ngôn ngữ lập trình hiện đại đơn giản và thuần hướng đối tượng, được phát triển bởi ông lớn Microsoft và cho ra mắt vào những năm 2000. Dựa trên hai ngôn ngữ mạnh nhất là C++ và Java, Microsoft đã cân bằng hai ngôn ngữ này và tạo ra C# [1].

2.1.2. Tổng quan về .NET CORE

.NET CORE được thừa kế từ .NET Framework, nó là một phiên bản đa nền tảng với một mã nguồn mở miễn phí trên nhiều hệ điều hành như Windows, Linux, macOS... cũng như C#.NET CORE cũng được phát triển bởi nhân viên của Microsoft với phiên bản .NET CORE được hoàn thành vào năm 2016 [2].

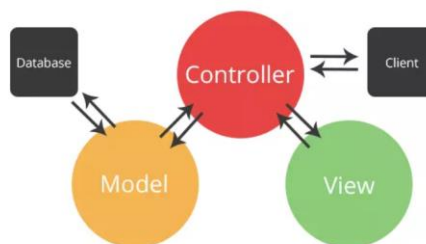
2.1.3. RESTful API

API là viết tắt của từ (Application Programming Interface) được hiểu nôm na là một giao diện lập trình ứng dụng, nó có khả năng giao tiếp giữa các ứng dụng, hệ điều hành như Chrome, Firefox, Window, ... thông qua mạng Internet.

REST được viết tắt theo từ (REpresentational State Transfer) gửi yêu cầu bằng cách sử dụng các phương thức của HTTP như DELETE để xóa, GET để lấy dữ liệu, ... Cho phép giao tiếp giữa các máy, nó như một dạng chuyển đổi cấu trúc dữ liệu.

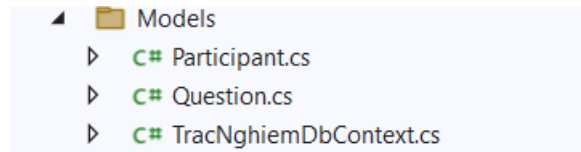
Với mỗi API được thiết kế cho các ứng dụng web thì được quản lý chúng phải có một tiêu chuẩn chung vì thế khái niệm **RESTful API** ra đời và được sử dụng phổ biến không những ở Web mà còn ở những ứng dụng Mobile [3].

2.1.3.1. Mô hình MVC (Model – View - Controller)



Hình 2.1.1: Hình ảnh minh họa Mô hình MVC

Model (M) là nơi chứa cấu trúc dữ liệu có khả năng lưu trữ dữ liệu của một ứng dụng. Model còn đóng vai trò liên kết giữa View và Controller.



Hình 2.1.2: Tạo thư mục Models

Dựa vào **Hình 2.1.2** Tạo ra các quan hệ thuộc tính cho cơ sở dữ liệu đó là 2 bảng Participant.cs và Question.cs.

```
public class Participant
{
    [Key]
    1 reference
    public int ParticipantId { get; set; }
    [Column(TypeName = "nvarchar(50)")]
    2 references
    public string Email { get; set; }
    [Column(TypeName = "nvarchar(50)")]
}
```

Hình 2.1.3: Lớp Participant.cs trong thư mục Models

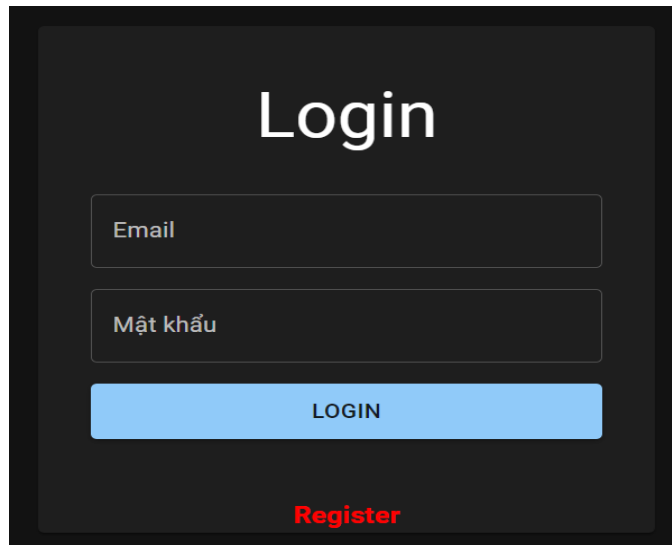
Trong **Hình 2.1.3** Khi có từ khóa [key] thì được coi là đặt cho biến ParticipantId thành khóa chính, còn từ khóa [Column(TypeName="nvarchar(50)")] là được phép ghi tối đa 50 kí tự cho biến Email. Kế tiếp là cách tạo Database.

```
public class TracNghiemDbContext:DbContext
{
    0 references
    public TracNghiemDbContext(DbContextOptions<TracNghiemDbContext> options)
        :base(options){}
    11 references
    public DbSet<Question> Questions { get; set; }
    11 references
    public DbSet<Participant> Participants { get; set; }
}
```

Hình 2.1.4: Tạo lớp TracNghiemDbContext.cs cho thư mục Models

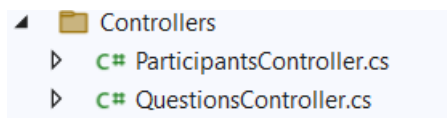
Nhìn vào **Hình 2.1.4** ta thấy dòng đầu lớp “TracNghiemDbContext” kế thừa lớp “DbContext” mà lớp “DbContext” là một phiên bản của lớp “EntityFrameworkCore” và được cài trong Lớp Dependencies dùng để truy vấn dữ liệu, kết nối cơ sở dữ liệu và lưu dữ liệu. Dòng thứ hai lớp “TracNghiemDbContext” nhận tham số từ bên trong “DbContextOptions”, tiếp theo tham số đó chứa các cài đặt và cấu hình cho cơ sở dữ liệu, “base(options)” lúc này truyền tham số options vào “DbContext”. Hai dòng cuối

“Dbset” cũng thuộc lớp “EntityFrameworkCore” được sử dụng để tập hợp các bảng trong Model và hai bảng này “Questions” và “Paricipants” đại diện để có thể truy vấn. Tiếp theo là phần View.



Hình 2.1.5: Giao diện View được thiết kế từ ReactJS

View (V) là lớp giao diện cho ứng dụng, View là nơi nhận request từ người dùng cho Controller xử lý trước khi trả kết quả. Theo **Hình 2.1.5** đây là giao diện thiết kế của View được sử dụng bởi ngôn ngữ ReactJS.



Hình 2.1.6: Tạo thư mục Controllers

Theo **Hình 2.1.6** thì **Controller** (C) là nơi tiếp nhận các request yêu cầu xử lý từ người dùng và là nơi xử lý các logic lấy đúng dữ liệu cần thiết từ Model và hiển thị ra cho View.

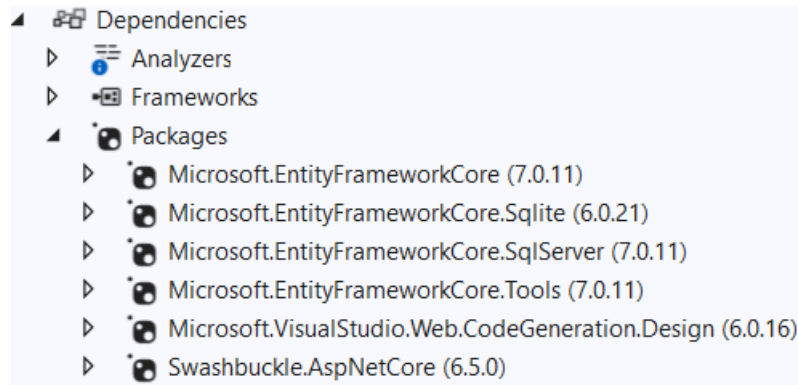
```
[Route("api/[controller]")]
[ApiController]
1 reference
public class ParticipantsController : ControllerBase
```

Hình 2.1.7: Hình ảnh minh họa về Route và API

Dựa vào **Hình 2.1.7** từ khóa [Route(“api/[controller]”)] thì ta có Route dùng để điều hướng các đường dẫn request đến các controller [2]. [controller] sẽ được thay thế tên là “Participants” và nó được xử lý từ “ParticipantsController” và ta sẽ có đường dẫn

“api/Participants”. Từ khóa [ApiController] nó được sử dụng để đánh dấu lớp “ParticipantsController”, nó còn cung cấp lợi ích cho việc xử lý yêu cầu các giao thức.

2.1.4. Cấu hình và cài đặt kết nối Database



Hình 2.1.8: Cài đặt thư viện Entity cho Packages

Dựa vào **Hình 2.1.8** ta có thể cài đặt các packages như sau ở Dependencies nhấp chuột phải và chọn Manage Nuget Packages ..., sau đó nó sẽ hiện bảng “Nuget” nhấp vào phần Browse ở bên phải trên cùng của bảng “Nuget” và ở phần “Search” bạn có thể ghi các dòng như trong **Hình 2.1.8**. Ở dòng đầu của gói Packages thì chúng đã được sử dụng cho **Hình 2.1.4**, tiếp đến dòng thứ hai và dòng thứ ba (Sqlite và SqlServer) chúng tương tự nhau đều được hỗ trợ để tương tác với cơ sở dữ liệu nhưng riêng dòng thứ ba (SqlServer) thì được tùy chỉnh riêng cho SQL Server. Ở dòng thứ tư (Tools) có thể sử dụng các câu lệnh cho Migrations, để đưa dữ liệu theo thư mục Model có lớp “TracNghiemDbContext.cs” vào SQL Server chúng ta cần cấu hình theo các bước và sử dụng các câu lệnh sau.

```
"ConnectionStrings": {
  "DevConnection": "Data Source=NHANHO\\SQLEXPRESS;Initial
```

Hình 2.1.9: Hình ảnh minh họa kết nối Database

Ở phần appsettings.json ta có dòng lệnh kết nối SQLServer "ConnectionStrings": { "DevConnection": "Data Source = NHANHO\\SQLEXPRESS; Initial Catalog = QuizDB; Integrated Security = True; Encrypt = False;" }

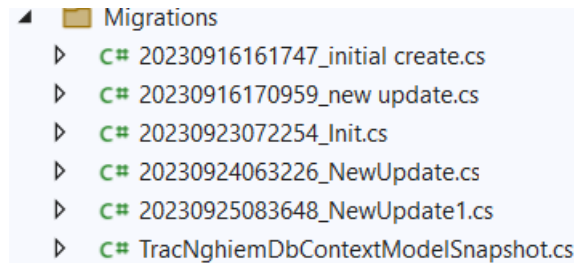
Theo **Hình 2.1.9** trên thì ConnectionStrings là phần cấu hình sử dụng để lưu trữ chuỗi kết nối, Devconnection là tên của chuỗi kết nối về phần bên trong “Data Source” kết nối đến địa chỉ máy chủ của “MS SQL Server”, “Initial Catalog” phần này xác định

tên và thực hiện trên cơ sở dữ liệu đó, “Integrated Security” lệnh này để xác thực thông tin đăng nhập vào SQL Server có đúng hay không, “Encrypt” phần này là phần mã hóa khi được kết nối đến. Về phần cấu hình ở “Program.cs”

```
builder.Services.AddDbContext<TracNghiemDbContext>(options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DevConnection")));
```

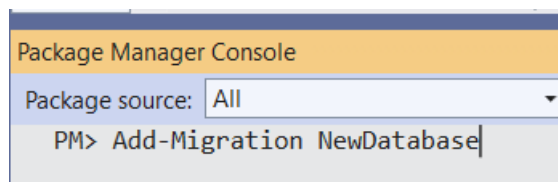
Hình 2.1.10: Hình ảnh minh họa cấu hình Database

Giải thích đơn giản theo **Hình 2.1.10** ta lấy chuỗi kết nối có tên “DevConnection” lúc ở trên **Hình 2.1.9** và phần “builder” cho phép bạn truy cập vào chúng từ lúc đó bạn có thể kết nối được với Database. Sau khi cấu hình xong chúng ta đến bước cuối là dựa trên lớp Migrations



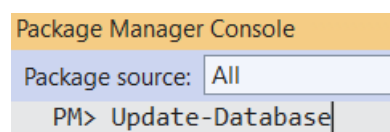
Hình 2.1.11: Tạo thư mục tự động Migrations

Để tạo ra được Migrations bạn vào phần “Tools” trên thanh công cụ của Visual rồi xuống phần “Nuget Packages Manager” và chọn “Package Manager Console” sau đó nó sẽ hiện cửa sổ “Console” bạn gõ lệnh như sau



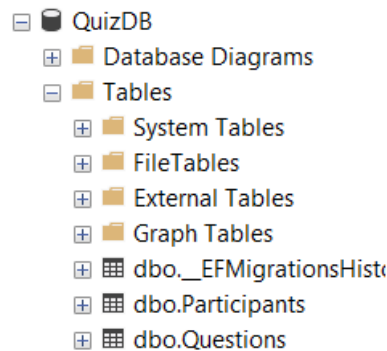
Hình 2.1.12: Dùng lệnh Add-Migration

Theo hình trên đây là lệnh “Add-Migration” là thêm vào Migration một class được tạo theo Model mà bạn đã tạo database như **Hình 2.1.4** giúp bạn lưu trữ dòng lệnh bạn đã tạo, dòng chữ “NewDatabase” là tên bạn đặt tùy ý cho class của bạn. Xong khi xong lệnh Add tiếp theo là



Hình 2.1.13: Dùng lệnh Update-Database

Lệnh này là lệnh cập nhật cơ sở dữ liệu của bạn vào MS SQL Server Khi cập nhật xong bạn có thể vào MS SQL Server để kiểm tra như **Hình 2.1.14**.



Hình 2.1.14: Database trong MS SQL Server

2.2. HTML + CSS

2.2.1. Tổng quan về HTML

HTML được viết tắt từ những cụm từ (Hypertext Markup Language) là một ngôn ngữ đánh dấu siêu văn bản. HTML được sử dụng để cấu tạo nên các trang web, ứng dụng, các đoạn văn bản..., các phần tử HTML này được ghi và được bọc bởi dấu ngoặc nhọn (<HTML>). Điều lưu ý là HTML không phải là ngôn ngữ lập trình [4].

2.2.2. Tổng quan về CSS

CSS được viết tắt là (Cascading Style Sheets) đây cũng không phải là ngôn ngữ lập trình nó được hiểu là một bộ cấu trúc kỹ thuật định kiểu lại các phần tử đã được sinh ra bởi ngôn ngữ đánh dấu, nói mạch lạc hơn và ngắn gọn thì nó cho phép thiết kế lại HTML như thay đổi được font chữ, màu chữ, kích thước bố cục cho các trang web, văn bản một cách bắt mắt người dùng [4].

2.3. Javascript

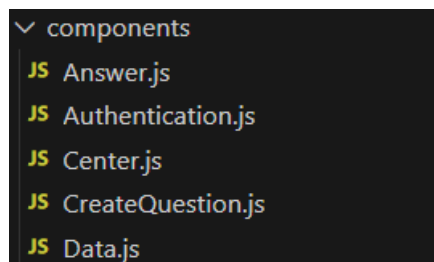
Javascript là một ngôn ngữ lập trình Web phía Client, Javascript cho phép truy xuất và thêm các phần tử vào trong trang Web, cập nhật thay đổi nội dung của thuộc tính của phần tử có trên trang Web. Nó còn được sử dụng để thực hiện các phép tính toán với dữ liệu hoặc xử lý sự kiện mà người dùng tương tác nên phần tử nào đó thuộc trang web. Javascript còn cấu hình được kích thước các trình duyệt kiểm tra dữ liệu người dùng và báo không hợp lệ [4].

2.4. ReactJS

2.4.1. Tổng quan về ReactJS

ReactJS nó là một mã nguồn mở được phát triển bởi ông lớn Facebook và được ra đời vào năm 2013. Nó là một thư viện chứa nhiều Javascript và kèm theo các HTML, CSS hỗ trợ. Mục tiêu tạo ra ReactJS là giúp các trang web sử dụng nhanh, gọn, khả năng mở rộng và dễ thực hiện cao [5].

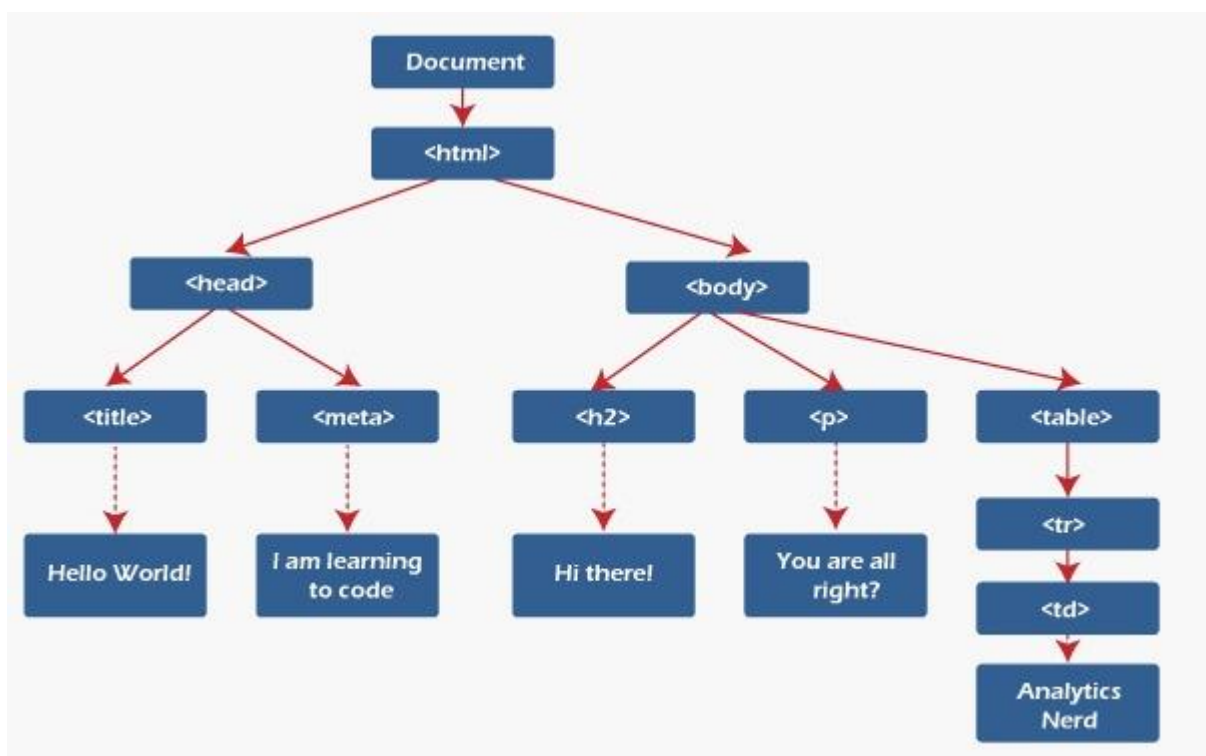
2.4.1.1. Component



Hình 2.4.1: Tạo thư mục Components

Component trong ReactJS giúp chia nhỏ giao diện người dùng để quản lý và tái sử dụng, các component khác nhau sẽ hiển thị các giao diện khác nhau.

2.4.1.2. DOM



Hình 2.4.2: Hình ảnh minh họa cây DOM

DOM được viết tắt từ (Document Object Model) là cấu trúc giao diện người dùng của trang Web viết theo dạng cây, cho phép truy cập vào HTML, thao tác và xem HTML dưới dạng **XML** (Extensible Markup Language là ngôn ngữ đánh dấu mở rộng).

2.4.1.3. Virtual DOM

Virtual DOM là một cây DOM ảo chỉ là một đại diện của DOM thật, mỗi khi trạng thái ứng dụng thay đổi thì DOM ảo sẽ được cập nhật thay vì DOM thật. Vì thế khi bạn sử dụng hàm hoặc thay đổi chức năng là các component thì Web sẽ không reload lại.

2.4.1.4. BrowserRouter

```
<BrowserRouter>
  <Routes>
    <Route path="/" element={<Login />} />
    <Route path="/register" element={<Register />} />
```

Hình 2.4.3: Định tuyến trong ReactJS

BrowserRouter khi sử dụng nó sẽ tạo ra một Router cho ứng dụng của bạn dùng để quản lý các định tuyến và hiển thị thành phần phù hợp dựa trên đường dẫn của trình duyệt.

2.4.1.5. Routes và Route

Route cho phép xác định một định tuyến cụ thể và liên kết nó với component mà bạn muốn hiển thị.

Routes tương tự Route nhưng nó xác định nhiều định tuyến cùng một lúc và **Routes** thường được dùng bao quanh nhóm các Route như **Hình 2.4.3**.

2.4.1.6. Hooks và Context

```
▼ hooks
  JS QuestionContext.js
  JS useForm.js
  JS useStateContext.js
```

Hình 2.4.4: Thư mục Hooks chứa các Context

Hooks dùng để quản lý trạng thái gồm những hàm cho phép kết nối React State ví dụ như một số hàm useState, useEffect, useContext.

Context thường được sử dụng để chia sẻ dữ liệu, ví dụ như trạng thái giữa các component và được truyền từ cao xuống cấp thấp trong cây component.

2.4.2. Cấu hình và cách hoạt động ReactJS

```

8  const root = ReactDOM.createRoot(document.getElementById('root'));
9  const darkTheme = createTheme({
10    palette: {
11      mode: 'dark',
12    },
13  })
14  root.render(
15    <React.StrictMode>
16      <ContextProvider>
17        <ThemeProvider theme={darkTheme}>
18          <CssBaseline />
19          <App />
20        </ThemeProvider>
21      </ContextProvider>
22    </React.StrictMode>
23  );
24  reportWebVitals();

```

Hình 2.4.5: Cách hoạt động Index.js

Tạo một root bằng cách sử dụng “ReactDOM.createRoot” và đối tượng root này sẽ được hiện thị lên giao diện của React trên cây DOM. Ở phần “root.render” được dùng để xây dựng giao diện và bên trong đó gồm các component có từng chức năng riêng.

```

function App() {
  return (
    <QuestionProvider>
      <BrowserRouter>
        <Routes>
          <Route path="/" element={<Login />} />
          <Route path="/register" element={<Register />} />
          <Route path="/questions" element={<Question />} />
          <Route path="/CreateQuestion" element={<CreateQuestion />} />
          <Route path="/UpdateQuestion/:id" element={<UpdateQuestion />} />
          <Route path="/participantresult" element={<ParticipantResult />} />
          <Route path="/importquestion" element={<ImportQuestion />} />
          <Route element={<Authentication />}>
            <Route path="/" element={<Layout />}>
              <Route path="/quiz" element={<Quiz />} />
              <Route path="/result" element={<Result />} />
            </Route>
          </Route>
        </Routes>
      </BrowserRouter>
    </QuestionProvider>
  );
}
export default App;

```

Hình 2.4.6: Cách hoạt động App.js sau Index.js

Trong hàm “App” nó được dùng để liên kết và hiển thị các component lên Web, với App nó có thể chứa các Route để điều hướng người dùng sang trang khác trong ứng dụng.

2.4.3. Kết nối Api giữa ReactJS và C#

```
import axios from "axios";
export const ENDPOINTS = {
  participant: 'Participants',
  question: 'Questions',
  getanswer: 'Questions/GetAnswers',
  getAllQuestions: 'Questions/GetAll',
  postquestion: 'Questions/PostQuestions'
}
export const BASE_URL = 'https://localhost:7002/';
export const createAPIEndpoint = endpoint => {
  let url = BASE_URL + 'api/' + endpoint + '/';
  return {
    fetch: () => axios.get(url),
    fetchById: id => axios.get(url + id),
    post: newRecord => axios.post(url, newRecord),
    put: (id, updateRecord) => axios.put(url + id, updateRecord),
    delete: id => axios.delete(url + id),
  }
}
```

Hình 2.4.7: Cấu trúc axios cho API

Đầu tiên ta cần import “axios” giúp gửi các yêu cầu HTTP đến các địa chỉ cụ thể và xử lý phản hồi từ máy chủ.

“const ENDPOINTS” dùng để chỉ định đến đường dẫn cụ thể.

“const BASE_URL” đây là nơi mà tất cả các địa chỉ API sẽ được gửi đến và thường là tên miền hoặc địa chỉ IP của máy chủ.

“const createAPIEndpoint” Đây là một hàm tạo ra API có các đối số endpoint được truyền vào và có các phương thức để thực hiện như GET, POST, PUT, DELETE... và trả về các chức năng tương ứng như là thêm, sửa, xóa cho các đường dẫn tương ứng.

Chương 3. XÂY DỰNG WEBSITE THI TRẮC NGHIỆM ONLINE BẰNG REACTJS VÀ .NET CORE API

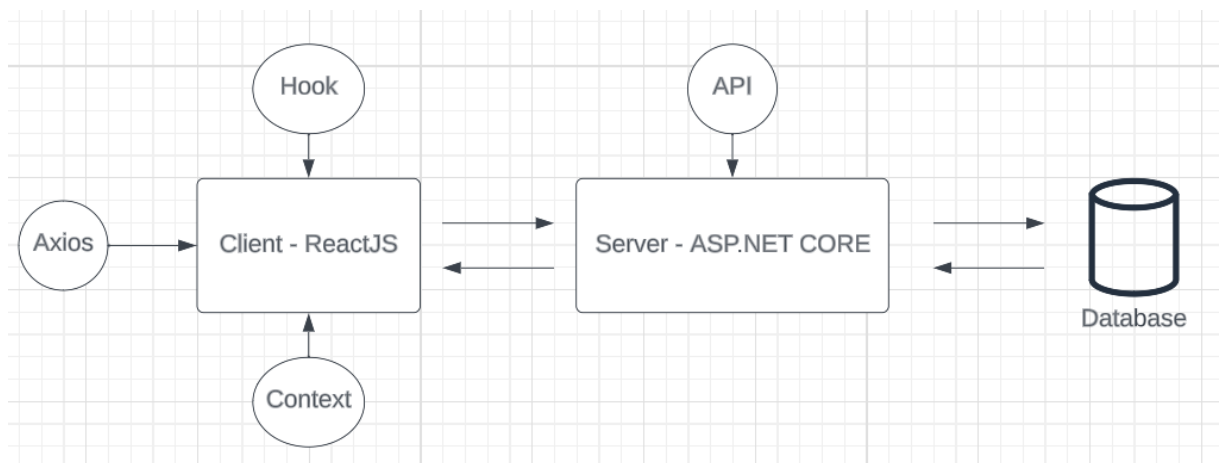
3.1. Giới thiệu hệ thống thi trắc nghiệm

Hệ thống Website thi trắc nghiệm này hỗ trợ cho người dùng có thể tham gia bài thi trực tuyến. Với mục đích kiểm tra kiến thức và đánh giá năng lực của từng người một cách trực quan. Hệ thống gồm các chức năng sau:

- Đăng nhập/ đăng ký tài khoản cá nhân theo người dùng.
- Thực hiện làm bài thi
- Xem kết quả khi hoàn thành bài thi
- Có thể thêm, sửa, xóa các câu hỏi
- Import dữ liệu câu hỏi từ file xls
- Export dữ liệu kết quả người dùng ra filexlsx

Khi sử dụng hệ thống này cho phép người dùng có thể tạo ra các câu hỏi nhanh chóng, không phải mất thời gian cho việc soạn trên giấy và in giấy, đối với người thi trắc nghiệm thì có thể tham gia vào bài thi một cách nhanh chóng, sau khi thực hiện trong bài thi kết quả sẽ được hiển thị ngay lập tức và lưu trữ điểm thi một cách nhanh chóng thuận tiện cho người chấm và người sử dụng.

3.2. Kiến trúc hệ thống



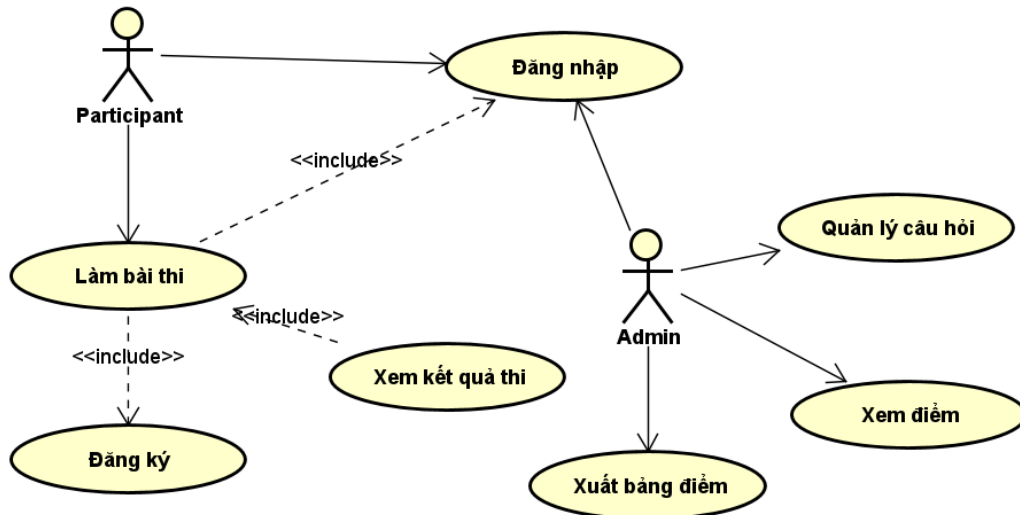
Hình 3.2.1: Kiến trúc hệ thống quản lý thi trắc nghiệm

Với kiến trúc và hệ thống ở **Hình 3.2.1** cho biết phía Server được xây dựng bởi ngôn ngữ C# và giao thức giao tiếp như HTTP với phía Client là ReactJS. Việc kết hợp hai công nghệ phổ biến này thì phía Server kết nối với CSDL để truy vấn, sànlọc dữ liệu

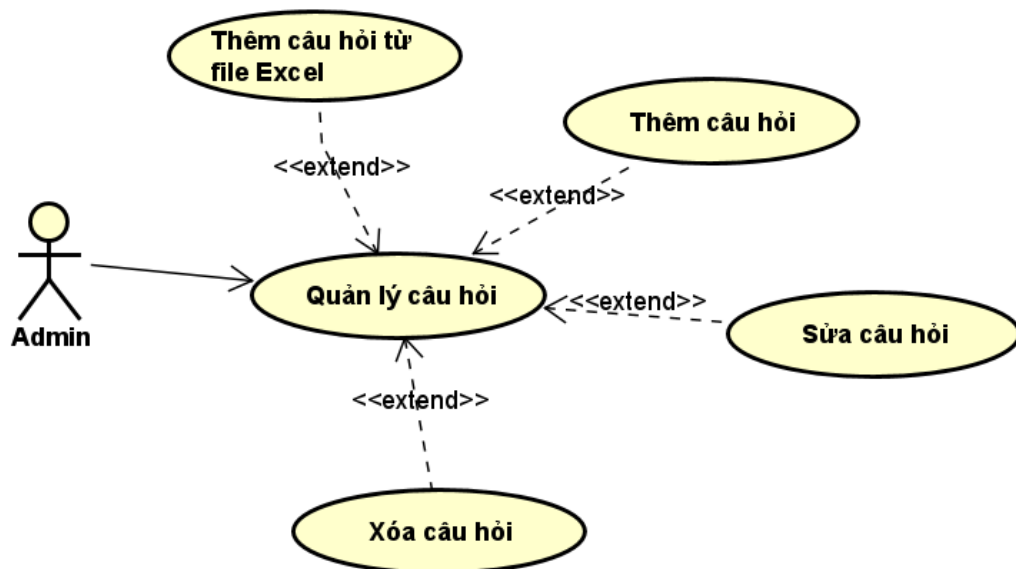
sau đó dùng API để trả request ngược lại cho phía Client nhận request và xử lý giao diện.

3.3. Phân tích và thiết kế hệ thống

3.3.1. Lược đồ UseCase



Lược đồ usecase 3.3.1.1: UseCase Tổng quát



Lược đồ usecase 3.3.1.2: UseCase Quản lý câu hỏi

3.3.2. Đặc tả UseCase

- UseCase “Đăng ký”

| Tiêu đề | Nội dung |
|------------------|--|
| UseCase ID: | UseCase 01 |
| UseCase Name: | Đăng ký |
| Vấn tắt: | UseCase này cho phép người dùng đăng ký tài khoản |
| Actor chính: | Người dùng |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc tài khoản phải có @gmail |
| Hậu điều kiện: | Đăng ký thành công |
| Luồng hoạt động: | 1. Chọn nút Register 2. Đăng ký tài khoản và mật khẩu 3. Bấm nút Đăng ký |
| Luồng thay thế: | 2.1 Nếu tài khoản không có @gmail thì báo lỗi. |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

Bảng 3.3.2-1: Bảng UseCase “Đăng ký”

Usecase “Đăng nhập”

| Tiêu đề | Nội dung |
|------------------|--|
| UseCase ID: | UseCase 02 |
| UseCase Name: | Đăng nhập |
| Vấn tắt: | UseCase này cho phép người dùng đăng nhập tài khoản |
| Actor chính: | Người dùng |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc phải có tài khoản |
| Hậu điều kiện: | Đăng nhập thành công |
| Luồng hoạt động: | 1. Nhập tài khoản + mật khẩu 2. Bấm nút đăng nhập |

| | |
|-----------------|--|
| Luồng thay thế: | 2.1 Tài khoản hoặc mật khẩu không đúng thì báo lỗi |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

Bảng 3.3.2-2: Bảng UseCase “Đăng nhập”

UseCase “Làm bài thi”

| Tiêu đề | Nội dung |
|------------------|--|
| UseCase ID: | UseCase 03 |
| UseCase Name: | Làm bài thi |
| Vấn tắt: | UseCase này cho phép người dùng đăng nhập vào và làm bài thi |
| Actor chính: | Người dùng |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc có tài khoản và đăng nhập |
| Hậu điều kiện: | Hiển thị kết quả bài thi |
| Luồng hoạt động: | 1. Chọn câu trả lời |
| Luồng thay thế: | Không có |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

Bảng 3.3.2-3: Bảng UseCase “Làm bài thi”

UseCase “Xem kết quả thi”

| Tiêu đề | Nội dung |
|-----------------|--|
| UseCase ID: | UseCase 04 |
| UseCase Name: | Xem kết quả thi |
| Vấn tắt: | UseCase này cho phép người xem kết quả sau khi thi |
| Actor chính: | Người dùng |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc có tài khoản và đăng nhập |

| | |
|------------------|---|
| Hậu điều kiện: | Xuất ra kết quả sau khi thi và lưu điểm |
| Luồng hoạt động: | 1. Xem câu trả lời đúng và sai |
| Luồng thay thế: | Không có |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

Bảng 3.3.2-4: Bảng UseCase “Xem kết quả thi”

UseCase “Quản lý câu hỏi”

| Tiêu đề | Nội dung |
|------------------|---|
| UseCase ID: | UseCase 05 |
| UseCase Name: | Quản lý câu hỏi |
| Vấn tắt: | UseCase này cho phép Admin thêm, sửa, xóa các câu hỏi |
| Actor chính: | Admin |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc có tài khoản Admin và đăng nhập |
| Hậu điều kiện: | Thêm, sửa, xóa, tạo câu hỏi thành công |
| Luồng hoạt động: | 1. Có thể tạo câu hỏi 2. Có thể chỉnh sửa câu hỏi 3. Có thể xóa câu hỏi 4. Có thể thêm câu hỏi từ file Excel |
| Luồng thay thế: | Không có |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

Bảng 3.3.2-5: Bảng UseCase “Quản lý câu hỏi”

UseCase “Xem điểm”

| Tiêu đề | Nội dung |
|---------------|--|
| UseCase ID: | UseCase 06 |
| UseCase Name: | Xem điểm |
| Vấn tắt: | UseCase này cho phép Admin xem điểm của người dùng |

| | |
|------------------|--|
| Actor chính: | Admin |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc có tài khoản Admin và đăng nhập |
| Hậu điều kiện: | Hiện kết quả bảng điểm của người dùng |
| Luồng hoạt động: | 1. Chọn xem điểm |
| Luồng thay thế: | Không có |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

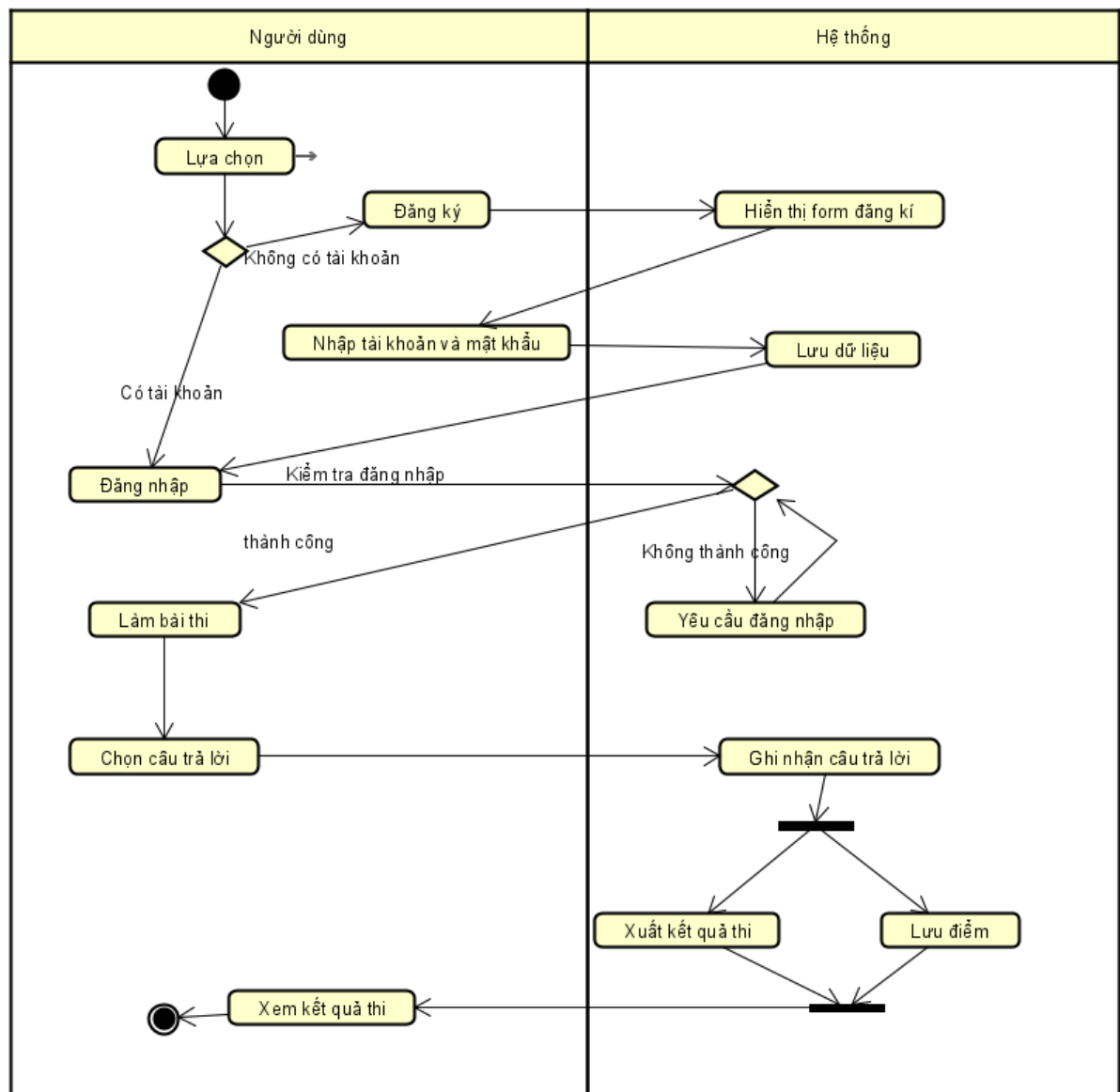
Bảng 3.3.2-6: Bảng UseCase “Xem điểm”

UseCase “Xuất bảng điểm”

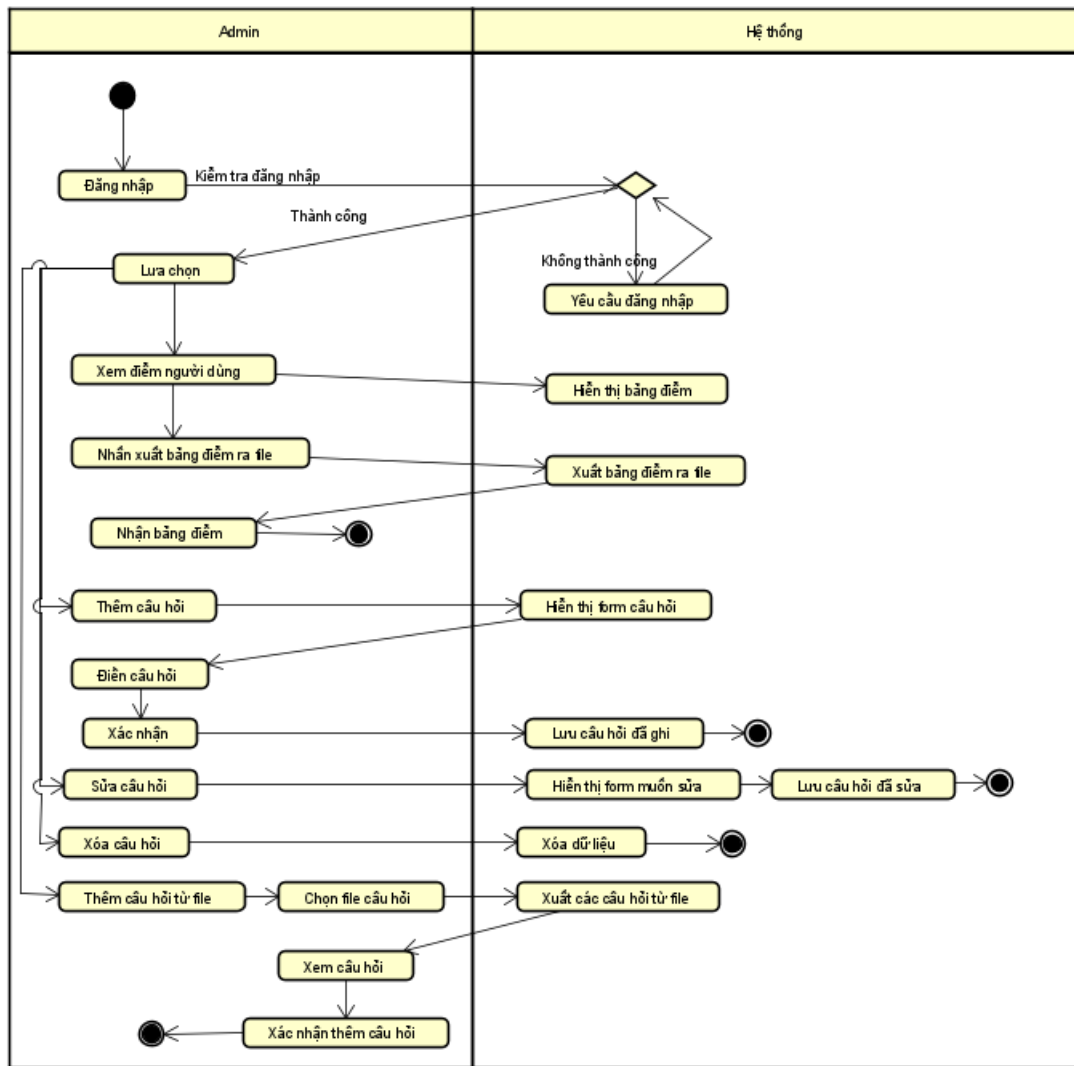
| Tiêu đề | Nội dung |
|------------------|--|
| UseCase ID: | UseCase 07 |
| UseCase Name: | Xuất bảng điểm |
| Vấn tắt: | UseCase này cho phép Admin có thể xuất bảng điểm ra file excel |
| Actor chính: | Admin |
| Actor phụ: | Không có |
| Tiền điều kiện: | Bắt buộc có tài khoản Admin và đăng nhập |
| Hậu điều kiện: | Xuất kết quả ra file excel điểm bảng điểm |
| Luồng hoạt động: | 1. Chọn mục xuất bảng điểm |
| Luồng thay thế: | Không có |
| Luồng ngoại lệ: | Mạng lỗi → không vào trang Web được |

Bảng 3.3.2-7: Bảng UseCase “Xuất bảng điểm”

3.3.3. Sơ đồ hoạt động



Hình 3.3.1: Sơ đồ hoạt động của Người dùng



Hình 3.3.2: Sơ đồ hoạt động của Admin

3.3.4. Database

Bảng Cơ sở dữ liệu “Questions”

| STT | Tên trường | Kiểu dữ liệu | Mô tả |
|-----|------------|---------------|----------------------|
| 1 | QnId | int | Mã câu hỏi |
| 2 | QnInWords | nvarchar(250) | Tên câu hỏi |
| 3 | ImageName | nvarchar(50) | Hình câu hỏi |
| 4 | Option1 | nvarchar(50) | Câu trả lời thứ nhất |
| 5 | Option2 | nvarchar(50) | Câu trả lời thứ hai |
| 6 | Option3 | nvarchar(50) | Câu trả lời thứ ba |

| | | | |
|---|---------|--------------|--------------------|
| 7 | Option4 | nvarchar(50) | Câu trả lời thứ tư |
| 8 | Answer | nvarchar(50) | Đáp án câu trả lời |

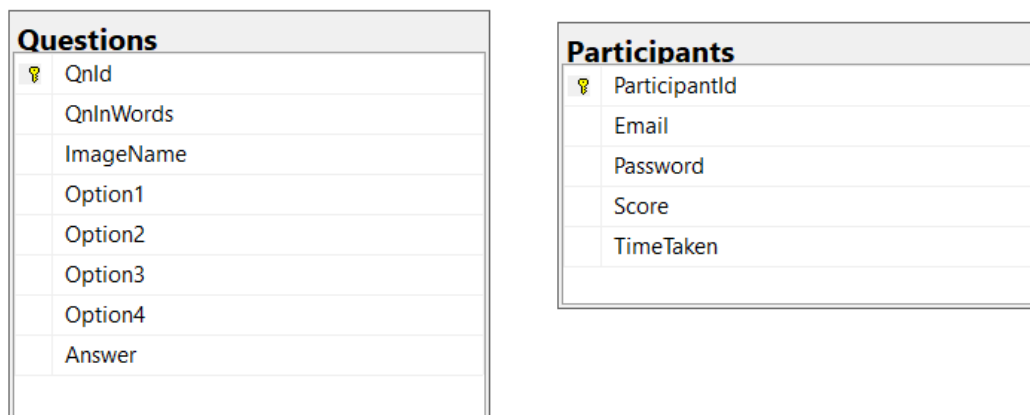
Bảng 3.3.4-1: Bảng “Questions”

Bảng cơ sở dữ liệu “Participants”

| STT | Tên trường | Kiểu dữ liệu | Mô tả |
|-----|---------------|--------------|---------------|
| 1 | ParticipantId | int | Mã người dùng |
| 2 | Email | nvarchar(50) | Email |
| 3 | Password | nvarchar(50) | Mật khẩu |
| 4 | Score | int | Lưu điểm |
| 5 | TimeTaken | int | Thời gian làm |

Bảng 3.3.4-2: Bảng “Participants”

Lược đồ cơ sở dữ liệu “QuizDB”



Hình 3.3.3: Cơ sở dữ liệu “QuizDB”

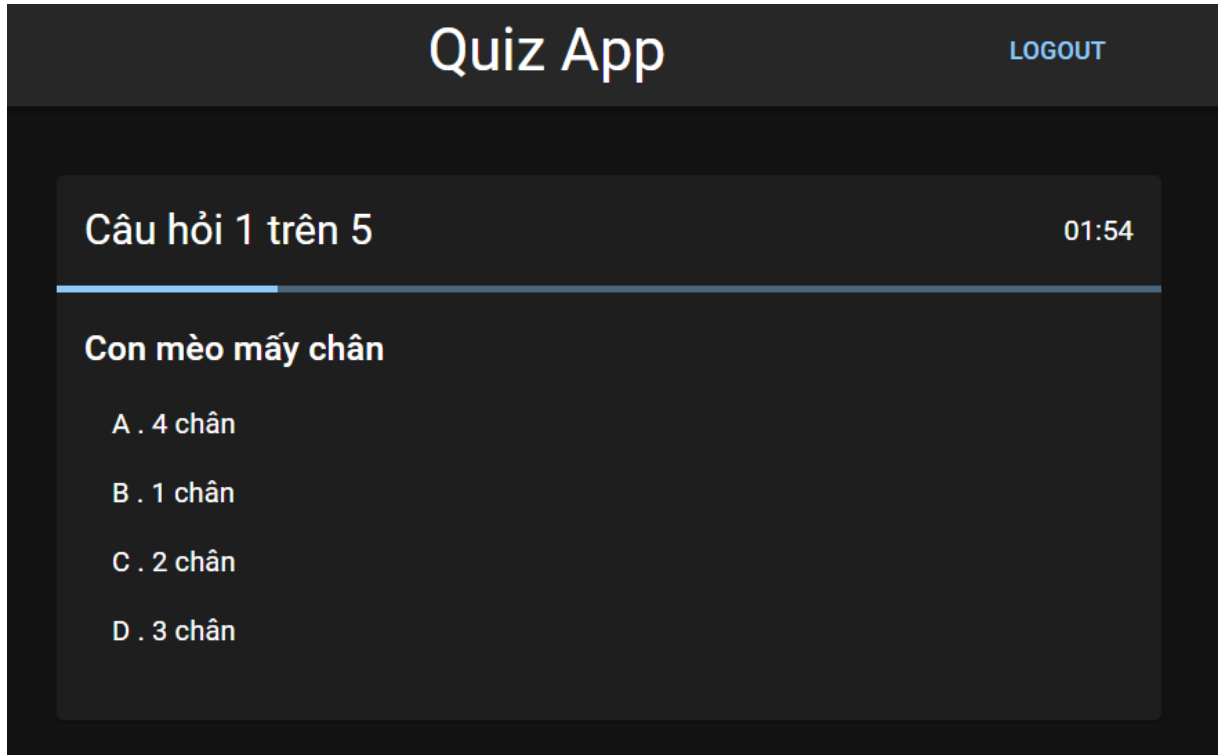
Ở bảng “Questions” người quản lý có thể điền câu hỏi mà mình muốn cũng như điền các câu trả lời và một đáp án đúng trong các câu trả lời đó để người tham gia bài thi đó có thể thực hiện.

Ở bảng “Participants” Cho phép lưu trữ thông tin của người tham gia bài thi bao gồm email, mật khẩu, số điểm và thời gian làm bài.

Ở đây không có mối quan hệ nào giữa hai bảng do bảng “Questions” hay bị thay đổi và câu trả lời cũng vậy nên tụi em đã gộp bảng lại nhằm mục đích dễ triển khai và quản lý, tăng hiệu suất truy xuất và xử lý dữ liệu nhanh.

3.4. Các nghiệp vụ đã phát triển

3.4.1. Làm bài thi



Hình 3.4.1: Giao diện làm bài thi

Về phần xử lý thời gian ta có:

```
let timer;
const time = 120;
const startTimer = () => {
  timer = setInterval(() => {
    setTimeLeft(prev => prev - 1)
  }, 1000)
}
```

Hình 3.4.2: Xử lý thời đếm lùi

Tạo biến “timer”, biến này dùng để lưu trữ biến đếm ngược trong hàm “startTimer”. còn biến “time” dùng để set thời gian bắt đầu trả lời cho từng câu hỏi là 120 giây. Để hiển thị các thông tin về câu hỏi và câu trả lời phải sử dụng qua API của C#.

```

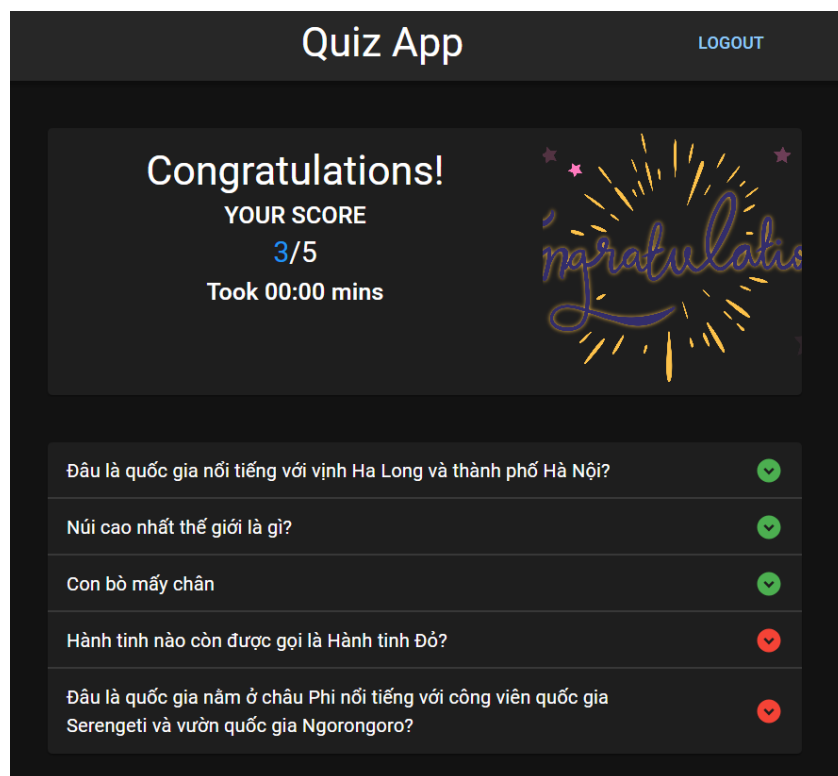
public class QuestionsController : ControllerBase
{
    private readonly TracNghiemDbContext _context;
    0 references
    public QuestionsController(TracNghiemDbContext context)
    {
        _context = context;
    }
    // GET: api/Questions
    [HttpGet]
    0 references
    public async Task<ActionResult<IEnumerable<Question>>> GetQuestions()
    {
        var random5Qns = await (_context.Questions.Select(x => new
        {
            QnId = x.QnId,
            QnInWords = x.QnInWords,
            ImageName = x.ImageName,
            Options = new string[] { x.Option1, x.Option2, x.Option3, x.Option4 },
            Answer = x.Answer
        })
            .OrderBy(y => Guid.NewGuid())
            .Take(5)
            ).ToListAsync();
        return Ok(random5Qns);
    }
}

```

Hình 3.4.3: Phương thức GET cho bảng “Questions”

Ở hàm này ta dùng phương thức GET là lấy dữ liệu, sau đó tạo ra biến “random5Qns” để lấy ngẫu nhiên 5 câu hỏi trong bảng “Questions” sau đó đưa API cho ReactJS theo **Hình 2.4.7** để hiển thị giao diện.

3.4.2. Xuất kết quả



Hình 3.4.4: Giao diện xuất kết quả sau khi làm bài

Xử lý phân so sánh giữa kết quả người dùng chọn với đáp án trong CSDL.

```
useEffect(() => {
  const ids = context.selectedOptions.map((x) => x.qnId)
  createAPIEndpoint(ENDPOINTS.getanswer)
    .post(ids)
    .then((res) => {
      const qna = context.selectedOptions.map((x) => ({
        ...x,
        ...res.data.find((y) => y.qnId === x.qnId),
      }));
      setQnAnswers(qna);
      caculator(qna);
    })
    .catch((err) => console.log(err));
}, []);
```

Hình 3.4.5: Hàm so sánh kết quả trả lời

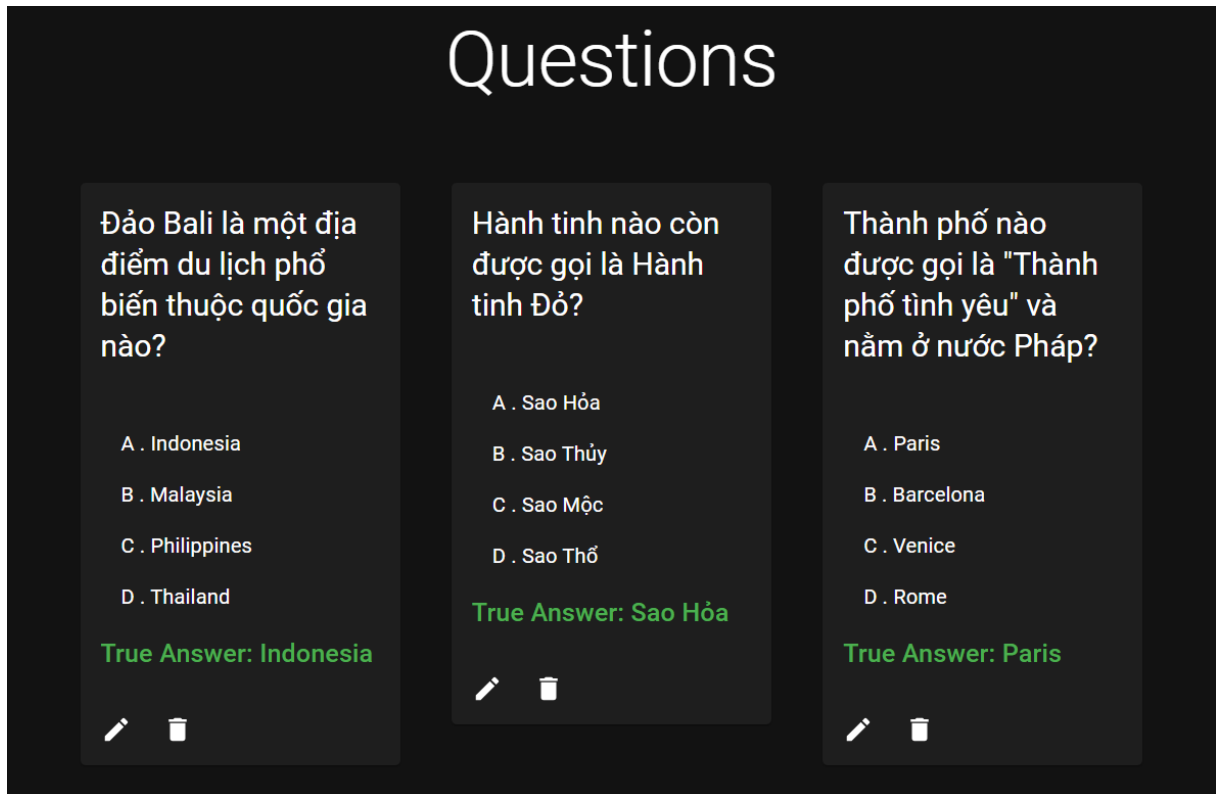
Đầu tiên ta tạo một mảng để chứa các danh sách từ các câu hỏi mà người dùng trả lời. Sau đó ta dùng API để lấy dữ liệu các câu hỏi có trong mảng trên từ CSDL. Bước tiếp là những câu hỏi có chung mã sẽ được so sánh với các mã đã hiển thị trong bài làm. Cuối cùng là so sánh kết quả người dùng chọn có trùng với đáp án có trong CSDL hay không.

```
const caculator = (qna) => {
  let tempScore = qna.reduce((acc, curr) => {
    return curr.answer === curr.selected ? acc + 1 : acc;
  }, 0);
  setScore(tempScore);
}
```

Hình 3.4.6: Hàm tính điểm câu trả lời đúng

Với hàm này sẽ tạo một biến cho phép cộng dồn số lần trả lời đúng của người tham gia bài thi như **Hình 3.4.5**.

3.4.3. Xem các câu hỏi



Hình 3.4.7: Giao diện hiện các câu hỏi bên Admin

```
const fetchQuestions = () => {
  createAPIEndpoint(ENDPOINTS.getAllQuestions)
    .fetch()
    .then(res => {
      console.log(res.data)
      setQuestions(res.data);
    })
    .catch(err => {
      console.log(err);
    });
};
```

Hình 3.4.8: Dùng phương GET để hiển thị các câu hỏi

Tạo một hàm và gọi API dạng GET để lấy thông tin toàn bộ câu hỏi có trong CSDL và dùng hàm “setQuestions” để cập nhật giá trị đó lên giao diện.

3.4.4. Thêm câu hỏi

Question *

Option1 *

Option2 *

Option3 *

Option4 *

Answer *

> LƯU

Hình 3.4.9: Giao diện thêm câu hỏi

```
const createQuestion = ({ qnInWords, option1, option2, option3, option4, answer }) => {
  const newQuestion = {
    qnInWords,
    option1,
    option2,
    option3,
    option4,
    answer
  };

  createAPIEndpoint(ENDPOINTS.question)
    .post(newQuestion)
    .then(
      res => {
        let data = res.data;
        setQuestions(prevQuestions => [...prevQuestions, data]);
        fetchQuestions();
      }
    )
    .catch(error => {
      console.error('Error:', error);
      // Xử lý lỗi nếu có
    });
}
```

Hình 3.4.10: Dùng phương thức POST để thêm dữ liệu

Tạo một đối tượng dùng để lưu các thông tin của câu hỏi được điền trên form, sau đó gọi API theo dạng POST của câu hỏi để truyền đối tượng vừa tạo vào. Khi gọi API thành công thì dữ liệu được truyền vào cơ sở dữ liệu để lưu trữ. Cuối cùng các dữ liệu của câu hỏi vừa được thêm sẽ được hiển thị qua phần hiển thị các câu hỏi qua **Hình 3.4.8**.

3.4.5. Sửa, xóa câu hỏi

Dựa vào **Hình 3.4.7** ở phía dưới của các câu hỏi có hình chỉnh sửa và xóa câu hỏi. Khi nhấn vào chỉnh sửa câu hỏi sẽ hiện thị giao diện và phần xử lý chức năng.

Question *

Đâu là quốc gia nổi tiếng với vịnh Ha Long và thành phố Hà Nội?

Option1 *

Cambodia

Option2 *

Laos

Option3 *

Vietnam

Option4 *

Thailand

Answer *

Vietnam

➤ SUBMIT

Hình 3.4.11: Giao diện chỉnh sửa câu hỏi

```
const updateQuestion = (id, value) => {
  createAPIEndpoint(ENDPOINTS.question)
    .put(id, value)
    .then(
      res => {
        fetchQuestions();
      }
    )
    .catch(error => {
      console.error(error);
      // xử lý lỗi nếu có
    });
}
```

Hình 3.4.12: Dùng phương thức PUT để cập nhật câu hỏi

Tạo một hàm nhận vào id của câu hỏi hiện đang có và giá trị thay đổi của câu hỏi. Ta gọi API dạng PUT để có thể cập nhật dữ liệu. Sau khi thành công thì chuyển về trang hiển thị và cập nhật dữ liệu lại sau khi đã thay đổi. Tiếp đến phần xóa dữ liệu được thể hiện qua phần xử lý sau.

```
const deleteQuestion = (id) => {
  createAPIEndpoint(ENDPOINTS.question)
    .delete(id)
  setQuestions(questions.filter((question) => question.qnId !== id))
}
```

Hình 3.4.13: Dùng phương thức Delete để xóa câu hỏi

Ở hàm xóa này ta truyền API dạng Delete để xóa theo mã câu hỏi. Sau khi xóa dữ liệu thành công, ta lọc lại mã của các câu hỏi để cập nhật lại giao diện với câu hỏi vừa mới xóa.

3.4.6. Import file

| Question | Option1 | Option2 | Option3 | Option4 | Answer |
|---|------------------|-------------|---------|--------------|-------------|
| Hành tinh nào còn được gọi là Hành tinh Đỏ? | Sao Hỏa | Sao Thủy | Sao Mộc | Sao Thổ | Sao Hỏa |
| Núi cao nhất thế giới là gì? | Núi Killimanjaro | Núi Everest | K2 | Núi McKinley | Núi Everest |

Hình 3.4.14: Giao diện Import file

```
const fileType = ['application/vnd.ms-excel'];
const handleFile = (e) => {
  let selectedFile = e.target.files[0];
  if (selectedFile) {
    if (selectedFile && fileType.includes(selectedFile.type)) {
      let reader = new FileReader();
      reader.readAsArrayBuffer(selectedFile);
      reader.onload = (e) => {
        setExcelFileError(null);
        setExcelFile(e.target.result);
      }
    }
    else {
      setExcelFileError('Please select only excel file types');
      setExcelFile(null);
    }
  }
  else {
    console.log('plz select your file');
  }
}
```

Hình 3.4.15: Xử lý Import file

Đầu tiên tạo mảng chứa các tệp được cho Excel. Khi thực hiện hàm, lúc đầu sẽ kiểm tra có phải là một tệp file không, nếu không phải là file Excel thì hãy chọn lại đúng tệp file. Ngược lại là tệp file sẽ bắt đầu đọc dữ liệu từ file Excel đã được chọn và hiện thị nút update.

```
const handleUpdate = () => {
  createAPIEndpoint(ENDPOINTS.postquestion)
    .post(excelData)
    .then(response => {
      console.log(response.data),
      navigate('/Questions')
    })
    .catch(error => {
      console.error('Error:', error);
      // xử lý lỗi nếu có
    });
};
```

Hình 3.4.16: Dùng phương thức POST để thêm dữ liệu từ file

Sau khi người dùng bấm vào nút update hàm trong Hình 3.4.16 sẽ bắt đầu đẩy dữ liệu vào CSDL và sau khi đẩy thành công thì sẽ chuyển về trang hiện thị các câu hỏi để có thể xem câu hỏi vừa được thêm từ file.

3.4.7. Export file

Score

[Export Participant Data](#)

| ID | Email | Password | Score | Time Taken |
|----|-----------------|----------|-------|------------|
| 1 | nhan@gmail.com | 123456 | 0 | 0 |
| 2 | u@gmail.com | 123456 | 3 | 0 |
| 3 | u@gmail.com | 12 | 0 | 0 |
| 4 | adsa@gmail.com | 123456 | 0 | 0 |
| 5 | khe@gmail.com | 1234567 | 0 | 0 |
| 6 | quynh@gmail.com | 123456 | 1 | 0 |
| 7 | Admin@gmail.com | 123456 | 0 | 0 |

Rows per page: 100 ▾
1–7 of 7
< >

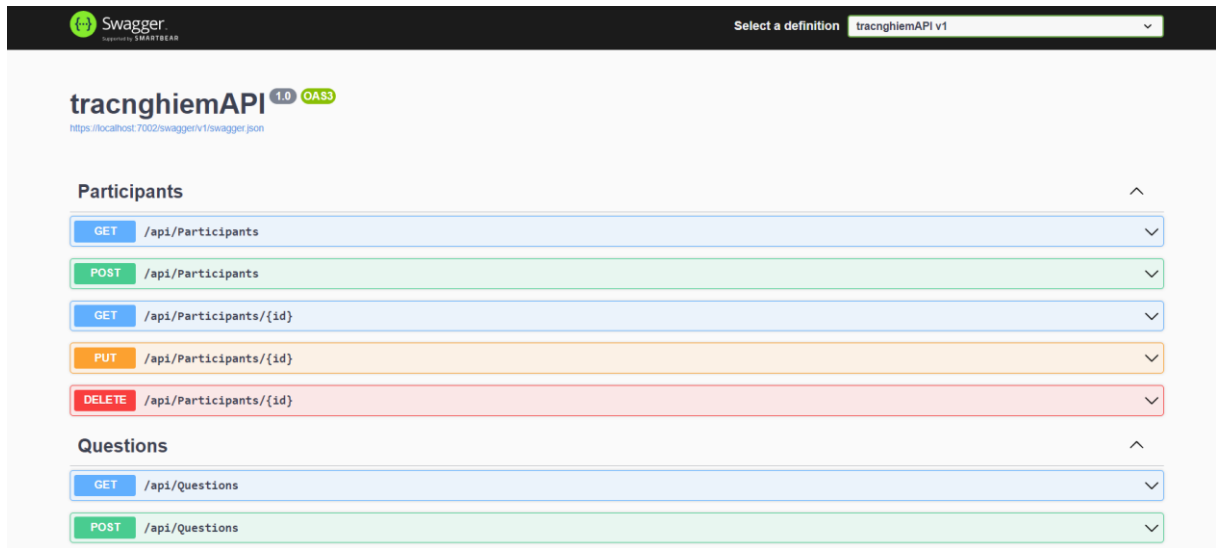
Hình 3.4.17: Giao diện xem kết quả từ người dùng

```
const prepareDataForCSV = () => {
  return rows.map((row) => ({
    id: row.participantId,
    Email: row.email,
    Password: row.password,
    Score: row.score,
    TimeTaken: row.timeTaken
  }));
};
return (
  <div style={{ height: 500, width: "90%", margin: "auto" }}>
    <Typography variant='h2' align='center' gutterBottom>Score</Typography>
    <CSVLink data={prepareDataForCSV()} className="btn btn-danger mb-3">Export Participant Data</CSVLink>
    <DataGrid rows={rows} columns={columns} getRowId={(row) => row.participantId}>
    </DataGrid>
  </div>
)
```

Hình 3.4.18: Xử lý xuất dữ liệu ra file

Khi nhấn vào dòng chữ “Export Participant Data” thì sẽ xuất ra file Excel với dữ liệu đã được định nghĩa và chuẩn bị ở tệp CSV trong hàm “prepareDataForCSV”. Qua đó file Excel sẽ được lưu vào thư mục của bạn.

3.4.8. Minh họa API



Hình 3.4.19: Giao diện các API bên C#

```
[
  {
    "qnId": 34,
    "qnInWords": "Đâu là quốc gia nổi tiếng với các tháp cổ của nền văn minh Maya?",
    "imageName": null,
    "options": [
      "Egypt",
      "Cambodia",
      "Mexico",
      "Greece"
    ],
    "answer": "Mexico"
  },
  {
    "qnId": 33,
    "qnInWords": "Địa danh nào nổi tiếng với tượng chúa Kitô Phục Sinh và nằm ở Brazil?",
    "imageName": null,
    "options": [
      "Buenos Aires",
      "Sao Paulo",
      "Rio de Janeiro",
      "Lima"
    ],
    "answer": "Rio de Janeiro"
  },
  {

```

Hình 3.4.20: Minh họa dữ liệu sau khi gọi API của “Questions”

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Từ lúc thực hiện đồ án chúng em đã nghiên cứu, tìm hiểu về cách thức hoạt động của những hệ thống web làm trắc nghiệm với sự hỗ trợ của thầy Phan Trần Minh Khuê chúng em đã xây dựng được một số chức năng để có thể cho người dùng sử dụng như là Chọn, thêm, sửa, xóa các câu hỏi, những câu hỏi được chọn một cách ngẫu nhiên với thứ tự các câu trả lời thay đổi theo mỗi lần làm bài, khả năng tiện dụng khi thêm câu hỏi từ file Excel và cò xuất bảng điểm một cách nhanh chóng.

Phần mềm của chúng em còn một số chức năng chưa có như định danh quét mã mặt người dùng khi làm bài để tránh gian lận, thêm dữ liệu câu hỏi từ file Excel bắt buộc phải là đuôi .xls và định dạng bị ràng buộc. Phần mềm phải có ít nhất năm câu hỏi trong CSDL.

4.2. Hướng phát triển

4.2.1. Ưu điểm

Với phần mềm chúng em đã xây dựng thì chúng em nhận thấy một số ưu điểm như sau giao diện thì dễ nhìn dễ hiểu và dễ sử dụng, những câu hỏi được bốc ngẫu nhiên theo từng người dùng gia với câu trả lời được xáo trộn vị trí, sau khi hoàn tất bài thi người dùng ngay lập tức nhận được kết quả và biết những câu trả lời sai. Chức năng quản lý câu hỏi và quản lý điểm người dùng trực quan.

4.2.2. Nhược điểm

Phần mềm của chúng em vẫn còn một số những chức năng chưa hoàn thiện và cần được cải thiện vì thế chúng em sẽ cố gắng khắc phục trong tương lai để có thể sửa chữa các chức năng sau các câu hỏi có thể được chỉnh sửa tùy ý mà không bị ràng buộc định dạng, thêm tính năng nhận diện khuôn mặt để tránh gian lận, có thể phân quyền được các vai trò, thêm được nhiều chức năng hơn ở phía người tham gia bài thi.

TÀI LIỆU THAM KHẢO

- [1] Trumhemcut, “C# là gì ? , ” 20/10/2005. [Trực tuyến]. Địa chỉ: <https://w.wiki/7jgJ>. [Truy cập 11/10/2023].
- [2] Lvphuong, “.NET CORE là gì?, ” 04/01/2022. [Trực tuyến]. Địa chỉ: <https://w.wiki/7jgk>. [Truy cập 11/10/2023].
- [3] Topdev.vn, “RESTful API là gì ? , ” 15/11/2016. [Trực tuyến]. Địa chỉ: <https://topdev.vn/blog/restful-api-la-gi/>. [Truy cập 11/10/2023].
- [4] Nguyễn Thị Mai Trang, “Tài liệu tham khảo môn ứng dụng web của trường đại học Mở TPHCM, ”17/02/2021.
- [5] apotech.fpt.edu.vn, “ReactJS là gì?, ”. [Trực tuyến]. Địa chỉ: <https://aptech.fpt.edu.vn/reactjs-la-gi.html>