**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: HỆ THỐNG QUẢN LÝ CÁC KHÓA HỌC**

**TRỰC TUYẾN LMS**

Giáo viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: NGUYỄN HỮU TIN

Lớp: CÔNG NGHỆ THÔNG TIN

Khóa: K59

TP. Hồ Chí Minh, năm 2022

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: HỆ THỐNG QUẢN LÝ CÁC KHÓA HỌC**

**TRỰC TUYẾN LMS**

Giáo viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: NGUYỄN HỮU TIN

Lớp: CÔNG NGHỆ THÔNG TIN

Khóa: K59

TP. Hồ Chí Minh, năm 2022

|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  **PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

|  |  |
| --- | --- |
| **Mã sinh viên:** 5951071106 | **Họ và Tên:** Nguyễn Hữu Tin |
| **Khóa:** K59 | **Lớp:** Công nghệ thông tin |

1. **Tên đề tài:** Hệ thống quản lý các khóa học trực tuyến LMS
2. **Mục đích yêu cầu**
3. **Mục đích**

Xây dựng hệ thống quản lý các khóa học về lập trình nhằm phục vụ cho việc dạy và học lập trình trực tuyến

1. **Yêu cầu**

* **Yêu cầu chức năng:** Hệ thống gồm ba nhóm chính: admin, instructor, student.
* **Admin:**
* Quản lý thống kê các khóa học, các instructor và các student.
* Phân quyền user
* Quản lý thanh toán cho instructor
* **Instructor:**
* Tạo các khóa học
* Quản lý thống kê các khóa học đã tạo
* Quản lý các bài đánh giá
* Tạo cuộc trò chuyện với các student
* Tạo buổi live Stream giảng dạy
* **Student:**
* Xem thông tin các khóa học
* Đăng ký, mua các khóa học
* Xem video, tài liệu, nội dung bài giảng của các khóa học đã mua
* Đánh giá, bình luận các khóa học
* **Yêu cầu phi chức năng**
  + Tốc độ: Nhận truy cập nhanh, truy xuất dữ liệu nhanh
  + Giao diện: Thân thiện với người sử dụng

1. **Nội dung đề tài**

* Tổng quan bài toán
* Khảo sát bài toán
* Phân tích và thiết kế hệ thống
* Lập trình xây dựng hệ thống quản lý các khóa học về lập trình trực tuyến
* Kiểm thử và chạy thực nghiệm

1. **Công nghệ, công cụ và ngôn ngữ lập trình**: API, C#, Angular/ Reactjs, .NET Core, Entity Frame work, Microsoft SQL Server, Visual studio.
2. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

Xây dựng được hệ thống quản lý các khóa học với các chức năng cơ bản để phục vụ cho việc tự học lập trình.

1. **Kế hoạch thực hiện**

* Tuần 1-2: Thiết kế và xây dựng cơ sở dữ liệu
* Tuần 2-3-4: Xây dựng cấu trúc hệ thống và API
* Tuần 5-6-7: Thiết kế xây dựng giao diện người dùng
* Các tuần cuối: Hoàn thiện các chức năng, kiểm thử hệ thống.

1. **Giáo viên và cán bộ hướng dẫn**

Họ tên: ThS. TRẦN PHONG NHÃ

Đơn vị công tác: Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

Điện thoại: 0906 761 014 Email: tpnha@utc2.edu.vn

|  |  |
| --- | --- |
| **Ngày ...... tháng ...... năm 2021**  **Trưởng BM Công nghệ Thông tin** | **Đã giao nhiệm vụ TKTN**  **Giảng viên hướng dẫn**  **Trần Phong Nhã** |

# LỜI CẢM ƠN

Lời nói đầu tiên, em xin kính gửi lời cảm ơn chân thành nhất tới Quý thầy cô trong Bộ môn Công Nghệ Thông Tin, cũng như Ban Giám Hiệu Trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh, đã cho phép em có thể thực hiện đề tài tốt nghiệp: XÂY DỰNG HỆ THỐNG CÁC KHÓA HỌC TRỰC TUYẾN LMS.

Trong quá trình thực hiện đề tài, em đã học hỏi và tích lũy được rất nhiều kiến thức cũng như kinh nghiệm quý báu. Bên cạnh đó, để hoàn thành được nhiệm vụ được giao này ngoài sự nỗ lực không ngừng của bản thân còn có sự hướng dẫn nhiệt tình từ thầy Trần Phong Nhã, em xin đặc biệt gửi lời cám ơn chân thành nhất tới thầy vì đã giúp đỡ em trong quá trình thực hiện đề tài.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài, nhưng chắc chắn rằng sẽ khó tránh khỏi những thiếu sót. Em rất mong nhận được những sự đánh giá, góp ý của Quý thầy cô để em có thể rút ra cho mình những bài học, kinh nghiệm quý báu.

Sau cùng, em cũng không biết nói gì hơn ngoài kính chúc Quý thầy cô trong Bộ môn Công Nghệ Thông Tin thật dồi dào sức khỏe và ngày càng gặt hái được nhiều thành công hơn nữa trong cuộc sống cũng như trong sự nghiệp giảng dạy của mình. Em xin chân thành cảm ơn !

***TP. Hồ Chí Minh, ngày … tháng … năm 2022***

**Sinh viên thực hiện**

**Nguyễn Hữu Tin**

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm 2022***  **Giảng viên hướng dẫn**  **Trần Phong Nhã** |

# MỤC LỤC

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP i](#_Toc106454077)

[LỜI CẢM ƠN iii](#_Toc106454078)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iv](#_Toc106454079)

[MỤC LỤC v](#_Toc106454080)

[BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ vii](#_Toc106454081)

[CHƯƠNG 1. MỞ ĐẦU 1](#_Toc106454082)

[1.1. Tổng quan về đề tài 1](#_Toc106454083)

[1.2. Mô tả chi tiết đề tài 1](#_Toc106454084)

[1.3. Công cụ sử dụng 2](#_Toc106454085)

[1.4. Cấu trúc báo cáo đồ án tốt nghiệp 2](#_Toc106454086)

[1.4.1. Chương 1: Mở đầu 2](#_Toc106454087)

[1.4.2. Chương 2: Cơ sở lý thuyết 2](#_Toc106454088)

[1.4.3. Chương 3: Phân tích và thiết kế 2](#_Toc106454089)

[1.4.4. Chương 4: Triển khai chương trình 2](#_Toc106454090)

[1.4.5. Chương 5: Kết quả - Kiến nghị 2](#_Toc106454091)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 3](#_Toc106454092)

[2.1. Tổng quan về ngôn ngữ lập trình C#. 3](#_Toc106454093)

[2.2. Tổng quan về SQL. 3](#_Toc106454094)

[2.3. Tìm hiểu về Entity Framework 4](#_Toc106454095)

[2.4. Tìm hiểu về ASP.NET Core. 6](#_Toc106454096)

[2.5. Tìm hiểu về Web API. 7](#_Toc106454097)

[2.6. Tìm hiểu về các Design pattern. 9](#_Toc106454098)

[2.6.1. Repository 9](#_Toc106454099)

[2.6.2. Unit of work 10](#_Toc106454100)

[2.6.3. Dependency injection. 10](#_Toc106454101)

[2.7. Tìm hiểu về HTML 12](#_Toc106454102)

[2.7.1. Giới thiệu 12](#_Toc106454103)

[2.7.2. Vai trò của HTML. 12](#_Toc106454104)

[2.8. Tìm hiểu về CSS. 13](#_Toc106454105)

[2.8.1. Giới thiệu về CSS 13](#_Toc106454106)

[2.8.2. Ưu điểm của CSS. 14](#_Toc106454107)

[2.9. Tìm hiểu về Reactjs. 14](#_Toc106454108)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ 16](#_Toc106454109)

[3.1. Đặt vấn đề 16](#_Toc106454110)

[3.2. Yêu cầu hệ thống 16](#_Toc106454111)

[3.2.1. Yêu cầu chức năng 16](#_Toc106454112)

[3.2.2. Mô tả hệ thống 17](#_Toc106454113)

[3.3. Thiết kế hệ thống 18](#_Toc106454114)

[3.3.1. Mô hình thực thể kết hợp ERD 18](#_Toc106454115)

[3.3.2. Sơ đồ phân rã chức năng 19](#_Toc106454116)

[3.3.3. Sơ đồ UseCase tổng quát 19](#_Toc106454117)

[3.3.4. Sơ đồ hoạt động 20](#_Toc106454118)

[3.3.5. Database Diagram 24](#_Toc106454119)

[CHƯƠNG 4. TRIỂN KHAI CHƯƠNG TRÌNH 25](#_Toc106454120)

[4.1. Vận hành hệ thống 25](#_Toc106454121)

[4.2. Xây dựng giao diện. 25](#_Toc106454122)

[4.2.1. Giao diện đăng nhập 25](#_Toc106454123)

[4.2.2. Giao diện Admin 26](#_Toc106454124)

[4.2.3. Giao diện Instructor 35](#_Toc106454125)

[4.2.4. Giao diện Student 40](#_Toc106454126)

[CHƯƠNG 5. KẾT QUẢ - KIẾN NGHỊ 45](#_Toc106454127)

[5.1. Kết quả đạt được 45](#_Toc106454128)

[5.2. Hạn chế 45](#_Toc106454129)

[5.3. Hướng phát triển 45](#_Toc106454130)

[TÀI LIỆU THAM KHẢO 46](#_Toc106454131)

# BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ

[Hình 2.1. Cấu trúc của Entity Framework. 5](#_Toc106454132)

[Hình 2.2. Hoạt động của Resful API. 8](#_Toc106454133)

[Hình 3.1. Mô hình thực thể ERD. 18](#_Toc106454134)

[Hình 3.2. Mô hình phân rã chức năng 19](#_Toc106454135)

[Hình 3.3. Mô hình usecase tổng quát. 19](#_Toc106454136)

[Hình 3.4. Mô hình hoạt động mua khóa học. 20](#_Toc106454137)

[Hình 3.5. Mô hình hoạt động thêm khóa học mới. 21](#_Toc106454138)

[Hình 3.6. Mô hình hoạt động đăng kí tài khoản giảng viên. 22](#_Toc106454139)

[Hình 3.7. Mô hình hoạt động phê duyệt khóa học 23](#_Toc106454140)

[Hình 3.8. Sơ đồ class 24](#_Toc106454141)

[Hình 4.1. Trang đăng nhập 25](#_Toc106454142)

[Hình 4.2. Trang đăng nhập thất bại. 26](#_Toc106454143)

[Hình 4.3. Giao diện admin. 26](#_Toc106454144)

[Hình 4.4. Thống kê doanh thu. 27](#_Toc106454145)

[Hình 4.5. Thống kê khóa học. 27](#_Toc106454146)

[Hình 4.6. Giao diện thêm người dùng mới. 28](#_Toc106454147)

[Hình 4.7. Giao diện sửa thông tin user. 28](#_Toc106454148)

[Hình 4.8. Giao diện thêm language. 29](#_Toc106454149)

[Hình 4.9. Giao diện sửa language. 29](#_Toc106454150)

[Hình 4.10. Giao diện Category. 30](#_Toc106454151)

[Hình 4.11. Giao diện SubCategory. 30](#_Toc106454152)

[Hình 4.12. Giao diện Student. 31](#_Toc106454153)

[Hình 4.13. Giao diện khi Block thành công. 31](#_Toc106454154)

[Hình 4.14. Giao diện Instructor. 32](#_Toc106454155)

[Hình 4.15. Giao diện thông tin chi tiết của Instructor. 32](#_Toc106454156)

[Hình 4.16. Giao diện Payout. 33](#_Toc106454157)

[Hình 4.17. Các lựa chọn khóa học 33](#_Toc106454158)

[Hình 4.18. Giao diện khóa học. 34](#_Toc106454159)

[Hình 4.19. Giao diện thông tin chi tiết của khóa học. 34](#_Toc106454160)

[Hình 4.20. Giao diện trang Dashboard của Instructor. 35](#_Toc106454161)

[Hình 4.21. Các khóa học mới gửi. 35](#_Toc106454162)

[Hình 4.22. Giao diện danh sách các khóa học của Instructor. 36](#_Toc106454163)

[Hình 4.23. Giao diện tạo khóa học. 36](#_Toc106454164)

[Hình 4.24. Giao diện thêm background cho khóa học. 37](#_Toc106454165)

[Hình 4.25. Giao diện tạo nội dung của khóa học. 37](#_Toc106454166)

[Hình 4.26. Upload video bài giảng và lưu bài học. 38](#_Toc106454167)

[Hình 4.27. Giao diện upload thành công. 38](#_Toc106454168)

[Hình 4.28. Giao diện trang Earning. 39](#_Toc106454169)

[Hình 4.29. Giao diện trang Payout của Instructor. 39](#_Toc106454170)

[Hình 4.30. Giao diện trang chủ. 40](#_Toc106454171)

[Hình 4.31. Giao diện trang thông tin khóa học. 40](#_Toc106454172)

[Hình 4.32. Giao diện trang thanh toán online. 41](#_Toc106454173)

[Hình 4.33. Các khóa học đã mua. 42](#_Toc106454174)

[Hình 4.34. Video bài giảng. 42](#_Toc106454175)

[Hình 4.35. Giao diện bình luận. 43](#_Toc106454176)

[Hình 4.36. Danh sách các giảng viên. 43](#_Toc106454177)

[Hình 4.37. Thông tin chi tiết cảu giảng viên. 44](#_Toc106454178)

# MỞ ĐẦU

## Tổng quan về đề tài

Hiện nay, các khóa học được mở ra nhằm nâng cao kỹ năng nghề nghiệp hoặc chuyên môn nào đó cho học viên. Người học đăng ký các khóa học này với mong muốn hoàn thiện các kỹ năng, kiến thức còn thiếu để có cơ hội tăng lương, lên chức hoặc chỉ đơn giản là làm tốt hơn chuyên môn của mình.

Các khóa học này xuất hiện ngày càng nhiều nhằm đáp ứng nhu cầu học ( của mọi đối tượng bao gồm cả già lẫn trẻ, cán bộ làm trong các cơ quan nhà nước đến người làm tư nhân, làm tự do, còn đi học hay đã về hưu…). Mục đích tham gia các khóa học cũng rất đa dạng, từ việc học để hoàn thiện kỹ năng, nâng cao chuyên môn đến học để nâng lương, thăng chức hoặc đi du học,… Tuy nhiên cũng chính vì sự xuất hiện ngày càng nhiều và tràn lan của đủ các loại khóa học mà việc thu hút học viên dường như khó khăn hơn.

Với mong muốn nhằm cải thiện các vấn đề nắm bắt thông tin khóa học, đăng ký khoá học tại trung tâm đào tạo được thực hiện nhanh hơn, mang tính chính xác cao. Từ thực tế đó, em xây dựng hệ thống khóa học trực tuyến.

Hệ thống khoá học trực tuyến là hệ thống cho phép người dùng thực hiện một cách nhanh chóng, chính xác việc nhập dữ liệu, lưu trữ, cập nhật thông tin các khoá học tại trung tâm đào tạo, các khóa học online thông qua các video bài giảng. Hệ thống sẽ giúp người dùng tránh việc dữ liệu bị mất, sắp xếp và lưu thông tin một cách hợp lý và logic, xử lý nhanh chóng các thông tin, tra cứu thông tin với một tốc độ truy cập nhanh và độ chính xác cao không nhầm lẫn giúp ta tiết kiệm nhiều thời gian hơn với giao diện làm việc thân thiện, tiện dụng đối với người sử dụng.

## Mô tả chi tiết đề tài

Hệ thống khóa học được chia làm 3 phần:

Phần 1 dành cho học viên, học viên có thể xem các các khóa học xem các thông tin của khóa học bao gồm: tên khóa học, mô tả của khóa học, cấu trúc của khóa học bao gồm các chương và các bài học của từng chương, xem được giá tiền của khóa học. Sau khi thanh toán onl để mua khóa học mà học viên muốn học viên có thể xem được cacsvideo bài giảng của khóa học đó. Học viên có thể tìm kiếm giảng viên, xem các thông tin của giảng viên cũng như các khóa học của giảng viên.

Phần 2 dành cho giảng viên (instructor), giảng viên sau khi đăng kí thành công tài khoản dành cho instructor có thể tạo các khóa học với các chương các bài học, upload video bài giảng. Giảng viên có thể thực hiện các chức nằng quản lý khóa học của mình, có thể block hoặc active khóa học, có thể xóa các khóa học đang ở chế độ nháp hoặc các khóa học đang chờ admin phê duyệt. Giảng viên có thể xem các thông tin thống kê và thu nhập trong tháng. Có thể yêu cầu ứng tiền trước với admin.

Phần 3 dành cho người quản lý (admin), người quản lý có thể thực hiện quản lý các thông tin về người dùng, quyền đăng nhập, các thông tin của khóa học, ngôn ngữ, các lĩnh vực của khóa học. Có thể phên duyệt các các khóa học mới được tạo từ instructor.

## Công cụ sử dụng

* Ngôn ngữ lập trình và công nghệ xử lý BackEnd: C#, .NET Core.
* Ngôn ngữ lập trình và thư viện xử lý FontEnd: HTML, CSS, JavaScript, Reactjs
* Thiết kế cơ sở dữ liệu: Microsoft SQL Server.
* Môi trường phát triển: Visual studio, Visual studio code, Potman.

## Cấu trúc báo cáo đồ án tốt nghiệp

### Chương 1: Mở đầu

### Chương 2: Cơ sở lý thuyết

### Chương 3: Phân tích và thiết kế

### Chương 4: Triển khai chương trình

### Chương 5: Kết quả - Kiến nghị

# CƠ SỞ LÝ THUYẾT

## Tổng quan về ngôn ngữ lập trình C#.

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

* **Ưu điểm**
* Gần gũi với các ngôn ngữ lập trình thông dụng (C++, Java, Pascal).
* Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của những ngôn ngữ đó.
* Cải tiến các khuyết điểm của C/C++ như con trỏ, các hiệu ứng phụ, . .
* Dễ tiếp cận, dễ phát triển.
* Được sự chống lưng của .NET Framework.
* **Nhược điểm**
* Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework.
* Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác. Hầu hết phải dựa vào windows.

## Tổng quan về SQL.

SQL là viết tắt của Structured Query Language, là ngôn ngữ truy vấn có cấu trúc, cho phép bạn truy cập và thao tác với các cơ sở dữ liệu để tạo, xóa, sửa đổi, trích xuất dữ liệu.

SQL cũng là ngôn ngữ tiêu chuẩn cho các hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản trị cơ sở dữ liệu (RDBMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres hay SQL Server đều lấy SQL làm ngôn ngữ cơ sở dữ liệu tiêu chuẩn.

Nếu tìm hiểu về SQL Server của Microsoft, chúng ta sẽ biết rằng ngoài SQL, hệ cơ sở dữ liệu này còn sử dụng T-SQL như ngôn ngữ "địa phương", Oracle SQL thì dùng thêmPL/SQL,...

SQL mang đến nhiều lợi ích như:

* Tạo cơ sở dữ liệu mới
* Tạo bảng mới trong cơ sở dữ liệu
* Tạo view (khung nhìn) mới
* Thực hiện truy vấn trên cơ sở dữ liệu
* Giup mô tả dữ liệu
* Tạo, chèn, xóa, sửa đổi bản ghi trong cơ sở dữ liệu
* Trích xuất dữ liệu từ cơ sở dữ liệu
* Thiết lập quyền trên bảng, thủ tục và view
* Nhúng các ngôn ngữ khác sử dụng mô-đun SQL, thư viện, trình biên dịch sẵn

## Tìm hiểu về Entity Framework

Entity Framework ra đời nhằm hỗ trợ sự tương tác giữa các ứng dụng trên nền tảng .NET với các cơ sở dữ liệu quan hệ. Hay, Entity Frmework chính là công cụ giúp ánh xạ giữa các đối tượng trong ứng dựng, phần mềm của bạn với các bảng của một cơ sở dữ liệu quan hệ.

Emntity Framework giúp các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng với ít mã hơn so với các ứng dụng truyền thống. Lợi ích lớn nhất của nó là giúp lập trình viên giảm thiểu việc lập trình mã nguồn để thực hiện truy cập và tương tác với cơ sở dữ liệu.

Hiện nay, Entity framwork là 1 framework mạnh để phát triển ứng dụng Web với sự hỗ trợ đông đảo của cộng đồng.

Cấu trúc của entity framework:

Graphical user interface, diagram

Description automatically generated

**Hình 2.1. Cấu trúc của Entity Framework.**

**Cấu trúc Entity Framework**

* EDM (Entity Data Model)

EDM bao gồm 3 phần :

* Conceptual Model: Chứa các model class và các quan hệ của nó. Nó độc lập với thiết kế bảng CSDL của bạn.
* Mapping: gồm có thông tin về cách Conceptual model kết nối với Storage model.
* Storage Model: Đây là database design model bao gồm các bảng, views, stored procedures, và những quan hệ và các khóa.
* LINQ to Entities

Là ngôn ngữ truy vấn sử dụng để viết các truy vấn tới object model. Trả về các thực thể được định nghĩa bên trong Conceptual model.

* Entity SQL

Đây là một ngôn ngữ truy vấn khác, nó giống LINQ to Entities.

* Object Service

Object service sẽ có trách nhiệm trong việc cụ thể hóa quá trình chuyển đổi dữ liệu trả về từ một entity client data provider tới một entity object structure.

* Entity Client Data Provider

Nhiệm vụ chính của nó là chuyển đổi L2E hoặc những truy vấn Entity SQL vào một truy vấn SQL. Nó được hiểu bởi CSDL cơ bản. Giao tiếp với ADO.Net data provider lần lượt gửi và nhận dữ liệu từ CSDL.

* ADO.NET Data Provider

Thực hiện giao tiếp với CSDL bằng sử dụng chuẩn ADO.Net.

## Tìm hiểu về ASP.NET Core.

ASP.NET Core là một web framework mã nguồn và được tối ưu hóa cho cloud để phát triển các ứng dụng web chạy trên nhiều nền tảng như Windows, Linux và Mac. Hiện tại, nó bao gồm MVC framework được kết hợp các tính năng của MVC và Web API thành một web framework duy nhất.

* Các ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên .NET Framework hoàn chỉnh.
* Nó đã được thiết kế để cung cấp một framework tối ưu cho các ứng dụng để triển khai tới cloud hoặc chạy on-premises.
* Nó bao gồm những modular với các thành phần tối thiểu, do đó bạn giữ được tính linh hoạt trong quá trình xây dựng các giải pháp của mình.

**Ưu điểm:**

* SP.NET Core có một số thay đổi kiến trúc dẫn đến modular framework nhỏ hơn.
* ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages.
* Điều này cho phép bạn tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.
* Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.

## Tìm hiểu về Web API.

**API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.  
 **Web API** là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng [JSON](https://topdev.vn/blog/json-la-gi/) hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Hoạt động của web API:

* Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.
* Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.
* Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.
* Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động , dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu…
* **RestfulAPI**

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

Diagram

Description automatically generated

**Hình 2.2. Hoạt động của Resful API.**

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

## Tìm hiểu về các Design pattern.

### Repository

Nhiều khi dự án đòi hỏi chúng ta phải định nghĩa các nghiệp vụ giữa tầng business(Business Service Layer) và tầng truy xuất dữ liệu (DAL). Những logic đó có thể được định nghĩa sử dụng Repository Pattern. Về cơ bản Repository là một lớp trung gian giữa hai tầng này. Trong project sử dụng Entity Framework với ASP.NET MVC thì Data chính là tầng chứa các lớp dbContext và class enttiy. Còn business logic chính là tầng xử lý nghiệp vụ của dự án. Repository hầu hết được sử dụng những chỗ cần điều chỉnh dữ liệu trước khi truyền xuống tầng data hoặc truyền lên trên business logic.

Lợi ích của việc sử dụng Repository Pattern:

* Tập trung hóa được các logic về xử lý dữ liệu hoặc business logic.
* Dễ dàng unit test
* Đưa ra kiến trúc linh hoạt
* Khi thay đổi logic của tầng data hoặc business logic, không cần thay đổi Repository
* **Generic Repository Pattern**

Trong các trường hợp khác, dự án lại đòi hỏi một cách tốt nhất để tạo ra tất cả các repository logic ở cùng một nơi. Chúng ta cần tạo 1 và chỉ 1 repository cho việc thao tác với toàn bộ các class entity. Vậy để giải quyết điều này chúng ta phải sử dụng Generic Repository Pattern.

Lợi ích của Generic Repository Pattern:

* Giảm thiểu sự trùng lặp code
* Đảm bảo các coder dùng chung 1 pattern
* Ít lỗi hơn
* Dễ dàng bảo trì sau này

### Unit of work

Unit Of Work được sử dụng để đảm bảo nhiều hành động như insert, update, delete...được thực thi trong cùng một transaction thống nhất. Nói đơn giản hơn, nghĩa là khi một hành động của người dùng tác động vào hệ thống, tất cả các hành động như insert, update, delete...phải thực hiện xong thì mới gọi là một transaction thành công. Gói tất cả các hành động đơn lẻ vào một transaction để đảm bảo tính toàn vẹn dữ liệu.

### Dependency injection.

Dependency Injection là một design pattern tuyệt vời cho phép chúng ta loại bỏ sự phụ thuộc cứng nhắc giữa các phần tử và làm cho ứng dụng trở nên linh hoạt mềm dẻo hơn, dễ mở rộng, dễ bảo trì.

* **Các loại Dependency Injection**
* Constructor injection: Các dependency (biến phụ thuộc) được cung cấp thông qua constructor (hàm tạo lớp).
* Setter injection: Các dependency sẽ được truyền vào 1 class thông qua các setter method (hàm setter).
* Interface injection: Dependency sẽ cung cấp một Interface, trong đó có chứa hàm có tên là Inject. Các client phải triển khai một Interface mà có một setter method dành cho việc nhận dependency và truyền nó vào class thông qua việc gọi hàm Inject của Interface đó.

Trong ba kiểu Inject thì Inject qua phương thức khởi tạo rất phổ biến vì tính linh hoạt, mềm dẻo, dễ xây dựng thư viện DI...

* **Ưu điểm**
* Giảm sự kết dính giữa các module
* Code dễ bảo trì, dễ thay thế module
* Rất dễ test và viết Unit Test
* Dễ dàng thấy quan hệ giữa các module (Vì các dependency đều được inject vào constructor)
* **Khuyết điểm**
* Khái niệm DI khá khó hiểu, các developer mới sẽ gặp khó khăn khi học

Sử dụng interface nên đôi khi sẽ khó debug, do không biết chính xác module nào được gọi

* Các object được khởi tạo toàn bộ ngay từ đầu, có thể làm giảm performance
* Làm tăng độ phức tạp của code
* **Sử dụng DI trong .NET core**

Sử dụng Dependency Injection thông qua các bước:

* Sử dụng một interface hoặc base class để trừu tượng hóa việc triển khai phụ thuộc.
* Đăng ký phần phụ thuộc trong service container. [ASP.NET](http://asp.net/) Core cho phép chúng ta đăng ký các dịch vụ ứng dụng của mình với IoC container, trong phương thức ConfigureServices của lớp Startup. Phương thức ConfigureServices bao gồm một tham số kiểu IServiceCollection . được sử dụng để đăng ký các dịch vụ ứng dụng.
* Đưa service vào phương thức khởi tạo của lớp mà nó được sử dụng. Framework sẽ tạo một thể hiện của sự phụ thuộc và loại bỏ nó khi nó không còn cần thiết nữa.
* **Các loại service lifetime khi đăng kí DI**

Bất cứ khi nào chúng ta yêu cầu Service, DI Container sẽ quyết định xem có tạo mới một thể hiện (instance) hay sử dụng lại thể hiện đã tạo từ trước đó. Vòng đời của Service phụ thuộc vào khi khởi tạo thể hiện và nó tồn tại bao lâu. Chúng ta định nghĩa vòng đời khi đăng ký Service. Có 3 mức độ vòng đời: addTransient, addScoped, addSingleton.

* **Transient**: Instance được khởi tạo mỗi lần tạo service
* **Scoped**: Instance được khởi tạo mỗi scope. ( Scope ở đây chính là mỗi request gửi đến ứng dụng). Trong cùng một scope thì service sẽ được tái sử dụng.
* **Singleton**: Instance của service được tạo duy nhất lúc khởi chạy ứng dụng và được dùng ở mọi nơi.

## Tìm hiểu về HTML

### Giới thiệu

HTML là chữ viết tắt của cụm từ HyperText Markup Language được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (sau đây có thể gọi là một tập tin HTML). Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html> ) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví <strong> **dụ </strong>** và ). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ <strong> **Đây là chữ in đậm** </strong>). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>). Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

### Vai trò của HTML.

HTML là một ngôn ngữ đánh dấu siêu văn bản nên nó sẽ có vai trò xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. Điều đó không có nghĩa là chỉ sử dụng HTML để tạo ra một website mà HTML chỉ đóng một vai trò hình thành trên website. HTML – Xây dựng cấu trúc và định dạng các siêu văn bản. CSS – Định dạng các siêu văn bản dạng thô tạo ra từ HTML thành một bố cục website, có màu sắc, ảnh nền… Javascript – Tạo ra các sự kiện tương tác với hành vi của người dùng (ví dụ nhấp vào ảnh trên nó sẽ có hiệu ứng phóng to). Dễ hiểu hơn, bạn hãy nghĩ rằng nếu website là một cơ thể hoàn chỉnh thì HTML chính là bộ xương của cơ thể đó, nó như là một cái khung sườn vậy. Như vậy, dù website thuộc thể loại nào, giao tiếp với ngôn ngữ lập trình nào để xử lý dữ liệu thì vẫn phải cần HTML để hiển thị nội dung ra cho người truy cập xem

## Tìm hiểu về CSS.

### Giới thiệu về CSS

CSS là chữ viết tắt của Cascading Style Sheets, nó chỉ đơn thuần là một dạng file text với phần tên mở rộng là .css. Trong Style Sheet này chứa những câu lệnh CSS. Mỗi một lệnh của CSS sẽ định dạng một phần nhất định của HTML ví dụ như: font của chữ, đường viền, màu nền, căn chỉnh hình ảnh…

Trước đây khi chưa có CSS, những người thiết kế web phải trộn lẫn giữa các thành phần trình bày và nội dung với nhau. Nhưng với sự xuất hiện của CSS, người ta có thể tách rời hoàn toàn phần trình bày và nội dung. Giúp cho phần code của trang web cũng gọn hơn và quan trọng hơn cả là dễ chỉnh sửa hơn.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

Những tag như <font> được ra mắt trong HTML phiên bản 3.2, nó gây rất nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ chỉ là một trang chứa văn bản mà không có gì khác.

CSS cho phép tạo các quy tắc chỉ định cách nội dung của một phần tử sẽ xuất hiện. Ví dụ: có thể chỉ định rằng nền của trang là màu kem, tất cả các đoạn văn sẽ xuất hiện bằng màu xám bằng kiểu chữ Arial hoặc tất cả các tiêu đề cấp một phải có màu xanh lam, in nghiêng, kiểu chữ Times (có thể tham khảo Chương 10 trong cuốn “Html&Css” [6] của John Ducket)

### Ưu điểm của CSS.

Sự khác biệt giữa site có CSS và không có CSS rất dễ nhận biết.

Trước khi sử dụng CSS, tất cả những phong cách của CSS cần được đính kèm vào trong HTML markup. Có nghĩa là bạn cần tách ra để xác định các thành phần như background, font colors, canh hàng…

CSS giúp định kiểu mọi thứ trên một file khác, bạn có thể tạo phong cách trước rồi sau đó tích hợp file CSS lên trên cùng của file HTML. Việc này giúp HTML markup rõ ràng và dễ quản lý hơn nhiều.

Tóm lại, với CSS bạn không cần lặp lại các mô tả cho từng thành phần. Nó tiết kiệm thời gian, làm code ngắn lại để bạn có thể kiểm soát lỗi dễ dàng hơn

## Tìm hiểu về Reactjs.

[ReactJS](https://reactjs.org/) là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,… Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:

* JSX.
* Virtual DOM.

# PHÂN TÍCH VÀ THIẾT KẾ

## Đặt vấn đề

Giáo dục là một trong những ngành giữ vai trò quan trọng và chủ chốt trong việc xây dựng một đất nước thịnh vượng, mà cụ thể là đầu tư từng cá nhân một về tri thức cũng như kỹ năng sống.

Với mức độ lan tỏa của mình, cuộc cách mạng 4.0 đã tạo ra sự thay đổi đối với thị trường lao động trong mọi khía cạnh, đặc biệt là đối với trình độ chuyên môn. Điều này đặt ra yêu cầu cho giáo dục là cần phải đào tạo nguồn nhân lực có đủ chuyên môn để thích nghi được với môi trường kỹ thuật mới.

Chính yêu cầu đó đã biến môi trường giáo dục vốn chỉ tập trung truyền tải những kiến thức hàn lâm thì nay đã đổi mới bằng việc cung cấp cho người học cả những kiến thức về kỹ năng bao gồm kỹ năng thực hành, kỹ năng làm việc nhóm, kỹ năng tư duy sáng tạo, kỹ năng phản biện.... Cách mạng 4.0 trong giáo dục cũng buộc người học phải chủ động thay đổi và chủ động hơn trong việc học tập của mình.

Ngày qua ngày, giáo dục không chỉ được biết đến với hình thức trường học, đại học mà còn là trung tâm và hình thức giáo dục online. Với sự bùng nổ của nhiều trường học, cơ sở giáo dục như vậy thì để giúp cho các trường học và trung tâm có thể tiếp cận hiệu quả nguồn học viên – phụ huynh, nổi bật hơn các trường khác, cũng như là đáp ứng tối đa những nhu cầu về thông tin cho người dùng thì website khóa học chính là giải pháp hàng đầu.

## Yêu cầu hệ thống

### Yêu cầu chức năng

Hệ thống khóa học bao gồm:

Trang Web chia ra ba đối tượng sử dụng, gồm người học viên thực hiện học các khóa học thi kiểm nghiệm, giảng viên quản trị viên thực hiện quản lý Website.

Ở giao diện người quản trị:

* Chức năng quản lý các khóa học.
* Quản lý các bài học online.
* Chức năng quản lý các đào tạo.
* Chức năng quản lý thông tin học viên.
* Chức năng quản lý nhân viên.
* Chức năng thống kê

Ở giao diện giảng viên:

* Xem các khóa học tại trung tâm.
* Tiến hành đăng ký các khóa học tại trung tâm.
* Xem các khóa học online. - Tiến hành thanh toán để sở hữu khóa học online
* Bình luận các khóa học đã sở hữu –
* Tham khảo và làm các bài kiểm tra.
* Đăng ký, đăng nhập và quên mật khẩu

Ở giao diện người dùng:

* Xem các khóa học hiện co
* Xem thông tin của khóa học
* Có thể mua khóa học
* Sau khi mua có thể xrm video hướng dẫn của khóa học
* Xem các thông tin về các khóa học đã mua.
* Xem thông tin về giảng viên.
* Xem các khóa học của giảng viên.

### Mô tả hệ thống

Hệ thống khóa học được chia làm 3 phần:

Phần 1 dành cho học viên, học viên có thể xem các các khóa học xem các thông tin của khóa học bao gồm: tên khóa học, mô tả của khóa học, cấu trúc của khóa học bao gồm các chương và các bài học của từng chương, xem được giá tiền của khóa học. Sau khi thanh toán online để mua khóa học mà học viên muốn học viên có thể xem được cacsvideo bài giảng của khóa học đó. Học viên có thể tìm kiếm giảng viên, xem các thông tin của giảng viên cũng như các khóa học của giảng viên.

Phần 2 dành cho giảng viên (instructor), giảng viên sau khi đăng kí thành công tài khoản dành cho instructor có thể tạo các khóa học với các chương các bài học, upload video bài giảng. Giảng viên có thể thực hiện các chức nằng quản lý khóa học của mình, có thể block hoặc active khóa học, có thể xóa các khóa học đang chờ admin phê duyệt, có thể xem các thông tin thống kê và thu nhập trong tháng. Có thể yêu cầu ứng tiền trước với admin.

Phần 3 dành cho người quản lý (admin), người quản lý có thể thực hiện quản lý các thông tin về người dùng, quyền đăng nhập, các thông tin của khóa học, ngôn ngữ, các lĩnh vực của khóa học. Có thể phên duyệt các các khóa học mới được tạo từ instructor.

## Thiết kế hệ thống

### Mô hình thực thể kết hợp ERD

Diagram, schematic

Description automatically generated

**Hình 3.1. Mô hình thực thể ERD.**

### Sơ đồ phân rã chức năng

Graphical user interface

Description automatically generated

**Hình 3.2. Mô hình phân rã chức năng**

### Sơ đồ UseCase tổng quát

Diagram

Description automatically generated

**Hình 3.3. Mô hình usecase tổng quát.**

### Sơ đồ hoạt động

* Sơ đồ hoạt động học viên mua khóa học: sau khi đăng nhập thành công học viên có thể xem các khóa học hiện đang có của hệ thống, sinh viên chọn khóa sau đó xem các thông tin về khóa học cũng như giá tiền của nó. Nếu như quyết định mua thì tiền s hành thanh toán.

Diagram

Description automatically generated

**Hình 3.4. Mô hình hoạt động mua khóa học.**

* Sơ đồ hoạt động instructor thêm khóa học mới: để thêm khóa học mới giảng viên chọn thêm khóa học mới sau đó nhập đầy đủ các nội dung của khóa học bao gồm các chương các bài của chương và upload video bài giảng. sau đó phải chờ admin phê duyệt khóa học đó.

Diagram

Description automatically generated

**Hình 3.5. Mô hình hoạt động thêm khóa học mới.**

* Sơ đồ hoạt động đăng kí tài khoản giảng viên (instructor): khi một giản viên muốn đăng kí một tài khoản instructor trong hệ thống để có thể tạo khóa học, giảng viên đó sẽ phải gửi yêu cầu đến admin. Sau khi tiếp nhận yêu cầu admin sẽ kiểm tra các thông tin liên quan và sẽ có buổi phỏng vấn kiểm tra trình đọ chuyên môn, nếu được sẽ tiến hành cấp tài khoản cho giảng viên.

Diagram

Description automatically generated

**Hình 3.6. Mô hình hoạt động đăng kí tài khoản giảng viên.**

* Sơ đồ hoạt động phê duyệt khóa học: sau khi nhận yêu cầu phê duyệt khóa học mới từ instructor, admin tiến hành kiểm tra nội dung nếu nội dung phù hợp thì phê duyệt cho khóa học.

Diagram

Description automatically generated

**Hình 3.7. Mô hình hoạt động phê duyệt khóa học**

### Database Diagram

Diagram

Description automatically generated

**Hình 3.8. Sơ đồ class**

# TRIỂN KHAI CHƯƠNG TRÌNH

## Vận hành hệ thống

Đối với quản trị viên: Quản trị viên có toàn quyền sử dụng trang web để điều hành hệ thống.

Đối với giảng viên: Giảng viên cần có một chiếc máy tính có kết nối internet và cần phải có tài khoản của instructor để có thể sử dụng.

Đối với người dùng bình thường: Người dùng bình thường cần có một chiếc máy tính hoặc điện thoại để kết nối Internet để có thể truy cập vào trang web và cần đăng ký tài khoản nếu muốn sử dụng được các chức năng của khách hàng.

## Xây dựng giao diện.

### Giao diện đăng nhập

Để có thể vào được trang dành cho admin hoặc giảng viên thì phải đăng nhập với tài khoản có role là admin hoặc instructor.

Graphical user interface, application

Description automatically generated

**Hình 4.1. Trang đăng nhập**

Nếu đăng nhập thất bại sẽ có thông báo.

Graphical user interface, application

Description automatically generated

**Hình 4.2. Trang đăng nhập thất bại.**

### Giao diện Admin

Khi đăng nhập bằng tài khoản admin, hệ thống sẽ chuyển đến giao diện của admin Dashboard.

Graphical user interface, chart, bar chart

Description automatically generated

**Hình 4.3. Giao diện admin.**

Trong trang dashboard có các thống kê khóa học và doanh thu của các instructor.

Chart, bar chart

Description automatically generated

**Hình 4.4. Thống kê doanh thu.**

Graphical user interface, application

Description automatically generated

**Hình 4.5. Thống kê khóa học.**

Khi chọn mục User hệ thống sẽ chuyển qua màn hình user. Ở trang này admin có thể thêm người dùng mới và cấp quyền cho người dùng đó.

Graphical user interface, application

Description automatically generated

**Hình 4.6. Giao diện thêm người dùng mới.**

Khi chọn vào nút edit admin có thể sửa lại thông tin cho người dùng.

Graphical user interface, application, table

Description automatically generated

**Hình 4.7. Giao diện sửa thông tin user.**

Tương tự như trang User, admin có thể thêm sửa và xóa các language.

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.8. Giao diện thêm language.**

Graphical user interface, application

Description automatically generated

**Hình 4.9. Giao diện sửa language.**

Trang Category và SubCategory cũng có các chức năng giống Language.

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.10. Giao diện Category.**

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.11. Giao diện SubCategory.**

Khi chọn vào trang student hệ thống sẽ hiển thị ra giao diện student.

Graphical user interface, application

Description automatically generated

**Hình 4.12. Giao diện Student.**

Admin có thể Block hoặc Un-Block Student.

Table

Description automatically generated

**Hình 4.13. Giao diện khi Block thành công.**

Khi người dùng chọn vào mục Instructor hệ thống sẽ hiển thị ra trang instructor với các chức năng tương tự student.

Table

Description automatically generated with medium confidence

**Hình 4.14. Giao diện Instructor.**

Admin có thể xem thông tin chi tiết của instructor.

Graphical user interface, application

Description automatically generated

**Hình 4.15. Giao diện thông tin chi tiết của Instructor.**

Khi người dùng chọn vào mục Payout admin có thể xem các yêu cầu thanh toán của instructor.

Graphical user interface, application, table

Description automatically generated

**Hình 4.16. Giao diện Payout.**

Ở phần quản lý khóa học có các lựa chọn sau:

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.17. Các lựa chọn khóa học**

Khi đó admin có thể xem danh sách các khóa học theo từng trạng thái. Admin có thể Block hoặc Un-Block các khóa học.

Trong phần Waiting for Approve là các khóa học đang chờ phê duyệt.

Graphical user interface, application, table

Description automatically generated

**Hình 4.18. Giao diện khóa học.**

Admin có thể xem chi tiết cảu khóa học đó để có thể đưa ra quyết định có phê duyệt cho khóa học đó hay không.

Graphical user interface, text, application

Description automatically generated

**Hình 4.19. Giao diện thông tin chi tiết của khóa học.**

### Giao diện Instructor

Khi người dùng đăng nhập bằng tài khoản của giảng viên hệ thống sẽ hiển thị trang dành cho giảng viên.

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.20. Giao diện trang Dashboard của Instructor.**

Ở trang Dashboard có các thông số thống kê về doanh thu, tổng các khóa học, số lượng học viên đăng ký mua khóa học và các khóa học đang chờ dc phê duyệt từ admin.

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.21. Các khóa học mới gửi.**

Khi người dùng chọn nút Courses thì hệ thống sẽ hiển thị danh sách các khóa học của giảng viên đó. Giảng viên cũng có thể Block hoặc Un-Block các khóa học, và có thể xóa các khóa học bản nháp hoặc các khóa học chưa được phê duyệt

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.22. Giao diện danh sách các khóa học của Instructor.**

Khi người dùng nhấn vào nút Create Course hệ thống sẽ hiển thị trang thêm khóa học

Graphical user interface, text, application, email

Description automatically generated

**Hình 4.23. Giao diện tạo khóa học.**

Ở phần đầu tiên General information người dùng nhập các thông tin của khóa học sau đó bấm next.

Sau đó sẽ thêm hình ảnh làm background cho khóa học và bấn next.

Graphical user interface, text, application

Description automatically generated

**Hình 4.24. Giao diện thêm background cho khóa học.**

Tiếp theo là tiến hành tạo các bài học của khóa học.

Graphical user interface, application, email

Description automatically generated

**Hình 4.25. Giao diện tạo nội dung của khóa học.**

Sau khi có được danh sách các bài học, tiến hành upload video bài giảng lên hệ thống.

Graphical user interface, application

Description automatically generated

**Hình 4.26. Upload video bài giảng và lưu bài học.**

Sau khi upload thành công sẽ nhận được thông báo, sau đó bấm next để hoàn tất quá trình tạo khóa học.

Graphical user interface

Description automatically generated with medium confidence

**Hình 4.27. Giao diện upload thành công.**

Khi người dùng nhấn vào mục Earning hệ thống sẽ hiển thị ra giao diện thống kê doanh thu, số khóa học đã được bán theo tháng và các khóa học hàng đầu.

Graphical user interface, application, website

Description automatically generated

**Hình 4.28. Giao diện trang Earning.**

Khi người dùng chọn vào mục Payout hệ thống sẽ hiển thị ra trang Payout để giảng viên có thể gửi yêu cầu thnah toán đến admin.

Graphical user interface, application

Description automatically generated

**Hình 4.29. Giao diện trang Payout của Instructor.**

### Giao diện Student

Khi không đăng nhập hoặc đăng nhập với tài khoản thường thì người dùng sẽ vô trang dành cho student.

Graphical user interface, application

Description automatically generated

**Hình 4.30. Giao diện trang chủ.**

Ở đây sẽ hiển thị tất cả các khóa học với giá tiền của khóa học.

Học viên có thể bấm chọn vào một khóa học để có thể xem chi tiết khóa học đó.

Graphical user interface, text, application, email

Description automatically generated

**Hình 4.31. Giao diện trang thông tin khóa học.**

Nếu muốn mua một khóa học nào đó người dùng cần đăng nhập sau đó bấm vào nút Buy Course. Hệ thống sẽ chuyển sang trang thanh toán online.

Graphical user interface, application

Description automatically generated

**Hình 4.32. Giao diện trang thanh toán online.**

Sau khi thah toán thành công, học viên sẽ vào mục My Course để xem các khóa học đã mua.

Graphical user interface, application

Description automatically generated

**Hình 4.33. Các khóa học đã mua.**

Học viên có thể chọn một khóa học đã mau để bắt đầu học.

A picture containing funnel chart

Description automatically generated

**Hình 4.34. Video bài giảng.**

Học viên có thể commnet để trao đổi về bài giảng.

Graphical user interface, text, application, email

Description automatically generated

**Hình 4.35. Giao diện bình luận.**

Khi người dùng chọn vào mục Instructor thì hệ thống sẽ hiển thị ra danh sách các giảng viên

Graphical user interface, application, Teams

Description automatically generated

**Hình 4.36. Danh sách các giảng viên.**

Người dùng có thể chọn một giảng viên bất kì để xem thông tin của giảng viên đó và các khóa học của giảng viên đó.

A screenshot of a computer

Description automatically generated with medium confidence

**Hình 4.37. Thông tin chi tiết cảu giảng viên.**

# KẾT QUẢ - KIẾN NGHỊ

## Kết quả đạt được

Trong quá trình làm đồ án, em đã học được rất nhiều kiến thức mới cũng như củng cố được những kiến thức cũ. Kết quả em đạt được :

- Biết cách thiết kế cơ sở dữ liệu và sử dụng công cụ Microsoft SQL Server để tạo cơ sở dữ liệu, tạo các ràng buộc dữ liệu và khai thác dữ liệu phù hợp với chương trình.

- Biết cách sử dụng Visual Studio Code xây dựng chức năng của website , tạo được giao diện phù hợp với web, dễ dàng sử dụng bằng HTML , CSS , JS.

- Biết cách sử dụng Visual studio và công nghệ ASP.NET core để xử lý BackEnd của một website.

## Hạn chế

Ngoài những kết quả đã đạt được, hệ thống còn vấp phải nhiều hạn chế cần phải giải quyết:

- Giao diện hệ thống được thiết kế theo cảm tính của cá nhân , chưa được đẹp hoàn hảo .

- Cơ sở dữ liệu chưa hoàn hảo và logic, dễ mắc một số lỗi đối với các trường hợp phức tạp ngoài thực tế.

- Còn thiếu một số tính năng nâng cao để ứng dụng sử dụng tiện lợi hơn.

## Hướng phát triển

Trong tương lai, chương trình sẽ được khắc phục những hạn chế . Song song với việc đó, chương trình cũng sẽ được bổ sung thêm nhiều tính năng mới nâng cao hơn nữa chất lượng cũng như nhiều nhu cầu khác của người học, nhằm giúp hệ thống quản lý chính xác nhất. Ngoài ra, nếu có thể, chương trình sẽ được tích hợp vào các chương trình, hệ thống khác và có thể sử dụng rộng rãi, phù hợp với các doanh nghiệp có quy mô lớn hơn phục vụ cho thương mại.

# TÀI LIỆU THAM KHẢO

[1] <https://fsoft-academy.edu.vn> [Accessed 1/4/2022]

[2] <https://feras.blog/how-to-use-asp-net-identity-and-identityserver4-in-your-solution/> [Accessed 15/3/2022]

[3] <https://enlabsoftware.com/development/how-to-implement-repository-unit-of-work-design-patterns-in-dot-net-core-practical-examples-part-one.html> [Accessed 15/3/2022]

[4] <https://code-maze.com/entity-framework-core-series/> [15/3/2022]

[5] [(269) F8 Official - YouTube](https://www.youtube.com/c/F8VNOfficial) [Accessed 15/5/2022]

[6] [Developer tools, technical documentation and coding examples | Microsoft Docs](https://docs.microsoft.com/en-us/)

[Accessed 23/4/2022]

[7] [Getting Started – React (reactjs.org)](https://reactjs.org/docs/getting-started.html) [Accessed 1/6/2022]