TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ ASP.NET**

**NĂM HỌC 2024-2025**

**XÂY DỰNG WEBSITE DIỄN ĐÀN SỨC KHỎE**

*Sinh viên thực hiện:*

Họ tên: Trần Thị Thu Tiền

MSSV: 170124272

Lớp: DK24TTC2

MSSV: 170124272

Lớp: DK24TTC2

*Giảng viên hướng dẫn:*

TS. Đoàn Phước Miền

***Gia Lai, tháng 11 năm 2025***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

…………, ngày…...tháng….năm 2025

**Giáo viên hướng dẫn**

*(ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

…………, ngày…..tháng….năm 2025

**Thành viên hội đồng**

*(ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Trong suốt quá trình thực hiện đề tài “Xây dựng website diễn đàn sức khỏe”, em đã nhận được rất nhiều sự giúp đỡ, hướng dẫn và động viên từ quý Thầy cô và bạn bè. Nhờ đó, em đã có thể hoàn thành đề tài đúng thời gian và đạt được những kết quả như mong đợi.

Em xin bày tỏ lòng biết ơn sâu sắc đến TS. Đoàn Phước Miền, giảng viên hướng dẫn, người đã tận tình chỉ bảo, định hướng và hỗ trợ em trong suốt quá trình nghiên cứu và hoàn thiện đề tài.

Em cũng xin cảm ơn bạn bè đã luôn hỗ trợ trao đổi tài liệu, góp ý và động viên em trong quá trình thực hiện đề tài. Mặc dù đã cố gắng hoàn thiện tốt nhất, đề tài không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của quý thầy cô để đề tài được hoàn thiện hơn.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc215214006)

[1.1. Lý do chọn đề tài 1](#_Toc215214007)

[1.2. Mục tiêu nghiên cứu 1](#_Toc215214008)

[1.3. Phương pháp nghiên cứu 2](#_Toc215214009)

[1.4. Mô tả mục tiêu chi tiết 3](#_Toc215214010)

[1.5. Đặc điểm kỹ thuật nổi bật 3](#_Toc215214011)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc215214012)

[2.1. Tổng quan về mô hình ASP.NET Core 9.0 MVC 4](#_Toc215214013)

[2.2. Entity Framework Core (EF Core) 9.0 6](#_Toc215214014)

[2.3. Mô hình kiến trúc MVC (Model-View-Controler) 7](#_Toc215214015)

[2.4. Công nghệ Frontend 8](#_Toc215214016)

[2.5. Bảo mật với ASP.NET Identity 8](#_Toc215214017)

[CHƯƠNG 3: HIỆN THỰC HÓA VÀ THIẾT KẾ HỆ THỐNG 10](#_Toc215214018)

[3.1. Giới thiệu hệ thống 10](#_Toc215214019)

[3.2. Phân tích hệ thống 10](#_Toc215214020)

[CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ NGHIÊN CỨU 19](#_Toc215214021)

[4.1. Quy trình cài đặt và khởi chạy 19](#_Toc215214022)

[4.2. Entity Framework 22](#_Toc215214023)

[4.3. Thiết kế Giao diện Người dùng (UX/UI) 22](#_Toc215214024)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 27](#_Toc215214025)

[5.1. Kết luận: 27](#_Toc215214026)

[5.2. Hạn chế: 27](#_Toc215214027)

[5.3. Hướng phát triển: 27](#_Toc215214028)

[DANH MỤC TÀI LIỆU THAM KHẢO 28](#_Toc215214029)

**DANH MỤC HÌNH ẢNH**

[**Hình 1: Mô hình kiến trúc MVC – Model – View – Controller.** 7](#_Toc215214108)

[**Hình 2: Thực thể Tài Khoản** 12](#_Toc215214109)

[**Hình 3: Lược đồ cơ sở dữ liệu** 12](#_Toc215214110)

[**Hình 4: Lược đồ USE CASE** 13](#_Toc215214111)

[**Hình 5: Lược đồ USE CASE Quản trị viên (Admin)** 14](#_Toc215214112)

[**Hình 6: Lược đồ USE CASE Khách và người dùng** 15](#_Toc215214113)

[**Hình 7: Tạo Project mới trên Visual Studio** 19](#_Toc215214114)

[**Hình 8: Chọn ASP.NET Core Web App (Model-View-Controller)** 20](#_Toc215214115)

[**Hình 9:Đặt tên cho Project và đường dẫn lưu** 20](#_Toc215214116)

[**Hình 10: Nhấn chọn Create** 21](#_Toc215214117)

[**Hình 11: Giao diện làm việc và cấu trúc cây thư mục của ASP.NET MVC** 21](file:///C:\Users\giakiet\Downloads\BC%20CĐ%20ASPNET%20TranThiThuTien_170124272_DK24TTC2.docx#_Toc215214118)

[**Hình 12: Package Manager Console sử dụng dòng lệnh** 22](file:///C:\Users\giakiet\Downloads\BC%20CĐ%20ASPNET%20TranThiThuTien_170124272_DK24TTC2.docx#_Toc215214119)

[**Hình 13: Trang chủ** 23](#_Toc215214120)

[**Hình 14: Giao diện quản lý chủ đề** 23](#_Toc215214121)

[**Hình 15: Giao diện danh sách chủ đề** 24](#_Toc215214122)

[**Hình 16: Giao diện Admin (tổng quan thống kê)** 25](#_Toc215214123)

[**Hình 17: Giao diện quản lý báo cáo** 25](#_Toc215214124)

[**Hình 18: Giao diện tạo thêm chủ đề** 26](#_Toc215214125)

**DANH MỤC CỤM TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| Từ viết tắt | Diễn giải |
| MVC | Model-View-Controller |
| SQL | Structured Query Language |
| HTML | HyperText Markup Language |
| UX/UI | User Experience/ User Interface |
| C# | C Sharp |
| CSDL | Cơ sở dữ liệu |
| ADO.NET | ActiveX Data Objects for .NET |
| SSMS | SQL Server Management Studio |
| EF | Entity Framework |
| ORM | Object-Relational Mapping Framework |
| ERD | Entity-Relationship Diagram |
| DFD | Data Flow Diagram |
| GUI | Graphical User Interface |
| RDBMS | Relational Database Management System |

**TÓM TẮT ĐỒ ÁN**

Đề tài “xây dựng website diễn đàn sức khoẻ” Đề tài “Xây dựng website diễn đàn sức khoẻ” nhằm tạo ra một nền tảng trực tuyến nơi người dùng có thể trao đổi thông tin, chia sẻ kinh nghiệm và đặt câu hỏi liên quan đến sức khoẻ. Hệ thống cho phép người dùng đăng ký, đăng nhập, tạo bài viết, bình luận và tương tác với các chủ đề sức khoẻ một cách dễ dàng.

Website được xây dựng trên nền tảng HTML, CSS, JavaScript cho giao diện và PHP/MySQL cho xử lý dữ liệu và quản lý cơ sở dữ liệu. Cơ sở dữ liệu được thiết kế với các bảng chính như: User, Post, Comment, Category, đảm bảo tính toàn vẹn dữ liệu thông qua các ràng buộc khóa chính – khóa ngoại.

Hệ thống hướng đến việc cung cấp một môi trường trao đổi lành mạnh, hỗ trợ học tập và nâng cao kiến thức sức khoẻ cộng đồng.

# 

1. TỔNG QUAN
   1. Lý do chọn đề tài

Trong những năm gần đây, nhu cầu tìm hiểu thông tin về sức khoẻ của cộng đồng ngày càng tăng. Tuy nhiên, phần lớn người dùng Internet gặp khó khăn khi tiếp cận nguồn thông tin chính thống hoặc khó tìm nơi để đặt câu hỏi, chia sẻ kinh nghiệm bệnh lý và nhờ tư vấn từ cộng đồng. Nhiều diễn đàn hoặc mạng xã hội hiện nay thiếu tính chuyên sâu, thiếu phân loại nội dung và không đảm bảo sự tương tác hiệu quả giữa người hỏi – người trả lời.

Vì vậy, em quyết định chọn đề tài “Xây dựng website diễn đàn sức khoẻ” nhằm tạo ra một môi trường trực tuyến nơi mọi người có thể:

* Trao đổi và chia sẻ kiến thức về sức khoẻ một cách thuận tiện.
* Đặt câu hỏi và nhận phản hồi từ cộng đồng nhanh chóng.
* Tương tác trong một không gian được phân loại rõ ràng theo chủ đề.
* Tiếp cận nguồn thông tin tổng hợp, dễ hiểu và thân thiện với người dùng.

Ngoài ra, việc xây dựng website diễn đàn giúp ứng dụng kiến thức đã học về lập trình web, thiết kế cơ sở dữ liệu, bảo mật thông tin, xử lý dữ liệu người dùng và thực hành xây dựng một hệ thống web hoàn chỉnh. Đây là một đề tài thiết thực, có ý nghĩa xã hội và phù hợp với xu hướng chuyển đổi số trong lĩnh vực chăm sóc sức khoẻ.

* 1. Mục tiêu nghiên cứu

Vận dụng kiến thức đã học để xây dựng một website bán hàng hoàn chỉnh: Ứng dụng lý thuyết về lập trình web, cơ sở dữ liệu SQL Server, mô hình MVC, HTML/CSS, Bootstrap [2], ngôn ngữ C# và công nghệ ASP.NET [3] để phát triển một hệ thống website thương mại điện tử thực tế.

Thiết kế giao diện người dùng thân thiện và dễ sử dụng: Tạo ra giao diện website trực quan, dễ thao tác cho cả người dùng lẫn quản trị viên, giúp nâng cao trải nghiệm người dùng (UX/UI).

Mục tiêu chính là xây dựng một ứng dụng web Sẵn sàng Sản xuất (Production Ready) với các tính năng MVP (Minimum Viable Product) đầy đủ.

* 1. Phương pháp nghiên cứu
* *Phương pháp nghiên cứu tài liệu*

Tìm hiểu kiến thức liên quan đến lập trình web: HTML, CSS, JavaScript, PHP, MySQL.

Tham khảo các giáo trình, tài liệu online, diễn đàn, các website mẫu để học hỏi cách tổ chức, xây dựng và triển khai hệ thống website diễn đàn sức khoẻ.

Nghiên cứu cấu trúc, chức năng, mô hình hoạt động của các diễn đàn sức khoẻ hiện có để rút ra ưu – nhược điểm.

* *Phương pháp khảo sát và phân tích hệ thống*

Khảo sát nhu cầu người dùng (sinh viên, người quan tâm sức khoẻ) thông qua câu hỏi và quan sát thực tế.

Xác định các chức năng cần thiết: đăng ký, đăng nhập, tạo bài viết, bình luận, phân loại chủ đề, tìm kiếm…

Phân tích yêu cầu hệ thống từ góc độ người dùng và quản trị viên để xác định các chức năng cần thiết.

Xây dựng sơ đồ chức năng, sơ đồ luồng dữ liệu (DFD), mô hình thực thể - quan hệ (ERD) để thiết kế hệ thống một cách có tổ chức và dễ triển khai.

* *Phương pháp thiết kế và lập trình*

Áp dụng quy trình phát triển phần mềm để thiết kế giao diện, xây dựng cơ sở dữ liệu và lập trình các chức năng chính của hệ thống theo mô hình phân lớp (3-tier hoặc MVC).

Sử dụng công cụ Visual Studio và hệ quản trị cơ sở dữ liệu SQL Server để phát triển và kiểm thử ứng dụng.

* *Phương pháp thực nghiệm*

Tiến hành chạy thử nghiệm hệ thống, kiểm tra tính đúng đắn của các chức năng như tìm kiếm, đặt hàng, quản lý sản phẩm và xử lý đơn hàng.

Ghi nhận các lỗi phát sinh trong quá trình sử dụng và thực hiện các bước sửa lỗi, cải tiến giao diện và tối ưu hiệu suất hệ thống.

* 1. Mô tả mục tiêu chi tiết
* *Chức năng:*

Triển khai đầy đủ hệ thống Tài khoản, Quản lý Nội dung (CRUD Chủ đề, Phản hồi), Tương tác Xã hội (Like, Tìm kiếm, Thông báo) và Quản trị Hệ thống.

* *Giao diện:*

Đạt chuẩn UX/UI premium với giao diện hiện đại, sử dụng gradient backgrounds và hiệu ứng glassmorphism. Đảm bảo Responsive design trên mọi thiết bị.

* *Hiệu suất:*

Đạt Performance cao thông qua tối ưu hóa Entity Framework Core.

* *Bảo mật:*

Áp dụng các biện pháp bảo mật mạnh mẽ như ASP.NET Identity, Role-based authorization, và chống tấn công XSS/CSRF.

* 1. Đặc điểm kỹ thuật nổi bật

Công nghệ Lõi: ASP.NET Core 9.0 MVC, C# 12.

Database: SQL Server LocalDB, quản lý qua Entity Framework Core 9.0.

Kiến trúc: Tuân thủ chặt chẽ mô hình MVC (Model-View-Controller).

Thiết kế: Sử dụng Bootstrap 5.3 và CSS3 tùy chỉnh để tạo giao diện hiện đại, mượt mà.

1. CƠ SỞ LÝ THUYẾT
   1. Tổng quan về mô hình ASP.NET Core 9.0 MVC

Khái niệm

MVC là viết tắt của Model - View - Controller, là một mô hình kiến trúc phần mềm được sử dụng phổ biến trong phát triển ứng dụng web. Mô hình này giúp phân tách rõ ràng các thành phần của ứng dụng, từ đó tăng tính tổ chức, dễ bảo trì và dễ mở rộng.

Là khung làm việc (framework) cho phép xây dựng ứng dụng web đa nền tảng, hiệu suất cao.

* Ưu điểm: Hiệu suất vượt trội so với các phiên bản .NET Framework trước, hỗ trợ đa nền tảng (Windows, Linux, macOS), và là nền tảng cho kiến trúc MVC.
* Sử dụng trong Dự án: Được sử dụng để định tuyến yêu cầu, quản lý trạng thái, và thực thi logic nghiệp vụ trong các Controller.

Các thành phần

Ba thành phần chính của mô hình MVC bao gồm:

* ***Model (M) – Mô hình dữ liệu***

*Chức năng chính:*

Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.

*Nhiệm vụ:*

Quản lý dữ liệu, trạng thái của ứng dụng.

Tương tác với cơ sở dữ liệu (CRUD: Create, Read, Update, Delete).

Cung cấp dữ liệu cho View thông qua Controller.

* ***Controller (C) – Bộ điều khiển***

*Chức năng chính:*

Là trung gian giữa Model và View. Nhận đầu vào từ người dùng và xử lý logic.

*Nhiệm vụ:*

Nhận yêu cầu từ người dùng (qua giao diện – View).

Gọi đến các phương thức của Model để xử lý dữ liệu.

Chọn View phù hợp để hiển thị kết quả.

* ***View (V) – Giao diện hiển thị***

*Chức năng chính:*

Hiển thị thông tin cho người dùng (UI – User Interface).

*Nhiệm vụ:*

Nhận dữ liệu từ Controller/Model và trình bày ra màn hình.

Lịch sử phát triển

MVC (Model – View – Controller) có một lịch sử phát triển khá lâu dài, bắt nguồn từ những năm 1970 bởi Trygve Reenskaug, là một nhà nghiên cứu tại Xerox PARC – một trong những trung tâm công nghệ tiên phong lúc bấy giờ. Ông phát triển mô hình MVC trong quá trình làm việc với ngôn ngữ Smalltalk để hỗ trợ phát triển giao diện người dùng (GUI) một cách rõ ràng, có tổ chức.

Giai đoạn phát triển ban đầu (1980s), MVC được tích hợp sâu vào các môi trường phát triển dựa trên Smalltalk-80.

Lúc này mô hình chủ yếu được dùng trong ứng dụng desktop GUI, chưa được phổ biến rộng rãi ra ngoài cộng đồng nghiên cứu.

Áp dụng vào Web & mở rộng (1990s – 2000s). Khi ứng dụng web phát triển mạnh (HTML, HTTP, trình duyệt...), nhu cầu tổ chức mã nguồn rõ ràng ngày càng cấp thiết.

Luồng xử lý trong MVC

Luồng xử lý thể hiện cách dữ liệu và sự kiện di chuyển giữa các thành phần trong hệ thống. Dưới đây là mô tả tuần tự các bước khi người dùng tương tác với hệ thống:

* ***Người dùng tương tác với hệ thống (qua View)***

Ví dụ: Người dùng nhấn nút "Đăng nhập", điền form, chọn menu...

Người dùng gửi **Request** (yêu cầu) đến **Controller.**

* ***Controller xử lý logic & tương tác với Model***

Controller tiếp nhận yêu cầu từ View và quyết định nên làm gì:

Kiểm tra dữ liệu đầu vào.

Gọi các phương thức tương ứng từ Model để xử lý dữ liệu.

* ***Model thao tác dữ liệu***

Model chịu trách nhiệm:

Tương tác với cơ sở dữ liệu.

Xử lý nghiệp vụ, tính toán, kiểm tra logic...

Sau khi hoàn thành, Model trả dữ liệu/ trạng thái cho Controller.

* ***Controller nhận kết quả và chọn View phù hợp***

Dựa trên kết quả xử lý từ Model, Controller quyết định:

Hiển thị View thành công hay thất bại.

Truyền dữ liệu cần thiết cho View để hiển thị.

View hiển thị dữ liệu cho người dùng

View nhận dữ liệu từ Controller và hiển thị ra giao diện người dùng.

Người dùng lại tiếp tục tương tác, bắt đầu vòng lặp mới.

* 1. Entity Framework Core (EF Core) 9.0

Tổng quan

EF Core là công cụ ORM (Object-Relational Mapper), cho phép nhà phát triển làm việc với cơ sở dữ liệu bằng các đối tượng .NET. Sử dụng Migrations để quản lý thay đổi schema cơ sở dữ liệu theo thời gian.

Vai trò.

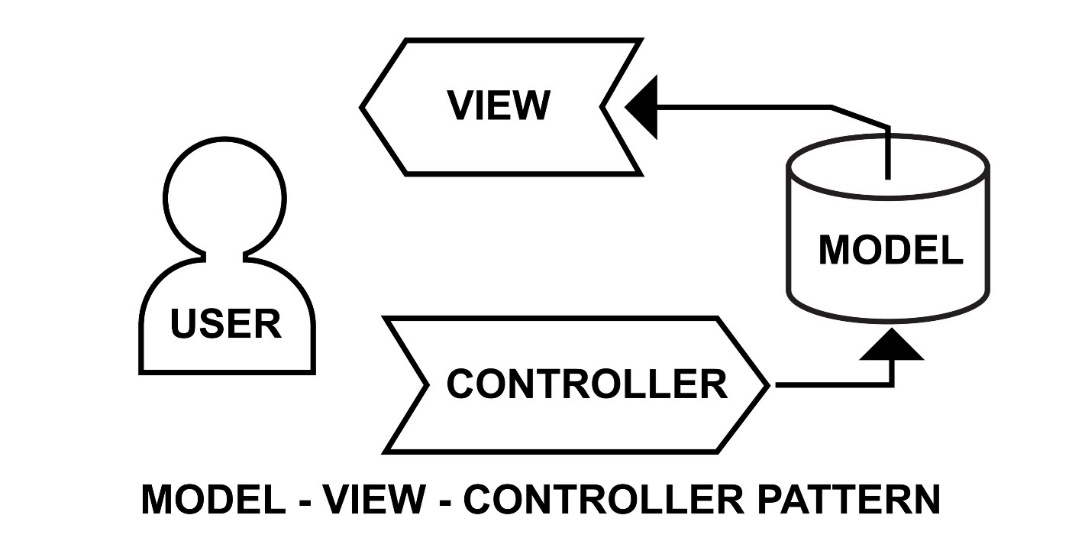
EF Core tạo ra một lớp trừu tượng giữa ứng dụng và SQL Server LocalDB. Nó xử lý việc ánh xạ giữa các lớp Model (như Topic, Reply) và các bảng trong cơ sở dữ liệu.

Lợi ích Bảo mật

EF Core tự động sử dụng parameterized queries, là biện pháp hiệu quả để phòng chống các cuộc tấn công SQL Injection.

* 1. Mô hình kiến trúc MVC (Model-View-Controler)

Mô hình



**Hình 1: Mô hình kiến trúc MVC – Model – View – Controller.**

Dự án áp dụng mô hình MVC để tách biệt các mối quan tâm:

* Controller (AccountController.cs, TopicsController.cs, v.v.): Chịu trách nhiệm nhận yêu cầu HTTP từ người dùng (User Request) và quyết định logic nghiệp vụ nào cần được thực hiện. Nó tương tác với Model để lấy hoặc cập nhật dữ liệu.
* Model (Topic.cs, Reply.cs, ApplicationDbContext.cs): Đại diện cho cấu trúc dữ liệu và xử lý logic liên quan đến dữ liệu (như validation, truy vấn).
* View (. cshtml Razor views): Chịu trách nhiệm trình bày giao diện người dùng dựa trên dữ liệu được Controller cung cấp qua các ViewModel.
  1. Công nghệ Frontend

Khái niệm

Frontend là phần giao diện người dùng (UI – User Interface) của một website hoặc ứng dụng web. Đây là toàn bộ những gì người dùng nhìn thấy và tương tác trực tiếp như: bố cục, màu sắc, menu, nút bấm, biểu mẫu, hiệu ứng… Công nghệ Frontend tập trung vào việc xây dựng trải nghiệm người dùng (UX) mượt mà, dễ dùng và trực quan.

Bootstrap 5.3

Cung cấp hệ thống lưới (grid system) linh hoạt và các thành phần UI dựng sẵn, đảm bảo tính responsive design cho mọi thiết bị.

CSS3 Custom Design

Được sử dụng để tạo ra giao diện UX/UI premium với các hiệu ứng hình ảnh hiện đại như glassmorphism effects và gradient backgrounds.

jQuery 3.x

Hỗ trợ các tác vụ JavaScript phía client, đặc biệt là trong việc xử lý các micro-interactions và các thao tác AJAX.

* 1. Bảo mật với ASP.NET Identity

ASP.NET Identity

ASP.NET Identity là hệ thống quản lý thành viên của Microsoft, cung cấp các tính năng xác thực (Authentication) và ủy quyền (Authorization) mạnh mẽ.

Password Hashing: Mật khẩu người dùng được lưu trữ dưới dạng hash an toàn, chống lại các cuộc tấn công Brute-force và Rainbow table.

Role-based Access Control: Phân quyền rõ ràng cho các vai trò Admin, Moderator, và User.

CSRF Protection

Sử dụng ValidateAntiForgeryToken cho tất cả các form gửi dữ liệu (POST requests) để chống lại tấn công Cross-Site Request Forgery.

XSS Protection

Razor engine tự động mã hóa (encode) các nội dung output, ngăn chặn tấn công Cross-Site Scripting bằng cách vô hiệu hóa mã độc được chèn vào.

1. HIỆN THỰC HÓA VÀ THIẾT KẾ HỆ THỐNG
   1. Giới thiệu hệ thống

Diễn đàn Sức khỏe là một nền tảng cộng đồng trực tuyến được xây dựng bằng ASP.NET Core MVC, cung cấp không gian cho người dùng chia sẻ kiến thức, kinh nghiệm, và tìm kiếm thông tin về sức khỏe. Dự án được thiết kế để giải quyết nhu cầu học hỏi từ cộng đồng về các lĩnh vực như dinh dưỡng, tập luyện, sức khỏe tinh thần và bệnh lý.

* 1. Phân tích hệ thống
* Phân tích Yêu cầu Chức năng (Functional Requirements)

| **Nhóm Chức năng** | **Chức năng Chi tiết** | **Trạng thái** |
| --- | --- | --- |
| **Hệ thống Tài khoản** | Đăng ký (qua email), Đăng nhập (với remember me), Quên mật khẩu (reset qua email), Quản lý Hồ sơ cá nhân (thông tin, ngày tham gia). | ✅ |
| **Quản lý Nội dung** | Tạo, Chỉnh sửa, Xóa chủ đề/phản hồi; Phân loại 5 chuyên mục; Hỗ trợ Rich content cơ bản; Đếm lượt xem tự động (View tracking). | ✅ |
| **Tương tác Xã hội** | Thích/Bỏ thích chủ đề và phản hồi; Tìm kiếm (tiêu đề & nội dung); Lọc theo chuyên mục; Nhận thông báo khi có phản hồi mới; Báo cáo nội dung vi phạm. | ✅ |
| **Quản trị Hệ thống** | Dashboard thống kê; Quản lý (Khóa/Mở khóa) người dùng; Xóa nội dung vi phạm; Xử lý và giải quyết các báo cáo. | ✅ |

* Yêu cầu Phi Chức năng (Non-Functional Requirements)

| **Yêu cầu** | **Mô tả** | **Phương thức Đạt được** |
| --- | --- | --- |
| **Hiệu suất** | Tải trang nhanh, xử lý truy vấn dữ liệu hiệu quả. | Tối ưu hóa truy vấn LINQ, sử dụng các pattern tối ưu của EF Core. |
| **Khả năng Mở rộng** | Dễ dàng thêm các tính năng mới trong tương lai (Roadmap). | Áp dụng kiến trúc MVC và Dependency Injection của ASP.NET Core. |
| **Bảo mật** | Chống lại các cuộc tấn công web phổ biến. | CSRF, XSS Protection, Password Hashing, HTTPS enforcement. |
| **Khả năng Sử dụng** | Giao diện trực quan, dễ tương tác trên mọi kích thước màn hình. | Responsive design (Bootstrap 5.3), UX/UI premium. |

Mô hình cơ sở dữ liệu

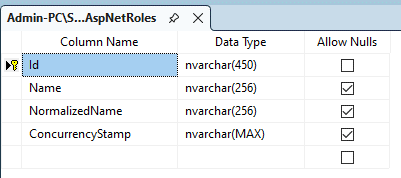
Xác định các thực thể

+Các thực thể trong hệ thống

* Thực thể Tài Khoản (User và Admin)
* Thực thể Courses. (Khóa học)
* Thực thể Lessons (bài giảng)
* Thực thể CourseProgresses (tiến độ)
* Thực thể Reviews (đánh giá nhận xét)
* Thực thể Orders (đơn mua)

+Thành phần của các thực thể

1. ***Thực thể Tài Khoản (User và Admin)***

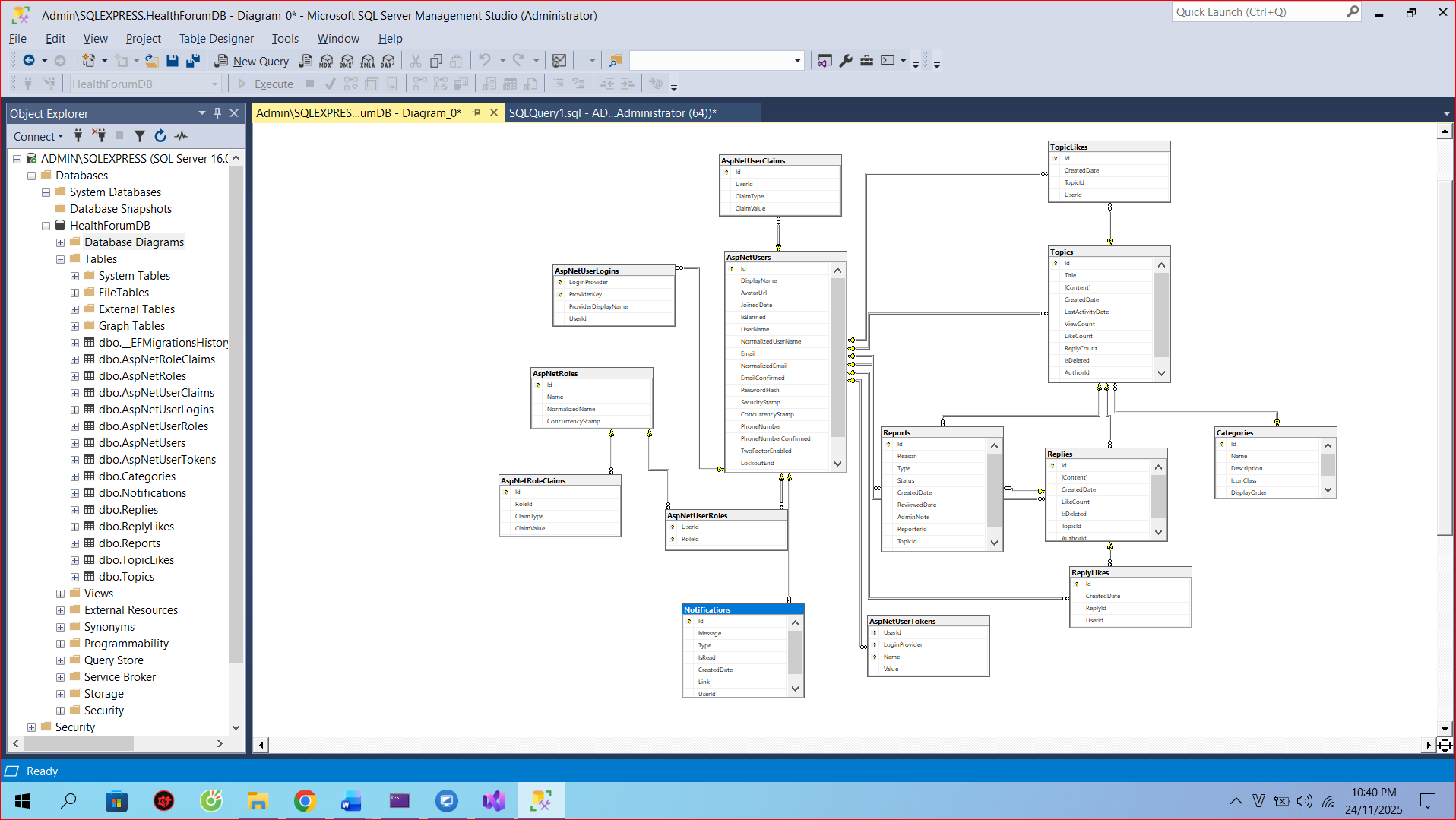
****

**Hình 2:** **Thực thể Tài Khoản**

- Các thuộc tính của thực thể:

* Id: khóa chính (phân biệt Admin, User, Teacher,…)
* Name: Tên đăng nhập (Dùng để đăng nhập)
* NormallizedName: tên (dùng để tìm kiếm)
* ConcurrencyStamp: kiểm tra xung đột dữ liệu.

Mô hình cơ sở dữ liệu

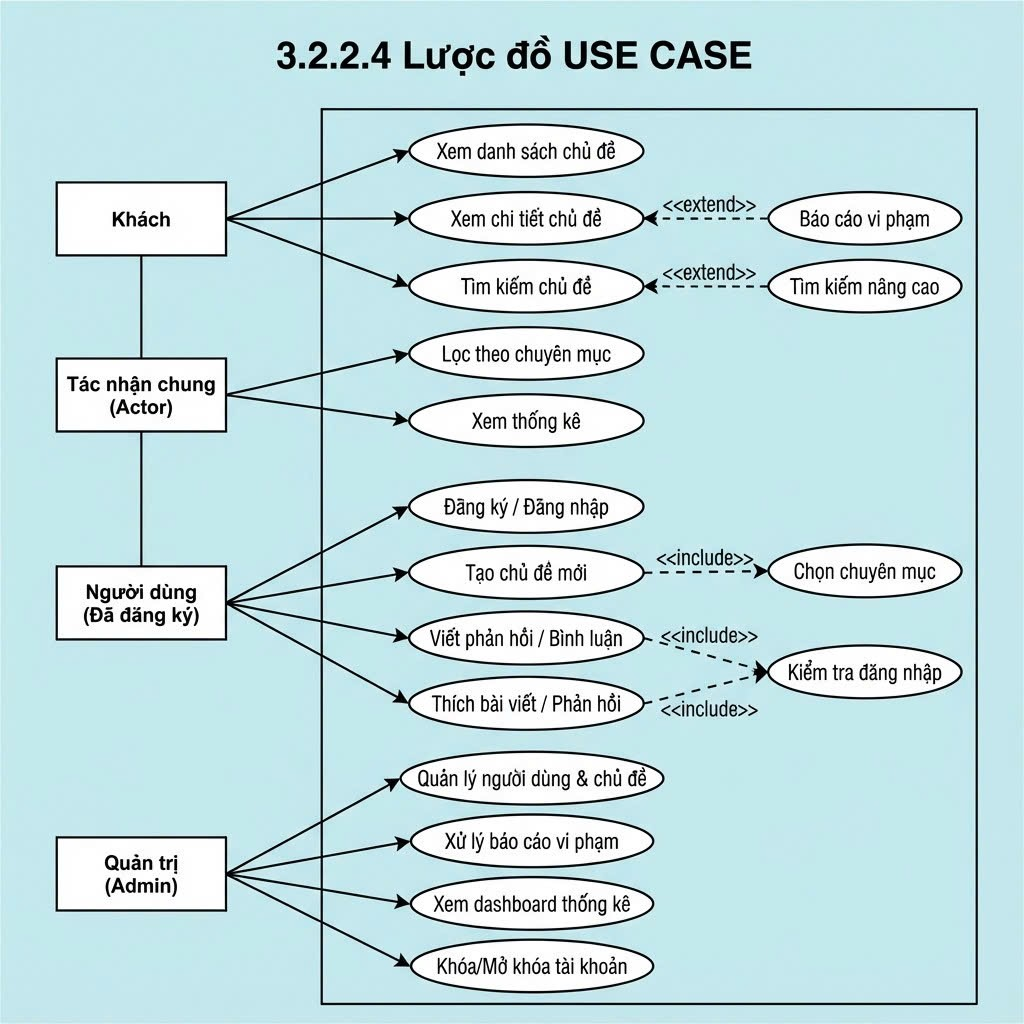


**Hình 3: Lược đồ cơ sở dữ liệu**

Các trường hợp sử dụng ( Use Case)

Biểu đồ Use Case (Use Case Diagram)

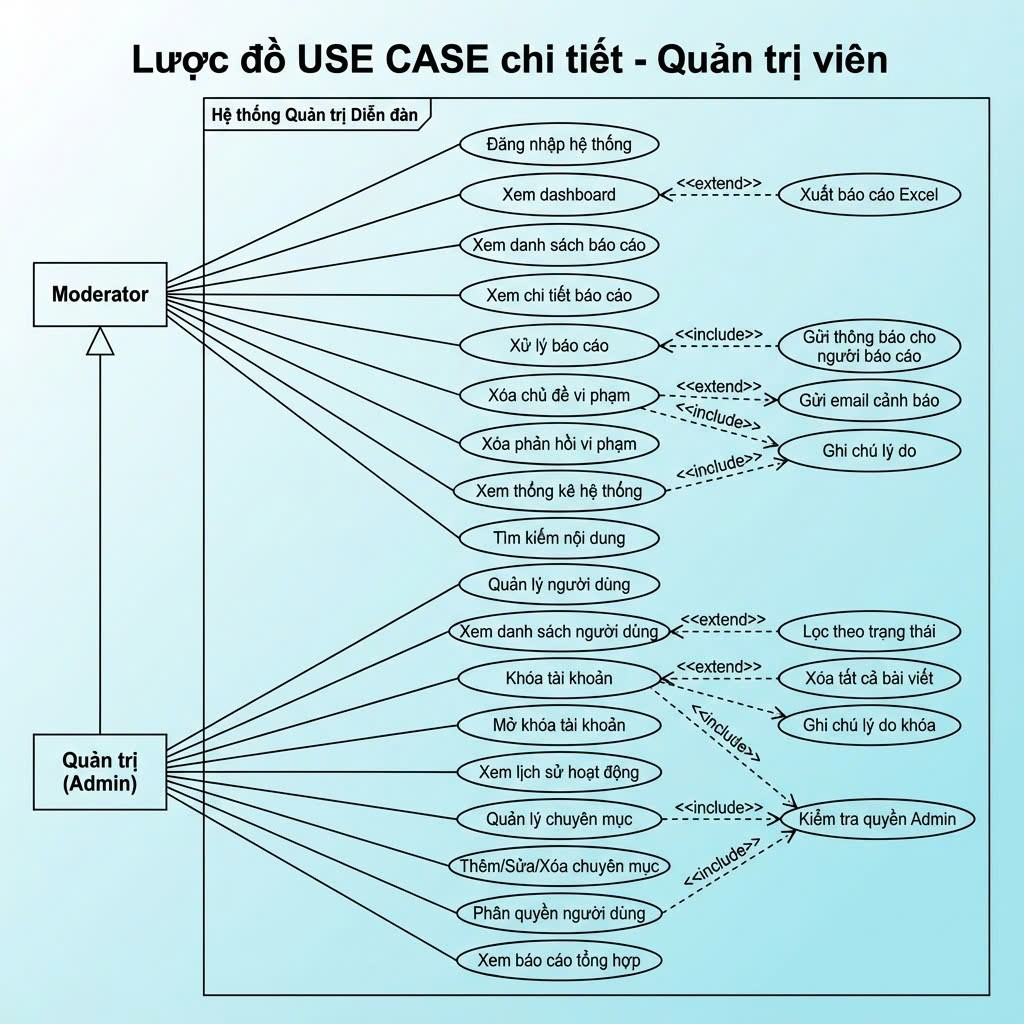
* **Mục đích:** Xác định các chức năng chính của hệ thống và cách người dùng (actors) tương tác với các chức năng đó.
* **Các Actors chính:**
  + **Khách hàng (Customer/Guest):** Người dùng chưa đăng nhập hoặc đã đăng nhập.
  + **Quản trị viên (Administrator):** Người quản lý toàn bộ hệ thống.



**Lược đồ USE CASE**

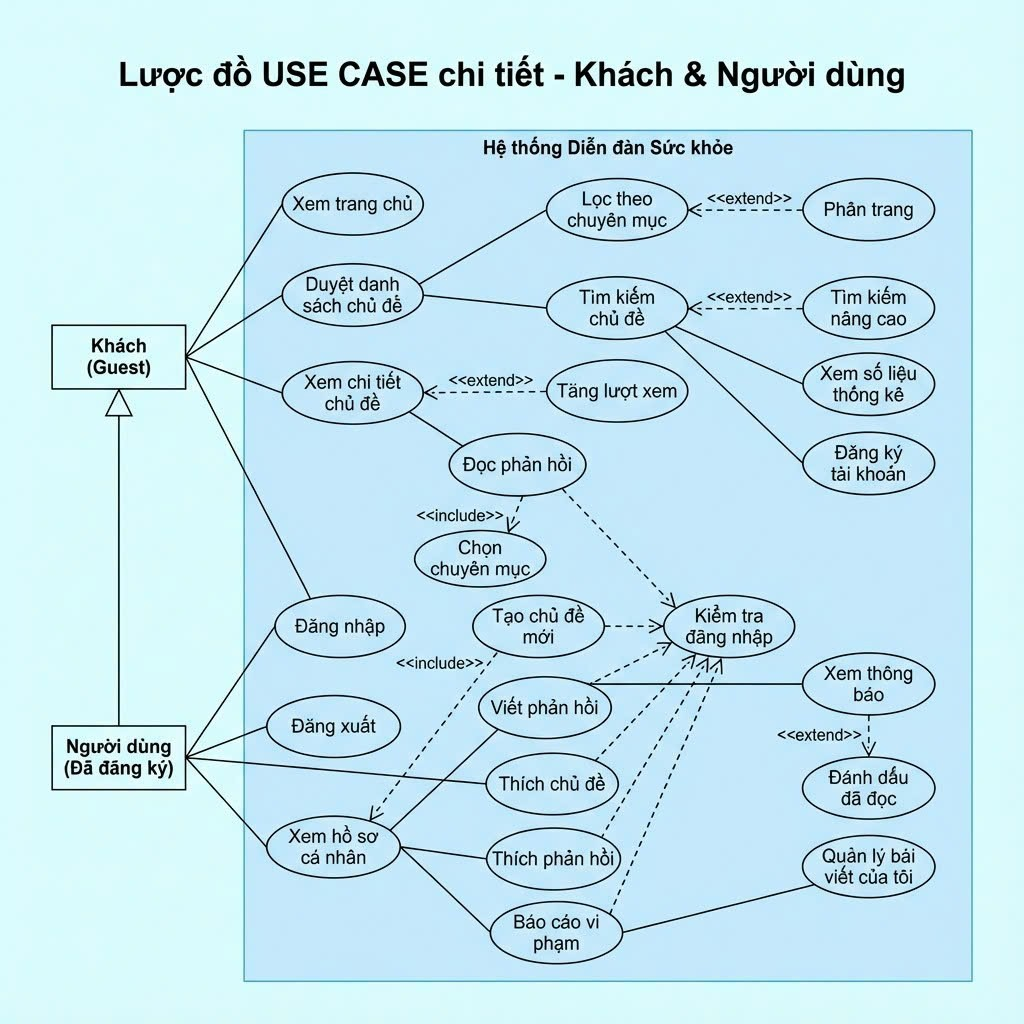
**Hình 4: Lược đồ USE CASE**

**Quản trị viên:**



**Hình 5: Lược đồ USE CASE Quản trị viên (Admin)**

**Khách hàng:**



**Hình 6: Lược đồ USE CASE Khách và người dùng**

Phân tầng ứng dụng

Tầng Trình bày (Presentation Layer): Bao gồm Views/ và wwwroot/. Chứa mã Razor HTML và các file tĩnh (CSS, JS, Fonts).

Tầng Logic Nghiệp vụ (Business Logic Layer): Bao gồm Controllers/. Chứa các lớp xử lý nghiệp vụ chính của ứng dụng (e.g., xác thực, tạo bài viết, kiểm duyệt).

Tầng Truy cập Dữ liệu (Data Access Layer): Bao gồm Models/ (DbContext) và Migrations/. Sử dụng EF Core để kết nối và thao tác với Database thông qua LINQ.

Cấu trúc Project

Cấu trúc thư mục tuân thủ chuẩn MVC của ASP.NET Core:

* Controllers/: Logic nghiệp vụ (Account, Topics, Admin,).
* Models/: Các Data Model (ApplicationUser, Topic, Reply,).
* ViewModels/: Các đối tượng dùng để truyền dữ liệu giữa Controller và View.
* Views/: Các file giao diện (. cshtml).
* wwwroot/: Tài nguyên tĩnh (CSS, JS, libs).

| **UC#** | **Tên Use Case** | **Actor** | **Mô tả Lưu đồ** |
| --- | --- | --- | --- |
| **UC1** | **Tìm kiếm thông tin sức khỏe** | Người dùng chưa đăng ký | Truy cập trang chủ $\rightarrow$ Xem/Lọc chủ đề $\rightarrow$ Sử dụng tìm kiếm để tìm thông tin $\rightarrow$ Đọc bài viết. |
| **UC2** | **Đặt câu hỏi về sức khỏe** | Người dùng đã đăng nhập | Đăng nhập $\rightarrow$ Click "Tạo chủ đề mới" $\rightarrow$ Chọn chuyên mục $\rightarrow$ Nhập nội dung $\rightarrow$ Đăng bài $\rightarrow$ Chờ phản hồi. |
| **UC4** | **Kiểm duyệt nội dung** | Admin/Moderator | Đăng nhập Admin $\rightarrow$ Xem dashboard $\rightarrow$ Kiểm tra/Xem xét báo cáo $\rightarrow$ Quyết định xóa hoặc giữ lại nội dung $\rightarrow$ Cập nhật trạng thái báo cáo. |

Phân quyền và vai trò (Role-based Authorization)

Hệ thống sử dụng ASP.NET Identity để quản lý ba vai trò chính:

**User (Người dùng thường):** Quyền cơ bản là tạo, đọc, cập nhật nội dung của chính mình, tương tác (Like, Reply, Report) với nội dung người khác.

**Moderator:** Có thêm quyền kiểm duyệt: Xóa các phản hồi không phù hợp, Quản lý (xử lý) các báo cáo nội dung vi phạm.

**Admin:** Quyền cao nhất, bao gồm quyền của Moderator cộng thêm quyền Quản lý người dùng (Khóa/Mở khóa tài khoản) và toàn quyền truy cập Admin Dashboard.

Thiết kế Cơ sở Dữ liệu (Database Schema)

Sử dụng SQL Server LocalDB với các bảng chính được ánh xạ qua EF Core Models.

**Bảng chính**

| **Bảng** | **Mô tả** | **Liên kết Quan hệ** |
| --- | --- | --- |
| **AspNetUsers** | Lưu trữ thông tin danh tính và vai trò. | 1: N với Topics, Replies, Reports, TopicLikes, ReplyLikes |
| **Topics** | Bài viết/Chủ đề do người dùng tạo. | N:1 với Categories; 1: N với Replies, TopicLikes |
| **Replies** | Bình luận cho một chủ đề cụ thể. | N:1 với Topics; 1: N với ReplyLikes |
| **Categories** | 5 chuyên mục chính của diễn đàn (Dinh dưỡng, Tập luyện,). | 1: N với Topics |

1. TRIỂN KHAI VÀ KẾT QUẢ NGHIÊN CỨU
   1. Quy trình cài đặt và khởi chạy

Quy trình đơn giản hóa nhờ sử dụng .NET Core CLI và SQL Server LocalDB

-Môi trường: Đảm bảo cài đặt .NET 9.0 SDK và SQL Server LocalDB.

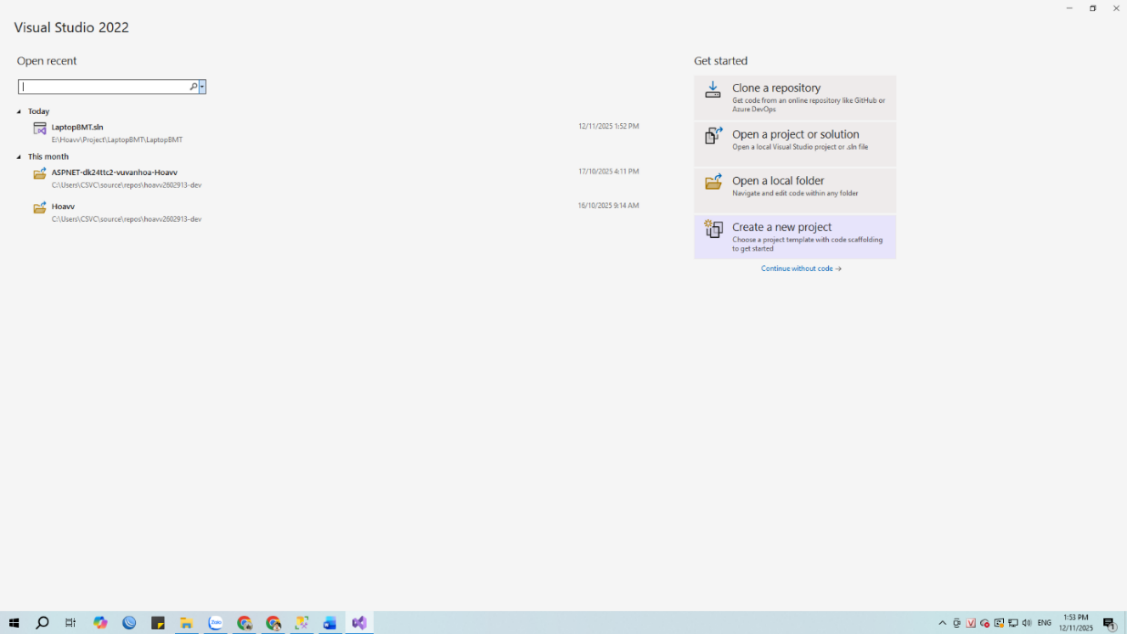
-Clone Source Code: Sử dụng lệnh git clone <repository-url>.

-Khôi phục Packages: Chạy dotnet restore để tải các dependencies (NuGet packages).

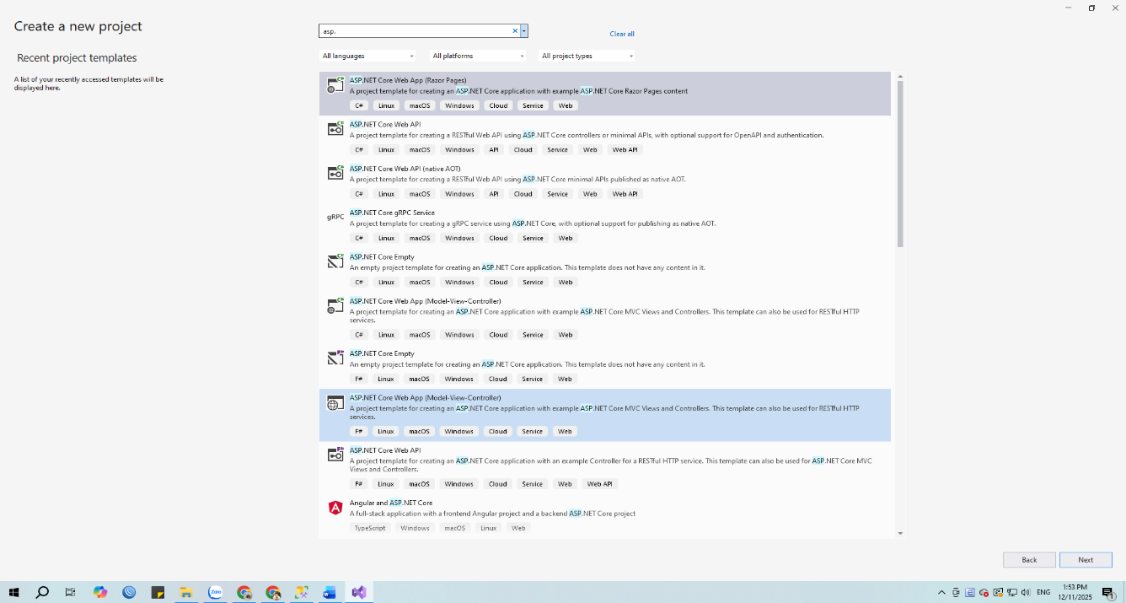
-Chạy Ứng dụng: Khởi động bằng dotnet run. Ứng dụng sẽ tự động chạy tại cổng http://localhost:5292.

-Tài khoản Mặc định: Hệ thống cung cấp tài khoản mẫu để kiểm thử: Admin (admin@healthforum.com / Admin@123) và User (nguyenvana@example.com / User@123).

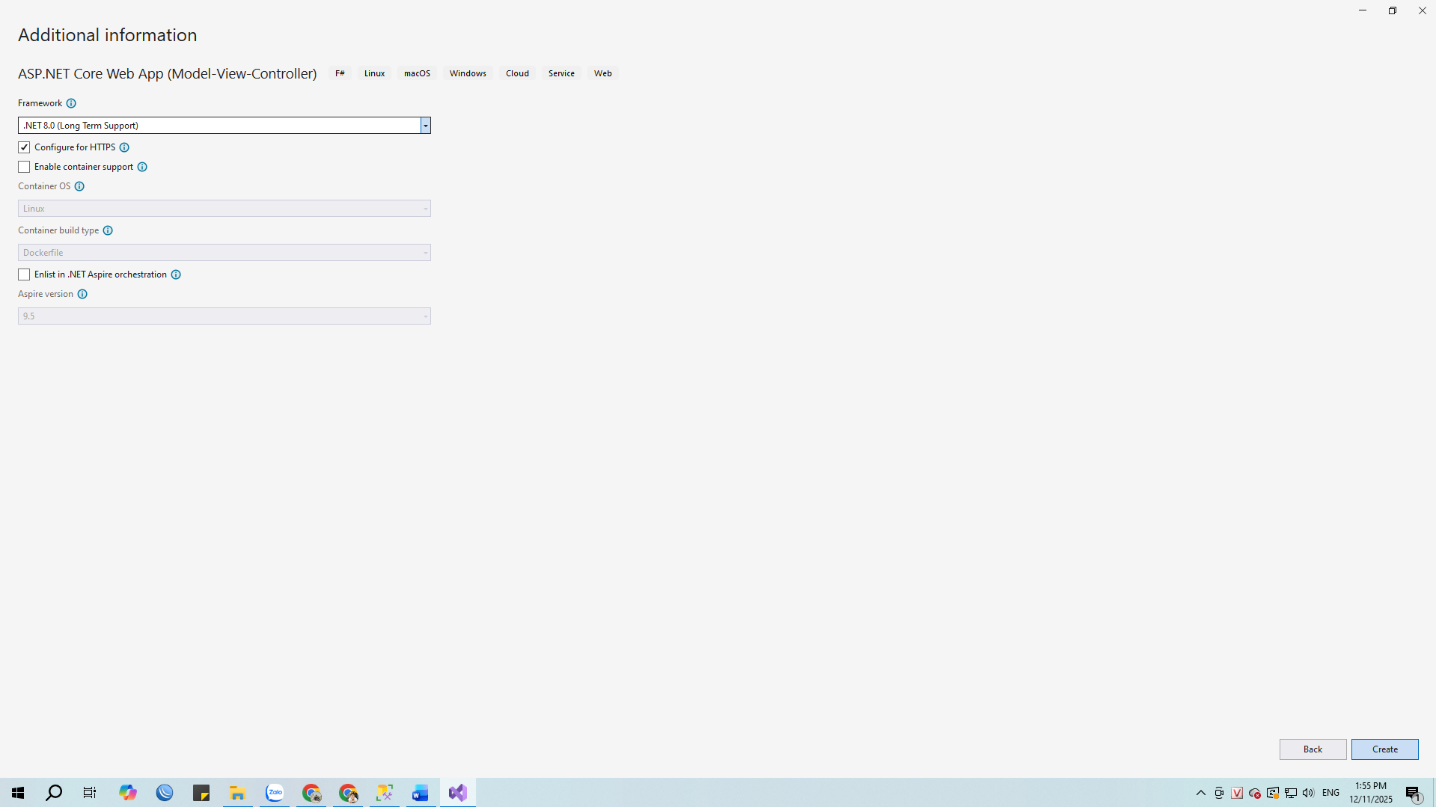
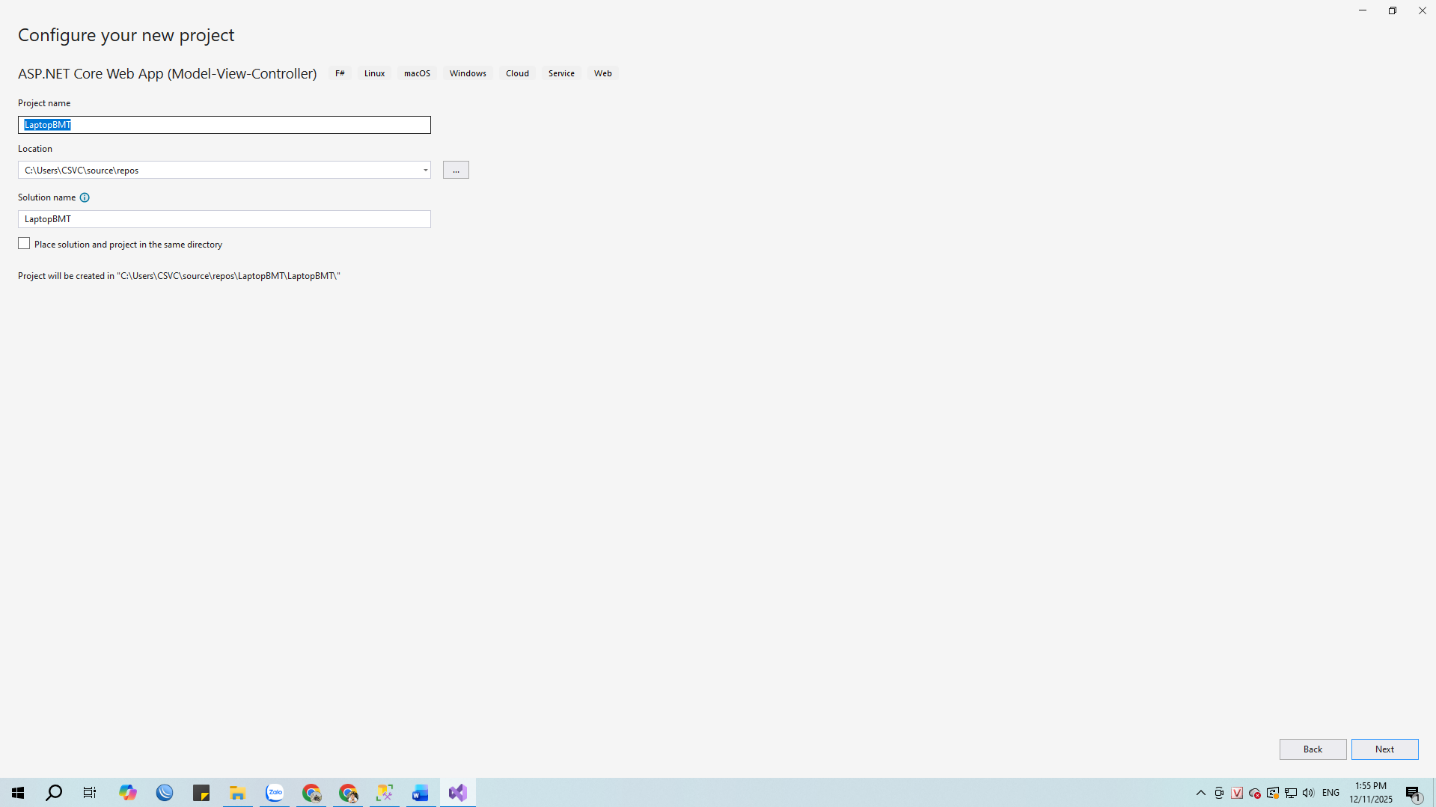
Tạo dự án với ASP.NET MVC

Bước 1: Mở Visual Studio Create a New Project

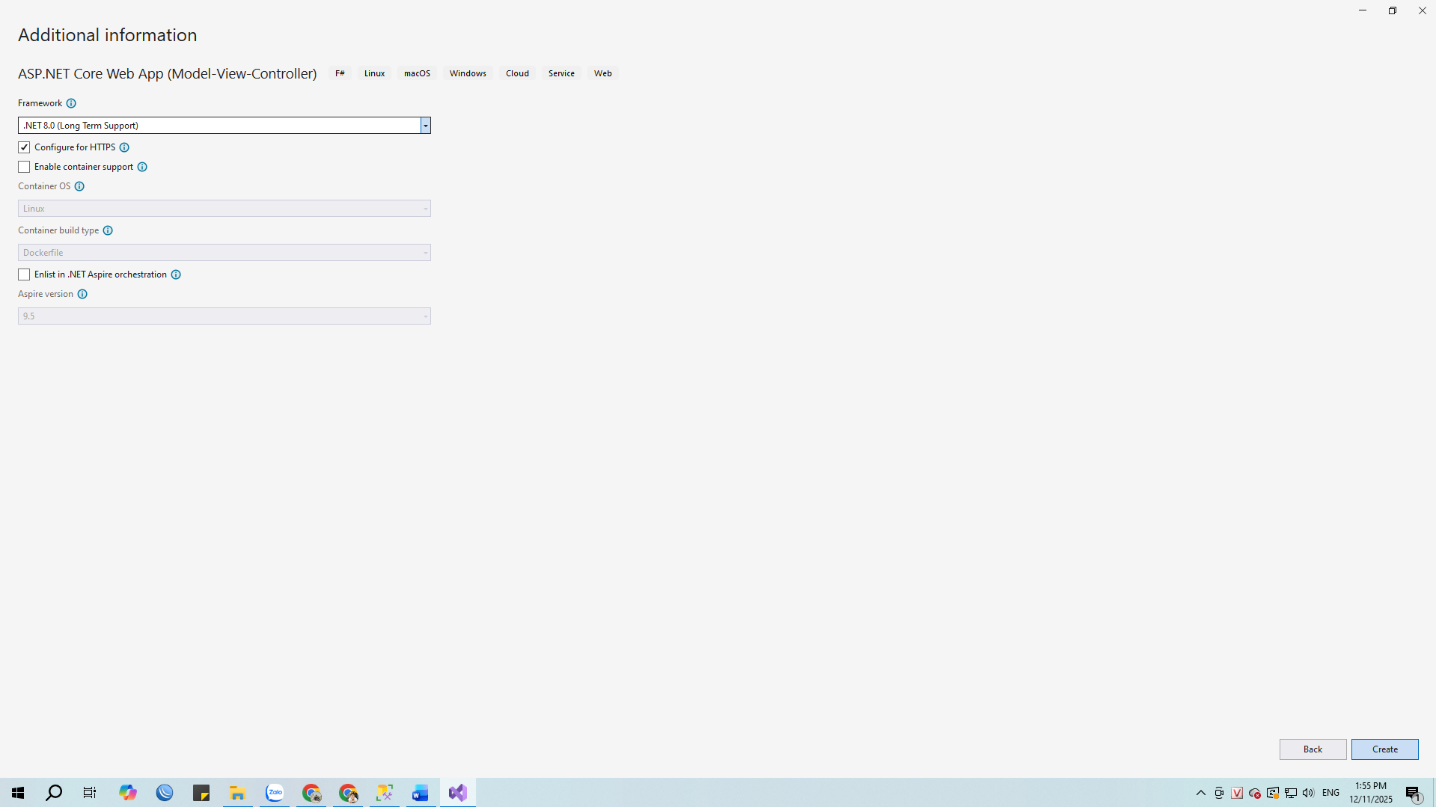
**Hình 7: Tạo Project mới trên Visual Studio**

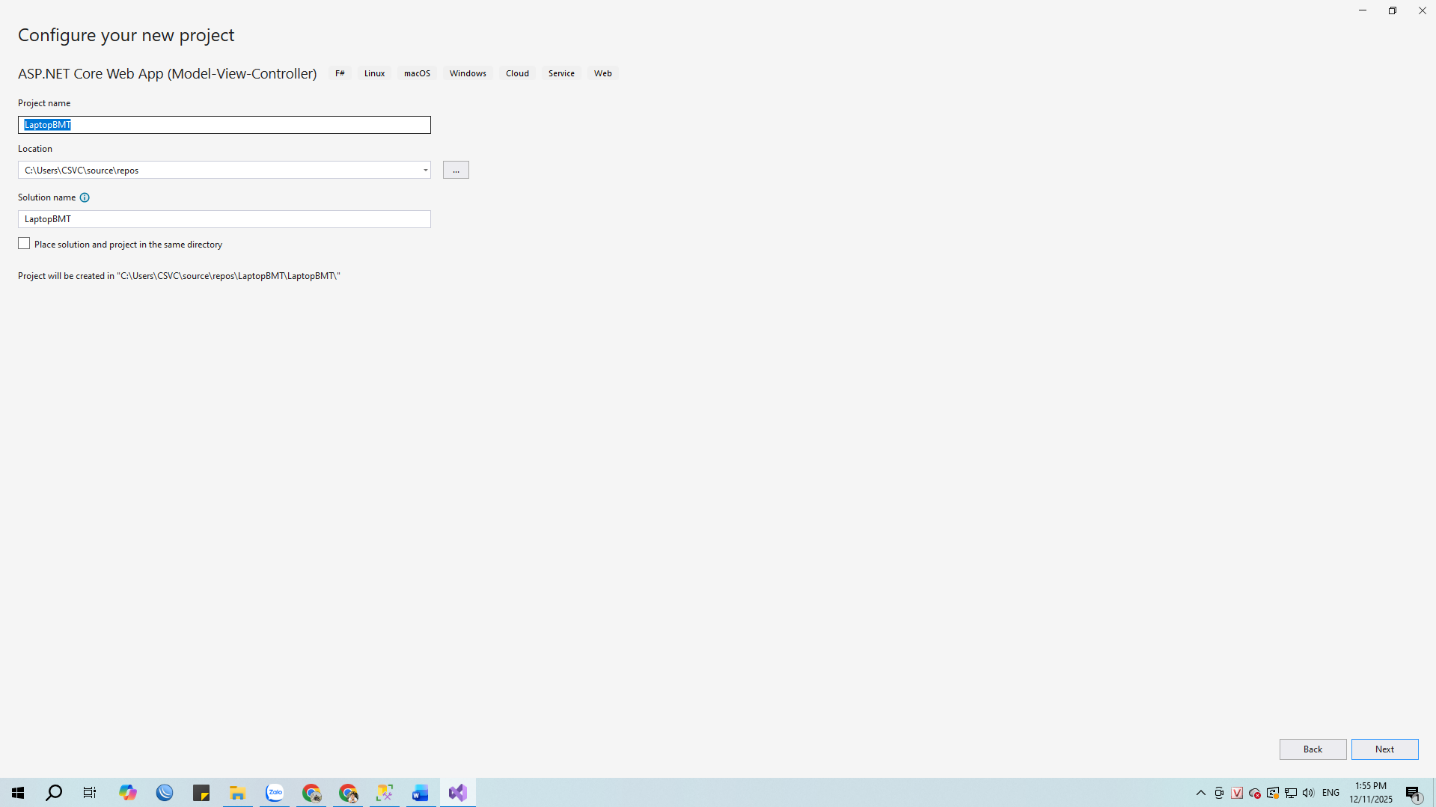
Bước 2: Chọn ASP.NET Core Web App (Model-View-Controller)

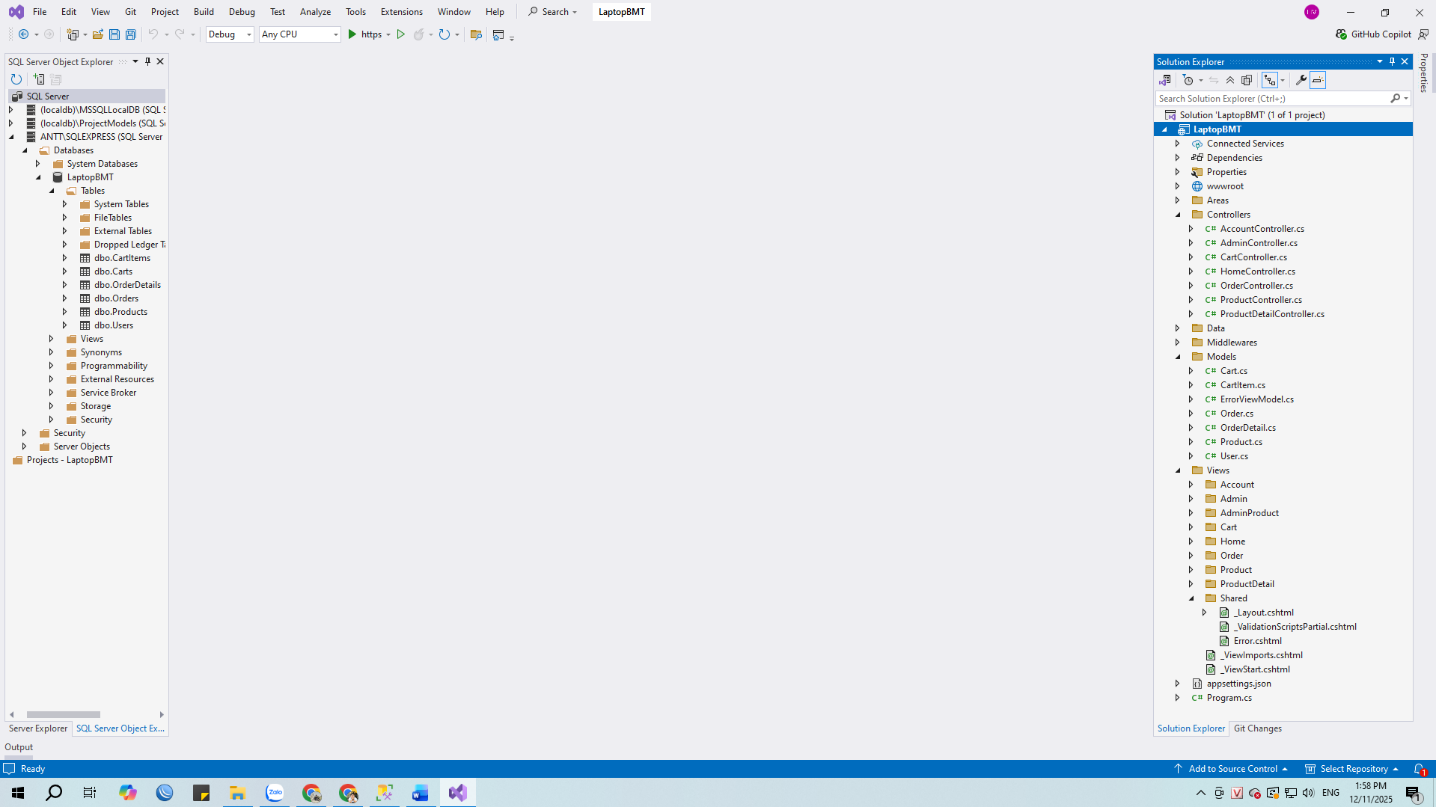
**Hình 8: Chọn ASP.NET Core Web App (Model-View-Controller)**

Bước 3: Đặt tên Project, đường dẫn thư mục lưu và nhấn “Next”

**Hình 9:Đặt tên cho Project và đường dẫn lưu**

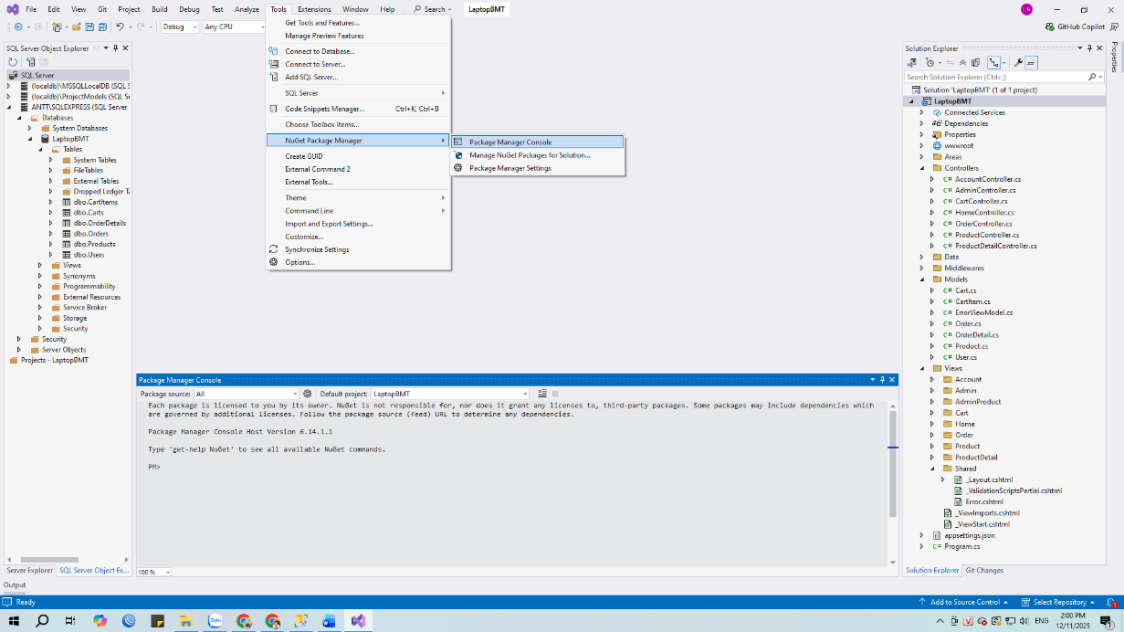
Bước 4: Nhấn “Create” để tiếp tục

**Hình 10: Nhấn chọn Create******

Cấu trúc cây thư mục của ASP.NET MVC

**Hình 11: Giao diện làm việc và cấu trúc cây thư mục của ASP.NET MVC**

* 1. Entity Framework

Vào Tools NuGet Package Manager Package Manager Console để có thể sử dụng dòng lệnh để cài đặt EF Core và provider cho SQL Server.

**Hình 12:** **Package Manager Console sử dụng dòng lệnh**

* 1. Thiết kế Giao diện Người dùng (UX/UI)

Giao diện được thiết kế để tạo cảm giác hiện đại, cao cấp (premium), tập trung vào trải nghiệm người dùng.

Sử dụng Hero section nổi bật với màu sắc gradient và các nút CTA rõ ràng. Danh sách chủ đề được trình bày một cách gọn gàng, nhấn mạnh vào metadata (lượt xem, tương tác) để hướng người dùng đến nội dung chất lượng.

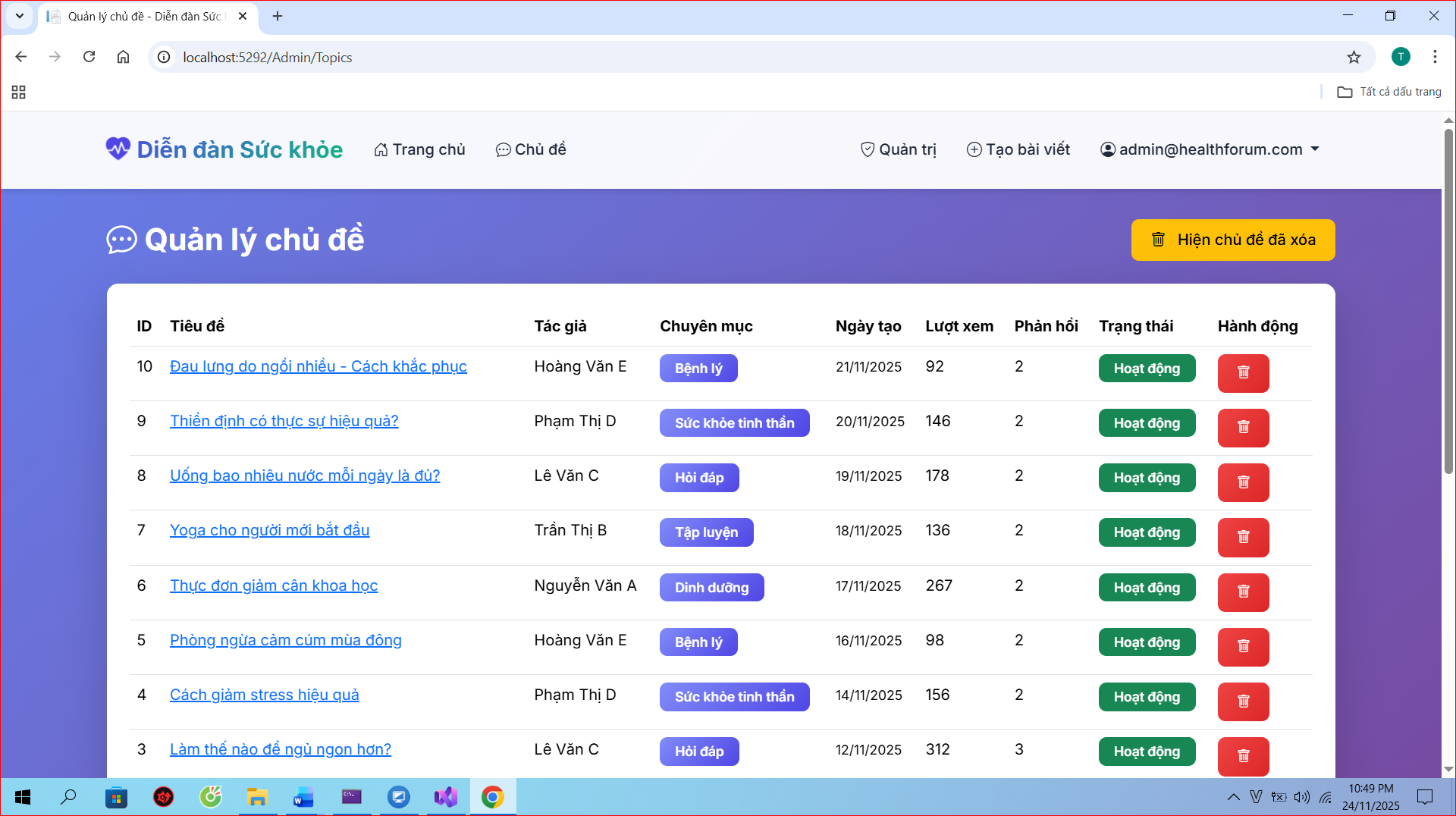
*Trang chủ của Diễn đàn Sức Khỏe* là nơi người dùng có thể xem tổng quan các chủ đề đang được thảo luận, tìm kiếm thông tin và truy cập nhanh vào các chuyên mục sức khỏe. Giao diện được thiết kế trực quan, dễ sử dụng cho mọi nhóm người dùng.

A screenshot of a computer

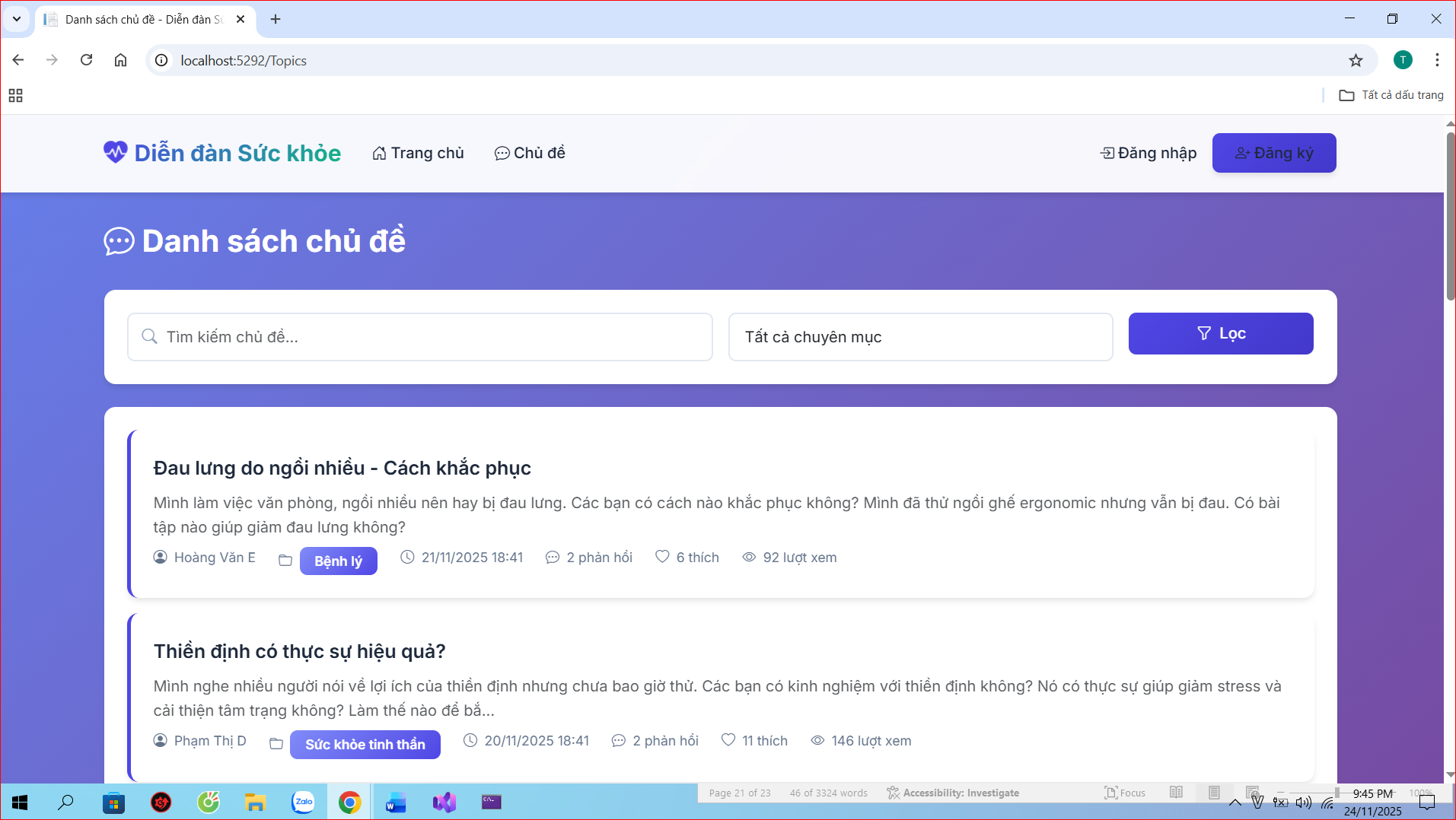
AI-generated content may be incorrect.

**Hình 13: Trang chủ**

*Trang chi tiết chủ đề*: Được thiết kế để tối ưu hóa việc đọc. Các chức năng tương tác như Like và Report được bố trí dễ tìm thấy. Phần phản hồi được tổ chức theo luồng thảo luận, khuyến khích sự chia sẻ kinh nghiệm.



**Hình 14: Giao diện quản lý chủ đề**



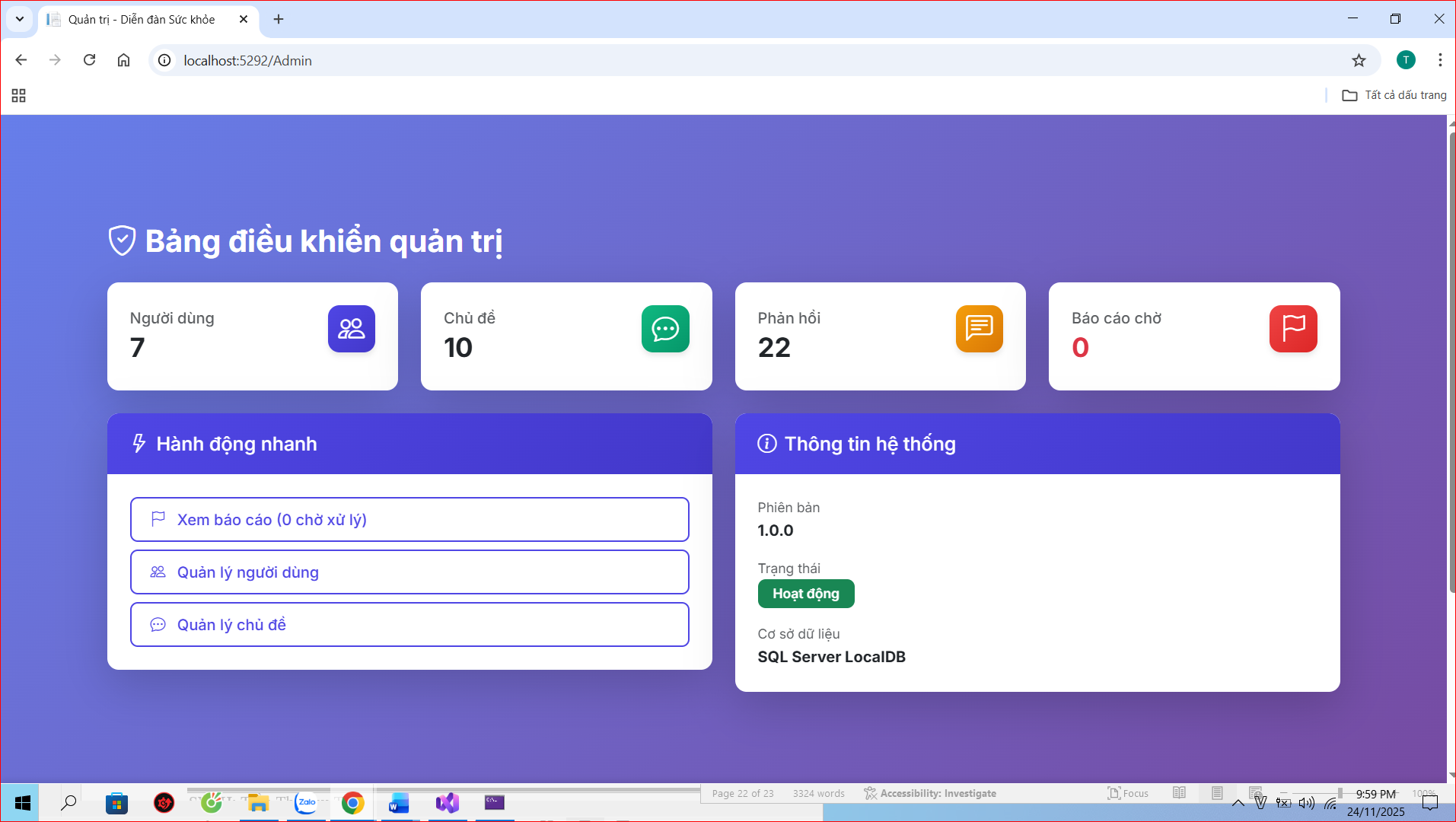
**Hình 15: Giao diện danh sách chủ đề**

Thiết kế Giao diện Quản trị (Admin Dashboard)

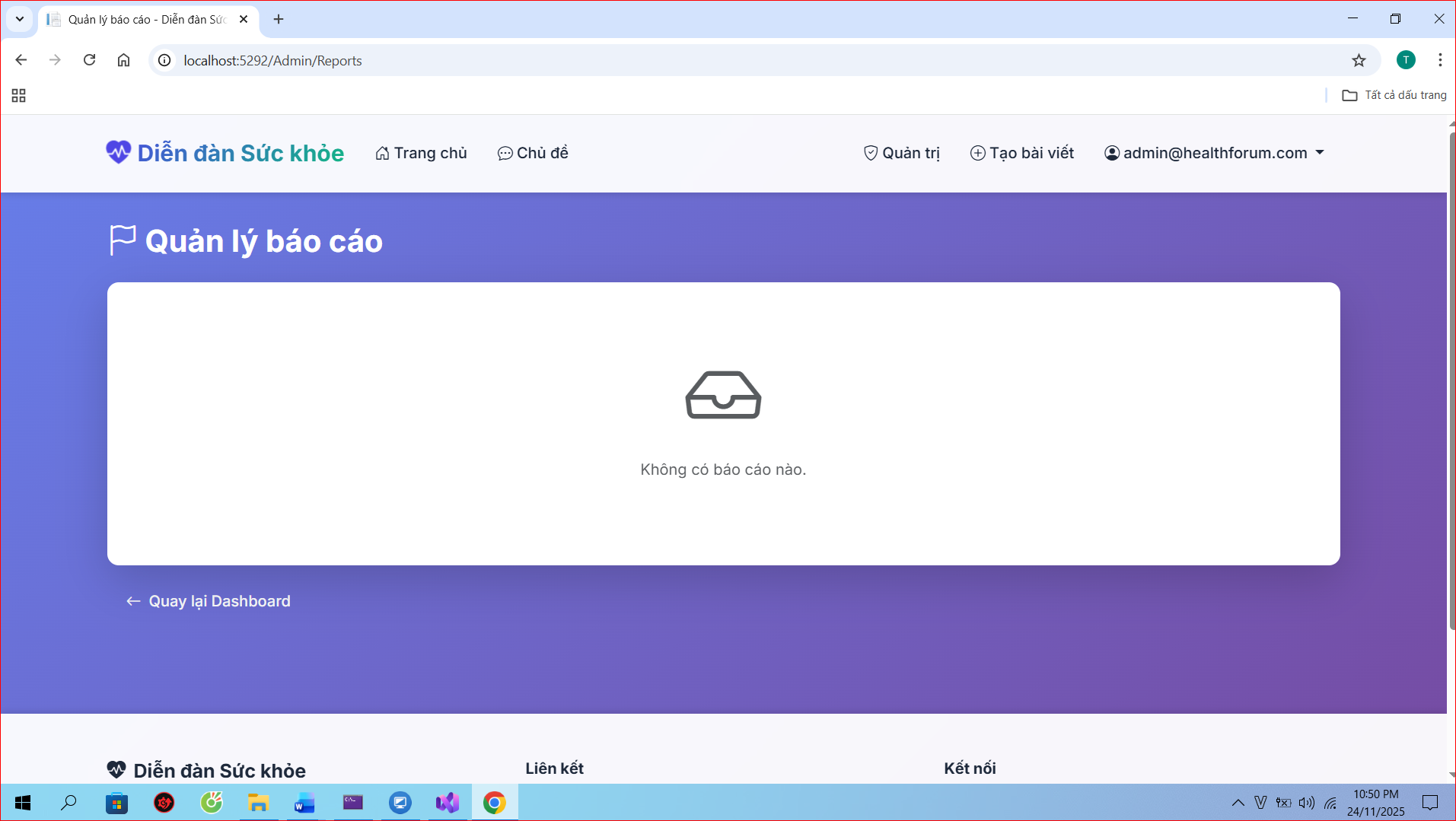
Dashboard được thiết kế tập trung vào chức năng và hiệu quả kiểm duyệt:

Tổng quan Thống kê: Cung cấp cái nhìn tổng thể về sức khỏe của diễn đàn (User, Topic, Report counts).

Quản lý Báo cáo: Đây là chức năng cốt lõi. Giao diện hiển thị các báo cáo với ưu tiên cao, cho phép Admin/Moderator xem nội dung vi phạm và đưa ra quyết định nhanh chóng (xóa hoặc bỏ qua).



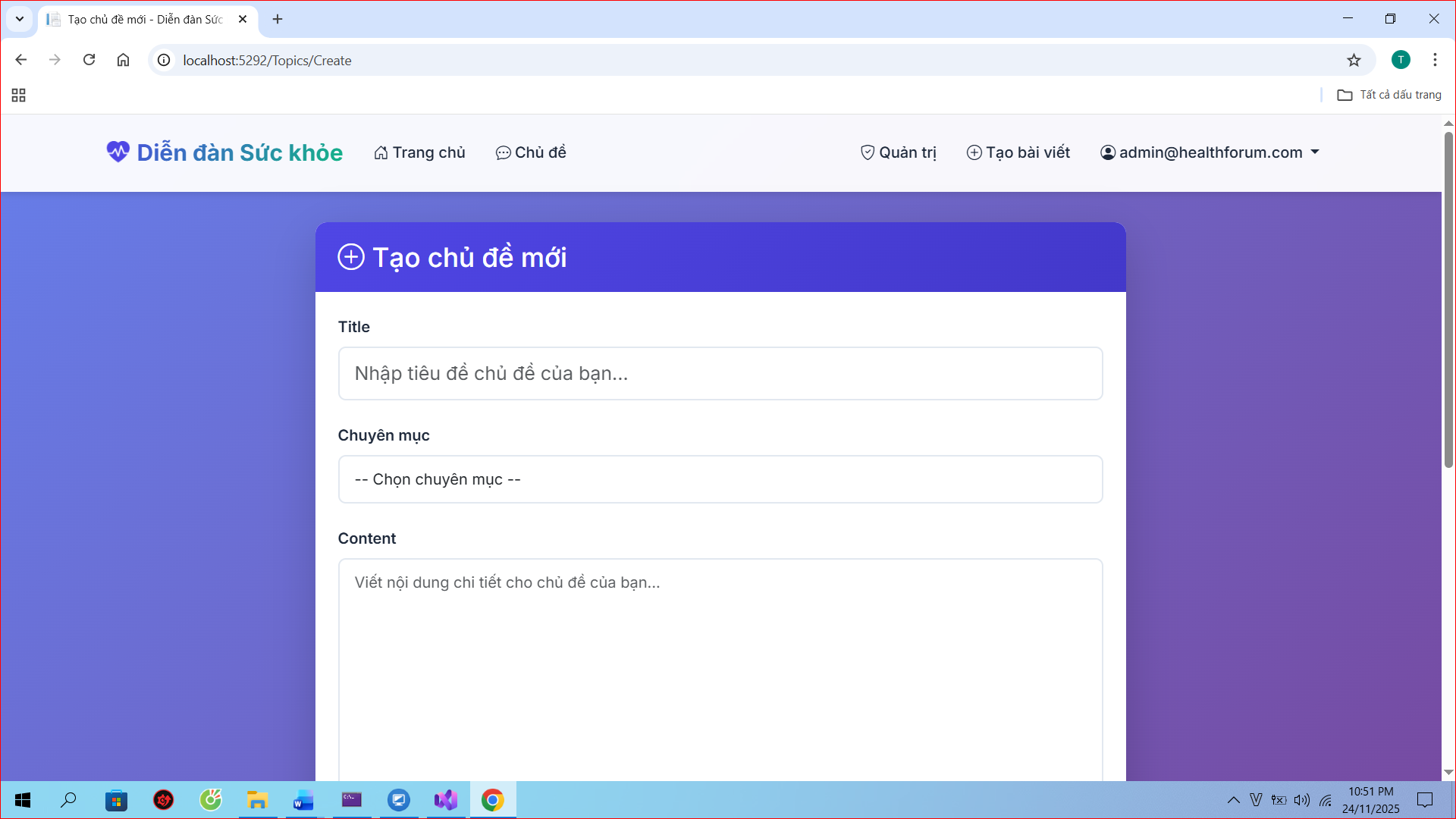
**Hình 16: Giao diện Admin (tổng quan thống kê)**



**Hình 17: Giao diện quản lý báo cáo**

Thiết kế Giao diện tạo thêm chủ đề

Giao diện tạo thêm chủ đề được thiết kế nhằm giúp người dùng đăng một chủ đề mới lên diễn đàn sức khỏe. Trang này có bố cục rõ ràng, đơn giản, dễ thao tác đối với mọi đối tượng người dùng.



**Hình 18: Giao diện tạo thêm chủ đề**

1. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN
   1. Kết luận:

Dự án Diễn đàn Sức khỏe là một minh chứng thành công cho việc ứng dụng công nghệ ASP.NET Core 9.0 MVC để xây dựng một nền tảng cộng đồng hiện đại, an toàn và hiệu suất cao. Giai đoạn MVP đã hoàn thành đầy đủ các yêu cầu chức năng cốt lõi.

* 1. Hạn chế:

Chưa xử lý một số lỗi bảo mật cho trang web.

Thiết kế cơ sở dữ liệu chưa được hoàn chỉnh và đầy đủ, chuyên nghiệp cho một trang web thương mại điện tử bán Laptop.

* 1. Hướng phát triển:

Upload ảnh trực tiếp cho bài viết; Tích hợp Rich text editor (TinyMCE/CKEditor); Hệ thống Email notifications; Tags và Hashtags cho chủ đề để nâng cao trải nghiệm người dùng và chất lượng nội dung.

Real-time chat (sử dụng SignalR); Hệ thống điểm và badges (Gamification); API endpoints (RESTful) cho bên thứ ba; Phát triển Mobile app (React Native/Flutter) để mở rộng tương tác thời gian thực và khả năng tiếp cận.

Bảo mật nâng cao cho trang web.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] Get started with ASP.NET Core MVC : <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio>

[2] Get started with Bootstrap :<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

[3] Chuyên đề ASP.NET- Tài liệu tham khảo của khóa học

**Github** : <https://github.com/ASPNET-DK24TTC2-tranthithutien-THUTIEN/Xay-dung-website-dien-dan-suc-khoe>