

## **ĐỒ ÁN TỐT NGHIỆP**

### **WEBSITE DIỄN ĐÀN DAISY FORUM**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : (1) Thầy Trần Đăng Khoa  
(2) Ths Nguyễn Huy Cường

Sinh viên thực hiện : Nguyễn Phú Đức

MSSV: 1911066051 Lớp: 19DTHE4

## LỜI CẢM ƠN

Lời cảm ơn đến các thầy cô trong khoa đã giúp đỡ, quan tâm và theo dõi quá trình học tập và nghiên cứu của em trong suốt thời gian qua. Em rất biết ơn sự quan tâm và động viên từ các thầy cô đã giúp em vượt qua những thử thách trong quá trình học tập và nghiên cứu.

Đặc biệt, em muốn gửi lời cảm ơn chân thành đến 2 thầy giảng viên hướng là thầy Trần Đăng Khoa và thầy Nguyễn Huy Cường, thầy đã giúp đỡ em rất nhiều trong quá trình thực hiện đồ án tốt nghiệp. Cả hai thầy đã dành thời gian quý báu để hướng dẫn em trong từng bước của đồ án, tận tình giải đáp các thắc mắc và đưa ra những gợi ý quý giá về phương pháp nghiên cứu và giải quyết vấn đề.

Nhờ sự hỗ trợ và tư vấn thầy, em đã có được những kiến thức, kỹ năng và kinh nghiệm quan trọng để hoàn thành đồ án tốt nghiệp. Em cảm thấy may mắn khi được học tập và làm việc dưới sự hướng dẫn của quý thầy tận tâm và giàu kinh nghiệm như vậy.

Em xin gửi lời cảm ơn sâu sắc đến quý thầy, vì đã đóng góp không nhỏ vào sự thành công của đồ án tốt nghiệp của em. Em sẽ luôn mang trong mình những giá trị mà 2 thầy đã truyền đạt, và sẽ cố gắng áp dụng kiến thức và kinh nghiệm đó để phát triển sự nghiệp trong tương lai.

## LỜI CAM ĐOAN

Lời cảm ơn đến các thầy cô trong khoa đã giúp đỡ, quan tâm và theo dõi quá trình học tập và nghiên cứu của em trong suốt thời gian qua. Em rất biết ơn sự quan tâm và động viên từ các thầy cô đã giúp em vượt qua những thử thách trong quá trình học tập và nghiên cứu.

Đặc biệt, em muốn gửi lời cảm ơn chân thành đến 2 thầy giảng viên hướng là thầy Trần Đăng Khoa và thầy Nguyễn Huy Cường, thầy đã giúp đỡ em rất nhiều trong quá trình thực hiện đồ án tốt nghiệp. Cả hai thầy đã dành thời gian quý báu để hướng dẫn em trong từng bước của đồ án, tận tình giải đáp các thắc mắc và đưa ra những gợi ý quý giá về phương pháp nghiên cứu và giải quyết vấn đề.

Nhờ sự hỗ trợ và tư vấn thầy, em đã có được những kiến thức, kỹ năng và kinh nghiệm quan trọng để hoàn thành đồ án tốt nghiệp. Em cảm thấy may mắn khi được học tập và làm việc dưới sự hướng dẫn của quý thầy tận tâm và giàu kinh nghiệm như vậy.

Em xin gửi lời cảm ơn sâu sắc đến quý thầy, vì đã đóng góp không nhỏ vào sự thành công của đồ án tốt nghiệp của em. Em sẽ luôn mang trong mình những giá trị mà 2 thầy đã truyền đạt, và sẽ cố gắng áp dụng kiến thức và kinh nghiệm đó để phát triển sự nghiệp trong tương lai.

# MỤC LỤC

<i>LỜI CẢM ƠN</i>	2
<i>PHỤ LỤC</i>	1
<i>CHƯƠNG 1: TỔNG QUAN</i>	3
1. LÝ DO CHỌN ĐỀ TÀI.....	3
2. MỤC TIÊU ĐỀ TÀI.....	3
3. TỔNG QUAN/CƠ SỞ LÝ LUẬN.....	4
4. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU.....	4
5. CÁC CÔNG NGHỆ VÀ KỸ THUẬT LẬP TRÌNH.....	5
5.1. Mô hình phát triển.....	5
5.2. Design pattern.....	7
5.3. Công nghệ sử dụng.....	8
5.4. Git.....	17
<i>CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ</i>	20
1. ĐẶC TẢ YÊU CẦU.....	20
2. THUỘC TÍNH THỰC THẾ.....	21
3. MÔ TẢ THUỘC TÍNH.....	22
3.1. Bảng thuộc tính Users.....	22
3.2. Bảng thuộc tính Categories.....	22
3.3. Bảng thuộc tính Knowledge Bases.....	23
3.4. Bảng thuộc tính Attachments.....	23
3.5. Bảng thuộc tính Reports.....	24
3.6. Bảng thuộc tính Comments.....	24
3.7. Bảng thuộc tính Roles.....	25
3.8. Bảng thuộc tính User Roles.....	25
3.9. Bảng thuộc tính Functions.....	25
3.10. Bảng thuộc tính Commands.....	26
3.11. Bảng thuộc tính Command Functions.....	26
3.12. Bảng thuộc tính Permissions.....	26
3.13. Bảng thuộc tính Votes.....	26
3.14. Bảng thuộc tính Labels.....	27
3.15. Bảng thuộc tính Label In Knowledge Bases.....	27

3.16. Bảng thuộc tính CacheTable.....	27
<b>4. SƠ ĐỒ.....</b>	<b>28</b>
4.1. Sơ đồ Use Case.....	28
4.2. Sơ đồ ERD.....	29
4.3. Sơ đồ Class.....	30
<b>CHƯƠNG 3: XÂY DỰNG WEBSITE VÀ KIỂM THỬ</b>	<b>31</b>
1. THỰC HIỆN.....	31
1.1. Backend API.....	31
1.2. Frontend.....	31
2. THUẬT TOÁN.....	33
2.1. Thuật toán đề xuất bài viết Contend Based.....	33
3. KIỂM THỬ	36
3.1. Kiểm thử giá trị nhập vào bằng UnitTest.....	36
3.2. Kiểm thử logic của chức năng bằng UnitTest.....	39
4. KẾT QUẢ.....	41
4.1. Backend Server.....	41
4.3. Web Portal.....	45
5. TỐI UU WEBSITE.....	49
5.1. Tối ưu tốc độ với Distributed Cache.....	49
5.2. Tối ưu bảo mật với Security headers.....	49
<b>CHƯƠNG 4: KẾT LUẬN</b>	<b>50</b>
1. KẾT LUẬN.....	50
2. ĐÁNH GIÁ.....	50
4.1. Tính mới.....	50
4.2. Đóng góp.....	50
3. HƯỚNG PHÁT TRIỂN.....	51
<b>TÀI LIỆU THAM KHẢO</b>	<b>52</b>

# **CHƯƠNG 1: TỔNG QUAN**

## **1. LÝ DO CHỌN ĐỀ TÀI**

Website diễn đàn là một công cụ rất quan trọng để kết nối và giao lưu giữa các thành viên trong một cộng đồng. Nó cho phép mọi người tham gia vào cuộc trò chuyện, chia sẻ kinh nghiệm và tư vấn cho nhau. Bằng cách tạo ra một website diễn đàn, bạn có thể giúp mọi người cải thiện kiến thức và kết nối với nhau một cách dễ dàng. Vì vậy, chọn website diễn đàn làm đề tài cho đồ án là một lựa chọn rất hợp lý.

Ngoài ra, website diễn đàn còn là một nền tảng tuyệt vời để phát triển các kỹ năng cá nhân như viết văn bản, giao tiếp và quản lý thời gian. Khi tham gia vào một diễn đàn, bạn sẽ được giao tiếp với nhiều thành viên khác nhau, từ đó rèn luyện kỹ năng giao tiếp và phát triển khả năng trao đổi ý kiến. Bạn cũng sẽ được đưa vào tình huống đặc biệt, từ đó học cách quản lý thời gian và các tình huống khác nhau. Viết bài trên diễn đàn cũng là một cách để bạn cải thiện kỹ năng viết văn bản của mình.

Ngoài ra, khi tham gia vào một diễn đàn công nghệ, bạn sẽ được cập nhật về các công nghệ mới nhất và các kỹ thuật phát triển website. Điều này giúp bạn tiếp cận với những thông tin mới nhất về công nghệ và giúp bạn phát triển kiến thức về lĩnh vực này. Tất cả những kinh nghiệm này sẽ rất hữu ích cho sự nghiệp của bạn trong tương lai.

## **2. MỤC TIÊU ĐỀ TÀI**

Mục tiêu của đề tài này là tạo ra một nền tảng diễn đàn công nghệ để kết nối cộng đồng, chia sẻ kiến thức và phát triển kỹ năng cá nhân. Cụ thể, đề tài sẽ tập trung vào các mục tiêu sau:

- Kết nối cộng đồng: Tạo ra một nền tảng cho phép các thành viên trong cộng đồng giao lưu và hợp tác với nhau. Điều này giúp xây dựng một môi trường tương tác tích cực giữa các thành viên, từ đó giúp tăng cường sự gắn kết và sự hỗ trợ cho nhau.
- Chia sẻ kiến thức: Cung cấp một nơi an toàn và trung thực để các thành viên chia sẻ và nhận được kiến thức. Điều này giúp xây dựng một cộng đồng học

tập, nơi mọi người có thể học hỏi và chia sẻ kiến thức của mình với nhau. Nhờ đó, các thành viên sẽ có được sự giúp đỡ và hỗ trợ trong việc giải quyết các vấn đề công nghệ một cách nhanh chóng và hiệu quả.

- Tạo ra một cộng đồng tự trị: Tạo ra một cộng đồng tự trị với các thành viên có thể giúp đỡ và hỗ trợ nhau. Điều này giúp xây dựng một môi trường tương tác tích cực giữa các thành viên, từ đó giúp tăng cường sự gắn kết và sự hỗ trợ cho nhau.
- Phát triển kỹ năng: Hỗ trợ các thành viên phát triển kỹ năng của họ, bao gồm viết văn bản, giao tiếp và quản lý thời gian. Để đạt được mục tiêu này, đề tài sẽ tập trung vào việc cung cấp các tài liệu hướng dẫn và chia sẻ kinh nghiệm từ các thành viên có kinh nghiệm trong lĩnh vực công nghệ. Các thành viên sẽ được khuyến khích tham gia vào các hoạt động trao đổi kiến thức trên diễn đàn, từ đó rèn luyện và phát triển các kỹ năng cá nhân của mình.
- Cập nhật công nghệ: Cung cấp cho các thành viên thông tin về các công nghệ mới nhất và kỹ thuật phát triển website. Điều này giúp các thành viên cập nhật được các thông tin mới nhất về công nghệ và có thể áp dụng vào công việc của mình. Đồng thời, các thành viên cũng sẽ được hỗ trợ trong việc giải quyết các vấn đề liên quan đến công nghệ một cách nhanh chóng và hiệu quả hơn.

### 3. TỔNG QUAN/ CƠ SỞ LÝ LUẬN

Website diễn đàn là một trang web cung cấp một nền tảng cho người dùng để trao đổi và chia sẻ thông tin về một chủ đề cụ thể hoặc nhiều chủ đề khác nhau. Nó cung cấp một cộng đồng để người dùng có thể tìm kiếm và đặt câu hỏi, trả lời câu hỏi và thảo luận về những chủ đề họ quan tâm. Website diễn đàn là một công cụ quan trọng để kết nối cộng đồng và giúp mọi người cải thiện kiến thức và kết nối với nhau một cách dễ dàng.

Cơ sở lý luận của đề tài này là xây dựng một nền tảng diễn đàn công nghệ, giúp kết nối cộng đồng, chia sẻ kiến thức và phát triển kỹ năng cá nhân. Nền tảng này sẽ được xây dựng trên cơ sở những nguyên tắc của diễn đàn, bao gồm sự trung thực, tính chất tương tác và sự hỗ trợ của cộng đồng. Đồng thời, đề tài cũng sẽ tập trung vào các yếu

tố thiết kế giao diện, tính năng, tính thân thiện với người dùng, tính bảo mật và tính ổn định của website.

## 4. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

Đối tượng nghiên cứu của đề tài này là website diễn đàn công nghệ. Phạm vi nghiên cứu bao gồm các yêu cầu, thiết kế, tính năng, tính thân thiện với người dùng, tính bảo mật và tính ổn định của website diễn đàn. Để đạt được mục tiêu của đề tài, phương pháp nghiên cứu sẽ bao gồm các bước sau:

- Tiến hành nghiên cứu thị trường để tìm hiểu về các website diễn đàn công nghệ đã có trên thị trường. Điều này giúp xác định các yếu tố cần thiết để xây dựng một website diễn đàn công nghệ hiệu quả.
- Tìm hiểu về các công nghệ và kỹ thuật phát triển website để xác định các công nghệ phù hợp để xây dựng website diễn đàn.
- Thiết kế giao diện và tính năng cho website diễn đàn công nghệ. Điều này bao gồm việc xác định các tính năng cần có để tạo ra một trải nghiệm người dùng tốt nhất và thân thiện nhất. Các tính năng này có thể bao gồm chức năng tìm kiếm, chức năng đăng ký và đăng nhập, chức năng đăng bài và trả lời bài viết, chức năng đăng ảnh, chức năng đánh giá và phản hồi bài viết, chức năng chat và hỗ trợ trực tuyến.
- Tiến hành kiểm thử và đánh giá tính năng của website. Điều này giúp xác định các vấn đề và lỗi trong quá trình phát triển, và từ đó cải thiện tính ổn định và tính bảo mật của website.
- Đưa website diễn đàn công nghệ vào hoạt động và quản lý hoạt động của website. Điều này bao gồm việc quản lý thành viên, quản lý bài viết, quản lý tính năng và nội dung của website.
- Đánh giá hiệu quả của website diễn đàn công nghệ. Điều này giúp đánh giá xem website đã đáp ứng được các mục tiêu được đề ra hay chưa, từ đó cải thiện và phát triển website trong tương lai.

## 5. CÁC CÔNG NGHỆ VÀ KỸ THUẬT LẬP TRÌNH

### 5.1. Mô hình phát triển

#### 5.1.1. Alige

Agile là một phương pháp phát triển phần mềm linh hoạt, là một hướng tiếp cận cụ thể cho việc quản lý dự án phần mềm. Nó gồm một quá trình làm việc tương tác và tích hợp để có thể đưa sản phẩm đến tay người dùng càng nhanh càng tốt.

Những phương pháp phát triển phần mềm theo cách truyền thống ngày càng bộc lộ nhiều nhược điểm và tỷ lệ các dự án thất bại cao trong thời kỳ bùng phát của ngành công nghệ. Nhận ra vấn đề đó, một số cá nhân và công ty riêng lẻ đã đưa ra các phương pháp phát triển phần mềm hiện đại hơn và khác nhau để thích ứng với tình hình mới.

Những phương thức phát triển phần mềm này giúp phần nào giải quyết được một số vấn đề nhưng lại phát sinh vấn đề khác về sự cộng tác, kỹ thuật, công cụ, hướng phát triển, chia sẻ ....

Vào năm 2001, bản tuyên ngôn Agile đã được thống nhất và ra đời bởi một nhóm người có uy tín trong phát triển phần mềm:

- **Individuals and interactions over processes and tools:** Cá nhân và sự tương tác hơn là quy trình và công cụ
- **Working software over comprehensive documentation:** Phần mềm chạy tốt hơn là tài liệu đầy đủ
- **Customer collaboration over contract negotiation:** Cộng tác với khách hàng hơn là đàm phán hợp đồng
- **Responding to change over following a plan:** Phản hồi với sự thay đổi hơn là bám theo kế hoạch

### 5.1.2. Scrum

Scrum là một “bộ khung làm việc” cơ bản để tiếp cận những công việc phức tạp. Dựa trên bộ khung này, nhóm làm việc có thể áp dụng những quy trình, kỹ thuật khác nhau cho công việc của mình... Nó là một thành viên của họ Agile.

Nó giúp loại bỏ những công đoạn phức tạp và chỉ tập trung vào những công đoạn cần thiết đáp ứng được nhu cầu của khách hàng đưa ra. Ba yếu tố nòng cốt tạo thành một mô hình quản lý tiến trình thực nghiệm gồm: sự minh bạch (transparency), thanh tra (inspection) và thích nghi (adaptation).

## Ba giá trị cốt lõi của Scrum

- *Minh bạch*

Muốn áp dụng thành công Scrum, các thông tin liên quan đến quá trình phải minh bạch và thông suốt. Các thông tin có thể là tầm nhìn của sản phẩm, yêu cầu của khách hàng, tiến độ công việc, các rào cản khác...

Từ đó mọi thành viên ở vai trò khác nhau có đầy đủ thông tin cần có để tiến hành quyết định trong việc nâng cao hiệu quả công việc.

- *Thanh tra*

Phải thường xuyên thanh tra các hoạt động trong Scrum và tiến độ đến đích để phát hiện các bất thường không theo ý muốn. Tần suất thanh tra không nên quá dày để khỏi ảnh hưởng đến công việc. Công tác thanh tra khi được thực hiện bởi người có kỹ năng tại các điểm quan trọng của công việc sẽ giúp cải tiến liên tục trong Scrum.

- *Thích nghi*

Scrum mang lợi thế là tính linh hoạt rất cao, nhờ đó mang lại tính thích nghi cao. Dựa vào thông tin liên tục và minh bạch từ quá trình thanh tra và làm việc, Scrum có thể cho lại các thay đổi tích cực, nhờ đó mang lại thành công cho dự án.

Lợi ích mà Scrum mang lại

Tính minh bạch, kiểm tra, và thích nghi là 3 nền tảng cơ bản của Scrum. Và dưới đây là những lý do tại sao nên dùng Scrum:

1. Cải thiện chất lượng phần mềm, dễ học và dễ sử dụng.
2. Rút ngắn thời gian phát hành phần mềm, cho phép khách hàng sử dụng sản phẩm sớm hơn.
3. Nâng cao tinh thần đồng đội, tối ưu hóa hiệu quả và nỗ lực của đội phát triển.
4. Gia tăng tỷ suất hoàn vốn đầu tư (ROI)
5. Tăng mức độ hài lòng của khách hàng
6. Kiểm soát dự án tốt, cải tiến liên tục
7. Giảm thiểu rủi ro khi xây dựng sản phẩm

### 5.2. Design pattern

- **Mô hình MVC (Model-View-Controller)**

Mô hình MVC là một trong những mô hình phát triển phổ biến nhất hiện nay, đặc biệt trong lĩnh vực phát triển ứng dụng web. Nó giúp tách biệt logic xử lý dữ liệu, logic nghiệp vụ và giao diện người dùng, giúp dễ dàng bảo trì và mở rộng ứng dụng. Do đó, việc sử dụng mô hình MVC trong dự án của bạn là một lựa chọn tốt và được khuyến khích.

- **Mô hình đang sử dụng cho phần Backend**

Mô hình đang sử dụng cho phần Backend là mô hình được cải tiến từ mô hình MVC nhằm tạo ra sự linh hoạt cho quá trình phát triển của dự án với 5 thành phần chính:

- **Controller:** chứa các lớp điều khiển, xử lý tương tác giữa người dùng và ứng dụng.
- **Data:** chứa các lớp xử lý dữ liệu, kết nối và truy xuất dữ liệu từ các nguồn khác nhau.
- **Model:** chứa các lớp mô hình, định nghĩa cấu trúc dữ liệu và cách thao tác với dữ liệu.
- **Service:** chứa các lớp cung cấp các dịch vụ hỗ trợ cho các lớp khác của ứng dụng.
- **View:** chứa các giao diện người dùng, hiển thị dữ liệu và tương tác với người dùng.

### 5.3. Công nghệ sử dụng

#### 5.3.1. Backend API



.NET 7.0



Hình 1.5.3.1a: .NET 7.0 (tham khảo:

<https://devblogs.microsoft.com/dotnet/announcing-dotnet-7/>)

- *.NET Core là gì?*

.NET Core là một nền tảng phát triển đa mục đích, mã nguồn mở được duy trì bởi Microsoft và cộng đồng .NET trên GitHub. Đó là nền tảng chéo (hỗ trợ Windows, macOS và Linux) và có thể được sử dụng để xây dựng các ứng dụng thiết bị, đám mây.

- *Đặc điểm của .NET Core*

- *Đa nền tảng:* Chạy trên các hệ điều hành Windows, macOS và Linux.
- *Nhất quán trên các kiến trúc:* có thể chạy mã nguồn của bạn với cùng một hành vi trên nhiều kiến trúc hệ thống, bao gồm x64, x86 và ARM.
- *Các công cụ dòng lệnh:* Bao gồm các công cụ dòng lệnh dễ sử dụng, có thể được sử dụng để phát triển cục bộ và trong các tình huống tích hợp liên tục.
- *Triển khai linh hoạt:* có thể cài đặt song song (cài đặt toàn người dùng hoặc toàn hệ thống). Có thể được sử dụng với các container Docker
- *Tương thích:* .NET Core tương thích với .NET Framework, Xamarin và Mono, thông qua .NET Standard.

- *Nguồn mở*: Nền tảng .NET Core là nguồn mở, sử dụng giấy phép MIT và Apache 2. .NET Core là một dự án .NET Foundation.
- *Được hỗ trợ bởi Microsoft*: .NET Core được Microsoft hỗ trợ, theo Hỗ trợ .NET Core.
- *Thành phần của .NET Core*:
  - *.NET Core runtime*: cung cấp một hệ thống kiểu, tải lăp ráp, trình thu gom rác, interop gốc và các dịch vụ cơ bản khác. Các thư viện khung .NET Core cung cấp các kiểu dữ liệu nguyên thủy, các kiểu thành phần ứng dụng và các tiện ích cơ bản.
  - *ASP.NET Core runtime*: cung cấp khung để xây dựng các ứng dụng kết nối internet, điện toán đám mây hiện đại, chẳng hạn như ứng dụng web, ứng dụng IoT và phụ trợ di động.
  - *.NET Core SDK và trình biên dịch ngôn ngữ (Roslyn và F#)*: cho phép trải nghiệm nhà phát triển .NET Core.
  - *Dotnet command*: được sử dụng để khởi chạy các ứng dụng .NET Core và các lệnh CLI. Nó chọn thời gian chạy và lưu trữ thời gian chạy, cung cấp chính sách tái lắp ráp và khởi chạy các ứng dụng và công cụ.
- *Lịch sử các phiên bản và kế hoạch phát triển .NET Core*:

*Bảng 1.5.3.1a: Lịch sử các phiên bản và kế hoạch phát triển .NET Core:*

Version	Ngày bắt đầu	Ngày kết thúc
.NET 7	8 thg 11, 2022	
.NET 6.0 (LTS)	9 thg 11, 2021	12 thg 11, 2024
.NET 5.0	10 thg 11, 2020	10 thg 5, 2022

.NET Core 3.1	3 thg 12, 2019	13 thg 12, 2022
.NET Core 3.0	23 thg 9, 2019	3 thg 3, 2020
.NET Core 2.2	4 thg 12, 2018	23 thg 12, 2019
.NET Core 2.1	30 thg 5, 2018	21 thg 8, 2021
.NET Core 2.0	14 thg 8, 2017	1 thg 10, 2018
.NET Core 1.1 (LTS)	16 thg 11, 2016	27 thg 6, 2019
.NET Core 1.0	27 thg 6 , 2016	27 thg 6, 2019

## □ Identity Server 4



Hình 1.5.3.1b: Identity Server 4 (tham khảo:

<https://nhatkynhoctap.blogspot.com/2017/08/tim-hieu-ve-identityserver-4.html>)

- *ASP.NET Core Identity là gì?*

ASP.NET Core Identity là một thư viện quản lý việc chứng thực và cấp quyền được dùng cho mọi loại project từ MVC, WebForms cho đến WebAPI. Nó đáp ứng các nhu cầu chính như sau:

- Dễ dàng tùy chỉnh profile của user
- Lưu trữ thông tin user trong cơ sở dữ liệu sử dụng Entity Framework Code First.
- Hỗ trợ unit test
- Giới hạn quyền truy cập theo quyền
- Cung cấp cơ chế làm việc với claim

- *Tại sao lại cần IdentityServer4?*

ASP.NET Identity có thể nhận một token bảo mật khi đăng nhập từ bên thứ 3 như Facebook, Google, Microsoft và Twitter. Nhưng nếu bạn muốn tự sinh ra mã token bảo mật cho các ứng dụng nội bộ của bạn giống như họ thì bạn cần phải có một thư viện của bên thứ 3 giống như IdentityServer4, OpenIddict.

Nếu làm một nghiên cứu nho nhỏ thì các bạn có thể thấy là IdentityServer4 rất phổ biến và hay được dùng để làm việc này.

- *IdentityServer4 là gì?*

IdentityServer4 là một framework hỗ trợ OpenID Connect và OAuth 2.0 trên ASP.NET Core. Nó cũng là một gói thư viện trên Nuget được dùng trong ASP.NET Core như một middleware cho phép đăng nhập/đăng xuất, cấp token, chứng thực và các giao thức chuẩn khác.

Để có cái nhìn tổng quan về kiến trúc của nó, các bạn có thể tham khảo thuật ngữ ở trang chủ của IdentityServer4:

- *User*: là con người, là bạn hoặc tôi đang sử dụng một client.
- *Client*: là một phần mềm ứng dụng kiểu như trình duyệt web, mobile app hay bất cứ cái gì đang cần gọi một API resource.
- *Resources*: là các API bạn cần bảo vệ bởi IdentityServer4
- *Access Token*: nó là token được sử dụng để truy cập vào API Resource
- *Refresh Token*: mỗi token đều có một thời gian hết hạn. Refresh token là việc lấy lại token mới mà không cần tương tác của người dùng. Client nên cho phép làm điều này bằng cách setup AllowOfflineAccess giá trị là true ở phần cài đặt client trong IdentityServer4.
- *Grant Type*: nó là loại tương tác giữa client và IdentityServer. Dựa trên client của bạn, có thể chọn loại grant type phù hợp.

## SQL Server 2019



*Hình 1.5.3.1c: SQL Server 2019 (tham khảo:*

*<https://support.cloud365.vn/cloud-app/sql-trong-cloud-server/>*

- *SQL Server là gì?*

SQL Server là gì? SQL Server hay Microsoft SQL Server là phần mềm ứng dụng cho hệ thống quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) được phát triển bởi Microsoft vào năm 1988. Nó được sử dụng để tạo, duy trì, quản lý và triển khai hệ thống RDBMS.

Phần mềm SQL Server được sử dụng khá rộng rãi vì nó được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn lên đến Tera – Byte cùng lúc phục vụ cho hàng ngàn user. Bên cạnh đó, ứng dụng này cung cấp đa dạng kiểu lập trình SQL từ ANSI SQL (SQL truyền thống) đến SQL và cả T-SQL (Transaction-SQL) được sử dụng cho cơ sở dữ liệu quan hệ nâng cao.

- *Các phiên bản SQL Server*

- *SQL Server 2012* cung cấp thêm những tính năng mới như chỉ mục cột lưu trữ, có thể được sử dụng để lưu trữ dữ liệu theo định dạng cột cho các ứng dụng. Mặt khác, phiên bản này có tính khả dụng cao và được trang bị công nghệ khắc phục sau thảm họa.
- *SQL Server 2014* đã thêm OLTP trong bộ nhớ, từ đó cho phép người dùng chạy các ứng dụng xử lý giao dịch trực tuyến. Một tính năng mới khác trong SQL 2014 đó là phần mở rộng vùng đệm bằng cách tích hợp bộ nhớ đệm và

ở đĩa – đây là cách thiết kế để tăng thông lượng I/O thông qua việc giảm dữ liệu từ các đĩa cứng thông thường.

- **SQL Server 2017** hỗ trợ chạy trên Linux, điều này làm SQL Server chuyển từ nền tảng cơ sở dữ liệu sang một hệ điều hành mã nguồn mở thường thấy trong các doanh nghiệp.Thêm vào đó, ở phiên bản này còn hỗ trợ ngôn ngữ lập trình Python, một ngôn ngữ mã nguồn mở được sử dụng rộng rãi trong các ứng dụng phân tích.
- **SQL Server 2019** cho phép người dùng kết hợp các vùng chứa SQL Server, HDFS và Spark với nhau bằng cách sử dụng tính năng Big Data Cluster mới. Thêm vào đó, một tính năng mới khác là khả năng phục hồi dữ liệu được tăng tốc nhanh hơn.

### 5.3.2. Front End

#### 5.3.2.1.1. Admin

##### □ Angular 15



*Hình 1.5.3.2.1.1 Angular 15 (tham khảo:*

*<https://www.patrickschadler.com/the-best-features-of-angular-15/>*

- *Angular là gì?*

Angular là một framework JavaScript và được viết bằng TypeScript. Google tạo ra framework này với mục đích viết nên giao diện web (Front-end) đúng chuẩn “ít nỗ lực”. Không chỉ đem lại những lợi ích như một framework, mà Angular còn giữ nguyên cấu trúc như ngôn ngữ lập trình tiêu chuẩn. Điều đó giúp các nhà phát triển vừa dễ mở rộng dự án cũng như việc bảo trì.

- *Ưu điểm của Angular*

- *Custom components:* Angular cho phép người dùng xây dựng các component (thành phần) của riêng họ, có thể đóng gói chức năng cùng với logic cho các component để tạo thành các phần có thể tái sử dụng. Các component này cũng hoạt động tốt với các component khác của web.
- *Data binding:* Data Binding là kỹ thuật dùng để tạo gắn kết giữa phần giao diện (UI) và dữ liệu thông qua phần business logic (tạm dịch – logic xử lý việc trao đổi thông tin dữ liệu với giao diện người dùng). Nhờ Data Binding, UI có thể tự động cập nhật lại để hiển thị các thay đổi trong dữ liệu.

- *Dependency injection*: Angular cho phép người dùng viết các câu lệnh và đưa chúng vào bất cứ nơi nào họ cần. Điều này cải thiện khả năng kiểm tra và khả năng tái sử dụng của các câu lệnh giống nhau.
  - *Testing*: Kiểm tra là công cụ hạng nhất của Angular vì ngay từ bản Angular 1 thì tính năng này đã được áp dụng. Bạn sẽ có thể kiểm tra mọi phần trong ứng dụng của mình điều này là điểm cộng nếu so với JavaScripts.
  - *Comprehensive (Tính toàn diện)*: Angular là framework được Google chống lưng nên đây là framework có “bảo hiểm” và cung cấp các giải pháp tiện ích toàn diện cho giao tiếp máy chủ.
  - *Browser Compatibility*: Angular là đa nền tảng và tương thích với nhiều trình duyệt. Một ứng dụng Angular thường có thể chạy trên tất cả các trình duyệt (Ví dụ: Chrome, Firefox) và hệ điều hành, chẳng hạn như Windows, macOS và Linux.
- *Nhược điểm của Angular*
    - *Steep learning curve (Đường cong học tập)*: Các thành phần cơ bản của Angular mà tất cả người dùng nên biết bao gồm chỉ thị (directives), mô-đun, trình trang trí (decorators), thành phần (components), dịch vụ (services), dependency injection, pipes, và mẫu (templates)... Các chủ đề nâng cao hơn bao gồm biên dịch AoT và Rx.js... Đối với người mới bắt đầu, Angular 4 có thể khó học vì nó là framework hoàn chỉnh.
    - *Limited SEO options*: Angular cung cấp các tùy chọn SEO hạn chế và khả năng tiếp cận kém với trình thu thập thông tin của công cụ tìm kiếm.
    - *Verbose/ Complex (Rườm rà)*: Một vấn đề phổ biến trong cộng đồng Angular là sự rườm rà của framework này. Nó cũng khá phức tạp so với các công cụ front-end khác.

### 5.3.2.2. Portal

#### ❑ Razor View



Hình 1.5.3.2.1.2 Razor View (tham khảo:

<https://www.telerik.com/blogs/exploring-aspnet-core-csharp-7-razor-view-blazor-view-existing-razor-project>)

- *View Engine là gì?*

View Engine có trách nhiệm xuất ra HTML cuối cùng khi Controller gọi Action Method. Controller action method trả về các kiểu khác nhau của response. Nó gọi là Action Result. ViewResult là một ActionResult nó trả ra HTML Response. ViewResult được cung cấp bởi View Engine. Nó sẽ sản sinh ra HTML.

- *Razor View Engine là gì?*

Razor View Engine là View Engine mặc định của ASP.NET Core. Nó lấy mã Razor trong file View và chuyển sang HTML response.

- *Razor Markup*

Controller trong MVC gọi View bằng cách gán dữ liệu để tạo giao diện. View phải có khả năng xử lý dữ liệu và tạo response. Điều này được xử lý bằng cách dùng Razor, nó cho chúng ta sử dụng C# code trong file HTML. Razor View Engine xử lý các lệnh này và tạo ra HTML.

Các file chứa Razor có đuôi .cshtml. Cú pháp Razor thường ngắn hơn và đơn giản hơn cũng dễ học như C# hoặc VB. Visual Studio IntelliSense cũng hỗ trợ cú pháp Razor.

## 5.4. Git

### □ Github



Hình 1.5.4 Github(tham khảo:

<https://www.freecodecamp.org/news/git-and-github-the-basics/>

- *Github là gì?*

Github là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. Github có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

- *Vài thông tin về GIT:*

- Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.
- Giúp đồng bộ source code của team lên 1 server.
- Hỗ trợ các thao tác kiểm tra source code trong quá trình làm việc (diff, check modifications, show history, merge source, ...)

GitHub có 2 phiên bản: miễn phí và trả phí. Với phiên bản có phí thường được các doanh nghiệp sử dụng để tăng khả năng quản lý team cũng như phân quyền bảo mật dự án.

Còn lại thì phần lớn chúng ta đều sử dụng Github với tài khoản miễn phí để lưu trữ source code.

Github cung cấp các tính năng social networking như feeds, followers, và network graph để các developer học hỏi kinh nghiệm của nhau thông qua lịch sử commit.

Nếu một comment để mô tả và giải thích một đoạn code. Thì với Github, commit message chính là phần mô tả hành động mà bạn thực hiện trên source code.

Github trở thành một yếu tố có sức ảnh hưởng lớn trong cộng đồng nguồn mở. Cùng với Linkedin, Github được coi là một sự thay thế cho CV của bạn. Các nhà tuyển dụng cũng rất hay tham khảo Github profile để hiểu về năng lực coding của ứng viên.

- *Tính năng của Github*
  - GitHub được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:
    - Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.
    - Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phần mềm thế nào, project phát triển ra sao.
    - Follow user: theo dõi hoạt động của người khác.

Có 2 cách tiếp cận GitHub: Tạo project của riêng mình Contribute cho project có sẵn: fork project có sẵn của người khác, sửa đổi, sau đó đề nghị họ cập nhật sửa đổi của mình (tạo pull request).

- *Một vài khái niệm của Git bạn cần nắm:*
  - **git**: là prefix của các lệnh được sử dụng dưới CLI
  - **branch**: được hiểu như là nhánh, thể hiện sự phân chia các version khi 2 version đó có sự sai khác nhất định và 2 version đều có sự khác nhau.
  - **commit**: là một điểm trên cây công việc (Work Tree ) hay gọi là cây phát triển công việc
  - **clone**: được gọi là nhân bản hay thực hiện nhân bản. Sử dụng để clone các project, repository trên các hệ thống chạy trên cơ sở là git, ví dụ như:

bitbucket, github, gitlab, cor(1 sản phẩm mã nguồn mở cho phép người dùng tự tạo git server cho riêng mình trên vps, server),... Việc clone này sẽ sao chép repository tại commit mình mong muốn, dùng để tiếp tục phát triển. Thao tác này sẽ tải toàn bộ mã nguồn, dữ liệu về máy tính của bạn.

- **folk:** Folk là thao tác thực hiện sao chép repository của chủ sở hữu khác về git account của mình. sử dụng và đối xử như 1 repository do mình tạo ra.
- **repository:** Kho quản lý dữ liệu, là nơi lưu trữ các dữ liệu, mã nguồn của project.
- **tag:** sử dụng để đánh dấu một commit khi bạn có quá nhiều commit tới mức không thể kiểm soát được.
- **remote:** sử dụng để điều khiển các nhánh từ một repository trên git server, đối xử với các nhánh trên remote tương tự như đối xử với các nhánh trên local
- **diff:** So sánh sự sai khác giữa phiên bản hiện tại với phiên bản muốn so sánh, nó sẽ thể hiện các sự khác nhau
- **.gitignore:** file mặc định của git sử dụng để loại bỏ (ignore) các thư mục, file mà mình không muốn push lên git server

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 1. ĐẶC TẢ YÊU CẦU

Website Daisy Forum là một website diễn đàn nhằm mục đích chia sẻ, giao lưu kiến thức, đặc biệt là mảng lập trình, những bài đăng của trang web thường là các bài viết chia sẻ về kiến thức, kinh nghiệm trong quá trình thực hiện dự án.

Trước tiên nói về phần người dùng, người dùng sẽ tạo tài khoản bằng cách cung cấp đầy đủ các thông tin về: họ tên, số điện thoại, ngày sinh, địa chỉ email. Ngoài ra người dùng có thể sử dụng Google, Facebook, Microsoft để đăng ký nhanh chóng nhờ việc sử dụng Open Id. Sau khi đăng ký xong, hệ thống sẽ gửi đường dẫn xác thực tài khoản về địa chỉ email mà người dùng đã nhập, nếu người dùng bỏ lỡ phần xác thực tài khoản thì người dùng có thể nhấn vào phần gửi lại đường dẫn xác nhận tài khoản. Khi đăng ký thành công người dùng có thể sử dụng để đăng nhập vào trang web, xem thông tin tài khoản, cập nhật thông tin tài khoản, thay đổi mật khẩu.

Về phần chính của trang web, đầu tiên khi đến với giao diện chính, các danh mục bài viết sẽ được hiện lên với cái thông tin tóm tắt như: tiêu đề, mô tả ngắn gọn, thể loại, nhãn dán, ngày đăng. Những danh mục bao gồm: bài viết mới đăng, bài viết xem nhiều và bài viết đề xuất. Ngoài ra trên thanh tiêu đề trang web sẽ chứa những thao tác nhanh như, tìm kiếm, xem theo thể loại, xem theo nhãn dán. Khi người dùng nhấn vào xem chi tiết bài viết thì các thông tin sẽ được hiện ra bao gồm: tiêu đề, người đăng, ngày đăng, thể loại, danh sách nhãn dán, mô tả, vấn đề, môi trường, các bước tái hiện, thông báo lỗi (nếu có), giải pháp (nếu có), danh sách tệp đính kèm (nếu có). Người dùng có thể thao tác với bài viết như: thích, bình luận và trả lời bình luận, báo cáo bài viết, khi sử dụng những tính năng này thì việc đăng nhập là yêu cầu bắt buộc. Đặc biệt khi bình luận trên trang web sẽ có phần kiểm tra những bình luận tiêu cực tự động, bình luận có thể bị ẩn hoặc xóa đi tùy vào mức độ nghiêm trọng.

Khi người dùng muốn tạo mới bài đăng thì người dùng cần cung cấp các thông tin về: tiêu đề, người đăng, ngày đăng, thể loại, danh sách nhãn dán, mô tả, vấn đề, môi trường, các bước tái hiện, thông báo lỗi(nếu có), giải pháp(nếu có), danh sách tệp đính kèm(nếu có).

Về phần quản lý trang web, quản trị viên sẽ có cách chức năng liên quan đến cách chức năng của trang web và phân quyền trên hệ thống. Quản trị viên có thể gán quyền

cho các hành động trên trang web, quản trị viên có thể xóa các tài khoản trên hệ thống, quản trị viên cũng có thể cấp quyền hoặc xóa quyền của các tài khoản. Ngoài ra quản trị viên cũng có các chức năng quản lý về bài viết, bình luận.

## 2. THUỘC TÍNH THỰC THẾ

- **Categories:** [Id](#), Name, SeoAlias, SeoDescription, SortOrder, ParentId, NumberOfTicket
- **KnowledgeBases:** [Id](#), [CategoryId](#), Title, SeoAlias, Description, Environment, Problem, StepToReproduce, ErrorMessage, Workaround, Note, OwnerUserId, Labels, CreateDate, LastModifiedDate, NumberOfComments, NumberOfVotes, NumberOfReports, ViewCount
- **Attachments:** [Id](#), [KnowledgeBaseId](#), FileName, AttachFilePath, FileType, FileSize, CreateDate, LastModifiedDate
- **Reports:** [Id](#), [KnowledgeBaseId](#), Content, ReportUserId, CreateDate, LastModifiedDate, IsProcessed
- **Comments:** [Id](#), [KnowledgeBaseId](#), Content, OwnerUserId, CreateDate, LastModifiedDate
- **Users:** [Id](#), FirstName, LastName, Dob, NumberOfKnowledgeBases, NumberOfVotes, NumberOfReports, UserName, Email, PhoneNumber, PasswordHash, CreateDate, LastModifiedDate
- **Roles:** [Id](#), Name
- **UserRoles:** [UserId](#), [RoleId](#)
- **Functions:** [Id](#), Name, Url, SortOrder, ParentId, Icon
- **Commands:** [Id](#), Name
- **CommandInFunctions:** [CommandId](#), [FunctionId](#)
- **Permissions:** [RoleId](#), [FunctionId](#), [CommandId](#)
- **Votes:** [KnowledgeBaseId](#), [UserId](#)
- **Labels:** [Id](#), Name,
- **LabelInKnowledgeBases:** [KnowledgeBaseId](#), [LabelId](#),
- **CacheTable:** [Id](#), Value, ExpiresAtTime, SlidingExpirationInSeconds, AbsoluteExpiration

### 3. MÔ TẢ THUỘC TÍNH

#### 3.1. Bảng thuộc tính Users

Bảng 2.3.1 Bảng thuộc tính Users

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	string	50	not null	Khóa chính
FirstName	string	50	not null	
LastName	string	50	not null	
Dob	datetime		not null	Kiểu ngày
NumberOfKnowledgeBases	int			
NumberOfVotes	int			
NumberOfReports	int			
UserName	string	256	not null	
Email	string	256	not null	
PhoneNumber	string	MAX	not null	
PasswordHash	string	MAX	not null	Chuỗi mã hóa
CreateDate	datetime		not null	
LastModifiedDate	datetime			

#### 3.2. Bảng thuộc tính Categories

Bảng 2.3.2 Bảng thuộc tính Categories

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	int		not null	Khóa chính
Name	string	200	not null	
SeoAlias	string	200	not null	
SeoDescription	string	500	not null	

SortOrder	int		not null	
ParentId	int			
NumberOfTicket	int			

### 3.3. Bảng thuộc tính Knowledge Bases

Bảng 2.3.3 Bảng thuộc tính Knowledge Bases

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	int		not null	Khóa chính
CategoryId	int		not null	Khóa ngoại
Title	string	MAX	not null	
SeoAlias	string	MAX	not null	
Description	string	MAX	not null	
Environment	string	MAX		
Problem	string	MAX	not null	
StepToReproduce	string	MAX		
ErrorMessage	string	MAX		
Workaround	string	MAX		
Note	string	MAX		
OwnerUserId	string	MAX	not null	
Labels	string	MAX	not null	
CreateDate	datetime		not null	
LastModifiedDate	datetime			
NumberOfComments	int			
NumberOfVotes	int			
NumberOfReports	int			
ViewCount	int			

### 3.4. Bảng thuộc tính Attachments

Bảng 2.3.4 Bảng thuộc tính Attachments

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	int		not null	Khóa chính
KnowledgeBaseId	int		not null	Khóa ngoại
FileName	string	200	not null	
AttachFilePath	string	200	not null	
FileType	string	4	not null	
FileSize	int		not null	
CreateDate	datetime		not null	
LastModifiedDate	datetime			

### 3.5. Bảng thuộc tính Reports

Bảng 2.3.5 Bảng thuộc tính Reports

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	int		not null	Khóa chính
KnowledgeBaseId	int		not null	Khóa ngoại
Content	string	500	not null	
ReportUserId	string	50	not null	Khóa ngoại
CreateDate	datetime		not null	
LastModifiedDate	datetime			
IsProcessed	bool			

### 3.6. Bảng thuộc tính Comments

*Bảng 2.3.5 Bảng thuộc tính Comments*

<b>Thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Chiều dài</b>	<b>Bắt buộc</b>	<b>Chú thích</b>
Id	int		not null	Khóa chính
KnowledgeBaseId	int		not null	Khóa ngoại
Content	string	MAX	not null	
OwnerUserId	string	50	not null	Khóa ngoại
CreateDate	datetime		not null	
LastModifiedDate	datetime			

### 3.7. Bảng thuộc tính Roles

*Bảng 2.3.7 Bảng thuộc tính Roles*

<b>Thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Chiều dài</b>	<b>Bắt buộc</b>	<b>Chú thích</b>
Id	string	50	not null	Khóa chính
Name	string	50	not null	

### 3.8. Bảng thuộc tính User Roles

*Bảng 2.3.8 Bảng thuộc tính User Roles*

<b>Thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Chiều dài</b>	<b>Bắt buộc</b>	<b>Chú thích</b>
UserId	string	50	not null	Khóa chính
RoleId	string	50	not null	Khóa chính

### 3.9. Bảng thuộc tính Functions

*Bảng 2.3.9 Bảng thuộc tính Functions*

<b>Thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Chiều dài</b>	<b>Bắt buộc</b>	<b>Chú thích</b>
Id	string	50	not null	Khóa chính
Name	string	200	not null	
Url	string	200	not null	
SortOrder	int		not null	Kiểu ngày
ParentId	int			
Icon	string	200		

### 3.10. Bảng thuộc tính Commands

Bảng 2.3.10 Bảng thuộc tính Commands

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	string	50	not null	Khóa chính
Name	string	200	not null	

### 3.11. Bảng thuộc tính Command Functions

Bảng 2.3.11 Bảng thuộc tính Command Functions

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
CommandId	string	50	not null	Khóa chính
FunctionId	string	50	not null	Khóa chính

### 3.12. Bảng thuộc tính Permissions

Bảng 2.3.12 Bảng thuộc tính Permissions

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
RoleId	string	50	not null	Khóa chính
FunctionId	string	50	not null	Khóa chính
CommandId	string	50	not null	Khóa chính

### 3.13. Bảng thuộc tính Votes

Bảng 2.3.13 Bảng thuộc tính Votes

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
KnowledgeBaseId	int		not null	Khóa chính
UserId	string	50	not null	Khóa chính

### 3.14. Bảng thuộc tính Labels

Bảng 2.3.14 Bảng thuộc tính Labels

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
Id	int		not null	Khóa chính
Name	string	50	not null	

### 3.15. Bảng thuộc tính Label In Knowledge Bases

Bảng 2.3.15 Bảng thuộc tính Label In Knowledge Bases

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích
KnowledgeBaseId	int		not null	Khóa chính
LabelId	int		not null	Khóa chính

### 3.16. Bảng thuộc tính CacheTable

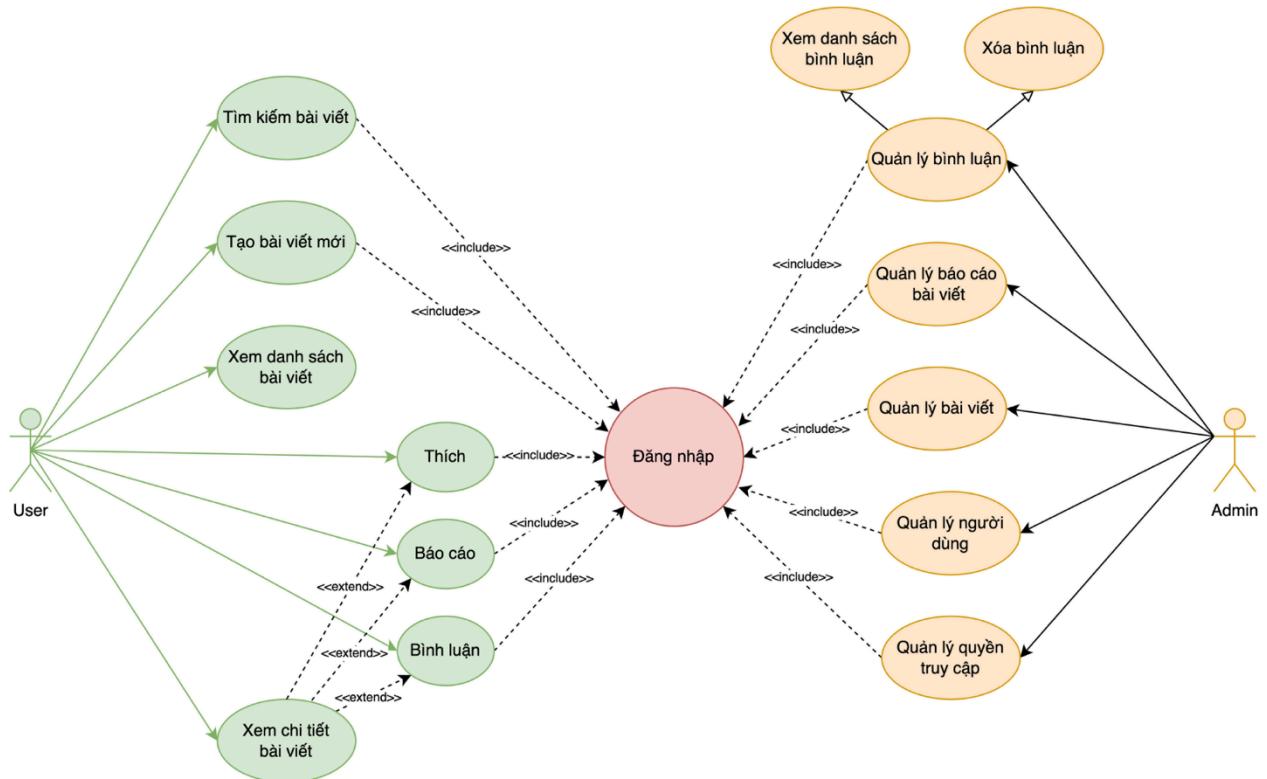
Bảng 2.3.16 Bảng thuộc tính CacheTable

Thuộc tính	Kiểu dữ liệu	Chiều dài	Bắt buộc	Chú thích

<b>Id</b>	string	449	not null	Khóa chính
<b>Value</b>	string	MAX	not null	
<b>ExpiresAtTime</b>	string	datetime	not null	
<b>SlidingExpirationInSeconds</b>	datetime	bigint	not null	Kiểu ngày
<b>AbsoluteExpiration</b>	int			

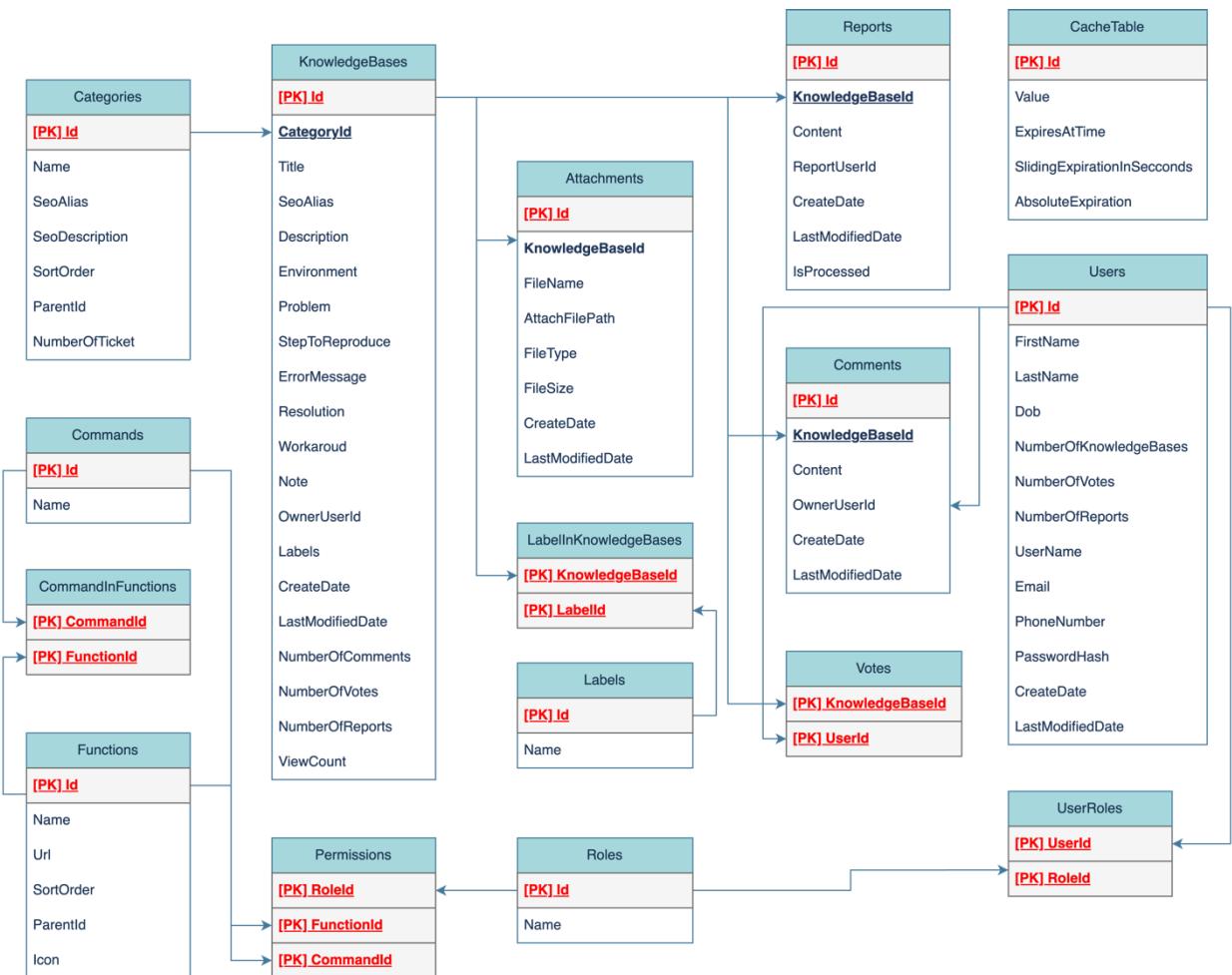
## 4. SƠ ĐỒ

### 4.1. Sơ đồ Use Case



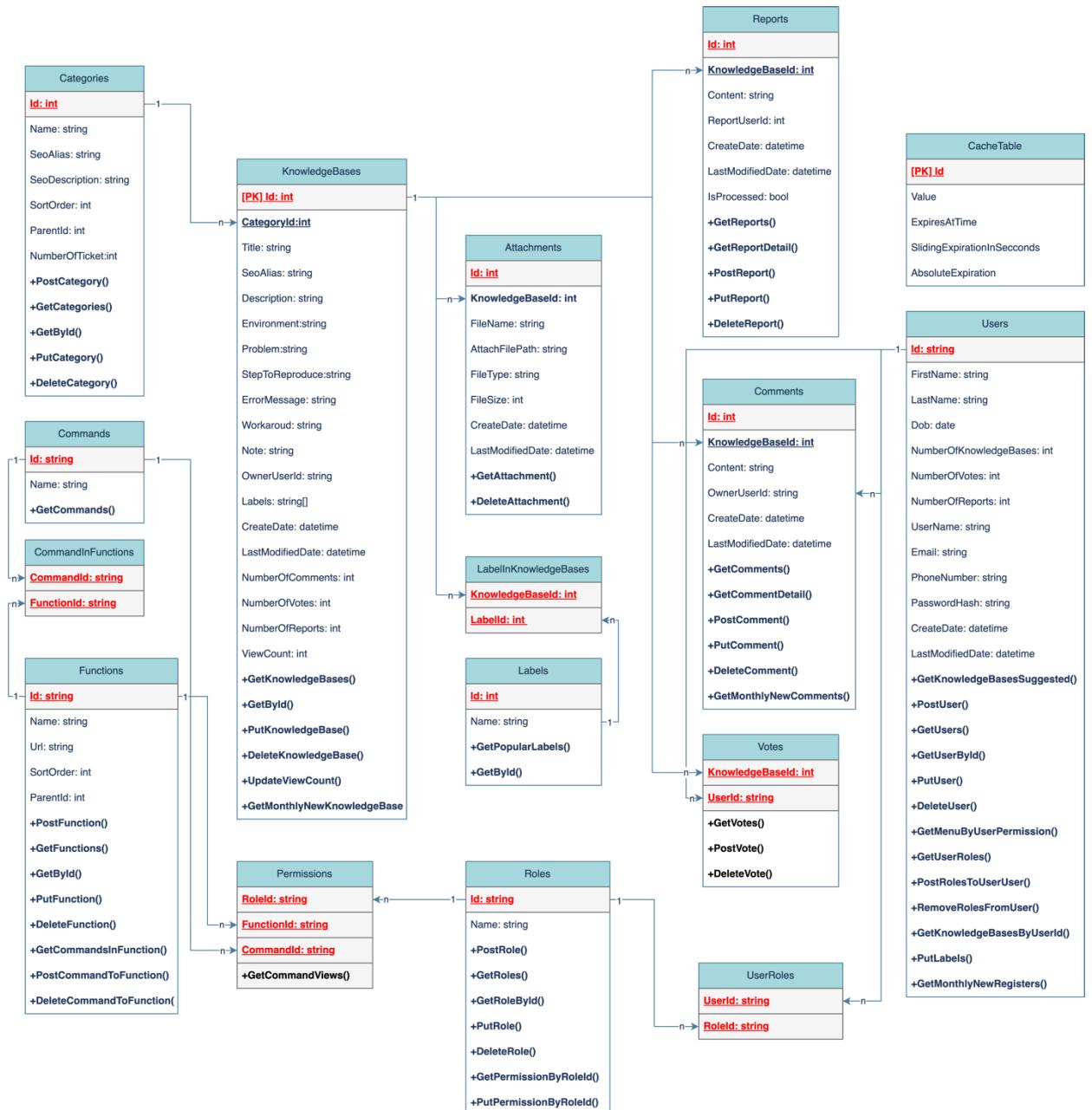
Hình 2.4.1:Sơ đồ Use Case

## 4.2. Sơ đồ ERD



Hình 2.4.2:Sơ đồ ERD

### 4.3. Sơ đồ Class



Hình 2.4.3:Sơ đồ Class

# CHƯƠNG 3: XÂY DỰNG WEBSITE VÀ KIỂM THỦ

## 1. THỰC HIỆN

### 1.1. Backend API

- **Data:** chứa các đối tượng, thuộc tính cho phần cơ sở dữ liệu, ngoài ra phần data còn chứa các cấu hình cho việc migrations tới cơ sở dữ liệu.
- **Controller:** chứa các xử lý chính để cung cấp các API với các method như GET, POST, PUT, DELETE, PATCH. Các API này được cấu hình CORS để cho phép Frontend được gọi đến. Các API được viết với chuẩn RESTful API để cung cấp ra các API chuẩn nhất
- **ViewModel:** chứa các đối tượng sử dụng làm request cho các controller và chứa các đối tượng view model để controller trả về đối tượng phù hợp. Ngoài ra, phần view model còn chứa các Validators các request khi nhận vào controller.
- **Identity:** là phần được tạo ra bởi Identity Server có sẵn trên dự án. Nó cung cấp đầy đủ các tính năng liên quan đến phần Authorization cũng như phần người dùng. Ngoài ra Identity Server cho phép người dùng cấu hình các chức năng đăng nhập bên ngoài thông qua OpenId như Google, Facebook, Microsoft,... Identity Server cũng cho phép người dùng cấu hình xác thực bảo mật 2 lớp thông qua SMS hoặc QR Code.
- **Service:** cung cấp các dịch vụ với phương thức DI cho controller với các dịch vụ như: SendMail, Cache Data, Upload File,...

### 1.2. Frontend

#### 1.2.1. Admin App

Phần Admin App được sử dụng Angular 16 để thực hiện cho trang web dưới dạng ứng dụng web giúp quản trị viên dễ dàng quản lý website. Phần Admin App gồm những phần chính như sau:

- **Layout:** chứa các controller, view, css, js cho phần giao diện tương tác chính với người dùng.
- **Login:** sử dụng thư viện oidc-client để gọi đến Identity Server cho Authentication và quản lý tài khoản.

- **Service:** là phần làm việc chính với API, giúp gọi đến API của Backend Server rồi cung cấp cho phần controller.
- **Model:** chứa các đối tượng sử dụng làm request cho phần service và chứa các đối tượng view model để nhận dữ liệu trả về từ API.

### *1.2.2. Web Portal*

Phần web portal là phần giao diện cho người dùng, phần này được sử dụng Razor View của .NET với mô hình Model-View-Controller-Service như sau:

- Model: Bao gồm các request và view model để gọi API đến Backend Server và sử dụng cho controller, view.
- View: Phần giao diện chính tương tác với người dùng, ngoài ra có sử dụng jQuery để load một số thành phần phức tạp hoặc kiểm tra dữ liệu nhập vào.
- Controller: Phần chức năng chính cho trang web, dữ liệu được nhận vào từ view sẽ được truyền xuống controller rồi sẽ được truyền tới server để gọi api.
- Service: Chứa các phương thức gọi API tới Backend Server.

## 2. THUẬT TOÁN

### 2.1. Thuật toán đề xuất bài viết Contend Based

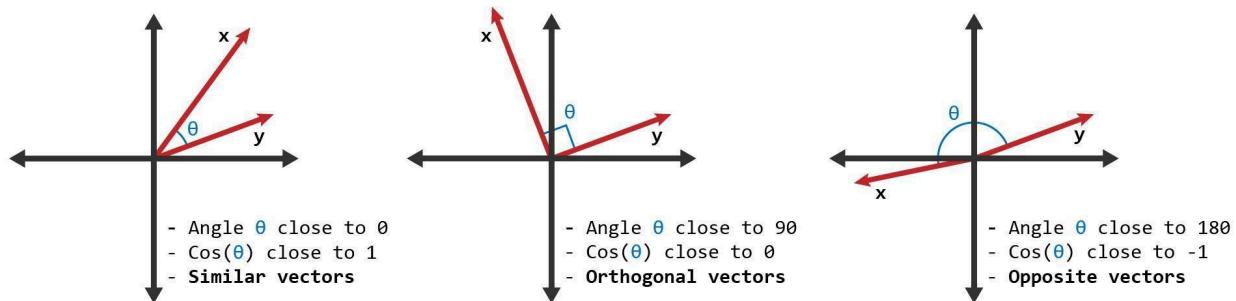
#### 2.1.1. Khái niệm

Thuật toán đề xuất bài viết content-based sử dụng độ đo cosine là một phương pháp được sử dụng trong hệ thống gợi ý nội dung. Thuật toán này dựa trên việc tính toán độ tương đồng giữa các bài viết dựa trên nội dung của chúng. Cụ thể, thuật toán sử dụng độ đo cosine để tính toán độ tương đồng giữa hai vector biểu diễn nội dung của hai bài viết.

Để áp dụng thuật toán này, trước tiên các bài viết cần được biểu diễn dưới dạng vector. Một cách phổ biến để biểu diễn nội dung của một bài viết dưới dạng vector là sử dụng mô hình Bag-of-Words (BoW). Trong mô hình BoW, mỗi từ trong bài viết được đưa vào một vector, và giá trị của vector tại vị trí tương ứng với từ đó được đặt bằng số lần xuất hiện của từ đó trong bài viết. Sau khi biểu diễn các bài viết dưới dạng vector, ta có thể tính toán độ tương đồng giữa chúng bằng cách sử dụng độ đo cosine.

Độ đo cosine là một độ đo độ tương đồng giữa hai vector trong không gian n chiều. Giá trị của độ đo cosine nằm trong khoảng từ -1 đến 1, và giá trị càng gần 1 thì hai vector càng giống nhau. Cụ thể, độ đo cosine được tính bằng cách lấy tích vô hướng của hai vector và chia cho tích của độ dài của hai vector đó.

Trong thuật toán đề xuất bài viết content-based sử dụng cosine, các bài viết có độ tương đồng cao sẽ được đề xuất cho người dùng khi họ đang xem một bài viết cụ thể. Thuật toán này có ưu điểm là không phụ thuộc vào thông tin về người dùng, mà chỉ dựa trên nội dung của các bài viết để đưa ra gợi ý, do đó có thể áp dụng cho những người dùng mới hoặc không có lịch sử sử dụng trước đó. Tuy nhiên, một nhược điểm của thuật toán này là nó không đưa ra những gợi ý mới mẻ hoặc đa dạng, mà chỉ đề xuất các bài viết giống hoặc liên quan đến bài viết đang xem.



*Hình 3.2.1.1a Công thức tính độ đo Cosine Similarity (tham khảo:*

<https://www.learndatasci.com/glossary/cosine-similarity/>)

### 2.1.1. Hiện thực vào dự án

Trong dự án website diễn đàn Daisy Forum, thuật toán để xuất bài viết content-based sử dụng độ đo cosine được ứng dụng vào bằng cách tính toán ra mức độ tương đồng từ những sở thích của người dùng thông qua những nhãn dán của bài viết.

*Bảng 3.2.1.1a Bảng ví dụ về vector của các bài viết*

#tag	c#	java	css	python	windows	arm
post1	0	1	0	1	1	0
post2	1	0	0	0	0	0
post3	0	0	1	0	0	1
post4	1	0	0	0	0	1
post5	1	1	1	0	0	0

*Bảng 3.2.1.1b Bảng ví dụ về vector của các user*

#tag	c#	java	css	python	windows	arm
user1	0	1	0	1	1	0
user2	1	0	0	0	0	0
user3	0	0	1	0	0	1
user4	1	0	0	0	0	1
user5	1	1	1	0	0	0

Từ những vector của bài post và vector của user ta tính được độ cosine qua công thức:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

Giá trị sau khi tính toán ra sẽ được sắp xếp theo thứ tự giảm dần thể hiện mức độ liên quan của bài đăng đến sở thích của người dùng, càng về gần 1 thì mức độ càng liên quan.

*Hình 3.2.1.1b Kết quả tính toán độ đo cosine của các bài viết*

```

Response body
[
  {
    "userId": "1",
    "tags": null,
    "postId": 4082,
    "cosine": 0.7071067811865475
  },
  {
    "userId": "1",
    "tags": null,
    "postId": 4153,
    "cosine": 0.5
  },
  {
    "userId": "1",
    "tags": null,
    "postId": 4182,
    "cosine": 0.5
  },
  {
    "userId": "1",
    "tags": null,
    "postId": 4294,
    "cosine": 0.5
  }
]

```

Download

### 3. KIỂM THỬ

#### 3.1. Kiểm thử giá trị nhập vào bằng UnitTest

Giá trị nhập vào được validate bằng thư viện FluentValidation. FluentValidation có thể được sử dụng trong các ứng dụng web ASP.NET Core để xác thực các mô hình sắp tới. Có hai cách tiếp cận chính để làm điều này:

- Xác thực thủ công
- Xác thực tự động

Với xác thực thủ công, bạn đưa trình xác thực vào bộ điều khiển của mình (hoặc điểm cuối api), gọi trình xác thực và hành động dựa trên kết quả. Đây là cách tiếp cận đơn giản và đáng tin cậy nhất.

Với xác thực tự động, FluentValidation cắm vào đường dẫn xác thực là một phần của ASP.NET Core MVC và cho phép các mô hình được xác thực trước khi một hành động của bộ điều khiển được gọi (trong quá trình liên kết mô hình). Cách tiếp cận xác thực này liền mạch hơn nhưng có một số nhược điểm:

Xác thực tự động không đồng bộ: Nếu trình xác thực của bạn chứa các quy tắc không đồng bộ thì trình xác thực của bạn sẽ không thể chạy. Bạn sẽ nhận được một ngoại lệ trong thời gian chạy nếu bạn cố gắng sử dụng trình xác thực không đồng bộ có xác thực tự động.

Xác thực tự động chỉ dành cho MVC: Tự động xác thực chỉ hoạt động với Bộ điều khiển MVC và Trang dao tạo. Nó không hoạt động với các phần hiện đại hơn của ASP.NET chẳng hạn như API tối thiểu hoặc Blazor.

Xác thực tự động khó gỡ lỗi: Bản chất 'kỳ diệu' của xác thực tự động khiến cho việc gỡ lỗi/khắc phục sự cố nếu xảy ra sự cố trở nên khó khăn vì có quá nhiều thứ được thực hiện ở hậu trường.

## Kết quả kiểm thử

- ✓ ✓ DaisyForum.ViewModels.UnitTesting.Contents
  - ✓ ✓ CategoryCreateRequestValidatorTest
    - ✓ Should\_Error\_Result\_When\_Miss\_Name(name: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_Name(name: null)
    - ✓ Should\_Error\_Result\_When\_Miss\_SeoAlias(seoAlias: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_SeoAlias(seoAlias: null)
    - ✓ Should\_Valid\_Result\_When\_Valid\_Request
  - ✓ ✓ CommentCreateRequestValidatorTest
    - ✓ Should\_Error\_Result\_When\_KnowledgeBaseId\_Is\_Zero
    - ✓ Should\_Error\_Result\_When\_Miss\_Content(content: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_Content(content: null)
    - ✓ Should\_Valid\_Result\_When\_Valid\_Request
  - ✓ ✓ KnowledgeBaseCreateRequestValidatorTest
    - ✓ Should\_Error\_Result\_When\_CategoryId\_Is\_Zero
    - ✓ Should\_Error\_Result\_When\_Miss\_Problem(problem: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_Problem(problem: null)
    - ✓ Should\_Error\_Result\_When\_Miss\_Title(title: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_Title(title: null)
    - ✓ Should\_Valid\_Result\_When\_Valid\_Request
  - ✓ ✓ LabelCreateRequestValidatorTest
    - ✓ Should\_Error\_Result\_When\_Miss\_Name(name: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_Name(name: null)
    - ✓ Should\_Valid\_Result\_When\_Valid\_Request
  - ✓ ✓ ReportCreateRequestValidatorTest
    - ✓ Should\_Error\_Result\_When\_KnowledgeBaseId\_Is\_Zero
    - ✓ Should\_Error\_Result\_When\_Miss\_Content(content: "")
    - ✓ Should\_Error\_Result\_When\_Miss\_Content(content: null)
    - ✓ Should\_Valid\_Result\_When\_Valid\_Request
  - ✓ ✓ VoteCreateRequestValidatorTest
    - ✓ Should\_Error\_Result\_When\_KnowledgeBaseId\_Is\_Zero
    - ✓ Should\_Valid\_Result\_When\_Valid\_Request

*Hình 2.3.1a: Kiểm thử giá trị input cho phần nội dung*

✓	DaisyForum.ViewModels.UnitTesting.Systems
✓	CommandAssignRequestValidatorTest
✓	Should_Error_Result_When_Request_CommandIds_Contains_Any_Empty_Item
✓	Should_Error_Result_When_Request_CommandIds_Empty
✓	Should_Error_Result_When_Request_Miss_CommandIds
✓	Should_Valid_Result_When_Valid_Request
✓	FunctionCreateRequestValidatorTest
✓	Should_Error_Result_When_Miss_Id(data: "")
✓	Should_Error_Result_When_Miss_Id(data: null)
✓	Should_Error_Result_When_Miss_Name(data: "")
✓	Should_Error_Result_When_Miss_Name(data: null)
✓	Should_Error_Result_When_Miss_Url(data: "")
✓	Should_Error_Result_When_Miss_Url(data: null)
✓	Should_Valid_Result_When_Valid_Request
✓	RoleAssignRequestValidatorTest
✓	Should_Error_Result_When_Request_Miss_RoleNames
✓	Should_Error_Result_When_Request_RoleNames_Contains_Empty_Item
✓	Should_Error_Result_When_Request_RoleNames_Is_Empty
✓	Should_Valid_Result_When_Valid_Request
✓	RoleCreateRequestValidatorTest
✓	Should_Error_Result_When_Request_Miss_RoleId
✓	Should_Error_Result_When_Request_Miss_RoleName
✓	Should_Error_Result_When_Request_Role_Empty
✓	Should_Valid_Result_When_Valid_Request
✓	UpdatePermissionRequestValidatorTest
✓	Should_Error_Result_When_Request_Miss_Permissions
✓	Should_Error_Result_When_Request_Permissions_Contains_Miss_CommandId_Item(data: "")
✓	Should_Error_Result_When_Request_Permissions_Contains_Miss_CommandId_Item(data: null)
✓	Should_Error_Result_When_Request_Permissions_Contains_Miss_FunctionId_Item(data: "")
✓	Should_Error_Result_When_Request_Permissions_Contains_Miss_FunctionId_Item(data: null)
✓	Should_Error_Result_When_Request_Permissions_Contains_Miss_RoleId_Item(data: "")
✓	Should_Error_Result_When_Request_Permissions_Contains_Miss_RoleId_Item(data: null)
✓	Should_Error_Result_When_Request_Permissions_Is_Empty
✓	Should_Valid_Result_When_Valid_Request
✓	UserCreateRequestValidatorTest
✓	Should_Error_Result_When_Miss_FirstName(data: "")
✓	Should_Error_Result_When_Miss_FirstName(data: null)
✓	Should_Error_Result_When_Miss_LastName(data: "")
✓	Should_Error_Result_When_Miss_LastName(data: null)
✓	Should_Error_Result_When_Miss_PhoneNumber(data: "")
✓	Should_Error_Result_When_Miss_PhoneNumber(data: null)
✓	Should_Error_Result_When_Miss_UserName(userName: "")
✓	Should_Error_Result_When_Miss_UserName(userName: null)
✓	Should_Error_Result_When_Password_Not_Match(data: "")
✓	Should_Error_Result_When_Password_Not_Match(data: "1234567")
✓	Should_Error_Result_When_Password_Not_Match(data: "Admin123")
✓	Should_Error_Result_When_Password_Not_Match(data: "sdasfaf")
✓	Should_Error_Result_When_Password_Not_Match(data: null)
✓	Should_Valid_Result_When_Valid_Request
✓	UserPasswordChangeRequestValidatorTest
✓	Should_Error_Result_When_Request_Miss_CurrentPassword(data: "")
✓	Should_Error_Result_When_Request_Miss_CurrentPassword(data: null)
✓	Should_Error_Result_When_Request_Miss_NewPassword(data: "")
✓	Should_Error_Result_When_Request_Miss_NewPassword(data: null)
✓	Should_Error_Result_When_Request_Miss_UserId
✓	Should_Error_Result_When_Request_NewPassword_Is_Not_Complex
✓	Should_Error_Result_When_Request_NewPassword_Is_NotEnough_8_Characters
✓	Should_Error_Result_When_Request_NewPassword_Is_Same_OldPassword
✓	Should_Valid_Result_When_Valid_Request

*Hình 2.3.1b: Kiểm thử giá trị input cho phần user*

### 3.2. Kiểm thử logic của chức năng bằng UnitTest

Phân kiểm thử logic cho trang web sử dụng thêm thư viện Mock giúp giả lập các thành phần DI của hệ thống để tiến hành kiểm tra.

```
✓ DaisyForum.BackendServer.UnitTest.Authorization
  ✓ ClaimRequirementAttributeTest
    ✓ Constructor_CreateInstance_ShouldBe_NotNull
  ✓ ClaimRequirementFilterTest
    ✓ Constructor_CreateInstance_ShouldBe_NotNull
    ✓ OnAuthorization_ValidClaims_SuccessResult
```

*Hình 2.3.2a: Kiểm thử logic cho phần Authorization*

✓	DaisyForum.BackendServer.UnitTesting.Controllers
✓	CategoriesControllerTest
✓	ShouldCreateInstance_NotNull_Success
✓	CommandsControllerTest
✓	ShouldCreateInstance_NotNull_Success
✓	FunctionsControllerTest
✓	DeleteFunction_ValidInput_Failed
✓	DeleteFunction_ValidInput_Success
✓	GetById_HasData_ReturnSuccess
✓	GetCommandsInFunction_ReturnSuccess
✓	GetFunctionsPaging_HasFilter_ReturnSuccess
✓	GetFunctionsPaging_NoFilter_ReturnSuccess
✓	GetFunction_HasData_ReturnSuccess
✓	PostFunction_ValidInput_Failed
✓	PostFunction_ValidInput_Success
✓	PutFunction_ValidInput_Failed
✓	PutFunction_ValidInput_Success
✓	Should_Create_Instance_Not_Null_Success
✓	KnowledgeBasesControllerTest
✓	PostKnowledgeBase_ValidInput_Success
✓	ShouldCreateInstance_NotNull_Success
✓	LabelsControllerTest
✓	ShouldCreateInstance_NotNull_Success
✓	PermissionsControllerTest
✓	ShouldCreateInstance_NotNull_Success
✓	RolesControllerTest
✓	DeleteRole_ValidInput_Failed
✓	DeleteRole_ValidInput_Success
✓	GetById_HasData_ReturnSuccess
✓	GetById_ThrowException_Failed
✓	GetRolesPaging_HasFilter_ReturnSuccess
✓	GetRolesPaging_NoFilter_ReturnSuccess
✓	GetRolesPaging_ThrowException_Failed
✓	GetRoles_HasData_ReturnSuccess
✓	GetRoles_ThrowException_Failed
✓	PostRole_ValidInput_Failed
✓	PostRole_ValidInput_Success
✓	PutRole_ValidInput_Failed
✓	PutRole_ValidInput_Success
✓	ShouldCreateInstance_NotNull_Success
✓	StatisticsControllerTest
✓	ShouldCreateInstance_NotNull_Success

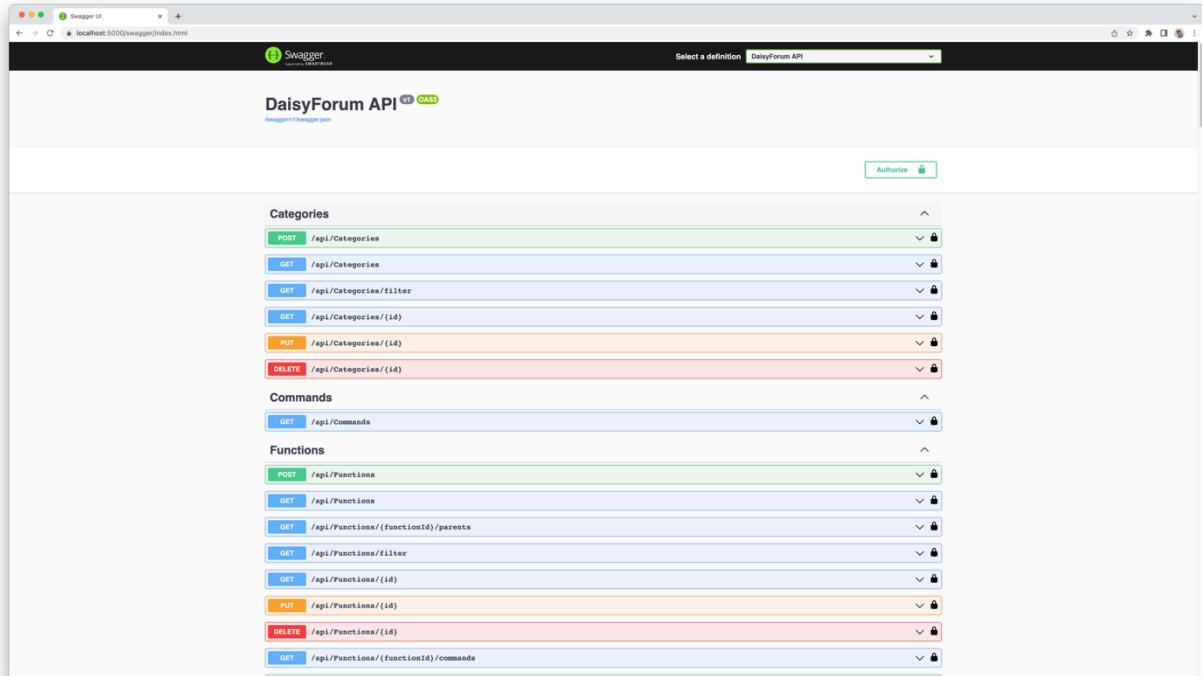
Hình 2.3.2b: Kiểm thử logic cho phần User

✓	UsersControllerTest
✓	DeleteUser_ValidInput_Failed
✓	DeleteUser_ValidInput_Success
✓	GetById_HasData_ReturnSuccess
✓	GetById_ThrowException_Failed
✓	GetUsersPaging_HasFilter_ReturnSuccess
✓	GetUsersPaging_NoFilter_ReturnSuccess
✓	GetUsersPaging_ThrowException_Failed
✓	GetUsers_HasData_ReturnSuccess
✓	GetUsers_ThrowException_Failed
✓	PostUser_ValidInput_Failed
✓	PutUser_ValidInput_Failed
✓	PutUser_ValidInput_Success
✓	ShouldCreateInstance_NotNull_Success
✓	VotesControllerTest
✓	GetVotes_ValidKbld_RecordMatch
✓	PostVote_NotFoundKbld_BadRequest
✓	DaisyForum.BackendServer.UnitTest.Helpers
✓	ApiBadRequestResponseTest
✓	Constructor.CreateInstance_Identity_Result_NotNull
✓	Constructor.CreateInstance_Message_NotNull
✓	Constructor.CreateInstance_ModelState_Valid_ShouldBe_Throw_Exception
✓	Constructor.CreateInstance_ShouldBe_NotNull
✓	ApiNotFoundResponseTest
✓	Constructor.CreateInstance_ShouldBe_NotNull
✓	ApiResponseTest
✓	Constructor.CreateInstance_ShouldBe_NotNull
✓	ErrorWrappingMiddlewareTest
✓	ErrorWrapping_Middleware_Should_Return_Error
✓	TextHelperTest
✓	ToUnsignString_ValidInput_SuccessResult

Hình 2.3.2c: Kiểm thử logic cho phần Nội dung

## 4. KẾT QUẢ

### 4.1. Backend Server



Hình 2.4.1a: Giao diện danh sách API Swagger

### Available authorizations

X

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

#### Bearer (OAuth2, implicit)

##### Authorized

Authorization URL: <https://localhost:5000/connect/authorize>

Flow: implicit

client\_id: \*\*\*\*\*

[Logout](#)

[Close](#)

Hình 2.4.1b: Giao diện Authorization Swagger

## 4.2. Admin App

Danh mục

Mã	Tên	Alias	Thứ tự
1	Ngôn ngữ lập trình	ngon-ngho-lap-trinh	0
2	Framework và thư viện	framework-va-thu-vien	0
14	.NET	.net	0
15	Angular	angular	0
16	React	react	0
17	Vue	vue	0
18	Node.js	nodejs	0
19	jQuery	jquery	0
3	Cơ sở dữ liệu	co-so-du-dieu	0
20	SQL Server	sql-server	0
21	MySQL	mysql	0
22	Oracle	oracle	0
23	MongoDB	mongodb	0
4	Công nghệ web và ứng dụng	cong-nghe-web-va-ung-dung	0
24	HTML	html	0
25	CSS	css	0
26	HTTP	http	0
27	SSL	ssl	0
38	Android Studio	android-studio	0
5	Công cụ và phần mềm	cong-cu-va-phien-mem	0
32	Visual Studio	visual-studio	0
33	Visual Studio Code	visual-studio-code	0
34	Eclipse	eclipse	0
35	Git	git	0
6	Mô hình thiết kế	mo-hinh-thiet-ke	0
7	Khái niệm và kỹ thuật lập trình	khai-niem-va-ky-thuat-lap-trinh	0
8	Vấn đề bảo mật và an ninh	van-de-bao-mat-va-an-ninh	0
9	Hệ điều hành	he-dieu-hanh	0

Hình 2.4.2a: Giao diện quản lý danh mục

Bài viết

Mã	Tiêu đề	Danh mục	Bình luận	Thích	Báo cáo
4081	Java Repeat Task Every Random Seconds	Ngôn ngữ lập trình	11	11	11
4082	Why are Java Optionals immutable?	Ngôn ngữ lập trình			
4083	Text Overlay Image with Darkened Opacity React Native	Ngôn ngữ lập trình			
4084	Why ternary operator in swift is so picky?	Ngôn ngữ lập trình			
4085	hideshow fab with scale animation	Ngôn ngữ lập trình			
4086	Accessing pointer member of the structure using structure pointer	Ngôn ngữ lập trình			
4087	How To Disable 2nd Saturday 4th Saturday, Sunday And Holiday Date Using jQuery Calendar	Ngôn ngữ lập trình			
4088	Resizing containers in bootstrap	Ngôn ngữ lập trình			
4089	Changing Theme in Windows 10 UWP App Programmatically	Ngôn ngữ lập trình			
4090	TextBox Value Disappears VB6	Ngôn ngữ lập trình			

Hình 2.4.2b: Giao diện quản lý bài viết

Tên	Mã	Thứ tự
Nội dung	CONTENT	0
Danh mục	CONTENT_CATEGORY	0
Trang	CONTENT_COMMENT	3
Bài viết	CONTENT KNOWLEDGE BASE	2
Báo xấu	CONTENT REPORT	3
Thống kê	DASHBOARD	1
Comment theo tháng	STATISTIC	0
Bài đăng hàng tháng	STATISTIC_MONTHLY COMMENT	3
Đăng ký từng tháng	STATISTIC_MONTHLY NEW KNOWLEDGE BASE	2
Hệ thống	SYSTEM	0
Chức năng	SYSTEM FUNCTION	0
Quyền hạn	SYSTEM PERMISSION	0
Nhóm quyền	SYSTEM ROLE	0
Người dùng	SYSTEM USER	0

Hình 2.4.2c: Giao diện quản lý chức năng

Chọn nhóm	Tên hành động
Nhóm người dùng	Thêm
Admin	Xóa
	Sửa
	Xem

Hình 2.4.2d: Giao diện quản lý quyền hạn

Mã ID	Mô tả nhóm ID
Admin	Admin
Member	Member

Hình 2.4.2e: Giao diện quản lý nhóm quyền

<input type="button" value="Thêm"/> <input type="button" value="Xóa"/> <input type="button" value="Sửa"/>					<input type="checkbox"/> <a href="#">Gán quyền</a>	<input type="text"/> Nhập tên người dùng...
Tài khoản	Tên	Họ	Email	Ngày sinh	Số điện thoại	
<input checked="" type="checkbox"/> admin	Quản trị	1	admin.ssdaisy@gmail.com	01/01/0001	0964732231	
<input type="checkbox"/> nguyenphuduc62001@gmail.com	Nguyễn	Phú Đức	nguyenphuduc62001@gmail.com	30/07/2001	0964732231	
Tổng số bản ghi: 2						<input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="6"/> <input type="button" value="7"/> <input type="button" value="8"/> <input type="button" value="9"/> <input type="button" value="10"/> <input type="button" value="▼"/>

*Hình 2.4.2f: Giao diện quản lý người dùng*

### 4.3. Web Portal



## Đăng nhập

Lưu đăng nhập?

Đăng nhập

[Quên mật khẩu?](#)

[Đăng ký tài khoản mới](#)

[Gửi lại email xác nhận](#)



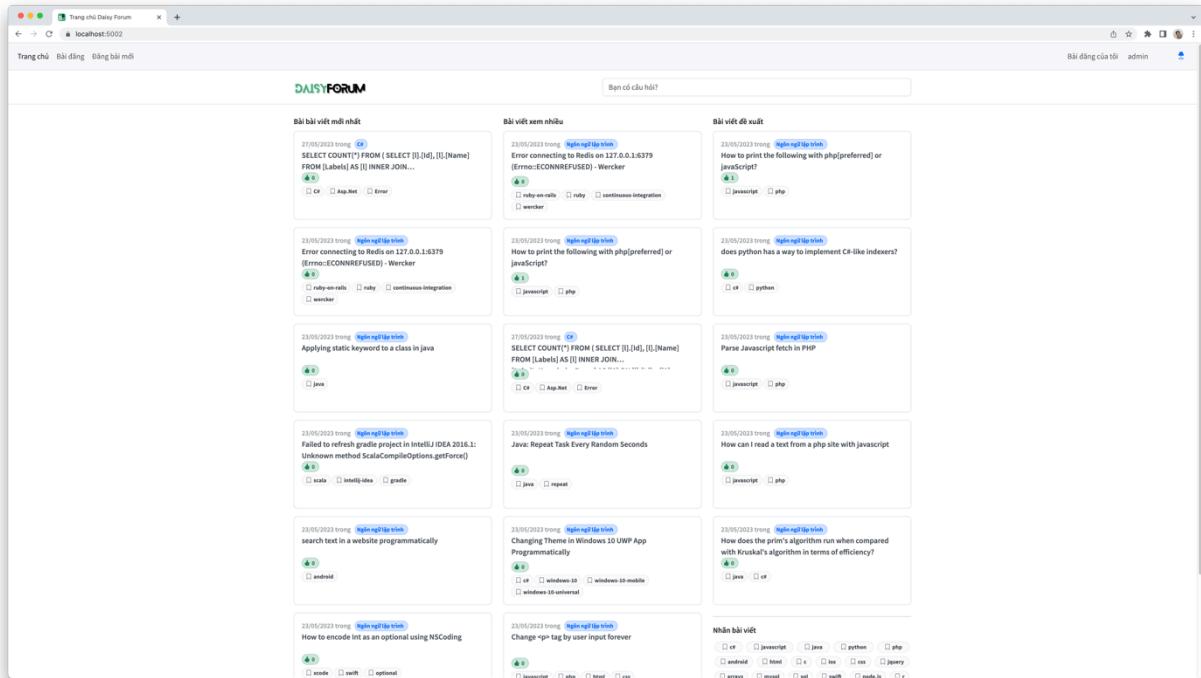
## Đăng ký tài khoản

Đăng ký

[→ Quay lại đăng nhập](#)

Hình 2.4.3a: Giao diện đăng nhập

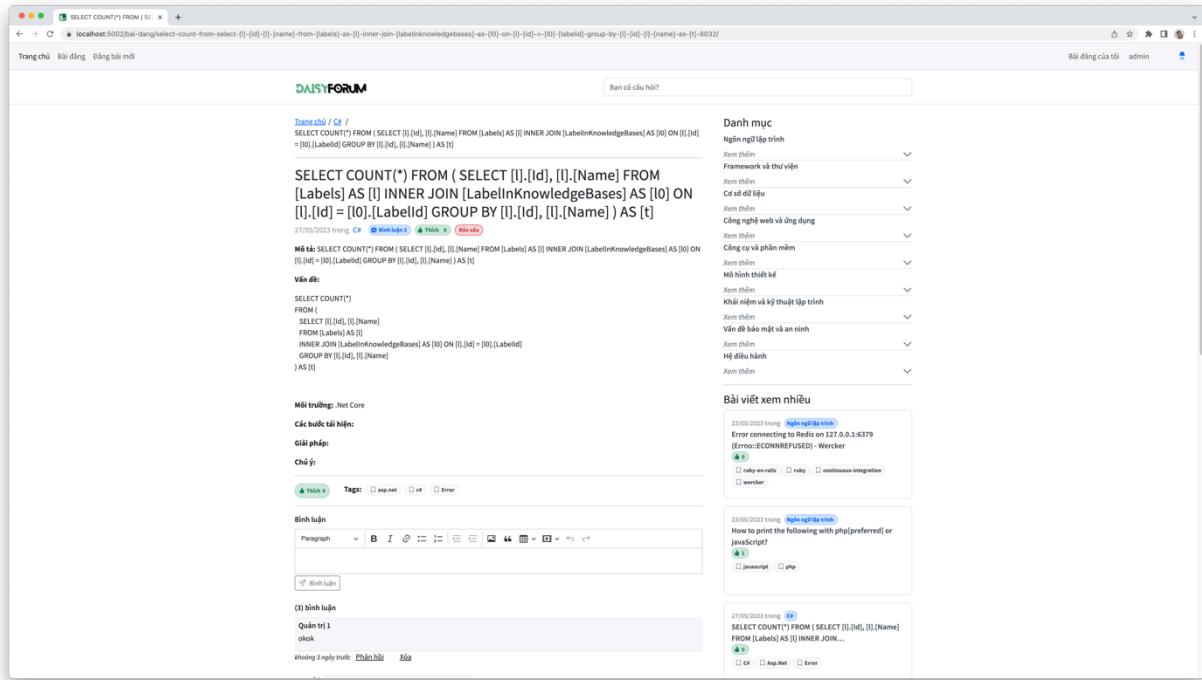
Hình 2.4.3b: Giao diện đăng ký



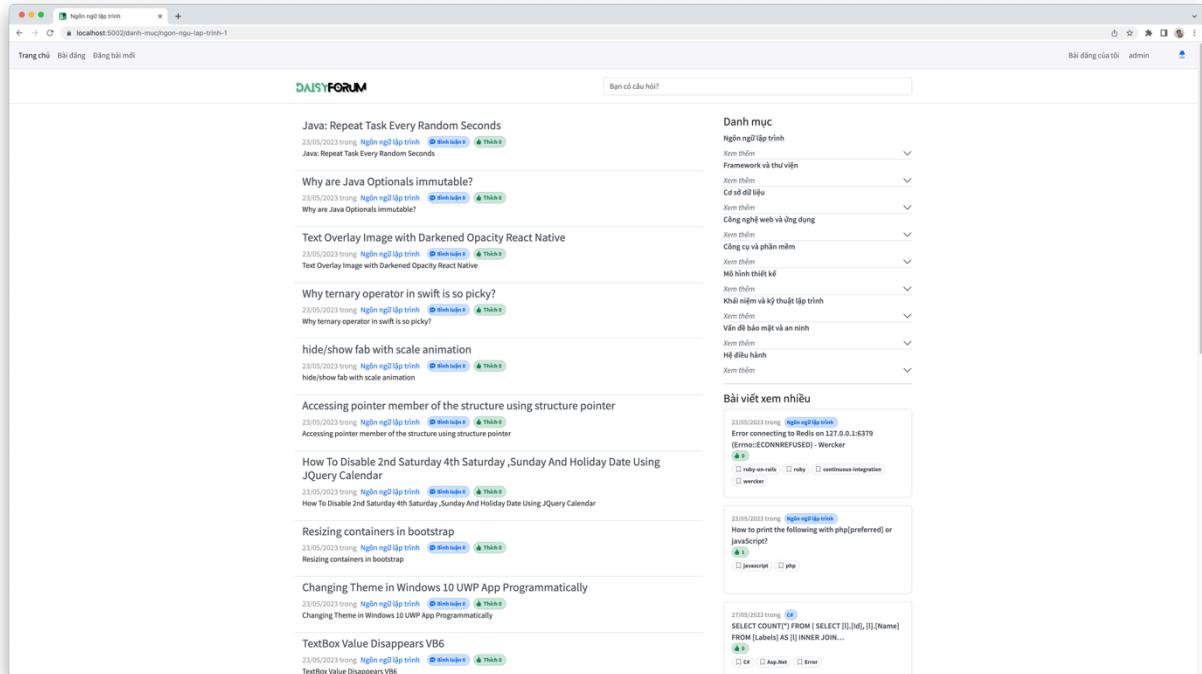
The screenshot shows the homepage of a forum titled "DAISYFORUM". The page displays a grid of 12 post cards, each containing a snippet of text, a timestamp, a category, and tags. The posts cover various topics like database errors, Java programming, and file handling.

Bài bài viết mới nhất	Bài viết xem nhiều	Bài viết đề xuất
23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> SELECT COUNT(*) FROM ( SELECT [...] ) AS [...] INNER JOIN...	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Error connecting to Redis on 127.0.0.1:6379 (Error=ECONNREFUSED) - Worker	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> How to print the following with php[preferred] or javascript?
23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Error connecting to Redis on 127.0.0.1:6379 (Error=ECONNREFUSED) - Worker	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> How to print the following with php[preferred] or javascript?	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Does python has a way to implement C-like indexers?
23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Applying static keyword to a class in java	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> SELECT COUNT(*) FROM ( SELECT [...] ) AS [...] INNER JOIN...	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Parse Javascript fetch in PHP
23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Failed to refresh gradle project in IntelliJ IDEA 2018.1. Unknown method GradleCompileOptions.getForce()	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Java: Repeat Task Every Random Seconds	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> How can I read a text from a php site with javascript
23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> search text in a website programmatically	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Changing Theme in Windows 10 UWP App Programmatically	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> How does the print algorithm run when compared with Kruskal's algorithm in terms of efficiency?
23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> How to encode int as an optional using NSCoder	23/05/2023 trong <a href="#">Ngôn ngữ lập trình</a> Change <p> tag by user input forever	Nhận bài viết:
		<input type="checkbox"/> c# <input type="checkbox"/> javascript <input type="checkbox"/> java <input type="checkbox"/> python <input type="checkbox"/> php <input type="checkbox"/> android <input type="checkbox"/> html <input type="checkbox"/> c <input type="checkbox"/> c++ <input type="checkbox"/> css <input type="checkbox"/> jquery <input type="checkbox"/> array <input type="checkbox"/> mysql <input type="checkbox"/> sql <input type="checkbox"/> swift <input type="checkbox"/> node.js <input type="checkbox"/>

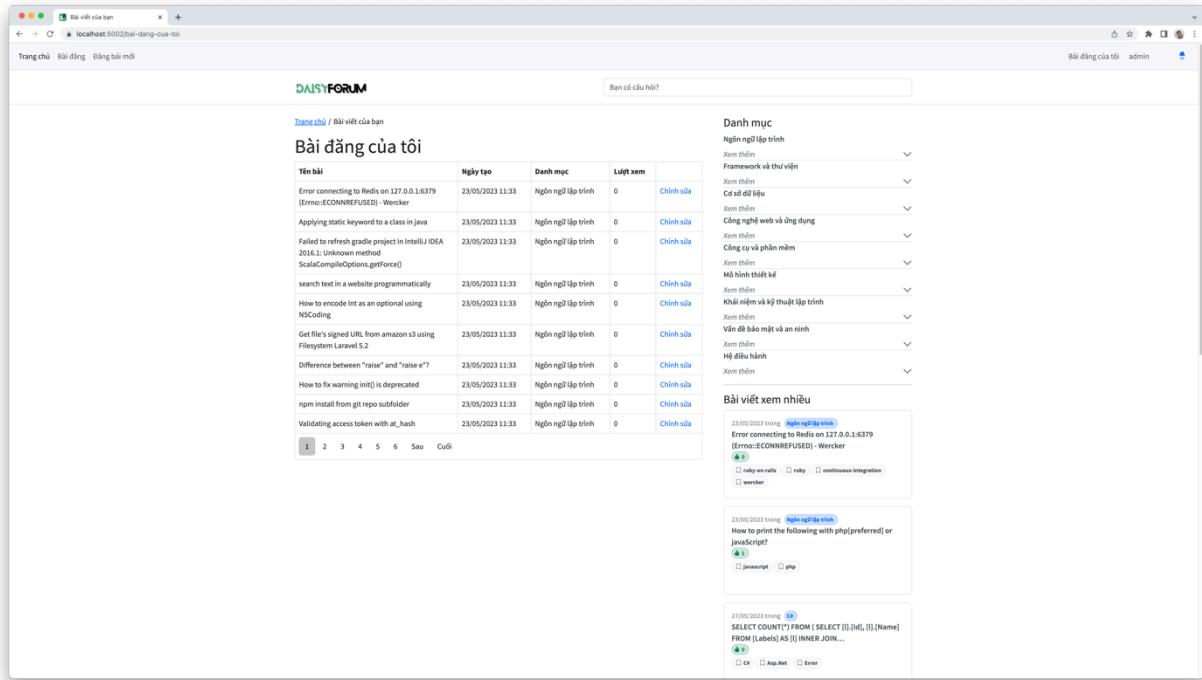
Hình 2.4.3c: Giao diện trang chủ



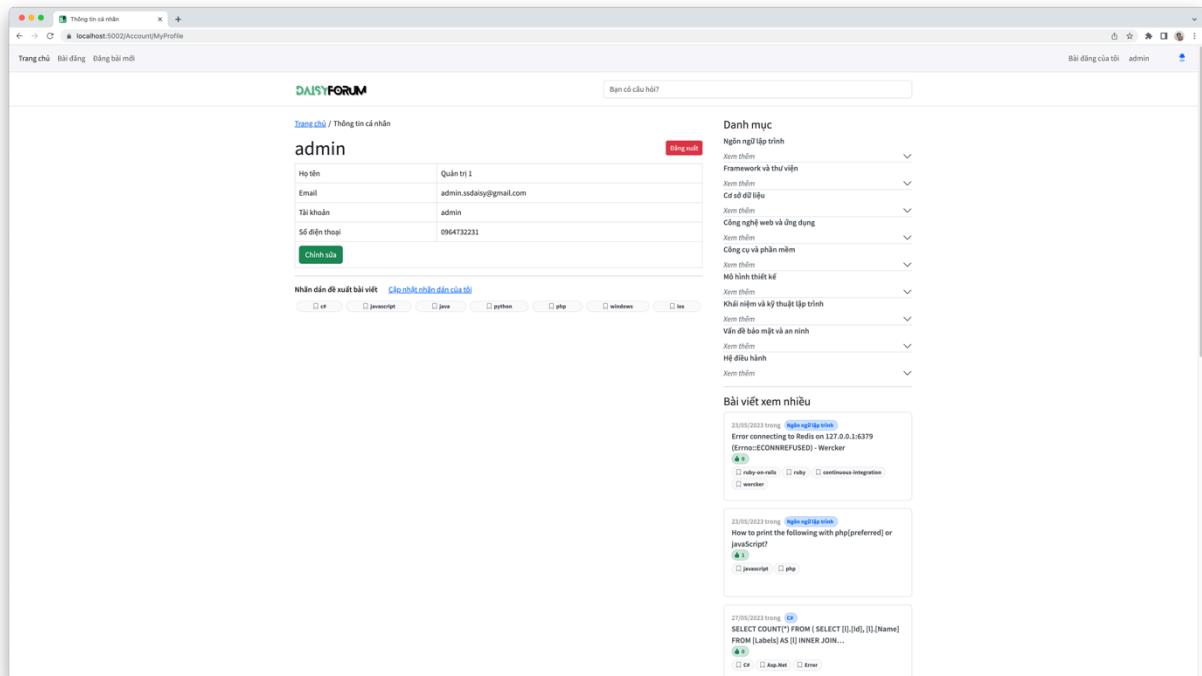
Hình 2.4.3d: Giao diện bài viết



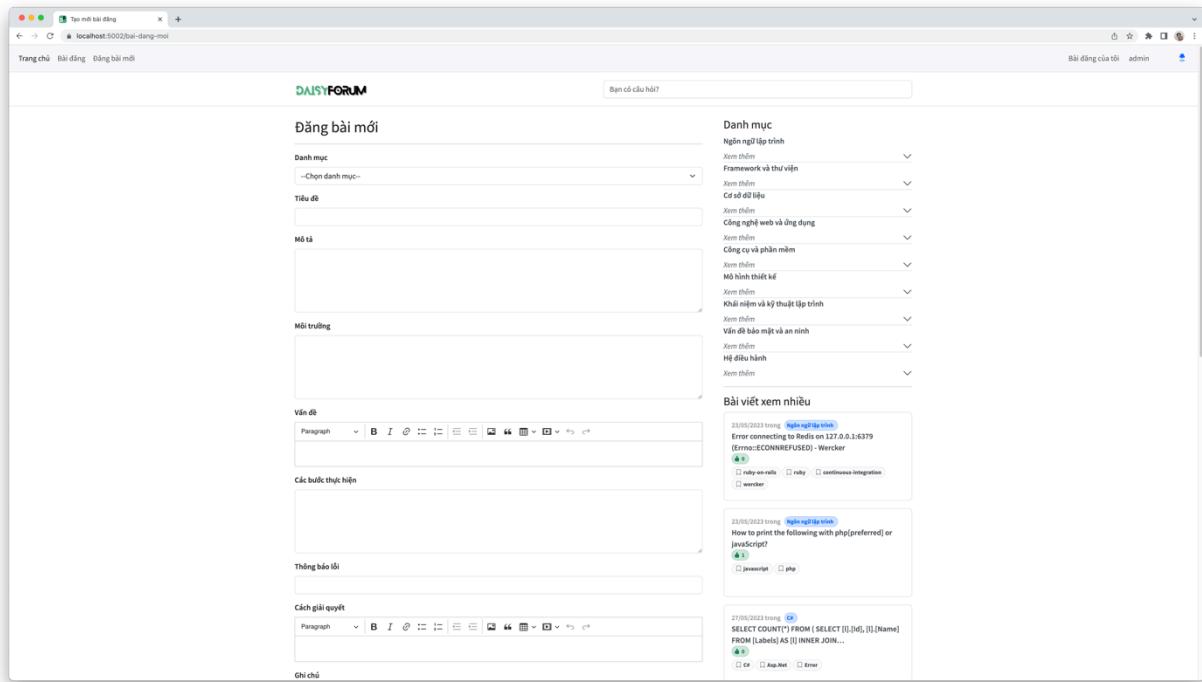
Hình 2.4.3e: Giao diện bài viết theo danh mục



Hình 2.4.3f: Giao diện quản lý bài viết



Hình 2.4.3g: Giao diện quản lý thông tin người dùng



Hình 2.4.3h: Giao diện đăng bài viết

## 5. TỐI UU WEBSITE

### 5.1. Tối ưu tốc độ với Distributed Cache

	<b>Id</b>	<b>Value</b>	<b>ExpiresAtTime</b>	<b>SlidingExpirationInSeconds</b>
1	Categories	0x5B7B224964223A312C224E616D65223A224E675C7530304634...	2023-05-31 11:18:31.8412790 +00:00	43200
2	LatestKnowledgeBases	0x5B7B224964223A383033322C2243617465676F72794964223A...	2023-05-31 01:18:28.7214480 +00:00	7200
3	PopularKnowledgeBases	0x5B7B224964223A383033312C2243617465676F72794964223A...	2023-05-31 23:18:31.8222540 +00:00	86400
4	PopularLabels	0x5B7B224964223A22632D222C224E616D65223A226323227D2C...	2023-05-31 11:18:28.7427850 +00:00	43200
5	RecentComments	0x5B7B224964223A352C22436F6E74656E74223A6E756C6C2C22...	2023-05-31 11:18:31.8286950 +00:00	43200

Hình 2.5.1: Tối ưu tối độ với Distributed Cache

### 5.2. Tối ưu bảo mật với Security headers

```
app.UseCsp(opts => opts
    .BlockAllMixedContent()
    .StyleSources(s => s.Self())
    .StyleSources(s => s.UnsafeInline())
    .FontSources(s => s.Self())
    .FormActions(s => s.Self())
    .FrameAncestors(s => s.Self())
    .ImageSources(s => s.Self())
    .ScriptSources(s => s.Self())
);
```

Hình 2.5.2: Tối ưu bảo mật với thư viện Security headers

## CHƯƠNG 4: KẾT LUẬN

### 1. KẾT LUẬN

- **Mức độ hoàn thành dự án:** 90%
- **Tích cực:**
  - Dự án xây dựng lên một trang diễn đàn với đầy đủ chức năng cơ bản.
  - Áp dụng tốt phần phân quyền cho quản trị viên
  - Ứng dụng thuật toán để xuất bài viết đến người dùng giúp quá trình sử dụng trở lên tiện dụng.
  - Ứng dụng việc đánh giá bình luận để phân loại bình luận tiêu cực một cách tự động.
  - Giải quyết được các vấn đề về bảo mật website.
  - Giải quyết được vấn đề về dữ liệu lớn.
- **Hạn chế:**
  - Dự án cần cải thiện khả năng phân loại bình luận giúp khả năng nhận diện trở lên chính xác hơn.
  - Dự án cần cải thiện khả năng để xuất bài viết.
  - Bổ sung thêm nhiều tính năng liên quan đến tiện ích sử dụng cho người dùng.

### 2. ĐÁNH GIÁ

#### 4.1. Tính mới

- Đề xuất bài viết liên quan đến nhu cầu của người dùng.
- Phân loại bình luận tiêu cực tự động.

#### 4.2. Đóng góp

- Kết nối cộng đồng: Tạo ra một nền tảng cho phép các thành viên trong cộng đồng giao lưu và hợp tác với nhau. Điều này giúp xây dựng một môi trường tương tác tích cực giữa các thành viên, từ đó giúp tăng cường sự gắn kết và sự hỗ trợ cho nhau.
- Chia sẻ kiến thức: Cung cấp một nơi an toàn và trung thực để các thành viên chia sẻ và nhận được kiến thức. Điều này giúp xây dựng một cộng đồng học tập, nơi mọi người có thể học hỏi và chia sẻ kiến thức của mình với nhau. Nhờ

đó, các thành viên sẽ có được sự giúp đỡ và hỗ trợ trong việc giải quyết các vấn đề công nghệ một cách nhanh chóng và hiệu quả.

- Tạo ra một cộng đồng tự trị: Tạo ra một cộng đồng tự trị với các thành viên có thể giúp đỡ và hỗ trợ nhau. Điều này giúp xây dựng một môi trường tương tác tích cực giữa các thành viên, từ đó giúp tăng cường sự gắn kết và sự hỗ trợ cho nhau.
- Phát triển kỹ năng: Hỗ trợ các thành viên phát triển kỹ năng của họ, bao gồm viết văn bản, giao tiếp và quản lý thời gian. Để đạt được mục tiêu này, đề tài sẽ tập trung vào việc cung cấp các tài liệu hướng dẫn và chia sẻ kinh nghiệm từ các thành viên có kinh nghiệm trong lĩnh vực công nghệ. Các thành viên sẽ được khuyến khích tham gia vào các hoạt động trao đổi kiến thức trên diễn đàn, từ đó rèn luyện và phát triển các kỹ năng cá nhân của mình.
- Cập nhật công nghệ: Cung cấp cho các thành viên thông tin về các công nghệ mới nhất và kỹ thuật phát triển website. Điều này giúp các thành viên cập nhật được các thông tin mới nhất về công nghệ và có thể áp dụng vào công việc của mình. Đồng thời, các thành viên cũng sẽ được hỗ trợ trong việc giải quyết các vấn đề liên quan đến công nghệ một cách nhanh chóng và hiệu quả hơn.

### **3. HƯỚNG PHÁT TRIỂN**

- Cải thiện mức độ nhanh chóng của tính năng đề xuất bài viết.
- Cải thiện mức độ chính xác của tính năng phân loại bình luận.
- Bổ sung thêm những tiện ích để tăng trải nghiệm người dùng
- Tăng hiệu năng của trang web.
- Kết hợp việc sử dụng web và app mobile.

## TÀI LIỆU THAM KHẢO

- Agile là gì? Scrum là gì? Các công cụ quản lý dự án theo Agile mà bạn nên biết  
<https://topdev.vn/blog/agile-la-gi-scrum-la-gi/>
- .NET Core là gì? Tổng quan về .Net Core  
<https://codegym.vn/blog/2020/06/29/net-core-la-gi-tong-quan-ve-net-core/>
- Razor View Engine trong ASP.NET Core MVC  
<https://tedu.com.vn/lap-trinh-aspnet-core/razor-view-engine-trong-aspnet-core-mvc-240.html>
- FluentValidator  
<https://docs.fluentvalidation.net/en/latest/aspnet.html>
- RESTful API là gì? Cách thiết kế RESTful API  
<https://topdev.vn/blog/restful-api-la-gi/>
- ASP.NET documentation  
<https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0>
- Bootstrap documentation  
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>