**Bộ giáo dục và đào tạo**

**Trường Đại học Ngoại ngữ - Tin học TP.HCM**



**Đồ án 1**

*Đề tài:*

**Xây dựng website meme**

GVHD: Th.s Vũ Đình ÁI

SVTH: Đỗ Tấn Lộc – 16DH110351

Phan Minh Trường – 16DH110251

Đỗ Đức Kiến – 16DH110347

**Tháng 5 Năm 2019**

**Lời mở đầu**

Vào ngày 27/6/2016, Microsoft chính thức phát hành mã nguồn mở của bộ .Net Framework nổi tiếng của mình với tên gọi mới là .Net Core 1.0 cùng với ASP.Net core 1.0 và Entity Framework Core 1.0 chạy được trên cả Mac và Linux thay vì chỉ chạy trên Windows như trước đây. Microsoft đã đầu tư rất nhiều vào mã nguồn mở, bằng chứng cho thấy là mặc dù chỉ mới ra mắt từ năm 2016 nhưng đến nay họ đã có bản .Net Core 2.2 hoàn chỉnh và .Net Core 3.0 Preview.

Trong cuộc sống hiện nay áp lực từ công việc cũng như việc học tập đối với các công nhân viên chức , các thầy cô giáo và các bạn học sinh , sinh viên là vô cùng lớn . Hiểu được điều này nhóm chúng em gồm 3 thành viên : Đỗ Tấn Lộc , Phan Minh Trường và Đỗ Đức Kiến với niềm đam mê vô tận đối với Game , Meme và Kamen Rider đã cùng nhau tạo ra website MemeTươi để có thể đem niềm đam mê của mình đến với mọi người cũng như giúp họ giải tỏa căng thẳng sau những giờ làm việc và học tập . Thú thật đây là lần đầu chúng em làm web nên cũng khó tránh khỏi những sai sót nhưng với sự hướng dẫn của thầy chúng em hy vọng trang web MemeTươi vẫn có đủ khả năng để phục vụ được cho mọi người.

Một số vấn đề nhóm chúng em đã nghiên cứu như ASP.net Core, cấu trúc JSON , cách sử dụng template để trang web đẹp hơn , Repository và Entity Framework Core. Trong quá trình nghiên cứu qua Internet và với sự giúp đỡ nhiệt tình của giảng viên hướng dẫn. Nhóm em bước đầu đã xây dựng được một trang web giải trí hoàn chỉnh với những tính năng cơ bản như thêm bài viết , tìm kiếm và phân trang. Do sự thiếu hiểu biết về công nghệ nên trang web vẫn còn những hạn chế mà nhóm chúng em hy vọng rằng mình sẽ khắc phục được trong tương lai.

Mục lục

Phần 1 Tổng quan về ASP.NET CORE 1

I. Lịch sử ASP.NET Core 1

I.1. ASP.NET 1

I.2. ASP.NET CORE 2

II. Một số tính năng của ASP.NET CORE 3

II.1. ASP.NET Core Identity 4

II.2. Razor Pages 5

III. Một số Pattern được sử dụng trong ASP.NET CORE 5

III.1. MVC 5

III.2. Repository pattern 6

Phần 2 Triển khai đề tài 8

I. Mô tả đề tài 8

I.1. Tổng quan: 8

I.2. Chức năng: 8

II. Xây dựng giao diện 9

II.1. Người dùng 9

II.2. Admin 11

III. Phân tích thiết kế cơ sở dữ liệu 16

III.1. Bảng Lịch sử Upvote 17

III.2. Bảng Thể loại 17

III.3. Bảng Meme (Bài viết) 18

III.4. Bảng Meme cần duyệt 19

Phần 3 Tổng kết 20

Tài liệu Tham Khảo 21

Danh mục hình ảnh

[Hình 1‑1 ASP.NET Core 2.2 1](file:///C:\Users\Loc%20Do\Desktop\DoAn1\Nhom%2011.docx#_Toc9430935)

[Hình 1‑2 Mô hình ASP.NET Core Identity 4](#_Toc9430936)

[Hình 1‑3 ASP.NET Core Razor Pages 5](#_Toc9430937)

[Hình 1‑4 Mô hình MVC 5](#_Toc9430938)

[Hình 1‑5 Repository pattern 6](#_Toc9430939)

[Hình 2‑1 Giao diện Trang chủ 9](#_Toc9430940)

[Hình 2‑2 Giao diện Chỉ tiết thể loại 10](#_Toc9430941)

[Hình 2‑3 Giao diện Chi tiết Bài Viết 11](#_Toc9430942)

[Hình 2‑4 Giao diện Quản lý meme 11](#_Toc9430943)

[Hình 2‑5 Giao diện Quản lý User 12](#_Toc9430944)

[Hình 2‑6 Giao diện Quản lý Upvote 12](#_Toc9430945)

[Hình 2‑7 Giao diện Quản lý Thể loại 13](#_Toc9430946)

[Hình 2‑8 Giao diện duyệt Meme 14](#_Toc9430947)

[Hình 2‑9 Giao diện Xem chi tiết bài cần duyệt 15](#_Toc9430948)

[Hình 2‑10 Sơ đồ CSDL 16](#_Toc9430949)

Danh mục từ viết tắt (nếu có)

|  |  |
| --- | --- |
| MVC | Model – View – Controller |
| ASP | Active Sever Pages |
| HTML | Hypertext Markup Language |

# Tổng quan về ASP.NET CORE

## Lịch sử ASP.NET Core

### ASP.NET

ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. Lần đầu tiên được đưa ra thị trường vào tháng 1 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages(ASP). ASP.NET được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi.NET language.

Hinh ‑ ASP.NET Core 2.2

Hình ‑ ASP.NET Core 2.2

Hình ‑ ASP.NET Core 2.2

ASP.NET MVC 1.0 được phát hành trong năm 2009, thực thi mô hình MVC (model–view–controller). ASP.NET MVC là một phần mềm mã nguồn mở, tách rời với thành phần độc quyền ASP.NET Web Forms. ASP.NET MVC đang nổi lên là phương pháp phát triển web mạnh nhất và phổ biến nhất trên nền ASP.NET hiện nay.

Sau khi phát hành phiên bản Internet Information Service 4.0 vào năm 1997, hãng Microsoft bắt đầu nghiên cứu một mô hình ứng dụng web để giải quyết những bất tiện của ASP, đặc biệt là việc tách riêng biệt phần thể hiện và phần nội dung cũng như cách viết mã rõ ràng hơn. Mark Anders, quản lý của nhóm IIS và Scott Guthrie, gia nhập Microsoft vào năm 1997 sau khi tốt nghiệp Đại học Duke, được giao nhiệm vụ định hình mô hình cần phát triển. Những thiết kế ban đầu được thực hiện trong vòng 2 tháng bởi Anders và Guthrie, Guthrie đã viết mã prototype đầu tiên trong khoảng thời gian nghỉ lễ Giáng sinh năm 1997.

### ASP.NET CORE

ASP.NET Core là gì?

ASP.NET CORE là một open – source mới và framework đa nền tảng cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile. Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-premises. Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux. Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

Hoàn cảnh ra đời của ASP.NET Core

Bản phát hành đầu tiên của ASP.NET đã xuất hiện cách đây 15 năm trước, nó là một phần của .NET Framework. Từ đó, hàng triệu lập trình viên đã sử dụng nó để xây dựng những ứng dụng web tuyệt vời, và trên những năm đó Microsoft đã phát triển thêm nhiều tính năng mới.

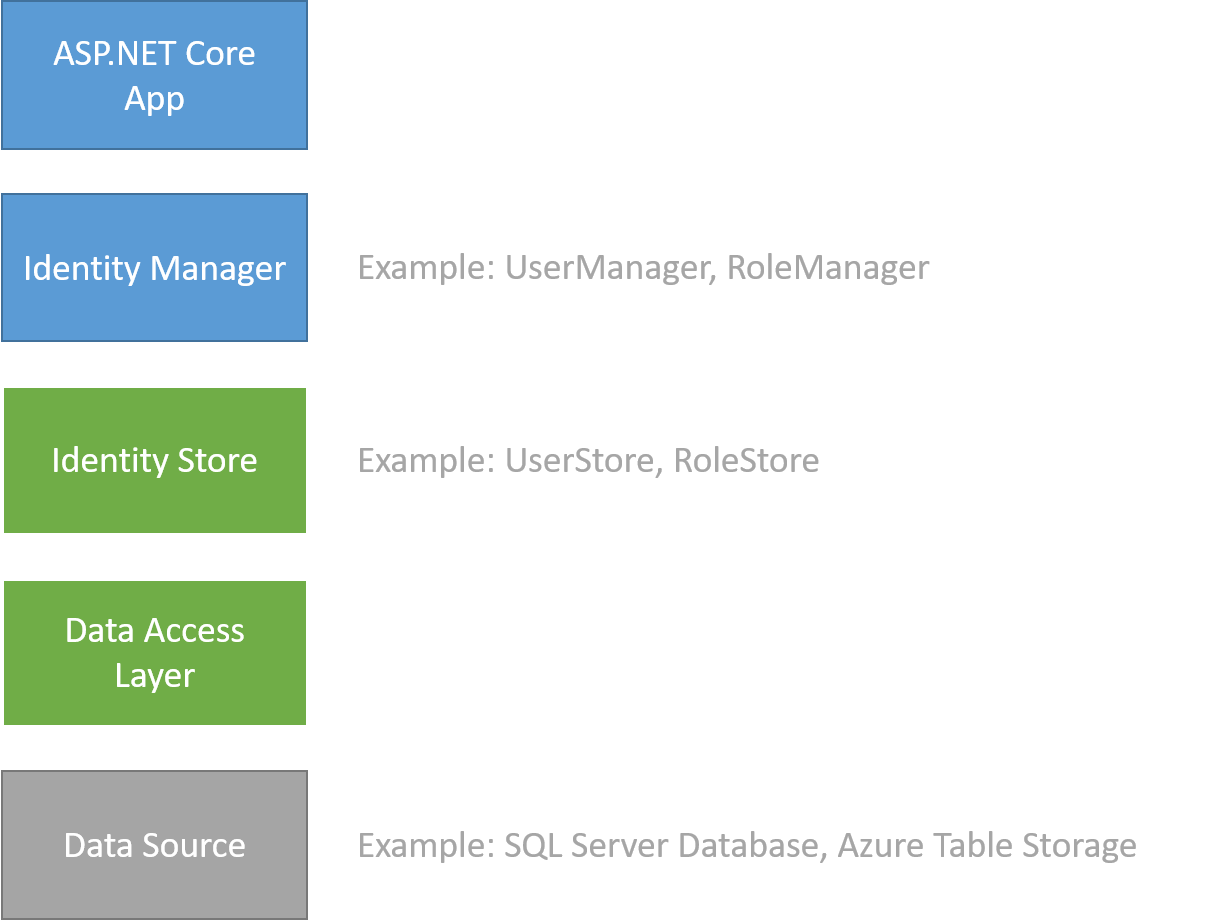
## Một số tính năng của ASP.NET CORE

ASP.NET Core có một số thay đổi kiến trúc lớn, đó là kết quả của việc học hỏi rất nhiều từ các framework module hóa khác. ASP.NET Core không còn dựa trên System.Web.dll nữa. Nó được dựa trên một tập hợp các gói, các module hay cũng được gọi là các Nuget packages. Điều này cho phép bạn tối ưu ứng dụng của bạn để chỉ bao gồm những packages nào cần thiết. Lợi ích của nó là giúp cho ứng dụng nhỏ hơn, bảo mật chặt chẽ hơn, giảm sự phức tạp, tối ưu hiệu suất hoạt động và giảm chi phí, thời gian cho việc phát triển.

Với ASP.NET Core bạn đạt được những nền tảng cải tiến dưới đây:

* Hợp nhất việc xây dựng web UI và web APIs.
* Tích hợp những client-side frameworks hiện đại và những luồng phát triển.
* Hệ thống cấu hình dựa trên môi trường đám mây thật sự.
* Dependency injection được xây dựng sẵn.
* HTTP request được tối ưu nhẹ hơn.
* Có thể host trên IIS hoặc self-host trong process của riêng bạn.
* Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning.
* Chuyển các thực thể, thành phần, module như những NuGet packages.
* Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại.
* Xây dựng và chạy đa nền tảng(Windows, Mac và Linux).
* Mã nguồn mở và tập trung vào cộng đồng.

### ASP.NET Core Identity

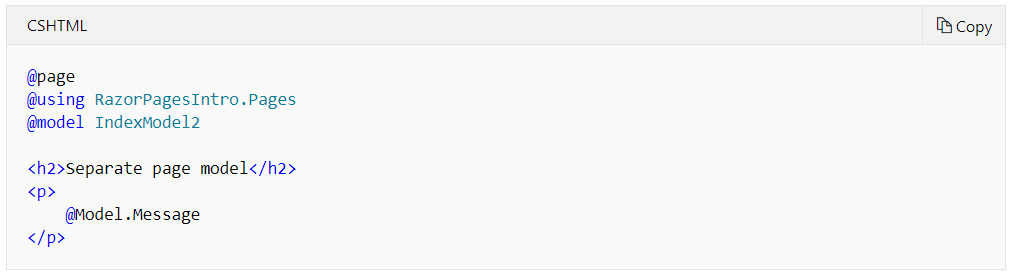


Hình ‑ Mô hình ASP.NET Core Identity

ASP.NET Core 2.1 đưa ra ASP.NET Core Identity như là một Razor Class Library. Các ứng dụng có sử dụng Identity có thể apply tính năng Identity scaffolder để thêm các source code được chọn chứa trong Identity Razor Class Library (RCL). Bạn có thể muốn gen ra source code mà bạn có thể thay đổi và custom. Ví dụ, bạn có thể chỉ thị cho scaffolder gen ra code sử dụng để cho việc đăng ký thành viên. Gen ra code có thể được ưu tiên hơn so với code hiện tại trong Identity RCL.

### Razor Pages

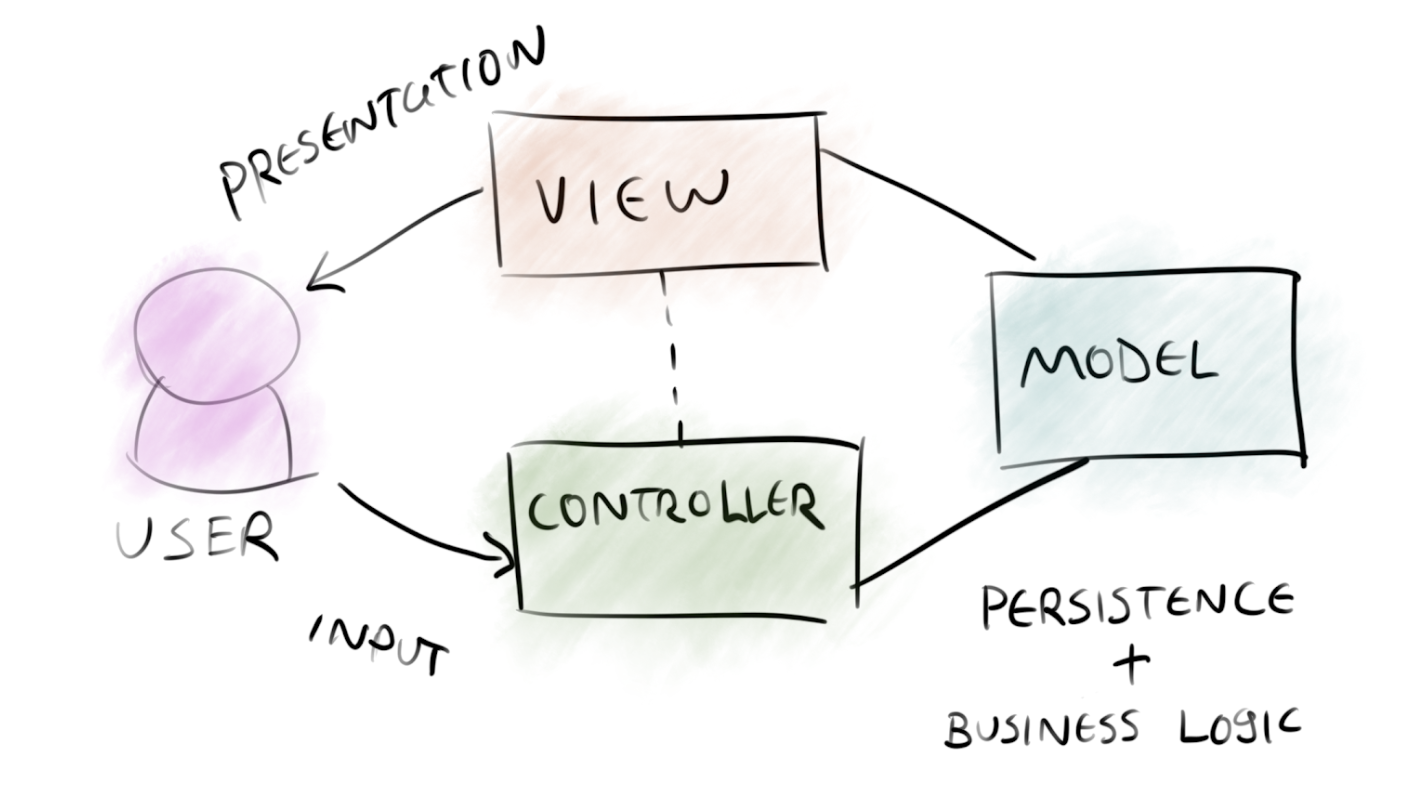
Razor pages là một tính năng mới của ASP.NET core MVC, nó giúp cho phân coding tập trung vào các trang mã hóa dễ dàng và hiệu quả hơn. Razor pages hoạt động với .NET core 2.0 hoặc các phiên bản cao hơn và được hỗ trợ bởi Visual Studio 2017 hoặc mới hơn.



Hình ‑ ASP.NET Core Razor Pages

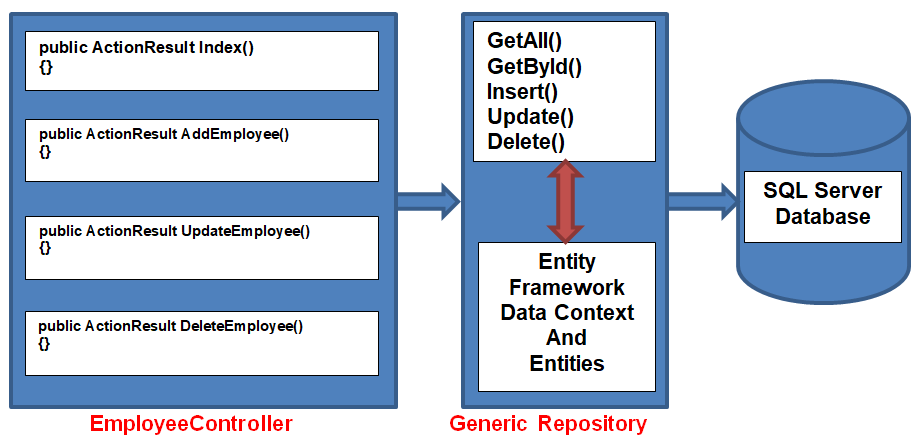
## Một số Pattern được sử dụng trong ASP.NET CORE

### MVC



Hình ‑ Mô hình MVC

### Repository pattern



Hình ‑ Repository pattern

**Khái niệm**

Repository Pattern là lớp trung gian giữa tầng Business Logic và Data Access, giúp cho việc truy cập dữ liệu chặt chẽ và bảo mật hơn.

Repository đóng vai trò là một lớp kết nối giữa tầng Business và Model của ứng dụng.

Thông thường thì các phần truy xuất, giao tiếp với database năm rải rác ở trong code, khi bạn muốn thực hiện một thao tác lên database thì phải tìm trong code cũng như tìm các thuộc tính trong bảng để xử lý. Điều này gây lãng phí thời gian và công sức rất nhiều.

Với Repository design pattern, thì việc thay đổi ở code sẽ không ảnh hưởng quá nhiều công sức chúng ra chỉnh sửa.

**Lợi ích của việc sử dụng Repository Pattern**

* Một nơi duy nhất để thay đổi quyền truy cập dữ liệu cũng như xử lý dữ liệu.
* Một nơi duy nhất chịu trách nhiệm cho việc mapping các bảng vào object.
* Tăng tính bảo mật và rõ ràng cho code.
* Rất dễ dàng để thay thế một Repository với một implementation giả cho việc testing, vì vậy bạn không cần chuẩn bị một cơ sở dữ liệu có sẵn.

**Generic Repository Pattern**

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

**Lợi ích của Generic Repository Pattern:**

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

# Triển khai đề tài

## Mô tả đề tài

### Tổng quan:

* Một trang web giải trí dành cho mọi người thư giãn sau những khoảng thời gian căng thẳng.
* Giao diện Website được viết bằng Bootstrap 4, CSS 3, và Razor Pages
* Dùng mô hình Three-Tire và tổ chức dữ liệu:
  + Presentation Layer (tầng 3): tầng này giúp hiển thị giao diện.
  + Business Logic Layer (tầng 2): tầng này chứa các hàm thực thi, giúp xử lý sự kiện xảy ra trên tầng Presentation Layer.
  + Data Access Layer (tầng 1): tầng này chứa hàm kết nối với các hệ quản trị cở sở dữ liệu SQL Server .
* Dùng Pattern MVC và Repository để tổ chức và quản lý dữ liệu.

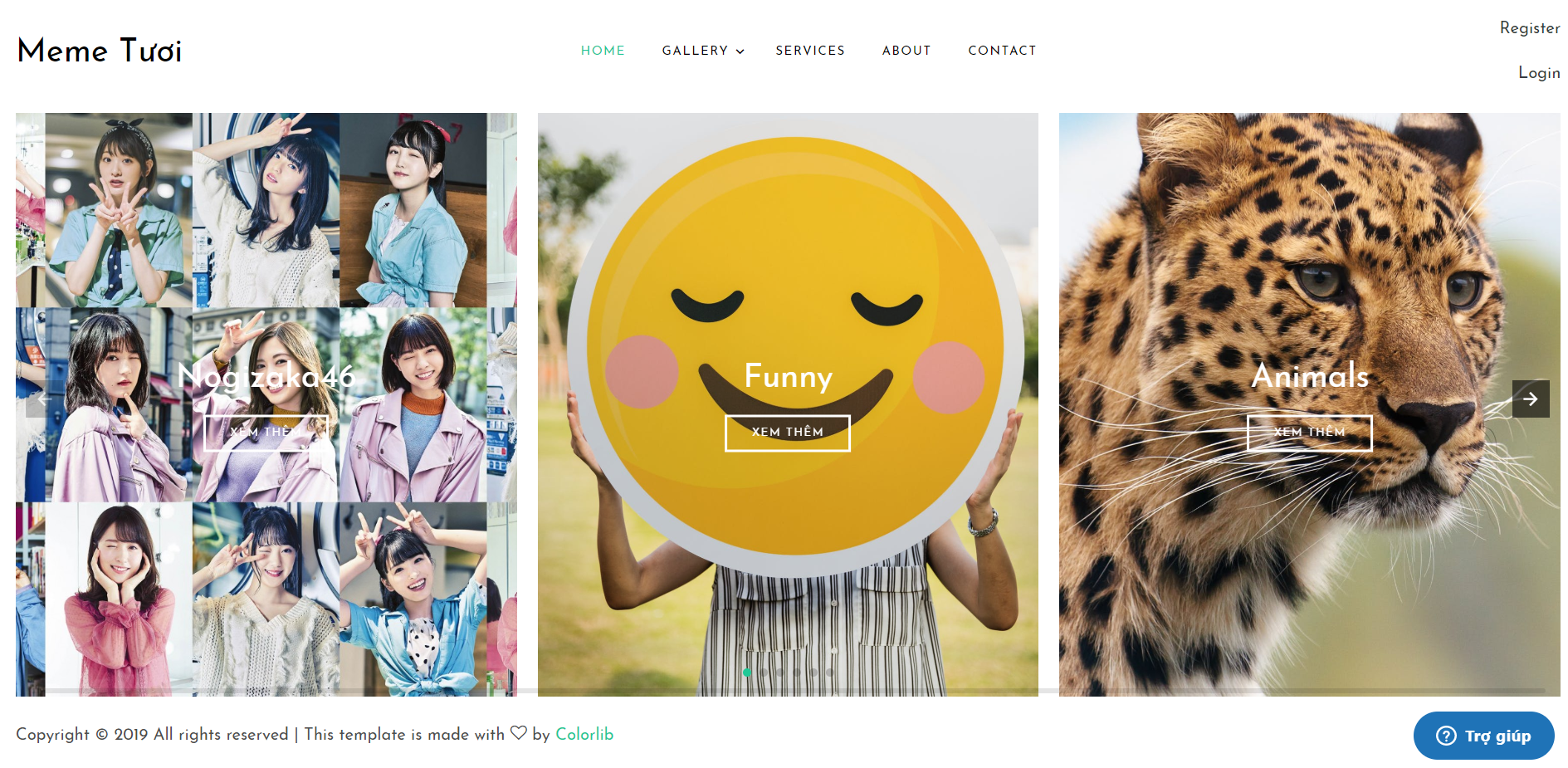
### Chức năng:

* Thành viên:
  + Đăng ký thành viên
  + Đăng meme hoặc video mới
  + Vote bài
  + Comment
* Admin:
  + Quản lý Meme
  + Quản lý Thể loại
  + Quản lý Lượt vote Meme
  + Quản lý User
  + Kiểm duyệt Meme do người dùng đăng lên

## Xây dựng giao diện

### Người dùng

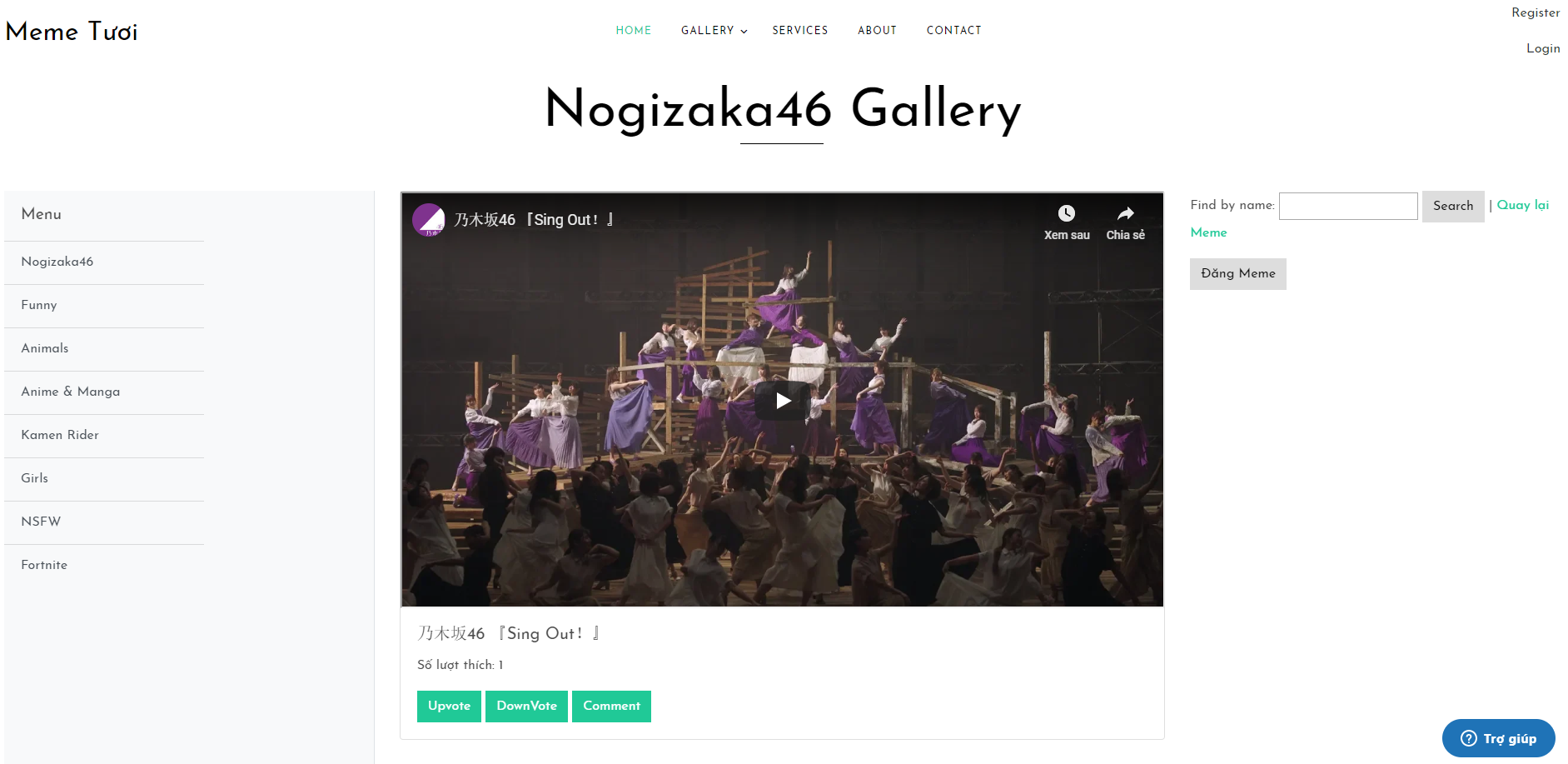
#### **Trang chủ**



Hình ‑ Giao diện Trang chủ

Trang chủ nhằm mục đích giới thiệu về trang web với những thể loại khác nhau. Người dùng bấm vào nút “Xem thêm” để vào thể loại mình muốn. Bên trên có 1 thanh memu, người dùng cũng có thể chọn thể loại mình muốn xem ở tag “Gallery”, những nút như “Services” , “About” , “Contact” dùng để tìm hiểu thêm về trang web . Ngoài ra người dùng còn có thể bấm vào nút “Trợ giúp” để chat với Admin.

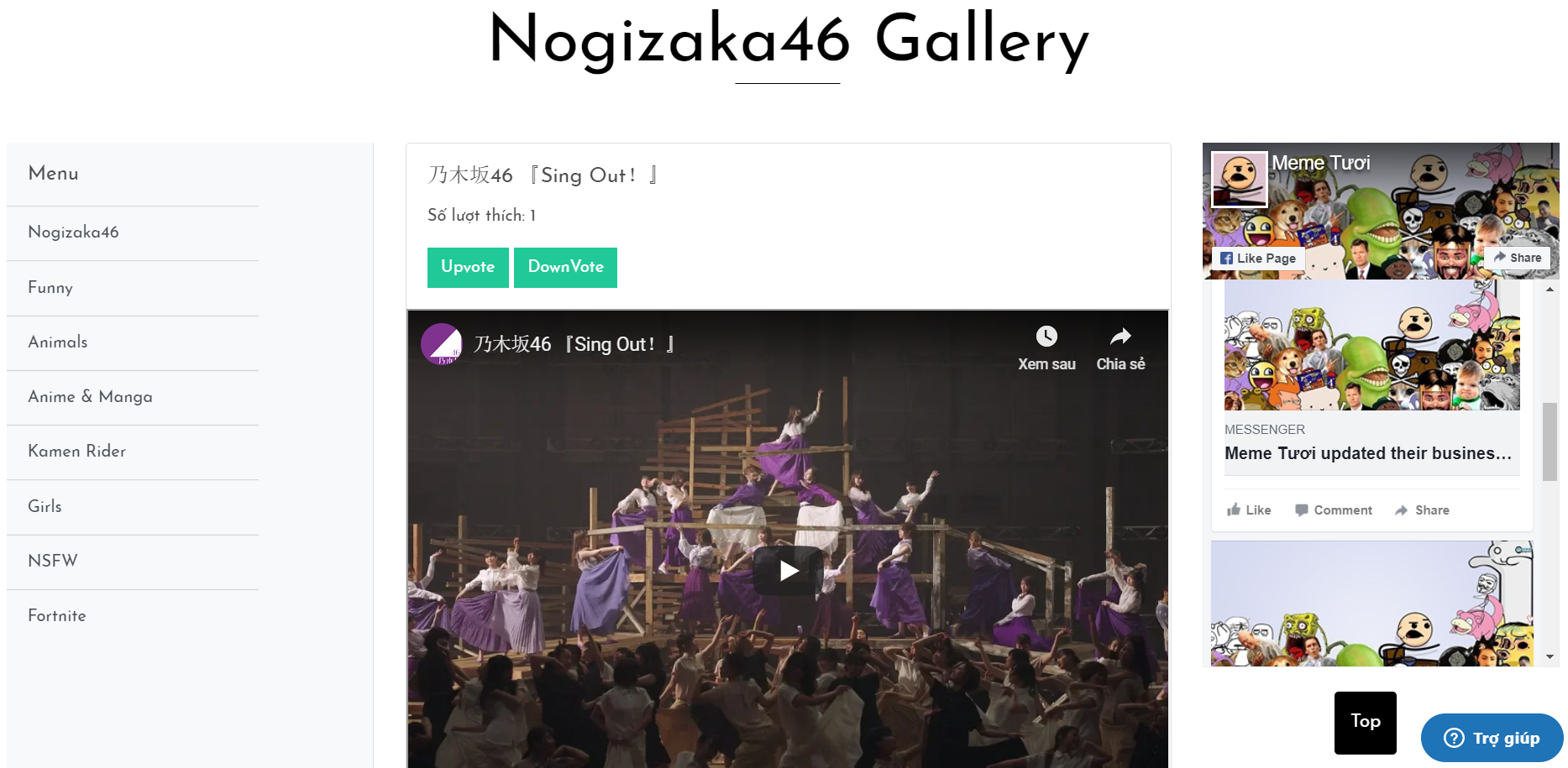
#### **Bên trong từng thể loại**

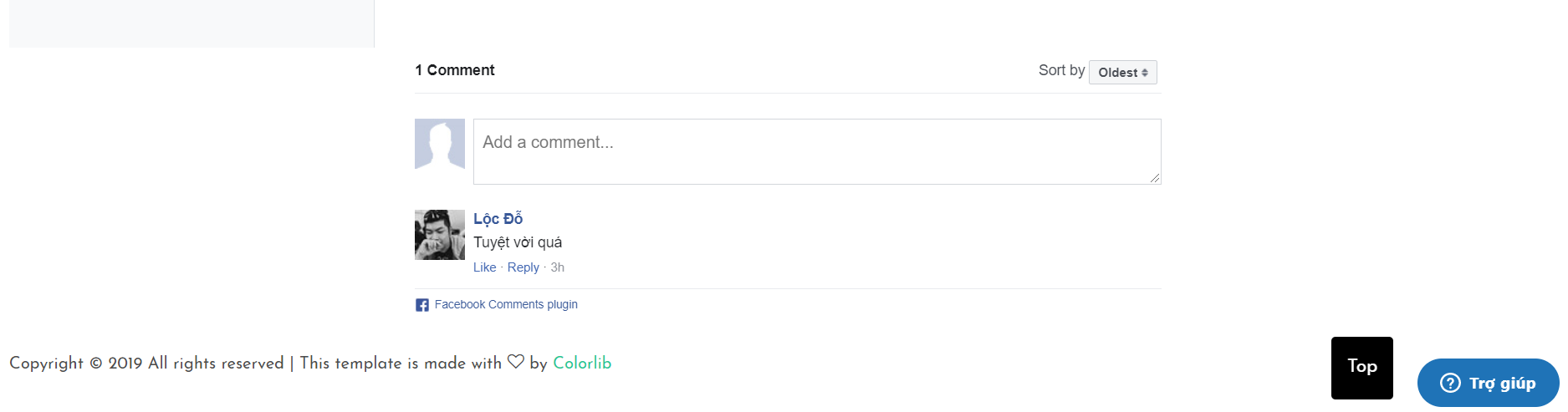


Hình ‑ Giao diện Chỉ tiết thể loại

Ở giao diện Gallery người dùng có thể xem toàn bộ Meme (Bài viết) trong thể loại đó và có thể “Thích” hoặc “Không thích” bài viết đó hoặc vào phần “Comment”.

#### **Chi tiết từng Meme (Bài viết)**



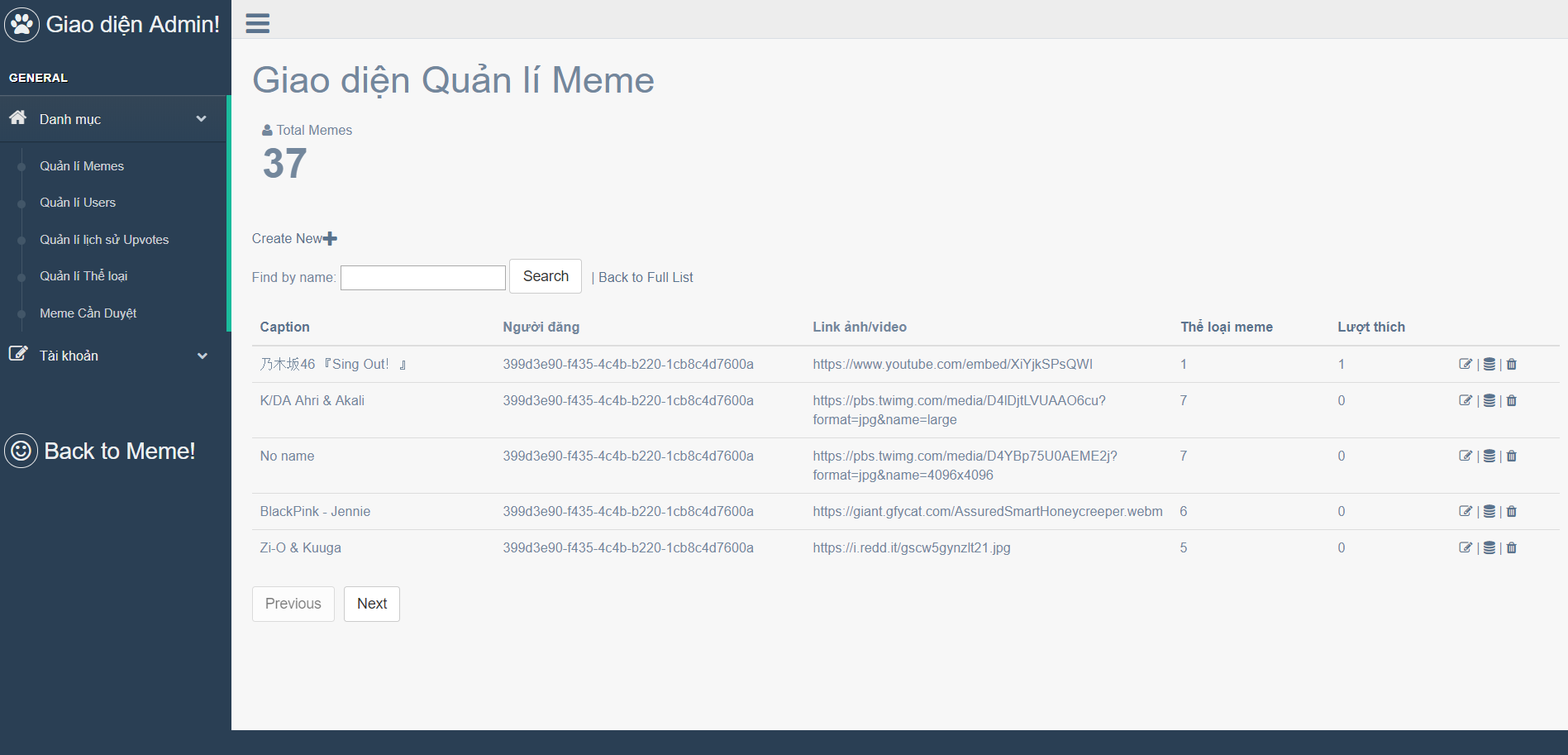


Hình ‑ Giao diện Chi tiết Bài Viết

Ở giao diện này người dùng giờ đã có thể comment cũng như bình chọn thích hoặc không thích bài viết . Người dùng cũng có thể quyết định xem có like page meme tươi hay không .

### Admin

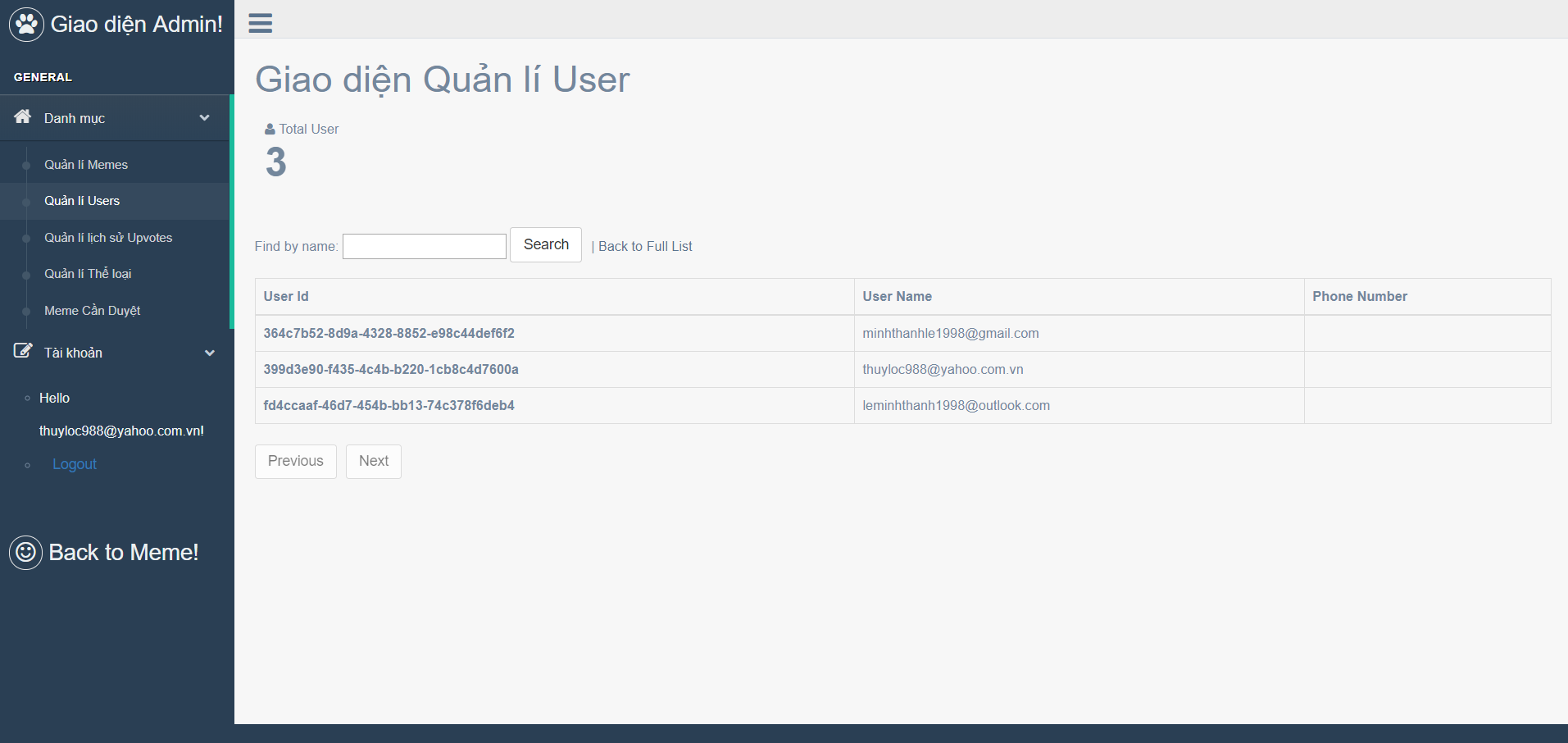
#### **Giao diện quản lý Meme**



Hình ‑ Giao diện Quản lý meme

Ở giao diện này Admin có thể dễ dàng quản lý các meme (bài viết) . Admin có thể Thêm , Xóa , Sửa , Xem chi tiết bất cứ bài viết nào . Ngoài ra , admin còn có thể sắp xếp theo tiêu đề , thể loại , lượt thích và còn có thể tìm kiếm . Ở giao diện này Admin cũng có thể quay lại trang chủ một cách nhanh chóng.

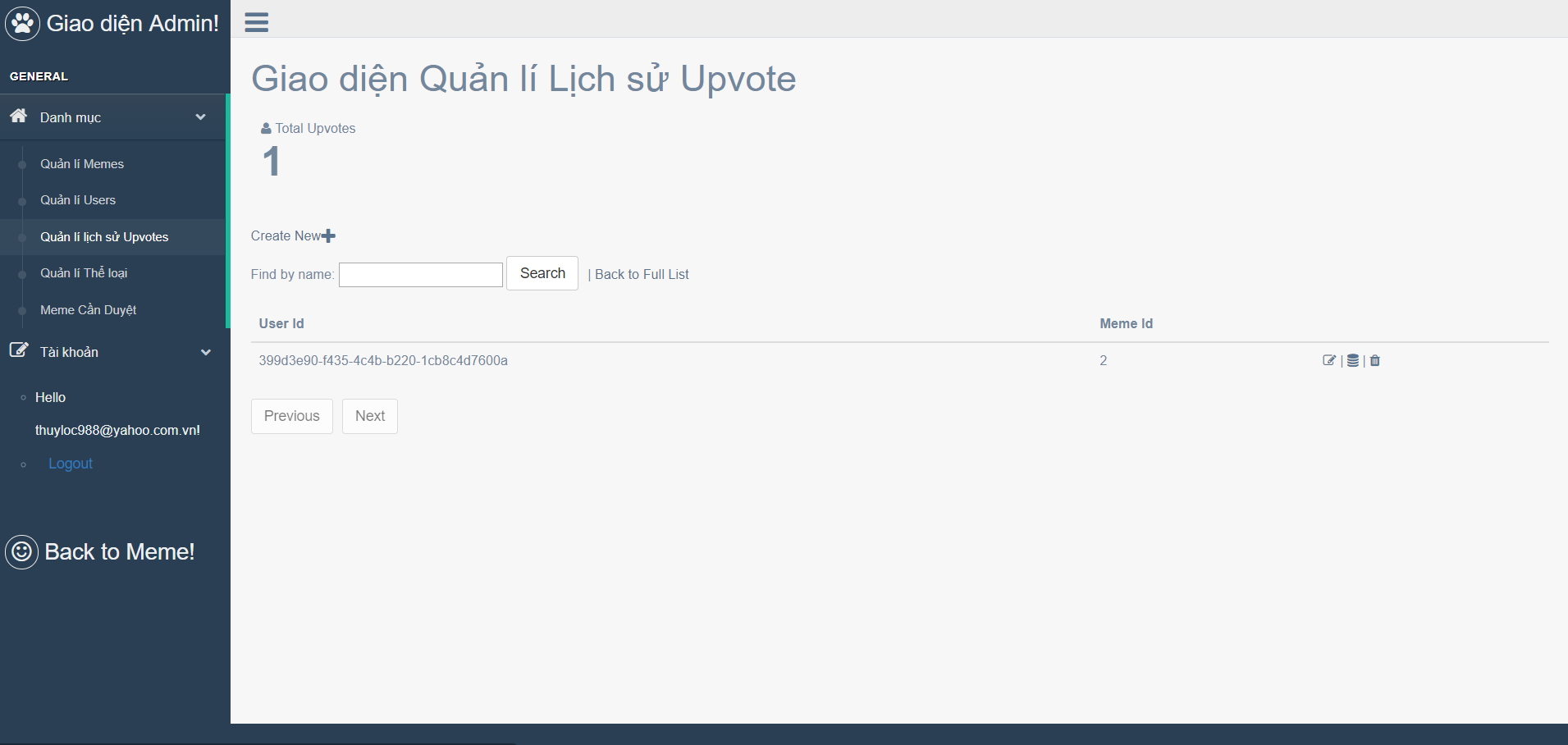
#### **Giao diện quản lý User**



Hình ‑ Giao diện Quản lý User

Ở giao diện này Admin có thể xem được trang web có bao nhiêu thành viên và có thể tìm kiếm thành viên thông qua số điện thoại , mã người dùng và User Name của người dùng đó.

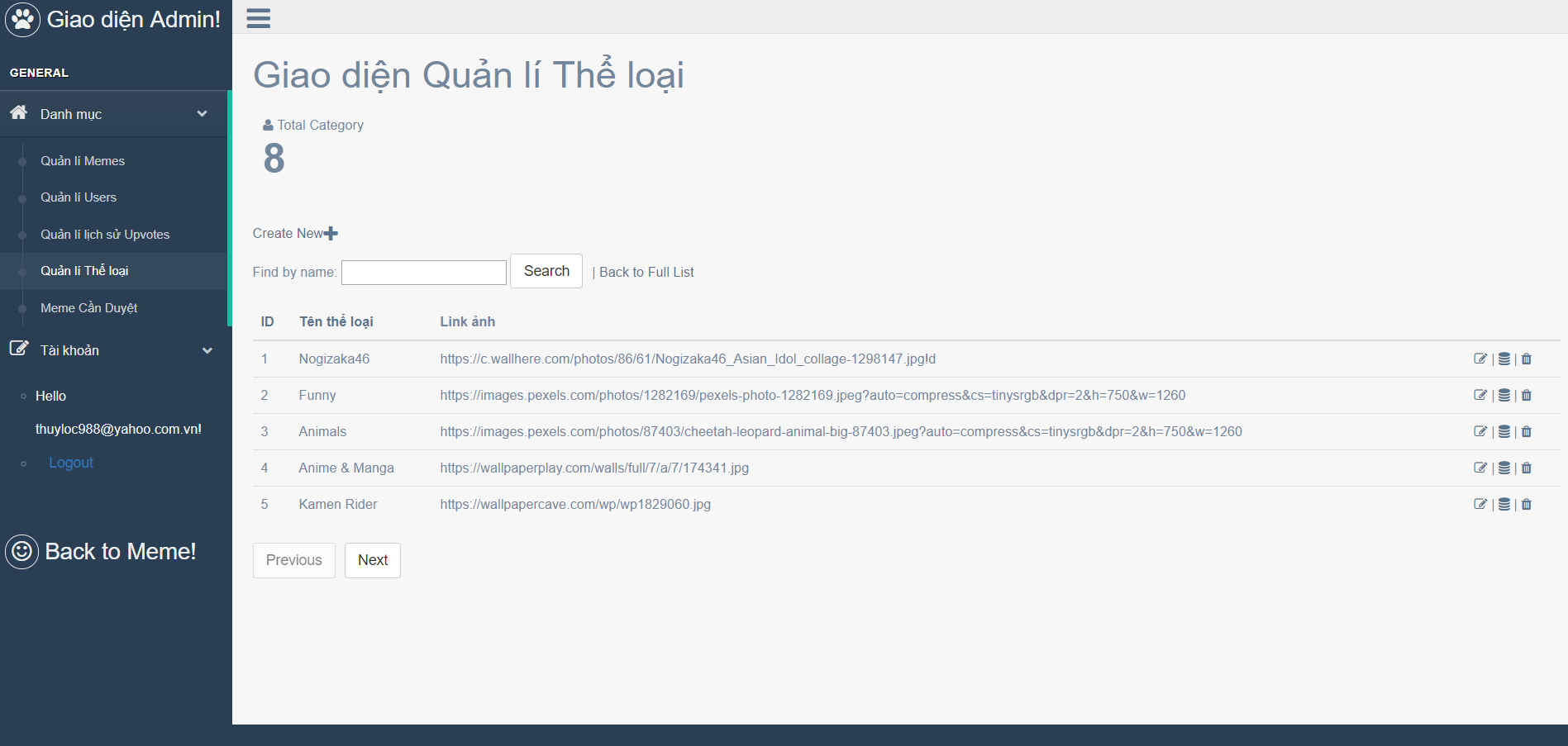
#### **Giao diện quản lý Lượt thích**



Hình ‑ Giao diện Quản lý Upvote

Ở giao diện này admin có thể theo dõi lượt thích của các thành viên . Thêm , xóa , sửa , xem chi tiết lượt thích và tìm kiếm nếu cần thiết.

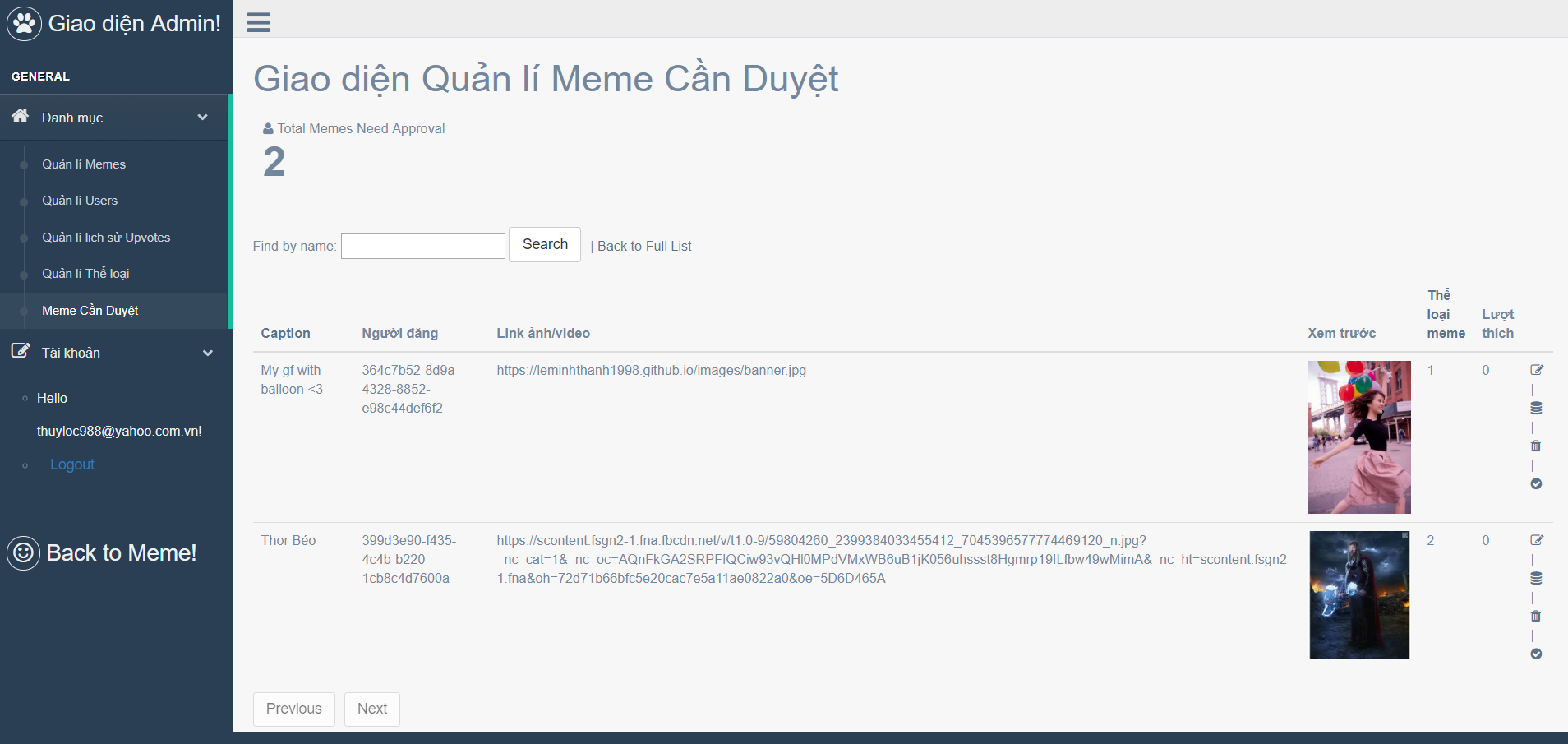
#### **Giao diện Quản lý Thể loại**



Hình ‑ Giao diện Quản lý Thể loại

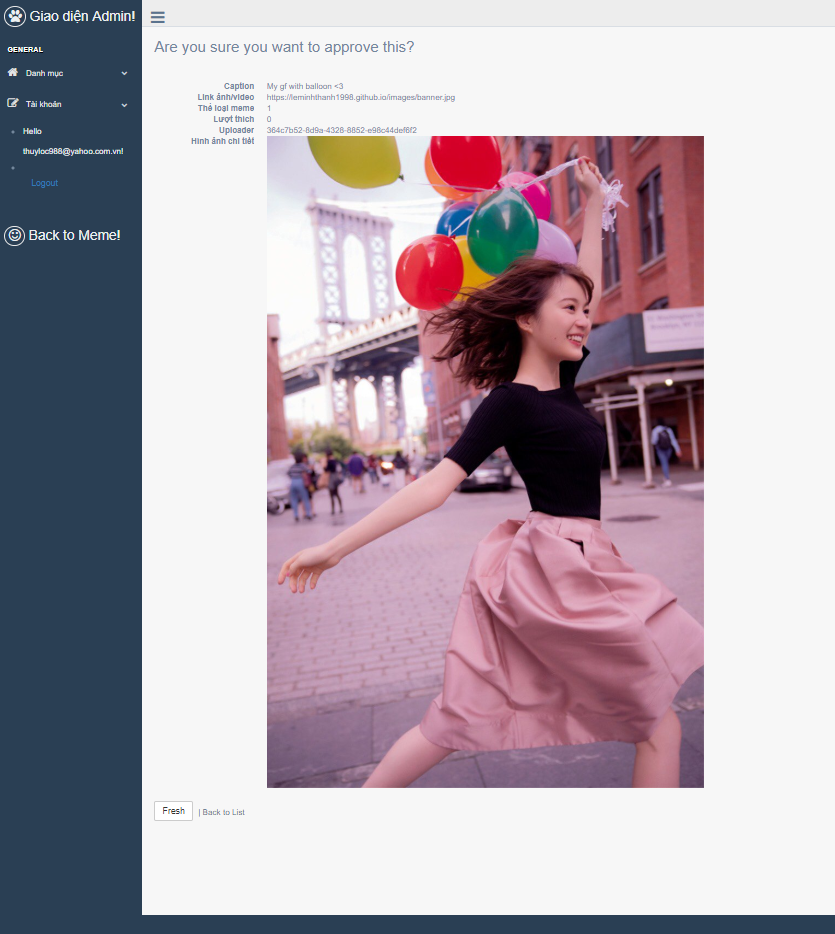
Giao diện này giúp admin có thể thêm , xóa , sửa , xem chi tiết , tìm kiếm , sắp xếp thể loại theo ID và tên thể loại.

#### **Giao diện duyệt Meme (Bài viết)**



Hình ‑ Giao diện duyệt Meme

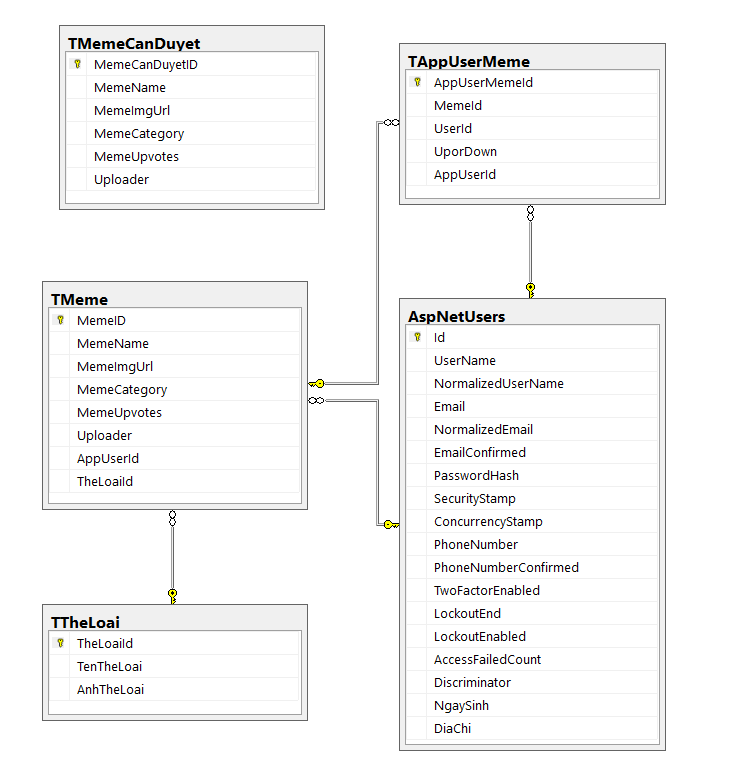
Giao diện này giúp admin có thể tìm kiếm , xem chi tiết meme cần duyệt , chỉnh sửa , chấp nhận hoặc hủy bỏ việc meme này có xuất hiện trên trang chính không .



Hình ‑ Giao diện Xem chi tiết bài cần duyệt

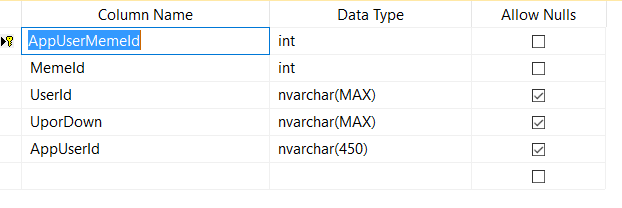
Giúp admin đánh giá chính xác hơn về meme.

## Phân tích thiết kế cơ sở dữ liệu



Hình ‑ Sơ đồ CSDL

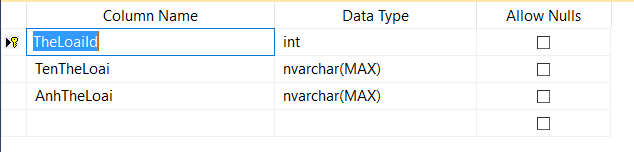
### Bảng Lịch sử Upvote



Bảng này được sinh ra từ mối quan hệ nhiều-nhiều của 2 bảng AppUser và bảng Meme chúng ta có:

* AppUserMemeId : Khóa chính của bảng này
* MemeId : Khóa chính của bảng Meme
* UserId : Khóa chính của bảng AspNetUser
* UporDown : Để kiểm tra người dùng đó thích hay không thích meme nào đó
* AppUserId: Khóa chính của bảng AppUser

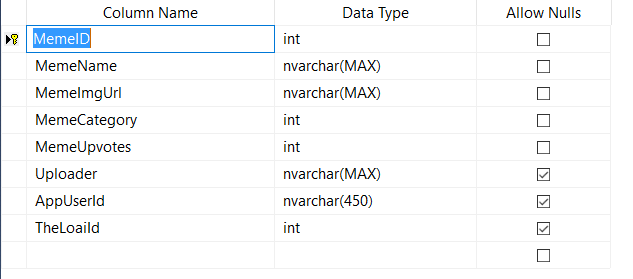
### Bảng Thể loại



Gồm có :

* TheLoaiId : Khóa chính của bảng Thể Loại
* TenTheLoai : Tên của thể loại dùng để hiển thị
* AnhTheLoai : Chứa đường dẫn ảnh của thể loại dùng để hiển thị

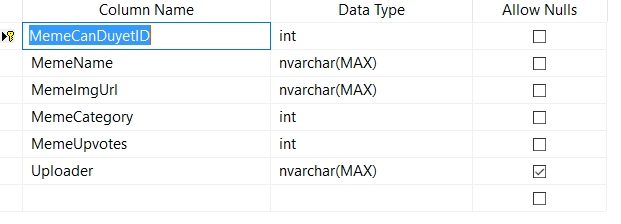
### Bảng Meme (Bài viết)



Gồm có:

* MemeId: Khóa chính của bảng Meme
* MemeName : Tiêu để của Meme
* MemeImgUrl : Đường dẫn hình ảnh của Meme để hiển thị
* MemeCategory : Thể loại của Meme
* MemeUpvote : Lượt thích của Meme
* Uploader : Người đăng Meme này
* AppUserId : Dùng để nối bảng
* TheLoaiId : Dùng để nối bảng

### Bảng Meme cần duyệt



* MemeCanDuyetId : Khóa chính của bảng Meme cần duyệt
* MemeName : Tiêu đề của Meme
* MemeName : Tiêu để của Meme
* MemeImgUrl : Đường dẫn hình ảnh của Meme để hiển thị
* MemeCategory : Thể loại của Meme
* MemeUpvote : Lượt thích của Meme
* Uploader : Người đăng Meme này

# Tổng kết

## Những gì đã đạt được sau khi hoàn thành đồ án

* Biết áp dụng Template vào một project ASP.NET Core
* Hiểu được cách hoạt động của một trang web ( Gồm trang người dùng và Admin)
* Biết cách xây dựng một trang web bằng ASP.NET Core MVC
* Tăng kỹ năng làm việc nhóm
* Học được các framework cũng như pattern mới

## Ưu và nhược điểm của đồ án

* Ưu điểm :
  + Giao diện đẹp và dễ dàng sử dụng
  + Người dùng có thể tạo tài khoản và tham gia bình chọn , đăng bài cũng như bình luận bằng Facebook
  + Có thể chat với admin để nhận hỗ trợ tư vấn
  + Cơ chế phân quyền giúp tăng tính bảo mật cho trang web
  + Trang web Responsive tốt
* Nhược điểm:
  + Chưa có đăng nhập bằng Google , Facebook
  + Chưa có Real-time
  + Chưa tận dụng được hết tính năng của ASP.NET Core
  + Việc phân quyền chưa được tối ưu
  + Phần quản lý tài khoản người dùng chưa thực sự tốt

## Hướng phát triển

* Sử dụng Progressive Web Apps cho trang web
* Thêm nhiều chức năng mới cho trang web như : báo cáo thống kê , tối ưu phân quyền , tối ưu hóa việc đăng bài của người dùng .

# Tài liệu Tham Khảo

x

|  |  |
| --- | --- |
| Tiếng Anh | |
| [1] | G. Sparks. (2000) An Introduction to modelling software systems using the Unified. [Online]. HYPERLINK " http://www.sparxsystems.com.au/," http://www.sparxsystems.com.au/, |
| [2] | D. Lebrknight R. LeMaster, *Object-Oriented Programming & Design”, CSCI 4448*.: University of Colorado, 2002. |
| [3] | K. Scott M. Fowler, *UML Distilled Second Edition – A Brief Guide to the Standard Object Modelling Langguage*.: Addison Wesley Book, 1999. |
| [4] | A. Munk-Madsen, P.A. Nielsen, J. Stage L. Mathiassen, *ObjectOriented Analysis&Design (OOA&D) – Concept, Principles & Methodology*. 2004. |
| Website | |
| [5] | UML Notion Guide. [Online]. HYPERLINK "http://etna.int-evry.fr/COURS/UML/notation/index.html" http://etna.int-evry.fr/COURS/UML/notation/index.html |
| Tiếng Việt | |
| [6] | Đặng Văn Đức, *Phân tích thiết kế hướng đối tượng bằng UML*. HCM, Việt Nam: Nhà xuất bản Giáo dục, 2001. |
| [7] | Dương Anh Đức, *Bài giảng Phân tích thiết kế hướng đối tượng sử dụng UML*. HCM, Việt Nam: Đại học KHTN - Đại học Quốc gia TP. HCM, 2000. |
| [8] | Nguyễn Văn Ba, *Phát triển hệ thống hướng đối tượng với UML 2.0 và C++*. Hà Nội, Việt Nam: Nhà xuất bản Đại học Quốc gia Hà Nội, 2005 |

x