

**UBND TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN CHUYÊN NGÀNH (841476)

**ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ BÀI VIẾT BLOG VỀ
GAME SỬ DỤNG ASP.NET**

SINH VIÊN: VÕ HẢI HÀ

MSSV: 3120410002

GIẢNG VIÊN PHỤ TRÁCH: PGS.TS NGUYỄN TUẤN ĐĂNG

LỚP: DCT1201

NĂM HỌC: 2023-2024

TP. HỒ CHÍ MINH, 6 THÁNG 12 NĂM 2023

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin ngày nay, blog là một thuật ngữ không hề quá xa lạ đối với các người dùng Internet, và việc tiếp cận các trang web blog thường rất phổ biến và không có quá nhiều sự khó khăn. Tuy nhiên trước đây ít ai biết được blog là một trong những thứ tạo nên nền tảng để phát triển Internet đến được hôm nay. Vì vậy, em quyết định chọn web blog làm đề tài chính cho đồ án chuyên ngành của mình. Website có các hầu hết các chức năng cho người dùng thực hiện các thao tác đọc blog và các chức năng cho người quản lí thực hiện việc kiểm soát hệ thống.

Trong quá trình thực hiện đề tài quản lý này nhóm chúng em chân thành cảm ơn thầy Nguyễn Tuấn Đăng đã tận tình hướng dẫn em trong suốt quá trình thực hiện, tạo điều kiện cho em hoàn thành chương trình.

Mục Lục

PHẦN I: TỔNG QUAN ĐỀ TÀI	7
I. Xác định đề tài nghiên cứu:	7
1. Tên đề tài:	7
2. Mục tiêu tổng quát:	7
3. Mục tiêu cụ thể:	7
II. Đối tượng nghiên cứu:	7
III. Phạm vi nghiên cứu, ứng dụng:	7
IV. Đóng góp mới (dự kiến)	7
V. Kế hoạch triển khai	8
1. Tuần 1-2-3 (từ ngày 6/9/2023-19/9/2023): Giai đoạn chuẩn bị	8
2. Tuần 4-5-6 (từ ngày 20/9/2023-10/10/2023): Giai đoạn phân tích thiết kế	8
3. Tuần 7-8-9 (từ ngày 11/10/2023-31/10/2023): Xây dựng dự án	8
4. Tuần 10-11 (từ ngày 1/11/2023-15/11/2023): Kiểm thử và triển khai	8
5. Tuần 12 (từ ngày 16/11/2023-23/11/2023): Viết báo cáo.	8
VI. Nghiên cứu tổng quan	9
1. Lịch sử vấn đề	9
2. Các công trình có liên quan	12
VII. Tài liệu tham khảo	12
PHẦN II: TỔNG QUAN VỀ PHẦN MỀM	13
I. Yêu cầu chung về phần mềm	13
1. Quản lý bài viết	13
2. Quản lý tài khoản người dùng	13
II. Các chức năng của phần mềm	13
1. Quản lý hệ thống: bao gồm các chức năng:	13
2. Quản lý danh mục: bao gồm các chức năng:	13
3. Quản lý bài viết: bao gồm các chức năng:	14
III. Đối tượng người dùng	14
1. Đối với admin	14
2. Đối với người dùng	14
IV. Các ràng buộc về dữ liệu	14
PHẦN III: THIẾT KẾ	15
I. Sơ đồ DFD mức 0:	15

II.	Sơ đồ ERD	15
III.	Sơ đồ Class	16
IV.	Sơ đồ chức năng.....	16
V.	Sơ đồ Sequence.....	17
1.	Đăng nhập.....	17
2.	Đăng kí	17
3.	Tạo blog.....	18
4.	Quản lý bài viết	18
5.	Quản lý thể loại	20
6.	Quản lý tags.....	22
7.	Quản lý tài khoản	24
VI.	Đặc tả yêu cầu.....	25
1.	Biểu đồ usecase	25
2.	Đặc tả Usecase.....	28
VII.	Bản vẽ giao diện	38
1.	Header	38
2.	Footer.....	38
3.	Đăng nhập.....	39
4.	Đăng kí	39
5.	Giao diện chính	40
6.	Danh sách các blog.....	40
7.	Trang blog	41
8.	Quản lý Tags.....	42
9.	Quản lý thể loại bài viết.	42
10.	Viết Blog.....	43
11.	Quản lý bài viết	43
VIII.	Cơ sở dữ liệu	44
1.	Các bảng (table).....	44
2.	Thực thể và thuộc tính.....	44
3.	Sơ đồ ER.....	46
PHẦN IV:	GIAO DIỆN SẢN PHẨM SAU LẬP TRÌNH	47
I.	Giao diện trang chủ.	47
II.	Giao diện đăng nhập và đăng kí tài khoản.....	48
III.	Danh mục sản phẩm tìm theo thể loại	49

IV.	Trang blog và phần bình luận dưới trang blog	49
V.	Giao diện chính trang admin.....	50
VI.	Giao diện phần viết blog cho admin.....	51
VII.	Giao diện quản lý các bài blog	51
VIII.	Giao diện quản lý các tài khoản	52
IX.	Giao diện quản lý các thể loại.....	52
X.	Giao diện quản lý các tags của bài viết	53
PHẦN V: SOURCE CODE		54
I.	Properties.....	54
1.	lauchSetting.json	54
2.	serviceDependencies.json.....	54
II.	Program.cs	54
III.	Appsettings.json.....	55
IV.	Data (DbContext).....	55
V.	Migrations.....	56
1.	InitialCreate (First migration)	56
2.	DbContextModelSnapshot	59
VI.	CSS	70
1.	Style.css (Giao diện chung).....	70
2.	Main.css (Giao diện admin)	132
3.	Các CSS khác	155
VII.	Javascript	156
1.	Main.js.....	156
2.	Các file JS khác	157
VIII.	Model.....	157
1.	Account.cs	157
2.	Blog.cs	158
3.	Category.cs	158
4.	Tag.cs.....	159
5.	Comment.cs	159
6.	Like.cs	160
IX.	Controller	160
1.	AAccountsController.cs (phía admin).....	160
2.	ABlogsController.cs (phía admin).....	162

3.	ACategoriesController.cs (phía admin).....	166
4.	ATagsController.cs (phía admin).....	169
5.	CommentsController.cs	171
6.	UBlogsController.cs (phía người dùng)	173
7.	UCategoriesController.cs (phía người dùng)	175
X.	Views	175
1.	Shared.....	175
2.	AAccounts (phía admin)	182
3.	ABlogs (phía admin)	186
4.	ACategories (phía admin)	194
5.	ATags (phía admin)	197
6.	UBlogs (phía người dùng).....	201
7.	UCategories (phía người dùng, chỉ có Index.cshtml)	207
8.	ULogin (phía người dùng, chỉ có Index.cshtml)	208
9.	USignup (phía người dùng, chỉ có index.cshtml)	209

PHẦN I: TỔNG QUAN ĐỀ TÀI

I. Xác định đề tài nghiên cứu:

1. Tên đề tài:

- Xây dựng website quản lý bài viết blog về game Minecraft sử dụng ASP.NET

2. Mục tiêu tổng quát:

- Phát triển một trang web blog cho phép người dùng truy cập để đọc các bài viết, đồng thời cho phép admin website đăng tải và quản lý các bài viết nhằm chia sẻ các thông tin về game Minecraft.

3. Mục tiêu cụ thể:

- Cho phép admin viết, đăng tải, chỉnh sửa, xoá, phân loại danh mục bài viết.
- Cho phép người dùng đăng kí, đăng nhập, tìm kiếm bài viết dựa theo từ khoá, danh mục và tiêu đề. Ngoài ra còn cho phép họ thích và bình luận về bài viết.

II. Đối tượng nghiên cứu:

- Đối tượng chính: C#, ASP.NET Core
- Cơ sở dữ liệu: SQL Server
- Mô hình thiết kế: Mô hình MVC

III. Phạm vi nghiên cứu, ứng dụng:

- Môi trường ứng dụng: Trình duyệt web.
- Phạm vi người dùng: toàn bộ người dùng Internet có đam mê về game.
- Ngôn ngữ lập trình: C#.

IV. Đóng góp mới (dự kiến)

V. Kế hoạch triển khai

1. Tuần 1-2-3 (từ ngày 6/9/2023-19/9/2023): Giai đoạn chuẩn bị
 - Xác định đề tài nghiên cứu và lập kế hoạch phát triển.
 - Nghiên cứu sơ lược về đề tài cần xác định.
2. Tuần 4-5-6 (từ ngày 20/9/2023-10/10/2023): Giai đoạn phân tích thiết kế
 - Xác định yêu cầu đề tài về chức năng và phi chức năng.
 - Thiết kế cơ sở dữ liệu, giao diện trang web.
3. Tuần 7-8-9 (từ ngày 11/10/2023-31/10/2023): Xây dựng dự án
 - Lập trình
 - Cài đặt cơ sở dữ liệu
4. Tuần 10-11 (từ ngày 1/11/2023-15/11/2023): Kiểm thử và triển khai
 - Kiểm tra & sửa lỗi
 - Triển khai ứng dụng lên hệ thống
5. Tuần 12 (từ ngày 16/11/2023-23/11/2023): Viết báo cáo.

VI. Nghiên cứu tổng quan

1. Lịch sử vấn đề

a. Sơ lược về Blog:

- Blog là phiên bản có tiền thân từ "weblog," là thuật ngữ thường dùng để mô tả các trang web cung cấp thông tin liên tục dưới dạng các bài viết. Các blog thường có một hoặc rất nhiều tác giả đồng viết.
- Thông thường các trang blog được đăng tải dưới dạng nhật kí hoặc được hiển thị theo trình tự thời gian ngược lại (bài viết mới nhất được xếp trên cùng).
- Một blog thường bao gồm các nội dung liên quan cá nhân hoặc một tập thể, có thể liên quan đến chính trị hay cơ bản chỉ tập trung về một vấn đề, một chủ đề nào đó, và luôn có ý kiến cá nhân của tác giả về mục tiêu của bài viết. Ngoài ra các bài blog cũng nhận được luồng bình luận từ người đọc nhằm tăng sự đa chiều, cũng như cải thiện chất lượng bài viết sau này nếu bình luận có tính đóng góp. [1]

b. Lịch sử phát triển của blog:

- 1994 - 1997: Giai đoạn khởi đầu:

Trong giai đoạn này, blogging xuất hiện lần đầu và vẫn còn tranh cãi về khi nào chính xác là thời điểm bắt đầu của blogging.

Justin Hall tạo ra trang web links.net vào tháng 01/1994, được coi là một trong những blogger đầu tiên.

Thuật ngữ "weblog" xuất hiện vào tháng 12/1996 bởi Jorn Barger.

Nhiều nền tảng blogging xuất hiện từ năm 1997 trở đi.

- 1998 - 2001: Sự xuất hiện của các tài nguyên hỗ trợ cho blogger:

Giai đoạn này chứng kiến sự tăng lên nhanh chóng của các nền tảng blogging và tài nguyên hỗ trợ blogger.

Open Diary, Blogger, LiveJournal, và Xanga là những nền tảng nổi bật trong giai đoạn này.

Blogging bắt đầu trở thành một cơ hội kiếm tiền và các blogger được coi trọng hơn.

- 2002: Một năm "đại phát" của blogging:

Technorati, công cụ tìm kiếm trên blog, ra đời vào tháng 02/2002.

Heather B. Armstrong bị sa thải vì đăng bài viết chỉ trích đồng nghiệp trên blog cá nhân.

Sự xuất hiện của Mommy Bloggers với các bài viết về việc nuôi dạy con.

Google ra mắt AdSense, cho phép blogger kiếm tiền từ quảng cáo trên blog của họ.

- 2003: Đà tăng trưởng được giữ vững:

Nền tảng như TypePad và WordPress ra đời.

Live blogging và live text trở nên phổ biến.

Google mua lại Pyra Labs, công ty phát triển Blogger.

- 2004 - 2005: Video và thông cáo báo chí:

Sự bùng nổ của blog video vào năm 2004.

YouTube xuất hiện và trở thành một nền tảng chia sẻ video lớn nhất.

Sự xuất hiện của tờ Huffington Post và BuzzFeed, mở đường cho mối giao thoa giữa blogging và báo chí.

- 2006 - 2007: Sự bùng nổ của tiểu blog (microblog) và các quy tắc:

Sự ra đời của microblogging với Twitter và Tumblr.

Sự phát triển của quy tắc và quyền lợi của các blogger trong việc quản lý bình luận và tương tác.

- 2008 - 2011: "Những năm tháng trầm lắng" của blog:

Giai đoạn này chứng kiến ít sự kiện quan trọng thúc đẩy blogging.

LinkedIn giới thiệu chương trình Influencers cho guest blog.

Blogging trở nên chuyên nghiệp hơn với một số blogger kiếm tiền từ nó.

- 2012: Medium ra đời:

Evan Williams, đồng sáng lập Blogger, tạo ra Medium.

Medium giới thiệu khái niệm Decentralized Content, cho phép chia sẻ nội dung gốc trên nền tảng khác.

LinkedIn mở chương trình Influencers cho mọi người dùng.

- Ý nghĩa: Có thể thấy Blogging đã phát triển từ các trang web đơn giản với văn bản và hình ảnh đến các nền tảng phức tạp hơn với tích hợp video, âm thanh, và nhiều tính năng tương tác. Công nghệ đã giúp nâng cao trải nghiệm của người đọc và người viết blog. Từ đó nó gây ảnh hưởng lớn đến cách thông tin được truyền tải và chia sẻ trực tuyến. Nó đã tạo ra một nền tảng cho người dùng bình thường và các chuyên gia chia sẻ kiến thức và quan điểm của họ về mọi chủ đề, từ công nghệ đến âm nhạc, sức khỏe, và nhiều lĩnh vực khác. Ngoài ra Blogging đã trở thành một nguồn thu nhập hấp dẫn, cho phép người viết blog kiếm tiền thông qua quảng cáo, tiếp thị liên kết, và sản phẩm và dịch vụ của họ. Điều này đã thúc đẩy sự phát triển của ngành công nghiệp liên quan đến sáng tạo nội dung trực tuyến. Không những thế, Blogging đã tạo ra các cộng đồng trực tuyến nơi mọi người có thể trao đổi ý kiến, thảo luận, và kết nối với nhau. Nó đã ảnh hưởng đến cách con người giao tiếp và xây dựng mối quan hệ trực tuyến. Tóm lại, điều này đã thể hiện sự thay đổi liên tục trong cách con người sáng tạo và tương tác trực tuyến, mặc dù có những thách thức và biến đổi, blogging vẫn tiếp tục là một phương tiện quan trọng để chia sẻ thông tin và ý kiến trên internet. [2]

c. Sơ lược về tựa game Minecraft:

- Minecraft là tựa game do Mojang phát hành vào 2011. Nó tập trung chủ yếu vào các khối (blocks) và các sinh vật trong game.
- Tựa game không hề có điểm đích cuối cùng, mà người chơi tự đặt ra cho mình cách chơi nó. Họ có thể cố gắng sinh tồn trong thế giới của game hoặc có thể thỏa sức sáng tạo bằng cách xây dựng thế giới của mình bằng các khối.
- Minecraft có giới hạn độ tuổi từ 7 trở lên và không có độ bạo lực cao, ngoài ra trò chơi có mặt ở đa dạng các nền tảng, nên việc tiếp cận với nhiều người chơi trên toàn thế giới là điều không đáng ngạc nhiên. [3]

d. Sơ lược về ASP.NET:

- ASP.NET là một mã nguồn mở do Microsoft tạo ra vào đầu những năm 2000 với mục tiêu giúp các nhà phát triển tạo các ứng dụng web, dịch vụ web và các trang web động.
- ASP.NET sử dụng nhiều ngôn ngữ lập trình khác nhau, trong đó C# và VB.net được sử dụng phổ biến nhất.

- ASP.NET được xem là một mô hình đáng để sử dụng bởi nó không yêu cầu máy tính có cấu hình cao do xây dựng trên môi trường Windows, các ứng dụng của ASP.NET được chuyển đổi thành các mã nguồn thực thi trong một quãng thời gian nhỏ. Ngoài ra nó cũng miễn phí và yêu cầu chi phí rất nhỏ đối với doanh nghiệp lớn. [4]

2. Các công trình có liên quan

- Planetminecraft.net
- Minecraftforum.net
- The-minecraft-blog.com

VII. Tài liệu tham khảo

- [1] Wordpress.org, “Introduction to Blogging,” 24 8 2023. [Trực tuyến]. Available: <https://wordpress.org/documentation/article/introduction-to-blogging/>. [Đã truy cập 19 9 2023].
- [2] A. Nguyen, “Tất tần tât về lịch sử phát triển của Blog,” Quản Trị Mạng, 15 9 2016. [Trực tuyến]. Available: <https://quantrimang.com/cong-nghe/tat-tan-tat-ve-lich-su-phat-trien-cua-blog-126260>. [Đã truy cập 20 9 2023].
- [3] P. Landin, “WHAT IS MINECRAFT?,” Mojang, 6 6 2023. [Trực tuyến]. Available: <https://www.minecraft.net/en-us/article/what-minecraft>. [Đã truy cập 9 19 2023].
- [4] N. Hưng, “ASP.NET là gì? So sánh giữa lập trình ASP.NET và PHP,” CÔNG TY CỔ PHẦN GIẢI PHÁP VÀ CÔNG NGHỆ VIETNIX, 6 7 2022. [Trực tuyến]. Available: <https://vietnix.vn/aspnet-la-gi/>. [Đã truy cập 20 9 2023].

PHẦN II: TỔNG QUAN VỀ PHẦN MỀM

I. Yêu cầu chung về phần mềm

Hệ thống phần mềm bao gồm: quản lý bài viết và quản lý tài khoản người dùng.

1. Quản lý bài viết

Admin viết bài blog bao gồm tiêu đề và nội dung, chọn loại bài viết và thêm tags (nếu có). Sau đó tiến hành đăng bài.

Admin có thể chỉnh sửa, xoá, và tìm kiếm các bài đã viết

2. Quản lý tài khoản người dùng

Admin có thể chỉnh sửa, xoá và tìm kiếm các tài khoản đã đăng kí

II. Các chức năng của phần mềm

1. Quản lý hệ thống: bao gồm các chức năng:

a. Quản lý thông tin

Thông tin tài khoản: Cho phép quản lý các thông tin về tài khoản và mật khẩu, đồng thời cung cấp chức năng đổi mật khẩu cho người dùng

Thông tin cá nhân: Cho phép quản lý các thông tin cơ bản về người dùng (gồm họ tên, giới tính,...)

b. Quản lý tài khoản

Hiển thị danh sách các tài khoản của người dùng để admin có thể xem thông tin của từng tài khoản. Ngoài ra cung cấp cho admin khả năng tìm, sửa và xoá tài khoản.

2. Quản lý danh mục: bao gồm các chức năng:

a. Quản lý thể loại bài viết

Hiển thị danh sách các thể loại bài viết, admin có thể tìm, sửa và xoá các danh mục khỏi danh sách

b. Quản lý tags

Hiển thị danh sách các tags bài viết, admin có thể tìm, sửa và xoá các danh mục khỏi danh sách

3. Quản lý bài viết: bao gồm các chức năng:

a. Quản lý bài viết

Hiện thị danh sách các bài viết, admin có thể tìm, sửa và xoá các danh mục khỏi danh sách

b. Đăng bài

Admin nhập tiêu đề, nội dung bài viết, chọn tags, thể loại và ấn đăng bài.

III. Đối tượng người dùng

1. Đối với admin

- Cho phép quản lý thông tin tài khoản, các danh mục, tags, bài viết
- Cho phép viết và đăng bài viết lên trang

2. Đối với người dùng

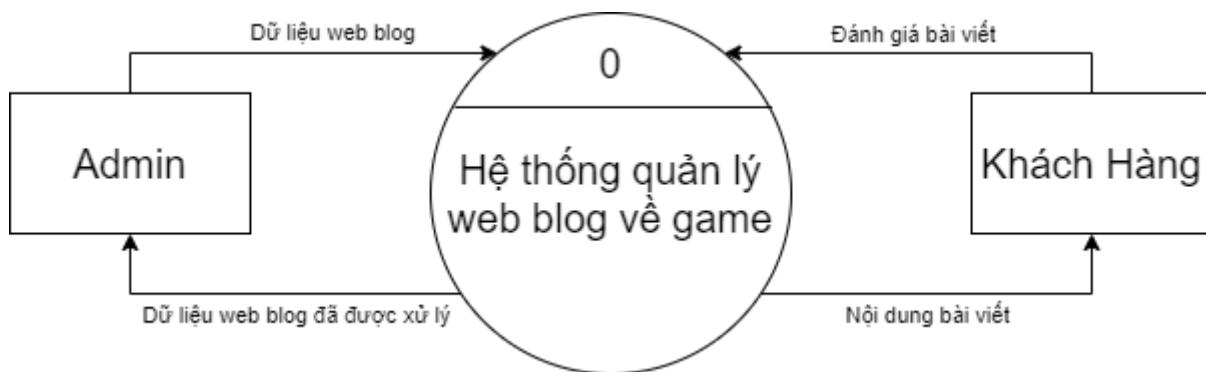
- Cho phép tạo tài khoản và quản lý các thông tin cá nhân
- Cho phép tìm và xem các bài viết được viết bởi admin thông qua từ khoá tìm kiếm, tags và danh mục, đồng thời cho phép để lại bình luận hoặc thích bài đăng.

IV. Các ràng buộc về dữ liệu

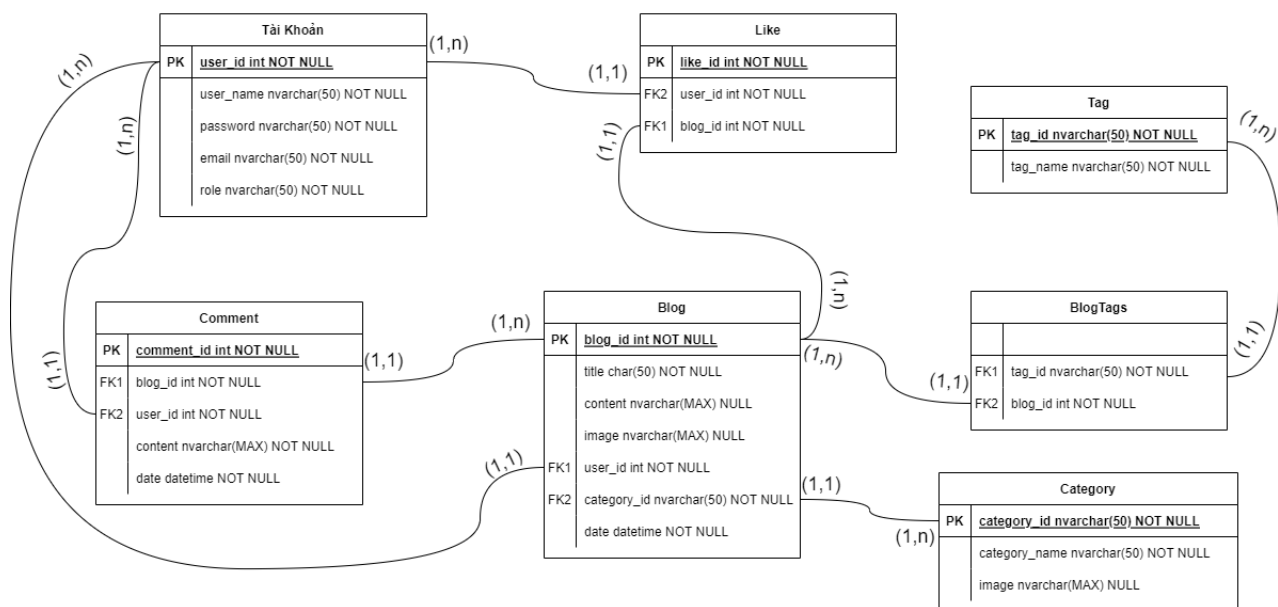
Ràng buộc về tài khoản: khi người dùng bình luận hoặc thích bài đăng, cần phải đăng nhập.

PHẦN III: THIẾT KẾ

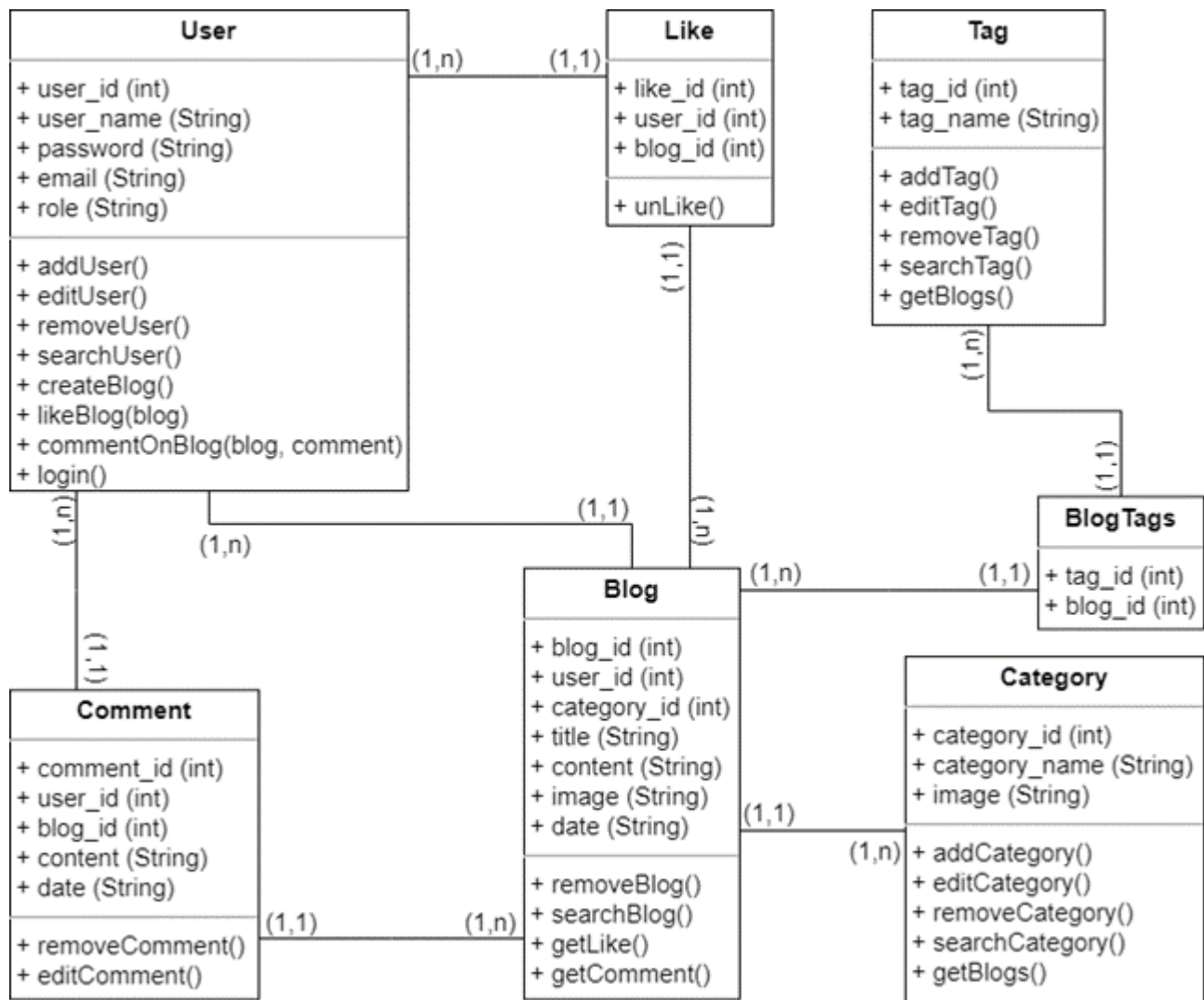
I. Sơ đồ DFD mức 0:



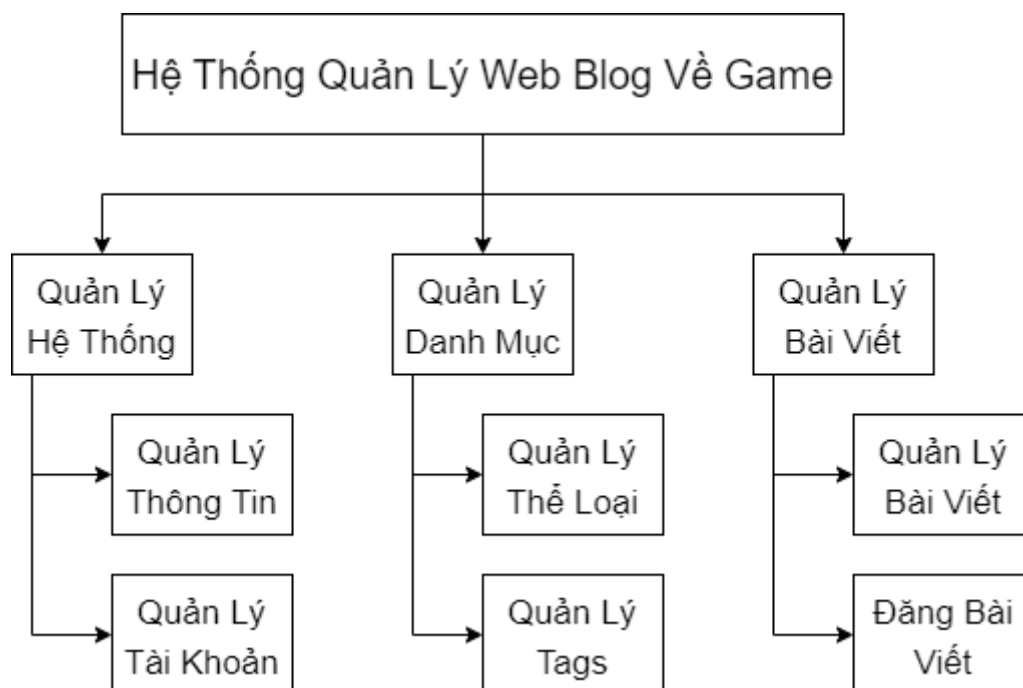
II. Sơ đồ ERD



III. Sơ đồ Class

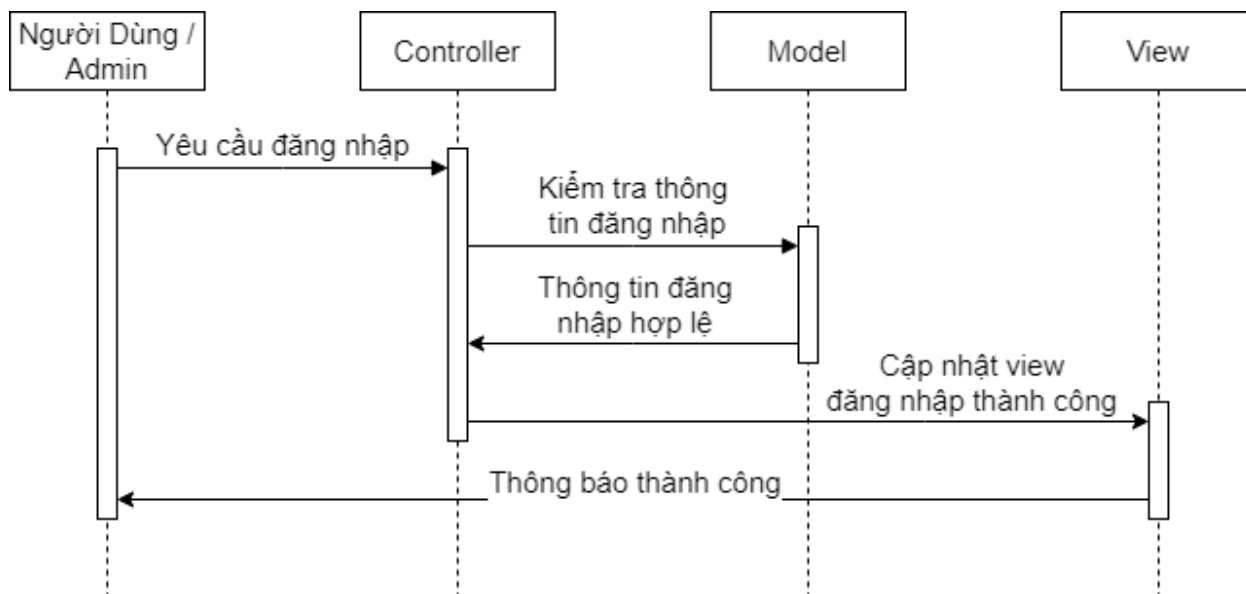


IV. Sơ đồ chức năng



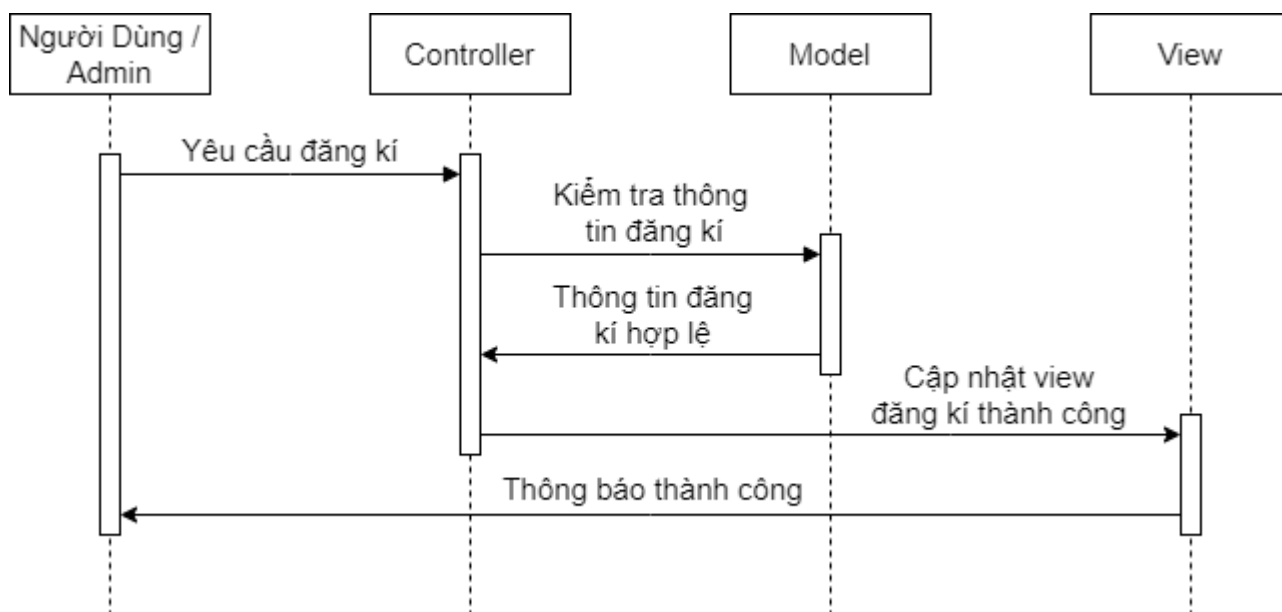
V. Sơ đồ Sequence

1. Đăng nhập



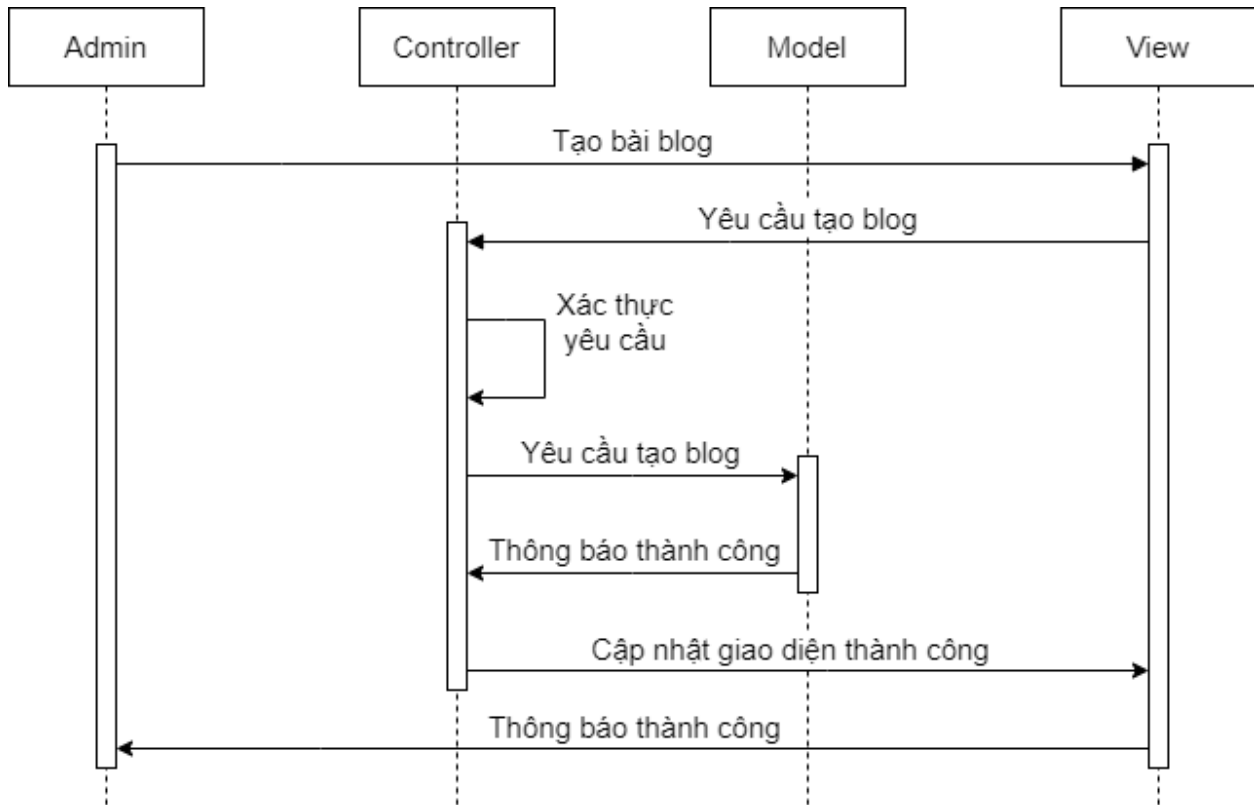
Khi truy cập thành công, người dùng sẽ gửi yêu cầu đăng nhập vào website, sau khi nhập đầy đủ thông tin, controller sẽ gửi yêu cầu kiểm tra thông tin đăng nhập cho model. Cập nhật view thất bại và thông báo không thành công nếu thông tin đăng nhập không hợp lệ và ngược lại.

2. Đăng kí



Khi truy cập thành công, người dùng sẽ gửi yêu cầu đăng kí vào website, sau khi nhập đầy đủ thông tin, controller sẽ gửi yêu cầu kiểm tra thông tin đăng kí cho model. Cập nhật view thất bại và thông báo không thành công nếu thông tin đăng kí không hợp lệ và ngược lại.

3. Tạo blog

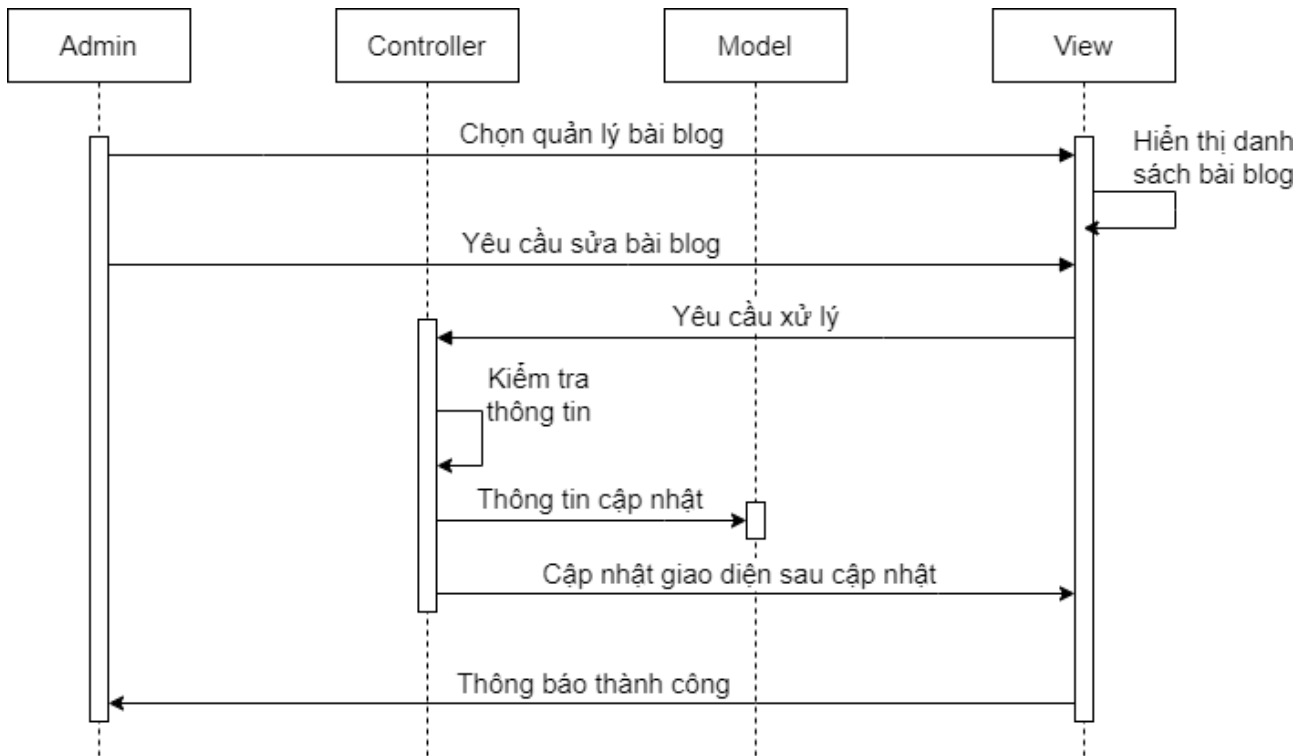


Sau khi viết xong blog, view sẽ gửi yêu cầu tạo blog cho controller và nó sẽ xác thực tài khoản cũng như nội dung bài blog, sau đó gửi yêu cầu tạo blog cho model và cập nhật giao diện tạo blog thành công. Giao diện sẽ cập nhật thất bại nếu quá trình xác thực yêu cầu thất bại.

4. Quản lý bài viết

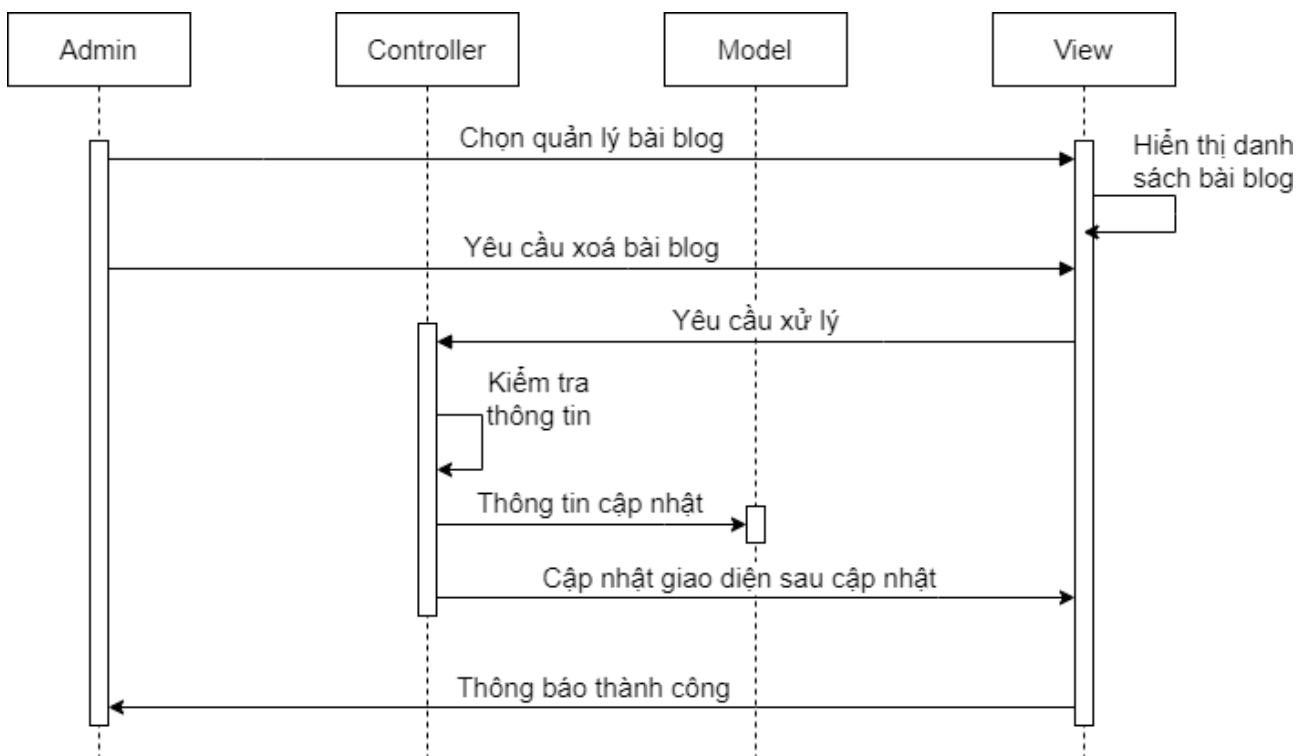
Sau khi yêu cầu quản lý blog, view sẽ hiển thị danh sách các blog đã được viết từ cơ sở dữ liệu. Tùy vào yêu cầu người dùng, hệ thống sẽ thực hiện các công việc dưới đây

a. Sửa bài viết



Khi người dùng yêu cầu sửa bài blog, view sẽ gửi nội dung chỉnh sửa cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

b. Xóa bài viết

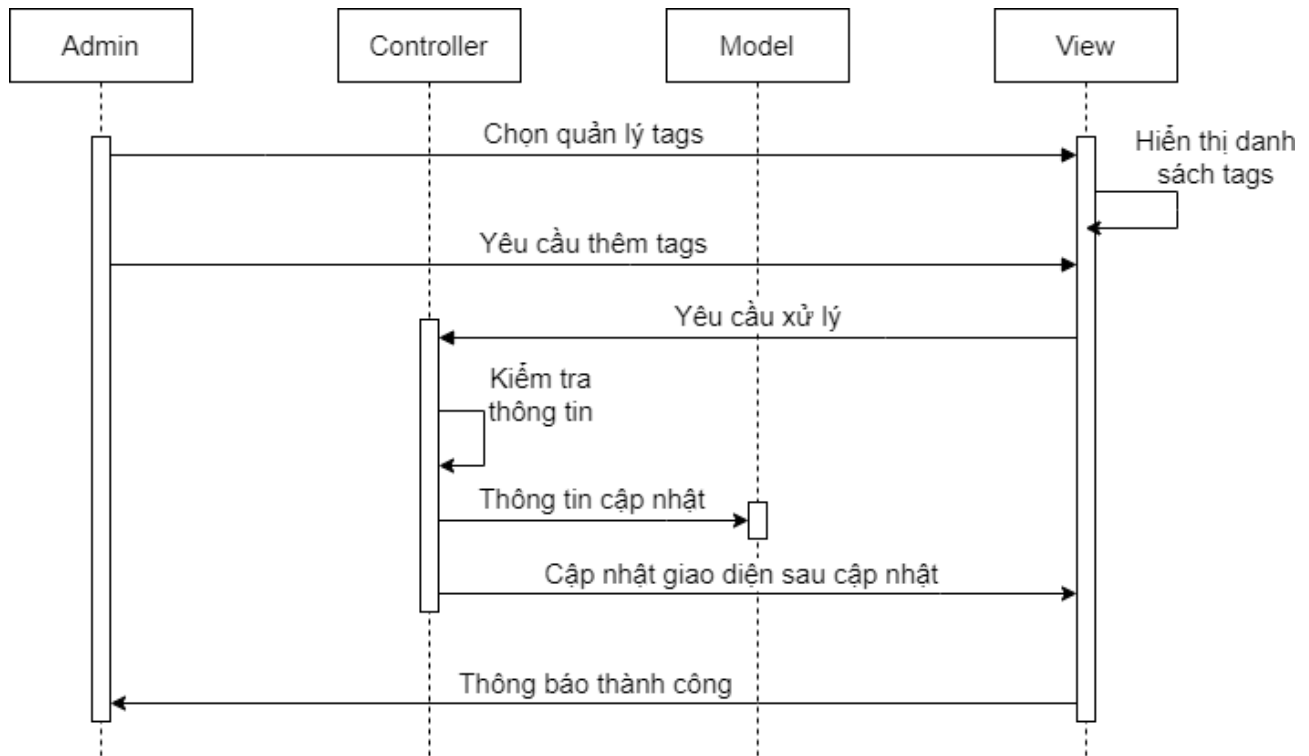


Khi người dùng yêu cầu xóa bài blog, view sẽ gửi yêu cầu xóa bài viết cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

5. Quản lý thẻ loại

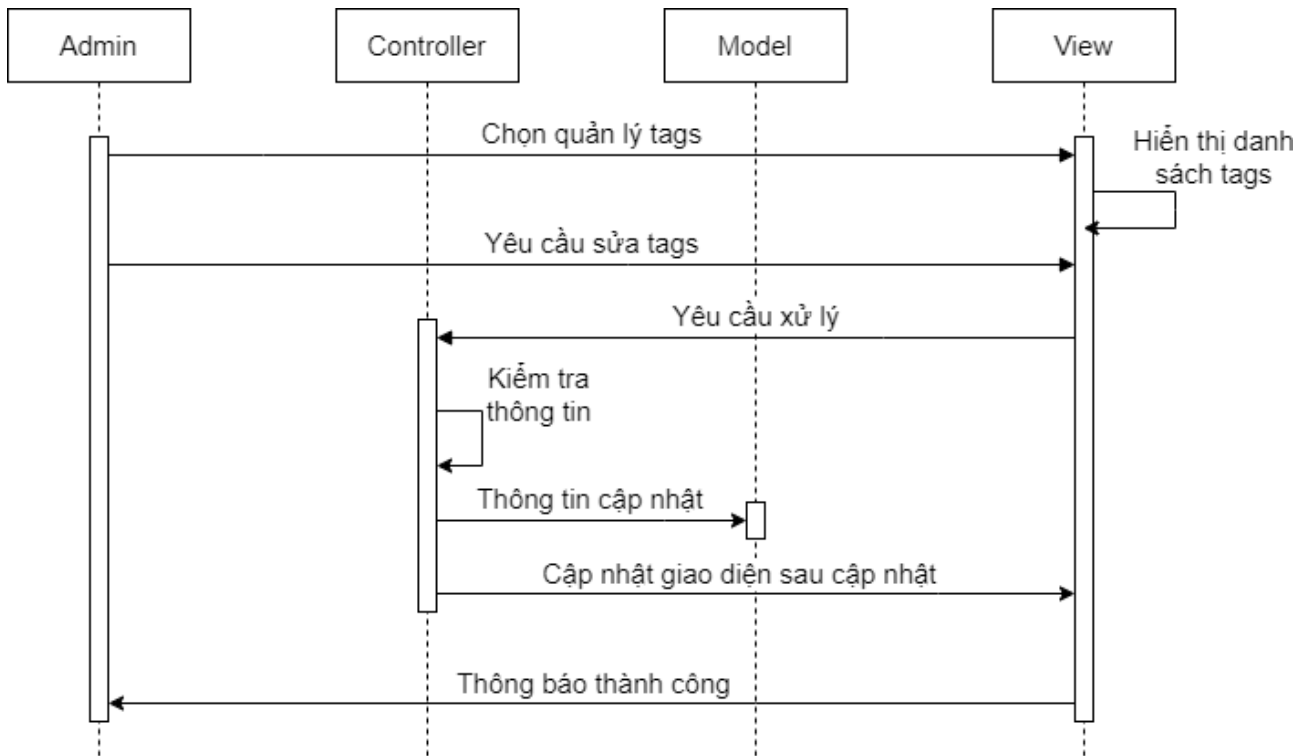
Sau khi yêu cầu quản lý thẻ loại, view sẽ hiển thị danh sách các thẻ loại đã được thêm từ cơ sở dữ liệu. Tùy vào yêu cầu người dùng, hệ thống sẽ thực hiện các công việc dưới đây

a. Thêm thẻ loại mới



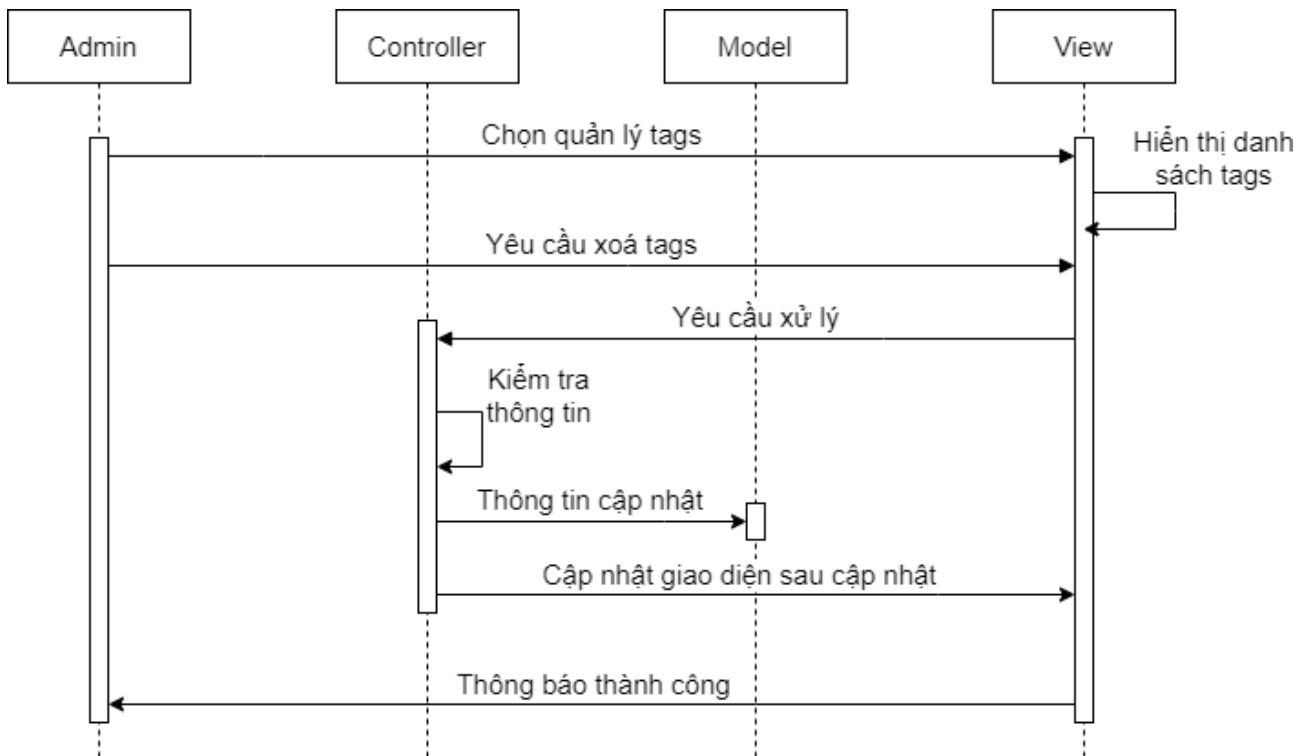
Khi người dùng yêu cầu thêm thẻ loại, view sẽ gửi thông tin thẻ loại mới cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

b. Sửa thể loại



Khi người dùng yêu cầu sửa thể loại, view sẽ gửi nội dung chỉnh sửa cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

c. Xóa thể loại

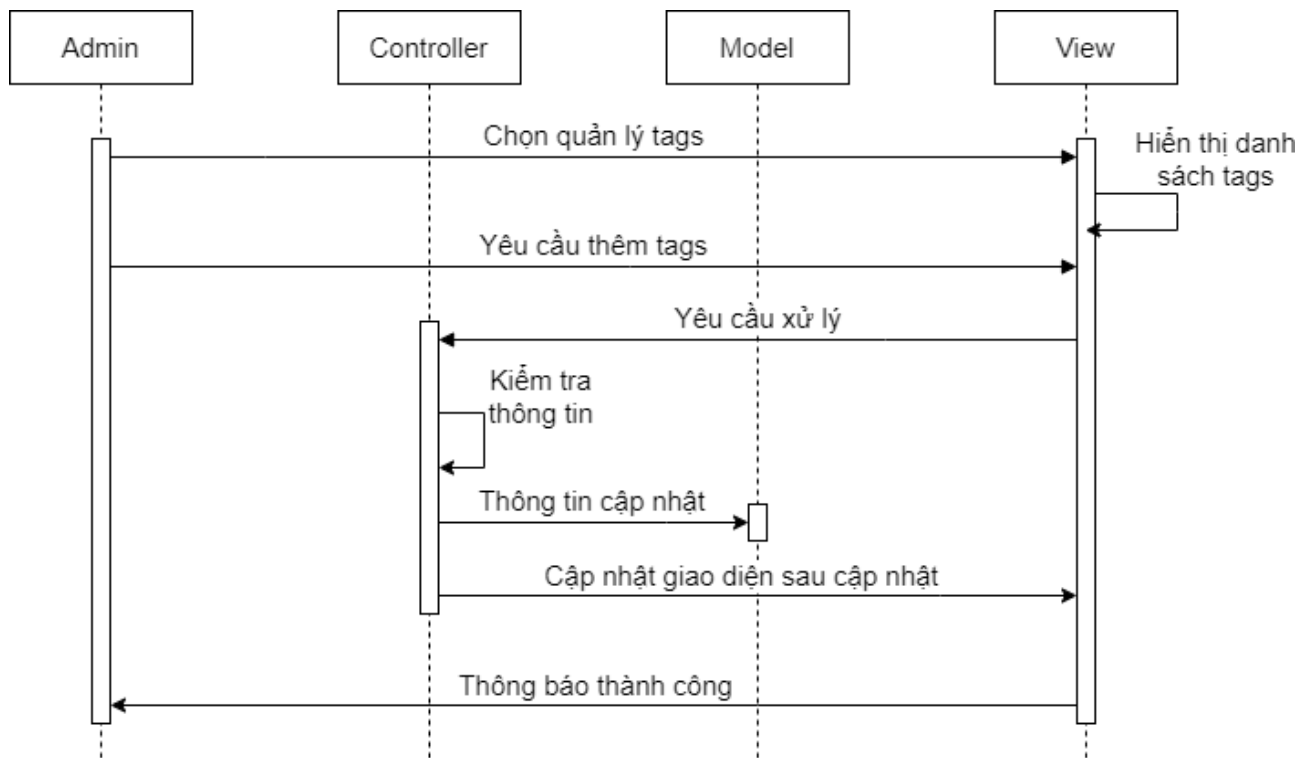


Khi người dùng yêu cầu sửa thể loại, view sẽ gửi yêu cầu xoá thể loại cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

6. Quản lý tags

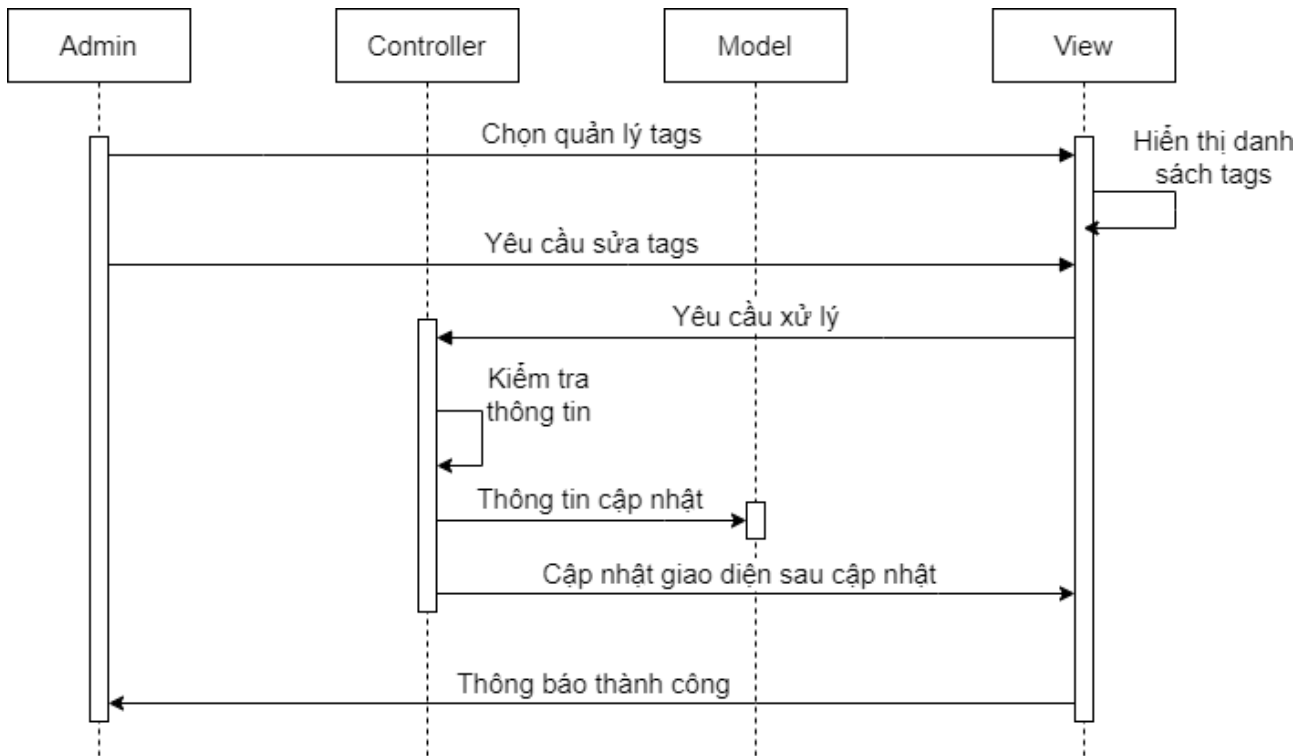
Sau khi yêu cầu quản lý tags, view sẽ hiển thị danh sách các tags đã được thêm từ cơ sở dữ liệu. Tùy vào yêu cầu người dùng, hệ thống sẽ thực hiện các công việc dưới đây

a. *Thêm tags*



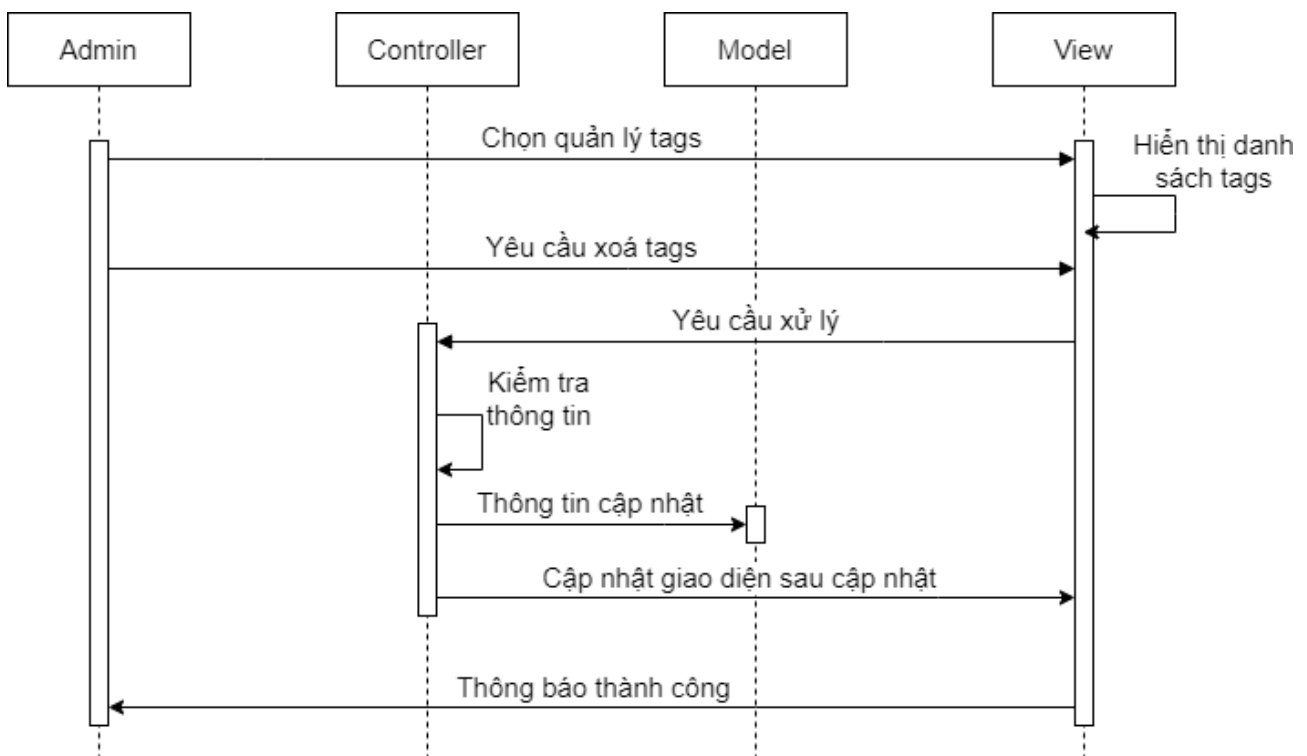
Khi người dùng yêu cầu thêm tags, view sẽ gửi thông tin tags mới cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

b. Sửa tags



Khi người dùng yêu cầu sửa tag, view sẽ gửi nội dung chỉnh sửa cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

c. Xóa tags

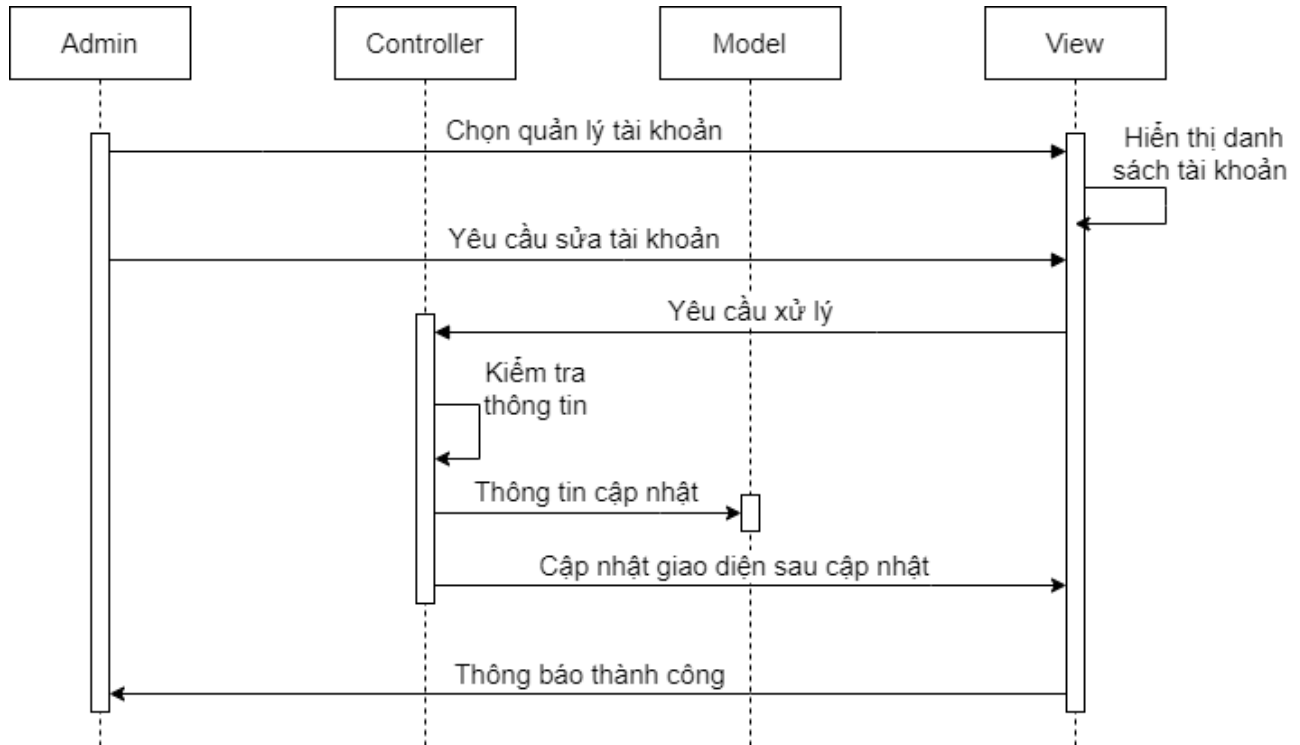


Khi người dùng yêu cầu xóa tag, view sẽ yêu cầu xóa tag cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

7. Quản lý tài khoản

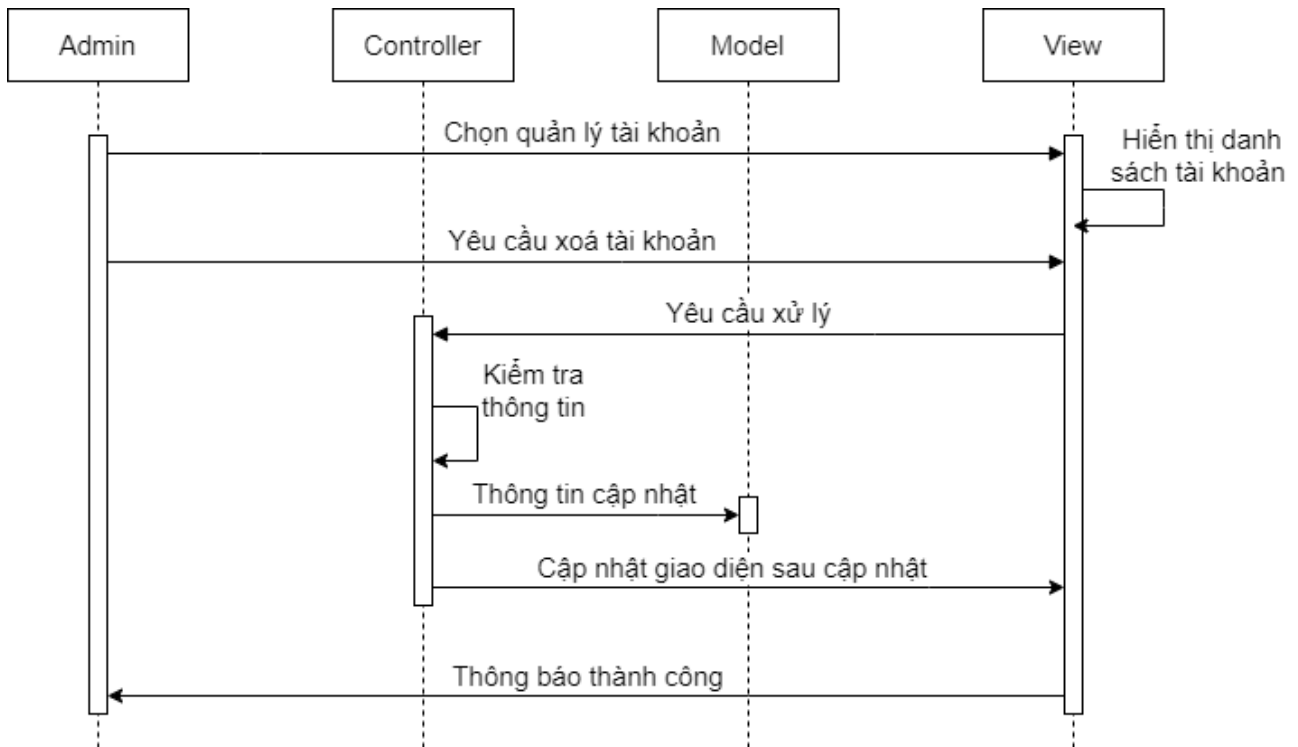
Sau khi yêu cầu quản lý tài khoản, view sẽ hiển thị danh sách các tài khoản đã được thêm từ cơ sở dữ liệu. Tùy vào yêu cầu người dùng, hệ thống sẽ thực hiện các công việc dưới đây

a. Sửa tài khoản



Khi người dùng yêu cầu sửa tài khoản, view sẽ gửi nội dung chỉnh sửa cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

b. Xóa tài khoản

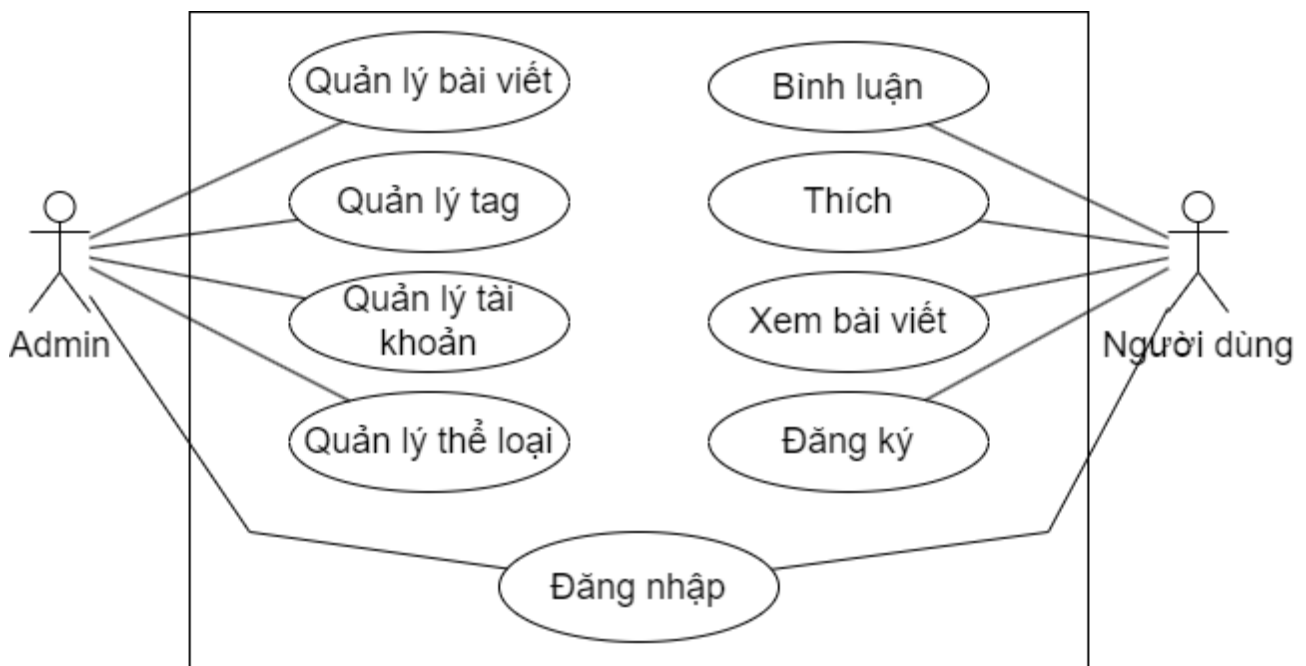


Khi người dùng yêu cầu xóa tài khoản, view sẽ yêu cầu xóa tài khoản cho controller kiểm tra trước khi gửi tín hiệu cập nhật cho model và cập nhật giao diện sau khi hoàn thành.

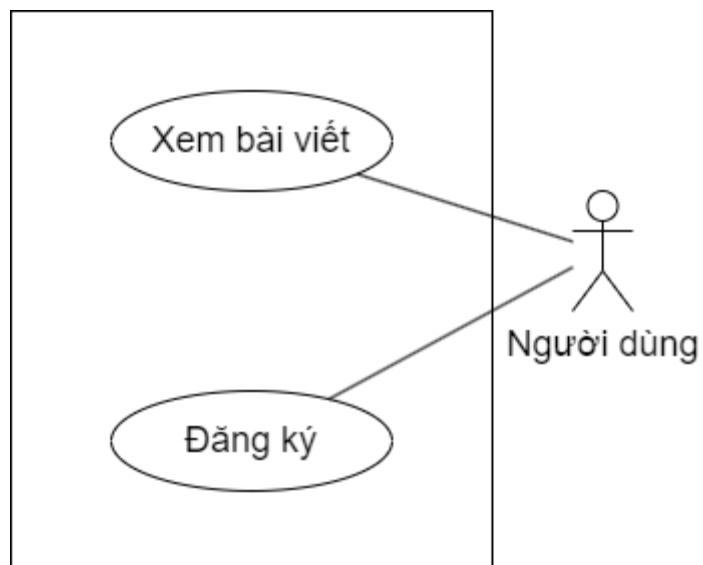
VI. Đặc tả yêu cầu

1. Biểu đồ usecase

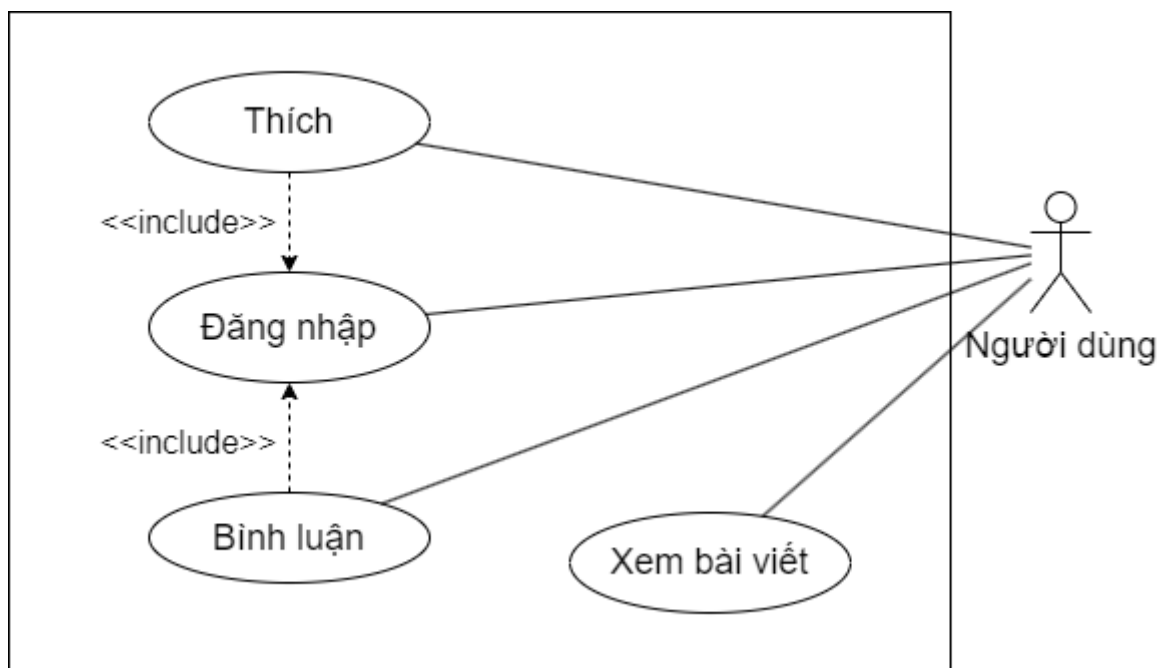
a. Usecase tổng quát



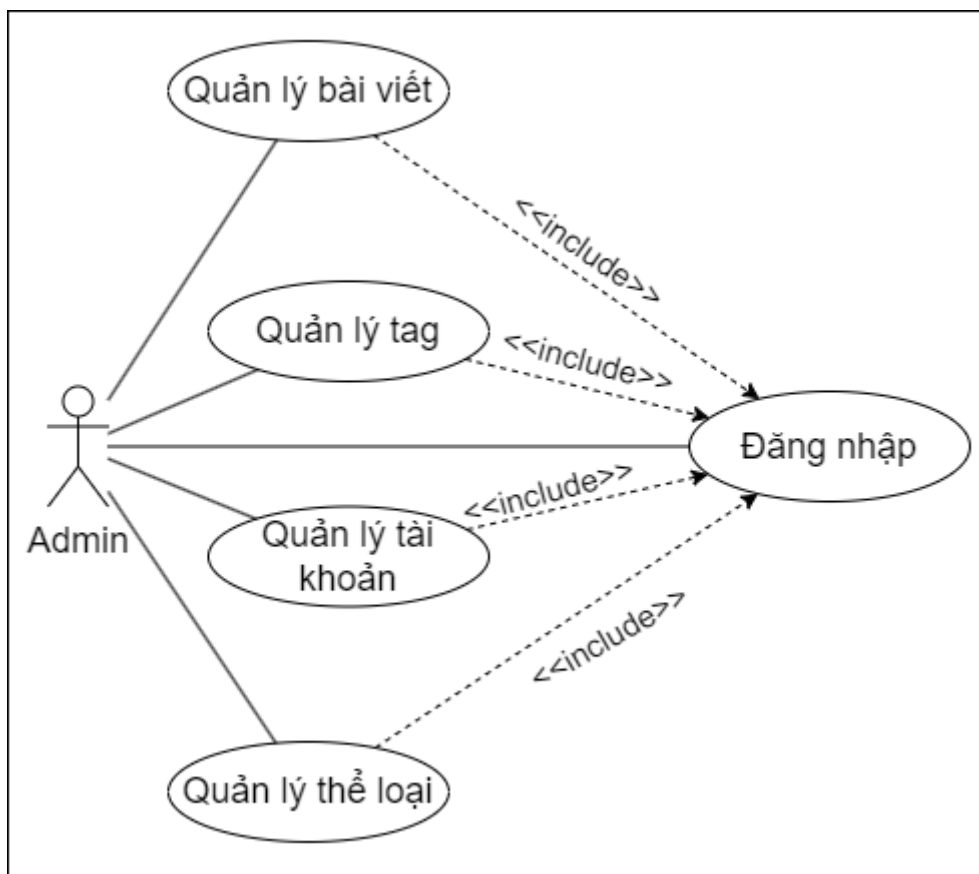
b. Usecase đối với người dùng mới



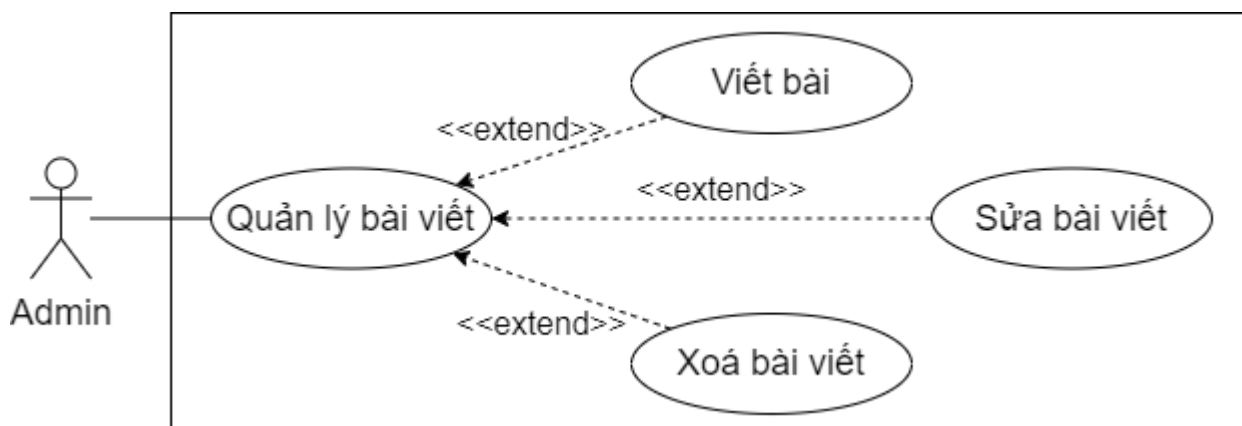
c. Usecase đối với người dùng cũ



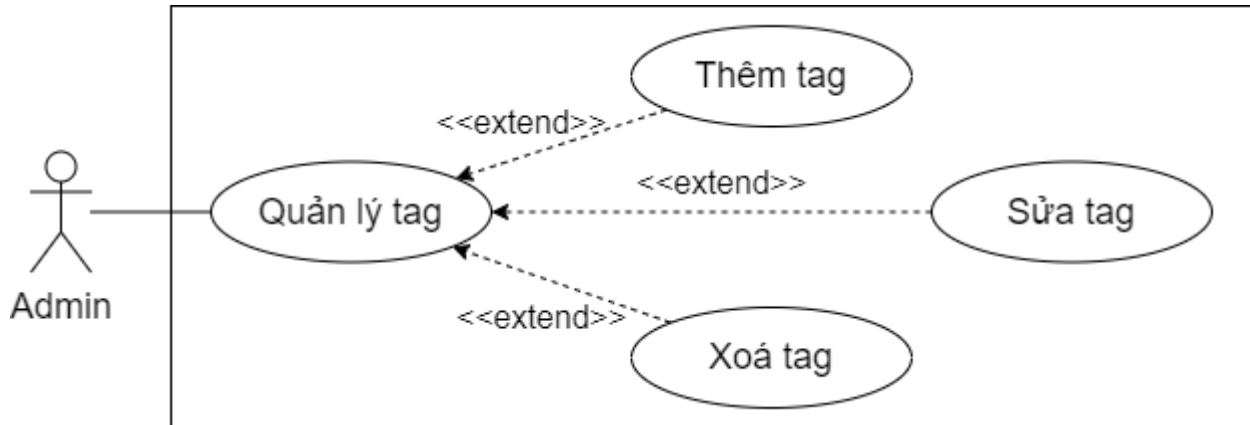
d. Usecase đối với admin



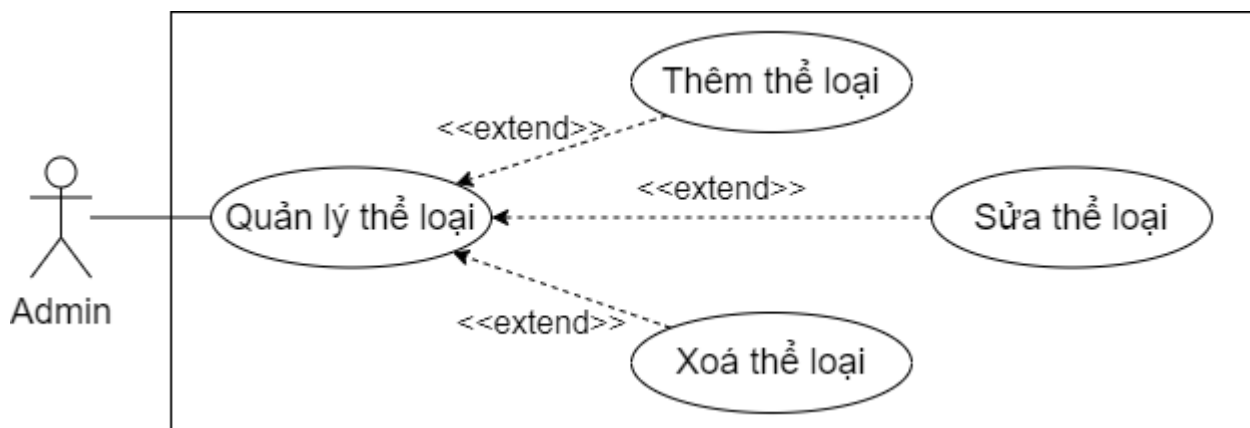
- Usecase quản lý bài viết



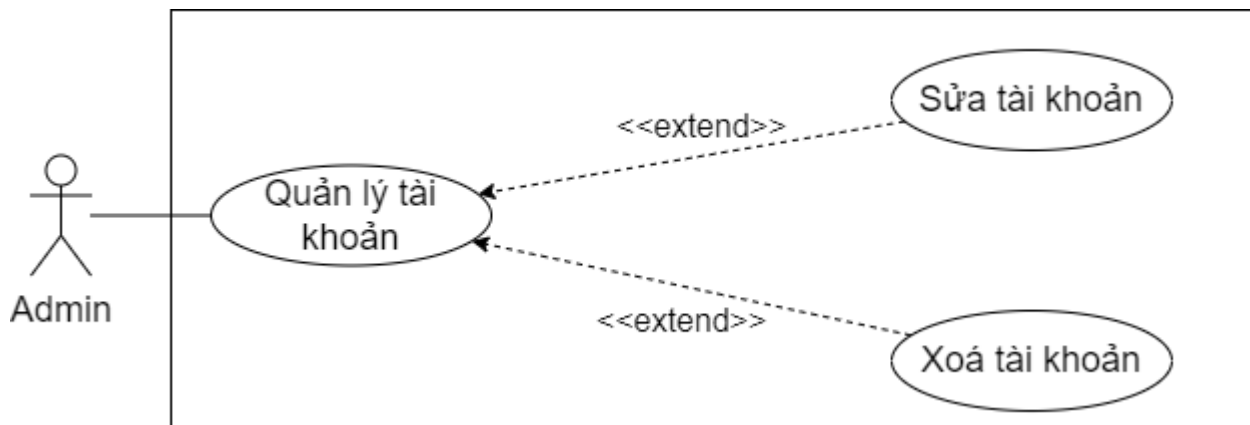
- Usecase quản lý tags



- Usecase quản lý thể loại bài viết



- Usecase quản lý tài khoản



2. Đặc tả Usecase

a. Đăng nhập

Tên usecase	Đăng nhập
Mô tả	Là người dùng, tôi muốn đăng nhập vào trang web
Đối tượng	Người dùng, Admin
Mức ưu tiên	Bắt buộc phải có

Kích hoạt	Người dùng muốn đăng nhập vào trang web
Điều kiện trước khi thực hiện	Người dùng đã có tài khoản Thiết bị của người dùng đã kết nối internet
Điều kiện sau khi thực hiện	Người dùng đăng nhập vào trang web thành công
Luồng thực hiện	1. Người dùng truy cập vào trang web 2. Người dùng click vào nút đăng nhập 3. Người dùng nhập tài khoản, mật khẩu và nhập lệnh đăng nhập 4. Hệ thống xác thực thông tin đăng nhập thành công và cho phép đăng nhập vào hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin đăng nhập thất bại 4c1. Người dùng huỷ đăng nhập – Usecase dừng lại 4c2. Người dùng click vào nút đăng kí – Usecase tiếp tục ở: Đăng kí.

b. Đăng kí

Tên usecase	Đăng kí
Mô tả	Là người dùng, tôi muốn đăng kí tài khoản mới để vào trang web
Đối tượng	Người dùng
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Người dùng muốn đăng kí vào trang web
Điều kiện trước khi thực hiện	Người dùng chưa có tài khoản Thiết bị của người dùng đã kết nối internet
Điều kiện sau khi thực hiện	Người dùng đăng kí vào trang web thành công Tài khoản mới được tạo và thêm vào database thành công
Luồng thực hiện	1. Người dùng truy cập vào trang web 2. Người dùng click vào nút đăng kí 3. Người dùng nhập tài khoản, mật khẩu, email và nhập lệnh đăng kí 4. Hệ thống xác thực thông tin đăng kí, tạo tài khoản mới thành công và cho phép đăng nhập vào hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin đăng kí thất bại 4c1. Người dùng huỷ đăng kí – Usecase dừng lại

	4c2. Người dùng click vào nút đăng nhập – Usecase tiếp tục ở: Đăng nhập.
--	--------------------------------------------------------------------------

c. Đăng bài

Tên usecase	Đăng bài
Mô tả	Là admin, tôi muốn viết bài viết mới và đăng lên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn viết bài viết mới
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin nhập và đăng bài viết vào trang web thành công Bài viết được tạo và thêm vào database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút viết bài 3. Admin nhập tiêu đề, mô tả, nội dung, hình ảnh và nhấn đăng bài 4. Hệ thống xác thực thông tin bài viết thành công và cho phép đăng bài vào hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin bài viết mới thất bại 4c1. Admin huỷ viết bài – Usecase dừng lại

d. Quản lý bài viết

- Sửa bài viết

Tên usecase	Sửa bài viết
Mô tả	Là admin, tôi muốn sửa bài viết đã đăng và cập nhật lại trên trang web
Đối tượng	Admin
Mức ưu tiên	Chức năng bổ sung
Kích hoạt	Admin muốn sửa bài viết cũ
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet

	Bài viết đã được viết và đăng lên trang web Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin sửa bài viết thành công Bài viết được sửa và cập nhật vào database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý bài viết 3. Admin chọn sửa một bài viết cần sửa trên sách bài viết 4. Admin nhập tiêu đề, mô tả, nội dung, hình ảnh mới và nhấn đăng bài 5. Hệ thống xác thực thông tin bài viết thành công và cho phép cập nhật bài viết vào hệ thống
Luồng ngoại lệ	5c. Hệ thống xác thực thông tin bài viết mới thất bại 5c1. Admin huỷ viết bài – Usecase dừng lại

- Xoá bài viết

Tên usecase	Xoá bài viết
Mô tả	Là admin, tôi muốn xoá bài viết đã đăng trên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn xoá bài viết cũ
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Bài viết đã được viết và đăng lên trang web Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin xoá bài viết thành công Bài viết được xoá khỏi database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý bài viết 3. Admin chọn xoá một bài viết cần xoá trên sách bài viết 4. Hệ thống xác thực thông tin và cho phép xoá bài viết khỏi hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin bài viết mới thất bại 4c1. Admin huỷ xoá bài – Usecase dừng lại

e. Quản lý tags

- Thêm tag

Tên usecase	Thêm tag
Mô tả	Là admin, tôi muốn tạo thêm tag cho bài viết trên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn thêm tag mới
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin thêm tag thành công Tag được tạo và cập nhật vào database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý tag 3. Admin nhập tên tag và nhấn lưu 4. Hệ thống xác thực thông tin tag thành công và cho phép thêm tag vào hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin tag mới thất bại 4c1. Admin huỷ thêm tag – Usecase dừng lại

- Sửa tag

Tên usecase	Sửa tag
Mô tả	Là admin, tôi muốn sửa tag đã thêm trên trang web
Đối tượng	Admin
Mức ưu tiên	Chức năng bổ sung
Kích hoạt	Admin muốn sửa tag cũ
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tag đã được thêm vào hệ thống Tài khoản của admin đã có phân quyền

Điều kiện sau khi thực hiện	Admin sửa tag thành công Tag được cập nhật vào database thành công
Luồng thực hiện	<ol style="list-style-type: none"> 1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý tag 3. Admin chọn một tag đã thêm từ danh sách các tag 4. Admin nhập tên tag và nhấn lưu 5. Hệ thống xác thực thông tin tag thành công và cho phép cập nhật tag vào hệ thống
Luồng ngoại lệ	5c. Hệ thống xác thực thông tin tag mới thất bại 5c1. Admin huỷ sửa tag – Usecase dừng lại

- Xoá tag

Tên usecase	Xoá tag
Mô tả	Là admin, tôi muốn xoá tag đã tạo trên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn xoá tag cũ
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tag đã được tạo trước đó Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin xoá tag thành công Tag được xoá khỏi database thành công
Luồng thực hiện	<ol style="list-style-type: none"> 1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý tag 3. Admin nhấn xoá tag cần xoá trên danh sách các tag 4. Hệ thống xác thực thông tin tag thành công và cho phép xoá tag khỏi hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin tag thất bại 4c1. Admin huỷ xoá tag – Usecase dừng lại

f. Quản lý thể loại

- Thêm thể loại

Tên usecase	Thêm thể loại
Mô tả	Là admin, tôi muốn tạo thêm thể loại cho bài viết trên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn thêm thể loại mới
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin thêm thể loại thành công Thể loại được tạo và cập nhật vào database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý thể loại 3. Admin nhập tên thể loại và nhấn lưu 4. Hệ thống xác thực thông tin thể loại thành công và cho phép thêm thể loại vào hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin thể loại mới thất bại 4c1. Admin hủy thêm thể loại – Usecase dừng lại

- Sửa thể loại

Tên usecase	Sửa thể loại
Mô tả	Là admin, tôi muốn sửa thể loại đã thêm trên trang web
Đối tượng	Admin
Mức ưu tiên	Chức năng bổ sung
Kích hoạt	Admin muốn sửa thể loại cũ
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Thể loại đã được thêm vào hệ thống Tài khoản của admin đã có phân quyền

Điều kiện sau khi thực hiện	Admin sửa thể loại thành công Thể loại được cập nhật vào database thành công
Luồng thực hiện	<ol style="list-style-type: none"> 1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý thể loại 3. Admin chọn một thể loại đã thêm từ danh sách các thể loại 4. Admin nhập tên thể loại và nhấn lưu 5. Hệ thống xác thực thông tin thể loại thành công và cho phép cập nhật thể loại vào hệ thống
Luồng ngoại lệ	5c. Hệ thống xác thực thông tin thể loại mới thất bại 5c1. Admin huỷ sửa thể loại – Usecase dừng lại

- Xóa thể loại

Tên usecase	Xóa thể loại
Mô tả	Là admin, tôi muốn xóa thể loại đã tạo trên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn xóa thể loại cũ
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet thể loại đã được tạo trước đó Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin xóa thể loại thành công thể loại được xóa khỏi database thành công
Luồng thực hiện	<ol style="list-style-type: none"> 1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý thể loại 3. Admin nhấn xóa thể loại cần xóa trên danh sách các thể loại 4. Hệ thống xác thực thông tin thể loại thành công và cho phép xóa thể loại khỏi hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin thể loại thất bại 4c1. Admin huỷ xóa thể loại – Usecase dừng lại

g. Quản lý tài khoản

- Sửa tài khoản

Tên usecase	Sửa tài khoản
Mô tả	Là admin, tôi muốn sửa tài khoản đã thêm trên trang web
Đối tượng	Admin
Mức ưu tiên	Chức năng bổ sung
Kích hoạt	Admin muốn sửa tài khoản
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tài khoản đã được thêm vào hệ thống Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin sửa tài khoản thành công Tài khoản được cập nhật vào database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý tài khoản 3. Admin chọn một tài khoản đã thêm từ danh sách các tài khoản 4. Admin thay đổi phân quyền hoặc các thông tin khác và nhấn lưu 5. Hệ thống xác thực thông tin tài khoản thành công và cho phép cập nhật tài khoản vào hệ thống
Luồng ngoại lệ	5c. Hệ thống xác thực thông tin tài khoản mới thất bại 5c1. Admin huỷ sửa tài khoản – Usecase dừng lại

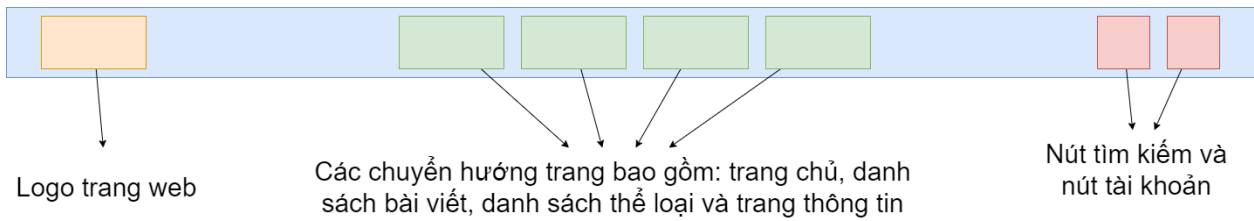
- Xóa tài khoản

Tên usecase	Xóa tài khoản
Mô tả	Là admin, tôi muốn xóa tài khoản đã tạo trên trang web
Đối tượng	Admin
Mức ưu tiên	Bắt buộc phải có
Kích hoạt	Admin muốn xóa tài khoản
Điều kiện trước khi thực hiện	Admin đã đăng nhập vào tài khoản admin Thiết bị của người dùng đã kết nối internet Tài khoản đã được tạo trước đó

	Tài khoản của admin đã có phân quyền
Điều kiện sau khi thực hiện	Admin xoá tài khoản thành công Tài khoản được xoá khỏi database thành công
Luồng thực hiện	1. Admin truy cập vào trang web 2. Admin nhấn vào nút quản lý tài khoản 3. Admin nhấn xoá tài khoản cần xoá trên danh sách các tài khoản 4. Hệ thống xác thực thông tin tài khoản thành công và cho phép xoá tài khoản khỏi hệ thống
Luồng ngoại lệ	4c. Hệ thống xác thực thông tin tài khoản thất bại 4c1. Admin huỷ xoá tài khoản – Usecase dừng lại

VII. Bản vẽ giao diện

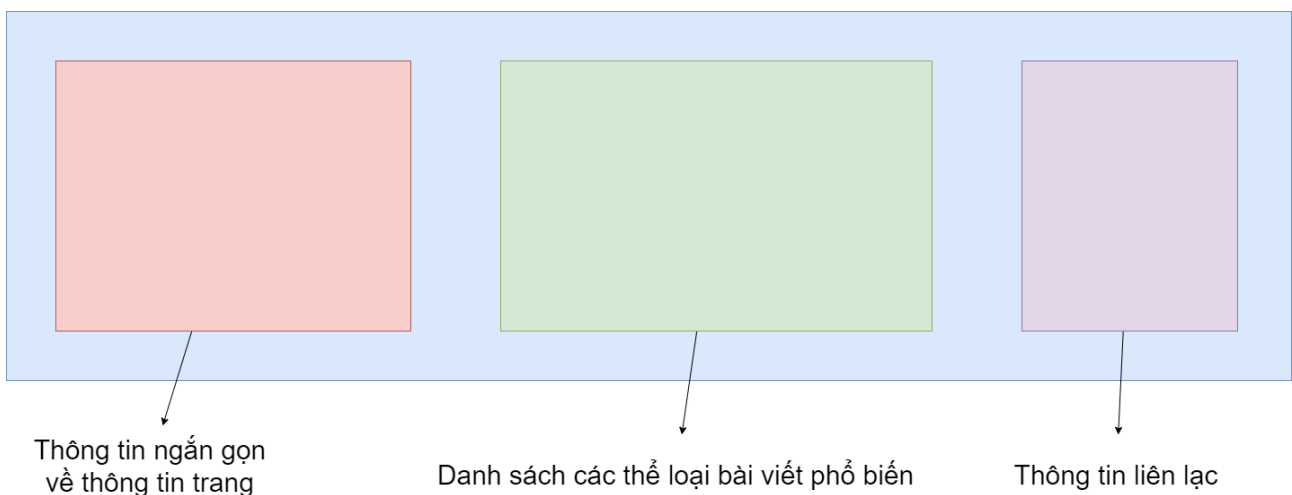
1. Header



Hiện thị ở đầu trang, thanh header này chứa toàn bộ các liên kết dẫn đến các danh mục quan trọng của trang web, bao gồm trang chủ, danh sách bài viết được phân loại theo các danh mục.

Ở góc trái và phải của thanh này sẽ là logo web và các chức năng khác như tìm kiếm và đăng nhập.

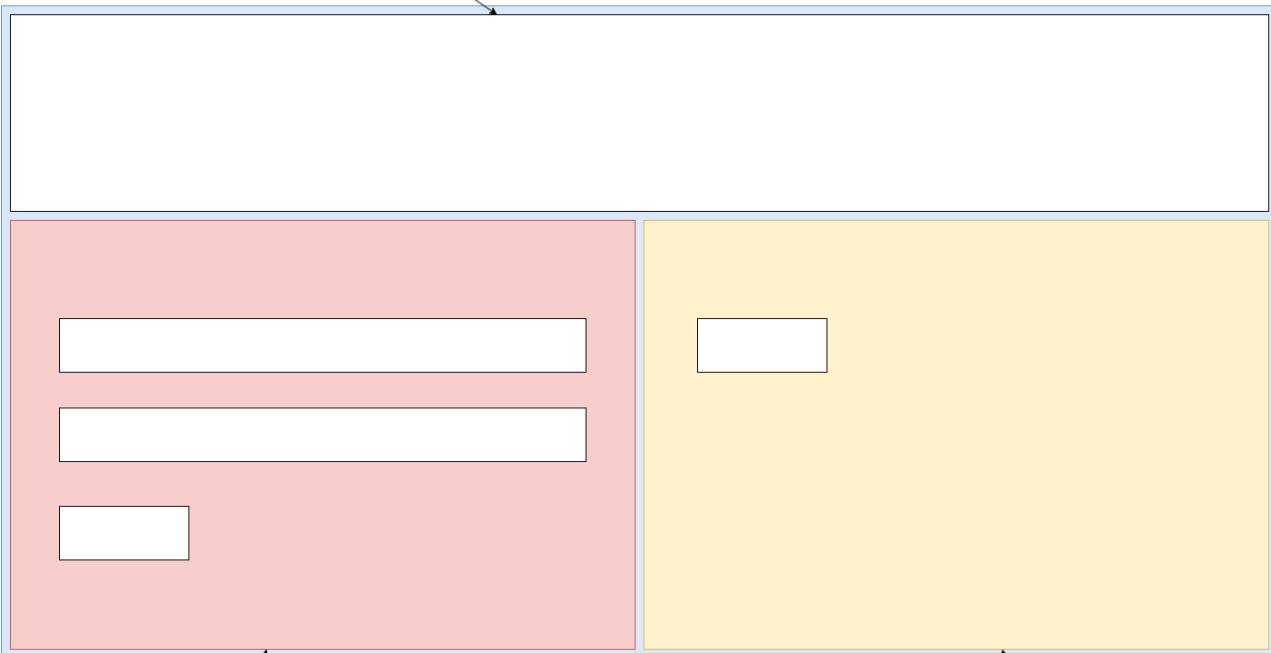
2. Footer



Footer hiển thị ở cuối trang, chứa ba phần thông tin gồm thông tin của trang web, các danh sách thể loại phổ biến và thông tin liên lạc đến chủ trang web.

3. Đăng nhập

Lời chào và yêu cầu đăng nhập



The diagram shows a login page layout. At the top is a white header box. Below it is a main content area divided into two columns. The left column has a light red background and contains three white input fields of varying lengths. The right column has a light yellow background and contains a single small white input field.

Phần nhập thông tin đăng nhập

Chuyển trang sang đăng kí nếu người dùng chưa có tài khoản

Trang đăng nhập có ba phần gồm phần giới thiệu, phần đăng nhập ở ô trái và chuyển hướng đăng kí ở ô phải nếu người dùng chưa có tài khoản.

4. Đăng kí

Lời chào và yêu cầu đăng kí



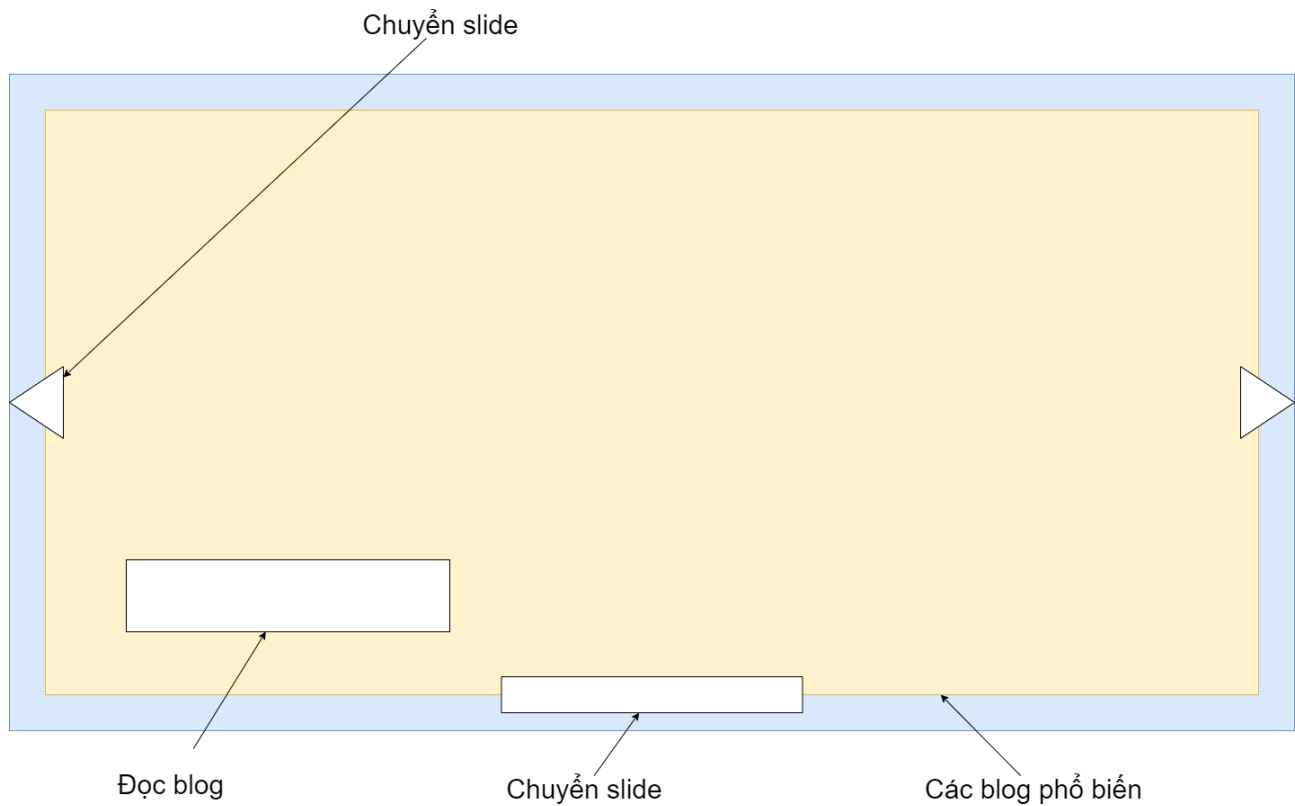
The diagram shows a registration page layout. At the top is a white header box. Below it is a main content area divided into two columns. The left column has a light red background and contains four white input fields of varying lengths. The right column has a light yellow background and is empty.

Phần nhập thông tin đăng kí

Giới thiệu về page

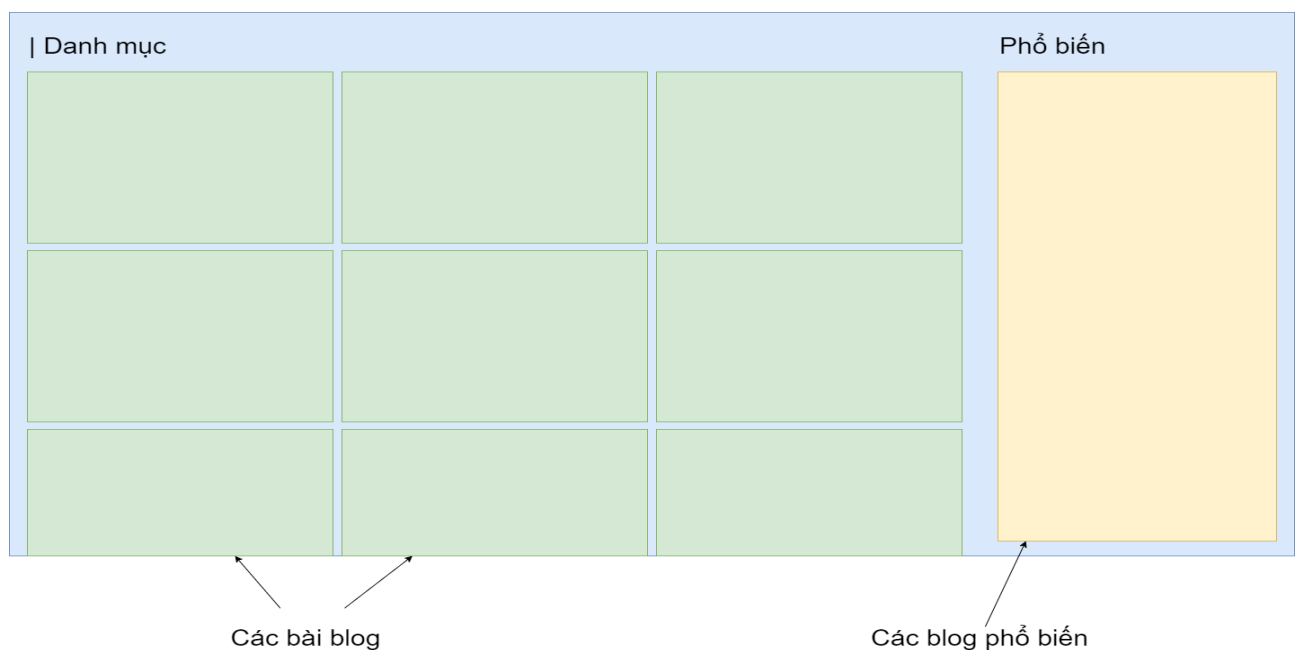
Tương tự như trang đăng nhập, trang đăng kí cũng có ba phần gồm phần giới thiệu và phần đăng kí ở ô trái. Tuy nhiên ô phải sẽ là phần giới thiệu về web.

5. Giao diện chính



Phần giao diện chính của web, hiển thị ba trang nổi bật nhất của website dưới dạng slide. Phần này sẽ được hiển thị khi truy cập trang chủ.

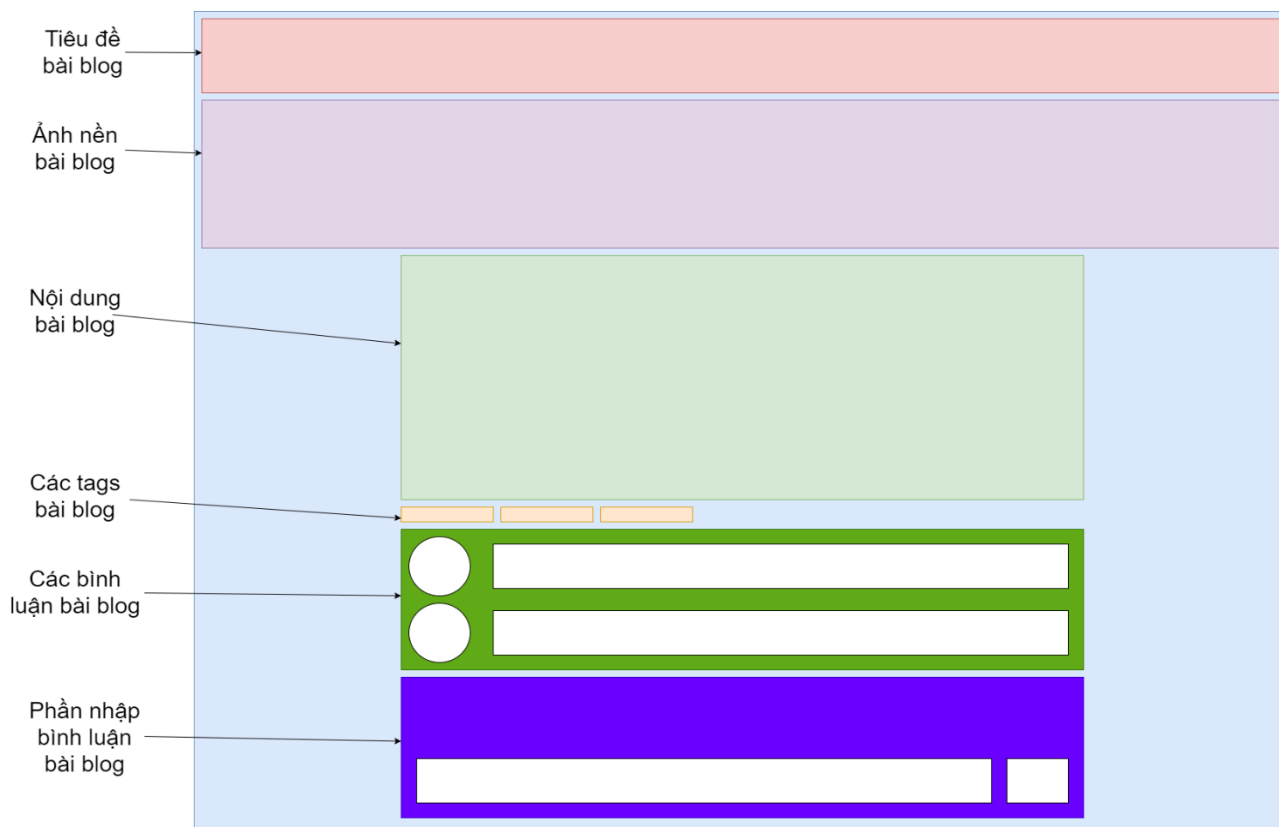
6. Danh sách các blog



Phần này sẽ hiển thị danh sách các blog, sẽ xuất hiện ở các trang bao gồm trang chủ, trang danh sách blog theo thể loại hay kết quả sau khi tìm kiếm.

Ngoài ra sẽ có một danh sách nhỏ khác chứa các blog phổ biến cố định.

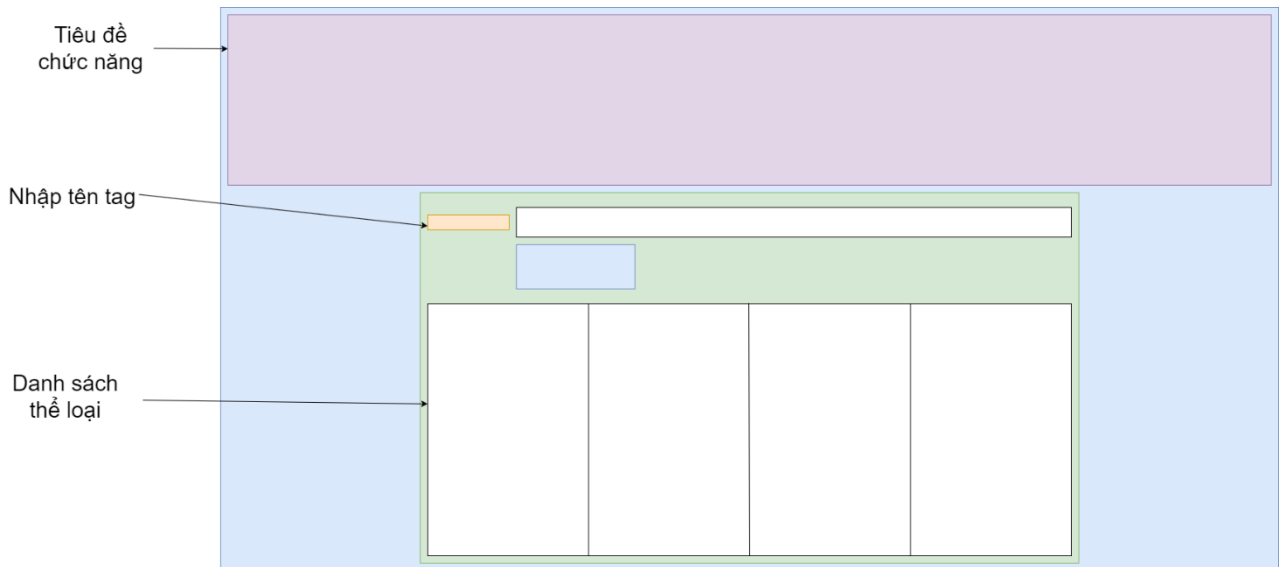
7. Trang blog



Mỗi bài viết hay blog sẽ có 3 phần chính gồm:

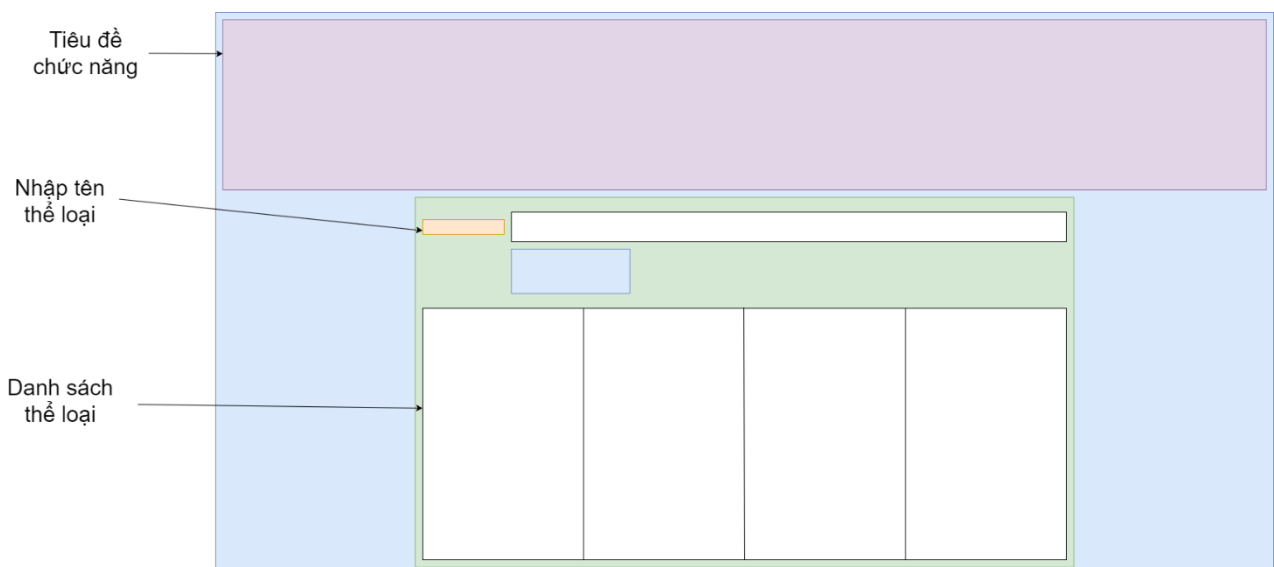
- Phần giới thiệu sẽ chứa tiêu đề và các thông tin liên quan như người viết, ngày đăng. Ngoài ra sẽ có một ảnh nền của blog ngay dưới thanh tiêu đề.
- Phần nội dung sẽ chứa toàn bộ nội dung của bài blog. Ngay cuối mỗi bài viết sẽ có danh sách các tags của blog.
- Phần bình luận giúp người đọc để lại bình luận cho các bài viết. Ngoài ra sẽ có nút like nên người đọc thích nó.

8. Quản lý Tags



Ở trang quản lý tags có hai phần chính gồm phần nhập thông tin tags và table hiển thị toàn bộ tag đã lưu. Ngoài ra trên table có hai button điều chỉnh và xoá tags.

9. Quản lý thể loại bài viết.



Tương tự như trang quản lý tag, trang quản lý thể loại cũng có hai phần chính gồm phần nhập thông tin thể loại và table hiển thị toàn bộ thể loại đã lưu. Ngoài ra trên table có hai button điều chỉnh và xoá thể loại.

10. Viết Blog.

The diagram illustrates the 'Viết Blog' (Write Blog) form. It features a light blue background with a central green-bordered form area. At the top of the form is a large purple rectangular field for the title. Below this are four orange rectangular buttons arranged vertically, each with a corresponding label to its left: 'Tiêu đề chức năng' (Title function), 'Nhập tiêu đề' (Enter title), 'Nhập mô tả' (Enter description), and 'Nhập nội dung' (Enter content). To the right of these buttons are three white input fields. The third input field is significantly larger than the others. Below the input fields are two more orange buttons, labeled 'Upload ảnh' (Upload image) and 'Chọn thể loại' (Select category). To the right of these buttons is a white dropdown menu with a small 'v' icon. Below the dropdown is a large white text area for tags, with a label 'Chọn các tags' (Select tags) to its left. At the bottom of the form is a white button labeled 'Lưu' (Save).

Chức năng viết blog có sáu mục thông tin cần lưu trữ bao gồm tiêu đề, mô tả, nội dung, ảnh, thể loại, tags. Sau khi nhập xong người viết có thể lưu.

11. Quản lý bài viết

The diagram shows the 'Quản lý bài viết' (Manage Posts) interface. It has a light blue background with a large purple rectangular field at the top for the title, labeled 'Tiêu đề chức năng' (Title function). Below this is a white table with five columns, labeled 'Danh sách bài viết' (List of posts). The table is currently empty.

Chức năng này hiển thị danh sách các bài viết đã đăng, kèm theo hai button để điều chỉnh và xoá bài viết.

VIII. Cơ sở dữ liệu

1. Các bảng (table)

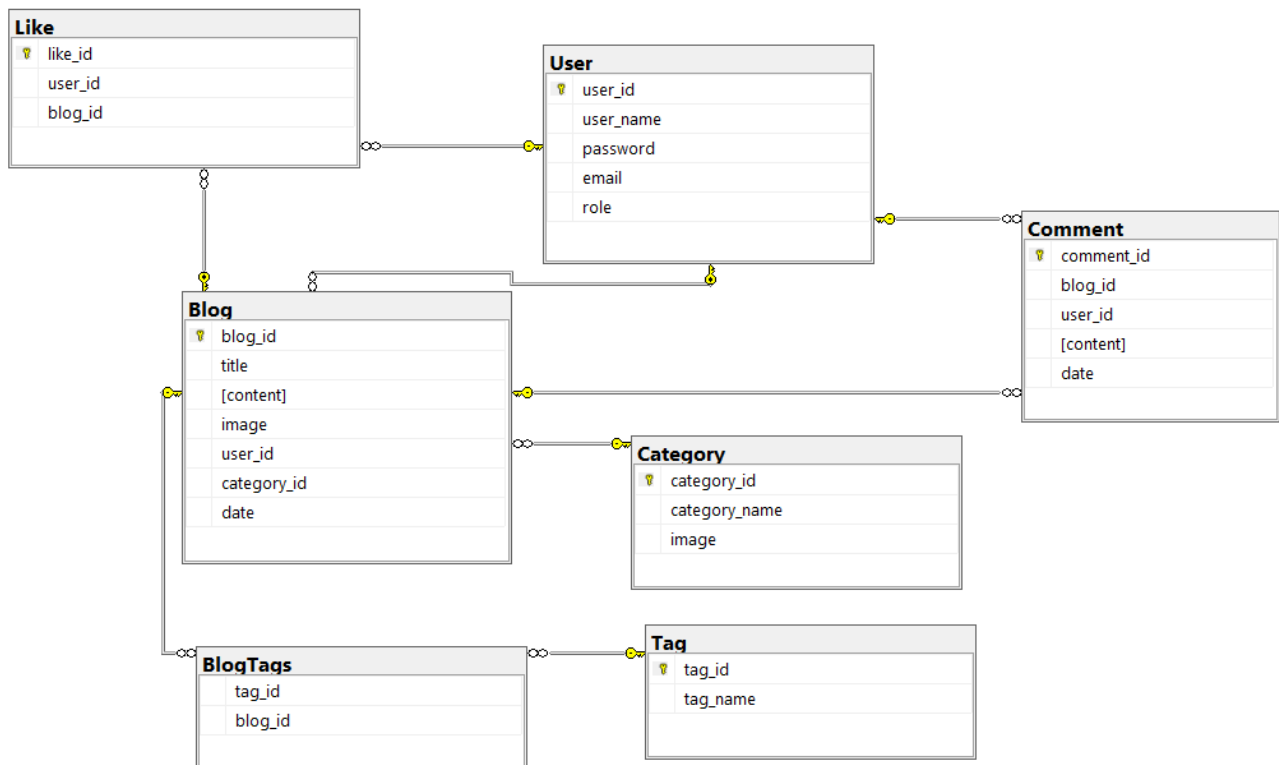
STT	Tên Bảng	Alias (bí danh)	Mô tả
1	Tài Khoản	USER	Cho biết tên người dùng, mật khẩu, email, chức vụ tài khoản Mỗi tài khoản có một mã duy nhất và không có tài khoản trùng tên người dùng.
2	Lượt thích	LIKE	Cho biết mã người dùng và mã bài viết được thích. Mỗi người chỉ được thích bài viết một lần và có thể thích nhiều bài viết
3	Bình luận	COMMENT	Cho biết mã người dùng và mã bài viết được bình luận. Mỗi người có thể bình luận bài viết nhiều lần và có thể bình luận nhiều bài viết
4	Từ khoá	TAGS	Cho biết mã và tên từ khoá Mỗi từ khoá sẽ được sử dụng trong nhiều bài viết.
5	Thể loại	CATEGORY	Cho biết mã, tên và hình ảnh minh họa thể loại Mỗi thể loại sẽ chứa nhiều bài viết.
6	Chi tiết từ khoá	BLOGTAGS	Cho biết mã bài viết và mã tag. Mỗi bài viết có nhiều chi tiết từ khoá, từ đó có thể lưu nhiều từ khoá khác nhau cho cùng một bài viết.
7	Bài viết	BLOG	Cho biết tiêu đề, người tạo, ngày tạo, ảnh nền, nội dung và thể loại. Mỗi bài viết có nhiều lượt thích, bình luận và từ khoá (tag) và chỉ có duy nhất một người viết, một thể loại.

2. Thực thể và thuộc tính

STT	Tên Bảng	Alias (bí danh)	Thuộc tính	Mô tả thuộc tính
1	Tài Khoản	USER	User_id	Mã tài khoản
			Password	Mật khẩu
			User_name	Tên tài khoản
			Email	Email

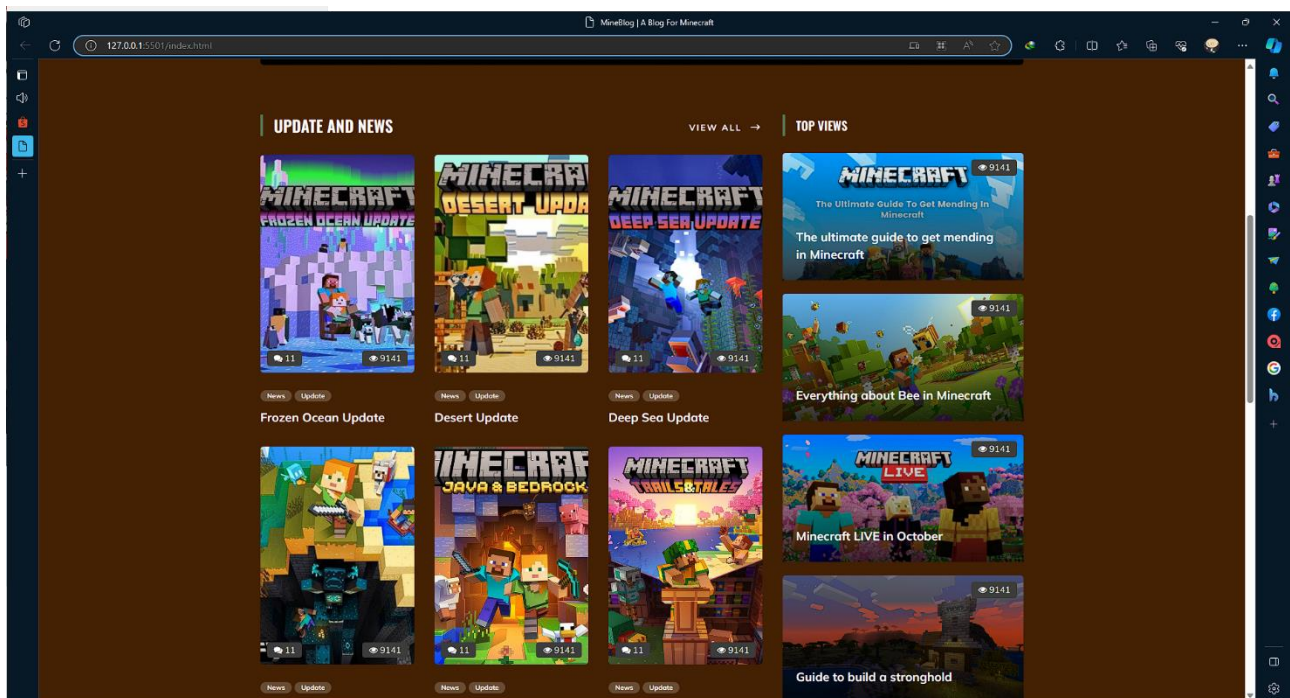
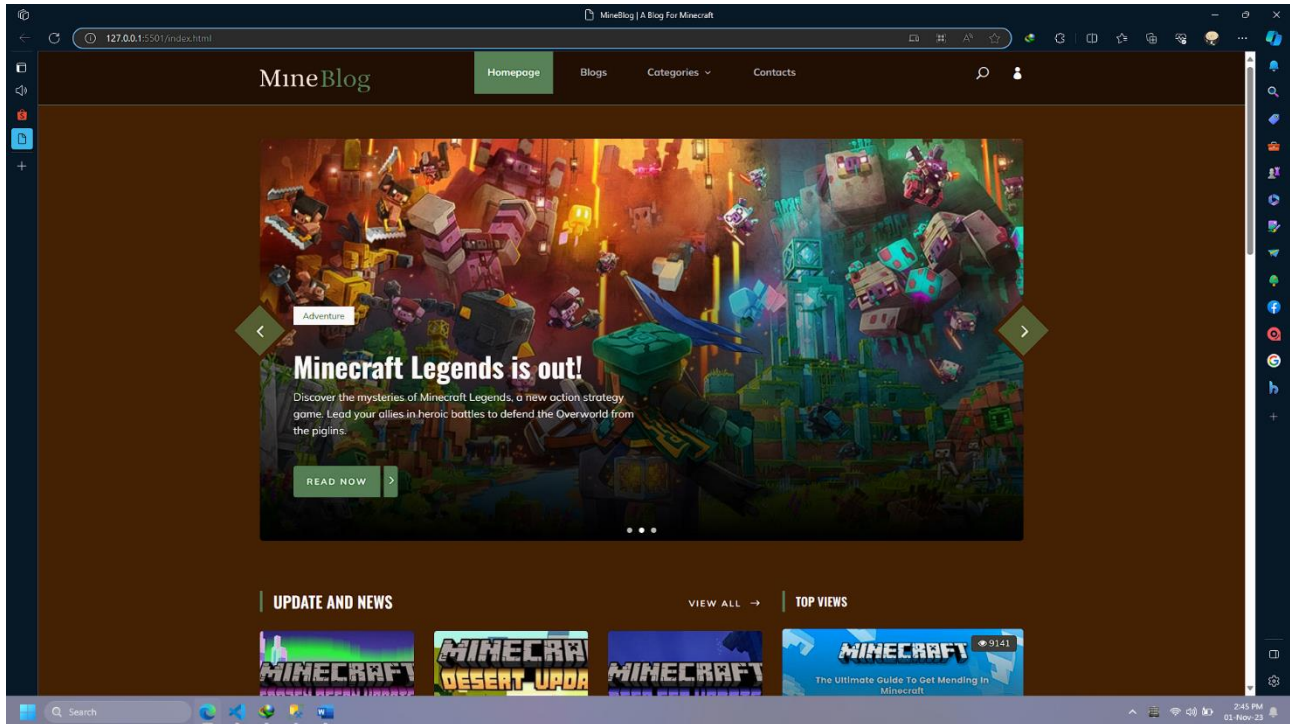
			Role	Loại tài khoản
2	Lượt thích	LIKE	Like_id	Mã lượt thích
			User_id	Tên tài khoản thích
			Blog_id	Tên bài được thích
3	Bình luận	COMMENT	Comment_id	Mã bình luận
			User_id	Tên tài khoản bình luận
			Blog_id	Tên bài được bình luận
			Content	Nội dung bình luận
			Date	Ngày bình luận
4	Từ khoá	TAGS	Tag_id	Mã tag
			Tag_name	Tên tag
5	Thể loại	CATEGORY	Category_id	Mã thể loại
			Category_name	Tên thể loại
			Image	ảnh minh hoạ
6	Chi tiết từ khoá	BLOGTAGS	Tag_id	Mã tags
			Blog_id	Mã bài blog
7	Bài viết	BLOG	Blog_id	Mã bài blog
			Title	Tiêu đề
			Content	Nội dung bài blog
			Image	ảnh minh hoạ
			User_id	Mã người viết
			Category_id	Mã thể loại
			Date	Ngày viết

3. Sơ đồ ER

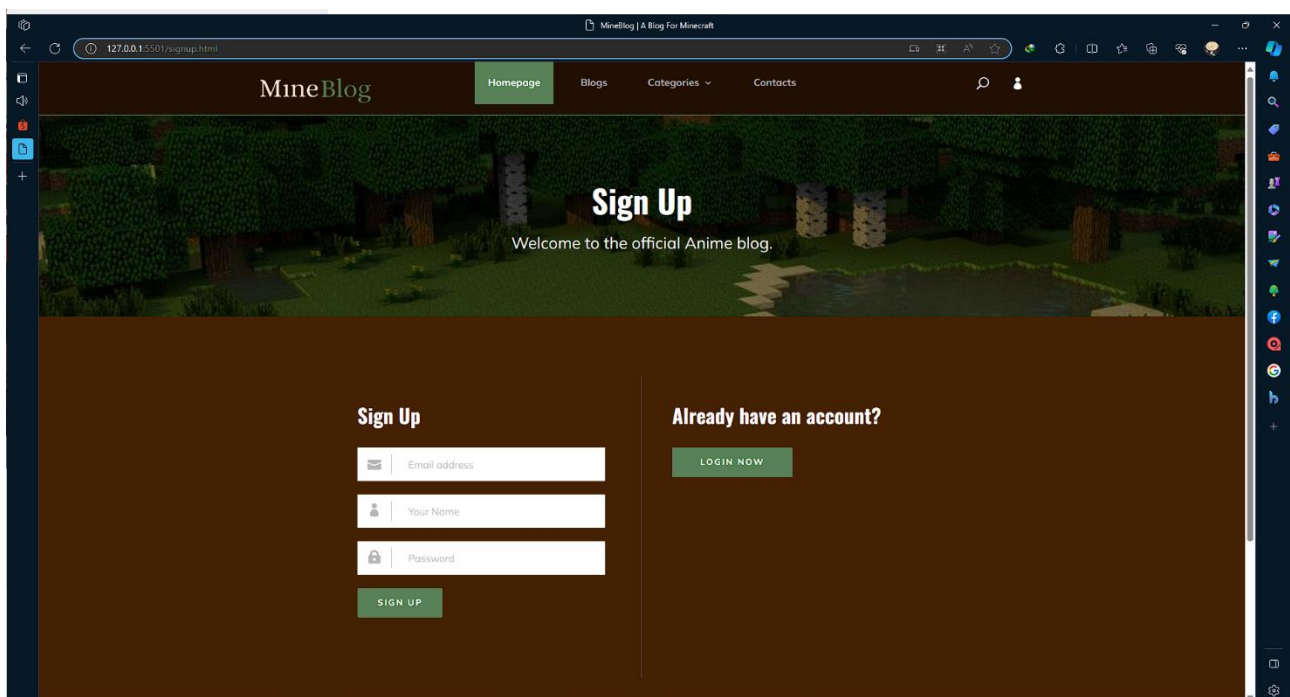
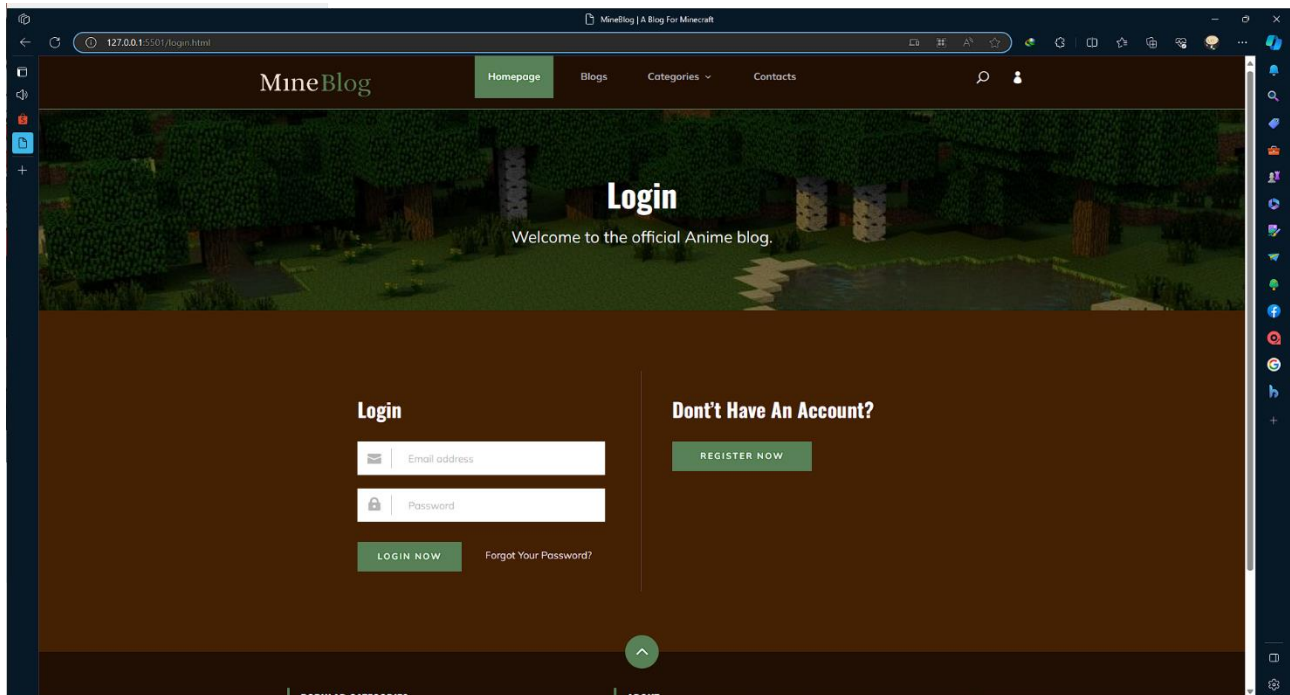


PHẦN IV: GIAO DIỆN SẢN PHẨM SAU LẬP TRÌNH

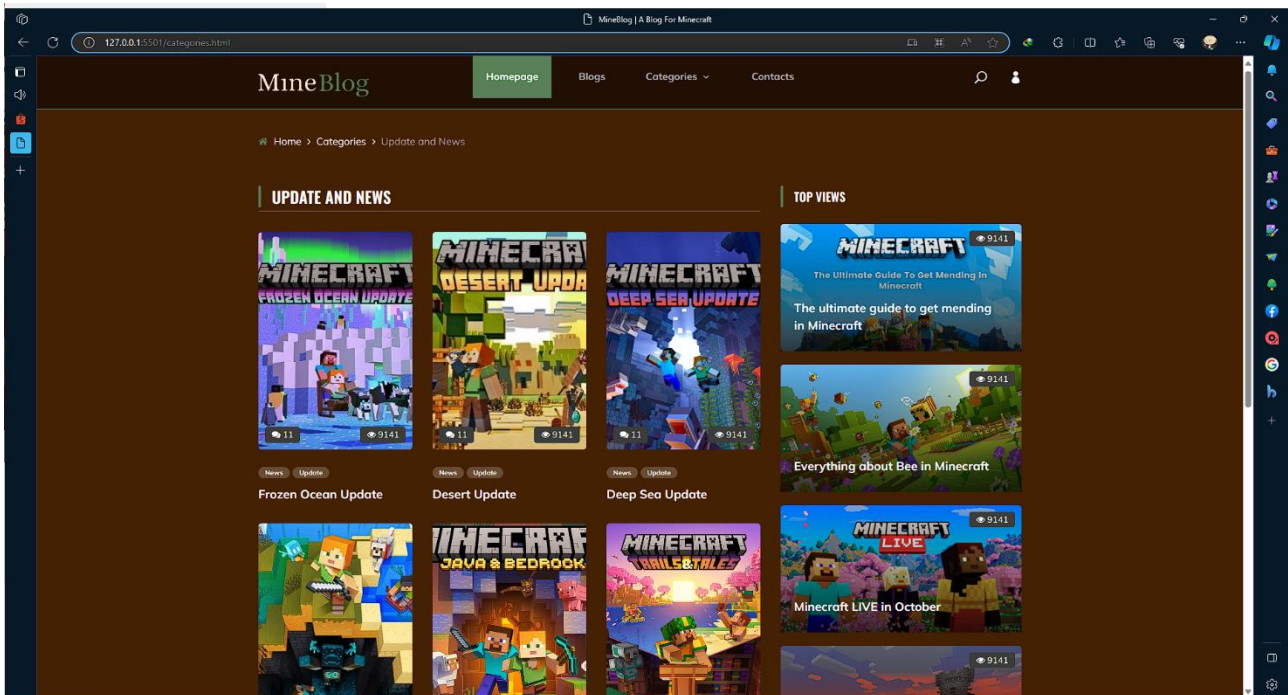
I. Giao diện trang chủ.



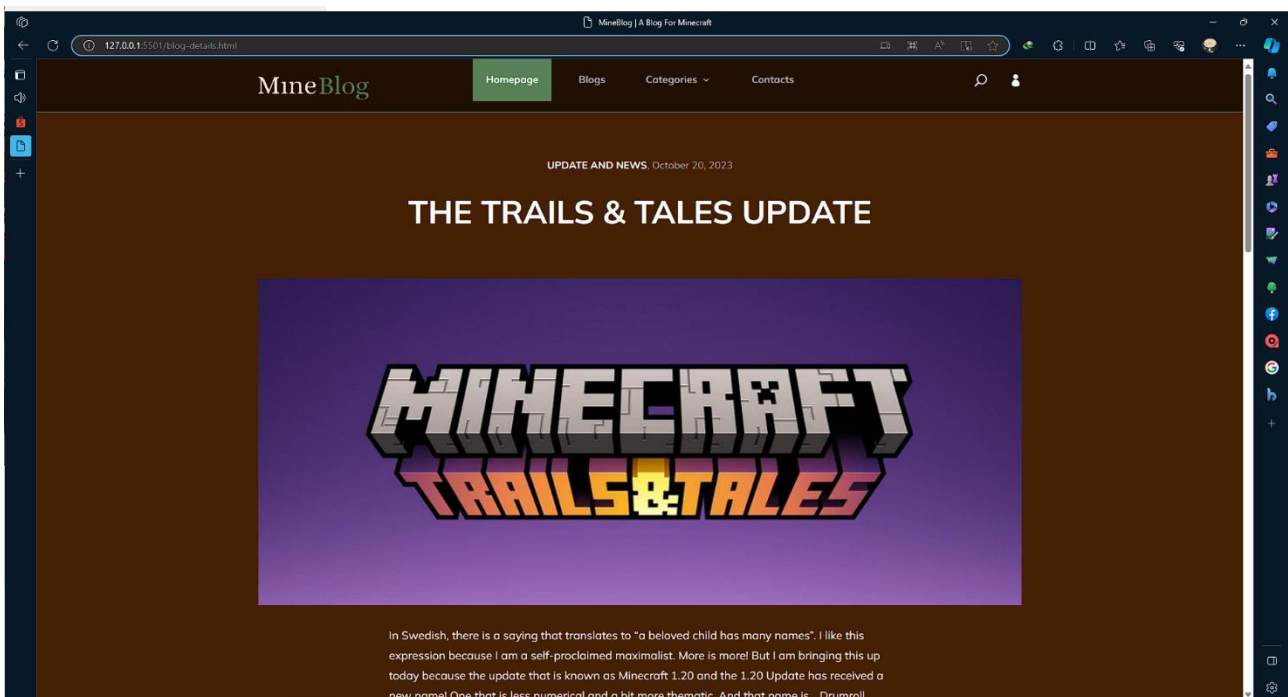
II. Giao diện đăng nhập và đăng kí tài khoản

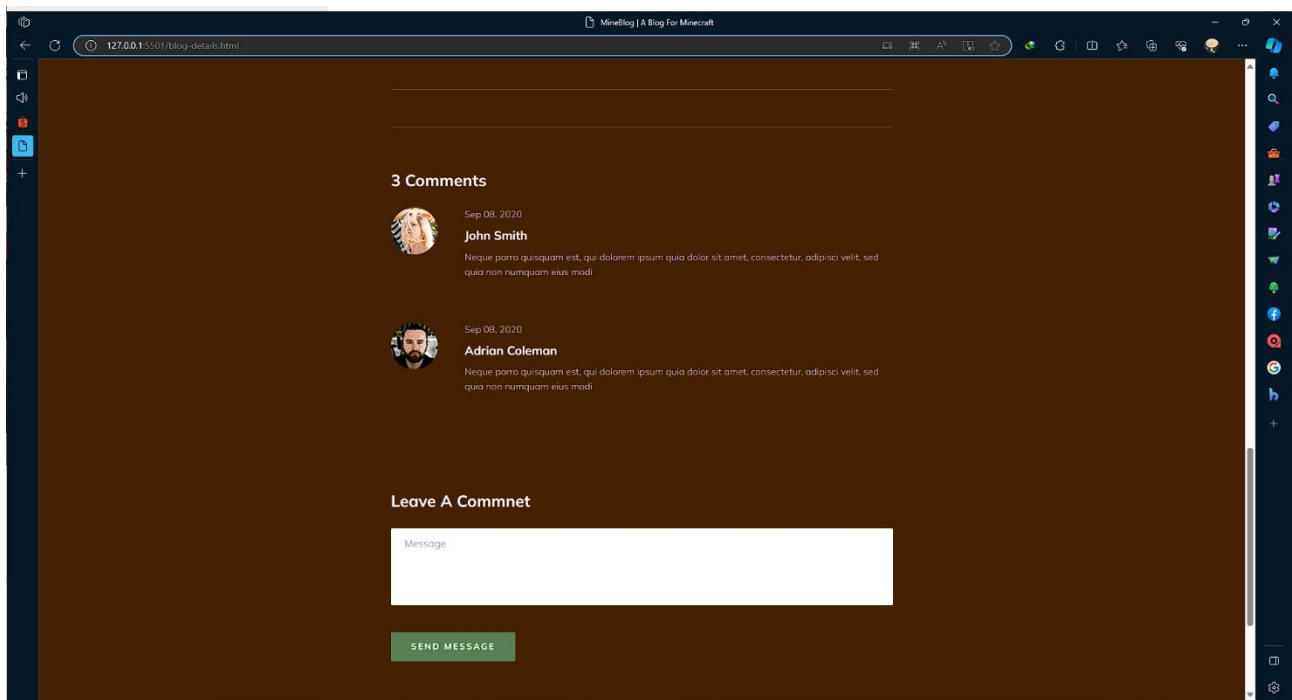


III. Danh mục sản phẩm tìm theo thể loại

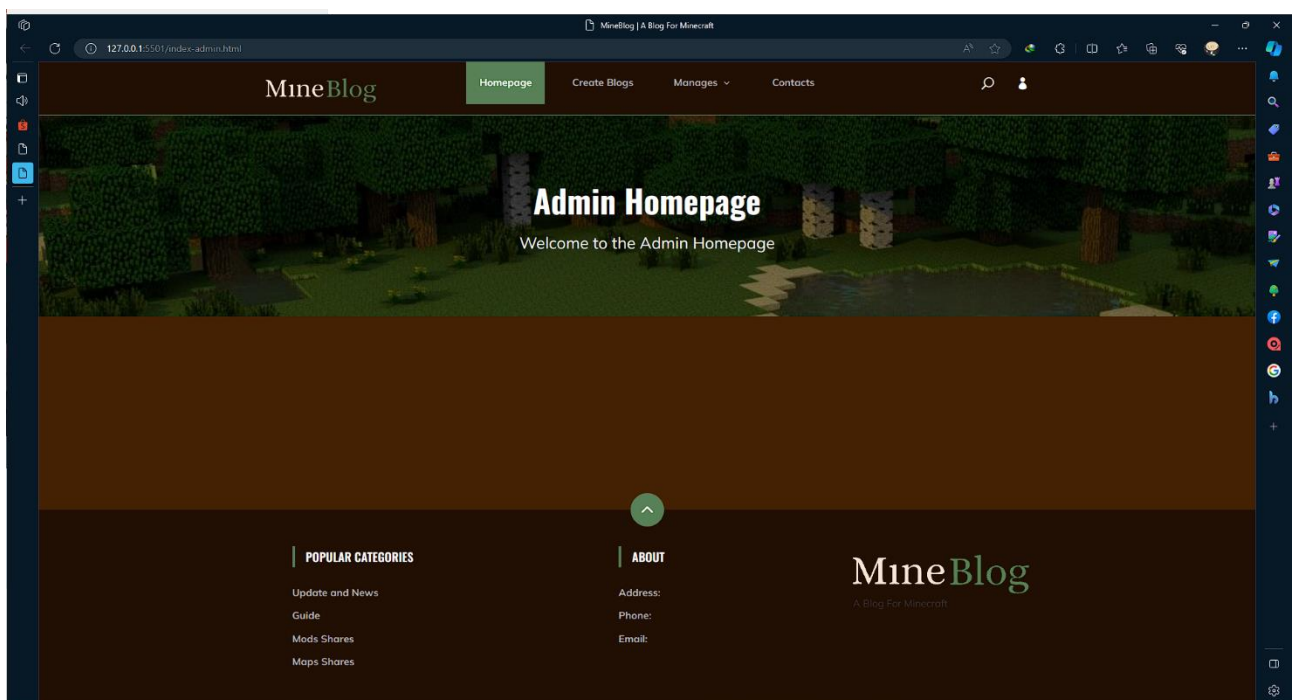


IV. Trang blog và phần bình luận dưới trang blog





V. Giao diện chính trang admin



VI. Giao diện phần viết blog cho admin

The screenshot shows the 'Create Blogs' form in the MineBlog admin interface. The form is set against a background image of a Minecraft forest. It includes fields for 'Blog Title', 'Short Description', and 'Content'. There is an 'Upload Image' section with a 'Choose file' button and a note 'No file chosen'. Below that is a 'Categories' dropdown menu with the text 'Choose your Blog's topic:' and 'Only pick one category'. There is also a 'Tags' section with a list of tags: 'Update', 'News', 'Mods', and 'Maps', with the text 'Only pick one category' below it. A 'Save Blog' button is at the bottom of the form.

Create Blogs
Write your new Blog

Blog Title

Short Description

Content

Upload Image No file chosen
Upload the image file for this blog. Max file size 5 MB

Categories
Only pick one category

Tags
Only pick one category

VII. Giao diện quản lý các bài blog

The screenshot shows the 'Blogs Management' page in the MineBlog admin interface. The page has a header with the MineBlog logo and navigation links: 'Homepage', 'Create Blogs', 'Manages', and 'Contacts'. The main content area features a 'Blogs Management' title and the text 'Manage your Blogs here.'. Below this is a table with columns: ID, Title, Category, Date, and Tool. The table contains three rows of data. At the bottom of the page, there are sections for 'POPULAR CATEGORIES' and 'ABOUT'.

Blogs Management
Manage your Blogs here.

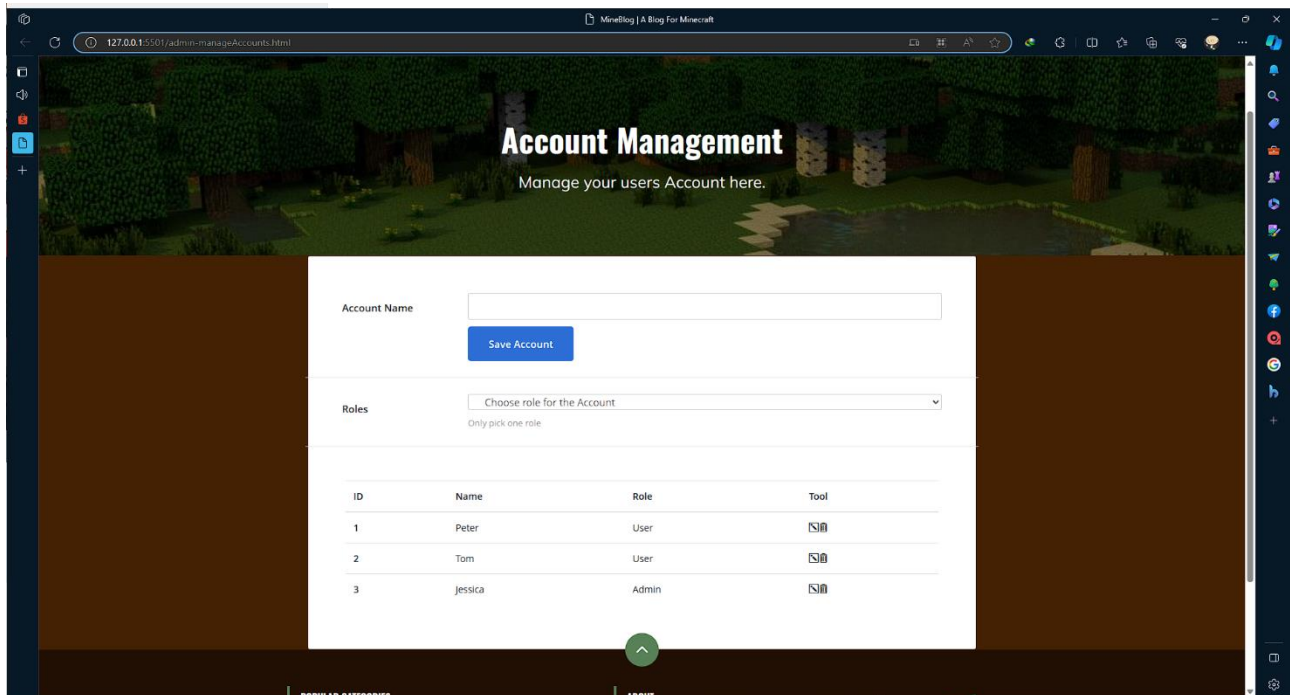
ID	Title	Category	Date	Tool
1	The ultimate guide to get mending in Minecraft	Update and News	Oct. 23, 2023	
2	The ultimate guide to get mending in Minecraft	Mods shares	Oct. 23, 2023	
3	The ultimate guide to get mending in Minecraft	Update and News	Oct. 23, 2023	

POPULAR CATEGORIES
Update and News
Guide
Mods Shares
Maps Shares

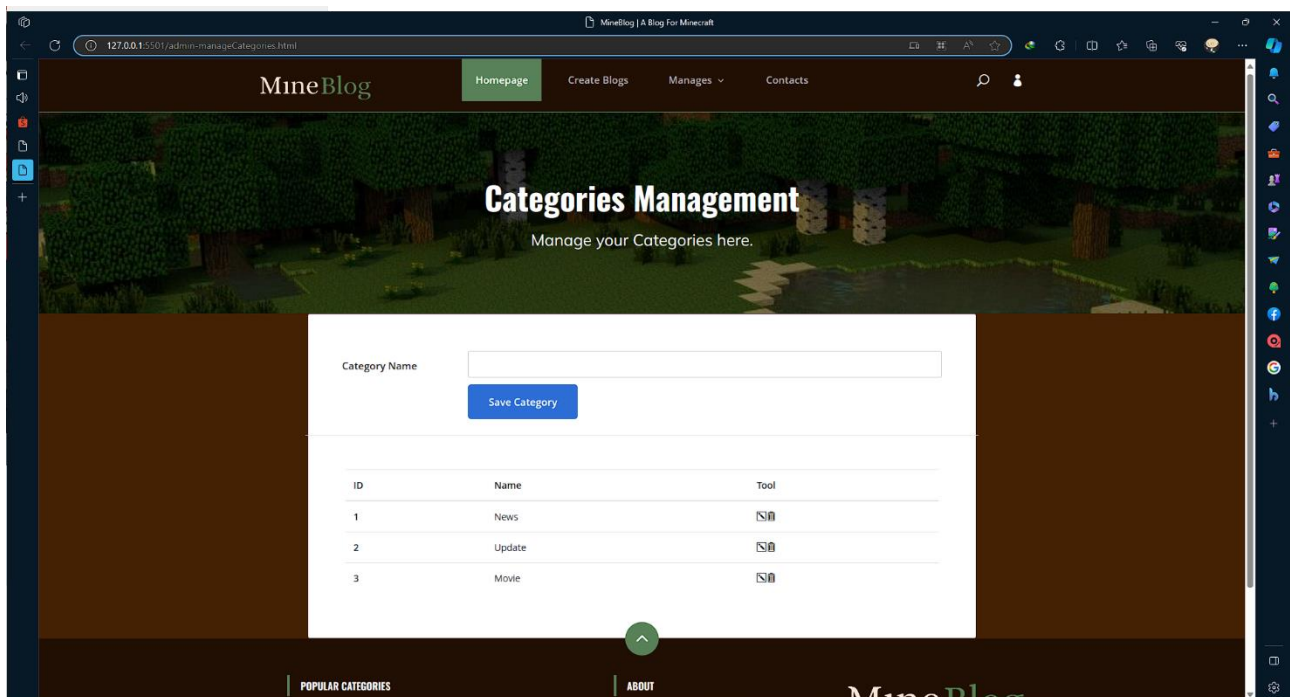
ABOUT
Address:
Phone:
Email:

MineBlog
A Blog For Minecraft

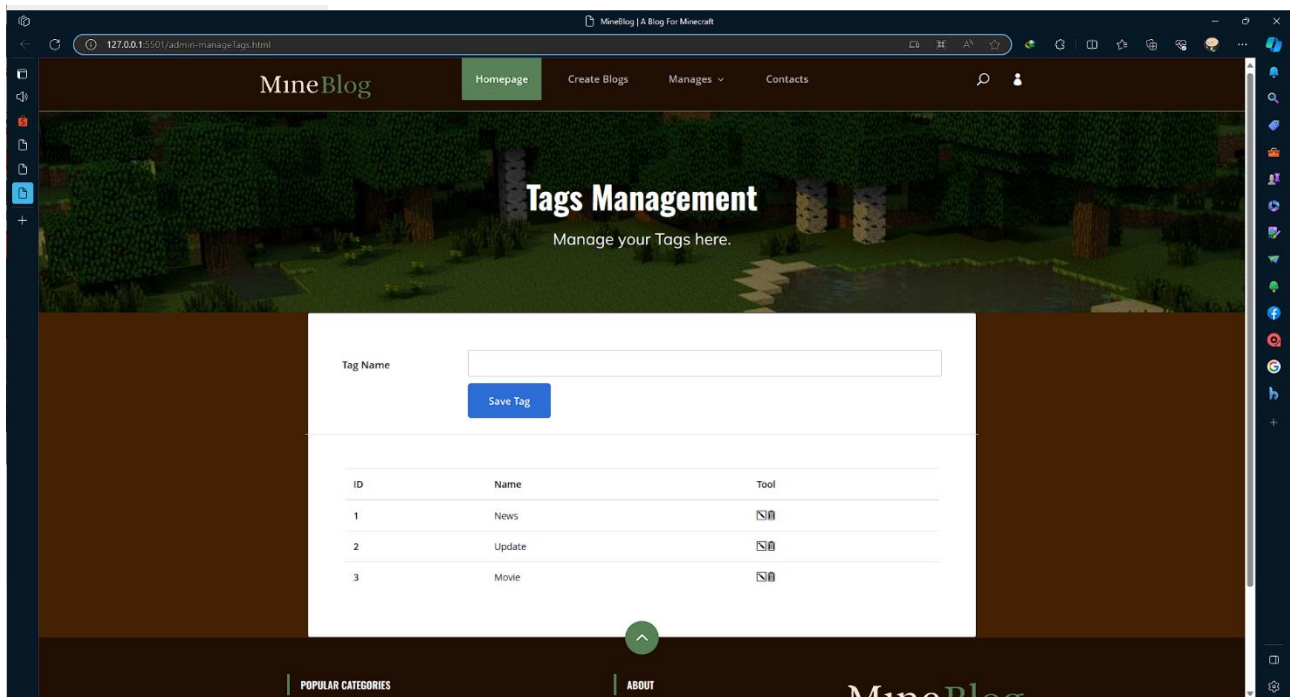
VIII. Giao diện quản lý các tài khoản



IX. Giao diện quản lý các thể loại



X. Giao diện quản lý các tags của bài viết



PHẦN V: SOURCE CODE

I. Properties

1. lauchSetting.json

```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:64344",
      "sslPort": 44316
    }
  },
  "profiles": {
    "http": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "applicationUrl": "http://localhost:5036",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "https": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "applicationUrl": "https://localhost:7127;http://localhost:5036",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

2. serviceDependencies.json

```
{
  "dependencies": {
    "secrets1": {
      "type": "secrets",
      "dynamicId": null
    },
    "mssql1": {
      "type": "mssql",
      "connectionId": "ConnectionStrings:MineBlogDbConnectionString",
      "dynamicId": null
    }
  }
}
```

II. Program.cs

```
using MineBlog.Data;
using Microsoft.EntityFrameworkCore;
```

```

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();
builder.Services.AddDbContext<MineBlogDbContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("MineBlogDbConnectionSt
ring")));

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production
    scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=UHome}/{action=Index}/{id?}");

app.Run();

```

III. Appsettings.json

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "MineBlogDbConnectionString": "Server = BRUH\\SQLEXPRESS; Database = MineBlog;
Trusted_Connection = True; TrustServerCertificate = Yes;"
  }
}

```

IV. Data (DbContext)

```

using Microsoft.EntityFrameworkCore;
using MineBlog.Models;

namespace MineBlog.Data
{
    public class MineBlogDbContext : DbContext
    {
        public MineBlogDbContext(DbContextOptions options) : base(options)
        {
        }
        public DbSet<Blog> Blog { get; set; }
        public DbSet<Tag> Tag { get; set; }
        public DbSet<Category> Category { get; set; }
        public DbSet<Account> Account { get; set; }
        public DbSet<Comment> Comment { get; set; }
        public DbSet<Like> Like { get; set; }
    }
}

```

```
    }
}
```

V. Migrations

1. InitialCreate (First migration)

```
using System;
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace MineBlog.Migrations
{
    /// <inheritdoc />
    public partial class InitialCreate : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Account",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Username = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    Email = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    Password = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    Phone = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    Role = table.Column<string>(type: "nvarchar(max)", nullable:
false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Account", x => x.Id);
                });

            migrationBuilder.CreateTable(
                name: "Category",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Name = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    ImageUrl = table.Column<string>(type: "nvarchar(max)", nullable:
false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Category", x => x.Id);
                });

            migrationBuilder.CreateTable(
                name: "Tag",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
```



```

        Name = table.Column<string>(type: "nvarchar(max)", nullable:
false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Tag", x => x.Id);
    });

migrationBuilder.CreateTable(
    name: "Blog",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Title = table.Column<string>(type: "nvarchar(max)", nullable:
false),
        Description = table.Column<string>(type: "nvarchar(max)",
nullable: false),
        UrlSlug = table.Column<string>(type: "nvarchar(max)", nullable:
false),
        Content = table.Column<string>(type: "nvarchar(max)", nullable:
false),
        ImageUrl = table.Column<string>(type: "nvarchar(max)", nullable:
false),
        Date = table.Column<DateTime>(type: "datetime2", nullable: false),
        CategoryId = table.Column<int>(type: "int", nullable: false),
        AuthorId = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Blog", x => x.Id);
        table.ForeignKey(
            name: "FK_Blog_Account_AuthorId",
            column: x => x.AuthorId,
            principalTable: "Account",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_Blog_Category_CategoryId",
            column: x => x.CategoryId,
            principalTable: "Category",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "BlogTag",
    columns: table => new
    {
        BlogsId = table.Column<int>(type: "int", nullable: false),
        TagsId = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_BlogTag", x => new { x.BlogsId, x.TagsId });
        table.ForeignKey(
            name: "FK_BlogTag_Blog_BlogsId",
            column: x => x.BlogsId,
            principalTable: "Blog",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_BlogTag_Tag_TagsId",
            column: x => x.TagsId,
            principalTable: "Tag",

```

```

        principalColumn: "Id",
        onDelete: ReferentialAction.Cascade);
});

migrationBuilder.CreateTable(
    name: "Comment",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Date = table.Column<DateTime>(type: "datetime2", nullable: false),
        Content = table.Column<string>(type: "nvarchar(500)", maxLength:
500, nullable: false),
        BlogId = table.Column<int>(type: "int", nullable: false),
        AccountId = table.Column<int>(type: "int", nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Comment", x => x.Id);
        table.ForeignKey(
            name: "FK_Comment_Account_AccountId",
            column: x => x.AccountId,
            principalTable: "Account",
            principalColumn: "Id");
        table.ForeignKey(
            name: "FK_Comment_Blog_BlogId",
            column: x => x.BlogId,
            principalTable: "Blog",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "Like",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        BlogId = table.Column<int>(type: "int", nullable: false),
        AccountId = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Like", x => x.Id);
        table.ForeignKey(
            name: "FK_Like_Account_AccountId",
            column: x => x.AccountId,
            principalTable: "Account",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_Like_Blog_BlogId",
            column: x => x.BlogId,
            principalTable: "Blog",
            principalColumn: "Id",
            onDelete: ReferentialAction.Restrict);
    });

migrationBuilder.CreateIndex(
    name: "IX_Blog_AuthorId",
    table: "Blog",
    column: "AuthorId");

migrationBuilder.CreateIndex(
    name: "IX_Blog_CategoryId",

```

```

        table: "Blog",
        column: "CategoryId");

migrationBuilder.CreateIndex(
    name: "IX_BlogTag_TagsId",
    table: "BlogTag",
    column: "TagsId");

migrationBuilder.CreateIndex(
    name: "IX_Comment_AccountId",
    table: "Comment",
    column: "AccountId");

migrationBuilder.CreateIndex(
    name: "IX_Comment_BlogId",
    table: "Comment",
    column: "BlogId");

migrationBuilder.CreateIndex(
    name: "IX_Comment_ParentCommentId",
    table: "Comment",
    column: "ParentCommentId");

migrationBuilder.CreateIndex(
    name: "IX_Like_AccountId",
    table: "Like",
    column: "AccountId");

migrationBuilder.CreateIndex(
    name: "IX_Like_BlogId",
    table: "Like",
    column: "BlogId");
}

/// <inheritdoc />
protected override void Down(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropTable(
        name: "BlogTag");

    migrationBuilder.DropTable(
        name: "Comment");

    migrationBuilder.DropTable(
        name: "Like");

    migrationBuilder.DropTable(
        name: "Tag");

    migrationBuilder.DropTable(
        name: "Blog");

    migrationBuilder.DropTable(
        name: "Account");

    migrationBuilder.DropTable(
        name: "Category");
}
}
}

```

2. DbContextModelSnapshot

```
// <auto-generated />
```

```

using System;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
using MineBlog.Data;

#nullable disable

namespace MineBlog.Migrations
{
    [DbContext(typeof(MineBlogDbContext))]
    partial class MineBlogDbContextModelSnapshot : ModelSnapshot
    {
        protected override void BuildModel(ModelBuilder modelBuilder)
        {
            #pragma warning disable 612, 618

            modelBuilder
                .HasAnnotation("ProductVersion", "7.0.12")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

            modelBuilder.Entity("BlogTag", b =>
            {
                b.Property<int>("BlogsId")
                    .HasColumnType("int");

                b.Property<int>("TagsId")
                    .HasColumnType("int");
            });
        }
    }
}

```

```

        b.HasKey("BlogsId", "TagsId");

        b.HasIndex("TagsId");

        b.ToTable("BlogTag");
    });

```

```

modelBuilder.Entity("MineBlog.Models.Account", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

```

```

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

```

```

        b.Property<string>("Email")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

```

```

        b.Property<string>("Password")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

```

```

        b.Property<string>("Phone")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

```

```

        b.Property<string>("Role")

```

```
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.Property<string>("Username")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.HasKey("Id");
```

```
b.ToTable("Account");  
});
```

```
modelBuilder.Entity("MineBlog.Models.Blog", b =>  
{  
    b.Property<int>("Id")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
b.Property<int>("AuthorId")  
.HasColumnType("int");
```

```
b.Property<int>("CategoryId")  
.HasColumnType("int");
```

```
b.Property<string>("Content")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```

        b.Property<DateTime>("Date")
            .HasColumnType("datetime2");

        b.Property<string>("Description")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.Property<string>("ImageUrl")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.Property<string>("Title")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.Property<string>("UrlSlug")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b.HasKey("Id");

        b.HasIndex("AuthorId");

        b.HasIndex("CategoryId");

        b.ToTable("Blog");
    });

```

```

modelBuilder.Entity("MineBlog.Models.Category", b =>

```

```
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
    b.Property<string>("ImageUrl")
        .IsRequired()
        .HasColumnType("nvarchar(max)");
```

```
    b.Property<string>("Name")
        .IsRequired()
        .HasColumnType("nvarchar(max)");
```

```
    b.HasKey("Id");
```

```
    b.ToTable("Category");
});
```

```
modelBuilder.Entity("MineBlog.Models.Comment", b =>
```

```
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
    b.Property<int?>("AccountId")
```



```

        .HasColumnType("int");

b.Property<int>("BlogId")
    .HasColumnType("int");

b.Property<string>("Content")
    .IsRequired()
    .HasMaxLength(500)
    .HasColumnType("nvarchar(500)");

b.Property<DateTime>("Date")
    .HasColumnType("datetime2");

b.Property<int?>("ParentCommentId")
    .HasColumnType("int");

b.HasKey("Id");

b.HasIndex("AccountId");

b.HasIndex("BlogId");

b.HasIndex("ParentCommentId");

b.ToTable("Comment");
});

modelBuilder.Entity("MineBlog.Models.Like", b =>
{
    b.Property<int>("Id")

```

```
.ValueGeneratedOnAdd()  
.HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
b.Property<int>("AccountId")  
.HasColumnType("int");
```

```
b.Property<int>("BlogId")  
.HasColumnType("int");
```

```
b.HasKey("Id");
```

```
b.HasIndex("AccountId");
```

```
b.HasIndex("BlogId");
```

```
b.ToTable("Like");  
});
```

```
modelBuilder.Entity("MineBlog.Models.Tag", b =>  
{  
    b.Property<int>("Id")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
b.Property<string>("Name")
```

```

        .IsRequired()
        .HasColumnType("nvarchar(max)");

b.HasKey("Id");

b.ToTable("Tag");
});

modelBuilder.Entity("BlogTag", b =>
{
    b.HasOne("MineBlog.Models.Blog", null)
        .WithMany()
        .HasForeignKey("BlogsId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("MineBlog.Models.Tag", null)
        .WithMany()
        .HasForeignKey("TagsId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();
});

modelBuilder.Entity("MineBlog.Models.Blog", b =>
{
    b.HasOne("MineBlog.Models.Account", "Author")
        .WithMany("Blogs")
        .HasForeignKey("AuthorId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();
}

```

```

        b.HasOne("MineBlog.Models.Category", "Category")
            .WithMany("Blogs")
            .HasForeignKey("CategoryId")
            .OnDelete(DeleteBehavior.Cascade)
            .IsRequired();

        b.Navigation("Author");

        b.Navigation("Category");
    });

modelBuilder.Entity("MineBlog.Models.Comment", b =>
{
    b.HasOne("MineBlog.Models.Account", null)
        .WithMany("Comments")
        .HasForeignKey("AccountId");

    b.HasOne("MineBlog.Models.Blog", "Blog")
        .WithMany("Comments")
        .HasForeignKey("BlogId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("MineBlog.Models.Comment", "ParentComment")
        .WithMany()
        .HasForeignKey("ParentCommentId");

    b.Navigation("Blog");
}

```

```
        b.Navigation("ParentComment");
    });
```

```
modelBuilder.Entity("MineBlog.Models.Like", b =>
{
    b.HasOne("MineBlog.Models.Account", "Account")
        .WithMany("Likes")
        .HasForeignKey("AccountId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("MineBlog.Models.Blog", "Blog")
        .WithMany("Likes")
        .HasForeignKey("BlogId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Account");

    b.Navigation("Blog");
});
```

```
modelBuilder.Entity("MineBlog.Models.Account", b =>
{
    b.Navigation("Blogs");

    b.Navigation("Comments");

    b.Navigation("Likes");
});
```

```

modelBuilder.Entity("MineBlog.Models.Blog", b =>
{
    b.Navigation("Comments");

    b.Navigation("Likes");
});

modelBuilder.Entity("MineBlog.Models.Category", b =>
{
    b.Navigation("Blogs");
});

#pragma warning restore 612, 618
}
}
}

```

VI. CSS

1. Style.css (Giao diện chung)

```

.dark-background {
    position: relative;

    /* Other styles for your container */

    /* Set the background image */
    background-image: url('@ViewBag.CategoryURL');
    background-size: cover;
    background-position: center;
}

.dark-background::before {
    content: "";
    position: absolute;

```

```

        top: 0;
        right: 0;
        bottom: 0;
        left: 0;
        background: rgba(0, 0, 0, 0.5); /* Adjust the last value (0.5) for the darkness
level */
        /* You can also use hex notation for the color, e.g., #00000080 for 50%
opacity */
    }
html,
body {
    height: 100%;
    font-family: "Mulish", sans-serif;
    -webkit-font-smoothing: antialiased;
    background: #452103;
}

h1,
h2,
h3,
h4,
h5,
h6 {
    margin: 0;
    color: #111111;
    font-weight: 400;
    font-family: "Mulish", sans-serif;
}

h1 {
    font-size: 70px;

```

```
}
```

```
h2 {  
    font-size: 36px;  
}
```

```
h3 {  
    font-size: 30px;  
}
```

```
h4 {  
    font-size: 24px;  
}
```

```
h5 {  
    font-size: 18px;  
}
```

```
h6 {  
    font-size: 16px;  
}
```

```
p {  
    font-size: 15px;  
    font-family: "Mulish", sans-serif;  
    color: #3d3d3d;  
    font-weight: 400;  
    line-height: 25px;  
    margin: 0 0 15px 0;  
}
```



```

img {
    max-width: 100%;
}

input:focus,
select:focus,
button:focus,
textarea:focus {
    outline: none;
}

a:hover {
    text-decoration: none;
    outline: none;
    color: white;
}

a:focus {
    text-decoration: none;
    outline: none;
    color: #412102;
}

ul,
ol {
    padding: 0;
    margin: 0;
}

/*-----

```

Helper CSS

-----*/

```
.section-title {  
    margin-bottom: 30px;  
}
```

```
.section-title h4,  
.section-title h5 {  
    color: #ffffff;  
    font-weight: 600;  
    line-height: 21px;  
    text-transform: uppercase;  
    padding-left: 20px;  
    position: relative;  
    font-family: "Oswald", sans-serif;  
}
```

```
.section-title h4:after,  
.section-title h5:after {  
    position: absolute;  
    left: 0;  
    top: -6px;  
    height: 32px;  
    width: 4px;  
    background: #588157;  
    content: "";  
}
```

```
.set-bg {  
    background-repeat: no-repeat;
```

```
background-size: cover;
background-position: top center;
}
```

```
.spad {
padding-top: 100px;
padding-bottom: 100px;
}
```

```
.text-white h1,
.text-white h2,
.text-white h3,
.text-white h4,
.text-white h5,
.text-white h6,
.text-white p,
.text-white span,
.text-white li,
.text-white a {
color: #fff;
}
```

```
/* buttons */
```

```
.primary-btn {
display: inline-block;
font-size: 13px;
font-weight: 700;
text-transform: uppercase;
color: #ffffff;
```

```
        letter-spacing: 2px;
    }
```

```
.primary-btn span {
    font-size: 18px;
    margin-left: 5px;
    position: relative;
    top: 3px;
}
```

```
.site-btn {
    font-size: 13px;
    color: #ffffff;
    background: #588157;
    font-weight: 700;
    border: none;
    border-radius: 2px;
    letter-spacing: 2px;
    text-transform: uppercase;
    display: inline-block;
    padding: 12px 30px;
}
```

```
/* Preloder */
```

```
#preloder {
    position: fixed;
    width: 100%;
    height: 100%;
    top: 0;
```

```

    left: 0;
    z-index: 999999;
    background: #000;
}

.loader {
    width: 40px;
    height: 40px;
    position: absolute;
    top: 50%;
    left: 50%;
    margin-top: -13px;
    margin-left: -13px;
    border-radius: 60px;
    animation: loader 0.8s linear infinite;
    -webkit-animation: loader 0.8s linear infinite;
}

@keyframes loader {
    0% {
        -webkit-transform: rotate(0deg);
        transform: rotate(0deg);
        border: 4px solid #f44336;
        border-left-color: transparent;
    }
    50% {
        -webkit-transform: rotate(180deg);
        transform: rotate(180deg);
        border: 4px solid #452103;
        border-left-color: transparent;
    }
}

```

```

    }
    100% {
        -webkit-transform: rotate(360deg);
        transform: rotate(360deg);
        border: 4px solid #f44336;
        border-left-color: transparent;
    }
}

```

```

@-webkit-keyframes loader {
    0% {
        -webkit-transform: rotate(0deg);
        border: 4px solid #f44336;
        border-left-color: transparent;
    }
    50% {
        -webkit-transform: rotate(180deg);
        border: 4px solid #452103;
        border-left-color: transparent;
    }
    100% {
        -webkit-transform: rotate(360deg);
        border: 4px solid #f44336;
        border-left-color: transparent;
    }
}

```

```

.spacial-controls {
    position: fixed;
    width: 111px;

```

```

height: 91px;
top: 0;
right: 0;
z-index: 999;
}

.spacial-controls .search-switch {
    display: block;
    height: 100%;
    padding-top: 30px;
    background: #452103;
    text-align: center;
    cursor: pointer;
}

.search-model {
    display: none;
    position: fixed;
    width: 100%;
    height: 100%;
    left: 0;
    top: 0;
    background: #000;
    z-index: 99999;
}

.search-model-form {
    padding: 0 15px;
}

```

```
.search-model-form input {  
    width: 500px;  
    font-size: 40px;  
    border: none;  
    border-bottom: 2px solid #333;  
    background: 0 0;  
    color: #999;  
}
```

```
.search-close-switch {  
    position: absolute;  
    width: 50px;  
    height: 50px;  
    background: #333;  
    color: #fff;  
    text-align: center;  
    border-radius: 50%;  
    font-size: 28px;  
    line-height: 28px;  
    top: 30px;  
    cursor: pointer;  
    display: -webkit-box;  
    display: -ms-flexbox;  
    display: flex;  
    -webkit-box-align: center;  
    -ms-flex-align: center;  
    align-items: center;  
    -webkit-box-pack: center;  
    -ms-flex-pack: center;  
    justify-content: center;
```



```
}
```

```
/*-----
```

```
Header
```

```
-----*/
```

```
.header {
```

```
    background: #210f04;
```

```
    border-bottom: solid #588157 1px;
```

```
}
```

```
.header__logo {
```

```
    padding: 20px 0 17px;
```

```
}
```

```
.header__logo a {
```

```
    display: inline-block;
```

```
}
```

```
.header__menu {
```

```
    text-align: center;
```

```
}
```

```
.header__menu ul li {
```

```
    list-style: none;
```

```
    display: inline-block;
```

```
    position: relative;
```

```
    margin-right: 16px;
```

```
}
```

```
.header__menu ul li.active a {  
    background: #588157;  
    color: #ffffff;  
}
```

```
.header__menu ul li:hover a {  
    color: #ffffff;  
}
```

```
.header__menu ul li:hover .dropdown {  
    top: 62px;  
    opacity: 1;  
    visibility: visible;  
}
```

```
.header__menu ul li:hover .dropdown li a {  
    background: transparent;  
}
```

```
.header__menu ul li:hover .dropdown li a:hover {  
    background: #588157;  
    color: white;  
}
```

```
.header__menu ul li:last-child {  
    margin-right: 0;  
}
```

```
.header__menu ul li .dropdown {  
    position: absolute;  
    left: 0;  
    top: 82px;
```

```
width: 150px;
background: #ffffff;
text-align: left;
padding: 5px 0;
z-index: 9;
opacity: 0;
visibility: hidden;
-webkit-transition: all, 0.3s;
-o-transition: all, 0.3s;
transition: all, 0.3s;
}
```

```
.header__menu ul li .dropdown li {
    display: block;
    margin-right: 0;
}
```

```
.header__menu ul li .dropdown li a {
    font-size: 14px;
    color: #111111;
    font-weight: 500;
    padding: 5px 20px;
}
```

```
.header__menu ul li a {
    font-size: 15px;
    color: #b7b7b7;
    display: block;
    font-weight: 700;
    -webkit-transition: all, 0.5s;
```

```
-o-transition: all, 0.5s;
transition: all, 0.5s;
padding: 20px;
}
```

```
.header__menu ul li a span {
    position: relative;
    font-size: 17px;
    top: 2px;
}
```

```
.header__right {
    text-align: right;
    padding: 20px 0 15px;
}
```

```
.header__right a {
    display: inline-block;
    font-size: 18px;
    color: #ffffff;
    margin-right: 30px;
}
```

```
.header__right a:last-child {
    margin-right: 0;
}
```

```
.header__right a:hover {
    text-decoration: none;
}
```

```
.slicknav_menu {
    display: none;
}
```

```
/*-----
Hero
-----*/
```

```
.hero {
    padding-top: 50px;
}
```

```
.hero__items {
    padding: 250px 0 42px 50px;
    border-radius: 5px;
    height: 600px;
    position: relative;
    overflow: hidden; /* Ensure that the gradient doesn't extend outside the element */
}
```

```
.hero__items::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%; /* Adjust the width to control how much of the image is darkened */
    height: 100%;
    background: linear-gradient(to bottom, rgba(0,0,0,0) 20%, rgba(0,0,0,1) 100%); /* Adjust
the color and opacity as needed */
    z-index: 1; /* Place it above the image */
}
```

```
}
```

```
.hero__text {  
    position: relative;  
    z-index: 9;  
}
```

```
.hero__text .label {  
    font-size: 13px;  
    color: #588157;  
    background: #ffffff;  
    padding: 5px 14px 3px;  
    display: inline-block;  
    position: relative;  
    top: -100px;  
    opacity: 0;  
    -webkit-transition: all, 0.2s;  
    -o-transition: all, 0.2s;  
    transition: all, 0.2s;  
}
```

```
.hero__text h2 {  
    color: #ffffff;  
    font-size: 42px;  
    font-family: "Oswald", sans-serif;  
    font-weight: 700;  
    line-height: 52px;  
    margin-top: 35px;  
    margin-bottom: 8px;  
    position: relative;
```

```
    top: -100px;
    opacity: 0;
    -webkit-transition: all, 0.4s;
    -o-transition: all, 0.4s;
    transition: all, 0.4s;
}
```

```
.hero__text p {
    color: #ffffff;
    font-size: 16px;
    margin-bottom: 40px;
    position: relative;
    top: -100px;
    opacity: 0;
    -webkit-transition: all, 0.6s;
    -o-transition: all, 0.6s;
    transition: all, 0.6s;
}
```

```
.hero__text a {
    position: relative;
    top: -100px;
    opacity: 0;
    -webkit-transition: all, 0.8s;
    -o-transition: all, 0.8s;
    transition: all, 0.8s;
}
```

```
.hero__text a span {
    font-size: 13px;
```

```
color: #ffffff;
background: #588157;
display: inline-block;
font-weight: 700;
letter-spacing: 2px;
text-transform: uppercase;
padding: 14px 20px;
border-radius: 4px 0 0 4px;
margin-right: 1px;
}
```

```
.hero__text a i {
    font-size: 20px;
    display: inline-block;
    background: #588157;
    padding: 11px 5px 16px 8px;
    color: #ffffff;
    border-radius: 0 4px 4px 0;
}
```

```
.hero__slider.owl-carousel .owl-item.active .hero__text .label {
    top: 0;
    opacity: 1;
}
```

```
.hero__slider.owl-carousel .owl-item.active .hero__text h2 {
    top: 0;
    opacity: 1;
}
```



```
.hero__slider.owl-carousel .owl-item.active .hero__text p {  
    top: 0;  
    opacity: 1;  
}
```

```
.hero__slider.owl-carousel .owl-item.active .hero__text a {  
    top: 0;  
    opacity: 1;  
}
```

```
.hero__slider.owl-carousel .owl-dots {  
    position: absolute;  
    left: 0;  
    bottom: 10px;  
    width: 100%;  
    text-align: center;  
}
```

```
.hero__slider.owl-carousel .owl-dots button {  
    height: 8px;  
    width: 8px;  
    background: #b7b7b7;  
    border-radius: 50%;  
    margin-right: 10px;  
}
```

```
.hero__slider.owl-carousel .owl-dots button.active {  
    background: #ffffff;  
}
```

```
.hero__slider.owl-carousel .owl-dots button:last-child {  
    margin-right: 0;  
}
```

```
.hero__slider.owl-carousel .owl-nav button {  
    font-size: 36px;  
    height: 66px;  
    width: 66px;  
    background: #452103;  
    line-height: 66px;  
    text-align: center;  
    color: #ffffff;  
    -webkit-transform: rotate(45deg);  
    -ms-transform: rotate(45deg);  
    transform: rotate(45deg);  
    position: relative;  
    z-index: 1;  
    position: absolute;  
    left: -33px;  
    top: 50%;  
    margin-top: -47px;  
}
```

```
.hero__slider.owl-carousel .owl-nav button.owl-next {  
    left: auto;  
    right: -33px;  
}
```

```
.hero__slider.owl-carousel .owl-nav button:after {  
    position: absolute;
```

```

    top: 6px;
    left: 0;
    right: 0;
    height: 54px;
    width: 54px;
    background: rgba(101, 145, 88, 0.548);
    content: "";
    z-index: -1;
    margin: 0 auto;
}

```

```

.hero__slider.owl-carousel .owl-nav button span {
    -webkit-transform: rotate(-45deg);
    -ms-transform: rotate(-45deg);
    transform: rotate(-45deg);
    display: block;
    z-index: 1;
}

```

```

/*-----
Product
-----*/

```

```

.product {
    padding-bottom: 60px;
    padding-top: 80px;
}

```

```

.product-page {
    padding-top: 60px;
}

```

```
}
```

```
.btn__all {  
    text-align: right;  
    margin-bottom: 30px;  
}
```

```
.trending__product {  
    margin-bottom: 50px;  
}
```

```
.popular__product {  
    margin-bottom: 50px;  
}
```

```
.recent__product {  
    margin-bottom: 50px;  
}
```

```
.product__item {  
    margin-bottom: 30px;  
}
```

```
.product__item__pic {  
    height: 325px;  
    position: relative;  
    border-radius: 5px;  
}
```

```
.product__item__pic .comment {
```

```
font-size: 13px;
color: #ffffff;
background: #3d3d3d;
display: inline-block;
padding: 2px 10px;
border-radius: 4px;
position: absolute;
left: 10px;
bottom: 10px;
}
```

```
.product__item__pic .view {
    font-size: 13px;
    color: #ffffff;
    background: #3d3d3d;
    display: inline-block;
    padding: 2px 10px;
    border-radius: 4px;
    position: absolute;
    right: 10px;
    bottom: 10px;
}
```

```
.product__item__text {
    padding-top: 20px;
}
```

```
.product__item__text ul {
    margin-bottom: 10px;
}
```

```
.product__item__text ul li {  
    list-style: none;  
    font-size: 15px;  
    color: #ffffff;  
    font-weight: 700;  
    padding: 1px 10px;  
    background: rgba(255, 255, 255, 0.2);  
    border-radius: 50px;  
    display: inline-block;  
}
```

```
.product__item__text h3 a {  
    color: #ffffff;  
    font-weight: 700;  
    line-height: 26px;  
}
```

```
.product__sidebar .section-title h5 {  
    color: #ffffff;  
    font-weight: 600;  
    font-family: "Oswald", sans-serif;  
    line-height: 21px;  
    text-transform: uppercase;  
    padding-left: 20px;  
    position: relative;  
}
```

```
.product__sidebar .section-title h5:after {  
    position: absolute;  
    left: 0;
```

```

    top: -6px;
    height: 32px;
    width: 4px;
    background: #588157;
    content: "";
}

.product__sidebar__view {
    position: relative;
    margin-bottom: 80px;
}

.product__sidebar__view__item {
    height: 190px;
    position: relative;
    border-radius: 5px;
    margin-bottom: 20px;
}

.product__sidebar__view__item::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%; /* Adjust the width to control how much of the image is darkened */
    height: 100%;
    background: linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.75)); /* Adjust the color and
opacity as needed */
    z-index: 1; /* Place it above the image */
}

```

```
.product__sidebar__view__item .view {  
    font-size: 13px;  
    color: #ffffff;  
    background: #3d3d3d;  
    display: inline-block;  
    padding: 2px 10px;  
    border-radius: 4px;  
    position: absolute;  
    right: 10px;  
    top: 10px;  
    z-index: 2; /* Place it above the image */  
}
```

```
.product__sidebar__view__item h5 {  
    position: absolute;  
    left: 0;  
    bottom: 25px;  
    width: 100%;  
    padding: 0 30px 0 20px;  
    z-index: 2; /* Place it above the image */  
}
```

```
.product__sidebar__view__item h5 a {  
    color: #ffffff;  
    font-weight: 700;  
    line-height: 26px;  
    z-index: 2; /* Place it above the image */  
}
```

```
.product__page__title {
```



```

border-bottom: 2px solid rgba(255, 255, 255, 0.2);
padding-bottom: 10px;
margin-bottom: 30px;
}

.product__page__title .section-title {
    margin-bottom: 0;
}

.product__page__title .product__page__filter {
    text-align: right;
}

.product__page__title .product__page__filter p {
    color: #ffffff;
    display: inline-block;
    margin-bottom: 0;
    margin-right: 16px;
}

.product__page__title .product__page__filter .nice-select {
    float: none;
    display: inline-block;
    font-size: 15px;
    color: #3d3d3d;
    font-weight: 700;
    border-radius: 0;
    padding-left: 15px;
    padding-right: 40px;
    height: 32px;

```

```

        line-height: 32px;
    }

.product__page__title .product__page__filter .nice-select:after {
    border-bottom: 2px solid #111;
    border-right: 2px solid #111;
    height: 8px;
    top: 47%;
    width: 8px;
    right: 15px;
}

.product__page__title .product__page__filter .nice-select .list {
    margin-top: 0;
    border-radius: 0;
}

.product__pagination {
    padding-top: 15px;
}

.product__pagination a {
    display: inline-block;
    font-size: 15px;
    color: #b7b7b7;
    font-weight: 600;
    height: 50px;
    width: 50px;
    border: 1px solid transparent;
    border-radius: 50%;

```

```

        line-height: 48px;
        text-align: center;
        -webkit-transition: all, 0.3s;
        -o-transition: all, 0.3s;
        transition: all, 0.3s;
    }

```

```

.product__pagination a:hover {
    color: #ffffff;
}

```

```

.product__pagination a.current-page {
    border: 1px solid #ffffff;
}

```

```

.product__pagination a i {
    color: #b7b7b7;
    font-size: 15px;
}

```

```

/*-----
Breadcrumb
-----*/

```

```

.breadcrumb-option {
    padding-top: 35px;
}

```

```

.breadcrumb__links a {
    font-size: 15px;
}

```

```

        color: #ffffff;
        margin-right: 18px;
        display: inline-block;
        position: relative;
    }

.breadcrumb__links a i {
    margin-right: 5px;
    color: #588157;
}

.breadcrumb__links a:after {
    position: absolute;
    right: -14px;
    top: 0;
    content: "□";
    font-family: "FontAwesome";
}

.breadcrumb__links span {
    font-size: 15px;
    color: #b7b7b7;
    display: inline-block;
}

/*-----
    Normal Breadcrumb
-----*/

.normal-breadcrumb {

```

```

height: 300px;
display: -webkit-box;
display: -ms-flexbox;
display: flex;
-webkit-box-align: center;
-ms-flex-align: center;
align-items: center;
-webkit-box-pack: center;
-ms-flex-pack: center;
justify-content: center;
}

```

```

.normal__breadcrumb__text h2 {
    color: #ffffff;
    font-size: 48px;
    font-family: "Oswald", sans-serif;
    font-weight: 700;
    margin-bottom: 22px;
}

```

```

.normal__breadcrumb__text p {
    color: #ffffff;
    font-size: 24px;
    margin-bottom: 0;
}

```

```

/*-----
    Blog
-----*/

```

```
.blog {  
    padding-top: 70px;  
    padding-bottom: 90px;  
}
```

```
.blog__item {  
    height: 580px;  
    position: relative;  
    margin-left: -10px;  
    margin-right: -10px;  
    margin-bottom: 10px;  
}
```

```
.blog__item.small__item {  
    height: 285px;  
}
```

```
.blog__item.small__item .blog__item__text {  
    padding: 0 30px;  
}
```

```
.blog__item.small__item .blog__item__text p {  
    margin-bottom: 5px;  
}
```

```
.blog__item.small__item .blog__item__text h4 a {  
    font-size: 20px;  
    line-height: 30px;  
}
```

```
.blog__item__text {  
    position: absolute;  
    left: 0;  
    bottom: 25px;  
    text-align: center;  
    width: 100%;  
    padding: 0 105px;  
}
```

```
.blog__item__text p {  
    color: #ffffff;  
    margin-bottom: 12px;  
}
```

```
.blog__item__text p span {  
    color: #588157;  
    margin-right: 5px;  
}
```

```
.blog__item__text h4 a {  
    color: #ffffff;  
    line-height: 34px;  
}
```

```
/*-----  
  
    Blog Details  
-----*/
```

```
.blog-details {  
    padding-top: 70px;
```

```
}
```

```
.blog__details__title {  
    text-align: center;  
    margin-bottom: 70px;  
}
```

```
.blog__details__title h6 {  
    font-size: 15px;  
    color: #ffffff;  
    font-weight: 700;  
    text-transform: uppercase;  
    margin-bottom: 30px;  
}
```

```
.blog__details__title h6 span {  
    color: #b7b7b7;  
    font-weight: 400;  
    text-transform: none;  
}
```

```
.blog__details__title h2 {  
    color: #ffffff;  
    font-size: 48px;  
    font-weight: 700;  
    line-height: 60px;  
    margin-bottom: 38px;  
}
```

```
.blog__details__title .blog__details__social a {
```



```

display: inline-block;
font-size: 15px;
color: #ffffff;
padding: 16px 35px 14px 20px;
border-radius: 2px;
margin-right: 6px;
}

.blog__details__title .blog__details__social a i {
margin-right: 6px;
}

.blog__details__pic {
text-align: center;
}

.blog__details__pic img {
display: inline-block;
}

.blog__details__pic__mini {
margin-bottom: 30px;
}

.blog__details__pic__mini img {
width: 100%;
}

.blog__details__text {
margin-bottom: 40px;
}

.blog__details__text p {

```

```

        color: #ffffff;
        font-size: 17px;
        line-height: 30px;
    }
.blog__details__text h2 {
    color: #ffffff;
    margin-bottom: 5px;
    margin-top: 20px;
    font-weight: bold;
}

.blog__details__item__text {
    margin-bottom: 42px;
}

.blog__details__item__text h4 {
    color: #ffffff;
    font-weight: 700;
    margin-bottom: 35px;
}

.blog__details__item__text img {
    min-width: 100%;
    margin-bottom: 26px;
}

.blog__details__item__text p {
    color: #ffffff;
    font-size: 17px;
    line-height: 30px;

```

```

    margin-bottom: 0;
}
.blog__details__tags {
    border-top: 1px solid rgba(255, 255, 255, 0.2);
    border-bottom: 1px solid rgba(255, 255, 255, 0.2);
    padding: 20px 0 15px;
    margin-bottom: 60px;
}
.blog__details__tags__one {
    border-bottom: 1px solid rgba(255, 255, 255, 0.2);
    padding: 20px 0 15px;
    margin-bottom: 60px;
}
.blog__details__tags__one a {
    display: inline-block;
    color: #b7b7b7;
    font-size: 12px;
    font-weight: 700;
    letter-spacing: 2px;
    text-transform: uppercase;
    background: rgba(255, 255, 255, 0.1);
    border-radius: 2px;
    margin-right: 6px;
    padding: 6px 15px;
    margin-bottom: 10px;
}

.blog__details__tags__one a:last-child {
    margin-right: 0;
}

```

```
.blog__details__tags a {
    display: inline-block;
    color: #b7b7b7;
    font-size: 12px;
    font-weight: 700;
    letter-spacing: 2px;
    text-transform: uppercase;
    background: rgba(255, 255, 255, 0.1);
    border-radius: 2px;
    margin-right: 6px;
    padding: 6px 15px;
    margin-bottom: 10px;
}
```

```
.blog__details__tags a:last-child {
    margin-right: 0;
}
```

```
.blog__details__btns {
    border-top: 1px solid rgba(255, 255, 255, 0.2);
    border-bottom: 1px solid rgba(255, 255, 255, 0.2);
    padding: 20px 0 15px;
    margin-bottom: 65px;
}
```

```
.blog__details__btns__item {
    margin-bottom: 20px;
}
```

```
.blog__details__btns__item.next__btn {
```

```

        text-align: right;
    }

.blog__details__btns__item h5 a {
    font-size: 17px;
    letter-spacing: 2px;
    color: #ffffff;
}

.blog__details__btns__item h5 a span {
    font-size: 30px;
    color: #b7b7b7;
    position: relative;
    top: 8px;
}

.blog__details__comment {
    border-bottom: 1px solid rgba(255, 255, 255, 0.1);
    padding-bottom: 30px;
}

.blog__details__comment h4 {
    color: #ffffff;
    font-weight: 700;
    margin-bottom: 26px;
}

.blog__details__comment__item {
    margin-bottom: 40px;
    overflow: hidden;

```

```
}
```

```
.blog__details__comment__item.blog__details__comment__item--reply {  
    padding-left: 112px;  
}
```

```
.blog__details__comment__item__pic {  
    float: left;  
    margin-right: 40px;  
}
```

```
.blog__details__comment__item__text {  
    overflow: hidden;  
}
```

```
.blog__details__comment__item__text span {  
    font-size: 14px;  
    color: #b7b7b7;  
    display: block;  
    margin-bottom: 10px;  
}
```

```
.blog__details__comment__item__text h5 {  
    color: #ffffff;  
    font-weight: 700;  
    margin-bottom: 10px;  
}
```

```
.blog__details__comment__item__text p {  
    color: #b7b7b7;
```

```

    font-size: 14px;
    line-height: 22px;
    margin-bottom: 25px;
}

.blog__details__comment__item__text a {
    display: inline-block;
    color: #ffffff;
    background: rgba(255, 255, 255, 0.1);
    padding: 6px 20px;
    letter-spacing: 2px;
    border-radius: 2px;
    margin-right: 14px;
    -webkit-transition: all, 0.3s;
    -o-transition: all, 0.3s;
    transition: all, 0.3s;
}

```

```

.blog__details__comment__item__text a:hover {
    background: #588157;
}

```

```

.blog__details__form {
    padding-top: 50px;
}

```

```

.blog__details__form h4 {
    color: #ffffff;
    font-weight: 700;
    margin-bottom: 26px;
}

```

```
}
```

```
.blog__details__form form input {
```

```
    height: 50px;
```

```
    width: 100%;
```

```
    background: #ffffff;
```

```
    font-size: 15px;
```

```
    color: #a6a6a6;
```

```
    padding-left: 20px;
```

```
    border-radius: 2px;
```

```
    border: none;
```

```
    margin-bottom: 30px;
```

```
}
```

```
.blog__details__form form input::-webkit-input-placeholder {
```

```
    color: #a6a6a6;
```

```
}
```

```
.blog__details__form form input::-moz-placeholder {
```

```
    color: #a6a6a6;
```

```
}
```

```
.blog__details__form form input:-ms-input-placeholder {
```

```
    color: #a6a6a6;
```

```
}
```

```
.blog__details__form form input::-ms-input-placeholder {
```

```
    color: #a6a6a6;
```

```
}
```



```
.blog__details__form form input::placeholder {  
    color: #a6a6a6;  
}
```

```
.blog__details__form form textarea {  
    height: 115px;  
    width: 100%;  
    background: #ffffff;  
    font-size: 15px;  
    color: #a6a6a6;  
    padding-left: 20px;  
    border-radius: 2px;  
    padding-top: 12px;  
    resize: none;  
    border: none;  
    margin-bottom: 34px;  
}
```

```
.blog__details__form form textarea::-webkit-input-placeholder {  
    color: #a6a6a6;  
}
```

```
.blog__details__form form textarea::-moz-placeholder {  
    color: #a6a6a6;  
}
```

```
.blog__details__form form textarea:-ms-input-placeholder {  
    color: #a6a6a6;  
}
```

```
.blog__details__form form textarea::-ms-input-placeholder {  
    color: #a6a6a6;  
}
```

```
.blog__details__form form textarea::placeholder {  
    color: #a6a6a6;  
}
```

```
/*-----  
Login  
-----*/
```

```
.login {  
    padding-top: 130px;  
    padding-bottom: 120px;  
}
```

```
.login__form {  
    position: relative;  
    padding-left: 145px;  
}
```

```
.login__form:after {  
    position: absolute;  
    right: -14px;  
    top: -40px;  
    height: 330px;  
    width: 1px;  
    background: rgba(255, 255, 255, 0.2);  
    content: "";
```

```
}
```

```
.login__form h3 {  
    color: #ffffff;  
    font-weight: 700;  
    font-family: "Oswald", sans-serif;  
    margin-bottom: 30px;  
}
```

```
.login__form form .input__item {  
    position: relative;  
    width: 370px;  
    margin-bottom: 20px;  
}
```

```
.login__form form .input__item:before {  
    position: absolute;  
    left: 50px;  
    top: 10px;  
    height: 30px;  
    width: 1px;  
    background: #b7b7b7;  
    content: "";  
}
```

```
.login__form form .input__item input {  
    height: 50px;  
    width: 100%;  
    font-size: 15px;  
    color: #b7b7b7;
```

```

background: #ffffff;
border: none;
padding-left: 76px;
}

.login__form form .input__item input::-webkit-input-placeholder {
    color: #b7b7b7;
}

.login__form form .input__item input::-moz-placeholder {
    color: #b7b7b7;
}

.login__form form .input__item input:-ms-input-placeholder {
    color: #b7b7b7;
}

.login__form form .input__item input::-ms-input-placeholder {
    color: #b7b7b7;
}

.login__form form .input__item input::placeholder {
    color: #b7b7b7;
}

.login__form form .input__item span {
    color: #b7b7b7;
    font-size: 20px;
    position: absolute;
    left: 15px;

```

```

        top: 13px;
    }

.login__form form button {
    border-radius: 0;
    margin-top: 10px;
}

.login__form .forget_pass {
    font-size: 15px;
    color: #ffffff;
    display: inline-block;
    position: absolute;
    right: 60px;
    bottom: 12px;
}

.login__register {
    padding-left: 30px;
}

.login__register h3 {
    color: #ffffff;
    font-weight: 700;
    font-family: "Oswald", sans-serif;
    margin-bottom: 30px;
}

.login__register .primary-btn {
    background: #588157;

```

```

        padding: 12px 42px;
    }

    /*-----
    Sign Up
    -----*/

    .signup {
        padding-top: 130px;
        padding-bottom: 150px;
    }

    .signup .login__form:after {
        height: 450px;
    }

    .signup .login__form h5 {
        font-size: 15px;
        color: #ffffff;
        margin-top: 36px;
    }

    .signup .login__form h5 a {
        color: #588157;
        font-weight: 700;
    }

    .signup .login__social__links {
        text-align: left;
        padding-left: 40px;
    }

```

```
}
```

```
.signup .login__social__links h3 {  
    color: #ffffff;  
    font-weight: 700;  
    font-family: "Oswald", sans-serif;  
    margin-bottom: 30px;  
}
```

```
.signup .login__social__links ul li a {  
    margin: 0;  
    text-align: center;  
}
```

```
/*-----  
  
Footer  
  
-----*/
```

```
.footer {  
    background: #210f04;  
    padding-top: 60px;  
    padding-bottom: 40px;  
    position: relative;  
}
```

```
.page-up {  
    position: absolute;  
    left: 50%;  
    top: -25px;  
    margin-left: -25px;
```

```
}
```

```
.page-up a {  
    display: inline-block;  
    font-size: 36px;  
    color: #ffffff;  
    height: 50px;  
    width: 50px;  
    background: #588157;  
    line-height: 50px;  
    text-align: center;  
    border-radius: 50%;  
}
```

```
.page-up a span {  
    position: relative;  
    top: 2px;  
    left: -1px;  
}
```

```
.footer__nav {  
    text-align: left;  
    margin-left: 40px;  
}
```

```
.footer__nav ul {  
    list-style: none;  
    position: relative;  
    margin-bottom: 10px;  
}
```



```
.footer__nav ul:last-child {  
    margin-right: 0;  
}
```

```
.footer__nav ul a {  
    font-size: 15px;  
    color: #b7b7b7;  
    font-weight: 700;  
}
```

```
.footer__copyright__text {  
    color: #b7b7b7;  
    margin-bottom: 0;  
    text-align: right;  
}
```

```
.footer__copyright__text a {  
    color: #588157;  
}
```

```
/*----- Responsive Media Quarries -----*/
```

```
@media only screen and (min-width: 1200px) and (max-width: 1300px) {  
    .hero {  
        overflow: hidden;  
    }  
}
```

```
@media only screen and (min-width: 1200px) {
```

```

        .container {
            max-width: 1170px;
        }
    }

/* Medium Device = 1200px */

@media only screen and (min-width: 992px) and (max-width: 1199px) {
    .hero {
        overflow: hidden;
    }
    .login__form {
        position: relative;
        padding-left: 32px;
    }
    .login__social__links ul li a {
        width: 380px;
    }
    .blog__item__text {
        padding: 0 50px;
    }
}

/* Tablet Device = 768px */

@media only screen and (min-width: 768px) and (max-width: 991px) {
    .hero {
        overflow: hidden;
    }
    .header {

```

```

        position: relative;
    }
    .header .container {
        position: relative;
    }
    .header__right {
        position: absolute;
        right: 120px;
        top: -42px;
        padding: 0;
    }
    .header__menu {
        display: none;
    }
    .slicknav_menu {
        background: transparent;
        padding: 0;
        display: block;
    }
    .slicknav_nav {
        position: absolute;
        left: 0;
        top: 60px;
        width: 100%;
        background: #ffffff;
        padding: 15px 30px;
        z-index: 9;
    }
    .slicknav_nav ul {
        margin: 0;

```

```

}
.slicknav_nav .slicknav_row,
.slicknav_nav a {
    padding: 7px 0;
    margin: 0;
    color: #111111;
    font-weight: 600;
}
.slicknav_btn {
    border-radius: 0;
    background-color: #222;
    position: absolute;
    right: 0;
    top: 9px;
}
.slicknav_nav .slicknav_arrow {
    color: #111111;
}
.slicknav_nav .slicknav_row:hover {
    border-radius: 0;
    background: transparent;
    color: #111111;
}
.slicknav_nav a:hover {
    border-radius: 0;
    background: transparent;
    color: #111111;
}
.product__sidebar {
    padding-top: 50px;
}

```

```

    }
    .footer__logo {
        text-align: center;
        margin-bottom: 20px;
    }
    .footer__nav {
        margin-bottom: 15px;
    }
    .footer__copyright__text {
        text-align: center;
    }
    .login__form {
        padding-left: 0;
        margin-bottom: 40px;
    }
    .login__form:after {
        display: none;
    }
    .login__form form .input__item {
        width: auto;
    }
    .login__register {
        padding-left: 0;
    }
    .signup .login__social__links {
        padding-left: 0;
    }
}

```

```

/* Wide Mobile = 480px */

```

```
@media only screen and (max-width: 767px) {  
    .hero {  
        overflow: hidden;  
    }  
    .header {  
        position: relative;  
    }  
    .header .container {  
        position: relative;  
    }  
    .header__right {  
        position: absolute;  
        right: 120px;  
        top: -42px;  
        padding: 0;  
    }  
    .header__menu {  
        display: none;  
    }  
    .slicknav_menu {  
        background: transparent;  
        padding: 0;  
        display: block;  
    }  
    .slicknav_nav {  
        position: absolute;  
        left: 0;  
        top: 60px;  
        width: 100%;  
    }  
}
```

```

        background: #ffffff;
        padding: 15px 30px;
        z-index: 9;
    }
    .slicknav_nav ul {
        margin: 0;
    }
    .slicknav_nav .slicknav_row,
    .slicknav_nav a {
        padding: 7px 0;
        margin: 0;
        color: #111111;
        font-weight: 600;
    }
    .slicknav_btn {
        border-radius: 0;
        background-color: #222;
        position: absolute;
        right: 0;
        top: 9px;
    }
    .slicknav_nav .slicknav_arrow {
        color: #111111;
    }
    .slicknav_nav .slicknav_row:hover {
        border-radius: 0;
        background: transparent;
        color: #111111;
    }
    .slicknav_nav a:hover {

```

```

        border-radius: 0;
        background: transparent;
        color: #111111;
    }
    .product__sidebar {
        padding-top: 50px;
    }
    .footer__logo {
        text-align: center;
        margin-bottom: 20px;
    }
    .footer__nav {
        margin-bottom: 15px;
    }
    .footer__copyright__text {
        text-align: center;
    }
    .blog__details__title h2 {
        font-size: 34px;
        line-height: normal;
    }
    .anime__details__pic {
        margin-bottom: 40px;
    }
    .anime__details__sidebar {
        padding-top: 50px;
    }
    .btn__all {
        text-align: left;
    }

```



```

.product__page__title .section-title {
    margin-bottom: 30px;
}

.product__page__title .product__page__filter {
    text-align: left;
}

.anime__details__rating {
    text-align: left;
    position: relative;
    margin-bottom: 20px;
}

.blog__details__social {
    overflow: hidden;
}

.blog__details__title .blog__details__social a {
    margin-right: 10px;
    margin-bottom: 10px;
    width: calc(50% - 10px);
    float: left;
}

.login__form {
    padding-left: 0;
    margin-bottom: 40px;
}

.login__form:after {
    display: none;
}

.login__form form .input__item {
    width: auto;
}

```

```

.signup .login__social__links {
    padding-left: 0;
}
.login__social__links ul li a {
    width: auto;
}
.blog__item__text {
    padding: 0 30px;
}
.login__register {
    padding-left: 0;
}
.product__sidebar__view .filter__controls li {
    margin-right: 2px;
}
.search-model-form input {
    width: 100%;
}
}

```

```

/* Small Device = 320px */

```

```

@media only screen and (max-width: 479px) {
    .hero__slider.owl-carousel .owl-nav {
        display: none;
    }
    .hero__items {
        padding: 250px 0 42px 15px;
    }
    .hero__text h2 {

```

```

        font-size: 32px;
    }
    .footer__nav ul li {
        margin-right: 10px;
    }
    .blog__details__title h2 {
        font-size: 30px;
        line-height: normal;
    }
    .blog__details__title .blog__details__social a {
        padding: 16px 25px 14px 20px;
    }
    .blog__details__comment__item.blog__details__comment__item--reply {
        padding-left: 0;
    }
    .blog__details__comment__item__pic {
        margin-right: 25px;
    }
    .blog__details__comment__item__text a {
        margin-right: 6px;
    }
    .login__social__links ul li a i {
        left: 20px;
    }
    .login__form .forget_pass {
        position: relative;
        left: 0;
        bottom: 0;
        margin-top: 25px;
    }

```

```

.header__right a {
    margin-right: 10px;
}

.anime__review__item__text h6 span {
    font-size: 12px;
}

.anime__review__item__text {
    padding: 18px 20px 20px;
}
}

```

2. Main.css (Giao diện admin)

```

.row {
    display: -webkit-box;
    display: -webkit-flex;
    display: -moz-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-flex-wrap: wrap;
    -ms-flex-wrap: wrap;
    flex-wrap: wrap;
}

.row .col-2:last-child .input-group-desc {
    margin-bottom: 0;
}

.row-space {
    -webkit-box-pack: justify;
    -webkit-justify-content: space-between;
}

```

```
-moz-box-pack: justify;
-ms-flex-pack: justify;
justify-content: space-between;
}
```

```
.row-refine {
  margin: 0 -15px;
}
```

```
.row-refine .col-3 .input-group-desc,
.row-refine .col-9 .input-group-desc {
  margin-bottom: 0;
}
```

```
.col-2 {
  width: -webkit-calc((100% - 30px) / 2);
  width: -moz-calc((100% - 30px) / 2);
  width: calc((100% - 30px) / 2);
}
```

```
@media (max-width: 767px) {
  .col-2 {
    width: 100%;
  }
}
```

```
.form-row {
  display: -webkit-box;
  display: -webkit-flex;
  display: -moz-box;
```

```
display: -ms-flexbox;
display: flex;
-webkit-flex-wrap: wrap;
-ms-flex-wrap: wrap;
flex-wrap: wrap;
-webkit-box-align: start;
-webkit-align-items: flex-start;
-moz-box-align: start;
-ms-flex-align: start;
align-items: flex-start;
padding: 24px 55px;
border-bottom: 1px solid #e5e5e5;
}
```

```
.form-row .name {
width: 188px;
color: #333;
font-size: 15px;
font-weight: 700;
margin-top: 11px;
}
```

```
.form-row .value {
width: -webkit-calc(100% - 188px);
width: -moz-calc(100% - 188px);
width: calc(100% - 188px);
}
```

```
@media (max-width: 767px) {
.form-row {
```

```

display: block;
padding: 24px 30px;
}

.form-row .name,
.form-row .value {
display: block;
width: 100%;
}

.form-row .name {
margin-top: 0;
margin-bottom: 12px;
}
}

/*
=====
=====

#BOX-SIZING

=====
===== */

/**
* More sensible default box-sizing:
* css-tricks.com/inheriting-box-sizing-probably-slightly-better-best-practice
*/
html {
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}

```

```

* {
padding: 0;
margin: 0;
}

*, *:before, *:after {
-webkit-box-sizing: inherit;
-moz-box-sizing: inherit;
box-sizing: inherit;
}

/*
=====
=====

#RESET

=====
===== */

/**
* A very simple reset that sits on top of Normalize.css.
*/
body,
h1, h2, h3, h4, h5, h6,
blockquote, p, pre,
dl, dd, ol, ul,
figure,
hr,
fieldset, legend {
margin: 0;
padding: 0;
}

```



```

/**
 * Remove trailing margins from nested lists.
 */
li > ol,
li > ul {
    margin-bottom: 0;
}

/**
 * Remove default table spacing.
 */
table {
    border-collapse: collapse;
    border-spacing: 0;
}

/**
 * 1. Reset Chrome and Firefox behaviour which sets a `min-width: min-content;`
 *    on fieldsets.
 */
fieldset {
    min-width: 0;
    /* [1] */
    border: 0;
}

button {
    outline: none;
    background: none;

```

```
border: none;
}
```

```
/*
```

```
=====
=====
```

```
#PAGE WRAPPER
```

```
=====
===== */
```

```
.page-wrapper {
  min-height: 100vh;
}
```

```
body {
  font-family: "Open Sans", "Arial", "Helvetica Neue", sans-serif;
  font-weight: 400;
  font-size: 14px;
}
```

```
h1, h2, h3, h4, h5, h6 {
  font-weight: 400;
}
```

```
h1 {
  font-size: 36px;
}
```

```
h2 {
  font-size: 30px;
}
```

```
h3 {  
    font-size: 24px;  
}
```

```
h4 {  
    font-size: 18px;  
}
```

```
h5 {  
    font-size: 15px;  
}
```

```
h6 {  
    font-size: 13px;  
}
```

```
a {  
    color: #452108;  
    text-decoration: none;  
    background-color: transparent;  
}
```

```
    a:hover {  
        color: #b75916;  
    }
```

```
/*
```

```
=====
```

```
#BACKGROUND
```

```
=====
```

```
*/
```

```
.bg-blue {  
  background: #2c6ed5;  
}
```

```
.bg-red {  
  background: #fa4251;  
}
```

```
.bg-dark {  
  background: #1a1a1a;  
}
```

```
.bg-gra-01 {  
  background: -webkit-gradient(linear, left bottom, left top, from(#fbc2eb), to(#a18cd1));  
  background: -webkit-linear-gradient(bottom, #fbc2eb 0%, #a18cd1 100%);  
  background: -moz-linear-gradient(bottom, #fbc2eb 0%, #a18cd1 100%);  
  background: -o-linear-gradient(bottom, #fbc2eb 0%, #a18cd1 100%);  
  background: linear-gradient(to top, #fbc2eb 0%, #a18cd1 100%);  
}
```

```
.bg-gra-02 {  
  background: -webkit-gradient(linear, left bottom, right top, from(#fc2c77), to(#6c4079));  
  background: -webkit-linear-gradient(bottom left, #fc2c77 0%, #6c4079 100%);  
  background: -moz-linear-gradient(bottom left, #fc2c77 0%, #6c4079 100%);  
  background: -o-linear-gradient(bottom left, #fc2c77 0%, #6c4079 100%);  
  background: linear-gradient(to top right, #fc2c77 0%, #6c4079 100%);  
}
```

```
.bg-gra-03 {  
  background: -webkit-gradient(linear, left bottom, right top, from(#08aeaa), to(#b721ff));
```

```

background: -webkit-linear-gradient(bottom left, #08aeaa 0%, #b721ff 100%);
background: -moz-linear-gradient(bottom left, #08aeaa 0%, #b721ff 100%);
background: -o-linear-gradient(bottom left, #08aeaa 0%, #b721ff 100%);
background: linear-gradient(to top right, #08aeaa 0%, #b721ff 100%);
}

/*
=====
=====

#SPACING

=====
===== */

.p-t-100 {
  padding-top: 100px;
}

.p-t-130 {
  padding-top: 130px;
}

.p-t-180 {
  padding-top: 180px;
}

.p-t-45 {
  padding-top: 45px;
}

.p-t-20 {
  padding-top: 20px;
}

```

```
}
```

```
.p-t-15 {  
  padding-top: 15px;  
}
```

```
.p-t-10 {  
  padding-top: 10px;  
}
```

```
.p-t-30 {  
  padding-top: 30px;  
}
```

```
.p-b-100 {  
  padding-bottom: 100px;  
}
```

```
.p-b-50 {  
  padding-bottom: 50px;  
}
```

```
.m-r-45 {  
  margin-right: 45px;  
}
```

```
.m-r-55 {  
  margin-right: 55px;  
}
```

```
.m-b-55 {  
    margin-bottom: 55px;  
}
```

```
/*
```

```
=====
```

```
#WRAPPER
```

```
=====
```

```
===== */
```

```
.wrapper {  
    margin: 0 auto;  
}
```

```
.wrapper--w960 {  
    max-width: 960px;  
}
```

```
.wrapper--w900 {  
    max-width: 900px;  
}
```

```
.wrapper--w790 {  
    max-width: 790px;  
}
```

```
.wrapper--w780 {  
    max-width: 780px;  
}
```

```
.wrapper--w1000 {  
    max-width: 1000px;  
}
```

```
/*
```

```
=====
```

```
#BUTTON
```

```
=====
```

```
===== */
```

```
.btn {  
    display: inline-block;  
    line-height: 50px;  
    padding: 0 30px;  
    -webkit-transition: all 0.4s ease;  
    -o-transition: all 0.4s ease;  
    -moz-transition: all 0.4s ease;  
    transition: all 0.4s ease;  
    cursor: pointer;  
    font-size: 15px;  
    text-transform: capitalize;  
    font-weight: 700;  
    color: #fff;  
    font-family: inherit;  
}
```

```
.btn--radius {  
    -webkit-border-radius: 3px;  
    -moz-border-radius: 3px;  
    border-radius: 3px;
```



```
}
```

```
.btn--radius-2 {  
  -webkit-border-radius: 5px;  
  -moz-border-radius: 5px;  
  border-radius: 5px;  
}
```

```
.btn--pill {  
  -webkit-border-radius: 20px;  
  -moz-border-radius: 20px;  
  border-radius: 20px;  
}
```

```
.btn--green {  
  background: #57b846;  
}
```

```
.btn--green:hover {  
  background: #4dae3c;  
}
```

```
.btn--blue {  
  background: #4272d7;  
}
```

```
.btn--blue:hover {  
  background: #3868cd;  
}
```

```
.btn--blue-2 {  
  background: #2c6ed5;  
}
```

```
.btn--blue-2:hover {  
  background: #185ac1;  
}
```

```
.btn--red {  
  background: #ff4b5a;  
}
```

```
.btn--red:hover {  
  background: #eb3746;  
}
```

```
/*
```

```
=====
```

```
#DATE PICKER
```

```
===== */
```

```
td.active {  
  background-color: #2c6ed5;  
}
```

```
input[type="date" i] {  
  padding: 14px;  
}
```

```
.table-condensed td, .table-condensed th {  
    font-size: 14px;  
    font-family: "Roboto", "Arial", "Helvetica Neue", sans-serif;  
    font-weight: 400;  
}
```

```
.daterangepicker td {  
    width: 40px;  
    height: 30px;  
}
```

```
.daterangepicker {  
    border: none;  
    -webkit-box-shadow: 0px 8px 20px 0px rgba(0, 0, 0, 0.15);  
    -moz-box-shadow: 0px 8px 20px 0px rgba(0, 0, 0, 0.15);  
    box-shadow: 0px 8px 20px 0px rgba(0, 0, 0, 0.15);  
    display: none;  
    border: 1px solid #e0e0e0;  
    margin-top: 5px;  
}
```

```
.daterangepicker::after, .daterangepicker::before {  
    display: none;  
}
```

```
.daterangepicker thead tr th {  
    padding: 10px 0;  
}
```

```
.daterangepicker .table-condensed th select {
```

```
border: 1px solid #ccc;
-webkit-border-radius: 3px;
-moz-border-radius: 3px;
border-radius: 3px;
font-size: 14px;
padding: 5px;
outline: none;
}
```

```
/*
```

```
=====
=====
```

```
#FORM
```

```
=====
===== */
```

```
input,
textarea,
select {
    outline: none;
    margin: 0;
    border: none;
    -webkit-box-shadow: none;
    -moz-box-shadow: none;
    box-shadow: none;
    width: 100%;
    font-size: 14px;
    font-family: inherit;
}
```

```
textarea {
```

```
    resize: vertical;
}
```

```
.input-group {
    position: relative;
    margin: 0;
}
```

```
.input--style-6,
.textarea--style-6 {
    background: transparent;
    line-height: 38px;
    border: 1px solid #cccccc;
    -webkit-border-radius: 3px;
    -moz-border-radius: 3px;
    border-radius: 3px;
    color: #666;
    font-size: 15px;
    -webkit-transition: all 0.4s ease;
    -o-transition: all 0.4s ease;
    -moz-transition: all 0.4s ease;
    transition: all 0.4s ease;
    padding: 0 20px;
}
```

```
.input--style-6::-webkit-input-placeholder,
.textarea--style-6::-webkit-input-placeholder {
    /* WebKit, Blink, Edge */
    color: #999;
}
```

```
.input--style-6:-moz-placeholder,  
.textarea--style-6:-moz-placeholder {  
    /* Mozilla Firefox 4 to 18 */  
    color: #999;  
    opacity: 1;  
}
```

```
.input--style-6::-moz-placeholder,  
.textarea--style-6::-moz-placeholder {  
    /* Mozilla Firefox 19+ */  
    color: #999;  
    opacity: 1;  
}
```

```
.input--style-6:-ms-input-placeholder,  
.textarea--style-6:-ms-input-placeholder {  
    /* Internet Explorer 10-11 */  
    color: #999;  
}
```

```
.input--style-6:-ms-input-placeholder,  
.textarea--style-6:-ms-input-placeholder {  
    /* Microsoft Edge */  
    color: #999;  
}
```

```
.input--style-6:focus,  
.textarea--style-6:focus {  
    -webkit-box-shadow: 0px 1px 5px 0px rgba(0, 0, 0, 0.15);
```

```
-moz-box-shadow: 0px 1px 5px 0px rgba(0, 0, 0, 0.15);
box-shadow: 0px 1px 5px 0px rgba(0, 0, 0, 0.15);
-webkit-transform: translateY(-3px);
-moz-transform: translateY(-3px);
-ms-transform: translateY(-3px);
-o-transform: translateY(-3px);
transform: translateY(-3px);
}
```

```
.textarea--style-6 {
  line-height: 1.2;
  min-height: 120px;
  max-height 800px;
  padding: 10px 20px;
}
```

```
.label--desc {
  font-size: 13px;
  color: #999;
  margin-top: 10px;
}
```

```
@media (max-width: 767px) {
  .label--desc {
    margin-top: 14px;
  }
}
```

```
.input-file {
  display: none;
```

```
}
```

```
.input-file + label {  
  font-size: 15px;  
  color: #fff;  
  color: white;  
  line-height: 40px;  
  background-color: #666666;  
  padding: 0 20px;  
  -webkit-border-radius: 3px;  
  -moz-border-radius: 3px;  
  border-radius: 3px;  
  display: inline-block;  
  margin-right: 15px;  
  -webkit-transition: all 0.4s ease;  
  -o-transition: all 0.4s ease;  
  -moz-transition: all 0.4s ease;  
  transition: all 0.4s ease;  
  cursor: pointer;  
}
```

```
.input-file:focus + label,  
.input-file + label:hover {  
  background: #1b1b1b;  
}
```

```
.input-file__info {  
  font-size: 15px;  
  color: #666;  
}
```



```

@media (max-width: 767px) {
    .input-file__info {
        display: block;
        margin-top: 6px;
    }
}

```

```

/*
=====
=====

```

```

    #TITLE

```

```

===== */

```

```

.title {
    font-size: 36px;
    font-weight: 700;
    text-align: left;
    color: #fff;
    margin-bottom: 24px;
}

```

```

@media (max-width: 767px) {
    .title {
        padding: 0 15px;
    }
}

```

```
/*
```

```
=====
```

```
#CARD
```

```
===== */
```

```
.card {  
  -webkit-border-radius: 3px;  
  -moz-border-radius: 3px;  
  border-radius: 3px;  
  background: #fff;  
}
```

```
.card-6 {  
  background: transparent;  
}
```

```
.card-6 .card-heading {  
  background: transparent;  
}
```

```
.card-6 .card-body {  
  background: #fff;  
  position: relative;  
  border: 1px solid #e5e5e5;  
  border-bottom: none;  
  padding: 50px 55px;  
  padding-bottom: 0;  
  -webkit-border-top-left-radius: 3px;  
  -moz-border-radius-topleft: 3px;
```

```
border-top-left-radius: 3px;
-webkit-border-top-right-radius: 3px;
-moz-border-radius-topright: 3px;
border-top-right-radius: 3px;
}
```

```
.card-6 .card-footer {
background: #fff;
border: 1px solid #e5e5e5;
border-top: none;
-webkit-border-bottom-left-radius: 3px;
-moz-border-radius-bottomleft: 3px;
border-bottom-left-radius: 3px;
-webkit-border-bottom-right-radius: 3px;
-moz-border-radius-bottomright: 3px;
border-bottom-right-radius: 3px;
padding: 50px 55px;
}
```

```
@media (max-width: 767px) {
.card-6 .card-footer {
padding: 50px 30px;
}
}
```

3. Các CSS khác

Ngoài hai file css trên, dự án còn sử dụng cái framework khác hỗ trợ phát triển giao diện gồm:

- Bootstrap
- Font-awesome
- Owl.carousel

VII. Javascript

1. Main.js

```
'use strict';

(function ($) {

    /*-----
       Preloader
    -----*/
    $(window).on('load', function () {
        $(".loader").fadeOut();
        $("#preloader").delay(200).fadeOut("slow");

        /*-----
           Filter
        -----*/
        $('.filter__controls li').on('click', function () {
            $('.filter__controls li').removeClass('active');
            $(this).addClass('active');
        });
        if ($('.filter__gallery').length > 0) {
            var containerEl = document.querySelector('.filter__gallery');
            var mixer = mixitup(containerEl);
        }
    });

    /*-----
       Background Set
    -----*/
    $('.set-bg').each(function () {
        var bg = $(this).data('setbg');
        $(this).css('background-image', 'url(' + bg + ')');
    });

    // Search model
    $('.search-switch').on('click', function () {
        $('.search-model').fadeIn(400);
    });

    $('.search-close-switch').on('click', function () {
        $('.search-model').fadeOut(400, function () {
            $('#search-input').val('');
        });
    });

    /*-----
       Navigation
    -----*/
    $(".mobile-menu").slicknav({
        prependTo: '#mobile-menu-wrap',
        allowParentLinks: true
    });

    /*-----
       Hero Slider
    -----*/
}
```

```

var hero_s = $(".hero__slider");
hero_s.owlCarousel({
    loop: true,
    margin: 0,
    items: 1,
    dots: true,
    nav: true,
    navText: ["<span class='arrow_carrot-left'></span>", "<span
class='arrow_carrot-right'></span>"],
    animateOut: 'fadeOut',
    animateIn: 'fadeIn',
    smartSpeed: 1200,
    autoHeight: false,
    autoplay: true,
    mouseDrag: false
});

/*-----
    Niceselect
-----*/
$('select').niceSelect();

/*-----
    Scroll To Top
-----*/
$("#scrollTopButton").click(function() {
    $("html, body").animate({ scrollTop: 0 }, "slow");
    return false;
});
})(jQuery);

```

2. Các file JS khác

Ngoài file js trên, dự án còn sử dụng các thư viện khác để hỗ trợ dự án gồm:

- Bootstrap
- Owl.carousel
- Jquery

VIII. Model

1. Account.cs

```

using System.ComponentModel.DataAnnotations;

namespace MineBlog.Models
{
    public class Account
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Username is required")]
        public string Username { get; set; }

        [Required(ErrorMessage = "Email is required")]
        [EmailAddress(ErrorMessage = "Invalid Email Address")]
        public string Email { get; set; }

        [Required(ErrorMessage = "Password is required")]
        public string Password { get; set; }
    }
}

```

```

        [Phone(ErrorMessage = "Invalid Phone Number")]
        public string Phone { get; set; }

        public string Role { get; set; }

        // Navigation properties
        public ICollection<Like> Likes { get; set; }
        public ICollection<Comment> Comments { get; set; }
        public ICollection<Blog> Blogs { get; set; }
    }
}

```

2. Blog.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace MineBlog.Models
{
    public class Blog
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Title is required")]
        public string Title { get; set; }

        [Required(ErrorMessage = "Description is required")]
        public string Description { get; set; }

        [Required(ErrorMessage = "URL Slug is required")]
        public string UrlSlug { get; set; }

        [Required(ErrorMessage = "Content is required")]
        public string Content { get; set; }

        [Url(ErrorMessage = "Invalid URL")]
        public string ImageUrl { get; set; }

        [DataType(DataType.DateTime)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd HH:mm:ss}",
            ApplyFormatInEditMode = true)]
        public DateTime Date { get; set; }

        // Foreign keys
        [Required(ErrorMessage = "Category is required")]
        public int CategoryId { get; set; }

        // Navigation properties
        public Category Category { get; set; }

        public ICollection<Tag> Tags { get; set; } = new List<Tag>();
        public ICollection<Like> Likes { get; set; }
        public ICollection<Comment> Comments { get; set; }

        [Required(ErrorMessage = "Author is required")]
        public int AuthorId { get; set; }

        public Account Author { get; set; }
    }
}

```

3. Category.cs

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

```

```

namespace MineBlog.Models
{
    public class Category
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Name is required")]
        public string Name { get; set; }

        [Url(ErrorMessage = "Invalid URL")]
        public string ImageUrl { get; set; }

        // Navigation properties
        public ICollection<Blog> Blogs { get; set; }
    }
}

```

4. Tag.cs

```

using System.Collections.Generic;

namespace MineBlog.Models
{
    public class Tag
    {
        public int Id { get; set; }
        public string Name { get; set; }

        // Navigation properties
        public ICollection<Blog> Blogs { get; set; }
    }
}

```

5. Comment.cs

```

using System;
using System.ComponentModel.DataAnnotations;

namespace MineBlog.Models
{
    public class Comment
    {
        public int Id { get; set; }

        [DataType(DataType.DateTime)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd HH:mm:ss}",
            ApplyFormatInEditMode = true)]
        public DateTime Date { get; set; }

        [Required(ErrorMessage = "Content is required")]
        [StringLength(500, ErrorMessage = "Content must be no longer than 500
characters")]
        public string Content { get; set; }

        // Foreign keys
        public int BlogId { get; set; }
        public int AccountId { get; set; }

        // Navigation properties
        public Blog Blog { get; set; }
        public Account Account { get; set; }
    }
}

```

6. Like.cs

```
namespace MineBlog.Models
{
    public class Like
    {
        public int Id { get; set; }

        // Foreign keys
        public int BlogId { get; set; }
        public int AccountId { get; set; }

        // Navigation properties
        public Blog Blog { get; set; }
        public Account Account { get; set; }
    }
}
```

IX. Controller

1. AAccountsController.cs (phía admin)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class AAccountsController : Controller
    {
        private readonly MineBlogDbContext _context;

        public AAccountsController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: AAccounts
        public async Task<IActionResult> Index()
        {
            return _context.Account != null ?
                View(await _context.Account.ToListAsync()) :
                Problem("Entity set 'MineBlogDbContext.Account' is null.");
        }

        // GET: AAccounts/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null || _context.Account == null)
            {
                return NotFound();
            }

            var account = await _context.Account
                .FirstOrDefaultAsync(m => m.Id == id);
            if (account == null)
            {
                return NotFound();
            }
        }
    }
}
```



```

        return View(account);
    }

    // GET: AAccounts/Create
    public IActionResult Create()
    {
        return View();
    }

    // POST: AAccounts/Create
    // To protect from overposting attacks, enable the specific properties you
    want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult>
Create([Bind("Id,Username,Email,Password,Phone,Role")] Account account)
    {
        if (true)
        {
            _context.Add(account);
            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        return View(account);
    }

    // GET: AAccounts/Edit/5
    public async Task<IActionResult> Edit(int? id)
    {
        if (id == null || _context.Account == null)
        {
            return NotFound();
        }

        var account = await _context.Account.FindAsync(id);
        if (account == null)
        {
            return NotFound();
        }
        return View(account);
    }

    // POST: AAccounts/Edit/5
    // To protect from overposting attacks, enable the specific properties you
    want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(int id,
[Bind("Id,Username,Email,Password,Phone,Role")] Account account)
    {
        if (id != account.Id)
        {
            return NotFound();
        }

        if (true)
        {
            try
            {
                _context.Update(account);
                await _context.SaveChangesAsync();
            }
            catch (DbUpdateConcurrencyException)

```

```

        {
            if (!AccountExists(account.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(account);
}

// GET: AAccounts/Delete/5
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Account == null)
    {
        return NotFound();
    }

    var account = await _context.Account
        .FirstOrDefaultAsync(m => m.Id == id);
    if (account == null)
    {
        return NotFound();
    }

    return View(account);
}

// POST: AAccounts/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Account == null)
    {
        return Problem("Entity set 'MineBlogDbContext.Account' is null.");
    }
    var account = await _context.Account.FindAsync(id);
    if (account != null)
    {
        _context.Account.Remove(account);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool AccountExists(int id)
{
    return (_context.Account?.Any(e => e.Id == id)).GetValueOrDefault();
}
}
}

```

2. ABlogsController.cs (phía admin)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;

```

```

using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class ABlogsController : Controller
    {
        private readonly MineBlogDbContext _context;

        public ABlogsController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: ABlogs
        public async Task<IActionResult> Index()
        {
            var mineBlogDbContext = _context.Blog.Include(b => b.Author).Include(b =>
b.Category);
            return View(await mineBlogDbContext.ToListAsync());
        }

        // GET: ABlogs/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null || _context.Blog == null)
            {
                return NotFound();
            }
            var blog = await _context.Blog
                .Include(b => b.Author)
                .Include(b => b.Category)
                .Include(b => b.Tags)
                .FirstOrDefaultAsync(m => m.Id == id);
            if (blog == null)
            {
                return NotFound();
            }

            return View(blog);
        }

        // GET: ABlogs/Create
        public IActionResult Create()
        {
            ViewData["AuthorId"] = new SelectList(_context.Account, "Id", "Username");
            ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name");
            ViewBag.Tags = _context.Tag.ToList();
            return View();
        }

        // POST: ABlogs/Create
        // To protect from overposting attacks, enable the specific properties you
want to bind to.
        // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Create(Blog blog, IFormFile image, int[]
tags)
        {
            if (true)
            {

```

```

        var fileName = Guid.NewGuid().ToString() +
Path.GetExtension(image.FileName);

        // Save the uploaded image to the wwwroot/images folder
        var filePath = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot", "img", "Blogs", fileName);
        var stream = new FileStream(filePath, FileMode.Create);
        image.CopyToAsync(stream);

        foreach (var tagId in tags)
        {
            var tag = _context.Tag.Find(tagId);
            if (tag != null)
            {
                blog.Tags.Add(tag);
            }
        }

        // Update the ImageUrl property with the relative path to the saved
image
        blog.ImageUrl = $"/img/Blogs/{fileName}";
        blog.Date = DateTime.Now;
        _context.Add(blog);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["AuthorId"] = new SelectList(_context.Account, "Id", "Email",
blog.AuthorId);
    ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name",
blog.CategoryId);
    ViewData["TagId"] = new SelectList(_context.Tag, "Id", "Name");
    return View(blog);
}

// GET: ABlogs/Edit/5
public async Task<IActionResult> Edit(int? id)
{
    if (id == null || _context.Blog == null)
    {
        return NotFound();
    }
    var blog = await _context.Blog.FindAsync(id);
    if (blog == null)
    {
        return NotFound();
    }
    ViewBag.Tags = _context.Tag.ToList();
    ViewData["AuthorId"] = new SelectList(_context.Account, "Id", "Email",
blog.AuthorId);
    ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name",
blog.CategoryId);
    return View(blog);
}

// POST: ABlogs/Edit/5
// To protect from overposting attacks, enable the specific properties you
want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id,
[Bind("Id,Title,Description,UrlSlug,Content,ImageUrl,Date,CategoryId,AuthorId")] Blog
blog, int[] tags)
{
    if (id != blog.Id)

```

```

    {
        return NotFound();
    }

    if (true)
    {
        try
        {
            var existingBlog = await _context.Blog
                .Include(b => b.Tags)
                .FirstOrDefaultAsync(b => b.Id == id);

            if (existingBlog == null)
            {
                return NotFound();
            }

            // Update tags
            existingBlog.Tags.Clear();

            if (tags != null && tags.Any())
            {
                foreach (var tagId in tags.Distinct())
                {
                    var tag = await _context.Tag.FindAsync(tagId);
                    if (tag != null)
                    {
                        existingBlog.Tags.Add(tag);
                    }
                }
            }
            _context.Update(existingBlog);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!BlogExists(blog.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    // Reload all tags and pass them to the view
    ViewBag.Tags = _context.Tag.ToList();
    ViewData["AuthorId"] = new SelectList(_context.Account, "Id", "Email",
blog.AuthorId);
    ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name",
blog.CategoryId);
    return View(blog);
}

// GET: ABlogs/Delete/5
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Blog == null)
    {
        return NotFound();
    }

    var blog = await _context.Blog

```

```

        .Include(b => b.Author)
        .Include(b => b.Category)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (blog == null)
    {
        return NotFound();
    }

    return View(blog);
}

// POST: ABlogs/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Blog == null)
    {
        return Problem("Entity set 'MineBlogDbContext.Blog' is null.");
    }
    var blog = await _context.Blog.FindAsync(id);
    if (blog != null)
    {
        _context.Blog.Remove(blog);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool BlogExists(int id)
{
    return (_context.Blog?.Any(e => e.Id == id)).GetValueOrDefault();
}
}
}

```

3. ACategoriesController.cs (phía admin)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class ACategoriesController : Controller
    {
        private readonly MineBlogDbContext _context;

        public ACategoriesController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: ACategories
        public async Task<IActionResult> Index()
        {
            return _context.Category != null ?
                View(await _context.Category.ToListAsync()) :

```

```

        Problem("Entity set 'MineBlogDbContext.Category' is
null.");
    }

    // GET: ACategories/Details/5
    public async Task<IActionResult> Details(int? id)
    {
        if (id == null || _context.Category == null)
        {
            return NotFound();
        }

        var category = await _context.Category
            .FirstOrDefaultAsync(m => m.Id == id);
        if (category == null)
        {
            return NotFound();
        }

        return View(category);
    }

    // GET: ACategories/Create
    public IActionResult Create()
    {
        return View();
    }

    // POST: ACategories/Create
    // To protect from overposting attacks, enable the specific properties you
    want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Create(Category category, IFormFile image)
    {
        if (true)
        {
            var fileName = Guid.NewGuid().ToString() +
Path.GetExtension(image.FileName);

            // Save the uploaded image to the wwwroot/images folder
            var filePath = Path.Combine(Directory.GetCurrentDirectory(),
"wwwroot", "img", "Category", fileName);
            var stream = new FileStream(filePath, FileMode.Create);
            image.CopyToAsync(stream);

            // Update the ImageUrl property with the relative path to the saved
image
            category.ImageUrl = $"/img/Category/{fileName}";
            _context.Add(category);
            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        return View(category);
    }

    // GET: ACategories/Edit/5
    public async Task<IActionResult> Edit(int? id)
    {
        if (id == null || _context.Category == null)
        {
            return NotFound();
        }
    }

```

```

        var category = await _context.Category.FindAsync(id);
        if (category == null)
        {
            return NotFound();
        }
        return View(category);
    }

    // POST: ACategories/Edit/5
    // To protect from overposting attacks, enable the specific properties you
    want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(int id, [Bind("Id,Name,ImageUrl")]
Category category)
    {
        if (id != category.Id)
        {
            return NotFound();
        }

        if (true)
        {
            try
            {
                _context.Update(category);
                await _context.SaveChangesAsync();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!CategoryExists(category.Id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }
            return RedirectToAction(nameof(Index));
        }
        return View(category);
    }

    // GET: ACategories/Delete/5
    public async Task<IActionResult> Delete(int? id)
    {
        if (id == null || _context.Category == null)
        {
            return NotFound();
        }

        var category = await _context.Category
            .FirstOrDefaultAsync(m => m.Id == id);
        if (category == null)
        {
            return NotFound();
        }

        return View(category);
    }

    // POST: ACategories/Delete/5

```



```

[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Category == null)
    {
        return Problem("Entity set 'MineBlogDbContext.Category' is null.");
    }
    var category = await _context.Category.FindAsync(id);
    if (category != null)
    {
        _context.Category.Remove(category);

        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }

    private bool CategoryExists(int id)
    {
        return (_context.Category?.Any(e => e.Id == id)).GetValueOrDefault();
    }
}

```

4. ATagsController.cs (phía admin)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class ATagsController : Controller
    {
        private readonly MineBlogDbContext _context;

        public ATagsController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: ATags
        public async Task<IActionResult> Index()
        {
            return _context.Tag != null ?
                View(await _context.Tag.ToListAsync()) :
                Problem("Entity set 'MineBlogDbContext.Tag' is null.");
        }

        // GET: ATags/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null || _context.Tag == null)
            {
                return NotFound();
            }

            var tag = await _context.Tag
                .FirstOrDefaultAsync(m => m.Id == id);

```

```

        if (tag == null)
        {
            return NotFound();
        }

        return View(tag);
    }

    // GET: ATags/Create
    public IActionResult Create()
    {
        return View();
    }

    // POST: ATags/Create
    // To protect from overposting attacks, enable the specific properties you
want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Create([Bind("Id,Name")] Tag tag)
    {
        if (true)
        {
            _context.Add(tag);
            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        return View(tag);
    }

    // GET: ATags/Edit/5
    public async Task<IActionResult> Edit(int? id)
    {
        if (id == null || _context.Tag == null)
        {
            return NotFound();
        }

        var tag = await _context.Tag.FindAsync(id);
        if (tag == null)
        {
            return NotFound();
        }
        return View(tag);
    }

    // POST: ATags/Edit/5
    // To protect from overposting attacks, enable the specific properties you
want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(int id, [Bind("Id,Name")] Tag tag)
    {
        if (id != tag.Id)
        {
            return NotFound();
        }

        if (true)
        {
            try
            {
                _context.Update(tag);
            }

```

```

        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!TagExists(tag.Id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }
    return RedirectToAction(nameof(Index));
}
return View(tag);
}

// GET: ATags/Delete/5
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Tag == null)
    {
        return NotFound();
    }

    var tag = await _context.Tag
        .FirstOrDefaultAsync(m => m.Id == id);
    if (tag == null)
    {
        return NotFound();
    }

    return View(tag);
}

// POST: ATags/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Tag == null)
    {
        return Problem("Entity set 'MineBlogDbContext.Tag' is null.");
    }
    var tag = await _context.Tag.FindAsync(id);
    if (tag != null)
    {
        _context.Tag.Remove(tag);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool TagExists(int id)
{
    return (_context.Tag?.Any(e => e.Id == id)).GetValueOrDefault();
}
}
}

```

5. CommentsController.cs

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Reflection.Metadata;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class CommentsController : Controller
    {
        private readonly MineBlogDbContext _context;

        public CommentsController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: Comments
        public async Task<IActionResult> Index()
        {
            var mineBlogDbContext = _context.Comment.Include(c => c.Account).Include(c
=> c.Blog);
            return View(await mineBlogDbContext.ToListAsync());
        }

        // GET: Comments/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null || _context.Comment == null)
            {
                return NotFound();
            }

            var comment = await _context.Comment
                .Include(c => c.Account)
                .Include(c => c.Blog)
                .FirstOrDefaultAsync(m => m.Id == id);
            if (comment == null)
            {
                return NotFound();
            }

            return View(comment);
        }

        // GET: Comments/Create
        public IActionResult Create()
        {
            ViewData["AccountId"] = new SelectList(_context.Account, "Id",
"Username");
            ViewData["BlogId"] = new SelectList(_context.Blog, "Id", "Title");
            return View();
        }

        // POST: Comments/Create
        // To protect from overposting attacks, enable the specific properties you
want to bind to.
        // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult>
Create([Bind("Id,Date,Content,BlogId,AccountId")] Comment comment)

```

```

        {
            if (true)
            {
                comment.Date = DateTime.Now;
                _context.Add(comment);
                await _context.SaveChangesAsync();
                return RedirectToAction("Details", "UBlogs", new { id = comment.BlogId
});
            }
            ViewData["AccountId"] = new SelectList(_context.Account, "Id", "Username",
comment.AccountId);
            ViewData["BlogId"] = new SelectList(_context.Blog, "Id", "Title",
comment.BlogId);
            return View(comment);
        }

// GET: Comments/Delete/5
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Comment == null)
    {
        return NotFound();
    }

    var comment = await _context.Comment
        .Include(c => c.Account)
        .Include(c => c.Blog)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (comment == null)
    {
        return NotFound();
    }

    return View(comment);
}

// POST: Comments/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Comment == null)
    {
        return Problem("Entity set 'MineBlogDbContext.Comment' is null.");
    }
    var comment = await _context.Comment.FindAsync(id);
    if (comment != null)
    {
        _context.Comment.Remove(comment);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool CommentExists(int id)
{
    return (_context.Comment?.Any(e => e.Id == id)).GetValueOrDefault();
}
}
}

```

6. UBlogsController.cs (phía người dùng)

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class UBlogsController : Controller
    {
        private readonly MineBlogDbContext _context;

        public UBlogsController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: ABlogs
        public async Task<IActionResult> Index()
        {
            var mineBlogDbContext = _context.Blog
                .Include(b => b.Author)
                .Include(b => b.Category)
                .Include(b => b.Tags);
            return View(await mineBlogDbContext.ToListAsync());
        }

        // GET: ABlogs
        public async Task<IActionResult> List(int categoryId)
        {
            var mineBlogDbContext = _context.Blog
                .Include(b => b.Author)
                .Include(b => b.Category)
                .Include(b => b.Tags)
                .Where(b => b.CategoryId == categoryId);
            var category = await _context.Category.FindAsync(categoryId);
            ViewBag.CategoryName = category.Name;
            ViewBag.CategoryURL = category.ImageUrl;
            return View(await mineBlogDbContext.ToListAsync());
        }

        // GET: ABlogs/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            ViewData["AccountId"] = new SelectList(_context.Account, "Id",
"Username");
            ViewBag.CurrentBlogId = id;
            if (id == null || _context.Blog == null)
            {
                return NotFound();
            }

            var blog = await _context.Blog
                .Include(b => b.Author)
                .Include(b => b.Category)
                .Include(b => b.Tags)
                .Include(b => b.Comments)
                .FirstOrDefaultAsync(m => m.Id == id);
            if (blog == null)
            {
                return NotFound();
            }
            ViewBag.CategoryID = blog.Category.Id;
            return View(blog);
        }
    }
}

```

```

    }
}

```

7. UCategoriesController.cs (phía người dùng)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MineBlog.Data;
using MineBlog.Models;

namespace MineBlog.Controllers
{
    public class UCategoriesController : Controller
    {
        private readonly MineBlogDbContext _context;

        public UCategoriesController(MineBlogDbContext context)
        {
            _context = context;
        }

        // GET: ACategories
        public async Task<IActionResult> Index()
        {
            return _context.Category != null ?
                View(await _context.Category.ToListAsync()) :
                Problem("Entity set 'MineBlogDbContext.Category' is
null.");
        }

        // GET: ACategories/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null || _context.Category == null)
            {
                return NotFound();
            }

            var category = await _context.Category
                .FirstOrDefaultAsync(m => m.Id == id);
            if (category == null)
            {
                return NotFound();
            }

            return View(category);
        }
    }
}

```

X. Views

1. Shared

a. _adminLayout.cshtml

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />

```

```

<meta name="description" content="Minecraft Blog">
<meta name="keywords" content="">
<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>@ViewData["Title"] - MineBlog</title>

<!-- Google Font -->
<link
href="https://fonts.googleapis.com/css2?family=Oswald:wght@300;400;500;600;700&display=swap" rel="stylesheet">
<link
href="https://fonts.googleapis.com/css2?family=Mulish:wght@300;400;500;600;700;800;900&display=swap"
rel="stylesheet">

<!-- Css Styles -->
<link rel="stylesheet" href="~/css/bootstrap.min.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/font-awesome.min.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/elegant-icons.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/plyr.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/nice-select.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/owl.carousel.min.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/slicknav.min.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/style.css" type="text/css" asp-append-version="true">
<link rel="stylesheet" href="~/css/main.css" type="text/css" asp-append-version="true">
</head>

<body>
<!-- Header Section Begin -->
<header class="header">
<div class="container">
<div class="row">
<div class="col-lg-2">
<div class="header__logo">
<a asp-controller="AHome" asp-action="Index">

</a>
</div>
</div>
<div class="col-lg-8">
<div class="header__nav">
<nav class="header__menu mobile-menu">
<ul>
<li class="active"><a asp-controller="AHome" asp-action="Index">Homepage</a></li>
<li><a asp-controller="ABlogs" asp-action="Create">Create Blogs</a></li>
<li>
<a href="#">Manages <span class="arrow_carrot-down"></span></a>
<ul class="dropdown">
<li><a asp-controller="ATags" asp-action="Index">Tags</a></li>

```



```

        <li><a asp-controller="ACategories" asp-
action="Index">Categories</a></li>
        <li><a asp-controller="ABlogs" asp-
action="Index">Blogs</a></li>
        <li><a asp-controller="AAccounts" asp-
action="Index">Users</a></li>
    </ul>
</li>
<li><a asp-controller="Home" asp-
action="Index">Contacts</a></li>
</ul>
</nav>
</div>
</div>
<div class="col-lg-2">
    <div class="header__right">
        <a href="#" class="search-switch"><span
class="icon_search"></span></a>
        <a asp-controller="Login" asp-action="Index"><span
class="icon_profile"></span></a>
    </div>
</div>
</div>
<div id="mobile-menu-wrap"></div>
</div>
</header>

<!-- Header End -->
@RenderBody()

<footer class="footer">
    <div class="page-up">
        <a href="#" id="scrollToTopButton"><span class="arrow_carrot-
up"></span></a>
    </div>
    <div class="container">
        <div class="row">
            <div class="col-lg-4">
                <div class="footer__nav">
                    <div class="section-title">
                        <h5>Popular Categories</h5>
                    </div>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="6">News and Update</a>
                    </ul>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="7">Mod and Resources Pack</a>
                    </ul>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="1002">Maps share</a>
                    </ul>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="2002">Tips and Tricks</a>
                    </ul>
                </div>
            </div>
            <div class="col-lg-4">
                <div class="footer__nav">
                    <div class="section-title">
                        <h5>About</h5>
                    </div>
                </div>
            </div>
        </div>
    </div>
</footer>

```



```

<!-- Css Styles -->
<link rel="stylesheet" href="~/css/bootstrap.min.css" type="text/css" asp-append-
version="true">
<link rel="stylesheet" href="~/css/font-awesome.min.css" type="text/css" asp-
append-version="true">
<link rel="stylesheet" href="~/css/elegant-icons.css" type="text/css" asp-append-
version="true">
<link rel="stylesheet" href="~/css/plyr.css" type="text/css" asp-append-
version="true">
<link rel="stylesheet" href="~/css/nice-select.css" type="text/css" asp-append-
version="true">
<link rel="stylesheet" href="~/css/owl.carousel.min.css" type="text/css" asp-
append-version="true">
<link rel="stylesheet" href="~/css/slicknav.min.css" type="text/css" asp-append-
version="true">
<link rel="stylesheet" href="~/css/style.css" type="text/css" asp-append-
version="true">
</head>
<body>
<!-- Page Preloder -->
<div id="preloader">
<div class="loader"></div>
</div>

<!-- Header Section Begin -->
<header class="header">
<div class="container">
<div class="row">
<div class="col-lg-2">
<div class="header__logo">
<a asp-controller="UHome" asp-action="Index">

</a>
</div>
</div>
<div class="col-lg-8">
<div class="header__nav">
<nav class="header__menu mobile-menu">
<ul>
<li class="active"><a asp-controller="UHome" asp-
action="Index">Homepage</a></li>
<li><a asp-controller="UBlogs" asp-
action="Index">Blogs</a></li>
<li>
<a asp-controller="UCategories" asp-
action="Index">Categories <span class="arrow_carrot-down"></span></a>
<ul class="dropdown">
<li><a asp-controller="UBlogs" asp-
action="List"
asp-route-categoryId="6">News and
Update</a>
</li>
<li>
<a asp-controller="UBlogs" asp-
action="List"
asp-route-categoryId="7">Mod and
Resources Pack</a>
</li>
<li>
<a asp-controller="UBlogs" asp-
action="List"
asp-route-categoryId="1002">Maps
share</a>
</li>
<li>

```

```

        <a asp-controller="UBlogs" asp-
action="List"
Tricks</a>
        asp-route-categoryId="2002">Tips and
        </li>
    </ul>
</li>
<li><a asp-controller="UHome" asp-
action="Index">Contacts</a></li>
    </ul>
</nav>
</div>
</div>
<div class="col-lg-2">
    <div class="header__right">
        <a href="#" class="search-switch"><span
class="icon_search"></span></a>
        <a asp-controller="ULogin" asp-action="Index"><span
class="icon_profile"></span></a>
    </div>
</div>
</div>
<div id="mobile-menu-wrap"></div>
</div>
</header>
<!-- Header End -->
@RenderBody()

<!-- Footer Section Begin -->
<footer class="footer">
    <div class="page-up">
        <a href="#" id="scrollToTopButton"><span class="arrow_carrot-
up"></span></a>
    </div>
    <div class="container">
        <div class="row">
            <div class="col-lg-4">
                <div class="footer__nav">
                    <div class="section-title">
                        <h5>Popular Categories</h5>
                    </div>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="6">News and Update</a>
                    </ul>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="7">Mod and Resources Pack</a>
                    </ul>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="1002">Maps share</a>
                    </ul>
                    <ul>
                        <a asp-controller="UBlogs" asp-action="List"
asp-route-categoryId="2002">Tips and Tricks</a>
                    </ul>
                </div>
            </div>
            <div class="col-lg-4">
                <div class="footer__nav">
                    <div class="section-title">
                        <h5>About</h5>
                    </div>
                </div>
            </div>
        </div>
    </div>
</footer>

```

```

        <ul><a>Address: 273 An Duong Vuong Street</a> </ul>
        <ul><a>Phone: 0333464563</a></ul>
        <ul><a>Email: voha308@gmail.com</a></ul>
    </div>
</div>
<div class="col-lg-4">
    <div class="footer__logo">
        <a asp-controller="UHome" asp-action="Index"></a>
        <p>A Blog For Minecraft</p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Footer Section End -->
<!-- Search model Begin -->
<div class="search-model">
    <div class="h-100 d-flex align-items-center justify-content-center">
        <div class="search-close-switch"><i class="icon_close"></i></div>
        <form class="search-model-form">
            <input type="text" id="search-input" placeholder="Search here....">
        </form>
    </div>
</div>
<!-- Search model end -->

<!-- Js Plugins -->
<script src="~/js/jquery-3.3.1.min.js"></script>
<script src="~/js/bootstrap.min.js"></script>
<script src="~/js/player.js"></script>
<script src="~/js/jquery.nice-select.min.js"></script>
<script src="~/js/mixitup.min.js"></script>
<script src="~/js/jquery.slicknav.js"></script>
<script src="~/js/owl.carousel.min.js"></script>
<script src="~/js/main.js"></script>
</body>
</html>

```

c. *_Comments.cshtml*

@model MineBlog.Models.Comment

```

<form asp-controller="Comments" asp-action="Create">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <input type="hidden" asp-for="AccountId" value="5" />
    <input type="hidden" asp-for="Date" value="null" />
    <div class="row">
        <div class="col-lg-12">
            <textarea asp-for="Content" placeholder="Message"></textarea>
            <span asp-validation-for="Content" class="text-danger"></span>
            <input asp-for="BlogId" type="hidden" value="@ViewBag.CurrentBlogId" />
            <div class="form-group" style="display: none">
                <label asp-for="AccountId" class="control-label"></label>
                <select asp-for="AccountId" class="form-control" asp-
items="ViewBag.AccountId"></select>
            </div>
            <button type="submit" value="Create" class="site-btn">Send
Message</button>
        </div>
    </div>
</form>

```

2. AAccounts (phía admin)

a. *Index.cshtml*

```
@model IEnumerable<MineBlog.Models.Account>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Account Management</h2>
                    <p>Manage your users Account here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <div class="card card-6">
        <div class="card-body">
            <div class="container">
                <a asp-action="Create">Create New</a>
            </div>
        </div>
        <div class="card-footer">
            <table class="table" style="background-color: white;">
                <thead>
                    <tr>
                        <th class="col-2">Username</th>
                        <th class="col-3">Email</th>
                        <th class="col-2">Phone</th>
                        <th class="col-2">Role</th>
                        <th class="col-3">Tools</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach(var obj in Model)
                    {
                        <tr>
                            <td>@Html.DisplayFor(modelItem => obj.Username)</td>
                            <td>@Html.DisplayFor(modelItem => obj.Email)</td>
                            <td>@Html.DisplayFor(modelItem => obj.Phone)</td>
                            <td>@Html.DisplayFor(modelItem => obj.Role)</td>
                            <td>
                                <a asp-action="Edit" asp-route-id="@obj.Id">Edit</a> |
                                <a asp-action="Delete" asp-route-
                                id="@obj.Id">Delete</a>
                            </td>
                        </tr>
                    }
                </tbody>
            </table>
        </div>
    </div>
</div>
```

</div>

b. Create.cshtml

```
@model MineBlog.Models.Account
```

```
@{
    ViewData["Title"] = "Create";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}
<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Account Management</h2>
                    <p>Manage your users Account here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Create">
        <div class="card card-6">
            <div class="card-body">
                <div>
                    <a asp-action="Index">Back to List</a>
                </div>
                <div class="card-body">
                    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                    <div class="form-row">
                        <div class="name">Account Name</div>
                        <div class="value">
                            <input class="input--style-6" type="text" asp-
for="Username">
                            <span asp-validation-for="Username" class="text-
danger"></span>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Account Email</div>
                        <div class="value">
                            <input class="input--style-6" type="text" asp-for="Email">
                            <span asp-validation-for="Email" class="text-
danger"></span>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Account Password</div>
                        <div class="value">
                            <input class="input--style-6" type="text" asp-
for="Password">
                            <span asp-validation-for="Password" class="text-
danger"></span>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Account Phone</div>

```

```

        <div class="value">
            <input class="input--style-6" type="text" asp-for="Phone">
            <span asp-validation-for="Phone" class="text-
danger"></span>
        </div>
    </div>
    <div class="form-row">
        <div class="name">Roles</div>
        <div class="value">
            <div class="input-group">
                <select class="form-select input--style-6" asp-
for="Role">
                    <option disabled selected>Choose role for the
Account</option>
                    <option value="admin">Administrater</option>
                    <option value="user">User</option>
                </select>
            </div>
            <div class="label--desc">Only pick one role</div>
        </div>
    </div>
    <div class="card-footer">
        <div class="form-row">
            <button class="btn btn--radius-2 btn--blue-2" type="submit"
style="margin-top: 10px;">Create Account</button>
        </div>
    </div>
</form>
</div>

```

c. Edit.cshtml

```

@model MineBlog.Models.Account

@{
    ViewData["Title"] = "Edit";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Account Management</h2>
                    <p>Manage your users Account here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Edit">
        <div class="card card-6">
            <div class="card-body">
                <div>
                    <a asp-action="Index">Back to List</a>
                </div>
            </div>
        </div>
    </form>

```



```

</div>
<div class="card-body">
    <!-- Hidden Edit -->
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <input type="hidden" asp-for="Id" />
    <input type="hidden" asp-for="Email" />
    <input type="hidden" asp-for="Password" />
    <input type="hidden" asp-for="Phone" />
    <!-- Hidden End -->
    <div class="form-row">
        <div class="name">Account Name</div>
        <div class="value">
            <input class="input--style-6" type="text" asp-
for="Username">
            <span asp-validation-for="Username" class="text-
danger"></span>
        </div>
    </div>
    <div class="form-row">
        <div class="name">Roles</div>
        <div class="value">
            <div class="input-group">
                <select class="form-select input--style-6" asp-
for="Role">
                    <option disabled selected>Choose role for the
Account</option>
                    <option value="admin">Administrater</option>
                    <option value="user">User</option>
                </select>
            </div>
            <div class="label--desc">Only pick one role</div>
        </div>
    </div>
</div>
<div class="card-footer">
    <div class="form-row">
        <button class="btn btn--radius-2 btn--blue-2" type="submit"
style="margin-top: 10px;">Save Account</button>
    </div>
</div>
</form>
</div>

```

d. Delete.cshtml

```
@model MineBlog.Models.Account
```

```

@{
    ViewData["Title"] = "Delete";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

```

```

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Account Management</h2>
                    <p>Are you sure you want to delete this?</p>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <div class="card card-6">
        <div class="card-body">
            <div>
                <a asp-action="Index">Back to List</a>
            </div>
        </div>
        <div class="card-footer">
            <table class="table" style="background-color: white;">
                <thead>
                    <tr>
                        <th class="col-2">Username</th>
                        <th class="col-3">Email</th>
                        <th class="col-2">Phone</th>
                        <th class="col-2">Role</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>@Html.DisplayFor(model => model.Username)</td>
                        <td>@Html.DisplayFor(model => model.Email)</td>
                        <td>@Html.DisplayFor(model => model.Phone)</td>
                        <td>@Html.DisplayFor(model => model.Role)</td>
                    </tr>
                </tbody>
            </table>
            <form asp-action="Delete">
                <input type="hidden" asp-for="Id" />
                <input type="submit" value="Delete" class="btn btn-danger" />
            </form>
        </div>
    </div>
</div>

```

```
</div>
```

3. ABlogs (phía admin)

a. *Index.cshtml*

```

@model IEnumerable<MineBlog.Models.Blog>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Blogs Management</h2>
                    <p>Manage your Blogs here.</p>
                </div>
            </div>
        </div>
    </div>
</section>

```

```

<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
  <div class="card card-6">
    <div class="card-body">
      <div class="container">
        <a asp-action="Create">Create New</a>
      </div>
    </div>
    <div class="card-footer">
      <table class="table" style="background-color: white;">
        <thead>
          <tr>
            <th class="col-4">Title</th>
            <th class="col-2">Category</th>
            <th class="col-2">Date</th>
            <th class="col-2">Author</th>
            <th class="col-2">Tools</th>
          </tr>
        </thead>
        <tbody>
          @foreach(var obj in Model)
          {
            <tr>
              <td>@Html.DisplayFor(modelItem => obj.Title)</td>
              <td>@Html.DisplayFor(modelItem => obj.Category.Name)</td>
              <td>@Html.DisplayFor(modelItem => obj.Date)</td>
              <td>@Html.DisplayFor(modelItem =>
obj.Author.Username)</td>
              <td>
                <a asp-action="Edit" asp-route-id="@obj.Id">Edit</a> |
                <a asp-action="Details" asp-route-
id="@obj.Id">Details</a> |
                <a asp-action="Delete" asp-route-
id="@obj.Id">Delete</a>
              </td>
            </tr>
          }
        </tbody>
      </table>
    </div>
  </div>
</div>

```

b. Create.cshtml

```

@model MineBlog.Models.Blog

@{
  ViewData["Title"] = "Create";
  Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
  <div class="container">
    <div class="row">
      <div class="col-lg-12 text-center">
        <div class="normal__breadcrumb__text">
          <h2>Create Blogs</h2>
          <p>Write your new Blog</p>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Create" enctype="multipart/form-data">
        <input type="hidden" asp-for="ImageUrl" value="null" />
        <input type="hidden" asp-for="UrlSlug" value="null" />
        <input type="hidden" asp-for="Date" value="null" />
        <div class="card card-6">
            <div class="card-body">
                <div>
                    <a asp-action="Index">Back to List</a>
                </div>
            </div>
            <div class="card-body">
                <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                <div class="form-row">
                    <div class="name">Blog Title</div>
                    <div class="value">
                        <input class="input--style-6" type="text" asp-for="Title">
                        <span asp-validation-for="Title" class="text-danger"></span>
                    </div>
                </div>
                <div class="form-row">
                    <div class="name">Short Description</div>
                    <div class="value">
                        <div class="input-group">
                            <input class="input--style-6" type="text" asp-
for="Description" placeholder="">
                            <span asp-validation-for="Description" class="text-
danger"></span>
                        </div>
                    </div>
                </div>
                <div class="form-row">
                    <div class="name">Content</div>
                    <div class="value">
                        <div class="input-group">
                            <textarea class="textarea--style-6" asp-for="Content"
placeholder="Write your content
here."></textarea>
                            <span asp-validation-for="Content" class="text-
danger"></span>
                        </div>
                    </div>
                </div>
                <div class="form-row">
                    <div class="name">Categories</div>
                    <div class="value">
                        <div class="input-group">
                            <select class="form-select input--style-6" asp-
for="CategoryId" asp-items="ViewBag.CategoryId"></select>
                        </div>
                    </div>
                </div>
                <div class="form-row">
                    <div class="name">Tags</div>
                    <div class="value">
                        <div class="input-group">
                            @foreach (var tag in ViewBag.Tags)
                            {
                                <label>

```

```

        <input type="checkbox" name="tags" value="@tag.Id"
/>
        @tag.Name
    </label>
    }
</div>
</div>
</div>
<div class="form-row">
    <div class="name">Author</div>
    <div class="value">
        <div class="input-group">
            <select class="form-select input--style-6" asp-
for="AuthorId" asp-items="ViewBag.AuthorId"></select>
        </div>
    </div>
</div>
<div class="form-row">
    <div class="name">Upload Image</div>
    <div class="value">
        <div class="input-group js-input-file">
            <input class="input-file" type="file" name="image"
accept="image/*" id="file">
            <label class="label--file" for="file">Choose file</label>
        </div>
        <div class="label--desc">Upload the image file for the blog.
Max file size 5 MB</div>
    </div>
</div>
</div>
<div class="card-footer">
    <button class="btn btn--radius-2 btn--blue-2" type="submit">Save
Blog</button>
</div>
</div>
</form>
</div>

```

c. Edit.cshtml

```

@model MineBlog.Models.Blog

@{
    ViewData["Title"] = "Edit";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}
<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Blogs Management</h2>
                    <p>Edit your Blogs here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->
<div class="wrapper wrapper--w1000">
    <form asp-action="Edit">
        <div class="card card-6">
            <div class="card-body">
                <div>

```

```

        <a asp-action="Index">Back to List</a>
    </div>
</div>
<div class="card-body">
    <!-- Hidden Edit -->
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <input type="hidden" asp-for="Id" />
    <input type="hidden" asp-for="UrlSlug" />
    <input type="hidden" asp-for="Date" />
    <input type="hidden" asp-for="AuthorId" />
    <input type="hidden" asp-for="ImageUrl" />
    <!-- Hidden Edit -->
    <div class="form-row">
        <div class="name">Blog Title</div>
        <div class="value">
            <input class="input--style-6" type="text" asp-for="Title">
            <span asp-validation-for="Title" class="text-danger"></span>
        </div>
    </div>
    <div class="form-row">
        <div class="name">Short Description</div>
        <div class="value">
            <div class="input-group">
                <input class="input--style-6" type="text" asp-
for="Description" placeholder="">
                <span asp-validation-for="Description" class="text-
danger"></span>
            </div>
        </div>
    </div>
    <div class="form-row">
        <div class="name">Content</div>
        <div class="value">
            <div class="input-group">
                <textarea class="textarea--style-6" asp-for="Content"
placeholder="Write your content
here."></textarea>
                <span asp-validation-for="Content" class="text-
danger"></span>
            </div>
        </div>
    </div>
    <div class="form-row">
        <div class="name">Categories</div>
        <div class="value">
            <div class="input-group">
                <select asp-for="CategoryId" class="form-control input--
style-6" asp-items="ViewBag.CategoryId">
                </select>
                <span asp-validation-for="CategoryId" class="text-
danger"></span>
            </div>
            <div class="label--desc">Only pick one category</div>
        </div>
    </div>
    <div class="form-row">
        <div class="name">Tags</div>
        <div class="value">
            <div class="input-group">
                @foreach (var tag in ViewBag.Tags)
                {
                    <label>
                        <input type="checkbox" name="tags" value="@tag.Id"
@tag.Name

```

```

        </label>
    }
</div>
</div>
</div>
</div>
<div class="card-footer">
    <button class="btn btn--radius-2 btn--blue-2" type="submit">Save
Blog</button>
</div>
</div>
</form>
</div>

```

d. Delete.cshtml

```

@model MineBlog.Models.Blog

@{
    ViewData["Title"] = "Delete";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}
<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Blog Management</h2>
                    <p>Are you sure you want to delete this?</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <div class="card card-6">
        <div class="card-body">
            <div>
                <a asp-action="Index">Back to List</a>
            </div>
        </div>
        <div class="card-footer">
            <table class="table" style="background-color: white;">
                <thead>
                    <tr>
                        <th class="col-4">Title</th>
                        <th class="col-2">Date</th>
                        <th class="col-3">Category</th>
                        <th class="col-3">Author</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>@Html.DisplayFor(model => model.Title)</td>
                        <td>@Html.DisplayFor(model => model.Date)</td>
                        <td>@Html.DisplayFor(model => model.Category.Name)</td>
                        <td>@Html.DisplayFor(model => model.Author.Username)</td>
                    </tr>
                </tbody>
            </table>
            <form asp-action="Delete">

```

```

        <input type="hidden" asp-for="Id" />
        <input type="submit" value="Delete" class="btn btn-danger" />
    </form>
</div>
</div>
</div>

```

e. Details.cshtml

```

@model MineBlog.Models.Blog

@{
    ViewData["Title"] = "Details";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Blog Details Section Begin -->
<section class="blog-details spad">
    <div class="container">
        <div class="row d-flex justify-content-center">
            <div class="col-lg-10">
                <div class="blog__details__title">
                    <h6>
                        @Html.DisplayFor(model => model.Category.Name)<span>
                            , @Html.DisplayFor(model => model.Date)
                            , by @Html.DisplayFor(model => model.Author.Username)
                        </span>
                    </h6>
                    <h2>@Html.DisplayFor(model => model.Title)</h2>
                </div>
            </div>
            <div class="col-lg-12">
                <div class="blog__details__pic">
                    <img src=@Html.Raw(Model.ImageUrl) alt="">
                </div>
            </div>
            <div class="col-lg-8">
                <div class="blog__details__content">
                    @{
                        // Assuming your model.Content is a string containing line
                        breaks
                        string[] lines = Model.Content.Split(new[] {
                            Environment.NewLine }, StringSplitOptions.None);
                    }
                    @foreach (var line in lines)
                    {
                        if (line.StartsWith($"{title$, "})
                        {
                            string titleContent = line.Substring($"{title$, ".Length");
                            <div class="blog__details__text">
                                <h2>@titleContent</h2>
                            </div>
                        }
                        else if (line.StartsWith("https://"))
                        {
                            string img = line.Substring("".Length);
                            <div class="blog__details__pic__mini">
                                </img>
                            </div>
                        }
                        else
                        {
                            <div class="blog__details__text">
                                <p>@line</p>
                            </div>
                        }
                    }
                </div>
            </div>
        </div>
    </div>
</section>

```


4. ACatagories (phía admin)

a. *Index.cshtml*

```
@model IEnumerable<MineBlog.Models.Category>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Categories Management</h2>
                    <p>Manage your Categories here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <div class="card card-6">
        <div class="card-body">
            <div class="container">
                <a asp-action="Create">Create New</a>
            </div>
        </div>
        <div class="card-footer">
            <table class="table " style="background-color: white;">
                <thead>
                    <tr>
                        <th class="col-3">Name</th>
                        <th class="col-5">Image</th>
                        <th class="col-4">Tool</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach (var item in Model)
                    {
                        <tr>
                            <td>@Html.DisplayFor(modelItem => item.Name)</td>
                            <td>@Html.DisplayFor(modelItem => item.ImageUrl)</td>
                            <td>
                                <a asp-action="Edit" asp-route-id="@item.Id">Edit</a>
                                <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
                            </td>
                        </tr>
                    }
                </tbody>
            </table>
        </div>
    </div>
</div>
```

b. *Create.cshtml*

```
@model MineBlog.Models.Category
```

```

@{
    ViewData["Title"] = "Create";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}
<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Category Management</h2>
                    <p>Manage your Category here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Create" enctype="multipart/form-data">
        <input type="hidden" asp-for="ImageUrl" value="null" />
        <div class="card card-6">
            <div class="card-body">
                <div>
                    <a asp-action="Index">Back to List</a>
                </div>
                <div class="card-body">
                    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                    <div class="form-row">
                        <div class="name">Category Name</div>
                        <div class="value">
                            <input class="input--style-6" type="text" asp-for="Name">
                            <span asp-validation-for="Name" class="text-danger"></span>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Upload Image</div>
                        <div class="value">
                            <div class="input-group js-input-file">
                                <input class="input-file" type="file" name="image"
                                accept="image/*" id="file">
                                <label class="label--file" for="file">Choose file</label>
                                <div class="label--desc">Upload the image file for the blog.
                                Max file size 5 MB</div>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="card-footer">
                    <div class="form-row">
                        <button class="btn btn--radius-2 btn--blue-2" type="submit"
                        style="margin-top: 10px;">Create Category</button>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>

```

c. Edit.cshtml

@model MineBlog.Models.Category

```

@{
    ViewData["Title"] = "Edit";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Category Management</h2>
                    <p>Manage your Category here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Edit">
        <div class="card card-6">
            <div class="card-body">
                <div>
                    <a asp-action="Index">Back to List</a>
                </div>
            </div>
            <div class="card-body">
                <!-- Hidden Edit -->
                <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                <input type="hidden" asp-for="Id" />
                <!-- Hidden End -->
                <div class="form-row">
                    <div class="name">Category Name</div>
                    <div class="value">
                        <input class="input--style-6" type="text" asp-for="Name">
                        <span asp-validation-for="Name" class="text-
danger"></span>
                    </div>
                </div>
                <div class="form-row">
                    <div class="name">Category Image</div>
                    <div class="value">
                        <input class="input--style-6" type="text" asp-
for="ImageUrl">
                        <span asp-validation-for="ImageUrl" class="text-
danger"></span>
                    </div>
                </div>
            </div>
            <div class="card-footer">
                <div class="form-row">
                    <button class="btn btn--radius-2 btn--blue-2" type="submit"
style="margin-top: 10px;">Save Category</button>
                </div>
            </div>
        </div>
    </form>
</div>

```

d. Delete.cshtml

```
@model MineBlog.Models.Category

@{
    ViewData["Title"] = "Delete";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Category Management</h2>
                    <p>Are you sure you want to delete this?</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <div class="card card-6">
        <div class="card-body">
            <div>
                <a asp-action="Index">Back to List</a>
            </div>
        </div>
        <div class="card-footer">
            <table class="table" style="background-color: white;">
                <thead>
                    <tr>
                        <th class="col-8">Category Name</th>
                        <th class="col-4">ImageUrl</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>@Html.DisplayFor(model => model.Name)</td>
                        <td>@Html.DisplayFor(model => model.ImageUrl)</td>
                    </tr>
                </tbody>
            </table>
            <form asp-action="Delete">
                <input type="hidden" asp-for="Id" />
                <input type="submit" value="Delete" class="btn btn-danger" />
            </form>
        </div>
    </div>
</div>
```

5. ATags (phía admin)

a. Index.cshtml

```
@model IEnumerable<MineBlog.Models.Tag>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}
```

```

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
  <div class="container">
    <div class="row">
      <div class="col-lg-12 text-center">
        <div class="normal__breadcrumb__text">
          <h2>Tag Management</h2>
          <p>Manage your Tag here.</p>
        </div>
      </div>
    </div>
  </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
  <div class="card card-6">
    <div class="card-body">
      <div class="container">
        <a asp-action="Create">Create New</a>
      </div>
    <div class="card-footer">
      <table class="table" style="background-color: white;">
        <thead>
          <tr>
            <th class="col-8">Name</th>
            <th class="col-4">Tool</th>
          </tr>
        </thead>
        <tbody>
          @foreach (var item in Model)
          {
            <tr>
              <td>@Html.DisplayFor(modelItem => item.Name)</td>
              <td>
                <a asp-action="Edit" asp-route-id="@item.Id">Edit</a>
                <a asp-action="Delete" asp-route-
id="@item.Id">Delete</a>
              </td>
            </tr>
          }
        </tbody>
      </table>
    </div>
  </div>
</div>

```

b. Create.cshtml

```

@model MineBlog.Models.Tag

@{
  ViewData["Title"] = "Create";
  Layout = "~/Views/Shared/_adminLayout.cshtml";
}
<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
  <div class="container">
    <div class="row">
      <div class="col-lg-12 text-center">
        <div class="normal__breadcrumb__text">

```

```

                <h2>Tag Management</h2>
                <p>Manage your Tag here.</p>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Create">
        <div class="card card-6">
            <div class="card-body">
                <div>
                    <a asp-action="Index">Back to List</a>
                </div>
                <div class="card-body">
                    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                    <div class="form-row">
                        <div class="name">Tag Name</div>
                        <div class="value">
                            <input class="input--style-6" type="text" asp-for="Name">
                            <span asp-validation-for="Name" class="text-
danger"></span>
                        </div>
                    </div>
                <div class="card-footer">
                    <div class="form-row">
                        <button class="btn btn--radius-2 btn--blue-2" type="submit"
style="margin-top: 10px;">Create Tag</button>
                    </div>
                </div>
            </div>
        </form>
    </div>

```

c. Edit.cshtml

```

@model MineBlog.Models.Tag

@{
    ViewData["Title"] = "Edit";
    Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Tag Management</h2>
                    <p>Manage your Tag here.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
    <form asp-action="Edit">
        <div class="card card-6">

```

```

<div class="card-body">
  <div>
    <a asp-action="Index">Back to List</a>
  </div>
</div>
<div class="card-body">
  <!-- Hidden Edit -->
  <div asp-validation-summary="ModelOnly" class="text-danger"></div>
  <input type="hidden" asp-for="Id" />
  <!-- Hidden End -->
  <div class="form-row">
    <div class="name">Tag Name</div>
    <div class="value">
      <input class="input--style-6" type="text" asp-for="Name">
      <span asp-validation-for="Name" class="text-
danger"></span>
    </div>
  </div>
</div>
<div class="card-footer">
  <div class="form-row">
    <button class="btn btn--radius-2 btn--blue-2" type="submit"
style="margin-top: 10px;">Save Tag</button>
  </div>
</div>
</form>
</div>

```

d. Delete.cshtml

```

@model MineBlog.Models.Tag

@{
  ViewData["Title"] = "Delete";
  Layout = "~/Views/Shared/_adminLayout.cshtml";
}

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" style="background-image: url(/img/normal-
breadcrumb.jpg)">
  <div class="container">
    <div class="row">
      <div class="col-lg-12 text-center">
        <div class="normal__breadcrumb__text">
          <h2>Tag Management</h2>
          <p>Are you sure you want to delete this?</p>
        </div>
      </div>
    </div>
  </div>
</section>
<!-- Normal Breadcrumb End -->

<div class="wrapper wrapper--w1000">
  <div class="card card-6">
    <div class="card-body">
      <div>
        <a asp-action="Index">Back to List</a>
      </div>
    </div>
    <div class="card-footer">
      <table class="table" style="background-color: white;">
        <thead>
          <tr>

```



```

        <th class="col-8">Tag Name</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>@Html.DisplayFor(model => model.Name)</td>
    </tr>
</tbody>
</table>
<form asp-action="Delete">
    <input type="hidden" asp-for="Id" />
    <input type="submit" value="Delete" class="btn btn-danger" />
</form>
</div>
</div>
</div>

```

6. UBlogs (phía người dùng)

a. *Index.cshtml*

```

@model IEnumerable<MineBlog.Models.Blog>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!-- Breadcrumb Begin -->
<div class="breadcrumb-option">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <div class="breadcrumb__links">
                    <a asp-controller="UHome" asp-action="Index"><i class="fa fa-home"></i> Home</a>
                    <a asp-controller="UBlogs" asp-action="Index">All Blogs</a>
                </div>
            </div>
        </div>
    </div>
</div>
<!-- Breadcrumb End -->

<section class="product-page spad">
    <div class="container">
        <div class="row">
            <div class="col-lg-8">
                <div class="product__page__content">
                    <div class="product__page__title">
                        <div class="row">
                            <div class="col-lg-8 col-md-8 col-sm-6">
                                <div class="section-title">
                                    <h4>All Blogs List</h4>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <div class="row">
                @foreach (var item in Model)
                {
                    <div class="col-12">
                        <div class="product__item">
                            <div class="product__item__pic set-bg" data-
setbg=@Html.Raw(item.ImageUrl)>

```

```

        </div>
        <div class="product__item__text">
            <ul>
                <li><a style="color:white;"
href="#">@Html.DisplayFor(modelItem => item.Category.Name)</a></li>
            </ul>
            <h3>
                <a asp-action="Details" asp-route-
id="@item.Id">@Html.DisplayFor(modelItem => item.Title)</a>
            </h3>
            <div class="blog__details__tags__one">
                @if (item.Tags != null && item.Tags.Any())
                {
                    <div class="blog__details__text">
                        <p>@Html.DisplayFor(modelItem =>
item.Description)</p>
                    </div>
                    foreach (var tag in item.Tags)
                    {
                        <a href="#">@tag.Name</a>
                    }
                }
            </div>
        </div>
    </div>
}
</div>
<div class="product__pagination">
    <a href="#" class="current-page">1</a>
    <a href="#">2</a>
    <a href="#">3</a>
    <a href="#">4</a>
    <a href="#">5</a>
    <a href="#"><i class="fa fa-angle-double-right"></i></a>
</div>
</div>
<div class="col-lg-4 col-md-6 col-sm-8">
    <div class="product__sidebar">
        <div class="product__sidebar__view">
            <div class="section-title">
                <h5>Top Views</h5>
            </div>
            <div class="filter__gallery">
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="img/sidebar/tv-1.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a href="#">The ultimate guide to get mending in
Minecraft</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="img/sidebar/tv-2.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a href="#">Everything about Bee in
Minecraft</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="img/sidebar/tv-3.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a href="#">Minecraft LIVE in October</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="img/sidebar/tv-4.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>

```



```

        <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-3.jpg">
            <div class="view"><i class="fa fa-eye"></i> 9141</div>
            <h5><a asp-action="Details" asp-route-
id="1013">Minecraft LIVE in October</a></h5>
            </div>
            <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-4.jpg">
            <div class="view"><i class="fa fa-eye"></i> 9141</div>
            <h5><a asp-action="Details" asp-route-
id="1014">Minecraft: Finding And Conquering A Stronghold</a></h5>
            </div>
            <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-5.jpg">
            <div class="view"><i class="fa fa-eye"></i> 9141</div>
            <h5><a asp-action="Details" asp-route-id="1015">What
is Minecraft: Education Edition?</a></h5>
            </div>
        </div>
    </div>
</div>
</section>
<!-- Product Section End -->

```

c. Details.cshtml

```

@model MineBlog.Models.Blog
@{
    ViewData["Title"] = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!-- Blog Details Section Begin -->
<section class="blog-details spad">
    <div class="container">
        <div class="row d-flex justify-content-center">
            <div class="col-lg-10">
                <div class="blog__details__title">
                    <h6>
                        <a style="text-decoration:none; color:white;" asp-
controller="UBlogs" asp-action="List" asp-route-categoryId="@ViewBag.CategoryId">
                            @Html.DisplayFor(model => model.Category.Name)</a><span>
                                , @Html.DisplayFor(model => model.Date)
                                , by @Html.DisplayFor(model => model.Author.Username)
                            </span>
                        </h6>
                    <h2>@Html.DisplayFor(model => model.Title)</h2>
                </div>
            </div>
            <div class="col-lg-12">
                <div class="blog__details__pic">
                    <img src=@Html.Raw(Model.ImageUrl) alt="">
                </div>
            </div>
            <div class="col-lg-8">
                <div class="blog__details__content">
                    @{
                        // Assuming your model.Content is a string containing line
breaks
                        string[] lines = Model.Content.Split(new[] {
Environment.NewLine }, StringSplitOptions.None);
                    }
                </div>
            </div>
        </div>
    </div>
</section>

```

```

@foreach (var line in lines)
{
    if (line.StartsWith("$title$, "))
    {
        string titleContent = line.Substring("$title$, ".Length);
        <div class="blog__details__text">
            <h2>@titleContent</h2>
        </div>
    }
    else if (line.StartsWith("https://"))
    {
        string img = line.Substring("".Length);
        <div class="blog__details__pic__mini">
            </img>
        </div>
    }
    else
    {
        <div class="blog__details__text">
            <p>@line</p>
        </div>
    }
}

<div class="blog__details__tags">
    @if (Model.Tags != null && Model.Tags.Any())
    {
        foreach (var tag in Model.Tags)
        {
            <a href="">@tag.Name</a>
        }
    }
</div>
<!-- Display existing comments -->
<div class="blog__details__comment">
    @if (Model.Comments != null && Model.Comments.Any())
    {
        <h4>@Model.Comments.Count Comments</h4>
        @foreach (var comment in Model.Comments)
        {
            <div class="blog__details__comment__item">
                <div class="blog__details__comment__item__pic">
                    

                </div>
                <div class="blog__details__comment__item__text">
                    <span>@comment.Date</span>
                    <h5>@comment.Account.Username</h5>
                    <p>
                        @comment.Content
                    </p>
                </div>
            </div>
        }
    }
</div>
<div class="blog__details__form">
    <h4>Leave A Comment</h4>
    @Html.Partial("_Comments", new MineBlog.Models.Comment())
</div>
</div>
</div>
</div>
</div>
</section>

```

<!-- Blog Details Section End -->

7. UCategories (phía người dùng, chỉ có Index.cshtml)

@model IEnumerable<MineBlog.Models.Category>

@{

 ViewData["Title"] = "Index";

 Layout = "~/Views/Shared/_Layout.cshtml";

}

<!-- Breadcrumb Begin -->

<div class="breadcrumb-option">

 <div class="container">

 <div class="row">

 <div class="col-lg-12">

 <div class="breadcrumb_links">

 <a asp-controller="UHome" asp-action="Index"><i class="fa fa-home"></i> Home

 <a asp-controller="UCategories" asp-action="Index">Categories

 </div>

 </div>

 </div>

 </div>

</div>

<!-- Breadcrumb End -->

<section class="product-page spad">

 <div class="container">

 <div class="row">

 <div class="col-lg-8">

 <div class="product__page__content">

 <div class="product__page__title">

 <div class="row">

 <div class="col-lg-8 col-md-8 col-sm-6">

 <div class="section-title">

 <h4>Category List</h4>

 </div>

 </div>

 </div>

 </div>

 <div class="row">

 @foreach (var item in Model)

 {

 <div class="col-12">

 <div class="product__item">

 <div class="product__item__pic set-bg" data-setbg=@Html.Raw(item.ImageUrl)>

 </div>

 <div class="product__item__text">

 <h3>

 <a asp-controller="UBlogs" asp-action="List" asp-route-categoryId="@item.Id">

 @Html.DisplayFor(modelItem =>

item.Name)

 </h3>

 <div class="blog__details__tags__one">

 </div>

 </div>

 </div>

 </div>

 }

 </div>

 </div>

 <div class="product__pagination">

```

        <a href="#" class="current-page">1</a>
        <a href="#">2</a>
        <a href="#">3</a>
        <a href="#">4</a>
        <a href="#">5</a>
        <a href="#"><i class="fa fa-angle-double-right"></i></a>
    </div>
</div>
<div class="col-lg-4 col-md-6 col-sm-8">
    <div class="product__sidebar">
        <div class="product__sidebar__view">
            <div class="section-title">
                <h5>Top Views</h5>
            </div>
            <div class="filter__gallery">
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-1.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a asp-action="Details" asp-route-id="1011">The
ultimate guide to get mending in Minecraft</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-2.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a asp-action="Details" asp-route-
id="1012">Everything about Bee in Minecraft</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-3.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a asp-action="Details" asp-route-
id="1013">Minecraft LIVE in October</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-4.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a asp-action="Details" asp-route-
id="1014">Minecraft: Finding And Conquering A Stronghold</a></h5>
                </div>
                <div class="product__sidebar__view__item set-bg mix" data-
setbg="../img/sidebar/tv-5.jpg">
                    <div class="view"><i class="fa fa-eye"></i> 9141</div>
                    <h5><a asp-action="Details" asp-route-id="1015">What
is Minecraft: Education Edition?</a></h5>
                </div>
            </div>
        </div>
    </div>
</div>
</section>
<!-- Product Section End -->

```

8. ULogin (phía người dùng, chỉ có Index.cshtml)

```
@model MineBlog.Models.Account
```

```

@{
    ViewData["Title"] = "Login";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```
<!-- Normal Breadcrumb Begin -->
```



```

<section class="normal-breadcrumb set-bg" data-setbg="@Url.Content("~/img/normal-
breadcrumb.jpg")">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Login</h2>
                    <p>Welcome to MineBlog.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->
<!-- Login Section Begin -->
<section class="login spad">
    <div class="container">
        <div class="row">
            <div class="col-lg-6">
                <div class="login__form">
                    <h3>Login</h3>
                    <form asp-action="Login" method="post">
                        <div asp-validation-summary="ModelOnly" class="text-
danger"></div>
                        <div class="input__item">
                            <input asp-for="Username" type="text" required
placeholder="Username">
                            <span class="icon_mail"></span>
                            <span asp-validation-for="Username" class="text-
danger"></span>
                        </div>
                        <div class="input__item">
                            <input asp-for="Password" type="password" required
placeholder="Password">
                            <span class="icon_lock"></span>
                            <span asp-validation-for="Password" class="text-
danger"></span>
                        </div>
                        <button type="submit" class="site-btn" value="Login">Login
Now</button>
                    </form>
                    <a href="#" class="forget_pass">Forgot Your Password?</a>
                </div>
            </div>
            <div class="col-lg-6">
                <div class="login__register">
                    <h3>Dont't Have An Account?</h3>
                    <a asp-controller="USignup" asp-action="Index" class="primary-
btn">Register Now</a>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Login Section End -->

```

9. USignup (phía người dùng, chỉ có index.cshtml)

```
@model MineBlog.Models.Account
```

```

@{
    ViewData["Title"] = "Sign Up";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

<!-- Normal Breadcrumb Begin -->
<section class="normal-breadcrumb set-bg" data-setbg="@Url.Content("~/img/normal-
breadcrumb.jpg")">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="normal__breadcrumb__text">
                    <h2>Sign Up</h2>
                    <p>Welcome to MineBlog.</p>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Normal Breadcrumb End -->
<!-- Signup Section Begin -->
<section class="signup spad">
    <div class="container">
        <div class="row">
            <div class="col-lg-6">
                <div class="login__form">
                    <h3>Sign Up</h3>
                    <form asp-action="Signup">
                        <div asp-validation-summary="ModelOnly" class="text-
danger"></div>
                        <div class="input__item">
                            <input asp-for="Email" type="email" placeholder="Email
address">
                            <span class="icon_mail"></span>
                        </div>
                        <div class="input__item">
                            <input asp-for="Username" type="text" placeholder="Your
Username">
                            <span class="icon_profile"></span>
                        </div>
                        <div class="input__item">
                            <input asp-for="Password" type="password"
placeholder="Password">
                            <span class="icon_lock"></span>
                        </div>
                    </form>
                    <button type="submit" value="Signup" class="site-btn">Sign
Up</button>
                </div>
            </div>
            <div class="col-lg-6">
                <div class="login__register">
                    <h3>Already have an account?</h3>
                    <a asp-controller="ULogin" asp-action="Index" class="primary-
btn">Login Now</a>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- Signup Section End -->

```