TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

**KHOA CÔNG NGHỆ THÔNG TIN**

-------------------------------------------------



**ĐỒ ÁN TỐT NGHIỆP**

Đề tài

Website giới thiệu game

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | : | Nguyễn Trọng Phúc |
| Sinh viên thực hiện | : | Nguyễn Tiến Đạt |
| Mã sinh viên | : | 191201375 |
| Lớp | : | Công Nghệ Thông Tin |
| Khóa | : | 60 |

**Hà Nội - 2023**

# **LỜI CẢM ƠN**

Để đồ án này đạt được kết quả như hiện nay em đã nhận được rất nhiều sự hỗ trợ và hướng dẫn của thầy Nguyễn Trọng Phúc. Xuất phát từ sự chân thành, em xin bày tỏ sự biết ơn sâu sắc nhất đến thầy.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến các thầy, cô giáo và nhà trường đã tạo điều kiện tốt nhất để em được môi trường học tập tốt nhất.

Mặc dù trong quá trình thực hiện đề tài em đã rất cố gắng nhưng do trình độ còn hạn chề về kiến thức, công nghệ cũng như nghiệp vụ nên trong quá trình thực hiện đề tài còn nhiều sự thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn!

Hà Nội, ngày 19 tháng 05 năm 2023

Sinh viên thực hiện

Nguyễn Tiến Đạt

**MỤC LỤC**

[**LỜI CẢM ƠN** 2](#_Toc135597515)

[**LỜI NÓI ĐẦU** 5](#_Toc135597516)

[**CHƯƠNG 1: KHẢO SÁT VÀ ĐẶT VẤN ĐỀ** 6](#_Toc135597517)

[**1.1. Bài toán** 6](#_Toc135597518)

[**1.2. Các tiêu chí khảo sát** 6](#_Toc135597519)

[**1.3. Tiến hành khảo sát** 6](#_Toc135597520)

[**1.3.1. Khảo sát website kenhgamez.com** 6](#_Toc135597521)

[**1.3.2. Khảo sát website daominhha.net** 10](#_Toc135597522)

[**1.4. Kết luận** 11](#_Toc135597523)

[**CHƯƠNG 2: CÁC CÔNG NGHỆ SỬ DỤNG** 13](#_Toc135597524)

[**2.1. MySQL** 13](#_Toc135597525)

[**2.1.1. Giới thiệu** 13](#_Toc135597526)

[**2.1.2. Kiến trúc của MYSQL** 13](#_Toc135597527)

[**Lớp Client** 14](#_Toc135597528)

[**Lớp SERVER** 14](#_Toc135597529)

[**Lớp Storage** 15](#_Toc135597530)

[**2.1.3. Ưu, nhược điểm** 15](#_Toc135597531)

[**2.1.4. So sánh với sql server** 16](#_Toc135597532)

[**2.2. MongoDB** 17](#_Toc135597533)

[**2.2.1. Giới thiệu** 17](#_Toc135597534)

[**2.2.2. Kiến trúc của Mongodb** 17](#_Toc135597535)

[**2.2.3. Ưu, nhược điểm** 18](#_Toc135597536)

[**2.3. NodeJS** 19](#_Toc135597537)

[**2.3.1. Giới thiệu** 19](#_Toc135597538)

[**2.3.2. Ưu, nhược điểm** 19](#_Toc135597539)

[2.4: ExpressJS 19](#_Toc135597540)

[**2.4.1. Giới thiệu** 19](#_Toc135597541)

[**2.4.2. Mô hình MC ( Model, Controller )** 20](#_Toc135597542)

[**2.5 NextJS** 20](#_Toc135597543)

[**2.5.1. Giới thiệu** 20](#_Toc135597544)

[**2.5.2. Ưu, nhược điểm NextJs** 21](#_Toc135597545)

[**CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 23](#_Toc135597546)

[**3.1 Sơ đồ phân rã chức năng BFD** 23](#_Toc135597547)

[**2.2: Phân tích nghiệp vụ các chức năng** 23](#_Toc135597548)

[**2.2.1: Quản lý hệ thống** 23](#_Toc135597549)

[**2.2.2: Quản lý người dùng** 24](#_Toc135597550)

[**2.2.3 Quản lý bài viết** 25](#_Toc135597551)

[**2.2.4 Quản lý thể loại** 26](#_Toc135597552)

[**2.2.5 Quản lý bình luận** 27](#_Toc135597553)

[**2.3: Sơ đồ luồng dữ liệu DFD** 28](#_Toc135597554)

[**2.3.1 Sơ đồ DFD mức ngữ cảnh** 28](#_Toc135597555)

[**2.3.2 Sơ đồ DFD mức 0** 29](#_Toc135597556)

[**2.3.3 Sơ đồ DFD mức 1** 30](#_Toc135597557)

[**2.3.4: Sơ đồ ERD** 35](#_Toc135597558)

[**2.3.5: Cơ sở dữ liệu** 36](#_Toc135597559)

[**CHƯƠNG 4: XÂY DỰNG WEBSITE** 37](#_Toc135597560)

[**4.1: Chức năng hiển thị danh sách bài viết** 37](#_Toc135597561)

[**4.2: Chức năng đăng bài viết** 37](#_Toc135597562)

[**4.3: Chức năng tìm kiếm bài viết** 38](#_Toc135597563)

[**4.4: Chức năng sửa bài viết** 38](#_Toc135597564)

[**4.5: Chức năng xóa bài viết** 39](#_Toc135597565)

[**4.6: Chức năng đăng ký** 40](#_Toc135597566)

[**4.7: Chức năng đăng nhập** 41](#_Toc135597567)

[**4.8: Chức năng bình luận** 41](#_Toc135597568)

[**4.8: Đăng nhập trên trang quản trị** 42](#_Toc135597569)

[**4.9: Trang quản trị** 42](#_Toc135597570)

[**4.10: Xem trước bài viết trên trang quản trị** 43](#_Toc135597571)

[**4.11: Quản lý người dùng** 43](#_Toc135597572)

[**4.12: Quản lý thể loại** 44](#_Toc135597573)

[**4.12: Quản lý bài viết** 44](#_Toc135597574)

[**KẾT LUẬN** 45](#_Toc135597575)

# **LỜI NÓI ĐẦU**

Công nghệ Thông tin (CNTT) là một trong những ngành đóng vai trò quan trọng trong sự phát triển của xã hội. Nó không chỉ thúc đẩy nhanh chóng quá trình tăng trưởng kinh tế, mà còn kéo theo sự biến đổi trong phương thức sáng tạo của cải, trong lối sống và tư duy của con người. Việc nhanh chóng đưa ứng dụng CNTT vào quá trình tự động hóa trong sản xuất kinh doanh, quảng bá sản phẩm, … là vấn đề đã, đang và sẽ luôn được quan tâm bởi lẽ CNTT có vai trò rất lớn trong mọi hoạt động kinh tế, xã hội - đặc biệt là cuộc Cách mạng Công nghiệp 4.0 đang tác động đến toàn thế giới hiện nay.

Có rất nhiều loại website, ví dụ như: website bán hàng, website quản lý, ứng dụng web… Trong đó có một loại website khá nổi bật đó là website giới thiệu. Website giới thiệu là một dạng website thường đi kèm với các tính năng phù hợp cho các loại giao diện chuyên về giới thiệu.

Do sự phát triển nhanh chóng của ngành công nghiệp 4.0, nhu cầu chơi các trò chơi điện tử ngày càng cao. Kéo theo đó là những loại game mới thú vị hấp dẫn người chơi.

Từ những lý do trên, em đã quyết định chọn đề tài làm website giới thiệu game.

Nội dung của đề tài gồm 4 chương

Chương 1: Khảo sát đề tài và đặt vấn đề

Trong chương này em sẽ trình bày bài toán về giới thiệu game online, giới hạn phạm vi bài toán. Dựa vào việc khảo sát những website thực tế để đưa ra các ưu, nhược điểm của vài trang web. Từ đó sẽ đưa ra được những chức năng phù hợp với website của mình.

Chương 2: Công nghệ sử dụng

Trong chương này, em sẽ trình bày về các công nghệ được sử dụng để xây dựng website

Chương 3: Thiết kế hệ thống

Ở chương này, em sẽ tập chung vào việc phân tích hệ thống, đưa ra các biểu đồ BFD, DFD, đặc tả của từng chức năng. Thiết kế cơ sở dữ liệu.

Chương 4: Kết quả

Em sẽ trình bày môi trường cài đặt cần thiết để website hoạt động. Tiếp theo sẽ là các hình ảnh về website mà em đã xây dựng được.

# **CHƯƠNG 1: KHẢO SÁT VÀ ĐẶT VẤN ĐỀ**

Chương đầu tiên sẽ thực hiện khảo sát 2 website giới thiệu game để từ đó phân tích các yêu cầu nghiệp vụ, xác định các chức năng cần có của đề tài.

**1.1. Bài toán**

Cùng với sự phát triển của khoa học - kỹ thuật, đời sống vật chất và tinh thần của con người ngày càng được nâng cao. Bên cạnh những nhu cầu về ăn mặc, sinh hoạt thì nhu cầu giải trí của con người cũng tăng lên. Và để đáp ứng điều này, vô số trò chơi (game) đã được ra đời. Do số lượng trò chơi này một nhiều nên sẽ không ít người gặp khó khăn trong việc tìm kiếm tựa game mình thích. Hơn nữa, có những người muốn giới thiệu game tới tất cả mọi người.

Từ đó bài toán đặt ra là cần có website giới thiệu game để phục vụ nhu cầu tìm kiếm và giới thiệu game.

**1.2. Các tiêu chí khảo sát**

- Giao diện

- Đối tượng phục vụ

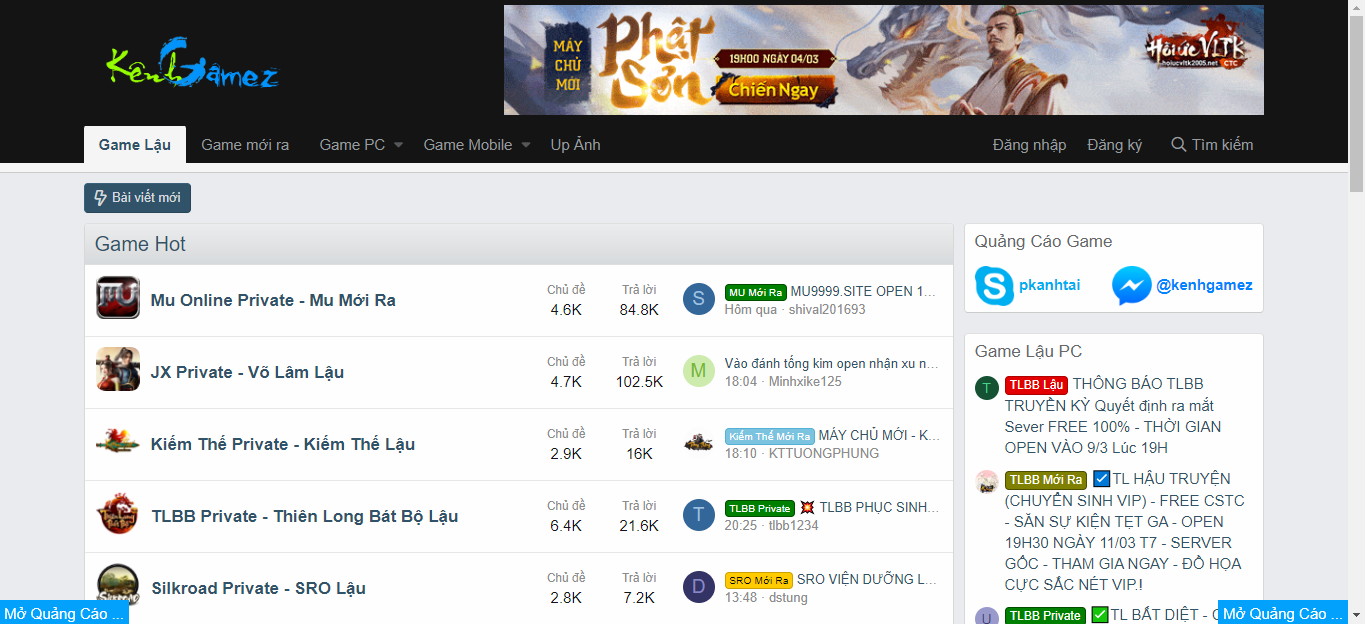
- Các chức năng chính

**1.3. Tiến hành khảo sát**

### **1.3.1. Khảo sát website kenhgamez.com**

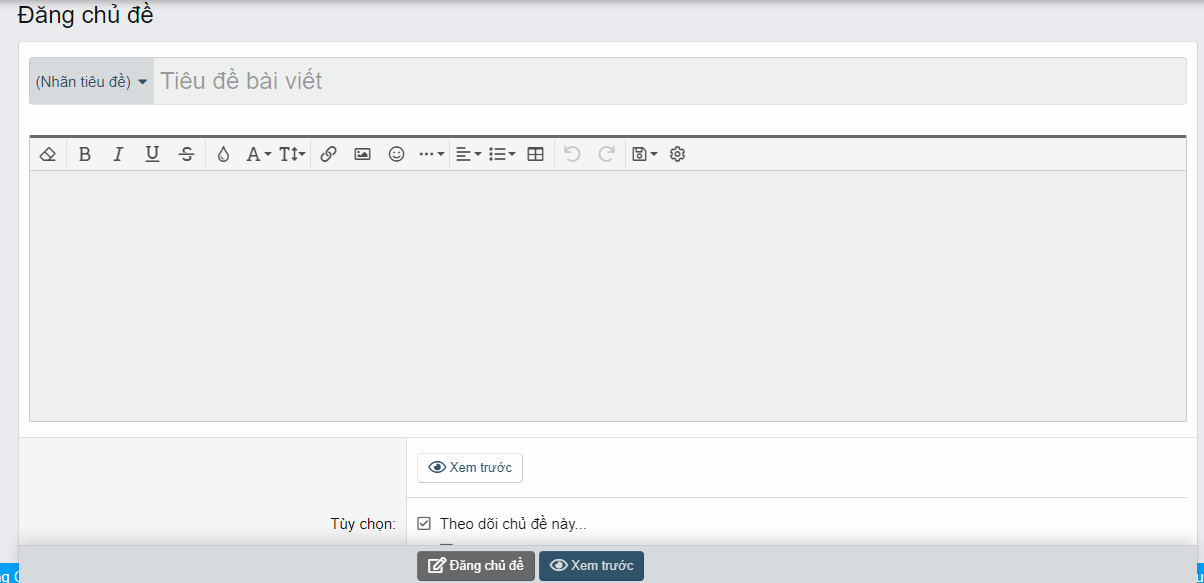
Đây là website giới thiệu game lâu đời. Có rất nhiều người đã đăng ký và đăng các bài viết của họ lên đây. Trang này chủ yếu giới thiệu các game từ khoảng 2010 trở về.

Một số hình ảnh của website

*Giao diện trang chủ*



*Giao diện đọc bài*



*Giao diện đăng bài viết*



*Giao diện chia sẻ bài viết*

- Các chức năng chính của web

\* Tìm kiếm bài viết

\* Đọc bài viết

\* Viết bài

\* Chia sẻ bài viết

- Ưu điểm web:

\* Giao diện dễ sử dụng

\* Người dùng có thể đăng bài viết để giới thiệu game của mình

\* Dễ dàng tìm kiếm các bài viết theo chủ đề, tác giả

\* Có tính năng chia sẻ bài

- Nhược điểm web:

\* Web chủ yếu tập chung vào các game từ đã ra đời từ 2010 trở về. Những game đó đa số phù hợp với những người từ 24 tuổi trở lên

\* Web chưa có tính năng lọc bài viết theo thể loại game.

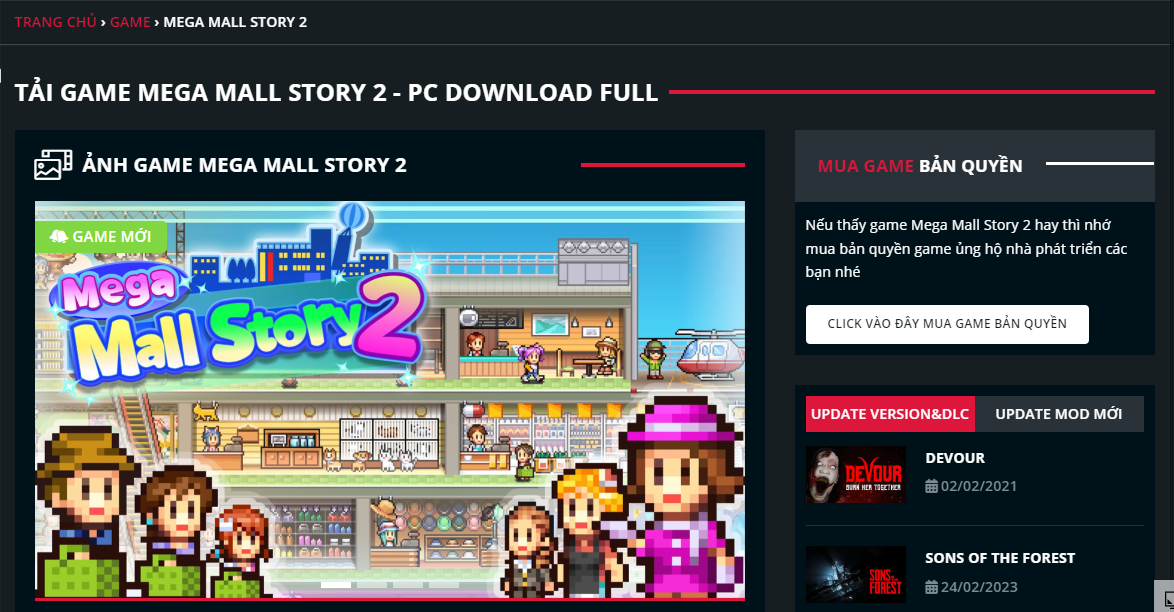
\* Chưa có tính năng bình luận

### **1.3.2. Khảo sát website daominhha.net**

Đây là trang web giới thiệu game đã được việt hóa. Trang web giới thiệu rất nhiều trò chơi hay. Có chức năng lọc trò chơi theo thể loại, nền tảng, cấu hình…



*Giao diện trang chủ*

**

*Giao diện đọc bài*

- Các chức năng chính của web

\* Tìm kiếm bài viết

\* Đọc bài viết

\* Lọc bài viết

- Ưu điểm web:

\* Giao diện đẹp mắt, dễ sử dụng

\* Dễ dàng tìm kiếm các bài viết theo thể loại, cấu hình, nền tảng

\* Người dùng có thể đánh giá và bình luận bài viết

- Nhược điểm web:

\* Không có tính năng chia sẻ bài viết

\* Người dùng không được đăng bài viết cá nhân

**1.4. Kết luận**

Từ những website em đã khảo sát, em sẽ xây dựng hệ thống website giới thiệu game với các chức năng sau:

* Người dùng:
* Tìm kiếm theo từ khóa
* Xem các bài viết theo thể loại
* Quản lý bình luận cá nhân
* Quản lý bài viết cá nhân
* Quản lý thông tin cá nhân
* Lịch sử xem bài viết
* Quản trị:
* Quản lý tài khoản
* Phân quyền
* Quản lý bài viết
* Quản lý thể loại

# **CHƯƠNG 2: CÁC CÔNG NGHỆ SỬ DỤNG**

## **2.1. MySQL**

### **2.1.1. Giới thiệu**

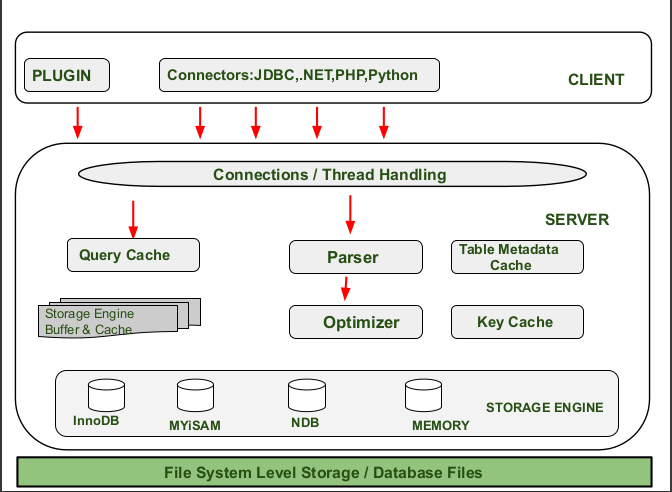
**MySQL** là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến trên thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/wiki/Novell_NetWare), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1),..

### **2.1.2. Kiến trúc của MYSQL**

MySQL là hệ thống Quản lý cơ sở dữ liệu quan hệ (Relational Database Management).

Kiến trúc của MYSQL mô tả mối quan hệ giữa các component với nhau. MYSQL tuân theo kiến trúc CLIENT-SERVER. Nó được thiết kế cho người dùng cuối có thể truy cập tài nguyên từ máy chủ bằng các dịch vụ mạng khác nhau. Kiến trúc của MYSQL chứa các lớp chính sau:

* Client
* Server
* Storage Layer



*Kiến trúc của MySQL*

#### **Lớp Client**

- Đây là tầng trên cùng của sơ đồ. Máy khách sẽ đưa ra các yêu cầu cho máy chủ với sự trợ giúp của **Lớp Client.** Máy khách đưa ra yêu cầu thông qua Command Prompt hoặc thông qua giao diện bằng cách sử dụng các câu lệnh SQL. Nếu các biểu thức hợp lệ và lệnh hợp lệ thì kết quả sẽ được in ra màn hình. Một sô dịch vụ quan trong của Lớp Client

* Connection Handling
* Authentication
* Security

- Connection Handling: Khi máy khách gửi yêu cầu đến máy chủ và máy chủ sẽ chấp nhận yêu cầu máy khách và máy khách được kết nối. Khi máy khách đã kết nối với máy chủ tại thời điểm đó, máy khách sẽ nhận được một luồng riêng. Với sự trợ giúp của luồng này tất các truy vấn từ máy khách sẽ được thực thi.

- Authentication: Thực hiện ở trên server khi máy khách đã kết nối tới MYSQL. Xác thực được thực hiện với sự trợ giúp của username và password.

- Security: Sau khi máy khách kết nối thành công tới MYSQL, Server sẽ kiểm tra xem ứng dụng đó có được đưa ra một số truy vấn nhất định đối với máy chủ không.

#### **Lớp SERVER**

- Lớp này chịu trách nhiệm cho tất cả các chức năng logic. Khi máy khách đưa ra các yêu cầu cho server, server sẽ cung cấp đầu ra ngay khi các lệnh được khớp.

- Thread handling: Khi client gửi yêu cầu đến server và server sẽ chấp nhận yêu cầu và client được kết nối. Khi client được kết nối tới server trong thời gian đó, client sẽ nhận ddwuocwc một thread của riêng nó. Thread này được cung cấp bởi thread handling. Các truy vấn của client do luồng thực thi cũng được xử lý bởi modun Thread handling.

- Parser: Máy chủ MySQL nhận truy vấn ở định dạng SQL. Sau khi nhận được một truy vấn, trước tiên nó cần được phân tích cú pháp. Quá trình này được gọi tắt là Trình phân tích cú pháp. MySQL phân tích cú pháp truy vấn để tạo cấu trúc bên trong (Parser tree). Trình phân tích cú pháp MySQL hoạt động như một trình biên dịch truyền đơn.

-Optimizer: Ngay sau khi quá trình phân tích cú pháp hoàn tất,  nhiều loại kỹ thuật tối ưu hóa khác nhau sẽ được áp dụng tại Khối trình tối ưu hóa. Những kỹ thuật này có thể bao gồm viết lại truy vấn, thứ tự quét bảng và chọn đúng chỉ mục để sử dụng, v.v.

#### **Lớp Storage**

- Lớp Công cụ lưu trữ này của Kiến trúc MYSQL làm cho nó trở nên độc đáo và thích hợp nhất cho các nhà phát triển . Do lớp MYSQL này, lớp được coi là RDBMS được sử dụng nhiều nhất và được sử dụng rộng rãi. Trong máy chủ MYSQL, đối với các tình huống và yêu cầu khác nhau, các loại công cụ lưu trữ khác nhau được sử dụng, đó là InnoDB , MYiSAM , NDB , Memory, v.v.

### **2.1.3. Ưu, nhược điểm**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| **- Dễ sử dụng**: **MySQL** là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.  **- Độ** **bảo mật cao**: **MySQL** rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều nhiều tính năng bảo mật thậm chí là ở cấp cao.  **- Đa tính năng**: **MySQL** hỗ trợ rất nhiều **chức năng SQL** được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.  **- Khả năng mở rộng và mạnh mẽ**: **MySQL** có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.  **- Nhanh chóng**: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi. | **- Giới hạn**: Theo thiết kế, **MySQL** không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.  **- Độ tin cậy**: Cách các chức năng cụ thể được xử lý với **MySQL** (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán,…) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.  **- Dung lượng hạn chế**: Nếu số bản ghi của bạn lớn dần lên thì việc truy xuất dữ liệu của bạn là khá khó khăn, khi đó chúng ta sẽ phải áp dụng nhiều biện pháp để tăng tốc độ truy xuất dữ liệu như là chia tải database này ra nhiều server, hoặc **tạo cache MySQL** |

### **2.1.4. So sánh với sql server**

|  |  |  |  |
| --- | --- | --- | --- |
|  | MySQL | SQL Server | |
| **Bản chất** | **MySQL**: MySQL thuộc sở hữu của Oracle, là một phần mềm mã nguồn mở, chạy trên hơn 20 nền tảng bao gồm Linux, Windows, OS / X, HP-UX, AIX, Netware. MySQL có thể kết hợp với mọi ngôn ngữ lập trình khác, thông thường là PHP. | **SQL Server**: SQL Server được phát triển bởi Microsoft. Microsoft đã xây dựng nhiều công cụ mạnh mẽ cho SQL Server, hỗ trợ lớn hơn cho RDBMS, bao gồm các công cụ phân tích dữ liệu. SQL Server hoạt động tốt trong môi trường .NET | |
| **Cú pháp** | Mặc dù cả **MySQL** và **SQL Server** đều dựa trên **SQL**, nhưng sự khác biệt về cú pháp vẫn là điểm nổi bật và đáng lưu ý. | | |
|  | |  |
| **Giá cả:** | **MySQL**: Ngược lại, MySQL sử dụng General Public License (mã nguồn mở), điều này làm cho nó hoàn toàn miễn phí để sử dụng. | | **SQL Server**: Microsoft yêu cầu người dùng mua giấy phép để truy cập các tính năng đầy đủ của SQL Server. |
| **Lưu trữ dữ liệu:** | MySQL cung cấp cho các nhà phát triển sự linh hoạt hơn nhiều, vì họ có thể sử dụng các công cụ khác nhau cho các bảng khác nhau dựa trên tốc độ, độ tin cậy hoặc một số thứ nguyên khác. Hai trong số các công cụ lưu trữ MySQL phổ biến nhất là InnoDB và MyISAM. | | SQL Server sử dụng một công cụ lưu trữ duy nhất do Microsoft phát triển. |
| **Sao lưu và phục hồi:** | Khi bạn sao lưu cơ sở dữ liệu MySQL, dữ liệu được trích xuất dưới dạng câu lệnh SQL. Do đó, việc sao lưu và khôi phục một lượng lớn dữ liệu có thể mất vĩnh viễn do thực hiện nhiều câu lệnh SQL. | | Người dùng SQL Server không phải trải qua tình huống đó. SQL Server sẽ không khóa cơ sở dữ liệu, vì vậy bạn vẫn có thể sử dụng nó trong quá trình sao lưu. |
| **Bảo mật:** | MySQL cho phép các tệp cơ sở dữ liệu của nó được chỉnh sửa và truy cập bởi các quy trình khác trong thời gian chạy. | | Điều này không xảy ra với SQL Server vì người dùng của nó được yêu cầu chạy một phiên bản để thực hiện chức năng. Nó làm cho SQL Server ít bị hack hơn vì dữ liệu không thể được thao tác hoặc truy cập trực tiếp. |

## **2.2. MongoDB**

### **2.2.1. Giới thiệu**

MongoDB là một cơ sở dữ liệu hướng tài liệu (document), một dạng cơ sở dữ liệu NoSQL. Vì thế, MongoDB sẽ tránh cấu trúc dựa trên bảng của cơ sở dữ liệu quan hệ để thích ứng với các tài liệu như JSON có một lược đồ rất linh hoạt được gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi bộ sưu tập sẽ có các kích thước và các tài liệu khác nhau. Dữ liệu được lưu trữ trong tài liệu dạng JSON nên truy vấn sẽ rất nhanh.

### **2.2.2. Kiến trúc của Mongodb**

Database: Hiểu một cách đơn giản, nó có thể được coi là vùng chứa vật lý cho dữ liệu. Mỗi database có tập file riêng trên file system với nhiều database hiện có trên một single MongoDB server.

Collection: Một nhóm các database document có thể được gọi là một collection. RDBMS tương đương với bộ sưu tập là một table. Toàn bộ collection tồn tại trong một single database. Không có schema nào khi nói đến các collection. Bên trong collection, các document khác nhau có thể có các trường khác nhau nhưng hầu hết các document trong một collection đều có cùng một mục đích hoặc phục vụ cùng một mục tiêu.

Document: Một tập hợp các cặp key – value có thể được chỉ định là một document. Các document được liên kết với các dynamic schema. Lợi ích của dynamic schema là document trong một collection không nhất thiết phải có cùng cấu trúc hoặc các trường. Ngoài ra, các trường phổ biến trong document của collection có thể có nhiều loại dữ liệu khác nhau.

**2.2.3. Ưu, nhược điểm**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| -Document oriented  -Hiệu suất cao  -Tính sẵn sàng cao – Nhân rộng  -Khả năng mở rộng cao – Sharding  -Năng động – Không có lược đồ cứng nhắc.  -Linh hoạt – thêm / xóa trường có ít hoặc không ảnh hưởng đến ứng dụng  -Dữ liệu không đồng nhất  -Không joins  -Phân phối được  -Biểu diễn dữ liệu trong JSON hoặc BSON  -Hỗ trợ không gian địa lý (Geospatial)  -Tích hợp dễ dàng với BigData Hadoop  -Ngôn ngữ truy vấn dựa trên tài liệu mạnh mẽ như SQL  -Các bản phân phối cloud như AWS, Microsoft, RedHat, dotCloud và SoftLayer, v.v … Trên thực tế, MongoDB được xây dựng cho cloud. Kiến trúc mở rộng quy mô tự nhiên của nó, được kích hoạt bởi sharding, liên kết tốt với quy mô và sự nhanh nhẹn có được nhờ điện toán đám mây. | - Một nhược điểm của NoSQL là hầu hết các giải pháp đều không tuân thủ ACID mạnh mẽ (Atomic, Consistency, Isolation, Durability) như các hệ thống RDBMS được thiết lập tốt hơn.  -Giao dịch phức tạp  -Không có chức năng hoặc thủ tục lưu trữ tồn tại nơi bạn có thể liên kết logic |

## **2.3. NodeJS**

### **2.3.1. Giới thiệu**

**NodeJS** là một môi trường runtime chạy [JavaScript](https://vietnix.vn/javascript-la-gi/) đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

### **2.3.2. Ưu, nhược điểm**

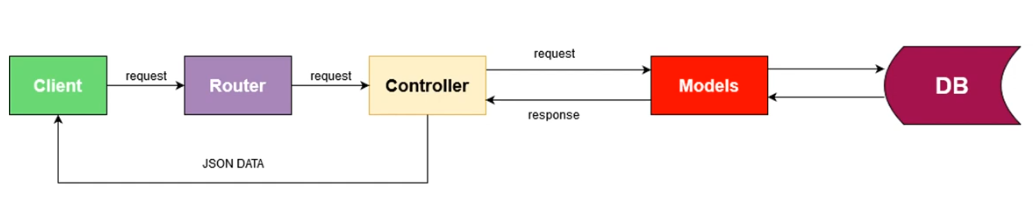
|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * - IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời. * - Sử dụng JavaScript – một ngôn ngữ lập trình dễ học. * - Chia sẻ cùng code ở cả phía client và server. * - [NPM](https://vietnix.vn/npm-la-gi/)(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ. * - Cộng đồng hỗ trợ tích cực. * - Cho phép stream các file có kích thước lớn | * - Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay. * - Khó thao tác với cơ sử dữ liệu quan hệ. * - Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác. * - Cần có kiến thức tốt về JavaScript. * - Không phù hợp với các tác vụ đòi hỏi nhiều CPU. |

## 2.4: ExpressJS

### **2.4.1. Giới thiệu**

Express JS là framework được xây dựng trên nền tảng của **Node.JS.** Nó cung cấp các tính năng để xây dựng các RESTful API, ứng dụng web.

### **2.4.2. Mô hình MC ( Model, Controller )**



Mô hình **MC** là mô hình giúp tổ chức code trở nên rõ ràng hơn, dễ sửa, nâng cấp …

**Client**: là người dùng sẽ gửi các yêu cầu ( request ) đến server.

**Router**: đề cập đến việc xác định cách ứng dụng phản hồi yêu cầu của máy khách tới một điểm cuối cụ thể, đó là một URI (hoặc đường dẫn) và một phương thức yêu cầu HTTP cụ thể (GET, POST, v.v.). Nhận yêu cầu ( request ) từ Client ( người dùng ) gửi đến controller.

**Controller**: Vì mỗi Router có các action khác nhau. Nên nhiệm vụ của Controller là chứa các action của router. Trong mỗi action sẽ yêu cầu Models làm một công việc nào đó ( lấy dữ liệu từ cơ sở dữ liệu, cập nhật dữ liệu cơ sở dữ liệu … ) và gửi lại về phía Client (JSON).

**Models**: thực hiện xử lý logic, truy vấn cơ sở dữ liệu trả kết quả cho Controller.

## **2.5 NextJS**

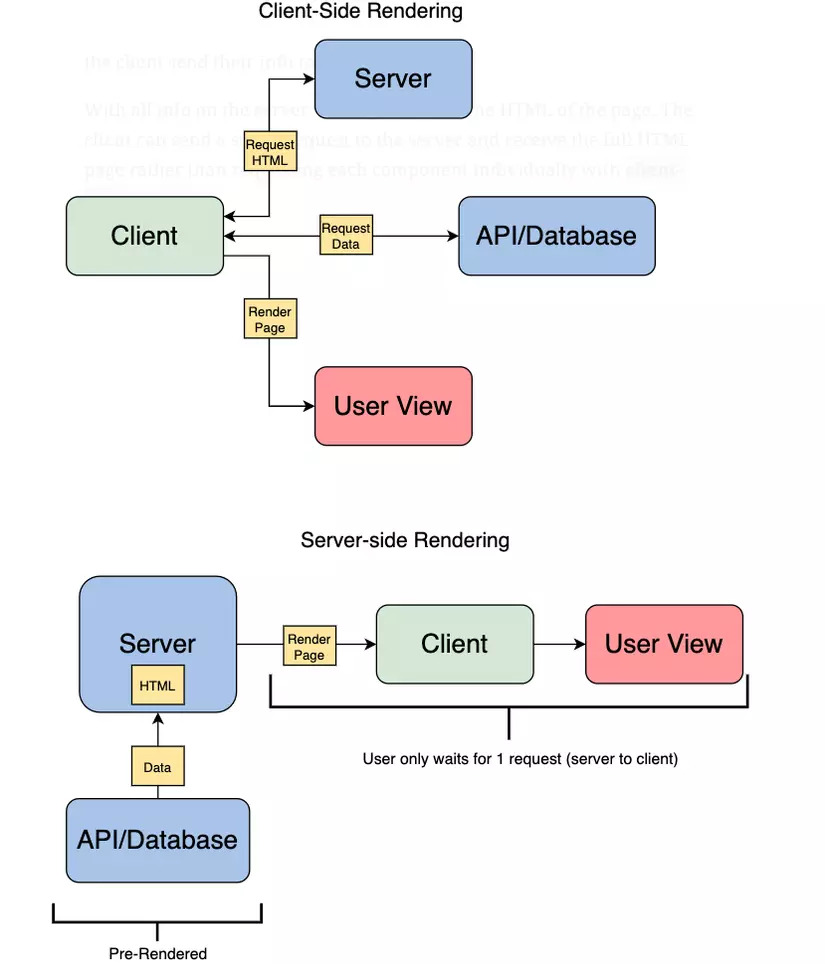
### **2.5.1. Giới thiệu**

Next.js là một framework front-end React được phát triển dưới dạng open-source bổ sung các khả năng tối ưu hóa như render phía máy chủ (SSR) và tạo trang web static. Next.js xây dựng dựa trên thư viện React, có nghĩa là các ứng dụng Next.js sử dụng core của React và chỉ thêm các tính năng bổ sung. Việc triển khai ứng dụng SSR cho phép máy chủ truy cập tất cả dữ liệu được yêu cầu và xử lý JavaScript cùng nhau để hiển thị trang. Sau đó, trang được gửi lại toàn bộ cho trình duyệt và ngay lập tức được hiển thị. SSR cho phép các trang web load trong thời gian nhỏ nhất và tăng trải nghiệm người dùng với khả năng phản hồi nhanh hơn.

Ngoài ra, sử dụng SSR cũng mang lại cho bạn lợi thế về SEO, giúp trang web của bạn hiển thị cao hơn trên các trang kết quả của công cụ tìm kiếm. SSR làm cho các trang web xếp hạng tốt hơn cho SEO vì chúng tải nhanh hơn và nhiều nội dung trang web có thể được quét bởi các SEO trackers. Thẻ <head> trong Next.js cũng cho phép bạn chỉnh sửa thẻ <head> của một trang web, điều mà bạn không thể thực hiện trong React. Thẻ <head> là một phần cốt lõi trong metadata của trang web và góp phần vào xếp hạng SEO của trang web.

### **2.5.2. Ưu, nhược điểm NextJs**

Ưu điểm chính của Next.js là hỗ trợ SSR tích hợp để tăng hiệu suất và SEO. Với tất cả thông tin trên server, nó sẽ xử lý để generate ra thông tin HTML của trang/ Sau đó Client có thể gửi một yêu cầu đến Server và nhận toàn bộ trang HTML thay vì yêu cầu từng thành phần riêng lẻ với Client Render.

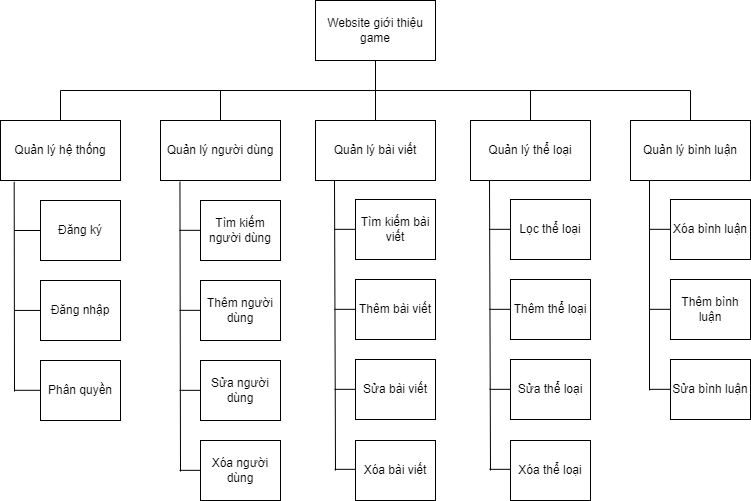


Nhược điểm thực sự duy nhất của Next.js là nó là một framework được cố định, có nghĩa là nó có một phương pháp và bộ công cụ cụ thể mà nó muốn bạn sử dụng để xây dựng các ứng dụng của mình. Tuy nhiên, các tùy chọn của Next.js sẽ phù hợp với phạm vi của hầu hết các dự án.

# **CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **3.1 Sơ đồ phân rã chức năng BFD**

Hệ thống gồm có 2 tác nhân: Quản trị viên và người dùng



## **2.2: Phân tích nghiệp vụ các chức năng**

### **2.2.1: Quản lý hệ thống**

\* Đăng ký:

- Tác nhân: Người dùng

- Mô tả:

+ Người dùng tiến hành nhập các thông tin như: tài khoản, mật khẩu, email… để tiến hành đăng ký.

+ Hệ thống tiến hành kiểm tra thông tin đăng ký

+ Nếu thông tin hợp lệ sẽ lưu xuống cơ sở dữ liệu, sau đó thông báo tới người dùng. Nếu thông tin không hợp lệ sẽ thông báo đăng ký thất bại tới người dùng.

\* Đăng nhập:

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng nhập hoặc quản trị viên thông tin tài khoản, mật khẩu để tiến hành đăng nhập

+ Hệ thống tiến hành kiểm tra thông tin đăng nhập

+Nếu thông tin đăng nhập chính xác thì sẽ điều hướng người dùng hoặc quản trị viên vào hệ thống

+ Nếu thông tin đăng nhập không chính xác sẽ yêu cầu người dùng nhập lại

\* Phân quyền

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Hệ thống sẽ dựa vào quyền của người dùng – được lưu trong cookie để xem người dùng là khách hàng hay quản trị viên. Từ đó sẽ kiểm tra các chức năng được cho phép của từng quyền

### **2.2.2: Quản lý người dùng**

\* Tìm kiếm người dùng

- Tác nhân: Quản trị viên, người dùng

- Mô tả:

+ Quản trị viên, người dùng viên nhập các thông tin tìm kiếm người dùng như: tên người dùng, email, tên tài khoản…

+ Hệ thống sẽ dựa vào thông tin quản trị viên hoặc người dùng nhập để lấy ra các người dùng hợp lệ.

\* Thêm người dùng

-Tác nhân: Quản trị viên

- Mô tả:

+ Quản trị viên nhập các thông tin của người dùng như: tên, tài khoản, email, mật khẩu …

+ Hệ thống sẽ dựa vào thông tin quản trị viên nhập để kiểm tra

+ Nếu thông tin hợp lệ thì tiến hành thêm thông tin người dùng xuống cơ sở dữ liệu và thông báo cho quản trị viên

+ Nếu thông tin không hợp lệ thì thông báo cho quản trị viên nhập lại

\* Sửa người dùng

- Tác nhân: Quản trị viên, người dùng

- Mô tả:

+ Người dùng, quản trị viên tiến hành sửa thông tin cá nhân của mình.

+ Hệ thống sẽ kiểm tra thông tin mà người dùng sửa và quyền của họ. Khách hàng chỉ được sửa thông tin cá nhân của mình. Quản trị viên có thể sửa thông tin của bất kì người dùng nào.

+ Nếu thông tin hợp lệ thì tiến hành cập nhật thông tin người dùng xuống cơ sở dữ liệu

+ Nếu thông tin không hợp lệ thì thông báo cho người dùng nhập lại

\* Xóa người dùng

-Tác nhân: Quản trị viên

- Mô tả:

+ Quản trị viên tiến hành chọn người dùng cần xoá

+ Hệ thống tiến hành xoá người dùng đó dưới cơ sở dữ liệu sau đó thông báo cho quản trị viên

### **2.2.3 Quản lý bài viết**

\* Tìm kiếm bài viết

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng nhập các thông tin bài viết cần tìm kiếm: tiêu đề, người đăng…

+ Hệ thống sẽ trả kết quả theo các yêu cầu tìm kiếm

\* Thêm bài viết

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng, quản trị viên nhập các thông tin bài viết như: tiêu đề, chủ đề, nội dung, thể loại…

+ Hệ thống sẽ kiểm tra thông tin người dùng nhập và quyền của họ

+ Nếu thông tin hợp lệ thì tiến hành thêm bài viết xuống cơ sở dữ liệu và thông báo cho người dùng.

+ Nếu thông tin không hợp lệ thì thông báo cho người dùng nhập lại

\* Sửa bài viết

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng, quản trị viên sửa thông tin bài viết của mình.

+ Nếu thông tin hợp lệ thì tiến hành cập nhật xuống cơ sở dữ liệu

+ Nếu không sẽ thông báo cho người dùng, quản trị viên

\* Xóa bài viết

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng chọn bài viết cần xóa. Quản trị viên có thể chọn bất kỳ bài viết nào để xóa

+ Hệ thống sẽ kiểm tra quyền người dùng. Người dùng chỉ được xóa thông tin bài viết của mình. Quản trị viên có thể xóa bất kỳ bài viết nào

+ Hệ thống sẽ kiểm tra quyền người dùng. Người dùng chỉ được xóa thông tin bài viết của mình. Quản trị viên có thể xóa bất kỳ bài viết nào

+ Nếu hợp lệ sẽ xóa bài viết khỏi cơ sở dữ liệu

### **2.2.4 Quản lý thể loại**

\* Lọc thể loại

- Tác nhân: người dùng, quản trị viên

- Mô tả :

+ Người dùng, quản trị viên chọn các thể loại để tìm kiếm các bài viết

+ Hệ thống sẽ dựa vào các thông tin thể loại người dùng, quản trị viên cần tìm kiếm để trả về kết quả tương ứng

\* Thêm thể loại

- Tác nhân: Quản trị viên

- Mô tả:

+ Quản trị viên nhập các thông tin thể loại cần thêm: tên thể loại

+ Hệ thống sẽ kiểm tra thông tin thể loại cần thêm

+ Nếu hợp lệ sẽ thêm vào cơ sở dữ liệu. Ngược lại sẽ thông báo lỗi tới quản trị viên

\* Sửa thể loại

- Tác nhân: Quản trị viên

- Mô tả:

+ Quản trị viên chọn thể loại cần sửa.

+ Hệ thống sẽ kiểm tra thông tin cần sửa và cập nhật xuống cơ sở dữ liệu

\* Xóa thể loại

- Tác nhân: Quản trị viên

- Mô tả:

+ Quản trị viên chọn thể loại cần xóa

+ Hệ thống tiến hành kiểm tra xem còn bài viết nào có chủ đề mà quản trị viên muốn xoá

+ Nếu còn bài viết nào thì tiến hành thông báo cho quản trị viên tiến hành chuyển chủ đề của các bài viết đó sang chủ đề khác, rồi mới tiến hành xoá chủ đề mà quản trị viên chọn

+ Nếu không còn bài viết nào thì tiến hành xoá chủ đề dưới cơ sở dữ liệu và thông báo cho quản trị viên

### **2.2.5 Quản lý bình luận**

\* Xóa bình luận

- Tác nhân: Quản trị viên, người dùng

- Mô tả:

+ Người dùng, quản trị viên chọn bình luận cần xóa. Quản trị viên có thể xóa bất cứ bình luận nào. Người dùng chỉ có thể xóa bình luận của mình

+ Hệ thống tiến hành xóa bình luận ở cơ sở dữ liệu

\* Thêm bình luận

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng, quản trị viên tiến hành nhập bình luận.

+ Hệ thống tiến hành thêm bình luận vào cơ sở dữ liệu

\* Sửa bình luận

- Tác nhân: Người dùng, quản trị viên

- Mô tả:

+ Người dùng, quản trị viên chọn bình luận cần sửa.

+ Hệ thống sẽ tiến hành cập nhật bình luận xuống cơ sở dữ liệu sau đó cập nhật lên giao diện người dùng

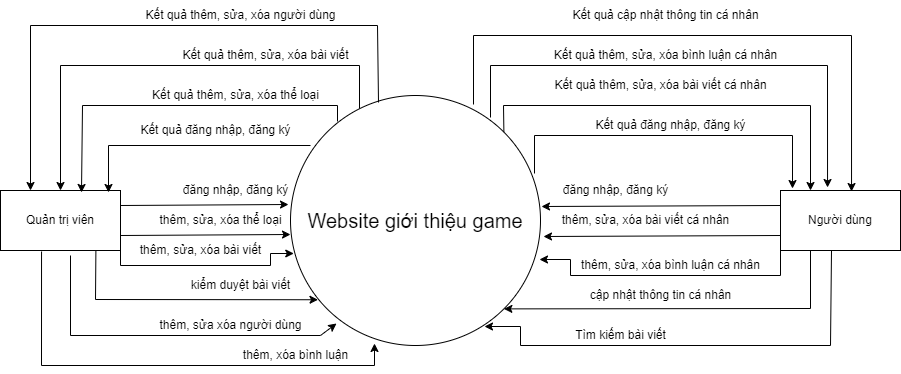
## **2.3: Sơ đồ luồng dữ liệu DFD**

### **2.3.1 Sơ đồ DFD mức ngữ cảnh**

\* Xác định các luồng dữ liệu

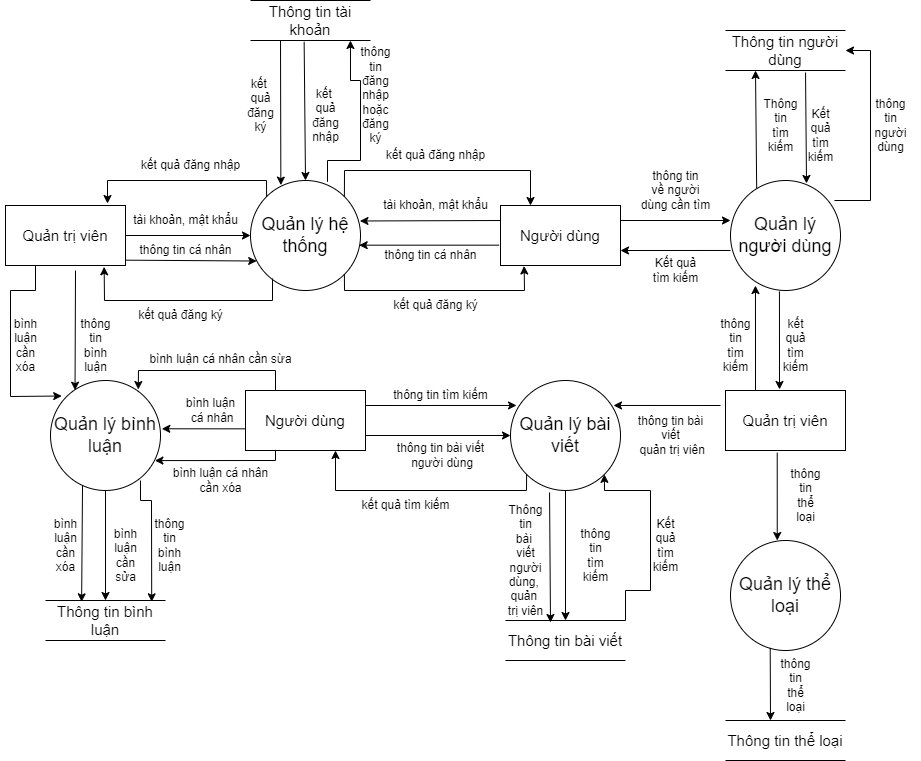
- Người dùng gửi các yêu cầu tới hệ thống ( tìm kiếm, đăng bài, … ) từ đó tùy yêu cầu sẽ được gửi tới quản trị viên, hệ thống

- Quản trị viên thực hiện các thao tác cập nhật thông tin, duyệt bài… trên hệ thống



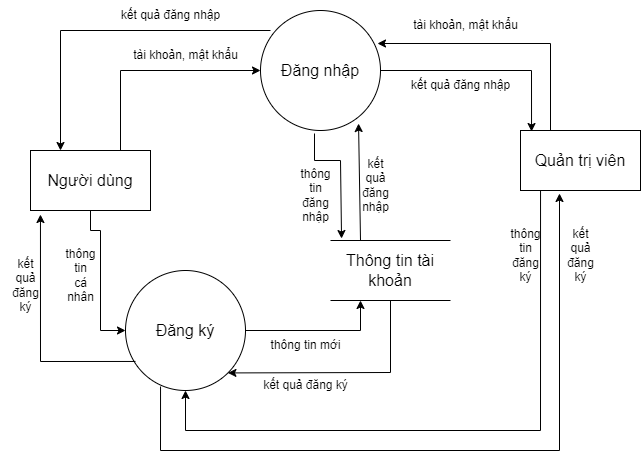
*Sơ đồ DFD mức ngữ cảnh*

### **2.3.2 Sơ đồ DFD mức 0**

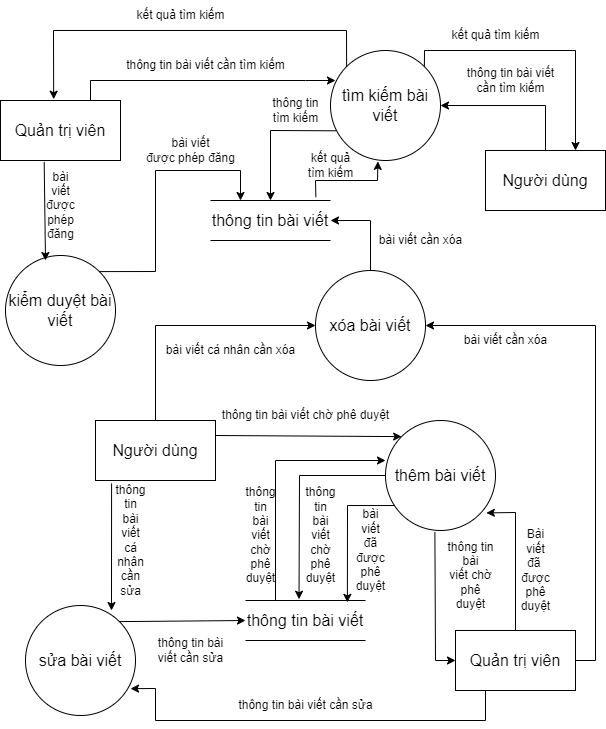


### **2.3.3 Sơ đồ DFD mức 1**

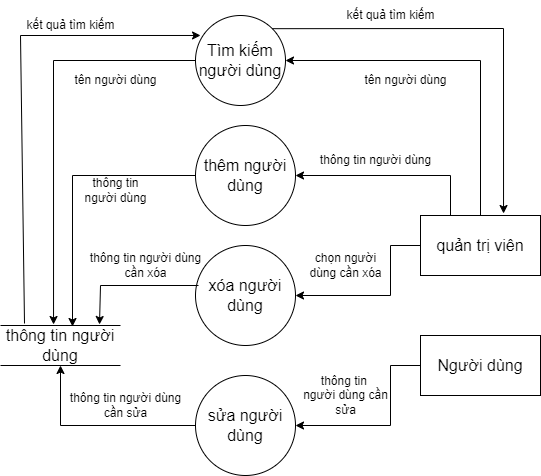
\* Sơ đồ DFD mức 1 quản lý hệ thống



\* Sơ đồ DFD mức 1 quản lý bài viết



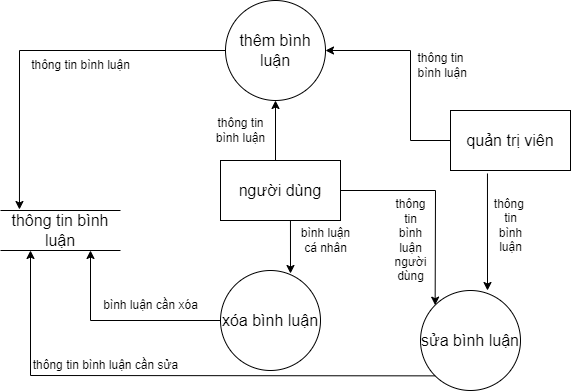
\* Sơ đồ DFD mức 1 quản lý người dùng



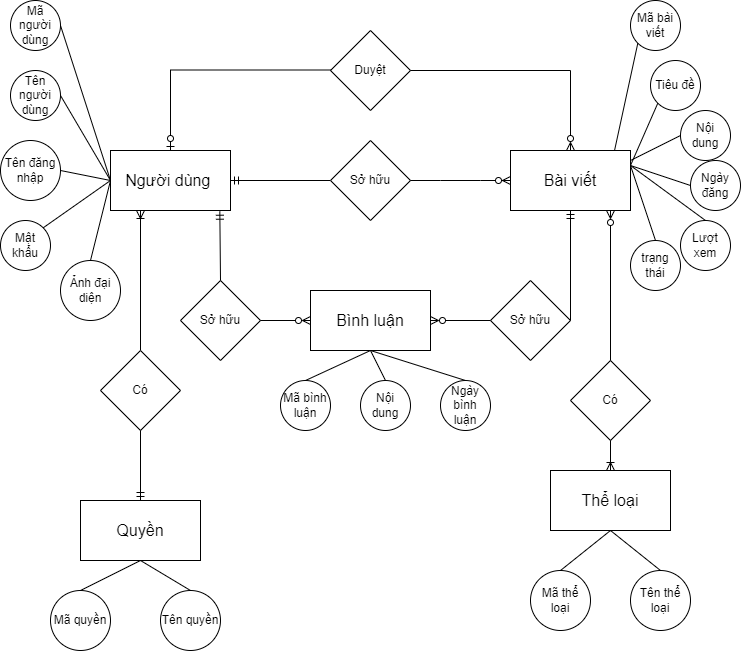
\* Sơ đồ DFD mức 1 quản lý thể loại



\* Sơ đồ DFD mức 1 quản lý bình luận



### **2.3.4: Sơ đồ ERD**



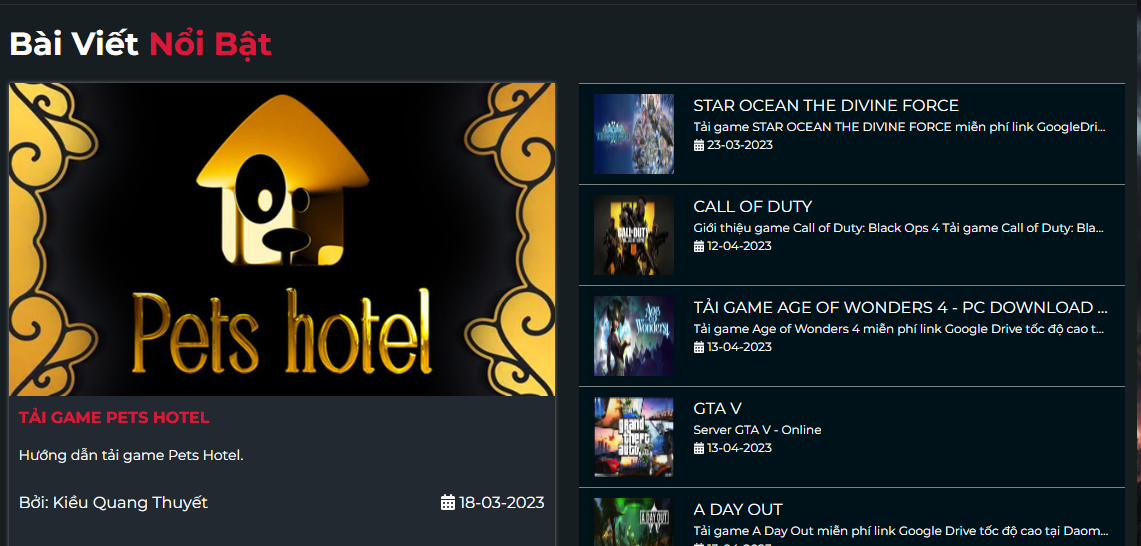
### **2.3.5: Cơ sở dữ liệu**

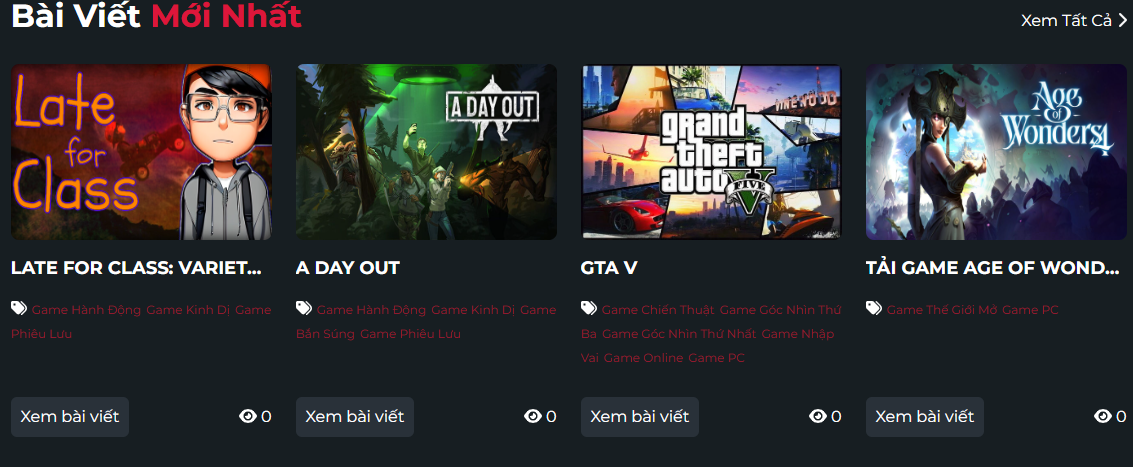


# **CHƯƠNG 4: XÂY DỰNG WEBSITE**

## **4.1: Chức năng hiển thị danh sách bài viết**

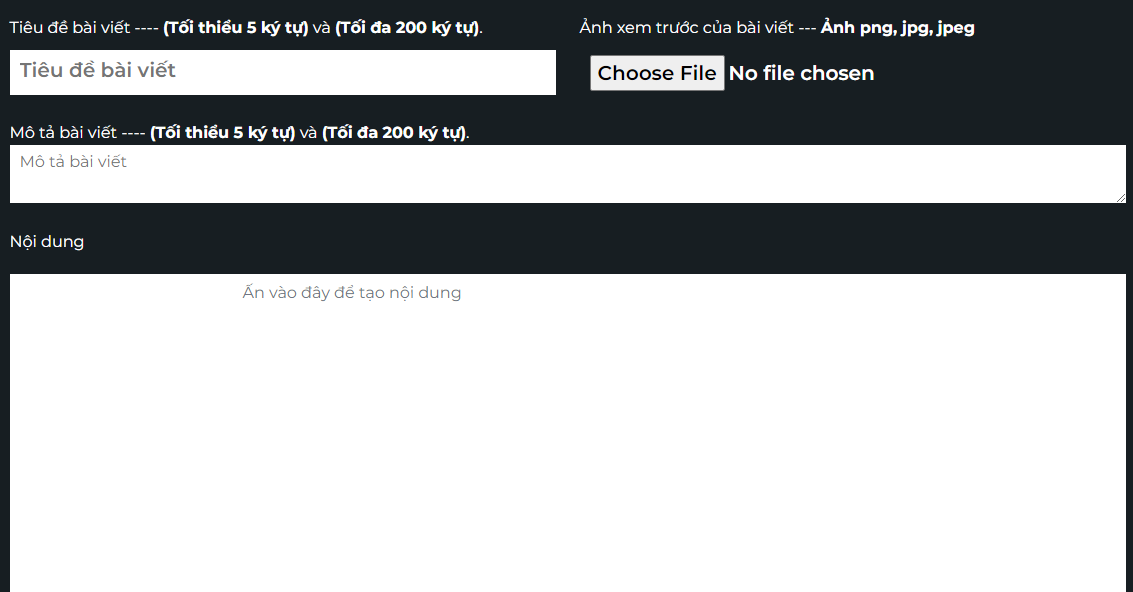
- Mô tả: Hiển thị danh sách bài viết ở trang chủ với các thông tin : Tác giả, tên bài viết, lượt đọc, ngày đăng.

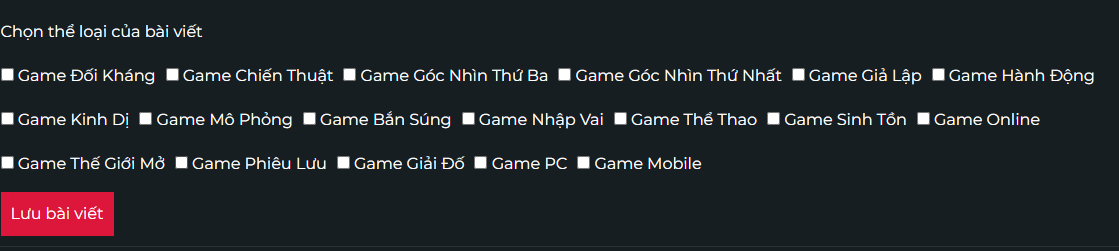




## **4.2: Chức năng đăng bài viết**

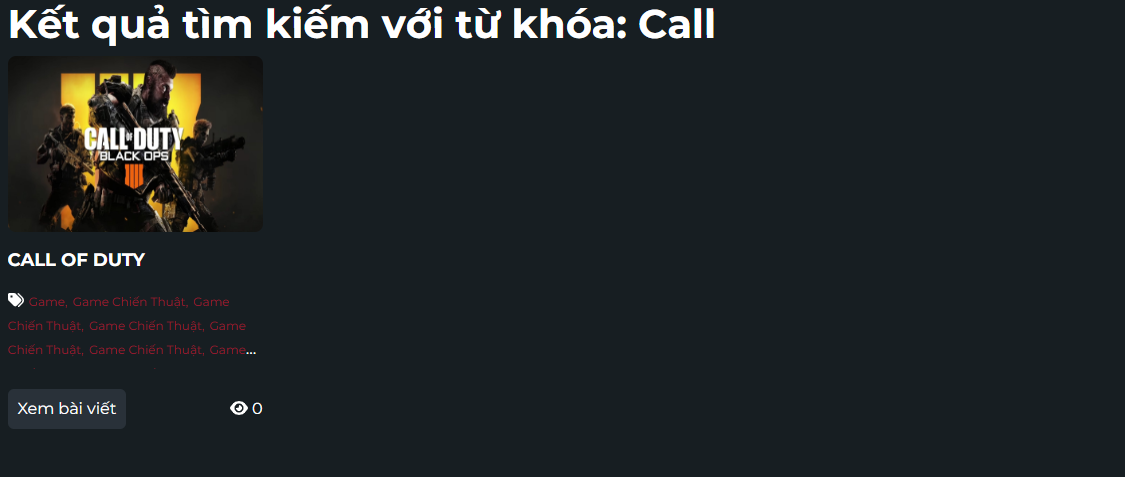
- Mô tả: Hiển thị giao diện soạn bài để người dùng có thể đăng các bài viết của mình.





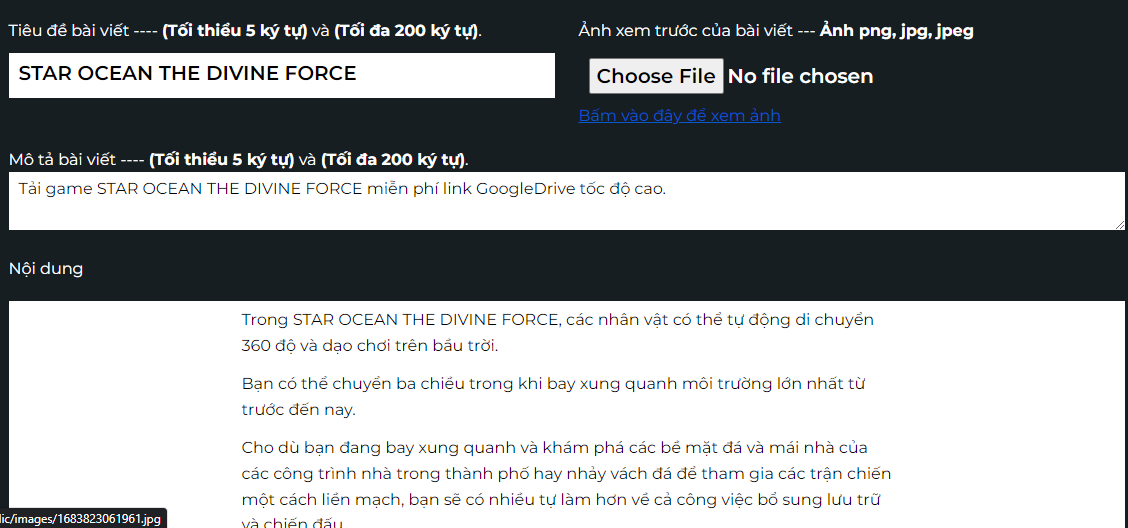
## **4.3: Chức năng tìm kiếm bài viết**

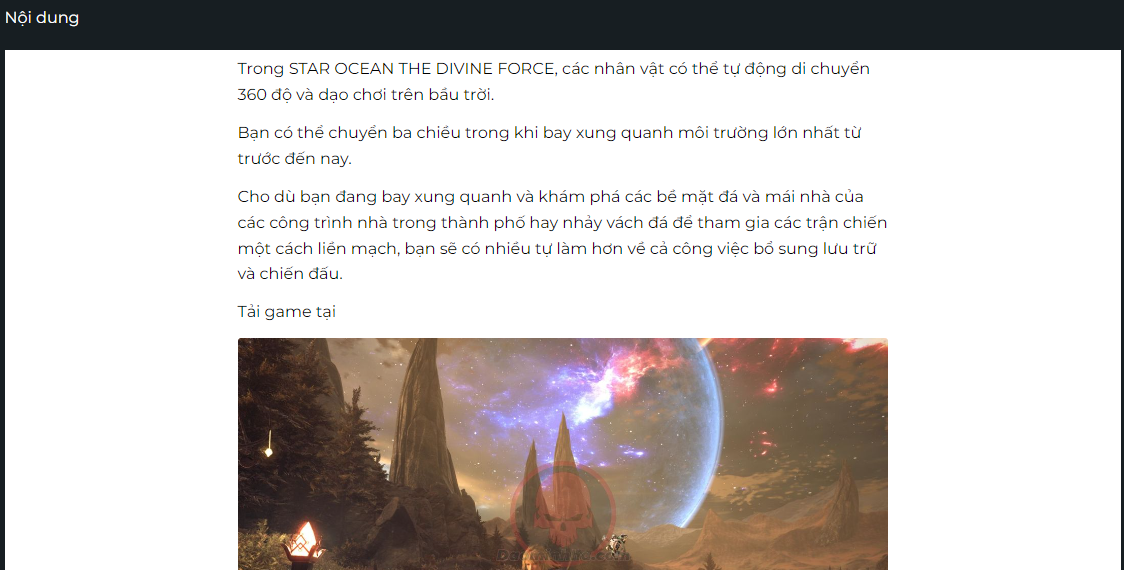
-Mô tả: người dùng có thể tìm kiếm bài viết theo tiêu đề



## **4.4: Chức năng sửa bài viết**

-Mô tả: người dùng có thể sửa lại bài viết của mình.





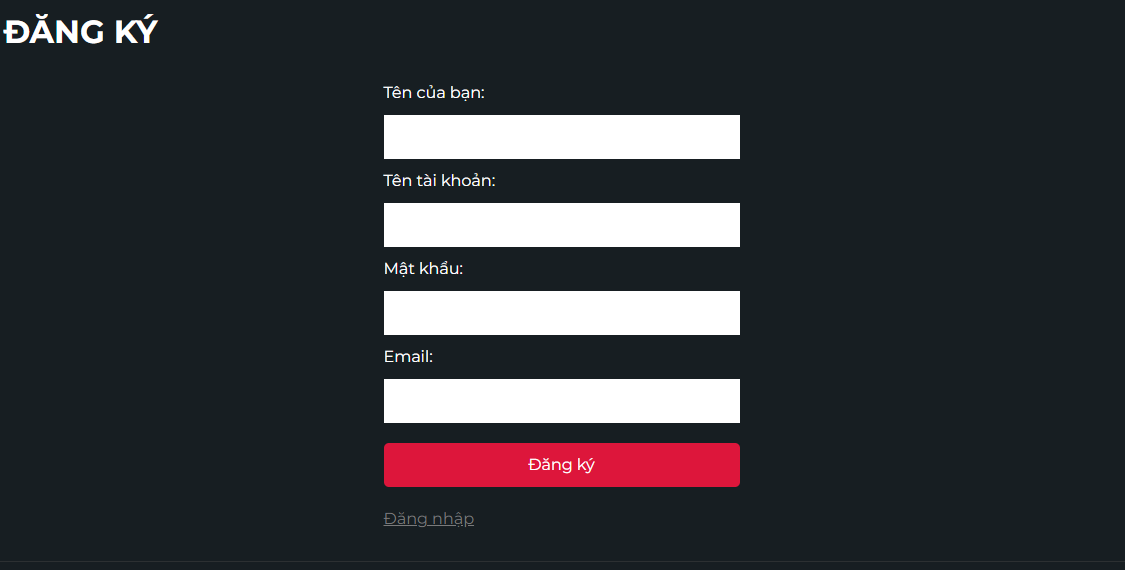
## **4.5: Chức năng xóa bài viết**

-Mô tả: người dùng có thể xóa bài viết của mình

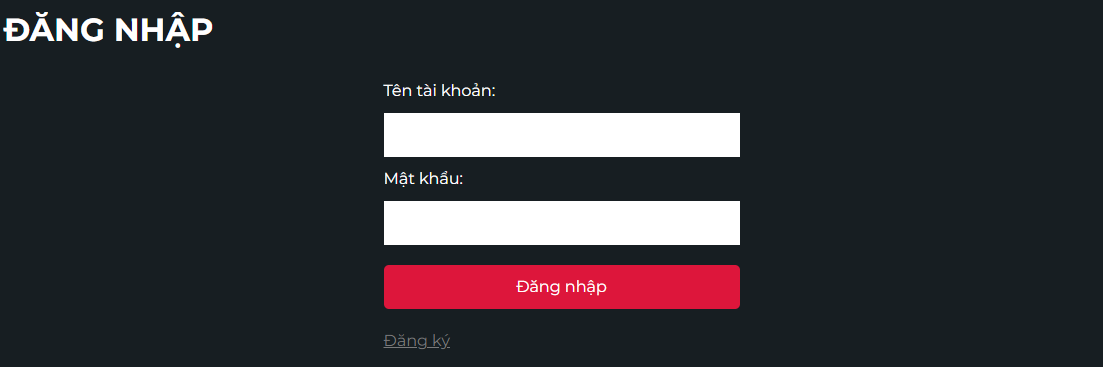


## **4.6: Chức năng đăng ký**

- Mô tả: người dùng có thể đăng ký tài khoản để sử dụng các tính năng của hệ thống.

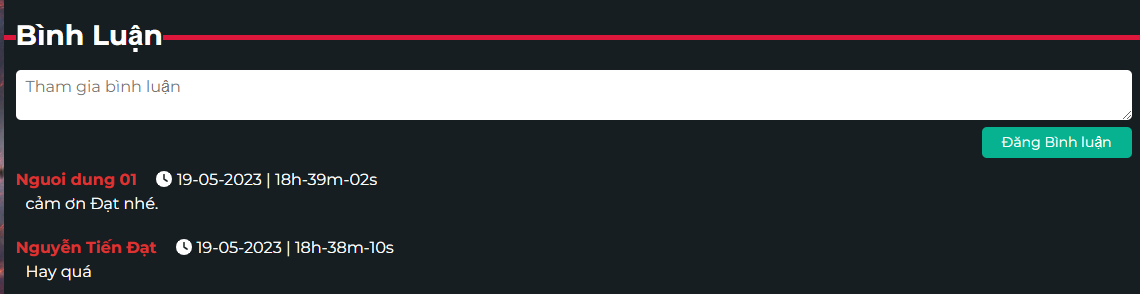


## **4.7: Chức năng đăng nhập**

- Mô tả: người dùng dùng tài khoản đã được đăng ký để đăng nhập vào hệ thống. 

## **4.8: Chức năng bình luận**

- Mô tả: người dùng có thể bình luận ở bất kỳ bài viết nào



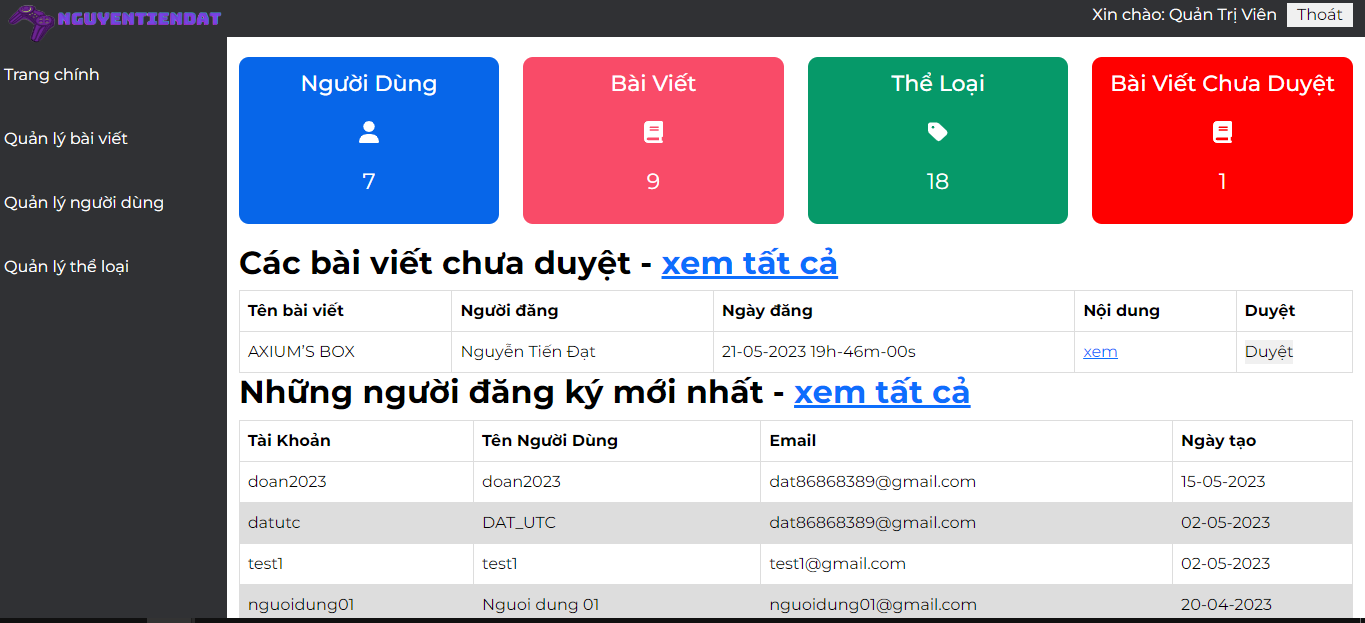
## **4.8: Đăng nhập trên trang quản trị**

- Mô tả: quản trị viên đăng nhập bằng tài khoản có quyền là quản trị để đăng nhập vào trang quản trị



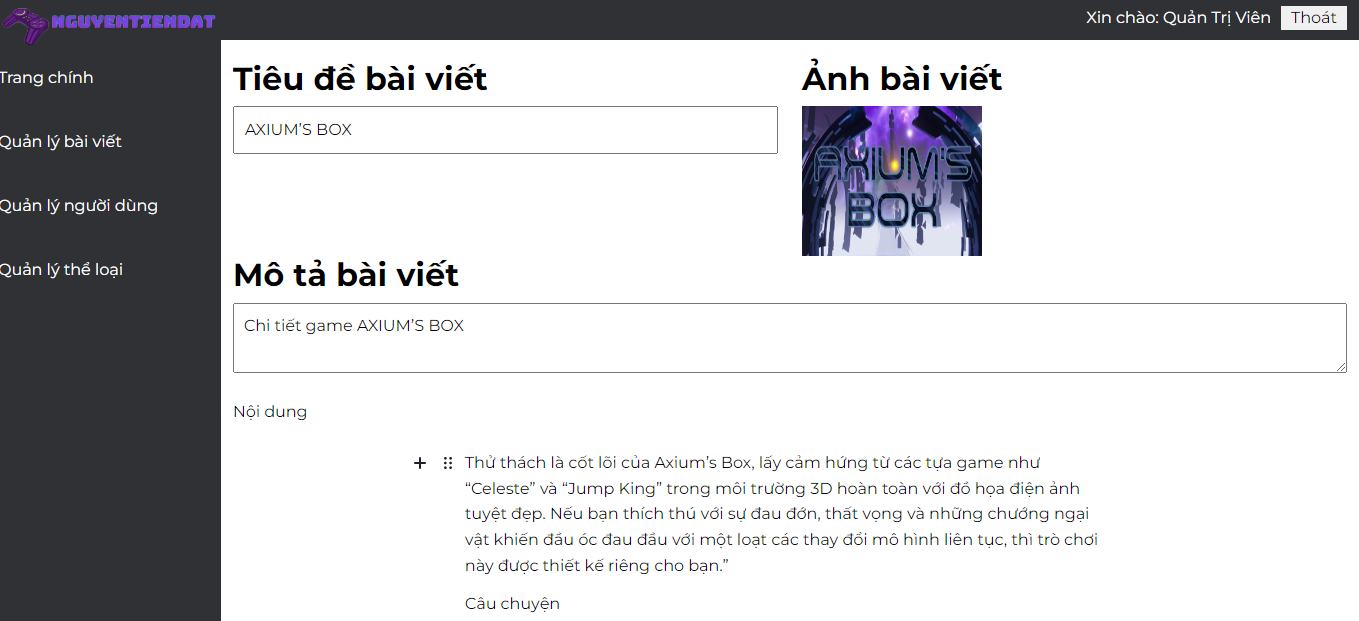
## **4.9: Trang quản trị**

- Mô tả: quản trị viên đăng nhập bằng tài khoản có quyền là quản trị để đăng nhập vào trang quản trị



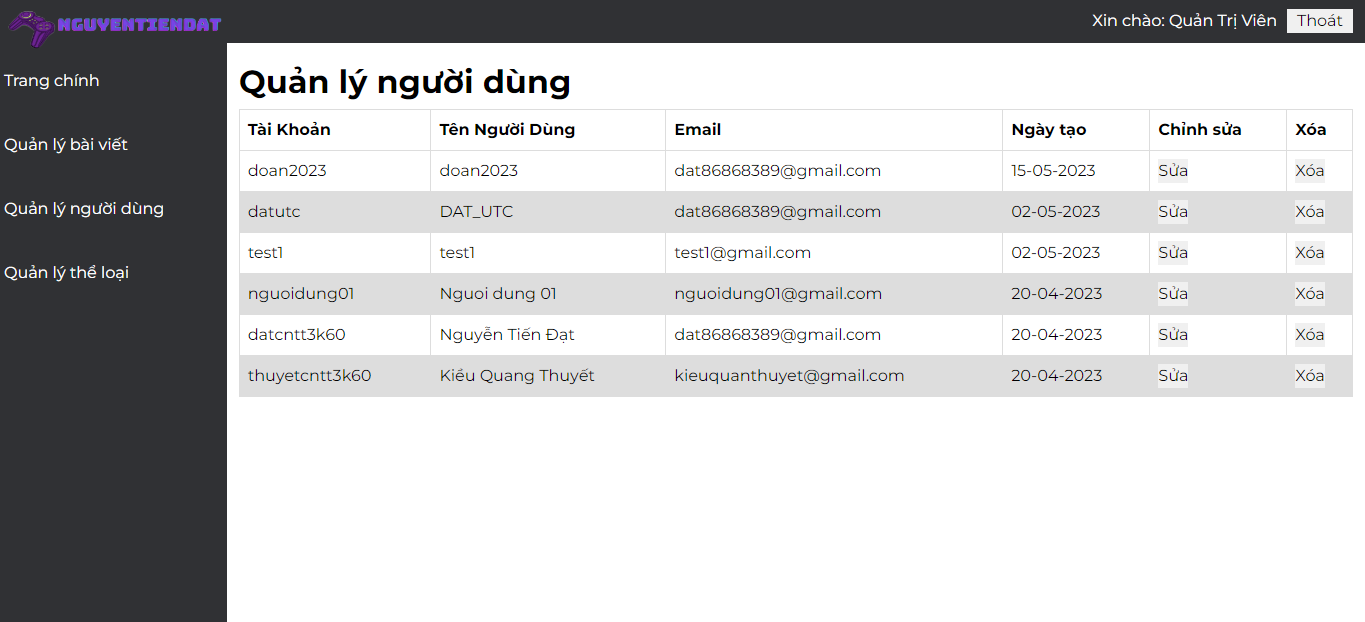
## **4.10: Xem trước bài viết trên trang quản trị**

- Mô tả: quản trị viên xem nội dung của bài định duyệt và ấn duyệt nếu thấy bài viết phù hợp. Ngược lại, sẽ không duyệt



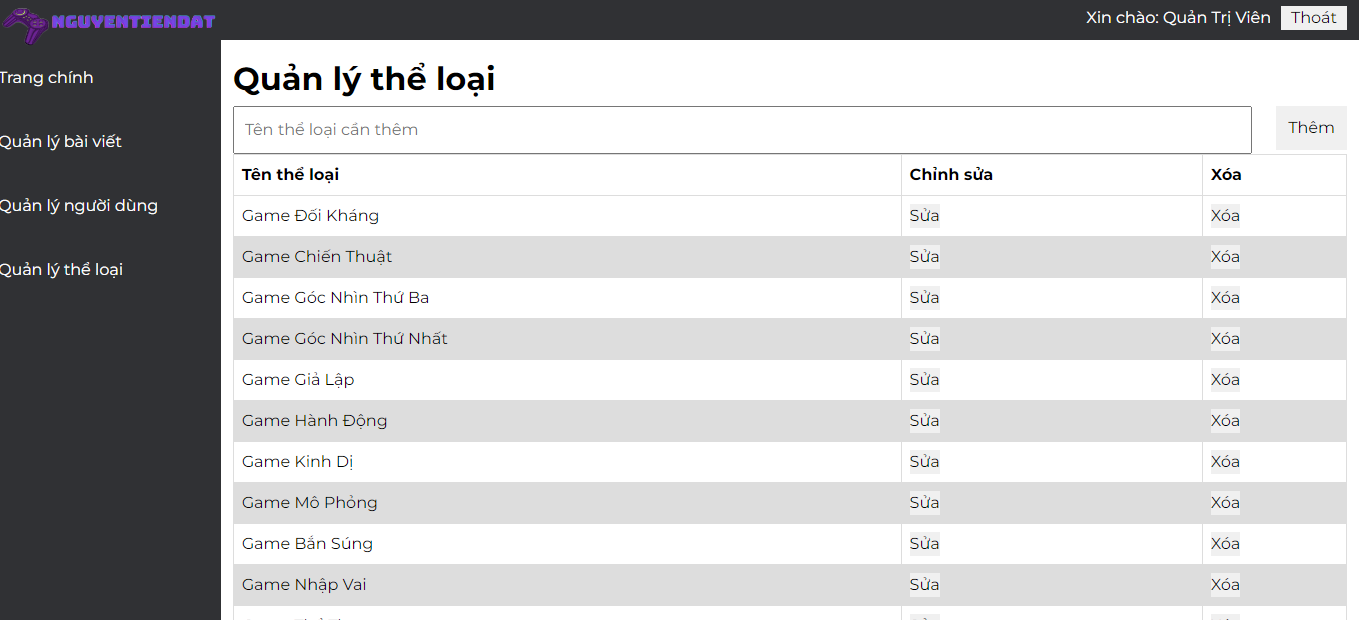
## **4.11: Quản lý người dùng**

- Mô tả: quản trị quản lý tất cả người dùng. Quản trị viên có thể sửa thông tin người dùng, xóa người dùng.



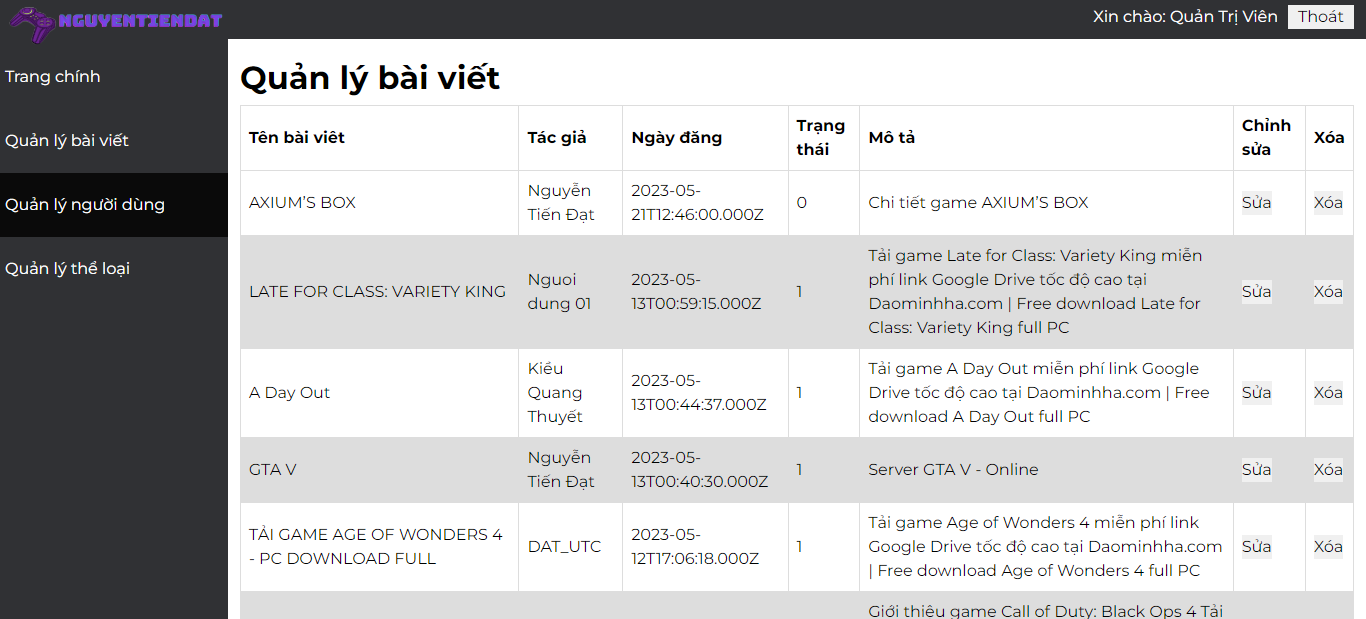
## **4.12: Quản lý thể loại**

- Mô tả: quản trị quản lý tất cả người dùng. Quản trị viên có thể thêm thể loại, chỉnh sửa thể loại và xóa thể loại



## **4.12: Quản lý bài viết**

- Mô tả: quản trị quản lý tất cả bài viết, có thể sửa hoặc xóa bài viết.



# **KẾT LUẬN**

Trong đồ án em tập trung phát triển các nghiệp vụ cơ bản của một website giới thiệu

1. **Kết quả đạt được:**

* Tìm hiểu về Bootstrap, JavaScript, HTML, CSS, JSX để xây dựng giao diện một website.
* Sử dụng ExpressJS ( framework NodeJS) để thiết kế Restful API phục vụ hiển thị và nhập liệu dữ liệu.
* Cách sử dụng cơ bản của công cụ soạn thảo EditorJS
* Sử dụng MySQL để quản trị cơ sở dữ liệu.
* Sử dụng Postman để kiểm thử các API
* Tìm hiểu và ứng dụng nghiệp vụ của một website giới thiệu
* Sử dụng Github để bảo lưu code.

1. **Hạn chế:**

Trong thời gian qua em đã cố gắng hết sức để tìm hiểu và thực hiện đề tài. Tuy nhiên với kinh nghiệm và thời gian có hạn nên không thể tránh khỏi nhưng thiếu sót trong đồ án. Cụ thể:

* Mẫu báo cáo còn hạn chế, chưa được đa dạng.
* Còn thiếu một vài module nâng cao trong nghiệp vụ
* Chưa áp dụng được tính năng tự động trong một vài module.

1. **Hướng phát triển đề tài:**

Với mong muốn hệ thống website tối ưu hơn, trong tương lai em sẽ phát triển thêm một số tính năng mới cho hệ thống. Nghiên cứu một số module khó hơn để có thể nâng cao khả năng code của bản thân.

TÀI LIỆU THAM KHẢO

[1] Tài liệu về ReactJS “https://react.dev/*”*

[2] Tài liệu về NextJS *“https://nextjs.org/”*

[3] Tài liệu về ExpressJS *“https://expressjs.com/”*

[4] Tài liệu MySQL *“https://www.w3schools.com/MySQL/default.asp”*

[5] Tài liệu EditorJS *“https://editorjs.io/”*