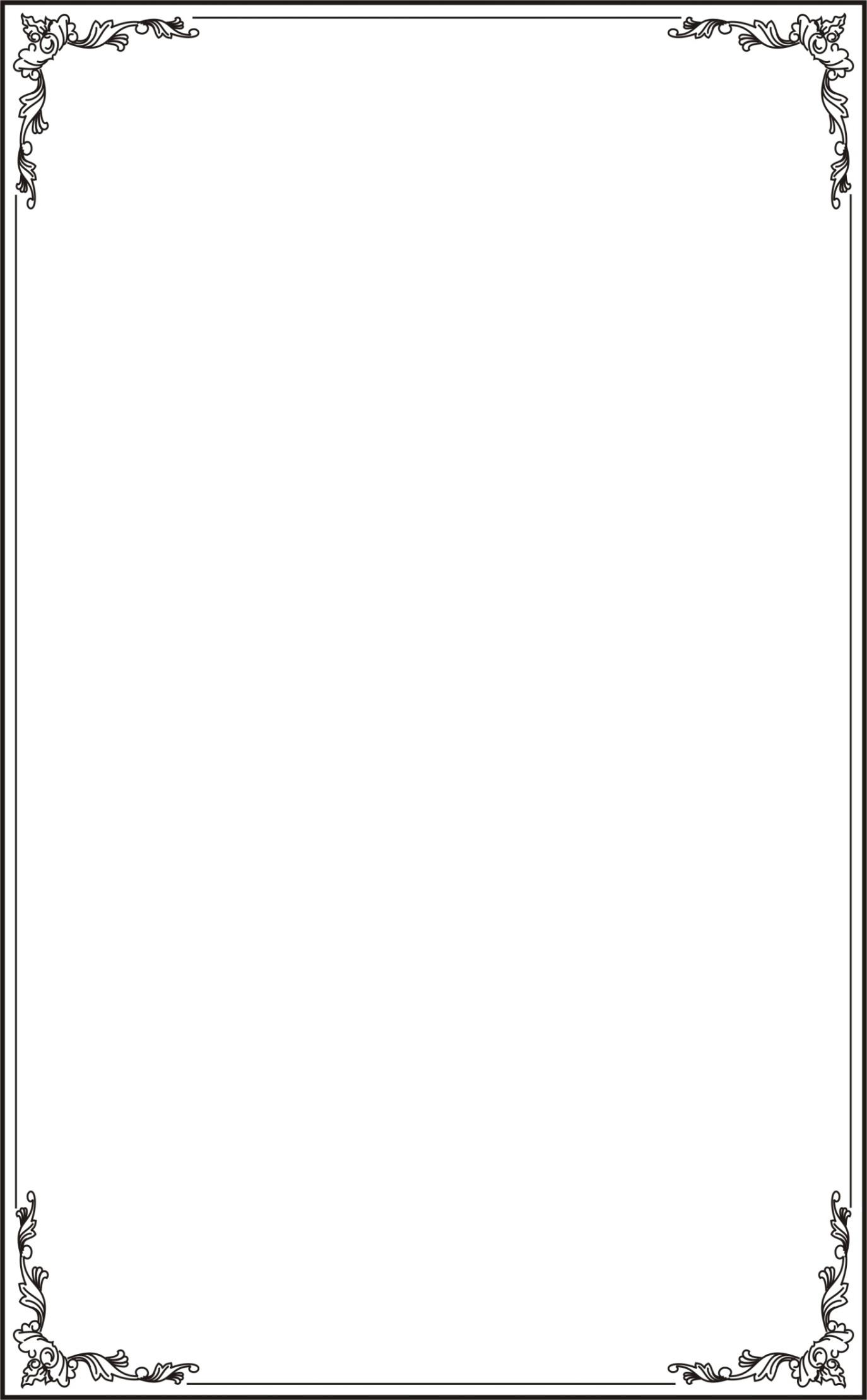
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

**TÌM HIỂU VỀ RSS FEED**

**VÀ ÁP DỤNG XÂY DỰNG WEBSITE TIN MỚI**

**SỬ DỤNG NODEJS VÀ JAVASCRIPT**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:** | TS. Nguyễn Thị Hoa Huệ |
| **Sinh viên:** | Đinh Khắc Tuấn |
| **Mã sinh viên:** | 2020605847 |
| **Lớp:** | 2020DHKTPM4 **Khóa:**15 |

**Hà Nội – Năm 2024**

# LỜI CẢM ƠN

Để bài tập lớn môn học được thực hiện thành công, em xin cảm ơn giảng viên hướng dẫn của cô **Nguyễn Thị Hoa Huệ**, Khoa Công nghệ thông tin, Trường đại học Công nghiệp Hà Nội đã chia sẻ những kiến thức quý báu, hướng dẫn tận tình và có những góp ý trong suốt quá trình thực hiện đề tài để em có thể hoàn thành đồ án một cách tốt nhất.

Sau khi nghiên cứu đề tài và kết thúc khóa học, em đã học hỏi và tích lũy được kiến thức và kinh nghiệm từ thầy đi trước để hoàn thiện và phát triển bản thân. Bên cạnh đó, đây cũng là cơ hội giúp em nhận ra mình cần hoàn thiện ssthêm những gì để chuẩn bị cho một hành trình dài phía trước.

Đặc biệt, em xin chân thành cảm ơn cô Nguyễn Thị Hoa Huệ .Người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua, cuốn báo cáo này không hẳn tránh khỏi những thiếu sót. Em rất mong muốn được những ý kiến đóng góp quý báu của thầy cô để bản báo cáo hoàn thiện hơn cũng như cải thiện kỹ năng của mình.

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc178448816)

[MỤC LỤC 2](#_Toc178448817)

[Danh mục hình ảnh 5](#_Toc178448818)

[Danh mục các bảng 7](#_Toc178448819)

[MỞ ĐẦU 8](#_Toc178448820)

[1. Lý do chọn đề tài 8](#_Toc178448821)

[2. Mục tiêu đề tài 9](#_Toc178448822)

[3. Nội dung nghiên cứu 9](#_Toc178448823)

[4. Phương pháp nghiên cứu 10](#_Toc178448824)

[5. Kết quả dự kiến 11](#_Toc178448825)

[6. Bố cục đề tài 11](#_Toc178448826)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 13](#_Toc178448827)

[1.1. Giới thiệu về RSS 13](#_Toc178448828)

[1.1.1. Khái niệm 13](#_Toc178448829)

[1.1.2. Cách sử dụng 13](#_Toc178448830)

[1.1.3. Lịch sử 14](#_Toc178448831)

[1.2. Khảo sát sơ bộ 16](#_Toc178448832)

[1.2.1. Mục đích 16](#_Toc178448833)

[1.2.2. Đối tượng khảo sát 17](#_Toc178448834)

[1.2.3. Phương pháp 17](#_Toc178448835)

[1.2.4. Kết quả sơ bộ 17](#_Toc178448836)

[1.3. Khảo sát hiện trạng cụ thể 18](#_Toc178448837)

[1.3.1. VnExpress 18](#_Toc178448838)

[1.3.2. Thanh Niên 18](#_Toc178448839)

[1.3.3. Tuổi Trẻ 19](#_Toc178448840)

[1.3.4. Người Đưa Tin 19](#_Toc178448841)

[1.3.5. Đánh giá tổng quan 20](#_Toc178448842)

[1.3.6. Cơ hội cải tiến 20](#_Toc178448843)

[1.3.7. Báo mới 20](#_Toc178448844)

[1.4. Các liên kết rss được sử dụng 21](#_Toc178448845)

[1.5. Công nghệ sử dụng để thực hiện đề tài 22](#_Toc178448846)

[1.5.1. Nextjs 22](#_Toc178448847)

[1.5.2. Tailwind css 23](#_Toc178448848)

[1.5.3. CI/CD 24](#_Toc178448849)

[1.5.4. Ant Design 25](#_Toc178448850)

[1.5.5. Docker 26](#_Toc178448851)

[1.5.6. SpringBoot 26](#_Toc178448852)

[1.6. Tài liệu api 27](#_Toc178448853)

[1.6.1. Giới thiệu 27](#_Toc178448854)

[1.6.2. Thông tin chung 28](#_Toc178448855)

[1.6.3. Thông tin các điểm cuối 28](#_Toc178448856)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 30](#_Toc178448857)

[2.1. Khảo sát bài toán 30](#_Toc178448858)

[2.1.1. Giới thiệu về trang web tin mới 30](#_Toc178448859)

[2.1.2. Mô tả bài toán 30](#_Toc178448860)

[2.2. Xác định các yêu cầu của bài toán 30](#_Toc178448861)

[2.2.1. Các yêu cầu chức năng 30](#_Toc178448862)

[2.2.2. Các yêu cầu phi chức năng 31](#_Toc178448863)

[2.3. Phân tích và thiết kế 32](#_Toc178448864)

[2.3.1. Mô hình hóa chức năng 32](#_Toc178448865)

[2.3.2. Mô tả các use case 37](#_Toc178448866)

[2.3.3. Biểu đồ thực hiện hóa các usecase 44](#_Toc178448867)

[2.3.4. Mô hình hóa dữ liệu 52](#_Toc178448868)

[2.4. Thiết kế giao diện 56](#_Toc178448869)

[2.4.1. Giao diện người dùng 56](#_Toc178448870)

[CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ 61](#_Toc178448871)

[3.1. Cài đặt chương trình 61](#_Toc178448872)

[3.1.1. Môi trường cài đặt 61](#_Toc178448873)

[3.1.2. Hướng dẫn cài đặt chương trình 62](#_Toc178448874)

[3.2. Kết quả đạt được 63](#_Toc178448875)

[3.2.1. Kết quả đạt được phía người dùng 63](#_Toc178448876)

[3.2.2. Kết quả đạt được phía quản trị viên 65](#_Toc178448877)

[3.3. Kiểm thử hệ thống 68](#_Toc178448878)

[3.3.1. Kế hoạch kiểm thử 68](#_Toc178448879)

[3.3.2. Kịch bản kiểm thử 68](#_Toc178448880)

[3.3.3. Kết quả kiểm thử 72](#_Toc178448881)

[CHƯƠNG 4: TRIỂN KHAI VÀ VẬN HÀNH 74](#_Toc178448882)

[4.1. Chuẩn bị môi trường triển khai 74](#_Toc178448883)

[4.2. Quy trình triển khai lên server 74](#_Toc178448884)

[KẾT LUẬN 78](#_Toc178448885)

[TÀI LIỆU THAM KHẢO 80](#_Toc178448886)

# Danh mục hình ảnh

[Hình 1.1: Điểm cuối advertisement-controller 28](#_Toc178375672)

[Hình 1.2: Điểm cuối advertising-field-controller 28](#_Toc178375673)

[Hình 1.3: Điểm cuối approval-request-controller 28](#_Toc178375674)

[Hình 1.4: Điểm cuối authentication-controller 29](#_Toc178375675)

[Hình 1.5: Điểm cuối role-controller 29](#_Toc178375676)

[Hình 1.6: Điểm cuối user-controller 29](#_Toc178375677)

[Hình 2.1: Biểu đồ use case toàn hệ thống 32](#_Toc178375678)

[Hình 2.2:Biểu đồ use case người đọc tin 33](#_Toc178375679)

[Hình 2.3:Biểu đồ use case nhà quảng cáo 34](#_Toc178375680)

[Hình 2.4:Biểu đồ use case admin 35](#_Toc178375681)

[Hình 2.5:Biểu đồ use case RSS Feed 36](#_Toc178375682)

[Hình 2.6:Biểu đồ lớp use case xem danh sách tin tức 44](#_Toc178375683)

[Hình 2.7:Biểu đồ trình tự use case xem danh sách tin tức 44](#_Toc178375684)

[Hình 2.8:Biểu đồ lớp use case tìm kiếm tin tức 45](#_Toc178375685)

[Hình 2.9:Biểu đồ trình tự use case tìm kiếm tin tức 45](#_Toc178375686)

[Hình 2.10:Biểu đồ lớp use case quản lý tài khoản 46](#_Toc178375687)

[Hình 2.11:Biểu đồ trình tự use case quản lý tài khoản 46](#_Toc178375688)

[Hình 2.12:Biểu đồ lớp use case quản lý quảng cáo 47](#_Toc178375689)

[Hình 2.13:Biểu đồ trình tự use case quản lý quảng cáo 47](#_Toc178375690)

[Hình 2.14:Biểu đồ lớp use case xem thống kê 48](#_Toc178375691)

[Hình 2.15:Biểu đồ trình tự use case xem thống kê 48](#_Toc178375692)

[Hình 2.16:Biểu đồ lớp use case gửi phê duyệt quảng cáo 49](#_Toc178375693)

[Hình 2.17:Biểu đồ trình tự use case gửi phê duyệt quảng cáo 49](#_Toc178375694)

[Hình 2.18:Biểu đồ lớp use case phê duyệt quảng cáo 50](#_Toc178375695)

[Hình 2.19:Biểu đồ trình tự use case phê duyệt quảng cáo 50](#_Toc178375696)

[Hình 2.20:Biểu đồ lớp use case thanh toán 51](#_Toc178375697)

[Hình 2.21:Biểu đồ trình tự use case thanh toán 51](#_Toc178375698)

[Hình 2.22:Biểu đồ thực thể liên hết 52](#_Toc178375699)

[Hình 2.23:Hình dung giao diện Trang chủ 56](#_Toc178375700)

[Hình 2.24:Hình dung giao diện danh mục tin tức 57](#_Toc178375701)

[Hình 2.25:Hình dung giao diện Tìn kiếm tin tức 58](#_Toc178375702)

[Hình 2.26:Hình dung giao diện trang tổng quan 59](#_Toc178375703)

[Hình 2.27:Hình dung giao diện quản lý tài khoản 59](#_Toc178375704)

[Hình 2.28:Hình dung giao diện quản lý quảng cáo 60](#_Toc178375705)

[Hình 2.29:Hình dung giao diện quản lý vai trò 60](#_Toc178375706)

[Hình 2.30:Hình dung giao diện hồ sơ cá nhân 61](#_Toc178375707)

[Hình 3.1:Giao diện trang chủ 64](#_Toc178375708)

[Hình 3.2:Giao diện danh mục tin tức 65](#_Toc178375709)

[Hình 3.3:Giao diện trang tim kiếm tin tức 65](#_Toc178375710)

[Hình 3.4:Giao diện tùy chọn chủ đề 66](#_Toc178375711)

[Hình 3.5:Giao diện trang tổng quan admin 66](#_Toc178375712)

[Hình 3.6:Giao diện trang quản lý tài khoản 67](#_Toc178375713)

[Hình 3.7:Giao diện trang quản lý lĩnh vực quảng cáo 67](#_Toc178375714)

[Hình 3.8:Giao diện trang quản lý quảng cáo 67](#_Toc178375715)

[Hình 3.9:Giao diện trang phê duyệt quảng cáo 68](#_Toc178375716)

[Hình 3.10:Giao diện trang quản lý vai trò 68](#_Toc178375717)

[Hình 3.11:Giao diện quản lý hồ sơ cá nhân 68](#_Toc178375718)

[Hình 3.12: Biểu đồ kết quả kiểm thử 74](#_Toc178375719)

# Danh mục các bảng

[Bảng 1: Liên kết RSS cho Trang Chủ 22](#_Toc178325605)

[Bảng 2: Liên kết RSS cho Các Chuyên Mục 22](#_Toc178325606)

[Bảng 3: Các tác nhân 33](#_Toc178325607)

[Bảng 4:Mô tả chi tiết use case xem danh sách tin tức 38](#_Toc178325608)

[Bảng 5:Mô tả chi tiết use case xem chi tiết tin tức 38](#_Toc178325609)

[Bảng 6:Mô tả chi tiết use case tùy chọn chủ đề 39](#_Toc178325610)

[Bảng 7:Mô tả chi tiết use case tùy chọn giao diện 39](#_Toc178325611)

[Bảng 8:Mô tả chi tiết use case xem trang chủ 40](#_Toc178325612)

[Bảng 9:Mô tả chi tiết use case quản lý tài khoản 40](#_Toc178325613)

[Bảng 10:Mô tả chi tiết use case quản lý vai trò 40](#_Toc178325614)

[Bảng 11:Mô tả chi tiết use case quản lý quảng cáo 41](#_Toc178325615)

[Bảng 12:Mô tả chi tiết use case quản lý hồ sơ cá nhân 41](#_Toc178325616)

[Bảng 13:Mô tả chi tiết use case xem thống kê 42](#_Toc178325617)

[Bảng 14:Mô tả chi tiết use case gửi phê duyệt quảng cáo 42](#_Toc178325618)

[Bảng 15:Mô tả chi tiết use case phê duyệt quảng cáo 43](#_Toc178325619)

[Bảng 16:Mô tả chi tiết use case quản lý doanh thu 43](#_Toc178325620)

[Bảng 17:Mô tả chi tiết use case thanh toán 44](#_Toc178325621)

[Bảng 18:Bảng users 54](#_Toc178325622)

[Bảng 19:Bảng role 54](#_Toc178325623)

[Bảng 20:Bảng advertisement 55](#_Toc178325624)

[Bảng 21:Bảng advertising\_field 56](#_Toc178325625)

[Bảng 22:Bảng revenue 56](#_Toc178325626)

[Bảng 23:Bảng approval\_request 56](#_Toc178325627)

[Bảng 24:Kịch bản kiểm thử 70](#_Toc178325628)

# MỞ ĐẦU

## 1. Lý do chọn đề tài

Hiện nay trên thế giới, công nghệ thông tin và truyền thông đang phát triển với tốc độ chóng mặt. Trong bối cảnh đó, việc cập nhật và chia sẻ thông tin nhanh chóng, hiệu quả trở thành một nhu cầu cấp thiết. RSS Feed, một công nghệ được phát triển để đáp ứng nhu cầu này, đang ngày càng được sử dụng rộng rãi trên các nền tảng trực tuyến. So với các phương thức truyền thống, RSS Feed mang lại nhiều ưu điểm vượt trội như khả năng cập nhật thông tin tự động, tiết kiệm thời gian cho người dùng, và tăng cường khả năng tiếp cận thông tin.

Hơn thế nữa, với lợi thế của Internet, RSS Feed cho phép người dùng dễ dàng theo dõi nhiều nguồn thông tin cùng lúc mà không cần phải truy cập trực tiếp vào từng trang web. Kết hợp với các ứng dụng đọc tin và các plugin tích hợp trên trình duyệt, RSS Feed càng trở nên thuận tiện và phổ biến trong việc tiêu thụ thông tin hàng ngày.

Do đó, việc áp dụng RSS Feed vào xây dựng website đang trở thành một xu hướng quan trọng trong phát triển web. Các trang tin tức, blog cá nhân, và cổng thông tin doanh nghiệp đều có thể hưởng lợi từ việc tích hợp RSS Feed, giúp tăng cường khả năng phân phối nội dung và tương tác với độc giả.

Trước thực tế đó, em lựa chọn đề tài: "Tìm hiểu về RSS Feed và áp dụng xây dựng website tin mới sử dụng nodejs và javascript " làm đề tài nghiên cứu của mình. Đề tài này không chỉ cho phép em đi sâu vào một công nghệ web quan trọng mà còn tạo cơ hội để áp dụng kiến thức đã học vào thực tế phát triển website. Việc nghiên cứu và triển khai RSS Feed sẽ giúp em hiểu sâu hơn về cách thức hoạt động của web hiện đại, đồng thời nâng cao kỹ năng lập trình web, tạo nền tảng vững chắc cho sự phát triển nghề nghiệp trong tương lai trong lĩnh vực công nghệ thông tin.

## 2. Mục tiêu đề tài

1. Nghiên cứu và tìm hiểu sâu về công nghệ RSS Feed, bao gồm cấu trúc, nguyên lý hoạt động và các ứng dụng phổ biến.
2. Phân tích và so sánh RSS Feed với các phương thức cập nhật thông tin khác, đánh giá ưu nhược điểm.
3. Thiết kế và xây dựng một website mẫu tích hợp RSS Feed, bao gồm:

* Giao diện người dùng thân thiện, dễ sử dụng
* Chức năng đọc và hiển thị nội dung từ các nguồn RSS Feed
* Công cụ quản lý và tổ chức các nguồn cấp dữ liệu
* Hệ thống tạo và xuất RSS Feed cho nội dung của website

1. Phát triển các tính năng quản trị cho website:

* Quản lý người dùng và phân quyền
* Quản lý nội dung và nguồn cấp dữ liệu
* Theo dõi và phân tích dữ liệu sử dụng

## 3. Nội dung nghiên cứu

1. Nghiên cứu, khảo sát thực tế các ứng dụng và website đang sử dụng RSS Feed hiệu quả trên thị trường hiện nay.
2. Nghiên cứu và tìm hiểu các kiến thức về:

* Công nghệ RSS Feed: cấu trúc, định dạng, và cách thức hoạt động
* Các framework và công cụ hỗ trợ xử lý RSS
* Hệ quản trị cơ sở dữ liệu để lưu trữ dữ liệu RSS
* Mô hình MVC (Model-View-Controller) để tổ chức cấu trúc của ứng dụng web

1. Khảo sát, phân tích và thiết kế hệ thống dựa trên các yêu cầu đề ra:

* Xác định các chức năng cần thiết cho website ứng dụng RSS Feed
* Thiết kế cơ sở dữ liệu phù hợp để lưu trữ thông tin về nguồn cấp dữ liệu và nội dung RSS
* Xây dựng các mô hình quan hệ và luồng dữ liệu trong hệ thống

1. Tìm hiểu và tích hợp các nguồn RSS Feed từ các trang tin tức, blog, và các nguồn thông tin khác vào hệ thống:

* Nghiên cứu cách thức lấy và phân tích dữ liệu từ các nguồn RSS Feed khác nhau
* Phát triển cơ chế cập nhật tự động để đồng bộ hóa nội dung mới từ các nguồn

1. Tìm hiểu và phát triển chức năng tạo RSS Feed cho nội dung của website:

* Nghiên cứu cách tạo và xuất RSS Feed chuẩn
* Phát triển tính năng cho phép người dùng tạo và quản lý RSS Feed của riêng họ

1. Rèn luyện và nâng cao các kiến thức về kiểm thử:

* Áp dụng các phương pháp kiểm thử đơn vị và kiểm thử tích hợp cho các chức năng liên quan đến RSS Feed
* Thực hiện kiểm thử hiệu năng để đảm bảo website có thể xử lý nhiều nguồn RSS Feed cùng lúc

1. Nghiên cứu và áp dụng các biện pháp bảo mật cho hệ thống:

* Tìm hiểu các vấn đề bảo mật liên quan đến việc sử dụng RSS Feed
* Áp dụng các biện pháp bảo vệ hệ thống khỏi các cuộc tấn công có thể xảy ra thông qua RSS Feed

## 4. Phương pháp nghiên cứu

1. Phương pháp nghiên cứu về mặt lý thuyết
2. Tiến hành nghiên cứu và thu thập các tài liệu liên quan đến RSS Feed, các ứng dụng và website sử dụng RSS Feed hiệu quả. Các tài liệu về công nghệ xử lý RSS, các framework hỗ trợ, cơ sở dữ liệu và các công nghệ web liên quan.
3. Tổng hợp và tiến hành phân tích các tài liệu thu thập, đặc biệt là các nghiên cứu về hiệu quả của RSS Feed trong việc cập nhật và chia sẻ thông tin.
4. Chọn lọc và sử dụng các tài liệu vào báo cáo đề tài nghiên cứu, đảm bảo tính cập nhật và phù hợp với mục tiêu đề ra.
5. Tiến hành khảo sát các website đang sử dụng RSS Feed, đưa ra những yêu cầu chức năng và phi chức năng cho hệ thống sẽ xây dựng.
6. Phân tích thiết kế hệ thống dựa trên những khảo sát và yêu cầu nêu ra, tập trung vào việc tối ưu hóa quá trình xử lý và hiển thị dữ liệu từ RSS Feed.
7. Sử dụng các công cụ, ngôn ngữ lập trình và framework phù hợp để tiến hành xây dựng website ứng dụng RSS Feed dựa trên các phân tích yêu cầu thực tế.
8. Thực hiện các bài kiểm thử để đảm bảo tính ổn định và hiệu quả của hệ thống, đặc biệt là khả năng xử lý nhiều nguồn RSS Feed cùng lúc.
9. Tổng hợp các kiến thức, kết quả nghiên cứu và hoàn thành báo cáo đề tài nghiên cứu.

## 5. Kết quả dự kiến

* Báo cáo đồ án tốt nghiệp
* Phần mềm ứng dụng website tin mới sử dụng nodejs và javascript

## 6. Bố cục đề tài

*Nội dung của báo cáo gồm 3 chương:*

**Chương 1: Tổng quan về đề tài**

Giới thiệu về các công nghệ, mô hình ứng dụng web trong thương mại điện tử. Các công nghệ, công cụ được sử dụng trong đề tài.

**Chương 2: Phân tích thiết kế hệ thống**

Từ những dữ liệu đã thu thập được qua khảo sát, tiến hành vẽ các biểu đồ chức năng của hệ thống ở mức tổng quát, mô tả chi tiết các chức năng, vẽ các biểu đồ trình tự, biểu đồ lớp… Từ những biểu đồ lớp thiết kế cơ sở dữ liệu phù hợp cho ứng dụng web

**Chương 3: Thử nghiệm và đánh giá**

Từ kết quả thu được qua bước phân tích thiết kế hệ thống, tiến hành cài đặt ứng dụng web đáp ứng được các chức năng đã đề ra. Lên kế hoạch kiểm thử và đánh giá hệ thống

# TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu về RSS

### Khái niệm

RSS là tài liệu định dạng tập tin thuộc họ XML dùng trong việc chia sẻ tin tức Web (Web syndication) được dùng bởi nhiều website tin tức và weblog.

Công nghệ của RSS cho phép người dùng Internet có thể đặt mua thông tin từ các websites có cung cấp khả năng RSS (RSS feeds); chúng thường là các site có nội dung thay đổi và được thêm vào thường xuyên. Để có thể dùng công nghệ này, người quản trị site đó tạo ra hay quản lý một phần mềm chuyên dụng (như là một hệ thống quản lý nội dung - content management system-CMS), mà với định dạng XML máy có thể đọc được, có thể biểu diễn các bài tin mới thành một danh sách, với một hoặc hai dòng cho mỗi bài tin và một liên kết đến bài tin đầy đủ đó. Khác với việc mua nhiều ấn bản của các tờ báo hay tạp chí in giấy, hầu hết việc mua RSS là miễn phí.

Định dạng RSS cung cấp nội dung web và tóm lược nội dung web cùng với các liên kết đến phiên bản đầy đủ của nội dung tin đó, và các siêu-dữ-liệu (meta-data) khác. Thông tin này được cung cấp dưới dạng một tập tin XML được gọi là một RSS feed, webfeed, RSS stream, hay RSS channel. Cùng với việc hỗ trợ cung cấp chia sẻ thông tin, RSS cho phép những độc giả thường xuyên của một website có thể theo dõi các cập nhật của site đó dùng aggregator (Người tổng hợp).

### Cách sử dụng

RSS được dùng phổ biến bởi cộng đồng weblog để chia sẻ những tiêu đề tin tức mới nhất hay toàn bộ nội dung của nó, và ngay cả các tập tin đa phương tiện đính kèm. (Xem podcasting, vodcasting, broadcasting, screencasting, Vloging, và MP3 blogs.) Vào giữa năm 2000, việc sử dụng RSS trở nên phổ biến đối với hãng tin tức lớn, bao gồm Reuters, CNN, và BBC. Những nhà cung cấp tin này cho phép các website khác tổng hợp những tiêu đề tin tức "được chia sẻ" hay cung cấp các tóm tắt ngắn gọn của các bản tin chính dưới nhiều hình thức thỏa hiệp khác nhau. RSS ngày nay được dùng cho nhiều mục đích, bao gồm tiếp thị, báo cáo lỗi (bug-reports), hay các hoạt động khác bao gồm cập nhật hay xuất bản định kì.

Một chương trình gọi là một feed reader (đầu đọc nguồn cấp dữ liệu) hay aggregator (công cụ tổng hợp) có thể kiểm tra xem một website có hỗ trợ RSS cho người dùng không và, nếu có, hiển thị những bài viết cập nhật nhất mà nó tìm thấy từ website đó. Ngày nay có thể tìm thấy RSS feeds trên rất nhiều Websites lớn, cũng như nhiều những site nhỏ.

Các công cụ đọc tin phía trình khách và công cụ aggregators thường được xây dựng thành một chương trình độc lập hoặc là một phần mở rộng của các chương trình có sẵn như trình duyệt web. Những chương trình như vậy có mặt trên nhiều hệ điều hành khác nhau. Xem danh sách các aggregators chuyên về tin tức.

Các phần mềm thu thập tin tức như trên không đòi hỏi phải cài đặt và có thể sử dụng trên các máy tính có kết nối Internet. Một số aggregators (công cụ tổng hợp) kết hợp khả năng chia sẻ tin tức, ví dụ: lấy mọi thông tin bóng đá từ nhiều nguồn tin và cung cấp thành một nguồn tin mới. Đây cũng chính là các động cơ tìm kiếm nội dung được đăng tải thông qua RSS feeds như Feedster hay Blogdigger.

Trên các trang web, RSS feeds thường được liên kết bằng một hình chữ nhật màu cam, có thể kèm theo các ký tự XML hay RSS.

### Lịch sử

Trước RSS, có nhiều định dạng khác cũng từng được dùng cho vấn đề chia sẻ thông tin, nhưng không có định dạng nào được dùng rộng rãi cho đến ngày nay, vì hầu hết chủ yếu dùng cho từng dịch vụ đơn. Ví dụ, năm 1997 Microsoft tạo ra Channel Definition Format cho chức năng Active Channel của Internet Explorer 4.0. Dave Winer cũng đã thiết kế định dạng XML cho việc chia sẻ thông tin riêng cho Scripting News weblog, ra đời năm 1997.

RDF (Resource Description Framework) Site Summary, phiên bản đầu tiên của RSS, được tạo ra bởi Dan Libby của Netscape vào tháng 3 năm 1999 dùng cho cổng điện tử My Netscape. Phiên bản này trở thành RSS 0.9. Vào tháng 7 năm 1999, đáp trả lại các đề nghị và góp ý, Libby đưa ra bản phác thảo ban đầu đặt tên là RSS 0.91 (RSS viết tắt của Rich Site Summary), nhằm đơn giản hóa định dạng và tích hợp một số phần trong định dạng scriptingNews của Winer. Từ đó, Libby đề xuất ra định dạng tương tự-RSS 1.0 thông qua cái gọi là Futures Document.

Chẳng bao lâu sau, Netscape không còn tập trung vào RSS/XML, bỏ rơi định dạng đó. Một nhóm làm việc và danh sách địa chỉ mail, RSS-DEV, được thành lập bởi nhiều người dùng và cộng đồng XML để tiếp tục phát triển nó. Cùng thời điểm, Winer đưa ra phiên bản sửa đổi của RSS 0.91 cho website Userland, vì nó đang được dùng trong sản phẩm của họ. Ông ta cho rằng đặc tả kĩ thuật của RSS 0.91 là tài sản riêng của công ty ông, UserLand Software. Vì chẳng có bên nào có tuyên bố chính thức về tên của định dạng, cho nên bây giờ có nhiều tên gọi.

Nhóm RSS-DEV tiếp tục đưa ra RSS 1.0 vào tháng 12 năm 2000 dựa trên bản phác thảo góp ý sửa đổi cho bản đặc tả kĩ thuật đưa ra bởi Tristan Louis. Giống với RSS 0.9 (không phải 0.91) bản này dựa vào đặc tả kĩ thuật của RDF, nhưng có tính khả mở hơn, với nhiều mục bắt nguồn từ các từ vựng metadata chuẩn như Dublin Core.

Mười chín ngày sau, Winer cho ra phiên bản RSS 0.92, a một vài chỉnh sửa có tính tương thích với các thay đổi của RSS 0.91 dựa trên cùng bản góp ý. Vào tháng 4 năm 2001, ông đưa ra bản phác thảo của RSS 0.93 mà hầu hết là giống với bản 0.92. Bản thảo RSS 0.94 ra đời vào tháng Tám, phục hồi lại những thay đổi trong bản 0.93, và thêm vào thuộc tính (attribute) type cho thành phần (element) description.

Vào tháng 9 năm 2002, Winer cho ra bản cuối cùng của RSS 0.92, bây giờ gọi là RSS 2.0 và nhấn mạnh "Really Simple Syndication" là nghĩa của ba ký tự viết tắt RSS. Đặc tả kĩ thuật của RSS 2.0 loại bỏ thuộc tính type từng được thêm vào trong RSS 0.94 và cho phép người dùng có thể thêm thành phần mở rộng nhờ dùng XML namespaces. Nhiều phiên bản của RSS 2.0 đã được ra đời, nhưng chỉ số của phiên bản thì vẫn không thay đổi.

Vào tháng Mười Một, 2002, Thời báo New York đã bắt đầu cung cấp cho người đọc khả năng mục các tin có hỗ trợ RSS feeds liên quan đến nhiều chủ đề khác nhau. Vào tháng Giêng, 2003, David Winer đã gọi việc dùng RSS của Thời báo New York Time là một "điểm nhấn" (tipping point) trong việc đưa định dạng RSS trở thành một chuẩn.

Vào tháng Bảy, 2003, Winer và Userland Software được cấp quyền sở hữu của đặc tả kĩ thuật RSS 2.0, Trung tâm Berkman về Xã hội và Internet của Harvard.

Winer đã bị phê bình vì đã đơn phương tạo ra định dạng mới và tự đưa ra số của phiên bản. Để đáp lại, đồng tác giả của RSS 1.0 Aaron Swartz đã đưa ra RSS 3.0, một định dạng văn bản không dựa trên XML. Định dạng đó chỉ là một sự bắt chước và chỉ được dùng rất ít.

Vào tháng Giêng 2005, Sean B. Palmer và Christopher Schmidt đã cho ra bản sơ thảo đầu tiên của RSS 1.1 Nó là bản sửa lỗi cho 1.0, loại bỏ những đặc tính ít dùng, đơn giản hóa cú pháp và nâng cao đặc tả kĩ thuật dựa vào các đặc tả RDF. Vào tháng 7 năm 2005, RSS 1.1 chỉ hơn một bài tập mang tính học thuật một ít.

## Khảo sát sơ bộ

### **Mục đích**

Mục đích chính của đề tài này là nghiên cứu sâu về RSS feed và ứng dụng nó vào việc xây dựng một website tin tức hiện đại có tên "Tin Mới". Website này sẽ tự động cập nhật và tổng hợp thông tin từ nhiều nguồn khác nhau, sử dụng Node.js và JavaScript làm nền tảng phát triển. Cụ thể là:

- Tìm hiểu cơ chế hoạt động và ưu điểm của RSS feed trong việc thu thập thông tin.

- Khám phá khả năng của Node.js trong việc xử lý dữ liệu RSS một cách hiệu quả.

- Áp dụng JavaScript để tạo giao diện người dùng động và tương tác.

- Xây dựng một hệ thống tin tức tự động, cập nhật liên tục và dễ dàng mở rộng.

### Đối tượng khảo sát

* RSS feed: Cấu trúc, cách thức hoạt động, các định dạng phổ biến.
* Node.js: Khả năng xử lý bất đồng bộ, các module hỗ trợ đọc và phân tích RSS feed.
* JavaScript: Các framework và thư viện front-end phù hợp cho việc hiển thị tin tức động.
* Springboot: Các thư viện phù hợp cho hệ thống
* Các website tin tức hiện có sử dụng RSS: Phân tích cấu trúc, giao diện và cách tổ chức thông tin, ví dụ trang Báo mới.

### Phương pháp

* Nghiên cứu tài liệu: Thu thập và phân tích các tài liệu kỹ thuật về RSS, Node.js, và JavaScript.
* Phân tích so sánh: Đánh giá ưu nhược điểm của các website tin tức đang sử dụng RSS.
* Thực nghiệm: Xây dựng các prototype nhỏ để kiểm tra khả năng tích hợp RSS với Node.js.
* Phỏng vấn và khảo sát: Thu thập ý kiến từ người dùng về nhu cầu và mong muốn đối với một website tin tức.

### Kết quả sơ bộ

* Xác định được tiềm năng của RSS feed trong việc tự động hóa quá trình cập nhật tin tức.
* Đánh giá được khả năng của Node.js trong việc xử lý hiệu quả các luồng dữ liệu RSS.
* Lựa chọn được các công nghệ JavaScript phù hợp cho phần front-end của website.
* Phác thảo được kiến trúc tổng thể của hệ thống "Tin Mới".

## Khảo sát hiện trạng cụ thể

Trong phần này, em sẽ phân tích chi tiết 4 trang web tin tức lớn của Việt Nam đang cung cấp RSS:

* VnExpress (https://vnexpress.net/rss/tin-moi-nhat.rss)
* Thanh Niên (https://thanhnien.vn/rss/tin-24h.rss)
* Tuổi Trẻ (https://tuoitre.vn/rss/tin-moi-nhat.rss)
* Người Đưa Tin (https://www.nguoiduatin.vn/trang-chu.rss)

Phân tích chi tiết:

### VnExpress

a) Đa dạng nguồn tin:

* Cung cấp tin tức từ nhiều lĩnh vực: thời sự, kinh doanh, thể thao, giải trí, khoa học, giáo dục.
* Có các chuyên mục riêng biệt cho từng lĩnh vực, giúp độc giả dễ dàng theo dõi các chủ đề quan tâm.

b) Tốc độ cập nhật:

* Cập nhật liên tục 24/7.
* Trung bình có khoảng 15-20 bài viết mới mỗi giờ trong các khung giờ cao điểm.

c) Thời gian tải trang:

* RSS feed tải nhanh, thường dưới 2 giây.
* Nội dung được tối ưu hóa cho việc đọc nhanh trên các ứng dụng đọc tin RSS.

### Thanh Niên

a) Đa dạng nguồn tin:

* Cung cấp tin tức từ nhiều lĩnh vực: thời sự, xã hội, thế giới, văn hóa, thể thao.
* Có các chuyên mục đặc biệt như "Người Trẻ", tập trung vào các vấn đề liên quan đến giới trẻ.

b) Tốc độ cập nhật:

* Cập nhật thường xuyên, tập trung vào các giờ cao điểm.
* Trung bình có khoảng 10-15 bài viết mới mỗi giờ.

c) Thời gian tải trang:

* RSS feed tải tương đối nhanh, thường trong khoảng 2-3 giây.
* Đôi khi có độ trễ nhỏ trong việc cập nhật tin mới nhất.

### Tuổi Trẻ

a) Đa dạng nguồn tin:

* Cung cấp tin tức từ nhiều lĩnh vực: thời sự, xã hội, thế giới, pháp luật, kinh tế.
* Có các chuyên mục đặc biệt như "Nhịp sống trẻ", tập trung vào các vấn đề liên quan đến giới trẻ và giáo dục.

b) Tốc độ cập nhật:

* Cập nhật đều đặn, với tần suất cao trong các giờ cao điểm.
* Trung bình có khoảng 12-18 bài viết mới mỗi giờ.

c) Thời gian tải trang:

* RSS feed tải khá nhanh, thường trong khoảng 2-3 giây.
* Nội dung được tối ưu hóa tốt cho việc đọc trên các ứng dụng RSS.

### Người Đưa Tin

a) Đa dạng nguồn tin:

* Cung cấp tin tức từ nhiều lĩnh vực: thời sự, xã hội, thế giới, pháp luật, đời sống, kinh tế.
* Có các chuyên mục đặc biệt như "Đa chiều", tập trung đa dạng các vấn đề.

b) Tốc độ cập nhật:

* Cập nhật đều đặn, với tần suất cao trong các giờ cao điểm.
* Trung bình có khoảng 10-15 bài viết mới mỗi giờ.

c) Thời gian tải trang:

* RSS feed tải khá nhanh, thường trong khoảng 2-3 giây.
* Nội dung được tối ưu hóa tốt cho việc đọc trên các ứng dụng RSS.

### Đánh giá tổng quan

Ưu điểm:

* Cả 4 trang báo đều cung cấp nguồn tin đa dạng, phong phú, bao quát nhiều lĩnh vực.
* Tốc độ cập nhật nhanh, đáp ứng nhu cầu thông tin kịp thời của độc giả.
* Thời gian tải trang RSS nhìn chung nhanh, tạo trải nghiệm đọc tin tốt cho người dùng.

Nhược điểm:

* Đôi khi có sự chênh lệch về tốc độ cập nhật giữa các trang, đặc biệt trong các sự kiện đột xuất.
* Thời gian tải trang có thể bị ảnh hưởng trong giờ cao điểm hoặc khi có nhiều người truy cập cùng lúc.

### Cơ hội cải tiến

* Cải thiện hơn nữa tốc độ cập nhật và đồng bộ hóa giữa trang web chính và RSS feed.
* Tối ưu hóa thời gian tải trang, đặc biệt là trong giờ cao điểm.
* Cân nhắc việc cung cấp các RSS feed chuyên biệt cho từng chủ đề để người dùng có thể tùy chọn theo sở thích.

### Báo mới

#### Giới thiệu

Báo mới là trang web tổng hợp tin tức trực tuyến, sử dụng RSS để thu thập và hiển thị tin tức từ nhiều nguồn báo uy tín khác nhau.

#### Đặc điểm chính

* Tổng hợp tin tức từ nhiều nguồn báo chính thống
* Cập nhật tin tức liên tục và tự động thông qua RSS
* Phân loại tin tức theo chủ đề: Thời sự, Kinh tế, Thể thao, Giải trí, Công nghệ, v.v.
* Giao diện thân thiện, dễ sử dụng trên nhiều thiết bị

#### Tính năng nổi bật

* Tự động cập nhật tin tức từ các nguồn RSS
* Tìm kiếm và lọc tin tức theo nguồn, chủ đề
* Tùy chọn cá nhân hóa feed tin tức
* Chia sẻ bài viết lên mạng xã hội
* Chế độ đọc offline (tùy chọn)

#### Mục tiêu

1. Cung cấp một nơi tập trung để đọc tin tức từ nhiều nguồn
2. Tiết kiệm thời gian cho người dùng trong việc theo dõi tin tức
3. Xây dựng cộng đồng độc giả quan tâm đến tin tức đa chiều

## Các liên kết rss được sử dụng

Bảng 1: Liên kết RSS cho Trang Chủ

|  |
| --- |
| **Trang Chủ** |
| <https://vnexpress.net/rss/tin-moi-nhat.rss> |
| <https://thanhnien.vn/rss/tin-24h.rss> |
| <https://tuoitre.vn/rss/tin-moi-nhat.rss> |
| <https://www.nguoiduatin.vn/trang-chu.rss> |

Bảng 2: Liên kết RSS cho Các Chuyên Mục

|  |  |
| --- | --- |
| **Chuyên Mục** | **URL** |
| Thời sự | <https://vnexpress.net/rss/thoi-su.rss> |
|  | <https://tuoitre.vn/rss/thoi-su.rss> |
|  | <https://www.nguoiduatin.vn/rss/toan-canh/tieu-diem.rss> |
| Thế giới | <https://vnexpress.net/rss/the-gioi.rss> |
|  | <https://tuoitre.vn/rss/the-gioi.rss> |
|  | <https://www.nguoiduatin.vn/rss/toan-canh/the-gioi.rss> |
| Kinh doanh | <https://vnexpress.net/rss/kinh-doanh.rss> |
|  | <https://tuoitre.vn/rss/kinh-doanh.rss> |
|  | <https://www.nguoiduatin.vn/rss/kinh-te.rss> |
| Giải trí | <https://vnexpress.net/rss/giai-tri.rss> |
|  | <https://tuoitre.vn/rss/giai-tri.rss> |
| Du lịch | <https://vnexpress.net/rss/du-lich.rss> |
|  | <https://tuoitre.vn/rss/du-lich.rss> |
| Đời sống | <https://vnexpress.net/rss/gia-dinh.rss> |
|  | <https://tuoitre.vn/rss/van-hoa.rss> |
|  | <https://www.nguoiduatin.vn/rss/doi-song/gia-dinh.rss> |
| Khoa học | <https://vnexpress.net/rss/khoa-hoc.rss> |
|  | <https://tuoitre.vn/rss/khoa-hoc.rss> |
| Giáo dục | <https://vnexpress.net/rss/giao-duc.rss> |
|  | <https://tuoitre.vn/rss/giao-duc.rss> |
|  | <https://www.nguoiduatin.vn/rss/xa-hoi/giao-duc.rss> |
| Tâm sự | <https://vnexpress.net/rss/tam-su.rss> |
|  | <https://tuoitre.vn/rss/nhip-song-tre.rss> |
|  | <https://www.nguoiduatin.vn/rss/da-chieu/talkshow.rss> |
| Thể thao | <https://vnexpress.net/rss/the-thao.rss> |
|  | <https://tuoitre.vn/rss/the-thao.rss> |
|  | <https://www.nguoiduatin.vn/rss/the-thao.rss> |

## Công nghệ sử dụng để thực hiện đề tài

### Nextjs

NextJS là một framework có mã nguồn mở được xây dựng trên nền tảng của React, cho phép xây dựng các trang web tĩnh có tốc độ siêu nhanh và thân thiện với người dùng, cũng như xây dựng các ứng dụng web React.

NextJS được ra đời vào năm 2016, thuộc sở hữu của Vercel. NextJS bắt đầu trở nên phổ biến vào năm 2018 và tiếp tục tăng trưởng mạnh mẽ trong cộng đồng phát triển web. Sự kết hợp của các tính năng như Server-side Rendering (SSR) với Static Site Generation (SSG) đã giúp NextJS trở thành sự lựa chọn hấp dẫn cho nhiều dự án.

**Đặc điểm chính của nextjs:**

* Server-side Rendering (SSR): Next.js hỗ trợ render trang web trên server, giúp cải thiện hiệu suất và SEO.
* Static Site Generation (SSG): Cho phép tạo các trang tĩnh tại thời điểm build.
* API Routes: Tích hợp API routes, cho phép tạo API endpoints trong cùng một dự án.
* Tối ưu hóa tự động: Tự động phân chia code, tối ưu hóa hình ảnh và font chữ.
* Routing dựa trên file system: Cấu trúc thư mục và file quyết định routing.
* Hot Module Replacement: Cập nhật thay đổi mà không cần refresh trình duyệt.
* TypeScript Support: Hỗ trợ TypeScript mặc định.
* CSS Support: Hỗ trợ CSS Modules, Sass, và các giải pháp styling khác.
* Image Optimization: Tối ưu hóa hình ảnh tự động.
* Zero Config: Cấu hình tối thiểu để bắt đầu một dự án.

### Tailwind css

Là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau và quan trọng, dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

**Đặc điểm chính của Tailwind**

* Utility-first: Sử dụng các lớp tiện ích nhỏ, có mục đích cụ thể để xây dựng giao diện.
* Highly customizable: Dễ dàng tùy chỉnh thông qua file cấu hình.
* Responsive design: Hỗ trợ thiết kế responsive với các tiền tố như sm:, md:, lg:, xl:.
* Low-level control: Cung cấp kiểm soát chi tiết về styling mà không cần viết CSS tùy chỉnh.
* Pseudo-classes: Hỗ trợ các pseudo-class như hover, focus, active.
* Dark mode: Hỗ trợ chế độ tối với tiền tố dark.
* Preflight: Reset CSS mặc định để đảm bảo tính nhất quán giữa các trình duyệt.
* JIT (Just-In-Time) mode: Tạo CSS động khi cần, giúp giảm kích thước file CSS.
* Plugins: Hệ thống plugin mạnh mẽ để mở rộng chức năng.
* PurgeCSS integration: Loại bỏ CSS không sử dụng trong production.
* Flexbox và Grid: Hỗ trợ mạnh mẽ cho layout với Flexbox và Grid.
* Mobile-first: Thiết kế theo hướng mobile-first.

### CI/CD

CI/CD là viết tắt của Continuous Integration/Continuous Deployment (Tích hợp liên tục/ Triển khai liên tục). Đây là một phương pháp phát triển phần mềm nhằm tối ưu hóa quy trình phát triển và triển khai ứng dụng.

Continuous Integration (CI) đề cập đến việc liên tục tích hợp các thay đổi vào mã nguồn của ứng dụng. Khi một nhóm phát triển làm việc cùng nhau, CI đảm bảo rằng các phiên bản mới nhất của mã nguồn được tích hợp vào một kho lưu trữ chung một cách tự động. Việc này giúp phát hiện sớm các lỗi hợp nhất và xung đột giữa các thành viên trong nhóm, đồng thời đảm bảo mã nguồn luôn ổn định.

Continuous Deployment (CD) liên quan đến việc triển khai tự động các phiên bản mới nhất của ứng dụng vào môi trường sản phẩm hoặc môi trường thử nghiệm. CD giúp giảm thời gian và công sức cần thiết để triển khai ứng dụng, đồng thời tăng tính nhất quán và độ tin cậy của quy trình triển khai.

Với CI/CD, mỗi khi có thay đổi trong mã nguồn, mã sẽ được tự động kiểm tra, tích hợp và triển khai một cách liên tục, giúp đảm bảo chất lượng phần mềm và tăng tốc độ phát triển. Nó cũng cho phép các nhóm phát triển và quản lý ứng dụng dễ dàng hợp nhất và triển khai các thay đổi một cách nhanh chóng và đáng tin cậy.

#### Mục đích

* Tự động hóa quy trình phát triển
* Phát hiện lỗi sớm
* Cải thiện chất lượng phần mềm
* Tăng tốc độ phát hành

#### Các thành phần chính

* Hệ thống quản lý mã nguồn (ví dụ: Git)
* Máy chủ CI/CD (ví dụ: Jenkins, GitLab CI, CircleCI)
* Công cụ kiểm thử tự động
* Môi trường triển khai

#### Quy trình CI

* Developers push code lên repository
* Tự động build và test
* Báo cáo kết quả

#### Quy trình CD

* Tự động triển khai đến môi trường staging
* Kiểm tra và xác nhận
* Triển khai tự động hoặc bán tự động đến production

### Ant Design

Ant Design là một thư viện UI (User Interface) phổ biến cho React, được phát triển bởi Ant Group (một công ty công nghệ Trung Quốc)

#### Các đặc điểm chính:

* Bộ sưu tập lớn các components: Buttons, forms, tables, modals, v.v.
* Thiết kế theo nguyên tắc của Material Design
* Hỗ trợ đầy đủ cho TypeScript
* Đáp ứng và thân thiện với thiết bị di động
* Chủ đề và tùy chỉnh: Dễ dàng thay đổi màu sắc và styles

#### Ưu điểm:

* Dễ sử dụng và tài liệu chi tiết
* Hiệu suất tốt
* Cộng đồng lớn và active
* Cập nhật thường xuyên

#### Nhược điểm:

* Bundle size tương đối lớn (có thể giảm bằng cách import từng component)

### Docker

Docker là một nền tảng mã nguồn mở được sử dụng để phát triển, triển khai và chạy các ứng dụng trong các môi trường được cô lập gọi là containers. Docker tạo ra một cấu trúc tiêu chuẩn để đóng gói ứng dụng cùng với tất cả các phụ thuộc của nó vào một container, đảm bảo rằng ứng dụng sẽ chạy một cách nhất quán trên các môi trường khác nhau.

### SpringBoot

Spring Boot là một framework thuộc hệ sinh thái Spring, được phát triển bởi Pivotal Software. Nó cung cấp một cách tiếp cận đơn giản và nhanh chóng để tạo ra các ứng dụng Spring, mà không cần phải cấu hình quá nhiều.

Đây là một số đặc điểm nổi bật của Spring Boot:

* **Tự động cấu hình (Auto-configuration):**

Spring Boot tự động cấu hình các thành phần cần thiết dựa trên các thư viện được thêm vào dự án.

Điều này giúp giảm đáng kể thời gian và nỗ lực cần thiết để thiết lập một ứng dụng Spring.

* **Starter dependencies:**

Spring Boot cung cấp các "starter dependencies" giúp bạn thêm các tính năng phổ biến vào ứng dụng một cách dễ dàng.

Ví dụ: spring-boot-starter-web, spring-boot-starter-data-jpa, spring-boot-starter-security, v.v.

* **Embedded servers:**

Spring Boot tích hợp sẵn các máy chủ web như Tomcat, Jetty hoặc Undertow, giúp bạn chạy ứng dụng mà không cần phải triển khai trên một máy chủ độc lập.

* **Actuator**:

Spring Boot Actuator cung cấp các endpoint để giám sát và quản lý ứng dụng, như health checks, metrics, information, v.v.

* **Profiles**:

Spring Boot cho phép bạn cấu hình các profile khác nhau (ví dụ: development, staging, production) để quản lý các cài đặt môi trường.

* **Tính linh hoạt và mở rộng:**

Spring Boot được xây dựng trên nền tảng của Spring Framework, vì vậy bạn vẫn có thể sử dụng tất cả các tính năng và công cụ của Spring.

## Tài liệu api

### Giới thiệu

API này được thiết kế để hỗ trợ quản lý hệ thống quảng cáo, với các chức năng chính bao gồm:

* Quản lý trang quảng cáo
* Quản lý chuyên mục quảng cáo
* Gửi yêu cầu phê duyệt quảng cáo
* Phê duyệt quảng cáo
* Quản lý tài khoản
* Quản lý vai trò
* Quản lý hồ sơ cá nhân

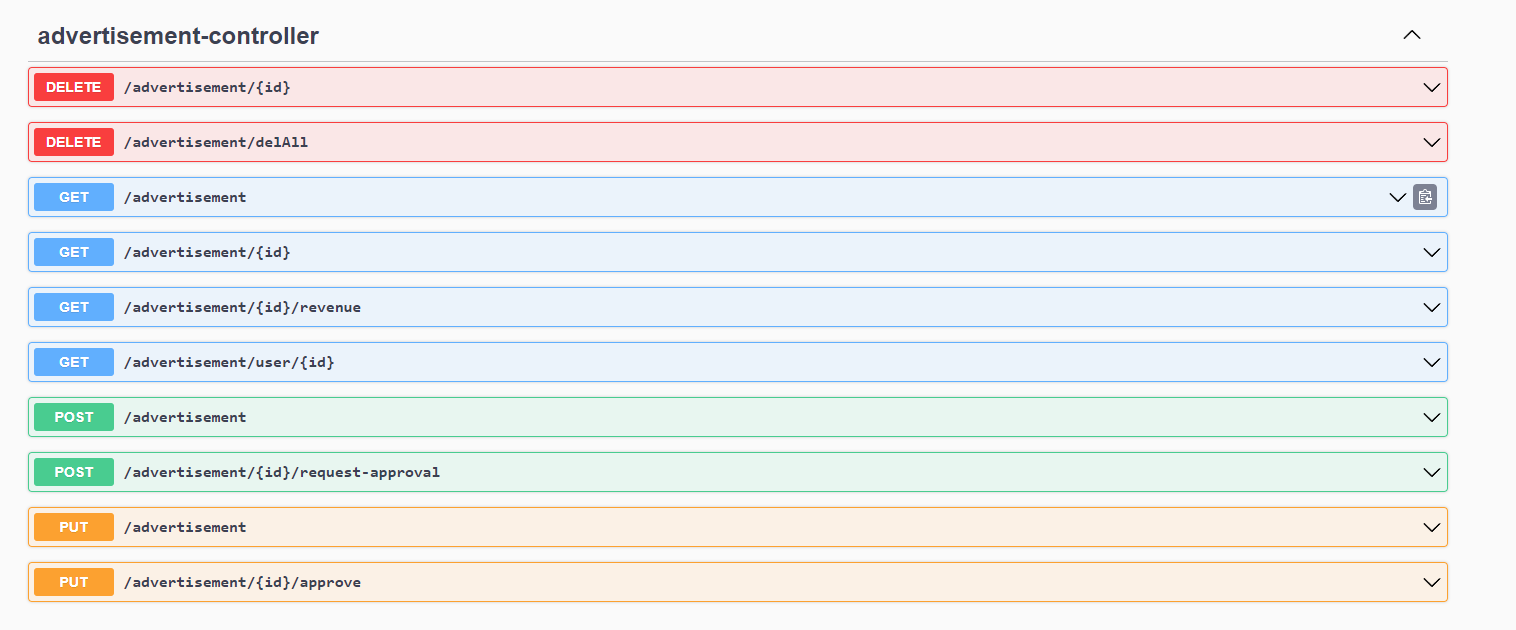
API cung cấp các endpoint để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên quảng cáo, xử lý quy trình phê duyệt, và quản lý dữ liệu liên quan đến hệ thống quảng cáo.

### Thông tin chung

* URL: [https://graduation-be-vlen.onrender.com/swagger ui/index.html](https://graduation-be-vlen.onrender.com/swagger-ui/index.html)
* Các controller: advertisement-controller, advertising-field-controller, approval-request-controller, advertising-field-controller, authentication-controller, role-controller, user-controller.

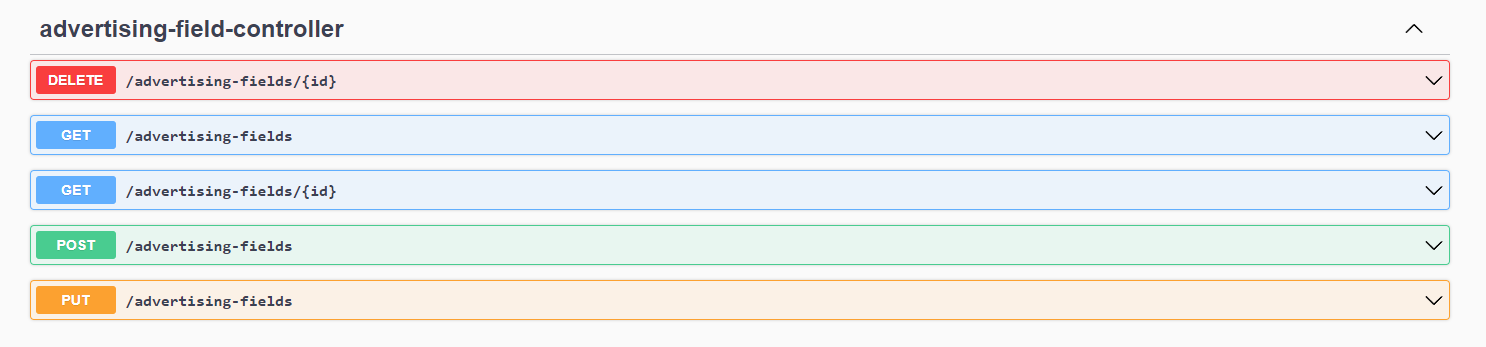
### Thông tin các điểm cuối

* advertisement-controller



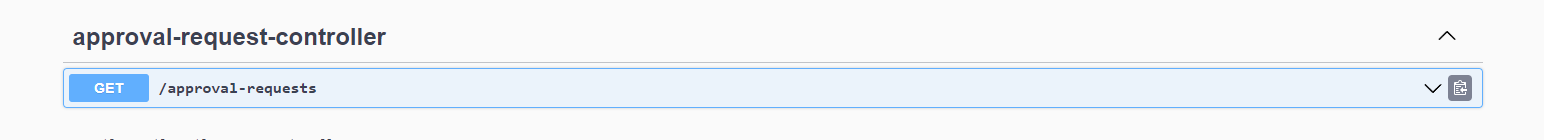
Hình 1.1: Điểm cuối advertisement-controller

* advertising-field-controller



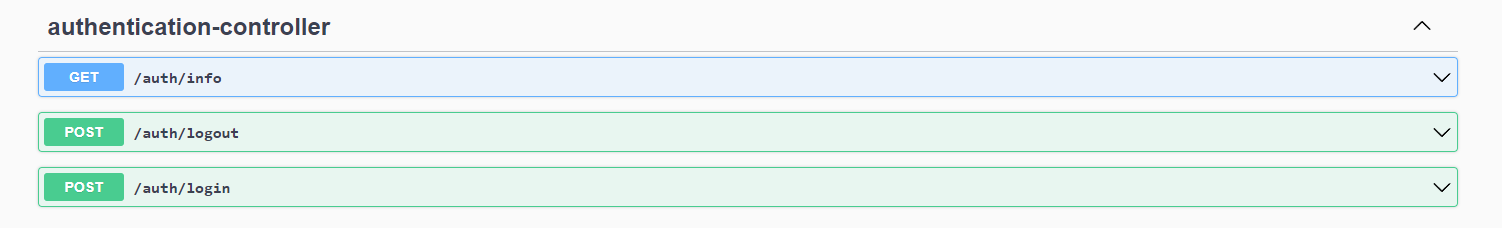
Hình 1.2: Điểm cuối advertising-field-controller

* approval-request-controller



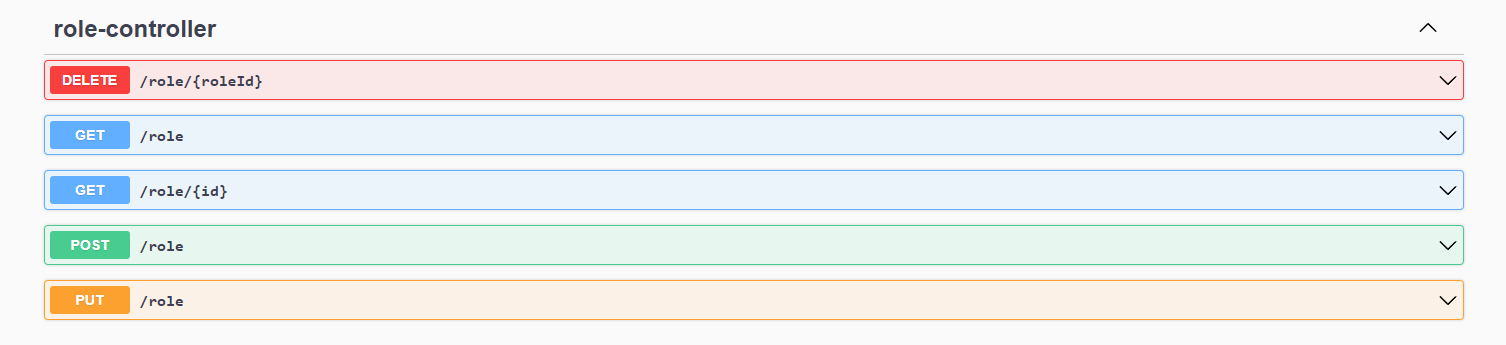
Hình 1.3: Điểm cuối approval-request-controller

* authentication-controller



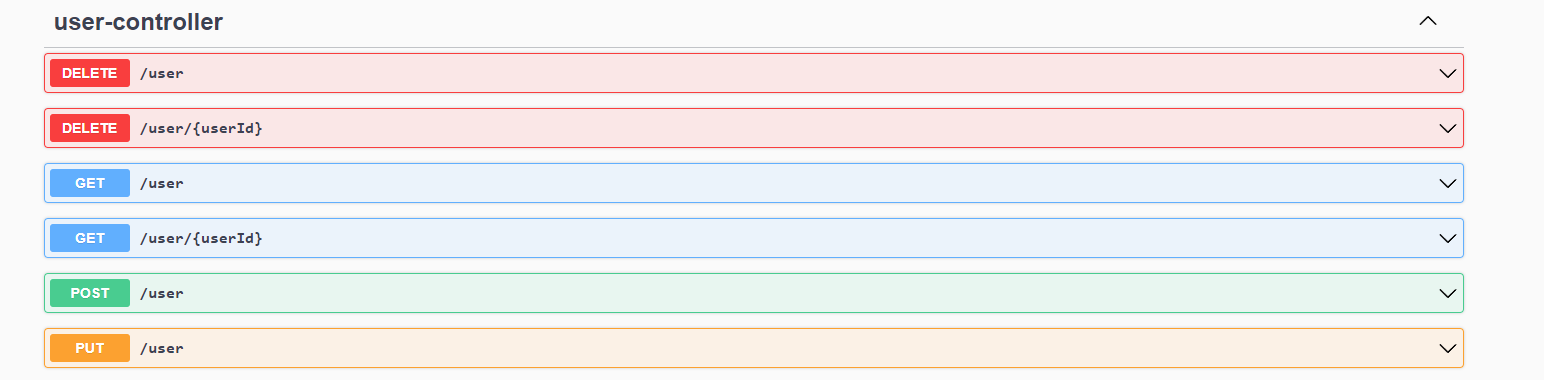
Hình 1.4: Điểm cuối authentication-controller

* role-controller



Hình 1.5: Điểm cuối role-controller

* user-controller



Hình 1.6: Điểm cuối user-controller

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Khảo sát bài toán

### Giới thiệu về trang web tin mới

Trang web Tin Mới là nguồn cung cấp thông tin đa dạng và cập nhật nhất. Em tổng hợp tin tức từ nhiều nguồn uy tín thông qua RSS, mang đến cho độc giả nội dung phong phú và kịp thời về mọi lĩnh vực.

Ngoài việc cung cấp tin tức, em còn cung cấp giải pháp quản lý quảng cáo hiệu quả cho các đối tác. Hệ thống của em cho phép theo dõi doanh thu quảng cáo chi tiết và thực hiện thanh toán một cách thuận tiện.

Với Tin Mới, người đọc tin sẽ luôn nắm bắt được thông tin mới nhất cùng cơ hội quảng bá thương hiệu hiệu quả.

### Mô tả bài toán

Bài toán xây dựng website Tin mới phải đáp ứng được các yêu cầu nghiệp vụ như sau:

1. Website phải có giao diện rõ ràng, dễ nhìn, dễ sử dụng.
2. Website phải đảm bảo tối ưu thời gian tải rss và hiển thị đúng.
3. Website phải đảm bảo về thời gian vận hàng thường xuyên, hạn chế tối đa thời gian ngắt quãng.
4. Với người đọc tin: Đáp ứng các chức năng tìm kiếm, xem danh mục tin tức, xem chi tiết bài viết một cách nhanh chóng, dễ dàng và thuận tiện nhất.
5. Với người quản trị: Có quyền kiểm soát mọi hoạt động của hệ thống trang website và thực hiện được các thao tác quản lý quảng cáo, quản lý tài khoản, quản lý vai trò, quản lý doanh thu … một cách linh hoạt thuận tiện và chính xác
6. Với nhà quảng cáo: Có quyền thực hiện được các thao tác quản lý quảng cáo, gửi phê duyệt quảng cáo … một cách linh hoạt và chính xác

## Xác định các yêu cầu của bài toán

### Các yêu cầu chức năng

* Tự động thu thập tin tức: Hệ thống có khả năng quét và cập nhật tin từ nhiều nguồn RSS khác nhau.
* Phân loại tin tức: Phân loại tin tức theo danh mục.
* Hiển thị tin tức: Trình bày tin tức theo layout rõ ràng, dễ đọc, có hình ảnh minh họa.
* Thay đổi bố cục giao diện: cho phép người dùng chọn những giao diện khác nhau
* Chọn theme cho trang web: cho phép người dùng chọn theme khác nhau
* Preview các thay đổi về website: cho phép người dùng xem trước các thay đổi giao diện trên website
* Tìm kiếm nâng cao: Cho phép người dùng tìm kiếm tin tức theo nhiều tiêu chí (từ khóa, ngày tháng, nguồn).
* Tùy chỉnh feed: Người dùng có thể tạo feed cá nhân dựa trên sở thích.

### Các yêu cầu phi chức năng

* Hiệu suất cao: Thời gian phản hồi nhanh, xử lý được lượng lớn request đồng thời.
* Giao diện responsive: Hiển thị tốt trên mọi thiết bị.
* Bảo mật: Đảm bảo an toàn thông tin, chống tấn công DDoS và SQL injection.
* Khả năng mở rộng: Dễ dàng thêm mới nguồn tin, tính năng mà không ảnh hưởng đến hệ thống.
* Độ tin cậy: Hệ thống hoạt động ổn định, có cơ chế backup và khôi phục dữ liệu.
* SEO friendly: Tối ưu hóa cho công cụ tìm kiếm để tăng lượng truy cập tự nhiên.

## Phân tích và thiết kế

### Mô hình hóa chức năng

Bảng 3: Các tác nhân

|  |  |  |
| --- | --- | --- |
| STT | Tên actor | Mô tả |
| 1 | RSS Feed (Website cung cấp rss) | Cung cấp data được cập nhật liên tục |
| 2 | Reader (Người đọc tin) | User có thể chọn chủ đề , layout giao diện , sử dụng chatbot , tìm kiếm bài viết , xem danh sách và chi tiết bài viết. |
| 3 | Advertiser (Nhà quảng cáo) | Advertiser có thể đăng nhập vào hệ thống với vai trò là nhà quảng cáo để tạo, chỉnh sửa, xóa quảng cáo, gửi phê duyệt. |
| 4 | Admin (Quản trị viên) | Admin có thể đăng nhập vào hệ thống với vai trò là quản trị viên để quản lý hệ thống. |

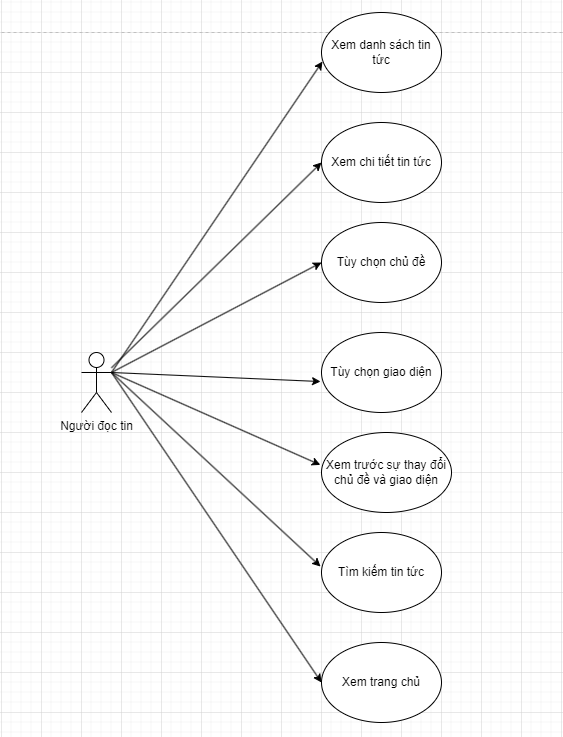
#### Biểu đồ UseCase

**Biểu đồ UseCase tổng quát**



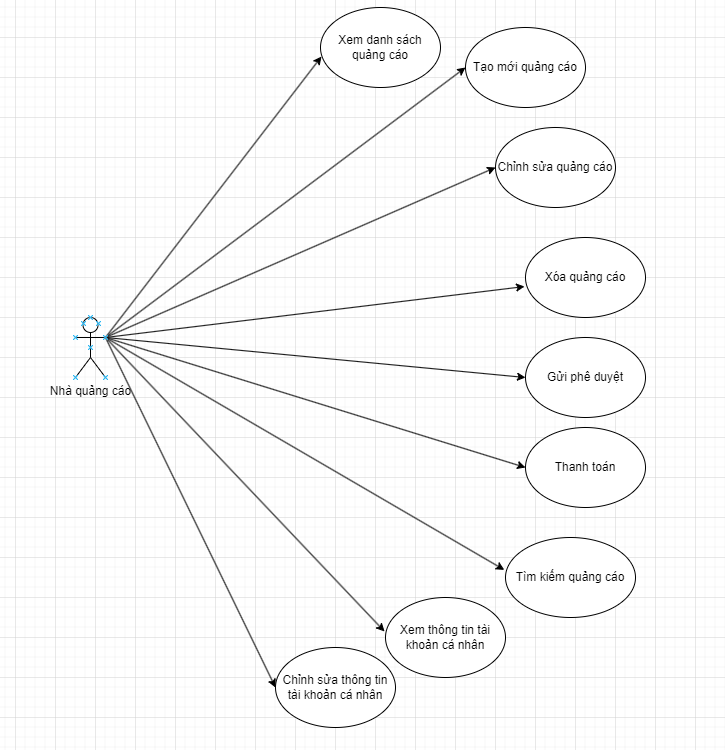
Hình 2.1: Biểu đồ use case toàn hệ thống

**Biểu đồ UseCase của actor người đọc tin**

****

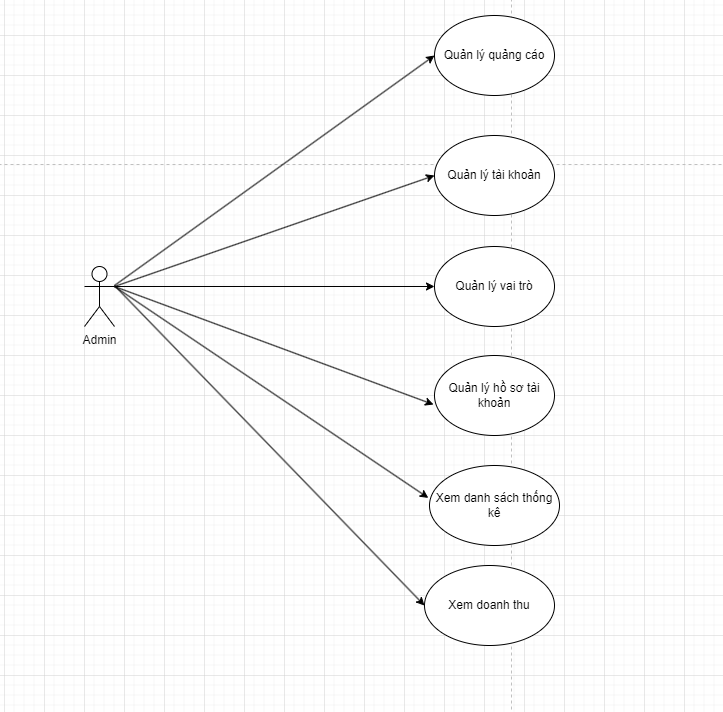
Hình 2.2:Biểu đồ use case người đọc tin

**Biểu đồ UseCase của actor nhà quảng cáo**



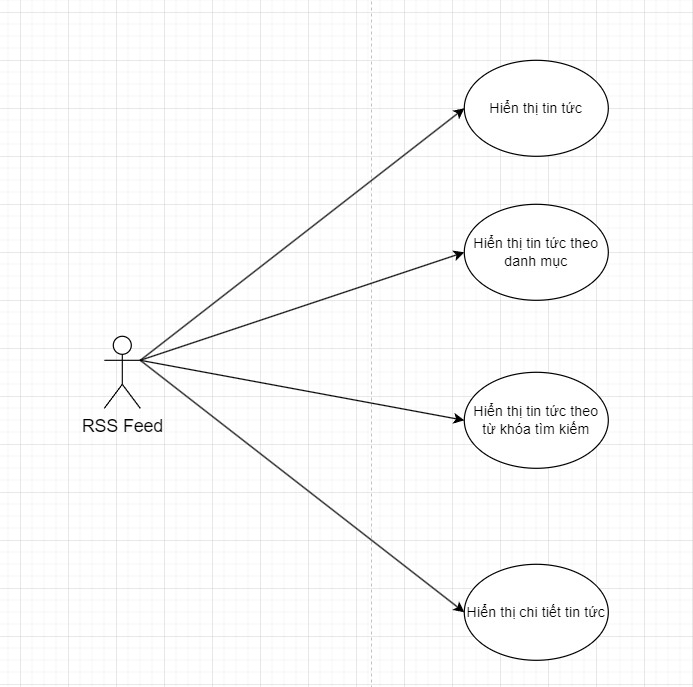
Hình 2.3:Biểu đồ use case nhà quảng cáo

**Biểu đồ UseCase của actor admin**



Hình 2.4:Biểu đồ use case admin

**Biểu đồ UseCase của RSS Feed**



Hình 2.5:Biểu đồ use case RSS Feed

### Mô tả các use case

#### Use case xem danh sách tin tức

Bảng 4:Mô tả chi tiết use case xem danh sách tin tức

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người đọc tin xem các bài viết thuộc danh mục |
| Actor | Người đọc tin |
| Tiền điều kiện | Không |
| Luồng cơ bản | 1. Use case này bắt đầu khi người đọc tin click vào các danh mục trên thanh điều hướng.  2. Hệ thống sẽ lấy data từ api hiển thị ra màn hình. Use case kết thúc. |
| Luồng rẽ nhánh | 1.Không đọc được dữ liệu từ rss feed: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không đọc được dữ liệu từ rss feed thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case xem chi tiết tin tức

Bảng 5:Mô tả chi tiết use case xem chi tiết tin tức

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người đọc tin xem chi tiết các bài viết tin tức |
| Actor | Người đọc tin |
| Tiền điều kiện | Không |
| Luồng cơ bản | 1. Use case này bắt đầu khi khách hàng click vào các bài viết bất kỳ ở trang chủ hoặc danh mục tin tức.  2. Hệ thống sẽ chuyển hướng người đọc sang trang chi tiết tin tức của trang báo.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không đọc được dữ liệu từ rss feed: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không đọc được dữ liệu từ rss feed thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case tùy chọn chủ đề

Bảng 6:Mô tả chi tiết use case tùy chọn chủ đề

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người đọc chọn chủ đề yêu thích |
| Actor | Người đọc tin |
| Tiền điều kiện | Không |
| Luồng cơ bản | 1. Use case này bắt đầu khi người đọc click vào icon bánh răng bên góc phải màn hình.  2. Hệ thống sẽ lấy data các chủ đề đang có và hiển thị ra màn hình.  3. Khi người dùng chọn chủ đề và click button “ Chọn ” hệ thống sẽ set lại chủ đề và cập nhật lại giao diện website.  Use case kết thúc. |
| Luồng rẽ nhánh | 1.Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu xảy ra lỗi thì hệ thống sẽ hiển thị một chủ đề mặc định và use case kết thúc. |

#### Use case tùy chọn giao diện

Bảng 7:Mô tả chi tiết use case tùy chọn giao diện

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người đọc chọn giao diện ưa thích |
| Actor | Người đọc tin |
| Tiền điều kiện | Không |
| Luồng cơ bản | 1. Use case này bắt đầu khi người đọc click vào icon bánh răng bên góc phải màn hình.  2. Hệ thống sẽ lấy data các layout giao diện đang có và hiển thị ra màn hình.  3. Khi người dùng chọn giao diện và click button “ Chọn ” hệ thống sẽ set giao diện và cập nhật lại giao diện website.  Use case kết thúc. |
| Luồng rẽ nhánh | 1.Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu xảy ra lỗi thì hệ thống sẽ hiển thị một giao diện mặc định và use case kết thúc. |

#### Use case xem trang chủ

Bảng 8:Mô tả chi tiết use case xem trang chủ

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép Người đọc tin xem các tin tức nóng, mới nhất ở trang chủ |
| Actor | Người đọc tin |
| Tiền điều kiện | Không |
| Luồng cơ bản | 1. Use case này bắt đầu khi người đọc tin truy cập vào website. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không đọc được dữ liệu từ rss feed: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không đọc được dữ liệu từ rss feed thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case quản lý tài khoản

Bảng 9:Mô tả chi tiết use case quản lý tài khoản

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép quản trị viên quản lý các tài khoản trong hệ thống |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản admin |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click icon đăng nhập.  2. Người quản trị đăng nhập bằng tài khoản có quyền là admin.  3. Chọn mục quản lý tài khoản, admin có thể tạo mới, chỉnh sửa, xóa, tìm kiếm tài khoản.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case quản lý vai trò

Bảng 10:Mô tả chi tiết use case quản lý vai trò

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người quản trị quản lý các vai trò trong hệ thống |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản admin |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click icon đăng nhập.  2. Người quản trị đăng nhập bằng tài khoản có quyền là admin.  3. Chọn mục quản lý vai trò, admin có thể tạo mới, chỉnh sửa, xóa, tìm kiếm vai trò.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case quản lý quảng cáo

Bảng 11:Mô tả chi tiết use case quản lý quảng cáo

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người quản trị quản lý các vai trò trong hệ thống |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản admin |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click icon đăng nhập.  2. Người quản trị đăng nhập bằng tài khoản có quyền là admin.  3. Chọn mục quản lý quảng cáo, admin có thể tạo mới, chỉnh sửa, xóa, tìm kiếm quảng cáo.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case quản lý hồ sơ cá nhân

Bảng 12:Mô tả chi tiết use case quản lý hồ sơ cá nhân

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người quản trị quản lý thông tin của tài khoản cá nhân |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản admin |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click icon đăng nhập.  2. Người quản trị đăng nhập bằng tài khoản có quyền là admin.  3. Người quản trị click tên hoặc ảnh đại diện và click nút “Hồ sơ”.  3. Admin có thể xem chi tiết, chỉnh sửa thông tin cá nhân.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case xem thống kê

Bảng 13:Mô tả chi tiết use case xem thống kê

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép người quản trị quản lý thông tin của tài khoản cá nhân |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản admin |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click icon đăng nhập.  2. Người quản trị đăng nhập bằng tài khoản có quyền là admin.  3. Người quản trị click tên hoặc ảnh đại diện và click nút “Hồ sơ”.  3. Admin có thể xem chi tiết, chỉnh sửa thông tin cá nhân.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case gửi phê duyệt quảng cáo

Bảng 14:Mô tả chi tiết use case gửi phê duyệt quảng cáo

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép nhà quảng cáo gửi yêu cầu phê duyệt quảng cáo lên người quản trị |
| Actor | Nhà quảng cáo |
| Tiền điều kiện | Đăng nhập bằng tài khoản nhà quảng cáo |
| Luồng cơ bản | 1. Use case này bắt đầu khi nhà quảng cáo click button gửi phê duyệt.  2. một modal hiển thị lên, nhà quảng cáo click button “OK”.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case phê duyệt quảng cáo

Bảng 15:Mô tả chi tiết use case phê duyệt quảng cáo

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép Người quản trị phê duyệt quảng cáo và hiển thị quảng cáo lên trang chủ |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản người quản trị |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click button phê duyệt.  2. Quảng cáo sẽ hiển thị bên ngoài trang chủ theo vị trí đã có trong thông tin.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

#### Use case quản lý doanh thu

Bảng 16:Mô tả chi tiết use case quản lý doanh thu

|  |  |
| --- | --- |
| Mô tả | Use case này cho phép Người quản trị quản lý doanh thu quảng cáo và xuất file excel doanh thu quảng cáo |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản người quản trị |
| Luồng cơ bản | 1. Use case này bắt đầu khi người quản trị click trang phê duyệt.  2. Admin có thể xem doanh thu, xuất file excel doanh thu quảng cáo.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

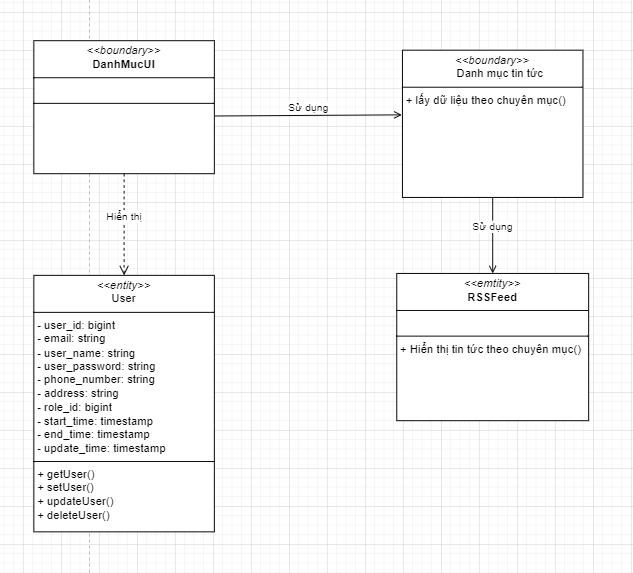
#### Use case thanh toán

Bảng 17:Mô tả chi tiết use case thanh toán

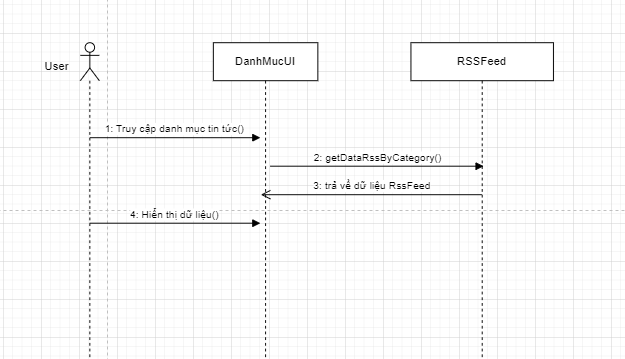
|  |  |
| --- | --- |
| Mô tả | Use case này cho phép Nhà quảng cáo thanh toán doanh thu quảng cáo |
| Actor | Admin |
| Tiền điều kiện | Đăng nhập bằng tài khoản có vai trò nhà quảng cáo |
| Luồng cơ bản | 1. Use case này bắt đầu khi nhà quảng cáo click nút thanh toán ở trang quản lý quảng cáo.  2. Nhà quảng cáo có thể chọn các phương thức thanh toán có sẵn.  3. Sau khi thanh toán sẽ có thông báo thành công và quay trở lại trang quản lý.  Use case kết thúc. |
| Luồng rẽ nhánh | 1. Không kết nối được dữ liệu cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |

### Biểu đồ thực hiện hóa các usecase

#### Use case xem danh sách tin tức

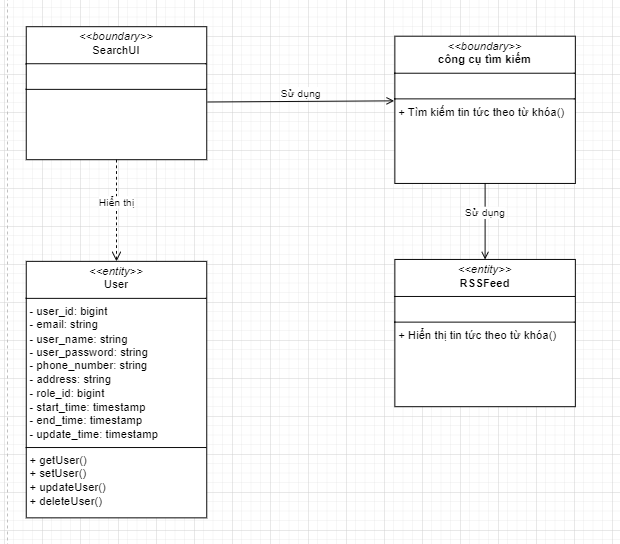
**Biểu đồ lớp**

Hình 2.6:Biểu đồ lớp use case xem danh sách tin tức

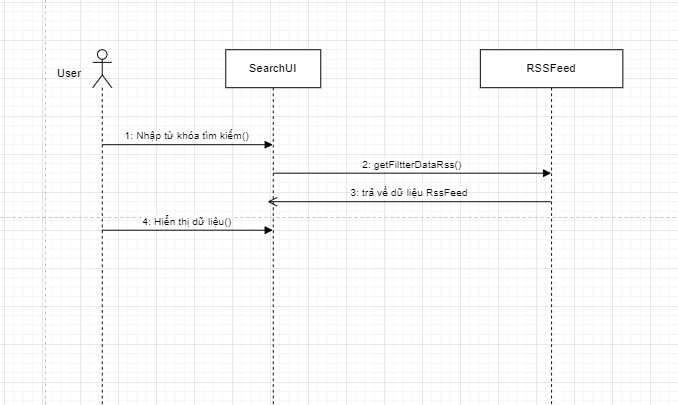
**Biểu đồ trình tự  
**

Hình 2.7:Biểu đồ trình tự use case xem danh sách tin tức

#### Use case tìm kiếm tin tức

**Biểu đồ lớp**

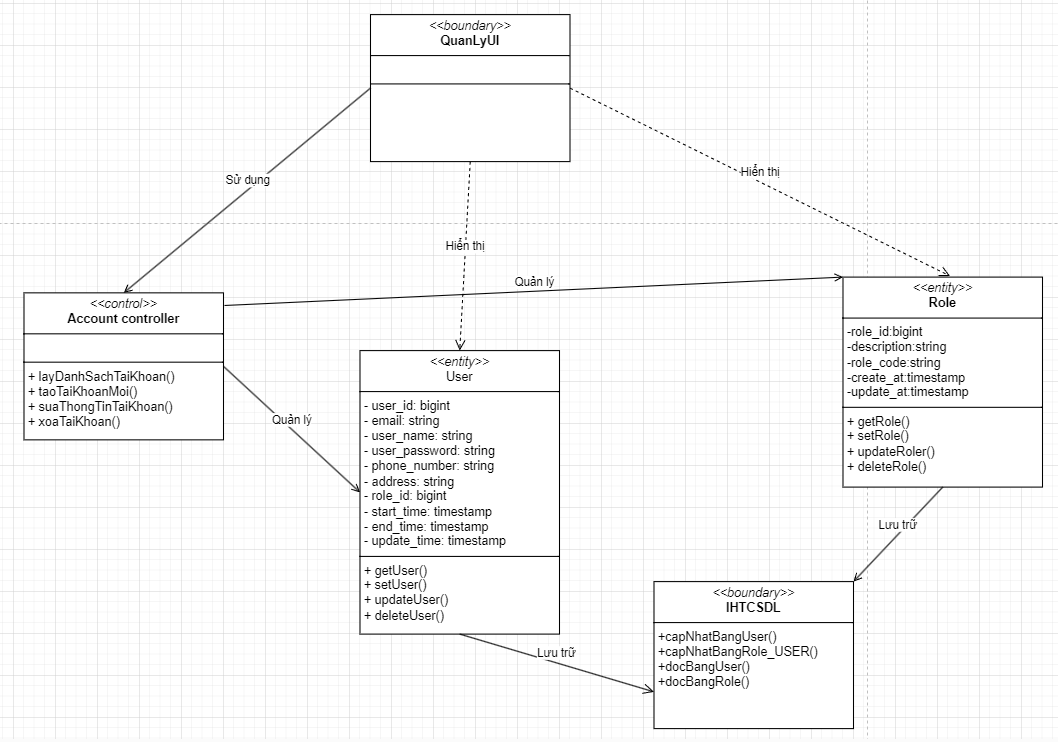
Hình 2.8:Biểu đồ lớp use case tìm kiếm tin tức

**Biểu đồ trình tự**  


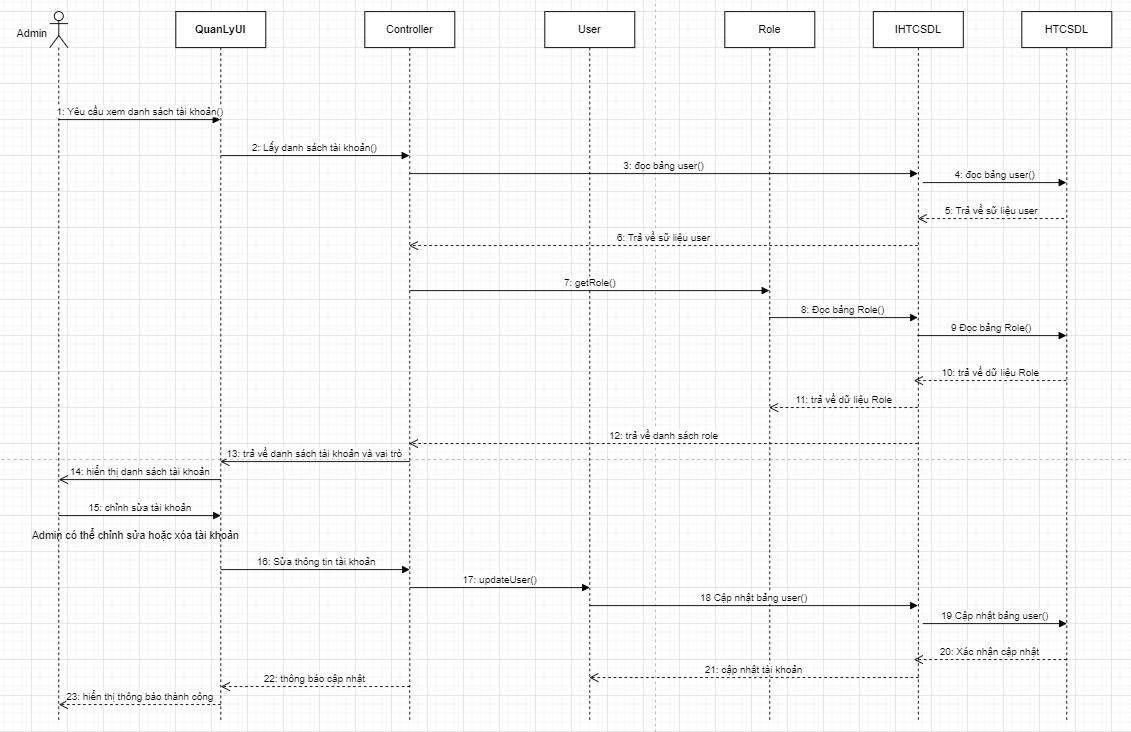
Hình 2.9:Biểu đồ trình tự use case tìm kiếm tin tức

#### Use case quản lý tài khoản

**Biểu đồ lớp**

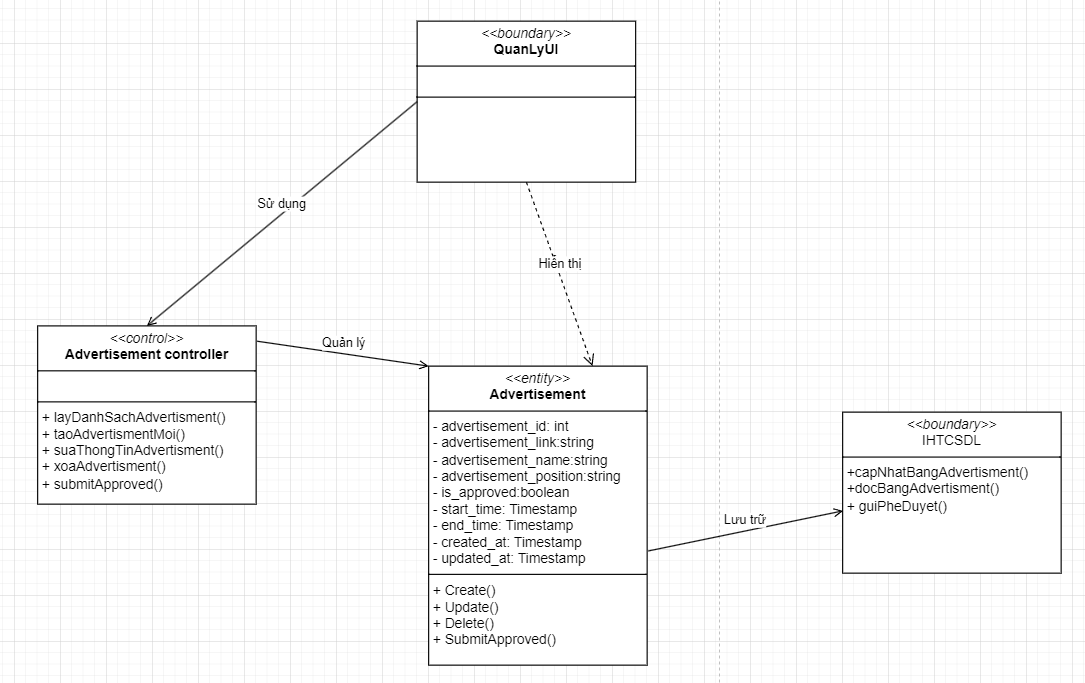


Hình 2.10:Biểu đồ lớp use case quản lý tài khoản

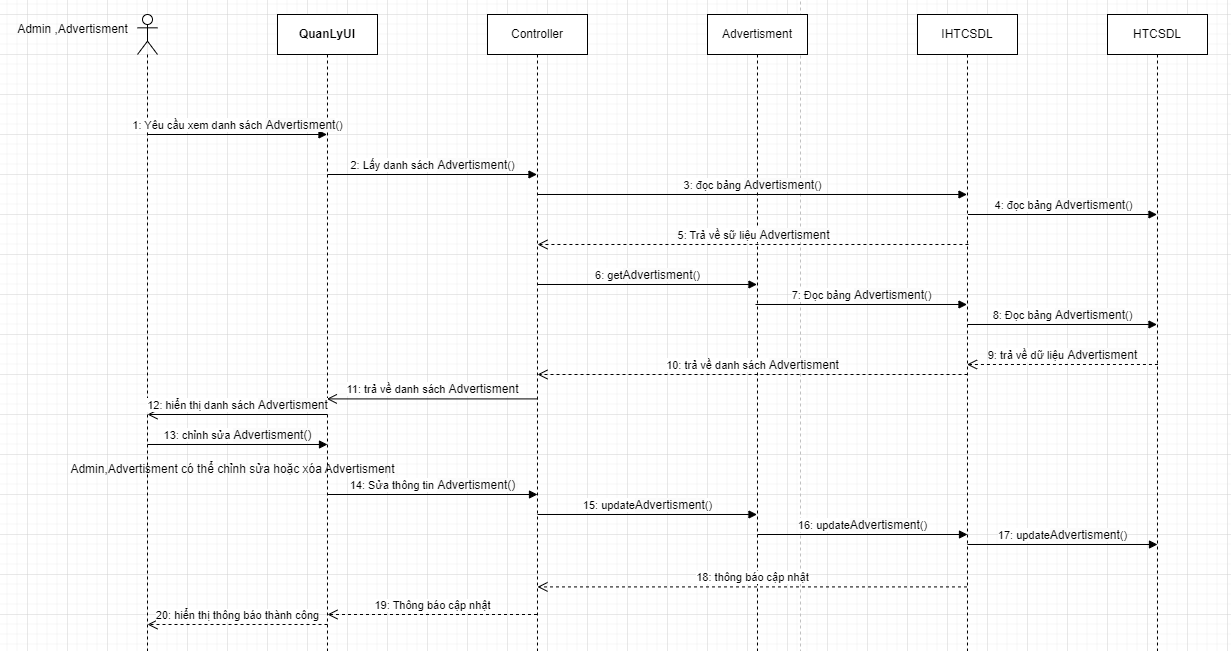
**Biểu đồ trình tự**  


Hình 2.11:Biểu đồ trình tự use case quản lý tài khoản

#### Use case quản lý quảng cáo

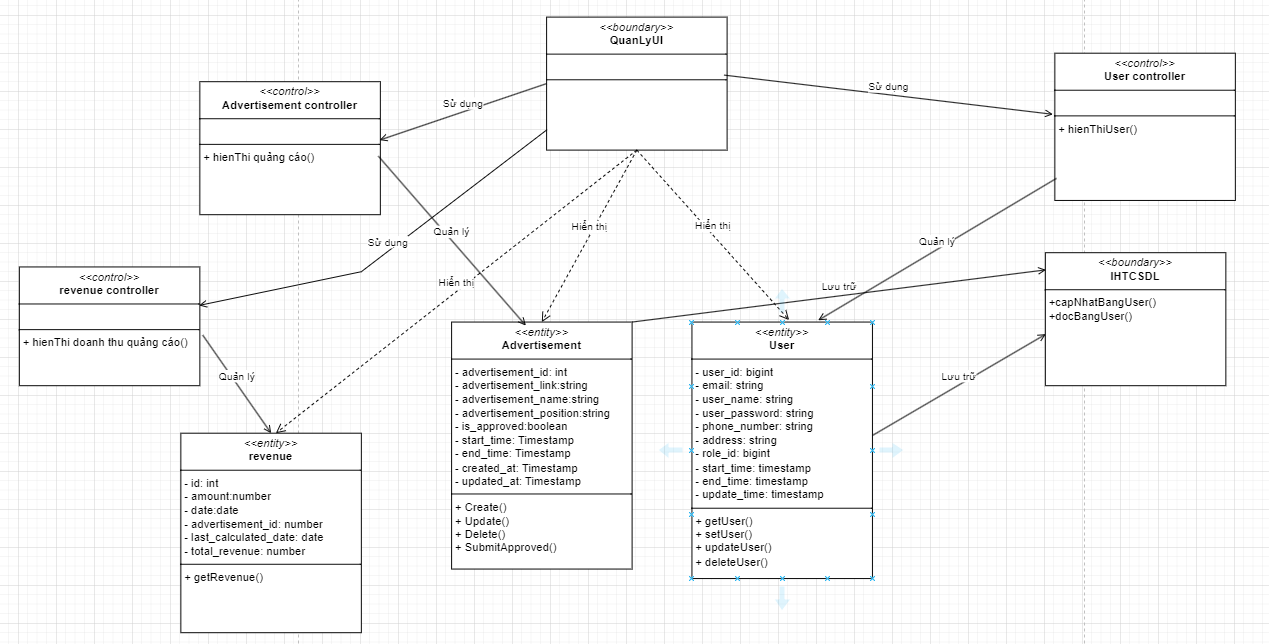
**Biểu đồ lớp**

Hình 2.12:Biểu đồ lớp use case quản lý quảng cáo

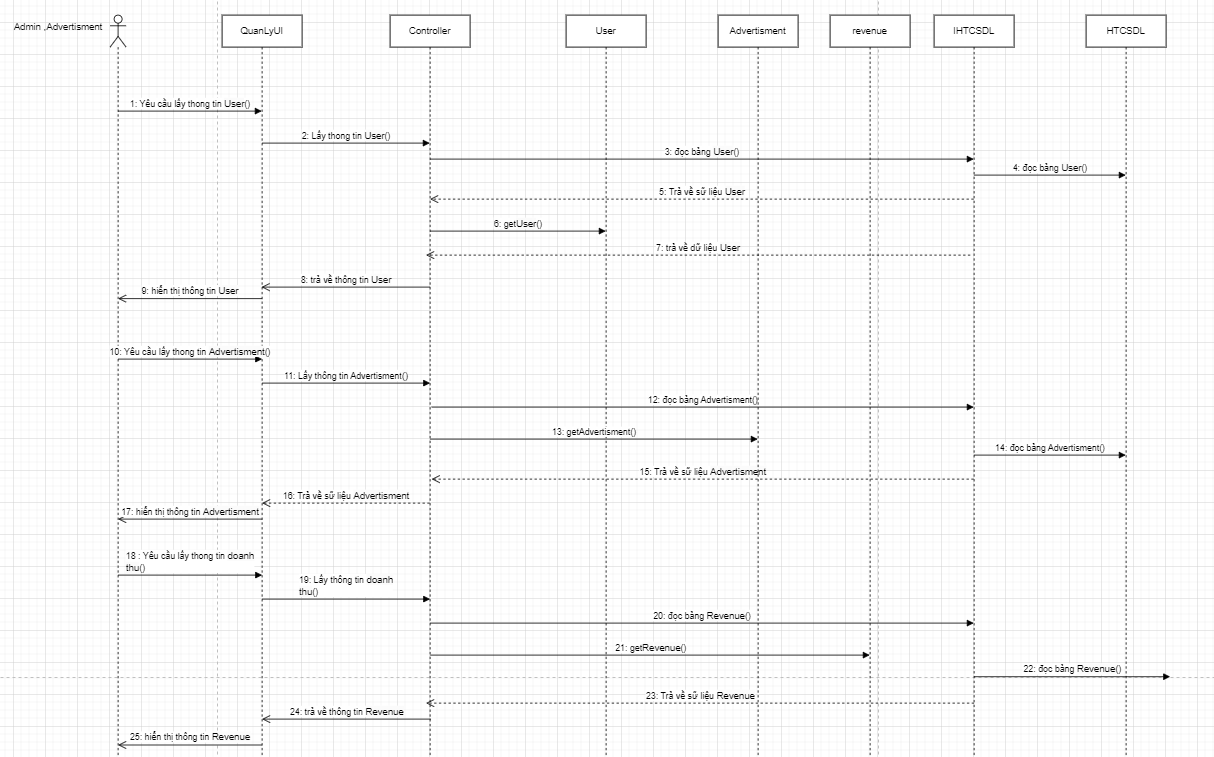
**Biểu đồ trình tự**

Hình 2.13:Biểu đồ trình tự use case quản lý quảng cáo

#### Use case xem thống kê

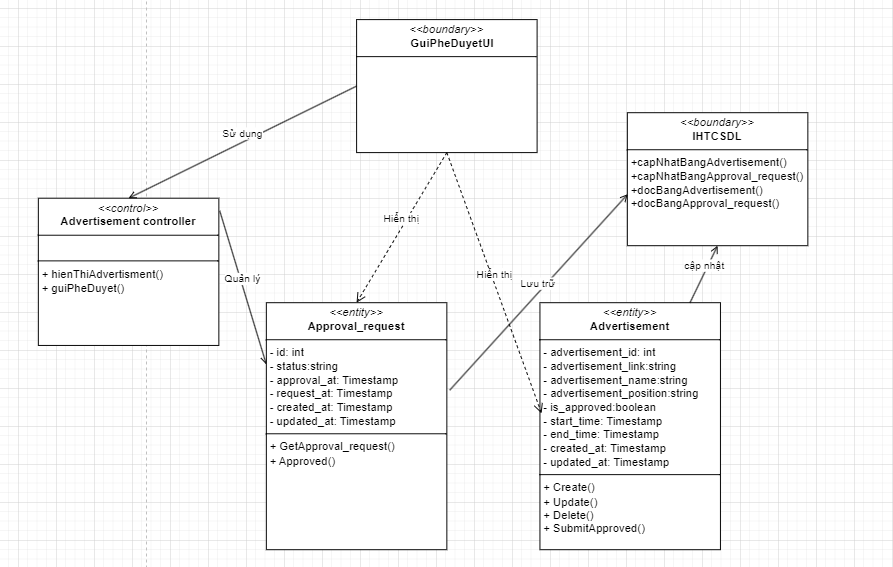
**Biểu đồ lớp**

Hình 2.14:Biểu đồ lớp use case xem thống kê

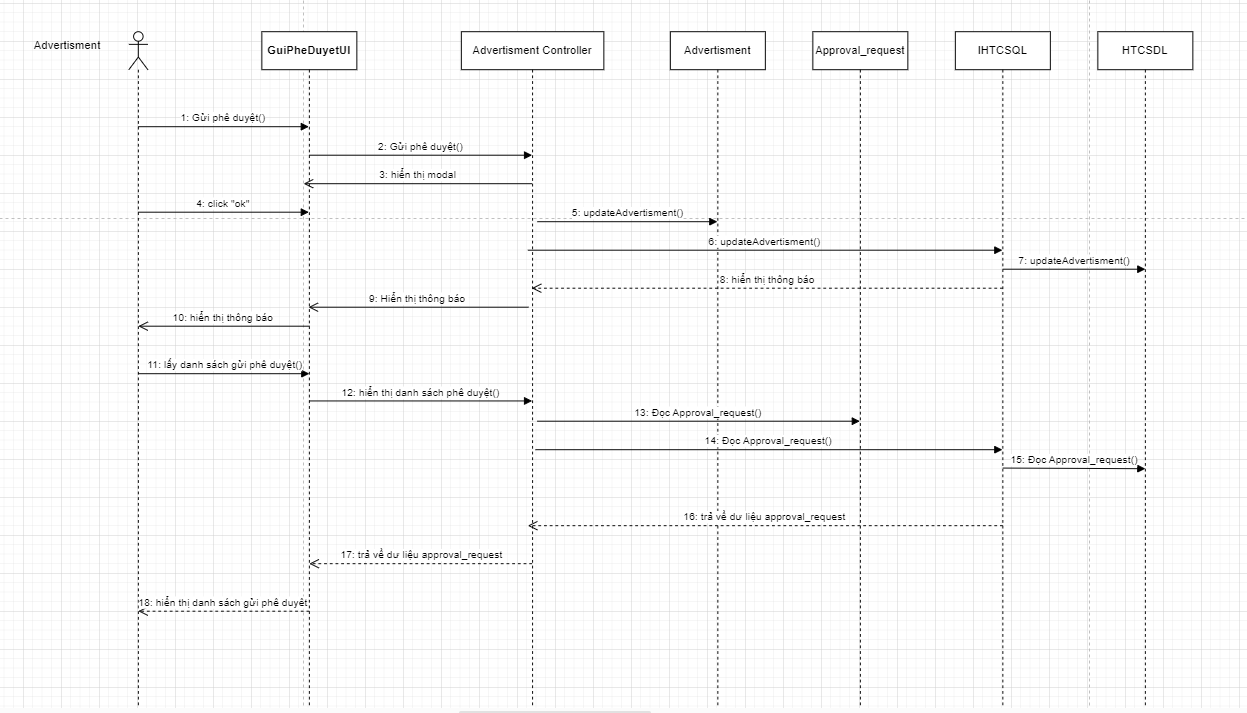
**Biểu đồ trình tự**

Hình 2.15:Biểu đồ trình tự use case xem thống kê

#### **Use case gửi phê duyệt quảng cáo**

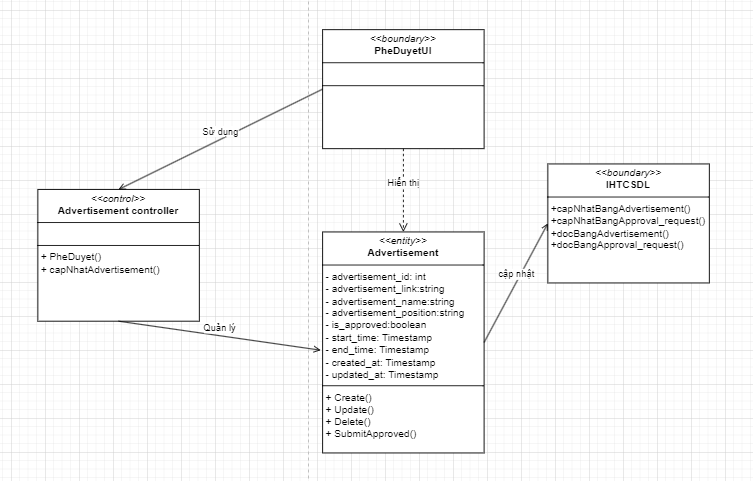
**Biểu đồ lớp  
**

Hình 2.16:Biểu đồ lớp use case gửi phê duyệt quảng cáo

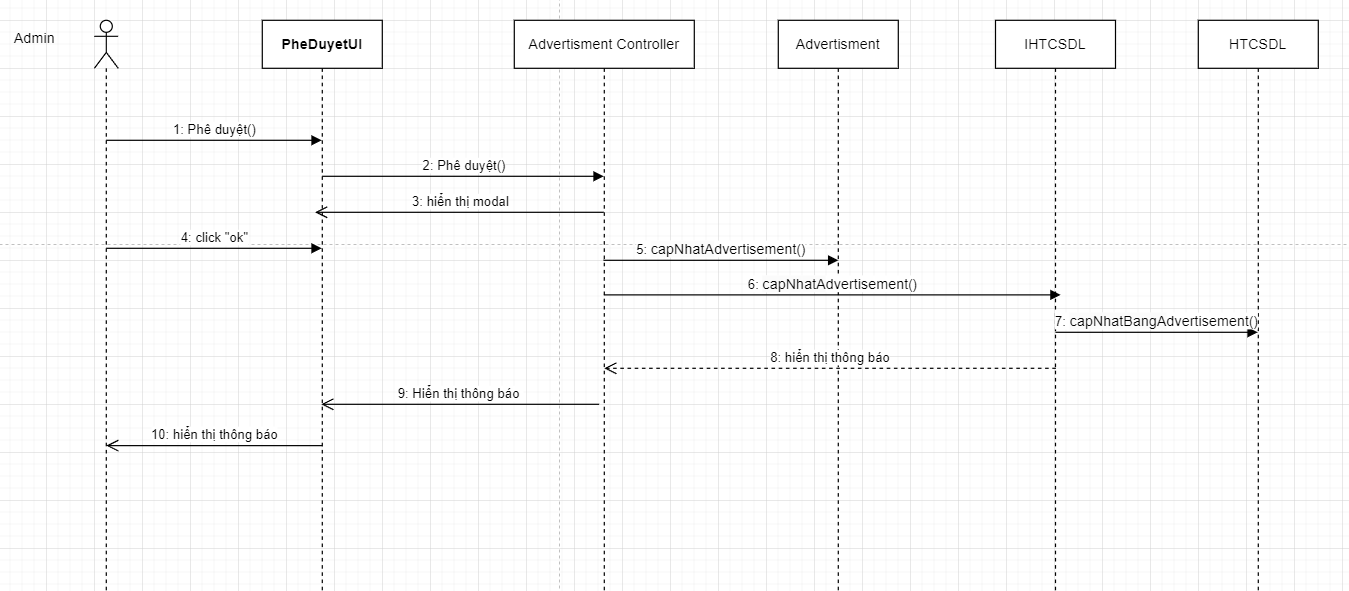
**Biểu đồ trình tự**

Hình 2.17:Biểu đồ trình tự use case gửi phê duyệt quảng cáo

#### Use case phê duyệt quảng cáo

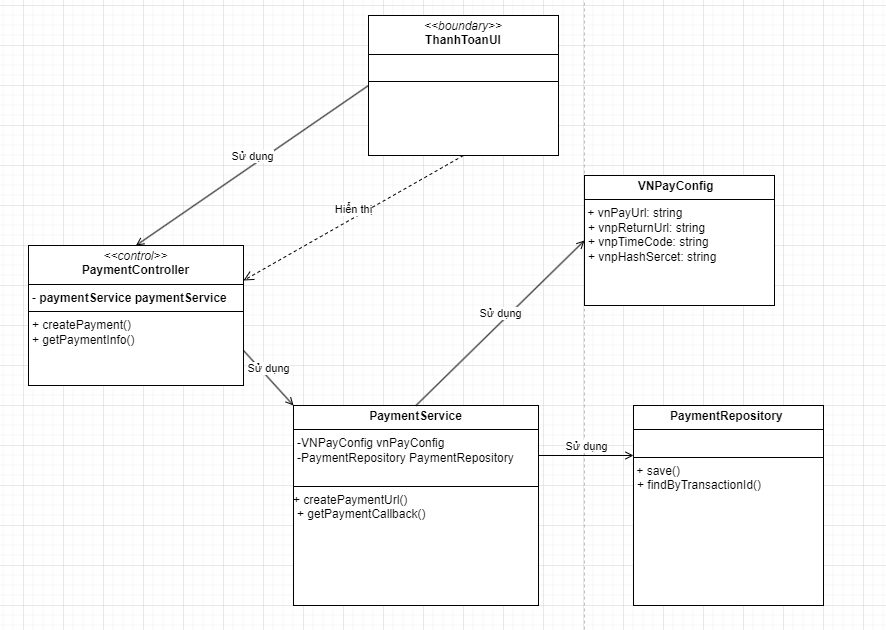
**Biểu đồ lớp**

Hình 2.18:Biểu đồ lớp use case phê duyệt quảng cáo

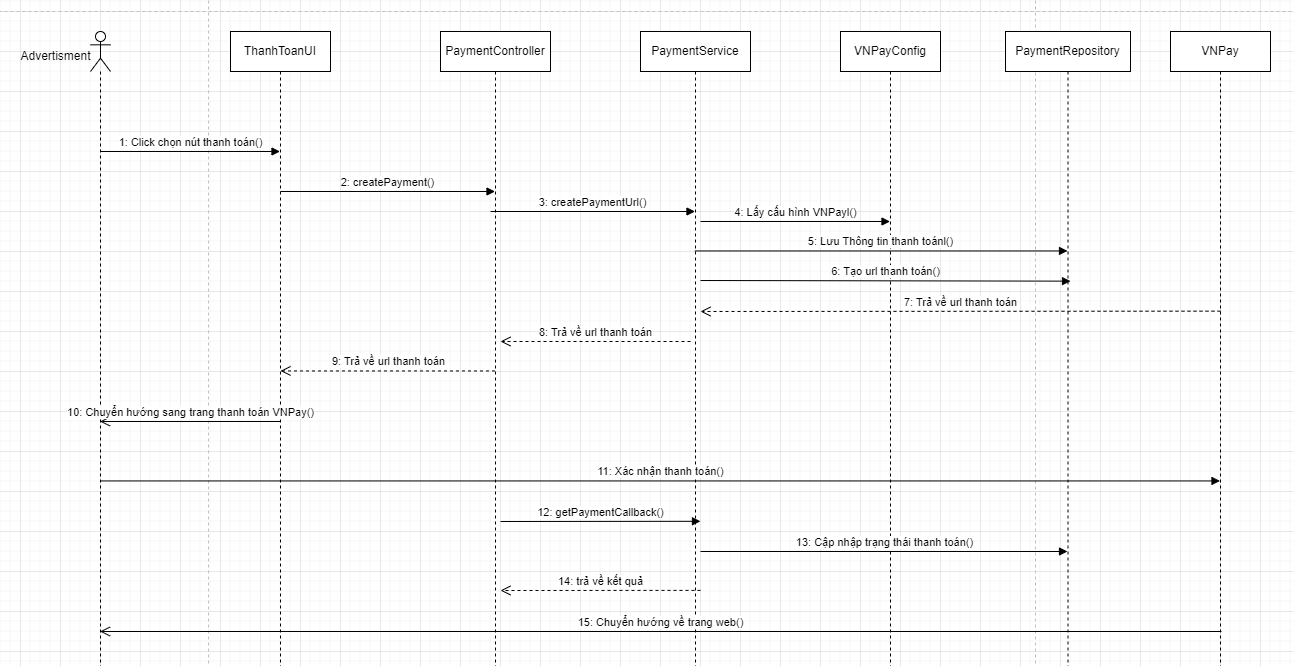
**Biểu đồ trình tự**

Hình 2.19:Biểu đồ trình tự use case phê duyệt quảng cáo

#### Use case thanh toán

**Biểu đồ lớp**

Hình 2.20:Biểu đồ lớp use case thanh toán

**Biểu đồ trình tự**  


Hình 2.21:Biểu đồ trình tự use case thanh toán

### Mô hình hóa dữ liệu

#### Biểu đồ thực thể liên kết

Hình 2.22:Biểu đồ thực thể liên hết

#### Mô tả cấu trúc cơ sở dữ liệu

Bảng 18:Bảng users

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| user\_id | bigint (IDENTITY) | Khóa chính, tự động tăng |
| end\_time | timestamp without time zone(6) | Thời gian kết thúc |
| start\_time | timestamp without time zone(6) | Thời gian bắt đầu |
| update\_time | timestamp without time zone(6) | Thời gian cập nhật |
| address | character varying(255) | Địa chỉ người dùng |
| email | character varying(255) | Email người dùng |
| phone\_number | character varying(255) | Số điện thoại |
| user\_name | character varying(255) | Tên người dùng |
| user\_password | character varying(255) | Mật khẩu người dùng |
| role\_id | bigint | Khóa ngoại liên kết với bảng role |
| updated\_at | timestamp without time zone(6) | Thời gian cập nhật gần nhất |
| created\_at | timestamp without time zone(6) | Thời gian tạo |

Bảng 19:Bảng role

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| role\_id | bigint (IDENTITY) | Khóa chính, tự động tăng |
| description | character varying(255) | Mô tả vai trò |
| role\_code | character varying(255) | Mã vai trò |
| updated\_at | timestamp without time zone(6) | Thời gian cập nhật gần nhất |
| created\_at | timestamp without time zone(6) | Thời gian tạo |

Bảng 20:Bảng advertisement

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| advertisement\_id | bigint (IDENTITY) | Khóa chính, tự động tăng |
| end\_time | timestamp without time zone(6) | Thời gian kết thúc quảng cáo |
| start\_time | timestamp without time zone(6) | Thời gian bắt đầu quảng cáo |
| update\_time | timestamp without time zone(6) | Thời gian cập nhật quảng cáo |
| advertisement\_link | character varying(255) | Liên kết quảng cáo |
| advertisement\_name | character varying(255) | Tên quảng cáo |
| advertisement\_position | character varying(255) | Vị trí quảng cáo |
| user\_id | bigint | Khóa ngoại liên kết với bảng users |
| updated\_at | timestamp without time zone(6) | Thời gian cập nhật gần nhất |
| created\_at | timestamp without time zone(6) | Thời gian tạo |
| end\_date | date | Ngày kết thúc |
| price | numeric(38,2) | Giá quảng cáo |
| start\_date | date | Ngày bắt đầu |
| status | character varying(255) | Trạng thái quảng cáo |
| advertisement\_field\_id | bigint | Khóa ngoại liên kết với bảng advertising\_field |

Bảng 21:Bảng advertising\_field

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| advertising\_field\_id | bigint (IDENTITY) | Khóa chính, tự động tăng |
| advertising\_field\_name | character varying(255) | Tên lĩnh vực quảng cáo |
| advertisement\_id | bigint | Khóa ngoại liên kết với bảng advertisement |

Bảng 22:Bảng revenue

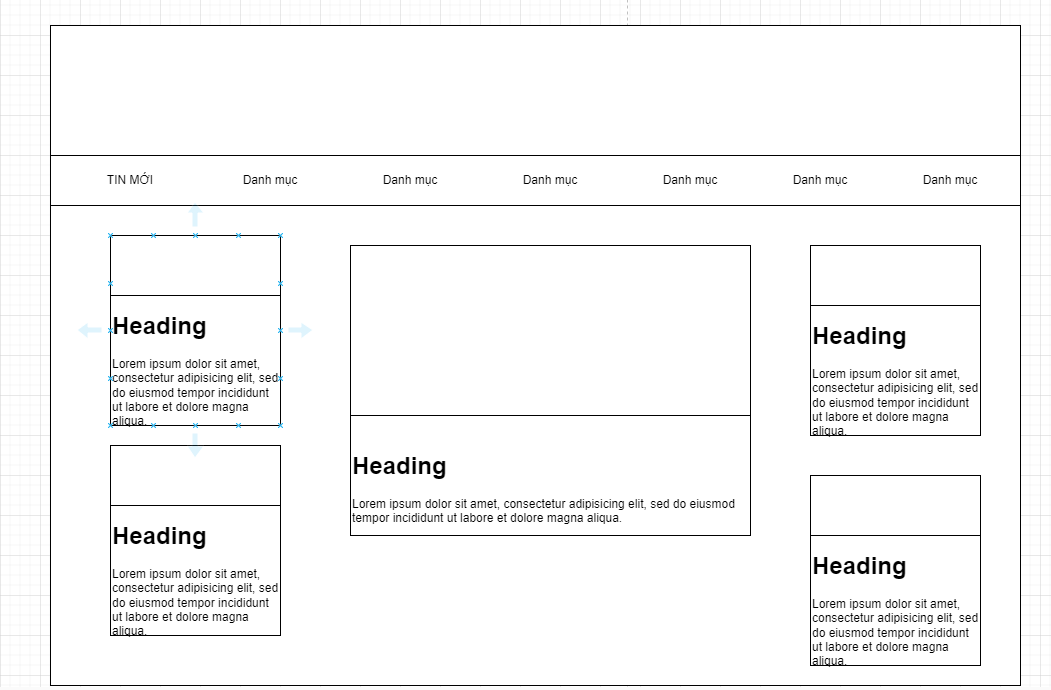
|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | bigint | Khóa chính, tự động tăng |
| amount | numeric(38,2) | Số tiền doanh thu |
| date | date | Ngày ghi nhận doanh thu |
| advertisement\_id | bigint | Khóa ngoại liên kết với bảng advertisement |
| last\_calculated\_date | date | Ngày tính toán gần nhất |
| total\_revenue | numeric(38,2) | Tổng doanh thu |

Bảng 23:Bảng approval\_request

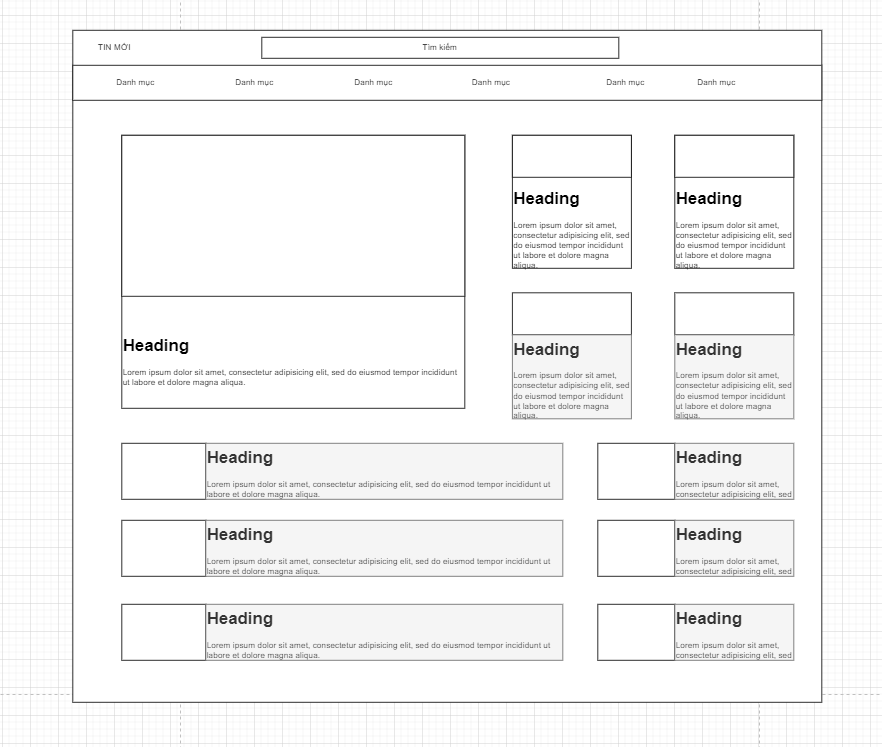
|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | bigint (IDENTITY) | Khóa chính, tự động tăng |
| created\_at | timestamp without time zone(6) | Thời gian tạo yêu cầu |
| updated\_at | timestamp without time zone(6) | Thời gian cập nhật yêu cầu |
| approved\_at | timestamp without time zone(6) | Thời gian phê duyệt |
| requested\_at | timestamp without time zone(6) | Thời gian yêu cầu |
| status | character varying(255) | Trạng thái yêu cầu |
| advertisement\_id | bigint | Khóa ngoại liên kết với bảng advertisement |

## Thiết kế giao diện

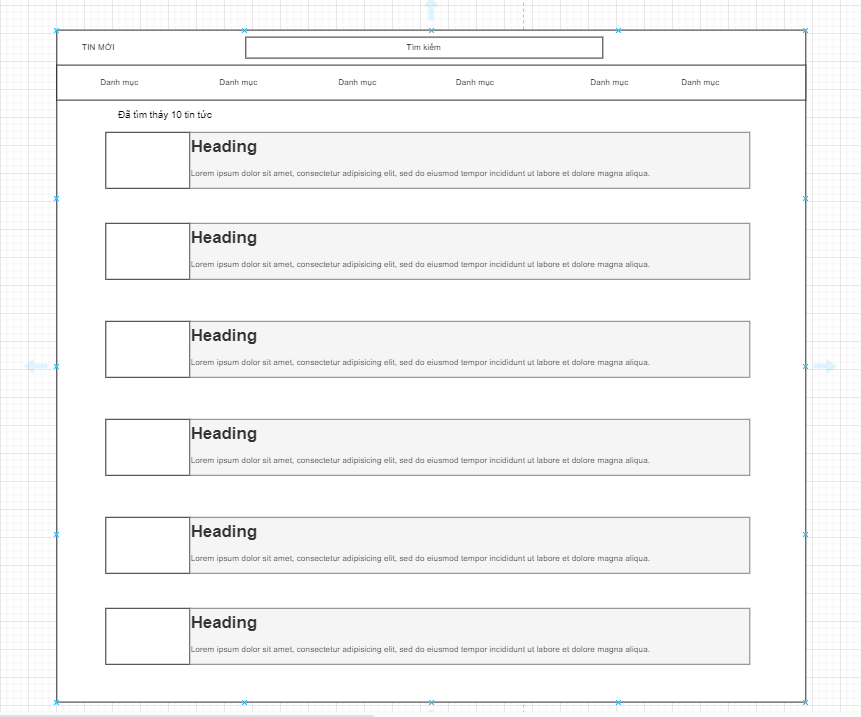
### Giao diện người dùng

Mô tả một số giao diện bên phía người dùng:  


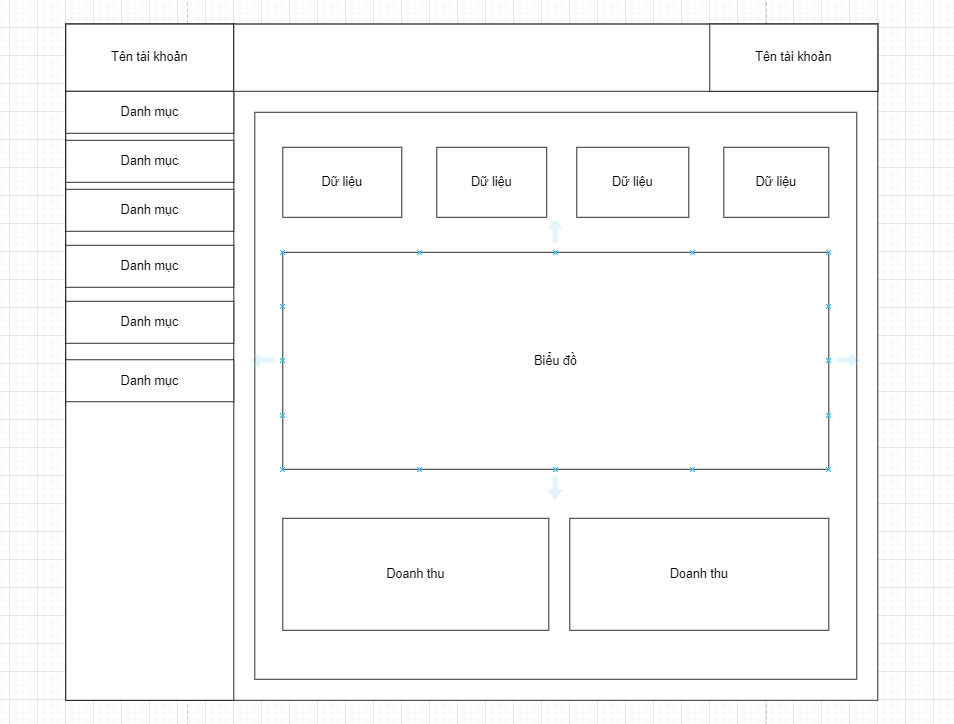
Hình 2.23:Hình dung giao diện Trang chủ



Hình 2.24:Hình dung giao diện danh mục tin tức



Hình 2.25:Hình dung giao diện Tìn kiếm tin tức



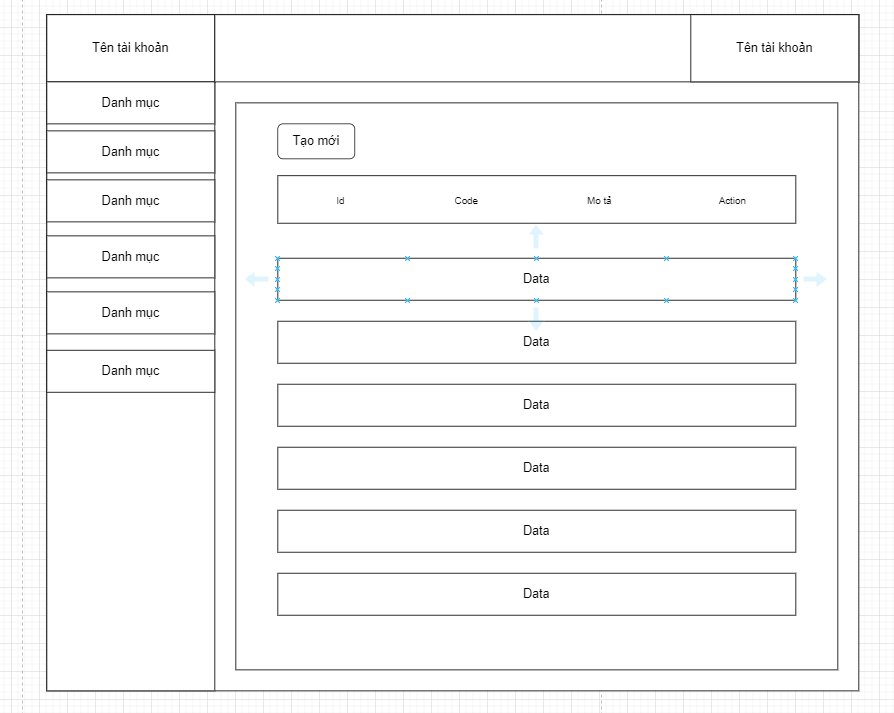
Hình 2.26:Hình dung giao diện trang tổng quan



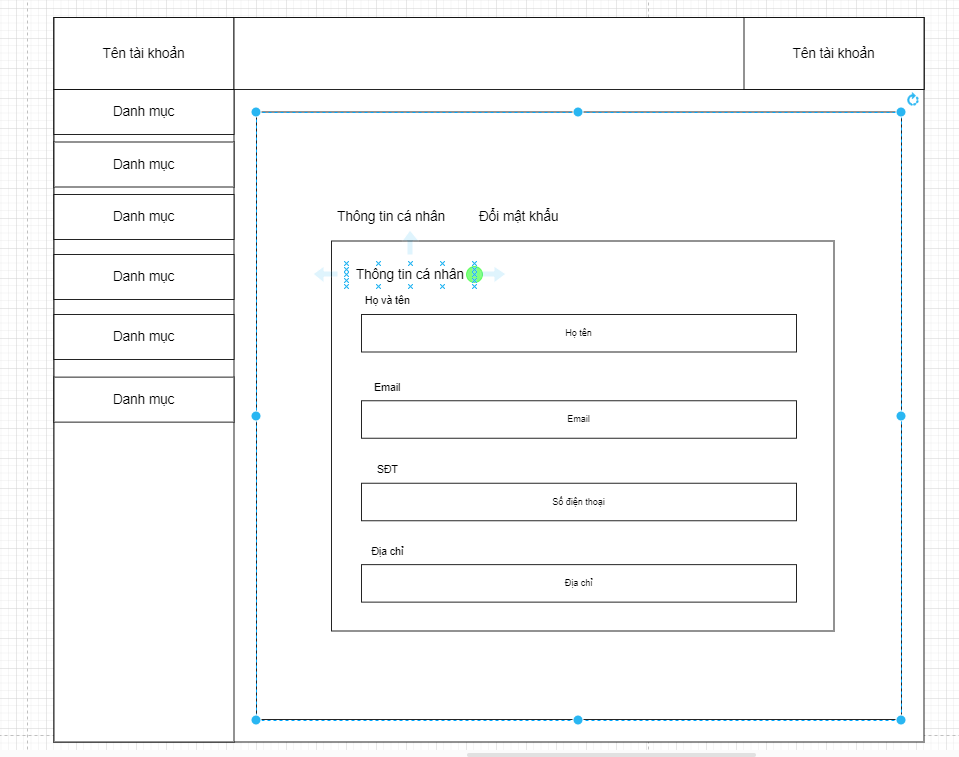
Hình 2.27:Hình dung giao diện quản lý tài khoản



Hình 2.28:Hình dung giao diện quản lý quảng cáo



Hình 2.29:Hình dung giao diện quản lý vai trò



Hình 2.30:Hình dung giao diện hồ sơ cá nhân

# THỬ NGHIỆM VÀ ĐÁNH GIÁ

## Cài đặt chương trình

### Môi trường cài đặt

Để hỗ trợ cho quá trình phát triển chương trình một cách thuận lợi việc cài đặt môi trường và các công cụ phát triển phần mềm là rất quan trọng.Dưới đây là một số cấu hình môi trường trong quá trình phát triển.

**- Front-end (Next.js):**

* + Node.js (phiên bản 18.x trở lên)
  + npm (Node Package Manager
  + Visual Studio Code
  + Git

**- Back-end (Spring Boot):**

* Java Development Kit (JDK) 17 trở lên
* Maven
* IntelliJ IDEA
* Git

**- Công cụ triển khai:**

* Docker (để đóng gói ứng dụng back-end)
* Vercel (để triển khai front-end)
* Render (để triển khai back-end)

**- Công cụ quản lý phiên bản:**

* Git
* Tài khoản GitHub hoặc GitLab
* Sourcetree

**- Công cụ kiểm thử:**

* Postman (để kiểm thử API)

**- Cơ sở dữ liệu:**

* PostgreSQL

### Hướng dẫn cài đặt chương trình

Cách cài đặt ứng dụng web thương mại điện tử kinh doanh đồ thể thao với Laravel:

#### Bên phía máy khách

- Bước 1: Truy cập vào đường dẫn git dưới đây để clone mã nguồn

chương trình về máy:

<https://github.com/dinhkhactuan/Project-Graduation>

- Bước 2: Sau khi clone mã nguồn chương về máy tiến hành mở mã nguồn chương trình lên bằng Visual Studio Code hoặc các phần mềm khác.

- Bước 3: Mở Terminal và di chuyển vào thư mục chứa toàn bộ mã nguồn bằng câu lệnh sau:

cd YouDirectoryName

- Bước 4: Cài đặt các phụ thuộc của chương trình bằng lệnh sau:  
 npm install or npm i

- Bước 5: Chạy chương trình bằng lệnh sau:  
 npm run dev or npm run build

- Bước 6: Bây giờ chương trình đã chạy trên máy bạn với url là:  
 localhost:3000

#### Bên phía máy chủ

- Bước 1: Truy cập vào đường dẫn git dưới đây để clone mã nguồn

chương trình về máy:

<https://github.com/dinhkhactuan/Graduation_Be>

- Bước 2: Sau khi clone mã nguồn chương về máy tiến hành mở mã nguồn chương trình lên bằng intellij hoặc các phần mềm khác.

- Bước 3: Mở Terminal và di chuyển vào thư mục chứa toàn bộ mã nguồn bằng câu lệnh sau:

cd YouDirectoryName

- Bước 4: Cài đặt các phụ thuộc của chương trình bằng lệnh sau:  
 mvn dependency:resolve

- Bước 5: Để xác nhận rằng tất cả đã được cài đặt chính xác, chạy:

mvn verify

- Bước 6: Cấu hình cơ sở dữ liệu:

Mở file src/main/resources/application.properties (hoặc application.yml)

Cập nhật thông tin kết nối cơ sở dữ liệu:|

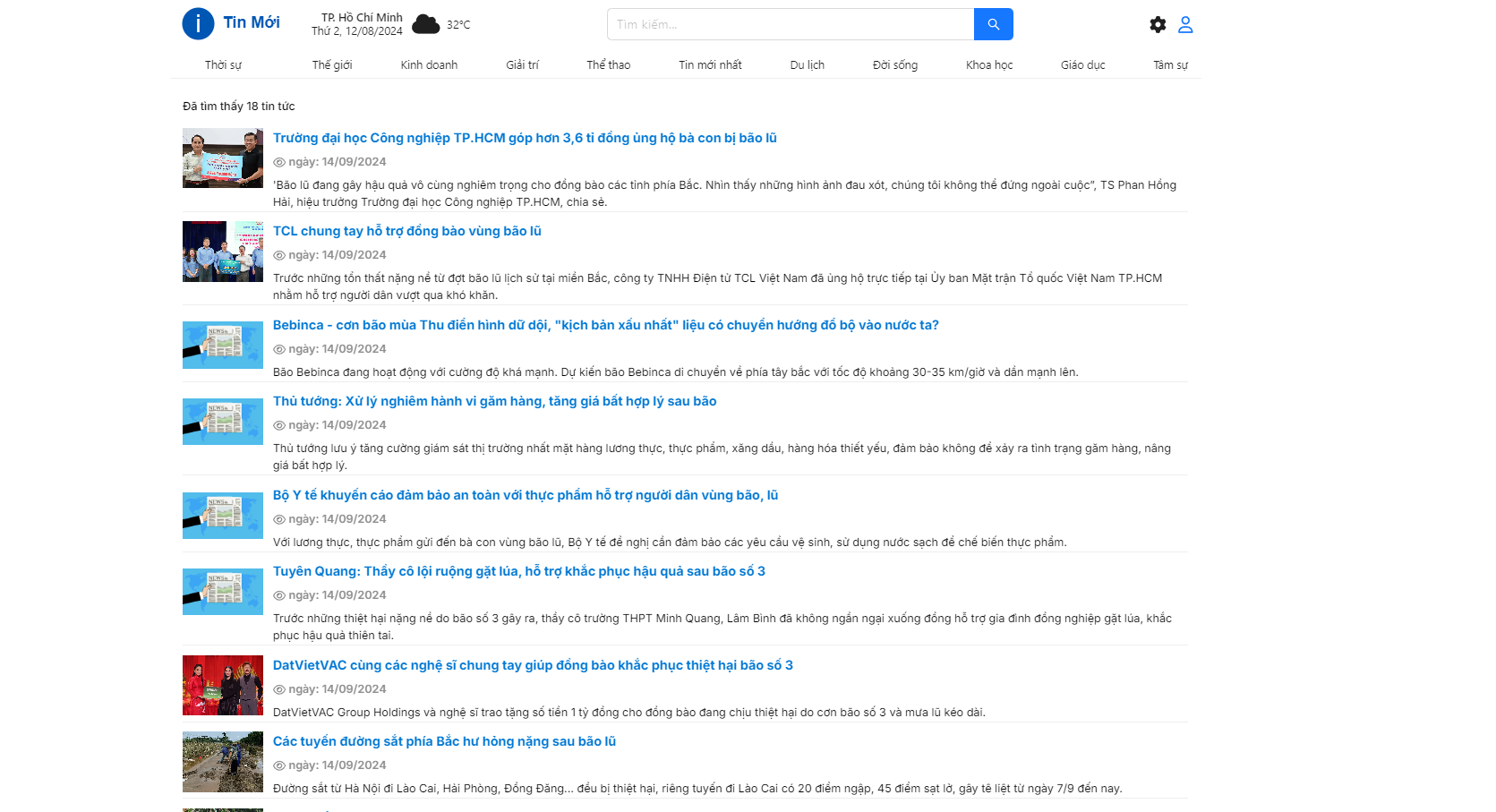
- Bước 7: Để chạy ứng dụng, sử dụng lệnh:  
 mvn spring-boot:run  
- Bước 7: Bây giờ chương trình đã chạy trên máy bạn với url là:  
 localhost:8080

## Kết quả đạt được

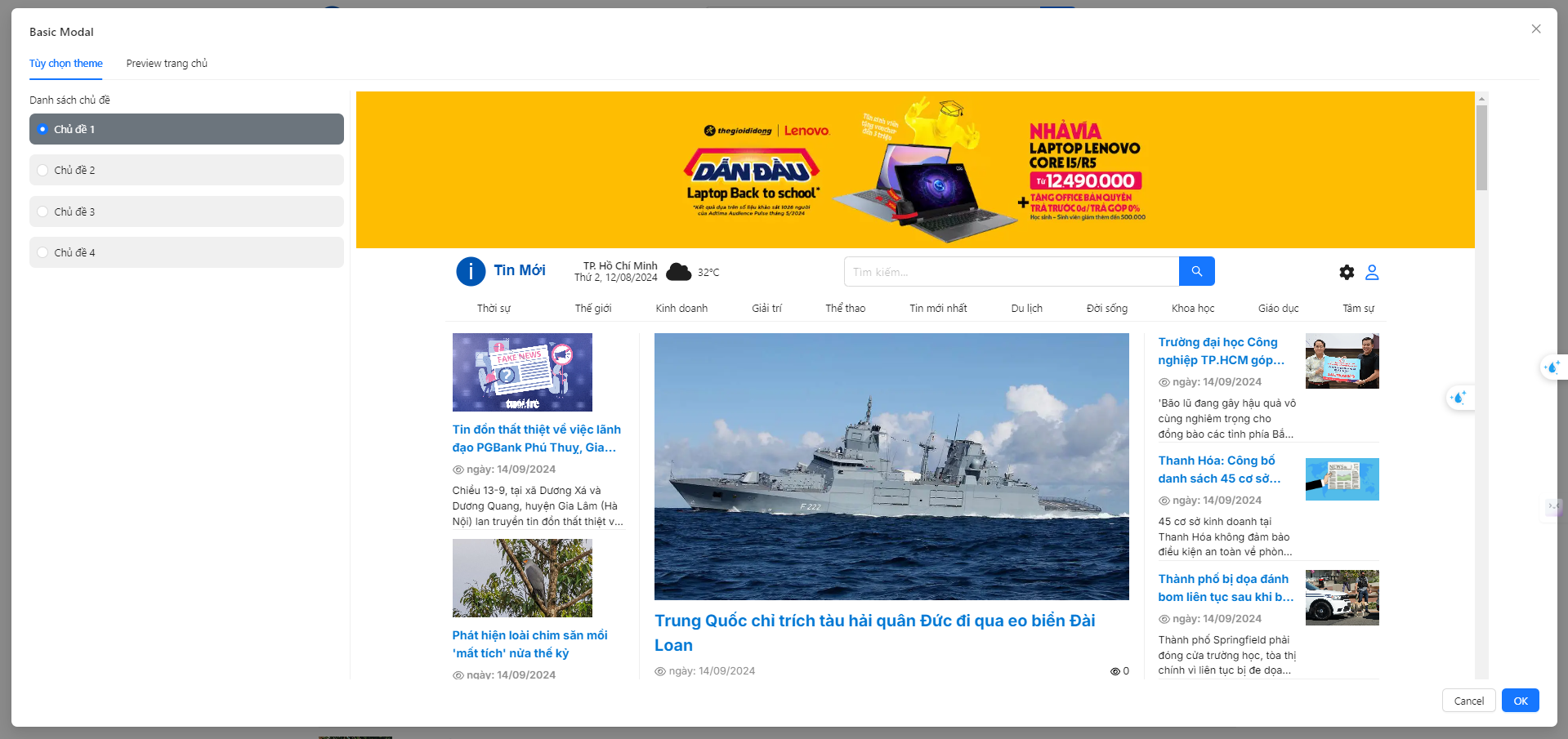
### Kết quả đạt được phía người dùng

Hình 3.1:Giao diện trang chủ



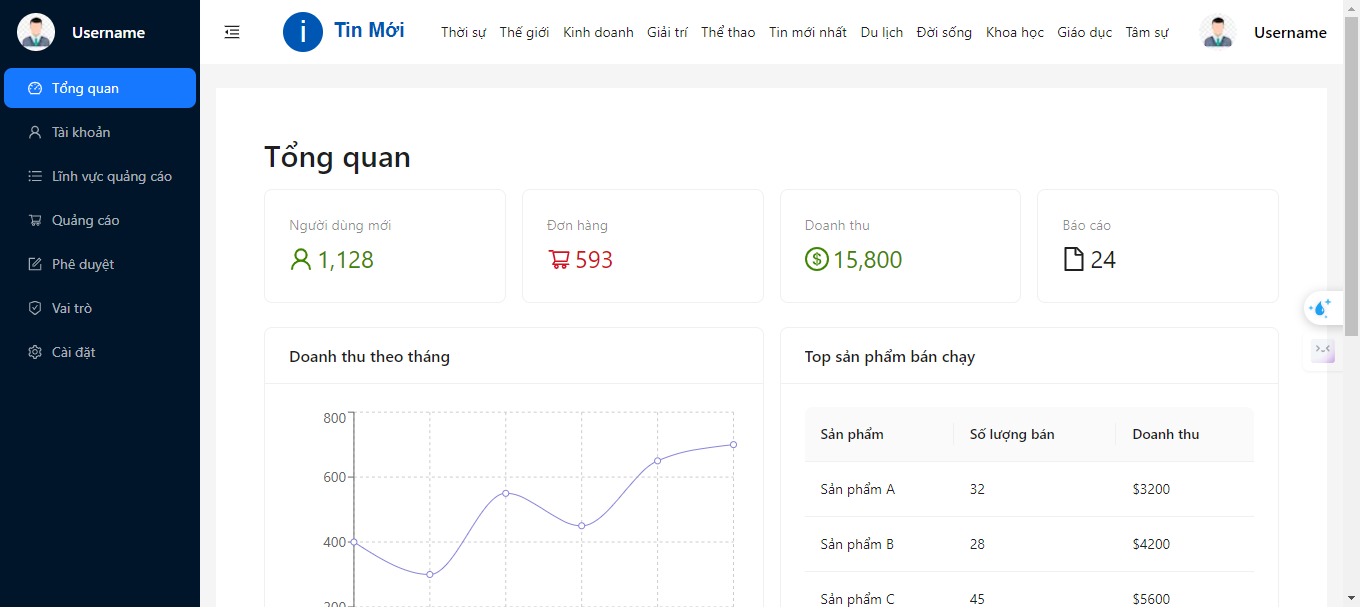
Hình 3.2:Giao diện danh mục tin tức  


Hình 3.3:Giao diện trang tim kiếm tin tức

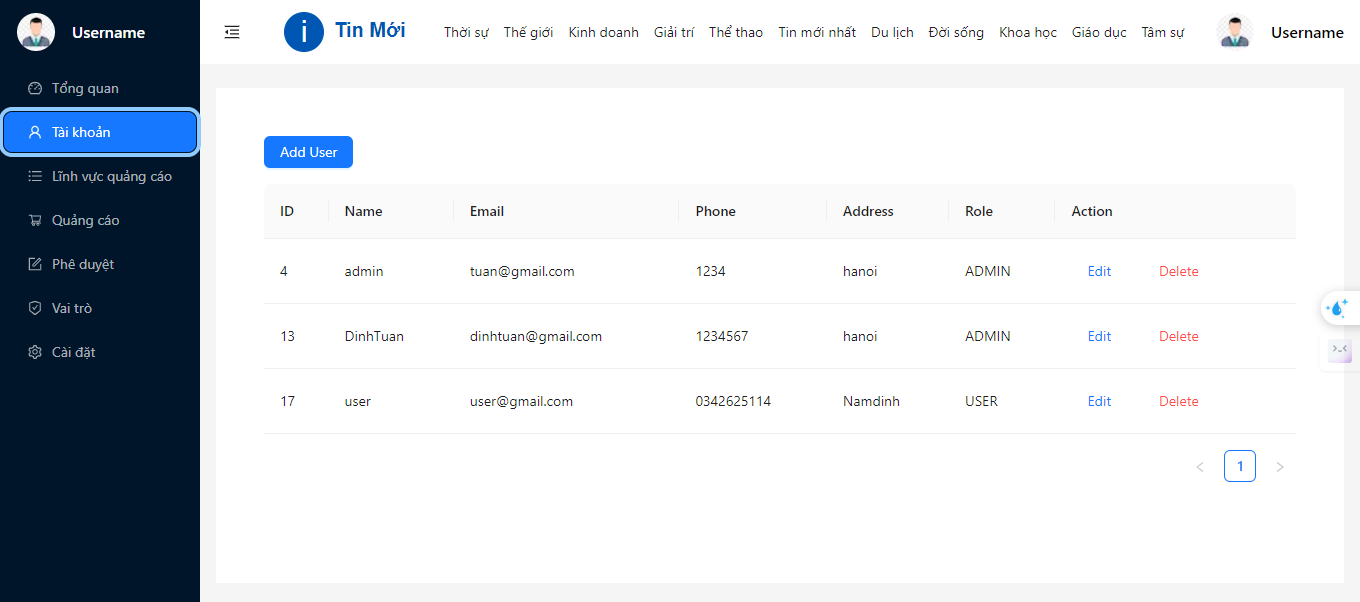


Hình 3.4:Giao diện tùy chọn chủ đề

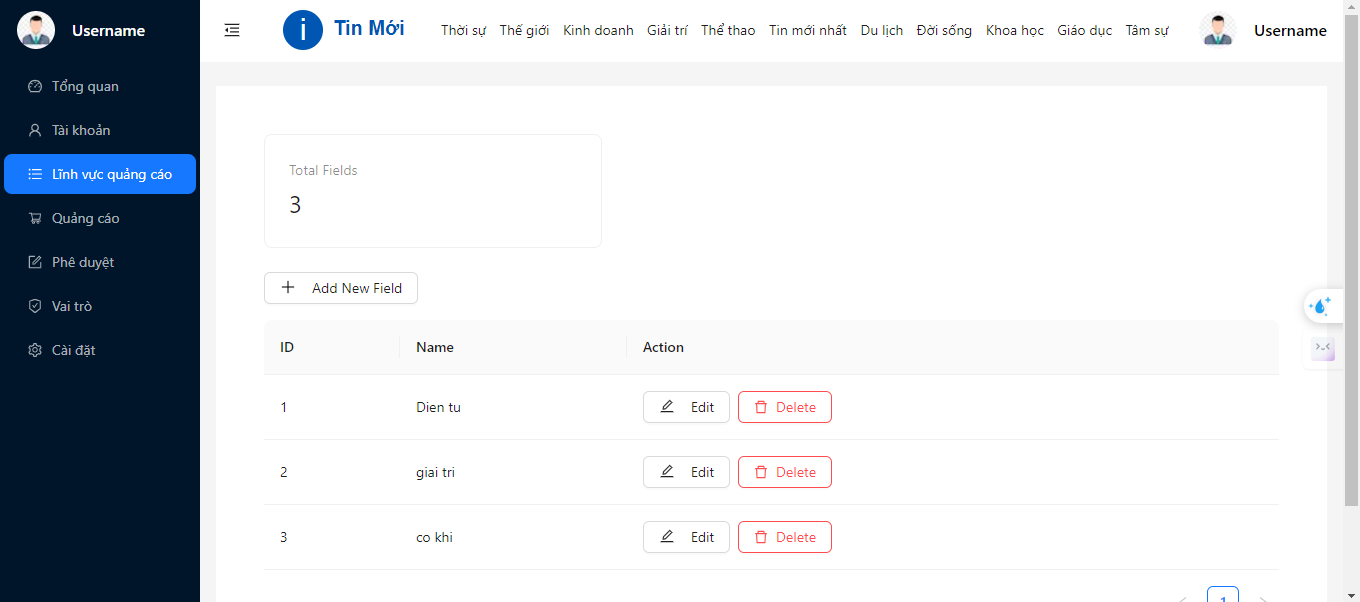
### Kết quả đạt được phía quản trị viên



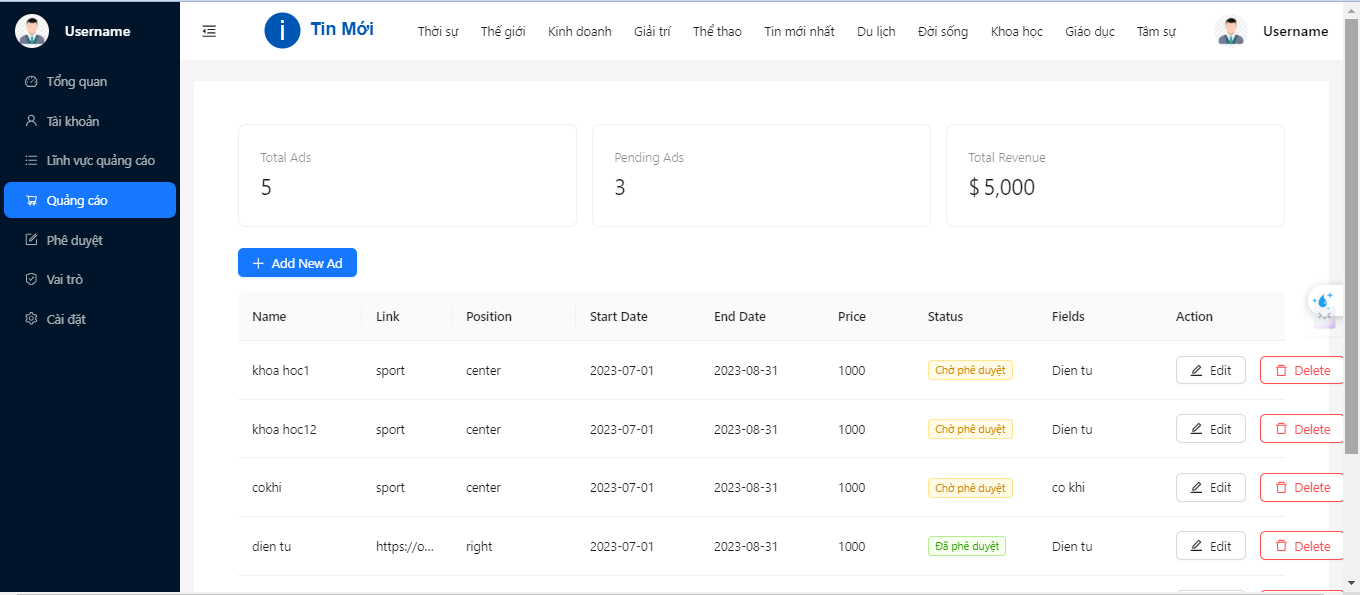
Hình 3.5:Giao diện trang tổng quan admin



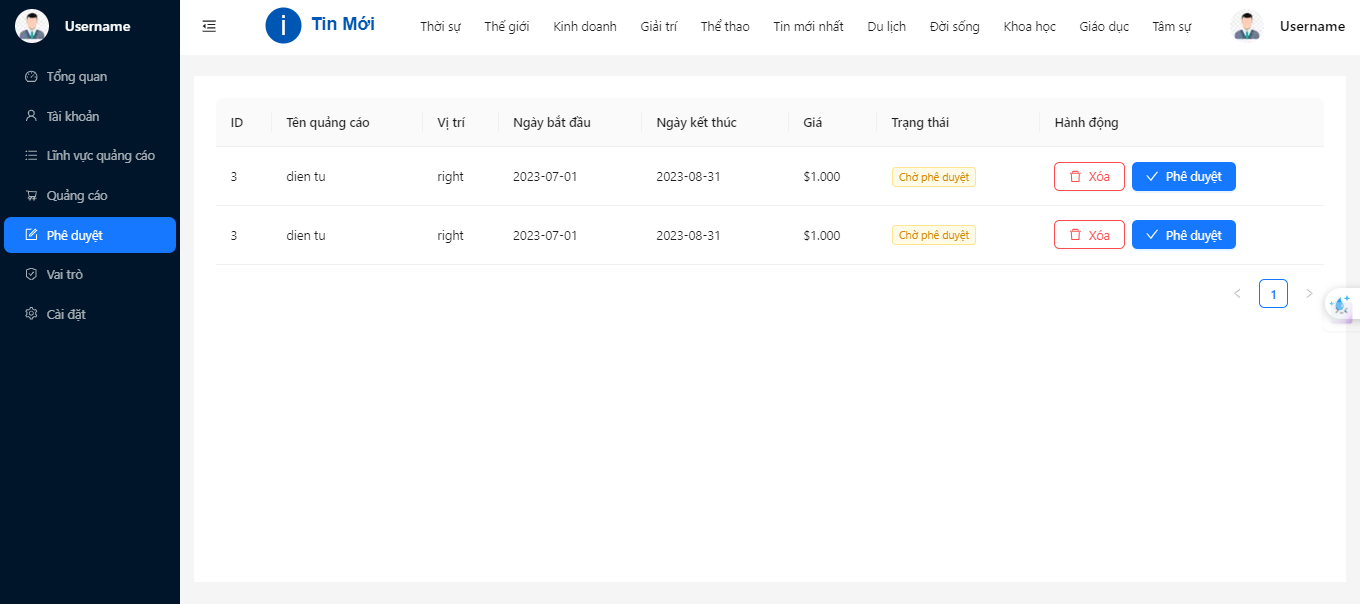
Hình 3.6:Giao diện trang quản lý tài khoản



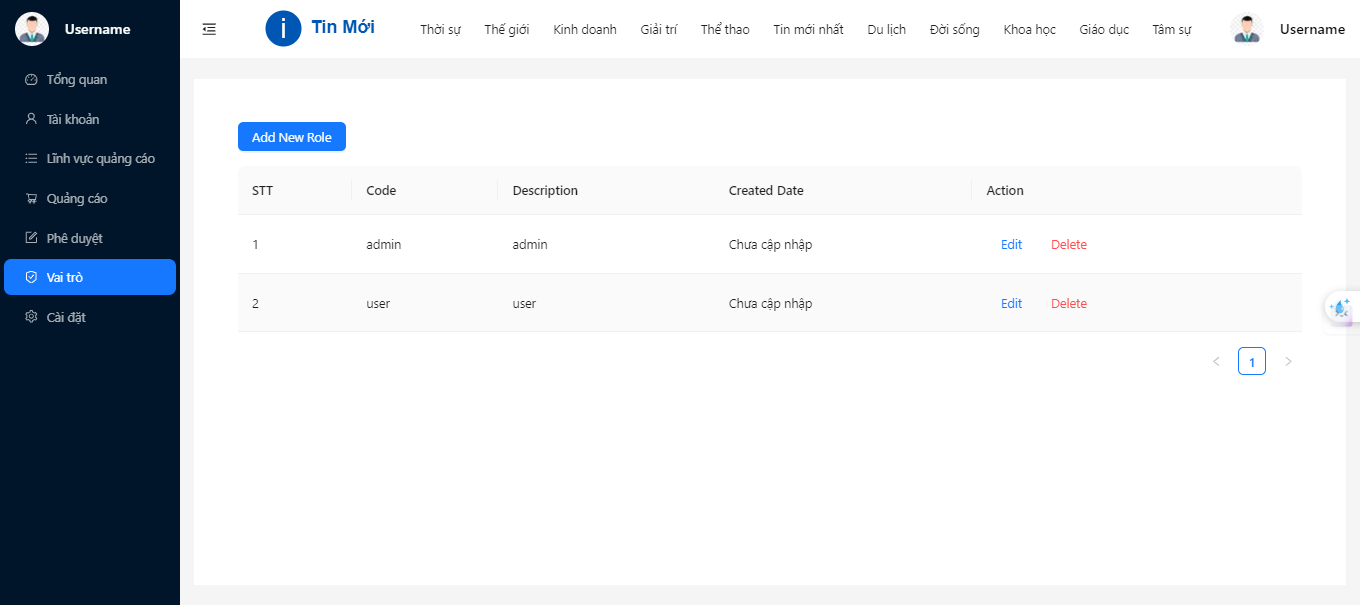
Hình 3.7:Giao diện trang quản lý lĩnh vực quảng cáo



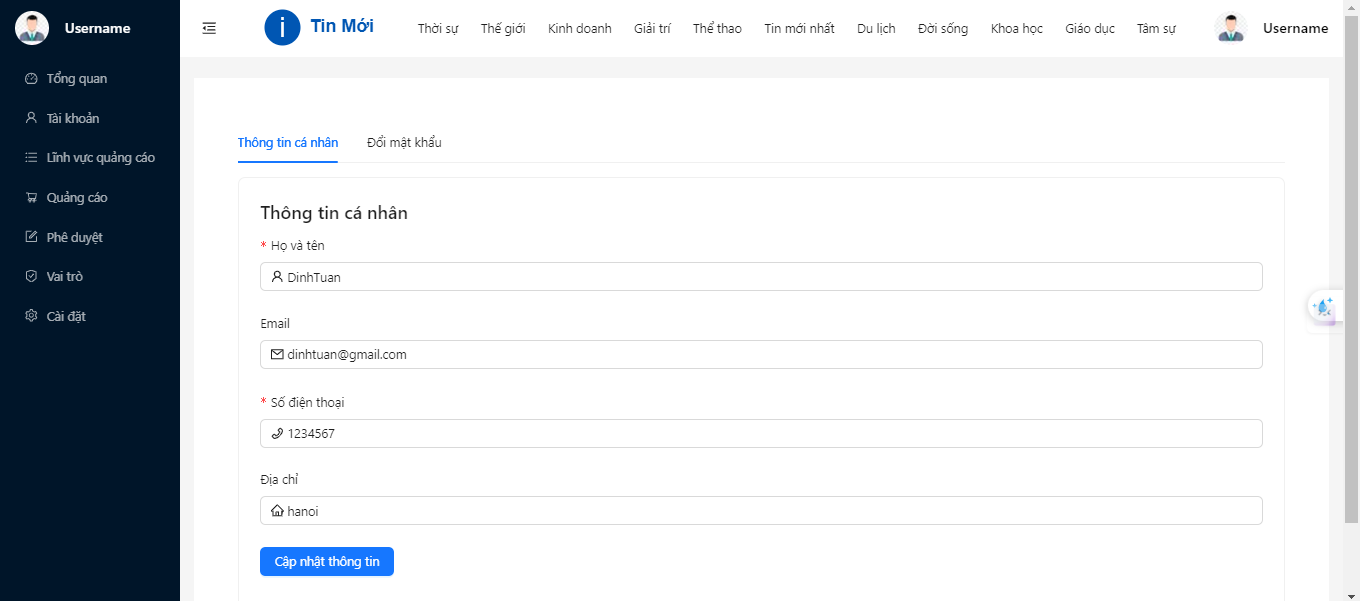
Hình 3.8:Giao diện trang quản lý quảng cáo



Hình 3.9:Giao diện trang phê duyệt quảng cáo



Hình 3.10:Giao diện trang quản lý vai trò



Hình 3.11:Giao diện quản lý hồ sơ cá nhân

## Kiểm thử hệ thống

### Kế hoạch kiểm thử

- Mục tiêu: Rà soát, phát hiện bug tồn tại của sản phẩm phần mềm, từ đó đưa ra kế hoạch sửa lỗi hiệu quả.

- Phạm vi: Kiểm thử các chức năng chính của website

- Dữ liệu kiểm thử: Sử dụng bộ dữ liệu tự tạo trong database bao gồm:

thông tin về các tài khoản, quảng cáo, vai trò,... Bộ dữ liệu này đầy đủ các thành phần ở mọi chức năng của hệ thống, có thể đảm bảo thực hiện kiểm thử mọi chức năng trong hệ thống.

- Dữ liệu rss kiểm thử: Sử dụng dữ liệu rss bao gồm các trang web bài báo tin tức xem các dữ liệu đã được hiển thị đúng chưa, có lỗi gì không.

- Chiến lược kiểm thử: Kiểm thử thủ công ở mức system test trên cả UI và

chức năng.

### **Kịch bản kiểm thử**

Bảng 24:Kịch bản kiểm thử

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | **Mục đích kiểm thử** | **Đầu vào** | **Đầu ra** | **Kết quả** |
| 1 | Hiển thị danh sách tin tức từ RSS ra trang chủ | Kiểm tra chức năng đọc và hiển thị tin tức từ nguồn RSS | URL nguồn RSS hợp lệ | Danh sách tin tức được hiển thị chính xác theo ngày giờ mới nhất | Đạt |
| 2 | Hiển thị danh mục tin tức từ RSS | Kiểm tra chức năng đọc và hiển thị tin tức từ nguồn RSS theo danh mục | URL nguồn RSS hợp lệ | Danh sách tin tức được hiển thị chính xác theo danh mục | Đạt |
| 3 | Tìn kiếm tin tức từ dữ liệu RSS | Kiểm tra chức năng tìm kiếm và hiển thị tin tức từ nguồn RSS | Các từ khóa hợp lệ | Danh sách tin tức chứa từ khóa được hiển thị chính xác | Đạt |
| 4 | Thay đổi chủ đề | Kiểm tra chức năng thay đổi chủ đề | Các tùy chọn sẵn có | Chủ đề hiển thị đúng với người đọc chọn | Đạt |
| 5 | Đăng nhập quản trị viên | Kiểm tra xác thực người dùng quản trị | Tên đăng nhập, vai trò hợp lệ và mật khẩu hợp lệ | Đăng nhập thành công, truy cập được trang quản trị đúng với vai trò | Đạt |
| 6 | Thêm tài khoản mới | Kiểm tra chức năng thêm tài khoản của quản trị viên | Thông tin tài khoản hợp lệ | Tài khoản được thêm vào hệ thống | Đạt |
| 7 | Chỉnh sửa tài khoản | Kiểm tra chức năng chỉnh sửa tài khoản của quản trị viên | Thông tin chỉnh sửa hợp lệ | Thông tin mới của tài khoản được cập nhật | Đạt |
| 8 | Xóa tài khoản | Kiểm tra chức năng xóa tài khoản của quản trị viên | ID tài khoản cần xóa | Tài khoản được xóa khỏi hệ thống | Đạt |
| 9 | Thêm mới vai trò | Kiểm tra chức năng thêm mới vai trò | Thông tin vai trò hợp lệ | Thêm vai trò mới vào hệ thống | Đạt |
| 10 | Chỉnh sửa vai trò | Kiểm tra chức năng chỉnh sửa vai trò | Thông tin mới vai trò hợp lệ | Thông tin của vai trò được cập nhật | Đạt |
| 11 | Xóa vai trò | Kiểm tra chức năng xóa vai trò của quản trị viên | ID vai trò cần xóa | Vai trò được xóa khỏi hệ thống | Đạt |
| 12 | Phê duyệt quảng cáo | Kiểm tra chức năng phê duyệt quảng cáo của quản trị viên | ID quảng cáo cần phê duyệt | Trạng thái quảng cáo được chuyển sang phê duyệt | Đạt |
| 13 | Đăng nhập nhà quảng cáo | Kiểm tra xác thực người dùng nhà quảng cáo | Tên đăng nhập và mật khẩu hợp lệ | Đăng nhập thành công, truy cập được trang quản lý quảng cáo | Đạt |
| 14 | Tạo chiến dịch quảng cáo mới | Kiểm tra chức năng tạo chiến dịch quảng cáo | Thông tin chiến dịch hợp lệ | Chiến dịch quảng cáo được tạo trong hệ thống | Đạt |
| 15 | Cập nhật trạng thái quảng cáo | Kiểm tra chức năng cập nhật trạng thái quảng cáo | ID quảng cáo và trạng thái mới | Trạng thái quảng cáo được cập nhật | Đạt |
| 16 | Gửi phê duyệt quảng cáo | Kiểm tra chức năng gửi phê duyệt quảng cáo | Quảng cáo hợp lệ | Gửi thanh công và hiển thị ra danh sách các quảng cáo gửi phê duyệt | Đạt |
| 17 | Tìm kiếm tin tức | Kiểm tra chức năng tìm kiếm tin tức | Từ khóa tìm kiếm | Danh sách tin tức phù hợp với từ khóa | Đạt |
| 18 | Phân trang danh sách tin tức | Kiểm tra chức năng phân trang | Số trang và số lượng tin trên mỗi trang | Hiển thị chính xác tin tức theo phân trang | Đạt |
| 19 | Thống kê lượt xem quảng cáo | Kiểm tra chức năng thống kê cho nhà quảng cáo | Khoảng thời gian thống kê | Báo cáo chính xác số lượt xem quảng cáo | Đạt |

### **Kết quả kiểm thử**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Function Name** | **Passed** | **Failed** | **N/A** | **Untested** | **Number of test cases** |
| 1 | Xem danh sách tin tức | 1 | 0 | 0 | 0 | 1 |
| 2 | Xem chi tiết tin tức | 1 | 0 | 0 | 0 | 1 |
| 3 | Tùy chọn chủ đề | 2 | 0 | 0 | 0 | 2 |
| 4 | Tùy chọn giao diện | 2 | 0 | 0 | 0 | 2 |
| 5 | Xem trang chủ | 2 | 0 | 0 | 0 | 2 |
| 6 | Quản lý tài khoản | 4 | 0 | 0 | 0 | 4 |
| 7 | Quản lý vai trò | 4 | 0 | 0 | 0 | 4 |
| 8 | Quảng lý quảng cáo | 6 | 0 | 0 | 0 | 6 |
| 9 | Quản lý hồ sơ cá nhân | 2 | 0 | 0 | 0 | 2 |
| 10 | Xem thống kê | 3 | 0 | 0 | 0 | 3 |
| 11 | Gửi phê duyệt quảng cáo | 2 | 0 | 0 | 0 | 2 |
| 12 | Phê duyệt quảng cáo | 2 | 0 | 0 | 0 | 2 |
| 13 | Quản lý doanh thu | 3 | 0 | 0 | 0 | 3 |
| 14 | Thanh toán | 2 | 0 | 0 | 0 | 2 |
|  | **SUM** | 36 | 0 | 0 | 0 | 36 |

Hình 3.12: Biểu đồ kết quả kiểm thử

**Test successful: 98.4%**

**Test failed: 2.6%**

# TRIỂN KHAI VÀ VẬN HÀNH

## Chuẩn bị môi trường triển khai

**Yêu cầu hệ thống cho Front-end**

1. Node.js phiên bản tương thích với Next.js
2. npm hoặc yarn package manager
3. Git repository đã được cấu hình
4. Tài khoản Vercel và GitHub được liên kết

**Yêu cầu hệ thống cho Back-end**

1. Java Development Kit (JDK) phiên bản tương thích với Spring Boot
2. Maven hoặc Gradle build tool
3. Docker installed và cấu hình
4. Git repository đã được cấu hình
5. Tài khoản Render.com và GitHub được liên kết

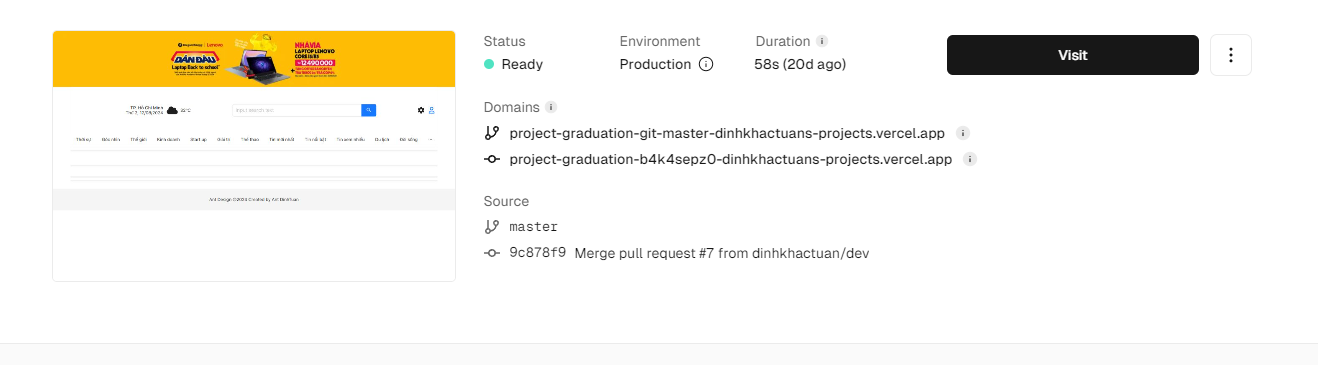
## Quy trình triển khai lên server

**Triển khai front-end lên Vercel**

1. Cấu hình GitHub Actions trong file Github-ci-cd.yml

****

1. Liên kết Vercel với GitHub repository
2. Nếu deploy thành công thì sẽ có giao diện như sau:

****

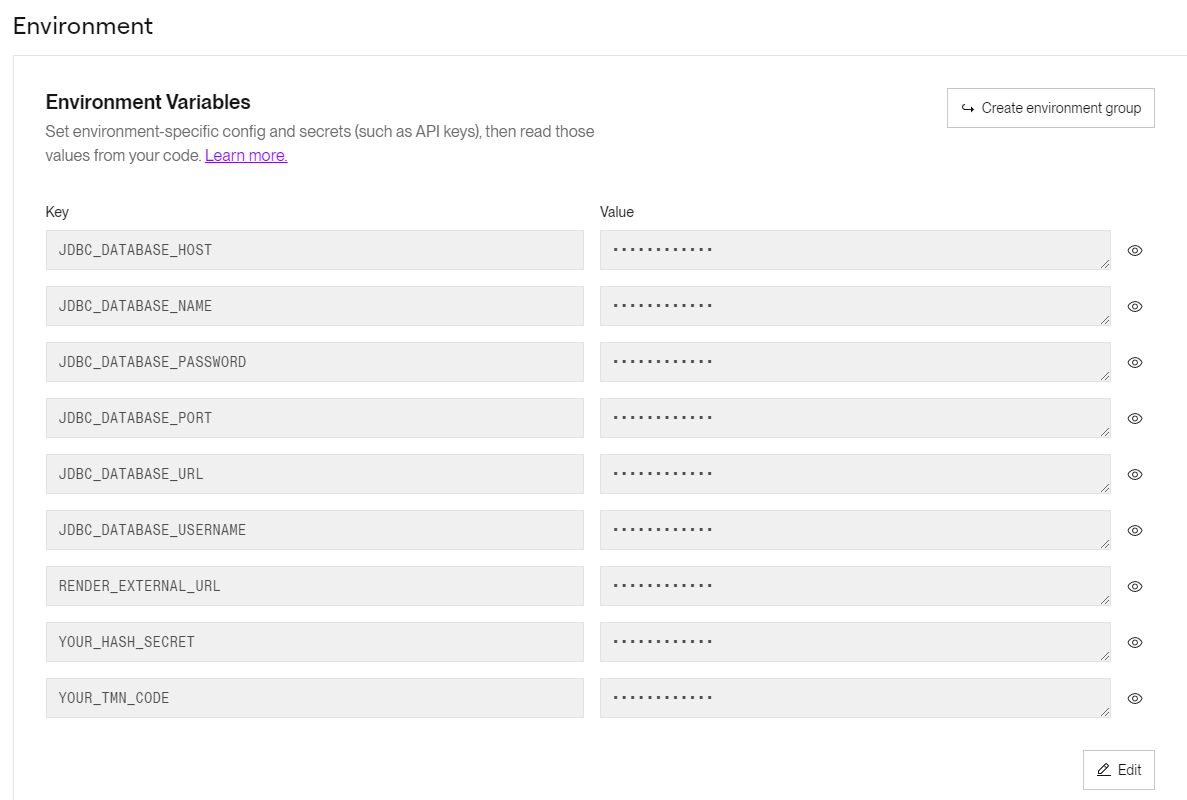
1. **Url truy cập trang web:** [**https://project-graduation-bice.vercel.app/**](https://project-graduation-bice.vercel.app/)
2. Tạo Dockerfile trong thư mục gốc của project Spring Boot:

****

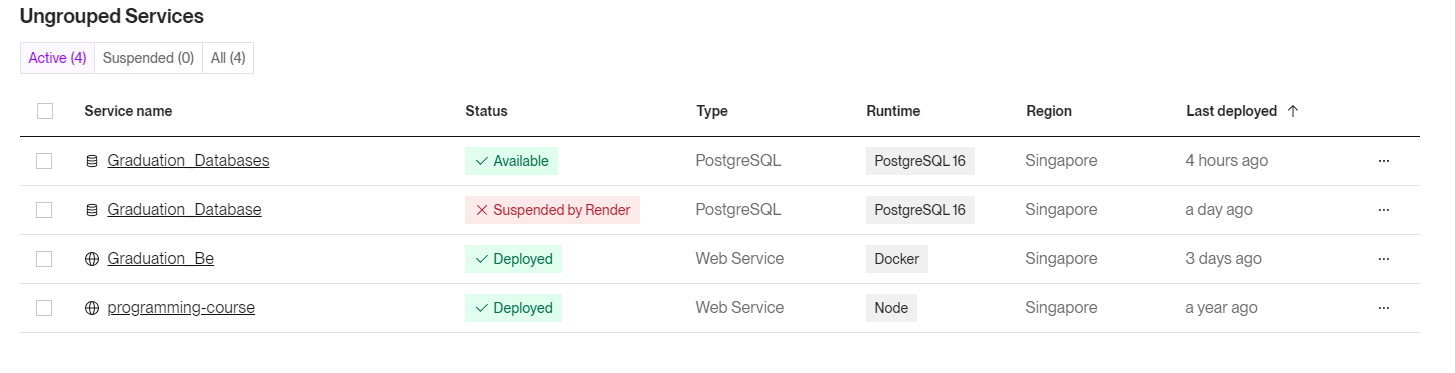
1. Cấu hình file application.properties trong Spring Boot:



1. Thiết lập biến môi trường cho render

****

1. Sau khi deploy nếu thành công sẽ có giao diện như sau:



1. **Url**: <https://graduation-be-vlen.onrender.com/>

# KẾT LUẬN

Qua quá trình thực hiện đồ án tốt nghiệp với đề tài "Tìm hiểu về RSS feed và áp dụng xây dựng website Tin Mới sử dụng nodejs và JavaScript", em đã thu được nhiều kinh nghiệm và kiến thức quý báu, chuẩn bị cho công việc sau khi tốt nghiệp. Cụ thể, em đã tiếp cận và nâng cao các kỹ năng sau:

- Sử dụng thành thạo Java và nắm vững các kiến thức về Spring Boot trong mô hình microservices, áp dụng nguyên lý OOP, và làm quen với các công nghệ như Docker, API RESTful.

- Nắm bắt và áp dụng được các kiến thức về HTML, CSS, JavaScript, TypeScript, Next.js và AntDesign để xây dựng giao diện người dùng hiện đại và responsive.

- Thực hành việc tối ưu hóa truy vấn cơ sở dữ liệu và xây dựng API hiệu quả để truyền tải dữ liệu giữa back-end và front-end.

- Hiểu sâu về cách thức hoạt động của RSS feed và ứng dụng trong việc tổng hợp tin tức từ nhiều nguồn.

Website Tin Mới đã đạt được các chức năng cơ bản như:

- Tự động cập nhật tin tức từ nhiều nguồn thông qua RSS feed.

- Hiển thị danh sách tin tức theo chủ đề, thời gian.

- Cho phép người dùng tìm kiếm, lọc tin tức theo từ khóa hoặc chủ đề.

- Cung cấp giao diện quản trị để quản lý nguồn tin, phân loại và kiểm duyệt nội dung.

Trong thời gian tới, em dự định phát triển thêm các chức năng sau để hoàn thiện website:

- Cải thiện thuật toán gợi ý tin tức dựa trên lịch sử đọc của người dùng.

- Thêm tính năng đánh giá và bình luận cho bài viết.

- Phát triển ứng dụng mobile để người dùng có thể truy cập tin tức mọi lúc, mọi nơi.

- Tích hợp các công cụ phân tích dữ liệu để theo dõi xu hướng tin tức và hành vi người dùng.

- Nâng cao bảo mật và tối ưu hiệu suất hệ thống để đáp ứng lượng truy cập lớn.

Qua đồ án này, em không chỉ nâng cao kỹ năng lập trình mà còn hiểu rõ hơn về quy trình phát triển một ứng dụng web từ ý tưởng đến sản phẩm hoàn chỉnh. Kinh nghiệm thu được từ việc xây dựng API quản lý bằng Spring Boot và tích hợp RSS feed sẽ là nền tảng quan trọng cho sự phát triển nghề nghiệp của em trong tương lai.

# TÀI LIỆU THAM KHẢO

[1]. Giáo trình thiết kế web - Trường đại học Công nghiệp Hà Nội.

[2]. Giáo trình Lập trình web api bằng Java Springboot – Trường đại học Công Nghiệp Hà Nội.

[3]. Trang tài liệu về kiến trúc Microservices cùng với Spring: <https://spring.io/blog/2015/07/14/microservices-with-spring>

[4]. Trang tài liệu về Antdesign: <https://ant.design/components/overview/>

[5]. Trang tài liệu về Springboot: <https://docs.spring.io/spring-boot/index.html>

[6]. Trang tài liệu về Nextjs: <https://nextjs.org/docs>