|  |  |
| --- | --- |
| logomobile.png | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN CÔNG CỤ MÔI TRƯỜNG**

**PHÁT TRIỂN PHẦN MỀM**

**ỨNG DỤNG CÔNG NGHỆ REACT NATIVE VÀ EXPO ĐỂ PHÁT TRIỂN SOCIAL APP**

Ngành : **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành : **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : Ths. **NGUYỄN ĐÌNH ÁNH**

Sinh viên thực hiện : **NGUYỄN QUANG HUY 1611060130**

**PHAN QUỐC CƯỜNG 1611060345**

**TRẦN VÕ TỐ QUYÊN 1611060229**

**TRẦN MINH TUẤN 1611060318**

**VÕ MINH DUY 1611060078**

TP. Hồ Chí Minh, 2020

# MỤC LỤC

**Đề mục Trang**

Trang bìa

[MỤC LỤC 2](#_Toc42940204)

[LỜI MỞ ĐẦU 4](#_Toc42940205)

[CHƯƠNG 1. TỔNG QUAN 5](#_Toc42940206)

[1.1. Giới thiệu đề tài 5](#_Toc42940207)

[1.1.1. Khảo sát thực trạng 5](#_Toc42940208)

[1.1.2. Mô tả 5](#_Toc42940209)

[1.1.4. Các bước thực hiện để hoàn thành dự án 6](#_Toc42940210)

[1.2. Ý tưởng hình thành ứng dụng 6](#_Toc42940211)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 7](#_Toc42940212)

[2.1. Các khái niệm 7](#_Toc42940213)

[2.1.1. Native vs Hybrid: 7](#_Toc42940214)

[2.1.2. Tổng quan về React Native 8](#_Toc42940215)

[2.1.3. Ưu và nhược điểm của 9](#_Toc42940216)

[2.1.4. Setup Environment – Development 10](#_Toc42940217)

[2.1.6. Tổng quan về Expo 11](#_Toc42940218)

[2.1.7. Những hạn chế của Expo 11](#_Toc42940219)

[CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM 13](#_Toc42940220)

[3.1. Các màn hình hiển thị 13](#_Toc42940221)

[3.1.1. Màn hình trang chủ 13](#_Toc42940222)

[3.1.2. Màn hình trang chủ (nội dung) 13](#_Toc42940223)

[3.1.3. Màn hình chi tiết sản phẩm 14](#_Toc42940224)

[3.1.4. Màn hình chi tiết sản phẩm (Bình luận của khách hàng) 14](#_Toc42940225)

[3.1.5. Màn hình admin (home) 15](#_Toc42940226)

[3.1.6. Màn hình admin quản lý người dùng 15](#_Toc42940227)

[3.1.6. Màn hình admin quản lý bài viết 16](#_Toc42940228)

[3.2. Thiết kế giao diện hệ thống 16](#_Toc42940229)

[3.2.1 Giao diện trang chủ 16](#_Toc42940230)

[3.2.2 Giao diện trang danh sách sản phẩm theo danh mục 17](#_Toc42940231)

[3.2.3 Giao diện trang hỗ trợ 17](#_Toc42940232)

[3.2.4 Giao diện trang chi tiết sản phẩm 18](#_Toc42940233)

[3.2.5 Giao diện trang chính admin 18](#_Toc42940234)

[3.2.6 Giao diện admin quản lý users 19](#_Toc42940235)

[3.2.7 Giao diện chức năng đăng bài 19](#_Toc42940236)

[3.2.8 Giao diện chức năng đăng bài 19](#_Toc42940237)

[3.2.9 Giao diện thêm sửa người dùng 20](#_Toc42940238)

[3.2.10 Giao diện thêm sửa người dùng 20](#_Toc42940239)

[3.2.11 Giao diện phpMyAdmin 20](#_Toc42940240)

[3.3. Sơ đồ thực thể kết hợp 20](#_Toc42940241)

[3.3.1 Bảng commet 20](#_Toc42940242)

[3.3.2 Bảng post 22](#_Toc42940243)

[3.3.3 Bảng post\_topic 22](#_Toc42940244)

[3.3.4 Bảng topics 22](#_Toc42940245)

[3.3.5 Bảng users 22](#_Toc42940246)

[CHƯƠNG 4. KẾT LUẬN 23](#_Toc42940247)

[4.1. Kết quả đạt được 23](#_Toc42940248)

[4.2. Đánh giá phần mềm 23](#_Toc42940249)

[4.2.1. Ưu điểm 23](#_Toc42940250)

[4.2.2. Nhược điểm 23](#_Toc42940251)

[4.3. Hướng phát triển và mở rộng đề tài 23](#_Toc42940252)

[Tài liệu tham khảo 24](#_Toc42940253)

# LỜI MỞ ĐẦU

Hiện tại ở Việt Nam sự phát triển về CNTT ngày càng khủng khiếp. Những công nghệ mới đã và đang làm thay đổi thế giới một cách nhanh chóng. Từ lúc mới ra đời cho đến nay smartphone đã có những bước tiến mạnh mẽ vì vậy mà những công nghệ kèm theo cũng đòi hỏi những nhà phát triển phần mềm viết ra nhiều phần mềm hơn để có thể sử dụng được trên smartphone chứ không còn đơn thuần là những ứng dụng nhắn tin gọi điện thông thường.

Biết được vấn đề này, vai trò và mục tiêu của bộ môn Công cụ mội trường phát triển phần mềm mang lại, chung em đã phát triển đề tài **“Ứng dụng công nghệ React Native và Expo để phát triển Social App”** từ bước mô tả yêu cầu, phân tích, thiết kế, hiện thực và kiểm thử.

Chúng em xin cám ơn thầy **ThS. Nguyễn Đình Ánh** đã truyền đạt cho chúng em những kiến thức, hiểu biết quý báu cùng niềm đam mê để chúng em có thể hoàn thành tốt đề tài này.

Do chưa có nhiều kinh nghiệm nghiên cứu, thực hành nên đề tài thực hiện còn nhiều thiếu sót, chúng em mong nhận được đóng góp ý kiến của thầy để đề tài được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

**Nguyễn Quang Huy**

**Phan Quốc Cường**

**Trần Võ Tố Quyên**

**Trần Minh Tuấn**

**Võ Minh Duy**

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Giới thiệu đề tài

Nhiều công ty sản xuất phần mềm trên di động hoặc những công ty lớn nhìn thấy đây là mảnh đất màu mỡ và tiềm năng phát triển của nó rất lớn. Đồng thời dựa theo số liệu thống kê như trên thì bắt đầu người dùng đã chịu chi tiền cho smartphone nhiều hơn, số lượng người dùng chịu bỏ tiền ra mua ứng dụng để sử dụng trên smartphone cũng tăng lên theo từng năm. Do đó kéo theo nhu cầu về công việc liên quan đến mảng smartphone nói chung và lập trình mobile nói riêng tăng lên chóng mặt. Đưa ra những tiềm năng và thử thách dành cho những ai muốn phát triển theo hướng lập trình ứng dụng di động. Nhu cầu dạy và học lập trình ứng dụng di động đang dần trở thành xu hướng của xã hội hiện nay.

### Khảo sát thực trạng

* Ngày nay chúng ta có hai nền tảng cạnh tranh chính là iOS và Android, mỗi kẻ một thế lực làm khuynh đảo giới lập trình. Từ đó sinh ra các công cụ hỗ trợ nhà phát triển ứng dụng có thể phát triển mỗi ứng dụng một cách dễ dàng chẳng hạn như Apple thì có XCode, Android thì có Android Studio.
* Nhưng cũng chính vì lẽ mỗi kẻ một thế lực nên không ai chơi với ai, mỗi bên xưng đế một “phương” cuối cùng con dân lập trình viên là những người phải chịu trận. Và cũng từ lẽ đó mà giang hồ sinh ra hai môn vang danh thiên hạ là Native và Hybrid? Trước khi đi sâu vào React Native thì chúng ta nên biết Native và Hybrid là cái chi chi và những lợi và hại giữa hai môn phái này. Thông qua đó chúng ta sẽ biết được tại sao mà React/React Native lại được khai sinh ra đời để giải quyết vấn đề nan giải của con dân lập trình viên di động.

### Mô tả

*Ưu điểm*

* Có thể tìm được được ngay tại thời điểm mong muốn và lựa chọn nhà cung cấp.

*Nhược điểm*

* Thông tin sản phẩm không nhiều, khó đặt hàng về việt nam

Chức năng của người dùng:

+ Cho phép đăng bài:gồm tiêu đề, hình ảnh, nội dung, loại sản phẩm.

+ Chức năng đăng kí.

+ Đăng nhập để đăng bài, bình luận,..

Chức năng admin:

+ Duyệt bài.

+ Quản lý tài khoản.

+ Duyệt Bình Luận.

+ Quản lý bài viết.

### 1.1.4. Các bước thực hiện để hoàn thành dự án

* Lập kế hoạch phát triển ứng dụng.
* Tìm hiểu về thêm về công cụ để lập trình và khởi chạy.
* Phân tích các yếu tố của ứng dụng.
* Thiết kế các giao diện ứng dụng.
* Lập trình ứng dụng.
* Kiểm tra ứng dụng và phát hành bản thử nghiệm.
* Biên soạn báo cáo và hướng dẫn sử dụng.

## 1.2. Ý tưởng hình thành ứng dụng

* Từ việc …..

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Các khái niệm

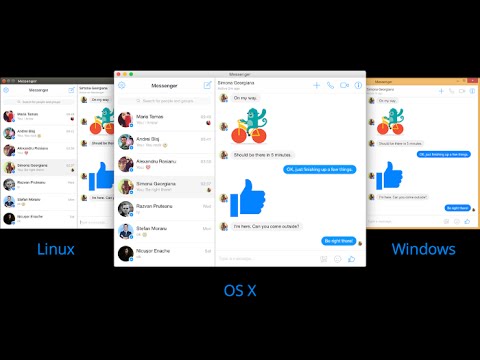
### 2.1.1. Native vs Hybrid:

Native app là ứng dụng được thiết kế riêng cho một hệ điều hành di động. App loại này không thể chạy trên các nền tảng khác, chẳng hạn như một ứng dụng native iOS sẽ được phát triển từ ngôn ngữ lập trình Swift và không thể chạy trên điện thoại Android.

Lợi thế lớn nhất của native app là chúng cho phép người dùng truy cập và sử dụng tất cả các tính năng một cách dễ dàng và sẽ ít khi gặp lỗi nếu được thiết kế một cách hoàn chỉnh. Native app cũng mang đến trải nghiệm tốc độ cao và hoàn chỉnh nhất nên hiện vẫn chiếm phần lớn tổng lượng ứng dụng hiện nay.

Tuy nhiên, cũng chính vì không thể chạy trên các hệ điều hành khác nên nhà phát triển muốn phủ sóng ứng dụng của mình trên nhiều nền tảng buộc phải code lại nó rất nhiều lần. Ví dụ như việc các startup phải phát triển ứng dụng của mình tới 3 lần cho các nền tảng phổ biến như iOS, Android và Windows là không hiếm khi xảy ra. Quá trình không chỉ đòi hỏi nguồn nhân lực tương đối mà còn cả thời gian, công sức và chi phí.

Do vậy, nhiều công ty chưa có nguồn lực dồi dào chọn cách phát triển app cho riêng 1 nền tảng trước. Nếu ứng dụng thành công trên nền tảng này, họ sẽ quay lại phát triển nó cho nền tảng khác. Một điều đáng chú ý nữa là các app chạy trên iOS 7 trở lên hiện đang mang về nhiều lợi nhuận nhất trong số tất cả các nền tảng nên lẽ dễ hiểu là ngày càng có nhiều công ty chọn thiết kế app cho iPhone đầu tiên.



Đúng như tên gọi, hybrid app là loại ứng dụng được thiết kế để chạy được trên nhiều nền tảng khác nhau. Chúng được phát triển dựa trên một ngôn ngữ lập trình tiêu chuẩn (chẳng hạn nhưu C# kết hợp với HTML5 và Javascript). Các tương tác của người dùng trên app loại này cũng thường nằm dưới sự quản lý của các plugin trong hệ điều hành.

Lợi thế lớn nhất của app Hybrid là chúng có thể hỗ trợ đa nền tảng và giúp các công ty tiết kiệm chi phí code nhiều lần. Về phía người dùng, miễn là ứng dụng được thiết kế tốt, thì họ cũng khó lòng phân biệt được đâu là native đâu là hybrid app. Với những framework hỗ trợ hư Ionic, Phonegap,… ngày càng được nâng cấp, rất nhiều startup nổi tiếng đã chuyển sang hybrid app, tiêu biểu là Uber, Evernote, Instagram, Gmail,…Rất bất ngờ phải không?

Tuy nhiên, hybrid cũng không phải là không có nhược điểm. Đối với ứng dụng đỏi hỏi những tương tác phức tạp thì hybrid app sẽ là bài toán khó cho các nhà phát triển. Hybrid có nhiều hạn chế khi lập trình viên muốn phát triển app của mình lên mức cao và tinh tế hơn. Chi phí duy trì ứng dụng hybrid cũng thường lớn hơn chi phí cho các ứng dụng native.

### 2.1.2. Tổng quan về React Native

Giới thiệu với các bạn, gương mặt đứa con của chúng ta hôm nay đây rồi React Native. React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động. Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái luôn, sợ chưa!!!

Một điểm hay ho nữa mà mình có đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là bạn không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa.

Và điểm lợi hại kế tiếp của React Native đó chính là chúng ta chỉ cần sử dụng JS để phát triển được một ứng dụng di động hoàn chỉnh, đồng thời giải quyết được các vấn đề mà Native App gặp phải mà mình đã nêu ở trên. Và rồi còn cả kết hợp với code native như Swift, Java, v.v… Tới đây mình chỉ có thể nói là “Thật là vi diệu”.

### 2.1.3. Ưu và nhược điểm của

Trước khi đi đến phần kế tiếp mình xin dừng lại chút xíu để phân tích xem liệu React Native có phải thần thánh không hay chỉ là cũng như bao framework khác đều có những điểm lợi và hại.

Ưu điểm:

* Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng nhanh chóng.
* Hiệu năng tương đối ổn định.
* Cộng đồng phát triển mạnh.
* Tiết kiệm tiền.
* Team phát triển nhỏ.
* Ứng dụng tin cậy và ổn định.
* Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
* Trải nghiệm người dùng tốt hơn là hybrid app.

Nhược điểm:

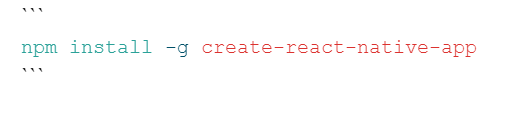
* Vẫn đòi hỏi native code.
* Hiệu năng sẽ thấp hơn với app thuần native code.
* Bảo mật không cao do dựa trên JS.
* Quản lý bộ nhớ.
* Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

### 2.1.4. Setup Environment – Development

React Native có thể chạy trên Windows, Linux, MacOS. Cách cài đặt bạn có thể xem trên trang hướng dẫn của Facebook hoặc làm theo mình bên dưới ở đây mình sẽ demo trên hệ điều hành linux hoặc cũng có thể bạn chỉ muốn trải nghiệm về cách lập trình thì sử dụng online editor EXPO.

Sau đây là các bước đơn giản nhất, không cần phải cài đặt Android Studio hay XCode để build ra file cài đặt. Các bạn muốn sử dụng cách mà có sử dụng Android Studio hay XCode có thể tham khảo trên trang của Facebook phần Building Project with Native Code. Ở đây mình sẽ chỉ viết code JS thuần không sử dụng bất kì Native Code nào.

1. **Trước hết bạn phải cài**[**NodeJS**](https://nodejs.org/en/)**.**
2. **Cài đặt**[**create-react-native-app**](https://www.npmjs.com/package/create-react-native-app)**.**



1. **Tạo project và run project mẫu.**



1. **Khởi chạy project trên điện thoại.**

Đầu tiên các bạn cài đặt Expo client trên điện thoại của các bạn Android, iOS. Kết nối cùng mạng wifi với máy tính mà bạn run project. Sau đó mở ứng dụng Expo Client lên và quét mã QR trên terminal của run project. Khi bạn thay đổi gì trong App.js thì ứng dụng của các bạn sẽ tự động thay đổi tự động.

### 2.1.6. Tổng quan về Expo

Expo là một bộ công cụ giúp dễ dàng viết các ứng dụng React Native. Trong bài này, tôi sẽ hướng dẫn cho bạn cách bạn có thể nhanh chóng tạo ra các ứng dụng React Native bằng Expo. Với Expo, các nhà phát triển có thể tạo ra các ứng dụng React Native mà không gặp bất kỳ sự phiền toái nào giống như khi cài đặt và cấu hình các phần mềm phụ trợ như Android Studio, Xcode hoặc tất cả các công cụ khác cần thiết để phát triển và chạy ứng dụng React Native. Trong này, tôi sẽ hướng dẫn cho bạn cách viết một trò chơi trí nhớ đơn giản bằng Expo. Cùng với đó bạn sẽ được tìm hiểu về những kiến thức sau đây: Cách sử dụng các công cụ do Expo cung cấp. Chúng bao gồm CLI, SDK và ứng dụng Expo client. Cách tạo ứng dụng React Native bằng Expo.

Expo là một framework dùng để phát triển nhanh các ứng dụng React Native. Nó giống như Laravel hay Symphony cho các nhà phát triển PHP, hoặc Ruby on Rails cho các nhà phát triển Ruby. Expo cung cấp một lớp nằm trên cùng của React Native API để giúp chúng dễ sử dụng và quản lý. Nó còn cung cấp các công cụ giúp bạn dễ dàng khởi tạo và kiểm thử các ứng dụng React Native. Sau cùng, nó cung cấp các thành phần UI và các dịch vụ thường chỉ có sẵn khi bạn cài đặt một thành phần React Native của bên thứ ba. Tất cả đều được cung cấp thông qua Expo SDK.

### 2.1.7. Những hạn chế của Expo

Trước khi đi sâu hơn, điều quan trọng là cần phải biết được một số hạn chế của Expo:

Expo không hỗ trợ thực thi code trong nền. Điều này có nghĩa là bạn không thể, ví dụ, chạy code để lắng nghe những thay đổi vị trí khi ứng dụng bị đóng.

Các ứng dụng Expo bị giới hạn trong các API gốc mà Expo SDK hỗ trợ. Điều này có nghĩa là nếu ứng dụng của bạn cần sử dụng một tính năng đặc trưng như giao tiếp với thiết bị ngoại vi thông qua Bluetooth, thì lựa chọn duy nhất để cài đặt chức năng đó là với React Native đơn thuần, hoặc bằng cách viết code bằng một thư viện có tên ExpoKit.

Expo trói buộc bạn vào bộ công cụ của họ. Điều này có nghĩa là bạn không thể đơn giản là cài đặt và sử dụng hầu hết các công cụ tuyệt vời có sẵn để phát triển React Native chẳng hạn như công cụ dòng lệnh, scaffolders và các framework UI. Nhưng điều tốt là Expo SDK tương thích với các ứng dụng React Native thuần tuý, vì vậy bạn sẽ không gặp bất kỳ vấn đề nào khi bạn gỡ ứng dụng khỏi Expo.

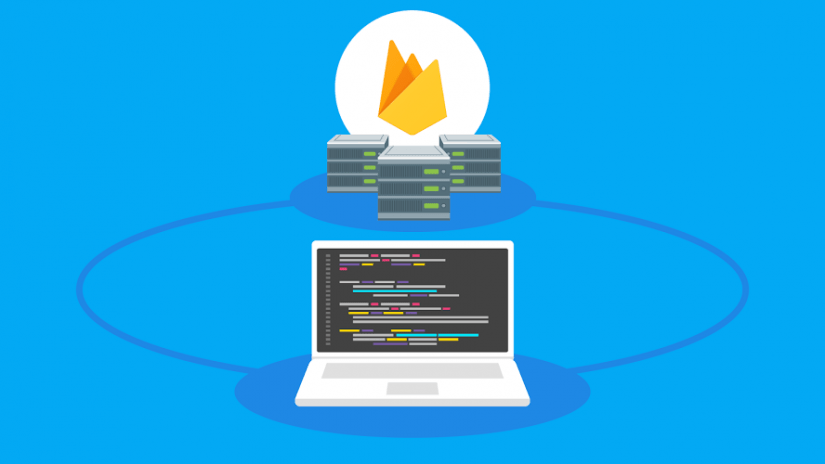
Mã nhị phân độc lập của các ứng dụng Expo chỉ có thể được built trực tuyến. Expo cung cấp một công cụ dòng lệnh có tên là Exp. Công cụ này cho phép các nhà phát triển bắt đầu quá trình built trên các máy chủ Expo. Khi đã hoàn tất, một URL sẽ được cung cấp để tải về tập tin .apk hoặc .ipa.

Ngay cả với những hạn chế này, điều quan trọng cần lưu ý là Expo là một framework đầy đủ chức năng với rất nhiều hỗ trợ cho các API thông dụng trên Android hoặc iOS. Điều này có nghĩa là nó đã bao quát cho bạn hầu hết các chức năng mà các ứng dụng thường cần. Vì vậy, thường không cần phải tìm bên ngoài Expo để cài đặt các tính năng gốc.

### 2.1.8. Tổng quan về firebase, firestore

Firebase của Google cung cấp hai loại cơ sở dữ liệu NoSQL cho người dùng có thể lưu trữ cơ sở dữ liệu đám mây và là giải pháp cho các ứng dụng di động với dữ liệu được đồng bộ hoá theo thời gian thực - Realtime Database và Cloud Firestore.

Giải pháp cho xây dựng ứng dụng không Server.



Hình: kết nối firebase

Firebase Realtime Database là gì ? Firebase Realtime Database là một cơ sở dữ liệu NoSQL được lưu trữ đám mây cho phép bạn lưu trữ và đồng bộ dữ liệu. Dữ liệu được lưu trữ dưới dạng JSON và được đồng bộ hóa theo thời gian thực cho mọi máy kết nối. Khi bạn xây dựng các ứng dụng đa nền tảng với SDK iOS, Android và JavaScript, tất cả các client của bạn sẽ chia sẻ một phiên bản Realtime Database và tự động cập nhật với dữ liệu mới nhất. Do đó đối với các ứng dụng di động yêu cầu trạng thái đồng bộ hóa giữa các máy trong thời gian thực thì đây là một giải pháp hiệu quả và có độ trễ thấp.



Hình: Database

**Tính năng chính Realtime :**

Firebase sử dụng đồng bộ hoá dữ liệu ,mỗi khi dữ liệu thay đổi mọi thiết bị kết nối sẽ nhận được thay đổi lập tức.

Offline : Ứng dụng sử dụng Firebase vẫn khả dụng khi ngoại tuyến vì Firebase Realtime Database SDK vẫn lưu dữ liệu của bạn trên local. Khi kết nối được thiết lập lại, thiết bị sẽ được cập nhật thay đổi nào đã bỏ lỡ và đồng bộ hóa dữ liệu hiện tại với máy chủ.

Sử dụng cho nhiều thiết bị : Firebase Realtime Database có thể được truy cập trực tiếp từ thiết bị di động hoặc trình duyệt web.

Khả năng mở rộng thông qua kết hợp cơ sở dữ liệu : Với Firebase Realtime Database trên gói Blaze (bản trả phí) cho phép mở rộng qua việc sử dụng kết hợp nhiều database trong cùng một Firebase project.

**Cách hoạt động:**

Firebase Realtime Database cho phép bạn xây dựng các ứng dụng dễ dàng, đa dạng ,an toàn với cơ sở dữ liệu trực tiếp từ client. Dữ liệu được duy trì cục bộ và ngay cả khi offline, các sự kiện thời gian thực vẫn sẽ được tiếp tục khi thiết bị lấy lại kết nối, Realtime Database sẽ đồng bộ hóa các thay đổi dữ liệu cục bộ với các dữ liệu từ máy chủ và tự động hợp nhất mọi xung đột.

Firebase Realtime database còn cung cấp một quy tắc bảo mật để xác định cách cấu trúc dữ liệu và quyền đọc ghi dữ liệu thường kết hợp với Authentication Firebase.

Realtime Database là một cơ sở dữ liệu NoQuery và do đó có các tối ưu hóa chức năng so với cơ sở dữ liệu quan hệ. Realtime Database được thiết kế cho việc thực hiện nhanh chóng. Điều này cho phép bạn xây dựng trải nghiệm thời gian thực tuyệt vời có thể phục vụ hàng triệu người dùng mà không cần suy nghĩ đến khả năng đáp ứng.

**Cloud Firestore là gì**

Cloud Firestore là cơ sở dữ liệu mới của Firebase phát triển dành cho ứng dụng di động. Nó là sự kế thừa của Realtime Database với mô hình dữ liệu mới và trực quan hơn. Cloud Firestore phong phú hơn, nhanh hơn và có khả năng mở rộng siêu việt hơn so với Realtime Database.

Giống như Firebase Realtime Database, nó giúp dữ liệu của bạn đồng bộ hóa trên các ứng dụng client thông qua việc đăng ký realtime và cung cấp hỗ trợ ngoại tuyến cho thiết bị di động và web. Cloud Firestore cũng cung cấp tích hợp với các sản phẩm khác của Firebase và Google Cloud Platform, bao gồm cả Cloud Functions.



Hình: Cloud Firestore

Tính năng chính Tính linh hoạt: Cloud Firestore hỗ trợ các cấu trúc dữ liệu linh hoạt, phân cấp dữ liệu. Lưu trữ dữ liệu của bạn trong các document , được tổ chức thành các collection. Các document có thể chứa các đối tượng phức tạp. Truy vấn tượng trưng : Bạn có thể sử dụng các truy vấn để truy xuất các document riêng lẻ hoặc để truy xuất tất cả các document trong collection khớp với các tham số truy vấn của bạn. Các truy vấn của bạn có thể bao gồm nhiều bộ lọc, kết hợp giữa bộ lọc và sắp xếp.

Cập nhật thời gian thực: Cloud Firestore sử dụng đồng bộ hóa dữ liệu để cập nhật dữ liệu trên mọi thiết bị được kết nối. Nó cũng được thiết kế để thực hiện các truy vấn tìm nạp một lần .

Hỗ trợ offline: Cloud Firestore lưu trữ dữ liệu tại local, vì vậy ứng dụng có thể viết, đọc, nghe và truy vấn dữ liệu ngay cả khi thiết bị ngoại tuyến. Khi thiết bị trở lại trực tuyến, Cloud Firestore sẽ đồng bộ hóa mọi thay đổi cục bộ lên Cloud Firestore.

Khả năng mở rộng: Mang đến khả năng từ Google Cloud Platform thiết kế để sử dụng cơ sở dữ liệu khó khăn nhất từ các ứng dụng lớn nhất thế giới.

**Cách hoạt động:**

Cloud Firestore là một cơ sở dữ liệu NoQuery được lưu trữ trên đám mây mà các ứng dụng IOS, Android, Web có thể truy cập trực tiếp thông qua SDK.Cloud Firestore cũng có sẵn trong Node.js, Java, Python và Go SDKs, REST và RPC APIs.

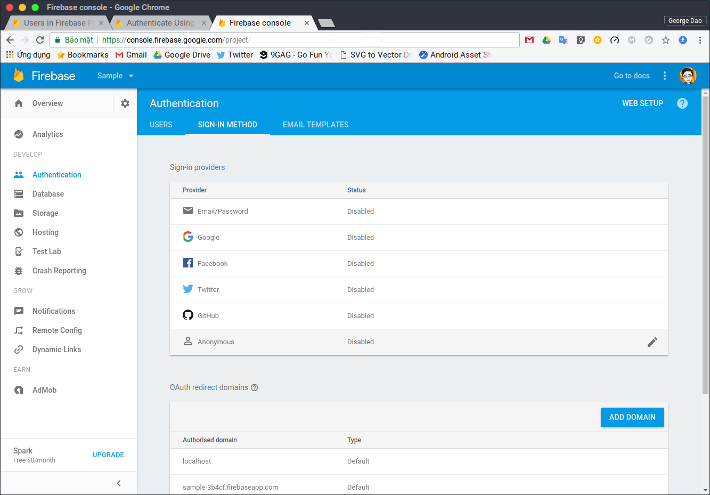
Được tổ chức theo mô hình dữ liệu NoQuery của Cloud Firestore, dữ liệu lưu trong các document ánh xạ tới các giá trị. Các document này được lưu trữ trong các collection cho bạn tổ chức dữ liệu và thực hiện truy vấn.

Bảo vệ quyền truy cập vào dữ liệu của bạn trong Cloud Firestore với Firebase Authentication cho Android, iOS và JavaScript hoặc nhận dạng và quản lý truy cập (IAM) .

### 2.1.8. Tổng quan về authentication

Kể từ khi trào lưu mạng xã hội bùng nổ, từ “cư dân mạng” ra đời, thì tuyệt đại đa số các “cư dân mạng” đó đều có tài khoản mạng xã hội, mà đa phần là Facebook, cũng như dùng Gmail làm địa chỉ email chính. Với sức ảnh hưởng lớn và bành trướng của Facebook sang nhiều lĩnh vực, từ một mạng xã hội thuần túy chia sẻ các cảm xúc, hình ảnh, sang một (nhà cung cấp) nền tảng cho các nhà phát triển khác xây dựng các dịch vụ của mình trên nền tảng đó, và sự phổ biến của Google đến mức “google” đã trở thành một động từ trong nhiều ngôn ngữ, thì người dùng nào cũng có xu hướng thích sign up và sign in gián tiếp qua các tài khoản Facebook hoặc Google của mình. Đơn giản, họ chỉ cần kết nối với tài khoản Facebook hoặc Google trong trang web hoặc ứng dụng, họ sẽ bỏ qua rất nhiều bước phiền phức như điền địa chỉ email (có khi tới 2 lần), điền u-xơ-nem mong muốn, điền pát-uộc 2 lần, xác nhận qua email… Chỉ với 1-2 chạm là họ sẽ dễ dàng hoàn thành thao tác sign up, hoặc sign in.

Thông thường, nếu bạn muốn dùng dịch vụ Authentication của Facebook, Twitter hay Google, thì bạn phải đăng kí với họ, và nhận được một mớ thông tin định danh loằng ngoằng. Và nếu bạn tích hợp bao nhiêu phương thức từ bao nhiêu nhà cung cấp, thì bạn có bấy nhiêu mớ kia để lưu trữ. Firebase Authentication ra đời đóng vai trò trung gian. Thay vì bạn đi theo hướng trực tiếp BẠN – NHÀ CUNG CẤP, bạn sẽ đi hướng gián tiếp BẠN – FIREBASE – NHÀ CUNG CẤP. Bạn sẽ chỉ chủ yếu “deal with” Firebase, phần còn lại thì “Ngọn lửa” sẽ lo hết. Để thông báo với Firebase là bạn sẽ sử dụng dịch vụ Authentication gián tiếp của họ, bạn cần truy cập Firebase Console, mở project của bạn. Trong phần Authentication, bạn chọn SIGN-IN METHOD và cho Enable các nhà cung cấp mà bạn muốn.



Hình: Authentication

# CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM

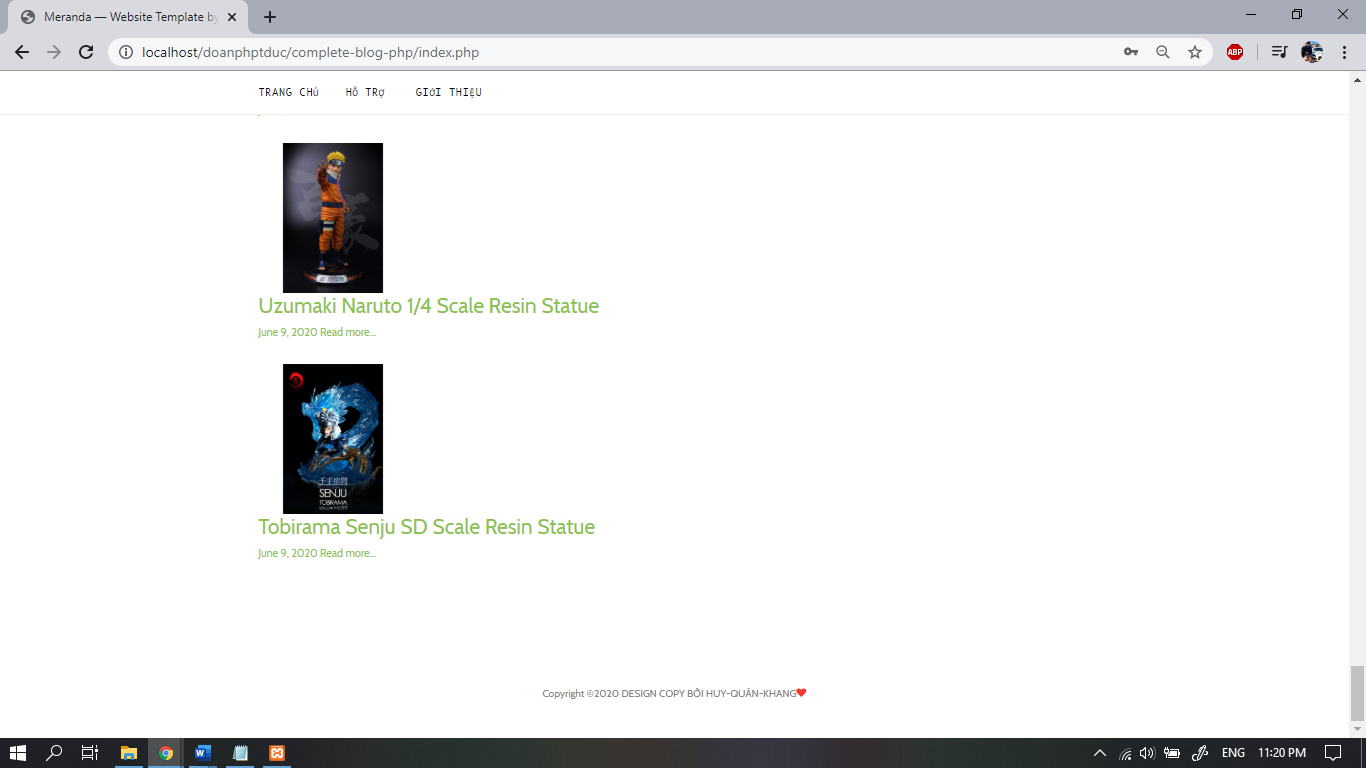
## 3.1. Các màn hình hiển thị

### 3.1.1. Màn hình trang chủ



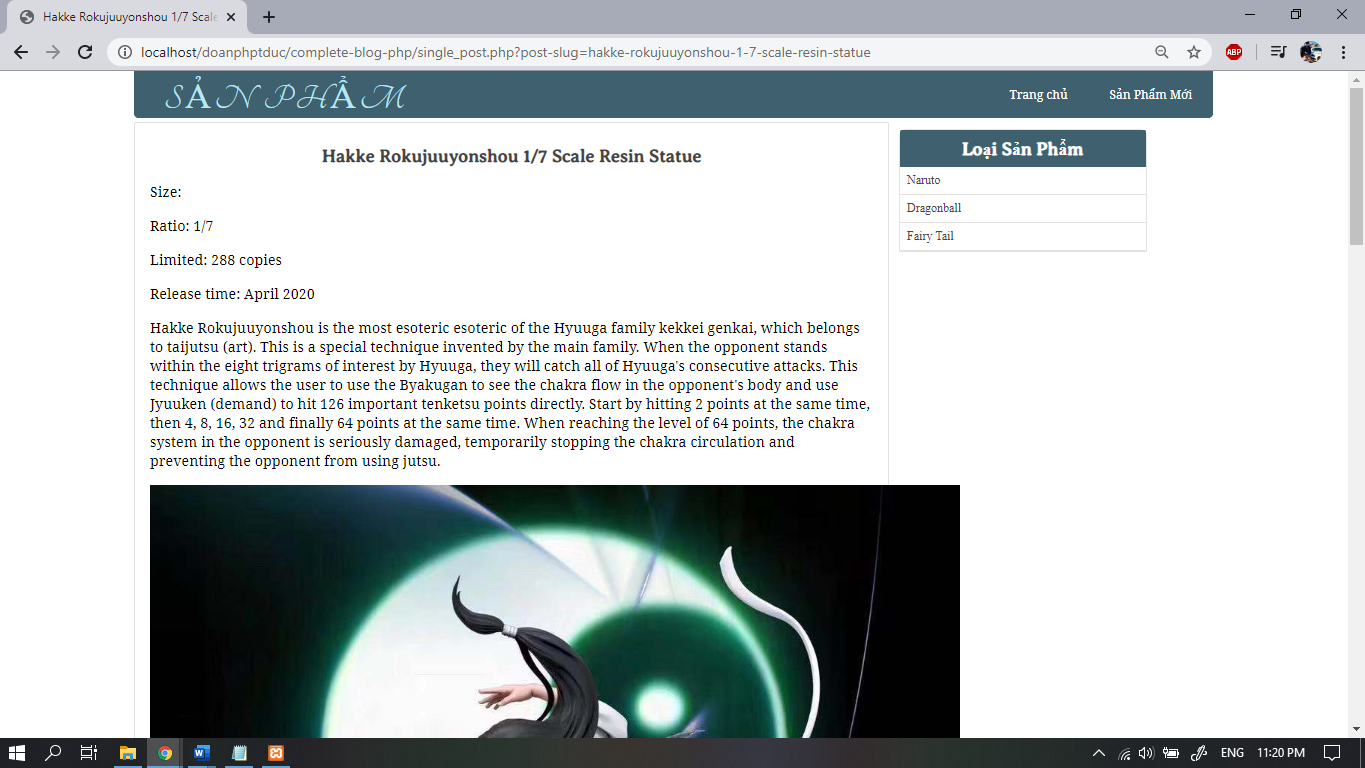
Hình 1: Giao diện trang chủ

### 3.1.2. Màn hình trang chủ (nội dung)



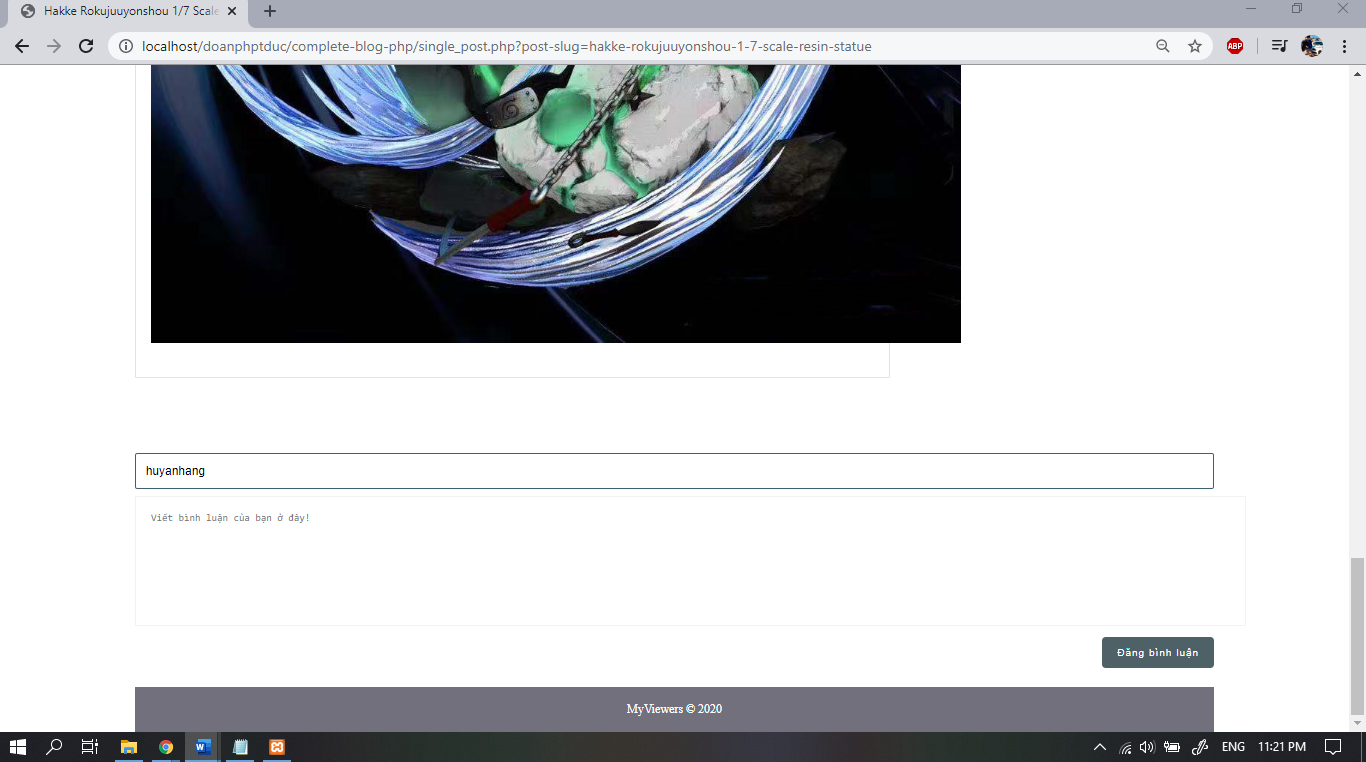
Hình 2: Giao diện trang chủ (nội dung)

### 3.1.3. Màn hình chi tiết sản phẩm



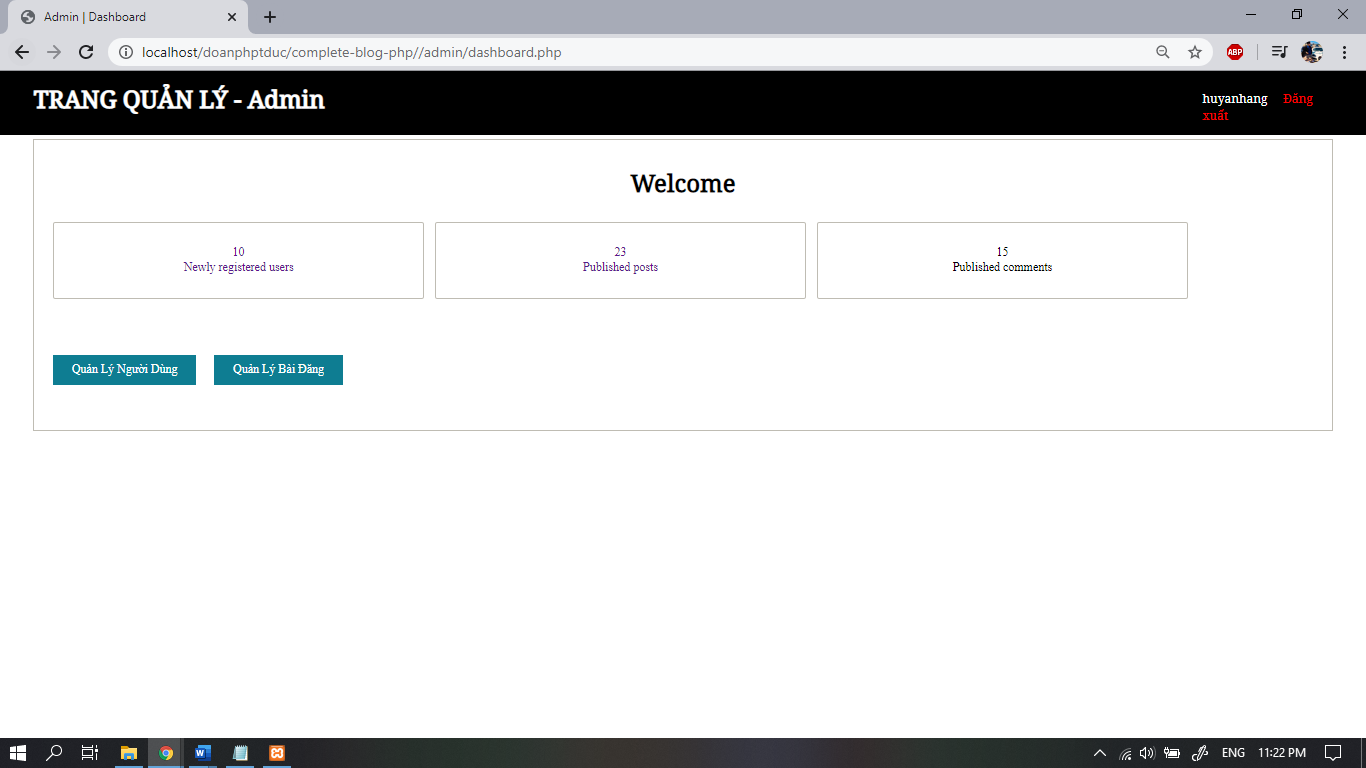
Hình 3: Giao diện chi tiết giản phẩm

### 3.1.4. Màn hình chi tiết sản phẩm (Bình luận của khách hàng)



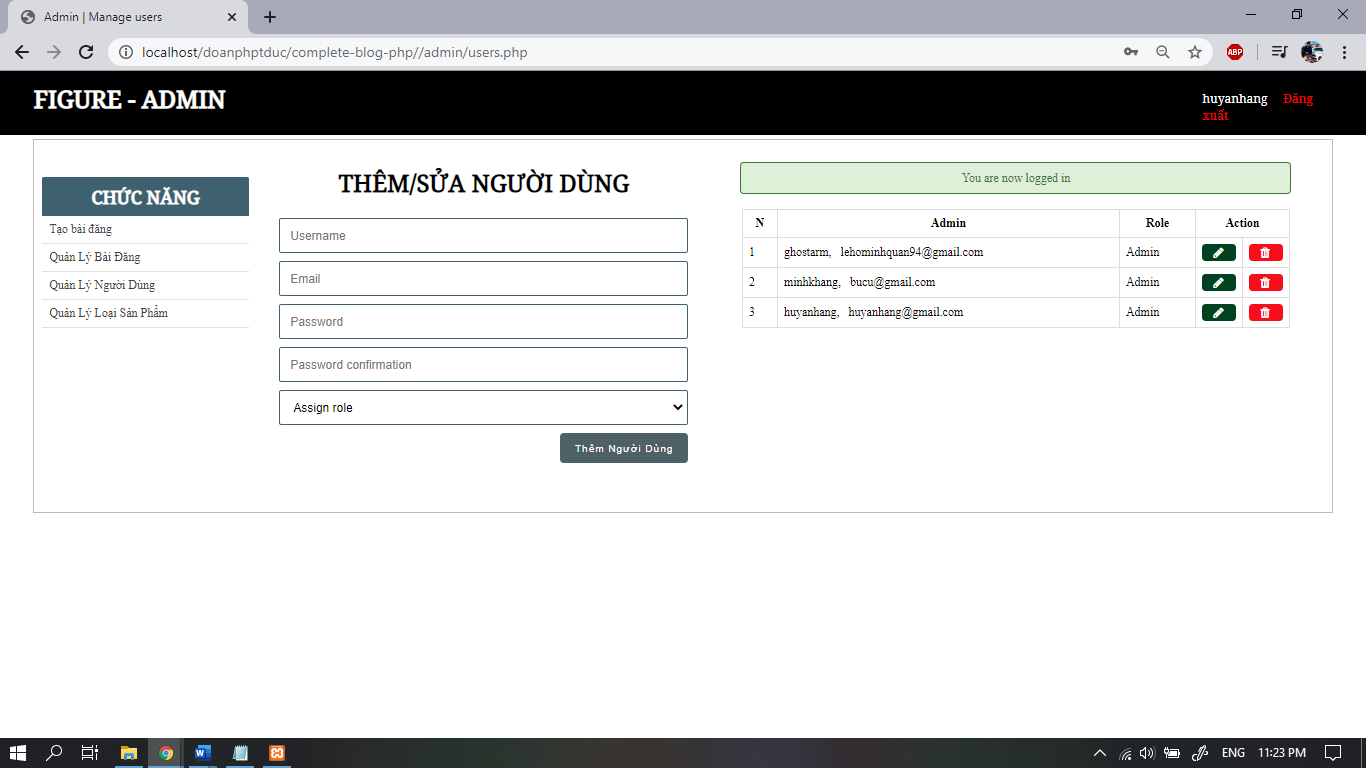
Hình 4: Giao diện chi tiết giản phẩm

### 3.1.5. Màn hình admin (home)



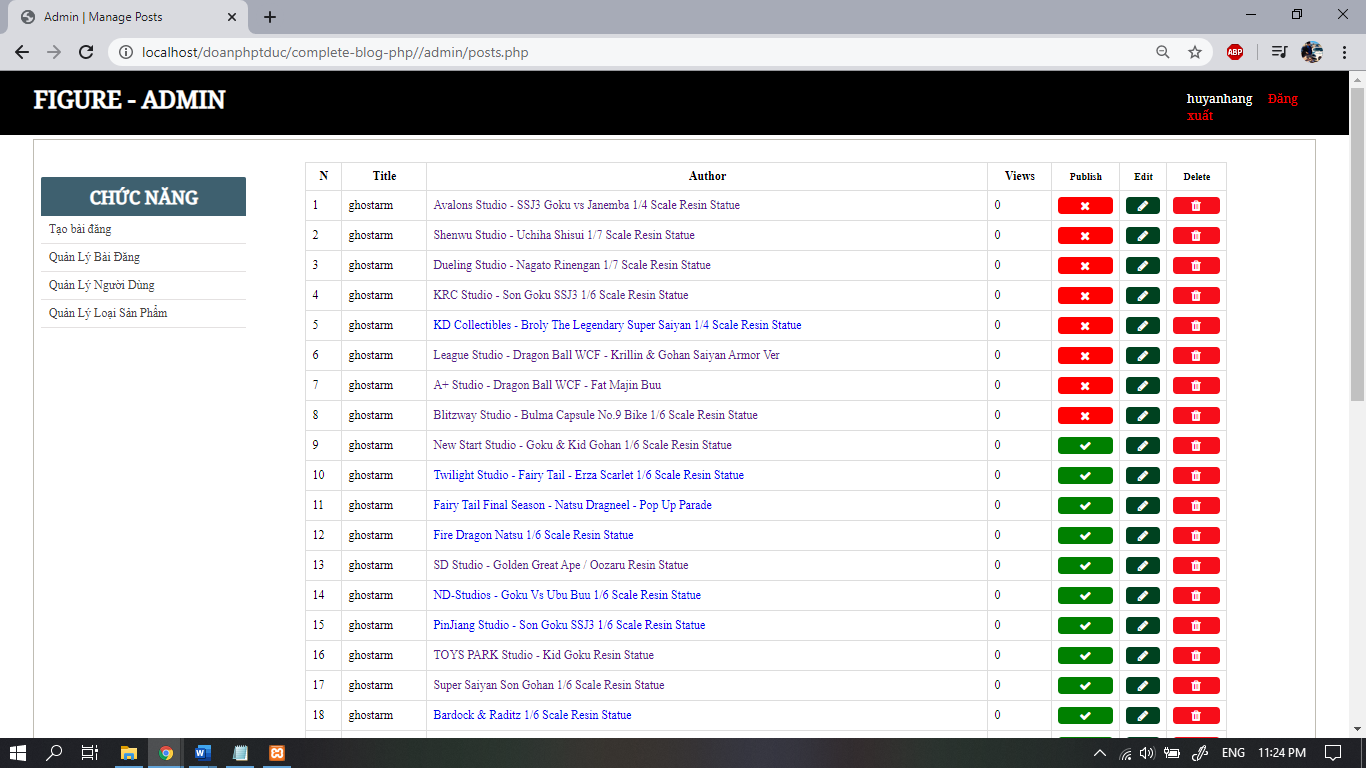
Hình 5: Giao diện trang chủ admin

### 3.1.6. Màn hình admin quản lý người dùng



Hình 6: Giao diện admin quản lý người dùng

### 3.1.6. Màn hình admin quản lý bài viết



Hình 7: Giao diện admin quản lý sản phẩm

## 3.2. Thiết kế giao diện hệ thống

### 3.2.1 Giao diện trang chủ

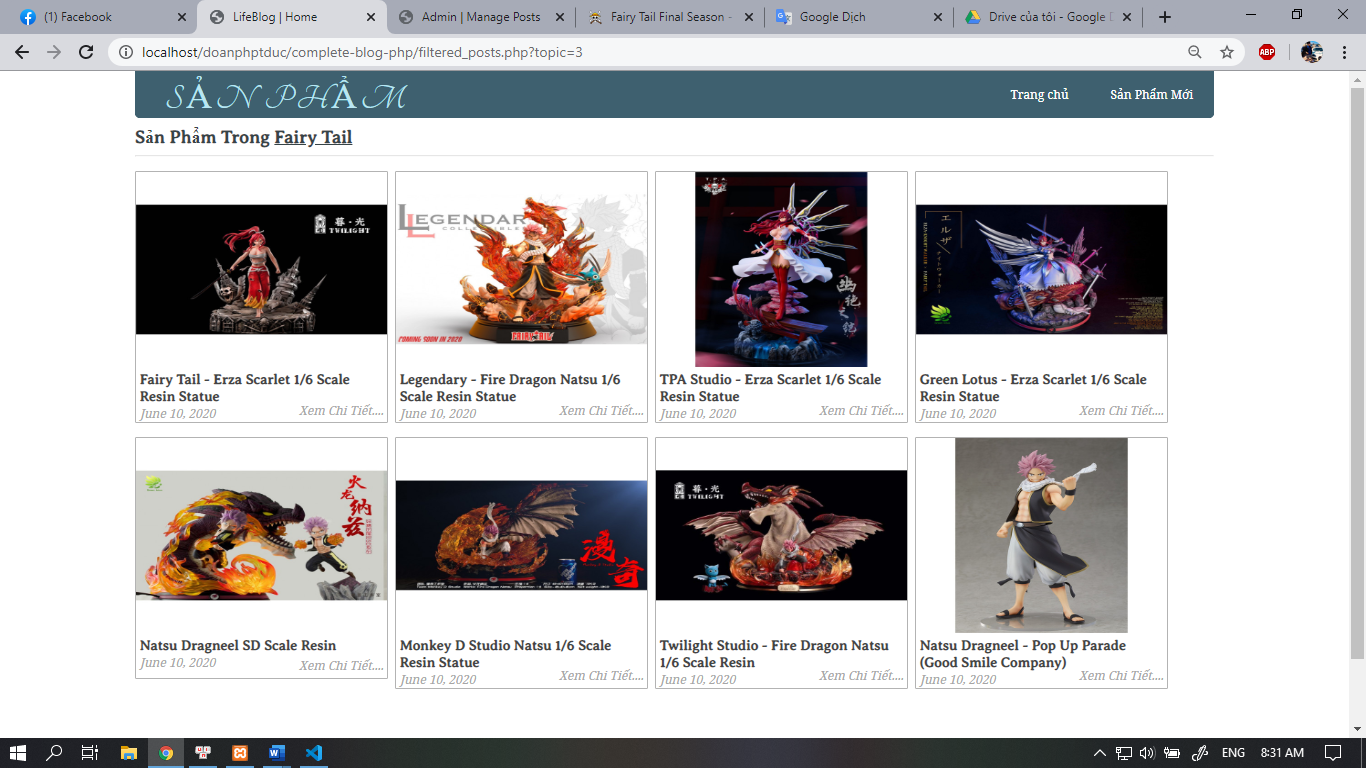
Đây là màn hình xuất hiện đầu tiên khi truy cập vào trang web thể hiện cho người dùng login vào trang admin thể hiện các nút trang chủ, hỗ trợ, giới thiệu. Hiển thị được các danh mục sản phẩm và danh sách các sản phẩm của website.



Hình 8: Giao diện trang chủ

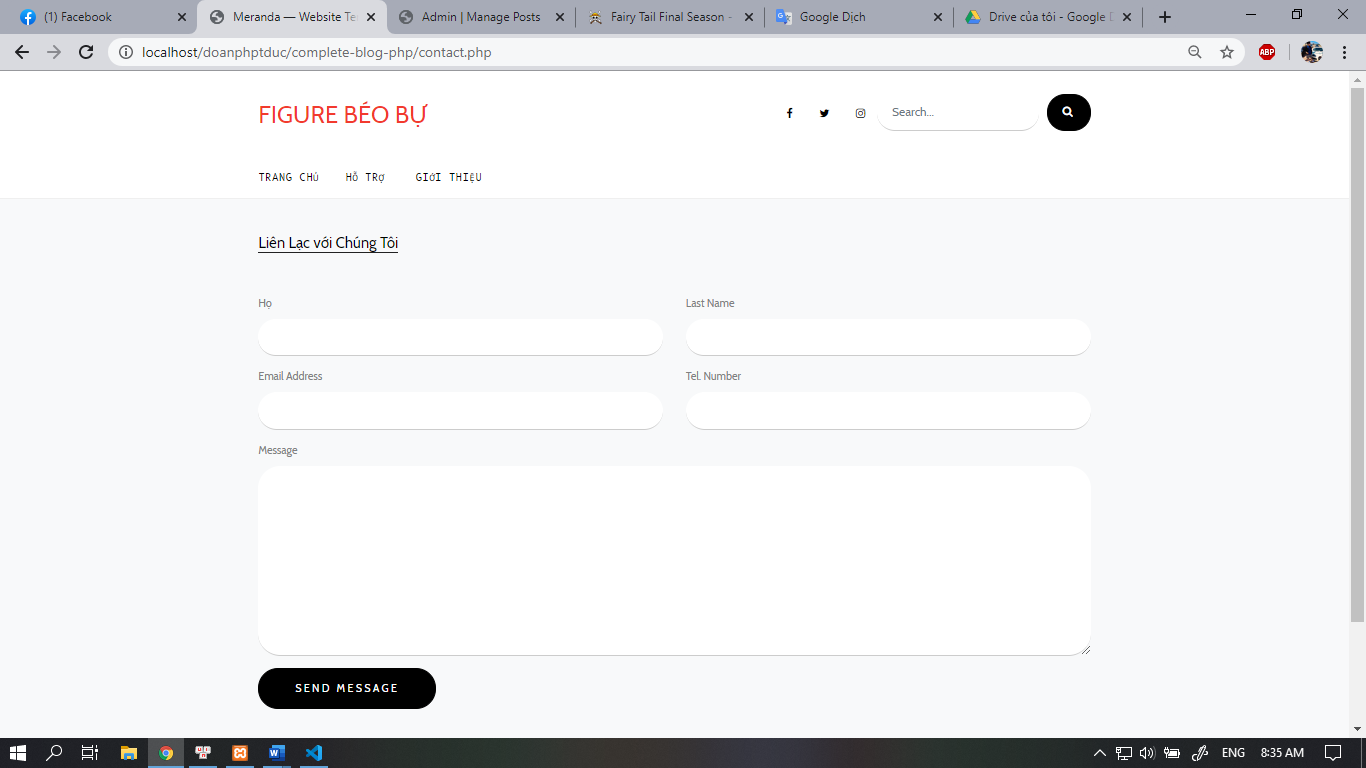
### 3.2.2 Giao diện trang danh sách sản phẩm theo danh mục

Hiển thị đươc các sản phẩm theo từng danh mục.



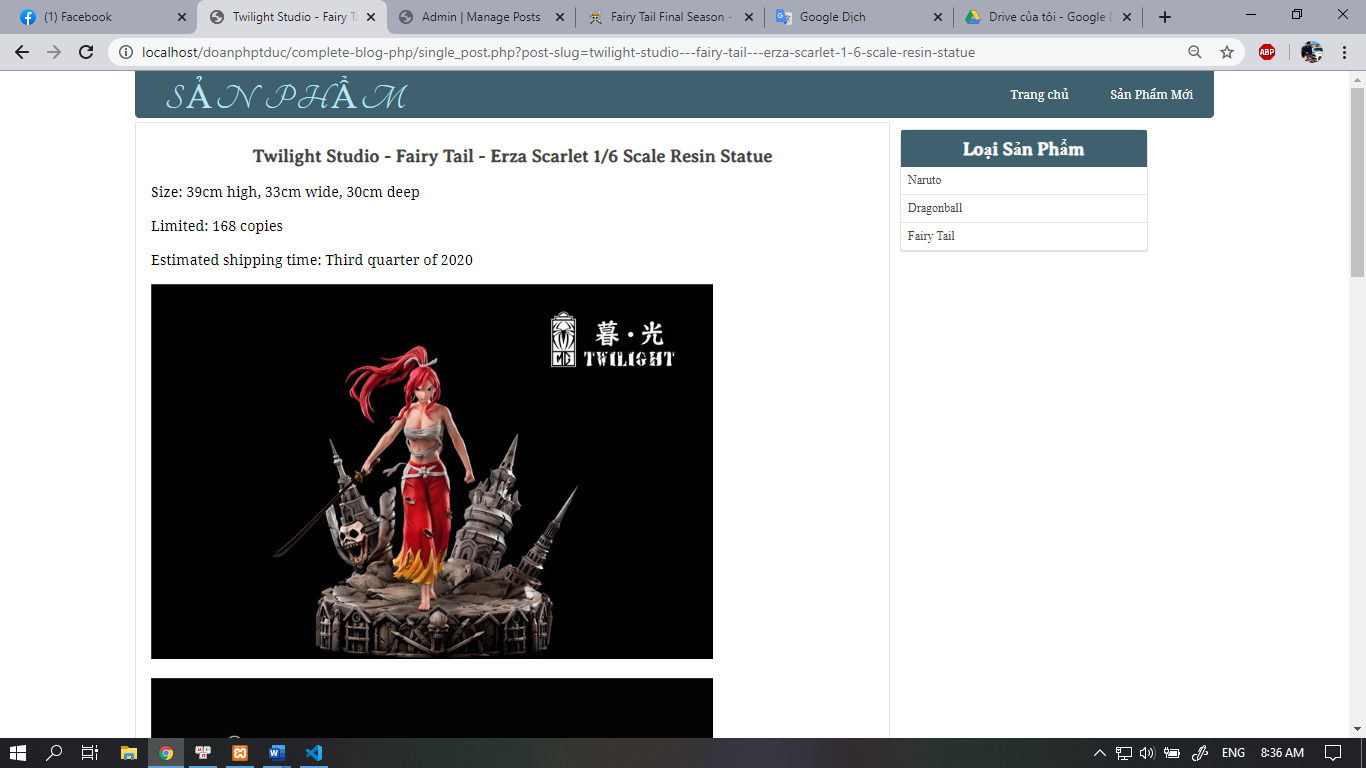
Hình 9: Giao diện trang danh mục

### 3.2.3 Giao diện trang hỗ trợ



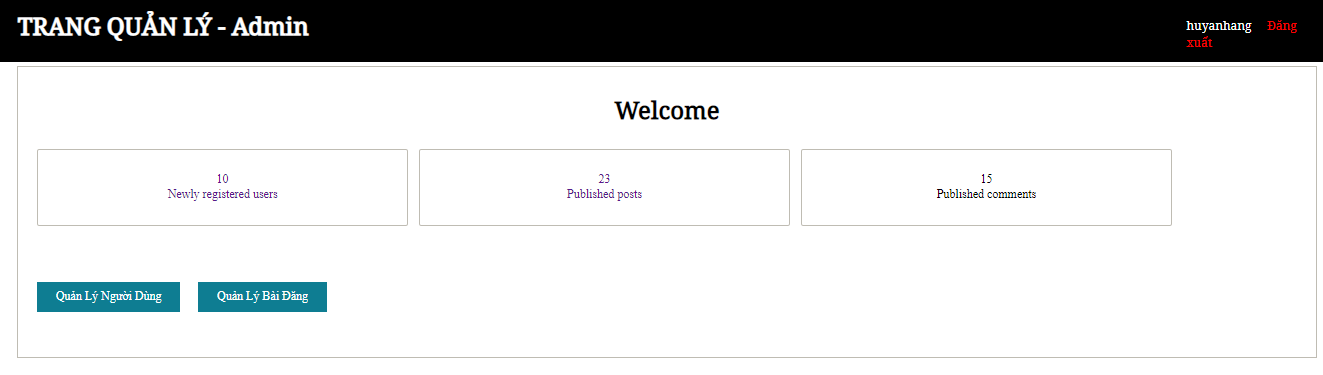
Hình 10: Giao diện trang hỗ trợ

### 3.2.4 Giao diện trang chi tiết sản phẩm



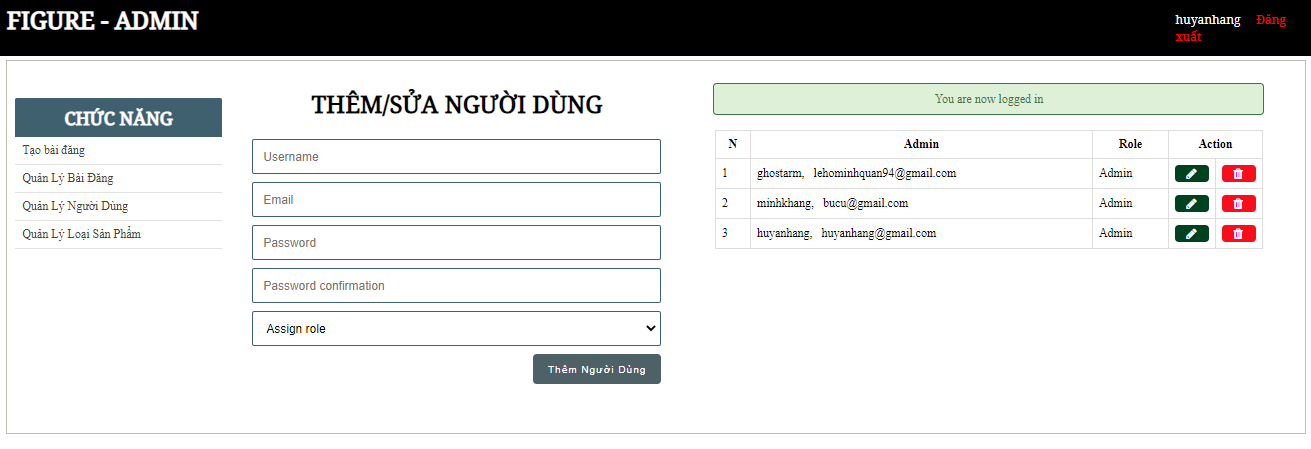
Hình 11: Giao diện trang Chi tiết sản phẩm

### 3.2.5 Giao diện trang chính admin



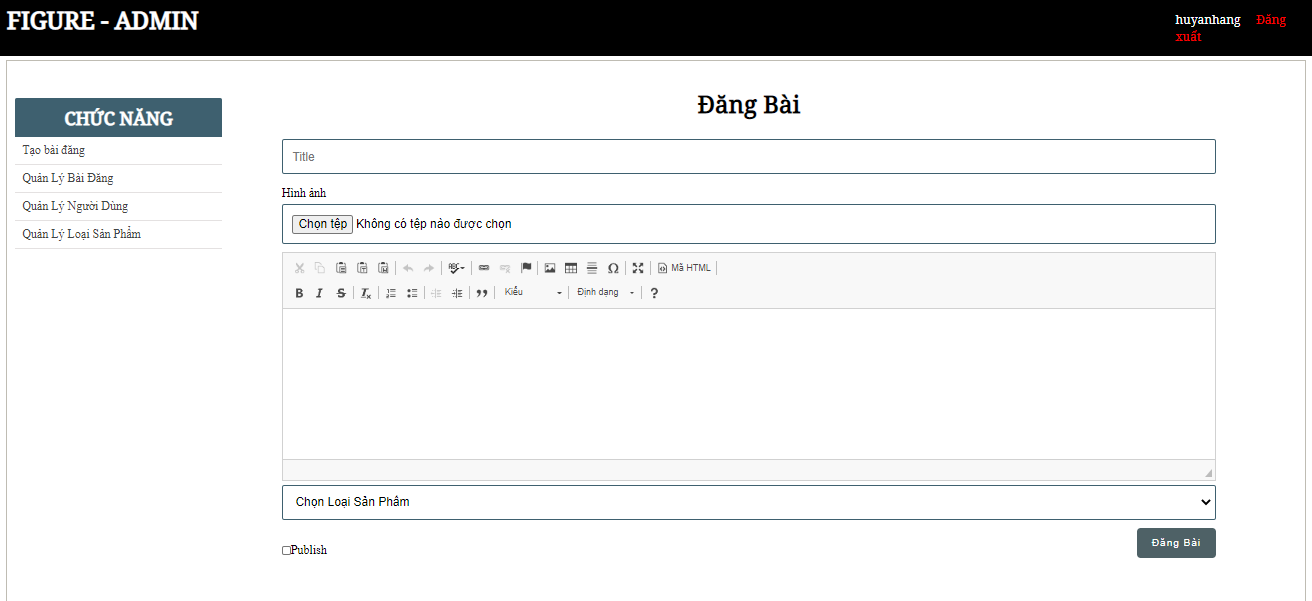
Màn hình giúp admin có thể quản lý được các thông tin trên website gồm: quản lý người dùng và quản lý sản phẩm.

### 3.2.6 Giao diện admin quản lý users

Dùng để tạo bài đăng, quản lý bài đăng, quản lý người dùng, quản lý loại sản phẩm.

### 3.2.7 Giao diện chức năng đăng bài

Dùng để đăng bài gồm các thông tin như: Tilte, hình ảnh, body, public

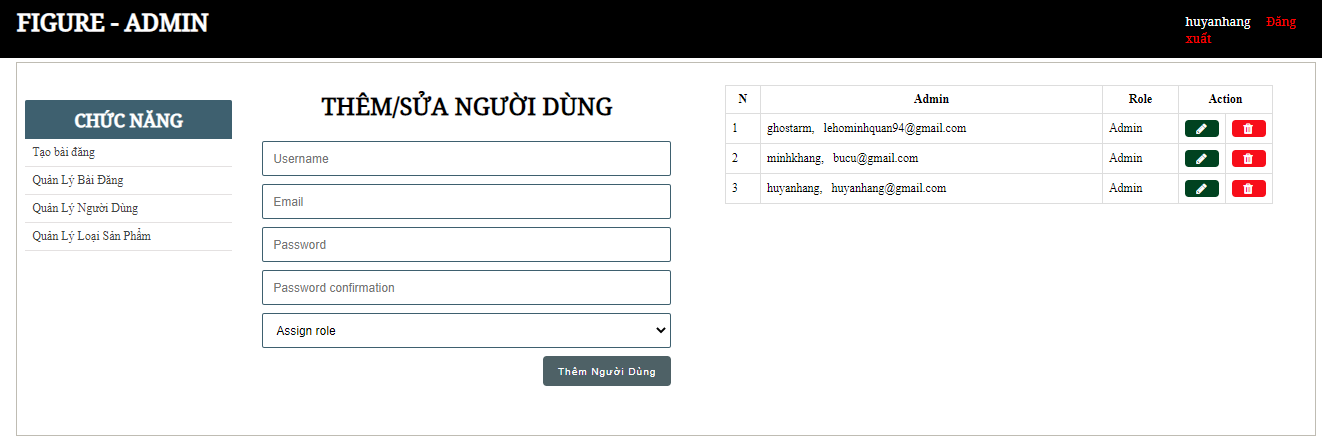


### 3.2.8 Giao diện chức năng đăng bài

Dùng để quản lý các bài đã đăng

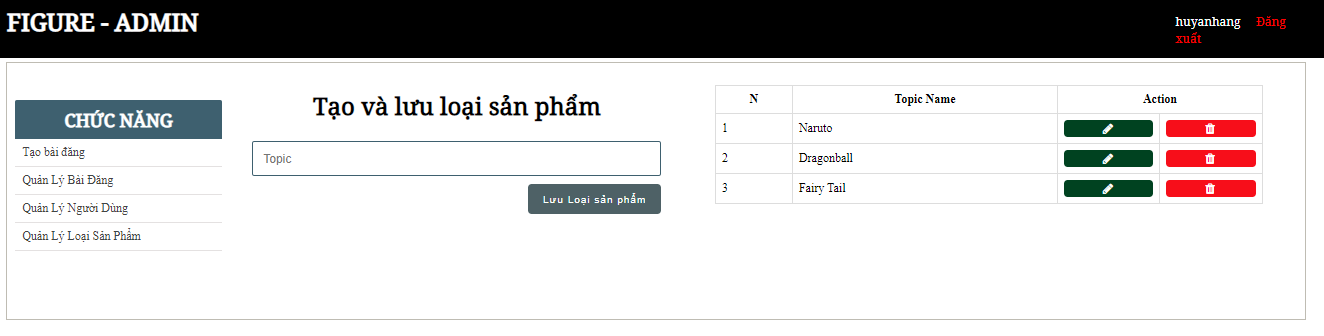
### 3.2.9 Giao diện thêm sửa người dùng

Thêm thông tin người đăng gồm: username, email, password,..

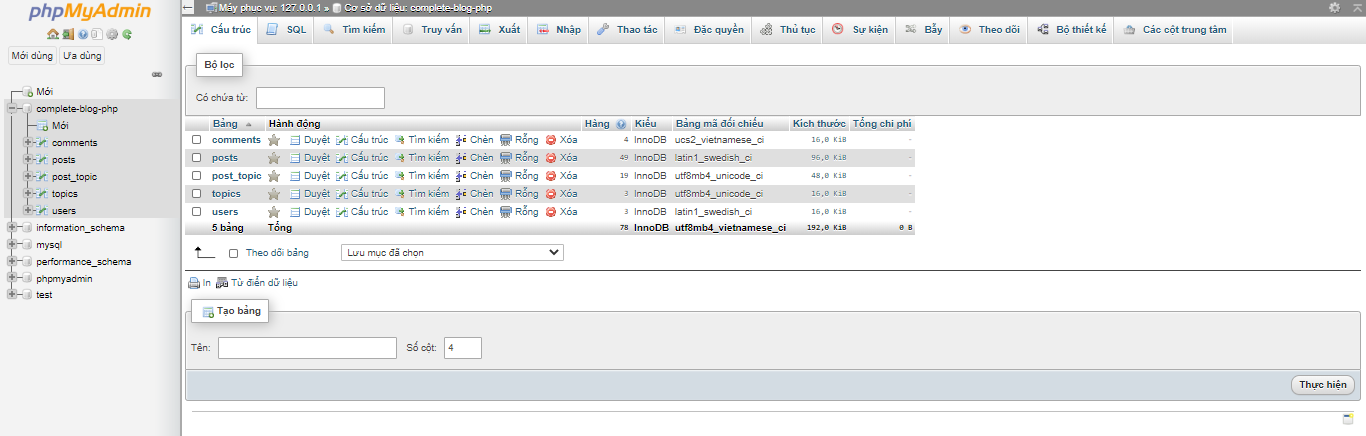


### 3.2.10 Giao diện thêm sửa người dùng

Tạo và lưu các loại sản phẩm



### 3.2.11 Giao diện phpMyAdmin



Hình 12: Giao diện trang phpMyAdmin

## 3.3. Sơ đồ thực thể kết hợp

### 3.3.1 Bảng commet

| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Ý nghĩa** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- |
| 1 | **IdComment** | int(11) | Khóa chính | Mã cmt | Not null |
| 2 | **Ten** | varchar(255) |  | Mã tên |  |
| 3 | **NoiDung** | varchar(255) |  | Nội dung |  |
| 4 | **IdBaiViet** | int(11) |  | Mã bài viết |  |

### 3.3.2 Bảng post

| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Ý nghĩa** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- |
| 1 | **id** | int(11) | Khóa chính | Mã id | Not null |
| 2 | **user\_id Chỉ mục** | int(11) | Khóa ngoại | Mã user |  |
| 3 | **title** | varchar(255) |  | Title |  |
| 4 | **slug** | varchar(255) | Khóa ngoại | Slug |  |
| 5 | **views** | int(11) |  | Xem |  |
| 6 | **Image** | varchar(255) |  | Hình |  |
| 7 | **body** | text |  | Nội dung |  |
| 8 | **published** | tinyint(1)s |  | Công khai |  |
| 9 | **created\_at** | timestamp |  | Tạo |  |
| 10 | **updated\_at** | timestamp |  | Cập nhật |  |

### 3.3.3 Bảng post\_topic

| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Ý nghĩa** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- |
| 1 | **id** | int(11) | Khóa chính | Mã id | Not null |
| 2 | **post\_id** | varchar(255) |  | Bài viết |  |
| 3 | **topic\_id** | varchar(255) |  | Topic |  |

### 3.3.4 Bảng topics

| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Ý nghĩa** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- |
| 1 | **id** | int(11) | Khóa chính | Mã id | Not null |
| 2 | **name** | varchar(255) |  | Tên |  |
| 3 | **slug** | varchar(255) |  | Slugs |  |

### 3.3.5 Bảng users

| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Ý nghĩa** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- |
| 1 | **id** | int(11) | Khóa chính | Mã id | Not null |
| 2 | **username** | varchar(255) |  | Mã user |  |
| 3 | **email** | varchar(255) |  | Title |  |
| 4 | **role** | enum('Author', 'Admin') |  | Slug |  |
| 5 | **password** | varchar(255) |  | Xem |  |
| 6 | **created\_at** | timestamp |  | Hình |  |
| 7 | **updated\_at** | timestamp |  | Nội dung |  |

# CHƯƠNG 4. KẾT LUẬN

## 4.1. Kết quả đạt được

Phần người dùng:

Thiết kế hoàn tất phần giao diện người dùng gồm màn hình đăng nhập, màn hình đăng kí, màn hình home, chi tiết các sản phẩm

Liên kết được các màn hình của website trang chủ sang các trang sản phẩm, chi tiết sản phẩm, trang đăng nhập, đăng kí

Có thể hiển thị hình ảnh lên website

Có thể comment vào sản phẩm.

Phần quản trị:

Giao diện thân thiện với quản trị viên

Có thể thêm xóa sửa bài viết

Thêm xóa, chỉnh sửa quyền tài khoản.

Thêm, xóa, sửa phần topic sản phẩm.

## 4.2. Đánh giá phần mềm

### 4.2.1. Ưu điểm

Hiển thị được hình ảnh lên web site, các tương tác, các nút có thể thực hiện được.

### 4.2.2. Nhược điểm

Vẫn đang trong quá trình phát triển vì vậy các chức năng trong website chưa thể thực hiện được.

Chức năng sửa còn một vài lỗi.

Chưa được phân trang.

Hình ảnh hiển thị đôi khi lỗi

## 4.3. Hướng phát triển và mở rộng đề tài

Người dùng có thể đặt hàng và thanh toán trực tiếp.

Điều chỉnh lại giao diện, phân trang, sữa các lỗi và hoàn thiện chức năng.

# Tài liệu tham khảo

Source code free trên mạng.

Tài liệu học tập của thầy Nguyễn Hồ Minh Đức