**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

***Đề tài:***

**XÂY DỰNG WEBSITE BÁN ĐIỆN THOẠI ONLINE**

**Giảng viên hướng dẫn**: TS. Hoàng Anh

**Sinh viên thực hiện:** Vũ Hữu Trùng Dương

**MSSV:** 0750080056

**Lớp:** 07\_ĐHCNTT\_2

**Khoá:** 2018-2022

**TP.HỒ CHÍ MINH THÁNG 03 NĂM 2023**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

***Đề tài:***

**XÂY DỰNG WEBSITE BÁN ĐIỆN THOẠI ONLINE**

**Giảng viên hướng dẫn**: TS. Hoàng Anh

**Sinh viên thực hiện:** Vũ Hữu Trùng Dương

**MSSV:** 0750080056

**Lớp:** 07\_ĐHCNTT\_2

**Khoá:** 2018-2022

**TP.HỒ CHÍ MINH THÁNG 03 NĂM 2023**

**Lời Cảm Ơn**

Sau thời gian học tập và nghiên cứu đề tài **“Xây dựng hệ thống website Shop bán điện thoại online”** em đã không ngừng được củng cố, mở mang thêm nhiều kiến thức và tầm hiểu biết về lĩnh vực Công nghệ Thông tin qua sự dạy dỗ, đào tạo tận tình của các thầy cô bộ môn trong khoa công nghệ thông tin và sự học hỏi lẫn nhau giữa bạn bè cùng khóa.

Thông qua bản báo cáo này, em xin được gửi lời chân thành cảm ơn đến Ban Giám Hiệu nhà Trường và cùng các thầy cô trong khoa đã tạo những điều kiện tốt nhất cho chúng em trong suốt thời gian học tập và sinh hoạt tại trường; xin chân thành cảm ơn thầy TS. Hoàng Anh giáo viên phụ trách hướng dẫn em tận tình trong suốt quá trình thực hiện đề tài này.

Em xin chân thành cảm ơn quý thầy cô đã nhiệt tình giảng dạy, truyền đạt những kiến thức, kinh nghiệm quý báu và hơn hết là tình cảm mà quý thầy cô đã dành cho em trong suốt quá trình giảng dạy.

Mặc dù cố gắng hoàn thành đồ án trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của thầy, cô và tất cả các bạn để đề tài ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

**TP. Hồ Chí Minh, ngày….. tháng….năm 2023**

**SINH VIÊN THỰC HIỆN**

**Vũ Hữu Trùng Dương**

**LỜI CAM ĐOAN**

Em xin cam đoan đây là công trình của bản thân. Các nội dung nghiên cứu và kết quả trong đề tài này là trung thực, chưa được ai công bố trong bất cứ công trình nào. Nếu có bất kỳ sự gian lận nào, em xin chịu trách nhiệm trước Hội đồng cũng như kết quả khóa luận của mình.

**SINH VIÊN THỰC HIỆN**

**Vũ Hữu Trùng Dương**

NHẬN XÉT GIÁO VIÊN HƯỚNG DẪN

**Kết luận:**

**TP. Hồ Chí Minh, ngày… tháng….năm 2023**

**GIẢNG VIÊN HƯỚNG DẪN**

NHẬN XÉT GIÁO VIÊN PHẢN BIỆN

**Kết luận:**

**TP. Hồ Chí Minh, ngày… tháng….năm 2023**

**GIẢNG VIÊN PHẢN BIỆN**

**LỊCH LÀM VIỆC**

**Giảng viên Hướng dẫn:** TS. Hoàng Anh

**Sinh viên**: Vũ Hữu Trùng Dương

|  |  |  |
| --- | --- | --- |
| Thời gian | Nội dung công việc được giao | Nhận xét GVHD  *(Ký tên)* |
| 01/10/2022-  04/10/2022 | Xác định yêu cầu nghiệp vụ của website |  |
| 05/10/2022-  08/10/2022 | Phân tích yêu cầu về các chức năng, giao diện |  |
| 09/10/2022-  13/10/2022 | Xây dựng cơ sở dữ liệu |  |
| 14/10/2022-  25/10/2022 | Thiết kế giao diện website |  |
| 26/10/2022-  19/11/2022 | Xây dựng các chức năng của hệ thống |  |
| 20/11/2022-  26/11/2022 | Kiểm thử các chức năng của hệ thống |  |
| 26/11/2022-  02/12/2022 | Làm báo cáo |  |

**TÓM TẮT NỘI DUNG**

* Website “Bán điện thoại online” đã giải quyết được các vấn đề thiết yếu của một hệ thống website bán hàng của cửa hàng một cách hiệu quả, tiện lợi như: quản lý thông tin sản phẩm, quản lý đơn hàng, quản lý thông tin liên hệ…
* Website đã đáp ứng được nhu cầu cơ bản của một trang web bán hàng trực tuyến của một cửa hàng.
* Website sử dụng ngôn ngữ lập trình HTML, CSS, JavaScript và framework JS là ReactJs để tạo giao diện và sử dụng công nghệ Nodejs trên nền tảng framework Express với cơ sở dữ liệu Mongodb đã đáp ứng được các chức năng sau:
  + Quản lý người dùng
  + Quản lý danh mục sản phẩm
  + Quản lý sản phẩm
  + Quản lý tin tức
  + Quản lý đơn hàng
  + Quản lý giỏ hàng
  + Quản lý đặt, mua hàng
  + Đăng ký / Đăng nhập
* Website vẫn đang trong giai đoạn hoàn thiện, vẫn còn một số tính năng chưa được tối ưu.
* Website cơ bản đã đáp ứng được các yêu cầu của người dùng, nhưng vẫn cần phải hoàn thiện thêm một số chức năng để đem lại trải nghiệm người dùng một cách hiệu quả và tiện lợi nhất.

MỤC LỤC

[**CHƯƠNG 1: TỔNG QUAN TÀI LIỆU 1**](#_Toc133269866)

[1.1 Tổng quan đề tài: 1](#_Toc133269867)

[1.2 Tổng quan về một số ngôn ngữ lập trình giao diện: 2](#_Toc133269868)

[1.3 Tổng quan về ngôn ngữ lập trình NodeJS 5](#_Toc133269869)

[1.4. Tổng quan về nền tảng ReactJs: 7](#_Toc133269870)

[1.5. Tổng quan về mô hình MVC: 10](#_Toc133269871)

[1.6. Tổng quan về Express framework: 13](#_Toc133269872)

[1.7. Tổng quan về cơ sở dữ liệu Mongodb: 14](#_Toc133269873)

[**CHƯƠNG 2: PHƯƠNG PHÁP THỰC HIỆN 20**](#_Toc133269874)

[2.1. Xác định yêu cầu: 20](#_Toc133269875)

[2.2. Phân tích bài toán 20](#_Toc133269876)

[2.3. lược đồ Use-case 22](#_Toc133269877)

[2.4. Thiết kế lược đồ activity diagram 34](#_Toc133269878)

[2.5. Thiết kế database cơ sở dữ liệu 41](#_Toc133269879)

[**CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM 43**](#_Toc133269880)

[3.1. Hướng dẫn cài đặt 43](#_Toc133269881)

[3.2 Kết quả đạt được: 49](#_Toc133269882)

[**CHƯƠNG 4: KẾT LUẬN 59**](#_Toc133269883)

[4.1. Kết quả đạt được 59](#_Toc133269884)

[4.2. Những hạn chế và định hướng 59](#_Toc133269885)

[**TÀI LIỆU THAM KHẢO 60**](#_Toc133269886)

**DANH MỤC SƠ ĐỒ, HÌNH**

Hình 1.1 Giới thiệu HTML……………………………………………………..3

Hình 1.1 Giới thiệu CSS………………………………………………………..3

Hình 1.3 Giới thiệu javaScript…………………………………………..……...4

Hình 1.4 Liên kết HTML, CSS, JavaScript…………………………….………4

Hình 1.5 ngôn ngữ lập trình NodeJs......................................................................5

Hình 1.6 Giới thiệu mô hình MVC…………………………………….……….8.

Hình 1.7 Giới thiệu Mongodb...………………………………………………...14

Hình 1.8 Các thành phần chính trong Mongodb...…………………..………….14

Hình 2.1 Lược đồ use-case quản trị viên…………………………….………….19

Hình 2.2 Lược đồ use-case khách hàng thành viên…………………….……….20

Hình 2.3 Lược đồ use-case khách vãng lai………………………………..….…20

Hình 2.4 Lược đồ chức năng đăng nhập…………………………………..….…32

Hình 2.5 Lược đồ chức năng đăng kí thành viên ……………………………….33

Hình 2.6 Lược đồ chức năng quản lý sản phẩm ………………………………...33

Hình 2.7 Lược đồ chức năng quản lý danh mục sản phẩm……………………...34

Hình 2.8 Lược đồ chức năng quản lý bài viết……………………………….…..34

Hình 2.9 Lược đồ chức năng quản lý đơn hàng …………………..………….…35

Hình 2.10 Lược đồ chức năng quản lý quyền truy cập …………………………35

Hình 2.11 Lược đồ chức năng quản lý tài khoản quản trị viên …………………36

Hình 2.12 Lược đồ chức năng quản lý giỏ hàng ……………………………......36

Hình 2.13 Lược đồ chức năng đặt hàng ………………………………………...37

Hình 2.14 Lược đồ chức năng thay đổi mật khẩu ………………………………37

Hình 3.7 giao diện trang chủ …………………………………..………………...46

Hình 3.8 Giao diện tất cả sản phẩm……………………….. …………………….47

Hình 3.9 Giao diện sản phẩm là điện thoại…… ………………………….……..47

Hình 3.10 giao diện sản phẩm là laptop…..………………………………………48

Hình 3.11 giao diện trang chi tiết sản phẩm …………………………………….48

Hình 3.12 giao diện giỏ hàng ………………………………………………….....49

Hình 3.13 giao diện trang giỏ hàng ……………………………………..…….....50

Hình 3.14 giao diện trang đăng nhập tài khoản ……………………………….....50

Hình 3.15 giao diện trang đăng kí tài khoản ……………………………………..51

Hình 3.16 giao diện trang thông tin tài khoản khách hàng ……………………....51

Hình 3.17 giao diện trang thanh toán ……………………………………………52

Hình 3.18 giao diện trang đăng nhập của admin…………………………………52

Hình 3.19 giao diện trang chủ admin ………………………………………….. 53

Hình 3.20 giao diện trang quản lý quyền truy cập ………………………………53

Hình 3.21 giao diện trang quản lý danh mục sản phẩm …………………………54

Hình 3.22 giao diện trang quản lý đơn hàng……………………………………..54

Hình 3.23 giao diện trang quản lý danh sách sản phẩm ……………………….55

**LỜI NÓI ĐẦU**

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hoá được xem là một trong những yếu tố mang tính quyết định trong hoạt động của tổ chức, cũng như các công ty. Nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.

Cùng với sự phát triển không ngừng về kỹ thuật máy tính và mạng điện tử, mạng internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên là một công cụ không thể thiếu là nền tảng chính cho sự truyền tải trao đổi thông tin trên toàn cầu.

Trong hoạt động sản xuất kinh doanh giờ đây, thương mại điện tử khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của doanh nghiệp. đối với một cửa hàng hay shop, việc quảng bá và giới thiệu đến khách hàng các sản phẩm mới đáp ứng được nhu cầu của khách hàng sẽ là cần thiết. vậy phải quảng bá thế nào: đó là xây dựng được một website cho cửa hàng quảng bá tất cả các sản phẩm mà người hoặc doanh nghiệp muốn bán.

Vì vậy, em đã thực hiện đồ án “ XÂY DỰNG WEBSITE BÁN ĐIỆN THOẠI ONLINE ”.

Người chủ cửa hàng có thể đưa các sản phẩm lên web và quản lý thông tin bằng website đó. Khách hàng có thể đặt mua hàng trên website mà không cần phải đến cửa hàng. Sẽ có nhân viên giao hàng tận nơi.

Với sự hướng dẫn tận tình của thầy Hoàng Anh em đã hoàn thành được đồ án này. Tuy đã cố gắng tìm hiểu, phân tích và thiết kế hệ thống nhưng chắc rằng không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và góp ý của quý thầy cô ban cố vấn. em xin chân thành cảm ơn.

# CHƯƠNG 1: TỔNG QUAN TÀI LIỆU

## **Tổng quan đề tài:**

* + 1. **Lý do chọn đề tài:**

Công nghệ thông tin không ngừng phát triển một cách mạnh mẽ và hiện đại. Các ứng dụng của công nghệ thông tin ngày càng đi sâu vào đời sống con người trở thành một bộ phận không thể thiếu của thế giới. Với xu thế toàn cầu hóa nền kinh tế thế giới, đặc biệt là nhu cầu trao đổi hàng hóa của con người ngày càng tăng về cả số lượng và chất lượng, nhu cầu sử dụng Internet ngày càng nhiều và các hình thức kinh doanh trên mạng ngày càng đa dạng và đã trở thành công cụ không thể thiếu.

Chính vì vậy đề tài: “Xây dựng website bán điện thoại online” được ra đời giúp cửa hàng bán điện thoại cụ thể hơn là các hãng điện thoại thông minh có thể áp dụng phương pháp bán hàng qua mạng (thương mại điện tử) để khách hàng dễ dàng tìm kiếm thông tin và không cần tốn nhiều thời gian và chi phí, cũng như có thể dễ dàng tiếp cận đối với nhiều loại khách hàng khác nhau.

* + 1. **Mục tiêu đề ra**

Xây dựng được trang web quản lý và bán ra những hãng điện thoại được ưa chuộng trên thị trường thông qua việc trao đổi trên thị trường thương mại điện tử.

* + 1. **Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của đề tài là các cửa hàng bán lẻ trên toàn quốc nói chung và các cửa hàng bán hàng trực tuyến nói riêng.

Phạm vi nghiên cứu là hệ thống website bán hàng và quản lí cửa hàng.

* + 1. **Phương pháp nghiên cứu**
* **Về lý thuyết:**

+ Tìm hiểu HTML, CSS, JavaScript và ngôn ngữ lập trình NodeJs

+ Tìm hiểu về framework ReactJS

+ Tìm hiểu về kỹ thuật lập trình, cách thực hoạt động và các đối tượng trong ứng dụng web theo mô hình MVC.

+ Tìm hiểu về cơ sở dữ liệu Mongodb

+ Tìm hiểu Express framework

* **Về thực nghiệm**

**+** Tìm hiểu về quy trình nghiệp vụ của các trang thương mại điện tử

+ Phân tích yêu cầu theo hướng dẫn của giảng viên hướng dẫn

+ Phân tích và thiết kế sản phẩm

+ Phân chia các module chức năng theo yêu cầu nghiệp vụ của trang thương mại điện tử

+ kiểm thử các chức năng hoàn thành và đưa vào hệ thống.

## **Tổng quan về một số ngôn ngữ lập trình giao diện:**

**1.2.1. Giới thiệu về HTML**

HTML (viết tắt của từ Hypertext Markup Language, hay là “Ngôn ngữ Đánh dấu Siêu văn bản”) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như CSS và các ngôn ngữ kịch bản giống như JavaScript.

Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện. HTML mô tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.

Các phần tử HTML là các khối xây dựng của các trang HTML. Với cấu trúc HTML, hình ảnh và các đối tượng khác như biểu mẫu tương tác có thể được nhúng vào trang được hiển thị. HTML cung cấp một phương tiện để tạo tài liệu có cấu trúc bằng cách biểu thị ngữ nghĩa cấu trúc cho văn bản như headings, paragraphs, lists,

links, quotes và các mục khác. Các phần tử HTML được phân định bằng các tags, được viết bằng dấu ngoặc nhọn.

Các tag khác như bao quanh và cung cấp thông tin về văn bản tài liệu và có thể bao gồm các thẻ khác làm phần tử phụ. Các trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để diễn giải nội dung của trang.



**Hình 1.1 HTML**

**1.2.2. CSS**

CSS là từ viết tắt của cụm từ Cascading Style Sheets, ngôn ngữ được sử dụng để tạo nên phong cách cho website. [CSS](https://quantrimang.com/hoc-css) mô tả cách các phần tử HTML hiển thị trên màn hình và các phương tiện khác.

CSS đóng vai trò như một công cụ giúp chúng ta thêm vào những thay đổi về mặt hình thức như đổi bố cục, màu sắc, font chữ,…

CSS hoạt động bằng cách khoanh vùng chọn dựa vào tên một thẻ HTML, ID hay Class. Từ đó, áp dụng những thuộc tính cần thay đổi lên vùng được chọn.

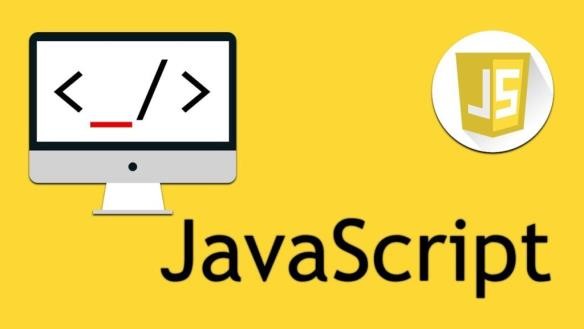


**Hình 1.2 CSS**

**1.2.3. JavaScript:**

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa. Javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... trên máy tính lẫn điện thoại.

Nhiệm vụ của Javascript là xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm / xóa / sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng.



**Hình 1.3 JavaScript**

**1.2.4. HTML, CSS, JavaScript liên quan thế nào với nhau?**

 Mặc dù HTML là một ngôn ngữ mạnh mẽ, nó vẫn không đủ tính năng để xây dựng một website chuyên nghiệp và hoàn toàn responsive. Chúng ta chỉ có thể dùng HTML để thêm element dạng văn bản và tạo ra giao diện cấu trúc cho nội dung.

**Hình 1.4 Liên kết HTML, CSS, JavaScript**

HTML tương thích hoàn toàn với 2 ngôn ngữ frontend là: CSS (Cascading Style Sheets), và JavaScript. Cùng với nhau, những ngôn ngữ này có thể kết hợp với nhau để tăng trải nghiệm người dùng và thiết lập các chức năng cao cấp khác:

* CSS chịu trách nhiệm cho việc thiết kế, tạo kiểu backgrounds, màu sắc, bố cục, phân cách hiệu ứng
* javaScript giúp tạo ra những chức năng động như : slider, pop-up và thư viện ảnh…

Hãy xem HTML giống như là một người, CMS giống như là quần áo, JavaScript là cách ứng xử và hành vi của người đó.

## **Tổng quan về ngôn ngữ lập trình NodeJS**

**1.3.1. khái niệm**

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript giúp chúng ta có thể xây dựng được các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp một cách nhanh chóng và dễ dàng mở rộng.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Node.js được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ..

Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao.

Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.



**Hình 1.5 ngôn ngữ lập trình NodeJs**

**1.3.2. Đặc điểm**

Các ứng dụng Nodejs được viết bằng javascript, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: “Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”

Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.

Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Các bạn có thể thấy cộng đồng Nodejs lớn như thế nào tại đây, các package đều hoàn toàn free.

Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

NodeJS được sử dụng để xây dựng rất nhiều loại ứng dụng khác nhau, trong đó phổ biến nhất gồm có:

Ứng dụng trò chuyện trong thời gian thực: Nhờ vào cấu trúc không đồng bộ đơn luồng, Node.JS rất thích hợp cho mục đích xử lý giao tiếp trong thời gian thực. Nền tảng này có thể dễ dàng mở rộng quy mô và thường dùng để tạo ra các chatbot. Bên cạnh đó, các tính năng liên quan đến ứng dụng trò chuyện như: chat nhiều người, thông báo đẩy,… cũng có thể dễ dàng được bổ sung nhờ NodeJS.

Internet of Things (IoT): Các ứng dụng IoT thường bao gồm nhiều bộ cảm biến phức tạp để gửi những phần dữ liệu nhỏ. Node.JS là một lựa chọn lý tưởng để xử lý các yêu cầu đồng thời này với tốc độ cực nhanh.

Truyền dữ liệu: Netflix là một trong số những công ty lớn trên thế giới chuyên sử dụng Node.JS cho mục đích truyền dữ liệu. Sở dĩ vì đây là một nền tảng nhẹ và cực nhanh, đồng thời còn cung cấp một API chuyên dùng để stream.

Các SPA (Single-page application) phức tạp: Trong SPA, toàn bộ ứng dụng được load vào trong một trang duy nhất, do đó sẽ có một số request được thực hiện trong nền. Vòng lặp sự kiện (event loop) của Node.JS cho phép xử lý các request theo hướng non-blocking.

Các ứng dụng REST dựa trên API: JavaScript được sử dụng trong cả frontend lẫn backend của trang. Do đó một server có thể dễ dàng giao tiếp với frontend qua REST API bằng Node.js. Bên cạnh đó, Node.JS còn cung cấp nhiều package như Express.js hay Koa để việc xây dựng ứng dụng web trở nên dễ dàng hơn bao giờ hết.

## **1.4. Tổng quan về nền tảng ReactJs:**

**1.4.1. Khái niệm về ReactJs:**

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,… Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng.

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện performance website thì những tính năng đặc biệt dưới đây có thể là lý do khiến bạn “chốt sale” với ReactJS và bắt đầu tìm hiểu nó từ bây giờ:

Phù hợp với đa dạng thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì bạn không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML và nó đã cung cấp cho bạn đủ loại “đồ chơi” để bạn có thể dùng cho nhiều trường hợp.

Tái sử dụng các Component: Nếu bạn xây dựng các Component đủ tốt, đủ flexible để có thể thoả các “yêu cầu” của nhiều dự án khác nhau, bạn chỉ tốn thời gian xây dựng ban đầu và sử dụng lại hầu như toàn bộ ở các dự án sau. Không chỉ riêng mỗi ReactJS mà các framework hiện nay cũng đều cho phép chúng ta thực hiện điều đó, ví dụ Flutter chẳng hạn.

Có thể sử dụng cho cả Mobile application: Hầu hết chúng ta đều biết rằng ReactJS được sử dụng cho việc lập trình website, nhưng thực chất nó được sinh ra không chỉ làm mỗi đều đó. Nếu bạn cần phát triển thêm ứng dụng Mobile, thì hãy sử dụng thêm React Native – một framework khác được phát triển cũng chính Facebook, bạn có thể dễ dàng “chia sẻ” các Component hoặc sử dung lại các Business Logic trong ứng dụng.

Thân thiện với SEO: SEO là một phần không thể thiếu để đưa thông tin website của bạn lên top đầu tìm kiếm của Google. Bản chất ReactJS là một thư viện JavaScript, Google Search Engine hiện nay đã crawl và index được code JavaScript, tuy nhiên bạn cũng cần thêm một vài thư viện khác để hỗ trợ điều này nhé!

Debug dễ dàng: Facebook đã phát hành 1 Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng. Điều đó giúp tăng tốc quá trình release sản phẩm cung như quá trình coding của bạn.

Công cụ phát triển web hot nhất hiện nay: Nếu bạn nhìn vào số liệu thống kê từ Google Trend ở Việt Nam ở hình bên dưới, dạo lướt qua các trang tuyển dụng hàng đầu ở Việt Nam như Topdev, Itviec, v.v bạn sẽ thấy số lượng tuyển dụng cho vị trí React Developer là cực kỳ lớn cùng với mức lương vô cùng hấp dẫn và độ phổ biến hiện tại của ReactJS trên thị trường Việt Nam là như thế nào.

Nếu bạn không sử dụng ReactJS (và JSX), website của bạn sẽ sử dụng HTML để cập nhật lại cây DOM cho chính bản nó (quá trình thay đổi diễn ra tự nhiên trên trang mà người dùng không cần phải tải lại trang), cách làm này sẽ ổn cho các website nhỏ, đơn giản, static website. Nhưng đối với các website lớn, đặc biệt là những website thiên về xử lý các tương tác của người dùng nhiều, điều này sẽ làm ảnh hưởng performance website cực kỳ nghiêm trọng bởi vì toàn bộ cây DOM phải reload lại mỗi lần người dùng nhấn vào tính năng yêu cầu phải tải lại trang).

Tuy nhiên, nếu bạn sử dụng JSX thì bạn sẽ giúp cây DOM cập nhật cho chính DOM đó, ReactJS đã khởi tạo một thứ gọi là Virtual DOM (DOM ảo). Virtual DOM (bản chất của nó theo đúng tên gọi) là bản copy của DOM thật trên trang đó, và ReactJS sử dụng bản copy đó để tìm kiếm đúng phần mà DOM thật cần cập nhật khi bất kỳ một sự kiện nào đó khiến thành phần trong nó thay đổi (chẳng hạn như user nhấn vào một nút bất kỳ).

Ví dụ, khi người dùng bình luận vào khung comment vào bất kỳ bài Blog nào trên website của bạn và nhấn “Enter”. Dĩ nhiên, người dùng của bạn sẽ cần phải thấy được bình luận của mình đã được thêm vào danh sách bình luận. Giả sử trong trường hợp không sử dụng ReactJS, toàn bộ cây DOM sẽ phải cập nhật để báo hiệu sự thay đổi mới này. Còn khi bạn sử dụng React, nó sẽ giúp bạn scan qua Virtual DOM để xem những gì đã thay đổi sau khi người dùng thực hiện hành động trên (trong trường hợp này, thêm mới bình luận) và lựa chọn đúng nơi đúng chỗ cần cập nhật sự thay đổi mà thôi.

## **1.5. Tổng quan về mô hình MVC:**

**1.5.1. Mô hình MVC là gì?**

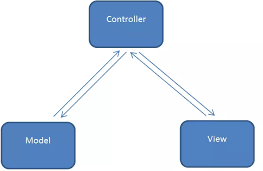
**MVC** là viết tắt của cụm từ “**Model-View-Controller**“. Đây là mô hình thiết kế sử

dụng trong kỹ thuật phần mềm. MVC là một để tạo lập giao

**mẫu kiến trúc phần mềm**

diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau như tên gọi: Model (dữ liệu), View (giao diện) và Controller (bộ điều khiển).

Là mô hình này được chia thành 3 phần trong soure code. Và mỗi phần đảm nhận vai trò và nhiệm vụ riêng biệt nhau và độc lập.



**Hình 1.6 Mô hình MVC**

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

* **Model:**

Model là thành phần trong mẫu thiết kế MVC được sử dụng để quản lý dữ liệu đó tức là trạng thái của ứng dụng trong bộ nhớ. Mô hình đại diện cho một tập hợp các lớp được sử dụng để mô tả logic xác thực của ứng dụng, logic nghiệp vụ và logic truy cập dữ liệu.

* **View:**

View trong mẫu thiết kế MVC được sử dụng để chứa logic để biểu diễn dữ liệu mô hình dưới dạng giao diện người dùng mà người dùng cuối có thể tương tác. Về cơ bản, khung nhìn được sử dụng để hiển thị dữ liệu miền (tức là dữ liệu kinh doanh) được bộ điều khiển cung cấp cho nó.

* **Controller:**

Controller là một tệp .cs (dành cho ngôn ngữ C #) có một số phương thức được gọi là Phương thức hành động. Khi một yêu cầu đến trên bộ điều khiển, phương thức hành động của bộ điều khiển sẽ xử lý các yêu cầu đó.

Controller là thành phần trong ứng dụng MVC được sử dụng để xử lý Yêu cầu HTTP đến và dựa trên hành động của người dùng, bộ điều khiển tương ứng sẽ làm việc với mô hình và chế độ xem, sau đó gửi phản hồi lại cho người dùng đã đưa ra yêu cầu ban đầu. Vì vậy, nó là cái sẽ tương tác với cả mô hình và khung nhìn để kiểm soát luồng thực thi ứng dụng.

**1.5.2. Luồng xử lí trong mô hình MVC:**

Luồng xử lí trong mô hình MVC:

* Khi một yêu cầu từ máy khách gửi đến server. Thì bị Controller trong mô hình MVC chặn lại để xem có URL, request hay sự kiện
* Sau đó Controller sẽ sử lí yêu cầu các bussiness logic của user rồi giao tiếp với model trong MVC
* Model sẽ gửi lại data cho Controller
* Cuối cùng khi xử lí yêu cầu xong thì Controller sẽ gửi dữ liệu về view và hiển thị

**View sẽ không giao tiếp trực tiếp với Model**. Sự tương tác giữa **View và Model** sẽ chỉ được xử lí qua **Controller.**

**1.5.3. Ưu nhược điểm của mô hình MVC**

* **Ưu điểm:** 
  + Hỗ trợ quá trình phát triển nhanh chóng: Với đặc điểm hoạt động độc lập của từng thành phần, các lập trình viên có thể làm việc đồng thời trên từng bộ phận khác nhau của mô hình này. MVC giúp bạn tiết kiệm rất nhiều thời gian.
  + Khả năng cung cấp đồng thời nhiều khung View: Với mô hình MVC, bạn có thể tạo ra đồng thời nhiều khung View cho Model.
  + Hỗ trợ các kỹ thuật không đồng bộ: MVC có thể hoạt động trên nền tảng JavaScript. Điều này có nghĩa là các ứng dụng MVC có thể hoạt động với các file PDF, các trình duyệt web cụ thể, và cả các widget máy tính.
  + Dễ dàng thao tác chỉnh sửa: Bộ phận Model hoạt động tách biệt với View đồng nghĩa với việc bạn có thể đưa ra các thay đổi, chỉnh sửa hoặc cập nhật dễ dàng ở từng bộ phận.
  + Giữ nguyên trạng thái data: Mô hình MVC truyền lại dữ liệu nhưng không định dạng lại dữ liệu. Do đó, các dữ liệu này có thể được dùng lại cho các thay đổi sau này.
  + Hỗ trợ các nền tảng phát triển SEO: Với mô hình MVC, bạn có thể dễ dàng tạo ra các mã SEO URL để thu hút lượng truy cập đối với ứng dụng bất kỳ.
* **Nhược điểm:** 
  + Khó khăn trong quá trình điều hướng code: Điều hướng khung có thể phức tạp vì mô hình này bao gồm nhiều lớp và yêu cầu người dùng thích ứng với các tiêu chí phân tách của MVC.
  + Không thích hợp việc phát triển các ứng dụng nhỏ vì mô hình này yêu cầu bạn lưu trữ một số lượng lớn các file.
  + Nhiều khung hoạt động đồng thời: Việc phân tách một tính năng thành ba bộ phận khác nhau dễ dẫn đến hiện tượng phân tán. Do đó, đòi hỏi các nhà phát triển phải duy trì tính nhất quán của nhiều bộ phận cùng một lúc.

# 1.6. Tổng quan về Express framework:

**1.6.1. Express framework là gì?**

Expressjs hay còn được viết là Express js, Express.js. Đây là một framework mã nguồn mở miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển.Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn.Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Expressjs cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN.Nhờ có thư viện Javascript của Express js đã giúp cho các nhà lập trình xây dựng nên các ứng dụng web hiệu quả và nhanh chóng hơn. Expressjs cũng được sử dụng để nâng cao các chức năng của Node.js.

Trên thực tế, nếu không sử dụng Express.js, bạn sẽ phải thực hiện rất nhiều bước lập trình phức tạp để xây dựng nên một API hiệu quả. Express js đã giúp cho việc lập trình trong Node.js trở nên dễ dàng hơn và có nhiều tính năng mới bổ sung.

Thời gian chính là tài sản quý giá của bất kỳ doanh nghiệp nào. Hơn thế, nhiều lập trình viên còn phải chịu áp lực xây dựng các ứng dụng web một cách hiệu quả trong thời gian ngắn. Nhưng để thực hiện tốt công việc này cần rất nhiều thời gian và Express.js chính là công cụ cứu cánh cho các nhà lập trình.

Express.js có thể giúp làm giảm một nửa thời gian viết mã mà vẫn xây dựng lên các ứng dụng web hiệu quả. Không chỉ trợ giúp về mặt thời gian Expressjs còn làm giảm những áp lực cần thiết để xây dựng với sự trợ giúp của các tính năng khác nhau của nó.Express js còn cung cấp một phần mềm trung gian đảm nhận nhiệm vụ đưa ra các quyết định để phản hồi chính xác những yêu cầu của khách hàng.

Nếu không có Express.js, các lập trình viên phải viết mã code riêng để xây dựng nên thành phần định tuyến. Đó là một công việc vô cùng tẻ nhạt và tốn thời gian. Express.js đã giúp cho công việc này trở nên đơn giản và hiệu quả hơn rất nhiều.

Phát triển máy chủ nhanh chóng: Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.

Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js.

Định tuyến - Routing: Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.

Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.

Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

# 1.7. Tổng quan về cơ sở dữ liệu Mongodb:

**1.7.1. Khái niệm**

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh..

**1.7.2. Lịch sử ra đời**

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

MongoDB đã trở thành một trong những NoSQL database nổi trội nhất bấy giờ, được dùng làm backend cho rất nhiều website như eBay, SourceForge và The New York Times.

Các feature của MongoDB gồm có:

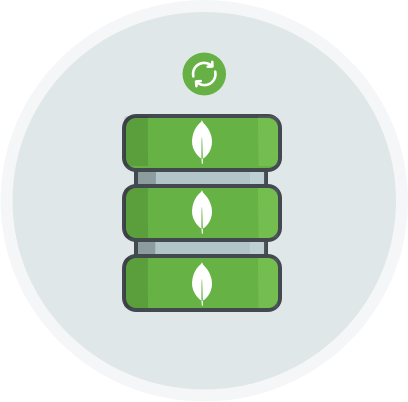
Các hỗ trợ search bằng field, các phép search thông thường, regular expression searches, và range queries.

Indexing: bất kì field nào trong BSON document cũng có thể được index.

Replication: có ý nghĩa là “nhân bản”, là có một phiên bản giống hệt phiên bản đang tồn tại, đang sử dụng. Với cơ sở dữ liệu, nhu cầu lưu trữ lớn, đòi hỏi cơ sở dữ liệu toàn vẹn, không bị mất mát trước những sự cố ngoài dự đoán là rất cao. Vì vậy, người ta nghĩ ra khái niệm “nhân bản”, tạo một phiên bản cơ sở dữ liệu giống hệt cơ sở dữ liệu đang tồn tại, và lưu trữ ở một nơi khác, đề phòng có sự cố.

Aggregation: Các Aggregation operation xử lý các bản ghi dữ liệu và trả về kết quả đã được tính toán. Các phép toán tập hợp nhóm các giá trị từ nhiều Document lại với nhau, và có thể thực hiện nhiều phép toán đa dạng trên dữ liệu đã được nhóm đó để trả về một kết quả duy nhất. Trong SQL, count(\*) và GROUP BY là tương đương với Aggregation trong MongoDB.

Lưu trữ file: MongoDB được dùng như một hệ thống file tận dụng những function trên và hoạt động như một cách phân phối qua sharding.



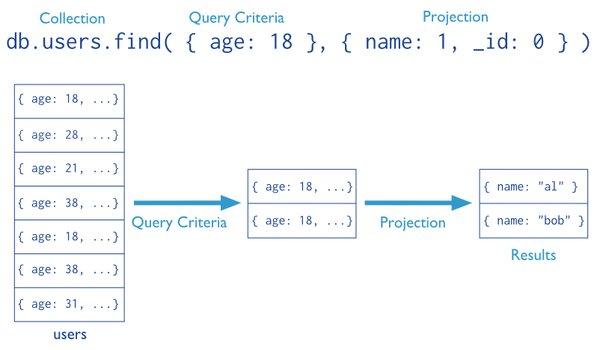
**Hình 1.7 Mongodb**

**1.7.3. Khi nào sử dụng Mongodb**

Quản lý và truyền tải content – Quản lý đa dạng nhiều product của content chỉ trong một kho lưu trữ data cho phép thay đổi và phản hồi nhanh chóng mà không chịu thêm phức tạp thêm từ hệ thống content.

Cấu trúc Mobile và Social – MongoDB cung cấp một platform có sẵn, phản xạ nhanh, và dễ mở rộng cho phép rất nhiều khả năng đột phá, phân tích real-time, và hỗ trợ toàn cầu.

Quản lý data khách hàng – Tận dụng khả năng query nhanh chóng cho phân tích real-time trên cơ sở dữ liệu người dùng cực lớn vớ các mô hình data phức tạp bằng các schema linh hoạt và tự động sharding cho mở rộng chiều ngang.



**Hình 1.8 Tính năng Mongodb**

MongoDB có các tính năng cơ bản như sau:

* Truy vấn
* Nhân rộng
* Cân bằng tải
* Lưu trữ tệp
* Tập hợp
* Thực thi Javascript phía máy chủ
* Giới hạn kích thước collection
* Giao dịch

Truy vấn ad hoc

Truy vấn ad hoc là một trong những tính năng tốt nhất của chương trình. Nó hỗ trợ các trường, truy vấn phạm vi và tìm kiếm các biểu thức để trả về các trường tài liệu cụ thể bao gồm các hàm JavaScript do người dùng xác định hoặc các truy vấn này được cấu hình và trả về mẫu kết quả ngẫu nhiên có kích thước nhất định. Bên cạnh đó, các trường trong MongoDB có thể được dùng để lập các chỉ mục chính và các chỉ mục phụ

Nhân rộng

Đây là tính năng mà chương trình cung cấp Replica set. Nó bao gồm hai hoặc nhiều bản sao của dữ liệu. Trong đó mỗi bản sao có thể đóng vai trò chính và phụ.

Trong quá trình nhân rộng, tất cả các dữ liệu ghi và đọc thực hiện trên bản sao chính.

Bản sao thứ cấp sử dụng bản sao tích hợp để duy trì bản sao dữ liệu của bản chính.

Trong trường hợp một bản sao chính thất bại, Replica set chọn ra một bản sao thứ cấp để trở thành bản sao chính tiếp theo. Replica thứ cấp có thể tùy chọn phục vụ các hoạt động nhưng dữ liệu cuối cùng chỉ nhất quán theo mặc định.

Cân bằng tải

Bằng cách sử dụng Sharding, MongoDB chia tỷ lệ theo chiều ngang để người dùng chọn một Shard key. Xác định các dữ liệu được phân phối trong collection. Nhìn chung, dữ liệu được chia thành các phạm vi và được phân phối đồng đều dựa trên các Shard key. Nó chạy trên nhiều máy chủ, cân bằng tải hoặc sao chép dữ liệu để giữ hệ thống luôn hoạt động trong trường hợp có lỗi về phần cứng.

Lưu trữ tệp

Với tính năng lưu trữ tệp, MongoDB được sử dụng như một hệ thống tệp (GridFS) giúp cân bằng tải và sao chép dữ liệu trên nhiều máy tính để lưu trữ tệp. Trong đó, GridFS chia một tệp ra thành các phần hoặc các đoạn và lưu trữ thành những tài liệu riêng biệt. Bạn có thể truy cập GridFS bằng tiện ích Mongofiles hoặc plugin cho Nginx và Lighttpd.

Tập hợp

Ở tính năng tập hợp, chương trình này cung cấp ba cách chính để thực hiện tập hợp là Aggregation Pipeline, chức năng Mapreduce và Single-purpose Aggregation. Trong đó, theo tài liệu của MongoDB thì Aggregation Pipeline được công nhận là cung cấp hiệu suất tốt hơn hầu hết các hoạt động tổng hợp.

Thực thi Javascript phía máy chủ

JavaScript thường được thực thi trong các truy vấn, các hàm tổng hợp và được gửi trực tiếp đến cơ sở dữ liệu

Giới hạn kích thước collection

MongoDB hỗ trợ collection có kích thước cố định được gọi là collection giới hạn. Nó có kích cỡ cố định theo sau thứ tự chèn làm tăng hiệu suất các hoạt động khác có liên quan đến dữ liệu. Khi dữ liệu vượt quá mức giới hạn thì các tài liệu cũ sẽ bị xóa mà không cần dùng bất cứ dòng lệnh nào.

Giao dịch

MongoDB còn có tính năng hỗ trợ cho các giao dịch ACID đa tài liệu, bắt đầu có hiệu dụng từ phiên bản 4.0 vào tháng 6 năm 2018**.**

.

# CHƯƠNG 2: PHƯƠNG PHÁP THỰC HIỆN

## **2.1. Xác định yêu cầu:**

Một cửa hàng bán điện thoại cụ thể là các hãng điện thoại mà cửa hàng đang bán. Cửa hàng yêu cầu hệ thống có các chức năng như sau:

1. **Quản trị**

* Quản lí danh sách thông tin sản phẩm
* Quản lí danh sách danh mục sản phẩm
* Quản lí danh sách tài khoản quản trị
* Quản lí danh sách tài khoản người dùng
* Quản lí danh sách đơn hàng
* Quản lí thống kê doanh thu của cửa hàng.

1. **Đặt hàng**

Khách hàng tham quan website, xem, lựa chọn mặt hàng. Nếu mua hàng khách đăng nhập để tiến hành đặt hàng. Khách hàng có thể xóa mặt hàng không muốn mua ra khỏi giỏ hàng của họ. Nếu đặt hàng thành công hàng sẽ được giao cho khách hàng và có thể theo dõi tình trạng đơn hàng.

1. **Bán hàng**

Hàng ngày, người quản trị kiểm tra các đơn đặt hàng và giao các đơn hàng hợp lệ cho bộ phận bán hàng. Bộ phận bán hàng nhận đơn hàng và phiếu xuất kho giao cho nhân viên giao hàng. Nhân viên giao hàng nhận hàng tại kho rồi giao hàng theo địa chỉ khách hàng. Nếu giao hàng thành công, nhân viên giao hàng sẽ gửi hóa đơn cho khách hàng và nhận tiền. Nhân viên giao lại hóa đơn và tiền cho bộ phận bán hàng. Nếu giao hàng không thành công, nhân viên giao hàng sẽ báo cáo lại cho bộ phận bán hàng và trả lại hàng.

## **2.2. Phân tích bài toán**

**Đối tượng sử dụng: có 3 đối tượng sử dụng hệ thống website:**

* Khách vãng lai: là người vào xem tham khảo sản phẩm, giá cả …., gửi yêu cầu hỗ trợ nhưng chưa đăng kí tài khoản thành viên
* Khách thành viên: là những khách hàng đã đăng kí vào hệ thống và được lưu thông tin, có quyền xem hàng, mua hàng và đặc biệt là quản lí được đơn hàng của mình…
* Người quản trị: là người toàn quyền kiểm soát hệ thống của website

**Cụ thể từng chức năng**

* **Khách vãng lai:**
* Tìm kiếm sản phẩm
* Đăng kí thành viên
* Xem thông tin của sản phẩm
* **Thành viên:**
* Đăng nhập, đăng xuất hệ thống.
* Thay đổi mật khẩu
* Tìm kiếm và xem thông tin của sản phẩm
* Đặt hàng, xem chi tiết đơn hàng và thanh toán
* Xem lại các đơn hàng của mình
* **Quản trị viên:**
* Quản lí khách hàng

+ Xem thông tin khách hàng

+ Xoá tài khoản khách hàng

* Quản lí sản phẩm:

+ Thêm, xoá, sửa sản phẩm

**+** Thay đổi trạng thái của sản phẩm

* Quản lí danh mục sản phẩm

+ Thêm, xoá, sửa danh mục sản phẩm

* Quản lí đơn hàng

+ Quản lí trạng thái đơn hàng

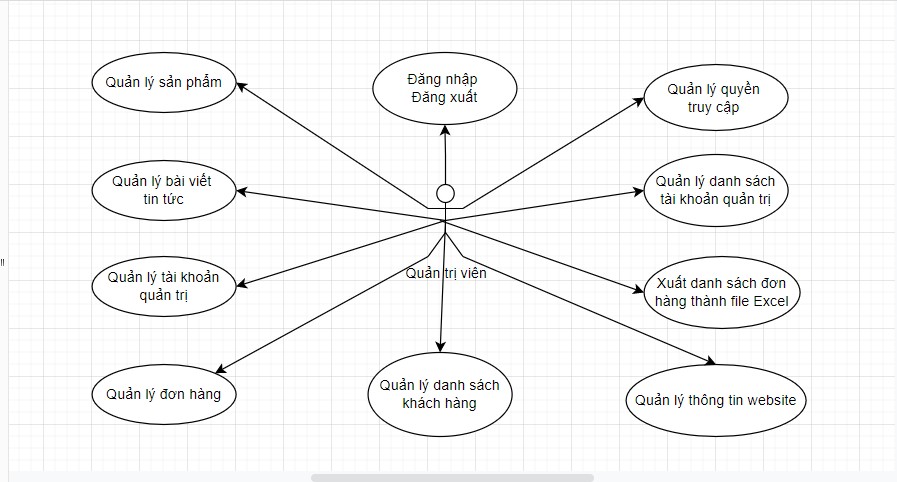
* Quản lí tài khoản quản trị

+ Quản lí quyền truy cập

+ Thêm xoá, sửa tài khoản quản trị

* Thống kê đơn hàng, khách hàng và doanh số của cửa hàng

## **2.3. lược đồ Use-case**



**Hình 2.1 lược đồ use-case quản trị viên**

Diagram, schematic

Description automatically generated

**Hình 2.2 lược đồ khách hàng thành viên**

Diagram

Description automatically generated

**Hình 2.3 lược đồ khách vãng lai**

**2.3.1. Đặc tả use-case đăng nhập**

Use-case này mô tả cách đăng nhập vào hệ thống website của cửa hàng

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi một actor muốn đăng nhập vào hệ thống.
* Hệ thống yêu cầu cầu các actor nhập tên đăng nhập và mật khẩu.
* Actor nhập tên và mật khẩu.
* Hệ thống kiểm tra tên đăng nhập và mật khẩu mà actor đã nhập và cho phép actor đăng nhập vào hệ thống.
* **Dòng sự kiện khác:**

Nếu trong dòng sự kiện chính các actor nhập tên đăng nhập và mật khẩu sai thì hệ thống sẽ thông báo lỗi. Actor có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc đăng nhập lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thì người đăng nhập sẽ có các quyền sử dụng hệ thống tương ứng. Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.2. Đặc tả use-case đăng kí thành viên:**

* **Tóm tắt:** use case này mô tả cách đăng ký thành viên của website cửa hàng
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi một actor muốn đăng ký vào hệ thống.
* Hệ thống yêu cầu cầu các actor nhập họ và tên, địa chỉ email, số điện thoại, địa chỉ và mật khẩu cài đặt.
* Actor điền đầy đủ thông tin.
* Hệ thống kiểm tra tên email và mật khẩu mà actor đã nhập và cho phép actor đăng ký vào hệ thống.
* **Dòng sự kiện khác:**

Nếu email mà actor điền vào đã có sẵn hoặc trùng với dữ liệu của hệ thống thì hệ thống sẽ báo lỗi và thông tin đăng ký tài khoản thành viên trên không được đăng ký.

Nếu mật khẩu xác nhận mà actor không giống với mật khẩu ban đầu thì thông tin tài khoản actor vừa đăng ký không thành công.

Việc đăng ký lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thì người đăng ký sẽ có các quyền sử dụng hệ thống tương ứng. Ngược lại trạng thái của hệ thống sẽ không đổi.

* **Mở rộng**

Mật khẩu actor nhập vào yêu cầu phải tối thiểu 6 ký tự.

**2.3.3 Đặc tả use-case danh mục sản phẩm**

* **Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý danh mục sản phẩm có trên hệ thống.
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người quản lí xem xét và quản lý những danh mục sản phẩm tại cửa hàng.
* Hệ thống yêu cầu người dùng lựa chọn phương thức quản lý bao gồm:
  + Thêm danh mục sản phẩm

Người dùng thêm thông tin cần thiết như: Tên danh mục sản phẩm, số thứ tự và chọn lưu.

Hệ thống thông báo lại và lưu thông tin của danh mục sản phẩm vào hệ thống.

Kết thúc phương thức.

* + Xoá danh mục sản phẩm

Người dùng có thể xóa danh mục sản phẩm nếu danh mục sản phẩm.

Hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* + Cập nhật thông tin danh mục sản phẩm

Người dùng có thể chỉnh sửa thông tin danh mục sản phẩm nếu thông tin danh mục sản phẩm đó không chính xác.

Hệ thống thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* **Dòng sự kiện khác:**

Người dùng có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc thay đổi lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thông tin danh mục sản phẩm trên hệ thống sẽ được thay đổi. Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.4. Đặc tả use-case quản lí sản phẩm**

* **Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý sản phẩm có trên hệ thống.
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi nhân viên xem xét và quản lý những sản phẩm tại cửa hàng.
* Hệ thống yêu cầu người dùng lựa chọn phương thức quản lý bao gồm:
  + Thêm sản phẩm

Người dùng thêm thông tin cần thiết như: Tên sản phẩm, giá, danh mục,

… và chọn lưu.

Hệ thống thông báo lại và lưu thông tin của sản phẩm vào hệ thống. Kết thúc phương thức.

* + Xoá sản phẩm

Người dùng có thể xóa sản phẩm, hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật

* + Cập nhật thông tin sản phẩm

Người dùng có thể chỉnh sửa thông tin sản phẩm nếu thông tin sản

Hệ thống thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* **Dòng sự kiện khác:**

Người dùng có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc thay đổi lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use-case thành công thông tin sản phẩm trên hệ thống sẽ được thay đổi.

Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.5. Đặc tả use-case quản lí bài viết**

* **Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý bài viết, tin tức có trên hệ thống.
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi nhân viên xem xét và quản lý những bài viết, tin tức tại cửa hàng.
* Hệ thống yêu cầu người dùng lựa chọn phương thức quản lý bao gồm:
  + Thêm bài viết

Người dùng thêm thông tin cần thiết như: Tên, tiêu đề, hình ảnh, nội dung bài viết và chọn lưu.

Hệ thống thông báo lại và lưu thông tin của bài viết vào hệ thống. Kết thúc phương thức.

* + Xoá bài viết

Người dùng có thể xóa bài viết nếu nó có vấn đề.

Hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* + Cập nhật thông tin sản phẩm

Người dùng có thể chỉnh sửa thông tin bài viết nếu thông tin bài viết.

Hệ thống thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* **Dòng sự kiện khác:**

Người dùng có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc thay đổi lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thông tin bài viết trên hệ thống sẽ được thay đổi. Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.6 Đặc tả use-case quản lí quyền truy cập quản trị viên**

* **Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý quyền truy cập trang quản trị viên có trên hệ thống.
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi quản trị viên cao nhất (chủ cửa hàng) quản lý danh mục quyền quản trị tại cửa hàng.
* Hệ thống yêu cầu người dùng lựa chọn phương thức quản lý bao gồm:
  + Thêm quyền truy cập

Người dùng thêm thông tin cần thiết như: Tên quyền, mô tả và chọn lưu. Hệ thống thông báo lại và lưu thông tin của quyền truy cập vào hệ thống. Kết thúc phương thức.

* + Xoá quyền truy cập

Người dùng có thể xóa quyền truy cập nếu nó có vấn đề. Hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* + Cập nhật thông tin quyền truy cập

Người dùng có thể chỉnh sửa thông tin quyền truy cập nếu thông tin quyền truy cập đó không chính xác.

Hệ thống thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* **Dòng sự kiện khác:**

Người dùng có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc thay đổi lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thông tin quyền truy cập trên hệ thống sẽ được thay đổi.

Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.7. Đặc tả use-case quản lí tài khoản quản trị**

* **Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý danh sách tài khoản quản trị viên có trên hệ thống.
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi quản trị viên cao nhất (chủ cửa hàng) quản lý tài khoản quản trị tại cửa hàng.
* Hệ thống yêu cầu người dùng lựa chọn phương thức quản lý bao gồm:
  + Thêm tài khoản

Người dùng thêm thông tin cần thiết như: Họ và tên, số điện thoại, email, mật khẩu, quyền truy cập và chọn lưu.

Hệ thống thông báo lại và lưu thông tin của tài khoản quản trị mới vào hệ thống.

Kết thúc phương thức.

* + Xoá tài khoản quản trị

Người dùng có thể xóa tài khoản quản trị đã được tạo. Hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật.

Kết thúc phương thức.

* **Dòng sự kiện khác:**

Người dùng có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc thay đổi lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thông tin tài khoản quản trị viên trên hệ thống sẽ được thay đổi. Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.8. Đặc tả use-case quản lí đơn hàng**

* **Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý danh sách đơn hàng khách hàng đã đặt có trên hệ thống.
* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi quản trị viên quản lý tất cả đơn hàng tại cửa hàng.
* Hệ thống yêu cầu người dùng lựa chọn phương thức quản lý bao gồm:
  + Lọc danh sách đơn hàng theo trạng thái

Người dùng lựa chọn lọc danh sách đơn hàng theo: Chờ xác nhận, chờ lấy hàng, đơn hủy, đơn trả…

Hệ thống sẽ lọc theo tình trạng đơn hàng mà người dùng đã lựa chọn và hiển thị ra màn hình danh sách đơn hàng.

Kết thúc phương thức.

* + Xem chi tiết đơn hàng

Người dùng có thể bấm xem chi tiết đơn hàng, bao gồm thông tin liên hệ của khách hàng và chi tiết hóa đơn đặt hàng mà khách hàng đã đặt mua. Kết thúc phương thức.

* + Xoá đơn hàng

Người dùng có thể xóa đơn hàng nếu nó có vấn đề hoặc theo mong muốn của quản trị viên.

Hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* + Cập nhật trạng thái đơn hàng

Người dùng có thể cập nhật trạng thái đơn hàng, lựa chọn trạng thái của đơn hàng.

Hệ thống sẽ thông báo và lưu trữ trạng thái mới cập nhật. Kết thúc phương thức.

* **Dòng sự kiện khác:**

Người dùng có thể quay trở về dòng đầu sự kiện hoặc hủy bỏ. Việc thay đổi lúc này use case kết thúc.

* **Sự kiện sau**

Nếu use case thành công thông tin đơn hàng trên hệ thống sẽ được thay đổi. Ngược lại trạng thái của hệ thống sẽ không đổi.

**2.3.9 Đặc tả use-case xem sản phẩm**

**Tóm tắt:** use case này cho phép người dùng thực hiện chức năng xem thông tin sản phẩm trên hệ thống website.

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người dùng xem thông tin sản phẩm có tại website của cửa hàng.
* Hệ thống sẽ cho phép người dùng có thể xem sản phẩm dựa vào các danh mục sản phẩm có trên website.
* Người có thể nhấn vào để xem thông tin chi tiết của sản phẩm như: giá thành, số lượng hàng còn trong kho, thông tin chi tiết của sản phẩm…
* Kết thúc phương thức
* **Dòng sự kiện khác:**

Không có.

**2.3.10. Đặc tả use-case tìm kiếm sản phẩm**

**Tóm tắt:** use case này cho phép người dùng thực hiện chức năng tìm kiếm sản phẩm theo tên trên hệ thống website.

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người dùng tìm kiếm sản phẩm theo tên có tại website của cửa hàng.
* Người dùng nhập các ký tự hoặc chữ vào thanh tìm kiếm
* Hệ thống sẽ liệt kê danh sách tất cả tên sản phẩm có thể trùng với ký tự hoặc chữ cái mà người dùng nhập vào.
* Người có thể nhấn vào các gợi ý hoặc sản phẩm có tên để xem thông tin chi tiết của sản phẩm như: giá thành, số lượng hàng còn trong kho, thông tin chi tiết của sản phẩm…
* Kết thúc phương thức.
* **Dòng sự kiện khác:**

Nếu người dùng nhập các ký tự hoặc chữ cái không trùng hoặc có tên với tên sản phẩm của cửa hàng thì thanh tìm kiếm sẽ không có bất kỳ gợi ý nào cho người dùng.

* **Dòng sự sau:**

Nếu use case thành công thì người dùng có thể chuyển sang trạng thái xem hàng.

**2.3.11 Đặc tả use-case quản lí giỏ hàng:**

**Tóm tắt:** use case này cho phép người dùng thực hiện chức năng quản lý sản phẩm có trong giỏ hàng.

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người dùng nhấn nút thêm sản phẩm vào giỏ hàng.
* Hệ thống sẽ tự động thêm tất cả số lượng sản phẩm mà người dùng đã thêm vào giỏ hàng
* Người dùng có thể thực hiện các thao tác quản lý của hàng của mình gồm:
  + Cập nhật số lượng sản phẩm đã có trong giỏ hàng.

Người dùng có thể tăng hoặc giảm số lượng sản phẩm đó.

Hệ thống sẽ tự động cập nhật lại số lượng và tổng giá tiền của đơn hàng. Phương thức kết thúc.

* + Xóa sản phẩm ra khỏi giỏ hàng

Người dùng nhấn nút xóa sản phẩm ra khỏi giỏ hàng.

Hệ thống xóa sản phẩm được chọn ra khỏi giỏ hàng và cập nhật tổng tiền của đơn hàng.

Phương thức kết thúc

* **Dòng sự kiện khác:**

Nếu người dùng xóa bỏ hết các sản phẩm trong giỏ hàng thì hệ thống sẽ hiện trong thông báo giỏ hàng rỗng và kết thúc trạng thái hoạt động

* **Dòng sự sau:**

Nếu use case thành công thì người dùng có thể chuyển sang trạng thái đặt hàng.

Hoặc cũng có thể trở về trạng thái xem hàng.

**2.3.12 Đặc tả use-case đặt – mua hàng**

**Tóm tắt:** use case này cho phép người dùng thực hiện chức năng đặt mua sản phẩm có trên website của hàng.

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người dùng thêm sản phẩm vào giỏ hàng và nhấn nút đặt hàng.
* Hệ thống sẽ kiểm tra và yêu cầu người dùng phải đăng nhập tài khoản của hệ thống mới được cho chép thực hiện hoạt động này.
* Người dùng đã đăng nhập vào hệ thống website sẽ có thể thực hiện các thao tác sau:
  + Cập nhật nhật lại thông tin đặt hàng của mình

Người dùng điền lại thông tin đặt hàng của mình vào form Sau đó lựa chọn phương thức thanh toán

Hệ thống sẽ lưu lại thông tin mới và tiến hành lưu thông tin đơn hàng vào hệ thống, sau đó thông báo cho người dùng đã đặt hàng thành công.

Phương thức kết thúc.

* **Dòng sự kiện khác:**

Nếu người dùng chưa đăng nhập hệ thống sẽ chuyển người dùng về trang đăng nhập. Nếu người dùng chưa có tài khoản thì sẽ tiến hành đăng ký toàn khoản thành viên vào hệ thống.

* **Dòng sự sau:**

Nếu use case thành công thì người dùng có thể chờ liên hệ xác nhận đơn hàng của nhân viên của hàng.

**2.3.13 Đặc tả use-case thay đổi mật khẩu**

**Tóm tắt:** use case này cho phép người dùng thực hiện chức năng thay đổi mật khẩu đăng ký thành viên của hệ thống website.

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người dùng vào quản lý tài khoản của mình lựa chọn đổi mật khẩu khi đã đăng nhập tài khoản vào hệ thống.
* Hệ thống sẽ yêu cầu người dùng điền mật khẩu cũ, mật khẩu mới và xác nhận lại mật khẩu mới
* Người dùng điền đầy đủ thông tin mật khẩu
* Hệ thống sẽ kiểm tra lại thông tin mật khẩu và lưu lại thay đổi thông tin mật khẩu mới
* **Dòng sự kiện khác:**

Nếu người dùng điền mật khẩu cũ không chính xác với mật khẩu đã lưu trong hệ thống. Hệ thống sẽ thông báo lỗi và không thay đổi mật khẩu mới thành công.

**2.3.14. Đặc tả use-case gửi phản hồi liên hệ**

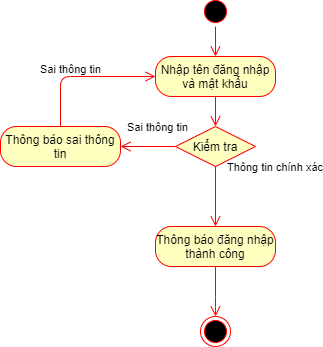
**Tóm tắt:** use case này cho phép người dùng thực hiện chức năng gửi phản hồi đến quản trị viên của cửa hàng trên hệ thống.

* **Dòng sự kiện:**
* **Dòng sự kiện chính:**
* Use case này bắt đầu khi người dùng vào gửi phản hồi đến quản trị viên của của hàng
* Hệ thống sẽ yêu cầu người dùng điền thông tin đầy đủ như: Họ tên, số điện thoại, email, địa chỉ và nội dung yêu cầu hỗ trợ hay phản hồi.
* Hệ thống sẽ lưu lại thông tin của khách hàng và thong báo khách hàng đã gửi phản hổi thành công.
* **Dòng sự kiện khác:**

Không có.

## **2.4. Thiết kế lược đồ activity diagram**

* **Lược đồ chức năng đăng nhập**



**Hình 2.4 lược đồ chức năng đăng nhập**

* **Lược đồ chức năng đăng ký thành viên**

Diagram

Description automatically generated

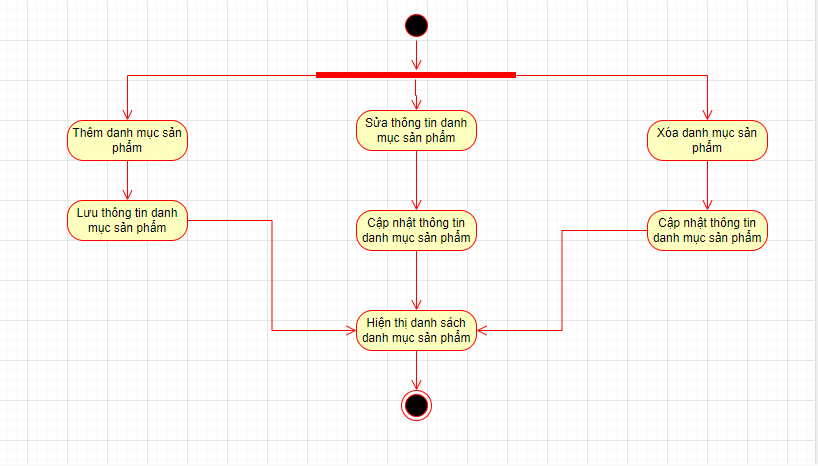
**Hình 2.5 lược đồ chức năng đăng ký thành viên**

* Diagram

  Description automatically generated**lược đồ chức năng quản lí sản phẩm**

**Hình 2.6 lược đồ chức năng quản lí sản phẩm**

* **Lược đồ chức năng quản lí danh mục sản phẩm**



**Hình 2.7 lược đồ chức năng quản lý danh mục sản phẩm**

* Diagram

  Description automatically generated**lược đồ chức năng quản lí bài viết**

**Hình 2.8 lược đồ chức năng quản lý bài viết**

* Diagram

  Description automatically generated**Lược đồ chức năng quản lí đơn hàng**

**Hình 2.9 lược đồ chức năng quản lí đơn hàng**

* **Lược đồ chức năng quản lí quyền truy cập**

Diagram

Description automatically generated

**Hình 2.10 lược đồ chức năng quản lí quyền truy cập**

* **Lược đồ chức năng quản lí tài khoản quản trị viên**

Diagram

Description automatically generated

**Hình 2.11 lược đồ chức năng quản lí tài khoản quản trị viên**

* **Lược đồ chức năng quản lí giỏ hàng**

Diagram

Description automatically generated

**Hình 2.12 lược đồ chức năng quản lí giỏ hàng**

* **Lược đồ chức năng đặt hàng**

Diagram

Description automatically generated

**Hình 2.13 lược đồ chức năng đặt hàng**

* **Lược đồ chức năng thay đổi mật khẩu**

Diagram

Description automatically generated

**Hình 2.14 lược đồ chức năng thay đổi mật khẩu**

## **2.5. Thiết kế database cơ sở dữ liệu**

* **Bảng product**

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| category | Object | Đối tượng thể loại |
| name | String | Tên sản phẩm |
| price | Number | Giá sản phẩm |
| release\_date | Date | Ngày tạo |
| img | String | Ảnh sản phẩm |
| describe | String | Mô tả sản phẩm |
| id\_category | String | Mã thể loại |
| view\_counts | Number | Số lượng xem |
| sales | Number | Số lượng mua |

* **Bảng category**

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id\_category | String | Mã thể loại |
| Name | String | Tên thể loại |

* **Bảng user**

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id\_user | String | Mã người dùng |
| Email | String | Email |
| FirstName | String | Họ |
| LastName | String | Tên |
| Pasword | String | Mật khẩu |
| Address | String | Địa chỉ |
| Phone\_number | String | Số điện thoại |
| Is\_admin | Boolean | Quyền quản trị |
| Is\_manage | Boolean | Quyền nhân viên |
| Is\_verify | Boolean | Xác thực |

* **Bảng cart**

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id\_user | String | Mã người dùng |
| Date | Date | Ngày |
| Product | Object | Đối tượng sản phẩm |

* **Bảng bill**

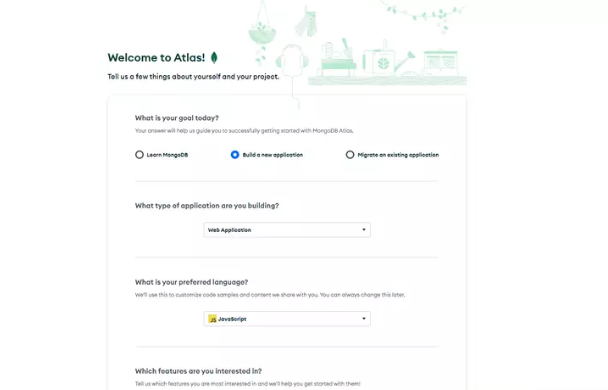
|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id\_user | String | Mã người dùng |
| Date | Date | Ngày |
| Product | Object | Đối tượng sản phẩm |
| Total | Number | Tổng tiền |
| Address | String | Địa chỉ giao hàng |
| Phone | String | Số điện thoại |
| Name | String | Họ tên |
| Is\_send | String | Trạng thái |

# CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM

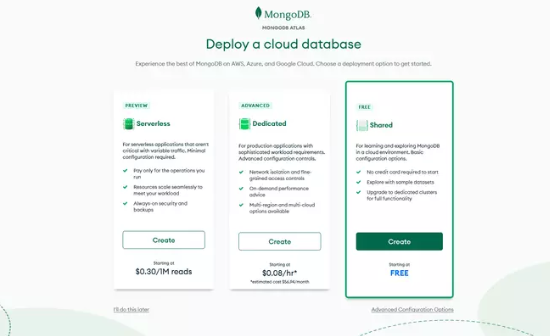
## **3.1. Hướng dẫn cài đặt**

**3.1.1 Cài đặt Mongodb**

Các bạn truy cập vào link: https://www.mongodb.com/atlas/database , sau đó hoàn thành các bước đăng ký và login xong chúng ta sẽ được đưa vào màn hình Welcome của mongodb atlas:

****

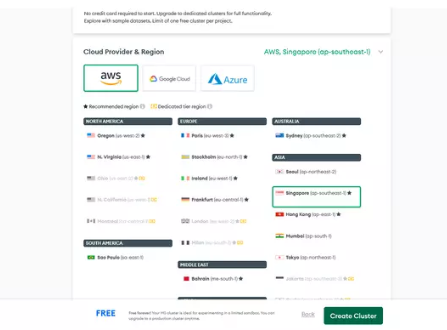
B2: Ok, nhập các thông tin và click button Finish bên dưới cùng và chuyển sang màn hình lựa chọn các options cloud database của bạn



B3: Bạn sẽ lựa chọn gói Shared, gói này hoàn toàn free và không yêu cầu credit card

B4: Sau khi chọn gói Shared các bạn sẽ được đưa tới màn hình tạo Cluster, ở đây các bạn có thể chọn nhà cung cấp dịch vụ cloud và hiện họ đang cung cấp 3 cloud provide là AWS, Google cloud và Azure

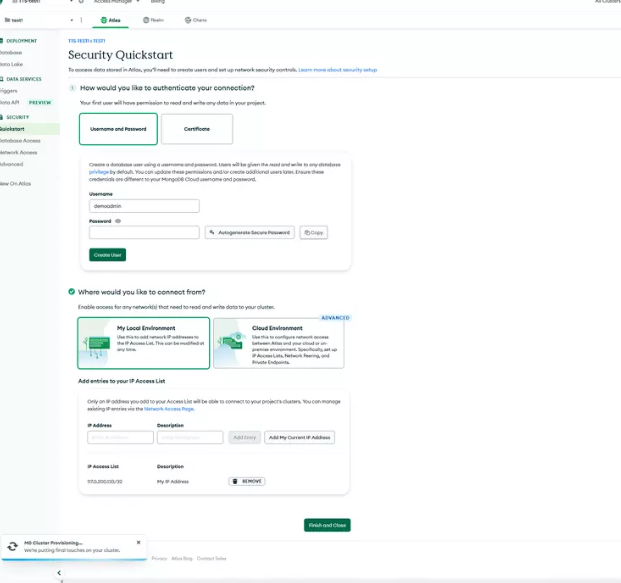
Đồng thời bạn cũng sẽ lựa chọn Region cho cloud db của bạn



B5 Sau khi nhấn tạo Cluster bạn sẽ di chuyển qua màn hình setup các thông tin kết nối tới cloud db này

Bao gồm: account info và access security

Phần địa chỉ IP các bạn có thể click button Add My Current IP Adress để thêm IP của mình vào danh sách các địa chi IP có thể kết nối tới cloud database này, rất an toàn phải không 😄 hoặc bạn cũng có thể chọn allow every where trong phần cài đặt Network access sau khi tạo xong Cluster



B6 Chờ vài giây chúng ta sẽ có 1 con Cluster được tạo và bạn sẽ nhìn thấy màn hình bên dưới, chúng ta có thể nhấn nút Connect để lấy database connection string hoặc click

**3.1.2.** **Cài đặt Microsoft SQL Server Managerment Studio:**

Cài đặt NodeJS và NPM - Install ReactJs

NodeJS là nền tảng cần thiết để phát triển ReactJS, trước khi cài đặt ReactJs, bạn cần phải cài đặt NodeJs.

Sau khi cài đặt thành công NodeJS, bạn có thể bắt đầu cài đặt React trên đó bằng npm. Bạn có thể cài đặt ReactJS theo hai cách

Sử dụng webpack và babel.

Sử dụng lệnh create-react-app .

Hướng dẫn 2 cách để Install ReactJs

Cách 1: Install ReactJs bằng webpack và babel

Webpack là một trình gói mô-đun (quản lý và tải các mô-đun độc lập). Nó có các mô-đun phụ thuộc và biên dịch chúng thành một gói (tệp) duy nhất. Bạn có thể sử dụng gói này trong khi phát triển ứng dụng bằng dòng lệnh hoặc bằng cách định cấu hình nó bằng tệp webpack.config.

Babel là một trình biên dịch và chuyển tiếp JavaScript. Nó được sử dụng để chuyển đổi mã nguồn này sang mã nguồn khác. Bằng cách sử dụng này, bạn sẽ có thể sử dụng các tính năng ES6 mới trong mã của mình, nơi babel chuyển đổi nó thành ES5 cũ thuần túy có thể chạy trên tất cả các trình duyệt.

Bước 1: Tạo Root Folder

Tạo 1 tập tin với tên reactApp trên desktop để cài đặt cái file theo yêu cầu, sử dụng lệnh mkdir.

C:\Users\username\Desktop>mkdir reactApp

C:\Users\username\Desktop>cd reactApp

Để tạo bất kỳ mô-đun nào, cần phải tạo tệp package.json . Do đó, sau khi Tạo thư mục, chúng ta cần tạo một tệp package.json . Để làm như vậy, bạn cần chạy lệnh npm init từ dấu nhắc lệnh.

C:\Users\username\Desktop\reactApp>npm init

Lệnh này hỏi thông tin về mô-đun như tên gói, mô tả, tác giả, v.v. bạn có thể bỏ qua các thông tin này bằng cách sử dụng tùy chọn –y.

C:\Users\username\Desktop\reactApp>npm init -y

Wrote to C:\reactApp\package.json:

Bước 2 - cài đặt React và react dom

Vì mục đích chính của ta là để cài đặt ReactJS và các gói dom của nó cũng như sử dụng ReactJS và React Dom của NPM tương ứng. Bạn có thể thêm các gói vào tệp package.json bằng cách sử dụng tùy chọn --save .

C:\Users\Tutorialspoint\Desktop\reactApp>npm install react --save

C:\Users\Tutorialspoint\Desktop\reactApp>npm install react-dom --save

Hoặc, bạn có thể cài đặt tất cả chúng bằng một lệnh duy nhất như -

C:\Users\username\Desktop\reactApp>npm install react react-dom --save

Bước 3 - Cài đặt webpack

Vì chúng ta đang sử dụng webpack để tạo gói cài đặt gói, webpack-dev-server và webpack-cli.

C:\Users\username\Desktop\reactApp>npm install webpack --save

C:\Users\username\Desktop\reactApp>npm install webpack-dev-server --save

C:\Users\username\Desktop\reactApp>npm install webpack-cli --save

Hoặc, bạn có thể cài đặt tất cả chúng bằng một lệnh duy nhất như -

C:\Users\username\Desktop\reactApp>npm install webpack webpack-dev-server webpack-cli --save

Bước 4 - Cài đặt babel

Cài đặt babel và các plugin babel-core, babel-loader, babel-preset-env, babel-preset-react và, html-webpack-plugin

C:\Users\username\Desktop\reactApp>npm install babel-core --save-dev

C:\Users\username\Desktop\reactApp>npm install babel-loader --save-dev

C:\Users\username\Desktop\reactApp>npm install babel-preset-env --save-dev

C:\Users\username\Desktop\reactApp>npm install babel-preset-react --save-dev

C:\Users\username\Desktop\reactApp>npm install html-webpack-plugin --save-dev

Hoặc, bạn có thể cài đặt tất cả chúng bằng một lệnh duy nhất như -

C:\Users\username\Desktop\reactApp>npm install babel-core babel-loader babel-preset-env

babel-preset-react html-webpack-plugin --save-dev

Bước 5: Tạo File

Để hoàn tất cài đặt, chúng ta cần tạo một số tệp nhất định, cụ thể là index.html, App.js, main.js, webpack.config.js và,. babelrc . Bạn có thể tạo các tệp này theo cách thủ công hoặc sử dụng dấu nhắc lệnh .

C:\Users\username\Desktop\reactApp>type nul > index.html

C:\Users\username\Desktop\reactApp>type nul > App.js

C:\Users\username\Desktop\reactApp>type nul > main.js

C:\Users\username\Desktop\reactApp>type nul > webpack.config.js

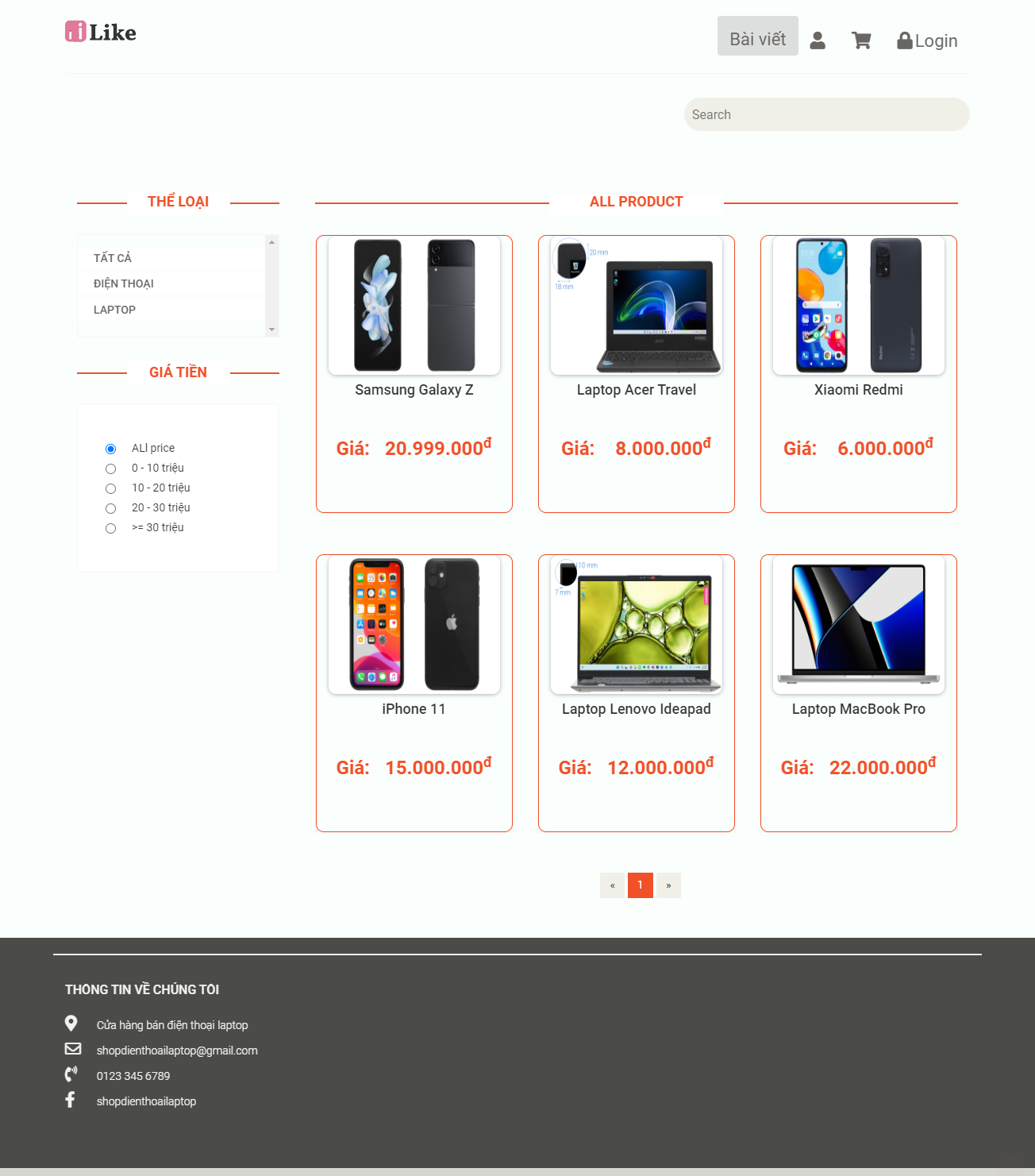
C:\Users\username\Desktop\reactApp>type nul > .babelrc

Bước 6 - Đặt Trình biên dịch, Máy chủ và Bộ tải

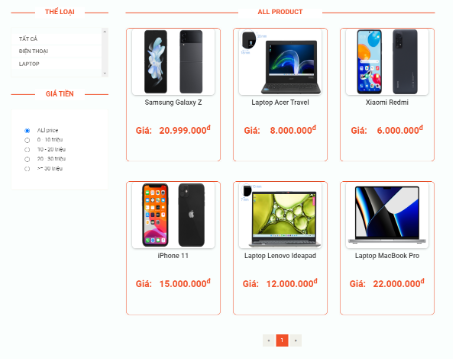
Mở tệp webpack-config.js và thêm mã sau. Chúng ta đang đặt điểm nhập webpack thành main.js. Đường dẫn đầu ra là nơi ứng dụng đi kèm sẽ được phân phát. Chúng ta cũng đang đặt máy chủ phát triển thành cổng 8001 . Bạn có thể chọn bất kỳ cổng nào bạn muốn.

## **Kết quả đạt được:**

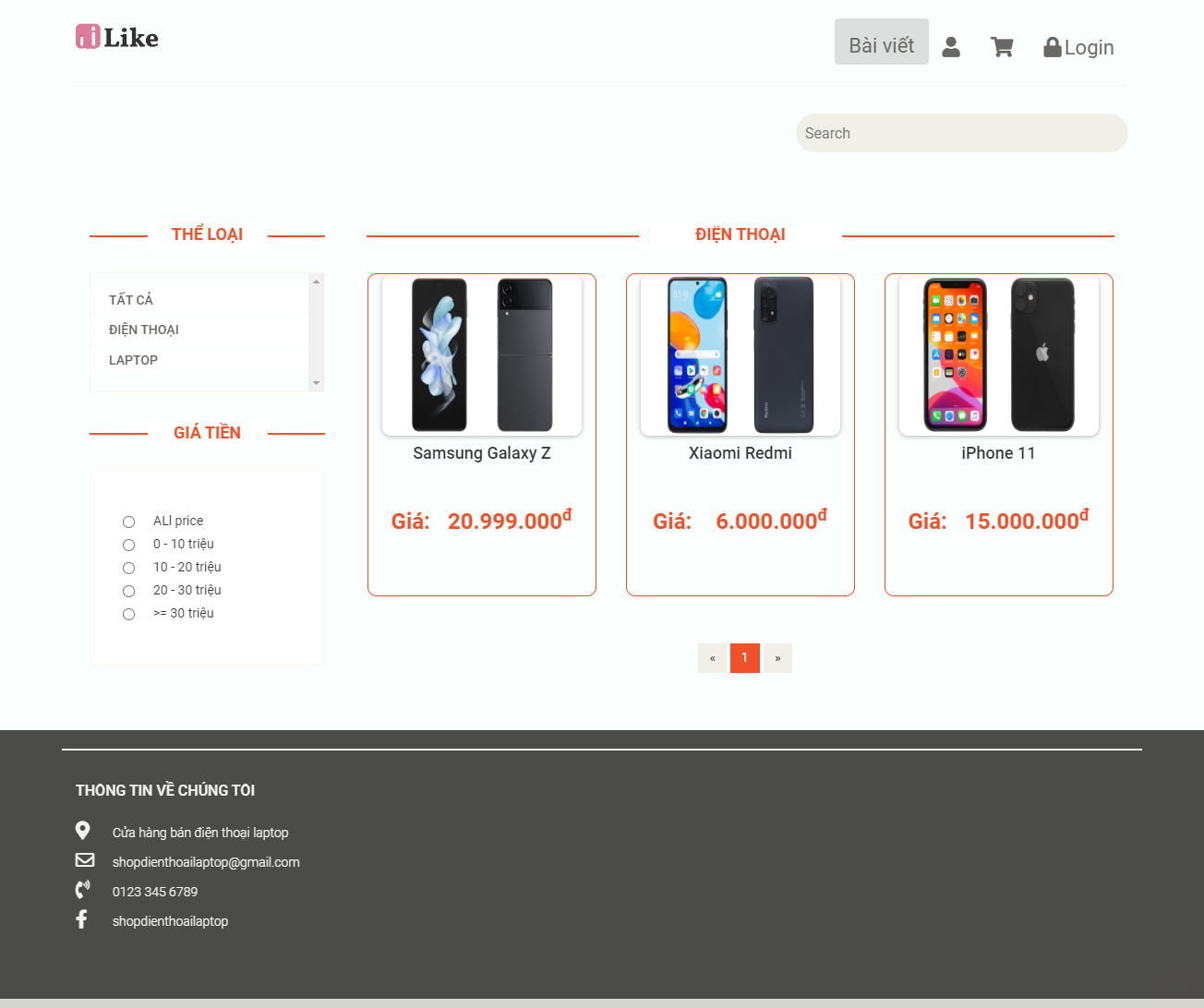
**3.2.1 Trang giao diện người dùng:**

****

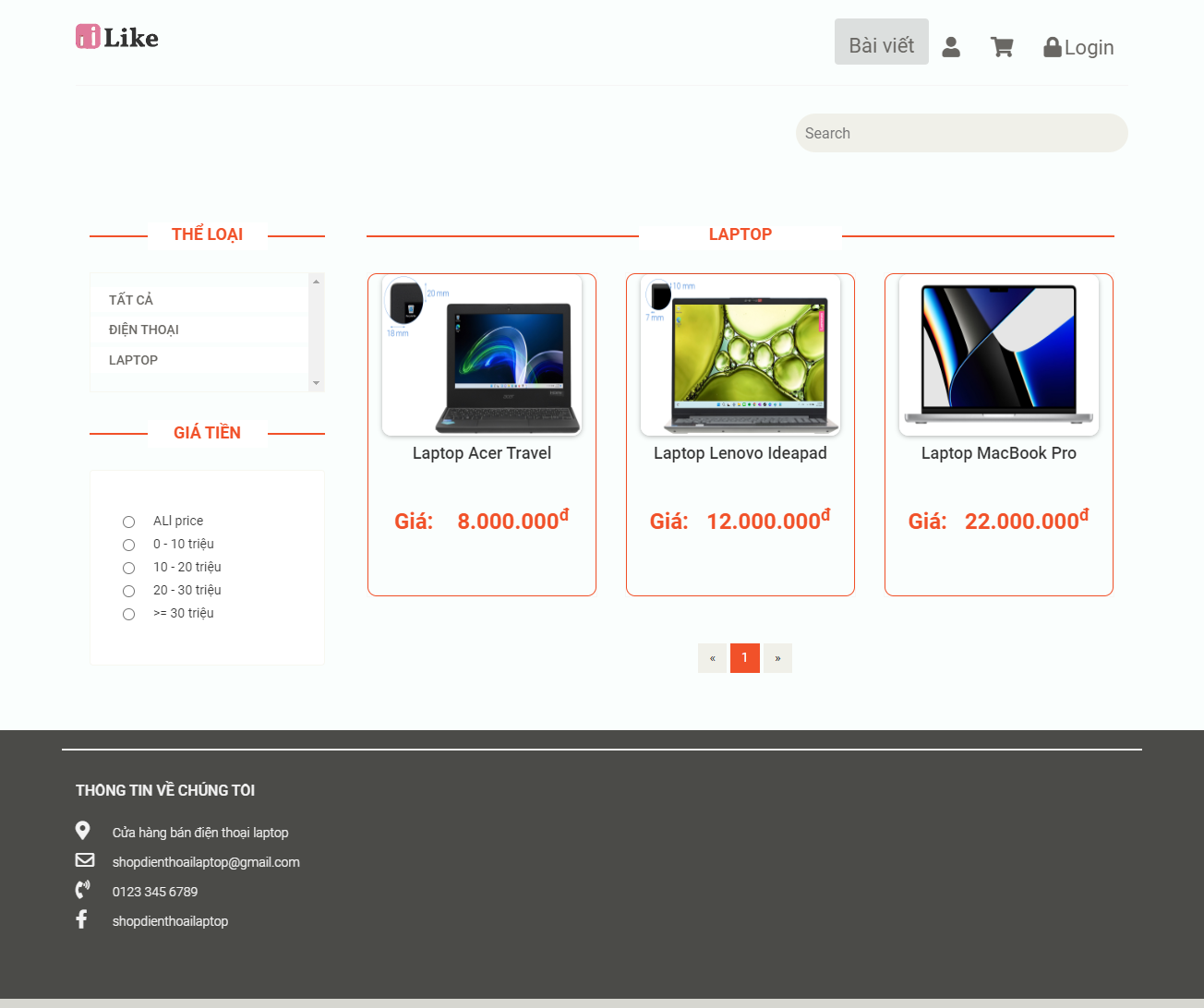
**Hình 3.7 Giao diện trang chủ**



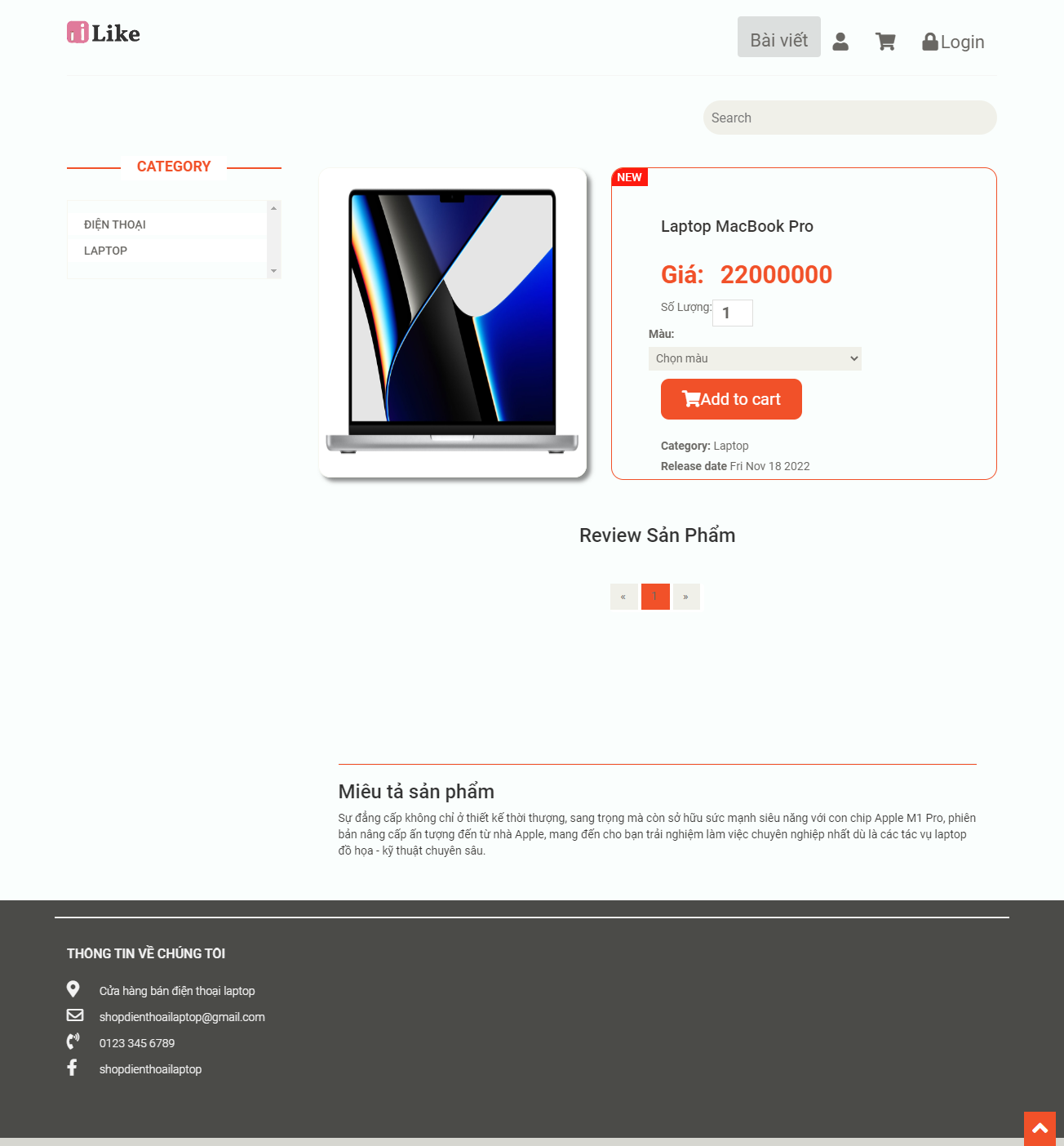
**Hình 3.8 Giao diện tất cả sản phẩm**

****

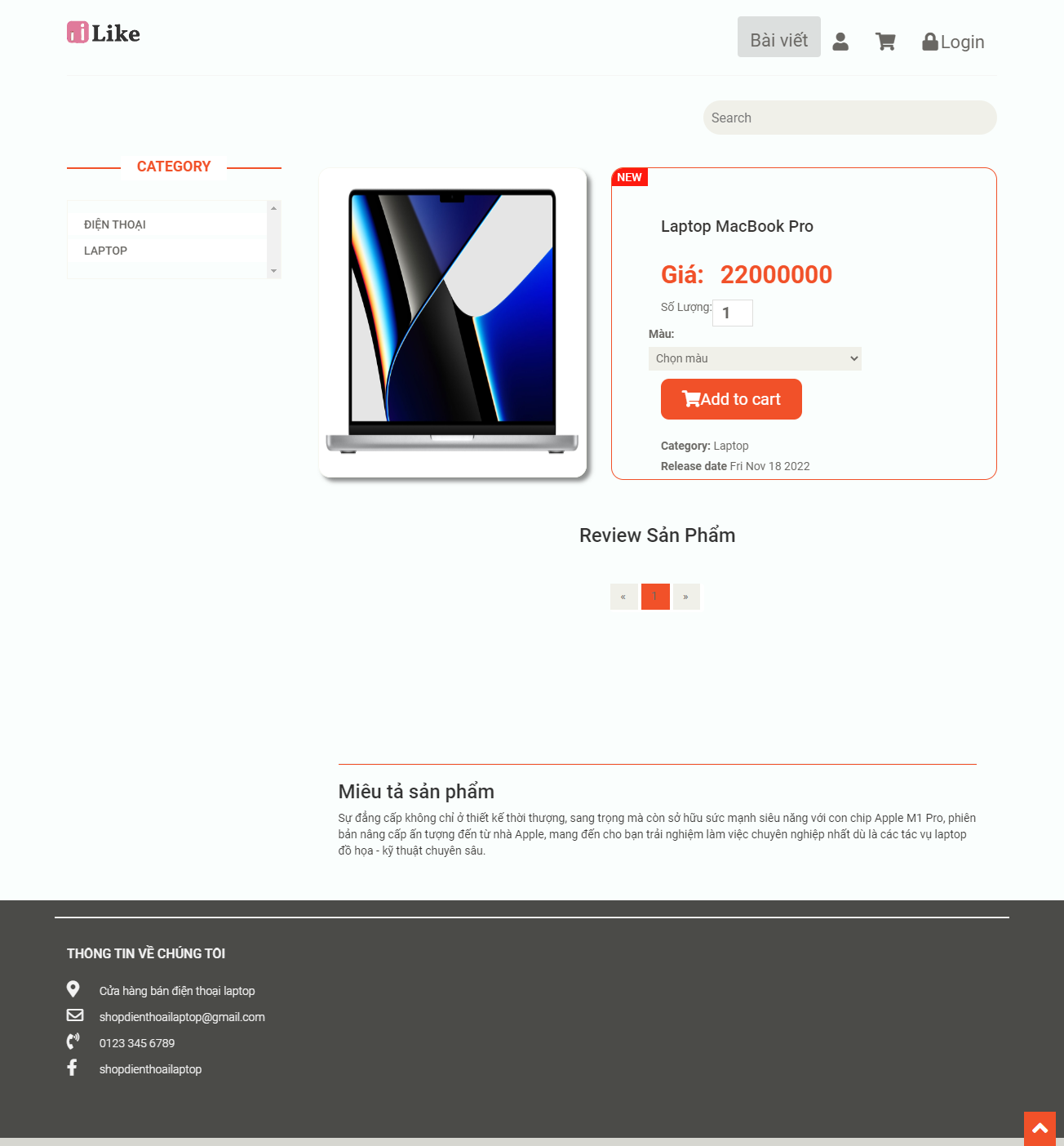
**Hình 3.9 Giao diện sản phẩm là điện thoại**



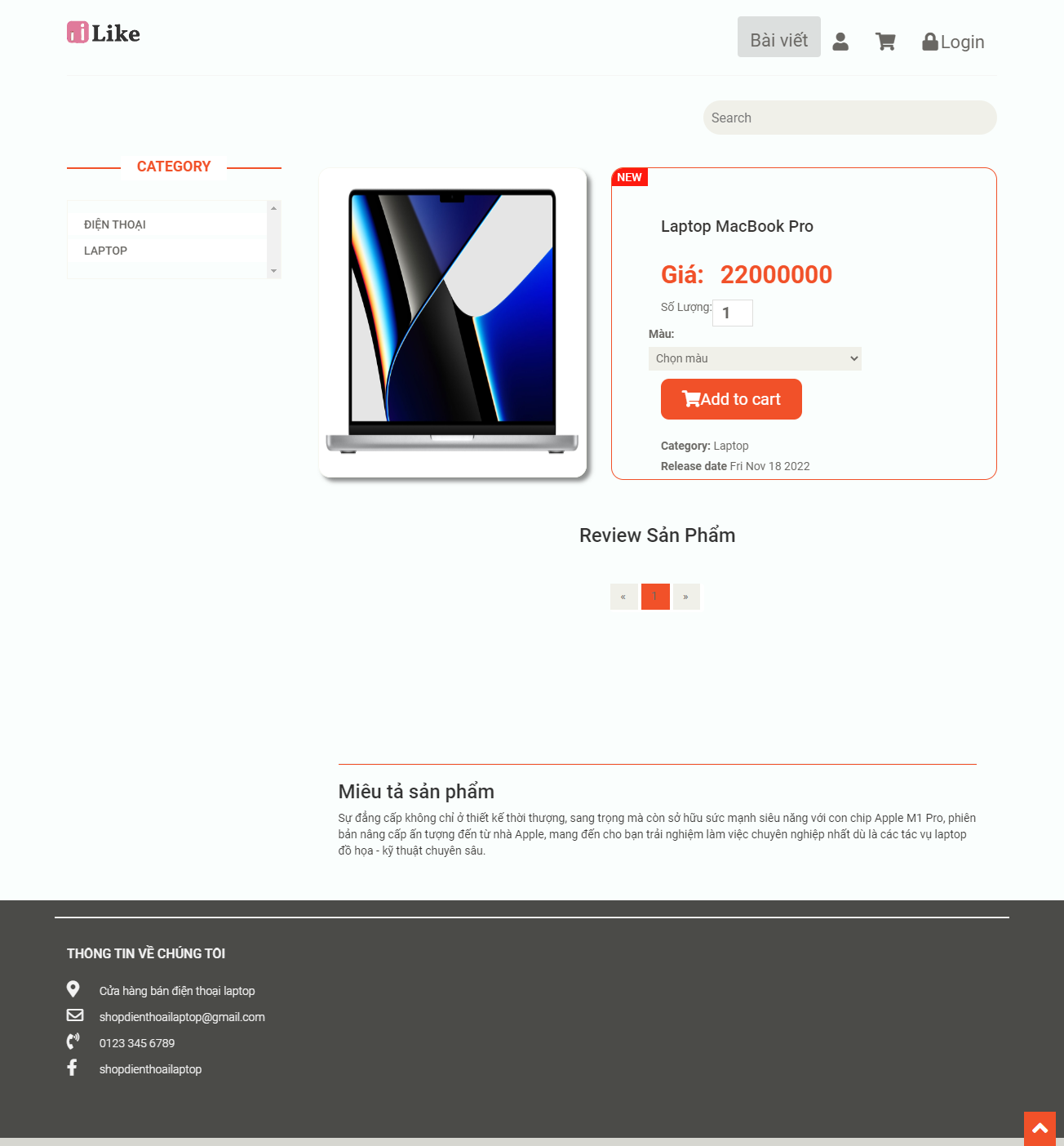
**Hình 3.10 Giao diện sản phẩm là laptop**

****

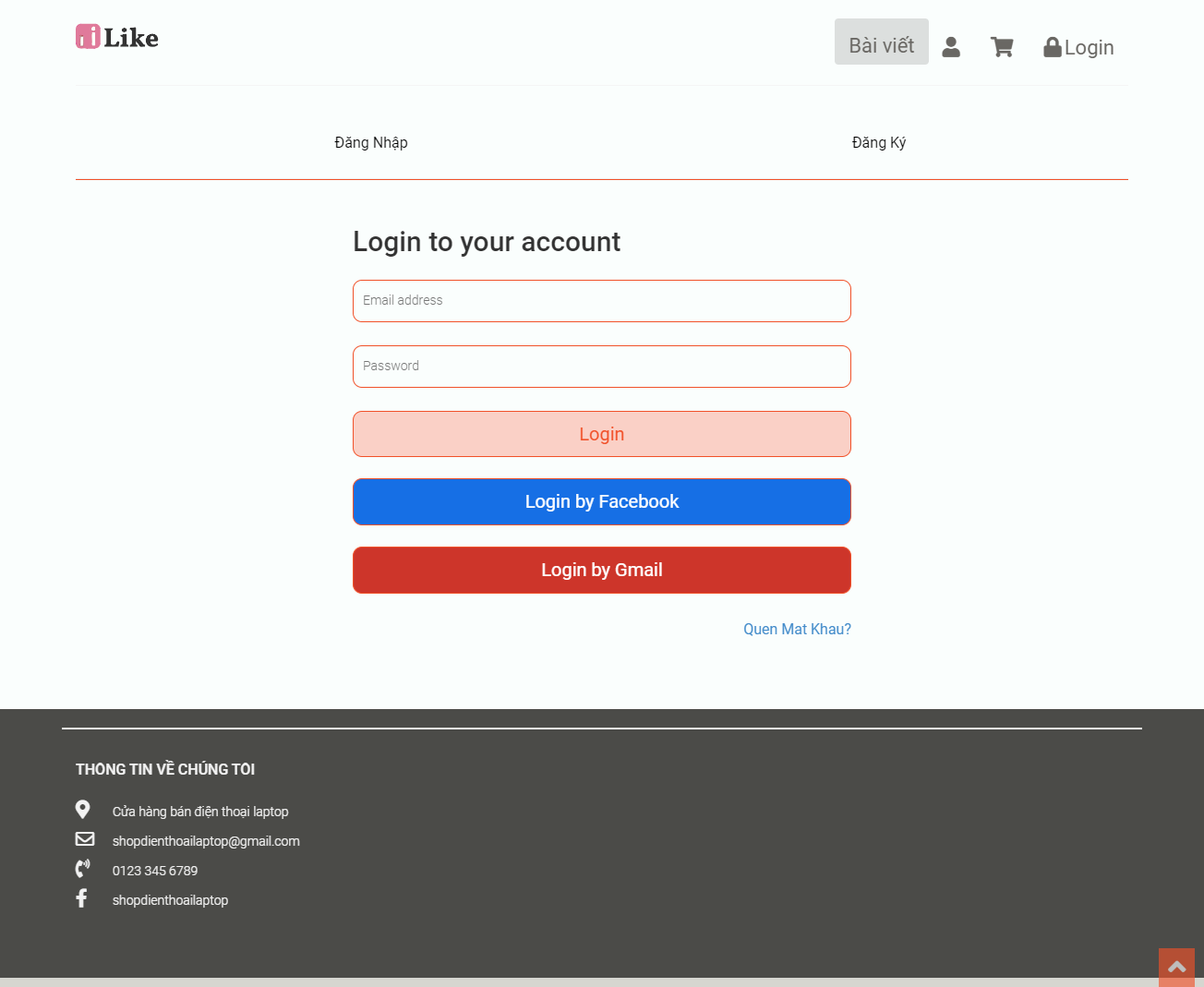
**Hình 3.11 giao diện trang chi tiết sản phẩm**

****

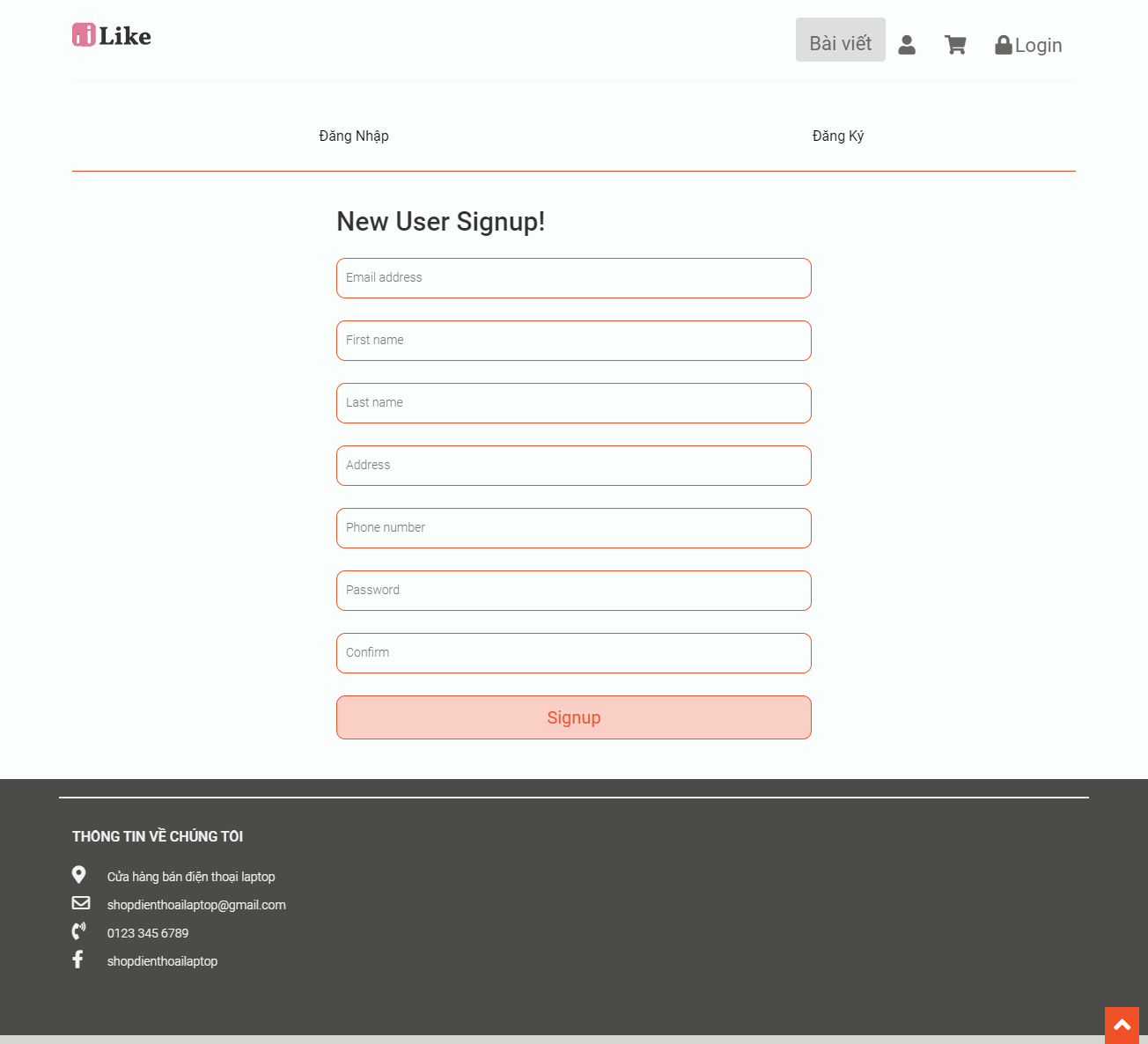
**Hình 3.12 Giao diện giỏ hàng**

****

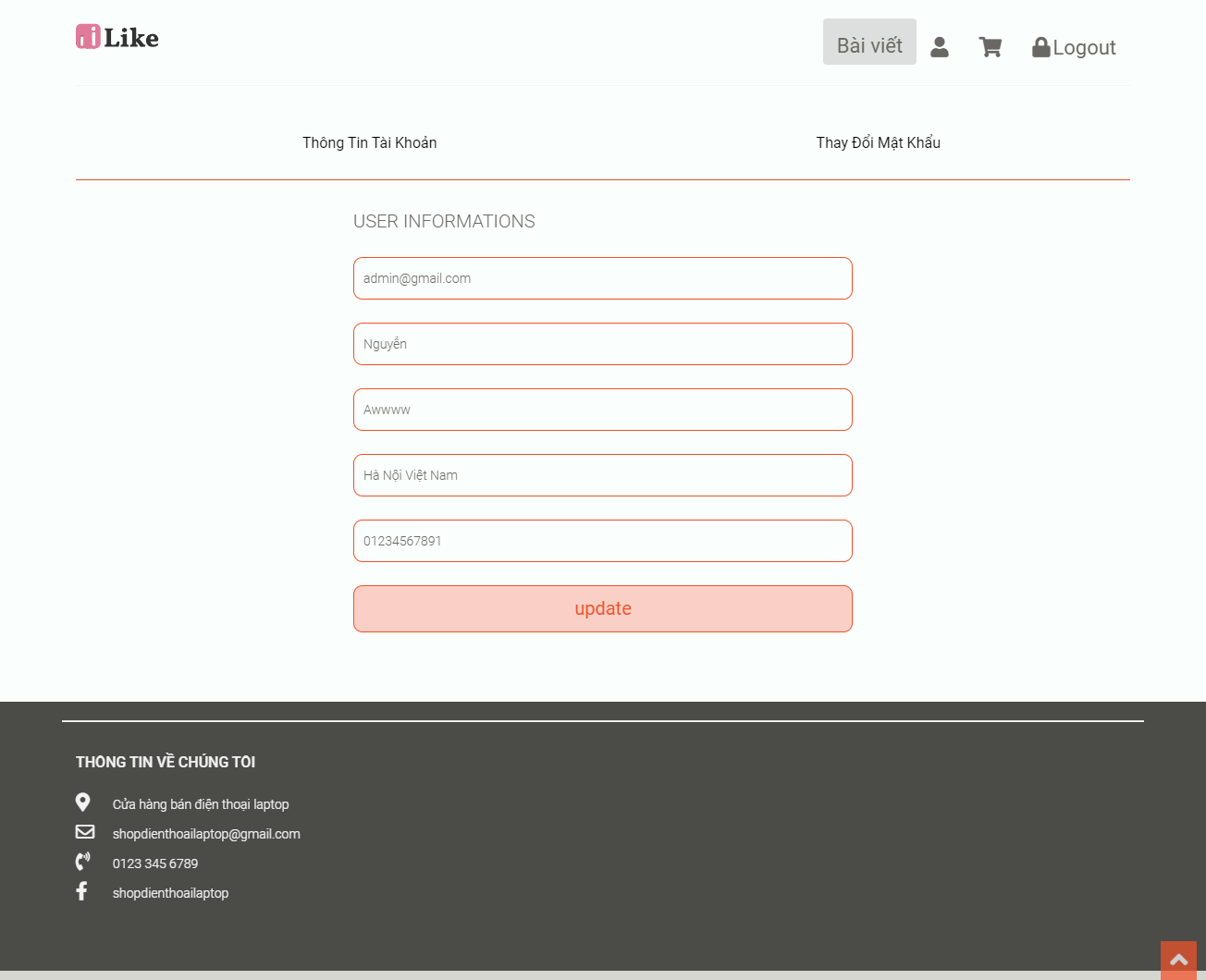
**Hình 3.13 Giao diện trang giỏ hàng**

****

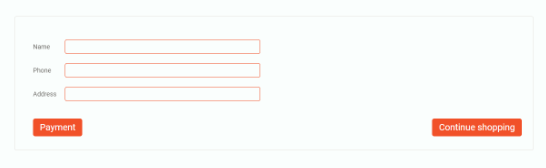
**Hình 3.14 Giao diện trang đăng nhập tài khoản**

****

**Hình 3.15 Giao diện trang đăng kí tài khoản**

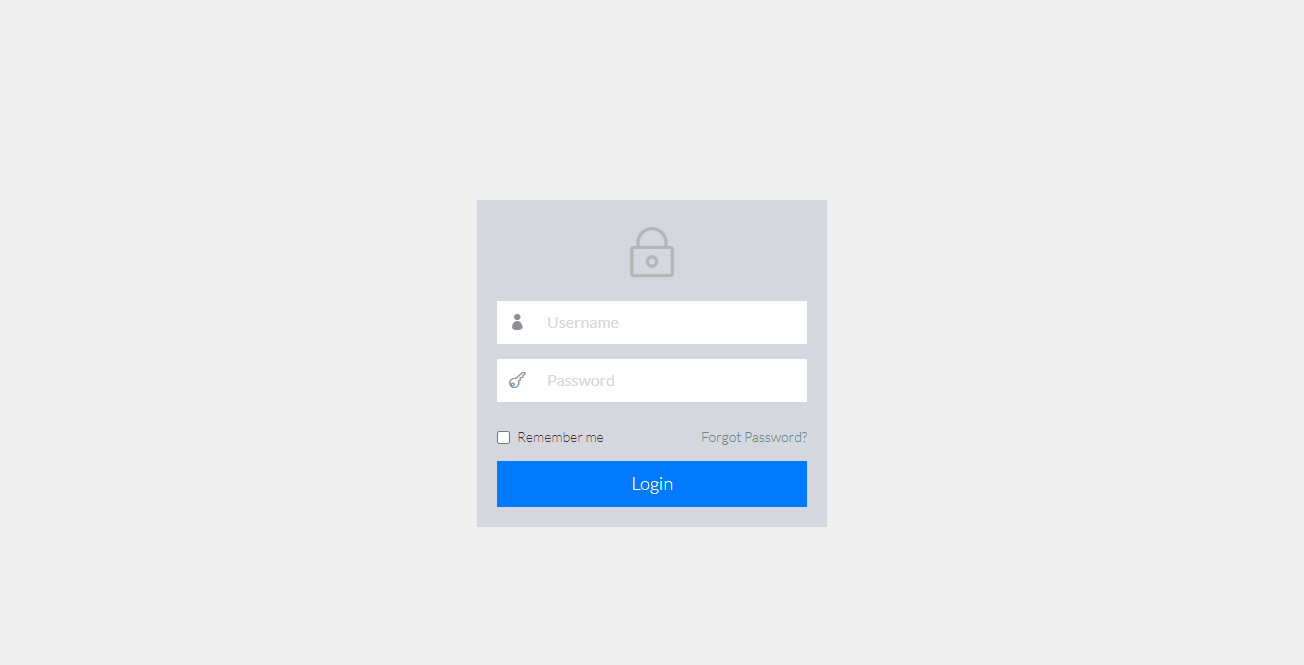
****

**Hình 3.16 Giao diện trang thông tin tài khoản khách hàng**

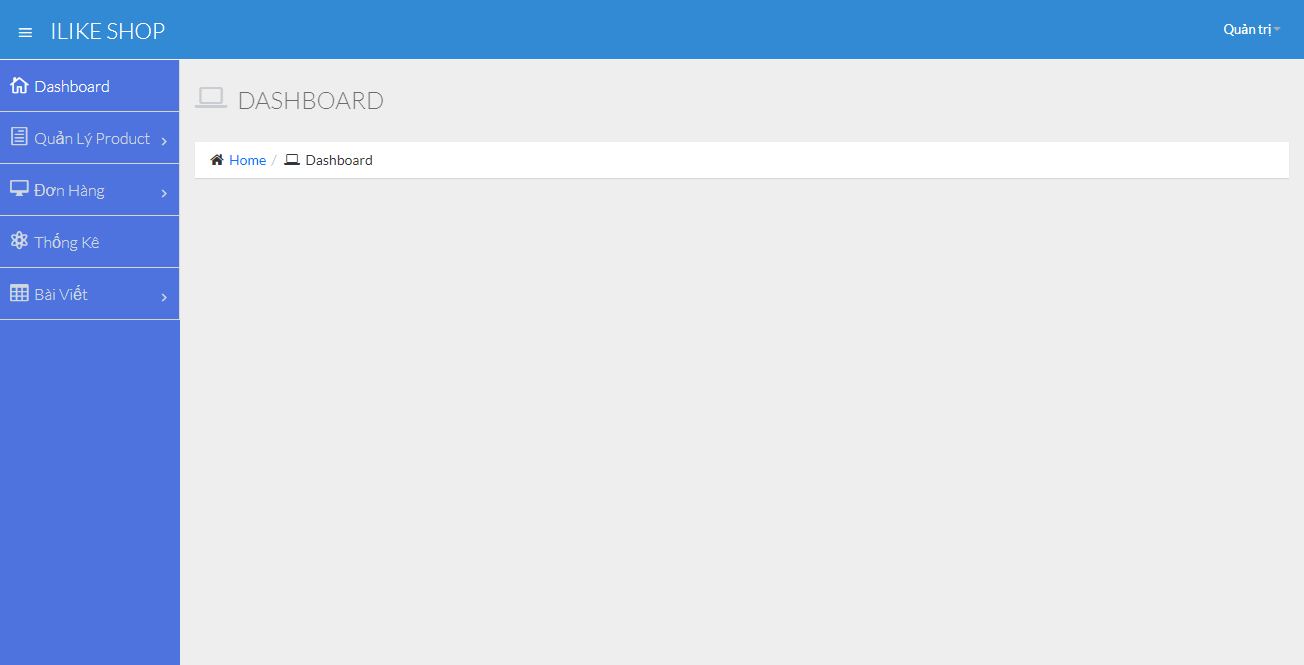
****

**Hình 3.17 giao diện trang thanh toán**

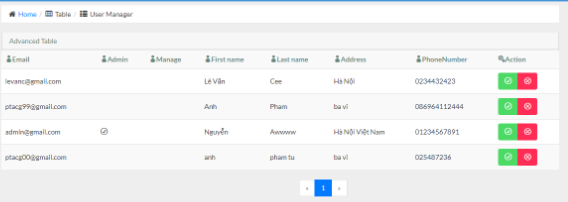
**3.2.2 Trang admin**

****

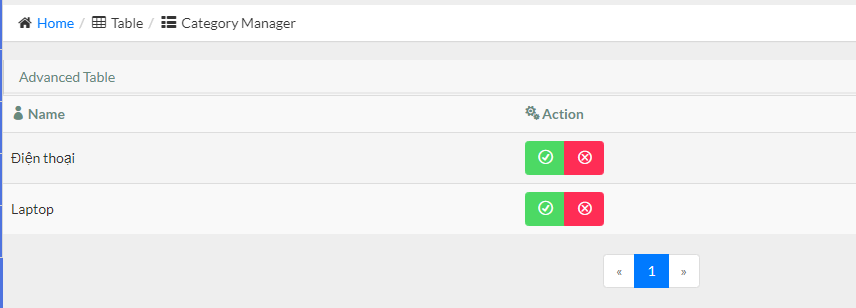
**Hình 3.18 Giao diện trang đăng nhập của admin**

****

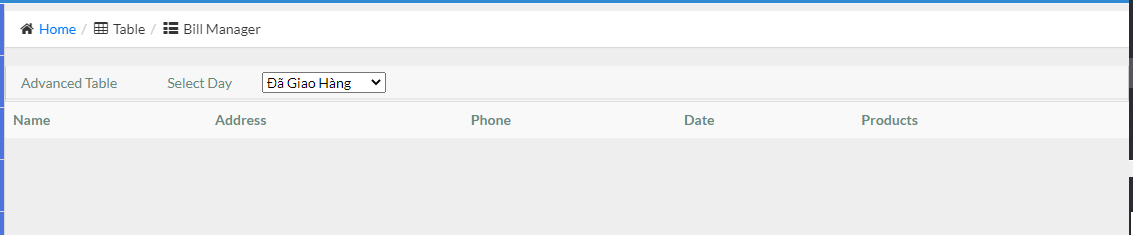
**Hình3.19 Giao diện trang chủ admin**

****

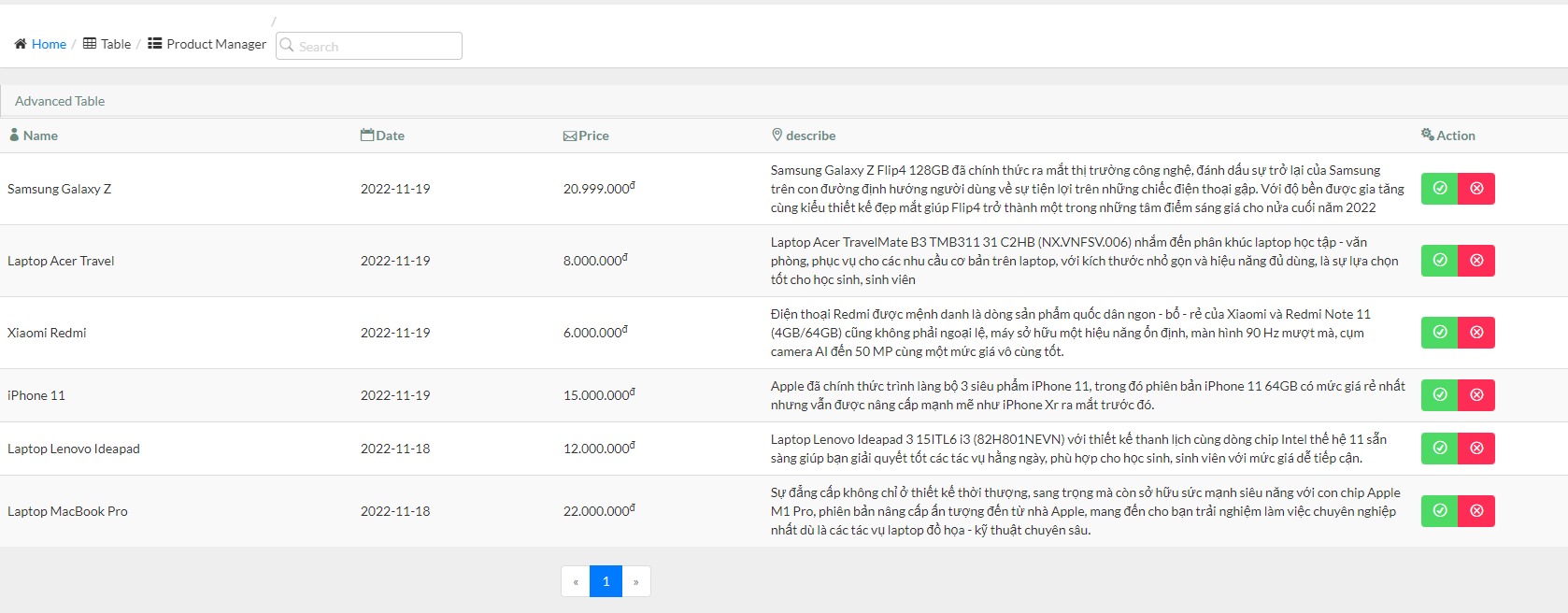
**Hình 3.20 Giao diện quản lí quyền truy cập**

****

**Hình 3.21 Giao diện quản lí danh mục sản phẩm**

****

**Hình 3.22 Giao diện quản lí đơn hàng**

****

**Hình 3.23 Giao diện quản lí danh sách sản phẩm**

# CHƯƠNG 4: KẾT LUẬN

## **4.1. Kết quả đạt được**

Thông qua quá trình phân tích và thiết kế, tìm hiểu các nội dung trên Internet. Em đã hoàn thành được ứng dụng website bán điện thoại online với các chức năng cần thiết cơ bản mà một trang thương mại điện tử:

1. **Đối với người sử dụng**

* Xem chi tiết thông tin giá cả của sản phẩm thương mại và đầy đủ thông tin của cửa hàng
* Cho phép khách hàng đăng ký thông tin để được mua sản phẩm
* Kiểm tra tính hợp lệ của khách hàng đăng nhập hệ thống
* Tạo đơn đặt hàng và chỉnh sửa đơn đặt hàng
* Có thể chạy trên nhiều hệ điều hành
* Trang web thân thiện dễ dàng sử dụng

1. **Đối với người quản trị**

* Xem thông tin của các mục cần quản lý : khách hàng, sản phẩm, đơn đặt hàng…
* Đưa ra form nhập dữ liệu về chi tiết sản phẩm đầy đủ
* Có thể sửa đổi cập nhật các dữ liệu trên.
* Giao diện trang web thân thiện dễ dàng cập nhật hay chỉnh sửa
* Trong việc tạo mới sản phẩm chương trình tự động sinh mã sản phẩm điều này thuận lợi cho người quản trị và dữ liệu luôn đồng nhất
* Các chức năng được thực hiện thông qua giao diện web
* Quản lí được các page của trang web từ banner footer đến các trang tin tức

## **4.2. Những hạn chế và định hướng**

Bên cạnh những kết quả đạt được thì sản phẩm còn có những hạn chế nhất định như chưa thanh toán được bằng hình thức visa card hay các hình thức thanh toán trực tuyến trên thị trường hiện nay…. Nhưng không dừng lại ở đó, em vẫn cố gắng tìm hiểu và hoàn thành chỉnh chu trong khả năng của bản thân. Hoàn chỉnh ứng dụng hơn trong tương lai.

# TÀI LIỆU THAM KHẢO

1. Youtube phạm Ta Ni xây dựng web site bán hàng online MERN stack ( *viết code back-end và front-end theo mẫu MERN Stack* )
2. <https://w3school.com> ( *chỉnh sửa CSS và các code chỉnh sửa giao diện* )
3. Lê văn phùng NXB thông tin và truyền thông. ( *phân tích thiết kế hệ thống thông tin* )
4. Chu kì Quang, *( giáo trình cơ sở dử liệu )* học viện công nghệ bưu chính viễn thông
5. Giáo trình lập trình ứng dụng web với MERN Stack tác giả Sachhoc.com…
6. Nguyễn Văn Du 2013 ( *đồ án tốt nghiệp slide share* )