TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**

---oOo---

**LUẬN VĂN TỐT NGHIỆP**

*Tên đề tài:*

TÌM HIỂU FRAMEWORK VUEJS VÀ NODEJS ÁP DỤNG XÂY DỰNG WEBSITE BÁN LAPTOP

Người hướng dẫn: **ThS. DƯƠNG VĂN ĐEO**

Sinh viên thực hiện:

**1. TRẦN KHÁNH DUY**

**2. PHẠM ĐĂNG HẢI DƯƠNG**

TP HỒ CHÍ MINH – Năm 2022

**LỜI CẢM ƠN**

Trước hết, chúng em gửi lời cảm ơn chân thành đến trường Đại học Công nghệ Sài Gòn và đặc biệt đến Khoa Công nghệ thông tin của trường vì đã tạo điều kiện thuận lợi cho chúng em được đào tạo, nghiên cứu, trao dồi các kĩ năng và thực hiện đề tài luận văn tốt nghiệp này.

Tiếp đến, chúng em trân trọng gửi lời cảm ơn sâu sắc đến thầy Dương Văn Đeo đã dành khoảng thởi gian quý báu không ít để hỗ trợ, hướng dẫn tận tình nhóm chúng em hoàn thành bài luận văn tốt nghiệp này dù gặp nhiều hạn chế trong thời gian qua.

Từ tận đáy lòng chúng em xin cảm ơn các đấng sinh thành đã luôn động viên, ủng hộ cả về vật chất lẫn tinh thần để chúng em có được trang bị những kiến thức quý báu trong suốt quãng thời gian học tập và làm việc tại trường.

Sau cùng, chúng em xin cảm ơn những sự giúp đỡ, hỗ trợ cũng như góp ý từ các anh chị, bạn bè trong quá trình thực hiện đề tài.

Với lượng kiến thức đã và đang tiếp thu, việc tránh khỏi những thiếu sót hay sơ xuất là điều không thể. Chúng em rất mong nhận được thêm các lời nhận xét, góp ý từ các quý thầy cô cũng như các bạn để phát triển đề tài tiến đến phiên bản thương mại và hoàn thiện nhất.

Chúng em chân thành cảm ơn tất cả.

**MỤC LỤC**

[**Chương 1. GIỚI THIỆU** 1](#_Toc108888096)

[**1.1 MỤC TIÊU** 1](#_Toc108888097)

[**1.2 GIẢI PHÁP CÔNG NGHỆ** 2](#_Toc108888098)

[1.2.1 VueJS Framework 5](#_Toc108888099)

[1.2.2 NodeJS Framework 9](#_Toc108888100)

[1.2.3 MySQL 18](#_Toc108888101)

[1.2.4 JSON Web Token (JWT) 19](#_Toc108888102)

[**Chương 2. MÔ TẢ NGHIỆP VỤ** 22](#_Toc108888103)

[**2.1 TỔNG QUAN** 22](#_Toc108888104)

[2.1.1 Quản trị viên 22](#_Toc108888105)

[2.1.2 Người mua hàng 24](#_Toc108888106)

[**2.2 SƠ ĐỒ CHỨC NĂNG** 25](#_Toc108888107)

[**2.3 QUY TRÌNH XỬ LÝ** 26](#_Toc108888108)

[2.3.1 Quy trình đăng ký 26](#_Toc108888109)

[2.3.2 Quy trình đăng nhập 26](#_Toc108888110)

[2.3.3 Quy trình đặt hàng 27](#_Toc108888111)

[2.3.4 Quy trình thanh toán 27](#_Toc108888112)

[**Chương 3. KIẾN TRÚC HỆ THỐNG** 28](#_Toc108888113)

[**3.1 USE-CASE TỔNG QUÁT** 28](#_Toc108888114)

[**3.2 SƠ ĐỒ CHI TIẾT** 29](#_Toc108888115)

[3.2.1 Use-case Đăng nhập (khách hàng) 29](#_Toc108888116)

[3.2.2 Use-case Đăng kí (khách hàng) 30](#_Toc108888117)

[3.2.3 Use-case Quản lý giỏ hàng 31](#_Toc108888118)

[3.2.4 Use-case Quản lý thông tin tài khoản 33](#_Toc108888119)

[3.2.5 Use-case Thanh toán 34](#_Toc108888120)

[3.2.6 Use-case Quản lý nhân viên 36](#_Toc108888121)

[3.2.7 Use-case Quản lý sản phẩm 38](#_Toc108888122)

[3.2.8 Use-case Quản lý linh kiện 40](#_Toc108888123)

[3.2.9 Use-case Quản lý đơn hàng 41](#_Toc108888124)

[3.2.10 Use-case Quản lý khách hàng 42](#_Toc108888125)

[**Chương 4. THÀNH PHẦN XỬ LÝ** 44](#_Toc108888126)

[**4.1 SƠ ĐỒ TUẦN TỰ** 44](#_Toc108888127)

[4.1.1 Sơ đồ tuần tự đăng nhập 44](#_Toc108888128)

[4.1.2 Sơ đồ tuần tự đăng ký 45](#_Toc108888129)

[4.1.3 Sơ đồ tuần tự quản lý giỏ hàng 46](#_Toc108888130)

[4.1.4 Sơ đồ tuần tự quản lý thông tin khách hàng 47](#_Toc108888131)

[4.1.5 Sơ đồ tuần tự thanh toán 48](#_Toc108888132)

[4.1.6 Sơ đồ tuần tự quản lý nhân viên 49](#_Toc108888133)

[4.1.7 Sơ đồ tuần tự quản lý sản phẩm 50](#_Toc108888134)

[4.1.8 Sơ đồ tuần tự quản lý linh kiện 51](#_Toc108888135)

[**Chương 5. THÀNH PHẦN DỮ LIỆU** 52](#_Toc108888136)

[**5.1 SƠ ĐỒ DỮ LIỆU LOGIC** 52](#_Toc108888137)

[**5.2 THIẾT KẾ CƠ SỞ DỮ LIỆU** 53](#_Toc108888138)

[**Chương 6. THÀNH PHẦN GIAO DIỆN** 56](#_Toc108888139)

[**6.1 GIAO DIỆN KHÁCH HÀNG** 56](#_Toc108888140)

[6.1.1 Giao diện đăng nhập khách hàng 56](#_Toc108888141)

[6.1.2 Giao diện đăng ký khách hàng 57](#_Toc108888142)

[6.1.3 Giao diện trang chủ 57](#_Toc108888143)

[6.1.4 Giao diện danh mục sản phẩm 58](#_Toc108888144)

[6.1.5 Giao diện chi tiết sản phẩm 60](#_Toc108888145)

[6.1.6 Giao diện giỏ hàng 61](#_Toc108888146)

[6.1.7 Giao diện thanh toán 62](#_Toc108888147)

[6.1.8 Giao diện thông tin khách hàng 62](#_Toc108888148)

[6.1.9 Giao diện lịch sử đơn hàng 64](#_Toc108888149)

[**6.2 GIAO DIỆN NHÀ QUẢN LÝ (ADMIN)** 65](#_Toc108888150)

[6.2.1 Giao diện đăng nhập 65](#_Toc108888151)

[6.2.2 Giao diện quản lý khách hàng 66](#_Toc108888152)

[6.2.3 Giao diện quản lý nhân viên 66](#_Toc108888153)

[6.2.4 Giao diện thêm nhân viên 67](#_Toc108888154)

[6.2.5 Giao diện sửa nhân viên 68](#_Toc108888155)

[6.2.6 Giao diện quản lý sản phẩm 68](#_Toc108888156)

[6.2.7 Giao diện thêm sản phẩm 69](#_Toc108888157)

[6.2.8 Giao diện sửa sản phẩm 70](#_Toc108888158)

[6.2.9 Giao diện quản lý linh kiện 72](#_Toc108888159)

[6.2.10 Giao diện quản lý đơn hàng 72](#_Toc108888160)

[6.2.11 Giao diện chi tiết đơn hàng 74](#_Toc108888161)

[**Chương 7. TỔNG KẾT** 75](#_Toc108888162)

[**7.1 KẾT LUẬN ĐỀ TÀI** 75](#_Toc108888163)

[**7.2 CHỨC NĂNG** 75](#_Toc108888164)

[7.2.1 Chức năng hoàn thiện 75](#_Toc108888165)

[7.2.2 Chức năng hạn chế 76](#_Toc108888166)

[**TÀI LIỆU THAM KHẢO** 77](#_Toc108888167)

**MỤC LỤC CÁC HÌNH VẼ**

[**Hình 1- 1**: Mô hình Client – Server 3](#_Toc108817144)

[**Hình 1- 2** : Web Services 3](#_Toc108817145)

[**Hình 1- 3** : Client và Server tương tác thông qua API 4](#_Toc108817146)

[**Hình 1- 4** : REST API 4](#_Toc108817147)

[**Hình 1- 5**: Vuejs sở hữu nhiều ưu điểm nổi bật 5](#_Toc108817148)

[**Hình 1- 6**: Cài đặt Vue 6](#_Toc108817149)

[**Hình 1- 7**: Vue, React và Angular 7](#_Toc108817150)

[**Hình 1- 8**: Số sao trên các dự án GitHub cho Angular, React và Vue 8](#_Toc108817151)

[**Hình 1- 9**: Các công ty sử dụng Vue.js 9](#_Toc108817152)

[**Hình 1- 10**: Giới thiệu NodeJS 10](#_Toc108817153)

[**Hình 1- 11**: Cài đặt NodeJS 12](#_Toc108817154)

[**Hình 1- 12**: Cấu trúc rẽ nhánh trong NodeJS 12](#_Toc108817155)

[**Hình 1- 13**: Cách dùng Async 14](#_Toc108817156)

[**Hình 1- 14**: Cách dùng lodash 14](#_Toc108817157)

[**Hình 1- 15** : Cách dùng Cheerio 15](#_Toc108817158)

[**Hình 1- 16**: Cách dùng Nodemailer 15](#_Toc108817159)

[**Hình 1- 17**: Cách dùng Faker 15](#_Toc108817160)

[**Hình 1- 18**: Cách dùng Morgan 16](#_Toc108817161)

[**Hình 1- 19**: Cách dùng Http-Errors 16](#_Toc108817162)

[**Hình 1- 20**: Cách dùng Moment 17](#_Toc108817163)

[**Hình 1- 21**: Logo MySQL 18](#_Toc108817164)

[**Hình 1- 22**: JWT 19](#_Toc108817165)

[**Hình 1- 23**: Cấu trúc của JWT 20](#_Toc108817166)

[**Hình 1- 24**: Sơ đồ hoạt động JWT 21](#_Toc108817167)

[**Bảng 1- 1**: So sánh giữa Angular, React và Vue[6] 8](#_Toc108817168)

[**Hình 2- 1**: Sơ đồ chức năng tổng quát 25](#_Toc108817176)

[**Hình 2- 2**: Quy trình đăng ký 26](#_Toc108817177)

[**Hình 2- 3**: Quy trình đăng nhập 26](#_Toc108817178)

[**Hình 2- 4**: Quy trình đặt hàng 27](#_Toc108817179)

[**Hình 2- 5**: Quy trình thanh toán 27](#_Toc108817180)

[**Hình 3- 1**: Sơ đồ Use-Case tổng quát 28](#_Toc108817187)

[**Hình 3- 2**: Sơ đồ Use-case đăng nhập (khách hàng) 29](#_Toc108817188)

[**Hình 3- 3**: Sơ đồ Use-case đăng kí (khách hàng) 30](#_Toc108817189)

[**Hình 3- 4**: Sơ đồ Use-case quản lý giỏ hàng 31](#_Toc108817190)

[**Hình 3- 5**: Sơ đồ Use-case quản lý thông tin tài khoản 33](#_Toc108817191)

[**Hình 3- 6**: Sơ đồ Use-case thanh toán 34](#_Toc108817192)

[**Hình 3- 7**: Sơ đồ Use-case quản lý nhân viên 36](#_Toc108817193)

[**Hình 3- 8**: Sơ đồ Use-case quản lý sản phẩm 38](#_Toc108817194)

[**Hình 3- 9**: Sơ đồ use-case quản lý linh kiện 40](#_Toc108817195)

[**Hình 3- 10**: Sơ đồ use-case quản lý đơn hàng 41](#_Toc108817196)

[**Hình 3- 11**: Sơ đồ use-case quản lý khách hàng 42](#_Toc108817197)

[**Bảng 3- 1**: Mô tả Use-case đăng nhập (khách hàng) 29](#_Toc108817289)

[**Bảng 3- 2**: Mô tả Use-case đăng kí (khách hàng) 30](#_Toc108817290)

[**Bảng 3- 3**: Mô tả Use-case quản lý giỏ hàng 32](#_Toc108817291)

[**Bảng 3- 4**: Mô tả Use-case quản lý thông tin tài khoản 33](#_Toc108817292)

[**Bảng 3- 5**: Mô tả Use-case thanh toán 35](#_Toc108817293)

[**Bảng 3- 6**: Mô tả Use-case quản lý nhân viên 36](#_Toc108817294)

[**Bảng 3- 7**: Mô tả Use-case quản lý sản phẩm 38](#_Toc108817295)

[**Bảng 3- 8**: Mô tả use-case quản lý linh kiện 40](#_Toc108817296)

[**Bảng 3- 9**: Mô tả use-case quản lý đơn hàng 41](#_Toc108817297)

[**Bảng 3- 10**: Mô tả use-case quản lý khách hàng 42](#_Toc108817298)

[**Hình 4- 1**: Sơ đồ tuần tự đăng nhập 44](#_Toc108875544)

[**Hình 4- 2**: Sơ đồ tuần tự đăng ký 45](#_Toc108875545)

[**Hình 4- 3**: Sơ đồ tuần tự quản lý giỏ hàng 46](#_Toc108875546)

[**Hình 4- 4**: Sơ đồ tuần tự quản lý thông tin khách hàng 47](#_Toc108875547)

[**Hình 4- 5**: Sơ đồ tuần tự thanh toán 48](#_Toc108875548)

[**Hình 4- 6**: Sơ đồ tuần tự quản lý nhân viên 49](#_Toc108875549)

[**Hình 4- 7**: Sơ đồ tuần tự quản lý sản phẩm 50](#_Toc108875550)

[**Hình 4- 8**: Sơ đồ tuần tự quản lý linh kiện 51](#_Toc108875551)

[**Hình 5- 1**: Sơ đồ dữ liệu Logic 52](#_Toc108887921)

[**Bảng 5- 1**: Bảng dữ liệu "nguoidung" 53](#_Toc108887936)

[**Bảng 5- 2**: Bảng dữ liệu "sanpham" 53](#_Toc108887937)

[**Bảng 5- 3**: Bảng dữ liệu "hang" 53](#_Toc108887938)

[**Bảng 5- 4**: Bảng dữ liệu "CPU" 54](#_Toc108887939)

[**Bảng 5- 5**: Bảng dữ liệu "hedieuhanh" 54](#_Toc108887940)

[**Bảng 5- 6**: Bảng dữ liệu "ocung" 54](#_Toc108887941)

[**Bảng 5- 7**: Bảng dữ liệu "ram" 54](#_Toc108887942)

[**Bảng 5- 8**: Bảng dữ liệu "chitietdonhang" 54](#_Toc108887943)

[**Bảng 5- 9**: Bảng dữ liệu "donhang" 54](#_Toc108887944)

[**Hình 6- 1**: Giao diện đăng nhập khách hàng 56](#_Toc108887946)

[**Hình 6- 2**: Giao diện đăng ký khách hàng 57](#_Toc108887947)

[**Hình 6- 3**: Giao diện trang chủ 58](#_Toc108887948)

[**Hình 6- 4**: Giao diện danh mục sản phẩm 59](#_Toc108887949)

[**Hình 6- 5**: Giao diện chi tiết sản phẩm 60](#_Toc108887950)

[**Hình 6- 6**: Giao diện giỏ hàng trống 61](#_Toc108887951)

[**Hình 6- 7**: Giao diện giỏ hàng 62](#_Toc108887952)

[**Hình 6- 8**: Giao diện thanh toán 62](#_Toc108887953)

[**Hình 6- 9**: Giao diện chi tiết khách hàng 63](#_Toc108887954)

[**Hình 6- 10**: Giao diện theo dỗi đơn hàng 64](#_Toc108887955)

[**Hình 6- 11**: Giao diện chi tiết đơn hàng 65](#_Toc108887956)

[**Hình 6- 12**: Giao diện đăng nhập cho admin 66](#_Toc108887957)

[**Hình 6- 13**: Giao diện quản lý khách hàng 66](#_Toc108887958)

[**Hình 6- 14**: Giao diện quản lý nhân viên 67](#_Toc108887959)

[**Hình 6- 15**: Giao diện thêm nhân viên 67](#_Toc108887960)

[**Hình 6- 16**: Giao diện sửa nhân viên 68](#_Toc108887961)

[**Hình 6- 17**: Giao diện quản lý sản phẩm 68](#_Toc108887962)

[**Hình 6- 18**: Giao diện quản lý sản phẩm 69](#_Toc108887963)

[**Hình 6- 19**: Giao diện thêm sản phẩm 69](#_Toc108887964)

[**Hình 6- 20**: Giao diện thêm sản phẩm 70](#_Toc108887965)

[**Hình 6- 21**: Giao diện sửa sản phẩm 71](#_Toc108887966)

[**Hình 6- 22**: Giao diện quản lý linh kiện 72](#_Toc108887967)

[**Hình 6- 23**: Giao diện quản lý đơn hàng 72](#_Toc108887968)

[**Hình 6- 24**: Giao diện quản lý đơn hàng 73](#_Toc108887969)

[**Hình 6- 25**: Giao diện chi tiết đơn hàng 74](#_Toc108887970)

# **Chương 1. GIỚI THIỆU**

## **1.1 MỤC TIÊU**

Trong cuộc sống hiện đại ngày nay, Internet là một phần không thể thiếu của đời sống hằng ngày. Internet có vị thế và vai trò đặc biệt quan trọng trong sự phát triển của xã hội cũng như phát triển các khía cạnh của các ngành nghề khác. Với sự phát triển có thể nói là vượt bậc đáng kể của internet trong các năm qua, khi mà phần trăm của người sử dụng Internet trên toàn thế giới đã và đang lên đến con số đáng kinh ngạc là 59,5% tương đương 4,66 tỉ người tăng 316 triệu người (7,3%) so với cùng kì năm ngoái theo VOV.VN và con số này tiếp tục tăng theo cấp số cộng. Còn ở Việt Nam con số này lên đến 72,1 triệu người tương đương 73,2% tổng dân số và tăng 3,4 triệu người tương đương 4,9% trong năm 2022 theo lptech.asia.[1]

Như đã biết chúng ta vừa vượt qua được giai đoạn kinh hoàng hay tệ hơn là khủng hoảng từ đại dịch COVID-19 và đang bắt đầu hồi phục, trong giai đoạn đó việc mua sắm online cũng như mua sắm các thiết bị có kết nối Internet như laptop, điện thoại (smartphone), máy tính bảng (tablet), …được đẩy mạnh. Đặc biệt là mảng laptop, thiết bị vi tính là mảng tăng trưởng mạnh nhất với lợi thế đa dạng, đa dụng nhằm đáp ứng toàn diện mọi nhu cầu cho các hoạt động làm việc cũng như mọi liên lạc trong công việc lẫn đời sống giải trí thông qua Internet, lợi thế vượt trội hơn hẳn so với các thiết bị cạnh tranh chủ yếu khác như điện thoại (smartphone), máy tính bảng (tablet).

Từ các yếu tố trên chúng em quyết định thực hiện đề tài “Xây dựng website bán laptop trực tuyến”. Phần vì những ưu thế của laptop đã được kể trên , phần vì những ưu điểm của việc kinh doanh online ở thời điểm hiện tại và tương lai tới.

Thông qua những báo cáo và thu thập thông tin từ các trang báo cũng như các nguồn đáng tin cậy, việc kinh doanh online hay lớn hơn là các sàn thương mại điện tử đã trở thành “miếng bánh ngon” cho các nhà đầu tư cũng như các dự án start-up từ nhỏ vừa cho đến lớn, chẳng hạn như những cái tên lớn ebay của Mỹ, Alibaba của Trung Quốc hay cái tên thân thuộc Shopee và Lazada đến từ nước bạn Singapore.

Việc kinh doanh qua internet trở nên phổ biến và ngày càng dễ dàng tiếp cận và sử dụng cho cả người dùng lẫn nhà kinh doanh. Cộng việc kết hợp với đơn vị vận chuyển thì khoảng cách địa lý không còn vấn đề nan giải như trước kia nữa. Dựa vào đó quyết định thực hiện đề tài “Xây dựng website bán laptop trực tuyến” lại càng trở nên đúng đắn. Với mục tiêu xây dựng Website là nơi mà khách hàng có thể truy cập để xem thông tin các sản phẩm và mua ngay sản phẩm, giao diện website dễ dàng sử dụng, thân thiện người dùng. Xây dựng các danh mục giúp khách hàng dễ dàng nhận biết các sản phẩm. Đối với nhà kinh doanh có các quy trình quản lý đơn hàng, lọc các đơn hàng đang vận chuyển. Thông tin sản phẩm, danh mục được cập nhật liên tục nhờ tính năng quản lý sản phẩm, quản lý danh mục và nhiều chức năng khác.

**Đối với khách hàng**

* Tìm kiếm
* Loại sản phẩm
* Chi tiết sản phẩm
* Giỏ hàng
* Đặt hàng
* Tình trạng đơn hàng

**Đối với người quản lý**

* Quản lý sản phẩm
* Quản lý người dùng
* Quản lý nhân viên
* Quản lý danh mục
* Quản lý hóa đơn

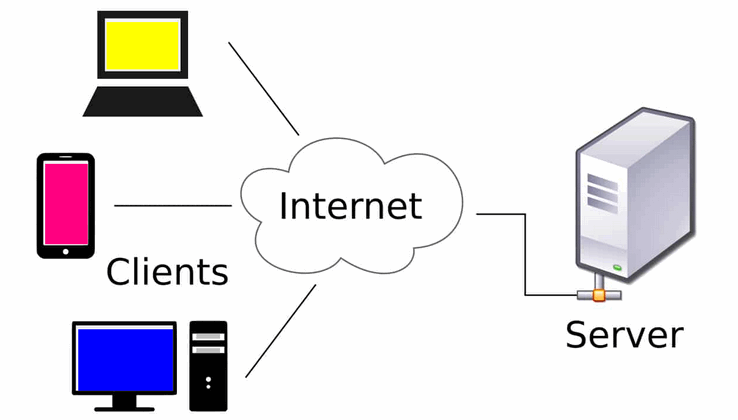
## **1.2 GIẢI PHÁP CÔNG NGHỆ**

Website còn gọi là Web, là một tập hợp trang web, thường chỉ nằm trong một tên miềm trên World Wide Web (WWW – mạng lưới toàn cầu) [2] của Internet. Một trang web được xây dựng từ HTML (HyperText Markup Language là ngôn ngữ dùng để tạo ra các thành phần của trang web), CSS (Cascading Style Sheets được dùng để tạo bố cục, trang trí trang web), JavaScript tạo ra các hành động. Theo thời gian yêu cầu về các chức năng, giao diện người dùng của các trang web ngày càng cao. Nhằm đáp ứng nhu cầu đó nhiều công nghệ, khung phần mềm được ra đời. Angular.js, React.js, Vue.js là ba khung phần mềm phổ biến hiện nay. Trong đó, VueJS được đánh giá là framework hàng đầu trong xây dựng, phát triển [giao diện người dùng](https://bizfly.vn/techblog/ui-ux-la-gi.html) (UI). Các ứng dụng được phát triển bởi VueJS cũng có tốc độ tải trang, xử lý rất nhanh. Ngoài ra, lập trình viên cũng có thể tạo các trang SPA - [Single Page Application](https://bizfly.vn/techblog/single-page-application-la-gi.html) [3] có độ phức tạp cao nếu kết hợp VueJS với SFC (Single-File Component)[4] và thư viện hỗ trợ.

Mô hình client-server[9] , bản chất thì nó là 2 máy tính giao tiếp và truyền tải dữ liệu cho nhau.

Máy tính đóng vai trò là máy khách – **Client**: Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.

Máy tính đóng vai trò là máy chủ – **Server**: Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn. Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP,...



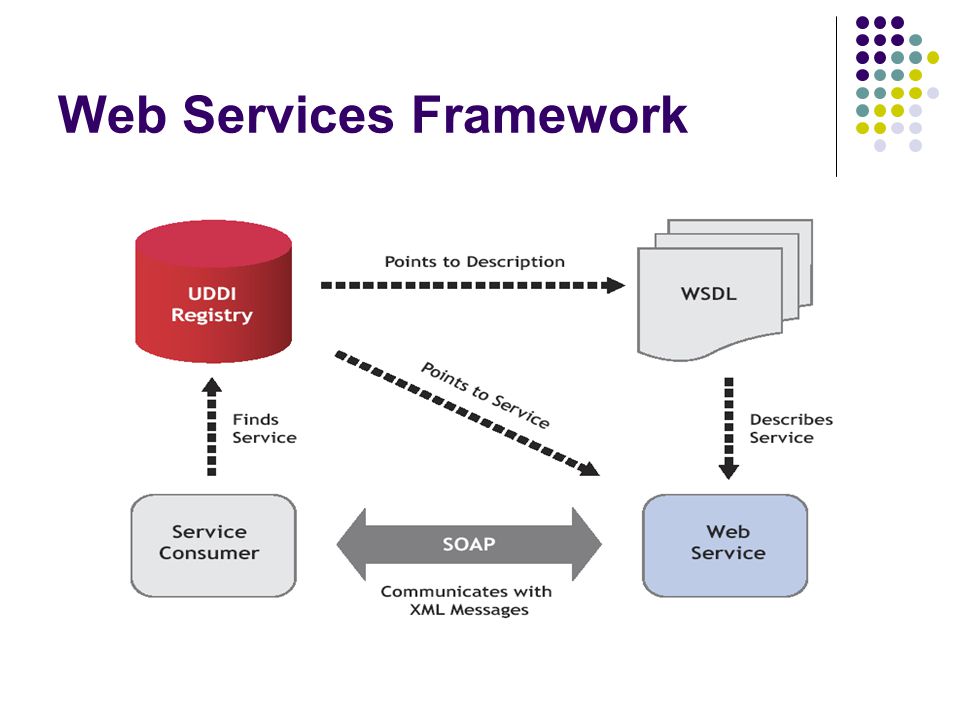
**Hình 1- 1**: Mô hình Client – Server

Web service là tập hợp các giao thức và tiêu chuẩn mở được sử dụng để trao đổi dữ liệu giữa các ứng dụng hoặc giữa các hệ thống. Nền tảng web service cơ bản là XML HTTP. Tất cả các web service chuẩn đều hoạt động bằng các thành phần sau:

- SOAP (là viết tắt của Simple Object Access Protocol) – giao thức truy cập đối tượng đơn giản: SOAP là một giao thức dựa trên XML đơn giản cho phép các ứng dụng trao đổi thông tin qua HTTP.

- UDDI (Universal Description, Discovery and Integration): UDDI là một tiêu chuẩn dựa trên XML để mô tả, xuất bản và tìm kiếm các dịch vụ web.

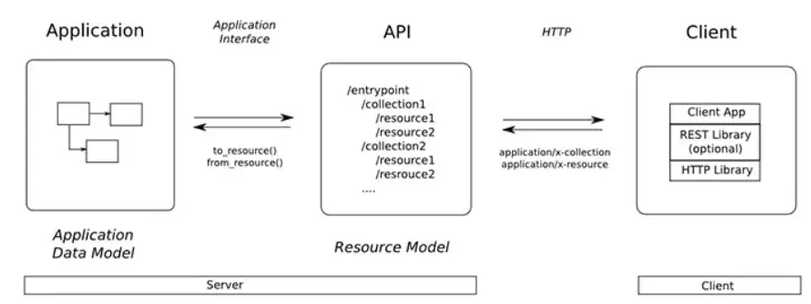
- WSDL (Web Services Description Language) – ngôn ngữ định nghĩa web service: WSDL là một ngôn ngữ dựa trên XML để mô tả các dịch vụ web và cách truy cập chúng.



**Hình 1- 2** : Web Services

API là viết tắt của Application Programming Interface – Giao diện lập trình ứng dụng. API là một tập hợp các quy trình, giao thức và công cụ để xây dựng các ứng dụng phần mềm.

Web API là dạng nâng cao của ứng dụng web để cung cấp dịch vụ trên các thiết bị khác nhau như máy tính xách tay, thiết bị di động và các thiết bị khác. Ứng dụng web giúp trao đổi thông tin trên internet và cũng giúp thực hiện một giao dịch an toàn trên các trang web.

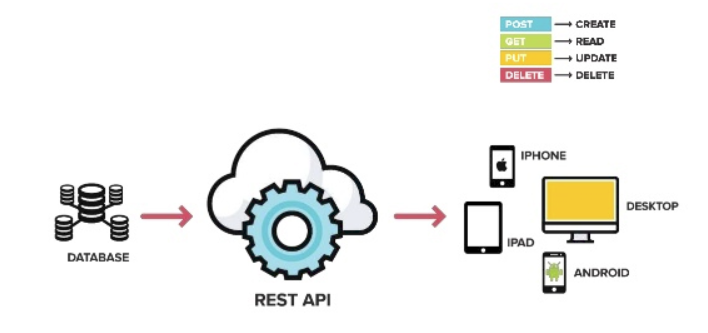


**Hình 1- 3** : Client và Server tương tác thông qua API

Restful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống ( tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và truyền tải qua HTTP.

* GET: để lấy một tài nguyên
* POST: để tạo tài nguyên
* PUT: để thay đổi trạng thái hoặc cập nhật tài nguyên, có thể là một đối tượng, tệp hoặc khối
* DELETE: để loại bỏ.

Đây là phương thức hoạt động được gọi là CRUD tương ứng với một Create, Read, Update, Delete



**Hình 1- 4** : REST API

### 1.2.1 VueJS Framework

#### 1.2.1.1 Giới thiệu

Vuejs được gọi tắt là Vue (được phát âm là /vjuː/, nó giống như từ view trong tiếng Anh). Vue.js là một framework rất linh động được dùng phổ biến để xây dựng nên các giao diện người dùng. Hoàn toàn khác với các framework nguyên khối thì Vue thường sở hữu thiết kế từ đầu theo những hướng cho phép cũng như khuyến khích làm việc để phát triển dễ dàng hơn các ứng dụng theo từng bước một.

Một  khi đã phát triển lớp giao diện (view layer) thì người dùng chỉ cần sử dụng loại thư viện lõi của Vue. Ngoài ra, nếu như bạn kết hợp với các kỹ thuật thiên hướng hiện đại thì Vue cũng có thể đáp ứng được dễ dàng mọi nhu cầu xây dựng ứng dụng của một trang với độ phức tạp cao hơn.

***Lịch sử ra đời*** [5]

Vue đã được tạo ra và phát triển bởi Evan You khi ông đang làm kỹ sư tại công ty Google. Lúc Evan đang sử dụng Angular cho một số dự án khác thì đã có phát biểu rằng: “Điều gì sẽ xảy ra như trích một phần mà mình thích về  Angular rồi xây dựng thêm một cái gì đó nhẹ nhàng hơn mà không cần phải chú ý đến những khái niệm bổ sung”.

Đây là phát biểu được  xem là bước đệm giúp cho Vue đã và đang có mặt tại thời điểm hiện tại. Vue đã được phát hành vào đầu năm 2014 và dự án này đã được tăng tải lên HackerNew, Echo Js ngay trong ngày đầu tiên ra mắt.



**Hình 1- 5**: Vuejs sở hữu nhiều ưu điểm nổi bật

#### 1.2.1.2 Lý do nên sử dụng Vuejs

Vue.js là một khung JavaScript tiến bộ được sử dụng để xây dựng giao diện người dùng của các ứng dụng hiện đại mà không cần sử dụng nhiều tài nguyên. Nó là sự kết hợp những gì tốt nhất của React và Angular.

Dưới đây là một số lý do nên sử dụng Vuejs.

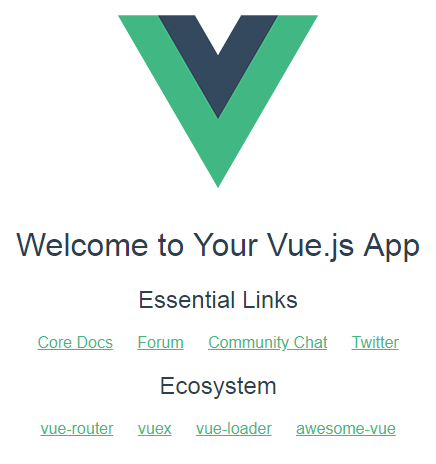
* Không cần bước “xây dựng”: Nếu sử dụng Vue thì bạn sẽ không cần phải trải qua quá nhiều bước để build (xây dựng) mà có thể đi thẳng vào vấn đề một cách dễ dàng. Bởi vì, Vue không cần sử dụng build tool (công cụ xây dựng) quá phức tạp mới có thể xây dựng ứng dụng, bạn chỉ cần khai báo một script là có thể phát triển một ứng dụng bằng Vue.
* Vue có thể tạo cấu trúc project nhanh chóng hơn nhờ vào CLI - command line interface (giao diện dòng lệnh).
* Hiện nay, tài liệu  về Vue ngày càng đa dạng, rõ ràng về ngôn ngữ nên bạn có thể dễ dàng trở thành chuyên gia về nó.
* Vue sở hữu một hệ sinh thái vững chắc nên có thể cung cấp một số add-ons rất hữu ích cho việc xây dựng một ứng dụng fontend điển hình nhất. Nó có thể bao gồm: vue-router, vuex, vue-test-utils, vue-dev-tools, vue-cli,…
* Core Vue sở hữu tính năng tối thiểu bởi khả năng tập trung vào việc render giao diện cho người dùng và các tương tác của nó. Chính vì vậy, nó sẽ cung cấp tối thiểu những tính năng cần thiết cho việc thiết kế và xây dựng kiến trúc. Nó sẽ giúp bạn loại bỏ được các tính năng không cần thiết ra khỏi thư viện trong lõi Vue.js và đảm bảo cho framework nhỏ gọn và mềm dẻo hơn.

#### 1.2.1.3 Cài đặt Vue

Dưới đây là một số cách cài đặt Vuejs mà bạn có thể tham khảo:

**Cách 1**: NPM – Node Package Manager (chương trình quản lý thư viện).

Nếu như muốn xây dựng một ứng dụng lớn với Vue thì lựa chọn NPM là vô cùng hợp lý. Có thể sử dụng câu lệnh “npm install vue”  để cài đặt Vue



**Hình 1- 6**: Cài đặt Vue

**Cách 2**: CLI

Hiện nay, Vue.js đang cung cấp một CLI có khả năng khởi tạo nhanh chóng các ứng dụng cho một trang. Bên trong CLI đã có chuyển bị sẵn những cài đặt phong phú nhằm đảm bảo cho quy trình tạo giao diện được hiện đại nhất.

Và chỉ mất vài phút thì bạn đã có thể bắt đầu cho việc xây dựng ứng dụng với những tính năng như: tùy chỉnh code khi save (lưu) và một số bản build (xây dựng) đã sẵn sàng để thực hiện deploy (triển khai) lên production (sản phẩm).

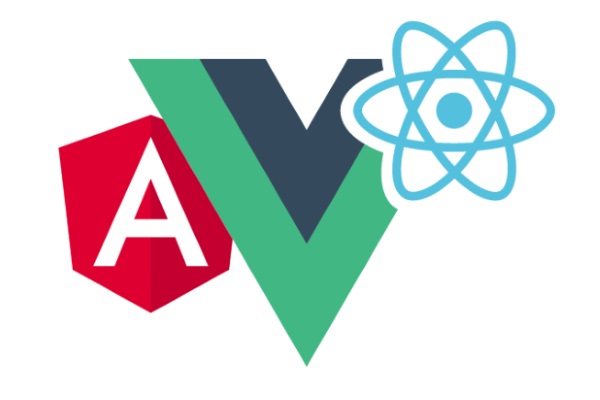
Bạn chỉ cần thực hiện như sau:

+ Tiến hành cài đặt vue-cli: npm install –global vue-cli

+ Tạo mới một dự án với template webpack (quản lý các module JavaScript.): vue init webpack my-project

+  Tiếp tục cài đặt thêm một số thư viện liên quan rồi tiến hành chạy thử.

#### 1.2.1.4 Ưu- nhược điểm về Vue.js



**Hình 1- 7**: Vue, React và Angular

*React và Vue*

Sự lựa chọn giữa React và Vue thường được tranh luận và nó không phải là một lựa chọn dễ dàng. Vue có một cộng đồng sôi động và ngày càng phát triển và đã chiếm được sự phổ biến so với React ở nhiều khía cạnh. Các nhà phát triển React vẫn đang tung ra rất nhiều thành phần và tính năng bổ sung mới, vì vậy không có dấu hiệu nào cho thấy React đang suy giảm.

Vue thường phù hợp hơn với các ứng dụng nhỏ hơn, ít phức tạp hơn và dễ học từ đầu hơn so với React. Vue có thể dễ dàng tích hợp hơn vào các dự án mới hoặc hiện tại và nhiều người cảm thấy việc sử dụng các mẫu HTML cùng với JSX là một lợi thế.

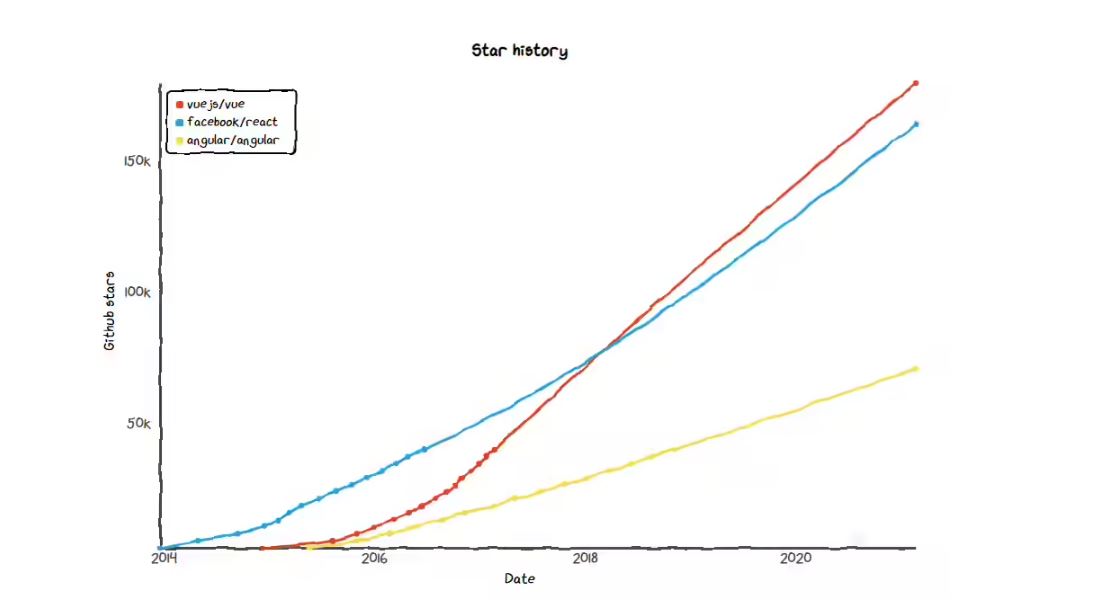
Nhìn chung, Vue có thể là lựa chọn tốt nhất nếu là nhà phát triển mới và không quen thuộc với các khái niệm JavaScript nâng cao, trong khi React khá phù hợp với các lập trình viên và nhà phát triển có kinh nghiệm đã làm việc với JavaScript hướng đối tượng, JavaScript chức năng và các khái niệm tương tự .

*Angular và Vue*

Angular và Vue là những thư viện rất khác nhau với các bộ tính năng và “đường cong học tập rất khác nhau”. Vue là lựa chọn rõ ràng cho các nhà phát triển ít kinh nghiệm và Angular sẽ được ưu tiên cho những người làm việc trên các ứng dụng lớn hơn.

Một thư viện lớn như Angular sẽ đòi hỏi sự chăm chỉ hơn trong việc cập nhật những gì mới, trong khi Vue sẽ ít đòi hỏi hơn về vấn đề này và thực tế là hai bản phát hành chính gần đây nhất của Vue nằm trong các kho lưu trữ riêng biệt.

Cũng cần lưu ý rằng Vue được tạo ra bởi một nhà phát triển trước đây đã làm việc trên Angular cho Google.

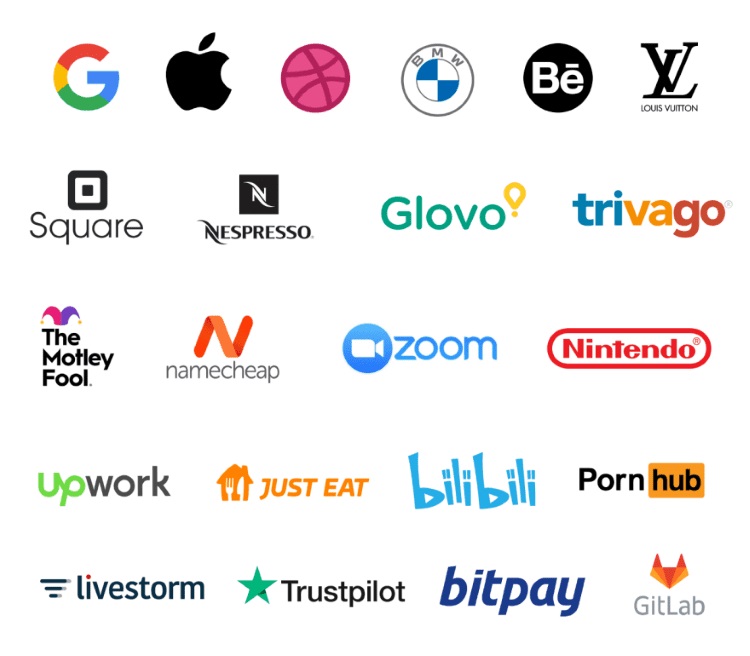


**Hình 1- 8**: Số sao trên các dự án GitHub cho Angular, React và Vue

**Bảng 1- 1**: So sánh giữa Angular, React và Vue[6]

|  |  |  |  |
| --- | --- | --- | --- |
|  | Angular | React | Vue |
| Phát hành lần đầu | 2010 | 2013 | 2014 |
| Trang web chính thức | [angle.io](https://angular.io/) | [reactjs.org](https://reactjs.org/) | [vuejs.org](https://vuejs.org/) |
| Phiên bản hiện tại | 13.x | 17.x | 3.x |
| Được sử dụng bởi | Google, Wix | Facebook, Uber | Alibaba, GitLab |
| Lượt theo dõi | 3.1k | 6.7k | 6.3k |
| Số sao | 78.4k | 180k | 218k |
| Chỉnh sửa | 20.6k | 36.5k | 35.7k |
| Người đóng góp | 1,500+ | 1,500+ | 400+ |

**Những công ty nào sử dụng Vue.js**

******

**Hình 1- 9**: Các công ty sử dụng Vue.js

#### 1.2.1.5 Một số thành phần cơ bản của Vuejs

* Component

Component(thành phần)[10] cho phép người lập trình chia giao diện người dùng thành các phần độc lập và có thể tái sử dụng khi cần thiết.

Vì là những đối tượng Vue tái sử dụng được, các component cùng nhận các tùy chọn như “new Vue”, ví dụ data, computed, watch, methods, và các hook vòng đời[11].

* Props

Props[12] trong Vuejs là một attribute (thuộc tính) tùy chỉnh dùng để truyền data(dữ liệu) từ component parent (thành phần cha) đến các component child (thành phần con). Để có thể nhận được dữ liệu từ các component parent thì component child phải khai báo prop mà nó mong muốn nhận được từ component parent, bằng cách đặt tên prop.

*Truyền dữ liệu bằng Props*[13]

Trong Vuejs tất cả các component đều có isolated scope (phạm vi cô lập) riêng. Điều này có nghĩa là không thể tham chiếu trực tiếp các dữ liệu ở component cha (parent) vào trong component con(child). Mà thay vào đó sẽ phải truyền dữ liệu qua *props*.

* Routing[14]

Cho phép người dùng dễ dàng điều hướng giữa các trang mà không cần phải tải lại toàn bộ trang. Sử dụng router để có thể điều hướng giữa các component.

* Event[15]

Dùng để xử lý các sự kiện trong Vuejs. Ví dụ muốn xử lý sự kiện click vào button element, dùng v-on để “lắng nghe” các sự kiện DOM-Document Object Model(mô hình đối tượng tài liệu)và thực thi JavaScript khi những sự kiện này được kích hoạt:

<button v-on:click=”tên sự kiện”></button> hoặc

<button @click=”tên sự kiện”></button>

Trong một số trường hợp, người lập trình cần gọi event.preventDefault() hoặc event.stopPropagation() bên trong một phương thức xử lý sự kiện nhầm giúp phương thức tập trung giải quyết logic dữ liệu.

Để giải quyết vấn đề này, Vue cung cấp các **event modifier** cho v-on. Event modfier là một hậu tố (postfix) cho directive(chỉ thị), được biểu thị bằng một dấu chấm.

* .stop
* .prevent
* .capture
* .self
* .once

Ví dụ:

<!-- sự kiện click sẽ không được propagate (lan truyền)  
 điều này tương đương với event.stopPropagation() -->  
<a v-on:click.stop="doThis"></a>

<!-- sự kiện submit sẽ không reload trang (tải lại trang)  
điều này tương đương với event.preventDefault() -->  
<form v-on:submit.prevent="onSubmit"></form>

<!-- ta có thể nối modifier với nhau -->  
<a v-on:click.stop.prevent="doThat"></a>  
  
<!-- chỉ có modifier, không có phương thức xử lí -->  
<form v-on:submit.prevent></form>

### 1.2.2 NodeJS Framework

#### 1.2.2.1 NodeJS là gì

**Định nghĩa NodeJS**

NodeJS là mã nguồn mở chạy trên môi trường V8 JavaScript runtime (một trình thông dịch [JavaScript](https://wiki.tino.org/html-css-javascript-la-gi/)chạy cực nhanh trên trình duyệt Chrome). NodeJS giúp các nhà phát triển xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

NodeJS có thể được dùng để xây dựng các loại ứng dụng khác nhau như các ứng dụng dòng lệnh, ứng dụng web, ứng dụng trò chuyện theo thời gian thực, máy chủ [REST API](https://wiki.tino.org/restful-api-la-gi/),.. Tuy nhiên, NodeJS thường được dùng chủ yếu để xây dựng các chương trình mạng như máy chủ web, tương tự như PHP, Java hoặc ASP.NET.

Vì đây là mã nguồn mở nên cho phép bạn sử dụng miễn phí và đồng thời được liên tục được chỉnh sửa, cải tiến bởi cộng đồng các nhà phát triển toàn cầu.



**Hình 1- 10**: Giới thiệu NodeJS

***Lịch sử ra đời của NodeJS* [7]**

NodeJS xuất hiện lần đầu vào năm 2009 bởi Ryan Dahl. Phiên bản đầu tiên của NodeJS chỉ hỗ trợ hệ điều hành Linux và MacOS X. Ban đầu, việc phát triển và bảo trì được thực hiện bởi Dahl. Sau đó được Joyent, một công ty phần mềm và dịch vụ tài trợ.

Vào tháng 1 năm 2010, một trình quản lý của NodeJS đã được giới thiệu để giúp các lập trình viên dễ dàng xuất bản và chia sẻ mã nguồn của các gói NodeJS, đồng thời đơn giản hóa việc cài đặt, gỡ cài đặt và cập nhật các phiên bản.

Vào năm 2011, Microsoft và Joyent đã cùng nhau phát triển phiên bản [Windows](https://wiki.tino.org/he-dieu-hanh-windows-la-gi/) của NodeJS, mở rộng số lượng hệ điều hành mà công cụ này có thể hỗ trợ và cung cấp thêm nhiều lựa chọn cho các nhà phát triển

Nền tảng NodeJS đã được hợp nhất với JS Foundation vào năm 2019 để tạo thành OpenJS Foundation gồm những nền tảng giúp quản lý dự án phát triển mã nguồn mở, phân tán của NodeJS.

#### **1.2.2.2 Đặc điểm và ứng dụng của NodeJS**

**Đặc điểm của NodeJS**

* NodeJS không cần đợi API (ứng dụng) trả dữ liệu về, do đó mọi APIs nằm trong thư viện NodeJS đều không được đồng bộ.
* Đây là một Platform (nền tảng) chứ không phải là một Framework. Do đó, NodeJS cho phép bạn có thể xây dựng các website một cách độc lập và nhanh chóng hơn.
* NodeJS có thể chạy trên đa nền tảng gồm: Window, MacOS, Linux.
* NodeJS được xem là một máy chủ đơn luồng và không thể hỗ trợ đa luồng.
* NodeJS không được xem là một ngôn ngữ lập trình, nên những người mới phải nắm chắc kiến thức lập trình căn bản như: các giao thức, Javascript,… mới có thể sử dụng NodeJS. Tuy nhiên, cộng đồng NodeJS thường rất lớn, sẵn sàng support cho bạn mọi lúc mọi nơi.
* Phần core của NodeJS thường được biết bằng ngôn ngữ C++[8] nên nó hiệu năng và tốc độ xử lý tương đối cao. Nhờ vậy, hầu hết các ứng dụng NodeJS đều có khả năng đáp ứng được thời gian thực chạy trên đa nền tảng, đa thiết bị,…

**Những ứng dụng của NodeJS trong lập trình web**

* NodeJS có thể tạo, mở, đọc, ghi, xóa và đóng các tệp ngay khi đang truy cập trên máy chủ.
* Xây dựng nội dung cho các trang web động.
* Thực hiện thu thập dữ liệu theo yêu cầu cụ thể.
* Thực hiện truy vấn, sửa, xóa, thêm các dữ liệu trong các hệ quản trị cơ sở như: Microsoft SQL Server, [MySQL](https://wiki.tino.org/mysql-la-gi/), [MongoDB](https://wiki.tino.org/mongodb-la-gi/), PostgreSQL.

#### 1.2.2.3 Điểm mạnh và hạn chế của của NodeJS

**Điểm mạnh**

* I/O (input – dữ liệu vào /output – dữ liệu ra) hướng sự kiện không đồng bộ giúp xử lý nhiều yêu cầu đồng thời.
* Đáp ứng được những yêu cầu về thời gian thực.
* Có tốc độ cực rất nhanh, đáp ứng được nhu cầu sử dụng của khách truy cập “khổng lồ” trong thời gian ngắn.
* Sử dụng JavaScript, một ngôn ngữ lập trình rất dễ học.
* Chia sẻ cùng một đoạn mã với cả phía máy chủ và máy khách.
* Npm và các module rất mạnh mẽ và vẫn đang tiếp tục phát triển.
* Có một cộng đồng lớn mạnh, có nhiều mã được chia sẻ qua github
* Tương thích với nhiều thiết bị, nhiều hệ điều hành như MacOS, Window, Linux,…

**Hạn chế**

* NodeJS không cung cấp khả năng mở rộng và không thể tận dụng lợi thế của nhiều lõi thường có trong phần cứng cấp máy chủ ngày nay.
* Thao tác với cơ sở dữ liệu quan hệ là một khó khăn nếu bạn đang sử dụng NodeJS
* Mỗi lần sử dụng lệnh gọi lại sẽ kết thúc với rất nhiều lệnh gọi lại lồng vào nhau.
* Nếu không hiểu rõ về JavaScript, bạn sẽ gặp khó khăn với NodeJS
* NodeJS không phù hợp với các tác vụ đòi hỏi nhiều CPU mà chỉ phù hợp với những I/O như máy chủ web.
* Nếu bạn có một web hosting (lưu trữ web) dùng chung, sẽ rất khó khăn nếu bạn tải lên một ứng dụng NodeJS. VPS và Dedicated server là một sự lựa chọn tốt hơn nhiều.

#### Hướng dẫn cách cài đặt và khai báo cho NodeJS

**Cài đặt NodeJS**

* **Bước 1:**Bạn truy cập vào địa chỉ “https://nodejs.org/en/download/” để cài đặt NodeJS.
* **Bước 2:** Tiến hành kiểm tra lại trạng thái cài đặt bằng cách: Vào **cmd** -> **node –v**
* **Bước 3:** Tạo một thư mục dùng để chứa các nội dung cho dự án. Trong đó, tạo 1 file có tên DuAn.js và nhập nội dung cần viết là:



**Hình 1- 11**: Cài đặt NodeJS

Để chương trình có thể chạy, bạn vào trong thư mục vừa tạo rồi tiến hành thao tác “Shift + chuột phải” vào thư mục đó và chọn Open PowerShell window here. Khi cửa sổ command line đã hiện lên, bạn hãy gõ lệnh: “node Duan.js”

Lúc này, cửa sổ command line sẽ hiện lên để bạn khai báo biến và hằng trong NodeJS

**Thực hiện khai báo biến và hằng trong NodeJS**

Khai báo biến: Trong NodeJS thường sử dụng 2 cách sau để khai báo biến.

* + Cách 1: Sử dụng từ khoá var

Var tên\_biến = giá trị;

* + Cách 2: Không sử dụng từ khoá var

Tên\_biến = giá trị;

Khai báo hằng: Việc thực hiện khai báo hằng sẽ tương tự như các ngôn ngữ khác và bạn có thể sử dụng từ khóa const để khai báo như sau:

* + const tên\_hằng = giá trị;
  + Cấu trúc rẽ nhánh trong NodeJS sẽ được viết giống như các dạng ngôn ngữ khác.

****

**Hình 1- 12**: Cấu trúc rẽ nhánh trong NodeJS

Đối với những cấu trúc vòng lặp dạng for, while, do… while sẽ có cấu trúc tương tự như các ngôn ngữ lập trình khác như: C/C++, PHP…

#### 1.2.2.5 NodeJS phù hợp cho các ứng dụng nào

Những loại ứng dụng dưới đây nên được viết bằng NodeJS:

* Fast File Upload: Những chương trình upload file với tốc độ cao.
* Websocket server: Các máy chủ của web socket gồm các dạng như: Game server, online chat,…
* Restful API: gồm các ứng dụng được dùng cho những ứng dụng khác thông qua API.
* Any Real-time Data Application: Những ứng dụng có yêu cầu cao về tốc độ thực hiện trong thời gian thực.
* Ad server: Các máy chủ quảng cáo.

Tại thời điểm này, chúng ta vẫn chưa thể cho rằng NodeJS đủ mạnh để thay thế hoàn toàn cho .NET, PHP hay Java. Nhưng trong tương lai gần, NodeJS hoàn toàn có thể trở thành một giải pháp tối ưu nhất cho web back-end.

#### 1.2.2.6 Các thư viện thường dùng trong NodeJS

* Express

Đây là một trong những module (hay framework) được sử dụng nhiều nhất, expressjs là bộ khung giúp bạn dựng các ứng dụng web trên Nodejs, bạn có thể dễ dàng custom router, controller, … với express.

* Làm ứng dụng MEAN stack (MongoDB, Express.js, Angular.js, Node.js)
* Có thể xử lý dễ dàng nhiều loại yêu cầu như GET, PUT, POST và DELETE
* Xây dựng các single page app, multi page app và hybrid web app.
* Async

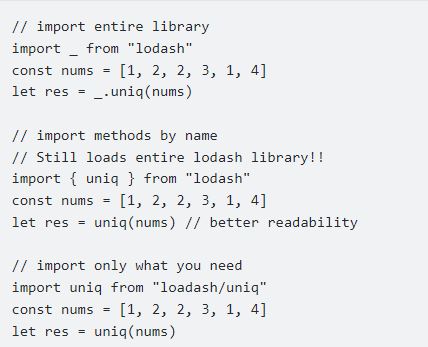
Workflow của nodejs chạy theo hướng bất đồng bộ và callback, đây là một trong những điểm mạnh của nodejs. Nhưng đây lại là nỗi khốn khổ của phần lớn begginner bởi không thể kiểm soát được các luồng của async. Async là module giúp bạn khử bất đồng bộ, chạy các hàm 1 cách tuần tự.



**Hình 1- 13**: Cách dùng Async

* Lodash

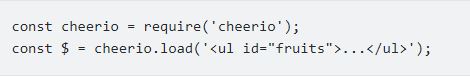
Lodash giúp bạn tăng sức mạnh cho ứng dụng javascript và nodejs với các hàm mở rộng để xử lý chuỗi, mảng, object,

…  


**Hình 1- 14**: Cách dùng lodash

* Cheerio

Cheerio phân tích cú pháp như HTML và cung cấp API để duyệt/thao tác cấu trúc dữ liệu kết quả



**Hình 1- 15** : Cách dùng Cheerio

* Nodemailer

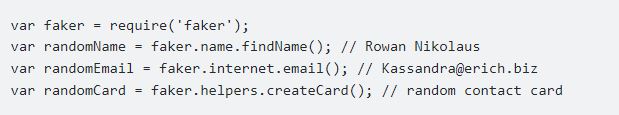
Gửi các email từ Node.JS, dễ dàng gửi các email - sử dụng SMTP



**Hình 1- 16**: Cách dùng Nodemailer

* Faker

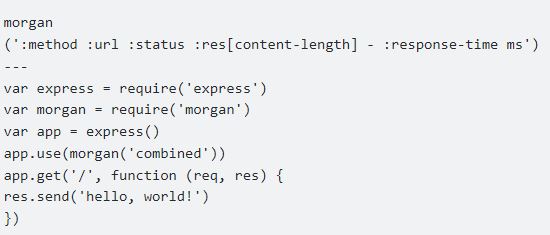
Tạo một lượng lớn dữ liệu giả trong trình duyệt và node.js. Xây dựng Front End UI và tương tác với dữ liệu trong khi quá trình xây dựng API chưa hoàn tất. Nhiều phương thức API bao gồm địa chỉ, công ty, cơ sở dữ liệu, hình ảnh, tên (FirstName, lastName)



**Hình 1- 17**: Cách dùng Faker

* Morgan

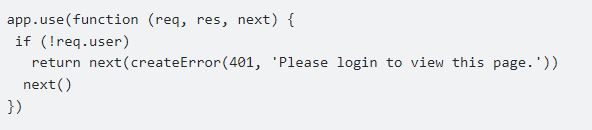
Phần mềm trung gian logger yêu cầu HTTP cho node.js



**Hình 1- 18**: Cách dùng Morgan

* Http-Errors

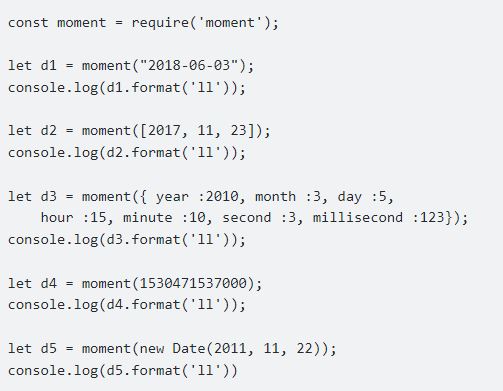
Tạo erros HTTP cho Express, Koa, Connect, v.v. Dễ dàng gửi phản hồi lỗi, nhiều thuộc tính lỗi có sẵn



**Hình 1- 19**: Cách dùng Http-Errors

* Moment

Cung cấp các hàm helper để xử lý, tính toán, xác thực, hiển thị,… thời gian trong javascript.



**Hình 1- 20**: Cách dùng Moment

* Mongoose

Mongoose có sẵn các hàm giúp tương tác đến CSDL MongoDB một cách dễ dàng.

* Mysql

mysql.js giúp bạn dễ dàng kết nối đến cơ sở dữ liệu mysql bằng Nodejs.

### 1.2.3 MySQL

#### 1.2.3.1 Giới thiệu MySQL là gì?



**Hình 1- 21**: Logo MySQL

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến hàng đầu trên thế giới và đặc biệt được ưa chuộng trong quá trình xây dựng, phát triển ứng dụng. Đây là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có khả năng thay đổi mô hình sử dụng phù hợp với điều kiện công việc.

MySQL được phát hành đâu tiền vào 23 tháng 5 năm 1995 và phiên bản ổn định 8.0.25, mã nguồn được viết bằng c, c++ và hệ điều hành có thể sử dụng là linux, Solaris, MacOS, Windows, FreeBSD. MySQL được sở hữu và tài trợ bởi công ty Thủy Điển MySQL AB, được mua lại bởi Sun Microsystems (nay là tập đoàn Oracle). Trong năm 2010, khi Oracle mua lại Sun, Widenius chi hai mã nguồn mở dựa Mysql để tạo MariaDB.

MySQL cũng có thể chạy trên các nền tảng điện toán đám mây như Microsoft Azure, Amazon EC2, Oracle Cloud Infrastructure

#### 1.2.3.2 Tính năng

* Triggers
* Cursors
* Cross-platform support
* Hỗ trợ SSL
* Truy vấn bộ nhớ đệm (Query caching)
* Hỗ trợ Unicode support
* Lập chỉ mục toàn văn bản và tìm kiếm (Full-text indexing and searching)

#### 1.2.3.3 Ưu - nhược điểm

*Ưu điểm:*

* Sử dụng dễ dàng: dễ sử dụng và hoạt động trên nhiều hệ điều hành
* Tính bảo mật cao: sở hữu nhiều tính năng bảo mật, thậm chí là bảo mật cấp cao
* Đa tính năng: hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp
* Khả năng mở rộng và mạnh mẽ: khả năng xử lý khối dữ liệu lớn và có thể mở rộng khi cần thiết.
* Tương thích trên nhiều hệ điều hành: tương thích để chạy trên nhiều hệ điều hành, như Novell NetWare, Windows \* Linux \*, nhiều loại UNIX \* (như Sun \* Solaris \*, AIX và DEC \* UNIX), OS / 2, FreeBSD \*,....
* Cho phép khôi phục: cho phép các transactionđược khôi phục, cam kết và phục hồi sự cố.

*Nhược điểm:*

* Bị hạn chế dung lượng

### 1.2.4 JSON Web Token (JWT)

#### 1.2.4.1 JWT – JSON Web Token là gì ?

**JSON Web Token** là một chuỗi mã hóa mà nguồn gốc ban đầu là một chuỗi **JSON**. Chuỗi thông tin dạng JSON bằng phương pháp mã hóa nào đó, nó trở thành 1 chuỗi ký tự lộn xộn nhìn vào sẽ rất khó hiểu.

Bảo mật **JWT** là phuơng pháp xác thực quyền truy cập (**Authentication**) bằng JSON Web Token.

#### 1.2.4.2 Cấu trúc của một JWT

Dưới đây là 1 JSON Web Token



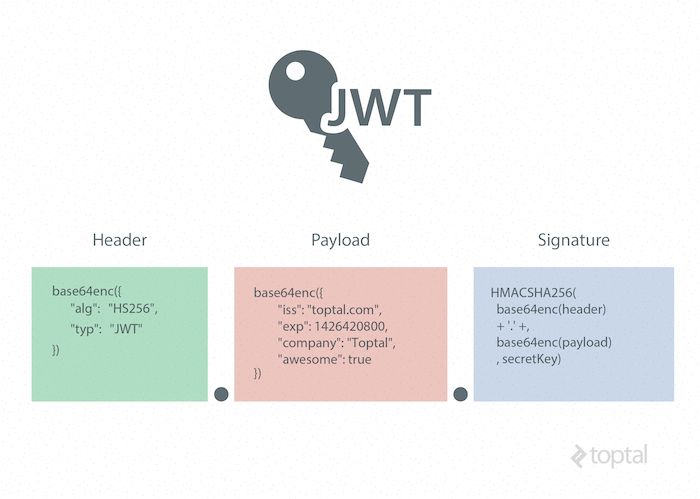
**Hình 1- 22**: JWT

JWT trên bao gồm 3 phần:

* Header
* Payload
* Signature

Cấu trúc của nó theo định dạng như sau

<base64-encoded header>.<base64-encoded payload>.<HMACSHA256(base64-encoded signature)>



**Hình 1- 23**: Cấu trúc của JWT

* Header

Header bao gồm hai phần chính:

* typ – Loại token (mặc định là JWT – cho biết đây là một Token JWT)
* alg – Thuật toán đã dùng để mã hóa (HMAC SHA256 – HS256 hoặc RSA).
* **Payload**

Là nơi chứa các nội dung của thông tin (claim). Thông tin truyền đi có thể là mô tả của 1 thực thể (ví dụ như người dùng) hoặc cũng có thể là các thông tin bổ sung thêm cho phần Header. Chúng được chia làm 3 loại: **reserved**, **public** và **private**

**Reserved**: là những thông tin đã được quy định ở trong IANA JSON Web Token Claims registry. Những thông tin này không có cái nào là bắt buộc cả. Tuy nhiên tùy vào từng ưng dụng bạn implement mà hãy ràng buộc yêu cầu bắt buộc đối với những thông tin cần thiết

* **iss** (issuer): tổ chức phát hành token (không bắt buộc)
* **sub** (subject): chủ đề của token (không bắt buộc)
* **aud** (audience): đối tượng sử dụng token (không bắt buộc)
* **exp** (expired time): thời điểm token sẽ hết hạn (không bắt buộc)
* **nbf** (not before time): token sẽ chưa hợp lệ trước thời điểm này
* **iat** (issued at): thời điểm token được phát hành, tính theo UNIX time
* **jti**: JWT ID

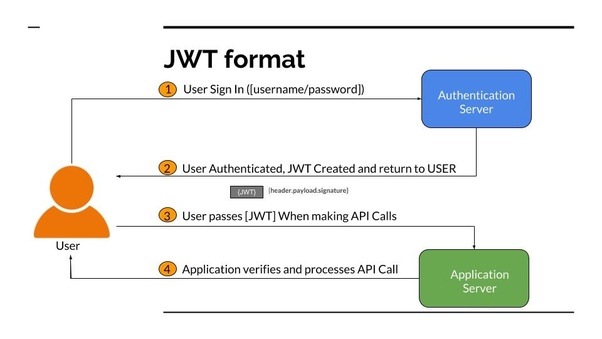
**Public**: Khóa có thể define tùy theo ý muốn của người sử dụng JWT. Tuy nhiên để tránh trùng lặp, khó nên được quy định ở trong IANA JSON Web Token Registry hoặc là 1 URI có chứa không gian tên không bị trùng lặp.

**Private**: Phần thông tin thêm dùng để truyền qua giữa các client

* **Signature**

Phần chữ ký được tạo bằng cách kết hợp 2 phần Header + Payload, rồi mã hóa nó lại bằng 1 giải thuật encode bất kỳ ví dụ như HMAC SHA-256

#### **1.2.4.3 Sơ đồ hệ thống sử dụng JWT**



**Hình 1- 24**: Sơ đồ hoạt động JWT

**Nhìn vào hình ta có thể thấy flow đi như sau:**

1. User thực hiện đăng nhập bằng cách gửi username/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (máy chủ xác thực)
2. Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
3. Người dùng nhận được JWT do Authentication Server vừa mới trả về làm “token” để thực hiện các “lệnh” tiếp theo đối với Application Server.
4. Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify(xác thực) JWT gửi lên. Nếu OK, tiếp tục thực hiện yêu cầu được gọi.

# **Chương 2. MÔ TẢ NGHIỆP VỤ**

## **2.1 TỔNG QUAN**

Ngày này trong nhịp sống hối hả của con người thì việc giành thời gian để ra ngoài để mua sắm trở nên là 1 điều quá xa sỉ.. Những lo lắng về giao thông không an toàn và hạn chế trong việc mua hàng truyền thống có thể tránh được trong khi mua sắm trực tuyến. Với mua sắm trực tuyến bạn cũng không cần phải lo lắng về điều kiện thời tiết. Người tiêu dùng và các khách hàng là những tổ chức, công ty,… đang dần chuyển sang mua sắm trực tuyến nhiều hơn nhằm tiết kiệm thời gian

Chính vì thế việc mua sắm online càng trở nên quan trọng và cần thiết,chỉ cần một cú click chuột thì họ có thể có được sản phẩm mà mình mong muốn.Việc mua sắm online có nhiều ưu điểm là có thể sở hữu mọi thứ thông qua các cú click chuột chứ không cần phải đến tận nơi để mua hàng. Sau khi vào [website bán hàn](http://thietkewebsitehcm.org/thiet-ke-website-ban-hang)g, chọn sản phẩm, chỉ cần đặt hàng (order) người bán sẽ mang sản phẩm đến tận nhà bạn. Mua sắm online cho phép mua hàng bất cứ khi nào bạn muốn. Các cửa hàng trên mạng không bao giờ đóng cửa, có thể mua sắm 24/24 giờ và 7 ngày trong tuần. Mua sắm ở các chợ, trung tâm thương mại hay cửa hàng rất khó để bạn có thể so sánh đặc điểm và giá của sản phẩm với nhau. Khi mua hàng online, bạn dễ dàng so sánh và đưa ra lựa chọn sản phẩm phù hợp nhất.

Nếu không có website, các đơn vị kinh doanh vẫn có thể hoạt động, tuy nhiên sẽ bỏ lỡ rất nhiều cơ hội từ những khách hàng tiềm năng và gặp khó khăn trong việc mở rộng, tiếp cận thị trường mới.

Trang web được xây dựng chuyên nghiệp có thể gây ấn tượng tốt trong tâm trí của khách hàng, thu hút tiềm năng rất lớn của cửa hàng. Có thể khách hàng sẽ trở lại trang web và mua hàng khi họ có yêu cầu. Web bán hàng sẽ phục vụ 2 đối tượng chính: quản trị viên, khách hàng.

* Quản trị viên: là người quản trị hệ thống, có các chức năng như: đăng nhập, quản lý sản phẩm, quản lý danh mục, quản lý khuyến mãi, quản lý banner, quản lý tin tức, quản lý nhân viên, quản lý tin tức, quản lý đơn hàng.
* Khách Hàng:
* Khách hàng mới: là khách hàng vãng lai chưa có tài khoản được sử dụng các chức năng sau: đăng ký, đăng nhập, xem sản phẩm, tìm kiếm sản phẩm.
* Khách hàng thành viên: là khách hàng đã đăng ký thành viên chỉ cần đăng nhập để quản lý đơn hàng, quản lý giỏ hàng, góp ý, bình luận.

### 2.1.1 Quản trị viên

#### 2.1.1.1 Quản lý sản phẩm

* Quản lý sản phẩm sẽ xem danh sách sản phẩm bao gồm thông tin chi tiết sản phẩm như tên sản phẩm, đường dẫn, danh mục, ngày tạo, người tạo, người cập nhập, ngày cập nhập.
* Khi quản trị muốn thêm sản phẩm phải chọn mục thêm sản phẩm sau đó người quản trị phải nhập các thông tin bao gồm tên sản phẩm, lựa chọn danh mục, số tiền mua sản phẩm, màu sắc sản phẩm và có những kích thước theo màu sắc sắc phẩm, hình ảnh sản phẩm và của màu sắc sản phẩm, trạng thái sản phẩm và mô tả thông tin của sản phẩm.
* Khi người quản trị chọn nút sửa trên một sản phẩm, hệ thống sẽ lấy thông tin sản phẩm và người quản trị sẽ tiến hành chỉnh sửa trên thông tin sản phẩm đó.

#### 2.1.1.2 Tạo tài khoản và phân quyền

* Khi tạo tài khoản cho trang quản trị, người tạo tài khoản sẽ phải điền các thông tin: họ tên, email, mật khẩu, số điện thoại, địa chỉ,giới tính.
* Nếu tạo tài khoản trên giao diện khách hàng thì tài khoản đó chỉ có các chức năng của người mua hàng như : đăng nhập, thông tin, xem chi tiết sản phẩm, mua hàng, thanh toán, giỏ hàng
* Nếu tạo tài khoản trên giao diện của Quản trị viên thì tài khoản đó có phân quyền là nhân viên
* Các trường thông tin nhập không được để trống, mật khẩu ít nhất 8 ký tự bao gôm chữ hoa, chữ thường, kí tự đặc biệt

#### 2.1.1.3 Đăng nhập

* Khi đăng nhập, người dùng sẽ dùng email và mật khẩu đã tạo. Hệ thống sẽ kiểm tra thông tin người dùng nhập vào nếu đúng sẽ chuyển đến trang chủ (là trang bán hàng hoặc là trang quản trị), nếu sai sẽ thông báo đến người dùng và yêu cầu đăng nhập lại.

#### 2.1.1.4 Quản lý khách hàng

* Quản lý khách hàng cho phép người quản trị xem thông tin tài khoản của khách hàng đăng nhập trên trang web bao gồm email, họ tên, địa chỉ, số điện thoại.
* Người quản trị còn có thể tiến hành vô hiệu hóa tài khoản người dùng và có thể tái kích hoạt.

#### 2.1.1.5 Quản lý nhân viên

* Quản lý nhân viên cho phép người quả trị xem thông tin tài khoản của nhân viên bao gồm: email, họ tên, số điện thoại, giới tính, địa chỉ
* Người quản trị có thể sửa thông tin của nhân viên khi cần thiết..
* Ngoài ra, người quản trị có thể vô hiệu hóa tài khoản nhân viên và có thể tái kích hoạt.

#### 2.1.1.6 Quản lý sản phẩm

* Quản lý sản phẩm bao gồm các nhiệm vụ thêm, xóa, sửa sản phẩm trên hệ thống website. Người quản lý có thể xem danh sách sản phẩm.
* Khi thêm sản phẩm người quản trị phải nhập các thông tin bao gồm: Tên sản phẩm, thương hiệu, giá, hình ảnh, các thông tin chi tiết CPU, RAM, ổ cứng, hệ điều hành thì chỉ cần chọn chúng trên bảng select
* Khi xóa sản phẩm thì sản phẩm đó chuyển sang trạng thái khác, chứ không biến mất hoàn toàn, để sau này khi cần có thể kích hoạt lại.
* Khi chỉnh sửa sản phẩm, hệ thống sẽ tiến hành lấy thông tin sản phẩm tiến hành thao tác chỉnh sửa dựa trên các thông tin đó.

#### 2.1.1.7 Quản lý linh kiện

* Quản lý linh kiện bao gồm các nhiệm vụ xem và thêm linh kiện trên hệ thống website.
* Khi thêm sản phẩm người quản trị phải nhập trực tiếp tên linh kiện cần thêm trong trang linh kiện
* Các thông tin của linh kiên sẽ được hiển thị qua trang tạo sản phẩm, người quản trị chỉ cần chọn đúng linh kiện của sản phẩm mà không cần phải nhập.

#### 2.1.1.8 Quản lý đơn hàng

* Quản lý đơn hàng ngoài xem danh sách đơn hàng, người quản trị còn có thể tiến hành các thao tác hủy đơn hàng, cập nhật đơn hàng
* Danh sách đơn hàng được chia làm hai danh sách: Danh sách đơn hàng chưa xử lý, và danh sách đơn hàng đã hoàn thành.
* Đối với danh sách đơn hàng chưa hoàn thành người quản trị có thể tiến hành chỉnh sửa trạng thái đơn hàng theo các giai đoạn: chưa xử lý, đã xác nhận, đang giao hàng, hoàn thành. Khi đơn hàng ở trạng thái hoàn thành sẽ tự động được chuyển đến danh sách đơn hàng đã hoàn thành.

### 2.1.2 Người mua hàng

Người mua hàng là đối tượng có nhu cầu sử dụng trang web để xem và mua sản phẩm.

Khách hàng được sử dụng các chức năng sau: xem sản phẩm, tìm kiếm sản phẩm, đăng ký, đăng nhập, quản lý thông tin cá nhân, quản lý đơn hàng, quản lý giỏ hàng.

#### 2.1.2.1 Đăng ký

* Khi khách hàng đăng kí trên website, hệ thống hiển thị ra form đăng kí và yêu cầu khách hàng nhập vào các thông tin: Họ tên, email, số điện thoại và mật khẩu, địa chỉ. Hệ thống sẽ tiến hành kiểm tra các thông tin như email và mật khẩu có hợp lệ không. Ví dụ: Email phải đúng định dạng email@gmail.com, mật khẩu phải 8 kí tự trở lên và có chữ hoa, chữ thường, kí tự đặc biệt.
* Khi đăng ký thành công thì khách hàng sẽ được chuyển qua trang đăng nhập vào hệ thống

#### 2.1.2.2 Đăng nhập

* Khi đăng nhập, người dùng sẽ dùng email và mật khẩu đã đăng kí. Hệ thống sẽ kiểm tra thông tin người dùng nhập vào nếu đúng sẽ chuyển người dùng đến trang chủ, nếu sai sẽ thông báo đến người dùng và yêu cầu đăng nhập lại.

#### 2.1.2.3 Xem sản phẩm

* Nhu cầu xem sản phẩm là điều không thể thiếu của khách hàng. Chức năng xem sản phẩm cho phép khách hàng xem tất cả sản phẩm có trong hệ thống khi truy cập vào trang web.
* Website có thể hiển thị sản phẩm theo danh mục hàng hóa giúp khách hàng dễ dàng hơn trong quá trình xem sản phẩm.
* Xem theo chủ đề như: sản phẩm mới, sản phẩm khuyến mãi hay sản phẩm bán chạy.

#### 2.1.2.4 Xem thông tin sản phẩm

* Khi chọn một sản phẩm để xem thông tin, hệ thống sẽ hiển thị các thông về sản phẩm đó bao gồm: Tên sản phẩm, giá, hình ảnh, thông tin chi tiết ở các góc độ khác nhau..
* Ngoài ra khi xem thông tin sản phẩm website sẽ hiển thị các sản phẩm tương tự theo danh mục đó.

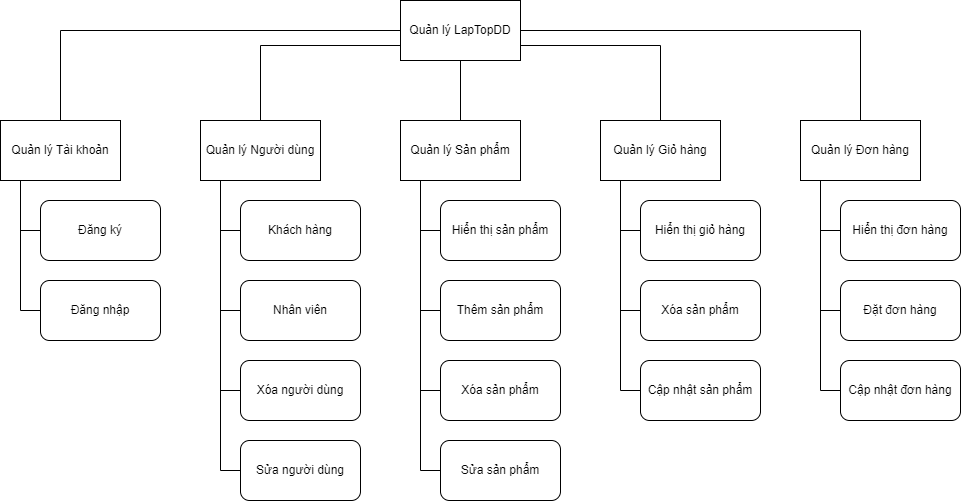
#### 2.1.2.5 Quản lý giỏ hàng

* Khi người dùng tìm được sản phẩm đúng với nhu cầu, muốn thêm sản phẩm vào giỏ hàng chỉ cần bấm vào thêm giỏ hàng. Sản phẩm sẽ được cập nhật vào giỏ hàng trên hệ thống
* Người dùng có thể xem thông tin giỏ hàng bao gồm: thông tin sản phẩm đã thêm vào giỏ hàng, số lượng và giá tiền từng loại sản phẩm cũng như tống giá tiền tất cả sản phẩm đã mua
* Giỏ hàng được mô phỏng như giỏ hàng trong thực tế, người dùng có thể thêm xóa sửa, cập nhật số lượng từng sản phẩm một cách dễ dàng.

#### 2.1.2.6 Thanh toán

* Khi người dùng đã chọn được các sản phẩm theo yêu cầu, muốn tiến hành thanh toán người dùng chỉ cần bấm chọn vào ô thanh toán trên giỏ hàng.

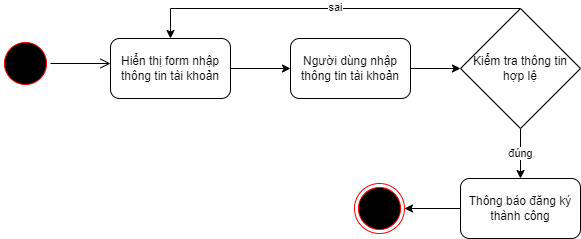
## **2.2 SƠ ĐỒ CHỨC NĂNG**



**Hình 2- 1**: Sơ đồ chức năng tổng quát

## **2.3 QUY TRÌNH XỬ LÝ**

### 2.3.1 Quy trình đăng ký

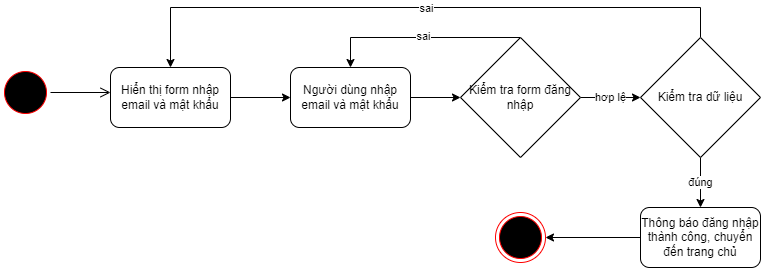


**Hình 2- 2**: Quy trình đăng ký

Yêu cầu:

* Ghi nhận đầy đủ thông tin nhập vào (tên, email, mật khẩu, số điện thoại, địa chỉ..)
* Email phải đúng đinh dạng [name@gmail.com](mailto:name@gmail.com),
* Mật khẩu ít nhất 8 kí tự bao gồm chữ hoa, chữ thường, kí tự đặc biệt.

### 2.3.2 Quy trình đăng nhập



**Hình 2- 3**: Quy trình đăng nhập

Yêu cầu:

* Ghi nhận đầy đủ thông tin nhập vào (email, mật khẩu)
* Email phải đúng đinh dạng [name@gmail.com](mailto:name@gmail.com)
* Mật khẩu nhập vào đủ yêu cầu
* Kiểm tra thông tin đúng email và mật khẩu

### 2.3.3 Quy trình đặt hàng



**Hình 2- 4**: Quy trình đặt hàng

Yêu cầu:

* Giỏ hàng hiện đúng sản phẩm khách hàng đặt
* Hiển thị thông tin cơ bản sản phẩm (tên, hình…)
* Hiển thi đúng số tiền

### 2.3.4 Quy trình thanh toán



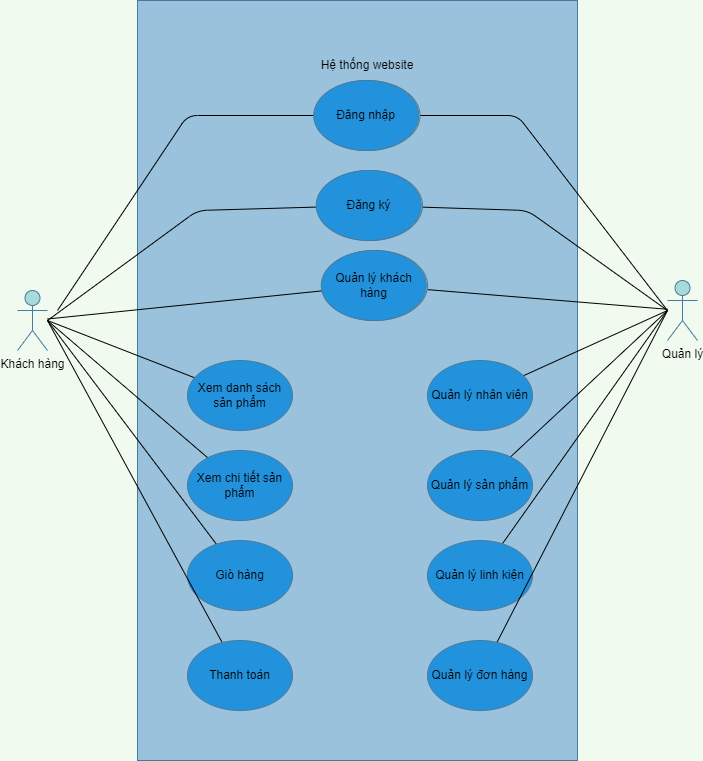
**Hình 2- 5**: Quy trình thanh toán

Yêu cầu:

* Nhập thông tin người mua nếu chưa đăng nhập
* Nếu đã đăng nhập lấy thông tin để làm thông tin người mua
* Hiển thị thông tin cơ bản sản phẩm (tên, hình…)
* Hiển thi đúng số tiền

# **Chương 3. KIẾN TRÚC HỆ THỐNG**

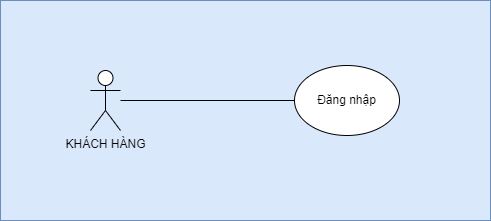
## **3.1 USE-CASE TỔNG QUÁT**



**Hình 3- 1**: Sơ đồ Use-Case tổng quát

## **3.2 SƠ ĐỒ CHI TIẾT**

### 3.2.1 Use-case Đăng nhập (khách hàng)

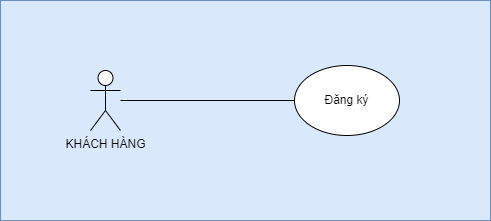


**Hình 3- 2**: Sơ đồ Use-case đăng nhập (khách hàng)

**Bảng 3- 1**: Mô tả Use-case đăng nhập (khách hàng)

|  |  |
| --- | --- |
| **Tên Use case** | **Đăng nhập (khách hàng)** |
| **Actor** | **Khách hàng** |
| **Mô tả** | **Use case cho phép khách hàng đăng nhập vào hệ thống** |
| Điều kiện sau khi kết thúc | * Khách hàng chưa đăng nhập * Khách hàng đã đăng nhập vào hệ thống và có thể sử dụng các chức năng của hệ thống cung cấp |
| Luồng sự kiện chính | 1. Khách hàng chọn chức năng đăng nhập 2. Hệ thống hiển thị ra form đăng nhập 3. Khách hàng nhập thông tin email và mật khẩu 4. Hệ thống kiểm tra rỗng 5. Hệ thống kiểm tra hợp lệ 6. Hiển thị thông báo 7. Chuyển sang trang chủ   **Rẽ nhánh 1:**  4.1 Kiểm tra thông tin bị rỗng quay lại bước 3  4.2 Kiểm tra thông tin không rỗng qua bước 5  **Rẽ nhánh 2:**  5.1 Thông tin không hợp lệ quay lại bước 3  5.2 Thông tin hợp lệ qua bước 6 |
| Luồng sự kiện phụ | Khách hàng nhấn nút Cancel  Hệ thống hủy màn hình đăng nhập |

### 3.2.2 Use-case Đăng kí (khách hàng)

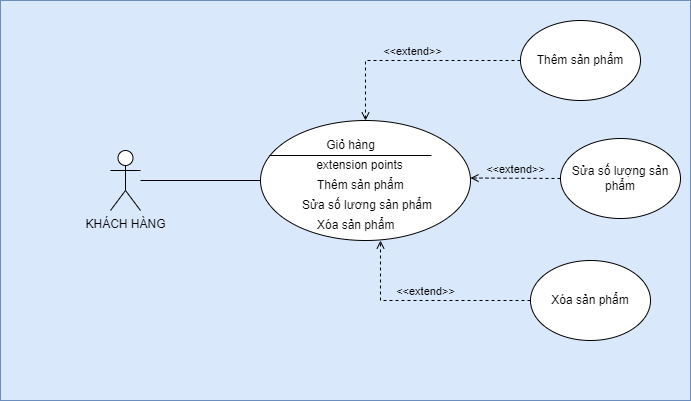


**Hình 3- 3**: Sơ đồ Use-case đăng kí (khách hàng)

**Bảng 3- 2**: Mô tả Use-case đăng kí (khách hàng)

|  |  |
| --- | --- |
| **Tên Use case** | **Đăng kí (khách hàng)** |
| **Actor** | **Khách hàng** |
| **Mô tả** | **Use case cho phép khách hàng đăng kí tài khoản** |
| Điều kiện sau khi kết thúc | * Khách hàng chưa đăng kí * Khách hàng đã đăng kí thành công tài khoản |
| Luồng sự kiện chính | 1. Khách hàng chọn chức năng đăng kí 2. Hệ thống hiển thị ra form đăng kí 3. Khách hàng nhập thông tin đầy đủ trong form 4. Hệ thống kiểm tra rỗng 5. Hệ thống kiểm tra hợp lệ 6. Hiển thị thông báo 7. Chuyển sang trang chủ   **Rẽ nhánh 1:**  4.1 Kiểm tra thông tin bị rỗng quay lại bước 3  4.2 Kiểm tra thông tin không rỗng qua bước 5  **Rẽ nhánh 2:**  5.1 Thông tin không hợp lệ quay lại bước 3  5.2 Thông tin hợp lệ qua bước 6 |
| Luồng sự kiện phụ | Khách hàng nhấn nút Cancel  Hệ thống hủy màn hình đăng kí  Đăng kí thành công chuyển sang màn hình đăng nhập |
|  |  |

### 3.2.3 Use-case Quản lý giỏ hàng

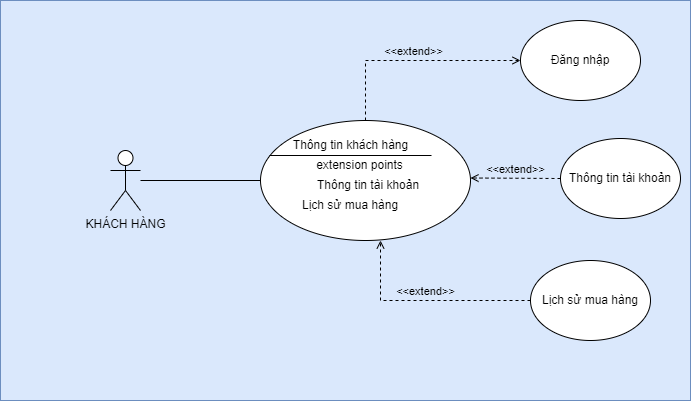


**Hình 3- 4**: Sơ đồ Use-case quản lý giỏ hàng

**Bảng 3- 3**: Mô tả Use-case quản lý giỏ hàng

|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý giỏ hàng** |
| **Actor** | **Khách hàng** |
| **Mô tả** | **Use case cho phép khách hàng thêm, xóa, sửa số lượng sản phẩm trong giỏ hàng** |
| Luồng sự kiện chính | 1. Khách hàng bấm vào biểu tượng giỏ hàng 2. Hệ thống hiển thị danh sách sản phẩm trong giỏ hàng  * Extend Use case Thêm sản phẩm * Extend Use case Sửa số lượng sản phẩm * Extend Use case Xóa sản phẩm |
| Luồng sự kiện phụ | Khách hàng chọn tiếp tục mua  Khách hàng chọn thanh toán |
| <Extend Use Case > | **Thêm sản phẩm:**   1. Khách hàng chọn sản phẩm 2. Khách hàng bấm nút mua sản phẩm 3. Hệ thống cập nhật thông tin và số lượng sản phẩm trong giỏ hàng 4. Hệ thống hiển thị lại danh sách sản phẩm trong giỏ hàng |
| <Extend Use Case > | **Sửa số lượng sản phẩm:**   1. Khách hàng chọn sản phẩm cần thay đổi 2. Khách hàng nhập số lượng mong muốn 3. Hệ thống cập nhật thông tin và số lượng sản phẩm trong giỏ hàng 4. Hệ thống hiển thị lại danh sách sản phẩm trong giỏ hàng |
| <Extend Use Case > | **Xóa sản phẩm:**   1. Khách hàng chọn sản phẩm cần xóa trong giỏ hàng 2. Khách hàng chọn xóa sản phẩm 3. Hệ thống cập nhật thông tin và số lượng sản phẩm trong giỏ hàng 4. Hệ thống hiển thị lại danh sách sản phẩm trong giỏ hàng |
|  |  |

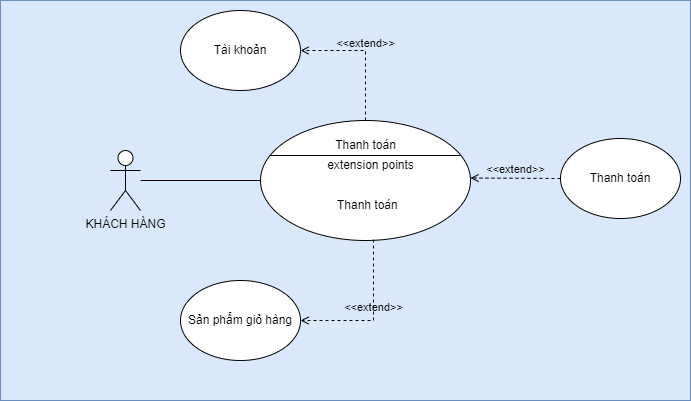
### 3.2.4 Use-case Quản lý thông tin tài khoản



**Hình 3- 5**: Sơ đồ Use-case quản lý thông tin tài khoản**Bảng 3- 4**: Mô tả Use-case quản lý thông tin tài khoản

|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý thông tin tài khoản** |
| **Actor** | **Khách hàng** |
| **Mô tả** | **Use case cho phép khách hàng xem thông tin cá nhân** |
| Luồng sự kiện chính | 1. Khách hàng đăng nhập vào hệ thống 2. Khách hàng chọn thông tin cá nhân 3. Hiển thị màn hình quản lý thông tin tài khoản  * Extend Use case Thông tin tài khoản * Extend Use case Lịch sử đơn hàng |
| <Extend Use Case > | **Thông tin tài khoản**   1. Hệ thống hiện ra thông tin khách hàng 2. Khách hàng chọn thông tin cần thay đổi 3. Khách hàng chỉnh sửa thông tin 4. Hệ thống cập nhật thông tin 5. Thông báo cập nhật thành công |
| <Extend Use Case > | **Lịch sử đơn hàng**   1. Hệ thống hiện ra thông tin đơn hàng 2. Khách hàng chọn hủy đơn hàng 3. Hệ thống cập nhật dữ liệu 4. Thông báo Hủy đơn hàng thành công   **Rẽ nhánh 1:**  2.1 Nếu đơn hàng đã xác nhận thì không thể hùy |
|  |  |

### 3.2.5 Use-case Thanh toán

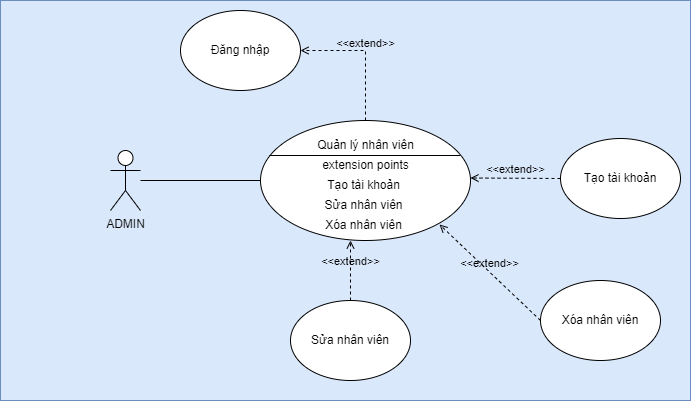


**Hình 3- 6**: Sơ đồ Use-case thanh toán

**Bảng 3- 5**: Mô tả Use-case thanh toán

|  |  |
| --- | --- |
| **Tên Use case** | **Thanh toán** |
| **Actor** | **Khách hàng** |
| **Mô tả** | **Use case cho phép khách hàng thanh toán sản phẩm** |
| Luồng sự kiện chính | 1. Khách hàng chọn chức năng thanh toán 2. Hệ thống hiển thị ra thông tin người mua 3. Hệ thống kiểm tra rỗng 4. Hệ thống kiểm tra hợp lệ 5. Hệ thống hiển thị ra thông tin đơn hàng 6. Chọn nút xác nhận 7. Hiển thị thông báo 8. Chuyển sang trang chủ   **Rẽ nhánh 1:**  2.1 Nếu đã đăng nhập thì hệ thống lấy thông tin của khách hàng  2.2 Nếu chưa đăng nhập thì hệ thống yêu cầu khách hàng nhập thông tin  **Rẽ nhánh 2:**  3.1 Thông tin rỗng yêu cầu khách hàng đăng nhập hoặc điền đầy đủ thông tin  3.2 Thông tin không rỗng qua bước 4  **Rẽ nhánh 3:**  4.1 Thông tin không hợp lệ yêu cầu khách hàng điền lại thông tin  4.2 Thông tin hợp lệ qua bước 5 |
| Luồng sự kiện phụ | Khách hàng chưa đăng nhập thì nhập thông tin người mua |
|  |  |

### 3.2.6 Use-case Quản lý nhân viên

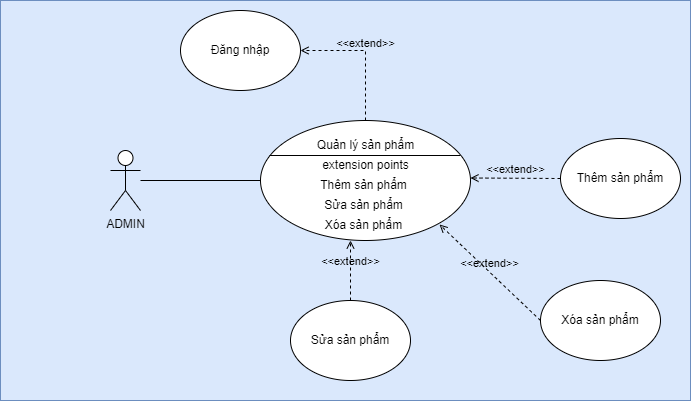


**Hình 3- 7**: Sơ đồ Use-case quản lý nhân viên

**Bảng 3- 6**: Mô tả Use-case quản lý nhân viên

|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý nhân viên** |
| **Actor** | **Chủ cửa hàng** |
| **Mô tả** | **Use case cho phép chủ cửa hàng tạo tài khoản cho nhân viên, xóa nhân viên, sửa thông tin nhân viên** |
| Luồng sự kiện chính | 1. Admin đăng nhập vào hệ thống 2. Admin chọn quản lý nhân viên 3. Hiển thị màn hình quản lý nhân viên  * Extend Use case Tạo tài khoản * Extend Use case Xóa nhân viên * Extend Use case Sửa nhân viên |
| <Extend Use Case > | **Tạo tài khoản nhân viên**   1. Hệ thống hiện ra form tạo nhân viên 2. Admin nhập thông tin nhân viên 3. Kiểm tra rỗng 4. Kiểm tra hợp lệ 5. Thông báo thêm thành công 6. Quay lại trang quản lý nhân viên   **Rẽ nhánh 1:**  3.1 Kiểm tra thông tin rỗng về bước 2  3.2 Thông tin không rỗng qua bước 4  **Rẽ nhánh 2:**  4.1 Kiểm tra không hợp lệ về bước 2  4.2 Kiểm tra hợp lệ qua bước 5 |
| <Extend Use Case > | **Xóa nhân viên**   1. Admin chọn nhân viên cần xoá. 2. Admin nhấn nút Xoá. 3. Kiểm tra xác nhận xóa nhân viên 4. Cập nhật CSDL. 5. Hiển thị thông báo.   **Rẽ nhánh 1:**  3.1 Kiểm tra xác nhận không xóa quay lại bước 1  3.2 Xác nhận xóa qua bước 4. |
| <Extend Use Case > | **Sửa nhân viên**   1. Admin chọn nhân viên cần sửa 2. Admin thay đổi thông tin 3. Kiểm tra rỗng 4. Kiểm tra hợp lệ 5. Cập nhật CSDL 6. Hiển thị thông báo   **Rẽ nhánh 1:**  3.1 Kiểm tra thông tin bị rỗng. Lập lại bước 2  3.2 Thông tin không rỗng qua bước 4.  **Rẽ nhánh 2:**  4.1 Kiểm tra thông tin không hợp lệ . Lập lại bước 2  4.2 Thông tin hợp lệ cập nhật dữ liệu. |
|  |  |

### 3.2.7 Use-case Quản lý sản phẩm

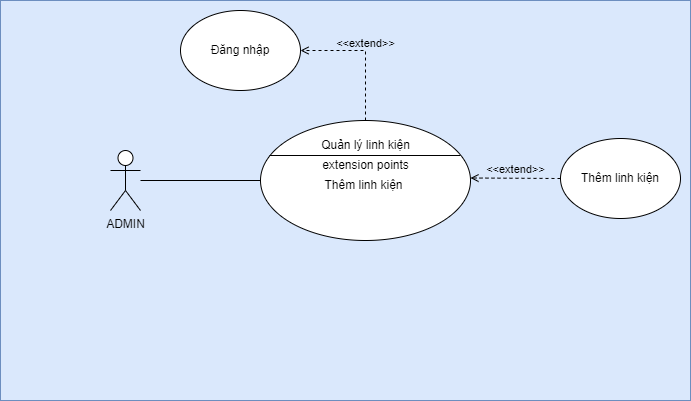


**Hình 3- 8**: Sơ đồ Use-case quản lý sản phẩm

**Bảng 3- 7**: Mô tả Use-case quản lý sản phẩm

|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý sản phẩm** |
| **Actor** | **Chủ cửa hàng** |
| **Mô tả** | **Use case cho phép chủ cửa hàng thêm, xóa, sửa sản phẩm** |
| Luồng sự kiện chính | 1. Admin đăng nhập vào hệ thống 2. Admin chọn quản lý sản phẩm 3. Hiển thị màn hình quản lý sản phẩm  * Extend Use case Thêm sản phẩm * Extend Use case Xóa sản phẩm * Extend Use case Sửa sản phẩm |
| <Extend Use Case > | **Thêm sản phẩm**   1. Admin chọn thêm sản phẩm 2. Admin nhập thông tin sản phẩm 3. Nhấn nút Thêm. 4. Kiểm tra thông tin rỗng 5. Kiểm tra trùng 6. Kiểm tra hợp lệ 7. Cập nhật CSDL. 8. Hiển thị thông báo.   **Rẽ nhánh 1:**  4.1 Kiểm tra thông tin rỗng về bước 2.  4.2 Thông tin không rỗng qua bước 4  **Rẽ nhánh 2:**  5.1 Kiểm tra dữ liệu trùng về bước 2.  5.2 Dữ liệu không trùng qua bước 5.  **Rẽ nhánh 3:**  6.1 Kiểm tra dữ liệu không hợp lệ về bước 2.  6.2 Dữ liệu hợp lệ lưu CSDL |
| <Extend Use Case > | **Xóa sản phẩm**   1. Admin chọn sản phẩm cần xóa 2. Nhấn nút Xoá 3. Kiểm tra xác nhận xóa thông tin 4. Cập nhật CSDL. 5. Hiển thị thông báo.   **Rẽ nhánh 1:**  3.1 Kiểm tra xác nhận không xóa thông tin quay lại bước 1  3.2 Xác nhận xóa qua bước 4. |
| <Extend Use Case > | **Sửa sản phẩm**   1. Admin chọn sửa sản phẩm 2. Nhập thông tin cần sửa 3. Nhấn nút Cập nhật 4. Kiểm tra rỗng 5. Kiểm tra hợp lệ 6. Cập nhật CSDL. 7. Hiển thị thông báo.   **Rẽ nhánh 1:**  4.1 Kiểm tra thông tin bị rỗng. Lập lại bước 2  4.2 Thông tin không rỗng qua bước 5  **Rẽ nhánh 2:**  5.1 Kiểm tra thông tin không hợp lệ . Lập lại bước 2  5.2 Thông tin hợp lệ. Qua bước 6 cập nhật dữ liệu |

### 3.2.8 Use-case Quản lý linh kiện

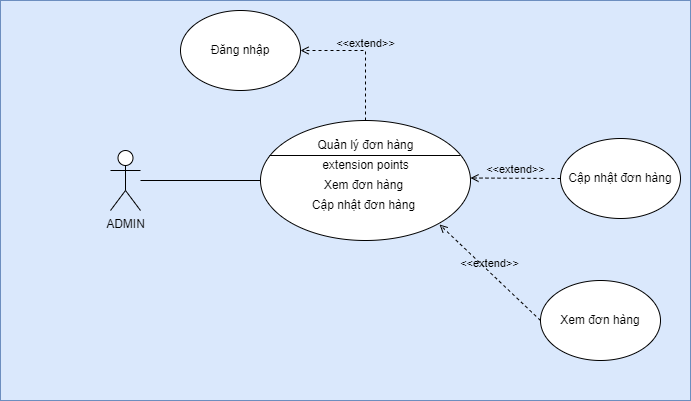


**Hình 3- 9**: Sơ đồ use-case quản lý linh kiện

**Bảng 3- 8**: Mô tả use-case quản lý linh kiện

|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý linh kiện** |
| **Actor** | **Chủ cửa hàng** |
| **Mô tả** | **Use case cho phép chủ cửa hàng thêm linh kiện cần thiết** |
| Luồng sự kiện chính | 1. Admin đăng nhập vào hệ thống 2. Admin chọn quản lý linh kiện 3. Hiển thị màn hình quản lý linh kiện  * Extend Use case Thêm linh kiện |
| <Extend Use Case > | **Thêm linh kiện**   1. Hệ thống hiện ra form thêm linh kiện 2. Admin nhập thông tin nhân viên 3. Kiểm tra rỗng 4. Kiểm tra hợp lệ 5. Thông báo thêm thành công 6. Quay lại trang quản lý linh kiện   **Rẽ nhánh 1:**  3.1 Kiểm tra thông tin rỗng về bước 2  3.2 Thông tin không rỗng qua bước 4  **Rẽ nhánh 2:**  4.1 Kiểm tra không hợp lệ về bước 2  4.2 Kiểm tra hợp lệ qua bước 5 |

### 3.2.9 Use-case Quản lý đơn hàng

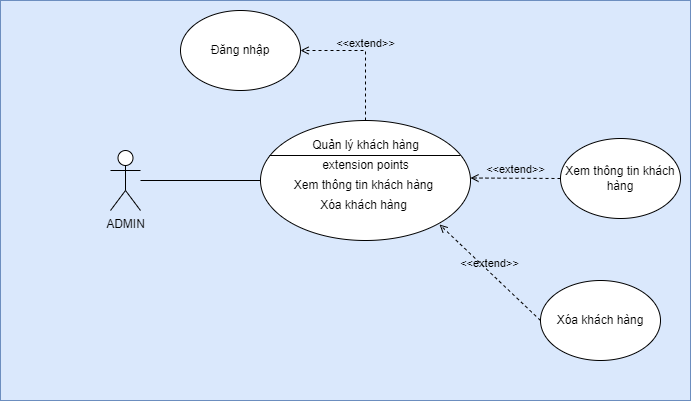


**Hình 3- 10**: Sơ đồ use-case quản lý đơn hàng

**Bảng 3- 9**: Mô tả use-case quản lý đơn hàng

|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý đơn hàng** |
| **Actor** | **Chủ cửa hàng** |
| **Mô tả** | **Use case cho phép chủ cửa hàng xem đơn hàng và cập nhật đơn hàng** |
| Luồng sự kiện chính | 1. Admin đăng nhập vào hệ thống 2. Admin chọn quản lý đơn hàng 3. Hiển thị màn hình quản lý đơn hàng  * Extend Use case Xem đơn hàng * Extend Use case Cập nhật đơn hàng |
| <Extend Use Case > | **Xem đơn hàng**   1. Admin chọn đơn hàng cần xem 2. Hiển thị chi tiết đơn hàng |
| <Extend Use Case > | **Cập nhật đơn hàng**   1. Admin chọn đơn hàng cần cập nhật 2. Admin chọn thay đổi trạng thái 3. Cập nhật CSDL. 4. Hiển thị thông báo. |

### 3.2.10 Use-case Quản lý khách hàng



**Hình 3- 11**: Sơ đồ use-case quản lý khách hàng

**Bảng 3- 10**: Mô tả use-case quản lý khách hàng

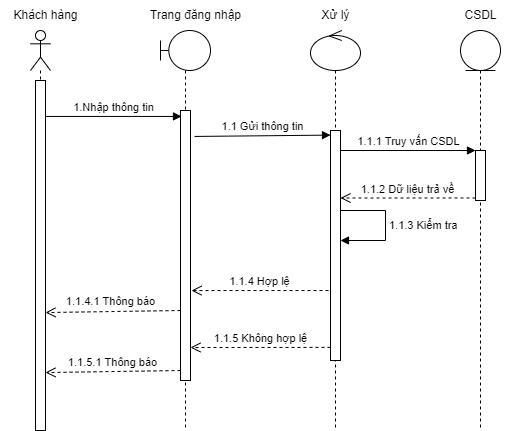
|  |  |
| --- | --- |
| **Tên Use case** | **Quản lý khách hàng** |
| **Actor** | **Chủ cửa hàng** |
| **Mô tả** | **Use case cho phép chủ cửa hàng xem thông tin khách hàng và xóa khách hàng** |
| Luồng sự kiện chính | 1. Admin đăng nhập vào hệ thống 2. Admin chọn quản lý khách hàng 3. Hiển thị màn hình quản lý khách hàng  * Extend Use case Xem khách hàng * Extend Use case Xóa khách hàng |
| <Extend Use Case > | **Xem khách hàng**   1. Admin chọn khách hàng cần xem 2. Hiển thị chi tiết khách hàng |
| <Extend Use Case > | **Xóa hàng**   1. Admin chọn khách hàng cần xóa 2. Nhấn nút Xoá 3. Kiểm tra xác nhận xóa thông tin 4. Cập nhật CSDL. 5. Hiển thị thông báo.   **Rẽ nhánh 1:**  3.1 Kiểm tra xác nhận không xóa thông tin quay lại bước 1  3.2 Xác nhận xóa qua bước 4. |

# 

# **Chương 4. THÀNH PHẦN XỬ LÝ**

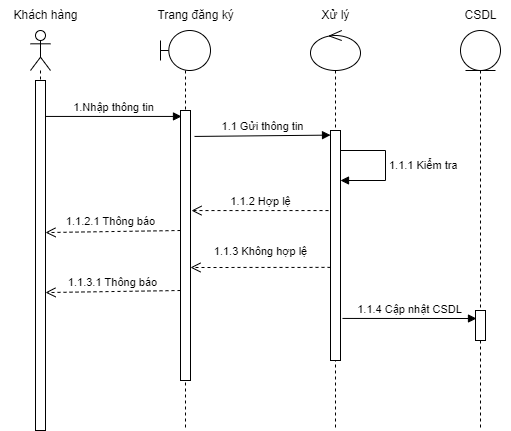
## **4.1 SƠ ĐỒ TUẦN TỰ**

### 4.1.1 Sơ đồ tuần tự đăng nhập



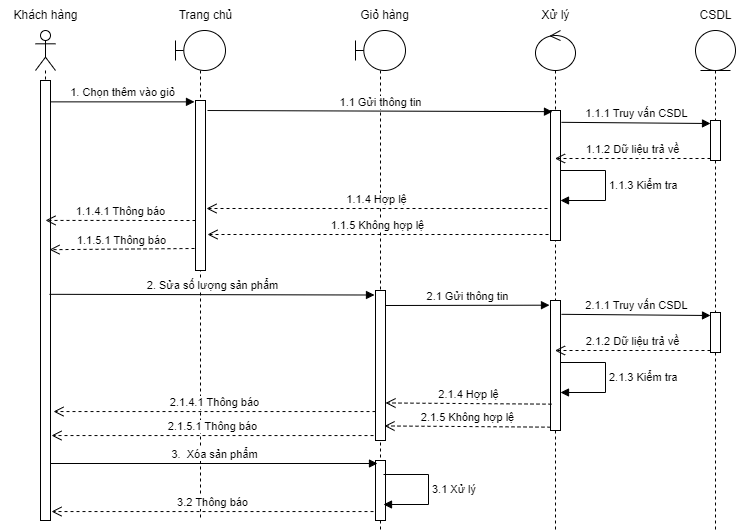
**Hình 4- 1**: Sơ đồ tuần tự đăng nhập

### 4.1.2 Sơ đồ tuần tự đăng ký



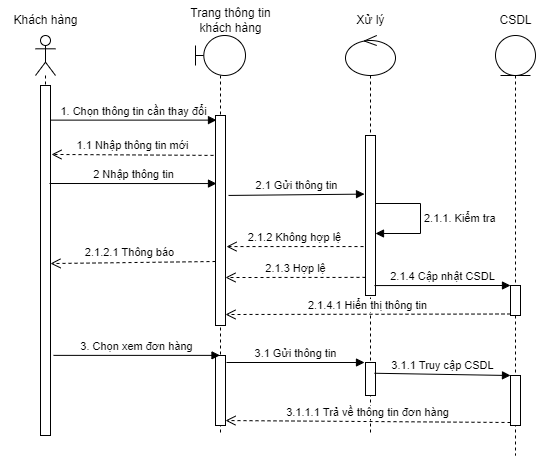
**Hình 4- 2**: Sơ đồ tuần tự đăng ký

### 4.1.3 Sơ đồ tuần tự quản lý giỏ hàng



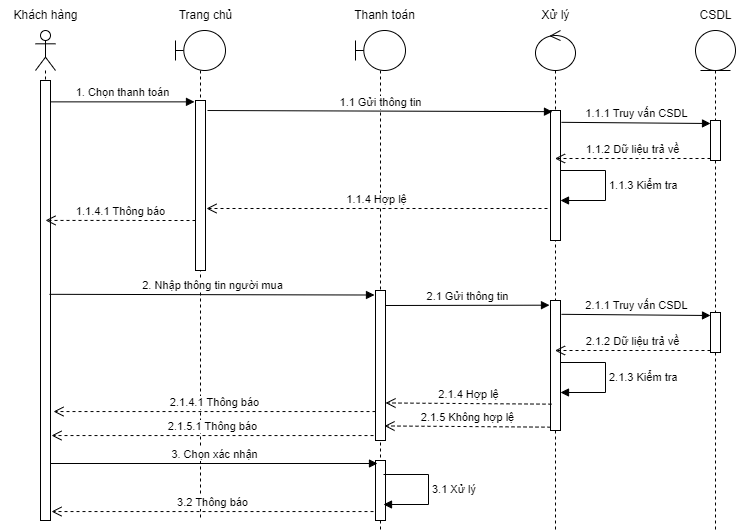
**Hình 4- 3**: Sơ đồ tuần tự quản lý giỏ hàng

### 4.1.4 Sơ đồ tuần tự quản lý thông tin khách hàng



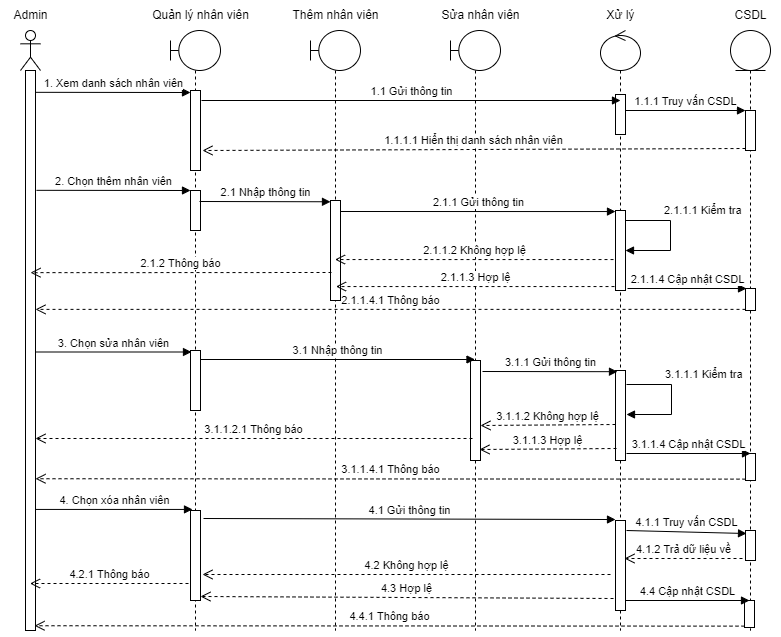
**Hình 4- 4**: Sơ đồ tuần tự quản lý thông tin khách hàng

### 4.1.5 Sơ đồ tuần tự thanh toán



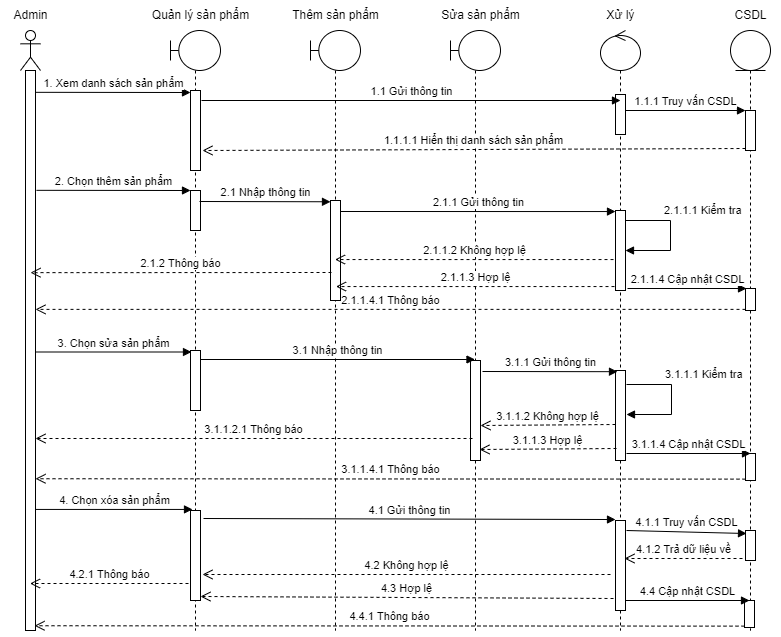
**Hình 4- 5**: Sơ đồ tuần tự thanh toán

### 4.1.6 Sơ đồ tuần tự quản lý nhân viên



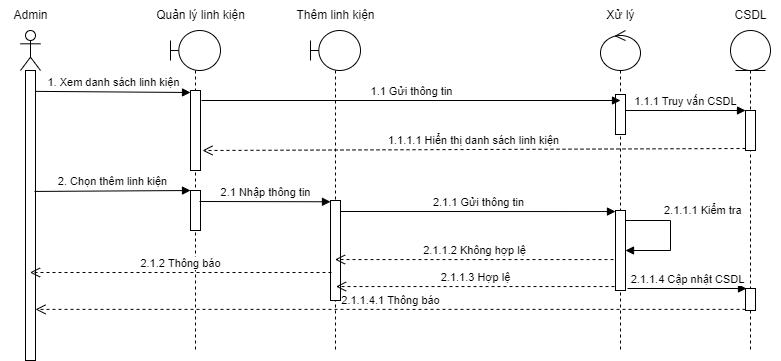
**Hình 4- 6**: Sơ đồ tuần tự quản lý nhân viên

### 4.1.7 Sơ đồ tuần tự quản lý sản phẩm



**Hình 4- 7**: Sơ đồ tuần tự quản lý sản phẩm

### 4.1.8 Sơ đồ tuần tự quản lý linh kiện

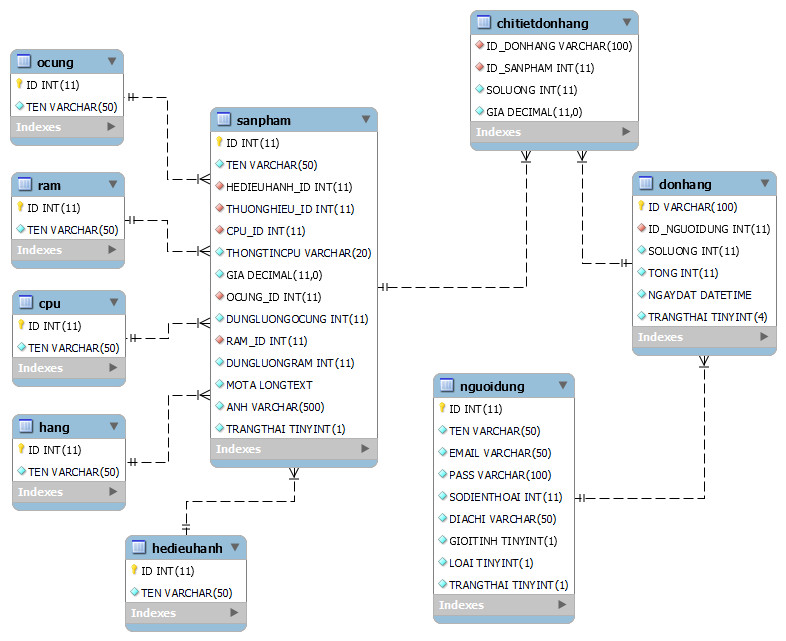


**Hình 4- 8**: Sơ đồ tuần tự quản lý linh kiện

# 

# **Chương 5. THÀNH PHẦN DỮ LIỆU**

## **5.1 SƠ ĐỒ DỮ LIỆU LOGIC**



**Hình 5- 1**: Sơ đồ dữ liệu Logic

## **5.2 THIẾT KẾ CƠ SỞ DỮ LIỆU**

**Bảng 5- 1**: Bảng dữ liệu "nguoidung"

| **Mô tả:** Bảng nguoidung chứa thông tin người dùng có tài khoản tại website | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã người dùng |
| TEN | varchar(50) |  |  | x | Tên người dùng |
| EMAIL | varchar(50) |  | x | x | Email người dùng |
| PASS | varchar(100) |  |  | x | Mật khẩu người dùng |
| SODIENTHOAI | int(11) |  | x | x | Số điện thoại người dùng |
| DIACHI | varchar(50) |  |  | x | Địa chỉ người dùng |
| GIOITINH | tinyint(1) |  |  | x | Giới tính người dùng |
| LOAI | tinyint(1) |  |  |  | 0 là khách hàng, 1 là nhân viên |
| TRANGTHAI | tinyint(1) |  |  |  | 0 là hoạt động , 1 là tắt kích hoạt ( khi tạo thì mặc định là 0) |

**Bảng 5- 2**: Bảng dữ liệu "sanpham"

| **Mô tả:** Bảng sanpham chứa thông tin sản phẩm | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã sản phẩm |
| TEN | varchar(50) |  | x | x | Tên sản phẩm |
| HEDIEUHANH\_ID | int(11) |  |  | x | Mã hệ điều hành |
| THUONGHIEU\_ID | int(11) |  |  | x | Mã thương hiệu |
| CPU\_ID | int(11) |  |  | x | Mã CPU |
| THONGTINCPU | varchar(20) |  |  | x | Thông tin CPU |
| GIA | decimal(11,0) |  |  | x | Giá sản phẩm |
| OCUNG\_ID | int(11) |  |  | x | Phân loại người dùng |
| DUNGLUONGOCUNG | int(11) |  |  | x | Thông tin ổ cứng |
| RAM\_ID | int(11) |  |  | x | Mã RAM |
| DUNGLUONGRAM | int(11) |  |  | x | Thông tin RAM |
| MOTA | longtext |  |  | x | Mô tả sản phẩm |
| ANH | varchar(500) |  | x | x | Hình ảnh sản phẩm |
| TRANGTHAI | tinyint(1) |  |  |  | 0 là hoạt động , 1 là tắt kích hoạt ( khi tạo thì mặc định là 0) |

**Bảng 5- 3**: Bảng dữ liệu "hang"

| **Mô tả:** Bảng hang chứa thông tin hãng sản xuất | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã hãng |
| TEN | varchar(50) |  | x | x | Tên hãng |

**Bảng 5- 4**: Bảng dữ liệu "CPU"

| **Mô tả:** Bảng CPU chứa thông tin CPU | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã CPU |
| TEN | varchar(50) |  | x | x | Tên CPU |

**Bảng 5- 5**: Bảng dữ liệu "hedieuhanh"

| **Mô tả:** Bảng hedieuhanh chứa thông tin hệ điều hành | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã hệ điều hành |
| TEN | varchar(50) |  | x | x | Tên hệ điều hành |

**Bảng 5- 6**: Bảng dữ liệu "ocung"

| **Mô tả:** Bảng ocung chứa thông tin ổ cứng | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã ổ cứng |
| TEN | varchar(50) |  | x | x | Tên ổ cứng |

**Bảng 5- 7**: Bảng dữ liệu "ram"

| **Mô tả:** Bảng ram chứa thông tin RAM | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | int(11) | x | x | x | Mã RAM |
| TEN | varchar(50) |  | x | x | Tên RAM |

**Bảng 5- 8**: Bảng dữ liệu "chitietdonhang"

| **Mô tả:** Bảng chitietdonhang chứa thông tin chi tiết đơn hàng | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID\_DONHANG | varchar(100) | x | x | x | Mã đơn hàng |
| ID\_SANPHAM | int(11) |  | x | x | Mã sản phẩm |
| SOLUONG | int(11) |  |  | x | Số lượng sản phẩm |
| GIA | decimal(11,0) |  |  | x | Giá sản phẩm |

**Bảng 5- 9**: Bảng dữ liệu "donhang"

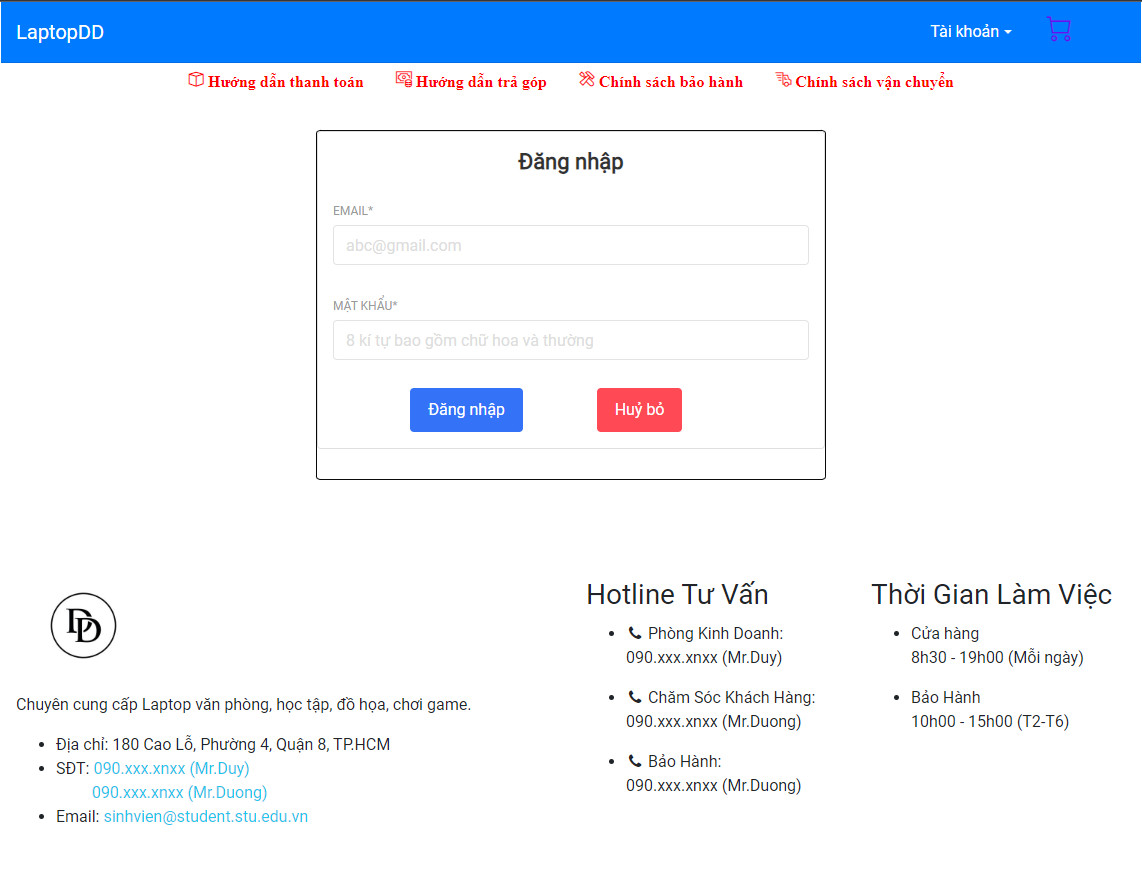
| **Mô tả:** Bảng donhang chứa thông tin đơn hàng | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| ID | varchar(100) | x | x | x | Mã đơn hàng |
| ID\_NGUOIDUNG | int(11) |  | x | x | Mã người dùng |
| SOLUONG | int(11) |  |  | x | Số lượng sản phẩm |
| TONG | int(11) |  |  | x | Tổng giá sản phẩm |
| NGAYDAT | datetime |  |  | x | Ngày đặt |
| TRANGTHAI | tinyint(4) |  |  |  | 0 chưa xác nhận, 1 xác nhận, 2 đang vận chuyển, 3 hoàn thành, -1 hủy |

**K**: key; **U**: unique; **M**: mandatory (not nul)

# **Chương 6. THÀNH PHẦN GIAO DIỆN**

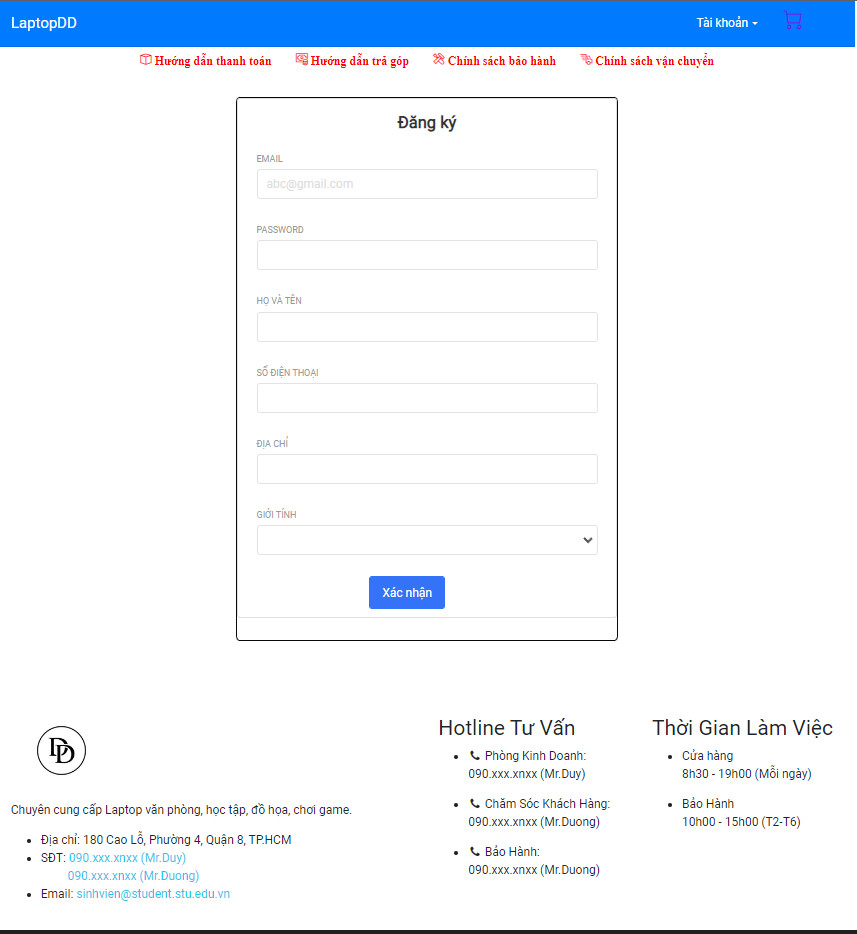
## **6.1 GIAO DIỆN KHÁCH HÀNG**

### 6.1.1 Giao diện đăng nhập khách hàng



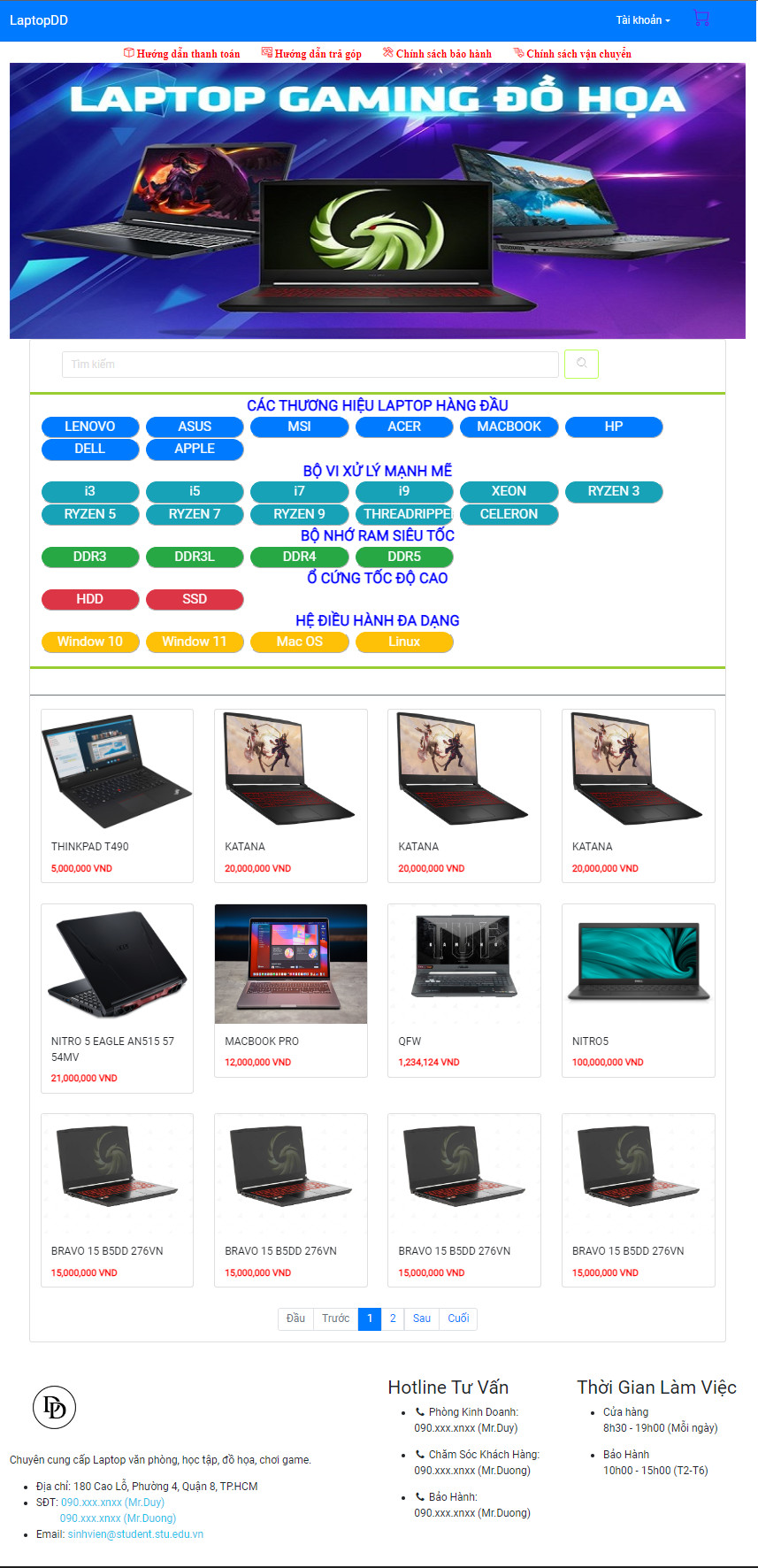
**Hình 6- 1**: Giao diện đăng nhập khách hàng

### 6.1.2 Giao diện đăng ký khách hàng



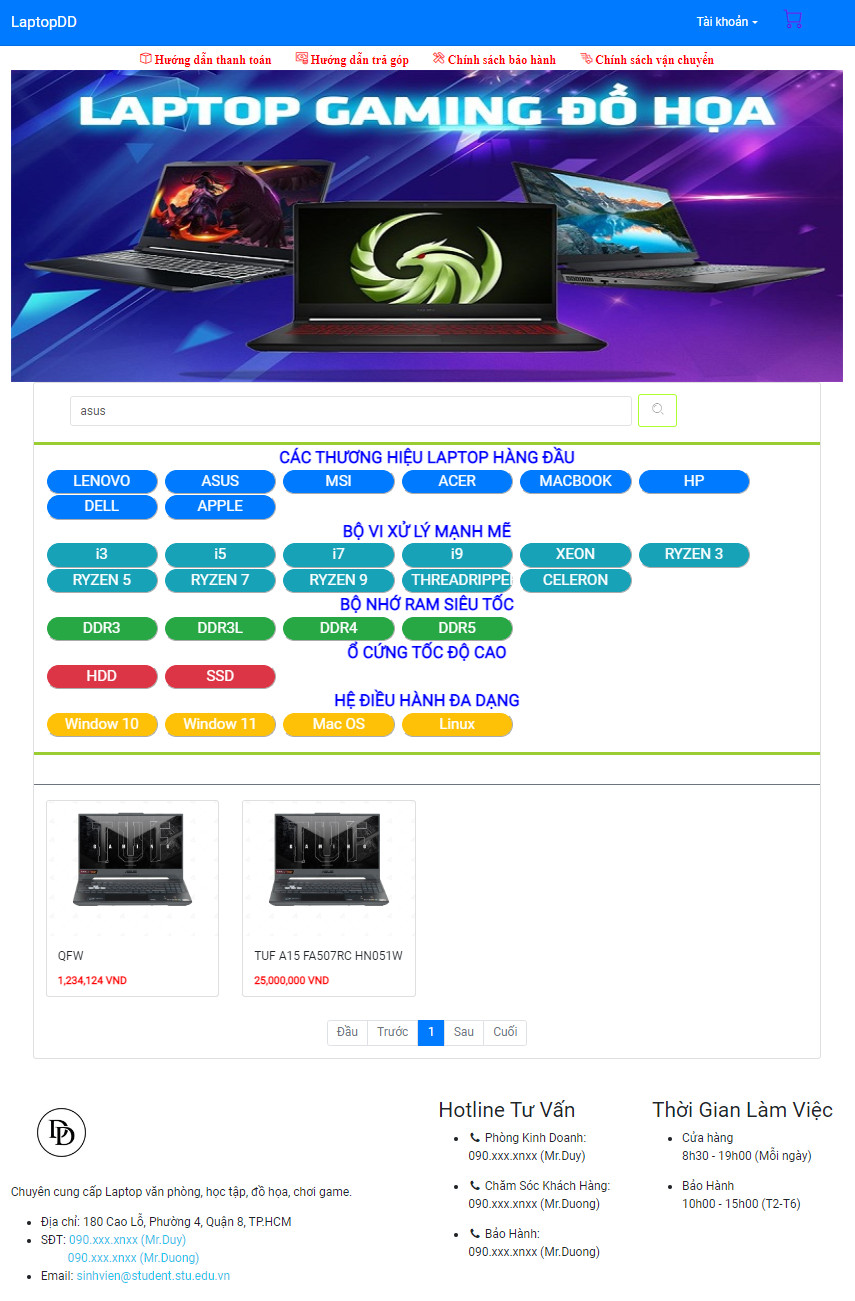
**Hình 6- 2**: Giao diện đăng ký khách hàng

### 6.1.3 Giao diện trang chủ



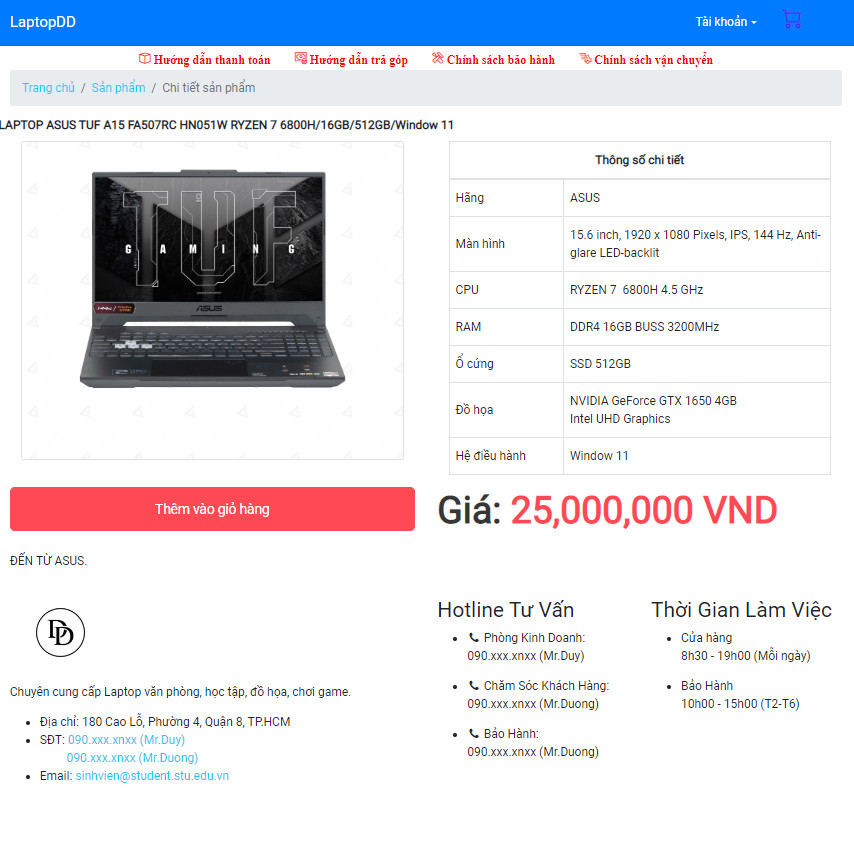
**Hình 6- 3**: Giao diện trang chủ

### 6.1.4 Giao diện danh mục sản phẩm



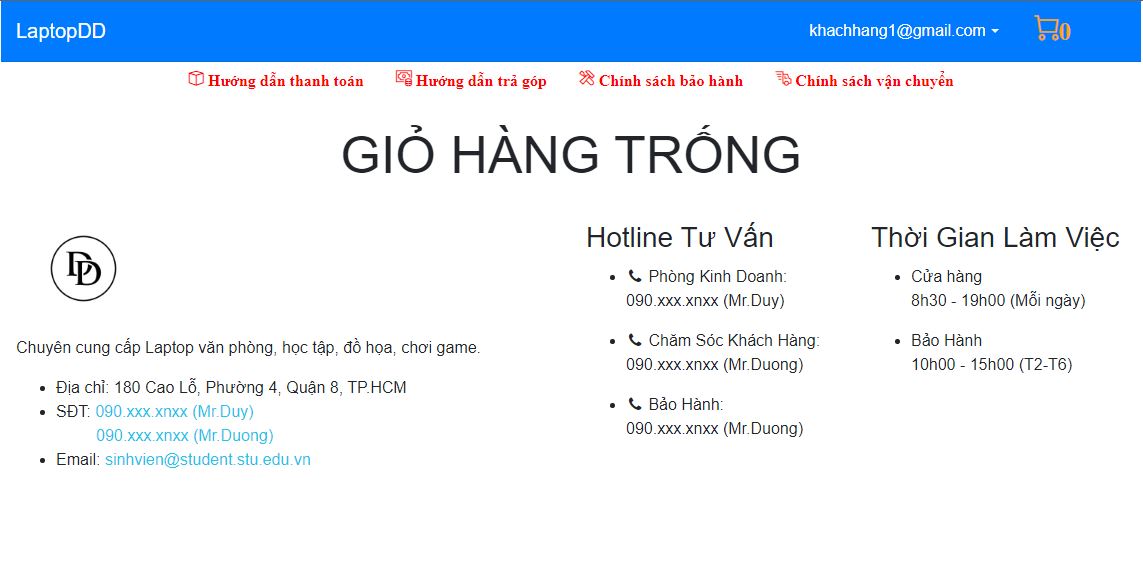
**Hình 6- 4**: Giao diện danh mục sản phẩm

### 6.1.5 Giao diện chi tiết sản phẩm

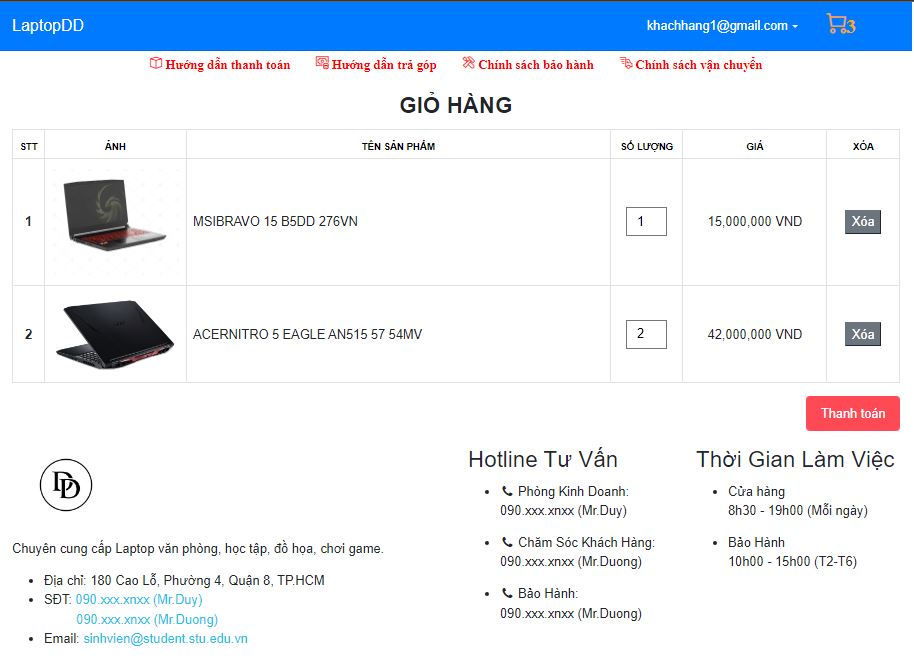


**Hình 6- 5**: Giao diện chi tiết sản phẩm

### 6.1.6 Giao diện giỏ hàng

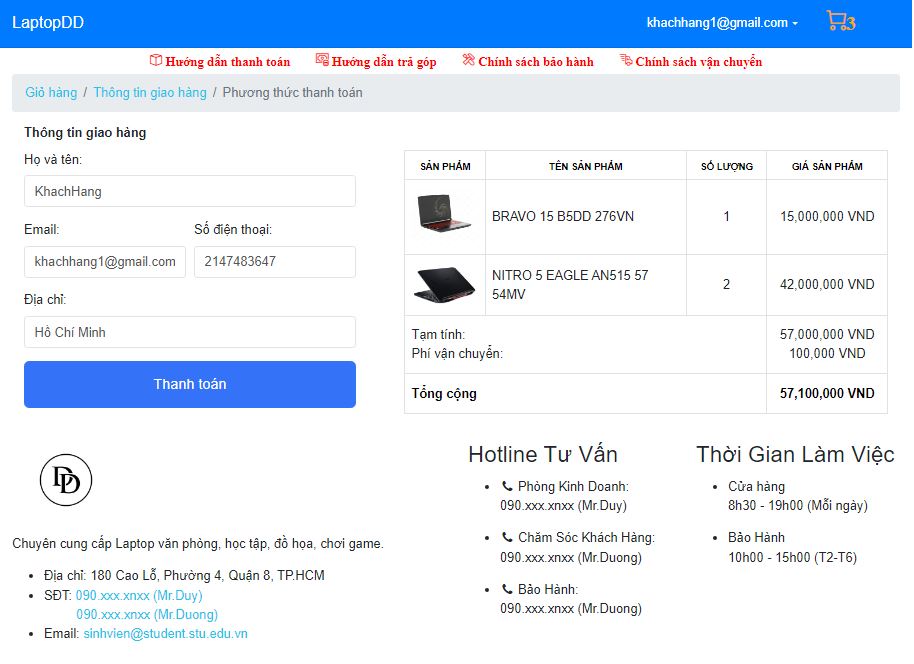


**Hình 6- 6**: Giao diện giỏ hàng trống



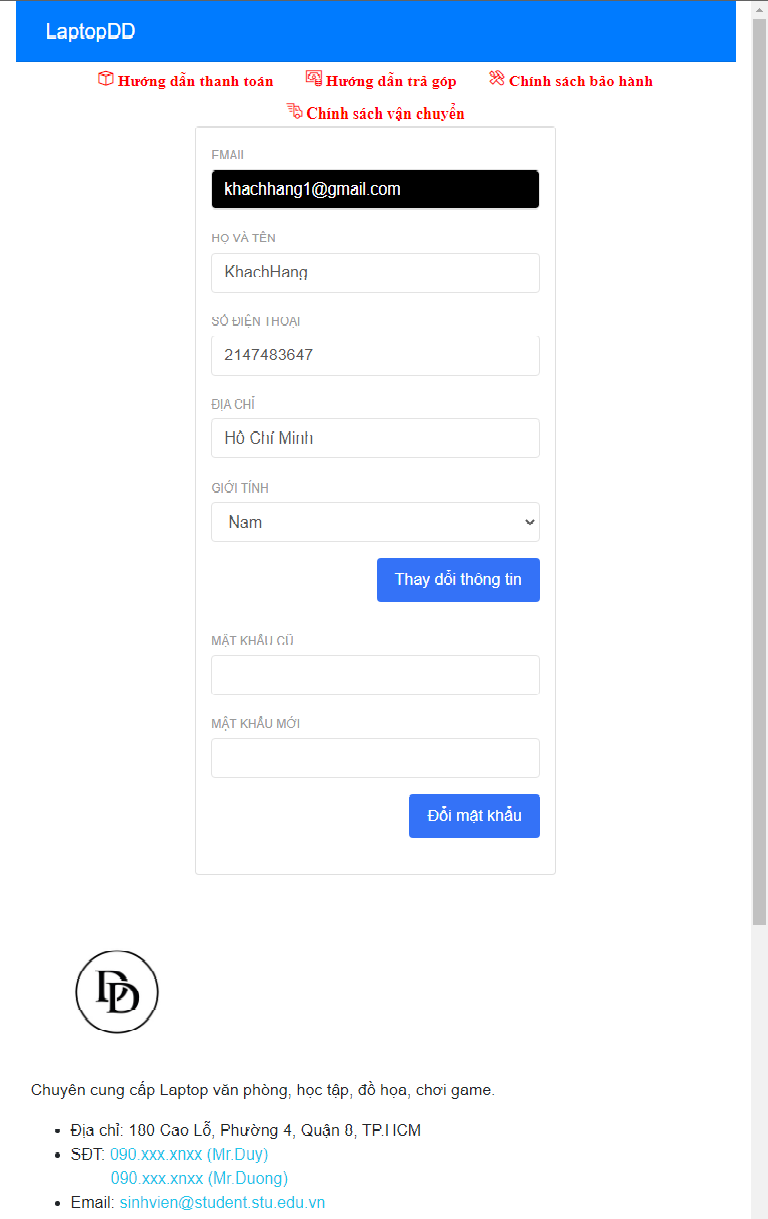
**Hình 6- 7**: Giao diện giỏ hàng

### 6.1.7 Giao diện thanh toán



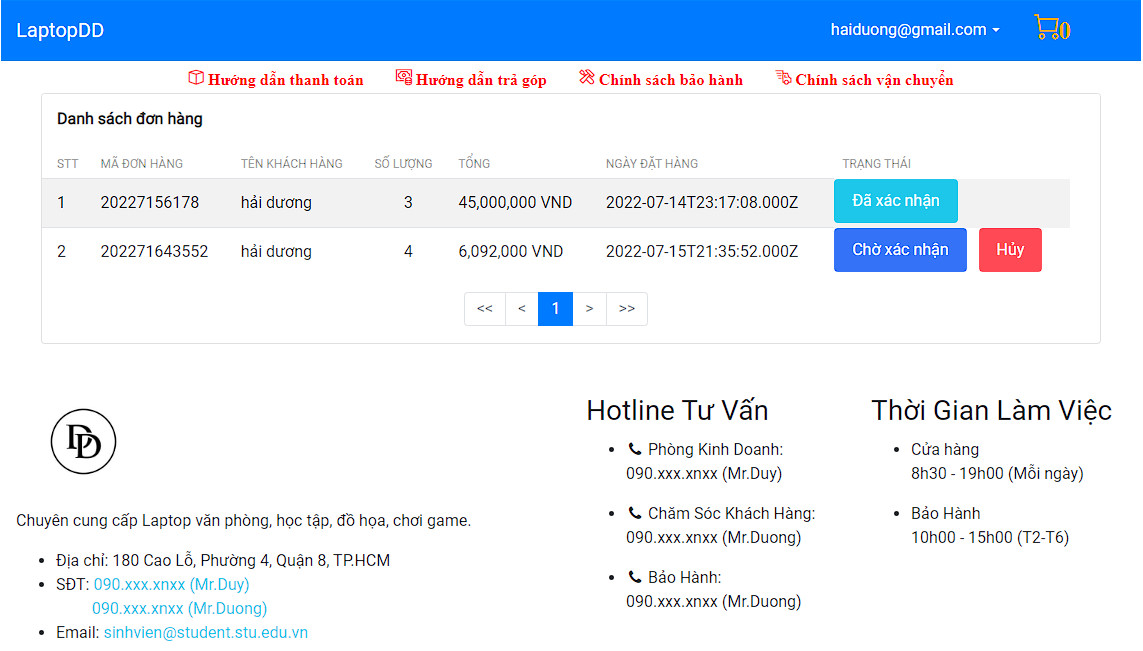
**Hình 6- 8**: Giao diện thanh toán

### 6.1.8 Giao diện thông tin khách hàng

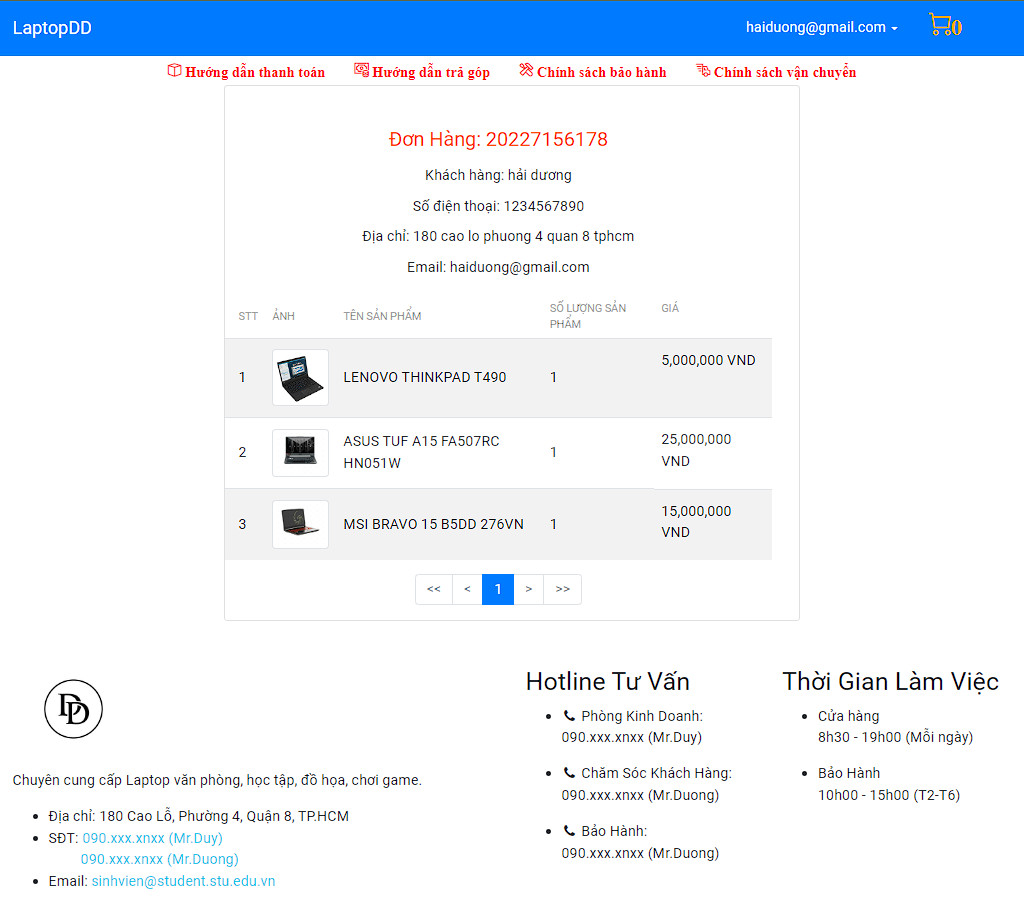


**Hình 6- 9**: Giao diện chi tiết khách hàng

### 6.1.9 Giao diện lịch sử đơn hàng



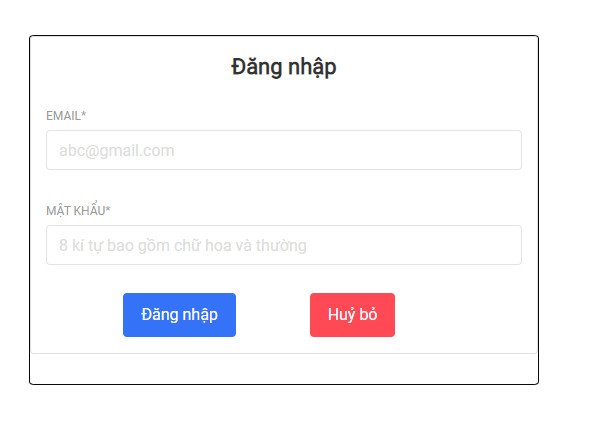
**Hình 6- 10**: Giao diện theo dỗi đơn hàng



**Hình 6- 11**: Giao diện chi tiết đơn hàng

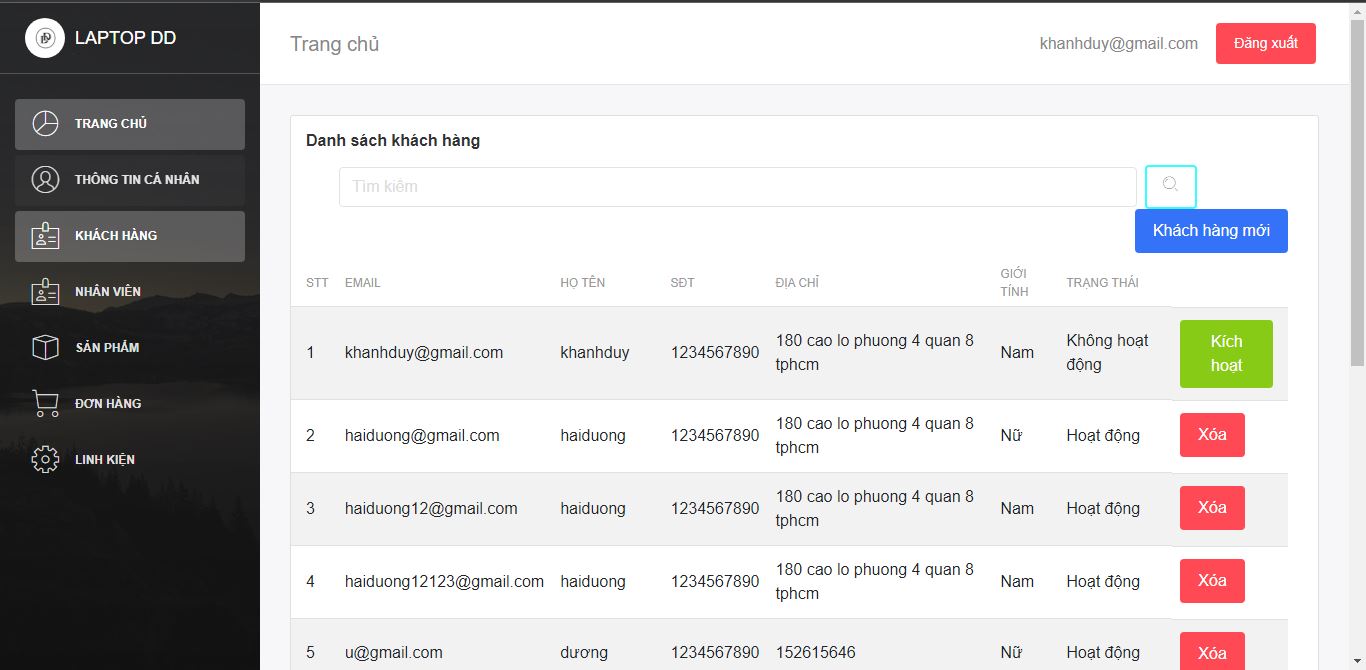
## **6.2** **GIAO DIỆN NHÀ QUẢN LÝ**

### 6.2.1 Giao diện đăng nhập



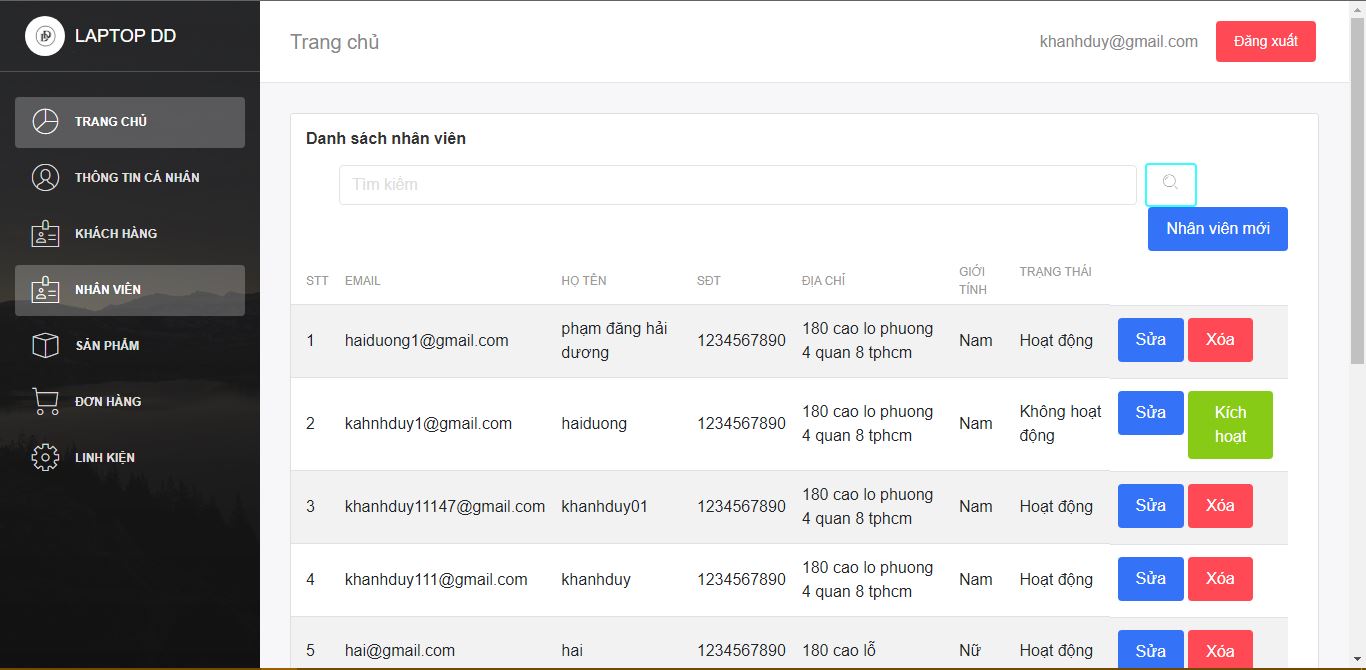
**Hình 6- 12**: Giao diện đăng nhập cho admin

### 6.2.2 Giao diện quản lý khách hàng



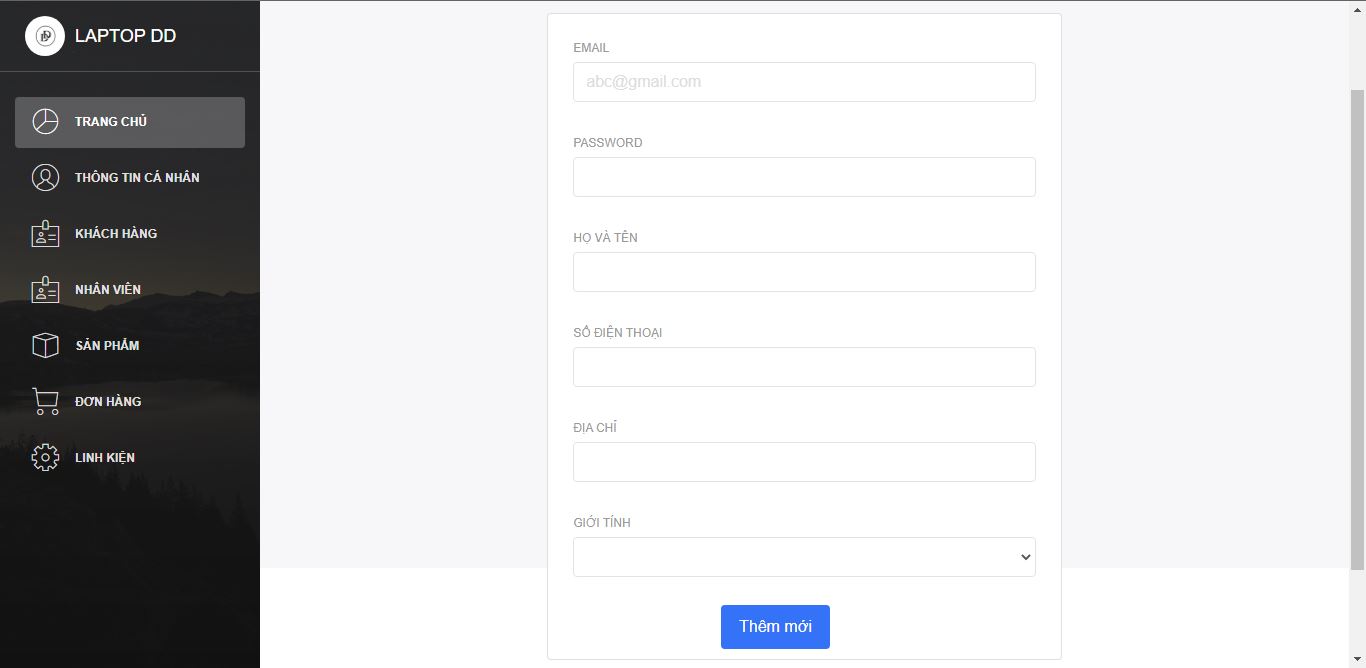
**Hình 6- 13**: Giao diện quản lý khách hàng

### 6.2.3 Giao diện quản lý nhân viên



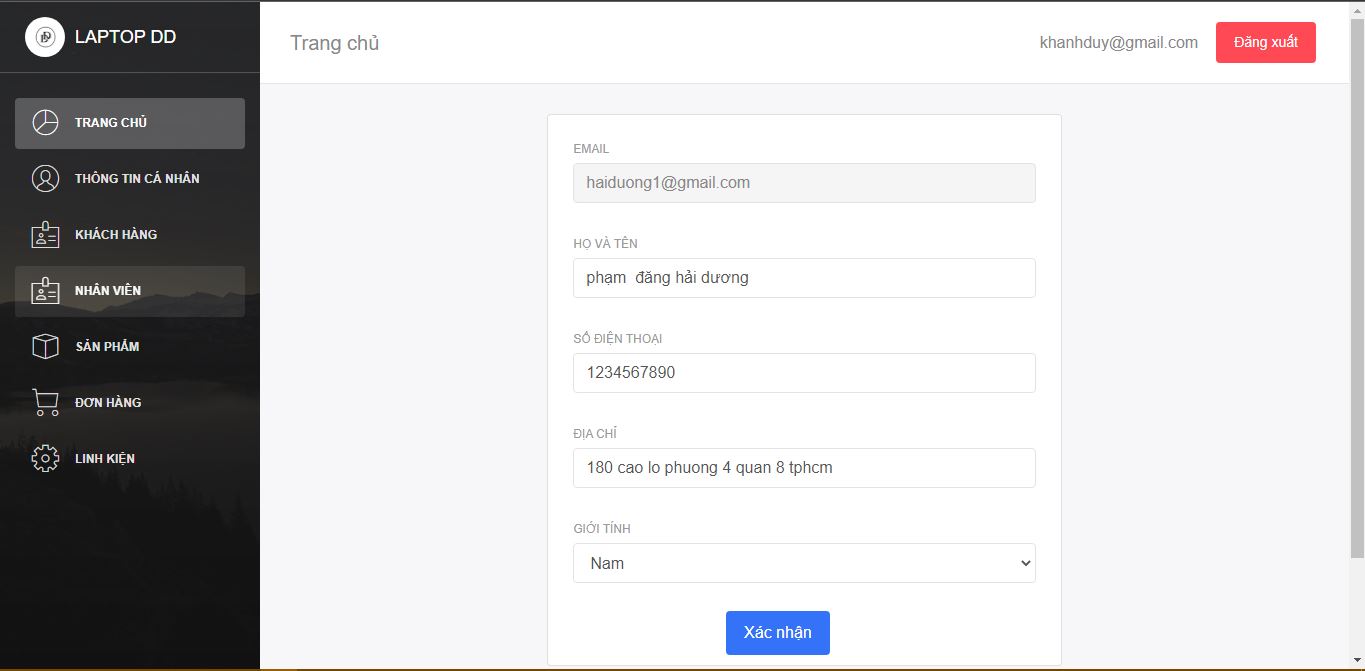
**Hình 6- 14**: Giao diện quản lý nhân viên

### 6.2.4 Giao diện thêm nhân viên



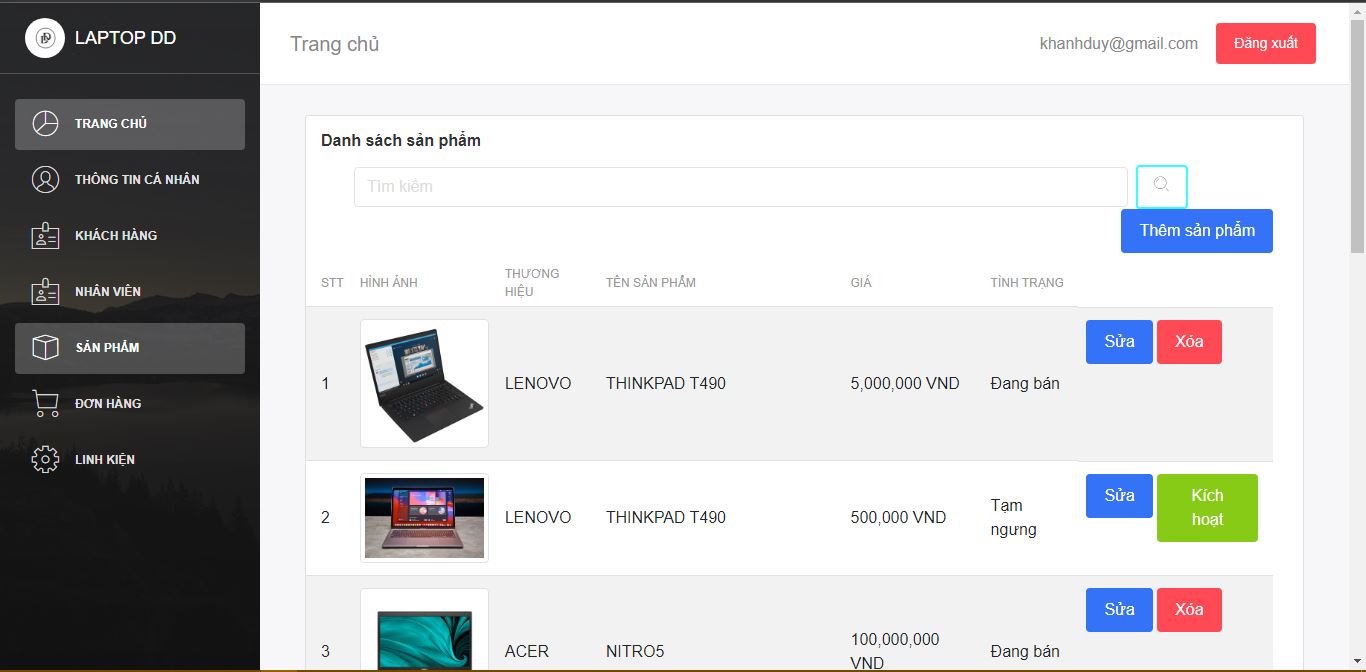
**Hình 6- 15**: Giao diện thêm nhân viên

### 6.2.5 Giao diện sửa nhân viên

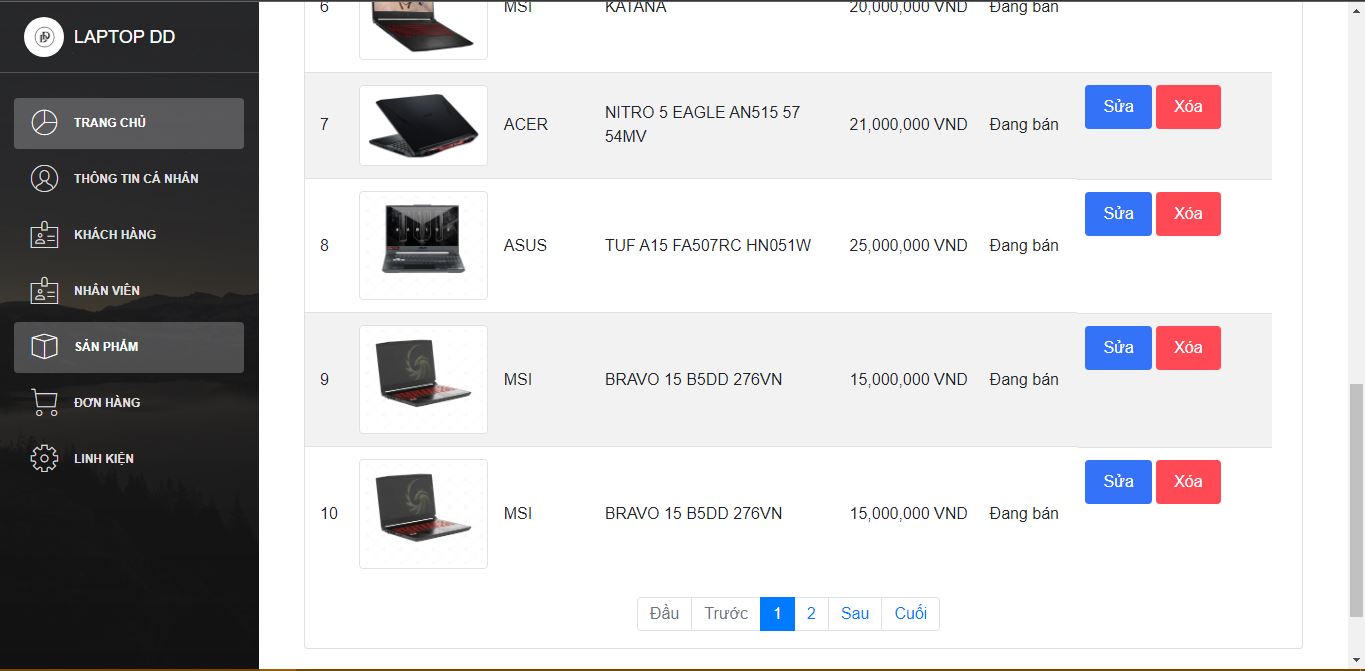


**Hình 6- 16**: Giao diện sửa nhân viên

### 6.2.6 Giao diện quản lý sản phẩm

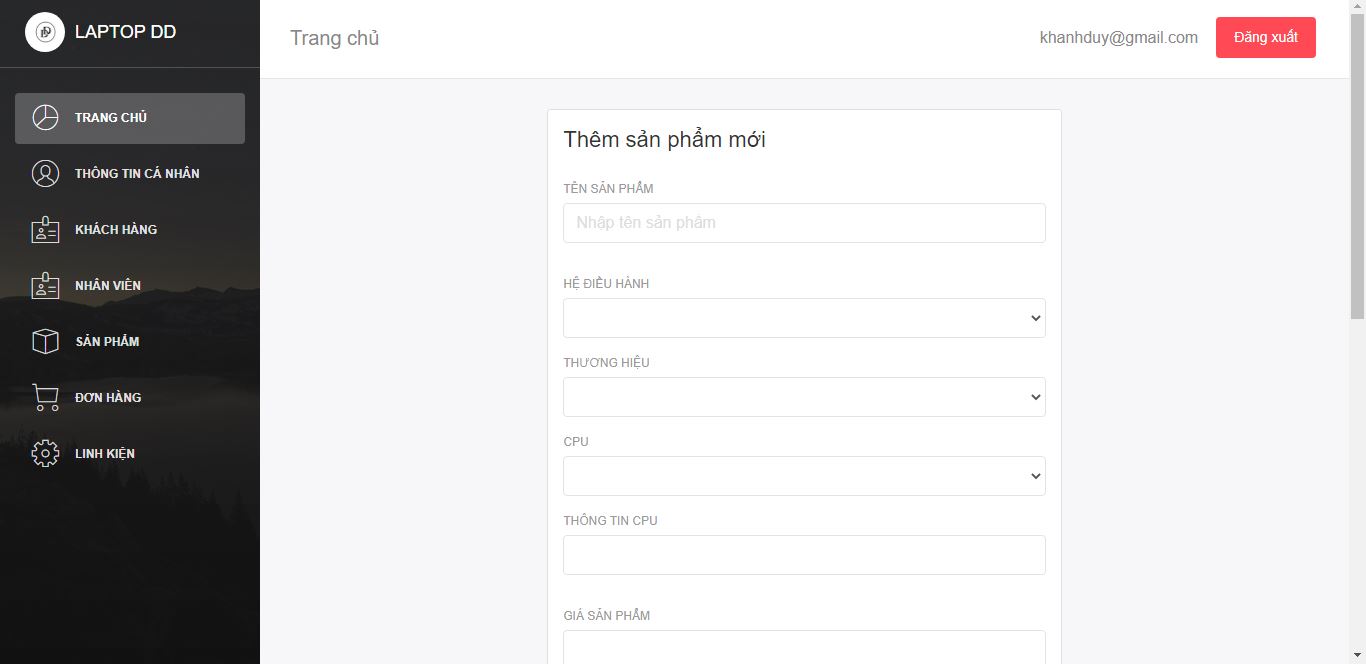


**Hình 6- 17**: Giao diện quản lý sản phẩm

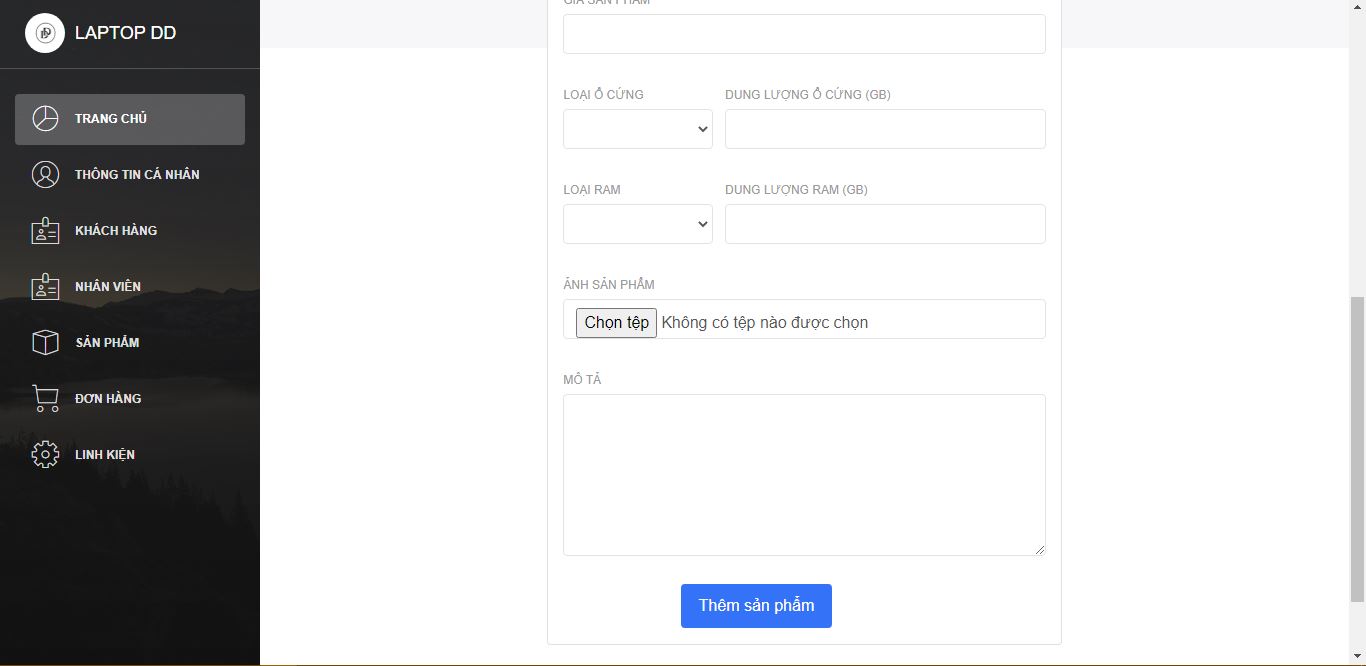


**Hình 6- 18**: Giao diện quản lý sản phẩm

### 6.2.7 Giao diện thêm sản phẩm

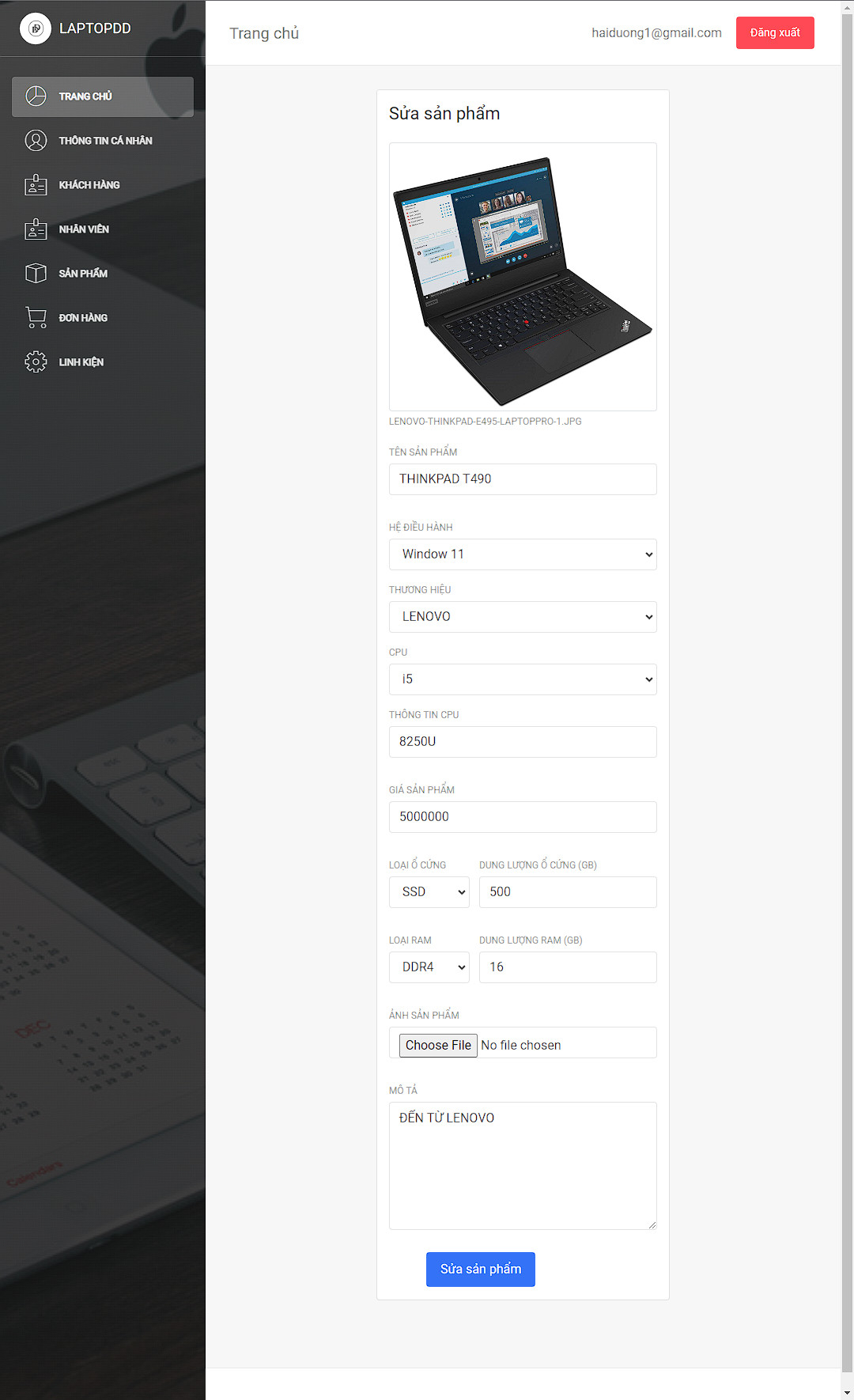


**Hình 6- 19**: Giao diện thêm sản phẩm



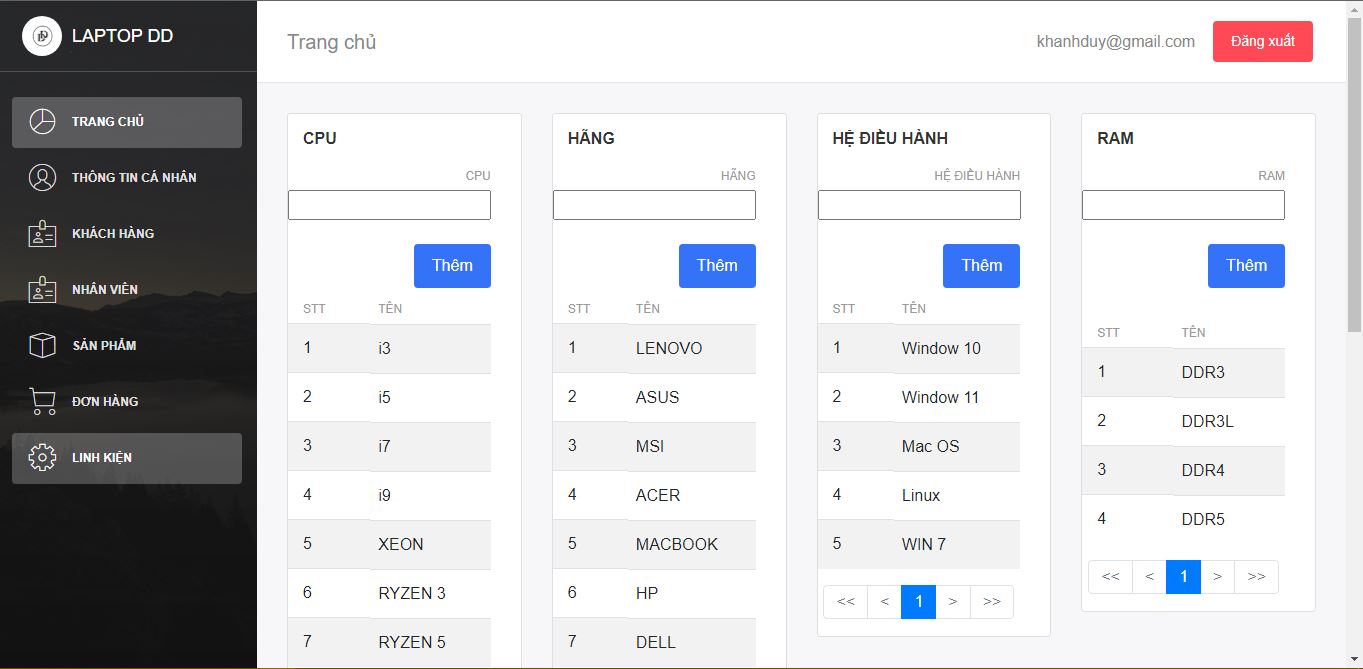
**Hình 6- 20**: Giao diện thêm sản phẩm

### 6.2.8 Giao diện sửa sản phẩm



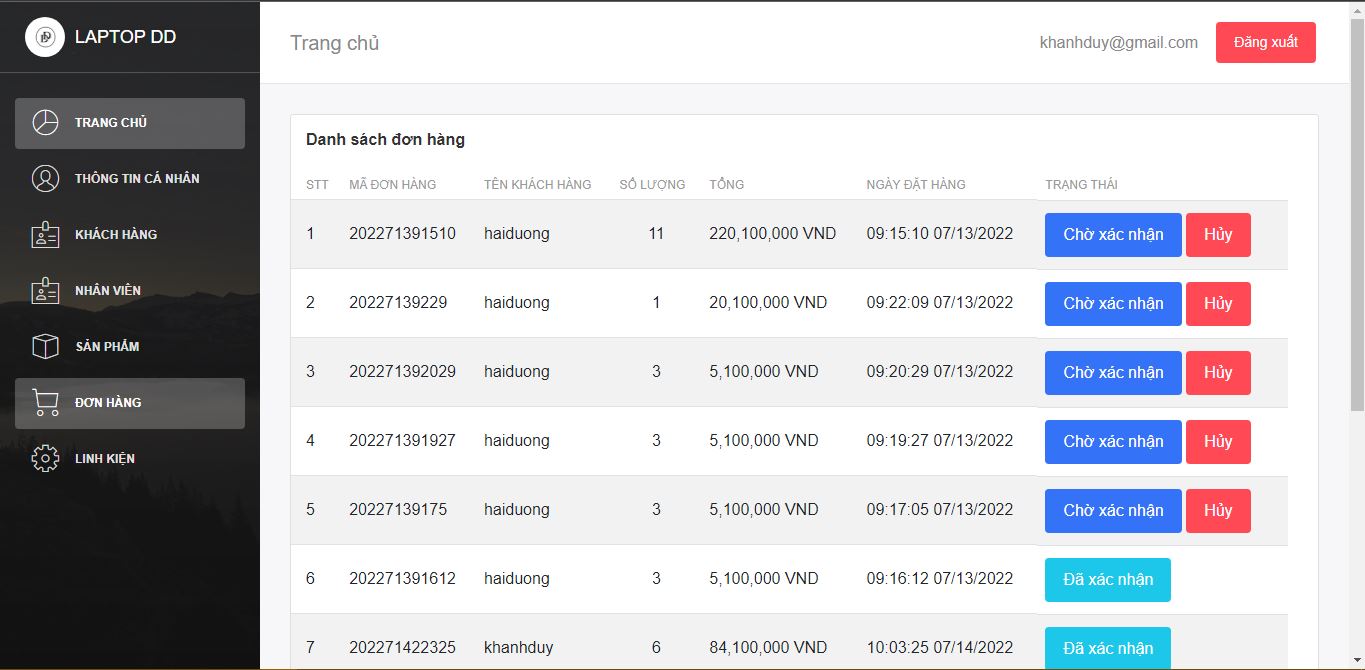
**Hình 6- 21**: Giao diện sửa sản phẩm

### 6.2.9 Giao diện quản lý linh kiện

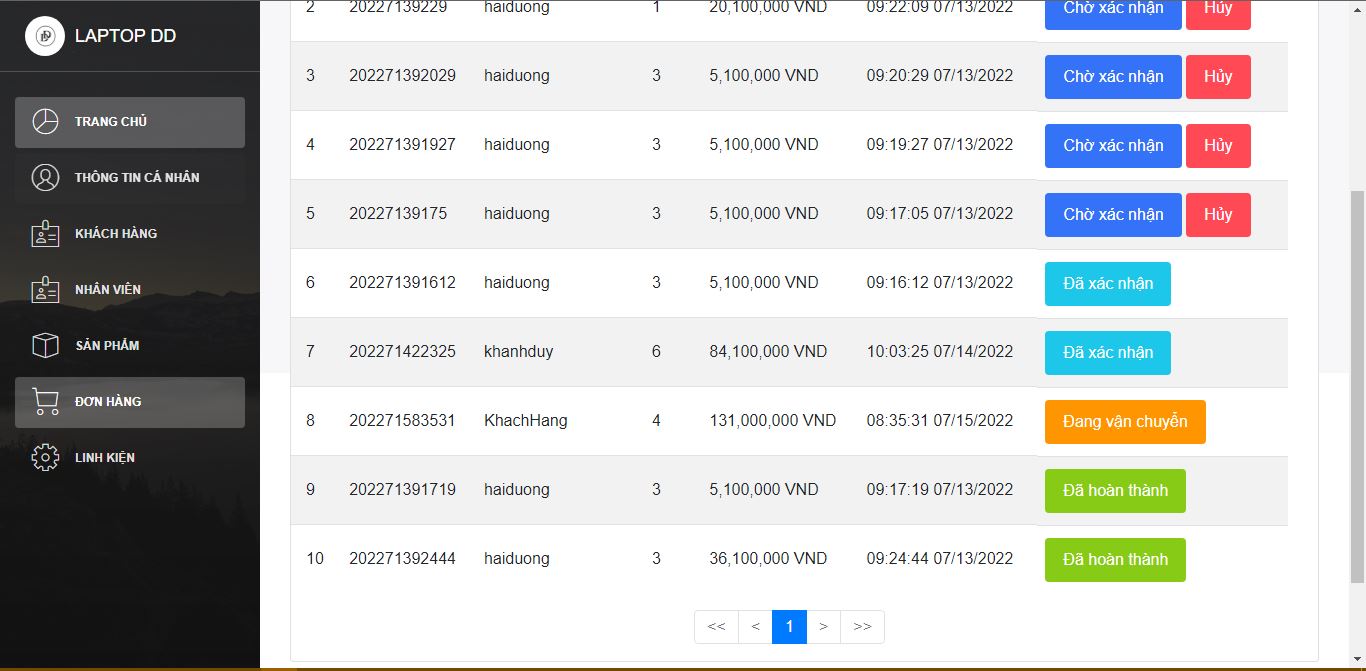


**Hình 6- 22**: Giao diện quản lý linh kiện

### 6.2.10 Giao diện quản lý đơn hàng

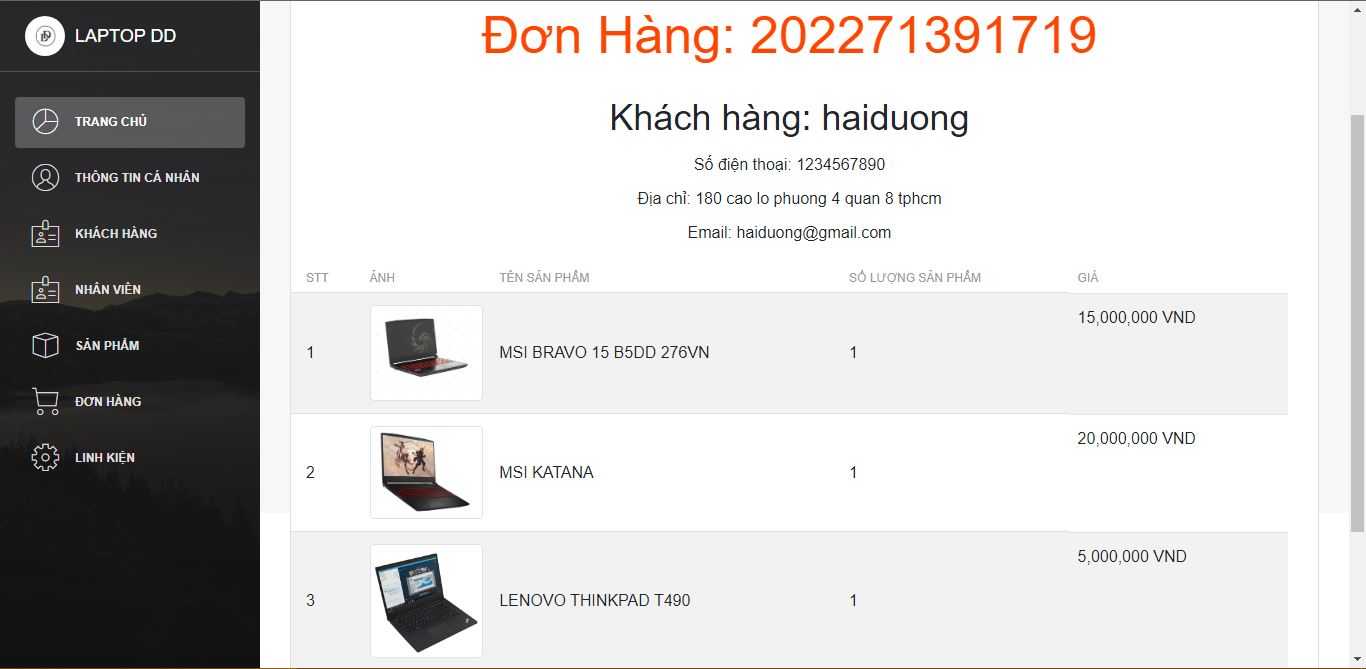


**Hình 6- 23**: Giao diện quản lý đơn hàng



**Hình 6- 24**: Giao diện quản lý đơn hàng

### 6.2.11 Giao diện chi tiết đơn hàng



**Hình 6- 25**: Giao diện chi tiết đơn hàng

# **Chương 7. TỔNG KẾT**

## **7.1 KẾT LUẬN ĐỀ TÀI**

Đề tài “Tìm hiểu VueJS và NodeJS. Áp dụng xây dựng website bán laptop” bao gồm nhiều ý nghĩa về mặt công nghệ cũng như tính năng. Để tài đã khái quát hoàn thành được các chức năng cần thiết của hệ thống. Website đã có thể cung cấp được một trang mua hàng trực tuyến với nhiều chức năng.

Tuy nhiên, trong thời gian ngắn ngoài công tác tìm hiểu ngôn ngữ mới, khảo sát nghiệp vụ đào tạo cũng như quá trình tìm tòi, học hỏi, tham khảo các tài liệu còn tương đối ngắn gọn, cùng với đó lượng người sử dụng ngôn ngữ mới này tương đối ít nên việc thảo luận trao đổi gặp nhiều khó khăn. Do đó chương trình không tránh khỏi những thiếu sót.

## **7.2 CHỨC NĂNG**

### 7.2.1 Chức năng hoàn thiện

* Khách hàng:
* Đăng ký
* Đăng nhập
* Trang chủ sản phẩm
* Xem chi tiết
* Thông tin cá nhân
* Xem giỏ hàng
* Mua hàng
* Xóa sản phẩm trong giỏ hàng
* Thay đổi số lượng sản phẩm trong giỏ hàng
* Thanh toán
* Quản lý:
* Đăng nhập
* Quản lý danh sách khách hàng
* Thêm khách hàng mới khi cần thiết
* Xóa – Kích hoạt khách hàng
* Quản lý danh sách nhân viên
* Thêm nhân viên
* Sửa nhân viên
* Xóa – Kích hoạt nhân viên
* Quản lý danh sách sản phẩm
* Thêm sản phẩm
* Sửa sản phẩm
* Xóa – Kích hoạt sản phẩm
* Quản lý danh sách đơn hàng

### 7.2.2 Chức năng hạn chế

* Khách hàng:
* Chưa đăng nhập bằng liên kết facebook, email
* Chưa có chức năng đánh giá, bình luận
* Chưa chat trực tiếp với hệ thống
* Quản lý:
* Chưa có chức năng khuyến mãi
* Chưa có chức năng tin tức, bài viết

# **TÀI LIỆU THAM KHẢO**

[1] [Báo cáo tổng quan thị trường Digital Việt Nam Năm 2022](https://lptech.asia/kien-thuc/bao-cao-tong-quan-thi-truong-digital-viet-nam-nam-2022" \l ":~:text=Qua%20c%C3%A1c%20bi%E1%BB%83u%20%C4%91%E1%BB%93%20s%E1%BB%91,s%E1%BB%91%20v%C3%A0o%20%C4%91%E1%BA%A7u%20n%C4%83m%202022.)

< [https://lptech.asia/kien-thuc/bao-cao-tong-quan-thi-truong-digital-viet-nam-nam-2022#](https://lptech.asia/kien-thuc/bao-cao-tong-quan-thi-truong-digital-viet-nam-nam-2022)>.

[2] World Wide Web là gì? < <https://vi.wikipedia.org/wiki/World_Wide_Web>>.

[3] SPA - [Single Page Application](https://bizfly.vn/techblog/single-page-application-la-gi.html) < https://en.wikipedia.org/wiki/Single-page\_application>.

[4]

[Single-File Components - Vue.js (SFC) < https://vuejs.org/api/sfc-spec.html>.](Single-File Components - Vue.js (SFC) < https://vuejs.org/api/sfc-spec.html>.ada)

[5] Lịch sử ra đời Vuejs <https://vi.wikipedia.org/wiki/Vue.js>

[6] So sánh giữa Angular, React, Vuejs < <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>>.

[7] Lịch sử ra đời Nodejs < <https://wiki.tino.org/nodejs-la-gi/>>

[8] Giới thiệu C++ < <https://vi.wikipedia.org/wiki/C%2B%2B>>.

[9] Mô hình client- server < <https://codelearn.io/sharing/tim-hieu-ve-mo-hinh-client-server>>.

[10] Giới thiệu component <<https://vuejs.org/guide/essentials/component-basics.html>>.

[11] Hook vòng đời <<https://allaravel.com/blog/tim-hieu-ve-vong-doi-thuc-the-trong-vue-js>.>

[12] Giới thiệu props <https://vi.vuejs.org/v2/guide/components-props.html>.

[13] Truyền dữ liệu qua props < https://hoc.tv/hoc-vuejs/props-trong-vuejs-2277>.

[14] Giới thiệu Routing < <https://vuejs.org/guide/scaling-up/routing.html>>.

[15] Giới thiệu Event < https://vi.vuejs.org/v2/guide/events.html>