**LỜI CẢM ƠN**

Em xin chân thành cảm ơn sâu sắc đến với quý thầy cô đã giúp đỡ em thực hiện đồ án lần này. Đặc biệt là thầy Nguyễn Mai Huy đã giúp đỡ, hướng dẫn tận tình, chỉ bảo em trong suốt thời gian học tập và thực hiện đồ án.

Đồng thời em cũng xin trân trọng gửi lời cảm ơn, tình cảm quý báu mà các thầy cô trong khoa CNTT đã truyền đạt kĩ năng, kinh nghiệm cho em, và em xin gửi lời cảm ơn đến Trường ĐH Nguyễn Tất Thành đã tạo điều kiện để em được học tập trong môi trường tuyệt vời như này.

Và cuối cùng em cũng xin được gửi lời cảm ơn đến với gia đình đã luôn luôn động viên, ủng hộ, những người bạn đã gắn bó, chia sẻ rất nhiều những kinh nghiệm, kiến thức để đồ án lần này được thực hiện một cách hoàn chỉnh.

Tuy nhiên, thời gian và lượng kiến thức có hạn nên em không thể phát huy hết khả năng, ý tưởng vào đồ án lần này. Trong quá trình lập trình website, không thể tránh những sai xót, kính mong nhận được sự đóng góp và cảm thông từ quý thầy cô.

Em xin chân thành cảm ơn.

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của CNTT và những lợi ích từ việc ứng dụng nó đi vào đời sống, máy tính và Internet đã không còn là những khái niệm xa lạ đối với mọi người, mà nó dần trở thành công cụ hữu ích hỗ trợ con người học tập, giải trí và làm việc.

Trong nền kinh tế hiện nay, với xu thế toàn cầu hóa nền kinh tế thế giới, mọi mặt của đời sống ngày càng được nâng cao, đặc biệt nhu cầu mua sắm các loại mặt hàng của con người ngày càng tăng. Môi trường Internet tự do, rộng lớn là mãnh đất màu mỡ cho các doanh nghiệp có thể quảng bá sản phẩm của mình đến người tiêu dùng một cách nhanh chóng, ít tốn kém, giúp mở rộng hoạt động kinh doanh, tăng trưởng doanh thu.

Năm 2021, khi con người vẫn còn đang đối mặt với đại dịch Covid thì việc quảng bá các mặt hàng online đã được đẩy mạnh để đáp ứng nhu cầu mọi người, sự tiện lợi, an toàn trong mùa dịch mà không cần phải đến nơi đông người để mua sắm đã góp phần thay đổi đời sống một cách hiện đại hóa.

Với mục tiêu tìm hiểu cấu trúc, cách thức hoạt động và khái quát hóa về thương mại điện tử trên mạng Internet ở Việt Nam, em đã xây dựng và đưa vào thử nghiệm thực tế một website bán giày sneaker trực tuyến. Đây sẽ là bước đi đầu tiên của em trước khi tiến cận với công việc quản lý website của một doanh nghiệp thực sự.

# 

# CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

1.1. Đặt vấn đề:

Vào những năm gần đây, việc ứng dụng CNTT về quản lý web ngày càng phát triển mạnh mẽ, giúp cho các cửa hàng kiếm được lợi nhuận nhiều hơn nhờ vào các trang website bán hàng trực tuyến. Hiểu được yêu cầu đó, em quyết định chọn đề tài “Xây dựng website bán giày sneaker” làm mục tiêu chính cho đồ án lần này, với mong muốn kinh doanh các sản phẩm giày thể thao đến từ Nike, Adidas, Puma, v.v... Ngoài ra, với giao diện website thân thiện cùng với tông màu đơn giản sẽ tạo thiện cảm cho khách hàng, giúp khách hàng chọn lựa những sản phẩm mình thích mà không cần tốn nhiều thời gian ra cửa hàng. Đó là sự tiện lợi mà mọi khách hàng nào cũng muốn!

Một trong các nhà phát triển phần mềm hàng đầu thế giới về lĩnh vực này là Microsoft, họ đã cho ra đời mô hình ASP.NET MVC 5 hỗ trợ người phát triển các ứng dụng web trên nền .NET Framework trong đó có tích hợp công nghệ LINQ với Entity Framework hỗ trợ làm việc với các loại CSDL.

Do đó, em chọn đề tài lần này để ứng dụng công nghệ phát triển web ASP.NET MVC 5, thực hiện lập trình web đáp ứng theo chương trình đào tạo của Trường, đồng thời mong muốn góp phần thúc đẩy ứng dụng CNTT vào Website để cho việc bán hàng trực tuyến được hoàn thiện, nhận được sự hài lòng đến từ khách hàng.

**1.2. Mục đích nghiên cứu:**

* Học được những kiến thức mới.
* Hiểu được cách hoạt động của ASP.NET MVC5, tương tác giữa Model – View – Controller.
* Tìm hiểu được các công nghệ hỗ trợ: HTML, CSS, BOOTSTRAP, JQUERY, Web Service, …
* Xây dựng trang web bán hàng giao diện chuẩn.
* Tìm hiểu về ngôn ngữ và kiến thức thức tương tác: LINQ to SQL và Entity Framework.
* Cách điều khiển dữ liệu thông qua Controller.
* Hiểu được cách tích hợp Google Map API đánh dấu địa điểm của website lên bản đồ, chỉ dẫn vị trí của khách hàng đến cửa hàng của website, giúp cho việc tìm kiếm cửa hàng dễ dàng hơn.
* Hiểu rõ về quá trình thanh toán trực tuyến, sự hỗ trợ từ các công nghệ để lưu trữ dữ liệu mua hàng của người dùng vào database.

1.3. Mục tiêu đề tài:

Là một website bán giày sneaker thì việc tạo ra giao diện đẹp mắt là điều quan trọng nhất, cách trình bày sản phẩm cũng như chi tiết sản phẩm, giá cả hợp lý sẽ gây ấn tượng tốt với khách hàng. Cả khách hàng và nhà quản lý sẽ cảm thấy thoải mái khi sử dụng một trang web đẹp và dễ truy cập. Với tiêu chí dễ dàng sử dụng dễ dàng chỉnh sửa, về phía người dùng có thể tham quan trang web, thoải mái mua sắm, tìm kiếm sản phẩm cần mua hoặc thêm vào danh sách yêu thích, về phía người quản trị có thể thêm, xóa, chỉnh sửa sản phẩm và theo dõi các hóa đơn, dữ liệu từ khách hàng.

* Mục tiêu đối với người dùng (User):
* Tham quan trang web.
* Tìm kiếm và lựa chọn sản phẩm cần mua.
* Sắp xếp sản phẩm theo danh mục cần tìm.
* Xem chi tiết 1 sản phẩm.
* Tạo tài khoản mua hàng trực tuyến.
* Thực hiện việc đặt hàng trực tuyến.
* Gửi liên hệ qua các địa chỉ liên hệ trên trang web.
* Xem vị trí cửa hàng trên Google Map thông qua trang liên hệ.
* Mục tiêu đối với người quản trị (Admin):
* Quản lý tất cả sản phẩm một cách dễ dàng.
* Thêm sản phẩm, loại sản phẩm, thương hiệu sản phẩm.
* Kiểm tra, theo dõi đơn đặt hàng, dữ liệu khách hàng.
* Theo dõi, phản hồi ý kiến khách hàng.

# CHƯƠNG II: TÌM HIỂU VỀ CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về ASP.NET:

2.1.1. Giới thiệu:

Đầu năm 2002, Microsoft giới thiệu một kĩ thuật lập trình Web khá mới mẻ với tên gọi ban đầu là ASP+, nhưng sau đổi thành ASP.Net. Với ASP.Net không những không cần đòi hỏi bạn phải biết các tag HTML, Web Design, mà nó còn hỗ trợ mạnh lập trình hướng đối tượng trong quá trình xây dựng và ứng dụng Web. ASP.Net là kỹ thuật lập trình và phát triển ứng dụng web ở phía Sever dựa trên nền tảng của .Net Framework. Hầu hết, những người mới đến với lập trình web đều bắt đầu tìm hiểu những kĩ thuật ở phía Client như: HTML, Java Script, CSS. Khi web browser yêu cầu một trang web (trang web sử dụng kỹ thuật Client), web server tìm trang web mà Client yêu cầu, sau đó gửi về Client. Client nhận kết quả trả về từ server và hiển thị lên màn hình. ASP.Net sử dụng lập trình ở phía server thì hoàn toàn khác, mã lệnh ở phía server sẽ được biên dịch và thi hành tại web server. Sau khi được server đọc, biên dịch và thi hành kết quả, kết quả tự động được chuyển sang HTML/JavaScript/CSS và trả về cho Client. Tất cả các xử lý lệnh ASP.Net đều được thực hiện tại server và do đó, gọi là kỹ thuật lập trình ở server. ASP.Net được Microsoft phát triển qua nhiều phiên bản từ ASP.Net 1.0, 1.1, 2.0, và gần đây nhất là phiên bản ASP.Net 5.

2.2. Mô hình lập trình web MVC của ASP.NET:

2.2.1. Khái niệm:

Mô hình MVC (Model-View-Controller) là kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm (đặc biết đối với phát triển ứng dụng web). Nó giúp cho tổ chức ứng dụng phân bố thành 3 phần khác nhau: Model, View, Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

Model: là các thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất CSDL, đối tượng mô tả dữ liệu như các Class, hàm xử lý… Model được giao nhiệm vụ cung cấp dữ liệu cho CSDL và lưu dữ liệu vào các kho chứa dữ liệu. Tất cả các nghiệp vụ logic được thực thi ở Model. Dữ liệu vào từ người dùng sẽ thông qua View để kiểm tra ở Model trước khí lưu vào cơ sở dữ liệu. Việc truy xuất, xác nhận và dữ liệu là một phần của Model.

View: View hiển thị các thông tin cho người dùng của ứng dụng và được giao nhiệm vụ cho việc nhận các dữ liệu vào từ người dùng, gởi đi các yêu cầu người dùng đến bộ điều khiển (Controller), sau đó là nhận lại các phản hồi từ bộ điều khiển và hiển thị kết quả cho người dùng. Các trang HTML, JS, các thư viện và các file nguồn là một phần của View.

Controller: Controller là tầng trung gian giữa Model và View. Controller được giao nhiệm vụ nhận các yêu cầu từ người dùng (phía máy khách). Một yêu cầu được nhận từ máy khách được thực hiện bởi một chức năng logic thích hợp từ thành phần Model và sau đó sinh ra các kết quả cho người dùng và được thành phần View hiển thị.

Một ứng dụng web được phát triển theo mô hình MVC, có nguyên lý hoạt động xử lý yêu cầu của người dùng như sau:

Khi có yêu cầu phát sinh từ người dùng (phía Client), yêu cầu này sẽ được gửi đến phía Server, tại đây Controller sẽ tiếp nhận yêu cầu để xử lý.

Bên trong Controller chứa nhiều phương thức (action), mỗi action tương ứng với 1 yêu cầu cụ thể từ người dùng. Controller sẽ lựa chọn action phù hợp với yêu cầu để xử lý. Trong quá trình xử lý, Controller tương tác với Model để có được dữ liệu mà người dùng mong muốn.

Sau khi Controller làm việc với Model để có được dữ liệu theo yêu cầu, Controller sẽ gửi về cho View thành phần Data Model, và nhiệm vụ của View sẽ chuyển Data Model nhận được thành dữ liệu ứng dụng và gửi trả về phía Client để hiển thị kết quả yêu cầu.

Diagram

Description automatically generated

**Hình 1 Mô hình MVC**

2.2.2. Ưu và nhược điểm khi sử dụng mô hình ASP.NET MVC để phát triển web:

### Ưu điểm

* Dễ dàng bảo trì mã, dễ dàng mở rộng và phát triển.
* Các thành phần của mô hình này có thể được kiểm tra hoàn toàn độc lập với người dùng.
* Dễ dàng hỗ trợ cho các khách hàng mới.
* Có thể thực hiện song song việc phát triển các thành phần khác nhau.
* Đơn giản hóa bằng cách chia web thành ba phần: Model, View, Controller.
* Chỉ sử dụng mẫu Front Controller để xử lý các yêu cầu của web thông qua một bộ điều khiển duy nhất.
* Hỗ trợ tốt nhất cho việc phát triển web theo hướng thử nghiệm.
* MVC hoạt động tốt với các ứng dụng được hỗ trợ bởi giới lập trình web.
* Thân thiện với công cụ tìm kiếm.
* Tất cả các đối tượng độc lập với nhau, do vậy bạn có thể kiểm tra một cách hoàn toàn riêng biệt.

Không những vậy, MVC tương đối nhẹ và tiết kiệm diện tích băng thông bởi nó không cần sử dụng Viewstate. Điều đó giúp website hoạt động tốt và ổn định khi người dùng thực hiện quá nhiều thao tác tương tác như gửi hay nhận dữ liệu liên tục.

### Nhược điểm

* Khó đọc, thay đổi, kiểm tra đơn vị hoặc sử dụng lại mô hình này.
* Đôi lúc điều hướng khung có thể phức tạp vì nó giới thiệu các lớp trừu tượng mới, đòi hỏi người dùng phải thích ứng với các tiêu chí phân tách của MVC.
* Không hỗ trợ việc xác thực chính thức.
* Vừa làm tăng sự phức tạp vừa làm giảm hiệu quả của dữ liệu.
* Gây khó khăn khi sử dụng với giao diện người dùng hiện đại.
* Bắt buộc phải có nhiều lập trình viên để tiến hành lập trình song song.
* Cần có kiến ​​thức tổng hợp về công nghệ.
* Bảo trì nhiều mã trong Bộ điều khiển.

MVC được sử dụng trong các dự án lớn. Với các dự án nhỏ, việc áp dụng mô hình MVC không được thích hợp bởi nó khá cồng kềnh. Nó cũng tiêu tốn nhiều thời gian để phát triển cũng như trung chuyển dữ liệu. Tuy nhiên, so với những mô hình khác, MVC vẫn là sự lựa chọn hàng đầu cho [lập trình ứng dụng](https://teky.edu.vn/blog/lap-trinh-ung-dung-di-dong/) nói chung và cả lập trình web nói riêng.

2.2.3. Razor và HTML Helper:

Razor:

Razor là một cú pháp đánh dấu cho phép bạn nhúng mã dựa trên máy chủ (Visual Basic và C#) vào các trang web.

Mã dựa trên máy chủ có thể tạo nội dung web động một cách nhanh chóng. Khi một trang web được gọi, máy chủ thực thi mã dựa trên máy chủ bên trong trang trước nó trả lại trang cho trình duyệt. Bằng cách chạy trên máy chủ, mã có thể thực hiện các tác vụ phức tạp như truy cập CSDL.

Razor dựa trên ASP.NET và được thiết kế để tạo web, các ứng dụng, nhưng nó dễ sử dụng và dễ học hơn.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Hình 2 Ví dụ Razor**

HTML Helper:

HMTL Helper giúp chúng ta viết phần tử HTML trong Razor sử dụng cú pháp thân thiện với HTML. Nó nhìn như là HTML chuẩn vậy nhưng code được xử lý bởi Razor Engine trên server và nó tận dụng đươc các ưu điểm của việc xử lý phía server. Razor được tạo sử dụng HMTLHelper  nhìn như phần tử HTML thuần. Nó thao tác với các phần tử HTML như thêm mới phần tử HTML hay thay thế các nội dung có sẵn bằng một cái mới.

Table

Description automatically generated

**Hình 3 HTML Helper**

2.2.4 ViewData – ViewBag:

ViewBag:

ViewBag trong ASP.NET MVC được sử dụng để chuyển dữ liệu tạm thời (không có trong model) từ controller sang view.

ViewBag là một thuộc tính kiểu dynamic của lớp ControllerBase là lớp cơ sở của lớp Controller

Có thể gán bất kỳ số thuộc tính và giá trị nào cho ViewBag. Nếu gán cùng một tên thuộc tính nhiều lần cho ViewBag, thì nó sẽ chỉ xem xét giá trị cuối cùng được gán cho thuộc tính. ViewBag chỉ truyền dữ liệu từ Controller sang View, không chuyển dữ liệu ngược lại.Graphical user interface

Description automatically generated with medium confidence

**Hình 4 Chuyển dữ liệu từ Controller sang View bằng ViewBag**

Trong View Index.cshtml, bạn có thể truy cập thuộc tính ViewBag.TotalStudents để hiển thị tổng số sinh viên trong View.

Graphical user interface, text, application

Description automatically generated

**Hình 5 Ví dụ đầu ra từ ViewBag**

Hạn chế của ViewBag:

ViewBag không yêu cầu ép kiểu trong khi truy xuất các giá trị từ nó. Điều này có thể ném ra một ngoại lệ thời gian chạy nếu sử dụng sai phương thức trên giá trị.

ViewBag là kiểu dynamic và bỏ qua kiểm tra kiểu tại thời điểm biên dịch. Vì vậy, tên thuộc tính ViewBag phải khớp trong Controller và View khi viết nó theo cách thủ công.

ViewData:

Trong ASP.NET MVC, ViewData tương tự ViewBag, truyền dữ liệu từ Controller sang View. ViewData có kiểu Dictionary, trong khi ViewBag là kiểu Dynamic. Tuy nhiên, cả hai đều lưu trữ dữ liệu trong cùng một từ điển nội bộ.

ViewData là một từ điển, vì vậy nó chứa các cặp khóa – giá trị trong đó mỗi khóa là một chuỗi.

Text

Description automatically generated with medium confidence

**Hình 6 Minh họa cách chuyển dữ liệu từ Controller sang View bằng ViewData**

ViewData và ViewBag đều sử dụng cùng một từ điển trong nội bộ. Vì vậy, bạn không thể để ViewData Key trùng với tên thuộc tính của ViewBag, nếu không nó sẽ ném ra một ngoại lệ lúc thực thi.

2.2.5. Layout View:

Một ứng dụng có thể chứa một phần giao diện người dùng cụ thể được giữ nguyên trong toàn bộ ứng dụng, chẳng hạn như phần đầu trang (header), thanh điều hướng bên trái (left navigation bar), thanh bên phải (right bar) hoặc phần chân trang (footer).

Layout View cho phép bạn xác định một mẫu trang web chung, mẫu này có thể được kế thừa trong nhiều view để cung cấp giao diện nhất quán trong nhiều trang của một ứng dụng.

Layout View loại bỏ mã hóa trùng lặp và nâng cao tốc độ phát triển và bảo trì. Layout View cho giao diện người dùng mẫu ở trên sẽ chứa các phần đầu trang, menu bên trái, thanh bên phải và chân trang. Nó có một trình giữ chỗ (placeholder) cho phần trung tâm thay đổi động.

Chart, treemap chart

Description automatically generated

**Hình 7 Layout View**

Sự khác nhau giữa RenderBody – RenderSection:

|  |  |
| --- | --- |
| **RenderBody** | **RenderSection** |
| Phương thức RenderBody () phải có trong dạng xem bố cục. | Phương thức RenderSection () là tùy chọn. |
| RenderBody () hiển thị tất cả nội dung của dạng xem con không được bao bọc trong phần được đặt tên. | RenderSection () chỉ hiển thị một phần của khung nhìn con được bao bọc bên dưới phần được đặt tên. |
| Không cho phép nhiều phương thức RenderBody () trong một chế độ xem bố cục duy nhất. | Nhiều phương thức RenderSection () được phép trong một chế độ xem bố cục duy nhất. |
| Phương thức RenderBody () không bao gồm bất kỳ tham số nào. | Phương thức RenderSection () bao gồm tham số boolean "bắt buộc", làm cho phần này là tùy chọn hoặc bắt buộc. Nếu tham số bắt buộc là true, thì dạng xem con phải chứa phần này. |

2.2.6. Entity Framework:

Entity Framework là một ánh xạ quan hệ đối tượng (ORM), là một loại công cụ giúp đơn giản hóa ánh xạ giữa các đối tượng trong phần mềm của bạn với các bảng và cột cơ sở dữ liệu quan hệ.

* Entity Framework (EF) là một framework ORM mã nguồn mở độc lập với .NET Framework.
* Entity Framework đảm nhiệm việc tạo ra các kết nối cơ sở dữ liệu và thực thi các lệnh, cũng như lấy kết quả truy vấn và tự động ánh xạ các kết quả đó thành các đối tượng trong ứng dụng của bạn.
* Entity Framework giúp theo dõi các thay đổi của đối tượng và cập nhật các thay đổi đó trở lại cơ sở dữ liệu cho bạn.

Lý do nên sử dụng Entity Framework:

Entity Framework được thiết kế nhằm mục đích tăng năng suất của nhà phát triển bằng cách giảm các công việc dư thừa như là duy trì dữ liệu được sử dụng trong các ứng dụng.

* Entity Framework có thể tạo các câu lệnh truy vấn cần thiết để đọc hoặc ghi dữ liệu trong CSDL và thực thi chúng cho bạn.
* Nếu bạn cần truy vấn dữ liệu, bạn có thể sử dụng LINQ to Entities.
* Entity Framework sẽ thực hiện các truy vấn có liên quan trong CSDL và sau đó ánh xạ kết quả trả về thành các đối tượng để bạn làm việc trong ứng dụng của mình.

2.2.7. LINQ:

LINQ (truy vấn tích hợp ngôn ngữ) là cú pháp truy vấn thống nhất trong C# và VB.NET để truy xuất dữ liệu từ các nguồn và định dạng khác nhau.

Nó được tích hợp trong C# hoặc VB.NET, do đó loại bỏ sự không phù hợp giữa các ngôn ngữ lập trình và CSDL, cũng như cung cấp một giao diện truy vấn duy nhất cho các nguồn dữ liệu khác nhau, chẳng hạn như danh sách , ADO.Net DataSet, XML Docs, web service, MS SQL Server và các CSDL khác.

Ưu điểm LINQ:

* Ngôn ngữ quen thuộc: Nhà phát triển không phải học ngôn ngữ truy vấn mới cho từng loại nguồn dữ liệu hoặc định dạng dữ liệu.
* Viết ít code: Nó làm giảm số lượng mã được viết so với cách tiếp cận truyền thống hơn.
* Code dễ đọc: LINQ làm cho mã dễ đọc hơn để các nhà phát triển có thể dễ dàng hiểu và bảo trì nó.
* Cách truy vấn chuẩn hóa nhiều nguồn dữ liệu: Có thể sử dụng cùng một cú pháp LINQ để truy vấn nhiều nguồn dữ liệu.
* An toàn khi biên dịch của các truy vấn: Nó cung cấp kiểm tra kiểu của các đối tượng tại thời gian biên dịch.
* Hỗ trợ IntelliSense: LINQ cung cấp IntelliSense cho các danh sách kiểu generic.
* Định hình dữ liệu: Bạn có thể truy xuất dữ liệu theo các hình dạng khác nhau.

2.2.8 Generic Collections:

Đối với nhiều ứng dụng, bạn muốn tạo và quản lý các nhóm đối tượng liên quan. Có hai cách để nhóm các đối tượng: Bằng cách tạo mảng đối tượng (arrays of object) và tạo bộ sưu tập đối tượng (Collections of object).

Mảng hữu ích nhất để tạo và làm việc với một số lượng các đối tượng được đánh máy mạnh.

Bộ sưu tập cung cấp một cách linh hoạt hơn để làm việc với các nhóm các đối tượng. Không giống như mảng, nhóm đối tượng bạn làm việc có thể phát triển và thu nhỏ động theo nhu cầu của ứng dụng thay đổi.

Một bộ sưu tập chung nhận được tất cả lợi ích của thuốc chung. Nó không cần đấm bốc và mở hộp trong khi lưu trữ hoặc lấy các vật phẩm,… Hiệu suất được cải thiện.

Bạn cũng có thể tạo các loại và phương pháp chung tùy chỉnh để cung cấp các giải pháp tổng quát của riêng bạn và các mẫu thiết kế an toàn và hiệu quả.

Trong hầu hết các trường hợp, bạn nên sử dụng lớp Danh sách được cung cấp bởi thư viện lớp .NET Framework thay vì tạo của riêng bạn.

Tham số loại T được sử dụng ở một số vị trí nơi bê tông loại thông thường sẽ được sử dụng để chỉ ra loại mặt hàng được lưu trữ trong danh sách.

# CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Phân tích thiết kế hệ thống và ứng dụng

3.1.1. Mô tả chức năng ứng dụng:

Một website bán sản phẩm ngoài giao diện đẹp, thân thiện ra, các chức năng người dùng và người quản trị phải dễ dàng sử dụng, tương tác chặt chẽ với nhau, giúp cho tâm lý khách hàng được thoải mái khi mua sắm, lựa chọn sản phẩm mà mình yêu thích. Bộ phận quản lý trước khi thực hiện các nghiệp vụ cần đăng nhập tài khoản và mật khẩu để thực hiện các chức năng cho người quản lý sản phẩm.

Chức năng ứng dụng đối với người quản trị:

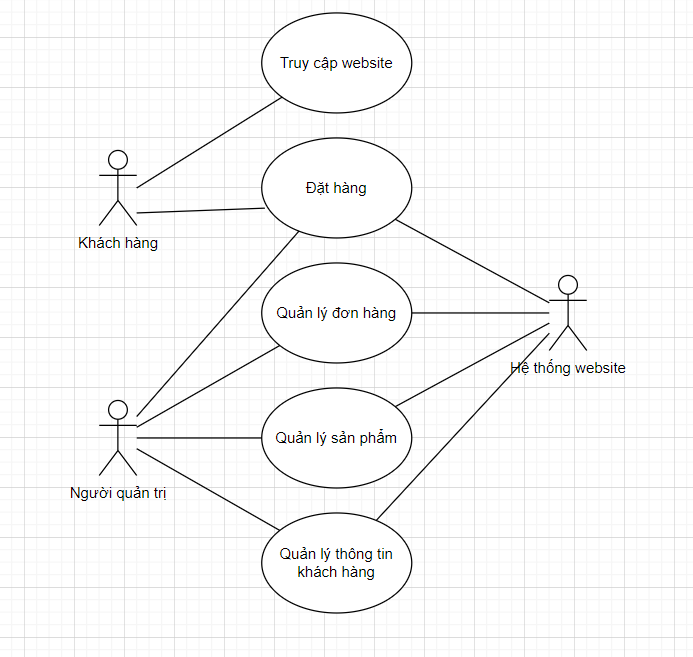
* Thêm, xóa, sửa, xem chi tiết sản phẩm.
* Thêm, xóa, sửa, xem chi tiết thương hiệu sản phẩm.
* Xem danh sách khách hàng.
* Xem danh sách các đơn đặt hàng.

Trang web còn phục vụ cho khách hàng với những chức năng như sau:

* Sắp xếp sản phẩm theo thương hiệu.
* Xem sản phẩm.
* Chọn và xem chi tiết sản phẩm.
* Đặt hàng trực tuyến.
* Xem chi tiết giỏ hàng.
* Cập nhật giỏ hàng (thêm, xóa các sản phẩm trong giỏ hàng).
* Đăng ký, đặt nhập tài khoản thành viên.

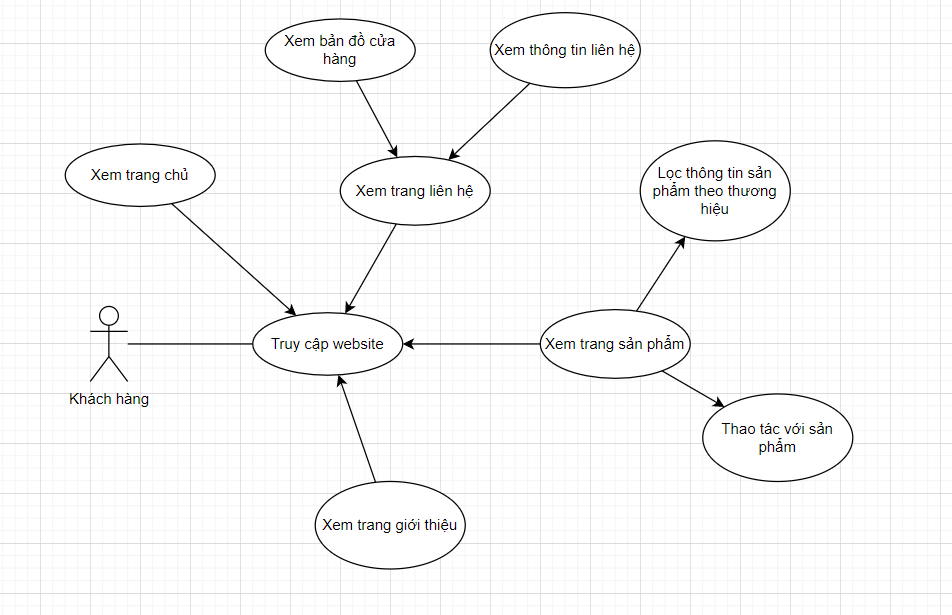
3.1.2. Sơ đồ usecase

Usecase tổng quát:



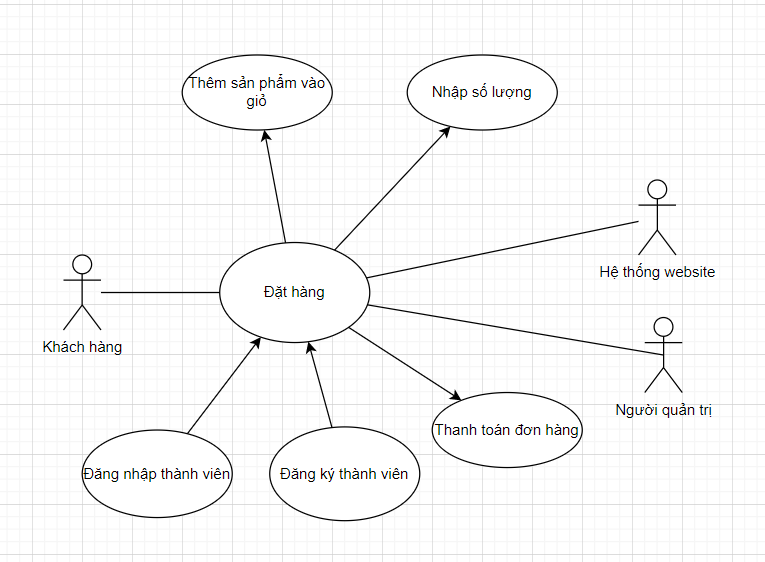
**Hình 8 Usecase tổng quát**

Usecase truy cập website



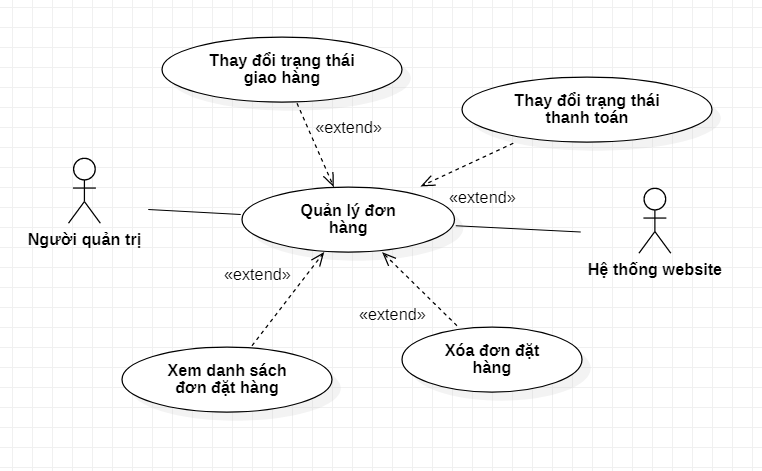
**Hình 9 Usecase truy cập website**

Usecase đặt hàng:



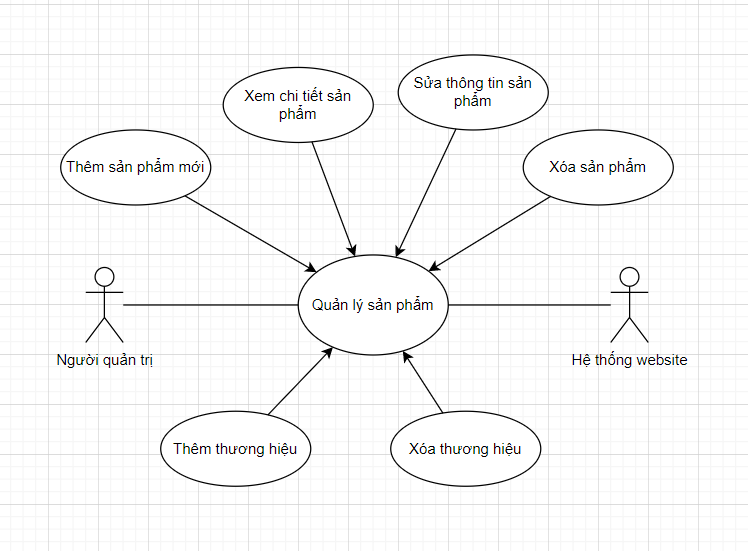
**Hình 10 Usecase truy cập website**

Usecase đơn hàng:



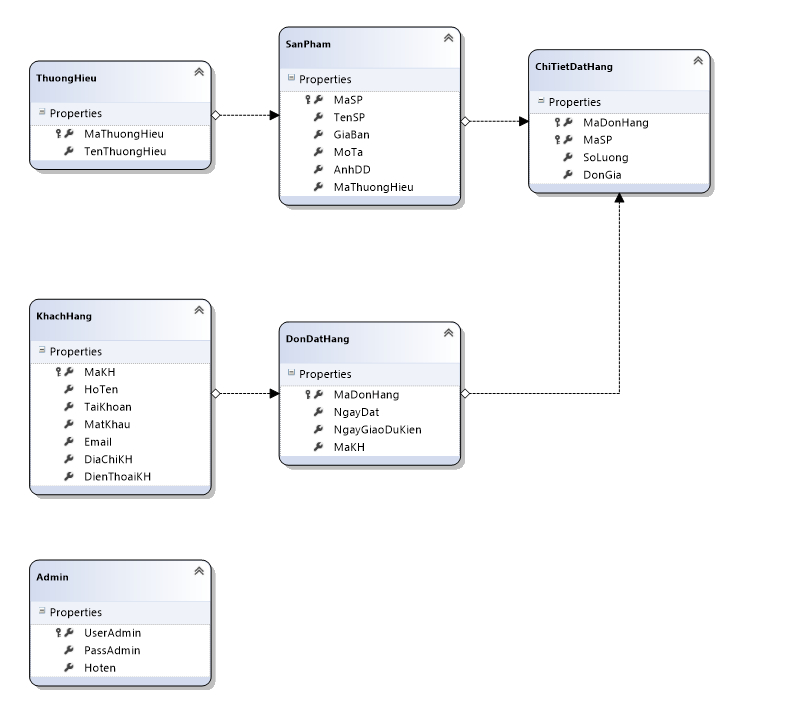
**Hình 11 Usecase truy cập website**

Usecase quản lý sản phẩm:



**Hình 12 Usecase truy cập website**

3.1.3. Mô hình khái niệm:



**Hình 13 Mô hình CSDL**

Các thành phầm chính trong mô hình khái niệm bao gồm:

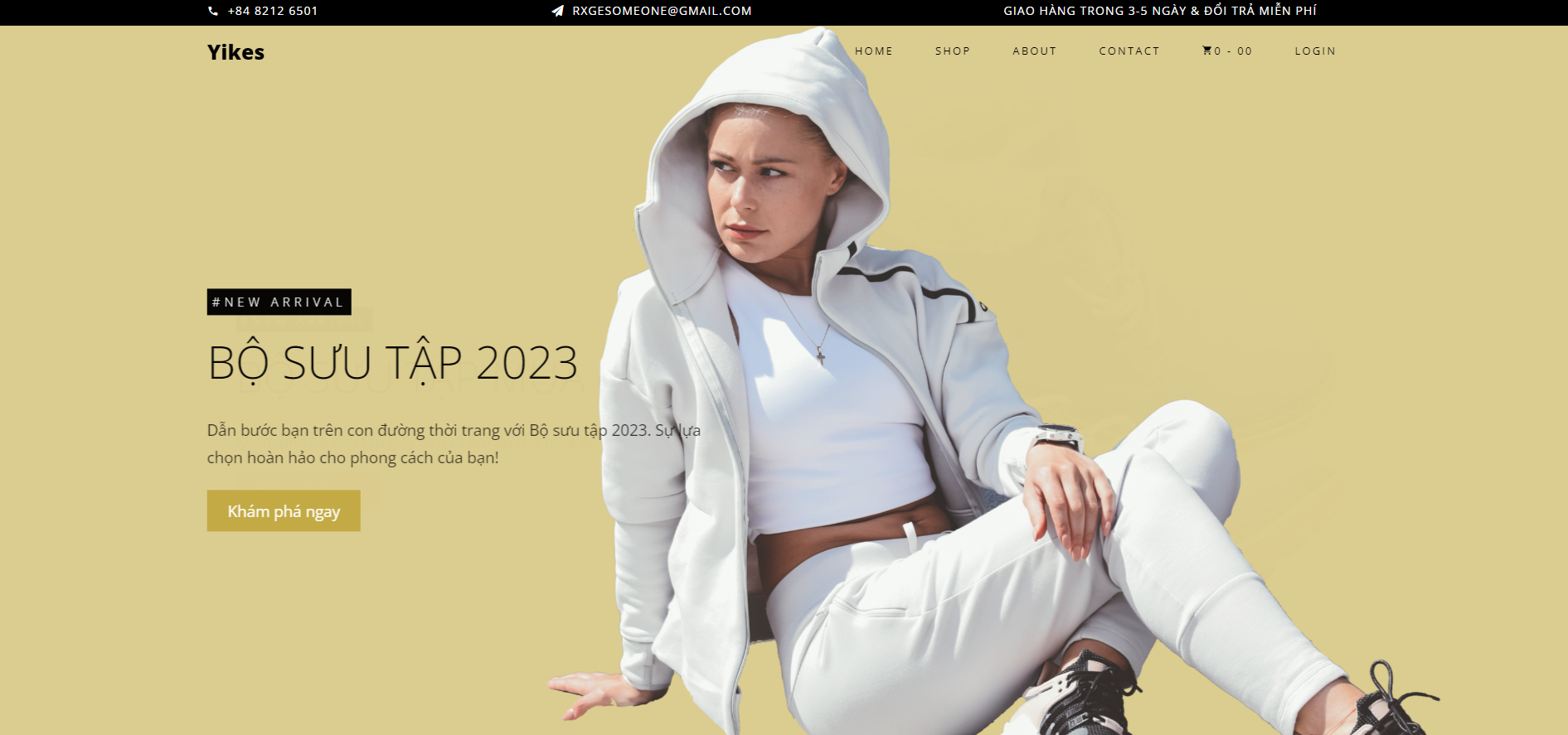
|  |  |
| --- | --- |
| **Tên thành phần** | **Thuộc tính** |
| ThuongHieu | MaThuongHieu (PK), TenThuongHieu |
| SanPham | MaSP (PK), TenSP, GiaBan, MoTa, AnhDD, MaThuongHieu (FK) |
| KhachHang | MaKH (PK), HoTen, TaiKhoan, MatKhau, Email, DiaChiKH, DienThoaiKH |
| DonDatHang | MaDonHang (PK), NgayDat, NgayGiaoDuKien, MaKH (FK) |
| ChiTietDatHang | MaDonHang (PK), MaSP (PK), SoLuong, DonGia |
| Admin | UserAdmin (PK), PassAdmin, Hoten |

# CHƯƠNG IV: THIẾT KẾ VÀ XÂY DỰNG WEBSITE BÁN GIÀY SNEAKER

4.1. Ý tưởng giao diện các trang Public và Admin

4.1.1. Giao diện các trang Public:

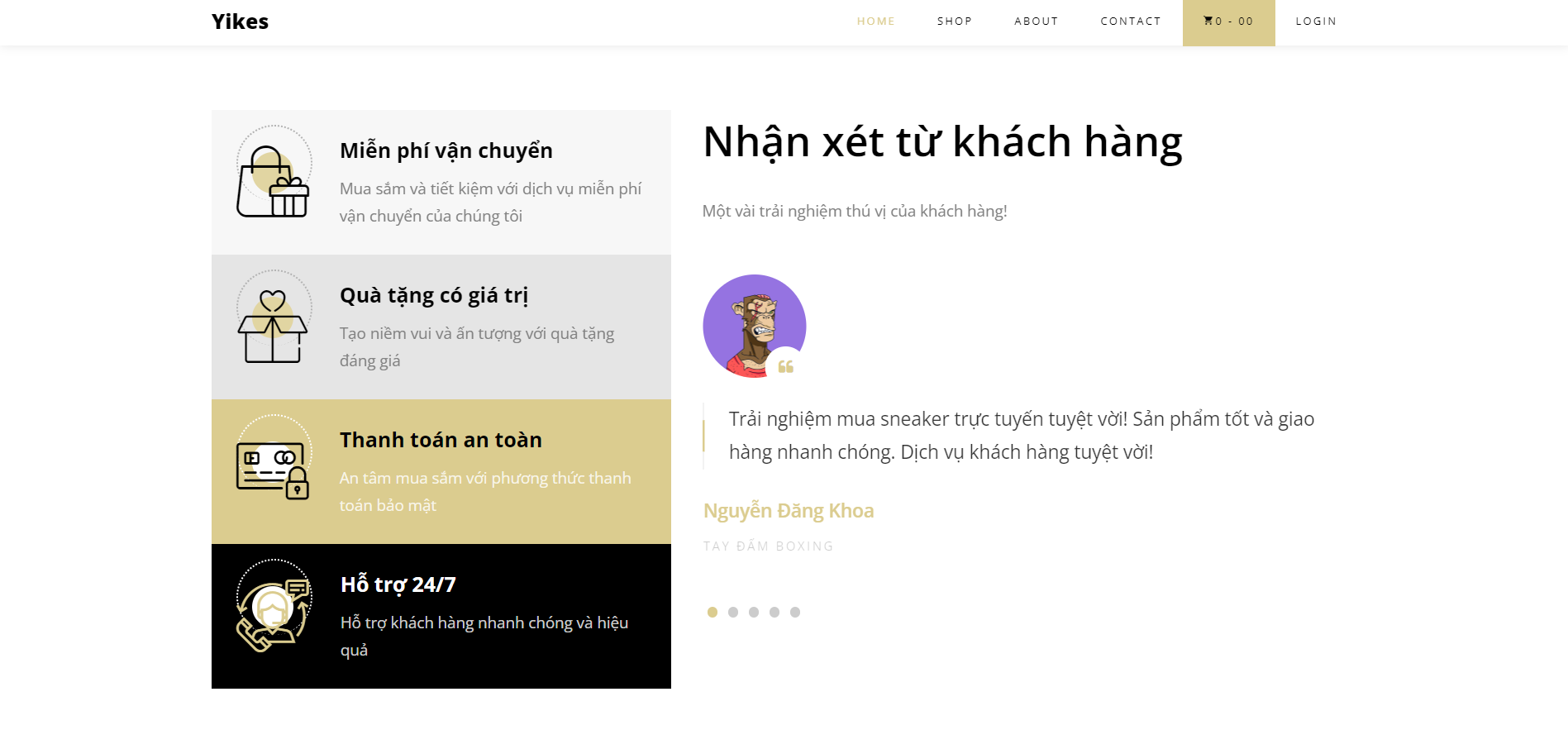
a. Trang chủ (Home):



**Hình 14 Giao diện trang chủ (1)**

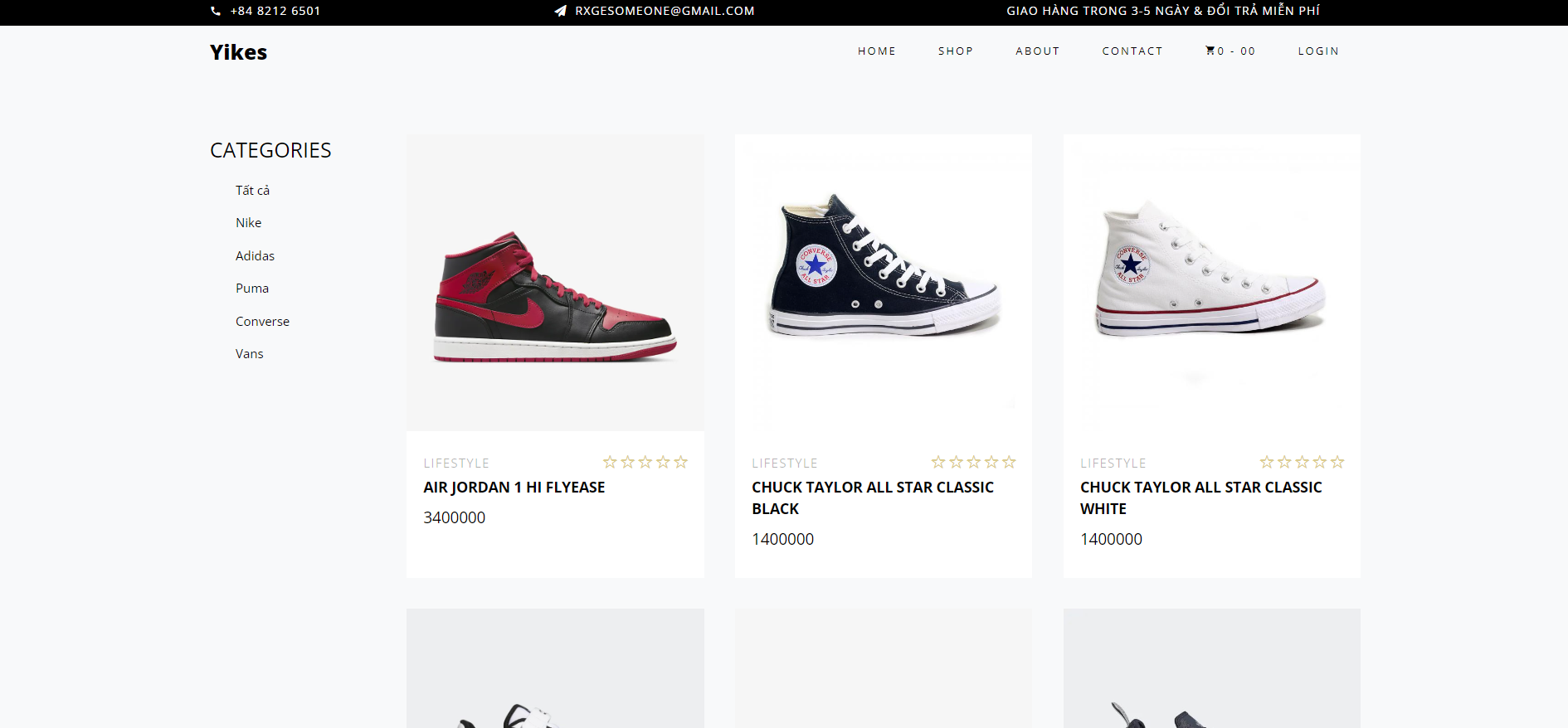


**Hình 15 Giao diện trang chủ (2)**

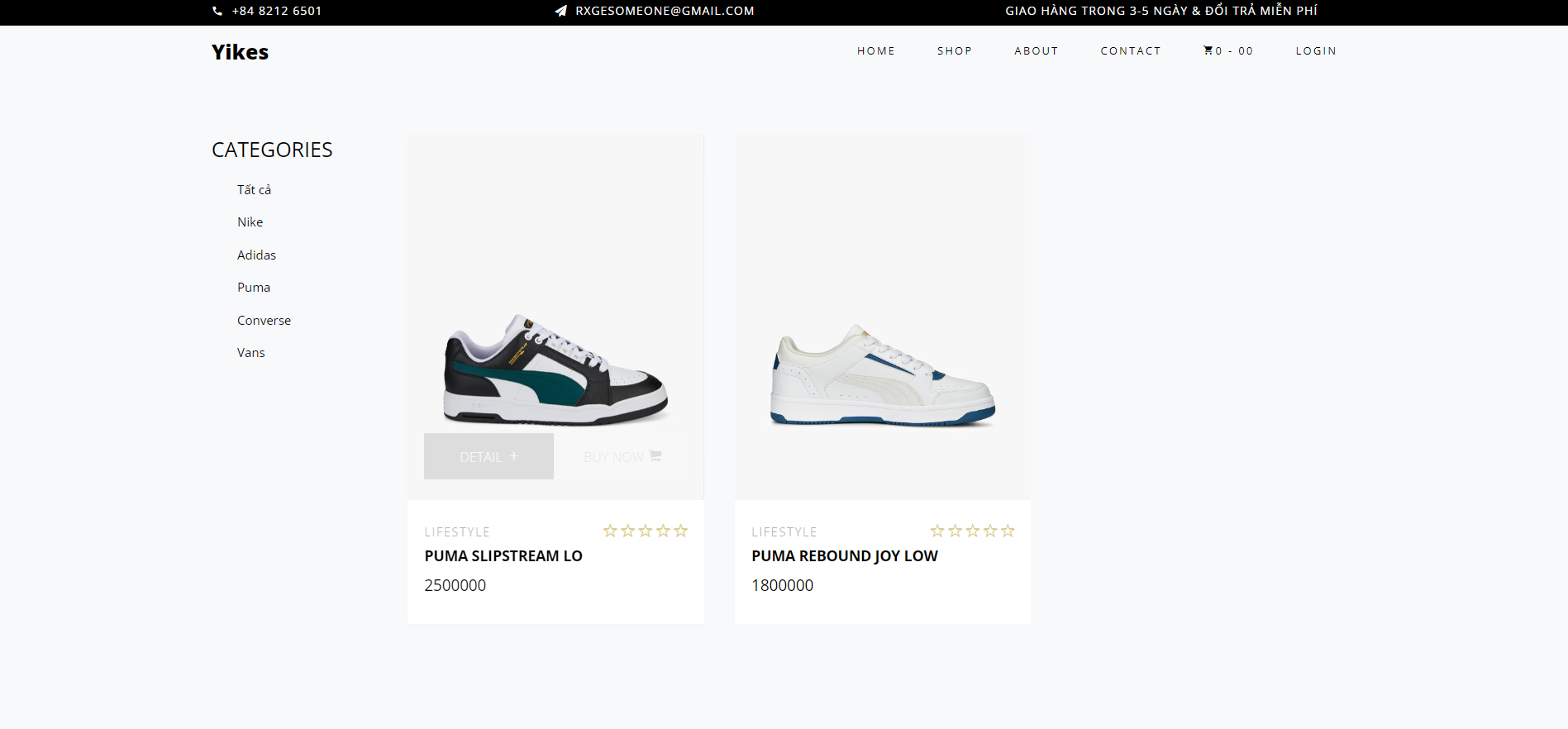


**Hình 16 Giao diện trang chủ (3)**

b. Trang sản phẩm (Sneaker)



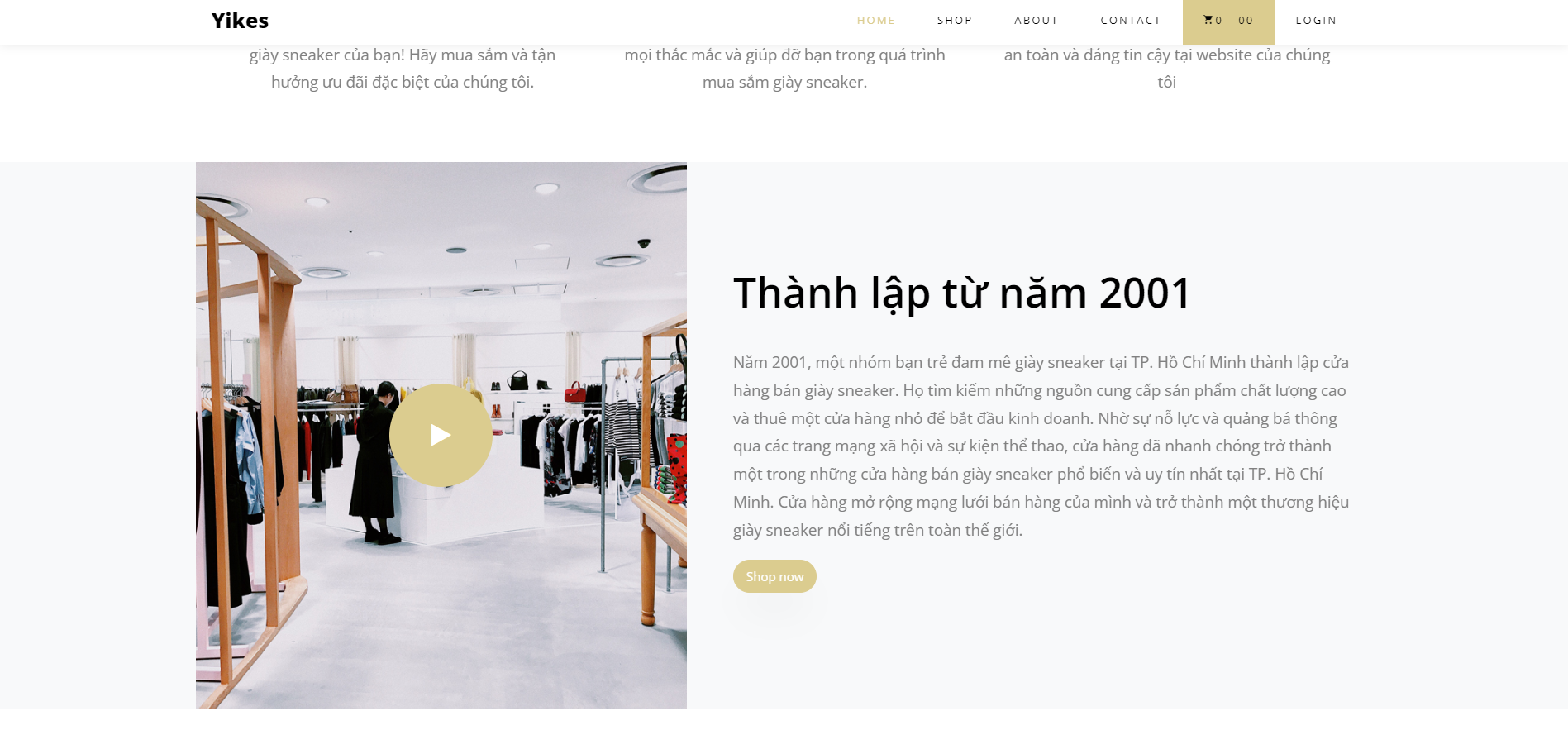
**Hình 17 Giao diện trang sản phẩm**



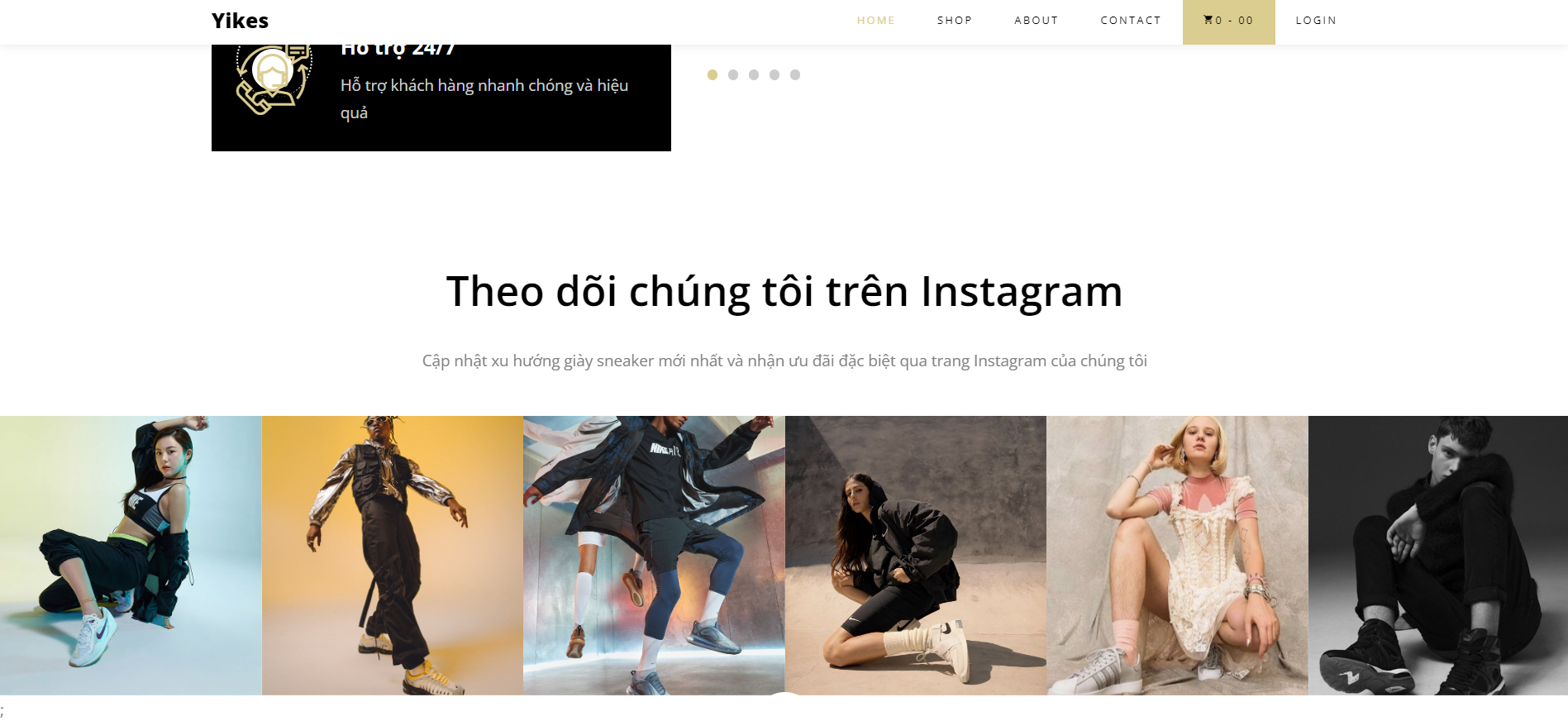
**Hình 18 Giao diện trang lọc danh sách theo thương hiệu**

c. Trang giới thiệu:

Giới thiệu cửa hàng, giá trị cốt lõi



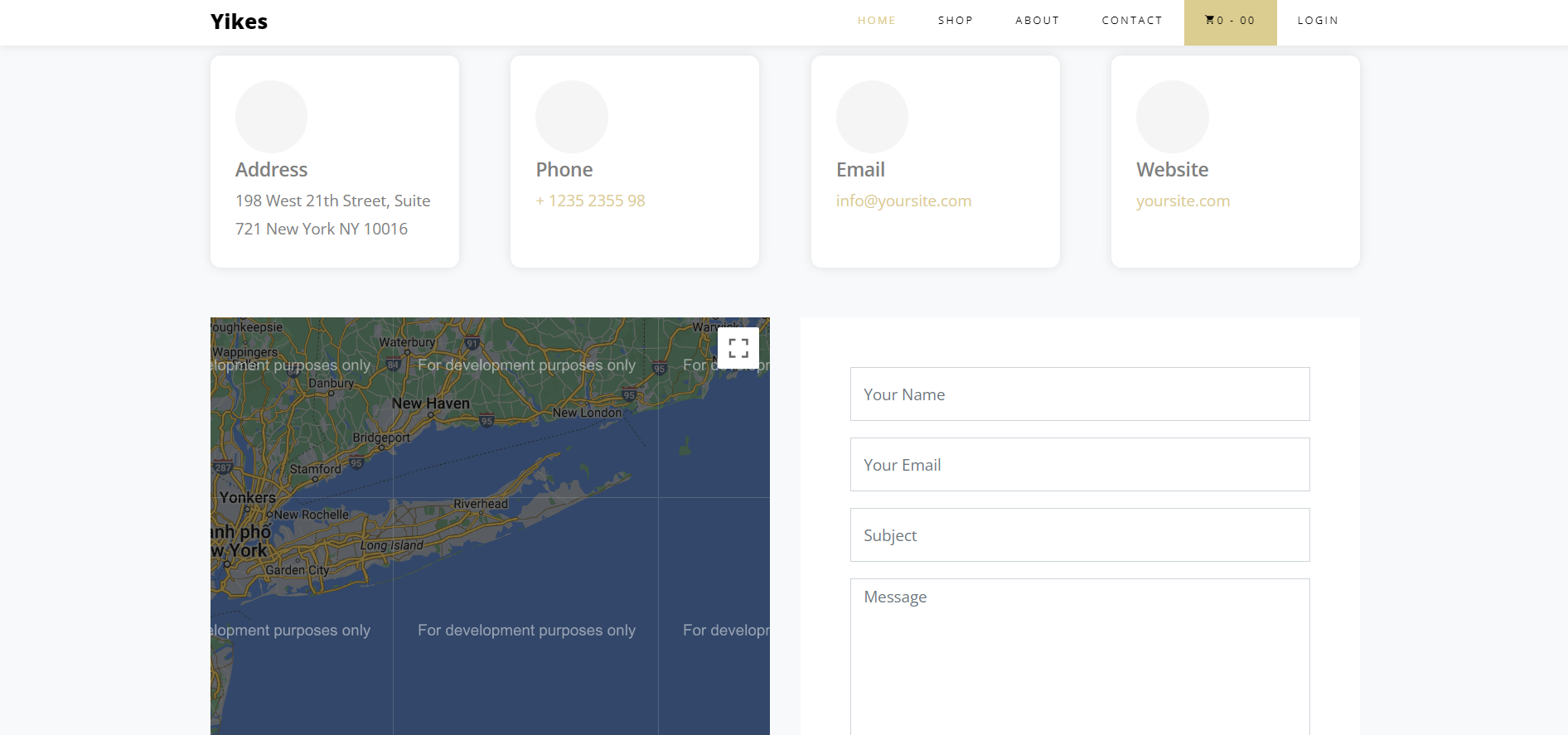
**Hình 19 Giao diện trang liên hệ (1)**



**Hình 20 Giao diện trang liên hệ (2)**

d. Trang liên hệ:

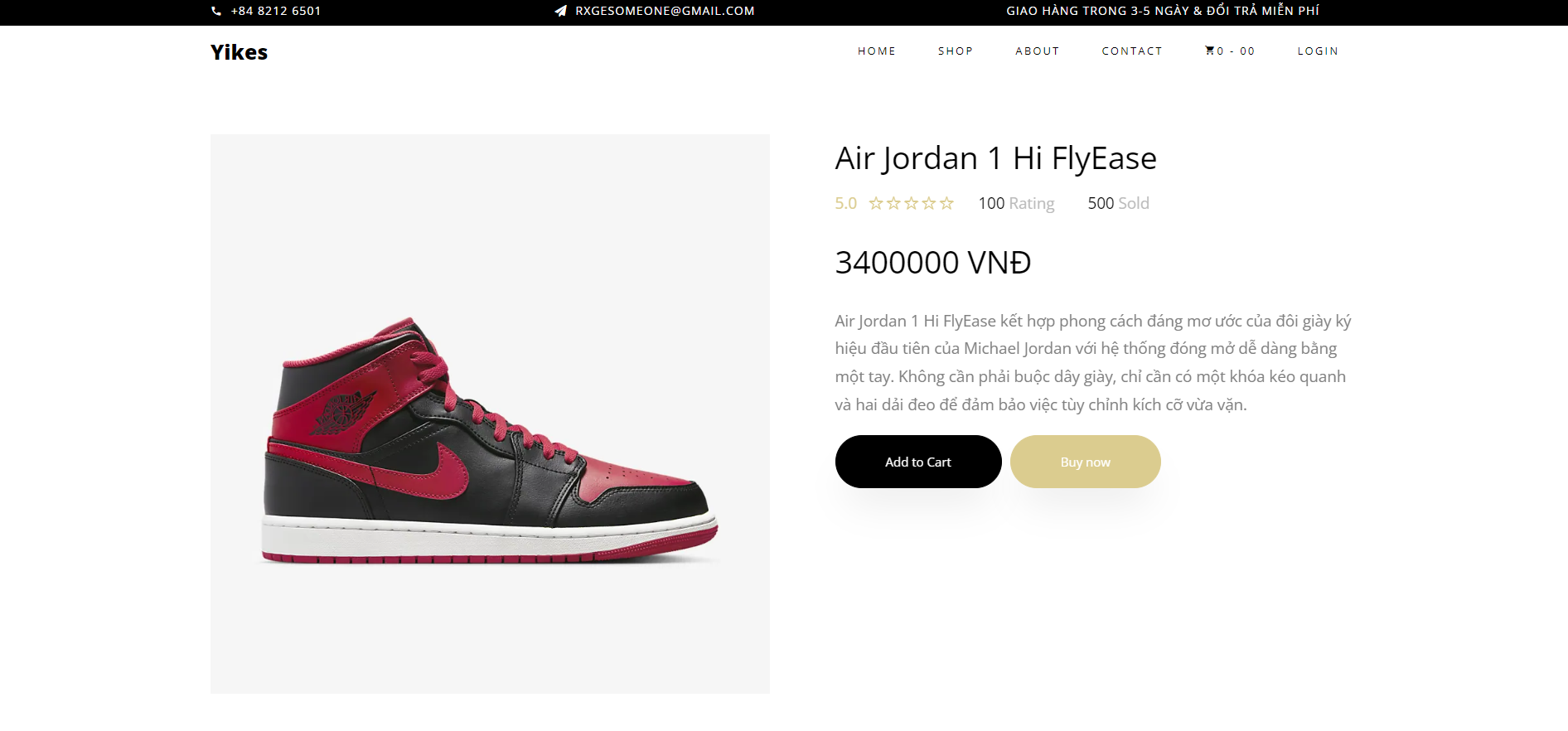
Cung cấp các thông tin cần thiết để khách hàng liên hệ, hỗ trợ tư vấn sản phẩm



**Hình 21 Trang liên hệ**

e. Trang xem chi tiết sản phẩm:

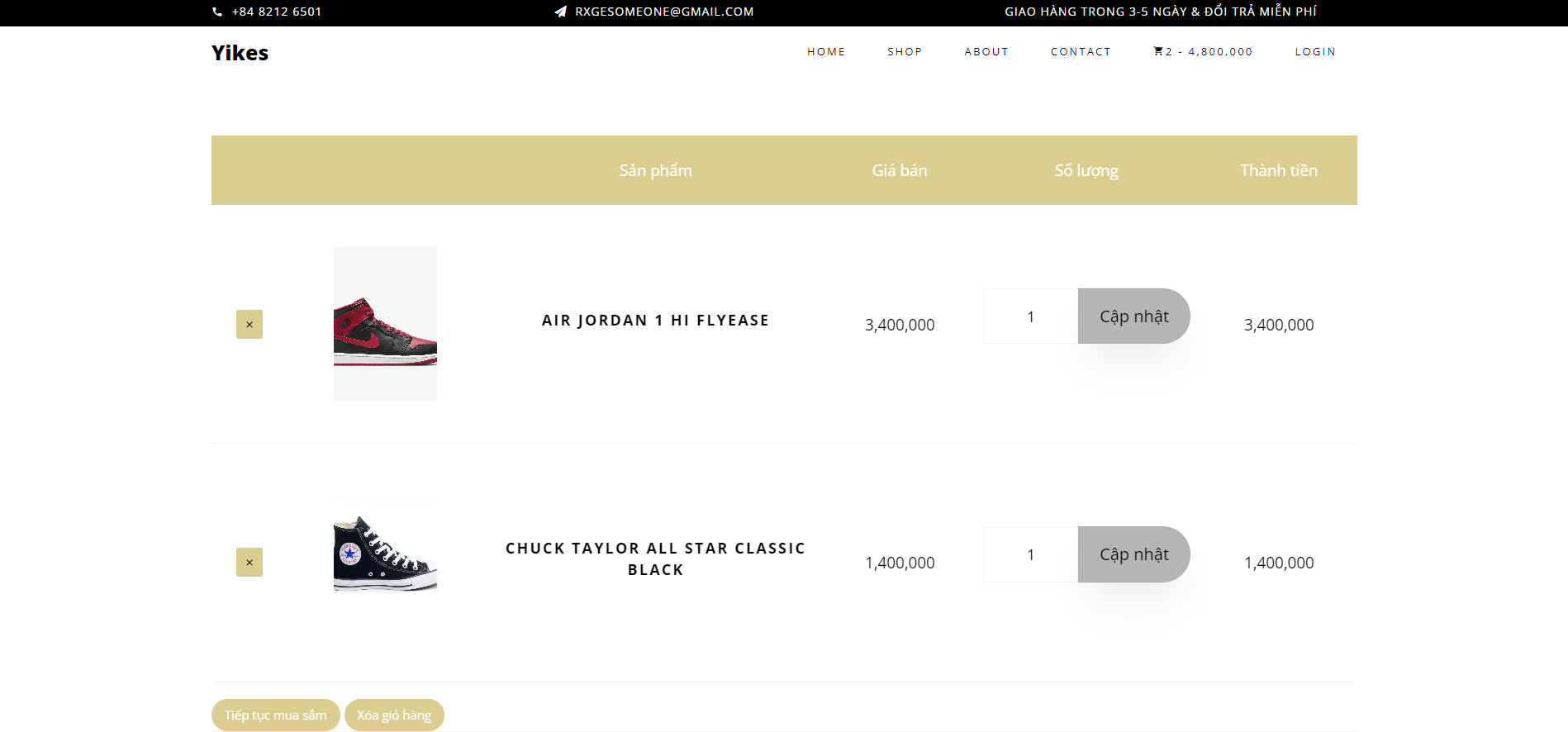
Muốn xem chi tiết một sản phẩm bất kì, bạn chỉ cần click vào Detail, một loạt thông tin chi tiết sản phẩm sẽ hiện ra.



**Hình 22 Trang giao diện chi tiết sản phẩm**

f. Trang giỏ hàng:

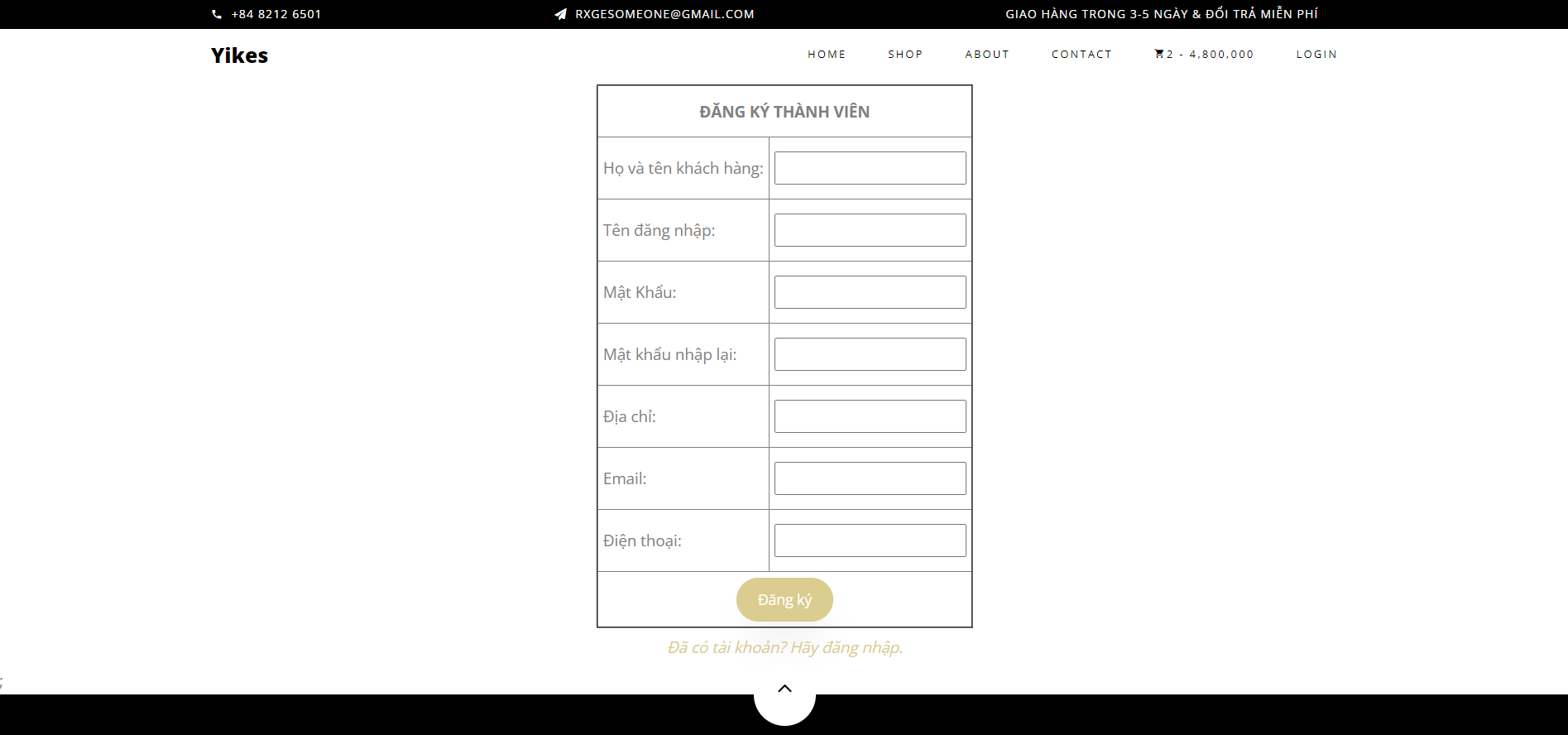
Sau khi thêm sản phẩm vào giỏ, bạn có thể điều chỉnh số lượng hoặc quay trở về để mua sản phẩm khác.



**Hình 23 Trang giao diện giỏ hàng**

g. Trang đăng ký tài khoản:

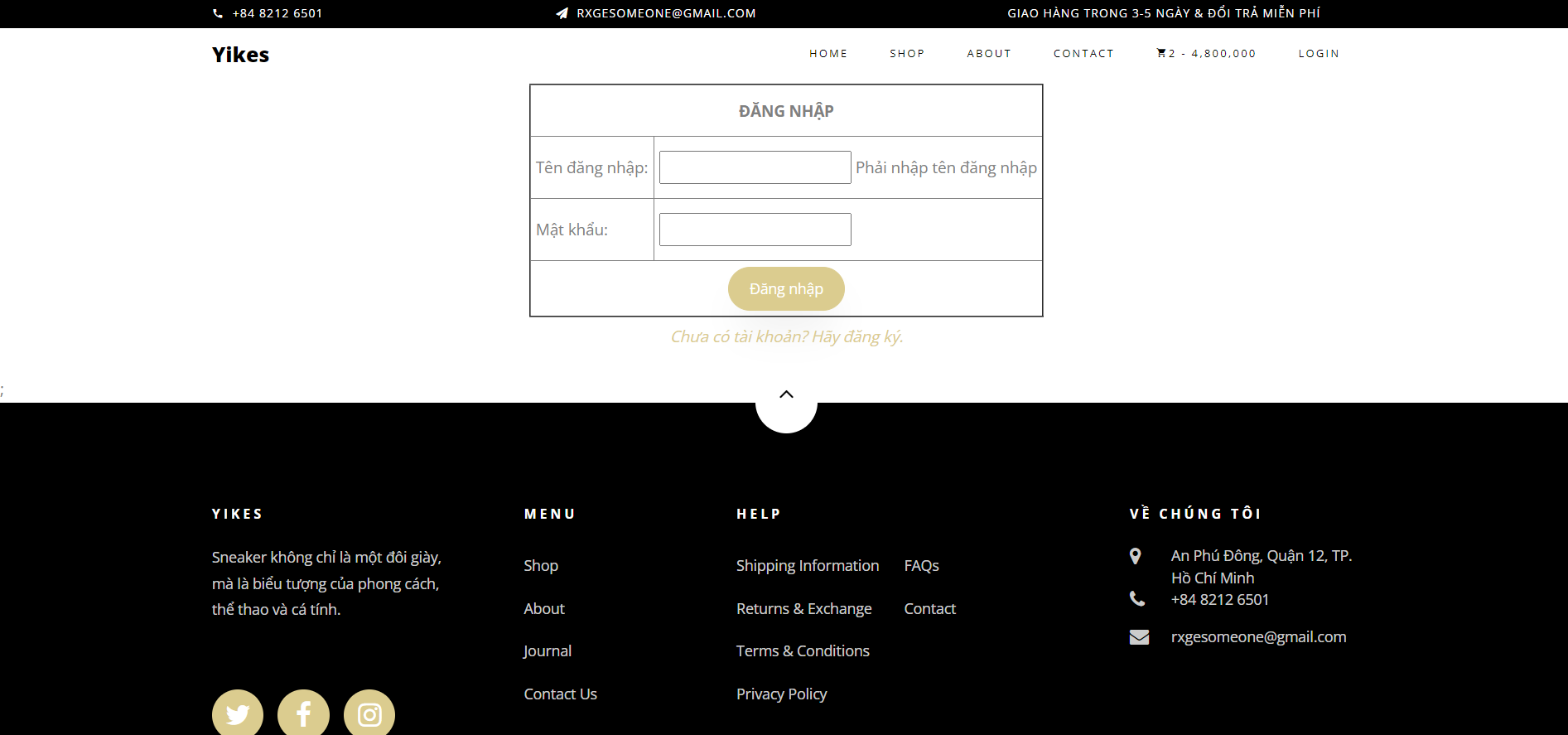
Khi bạn chọn tiếp tục mua hàng ở trang giỏ hàng, trang web sẽ chuyển hướng đến trang đăng kí tài khoản đặt hàng.



**Hình 24 Trang đăng ký khách hàng**

h. Trang đăng nhập tài khoản.

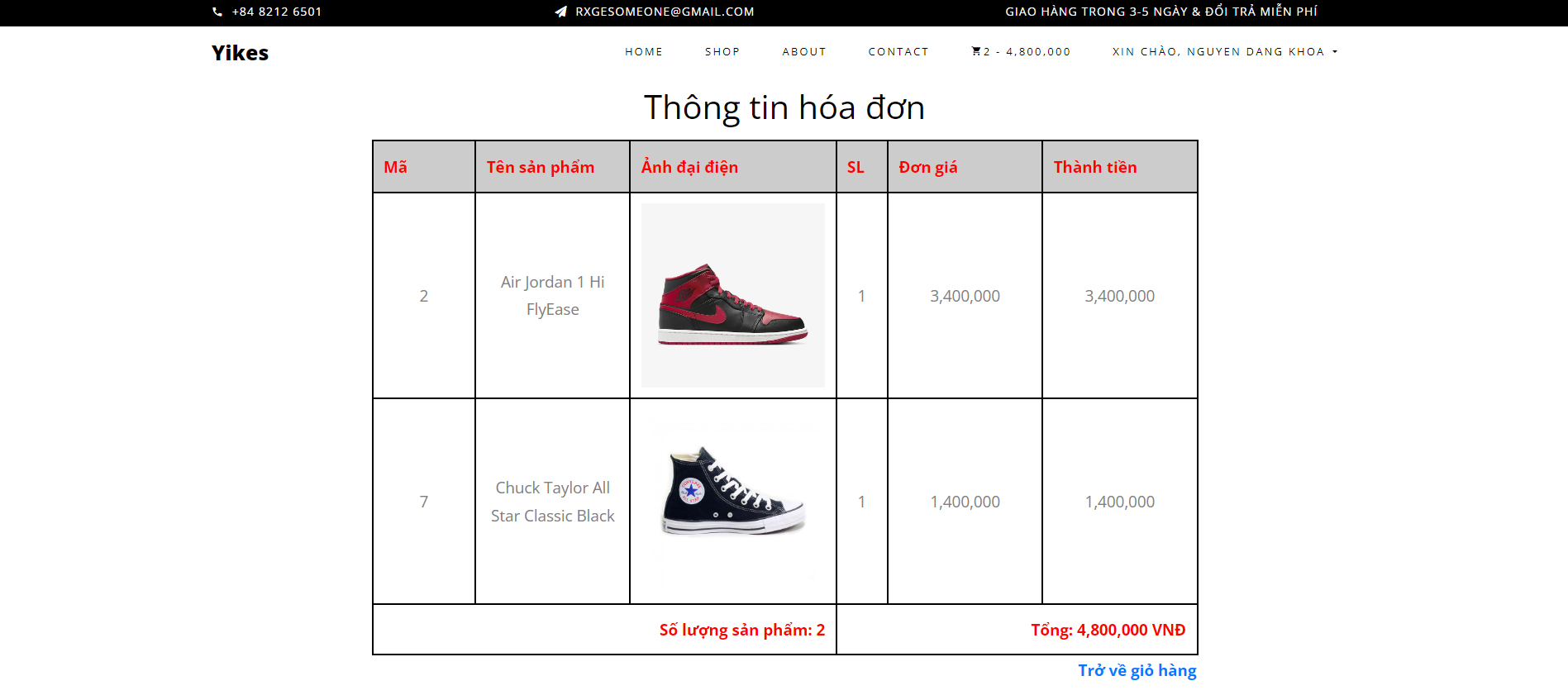
Sau khi đăng ký tài khoản thành công, người dùng sẽ tự chuyển hướng sang trang đăng nhập



**Hình 25 Trang đăng nhập khách hàng**

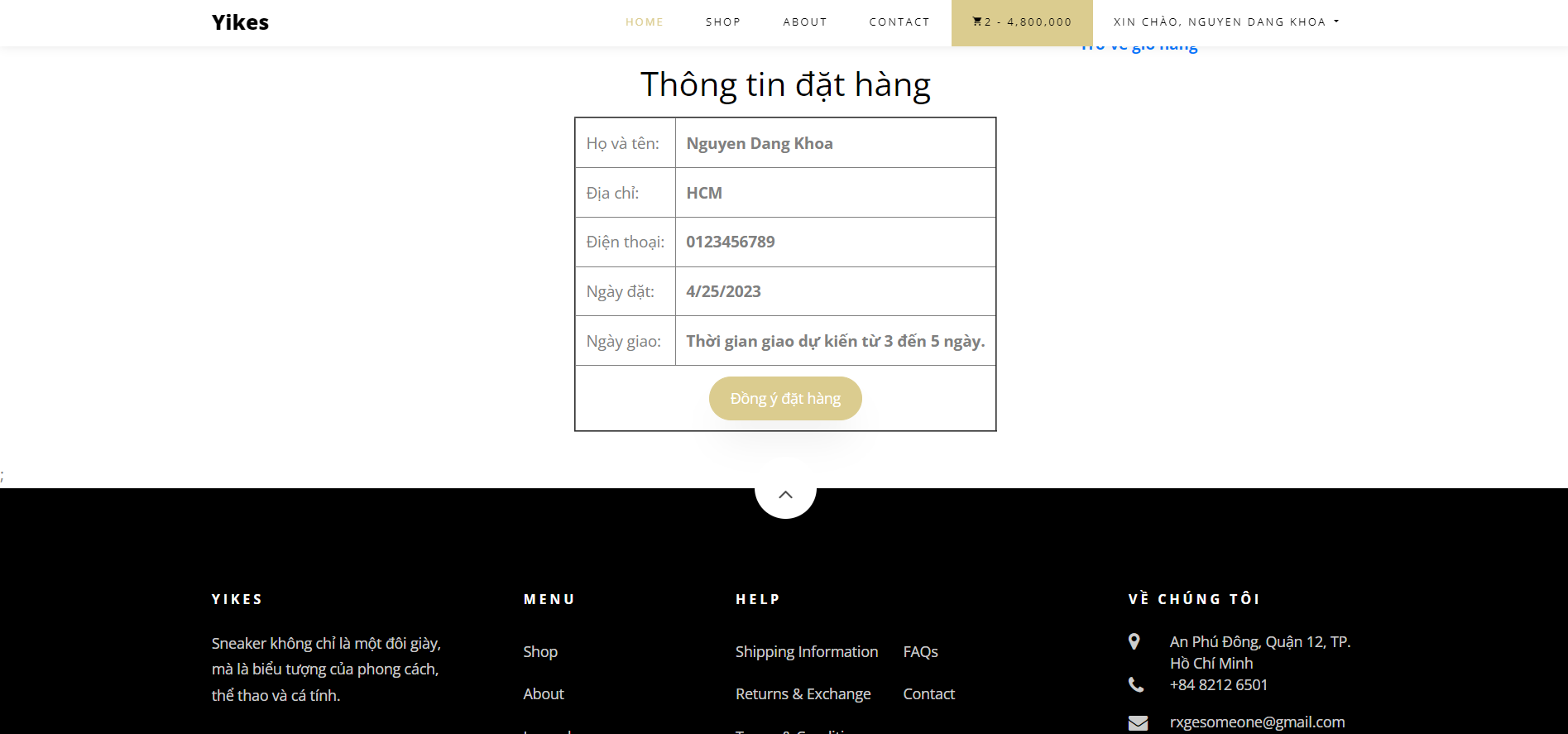
i. Trang đặt hàng

Sau khi người dùng đăng nhập, website sẽ chuyển hướng sang trang đặt hàng và hiển thị thông tin hóa đơn của đơn hàng khi nãy.



**Hình 26 Trang đặt hàng (1)**

Người dùng cần kiểm tra thông tin hóa đơn và thông tin giao hàng trước khi nhấn đặt hàng.



**Hình 27 Trang đặt hàng (2)**

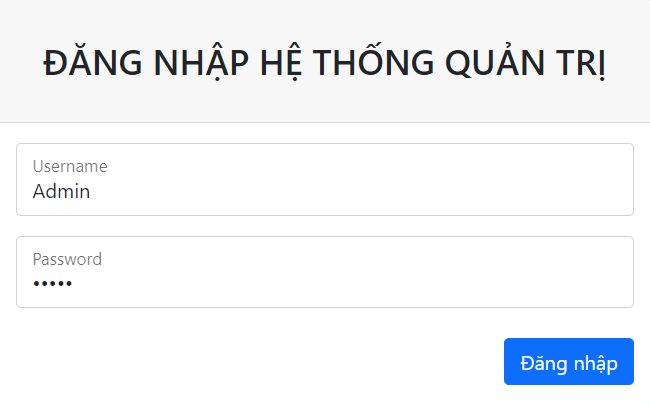
4.1.2. Giao diện trang quản trị (Admin):

Trên thanh địa chỉ localhost, gõ chữ “admin” đằng sau để vào giao diện đăng nhập.



a. Trang đăng nhập quyền quản trị:

Tài khoản Admin/ mật khẩu 12345



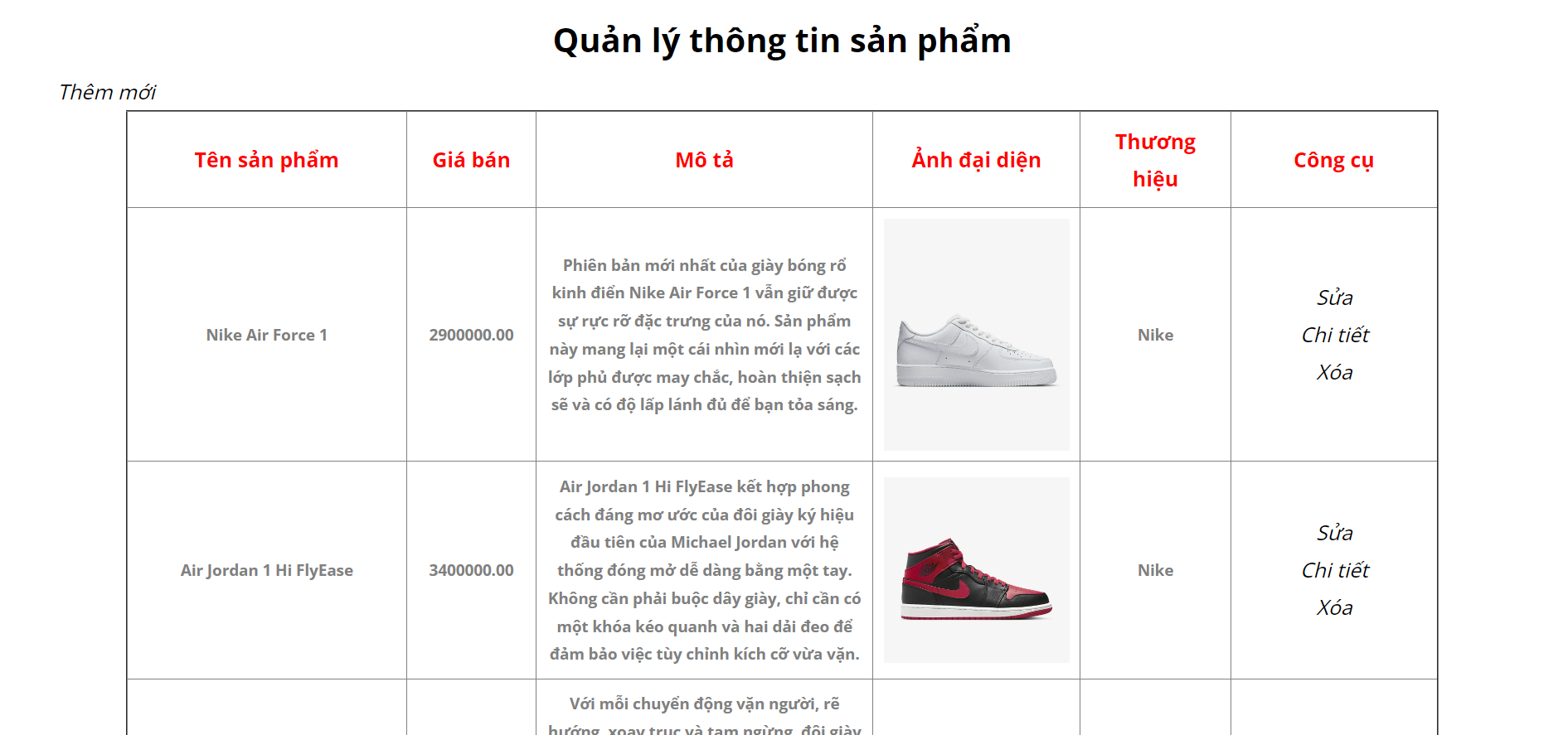
**Hình 28 Trang login quyền quản trị**

b. Trang quản lý đơn hàng:



**Hình 29 Trang index admin**

c. Trang quản lý thông tin sản phẩm:



**Hình 30 Trang quản lý thông tin sản phẩm**

Có thể thực hiện các thao tác thêm, xóa, sửa, xem chi tiết sản phẩm.

d. Trang quản lý thương hiệu



**Hình 31 Trang quản lý thương hiệu**

# CHƯƠNG V: TỔNG KẾT

1. Nhận xét:

* Ưu điểm:
* Hoàn thành 1 website bán hàng online với những tính năng cơ bản nhất cho người dùng và người quản trị.
* Học được cách duy trì và hoạt động của các website bán hàng online hiện nay.
* Nhược điểm:
* Kiến thức còn hạn chế, nhiều kĩ thuật chưa được nắm, cần phải trôi dàu thêm.

1. Kết luận:

Tuy đã hoàn thành xong đồ án, với nhiều kiến thức, kỹ năng còn hạn chế, nhiều thứ còn sai xót, mong thầy (cô) có thể chỉ dẫn và góp ý để em hoàn thiện sản phẩm một cách tốt nhất, hoàn hảo nhất đến tay người tiêu dùng.