

**TP.HCM – Tháng 11, năm 2024**

**Giảng viên hướng dẫn: Nguyễn Ngọc Duy**

**Lớp: D21CQCNPN01-N**

**Sinh viên thực hiện:**

**Bùi Văn Minh - N21DCCN052**

***Đề tài: Xây dựng website bán giày thể thao***

**ĐỒ ÁN MÔN PHÁT TRIỂN CÁC HỆ**

**THỐNG THÔNG MINH**

**KHOA CÔNG NGHỆ THÔNG TIN 2**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

**LỜI CẢM ƠN**

Trước tiên, em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Ngọc Duy, giảng viên Khoa Công nghệ Thông tin 2. Sự hỗ trợ, hướng dẫn tận tình và những góp ý quý báu của thầy đã giúp em rất nhiều trong quá trình thực hiện đề tài "Xây dựng website bán giày thể thao (gợi ý sản phẩm dựa trên mô tả sản phẩm)".

Trong quá trình thực hiện báo cáo, em nhận thấy vẫn còn nhiều thiếu sót, vì vậy em rất mong thầy tiếp tục hỗ trợ và đưa ra những phản hồi, góp ý chi tiết về mặt chuyên môn. Những ý kiến của thầy sẽ giúp em hoàn thiện sản phẩm và báo cáo một cách tốt nhất.

*Tp. Hồ Chí Minh, ngày tháng 20*

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |  |
| --- | --- |
|  | *Tp. Hồ Chí Minh, ngày tháng 20* |
|  | **Gỉảng viên hướng dẫn** |

**MỤC LỤC**

[I. Giới thiệu đề tài 4](#_Toc180533609)

[**1.** **Tên đề tài** 4](#_Toc180533610)

[**2.** **Lý do chọn đề tài** 4](#_Toc180533611)

[**3.** **Mục tiêu nghiên cứu** 4](#_Toc180533612)

[II. Tìm hiểu lý thuyết 4](#_Toc180533613)

[**1.** **Hệ thống thông minh** 4](#_Toc180533614)

[**2.** **Học máy và hệ thống thông minh** 5](#_Toc180533615)

[**3.** **Xây dựng và triển khai hệ thống thông minh** 6](#_Toc180533616)

[III. Phân tích thiết kế. 7](#_Toc180533617)

[**1. Mô tả hệ thống bằng ngôn ngữ tự nhiên.** 7](#_Toc180533618)

[**2. Công nghệ và tài nguyên sử dụng.** 8](#_Toc180533619)

[**2.1 Nodejs** 8](#_Toc180533620)

[**2.2 Express.js** 8](#_Toc180533621)

[**2.3 MongoDB** 9](#_Toc180533622)

[**3. Mô hình Use Case.** 9](#_Toc180533623)

[**4. Sơ đồ ERD.** 10](#_Toc180533624)

[IV. Thiết kế thông minh 10](#_Toc180533625)

[**1. Hệ thống đề xuất sản phẩm dựa trên sản phẩm khách hàng đang quan tâm** 10](#_Toc180533626)

[**1.1. Thu thập và xây dựng cơ sở dữ liệu** 10](#_Toc180533627)

[**2.1 Ưu nhược điểm của mô hình** 11](#_Toc180533628)

[**2. Giao diện (UI)** 12](#_Toc180533629)

[**2.1 Giao diện Đăng nhập/ Đăng ký** 12](#_Toc180533630)

[**2.2 Giao diện Admin** 13](#_Toc180533631)

[**2.3.Giao diện người dùng** 14](#_Toc180533632)

# Chương 1: GIỚI THIỆU ĐỀ TÀI

## 1. Tên đề tài

Website bán giày thể thao (gợi ý sản phẩm dựa trên mô tả sản phẩm)

## 2. Lý do chọn đề tài

* **Xu hướng phát triển thương mại điện tử**: Ngày nay, thương mại điện tử đang phát triển mạnh mẽ, đặc biệt là trong lĩnh vực thời trang và giày dép. Các sản phẩm giày thể thao đang rất được ưa chuộng bởi giới trẻ và người yêu thích thể thao. Việc xây dựng một website bán giày thể thao có tính năng gợi ý sản phẩm sẽ mang lại trải nghiệm mua sắm trực tuyến tốt hơn cho người dùng.
* **Tăng cường trải nghiệm người dùng**: Các trang web thương mại điện tử hiện nay thường gặp khó khăn trong việc giúp khách hàng tìm được sản phẩm họ mong muốn một cách nhanh chóng và dễ dàng. Tính năng gợi ý sản phẩm dựa trên mô tả có thể giúp khách hàng dễ dàng khám phá các sản phẩm phù hợp mà không cần tìm kiếm phức tạp.
* **Ứng dụng công nghệ gợi ý sản phẩm**: Việc áp dụng các thuật toán máy học và trí tuệ nhân tạo (AI) vào website thương mại điện tử giúp cải thiện khả năng gợi ý sản phẩm dựa trên hành vi và nhu cầu của người dùng. Đây là một xu hướng mới và có tiềm năng lớn trong ngành thương mại điện tử.
* **Tăng tỷ lệ chuyển đổi và doanh thu**: Một hệ thống gợi ý sản phẩm hiệu quả có thể giúp tăng tỷ lệ chuyển đổi (conversion rate) và doanh thu cho website bán hàng, bằng cách gợi ý những sản phẩm khách hàng có khả năng mua cao nhất.

## 3. Mục tiêu nghiên cứu

* **Phát triển một website bán giày thể thao hiện đại**: Xây dựng một website với giao diện thân thiện, dễ sử dụng, đáp ứng nhu cầu mua sắm trực tuyến của người dùng, đặc biệt tập trung vào sản phẩm giày thể thao.
* **Tích hợp tính năng gợi ý sản phẩm dựa trên mô tả**: Nghiên cứu và áp dụng các thuật toán gợi ý sản phẩm (recommender system) dựa trên mô tả sản phẩm và hành vi người dùng. Đảm bảo rằng hệ thống có thể hiểu được mô tả sản phẩm và đưa ra những gợi ý chính xác.
* **Nâng cao trải nghiệm người dùng**: Mục tiêu chính là giúp người dùng dễ dàng tìm thấy sản phẩm họ muốn mua thông qua các gợi ý sản phẩm phù hợp. Tạo ra một trải nghiệm mua sắm cá nhân hóa hơn cho khách hàng.
* **Đánh giá hiệu quả của hệ thống gợi ý**: Phân tích dữ liệu người dùng và đánh giá mức độ hài lòng của khách hàng, cũng như so sánh tỷ lệ chuyển đổi trước và sau khi tích hợp tính năng gợi ý sản phẩm.

# Chương 2: Tìm hiểu lý thuyết

## 1. Hệ thống thông minh

* **Khái niệm:** Hệ thống thông minh là một loại hệ thống công nghệ có khả năng tự động hóa và tối ưu hóa các quy trình thông qua việc kết nối người dùng với trí tuệ nhân tạo (AI). Hệ thống này được thiết kế để giúp người dùng đạt được các mục tiêu có ý nghĩa, từ việc ra quyết định đến việc tối ưu hóa các nguồn lực trong nhiều lĩnh vực khác nhau.
* **Hệ thống thông minh ra đời dựa trên các yêu cầu:**
  + - **Xử lý thông tin**: Hệ thống thông minh có khả năng thu thập và xử lý dữ liệu từ nhiều nguồn khác nhau, bao gồm dữ liệu cấu trúc và phi cấu trúc, từ cảm biến, mạng xã hội, và các cơ sở dữ liệu lớn. Việc xử lý thông tin này cho phép hệ thống nhận diện và phân tích các mẫu dữ liệu để đưa ra những quyết định thông minh.
    - **Yêu cầu hiệu suất**: Mức độ chính xác của hệ thống thông minh phụ thuộc vào ứng dụng cụ thể. Trong một số lĩnh vực như y tế hay tài chính, độ chính xác có thể được yêu cầu ở mức cao để đảm bảo kết quả đúng đắn, trong khi ở các lĩnh vực khác có thể chấp nhận được mức độ chính xác thấp hơn.
    - Yêu cầu an ninh và quyền riêng tư: Bảo vệ dữ liệu là một yếu tố then chốt trong bất kỳ hệ thống thông minh nào. Hệ thống cần đảm bảo an toàn cho dữ liệu người dùng và tránh bị tấn công mạng thông qua các biện pháp bảo mật như mã hóa, xác thực người dùng và giám sát an ninh mạng
    - **Yêu cầu đạo đức và xã hội**: Hệ thống thông minh phải được thiết kế và đào tạo để hoạt động theo các tiêu chuẩn đạo đức, đảm bảo rằng các quyết định được đưa ra không gây hại cho con người và xã hội. Điều này bao gồm việc ngăn ngừa sự thiên lệch trong các thuật toán và đảm bảo tính minh bạch trong cách thức hệ thống hoạt động
    - **Yêu cầu kỹ thuật**: Để hỗ trợ hoạt động của hệ thống thông minh, cần có cơ sở hạ tầng phần cứng và phần mềm vững chắc. Điều này bao gồm việc sử dụng các máy chủ mạnh mẽ, mạng lưới an toàn và các công cụ phân tích dữ liệu hiện đại.
    - **Yêu cầu ngữ cảnh và ứng dụng cụ thể**: Hệ thống thông minh cần phục vụ một mục tiêu cụ thể trong các lĩnh vực như chăm sóc sức khỏe, tài chính, giáo dục và công nghiệp. Ví dụ, trong chăm sóc sức khỏe, hệ thống có thể được sử dụng để chẩn đoán bệnh; trong lĩnh vực tài chính, để tư vấn đầu tư và quản lý rủi ro.
* **Ứng dụng** 
  + **Các hệ thống khuyến nghị và cảnh báo**: Hệ thống thông minh có thể được áp dụng để phát triển các hệ thống khuyến nghị, giúp người dùng tìm kiếm sản phẩm hoặc dịch vụ phù hợp với nhu cầu của họ. Ví dụ:
* **Gợi ý sản phẩm**: Các trang thương mại điện tử thường sử dụng hệ thống khuyến nghị để gợi ý sản phẩm dựa trên lịch sử mua sắm và sở thích của người dùng.
* **Cảnh báo thông minh**: Hệ thống có thể đưa ra cảnh báo cho người dùng khi có những thay đổi quan trọng trong môi trường, như thông báo về thời tiết xấu, hay những tin tức quan trọng liên quan đến lĩnh vực mà người dùng quan tâm.
  + **Hệ thống tự động hóa**: Hệ thống thông minh còn được áp dụng trong các lĩnh vực tự động hóa như robot công nghiệp, quản lý sản xuất, và logistic, giúp nâng cao hiệu quả công việc và giảm thiểu sai sót con người.
  + **Phân tích dữ liệu lớn**: Hệ thống thông minh có khả năng xử lý và phân tích lượng dữ liệu lớn (big data) để tìm ra các xu hướng, mẫu hành vi và dự đoán tương lai, từ đó hỗ trợ các quyết định kinh doanh chiến lược.
  + **Trí tuệ nhân tạo trong y tế**: Hệ thống thông minh được áp dụng trong lĩnh vực y tế để hỗ trợ chẩn đoán bệnh, cá nhân hóa điều trị, và theo dõi sức khỏe của bệnh nhân thông qua các thiết bị đeo.
  + **Giáo dục thông minh**: Trong lĩnh vực giáo dục, hệ thống thông minh có thể cung cấp các trải nghiệm học tập cá nhân hóa, đánh giá tiến độ học tập của học sinh và đề xuất các phương pháp học tập hiệu quả hơn.

## 2. Học máy và hệ thống thông minh

* **Học máy (Machine Learning)**
  + **Định nghĩa**: Học máy là một phân nhánh của trí tuệ nhân tạo (AI) tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính học hỏi từ dữ liệu và cải thiện hiệu suất theo thời gian mà không cần phải lập trình lại cụ thể.
  + **Mục tiêu chính**: Mục tiêu của học máy là phát triển các phương pháp giúp máy tính nhận diện mẫu, dự đoán kết quả và ra quyết định dựa trên dữ liệu mà nó đã tiếp nhận. Điều này cho phép hệ thống tự động hóa nhiều quy trình và nâng cao độ chính xác trong việc xử lý thông tin
  + **Phân loại**: Học máy có nhiều loại, bao gồm:
    - **Học có giám sát (Supervised Learning)**: Trong mô hình này, máy học từ một tập dữ liệu đã được gán nhãn, nghĩa là mỗi mẫu dữ liệu có một đầu ra đã biết. Ví dụ, nhận diện hình ảnh hoặc dự đoán giá trị.
    - **Học không giám sát (Unsupervised Learning)**: Hệ thống học từ tập dữ liệu không có nhãn, cố gắng tìm ra cấu trúc hoặc mẫu trong dữ liệu. Ví dụ, phân cụm khách hàng dựa trên hành vi mua sắm.
    - **Học tăng cường (Reinforcement Learning)**: Một phương pháp mà trong đó hệ thống học qua việc tương tác với môi trường và nhận thưởng hoặc phạt dựa trên hành động của mình. Điều này thường được sử dụng trong trò chơi và robot.
* **Hệ thống thông minh (Artificial Intelligence – AI)**
  + **Định nghĩa**: Hệ thống thông minh là một lĩnh vực trong AI mà mục tiêu chính là phát triển các hệ thống có khả năng hoạt động như con người, có khả năng học hỏi, lý luận, giải quyết vấn đề và ra quyết định trong các tình huống phức tạp.
  + **Mục tiêu chính**: Mục tiêu của AI là tạo ra các máy móc hoặc hệ thống có khả năng tự động thực hiện các nhiệm vụ mà thường cần đến trí thông minh của con người. Điều này bao gồm việc nhận diện giọng nói, xử lý ngôn ngữ tự nhiên, hiểu hình ảnh, và thậm chí tham gia vào các cuộc đối thoại phức tạp. AI không chỉ giúp tự động hóa các tác vụ mà còn mở rộng khả năng của con người trong nhiều lĩnh vực, từ chăm sóc sức khỏe đến tài chính và giáo dục

## 3. Xây dựng và triển khai hệ thống thông minh

### 3.1. Quy trình xây dựng

* Xây dựng hệ thống thông minh là một quá trình phức tạp và đòi hỏi sự cân nhắc kỹ lưỡng. Quy trìn này thường bao gồm các bước sau:

1. Thu thập và xử lý dữ liệu

* **Thu thập dữ liệu:** Bước đầu tiên trong việc xây dựng hệ thống thông minh là thu thập dữ liệu từ nhiều nguồn khác nhau, chẳng hạn như cảm biến, cơ sở dữ liệu, hoặc thông qua tương tác của người dùng. Dữ liệu cần phải được xử lý và làm sạch để loại bỏ những thông tin không cần thiết và khắc phục các lỗi dữ liệu. Quá trình này rất quan trọng vì chất lượng dữ liệu sẽ ảnh hưởng trực tiếp đến độ chính xác và hiệu suất của hệ thống.
* **Xử lý dữ liệu:** Sau khi thu thập, dữ liệu cần được xử lý và làm sạch. Điều này bao gồm:
* **Loại bỏ dữ liệu không cần thiết:** Dữ liệu không liên quan hoặc không đáng tin cậy cần được loại bỏ.
* **Khắc phục lỗi dữ liệu:** Dữ liệu bị thiếu hoặc sai lệch cần được sửa chữa hoặc bù đắp bằng các kỹ thuật như nội suy.
* **Chuẩn hóa và chuyển đổi dữ liệu:** Đảm bảo rằng dữ liệu được định dạng thống nhất và có thể dễ dàng phân tích.

2. Thiết kế và phát triển mô hình

* **Phân tích yêu cầu:** Dựa trên dữ liệu đã xử lý, xác định các yêu cầu cho mô hình thông minh. Điều này có thể bao gồm các tính năng mà hệ thống cần hỗ trợ và các tiêu chí đánh giá hiệu suất.
* **Lựa chọn thuật toán:** Chọn các thuật toán học máy hoặc học sâu phù hợp với yêu cầu đã xác định. Việc này bao gồm việc quyết định giữa các mô hình như hồi quy, cây quyết định, mạng nơ-ron, v.v.
* **Huấn luyện mô hình:** Sử dụng tập dữ liệu đã xử lý để huấn luyện mô hình. Quá trình này có thể bao gồm việc điều chỉnh các tham số để tối ưu hóa hiệu suất.

3. Kiểm thử mô hình

* **Kiểm thử với tập dữ liệu chưa thấy:** Đánh giá mô hình trên tập dữ liệu mà nó chưa thấy trước đó để đảm bảo rằng mô hình có thể tổng quát và không chỉ hoạt động tốt trên tập dữ liệu huấn luyện.
* **Đánh giá hiệu suất:** Sử dụng các chỉ số như độ chính xác, độ nhạy, và độ đặc hiệu để đánh giá hiệu suất của mô hình. Nếu mô hình không đạt yêu cầu, cần quay lại các bước trước đó để điều chỉnh.

### 3.2. Triển khai hệ thống

* **Triển khai hệ thống thông minh** là giai đoạn cuối cùng của quy trình xây dựng. Trong giai đoạn này, mô hình đã phát triển và được thử nghiệm sẽ được tích hợp vào các ứng dụng hoặc hệ thống hiện tại. Cần phải đảm bảo rằng hệ thống hoạt động ổn định, có khả năng mở rộng và dễ dàng bảo trì. Các bước quan trọng trong giai đoạn này bao gồm:
* **Tích hợp hệ thống:** Kết hợp mô hình vào nền tảng ứng dụng. Điều này có thể bao gồm việc thiết lập API để mô hình có thể nhận dữ liệu đầu vào và trả kết quả ra ngoài.
* **Theo dõi và bảo trì:** Sau khi triển khai, việc theo dõi hiệu suất của hệ thống là rất quan trọng. Cần có cơ chế để phát hiện lỗi và điều chỉnh mô hình nếu cần thiết. Các công cụ giám sát có thể được sử dụng để theo dõi hiệu suất và tải của hệ thống.

### 3.3. Tổ chức hệ thống

* **Tổ chức hệ thống thông minh** liên quan đến cách các thành phần của hệ thống hoạt động cùng nhau. Các yếu tố cần xem xét bao gồm:
* **Xác định các module khác nhau:** Mỗi phần của hệ thống cần được tổ chức thành các module rõ ràng. Ví dụ, một module có thể chịu trách nhiệm về thu thập dữ liệu, trong khi module khác có thể xử lý dữ liệu và thực hiện phân tích.
* **Tương tác và giao tiếp giữa các module:** Đảm bảo rằng các module có thể giao tiếp với nhau một cách hiệu quả. Điều này có thể bao gồm việc sử dụng các giao thức API hoặc hệ thống message queue để truyền dữ liệu
* **Quản lý dữ liệu và tài nguyên:** Quản lý việc lưu trữ và truy cập dữ liệu, đảm bảo rằng tài nguyên được sử dụng hiệu quả và không gây ra tắc nghẽn trong quá trình hoạt động của hệ thống.

### 3.4. Vấn đề về trade-offs trong xây dựng hệ thống thông minh

* Trong quá trình xây dựng hệ thống thông minh thường có những trade-offs cần được xem xét:
  + - * **Trade-off giữa độ chính xác và tốc độ**: Đôi khi, bạn phải hi sinh độ chính xác để có tốc độ nhanh hơn và ngược lại
      * **Trade-off giữa tiêu chuẩn đạo đức và hiệu suất**: Cân nhắc giữa việc sử dụng dữ liệu có thể gây ra vấn đề về quyền riêng tư và đạo đức và việc đạt được hiệu suất tốt hơn.

### 3.5. Đạo đức trong xây dựng hệ thống

* Việc xây dựng hệ thống thông minh đòi hỏi tuân thủ nguyên tắc đạo đức. Một số điểm cần lưu ý bao gồm:
* **Đảm bảo không gây hại:** Hệ thống không được gây hại cho cá nhân hoặc xã hội. Cần có các biện pháp bảo vệ để đảm bảo rằng các quyết định mà hệ thống đưa ra không dẫn đến hậu quả tiêu cực cho người dùng hoặc cộng đồng.
* **Quyền riêng tư của người dùng:** Hệ thống cần tuân thủ các quy định về bảo vệ dữ liệu cá nhân và đảm bảo quyền riêng tư của người dùng. Cần thông báo cho người dùng về cách thức thu thập và sử dụng dữ liệu của họ.
* **Tránh thiên vị:** Cần tránh tình trạng thiên vị không công bằng trong mô hình hoặc quyết định của hệ thống. Điều này có thể đạt được thông qua việc đảm bảo rằng dữ liệu huấn luyện đại diện cho tất cả các nhóm người dùng và không thiên lệch về giới tính, chủng tộc, hay độ tuổi.
* **Phát triển có trách nhiệm:** Các nhà phát triển và tổ chức cần có trách nhiệm với việc xây dựng và triển khai hệ thống thông minh, đảm bảo rằng các quyết định của họ không chỉ mang lại lợi ích kinh tế mà còn đáp ứng các tiêu chuẩn đạo đức và xã hội.

## 4. Các công nghệ sử dụng

### 4.1. Node.js

* **Mô tả về Node.js:**
* **Node.js** là một nền tảng JavaScript phía server, chạy trên engine V8 của Google, giúp thực thi mã JavaScript với hiệu suất cao.
* Node.js được thiết kế dựa trên mô hình **event-driven** (hướng sự kiện) và **non-blocking I/O** (I/O không đồng bộ), giúp tối ưu hóa hiệu suất khi xử lý các yêu cầu đồng thời.
* **Lợi ích của Node.js trong dự án**
* **Hiệu suất cao và khả năng xử lý đồng thời tốt**: Nhờ mô hình không đồng bộ, Node.js có thể xử lý nhiều kết nối cùng lúc mà không bị chậm, phù hợp cho ứng dụng thương mại điện tử với lượng truy cập lớn.
* **Nhất quán trong ngôn ngữ lập trình**: Việc sử dụng JavaScript cho cả frontend và backend giúp giảm sự phức tạp trong quản lý code, đồng thời tăng tốc độ phát triển.
* **Hệ sinh thái phong phú**: Với **npm** (Node Package Manager), có hàng ngàn module và thư viện có sẵn, hỗ trợ nhiều chức năng từ xử lý HTTP, kết nối cơ sở dữ liệu, đến các công cụ phát triển khác.
* **Ứng dụng của Node.js trong hệ thống**
* Node.js được sử dụng để phát triển **server backend**, nơi xử lý tất cả các yêu cầu HTTP từ frontend, kết nối với cơ sở dữ liệu MongoDB, và cung cấp các API cho người dùng.
* Hỗ trợ việc phát triển các **API RESTful**, cho phép giao tiếp dữ liệu dễ dàng giữa frontend và backend.

### 4.2. Express.js

* **Mô tả về Express.js**
* **Express.js** là một framework web nhẹ và mạnh mẽ cho Node.js, giúp đơn giản hóa việc phát triển các ứng dụng web và API.
* Express.js cung cấp một bộ công cụ linh hoạt cho việc quản lý **routing**, **middleware**, và xử lý các yêu cầu HTTP.
* **Lợi ích của Express.js trong dự án**
* **Routing hiệu quả**: Express.js cho phép thiết lập các route rõ ràng và dễ quản lý, giúp phân loại và điều hướng các yêu cầu đến đúng chức năng của ứng dụng.
* **Middleware mạnh mẽ**: Các middleware trong Express.js giúp xử lý các yêu cầu trước khi đến các route chính, hỗ trợ các chức năng như phân tích dữ liệu JSON, xử lý xác thực, và quản lý lỗi.
* **Tích hợp dễ dàng**: Express.js tích hợp tốt với các công nghệ khác như MongoDB, Handlebars, và các hệ thống microservice, giúp xây dựng một kiến trúc thống nhất và dễ mở rộng.
* **Vai trò của Express.js trong hệ thống**
* Quản lý các **API endpoint** cho các chức năng như tìm kiếm sản phẩm, đăng ký người dùng, đăng nhập, quản lý giỏ hàng, và xử lý thanh toán.
* Sử dụng middleware để xử lý dữ liệu và quản lý bảo mật kết nối giữa frontend và backend.
* Hỗ trợ kết nối với cơ sở dữ liệu MongoDB thông qua các API, đồng thời tích hợp Handlebars để render giao diện phía server.

### 4.3. MongoDB

* Mô tả về MongoDB
* **MongoDB** là một cơ sở dữ liệu NoSQL dạng tài liệu, cho phép lưu trữ dữ liệu dưới dạng JSON (BSON). Điều này mang lại tính linh hoạt cao khi lưu trữ dữ liệu không có cấu trúc cố định.
* MongoDB hỗ trợ tính năng **sharding** và **replication**, giúp tối ưu hóa hiệu suất và đảm bảo độ tin cậy cho hệ thống.
* Ưu điểm của MongoDB
* **Khả năng mở rộng linh hoạt**: MongoDB có thể dễ dàng mở rộng theo chiều ngang bằng cách thêm nhiều server vào hệ thống.
* **Tốc độ truy vấn nhanh**: Với khả năng lưu trữ dữ liệu dạng tài liệu, MongoDB cho phép truy vấn dữ liệu nhanh chóng, phù hợp cho các ứng dụng cần xử lý dữ liệu lớn.
* **Dễ dàng tích hợp**: MongoDB tích hợp tốt với Node.js thông qua thư viện **Mongoose**, giúp dễ dàng thực hiện các thao tác CRUD (Create, Read, Update, Delete).
* Ứng dụng của MongoDB trong hệ thống
* Lưu trữ thông tin sản phẩm, người dùng, đơn hàng, và dữ liệu gợi ý sản phẩm.
* Tạo các collection khác nhau cho các loại dữ liệu khác nhau như "products", "users", "orders"
* Sử dụng các chỉ mục (index) để tối ưu hóa hiệu suất truy vấn dữ liệu.

### 4.4. Handlebars

* **Mô tả về Handlebars**
* **Handlebars** là một template engine cho Node.js, giúp tạo giao diện động phía server. Handlebars sử dụng cú pháp đơn giản, dễ đọc, cho phép nhúng dữ liệu trực tiếp vào HTML.
* **Lợi ích của Handlebars**
* **Dễ học và sử dụng**: Cú pháp Handlebars đơn giản, dễ dàng tích hợp với Express.js và Node.js.
* **Tái sử dụng mã hiệu quả**: Hỗ trợ sử dụng các template, partial views, và layout, giúp giảm thiểu sự lặp lại mã và dễ dàng bảo trì.
* **Tối ưu hóa hiệu suất**: Render giao diện phía server giúp giảm tải cho trình duyệt, đồng thời tăng tốc độ hiển thị trang.

### 4.5. HTML & CSS

* Mô tả về HTML
* Mô tả về CSS

# III. Phân tích thiết kế.

## 1. Mô tả hệ thống bằng ngôn ngữ tự nhiên.

* **Hệ thống bán giày thể thao** là một nền tảng trực tuyến cho phép khách hàng, nhân viên quản lý, thông tin tài khoản và các mặt hàng giày thể thao.
* **Khách hàng:**
* Khách hàng là người dùng cuối sử dụng hệ thống để mua sắm giày. Họ có thể là cá nhân hoặc tổ chức có nhu cầu mua sắm các sản phẩm thể thao.
* Khách hàng có thể tạo tài khoản trên hệ thống để lưu trữ thông tin cá nhân, địa chỉ giao hàng và lịch sử đơn hàng.
* Hệ thống hỗ trợ tính năng tìm kiếm và duyệt danh mục sản phẩm, cho phép khách hàng thêm sản phẩm vào giỏ hàng và thực hiện thanh toán trực tuyến
* Hệ thống gợi ý sản phẩm dựa trên mô tả sản phẩm và tính toán độ tương đồng giữa các sản phẩm với sản phẩm đang được khách hàng xem.
* **Nhân viên quản lý:**
* Nhân viên quản lý là những người điều hành và quản trị hệ thống bán giày.
* Họ có quyền truy cập vào các chức năng quản lý quan trọng như quản lý tài khoản khách hàng, sản phẩm, và đơn hàng.
* Chức năng của họ bao gồm việc tạo, chỉnh sửa và xóa các bản ghi sản phẩm giày, cũng như quản lý trạng thái đơn hàng và hỗ trợ khách hàng khi có vấn đề phát sinh.
* **Thông tin tài khoản:**
* Thông tin tài khoản là phần của hệ thống lưu trữ thông tin cá nhân của khách hàng và nhân viên quản lý.
* Đối với khách hàng, thông tin tài khoản bao gồm tên, địa chỉ, thông tin liên hệ, và lịch sử mua hàng.
* Đối với nhân viên quản lý, thông tin tài khoản bao gồm quyền hạn để quản lý sản phẩm, danh mục, và đơn hàng.
* **Các mặt hàng giày thể thao:**
* Các mặt hàng giày thể thao là tập hợp các sản phẩm thể thao có sẵn để bán trên hệ thống.
* Mỗi sản phẩm quần áo có thông tin cụ thể như tên sản phẩm, mô tả, kích thước, màu sắc, giá cả, và số lượng tồn kho.
* Các sản phẩm được phân loại thành các danh mục khác nhau để khách hàng dễ dàng tìm kiếm và lựa chọn, đồng thời có hệ thống gợi ý các sản phẩm tương tự hoặc phù hợp với sở thích của khách hàng.

## 2. Công nghệ và tài nguyên sử dụng.

### 2.1 Nodejs

* Node.js xuất hiện vào năm 2009, được phát triển bởi Ryan Dahl, là một nền tảng xây dựng ứng dụng sử dụng JavaScript bên máy chủ. Node.js cho phép các nhà phát triển chạy mã JavaScript bên ngoài trình duyệt, giúp tạo ra các ứng dụng web mạnh mẽ và hiệu quả.
* Node.js hoạt động trên mô hình phi đồng bộ và hướng sự kiện, cho phép xử lý nhiều kết nối đồng thời mà không bị chặn. Điều này giúp tăng cường hiệu suất và khả năng mở rộng của ứng dụng. Một trong những điểm mạnh của Node.js là sử dụng V8 JavaScript Engine của Google, mang lại tốc độ thực thi nhanh chóng cho các ứng dụng.
* Ngày nay, Node.js được sử dụng cho nhiều mục đích như sau:
* Phát triển ứng dụng web: Node.js cho phép xây dựng các ứng dụng web động và thời gian thực, như trò chuyện trực tuyến và trò chơi đa người.
* Xây dựng API: Node.js rất phổ biến trong việc tạo ra các API RESTful cho các ứng dụng di động và web, cho phép giao tiếp giữa máy chủ và client.
* Xử lý dữ liệu theo thời gian thực: Node.js có khả năng xử lý luồng dữ liệu theo thời gian thực, thích hợp cho các ứng dụng như ứng dụng tài chính và theo dõi trực tuyến.
* Phát triển microservices: Với khả năng mở rộng tốt, Node.js là lựa chọn lý tưởng cho kiến trúc microservices, giúp chia nhỏ các ứng dụng lớn thành các dịch vụ nhỏ hơn và dễ quản lý

### 2.2 Express.js

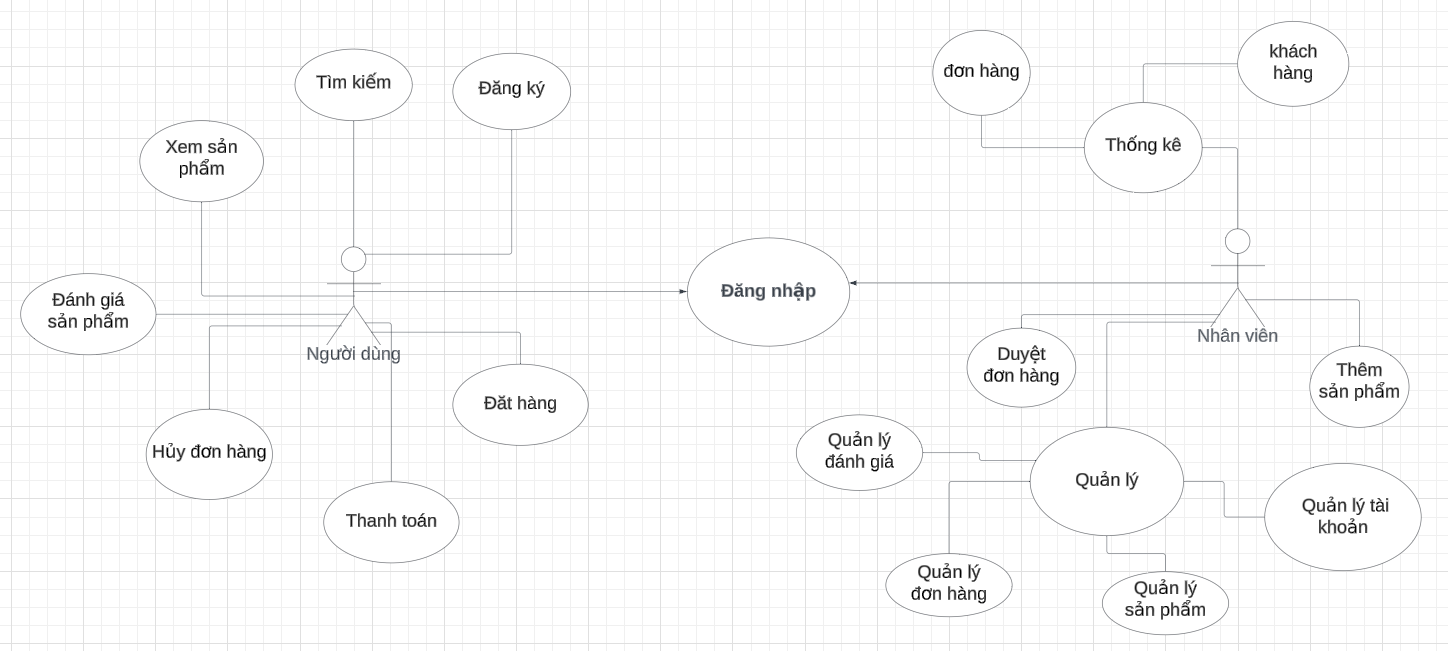
* Express.js là một framework web nhẹ và mạnh mẽ cho Node.js, được sử dụng để xây dựng các ứng dụng web và API. Nó cung cấp một bộ công cụ đơn giản và linh hoạt để phát triển các ứng dụng web, cho phép nhà phát triển xây dựng một ứng dụng phía máy chủ một cách nhanh chóng và hiệu quả.
* Một số đặc điểm chính của Express.js:
* **Đơn giản và nhẹ**: Express.js cung cấp một lớp trừu tượng nhẹ trên Node.js, giúp dễ dàng tạo ra các ứng dụng web mà không phải viết quá nhiều mã lệnh.
* **Routing mạnh mẽ**: Express.js có hệ thống định tuyến linh hoạt, cho phép định nghĩa các đường dẫn URL và cách xử lý chúng một cách dễ dàng.
* **Middleware**: Express.js hỗ trợ middleware, là các hàm được thực hiện trước khi gửi phản hồi tới client, giúp dễ dàng xử lý dữ liệu, xác thực, logging và các công việc khác.
* **Hỗ trợ API RESTful**: Express.js được tối ưu hóa để phát triển các API RESTful, cho phép dễ dàng quản lý và triển khai các dịch vụ web.
* **Tích hợp dễ dàng với cơ sở dữ liệu**: Express.js hoạt động tốt với các hệ quản trị cơ sở dữ liệu như MongoDB, MySQL, PostgreSQL, cho phép dễ dàng quản lý dữ liệu và tích hợp vào ứng dụng.

### 2.3 MongoDB

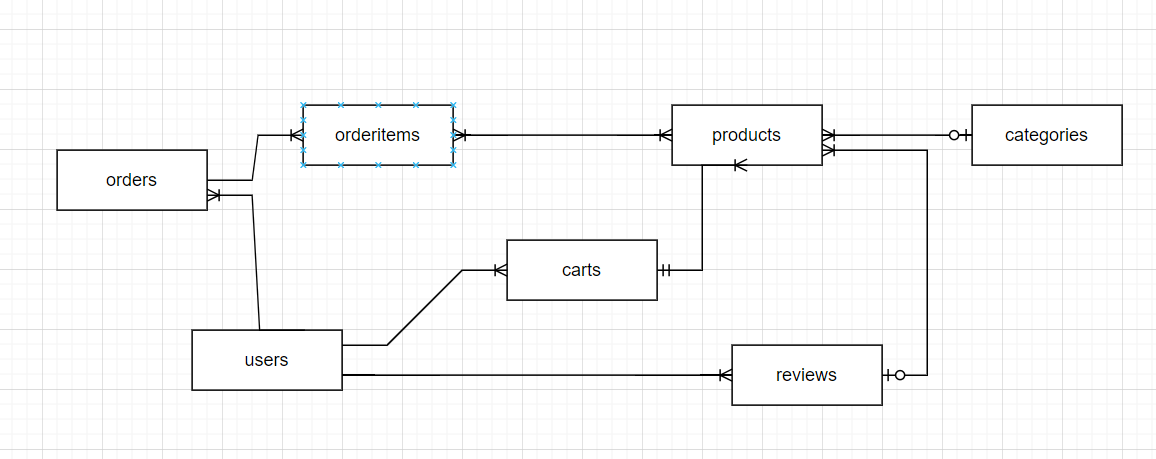
* MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, được phát triển bởi công ty MongoDB Inc. Đây là cơ sở dữ liệu hướng tài liệu, sử dụng cấu trúc dữ liệu JSON-like (BSON), giúp lưu trữ và quản lý dữ liệu một cách linh hoạt hơn so với các hệ cơ sở dữ liệu quan hệ (SQL).
* Một số đặc điểm nổi bật của MongoDB:
* **Lưu trữ dữ liệu dưới dạng tài liệu**: Thay vì lưu trữ dữ liệu trong các bảng có quan hệ như SQL, MongoDB lưu trữ dưới dạng tài liệu BSON, giúp dễ dàng mở rộng và thay đổi cấu trúc.
* **Tính linh hoạt và mở rộng**: MongoDB có khả năng mở rộng ngang (scale horizontally), hỗ trợ phân tán dữ liệu trên nhiều máy chủ khác nhau.
* **Hỗ trợ truy vấn phong phú**: MongoDB cung cấp các phương thức truy vấn mạnh mẽ, bao gồm truy vấn theo trường, lọc, sắp xếp và phân trang.
* **Khả năng tích hợp**: MongoDB dễ dàng tích hợp với nhiều ngôn ngữ lập trình và công nghệ phổ biến như Node.js, Python, Java,...

## 3. Mô hình Use Case.

* **UseCase cho toàn bộ hệ thống**



## 4. Sơ đồ ERD.



## 

# IV. Thiết kế thông minh

## 1. Hệ thống đề xuất sản phẩm dựa trên sản phẩm khách hàng đang quan tâm

### 1.1. Thu thập và xây dựng cơ sở dữ liệu

* **Bước 1: thu thập dữ liệu sản phẩm**
* Đặc điểm giày: tên, kích cỡ , mô tả,…
* Loại hình sử dụng: giày chạy bộ, đá bóng,…
* **Bước 2: thu thập dữ liệu khách hàng**
* Hành vi tương tác: sản phẩm khách hàng xem, click, mua, thêm vào giỏ hàng,…
* Sở thích và lịch sử mua hàng: Thương hiệu ưa thích, loại giày đã mua
* **Bước 3: làm sạch và chuẩn hóa dữ liệu**

 Xóa dữ liệu trùng lặp hoặc không hợp lệ.

 Chuẩn hóa thông tin (ví dụ: thống nhất đơn vị kích cỡ EU/US).

 Xử lý dữ liệu thiếu bằng cách suy đoán dựa trên dữ liệu tương tự.

1.2.Tính toán điểm tương đồng giữa sản phẩm đang xem và các sản phẩm khác

* Sử dụng Cosine similarity để tìm các sản phẩm tương đồng

Cosine Similarity tính góc giữa hai vector đặc trưng của sản phẩm



Nếu giá trị gần 1, hai sản phẩm rất tường đồng

* **Bước 4: phát triển mô hình gợi ý**
* **KNN (K-Nearest Neighbors)**:
* Tìm **K sản phẩm gần nhất** dựa trên đặc trưng đã chuẩn hóa và đo độ tương đồng.
* **Mạng nơ-ron nhúng (Embedding Network)**:
* Nếu dữ liệu phức tạp (ví dụ: hình ảnh hoặc mô tả dài), dùng **mạng nơ-ron** để mã hóa các sản phẩm thành các vector nhúng trong không gian nhiều chiều.
* **Bước 5: Kiểm thử mô hình**

 **Đánh giá bằng các chỉ số**:

* **Precision**: Tỷ lệ sản phẩm gợi ý đúng so với tất cả sản phẩm được đề xuất.
* **Recall**: Tỷ lệ sản phẩm phù hợp trong tất cả các sản phẩm có thể được gợi ý.
* **RMSE (Root Mean Square Error)**: Đánh giá sai số giữa giá trị dự đoán và thực tế.

 **A/B Testing**: So sánh hiệu quả giữa các mô hình gợi ý khác nhau (ví dụ: mô hình KNN vs. mô hình mạng nơ-ron).

* **Bước 6: Triển khai API gợi ý**

**API** sẽ nhận thông tin về sản phẩm đang xem (mã sản phẩm, danh mục) và trả về danh sách **n sản phẩm tương tự** dựa trên độ tương đồng đã tính toán.

* **Bước 7:Theo dõi và tối ưu hóa**

 **Thu thập phản hồi của người dùng**:

* Theo dõi tỷ lệ click (CTR) và tỷ lệ chuyển đổi (conversion rate).

 **Cập nhật mô hình định kỳ**:

* Huấn luyện lại mô hình khi có dữ liệu mới hoặc thay đổi xu hướng tiêu dùng.

 **Xử lý lỗi và cải tiến hiệu năng**:

* Tối ưu hóa thời gian phản hồi của API và cải thiện thuật toán khi cần.

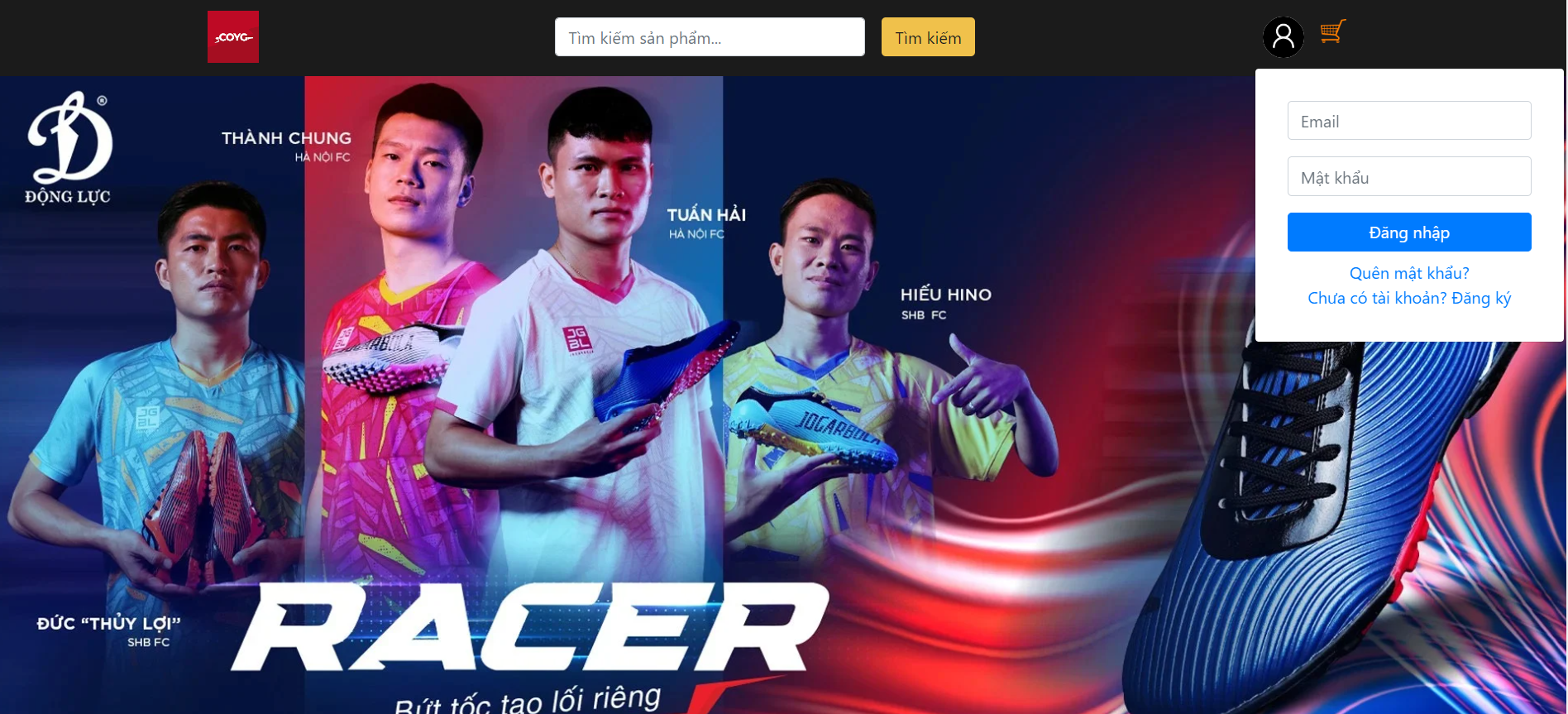
### 2.1 Ưu nhược điểm của mô hình

* Ưu điểm
* Cá nhân hóa cao: mô hình có khả năng gợi ý sản phẩm dựa trên sở thích và nhu cầu cụ thể của từng khách hàng, từ đó tạo ra trải nghiệm mua sắm tốt hơn.
* **Dễ hiểu và dễ triển khai:** Phương pháp Content-based Filtering thường dễ hiểu và có thể được triển khai mà không cần quá nhiều dữ liệu người dùng. Chỉ cần thông tin về sản phẩm và mô tả sản phẩm là đủ để bắt đầu.
* **Không cần dữ liệu từ người dùng khác:** Khác với các phương pháp Collaborative Filtering, mô hình này không phụ thuộc vào dữ liệu của người dùng khác, giúp tránh vấn đề "cold start" (khi không có đủ thông tin người dùng để gợi ý).
* **Cập nhật linh hoạt:** Dễ dàng cập nhật và thêm sản phẩm mới vào hệ thống mà không cần phải thay đổi cách hoạt động của mô hình. Sản phẩm mới sẽ được tự động đưa vào vòng lặp phân tích và gợi ý.
* **Tính chính xác cao cho sản phẩm tương tự:** Mô hình có khả năng gợi ý chính xác các sản phẩm tương tự dựa trên các đặc điểm đã được xác định (chẳng hạn như thương hiệu, loại, tính năng).
* **Nhược điểm**
* **Giới hạn trong việc khám phá sản phẩm mới**: Mô hình chỉ gợi ý các sản phẩm tương tự với sản phẩm đang xem, dẫn đến khả năng giới hạn trong việc khám phá các sản phẩm mới mà người dùng có thể quan tâm.
* Thiếu sự đa dạng: Gợi ý có thể trở nên đơn điệu và thiếu đa dạng. Người dùng có thể nhận được gợi ý tương tự nhau, dẫn đến trải nghiệm nhàm chán.
* Chất lượng dữ liệu ảnh hưởng lớn: Nếu thông tin sản phẩm không đầy đủ hoặc không chính xác (ví dụ: mô tả sản phẩm không rõ ràng), chất lượng gợi ý sẽ bị ảnh hưởng nghiêm trọng.
* Khó khăn trong việc xử lý dữ liệu không có cấu trúc: Mô hình có thể gặp khó khăn khi phải xử lý các loại dữ liệu không có cấu trúc, chẳng hạn như hình ảnh sản phẩm, nếu không có các kỹ thuật học sâu phù hợp.
* Phụ thuộc vào sự phong phú của thông tin sản phẩm: Chất lượng gợi ý phụ thuộc vào việc có đủ thông tin và đặc điểm mô tả sản phẩm. Nếu sản phẩm không được mô tả đầy đủ, mô hình sẽ không hoạt động hiệu quả.

## 2. Giao diện (UI)

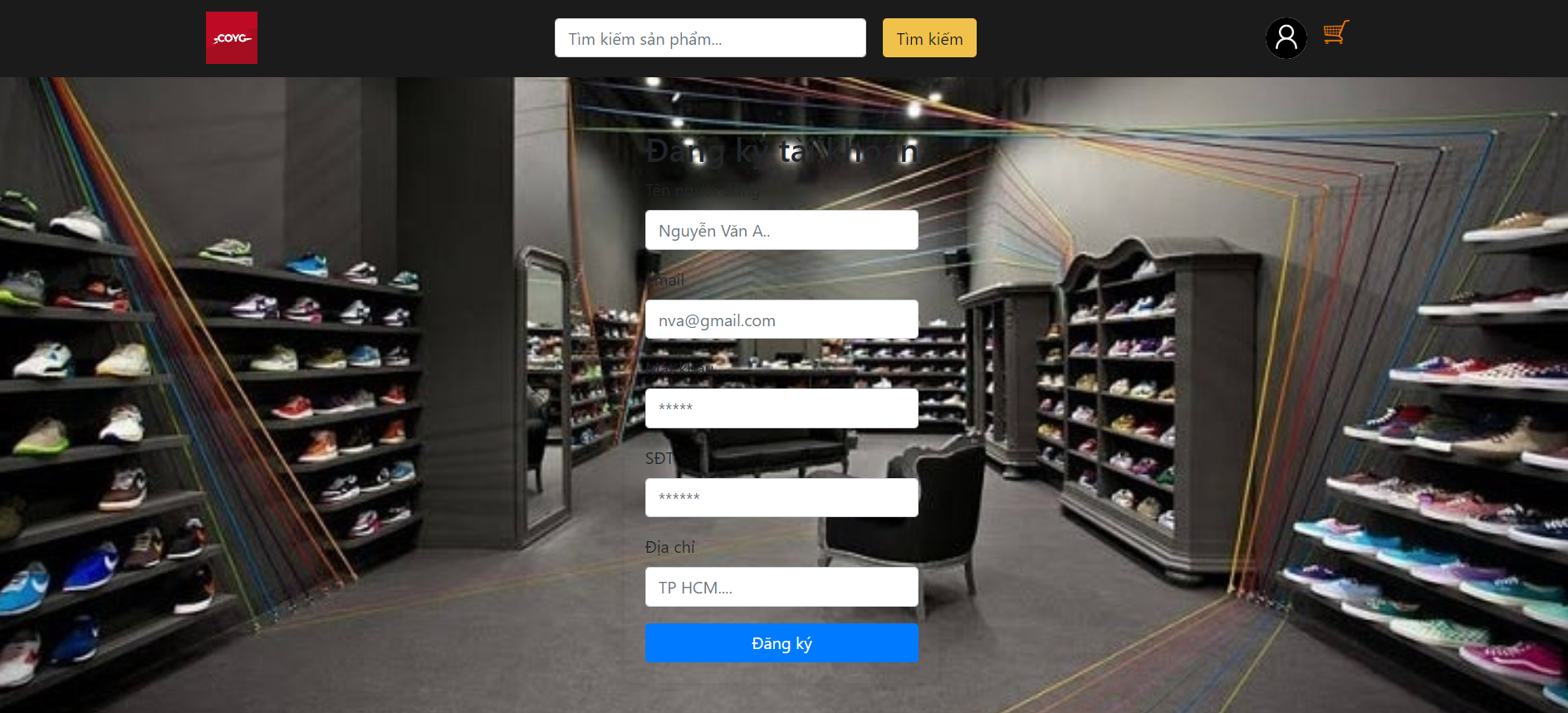
### 2.1 Giao diện Đăng nhập/ Đăng ký

- **Đăng nhập tài khoản**

****

* Giao diện đăng nhập cho phép người dùng đăng nhập vào tài khoản bằng email và mật khẩu ở bên góc phải màn hình
* Nếu người dùng chưa có tài khoản, bấm vào nút chưa có tài khoản và đăng ký để tạo tài khoản mới
* Nếu người dùng quên mật khẩu, bấm vào “quên mật khẩu” để tạo lại mật khẩu mới

- **Đăng ký tài khoản**

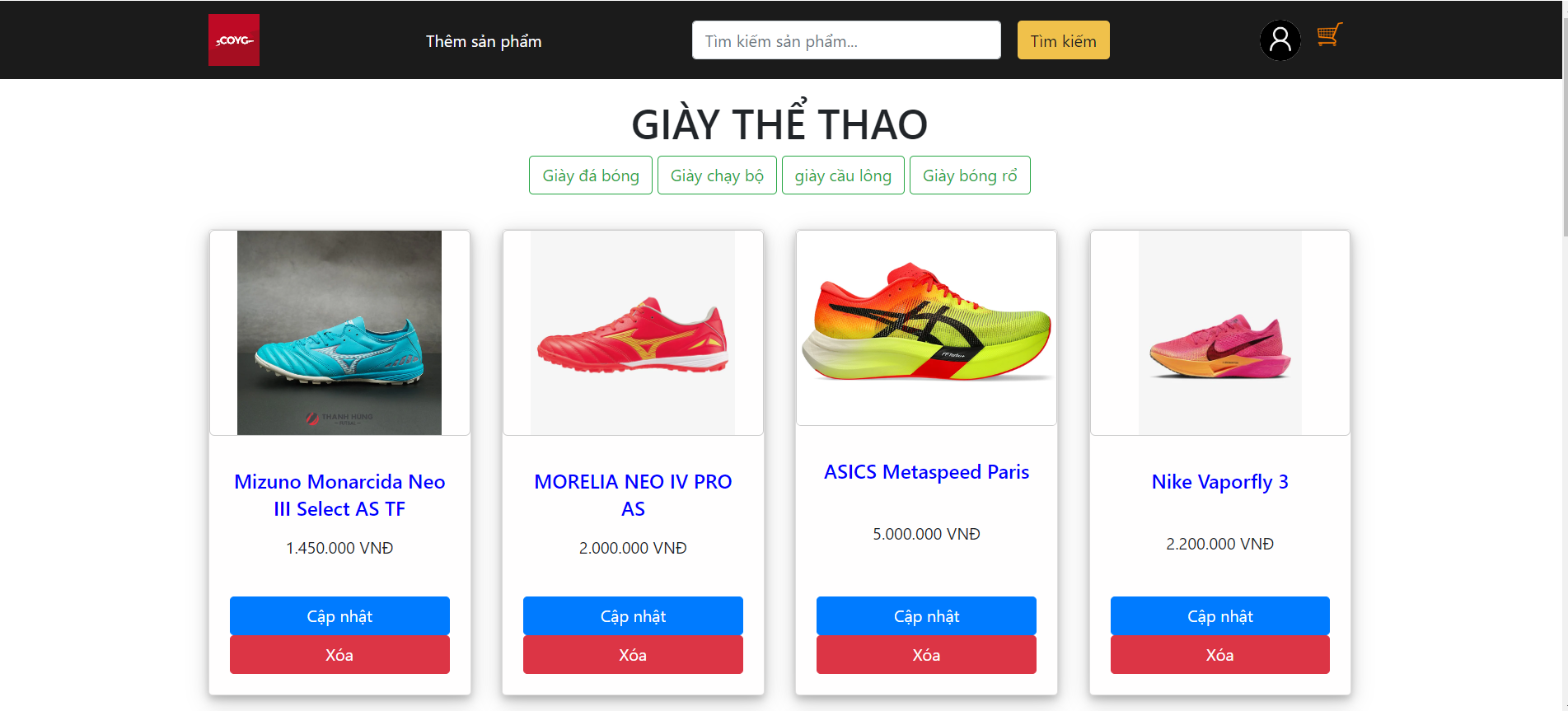
****

* Giao diện đăng ký cho phép người dùng tạo tài khoản mới bằng cách điền         thông tin cá nhân.
* Các trường bao gồm: họ và tên lót, tên, tên người dùng, mật khẩu, và email.
* Sau khi hoàn tất, người dùng nhấn vào nút "Đăng ký" để hoàn thành quá trình  đăng ký.

### 2.2 Giao diện Admin

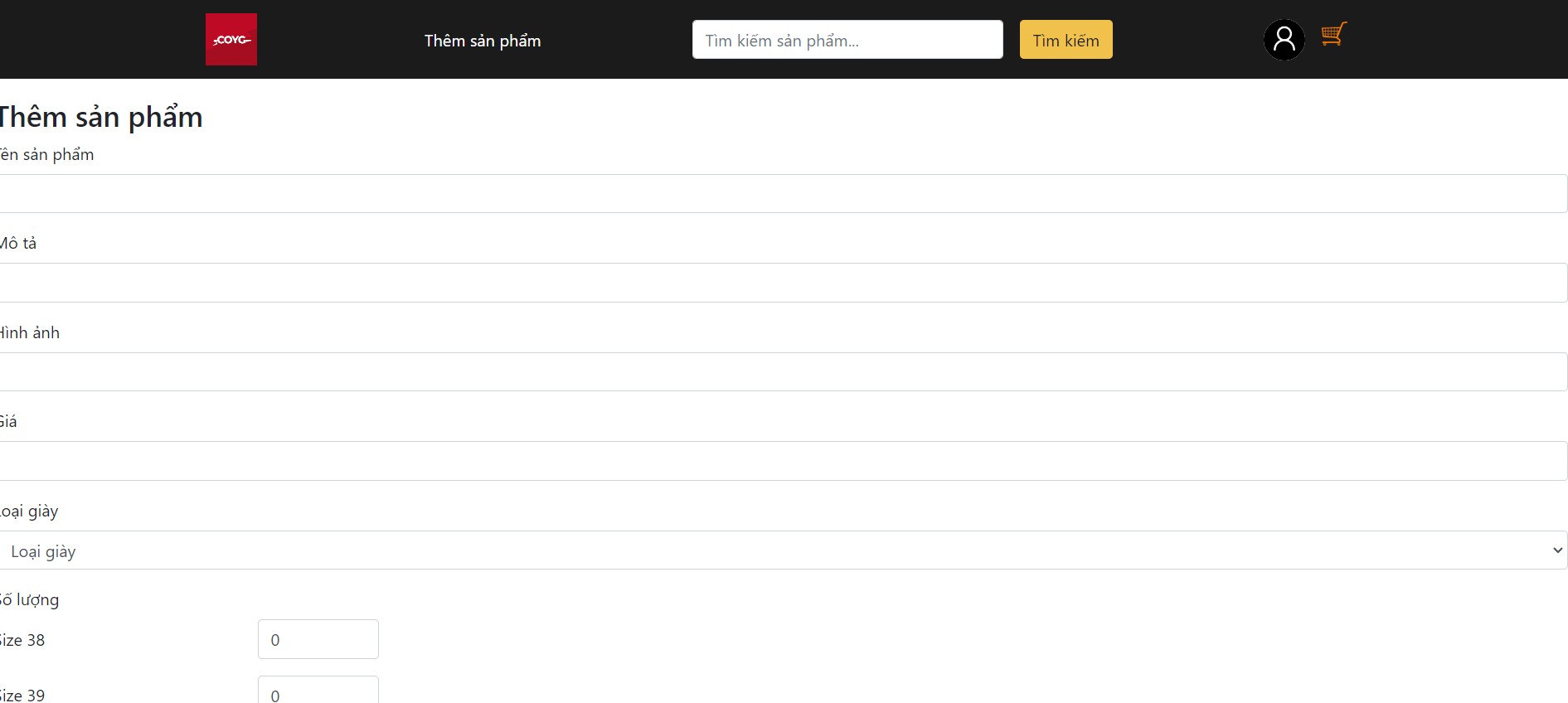
#### **- Trang chủ**

#### 

****

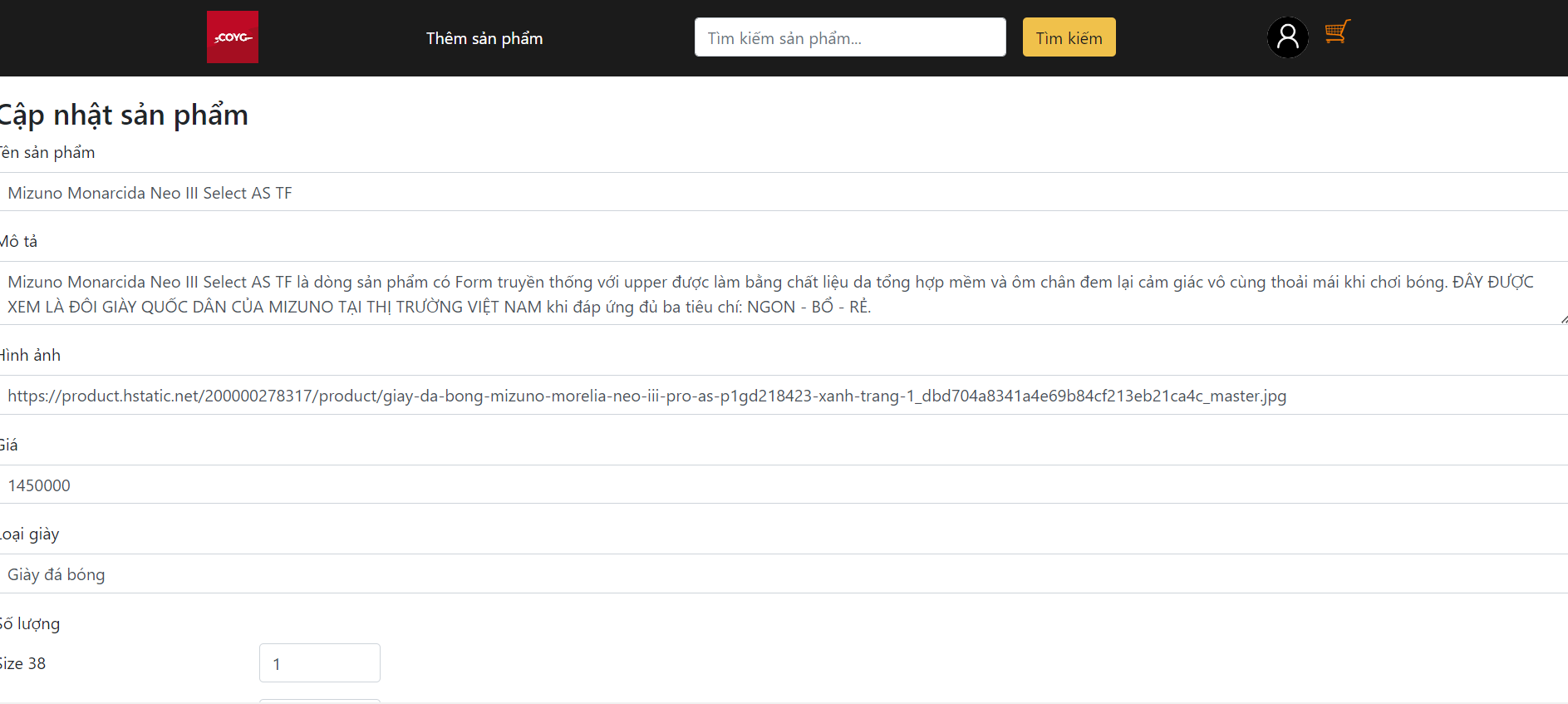
* Trang hiển thị tất cả sản phẩm giày trên hệ thống
* Nút tìm kiếm để tìm kiếm theo tên sản phẩm
* Nút thêm sản phẩm sẽ dùng để thêm sản phẩm mới nếu cửa hàng có sản phẩm mới
* Dưới mỗi sản phẩm sẽ có nút cập nhật và xóa những sản phẩm trong cửa hàng

**- Thêm sản phẩm**

****

* Giao diện thêm sản phẩm để thêm những sản phẩm mới cho cửa hàng bao gồm thông tin tên, mô tả, hình ảnh, loại giày và số lượng giày

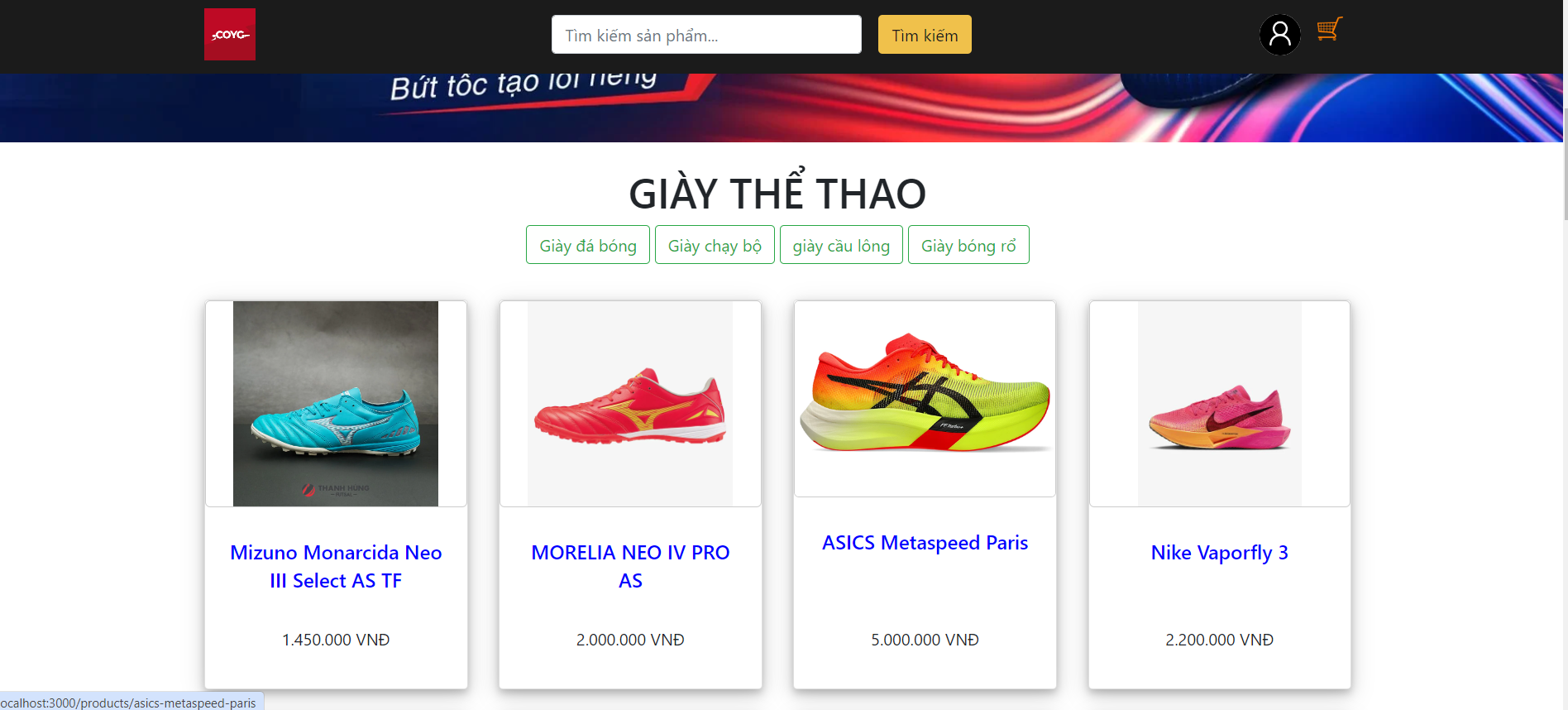
**- Cập nhật sản phẩm**

****

* Giao diện cập nhật sản phẩm để thay đổi thông tin sản phẩm đang có trong cửa hàng

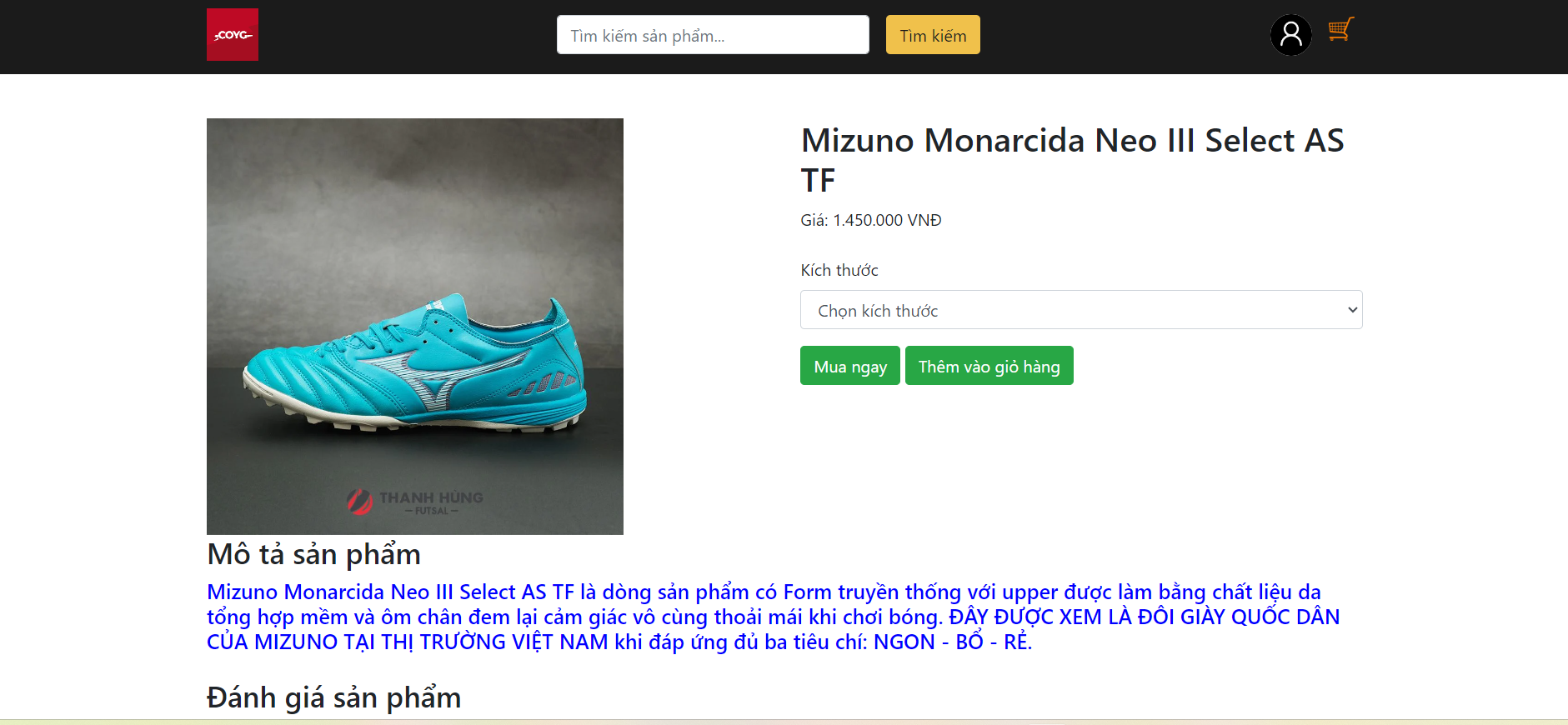
### 2.3.Giao diện người dùng

**- Trang chủ**

****

* Giao diện người dùng hiển thị sản phẩm có trong cửa hàng và người dùng có thể lọc theo từng sản phẩm với nhu cầu thể thao của mình
* Thanh “tìm kiếm” để người dùng có thể tìm kiếm sản phẩm theo tên sản phẩm mà người dùng muốn tìm
* Góc phải là biểu tượng thông tin người dùng và giỏ hàng hiển thị sản phẩm khách hàng đã thêm vào

**- Chi tiết sản phẩm**

****

* Giao diện chi tiết sản phẩm hiển thị chi tiết thông tin giá cả, mô tả và đánh giá sản phẩm mà khách hàng đang xem
* Nút “mua ngay” chuyển đến trang thanh toán để mua sản phẩm
* Nút “thêm vào giỏ hàng” để thêm sản phẩm vào giỏ hàng