**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

🙥🙥🙥🙥🕮🙥🙥🙥🙥



**KHOA CÔNG NGHỆ THÔNG TIN 2**

**BÁO CÁO ĐỒ ÁN MÔN HỌC PHÁT TRIỂN CÁC HỆ THỐNG THÔNG MINH**

**ĐỀ TÀI:**

**PHÁT TRIỂN HỆ THỐNG GỢI Ý SẢN PHẨM THÔNG MINH CHO WEBSITE BÁN THIẾT BỊ ĐIỆN TỬ**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn** | **:** | **Nguyễn Ngọc Duy** |
| **Lớp** | **:** | **D22CQCNPM02-N** |
| **Sinh viên thực hiện** | **:** | **Y Cao Nguyên Byă** |
| **MSSV** | **:** | **N22DCCN200** |

***TP.Hồ Chí Minh, tháng 11 năm 2025***

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Ngọc Duy vì những bài giảng bổ ích và tận tâm trong suốt thời gian qua. Nhờ sự hướng dẫn tận tình của các thầy, em đã không chỉ nắm vững kiến thức mà còn phát triển được nhiều kỹ năng quan trọng trong lĩnh vực chuyên môn của mình.

Thầy luôn là nguồn cảm hứng để em nỗ lực học tập và rèn luyện bản thân.

Những lời khuyên và kiến thức mà thầy truyền đạt sẽ là hành trang quý báu trên con đường học vấn và sự nghiệp của em sau này.

Em cảm ơn thầy đã luôn kiên nhẫn giải đáp mọi thắc mắc, động viên và hỗ trợ em cũng như các bạn trong lớp. Sự nhiệt huyết và tâm huyết của thầy là động lực lớn lao để chúng em vượt qua những khó khăn và thử thách trong quá trình học tập.

Một lần nữa, em xin chân thành cảm ơn và kính chúc thầy luôn dồi dào sức khỏe, hạnh phúc và tiếp tục thành công trong sự nghiệp giáo dục.

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc213218174)

[MỤC LỤC 2](#_Toc213218175)

[1. Giới thiệu đề tài. 1](#_Toc213218176)

[1.1. Đặt vấn đề. 1](#_Toc213218177)

[1.2. Mục tiêu đề tài. 1](#_Toc213218178)

[1.3. Đối tượng và phạm vi nghiên cứu. 2](#_Toc213218179)

[2. Tìm hiểu lý thuyết. 3](#_Toc213218180)

[2.1. Tổng quan về hệ thống thông minh. 3](#_Toc213218181)

[2.1.1. Định nghĩa. 3](#_Toc213218182)

[2.1.2. Các đặc trưng cơ bản của một Hệ thống thông minh. 3](#_Toc213218183)

[2.1.3. Vai trò và ứng dụng trong thực tiễn. 4](#_Toc213218184)

[2.2. Giới thiệu về Học máy (Machine Learning). 5](#_Toc213218185)

[2.2.1. Khái niệm và mối quan hệ với Hệ thống thông minh. 5](#_Toc213218186)

[2.2.2. Quy trình hoạt động của một mô hình Học máy. 6](#_Toc213218187)

[2.2.3. Phân loại các bài toán học máy. 7](#_Toc213218188)

[2.3. Hệ thống gợi ý (Recommendation Systems). 8](#_Toc213218189)

[2.3.1. Định nghĩa và bài toán cốt lõi. 8](#_Toc213218190)

[2.3.2. Phân loại chi tiết các phương pháp gợi ý. 9](#_Toc213218191)

[2.4. Các thuật toán gợi ý nền tảng. 13](#_Toc213218192)

[2.4.1. Lọc cộng tác dựa trên mục (Item-Based Collaborative Filtering - IBCF) 13](#_Toc213218193)

[2.4.2. Lọc Cộng tác và Thuật toán Phân rã Ma trận (ALS). 14](#_Toc213218194)

[2.4.3. Lọc Nội dung và các Thuật toán Xử lý Văn bản. 16](#_Toc213218195)

[2.5. Các độ đo đánh giá hệ thống gợi ý. 17](#_Toc213218196)

[2.5.1. Khái niệm về đánh giá Offline. 18](#_Toc213218197)

[2.5.2. Các độ đo xếp hạng cho Top-N (Top-N Ranking Metrics). 18](#_Toc213218198)

[3. Phân tích và thiết kế hệ thống. 20](#_Toc213218199)

[3.1. Phân tích yêu cầu nghiệp vụ. 20](#_Toc213218200)

[3.1.1. Các tác nhân. 20](#_Toc213218201)

[3.1.2. Yêu cầu chức năng theo nghiệp vụ. 23](#_Toc213218202)

[3.2. Phân tích yêu cầu hệ thống. 27](#_Toc213218203)

[3.2.1. Đặc tả chi tiết Use Case (Use Case Specifications). 27](#_Toc213218204)

[3.2.2. Sơ đồ Hoạt động (Activity Diagrams). 29](#_Toc213218205)

[3.3. Thiết kế Cơ sở dữ liệu. 32](#_Toc213218206)

[3.3.1. Mô tả chi tiết các bảng. 32](#_Toc213218207)

[3.4. Thiết kế kiến trúc hệ thống. 37](#_Toc213218208)

[3.4.1. Thiết kế API. 37](#_Toc213218209)

[3.5. Thiết kế chi tiết module gợi ý. 41](#_Toc213218210)

[3.5.1. Xác định nguồn dữ liệu cho Hệ thống Gợi ý. 41](#_Toc213218211)

[3.5.2. Mô hình hóa Dữ liệu Sự kiện. 41](#_Toc213218212)

[Lựa chọn Phương pháp và Thuật toán Gợi ý. 44](#_Toc213218213)

[3.6. Thiết kế giao diện. 45](#_Toc213218214)

[4. Triển khai hệ thống. 47](#_Toc213218215)

[4.1. Yêu cầu hệ thống. 47](#_Toc213218216)

[4.2. Cài đặt và chạy 47](#_Toc213218217)

[5. Kết luận. 52](#_Toc213218218)

# Giới thiệu đề tài.

# Đặt vấn đề.

Trong kỷ nguyên số, thương mại điện tử đã trở thành một chiến trường cạnh tranh khốc liệt, đặc biệt trong ngành hàng thiết bị điện tử với sự đa dạng và tốc độ cập nhật sản phẩm chóng mặt. Người dùng thường bị "nhấn chìm" trong vô số lựa chọn, trong khi doanh nghiệp phải đối mặt với thách thức làm thế nào để hiểu và đáp ứng chính xác nhu cầu cá nhân của từng khách hàng.

Đây chính là bối cảnh lý tưởng để ứng dụng các **hệ thống thông minh (Intelligent Systems)**, một lĩnh vực cốt lõi được đề cập trong môn học. Các hệ thống này, đặc biệt là hệ thống gợi ý (Recommendation Systems), sử dụng các thuật toán học máy để phân tích dữ liệu và tự động đưa ra những đề xuất giá trị, giải quyết hiệu quả bài toán cá nhân hóa trải nghiệm người dùng.

Vì vậy, việc lựa chọn đề tài **"Phát triển Hệ thống gợi ý thông minh cho website bán thiết bị điện tử"** không chỉ giải quyết một vấn đề thực tiễn cấp thiết, mà còn là cơ hội tuyệt vời để vận dụng và đào sâu các kiến thức, kỹ năng đã được học trong môn "Phát triển các Hệ thống thông minh" vào một dự án cụ thể.

# Mục tiêu đề tài.

Mục tiêu trọng tâm của đồ án môn học này là thực hiện một chu trình hoàn chỉnh trong việc **phát triển một hệ thống thông minh**, từ khâu lên ý tưởng đến đánh giá sản phẩm. Các mục tiêu cụ thể bao gồm:

**Vận dụng kiến thức môn học:** Áp dụng các khái niệm về hệ thống thông minh, học máy để phân tích bài toán gợi ý sản phẩm trong lĩnh vực thương mại điện tử.

**Nghiên cứu và lựa chọn mô hình:** Tìm hiểu, so sánh các kỹ thuật gợi ý phổ biến (Lọc cộng tác, Lọc dựa trên nội dung, Hybrid) và lựa chọn thuật toán phù hợp với đặc thù dữ liệu và yêu cầu của bài toán.

**Xây dựng hệ thống:**

Thiết kế kiến trúc và luồng xử lý dữ liệu cho một hệ thống gợi ý.

Tiến hành cài đặt (implement) mô hình gợi ý bằng ngôn ngữ lập trình Python và các thư viện học máy thông dụng (ví dụ: Scikit-learn, Pandas).

Xây dựng một ứng dụng web đơn giản để mô phỏng và trình diễn khả năng của hệ thống.

**Đánh giá hiệu năng:** Thiết lập phương pháp và sử dụng các độ đo (metrics) chuẩn để đánh giá mức độ chính xác và hiệu quả của mô hình đã xây dựng.

# Đối tượng và phạm vi nghiên cứu.

* **Ý nghĩa đối với môn học:** Đề tài là bài tập thực hành tổng hợp, giúp sinh viên chứng minh khả năng áp dụng lý thuyết vào thực tế. Nó bao quát toàn bộ vòng đời phát triển một hệ thống thông minh: từ phân tích yêu cầu, lựa chọn mô hình, xử lý dữ liệu, lập trình cho đến kiểm thử và đánh giá.
* **Ý nghĩa thực tiễn:** Kết quả của đề tài có thể được xem như một mô hình tham khảo, là tiền đề để phát triển các hệ thống gợi ý phức tạp hơn, giúp các doanh nghiệp vừa và nhỏ cải thiện trải nghiệm khách hàng và tăng hiệu quả kinh doanh.

Đối tượng nghiên cứu:

* Quy trình phát triển một hệ thống thông minh.
* Các mô hình học máy ứng dụng trong hệ thống gợi ý.
* Dữ liệu hành vi người dùng (user-item interactions) trên các nền tảng thương mại điện tử.

Phạm vi nghiên cứu:

* Về thuật toán: Tập trung vào việc cài đặt và thử nghiệm một hoặc hai thuật toán gợi ý cơ bản (ví dụ: User-based/Item-based Collaborative Filtering).
* Về dữ liệu: Sử dụng một bộ dữ liệu có sẵn (public dataset) về đánh giá sản phẩm điện tử hoặc một bộ dữ liệu mô phỏng.
* Về sản phẩm: Đồ án là một sản phẩm prototype, tập trung vào chức năng lõi là gợi ý, không đi sâu vào các tính năng hoàn chỉnh của một website thương mại điện tử.

# Tìm hiểu lý thuyết.

# Tổng quan về hệ thống thông minh.

# Định nghĩa.

Hệ thống thông minh (Intelligent System) là một hệ thống máy tính có khả năng mô phỏng các khía cạnh của trí tuệ con người. Thay vì chỉ thực thi các chỉ lệnh được lập trình sẵn một cách máy móc, hệ thống thông minh được thiết kế để có thể tự học hỏi từ dữ liệu, suy luận để đưa ra quyết định, và thích nghi với môi trường thay đổi.

Về bản chất, hệ thống thông minh là một ứng dụng thực tiễn của Trí tuệ nhân tạo (Artificial Intelligence - AI). Nó sử dụng các kỹ thuật và thuật toán của AI để giải quyết các bài toán phức tạp mà các hệ thống thông thường khó có thể xử lý hiệu quả, đặc biệt là các bài toán đòi hỏi khả năng nhận thức, phân tích và phán đoán.

# Các đặc trưng cơ bản của một Hệ thống thông minh.

Một hệ thống được xem là "thông minh" khi nó sở hữu một hoặc nhiều các đặc trưng sau:

* Khả năng học hỏi (Learning): Đây là đặc trưng quan trọng nhất. Hệ thống có thể tự động cải thiện hiệu năng của mình thông qua kinh nghiệm hoặc dữ liệu thu thập được mà không cần sự can thiệp trực tiếp của con người. Công cụ chính để thực hiện khả năng này là Học máy (Machine Learning).
* Khả năng suy luận (Reasoning): Hệ thống có khả năng sử dụng các tri thức, quy tắc đã biết để đưa ra các kết luận mới, phán đoán hoặc dự báo cho các tình huống chưa từng gặp.
* Khả năng thích nghi (Adaptation): Hệ thống có thể tự điều chỉnh hành vi của mình để phù hợp với sự thay đổi của môi trường hoặc yêu cầu đầu vào.
* Xử lý thông tin không chắc chắn: Dữ liệu trong thế giới thực thường không đầy đủ, nhiễu và không chắc chắn. Hệ thống thông minh có khả năng làm việc hiệu quả với các loại dữ liệu này, thường thông qua các mô hình xác suất.
* Tương tác tự nhiên: Nhiều hệ thống thông minh có khả năng giao tiếp và tương tác với con người một cách tự nhiên, ví dụ như qua ngôn ngữ nói hoặc hình ảnh.

# Vai trò và ứng dụng trong thực tiễn.

Hệ thống thông minh đang ngày càng đóng vai trò then chốt trong mọi mặt của đời sống và công nghệ. Các ứng dụng của nó không chỉ giúp tự động hóa các tác vụ lặp đi lặp lại mà còn giải quyết được những vấn đề mang tính nhận thức cao. Một số ví dụ tiêu biểu bao gồm:

* Trợ lý ảo: Siri của Apple, Google Assistant, Amazon Alexa có khả năng hiểu và phản hồi ngôn ngữ tự nhiên của con người.
* Xe tự hành: Sử dụng các cảm biến và thuật toán AI để nhận diện môi trường, phân tích tình huống giao thông và tự động ra quyết định điều khiển.
* Nhận dạng hình ảnh và giọng nói: Ứng dụng trong việc mở khóa bằng khuôn mặt (Face ID), phân loại ảnh tự động, chuyển đổi giọng nói thành văn bản.
* Chẩn đoán y khoa: Các hệ thống thông minh có thể phân tích hình ảnh y tế (X-quang, MRI) để phát hiện sớm các dấu hiệu bệnh lý với độ chính xác cao.

Một trong những ứng dụng thương mại thành công và phổ biến nhất của hệ thống thông minh chính là Hệ thống gợi ý (Recommendation Systems). Hệ thống này thể hiện đầy đủ các đặc trưng của một hệ thống thông minh: nó học hỏi từ hành vi (lượt xem, lượt mua) của người dùng, suy luận ra sở thích tiềm ẩn của họ, và thích nghi bằng cách liên tục cập nhật các gợi ý cho phù hợp. Chính vì vậy, việc phát triển một hệ thống gợi ý là một bài toán điển hình và tiêu biểu trong lĩnh vực phát triển các hệ thống thông minh.

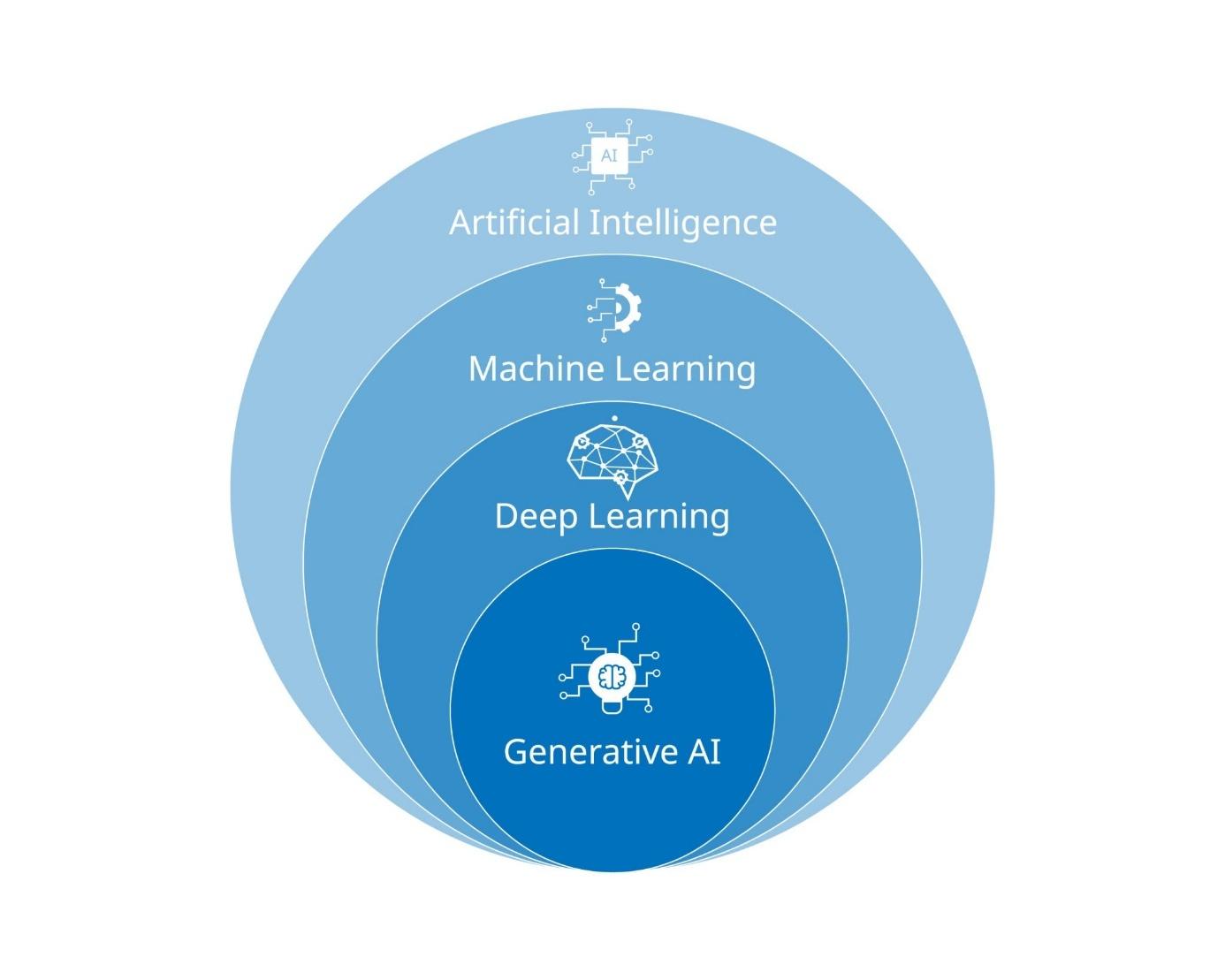
# Giới thiệu về Học máy (Machine Learning).

Học máy không chỉ là một công cụ, mà là động cơ (engine) cung cấp khả năng học hỏi cho các hệ thống thông minh. Nó là tập hợp các phương pháp và thuật toán cho phép máy tính tự động trích xuất các quy luật (patterns) từ dữ liệu, từ đó đưa ra các dự đoán hoặc quyết định mà không cần được lập trình một cách tường minh cho từng trường hợp cụ thể

# Khái niệm và mối quan hệ với Hệ thống thông minh.

Học máy là một lĩnh vực con của Trí tuệ nhân tạo (AI), tập trung vào việc nghiên cứu và xây dựng các kỹ thuật cho phép máy tính có thể "học" từ dữ liệu. Thay vì được lập trình một cách tường minh để thực hiện một tác vụ, máy tính sẽ tự xây dựng và cải thiện một mô hình (model) dựa trên các mẫu (patterns) và tri thức mà nó tự rút ra từ dữ liệu đầu vào.

Mối quan hệ giữa Học máy và Hệ thống thông minh là mối quan hệ giữa công cụ và sản phẩm. Nếu Hệ thống thông minh là một ngôi nhà, thì Học máy chính là bộ công cụ (búa, cưa, máy khoan) và các kỹ thuật xây dựng giúp tạo nên ngôi nhà đó. Cụ thể, Học máy cung cấp cơ chế học hỏi, một trong những đặc tính cốt lõi và quan trọng nhất của một hệ thống thông minh.



# Quy trình hoạt động của một mô hình Học máy.

Ta có thể hình dung một mô hình học máy giống như một "học sinh". Quy trình "học" của nó thường bao gồm các bước cơ bản sau:

**Dữ liệu (Data):** "Học sinh" cần có "sách giáo khoa", đó chính là dữ liệu. Dữ liệu này chứa các thông tin đầu vào và trong nhiều trường hợp, cả kết quả mong muốn.

**Đặc trưng (Features):** Là các thuộc tính, các cột thông tin trong bộ dữ liệu được sử dụng để mô tả một đối tượng. Ví dụ, đối với một sản phẩm điện thoại, các đặc trưng có thể là: thương hiệu, giá, kích thước màn hình, dung lượng pin.

**Huấn luyện (Training):** "Học sinh" sẽ "đọc sách" và cố gắng tìm ra mối liên hệ giữa các đặc trưng đầu vào và kết quả đầu ra. Quá trình này được gọi là huấn luyện mô hình. Kết quả của quá trình này là một **mô hình (model)** – một cấu trúc toán học đã "học" được các quy luật từ dữ liệu.

**Dự đoán (Prediction):** Sau khi đã được huấn luyện, "học sinh" có thể "làm bài kiểm tra". Khi nhận được một dữ liệu mới (chưa có kết quả), mô hình sẽ áp dụng những quy luật đã học để đưa ra dự đoán.

# Phân loại các bài toán học máy.

Các thuật toán Học máy thường được phân loại thành ba nhóm chính dựa trên cách chúng "học" từ dữ liệu:

**Học có giám sát (Supervised Learning):**

* Đây là phương pháp học phổ biến nhất. Mô hình sẽ học từ một bộ dữ liệu đã được "gán nhãn" (labeled data), nghĩa là mỗi điểm dữ liệu đầu vào đều đi kèm với một kết quả đầu ra chính xác.
* Mục tiêu: Học ra một "quy luật" tổng quát để có thể dự đoán kết quả đầu ra cho các dữ liệu mới chưa từng thấy.
* Ví dụ:
  + Phân loại (Classification): Dự đoán email là spam hay không spam.
  + Hồi quy (Regression): Dự đoán giá của một ngôi nhà dựa trên diện tích, số phòng ngủ.

**Học không giám sát (Unsupervised Learning):**

* Với phương pháp này, mô hình sẽ làm việc với bộ dữ liệu "không có nhãn". Nhiệm vụ của nó là tự khám phá ra các cấu trúc, quy luật hay các cụm (clusters) tiềm ẩn trong dữ liệu.
* Mục tiêu: Tìm ra sự tương đồng và các mẫu ẩn trong dữ liệu.
* Ví dụ:
  + Phân cụm (Clustering): Phân nhóm khách hàng có cùng hành vi mua sắm.
  + Luật kết hợp (Association Rule Mining): Tìm ra các sản phẩm thường được mua cùng nhau (ví dụ: "khách hàng mua bánh mì thường mua cả sữa").
  + Mức độ liên quan: Đây là nhánh học máy cực kỳ quan trọng đối với các hệ thống gợi ý, vì các thuật toán như Lọc cộng tác (Collaborative Filtering) về bản chất là một bài toán học không giám sát, tự tìm ra các nhóm người dùng hoặc nhóm sản phẩm tương tự nhau.

**Học củng cố (Reinforcement Learning):**

* Đây là phương pháp học mô phỏng cách con người học thông qua thử và sai. Một "tác nhân" (agent) sẽ học cách hành động trong một "môi trường" (environment) để tối đa hóa một "phần thưởng" (reward) nào đó.
* Mục tiêu: Tìm ra một chuỗi hành động tối ưu (chính sách) để đạt được mục tiêu.
  + Ví dụ: Huấn luyện AI chơi game, điều khiển robot tự hành, tối ưu hóa chuỗi cung ứng.

# Hệ thống gợi ý (Recommendation Systems).

Sau khi đã có nền tảng về Học máy, chúng ta sẽ đi sâu vào lĩnh vực ứng dụng trọng tâm của đề tài. Hệ thống gợi ý không chỉ là một ứng dụng đơn lẻ mà là một lĩnh vực nghiên cứu quan trọng, là cầu nối giữa dữ liệu hành vi người dùng và giá trị kinh doanh, thể hiện một cách rõ ràng nhất sức mạnh của các hệ thống thông minh.

# Định nghĩa và bài toán cốt lõi.

Hệ thống gợi ý (Recommendation System) là một lớp của các hệ thống lọc thông tin, có nhiệm vụ xử lý và khai thác dữ liệu đầu vào (như hành vi người dùng, thuộc tính sản phẩm) để đưa ra những đề xuất phù hợp và được cá nhân hóa cho từng người dùng.

Bài toán cốt lõi mà một hệ thống gợi ý cần giải quyết có thể được phát biểu như sau: "Với một tập hợp người dùng U và một tập hợp sản phẩm I, cùng với dữ liệu về sự tương tác giữa người dùng và sản phẩm (ví dụ: ma trận đánh giá R), hãy dự đoán giá trị của những tương tác chưa xảy ra, từ đó đề xuất cho người dùng u những sản phẩm i mà họ có khả năng sẽ yêu thích nhất."

Dữ liệu tương tác này thường được biểu diễn dưới dạng một ma trận User-Item, trong đó các hàng đại diện cho người dùng, các cột đại diện cho sản phẩm, và giá trị tại ô (u, i) là đánh giá (rating) của người dùng u cho sản phẩm i. Ma trận này thường cực kỳ thưa thớt (sparse) vì một người dùng chỉ tương tác với một phần rất nhỏ trong tổng số sản phẩm.

# Phân loại chi tiết các phương pháp gợi ý.

Các phương pháp xây dựng hệ thống gợi ý rất đa dạng, mỗi loại có ưu và nhược điểm riêng, phù hợp với các bối cảnh bài toán khác nhau.

1. Lọc dựa trên sự phổ biến (Popularity-Based Filtering).

Đây là phương pháp tiếp cận đơn giản nhất, hoạt động như một baseline để so sánh với các mô hình phức tạp hơn.

* Nguyên lý: Hệ thống sẽ xếp hạng tất cả các sản phẩm dựa trên một độ đo phổ biến nào đó (ví dụ: số lượt mua, số lượt xem, điểm đánh giá trung bình cao nhất) và đề xuất top-N sản phẩm đứng đầu cho tất cả người dùng.
* Ưu điểm: Cực kỳ dễ triển khai, không đòi hỏi dữ liệu lịch sử của người dùng cụ thể, do đó không gặp phải vấn đề "khởi đầu lạnh" (cold start) đối với người dùng mới.
* Nhược điểm: Hoàn toàn không có tính cá nhân hóa. Gợi ý mang tính đại chúng và bỏ qua sở thích riêng của từng cá nhân. Nó cũng có xu hướng tạo ra "vòng lặp phổ biến" (popularity bias), khiến các sản phẩm đã nổi tiếng lại càng nổi tiếng hơn.

1. Lọc dựa trên nội dung (Content-Based Filtering).

Phương pháp này đưa ra gợi ý dựa trên nguyên tắc: "Nếu bạn đã thích một sản phẩm, bạn cũng sẽ thích những sản phẩm khác có đặc tính tương tự."

* Nguyên lý: Hệ thống tập trung vào việc phân tích thuộc tính (attributes) của các sản phẩm. Ví dụ, đối với một chiếc laptop, thuộc tính có thể là: thương hiệu=Dell, CPU=Core i7, RAM=16GB, loại=gaming.
* Quy trình hoạt động:
  1. Trích xuất đặc trưng sản phẩm (Item Feature Extraction): Biến các thuộc tính của sản phẩm thành một vector số học (item profile). Các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) như TF-IDF thường được dùng để xử lý các thuộc tính dạng văn bản (mô tả sản phẩm).
  2. Xây dựng hồ sơ người dùng (User Profile Construction): Dựa trên lịch sử tương tác của người dùng (ví dụ: các sản phẩm đã mua, đã đánh giá 5 sao), hệ thống sẽ xây dựng một hồ sơ sở thích của người dùng đó. Về cơ bản, hồ sơ này là một phiên bản tổng hợp hoặc trung bình của các vector đặc trưng của những sản phẩm mà người dùng yêu thích.
  3. Gợi ý: Hệ thống tính toán độ tương đồng (ví dụ: dùng độ tương đồng Cosine) giữa vector hồ sơ người dùng và vector đặc trưng của tất cả các sản phẩm mà người dùng chưa tương tác. Các sản phẩm có độ tương đồng cao nhất sẽ được gợi ý.
* Ưu điểm:
  + Giải quyết vấn đề khởi đầu lạnh cho sản phẩm mới: Một sản phẩm mới chỉ cần có mô tả là có thể được đưa vào hệ thống và gợi ý ngay lập tức.
  + Gợi ý có tính giải thích cao: Dễ dàng giải thích cho người dùng tại sao một sản phẩm được gợi ý ("Vì bạn đã thích laptop Dell, chúng tôi gợi ý laptop Dell khác cho bạn").
* Nhược điểm:
  + Hạn chế về tính mới lạ (Limited Serendipity): Hệ thống có xu hướng "đóng khung" người dùng trong một vòng lặp sở thích, khó gợi ý được những sản phẩm thuộc lĩnh vực hoàn toàn mới mà người dùng có thể sẽ thích.
  + Đòi hỏi kỹ thuật trích xuất đặc trưng phức tạp: Chất lượng gợi ý phụ thuộc rất nhiều vào việc các thuộc tính sản phẩm có được mô tả và xử lý tốt hay không.

1. Lọc cộng tác (Collaborative Filtering - CF).

* Đây là phương pháp được sử dụng rộng rãi và thành công nhất, là trái tim của nhiều hệ thống gợi ý khổng lồ như Amazon, Netflix.
* Nguyên lý: Phương pháp này khai thác "trí tuệ đám đông" dựa trên giả định cốt lõi: "Những người dùng đồng ý trong quá khứ có khả năng sẽ tiếp tục đồng ý trong tương lai.". Nó không cần biết gì về thuộc tính sản phẩm, mà chỉ dựa hoàn toàn vào ma trận tương tác User-Item.
* Các cách tiếp cận chính:
  + CF dựa trên bộ nhớ (Memory-based CF): Phương pháp này sử dụng toàn bộ ma trận User-Item để tính toán độ tương đồng và đưa ra dự đoán trực tiếp.
    - Dựa trên người dùng (User-based CF): Để dự đoán đánh giá của người dùng A cho sản phẩm X, hệ thống sẽ: 1) Tìm ra top-K người dùng có "khẩu vị" giống A nhất (những người đã đánh giá các sản phẩm khác tương tự A). 2) Lấy trung bình có trọng số các đánh giá của nhóm người dùng này cho sản phẩm X để làm dự đoán cho A.
    - Dựa trên sản phẩm (Item-based CF): Để dự đoán đánh giá của người dùng A cho sản phẩm X, hệ thống sẽ: 1) Tìm ra top-K sản phẩm giống với X nhất (dựa trên việc chúng được cùng một nhóm người dùng đánh giá tương tự nhau). 2) Lấy trung bình có trọng số các đánh giá mà người dùng A đã dành cho các sản phẩm tương tự đó để làm dự đoán cho X. Đây là phương pháp thường hiệu quả và ổn định hơn User-based CF trong thực tế.
  + CF dựa trên mô hình (Model-based CF): Thay vì dùng toàn bộ ma trận, phương pháp này xây dựng một mô hình học máy để "học" và "nén" thông tin từ ma trận đó.
    - Kỹ thuật tiêu biểu - Phân rã ma trận (Matrix Factorization): Kỹ thuật này giả định rằng sở thích của người dùng và đặc tính của sản phẩm có thể được biểu diễn trong một không gian "đặc trưng ẩn" (latent features) có số chiều thấp hơn. Nó sẽ phân rã ma trận User-Item R (kích thước m x n) thành hai ma trận P (user-features, m x k) và Q (item-features, n x k), sao cho R ≈ P x Qᵀ. Các đặc trưng ẩn này có thể đại diện cho các yếu tố như "thể loại phim" (hành động, hài), "đạo diễn", "diễn viên" mà mô hình tự học được.
* Ưu điểm:
  + Khả năng tạo ra sự bất ngờ (Serendipity): Có thể gợi ý những sản phẩm hoàn toàn khác biệt với những gì người dùng đã tương tác, dựa trên sở thích của những người dùng tương đồng.
  + Không cần thông tin về sản phẩm: Hoạt động tốt ngay cả khi không có mô tả chi tiết về sản phẩm.
* Nhược điểm:
  + Vấn đề khởi đầu lạnh (Cold Start): Gặp khó khăn nghiêm trọng với người dùng mới (không có lịch sử) và sản phẩm mới (không có ai đánh giá).
  + Dữ liệu thưa thớt (Sparsity): Khi ma trận tương tác quá thưa, việc tìm kiếm sự tương đồng trở nên khó khăn và kém tin cậy.

1. Phương pháp lai (Hybrid Methods).

Nguyên lý: "Không có phương pháp nào là tốt nhất cho mọi trường hợp". Các hệ thống lai ra đời bằng cách kết hợp hai hay nhiều kỹ thuật gợi ý để tận dụng ưu điểm và hạn chế nhược điểm của từng phương pháp.

Ví dụ: Netflix kết hợp thuật toán phân rã ma trận (CF) với các mô hình phân tích thuộc tính phim (Content-based) để đưa ra gợi ý cuối cùng. Một cách tiếp cận phổ biến là sử dụng Content-based để giải quyết vấn đề khởi đầu lạnh cho sản phẩm mới, và dùng CF cho các sản phẩm đã có đủ dữ liệu tương tác.

# Các thuật toán gợi ý nền tảng.

Phần này cung cấp một cái nhìn sâu sắc về cơ sở lý thuyết và cơ chế hoạt động của các thuật toán cốt lõi được lựa chọn để xây dựng module gợi ý. Việc hiểu rõ bản chất của các thuật toán này là nền tảng cho các quyết định thiết kế và triển khai ở Chương 3.

# Lọc cộng tác dựa trên mục (Item-Based Collaborative Filtering - IBCF)

**Lọc cộng tác (Collaborative Filtering - CF)** là một trong những họ kỹ thuật thành công và phổ biến nhất trong các hệ thống gợi ý. Nguyên lý chung của nó là dựa vào "trí tuệ đám đông" để đưa ra dự đoán, bằng cách thu thập và phân tích hành vi (ví dụ: đánh giá, lượt mua, lượt xem) từ một tập hợp lớn người dùng.

Họ Lọc cộng tác được chia làm hai hướng tiếp cận chính:

1. **User-Based CF (Lọc cộng tác dựa trên người dùng):** Tìm những người dùng có "khẩu vị" *tương tự* như bạn và gợi ý những mục (item) mà họ thích.
2. **Item-Based CF (Lọc cộng tác dựa trên mục):** Tìm những mục *tương tự* với những mục mà bạn đã thích trong quá khứ và gợi ý chúng.

**Item-Based Collaborative Filtering (IBCF)**, được tiên phong và phổ biến bởi Amazon.com, hoạt động dựa trên nguyên lý cốt lõi:

"Một người dùng sẽ có xu hướng thích một mục, nếu mục đó **tương tự** với các mục khác mà người dùng đó đã thích."

Ví dụ: Nếu hệ thống phát hiện rằng phần lớn những người mua "Harry Potter - Tập 1" cũng mua "Harry Potter - Tập 2", nó sẽ xác định hai mục này có độ tương đồng cao. Khi một người dùng mới mua "Tập 1", hệ thống sẽ ngay lập tức gợi ý "Tập 2".

Thuật toán này tập trung vào việc xây dựng mối quan hệ tương đồng giữa các **mục (item)**, thay vì giữa các **người dùng (user)**.

Quy trình của IBCF được chia thành hai giai đoạn rõ rệt:

**Giai đoạn Offline: Xây dựng Ma trận Tương đồng (Item-Item Similarity Matrix)**

Đây là bước tính toán nặng, được thực hiện định kỳ (ví dụ: hàng đêm) và lưu lại để sử dụng sau.

1. **Đầu vào:** Ma trận Tương tác Người dùng-Mục (User-Item Interaction Matrix). Đây là một ma trận thưa (sparse matrix) khổng lồ, nơi các hàng là người dùng u, các cột là mục i, và giá trị ru,i là đánh giá (rating) của người dùng u cho mục i.
2. **Quá trình:** Với mỗi cặp mục (item i, item j), hệ thống sẽ tìm tất cả những người dùng u đã đánh giá *cả hai* mục này.
3. **Tính toán:** Dựa trên tập đánh giá chung này, một "hệ số tương đồng" (similarity score) si, j được tính toán cho cặp (i, j).
4. **Đầu ra:** Một ma trận vuông Tương đồng Mục-Mục (Item-Item Similarity Matrix), nơi si,j biểu thị mức độ tương đồng giữa item I và item j.

**Giai đoạn Online: Tạo dự đoán (Prediction Generation)**

Đây là bước được thực hiện "ngay lập tức" (real-time) khi một người dùng u yêu cầu gợi ý hoặc truy cập một trang sản phẩm i.

1. **Mục tiêu:** Dự đoán đánh giá PU,I (mà người dùng u có thể sẽ đưa ra) cho mục i (mà u chưa tương tác).
2. **Thu thập:** Hệ thống lấy ra tất cả các mục j mà người dùng u đã đánh giá (ru,j).
3. **Tra cứu:** Từ ma trận tương đồng (đã tính ở bước offline), hệ thống lấy ra các hệ số tương đồng si,j(giữa mục i và các mục j).
4. **Dự đoán:** Hệ thống tính toán PU,I bằng cách lấy **trung bình có trọng số** (weighted average) của các đánh giá ru,j mà u đã thực hiện. Trọng số chính là độ tương đồng si,j.

# Lọc Cộng tác và Thuật toán Phân rã Ma trận (ALS).

1. Ý tưởng cốt lõi của Lọc Cộng tác.

Lọc cộng tác (Collaborative Filtering - CF) là phương pháp gợi ý mạnh mẽ nhất, hoạt động dựa trên nguyên tắc "trí tuệ đám đông". Giả định cơ bản là nếu hai người dùng có những hành vi tương đồng trong quá khứ (ví dụ: cùng mua, cùng xem, hoặc cùng đánh giá cao những sản phẩm giống nhau), thì họ có khả năng sẽ tiếp tục có sở thích tương đồng trong tương lai.

1. Phân rã Ma trận (Matrix Factorization).

Thay vì tìm kiếm sự tương đồng trực tiếp giữa hàng triệu người dùng, các kỹ thuật Phân rã Ma trận (MF) hiện đại tiếp cận vấn đề bằng cách học các đặc trưng ẩn (latent features). Ý tưởng là sở thích của người dùng và các đặc tính của sản phẩm có thể được biểu diễn trong một không gian vector có số chiều thấp hơn.

Ví dụ, sở thích của một người dùng đối với một chiếc điện thoại có thể được quyết định bởi các yếu tố ẩn như: "mức độ ưa chuộng hệ sinh thái Apple", "sự quan tâm đến khả năng nhiếp ảnh", "nhu cầu về thời lượng pin", và "sự nhạy cảm về giá". Tương tự, mỗi chiếc điện thoại cũng có các đặc trưng ẩn tương ứng với những yếu tố này.

Về mặt toán học, MF cố gắng phân rã ma trận tương tác R (kích thước m người dùng × n sản phẩm) thành tích của hai ma trận có số chiều thấp hơn: ma trận đặc trưng người dùng P (kích thước m × k) và ma trận đặc trưng sản phẩm Q (kích thước n × k).​

Trong đó, k là số lượng đặc trưng ẩn, một siêu tham số (hyperparameter) quan trọng cần được lựa chọn. Mỗi hàng trong P là một vector đại diện cho một người dùng, và mỗi hàng trong Q là một vector đại diện cho một sản phẩm trong không gian k chiều đó.

1. Thuật toán ALS(Alternating Least Squares).

ALS là một thuật toán tối ưu lặp đi lặp lại để tìm ra hai ma trận P và Q. Mục tiêu của nó là tối thiểu hóa tổng bình phương sai số giữa các đánh giá thực tế và các đánh giá dự đoán (tích của vector người dùng và vector sản phẩm).

Hàm mục tiêu có dạng:

Trong đó:

là tập hợp các cặp(user, item) đã có đánh giá.

là đánh giá thực tế của người dùng u cho sản phẩm i.

là vector đặc trưng cho người dùng *u* (một hàng của **P**).

là vector đặc trưng cho sản phẩm *i* (một hàng của **Q**).

là tham số điều chuẩn (regularization parameter) để tránh hiện tượng quá khớp (overfitting).

Tên gọi "Bình phương Tối thiểu Luân phiên" mô tả chính xác cách thuật toán hoạt động:

1. Khởi tạo: Khởi tạo ngẫu nhiên ma trận P và Q.
2. Cố định P, Tối ưu Q: Giữ nguyên các vector đặc trưng người dùng, thuật toán sẽ tìm ra các vector đặc trưng sản phẩm (Q) tối ưu nhất để giảm thiểu hàm lỗi.
3. Cố định Q, Tối ưu P: Ngược lại, giữ nguyên các vector Q vừa tính được, thuật toán sẽ tìm ra các vector người dùng (P) tối ưu nhất.
4. Lặp lại: Quá trình 2 và 3 được lặp lại luân phiên cho đến khi mô hình hội tụ (khi hàm lỗi không còn giảm đáng kể).

Ưu điểm lớn nhất của ALS là các bước tính toán trong mỗi vòng lặp có thể được song song hóa một cách hiệu quả, khiến nó trở thành lựa chọn hàng đầu cho việc triển khai trên các nền tảng xử lý dữ liệu phân tán như Apache Spark.

# Lọc Nội dung và các Thuật toán Xử lý Văn bản.

1. Ý tưởng cốt lõi của Lọc Nội dung

Lọc dựa trên Nội dung (Content-Based Filtering) đưa ra gợi ý dựa trên nguyên tắc: "Gợi ý các sản phẩm có thuộc tính tương tự với những sản phẩm mà người dùng đã thích trước đây." Để làm được điều này, hệ thống cần một phương pháp để "đọc" và "so sánh" nội dung của các sản phẩm.

1. TF-IDF (Term Frequency - Inverse Document Frequency)

Để máy tính có thể so sánh các mô tả sản phẩm, chúng ta cần chuyển đổi chúng từ dạng văn bản sang dạng vector số học. TF-IDF là một kỹ thuật trọng số thống kê mạnh mẽ để thực hiện việc này. Nó đánh giá tầm quan trọng của một từ (term) trong một tài liệu (document) cụ thể, trong bối cảnh của toàn bộ kho tài liệu.

Term Frequency (TF): Đo lường tần suất xuất hiện của một từ trong một tài liệu. Từ nào xuất hiện càng nhiều thì càng có khả năng mô tả rõ hơn về tài liệu đó.

Inverse Document Frequency (IDF): Đo lường mức độ "hiếm" của một từ trên toàn bộ tập tài liệu. Những từ xuất hiện trong nhiều tài liệu (ví dụ: "và", "của", "tốt") sẽ bị coi là ít quan trọng hơn, trong khi những từ đặc trưng chỉ xuất hiện trong một vài tài liệu (ví dụ: "Snapdragon", "OLED") sẽ được coi là quan trọng hơn.

Điểm số TF-IDF cuối cùng là tích của hai giá trị trên. Sau khi áp dụng, mỗi mô tả sản phẩm sẽ được biểu diễn bằng một vector mà mỗi chiều của nó là điểm TF-IDF của một từ trong từ điển.

1. Độ tương đồng Cosine (Cosine Similarity).

Sau khi đã có các vector sản phẩm, chúng ta cần một phương pháp để đo lường mức độ tương đồng giữa chúng. Độ tương đồng Cosine là một lựa chọn phổ biến và hiệu quả. Thay vì đo khoảng cách Euclid, nó đo lường giá trị cosine của góc (θ) giữa hai vector trong không gian đa chiều.

Kết quả của độ tương đồng Cosine luôn nằm trong khoảng từ 0 đến 1. Một giá trị gần 1 cho thấy hai sản phẩm có nội dung rất tương đồng (góc giữa các vector rất nhỏ), trong khi một giá trị gần 0 cho thấy chúng ít liên quan.

# Các độ đo đánh giá hệ thống gợi ý.

Việc xây dựng một mô hình gợi ý là chưa đủ, chúng ta cần phải có phương pháp để đo lường và định lượng hiệu quả của nó. Mục này trình bày các độ đo (metrics) phổ biến và quan trọng, đóng vai trò là cơ sở để đánh giá và so sánh các thuật toán sẽ được triển khai ở chương sau.

# Khái niệm về đánh giá Offline.

Trong khuôn khổ của một đồ án, việc đánh giá thường được thực hiện offline. Quy trình này diễn ra như sau:

Phân chia dữ liệu: Bộ dữ liệu tương tác User-Item sẽ được chia thành hai phần:

Tập huấn luyện (Training set): Dùng để xây dựng (huấn luyện) mô hình gợi ý.

Tập kiểm thử (Test set): Dùng để đánh giá. Tập này chứa các tương tác mà mô hình chưa từng "nhìn thấy".

Dự đoán và so sánh: Mô hình sau khi được huấn luyện sẽ đưa ra các gợi ý. Các gợi ý này sau đó sẽ được so sánh với dữ liệu thực tế trong tập kiểm thử để tính toán các độ đo hiệu năng.

# Các độ đo xếp hạng cho Top-N (Top-N Ranking Metrics).

Trong thực tế, người dùng quan tâm nhất đến chất lượng của danh sách top-N (ví dụ: 10 sản phẩm hàng đầu) mà hệ thống đề xuất. Do đó, các độ đo sau đây là cực kỳ quan trọng.

1. Precision@k (Độ chính xác tại k).

* Câu hỏi trả lời: "Trong số k sản phẩm được gợi ý, tỷ lệ sản phẩm mà người dùng thực sự quan tâm là bao nhiêu?"
* Ý nghĩa: Đo lường mức độ "chính xác" của danh sách gợi ý. Precision cao có nghĩa là các sản phẩm được đề xuất hầu hết đều phù hợp.
* Công thức:
* Ví dụ: Nếu hệ thống gợi ý 10 sản phẩm (k=10) cho một người dùng, và trong số đó có 4 sản phẩm nằm trong danh sách các sản phẩm mà người dùng thực sự đã mua ở tập kiểm thử, thì:
* Precision@10 = 4 / 10 = 0.4

1. Recall@k (Độ phủ tại k).

* Câu hỏi trả lời: "Trong tổng số các sản phẩm mà người dùng thực sự quan tâm, hệ thống đã tìm thấy và gợi ý được bao nhiêu phần trăm trong top k?"
* Ý nghĩa: Đo lường khả năng "tìm lại" tất cả các sản phẩm phù hợp. Recall cao có nghĩa là hệ thống không bỏ sót nhiều sản phẩm mà người dùng sẽ thích.
* Công thức:

* Ví dụ: Giả sử người dùng đó đã mua tổng cộng 8 sản phẩm trong tập kiểm thử. Hệ thống gợi ý top 10 và tìm thấy được 4 sản phẩm trong số đó, thì:

Recall@10 = 4 / 8 = 0.5

**Lưu ý về Precision và Recall**: Hai độ đo này thường có sự đánh đổi. Một hệ thống cố gắng gợi ý thật nhiều để không bỏ sót (tăng Recall) thì có thể sẽ gợi ý cả những sản phẩm không liên quan (giảm Precision), và ngược lại.

1. F1-Score@k.

Ý nghĩa: Là một chỉ số duy nhất kết hợp cả Precision và Recall, giúp đưa ra một cái nhìn cân bằng hơn về hiệu suất của mô hình. F1-Score là trung bình điều hòa (harmonic mean) của hai chỉ số này, nó sẽ có giá trị cao chỉ khi cả Precision và Recall đều cao.

Công thức:

# Phân tích và thiết kế hệ thống.

# Phân tích yêu cầu nghiệp vụ.

# Các tác nhân.

1. Khách hàng (Customer) .

Mô tả: Đây là tác nhân trung tâm của hệ thống, là người dùng cuối cùng có nhu cầu tìm kiếm, khám phá và mua sắm các thiết bị điện tử. Toàn bộ trải nghiệm người dùng và các luồng kinh doanh đều xoay quanh việc phục vụ tác nhân này.

Mục tiêu chính:

* Tìm được sản phẩm đúng với nhu cầu và ngân sách một cách nhanh chóng.
* Nhận được các gợi ý sản phẩm phù hợp, khám phá các lựa chọn mới.
* Thực hiện quy trình mua hàng và thanh toán một cách an toàn, tiện lợi.
* Theo dõi được tình trạng đơn hàng và quản lý lịch sử mua sắm.

Các hành vi/kịch bản chính:

* Đăng ký tài khoản, đăng nhập, quản lý thông tin cá nhân.
* Sử dụng công cụ tìm kiếm và các bộ lọc (theo giá, thương hiệu, thông số kỹ thuật).
* Xem chi tiết sản phẩm, đọc mô tả, xem hình ảnh và đọc các đánh giá từ người dùng khác.
* Tương tác với hệ thống gợi ý (click vào các sản phẩm trong mục "Dành cho bạn", "Sản phẩm tương tự").
* Thêm sản phẩm vào giỏ hàng, tiến hành thanh toán.

Để lại đánh giá (rating) và bình luận (review) cho sản phẩm đã mua.

* Kỳ vọng đối với Hệ thống:
* Giao diện thân thiện, dễ sử dụng, tốc độ tải trang nhanh.
* Thông tin sản phẩm đầy đủ, chính xác.
* Trải nghiệm mua sắm được cá nhân hóa, cảm thấy hệ thống "hiểu" được mình.
* Bảo mật thông tin cá nhân và thông tin thanh toán.

1. Người bán (Seller).

Mô tả: Là các đối tác kinh doanh (cá nhân hoặc doanh nghiệp) sử dụng nền tảng để đăng tải và bán sản phẩm của họ. Đây là tác nhân quyết định sự đa dạng của sản phẩm và khả năng mở rộng của hệ thống.

Mục tiêu chính:

* Tiếp cận được lượng lớn khách hàng tiềm năng.
* Quản lý sản phẩm, đơn hàng và tồn kho một cách hiệu quả.
* Theo dõi hiệu quả kinh doanh thông qua các báo cáo, thống kê.
* Tối đa hóa doanh thu và lợi nhuận.

Các hành vi/kịch bản chính:

* Đăng ký gian hàng và được hệ thống phê duyệt.
* Đăng tải sản phẩm mới, cập nhật thông tin, giá cả và số lượng tồn kho.
* Xem và xử lý các đơn hàng mới (xác nhận đơn, đóng gói, giao hàng).
* Xem bảng điều khiển (dashboard) với các thống kê về doanh thu, số đơn hàng, sản phẩm bán chạy.
* Phản hồi các câu hỏi hoặc đánh giá của khách hàng về sản phẩm của mình.

Kỳ vọng đối với Hệ thống:

* Một "Kênh người bán" (Seller Center) có giao diện quản lý rõ ràng, dễ thao tác.
* Hệ thống xử lý đơn hàng và thanh toán minh bạch, chính xác và đúng hạn.
* Các công cụ phân tích dữ liệu trực quan để hỗ trợ ra quyết định kinh doanh.
* Cơ chế hỗ trợ nhanh chóng khi gặp sự cố.

1. Nhân viên (Staff).

Mô tả: Là nhân viên của công ty chủ quản nền tảng, chịu trách nhiệm vận hành hàng ngày để đảm bảo chất lượng và sự trơn tru của toàn bộ sàn thương mại điện tử.

Mục tiêu chính:

* Đảm bảo các sản phẩm được đăng bán tuân thủ tiêu chuẩn chất lượng và chính sách.
* Hỗ trợ các tác nhân khác (Khách hàng, Người bán) giải quyết các vấn đề phát sinh.
* Quản lý nội dung và các danh mục chung của website.

Các hành vi/kịch bản chính:

* Kiểm duyệt nội dung: Xem và phê duyệt hoặc từ chối các sản phẩm mới do Người bán đăng tải.
* Hỗ trợ người dùng: Tiếp nhận và xử lý các khiếu nại, yêu cầu hỗ trợ từ Khách hàng và Người bán.
* Quản lý danh mục: Tạo, sửa, xóa các danh mục sản phẩm (ví dụ: "Điện thoại", "Laptop", "Phụ kiện").

Kỳ vọng đối với Hệ thống:

* Các công cụ quản trị (admin tools) hiệu quả, quy trình rõ ràng để thực hiện công việc.
* Hệ thống phân quyền hợp lý, chỉ cho phép truy cập các chức năng cần thiết.

1. Quản trị viên (Administrator).

Mô tả: Là người dùng có quyền hạn cao nhất trong hệ thống, chịu trách nhiệm về cả mặt chiến lược kinh doanh và quản trị kỹ thuật tổng thể. Trong giai đoạn đầu, vai trò này bao quát toàn bộ các quyền quản lý cấp cao.

Mục tiêu chính:

* Theo dõi và thúc đẩy sự tăng trưởng của toàn bộ nền tảng.
* Đảm bảo hệ thống hoạt động ổn định, an toàn và bảo mật.
* Có cái nhìn tổng quan và toàn quyền kiểm soát mọi hoạt động trên sàn.

Các hành vi/kịch bản chính:

* Quản lý người dùng: Xem, tạo, khóa hoặc xóa bất kỳ tài khoản nào (Customer, Seller, Staff).
* Phân tích kinh doanh: Truy cập các báo cáo tài chính, báo cáo tăng trưởng tổng thể, hiệu suất của các Người bán hàng đầu.
* Quản lý chiến dịch: Tạo và quản lý các chương trình khuyến mãi, banner quảng cáo trên toàn trang.
* Cấu hình hệ thống: Thiết lập các thông số quan trọng như phí hoa hồng cho Người bán, cấu hình cổng thanh toán, chỉnh sửa các chính sách.

Kỳ vọng đối với Hệ thống:

* Bảng điều khiển quản trị (Admin Dashboard) toàn diện, cung cấp cái nhìn 360 độ về hệ thống.
* Các công cụ mạnh mẽ để quản lý và cấu hình.
* Hệ thống ghi log (logging) và báo cáo lỗi chi tiết để phục vụ việc chẩn đoán sự cố.

# Yêu cầu chức năng theo nghiệp vụ.

Tác nhân: Người dùng (User)

| **Mã chức năng** | **Tên chức năng** | **Mô tả chi tiết** |
| --- | --- | --- |
| FC01 | Đăng ký tài khoản | Cho phép người dùng tạo tài khoản mới bằng email, số điện thoại hoặc tài khoản mạng xã hội. |
| FC02 | Đăng nhập | Cho phép người dùng đăng nhập vào hệ thống bằng thông tin đã đăng ký. |
| FC03 | Cập nhật thông tin cá nhân | Người dùng có thể chỉnh sửa thông tin như tên, mật khẩu, ảnh đại diện, địa chỉ, email, v.v. |

Tác nhân: Khách hàng (Customer)

| **Mã chức năng** | **Tên chức năng** | **Mô tả chi tiết** |
| --- | --- | --- |
| FC04 | Tìm kiếm & Xem sản phẩm | Cho phép khách hàng tìm kiếm sản phẩm theo tên, danh mục hoặc bộ lọc. Có thể xem chi tiết sản phẩm, mô tả, đánh giá, và giá bán. |
| FC05 | Quản lý giỏ hàng | Thêm, sửa, xóa sản phẩm trong giỏ hàng. Hệ thống tự động cập nhật tổng tiền và số lượng. |
| FC06 | Đặt hàng | Tiến hành đặt hàng từ giỏ, xác nhận địa chỉ giao hàng và phương thức thanh toán. |
| FC07 | Thanh toán | Hỗ trợ thanh toán qua nhiều hình thức: COD, ví điện tử, hoặc thẻ ngân hàng. |
| FC08 | Đánh giá sản phẩm | Sau khi mua hàng, khách hàng có thể viết đánh giá, chấm sao cho sản phẩm đã mua. |
| FC09 | Xem gợi ý sản phẩm | Hệ thống đề xuất sản phẩm dựa trên lịch sử xem, tìm kiếm, hoặc mua hàng của khách hàng. |
| FC10 | Theo dõi đơn hàng | Xem trạng thái đơn hàng (đã xác nhận, đang giao, hoàn thành, bị hủy) và chi tiết lộ trình. |

Tác nhân: Người bán (Seller)

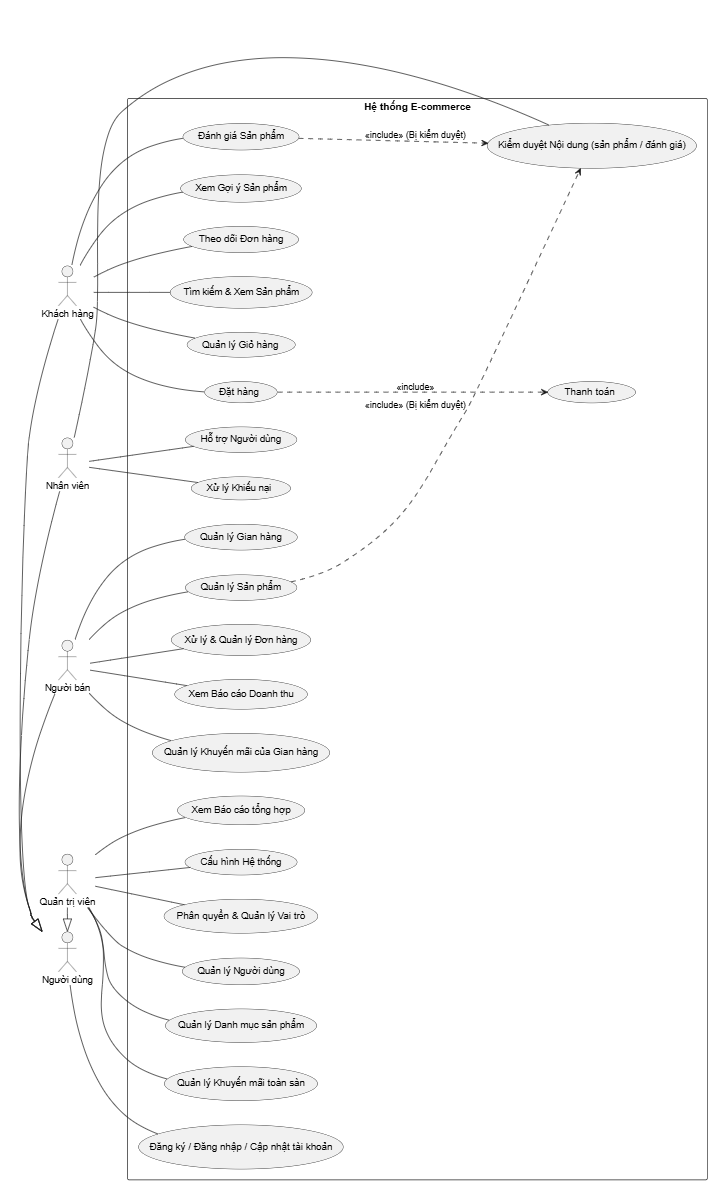
| **Mã chức năng** | **Tên chức năng** | **Mô tả chi tiết** |
| --- | --- | --- |
| FC11 | Quản lý gian hàng | Xem và chỉnh sửa thông tin gian hàng: tên, logo, mô tả, địa chỉ, thông tin liên hệ. |
| FC12 | Quản lý sản phẩm | Thêm mới, cập nhật hoặc xóa sản phẩm; quản lý hình ảnh, mô tả, giá bán, tồn kho. |
| FC13 | Xử lý & Quản lý đơn hàng | Xem danh sách đơn hàng của khách, xác nhận hoặc từ chối, cập nhật trạng thái vận chuyển. |
| FC14 | Xem báo cáo doanh thu | Thống kê doanh số theo ngày, tuần, tháng; xem biểu đồ doanh thu và sản phẩm bán chạy. |
| FC15 | Quản lý khuyến mãi của gian hàng | Tạo, chỉnh sửa hoặc xóa chương trình giảm giá, voucher riêng cho gian hàng. |

Tác nhân: Nhân viên (Staff)

| **Mã chức năng** | **Tên chức năng** | **Mô tả chi tiết** |
| --- | --- | --- |
| FC16 | Kiểm duyệt sản phẩm | Duyệt hoặc từ chối sản phẩm mới do người bán thêm vào, đảm bảo nội dung hợp lệ. |
| FC17 | Kiểm duyệt đánh giá | Xem và xác nhận hoặc ẩn các đánh giá vi phạm chính sách. |
| FC18 | Hỗ trợ người dùng | Tiếp nhận yêu cầu hỗ trợ, giải đáp thắc mắc từ khách hàng hoặc người bán. |
| FC19 | Xử lý khiếu nại | Nhận và xử lý các báo cáo, khiếu nại về đơn hàng, sản phẩm hoặc hành vi gian lận. |

Tác nhân: Quản trị viên (Administrator)

| **Mã chức năng** | **Tên chức năng** | **Mô tả chi tiết** |
| --- | --- | --- |
| FC20 | Quản lý người dùng | Thêm, khóa, hoặc xóa tài khoản người dùng (khách hàng, người bán, nhân viên). |
| FC21 | Quản lý danh mục sản phẩm | Thêm, chỉnh sửa, xóa danh mục cấp 1–n để phục vụ tìm kiếm và hiển thị sản phẩm. |
| FC22 | Quản lý khuyến mãi toàn sàn | Tạo hoặc điều chỉnh các chương trình khuyến mãi áp dụng cho toàn bộ sàn thương mại. |
| FC23 | Xem báo cáo tổng hợp | Thống kê tổng quan hệ thống: số người dùng, doanh thu, lượt truy cập, đơn hàng,... |
| FC24 | Cấu hình hệ thống | Quản lý các thông số hệ thống (SMTP, API key, cổng thanh toán,...). |
| FC25 | Phân quyền & Quản lý vai trò | Cấp quyền, phân vai trò cho người dùng; định nghĩa quyền hạn cho từng nhóm. |



# Phân tích yêu cầu hệ thống.

# Đặc tả chi tiết Use Case (Use Case Specifications).

1. UC-01: Đăng ký tài khoản

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| **Tên Use Case** | Đăng ký tài khoản |
| **Mã** | UC-01 |
| **Tác nhân chính** | Người dùng chưa xác thực (Guest) |
| **Mô tả ngắn gọn** | Cho phép người dùng tạo một tài khoản mới (Khách hàng hoặc Người bán) trên hệ thống. |
| **Điều kiện tiên quyết** | Người dùng chưa đăng nhập vào hệ thống. |
| **Điều kiện sau khi thực hiện** | Một tài khoản mới được tạo trong CSDL với trạng thái “Chờ xác thực”. Người dùng nhận được email để kích hoạt. |

**Luồng sự kiện chính (Main Flow):**

1. Người dùng nhấn vào nút **“Đăng ký”**.
2. Hệ thống hiển thị form đăng ký, yêu cầu chọn loại tài khoản (Khách hàng/Người bán) và nhập thông tin (email, mật khẩu, số điện thoại...).
3. Người dùng điền thông tin và nhấn **“Gửi”**.
4. Hệ thống kiểm tra tính hợp lệ của thông tin (email chưa tồn tại, mật khẩu đủ mạnh).
5. Hệ thống tạo tài khoản mới và lưu vào CSDL.
6. Hệ thống gửi email chứa liên kết kích hoạt.
7. Hiển thị thông báo đăng ký thành công và hướng dẫn kiểm tra email.

**Luồng rẽ nhánh (Exceptions):**

* **E1:** Email đã tồn tại → Hiển thị lỗi “Email này đã được sử dụng”.
* **E2:** Thông tin không hợp lệ → Báo lỗi cụ thể (mật khẩu yếu, số điện thoại sai định dạng...).

1. UC-02: Đăng nhập

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| **Tên Use Case** | Đăng ký tài khoản |
| **Mã** | UC-01 |
| **Tác nhân chính** | Người dùng chưa xác thực (Guest) |
| **Mô tả ngắn gọn** | Cho phép người dùng tạo một tài khoản mới (Khách hàng hoặc Người bán) trên hệ thống. |
| **Điều kiện tiên quyết** | Người dùng chưa đăng nhập vào hệ thống. |
| **Điều kiện sau khi thực hiện** | Một tài khoản mới được tạo trong CSDL với trạng thái “Chờ xác thực”. Người dùng nhận được email để kích hoạt. |

**Luồng sự kiện chính (Main Flow):**

1. Người dùng nhấn **“Đăng nhập”**.
2. Hệ thống hiển thị form (email/số điện thoại và mật khẩu).
3. Người dùng nhập thông tin và gửi.
4. Hệ thống kiểm tra thông tin với CSDL.
5. Nếu hợp lệ → tạo session, lưu thông tin người dùng.
6. Chuyển hướng đến trang chủ hoặc trang quản lý phù hợp.

**Luồng rẽ nhánh (Exceptions):**

* **E1:** Sai thông tin → Báo “Email hoặc mật khẩu không chính xác”.
* **E2:** Tài khoản chưa kích hoạt hoặc bị khóa → Hiển thị thông báo tương ứng.

1. UC-03: Quản lý sản phẩm (Người bán)

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| **Tên Use Case** | Quản lý sản phẩm |
| **Mã** | UC-03 |
| **Tác nhân chính** | Người bán (Seller) |
| **Mô tả ngắn gọn** | Cho phép người bán thêm, cập nhật và quản lý tồn kho sản phẩm trong gian hàng của mình. |
| **Điều kiện tiên quyết** | Người bán đã đăng nhập vào Kênh Người bán. |
| **Điều kiện sau khi thực hiện** | Sản phẩm mới được tạo với trạng thái “Chờ duyệt”; hệ thống gửi thông báo cho nhân viên kiểm duyệt. |

**Luồng sự kiện chính (Thêm mới):**

1. Người bán chọn “Thêm sản phẩm mới”.
2. Hệ thống hiển thị form nhập thông tin (tên, mô tả, giá, danh mục, hình ảnh...).
3. Người bán điền thông tin và nhấn “Lưu”.
4. Hệ thống kiểm tra dữ liệu hợp lệ.
5. Lưu sản phẩm vào CSDL với trạng thái **“Chờ duyệt”**.
6. Gửi thông báo đến Nhân viên để kiểm duyệt.

**Luồng rẽ nhánh (Exceptions):**

* **E1:** Thiếu thông tin → Báo lỗi, tô sáng trường bị thiếu.
* **E2:** Dữ liệu không hợp lệ → Báo cụ thể (giá không hợp lệ, hình ảnh quá lớn...).

1. UC-04: Xem gợi ý sản phẩm

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| **Tên Use Case** | Xem gợi ý sản phẩm |
| **Mã** | UC-04 |
| **Tác nhân chính** | Khách hàng (Customer) |
| **Mô tả ngắn gọn** | Hệ thống tự động hiển thị danh sách sản phẩm gợi ý phù hợp hành vi khách hàng. |
| **Điều kiện tiên quyết** | Khách hàng đang ở trang có module gợi ý (trang chủ, trang chi tiết...). |
| **Điều kiện sau khi thực hiện** | Danh sách sản phẩm được hiển thị. |

**Luồng sự kiện chính (Main Flow):**

1. Khách hàng tải trang web.
2. Frontend gửi yêu cầu API đến Backend (kèm ID người dùng).
3. Backend gọi Module Gợi ý.
4. Module Gợi ý xử lý, trả về danh sách sản phẩm phù hợp.
5. Backend truy vấn chi tiết sản phẩm và trả về cho Frontend.
6. Frontend hiển thị danh sách gợi ý.

**Luồng rẽ nhánh (Exceptions):**

* **E1:** Khách hàng mới (Cold Start) → Hiển thị sản phẩm bán chạy hoặc phổ biến.

1. UC-05: Quy trình Mua hang

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| **Tên Use Case** | Quy trình Mua hàng |
| **Mã** | UC-05 |
| **Tác nhân chính** | Khách hàng (Customer) |
| **Mô tả ngắn gọn** | Mô tả luồng hoàn chỉnh từ giỏ hàng → đặt hàng → thanh toán. |
| **Điều kiện tiên quyết** | Khách hàng đã đăng nhập và có sản phẩm trong giỏ hàng. |
| **Điều kiện sau khi thực hiện** | Đơn hàng mới được tạo, tồn kho được cập nhật, thông báo gửi đến khách hàng và người bán. |

**Luồng sự kiện chính (Main Flow):**

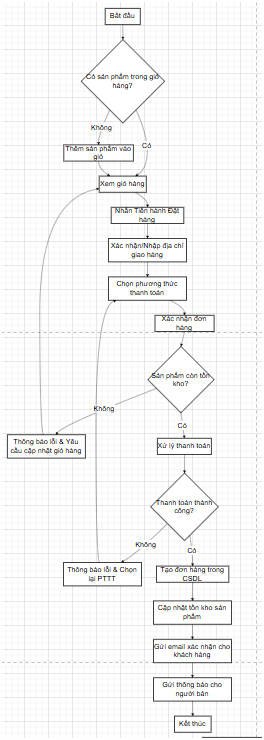
1. Xem giỏ hàng → nhấn “Tiến hành đặt hàng”.
2. Nhập hoặc xác nhận địa chỉ giao hàng.
3. Chọn phương thức thanh toán.
4. Xem tóm tắt đơn hàng.
5. Nhấn “Hoàn tất đơn hàng”.
6. Hệ thống kiểm tra tồn kho.
7. (include) Luồng xử lý thanh toán.
8. Nếu thành công → tạo đơn hàng mới trong CSDL.
9. Cập nhật tồn kho.
10. Gửi email xác nhận cho khách hàng và thông báo cho người bán.

**Luồng rẽ nhánh (Exceptions):**

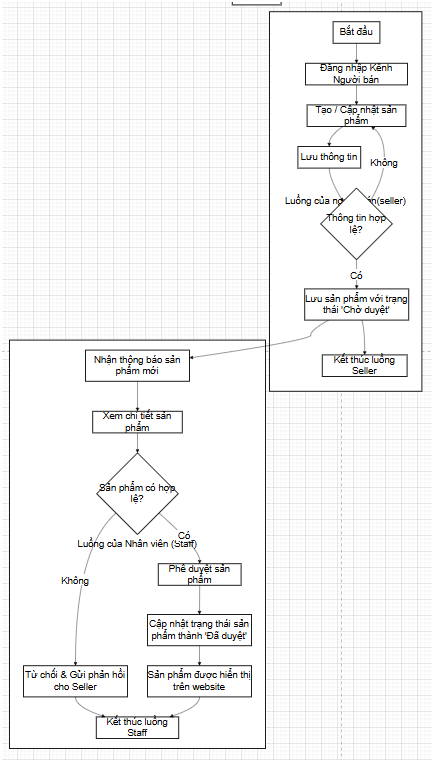
* **E1:** Hết hàng → báo lỗi, yêu cầu cập nhật giỏ hàng.
* **E2:** Thanh toán thất bại → báo lỗi, cho phép thử lại hoặc đổi phương thức.

# Sơ đồ Hoạt động (Activity Diagrams).

Sơ đồ: Quy trình Mua hàng

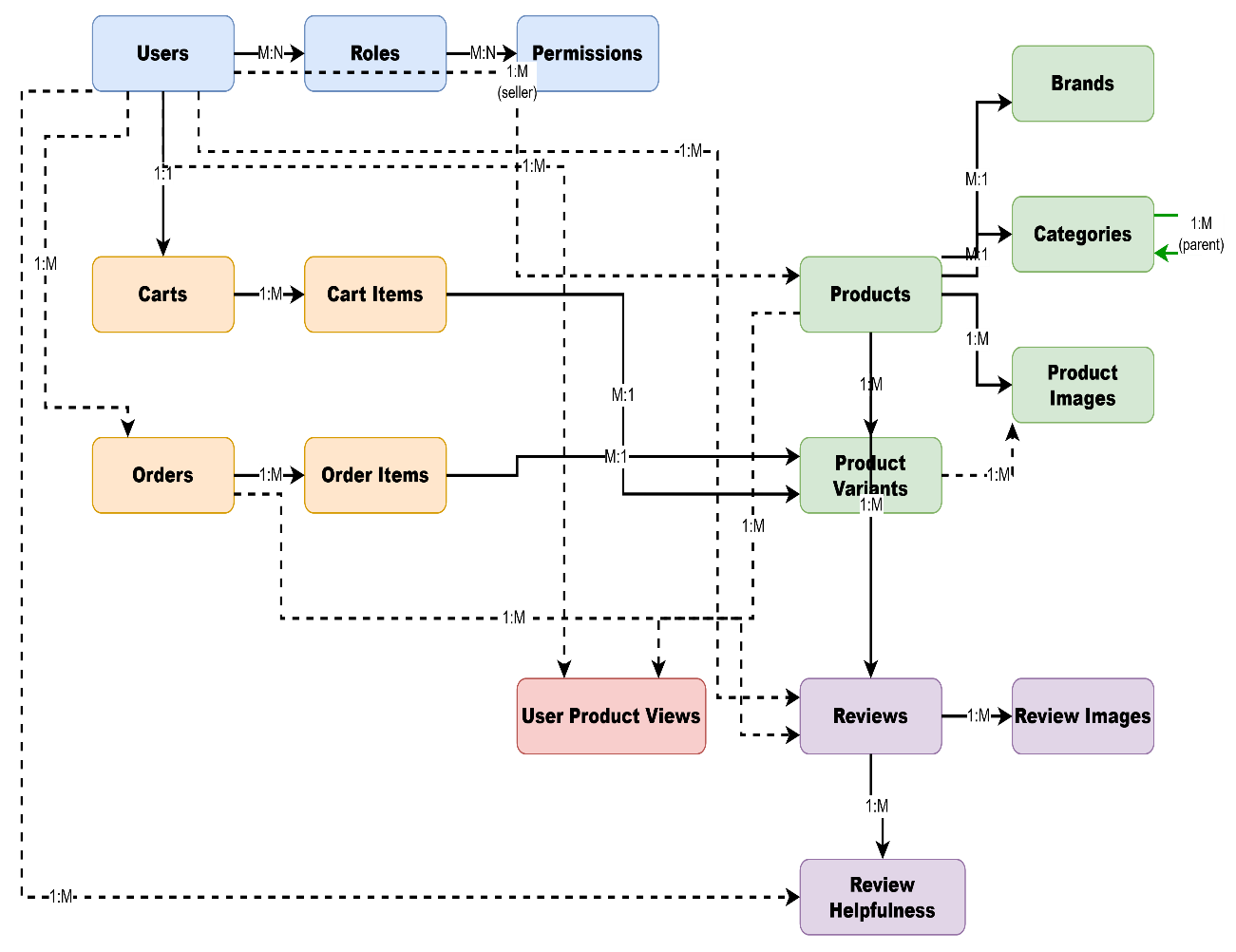


Sơ đồ: Quy trình Quản lý & Kiểm duyệt Sản phẩm



# Thiết kế Cơ sở dữ liệu.

# Sơ đồ ERD.



# Mô tả chi tiết các bảng.

**1. Roles (Vai trò)**

**Mô tả:**  
Quản lý các vai trò (roles) người dùng trong hệ thống. Việc tách riêng bảng vai trò giúp hệ thống dễ dàng phân quyền (authorization) và mở rộng linh hoạt.

**Mối quan hệ:**

* Một Role có thể được gán cho nhiều Users (1–N).
* Có quan hệ Nhiều–Nhiều (Many–Many) với Permissions thông qua bảng role\_permissions.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| role\_id | BINARY(16) | Khóa chính (Primary key), định danh duy nhất cho mỗi vai trò. |
| name | VARCHAR(255) | Tên vai trò (ví dụ: Customer, Seller, Admin). |
| created\_at | DATETIME(6) | Thời điểm tạo. |
| updated\_at | DATETIME(6) | Thời điểm cập nhật. |

**2. Permissions (Quyền hạn)**

**Mô tả:**  
Xác định các hành động cụ thể mà người dùng được phép thực hiện, phục vụ cho cơ chế RBAC (Role-Based Access Control – Kiểm soát truy cập dựa trên vai trò).

**Mối quan hệ:**

* Có quan hệ Nhiều–Nhiều với Roles thông qua bảng role\_permissions.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| permission\_id | BINARY(16) | Khóa chính (Primary key). |
| name | VARCHAR(255) | Tên quyền duy nhất, dùng để kiểm tra trong mã nguồn (ví dụ: product.create, order.manage). |
| created\_at | DATETIME(6) | Thời điểm tạo. |
| updated\_at | DATETIME(6) | Thời điểm cập nhật. |

**3. Users (Người dùng)**

**Mô tả:**  
Lưu trữ thông tin xác thực (authentication) và thông tin cá nhân cơ bản của người dùng.

**Mối quan hệ:**

* Liên kết với Roles để xác định quyền.
* Là cha (Parent) của Orders, Carts, Reviews và View\_History.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| user\_id | BINARY(16) | Khóa chính (Primary key). |
| role\_id | BINARY(16) | Khóa ngoại (Foreign key) → Roles. |
| email | VARCHAR(255) | Địa chỉ email duy nhất. |
| username | VARCHAR(255) | Tên đăng nhập. |
| password\_hash | VARCHAR(255) | Mật khẩu đã được băm (hash). |
| is\_active | BIT | Trạng thái hoạt động của người dùng. |
| created\_at | DATETIME(6) | Thời điểm tạo. |
| updated\_at | DATETIME(6) | Thời điểm cập nhật. |

**4. Brands (Thương hiệu)**

**Mô tả:**  
Đại diện cho thương hiệu của sản phẩm, giúp phân loại và gợi ý sản phẩm theo nhãn hàng.

**Mối quan hệ:**

* Một Brand có thể có nhiều Products (1–N).

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính (Primary key). |
| name | VARCHAR(255) | Tên thương hiệu. |
| is\_active | BIT | Trạng thái hoạt động. |
| created\_at | DATETIME(6) | Thời điểm tạo. |
| updated\_at | DATETIME(6) | Thời điểm cập nhật. |

**5. Categories (Danh mục)**

**Mô tả:**  
Phân loại sản phẩm giúp người dùng tìm kiếm và quản lý dễ dàng hơn.

**Mối quan hệ:**

* Một Category có thể chứa nhiều Products (1–N).
* Có thể có quan hệ cha–con (self-reference).

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính (Primary key). |
| parent\_id | BINARY(16) | Khóa ngoại (Foreign key) → Categories. |
| name | VARCHAR(255) | Tên danh mục. |
| slug | VARCHAR(255) | Đường dẫn thân thiện (URL slug). |
| created\_at | DATETIME(6) | Thời điểm tạo. |
| updated\_at | DATETIME(6) | Thời điểm cập nhật. |

**6. Products (Sản phẩm)**

**Mô tả:**  
Lưu thông tin chung của sản phẩm, không phụ thuộc vào biến thể (variant).

**Mối quan hệ:**

* Thuộc về Category và Brand.
* Có quan hệ 1–N với Product\_Variants.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính. |
| category\_id | BINARY(16) | Khóa ngoại → Categories. |
| brand\_id | BINARY(16) | Khóa ngoại → Brands. |
| name | VARCHAR(255) | Tên sản phẩm. |
| slug | VARCHAR(255) | Đường dẫn URL. |
| description | TEXT | Mô tả chi tiết sản phẩm. |
| is\_active | BIT | Trạng thái hiển thị sản phẩm. |
| created\_at | DATETIME(6) | Thời điểm tạo. |
| updated\_at | DATETIME(6) | Thời điểm cập nhật. |

**7. Product\_Variants (Biến thể sản phẩm)**

**Mô tả:**  
Đại diện cho từng mặt hàng cụ thể có thể bán được (theo màu sắc, dung lượng, kích cỡ…).

**Mối quan hệ:**

* Thuộc về Product.
* Được tham chiếu bởi Cart\_Items và Order\_Items.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính. |
| product\_id | BINARY(16) | Khóa ngoại → Products. |
| sku | VARCHAR(255) | Mã định danh hàng hóa (SKU). |
| price | DECIMAL(10,2) | Giá bán. |
| stock | INT | Số lượng tồn kho. |
| attributes | JSON | Đặc tính như màu, dung lượng, kích cỡ. |

**8. Product\_Images (Ảnh sản phẩm)**

**Mô tả:**  
Lưu danh sách ảnh minh họa cho sản phẩm.

**Mối quan hệ:**

* Một Product có thể có nhiều ảnh.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính. |
| product\_id | BINARY(16) | Khóa ngoại → Products. |
| image\_url | VARCHAR(255) | Đường dẫn ảnh. |
| is\_primary | BIT | Đánh dấu ảnh chính. |

**9. Carts (Giỏ hàng) & Cart\_Items (Chi tiết giỏ hàng)**

**Mô tả:**  
Lưu trữ tạm thời các sản phẩm mà người dùng dự định mua.

**Mối quan hệ:**

* Một User có một Cart.
* Một Cart có nhiều Cart\_Items.
* Cart\_Items liên kết đến Product\_Variants.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Carts | id | BINARY(16) | Khóa chính. |
|  | user\_id | BINARY(16) | Khóa ngoại → Users. |
|  | created\_at | DATETIME(6) | Thời điểm tạo giỏ hàng. |
| Cart\_Items | cart\_id | BINARY(16) | Khóa ngoại → Carts. |
|  | variant\_id | BINARY(16) | Khóa ngoại → Product\_Variants. |
|  | quantity | INT | Số lượng. |

**10. Orders (Đơn hàng) & Order\_Items (Chi tiết đơn hàng)**

**Mô tả:**  
Ghi nhận thông tin giao dịch mua bán chính thức.

**Mối quan hệ:**

* Một User có thể có nhiều Orders.
* Một Order có nhiều Order\_Items.
* Order\_Items liên kết đến Product\_Variants.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| Orders | id | BINARY(16) | Khóa chính. |
|  | user\_id | BINARY(16) | Khóa ngoại → Users. |
|  | status | VARCHAR(50) | Trạng thái đơn hàng. |
|  | shipping\_address | VARCHAR(255) | Địa chỉ giao hàng. |
|  | created\_at | DATETIME(6) | Thời điểm đặt hàng. |
| Order\_Items | id | BINARY(16) | Khóa chính. |
|  | order\_id | BINARY(16) | Khóa ngoại → Orders. |
|  | variant\_id | BINARY(16) | Khóa ngoại → Product\_Variants. |
|  | quantity | INT | Số lượng. |
|  | price\_at\_purchase | DECIMAL(10,2) | Giá tại thời điểm mua. |

**11. Payments (Thanh toán)**

**Mô tả:**  
Theo dõi giao dịch thanh toán cho từng đơn hàng.

**Mối quan hệ:**

* Mỗi Order có một Payment tương ứng (1–1).

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính. |
| order\_id | BINARY(16) | Khóa ngoại → Orders. |
| payment\_method | VARCHAR(50) | Hình thức thanh toán (COD, VNPay, Momo…). |
| status | VARCHAR(50) | Trạng thái thanh toán. |
| transaction\_id | VARCHAR(255) | Mã giao dịch. |

**12. Reviews (Đánh giá)**

**Mô tả:**  
Thu thập đánh giá và phản hồi của người dùng về sản phẩm.

**Mối quan hệ:**

* Một User có thể đánh giá nhiều Products.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính. |
| user\_id | BINARY(16) | Khóa ngoại → Users. |
| product\_id | BINARY(16) | Khóa ngoại → Products. |
| rating | INT | Điểm đánh giá. |
| comment | TEXT | Nội dung bình luận. |
| created\_at | DATETIME(6) | Thời điểm tạo. |

**13. View\_History (Lịch sử xem sản phẩm)**

**Mô tả:**  
Ghi nhận hành vi xem sản phẩm của người dùng, phục vụ hệ thống gợi ý (recommendation system).

**Mối quan hệ:**

* Quan hệ N–N giữa Users và Products.

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| id | BINARY(16) | Khóa chính. |
| user\_id | BINARY(16) | Khóa ngoại → Users. |
| product\_id | BINARY(16) | Khóa ngoại → Products. |
| viewed\_at | DATETIME(6) | Thời điểm xem sản phẩm. |

# Thiết kế kiến trúc hệ thống.

# Thiết kế API.

|  |  |  |
| --- | --- | --- |
| **POST** | /api/auth/login | Đăng nhập |
| **POST** | /api/auth/register | Đăng ký tài khoản |
| **POST** | /api/auth/refresh-token | Làm mới access token |
| **GET** | /api/auth/me | Lấy thông tin user hiện tại |
| **GET** | /api/users | Lấy danh sách users |
| **GET** | /api/users/search | Tìm kiếm users |
| **GET** | /api/users/role/{roleName} | Lấy users theo role |
| **GET** | /api/users/{id} | Lấy user by ID |
| **POST** | /api/users | Tạo user mới |
| **PUT** | /api/users/{id} | Cập nhật user |
| **POST** | /api/users/{id}/roles | Gán roles cho user |
| **DELETE** | /api/users/{id}/roles | Xóa roles của user |
| **POST** | /api/users/{id}/change-password | Đổi mật khẩu |
| **POST** | /api/users/{id}/verify-email | Verify email |
| **DELETE** | /api/users/{id} | Soft delete user |
| **PATCH** | /api/users/{id}/restore | Restore user đã xóa |
| **GET** | /api/roles | Lấy tất cả roles |
| **GET** | /api/roles/{id} | Lấy role by ID |
| **POST** | /api/roles | Tạo role mới |
| **PUT** | /api/roles/{id} | Cập nhật role |
| **POST** | /api/roles/{id}/permissions | Gán permissions cho role |
| **DELETE** | /api/roles/{id}/permissions | Xóa permissions của role |
| **DELETE** | /api/roles/{id} | Xóa role |
| **GET** | /api/permissions | Lấy tất cả permissions |
| **GET** | /api/permissions/{id} | Lấy permission by ID |
| **POST** | /api/permissions | Tạo permission mới |
| **PUT** | /api/permissions/{id} | Cập nhật permission |
| **DELETE** | /api/permissions/{id} | Xóa permission |
| **GET** | /api/products | Lấy tất cả products |
| **GET** | /api/products/{id} | Chi tiết product by ID |
| **GET** | /api/products/slug/{slug} | Chi tiết product by slug |
| **GET** | /api/products/search | Tìm kiếm products |
| **GET** | /api/products/category/{categoryId} | Products theo category |
| **GET** | /api/products/brand/{brandId} | Products theo brand |
| **GET** | /api/products/trending | Sản phẩm trending |
| **GET** | /api/products/best-sellers | Sản phẩm bán chạy |
| **GET** | /api/products/top-rated | Sản phẩm rating cao |
| **GET** | /api/products/{id}/similar | Sản phẩm tương tự |
| **POST** | /api/products | Tạo product mới |
| **GET** | /api/products/my-products | Lấy products của seller |
| **PUT** | /api/products/{id} | Cập nhật product |
| **DELETE** | /api/products/{id} | Xóa product |
| **GET** | /api/products/pending | Products chờ duyệt |
| **GET** | /api/products/admin/all | Tất cả products (admin) |
| **GET** | /api/products/status/{status} | Products theo status |
| **POST** | /api/products/{id}/approve | Duyệt product |
| **POST** | /api/products/{id}/reject | Từ chối product |
| **GET** | /api/categories | Lấy tất cả categories |
| **GET** | /api/categories/root | Root categories with children |
| **GET** | /api/categories/{id} | Category by ID |
| **GET** | /api/categories/slug/{slug} | Category by slug |
| **GET** | /api/categories/{id}/children | Sub-categories |
| **POST** | /api/categories | Tạo category |
| **PUT** | /api/categories/{id} | Cập nhật category |
| **DELETE** | /api/categories/{id} | Xóa category |
| **PATCH** | /api/categories/{id}/restore | Restore category |
| **GET** | /api/brands | Lấy tất cả brands |
| **GET** | /api/brands/{id} | Brand by ID |
| **GET** | /api/brands/slug/{slug} | Brand by slug |
| **POST** | /api/brands | Tạo brand |
| **PUT** | /api/brands/{id} | Cập nhật brand |
| **DELETE** | /api/brands/{id} | Xóa brand |
| **PATCH** | /api/brands/{id}/restore | Restore brand |
| **GET** | /api/cart | Lấy giỏ hàng hiện tại |
| **POST** | /api/cart/items | Thêm item vào giỏ |
| **PUT** | /api/cart/items/{itemId} | Cập nhật quantity |
| **DELETE** | /api/cart/items/{itemId} | Xóa item khỏi giỏ |
| **DELETE** | /api/cart | Xóa toàn bộ giỏ hàng |
| **POST** | /api/cart/sync | Sync cart data |
| **GET** | /api/cart/count | Đếm số item trong giỏ |
| **POST** | /api/orders | Tạo order từ cart |
| **GET** | /api/orders/my-orders | Lấy orders của mình |
| **GET** | /api/orders/{orderId} | Chi tiết order |
| **GET** | /api/orders/number/{orderNumber} | Order by order number |
| **POST** | /api/orders/{orderId}/cancel | Hủy order |
| **GET** | /api/orders | Tất cả orders |
| **GET** | /api/orders/status/{status} | Orders theo status |
| **GET** | /api/orders/payment-status/{status} | Orders theo payment status |
| **GET** | /api/orders/pending | Pending orders |
| **GET** | /api/orders/search | Tìm kiếm orders |
| **GET** | /api/orders/date-range | Orders trong khoảng thời gian |
| **POST** | /api/orders/{orderId}/confirm | Xác nhận order |
| **POST** | /api/orders/{orderId}/processing | Đánh dấu đang xử lý |
| **POST** | /api/orders/{orderId}/ship | Đánh dấu đã ship |
| **POST** | /api/orders/{orderId}/deliver | Đánh dấu đã giao |
| **PUT** | /api/orders/{orderId}/status | Cập nhật status |
| **POST** | /api/orders/{orderId}/payment/paid | Đánh dấu đã thanh toán |
| **POST** | /api/orders/{orderId}/payment/failed | Đánh dấu thanh toán thất bại |
| **GET** | /api/orders/statistics | Thống kê orders |
| **GET** | /api/orders/revenue/total | Tổng doanh thu |
| **GET** | /api/orders/revenue/date-range | Doanh thu theo thời gian |
| **GET** | /api/reviews/product/{productId} | Reviews của product |
| **GET** | /api/reviews/product/{productId}/rating/{rating} | Filter by rating |
| **GET** | /api/reviews/product/{productId}/verified | Verified purchase reviews |
| **GET** | /api/reviews/product/{productId}/helpful | Most helpful reviews |
| **GET** | /api/reviews/product/{productId}/summary | Rating summary |
| **GET** | /api/reviews/{reviewId} | Review details |
| **POST** | /api/reviews | Tạo review |
| **PUT** | /api/reviews/{reviewId} | Cập nhật review |
| **DELETE** | /api/reviews/{reviewId} | Xóa review |
| **GET** | /api/reviews/my-reviews | Reviews của mình |
| **POST** | /api/reviews/{reviewId}/vote | Vote helpful/not helpful |
| **DELETE** | /api/reviews/{reviewId}/vote | Xóa vote |
| **GET** | /api/reviews/status/{status} | Reviews theo status |
| **GET** | /api/reviews/pending | Pending reviews |
| **POST** | /api/reviews/{reviewId}/approve | Duyệt review |
| **POST** | /api/reviews/{reviewId}/reject | Từ chối review |
| **POST** | /api/reviews/{reviewId}/reply | Reply review |
| **GET** | /api/recommendations/homepage | Homepage recommendations |
| **GET** | /api/recommendations/trending | Trending products |
| **GET** | /api/recommendations/best-sellers | Best sellers |
| **GET** | /api/recommendations/top-rated | Top rated |
| **GET** | /api/recommendations/new-arrivals | New arrivals |
| **GET** | /api/recommendations/similar/{productId} | Similar products |
| **GET** | /api/recommendations/bought-together/{productId} | Frequently bought together |
| **GET** | /api/recommendations/for-you | Personalized recommendations |

# Thiết kế chi tiết module gợi ý.

# Xác định nguồn dữ liệu cho Hệ thống Gợi ý.

1. Dữ liệu Tường minh (Explicit Feedback):

Tín hiệu chất lượng cao, thể hiện rõ ý định của người dùng.

Đánh giá (Rating): Điểm từ 1-5 sao.

Danh sách yêu thích (Wishlist): Hành động lưu sản phẩm.

1. Dữ liệu Ngầm định (Implicit Feedback):

Dữ liệu hành vi có khối lượng lớn, cần được suy luận.

Lịch sử Mua hàng: Tín hiệu mạnh nhất.

Thêm vào giỏ hàng: Tín hiệu về ý định mua rất cao.

Lịch sử xem: Tín hiệu về sự quan tâm ban đầu.

1. Dữ liệu Thuộc tính (Attribute/Content Data):

Dữ liệu mô tả, cần thiết cho việc giải quyết các vấn đề cold-start.

Thuộc tính Sản phẩm: Danh mục, thương hiệu, thông số kỹ thuật, mô tả.

Thuộc tính Người dùng: Các thông tin nhân khẩu học cơ bản thu thập được khi đăng ký (nếu có).

# Mô hình hóa Dữ liệu Sự kiện.

Để hệ thống có thể thu thập và xử lý dữ liệu một cách nhất quán, tất cả các hành vi của người dùng sẽ được ghi nhận, Hệ thống thu thập dữ liệu từ hai nguồn chính trong cơ sở dữ liệu MySQL:

**1. Dữ liệu đầu vào**

**1.1. Nguồn dữ liệu**

Hệ thống thu thập dữ liệu từ hai nguồn chính trong cơ sở dữ liệu MySQL:

**1.1.1. Dữ liệu mua hàng (Purchase Data)**

**Bảng sử dụng:** orders, order\_items, product\_variants

**Câu truy vấn SQL:**

SELECT

HEX(o.user\_id) as user\_id,

HEX(pv.product\_id) as product\_id,

oi.quantity,

o.created\_at as purchase\_date

FROM orders o

INNER JOIN order\_items oi ON o.id = oi.order\_id

INNER JOIN product\_variants pv ON oi.variant\_id = pv.id

WHERE o.order\_status IN ('CONFIRMED', 'DELIVERED', 'SHIPPED')

AND o.is\_active = 1

AND oi.is\_active = 1

ORDER BY o.created\_at DESC

**Đặc điểm dữ liệu:**

* Chỉ lấy đơn hàng đã được xác nhận, đang giao hoặc đã giao
* Loại bỏ đơn hàng bị hủy hoặc không hợp lệ
* Bao gồm thông tin số lượng sản phẩm được mua
* Sắp xếp theo thời gian mua gần nhất

**Cấu trúc dữ liệu:**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| user\_id | VARCHAR(32) | ID người dùng (Hex format) |
| product\_id | VARCHAR(32) | ID sản phẩm (Hex format) |
| quantity | INTEGER | Số lượng sản phẩm đã mua |
| purchase\_date | TIMESTAMP | Thời điểm mua hàng |

**1.1.2. Dữ liệu lượt xem (View Data)**

**Bảng sử dụng:** user\_product\_views

**Câu truy vấn SQL:**

SELECT

HEX(user\_id) as user\_id,

HEX(product\_id) as product\_id,

view\_count,

last\_viewed\_at

FROM user\_product\_views

**Đặc điểm dữ liệu:**

* Lưu trữ số lần người dùng xem một sản phẩm
* Cập nhật thời gian xem gần nhất
* Dữ liệu ẩn cho thấy sở thích của người dùng

**Cấu trúc dữ liệu:**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| user\_id | VARCHAR(32) | ID người dùng (Hex format) |
| product\_id | VARCHAR(32) | ID sản phẩm (Hex format) |
| view\_count | INTEGER | Số lần xem sản phẩm |
| last\_viewed\_at | TIMESTAMP | Lần xem cuối cùng |

**1.2. Thống kê dữ liệu**

Dữ liệu được thu thập từ hệ thống production bao gồm:

* **Số lượng người dùng**: Từ 50 đến hàng nghìn người dùng
* **Số lượng sản phẩm**: Tối thiểu 10 sản phẩm
* **Số lượng tương tác**: Tổng hợp từ purchase và view records
* **Độ thưa (Sparsity)**: Thường > 99% (điển hình của hệ thống E-Commerce)

**1.3. Xử lý dữ liệu**

**1.3.1. Chuẩn hóa ID**

Tất cả Product ID và User ID được chuyển đổi từ định dạng UUID sang Hex format để tối ưu hóa hiệu suất:

UUID: ba6f86d7-6ca4-4499-aee9-108f60a9a476

Hex: BA6F86D76CA44499AEE9108F60A9A476

**1.3.2. Tổng hợp dữ liệu**

**Purchase Data:**

* Nhóm theo (user\_id, product\_id)
* Tính tổng số lượng (quantity) cho mỗi cặp user-product
* Kết quả: Ma trận thưa với giá trị là tổng số lượng đã mua

**View Data:**

* Nhóm theo (user\_id, product\_id)
* Tính tổng số lần xem (view\_count)
* Kết quả: Ma trận thưa với giá trị là số lần đã xem

# Lựa chọn Phương pháp và Thuật toán Gợi ý.

Chúng ta sẽ áp dụng một Chiến lược Lai Chuyển đổi theo Ngữ cảnh (Contextual Switching Hybrid Strategy). Thay vì trộn lẫn kết quả một cách phức tạp, hệ thống sẽ lựa chọn phương pháp gợi ý phù hợp nhất tùy thuộc vào ngữ cảnh của người dùng và vị trí trên trang web.

1. Phương pháp chính: Lọc cộng tác dựa trên mục (Item-Based Collaborative Filtering - IBCF)

Tính ổn định (Stability): Mối quan hệ tương đồng giữa các mục (ví dụ: "iPhone 15" và "Ốp lưng iPhone 15") thay đổi chậm hơn nhiều so với sở thích của một người dùng. Do đó, ma trận tương đồng không cần phải tính toán lại liên tục, giúp giảm tải cho hệ thống.

Khả năng mở rộng (Scalability): Việc tính toán nặng (ma trận tương đồng) được thực hiện offline. Quá trình dự đoán (online) rất nhanh, chỉ liên quan đến việc tra cứu và tính toán trên một tập mục nhỏ (những mục người dùng đã tương tác).

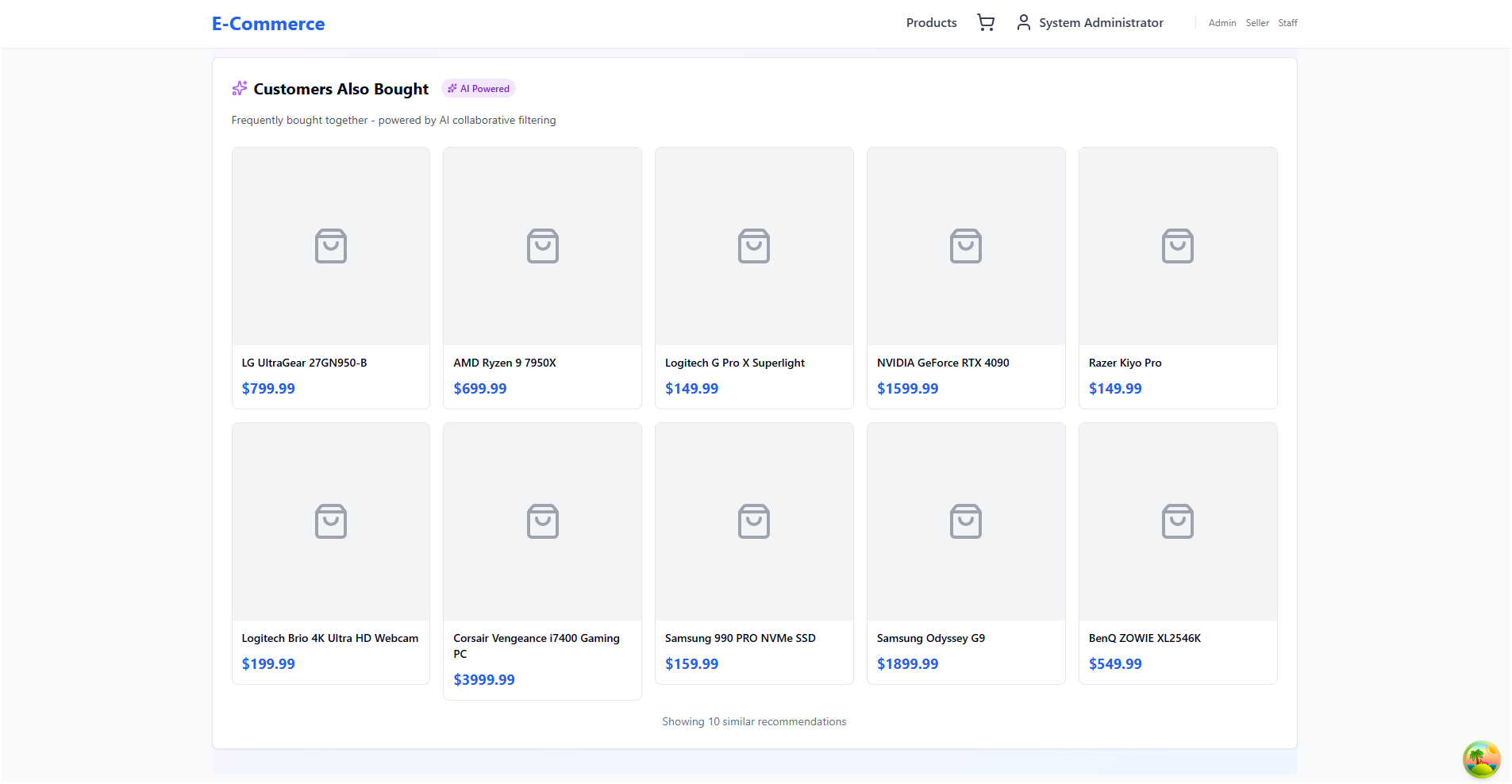
Tính giải thích được (Explainability): Hệ thống có thể dễ dàng giải thích lý do gợi ý, tăng độ tin cậy: "Chúng tôi gợi ý mục [A] cho bạn vì bạn đã thích/mua [B]" (với A và B có sA,B cao).Hiệu quả với dữ liệu thưa: Hoạt động tốt ngay cả khi người dùng chỉ tương tác với một vài mục.

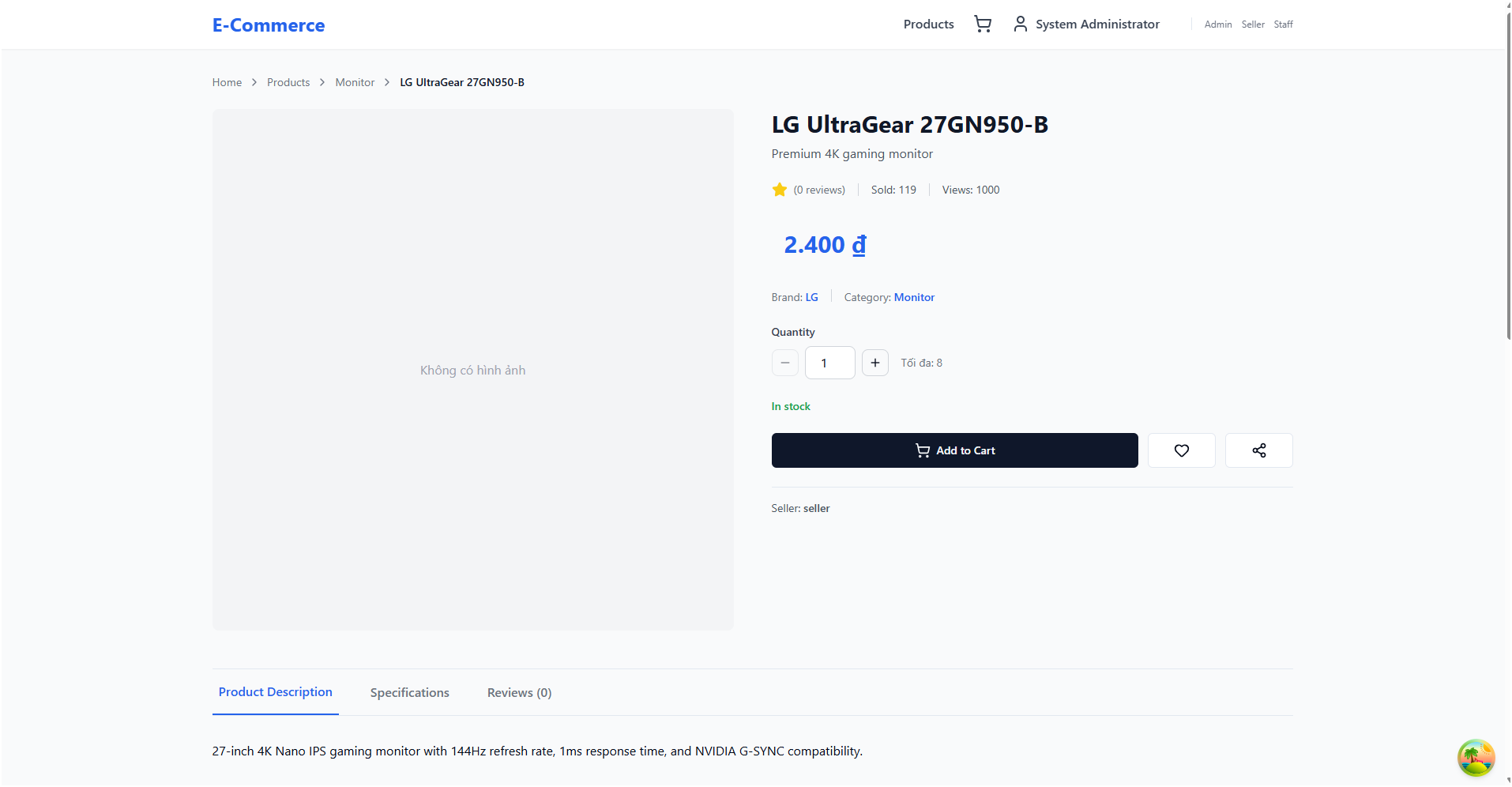
1. Phương pháp dự phòng: Dựa trên sự Phổ biến (Popularity-Based)

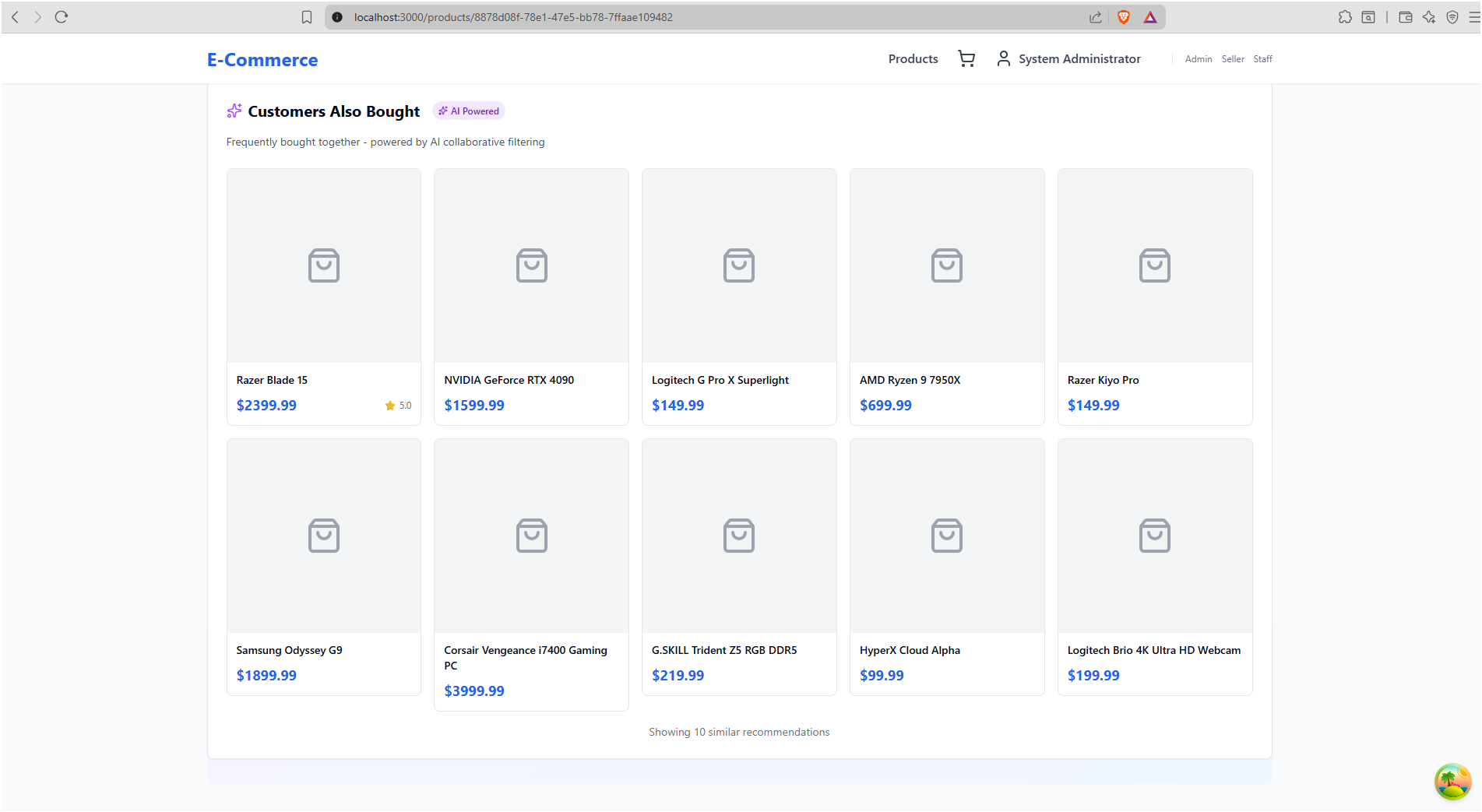
Mục đích: Là giải pháp an toàn cho vấn đề cold-start của người dùng mới.

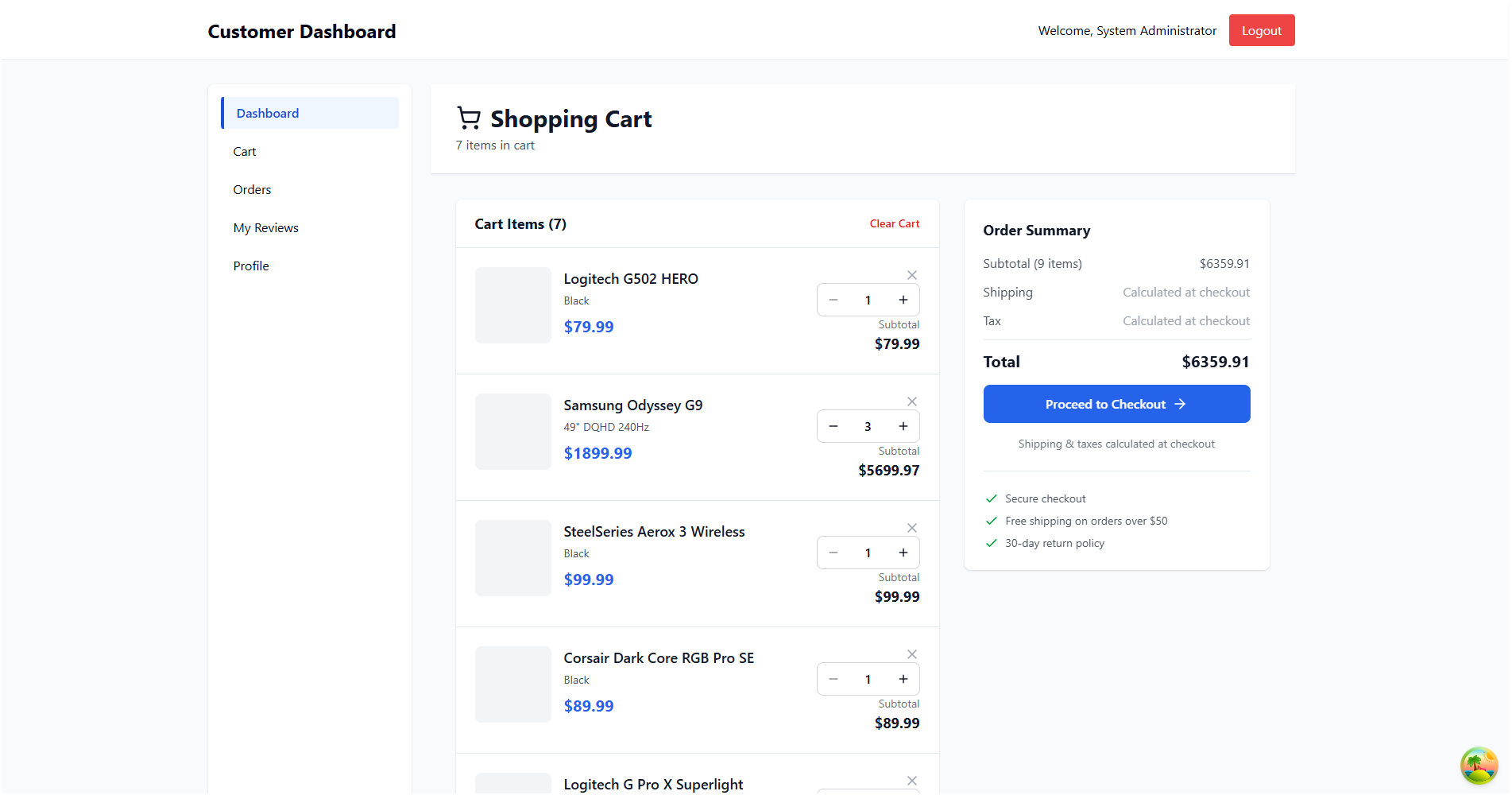
Thuật toán: Xếp hạng sản phẩm theo tổng lượt mua hoặc xem trong 30 ngày gần nhất.

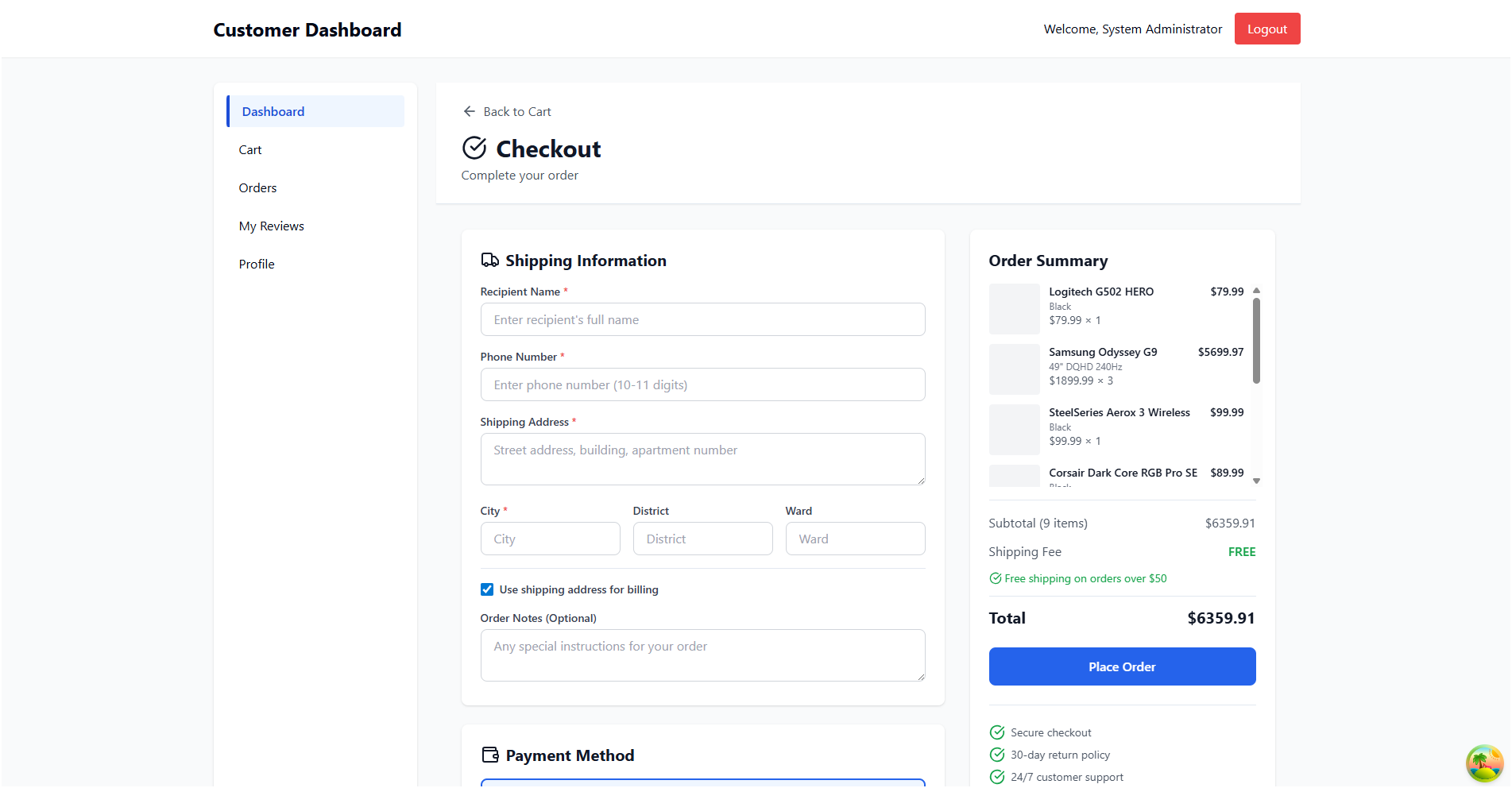
# Thiết kế giao diện.

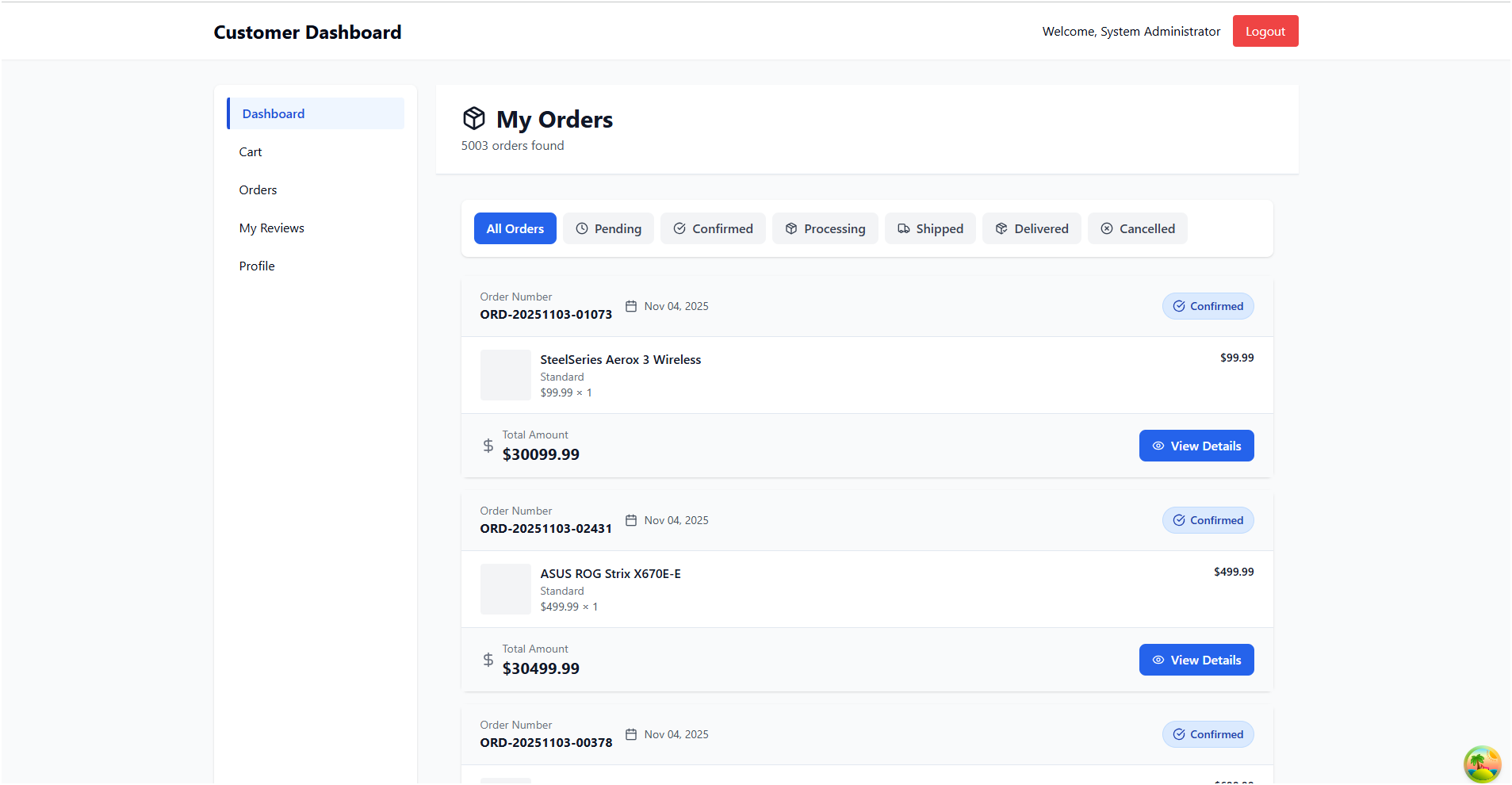












# Triển khai hệ thống.

# Yêu cầu hệ thống.

# Cài đặt và chạy

Mã nguồn: <https://github.com/CaoNguyen-1883/HTTM>

**Cách 1: Dùng Docker (Đơn giản nhất)**

**Bước 1: Clone repository**

git clone https://github.com/your-username/ecommerce-platform.git

cd ecommerce-platform

**Bước 2: Cấu hình môi trường**

# Copy file environment mẫu

cp .env.example .env

# Sửa file .env với text editor của bạn

nano .env

Các giá trị cần sửa trong file .env:

# Đổi password database

DB\_PASSWORD=your-secure-password-here

# Đổi JWT secret

JWT\_SECRET=your-very-long-random-secret-key-at-least-256-bits

# Tắt test users khi production

TEST\_USERS=false

**Bước 3: Khởi động toàn bộ hệ thống**

# Sử dụng Makefile

make init

# Hoặc dùng Docker Compose trực tiếp

docker-compose build

docker-compose up -d

**Bước 4: Đợi services khởi động**

# Xem logs để theo dõi quá trình khởi động

make logs

# Hoặc

docker-compose logs -f

Đợi cho đến khi thấy dòng:

* Backend: "Started EcommerceApplication"
* ML Service: "Application startup complete"

**Bước 5: Train ML model lần đầu**

# Train model

make train

# Hoặc

docker-compose exec ml-service python src/train.py

**Bước 6: Truy cập ứng dụng**

|  |  |  |
| --- | --- | --- |
| **Service** | **URL** | **Mô tả** |
| Frontend | <http://localhost> | Giao diện chính |
| Backend API | <http://localhost:8080> | Spring Boot API |
| Swagger UI | <http://localhost:8080/swagger-ui.html> | API documentation |
| ML API | <http://localhost:8000> | Python FastAPI |
| ML API Docs | <http://localhost:8000/docs> | ML API documentation |

**Bước 7: Đăng nhập với tài khoản test**

Nếu bạn để TEST\_USERS=true trong file .env, có thể dùng các tài khoản sau:

|  |  |  |
| --- | --- | --- |
| **Vai trò** | **Email** | **Password** |
| Admin | [admin@ecommerce.com](mailto:admin@ecommerce.com) | Admin@123 |
| Staff | [staff@ecommerce.com](mailto:staff@ecommerce.com) | Staff@123 |
| Seller | [seller@ecommerce.com](mailto:seller@ecommerce.com) | Seller@123 |
| Customer | [customer@ecommerce.com](mailto:customer@ecommerce.com) | Customer@123 |

**Cách 2: Chạy thủ công (Không dùng Docker)**

**A. Setup Database**

# Cài đặt MySQL 8.0

# Tạo database

mysql -u root -p

CREATE DATABASE ecommerce\_db CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

CREATE USER 'ecommerce'@'localhost' IDENTIFIED BY 'your-password';

GRANT ALL PRIVILEGES ON ecommerce\_db.\* TO 'ecommerce'@'localhost';

FLUSH PRIVILEGES;

EXIT;

# Cài đặt Redis (optional)

# Ubuntu/Debian

sudo apt-get install redis-server

sudo systemctl start redis

# macOS

brew install redis

brew services start redis

**B. Chạy Backend (Spring Boot)**

cd ecommerce

# Sửa file application.yml hoặc tạo application-dev.yml

# Cấu hình database connection

# Build và chạy

mvn clean install

mvn spring-boot:run

# Hoặc chạy jar file

java -jar target/ecommerce-0.0.1-SNAPSHOT.jar

Backend sẽ chạy trên: [http://localhost:8080](http://localhost:8080/)

**C. Chạy ML Service (Python)**

cd ml-recommendation

# Tạo virtual environment

python -m venv venv

source venv/bin/activate # Linux/Mac

# hoặc

venv\Scripts\activate # Windows

# Cài đặt dependencies

pip install -r requirements.txt

# Cấu hình database connection (environment variables)

export DB\_HOST=localhost

export DB\_PORT=3306

export DB\_NAME=ecommerce\_db

export DB\_USER=ecommerce

export DB\_PASSWORD=your-password

# Train model lần đầu

python src/train.py

# Chạy API server

python src/api.py

ML API sẽ chạy trên: [http://localhost:8000](http://localhost:8000/)

**D. Chạy Frontend (React)**

cd frontend

# Cài đặt dependencies

npm install

# Chạy development server

npm run dev

Frontend sẽ chạy trên: [http://localhost:3000](http://localhost:3000/)

**Lưu ý:** Khi chạy thủ công, bạn cần cấu hình CORS và API endpoints trong frontend:

* Tạo file .env.local trong folder frontend
* Thêm: VITE\_API\_URL=http://localhost:8080/api
* Thêm: VITE\_ML\_API\_URL=http://localhost:8000

# Kết luận.

Đồ án là minh chứng cho việc kết hợp giữa Software Engineering và Machine Learning có thể tạo ra giá trị thực tế, giải quyết bài toán business một cách hiệu quả.

Mặc dù còn nhiều hạn chế và điểm cần cải thiện, nhưng hệ thống đã đạt được mục tiêu cốt lõi: xây dựng một E-Commerce platform hoạt động tốt với khả năng gợi ý sản phẩm thông minh dựa trên Machine Learning. Đây là nền tảng vững chắc để phát triển tiếp thành một sản phẩm hoàn thiện và có thể đưa vào sử dụng thực tế