TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP.HCM

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ ĐIỆN, ĐIỆN TỬ**



**ĐỒ ÁN TRÍ TUỆ NHÂN TẠO**

**TÊN ĐỀ TÀI**

**XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ SÁCH TÍCH HỢP HỆ THỐNG GỢI Ý SẢN PHẨM DỰA TRÊN THUẬT TOÁN COLLABORATIVE FILTERING**

**Giảng viên hướng dẫn:** Bùi Trọng Hiếu

**Nhóm sinh viên thực hiện:** 2251120219 Đoàn Minh Khôi

2251120271 Trần Gia Bảo

2251120249 Huỳnh Đại Thắng

079205013990 Phạm Hà Võ

072205012353 Ngô Hoài Trọng Phúc

***Thành phố Hồ Chí Minh, ngày tháng 1 năm 2026***

MỤC LỤC

[MỤC LỤC i](#_Toc220444864)

[DANH MỤC HÌNH ẢNH iii](#_Toc220444865)

[LỜI MỞ ĐẦU v](#_Toc220444866)

[LỜI CẢM ƠN vi](#_Toc220444867)

[PHÂN CÔNG CÔNG VIỆC CHO CÁC THÀNH VIÊN NHÓM 6](#_Toc220444868)

[CHƯƠNG 1. tổng quan 3](#_Toc220444869)

[1.1. Giới thiệu chung 3](#_Toc220444870)

[1.2. Khảo sát và phân tích các đề tài liên quan đến vấn đề cần giải quyết 3](#_Toc220444871)

[1.2.1. Vấn đề cần giải quyết 3](#_Toc220444872)

[1.2.2. Mục tiêu của đề tài 3](#_Toc220444873)

[CHƯƠNG 2. GIẢI PHÁP ĐỀ XUẤT 4](#_Toc220444874)

[2.1. Cơ sở lý thuyết 4](#_Toc220444875)

[2.2. Lịch sử phát triển 5](#_Toc220444876)

[2.3. Phương pháp / chỉ tiêu đánh giá 5](#_Toc220444877)

[2.3.1. Đánh giá mô hình (offline evaluation): 5](#_Toc220444878)

[2.3.2. Đánh giá trên hệ thống thực tế (online evaluation): 5](#_Toc220444879)

[2.4. Công thức 6](#_Toc220444880)

[2.4.1. User-based Collaborative Filtering (UserCF) 6](#_Toc220444881)

[2.4.1.1. Tính độ tương đồng 6](#_Toc220444882)

[2.4.1.2. Dự đoán cho user u với item i 6](#_Toc220444883)

[2.4.1.3. Chuẩn hóa & Ensemble 6](#_Toc220444884)

[2.4.2. Item-based Collaborative Filtering (ItemCF) 6](#_Toc220444885)

[2.4.2.1. Tính độ tương đồng 6](#_Toc220444886)

[2.4.2.2. Dự đoán cho user **u** với item **i** 7](#_Toc220444887)

[2.4.2.3. Chuẩn hóa & Ensemble 7](#_Toc220444888)

[2.4.3. Matrix Factorization (MF) 7](#_Toc220444889)

[2.4.3.1. Ý tưởng chính 7](#_Toc220444890)

[2.4.3.2. Hàm mất mát 8](#_Toc220444891)

[2.4.3.3. Thuật toán huấn luyện (SGD) 8](#_Toc220444892)

[2.4.3.4. Dự đoán 8](#_Toc220444893)

[2.5. Kết quả dự kiến của đề tài 8](#_Toc220444894)

[2.6. Công nghệ sử dụng 9](#_Toc220444895)

[2.7. Link cài đặt dự án 9](#_Toc220444896)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 9](#_Toc220444897)

[3.1. Phân tích chức năng 9](#_Toc220444898)

[3.1.1. Quản lý tài khoản và đăng nhập 9](#_Toc220444899)

[3.1.2. Duyệt và tìm kiếm sản phẩm 9](#_Toc220444900)

[3.1.3. Tương tác với sản phẩm (User Interactions) 9](#_Toc220444901)

[3.1.4. Giỏ hàng & đơn hàng 9](#_Toc220444902)

[3.1.5. Chức năng quản trị (Admin) 9](#_Toc220444903)

[3.1.6. Hệ thống gợi ý (Recommendation System) 10](#_Toc220444904)

[3.1.6.1. Thu thập dữ liệu hành vi từ bảng UserInteractions. 10](#_Toc220444905)

[3.1.6.2. Huấn luyện 3 mô hình: 10](#_Toc220444906)

[3.1.6.3. Đánh giá model: 10](#_Toc220444907)

[3.1.6.4. Triển khai mô hình qua API (FastAPI). 10](#_Toc220444908)

[3.1.6.5. Tích hợp kết quả gợi ý cá nhân hóa hiển thị trực tiếp trong website ASP.NET MVC. 11](#_Toc220444909)

[3.2. Thiết kế 11](#_Toc220444910)

[3.2.1. Kiến trúc tổng thể 11](#_Toc220444911)

[3.2.2. Biểu đồ chức năng (Use case chính) 11](#_Toc220444912)

[3.2.3. Thiết kế hệ thống gợi ý 12](#_Toc220444913)

[3.2.4. Sơ đồ phân rã chức năng 12](#_Toc220444914)

[3.2.5. Luồng dữ liệu chính 15](#_Toc220444915)

[3.2.6. Diagram 16](#_Toc220444916)

[3.2.7. ERD chi tiết 16](#_Toc220444917)

[3.2.8. Mô hình quan hệ (Relational Model) 17](#_Toc220444918)

[3.2.9. Chi tiết các bảng 18](#_Toc220444919)

[3.2.9.1. Bảng AspNetUsers 18](#_Toc220444920)

[3.2.9.2. Bảng AspNetRoles 19](#_Toc220444921)

[3.2.9.3. Bảng AspNetUserRoles 19](#_Toc220444922)

[3.2.9.4. Bảng AspNetUserClaims 19](#_Toc220444923)

[3.2.9.5. Bảng AspNetRoleClaims 20](#_Toc220444924)

[3.2.9.6. Bảng AspNetUserLogins 20](#_Toc220444925)

[3.2.9.7. Bảng AspNetUserTokens 20](#_Toc220444926)

[3.2.9.8. Bảng Genre 21](#_Toc220444927)

[3.2.9.9. Bảng Book 21](#_Toc220444928)

[3.2.9.10. Bảng Stock 21](#_Toc220444929)

[3.2.9.11. Bảng ShoppingCart 22](#_Toc220444930)

[3.2.9.12. Bảng CartDetail 22](#_Toc220444931)

[3.2.9.13. Bảng Order 22](#_Toc220444932)

[3.2.9.14. Bảng OrderDetail 23](#_Toc220444933)

[3.2.9.15. Bảng OrderStatus 24](#_Toc220444934)

[3.2.9.16. Bảng UserInteractions 24](#_Toc220444935)

[3.2.9.17. Bảng \_\_EFMigrationsHistory 25](#_Toc220444936)

[CHƯƠNG 4. KIỂM THỬ 25](#_Toc220444937)

[5. KẾT QUẢ, KẾT LUẬN/ HƯỚNG PHÁT TRIỂN 34](#_Toc220444938)

[Tài liệu tham khảo 37](#_Toc220444939)

DANH MỤC HÌNH ẢNH

[Hình 1. Sơ đồ phân rã chức năng 1. DFD mức 0 (Context Diagram) 12](#_Toc220444969)

[Hình 2. Sơ đồ phân rã chức năng 2. DFD mức 1 - Process 1.0: Quản lý tài khoản 13](#_Toc220444970)

[Hình 3. Sơ đồ phân rã chức năng 3. DFD mức 1 - Process 2.0: Quản lý sản phẩm 13](#_Toc220444971)

[Hình 4. Sơ đồ phân rã chức năng 4. DFD mức 1 - Process 3.0: Quản lý giỏ hàng và đơn hàng 13](#_Toc220444972)

[Hình 5. Sơ đồ phân rã chức năng 5. DFD mức 1 - Process 4.0: Thu thập hành vi người dùng 14](#_Toc220444973)

[Hình 6. Sơ đồ phân rã chức năng 6. DFD mức 1 - Process 5.0: Hệ thống gợi ý 14](#_Toc220444974)

[Hình 7. Luồng dữ liệu 1 15](#_Toc220444975)

[Hình 8. Luồng dữ liệu 1 15](#_Toc220444976)

[Hình 9. Sơ đồ phân rã chức năng Diagram 16](#_Toc220444977)

[Hình 10. Test API User-based Collaborative Filtering 25](#_Toc220444978)

[Hình 11. Test API Item-based Collaborative Filtering 26](#_Toc220444979)

[Hình 12. Test API Matrix Factorization 26](#_Toc220444980)

[Hình 13. Lấy dữ liệu từ trường Rating 27](#_Toc220444981)

[Hình 14. Lấy dữ liệu từ trường Score 28](#_Toc220444982)

[Hình 15. Ma trận Rating và Implicit 28](#_Toc220444983)

[Hình 16. Ma trận tương đồng 29](#_Toc220444984)

[Hình 17. Gợi ý sản phẩm 29](#_Toc220444985)

[Hình 18. Hiển thị sản phẩm gợi ý ra Website 29](#_Toc220444986)

[Hình 19. Trang chủ website 29](#_Toc220444987)

[Hình 20. Bảng xếp hạng sách bán chạy 30](#_Toc220444988)

[Hình 21. Trang sản phẩm 30](#_Toc220444989)

[Hình 22. Quản lí tài khoản 30](#_Toc220444990)

[Hình 23. Quản lí giỏ hàng 31](#_Toc220444991)

[Hình 24. Quản lí đơn hàng 31](#_Toc220444992)

[Hình 25. Quản lí của admin 32](#_Toc220444993)

[Hình 26. Quản lí danh sách đơn hàng 32](#_Toc220444994)

[Hình 27. Quản lí tồn kho 32](#_Toc220444995)

[Hình 28. Quản lí thể loại 33](#_Toc220444996)

[Hình 29. Quản lí sách 33](#_Toc220444997)

[Hình 30. Quản lí sách bán chạy theo khoảng thời gian nhất định 33](#_Toc220444998)

[Hình 31. Quản lí hệ thống gợi ý 34](#_Toc220444999)

[Hình 32. Đánh giá thuật toán gợi ý 34](#_Toc220445000)

LỜI MỞ ĐẦU

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ và thương mại điện tử ngày càng trở thành xu hướng tất yếu, việc xây dựng các hệ thống bán hàng trực tuyến thông minh và hiệu quả đã trở thành nhu cầu cấp thiết. Đặc biệt, với sự bùng nổ thông tin và lượng sản phẩm khổng lồ trên các nền tảng thương mại điện tử, người dùng thường gặp khó khăn trong việc tìm kiếm và lựa chọn sản phẩm phù hợp với nhu cầu và sở thích của mình. Chính vì vậy, các hệ thống gợi ý (Recommendation Systems) đã ra đời như một giải pháp công nghệ quan trọng, giúp cá nhân hóa trải nghiệm mua sắm và nâng cao sự hài lòng của khách hàng.

Nhận thức được tầm quan trọng của vấn đề này, nhóm chúng em đã lựa chọn đề tài **"Xây dựng hệ thống thương mại điện tử bán sách trực tuyến tích hợp hệ thống gợi ý thông minh"** cho bài tập lớn môn Trí tuệ nhân tạo. Dự án không chỉ tập trung vào việc xây dựng một nền tảng thương mại điện tử hoàn chỉnh với đầy đủ các chức năng cơ bản như quản lý sản phẩm, giỏ hàng, đơn hàng, thanh toán, mà còn đặc biệt chú trọng đến việc ứng dụng các thuật toán Machine Learning để xây dựng hệ thống gợi ý sách cá nhân hóa cho từng người dùng.

LỜI CẢM ƠN

Trước hết, nhóm 3 xin gửi lời cảm ơn chân thành và sâu sắc nhất tới thầy Bùi Trọng Hiếu, người đã tận tình hướng dẫn, định hướng và hỗ trợ nhóm em trong suốt quá trình thực hiện học phần “Trí tuệ nhân tạo”. Thầy đã không chỉ truyền đạt cho em những kiến thức chuyên môn quý giá mà còn đưa ra những góp ý chi tiết, giúp nhóm em hoàn thiện từng phần của đề tài một cách khoa học và hiệu quả hơn.

Nhờ sự chỉ dẫn tận tâm của thầy, nhóm 3 đã có thể hoàn thành đồ án với chủ đề “Xây dựng website thương mại điện tử sách tích hợp hệ thống gợi ý sản phẩm dựa trên thuật toán collaborative filtering”. Trong quá trình thực hiện, nhóm em đã học hỏi và nâng cao được nhiều kỹ năng quan trọng như: lập trình web với ASP.NET, thiết kế cơ sở dữ liệu, xây dựng chức năng giỏ hàng – thanh toán, tích hợp hệ thống gợi ý sản phẩm và phát triển các module quản lý trong hệ thống thương mại điện tử. Đây đều là những kiến thức hữu ích, góp phần củng cố nền tảng chuyên môn của em trong lĩnh vực công nghệ thông tin.

Nhóm 3 cũng xin gửi lời cảm ơn đến các thầy cô trong khoa đã tạo điều kiện học tập thuận lợi, cũng như truyền đạt cho em kiến thức trong suốt thời gian qua. Đồng thời, nhóm em trân trọng cảm ơn các bạn bè đã hỗ trợ, góp ý, và đồng hành cùng em trong quá trình xây dựng giao diện, xử lý lỗi và hoàn thiện hệ thống.

Đồ án không chỉ là một sản phẩm học thuật, mà còn là cơ hội để nhóm em tiếp cận thực tế, rèn luyện khả năng tự học, tư duy hệ thống và giải quyết vấn đề. Những trải nghiệm quý báu này sẽ là hành trang để em tự tin hơn trên con đường phát triển nghề nghiệp sau này.

Em xin chân thành cảm ơn!

PHÂN CÔNG CÔNG VIỆC CHO CÁC THÀNH VIÊN NHÓM

|  |  |  |
| --- | --- | --- |
| **MSSV** | **HỌ VÀ TÊN** | **CÔNG VIỆC** |
| 2251120219 | Đoàn Minh Khôi | Xây dựng Website ASP.NET và hỗ trợ xây dựng 3 thuật toán UserCF, ItemCF, Matrix Factorization |
| 079205013990 | Phạm Hà Võ | Xây dựng thuật toán UserCF |
| 072205012353 | Ngô Hoài Trọng Phúc | Xây dựng thuật toán MF |
| 2251120271 | Trần Gia Bảo | Xây dựng thuật toán ItemCF |
| 2251120249 | Huỳnh Đại Thắng | Đánh giá thuật toán thông qua các chỉ số PRECISION, RECALL, NDCG |

# tổng quan

## Giới thiệu chung

Trong kỷ nguyên thương mại điện tử, lượng sản phẩm trên các website ngày càng nhiều, gây khó khăn cho người dùng khi lựa chọn. Vì vậy, việc tích hợp hệ thống gợi ý sản phẩm là rất cần thiết nhằm cá nhân hóa trải nghiệm, tiết kiệm thời gian tìm kiếm và tăng doanh thu cho doanh nghiệp. Nhóm 3 quyết định xây dựng Website Bán Sách Trực Tuyến tích hợp hệ thống gợi ý sản phẩm dựa trên thuật toán Collaborative Filtering có tên là BookStore Online.

## Khảo sát và phân tích các đề tài liên quan đến vấn đề cần giải quyết

Hiện nay, nhiều nền tảng lớn (Amazon, Tiki, Fahasa…) đã áp dụng hệ thống gợi ý dựa trên User-based Collaborative Filtering, Item-based Collaborative Filtering và Matrix Factorization để cá nhân hóa trải nghiệm khách hàng. Tuy nhiên, trong phạm vi học tập và nghiên cứu, số lượng đề tài tích hợp gợi ý trực tiếp vào một website thương mại điện tử hoàn chỉnh (bao gồm quản lý tài khoản, sản phẩm, giỏ hàng, đơn hàng và hệ thống gợi ý) còn hạn chế. Đề tài tập trung vào việc xây dựng một website bán sách sử dụng ASP.NET MVC kết hợp hệ thống gợi ý dựa trên Collaborative Filtering sử dụng Python. Người dùng khi xem, thêm giỏ hàng, mua hàng hay đánh giá sản phẩm sẽ được ghi nhận hành vi, từ đó mô hình học máy xử lý và đưa ra các gợi ý cá nhân hóa.

### Vấn đề cần giải quyết

* Người dùng thường bỏ lỡ những cuốn sách phù hợp do không có gợi ý cá nhân hóa.
* Hệ thống tìm kiếm/danh mục truyền thống (dựa trên từ khóa hoặc lọc tĩnh) không tận dụng được lịch sử hành vi của người dùng.
* Cần một cơ chế thu thập, xử lý hành vi và trả về gợi ý theo thời gian thực/near-real-time phục vụ giao diện web.

### Mục tiêu của đề tài

Phát triển một e-commerce website bán sách bằng ASP.NET Core MVC tích hợp một hệ thống gợi ý sản phẩm (recommendation system) bằng Python, cụ thể:

1. Thu thập đầy đủ hành vi người dùng (xem, thêm giỏ, mua, đánh giá).
2. Nghiên cứu, hiện thực và so sánh ba phương pháp phổ biến: User-based CF, Item-based CF, Matrix Factorization.
3. Kết hợp cả implicit feedback (hành vi) và explicit feedback (đánh giá sao).
4. Triển khai mô hình dưới dạng API (FastAPI) để website (ASP.NET MVC) gọi lấy gợi ý.
5. Đánh giá chất lượng gợi ý bằng các chỉ số phù hợp (Precision@K, Recall@K, NDCG@K):

* Precision@K – Precision at K: trong top-K item gợi ý, có bao nhiêu phần trăm là item đúng.
* Recall@K – Recall at K: trong tất cả item đúng (ground truth), có bao nhiêu phần trăm được xuất hiện trong top-K gợi ý.
* NDCG@K – Normalized Discounted Cumulative Gain at K: đánh giá chất lượng xếp hạng của top-K gợi ý. Không chỉ quan tâm item đúng có trong danh sách, mà còn xem chúng nằm ở vị trí nào. Công thức tính NDCG@K:

1. DCG@K (Discounted Cumulative Gain)



Trong đó:

* : vị trí trong danh sách gợi ý (1→K).
* : mức độ liên quan (relevance) của item tại vị trí .
* Với implicit feedback (thích / không thích): .
* Với rating:  có thể là điểm số (vd: 1–5).
* : yếu tố giảm dần trọng số theo vị trí (item ở đầu được coi trọng hơn).

1. IDCG@K (Ideal DCG)

khi danh sách gợi ý được sắp xếp lý tưởng (item đúng ở đầu tiên)

* Tức là sắp tất cả item đúng vào trước trong top-K.

1. NDCG@K (Normalized DCG)



* Chuẩn hóa về khoảng 0 → 1.
* Nếu NDCG@K = 1 → danh sách gợi ý hoàn hảo (item đúng ở đầu tiên).

# GIẢI PHÁP ĐỀ XUẤT

## Cơ sở lý thuyết

Hệ thống gợi ý (Recommendation System) là một nhánh quan trọng của Machine Learning, giúp dự đoán sở thích của người dùng để đưa ra sản phẩm phù hợp. Trong đề tài này, cơ sở lý thuyết tập trung vào **Collaborative Filtering (CF)** và **Matrix Factorization (MF)**:

* **User-based Collaborative Filtering (UserCF):** Gợi ý sản phẩm dựa trên sự tương đồng giữa người dùng. Nếu hai người dùng có hành vi, đánh giá tương tự nhau, sản phẩm mà một người quan tâm sẽ có khả năng cao được gợi ý cho người còn lại.
* **Item-based Collaborative Filtering (ItemCF):** Gợi ý sản phẩm dựa trên sự tương đồng giữa các sản phẩm. Nếu hai sản phẩm được nhiều người dùng cùng quan tâm/mua, thì khi người dùng xem sản phẩm A, hệ thống sẽ gợi ý thêm sản phẩm B.
* **Matrix Factorization (MF):**Một phương pháp học ẩn, phân rã ma trận User-Item thành các vector đặc trưng tiềm ẩn. Phương pháp này giải quyết vấn đề dữ liệu thưa (sparse data) và thường cho kết quả gợi ý chính xác hơn.

Ngoài ra, đề tài còn sử dụng:

* **Explicit Feedback (đánh giá số sao):** người dùng trực tiếp chấm điểm (1–5 sao).
* **Implicit Feedback (hành vi):** các hành động như xem sách, thêm giỏ hàng, mua hàng.
* **ASP.NET MVC** để xây dựng website và quản lý logic người dùng.
* **FastAPI (Python)** để triển khai API mô hình gợi ý, kết nối trực tiếp với website ASP.NET.

## Lịch sử phát triển

Hệ thống gợi ý bắt đầu được nghiên cứu từ những năm 1990, tiêu biểu là **GroupLens (1994)** với gợi ý tin tức. Đến đầu những năm 2000, Amazon áp dụng **Item-based CF** vào thương mại điện tử, mở ra kỷ nguyên cá nhân hóa trải nghiệm mua sắm.

Sau đó, với sự phát triển của dữ liệu lớn và Machine Learning, các phương pháp **Matrix Factorization** trở thành trụ cột trong nhiều nền tảng (Netflix Prize 2006 là ví dụ nổi bật). Hiện nay, các công ty lớn (Netflix, Spotify, Shopee, Tiki, Fahasa) đều tích hợp hệ thống gợi ý dựa trên nhiều mô hình kết hợp.

Trong phạm vi nghiên cứu học thuật, nhiều đề tài trong nước đã đề cập đến gợi ý sản phẩm hoặc gợi ý nội dung, nhưng thường tập trung vào mô hình thuật toán mà ít tích hợp vào một hệ thống web thương mại điện tử hoàn chỉnh. Đề tài này kế thừa lý thuyết từ các nghiên cứu trước và áp dụng trực tiếp vào một website bán sách.

## Phương pháp / chỉ tiêu đánh giá

Để đánh giá hệ thống gợi ý, đề tài sử dụng cả hai hướng:

### Đánh giá mô hình (offline evaluation):

* + **Precision@K:** tỉ lệ gợi ý đúng trong top-K.
  + **Recall@K:** mức độ bao phủ sản phẩm đúng trong top-K.
  + **NDCG@K:** đánh giá chất lượng thứ hạng của danh sách gợi ý.

### Đánh giá trên hệ thống thực tế (online evaluation):

* + Mức độ đa dạng và hợp lý của danh sách gợi ý.
  + Trải nghiệm thực tế của người dùng (hữu ích, dễ tìm sách phù hợp).
  + Ảnh hưởng đến hành vi mua sắm (tăng lượt xem, tăng tỉ lệ mua).

Dữ liệu để đánh giá được lấy trực tiếp từ **bảng UserInteractions**, ghi nhận hành vi:

* Xem sách → +1 điểm
* Thêm giỏ hàng → +3 điểm
* Mua hàng → +5 điểm
* Đánh giá sao → từ -10 đến +10 tùy mức sao (1 sao -10 điểm, 2 sao -5 điểm, 3 sao 0 điểm, 4 sao + 5 điểm, 5 sao +10 điểm )

## Công thức

### User-based Collaborative Filtering (UserCF)

#### Tính độ tương đồng

* Explicit (rating): Pearson correlation



Trong đó:

* + : tập user cùng đánh giá cả 2 item .
  + : rating của user cho item .
  + : trung bình rating của item .
  + Implict(hành vi): Cosine similarity



Trong đó vector hành vi theo từng user: view = 1, add\_to\_cart = 3, purchase = 5.

#### Dự đoán cho user u với item i

* Từ explicit:



* Từ implicit:



#### Chuẩn hóa & Ensemble

* Chuẩn hóa explicit về [0, 1]:



* Chuẩn hóa implicit về [0, 1]:



* Kết hợp (ensemble):



### Item-based Collaborative Filtering (ItemCF)

#### Tính độ tương đồng

* Explicit (rating): Pearson correlation



Trong đó:

* : tập user cùng đánh giá cả 2 item i, j.
* : rating của user cho item.
* : trung bình rating của item .
* **Implicit (hành vi): Cosine similarity**



Trong đó vector hành vi theo từng user: view = 1, add\_to\_cart = 3, purchase = 5.

#### Dự đoán cho user **u** với item **i**

* **Từ explicit:**



* **Từ implicit:**



#### Chuẩn hóa & Ensemble

* Chuẩn hóa explicit về [0, 1]:



* Chuẩn hóa implicit về [0, 1]:



* Kết hợp (ensemble):



### Matrix Factorization (MF)

#### Ý tưởng chính

* Biểu diễn User và Item dưới dạng vector trong không gian tiềm ẩn (latent space)
* Ma trận tương tác R (user-item) được xấp xỉ bởi:



Trong đó:

* + : ma trận đặ trưng ẩn của user.
  + :ma trận đặc trưng ẩn của item.
  + : số chiều latent factors.

#### Hàm mất mát

Sử dụng **Mean Squared Error (MSE)** có regularization:



Trong đó:

* : tập các cặp (user, item) có tương tác.
* : hệ số regularization để tránh overfitting.

#### Thuật toán huấn luyện (SGD)

* Với mỗi  có :
* **Sai số:**



* **Cập nhật tham số:**





Trong đó:  là learning rate.

#### Dự đoán

* Điểm dự đoán (raw score) cho user  với item :



* Sau đó chuẩn hóa giá trị về khoảng [0, 1]:



## Kết quả dự kiến của đề tài

Đề tài hướng tới xây dựng một website bán sách có đầy đủ chức năng cơ bản (quản lý tài khoản, sản phẩm, giỏ hàng, đơn hàng, admin) và tích hợp hệ thống gợi ý sản phẩm cá nhân hóa.

Các kết quả kỳ vọng:

* Xây dựng thành công website bán sách bằng **ASP.NET MVC**.
* Lưu trữ và xử lý dữ liệu hành vi người dùng qua **UserInteractions**.
* Huấn luyện và triển khai được 3 mô hình gợi ý: **UserCF, ItemCF, Matrix Factorization**.
* Tích hợp **API FastAPI** vào website để trả về gợi ý theo từng người dùng.
* Hiển thị danh sách gợi ý sách cá nhân hóa trong giao diện (trang chủ, trang chi tiết sản phẩm).
* Đánh giá được hiệu quả của từng mô hình dựa trên chỉ tiêu đã nêu và chọn được mô hình huấn luyện phù hợp.

Kết quả này không chỉ mang ý nghĩa học thuật mà còn có khả năng mở rộng thành giải pháp thực tế trong thương mại điện tử.

## Công nghệ sử dụng

* Website e-commerce bán sách: ASP.NET MVC.
* Hệ thống gợi ý: Python.
* Test API: Postman.
* Cơ sở dữ liệu: SQL Server Management Studio

## Link cài đặt dự án

*https://github.com/MinhKhoi74/E-commerce\_BookStore-Recommend-system-MK.git*

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân tích chức năng

Hệ thống website thương mại điện tử kết hợp hệ thống gợi ý được phân tích thành các nhóm chức năng chính:

### Quản lý tài khoản và đăng nhập

* **Đăng ký/Đăng nhập** tài khoản thông qua **ASP.NET Identity**.
* Hỗ trợ **đăng nhập bằng Google** (External Login), link xác thực được gửi đến hộp thư gmail. Chỉ cho phép đăng nhập sau khi xác thực thành công.
* Người dùng quản lý được **thông tin cá nhân**: email, số điện thoại, mật khẩu.

### Duyệt và tìm kiếm sản phẩm

* Hiển thị danh mục sách theo **thể loại**.
* Trang chi tiết sách gồm hình ảnh, thông tin chi tiết, đánh giá.
* Tìm kiếm theo từ khóa.

### Tương tác với sản phẩm (User Interactions)

* Xem chi tiết sách → lưu hành vi (Score = 1).
* Thêm vào giỏ hàng → lưu hành vi (Score = 3).
* Thanh toán mua hàng → lưu hành vi (Score = 5).
* Đánh giá sản phẩm sau khi mua (1–5 sao) → **explicit feedback** (quy đổi thành giá trị từ -10 đến +10 (1 sao -10 điểm, 2 sao -5 điểm, 3 sao 0 điểm, 4 sao + 5 điểm, 5 sao +10 điểm)).

### Giỏ hàng & đơn hàng

* Quản lý giỏ hàng: xem, thêm, xóa, cập nhật số lượng sách.
* Thanh toán (COD).
* Xem lại **lịch sử mua hàng** và chi tiết đơn hàng.

### Chức năng quản trị (Admin)

* Quản lý đơn hàng: xem, sửa trạng thái đơn hàng, chỉnh sửa trạng thái thanh toán.
* Quản lý tồn kho: tìm kiếm sản phẩm, cập nhật số lượng tồn kho.
* Quản lý thể loại sách: xem, thêm, sửa, xóa.
* Quản lý sách: xem, thêm, sửa, xóa sách.
* Thống kê sách bán chạy trong khoảng thời gian tùy chọn.

### Hệ thống gợi ý (Recommendation System)

#### Thu thập dữ liệu hành vi từ bảng UserInteractions.

* Chia dữ liệu train/test
* Mỗi user giữ lại ~20% interactions làm test, phần còn lại cho train
* Mục đích: test dữ liệu này để kiểm tra model dự đoán/gợi ý có khớp thực tế không.

#### Huấn luyện 3 mô hình:

* User-based Collaborative Filtering.
* Item-based Collaborative Filtering.
* Matrix Factorization.

#### Đánh giá model:

* Với mỗi user trong test set:
* Lấy danh sách gợi ý top-K từ model.
* So sánh với items thực sự liên quan.
* Tính NDCG@K.
* Trung bình NDCG@K trên tất cả users = điểm cuối cùng.
* Model nào có NDCG@K cao nhất = model tốt nhất cho bài toán gợi ý.

#### Triển khai mô hình qua API (FastAPI).

* /recommend/personal/{user\_id}
* Mục đích: Trả về danh sách sách gợi ý dựa trên thuật toán đã được lựa chọn tùy theo số hành vi của người dùng
* Input:
* user\_id (người dùng cần gợi ý)
* Output: Danh sách sách gợi ý và các thông số của người dùng như: thuật toán phù hợp, loại người dùng, điểm tương tác.
* /recommend/usercf/{user\_id}
* Mục đích: Gợi ý dựa trên User-based Collaborative Filtering (UserCF).
* Input:
* user\_id
* Output: Danh sách sách gợi ý từ UserCF.
* /evaluate
* Mục đích: Đánh giá thuật toán
* Input:
* Output: đánh giá thuật toán theo các chỉ số Precision@K Recall@K NDCG@K
* /recommend/itemcf/{user\_id}
* Mục đích: Gợi ý dựa trên Item-based Collaborative Filtering (ItemCF).
* Input:
* user\_id
* Output: Danh sách sách gợi ý từ ItemCF.
* /recommend/mf/{user\_id}
* Mục đích: Gợi ý dựa trên Matrix Factorization (MF).
* Input:
* user\_id
* Output: Danh sách sách gợi ý từ MF.
* /recommender/config/{user\_id}
* Mục đích: Xem các thông số ảnh hưởng đến thuật toán
* Input:
* user\_id
* Output: Các thông số như: tổng số hành vi, tổng người dùng, tổng đánh giá, tổng điểm hành vi, ngưỡng phân loại người dùng, trọng số của các hành vi.

#### Tích hợp kết quả gợi ý cá nhân hóa hiển thị trực tiếp trong website ASP.NET MVC.

## Thiết kế

### Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình **kết hợp Web App ASP.NET MVC và dịch vụ AI Python**:

* **ASP.NET MVC**: chịu trách nhiệm giao diện web, xử lý nghiệp vụ (giỏ hàng, thanh toán, quản lý tài khoản, đơn hàng, tìm kiếm, giao diện, thống kê, admin).
* **SQL Server**: lưu trữ dữ liệu sản phẩm, người dùng, đơn hàng, bảng UserInteractions.
* **Python (FastAPI)**: huấn luyện và triển khai mô hình gợi ý.
* **Giao tiếp ASP.NET ↔ FastAPI**: thông qua REST API.

### Biểu đồ chức năng (Use case chính)

* Người dùng:
* Đăng nhập, đăng ký.
* Tìm kiếm sách.
* Mua hàng.
* Xem lịch sử đơn hàng.
* Đánh giá sản phẩm (1-5 sao).
* Nhận gợi ý sách cá nhân hóa.
* Admin:
  + Chức năng giống người dùng
  + Quản lý người dùng
  + Quản lý đơn hàng (xem, cập nhật trạng thái)
  + Quản lý tồn kho
  + Thống kê doanh thu, sách, tồn kho, sách
  + Quản lý danh mục sách (CRUD)
* Hệ thống gợi ý: xử lý dữ liệu tương tác, huấn luyện mô hình, trả kết quả gợi ý.

### Thiết kế hệ thống gợi ý

* Pipeline:
  1. Thu thập dữ liệu từ DB.
  2. Tách **explicit feedback** (Rating) và **implicit feedback** (Score).
  3. Huấn luyện mô hình (UserCF, ItemCF, MF).
  4. Cung cấp API dự đoán top-N sản phẩm cho mỗi user.
  5. ASP.NET gọi API để hiển thị gợi ý cá nhân hóa trên giao diện.

### Sơ đồ phân rã chức năng

A screenshot of a computer

AI-generated content may be incorrect.

* + - * 1. Sơ đồ phân rã chức năng 1. DFD mức 0 (Context Diagram)

A yellow circle with black text and arrows

AI-generated content may be incorrect.

* + - * 1. Sơ đồ phân rã chức năng 2. DFD mức 1 - Process 1.0: Quản lý tài khoản

A diagram of a circle with text

AI-generated content may be incorrect.

* + - * 1. Sơ đồ phân rã chức năng 3. DFD mức 1 - Process 2.0: Quản lý sản phẩm

A diagram of a diagram

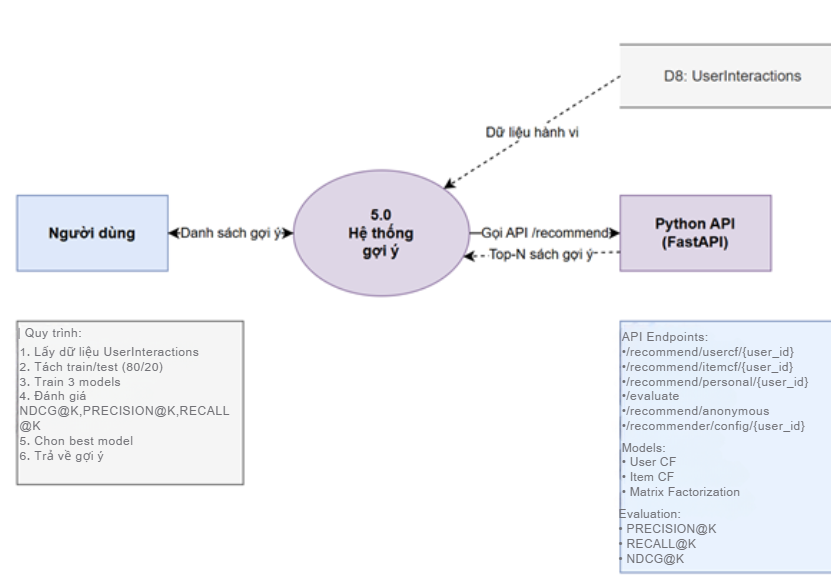
AI-generated content may be incorrect.

* + - * 1. Sơ đồ phân rã chức năng 4. DFD mức 1 - Process 3.0: Quản lý giỏ hàng và đơn hàng

A diagram of a diagram

AI-generated content may be incorrect.

* + - * 1. Sơ đồ phân rã chức năng 5. DFD mức 1 - Process 4.0: Thu thập hành vi người dùng



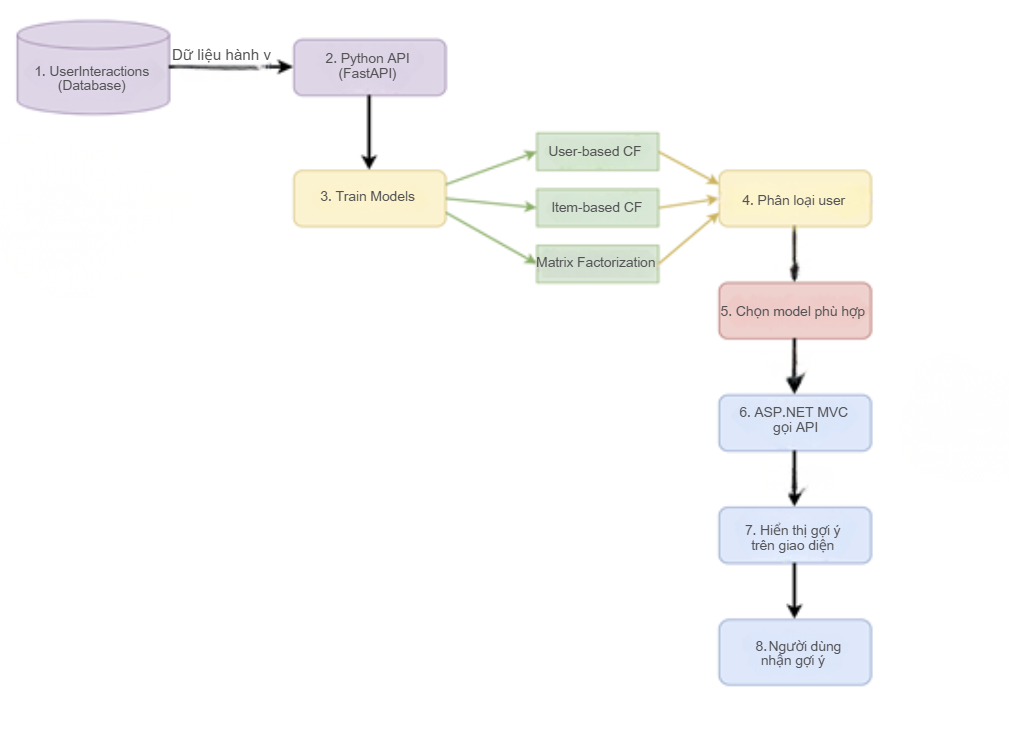
* + - * 1. Sơ đồ phân rã chức năng 6. DFD mức 1 - Process 5.0: Hệ thống gợi ý

### Luồng dữ liệu chính

A diagram of a flowchart

AI-generated content may be incorrect.

* + - * 1. Luồng dữ liệu 1



* + - * 1. Luồng dữ liệu 1

### Diagram



* + - * 1. Sơ đồ phân rã chức năng Diagram

### ERD chi tiết

**Các thực thể chính:**

1. **AspNetUsers** - Quản lý người dùng
2. **Genre** - Thể loại sách
3. **Book** - Sản phẩm sách
4. **CartDetail** - Chi tiết giỏ hàng
5. **Order** - Đơn hàng
6. **OrderDetail** - Chi tiết đơn hàng
7. **OrderStatus** - Trạng thái đơn hàng
8. **UserInteractions** - Tương tác người dùng
9. **Stock** - Tồn kho

**Các mối quan hệ chính:**

* **AspNetUsers (1) ←→ (N) Order**: Một người dùng có nhiều đơn hàng
* **AspNetUsers (1) ←→ (N) CartDetail**: Một người dùng có nhiều sản phẩm trong giỏ
* **AspNetUsers (1) ←→ (N) UserInteractions**: Một người dùng có nhiều tương tác
* **Genre (1) ←→ (N) Book**: Một thể loại có nhiều sách
* **Book (1) ←→ (N) CartDetail**: Một sách có thể nằm trong nhiều giỏ hàng
* **Book (1) ←→ (N) OrderDetail**: Một sách có thể nằm trong nhiều đơn hàng
* **Book (1) ←→ (N) UserInteractions**: Một sách có nhiều tương tác
* **Book (1) ←→ (1) Stock**: Một sách có một bản ghi tồn kho
* **Order (1) ←→ (N) OrderDetail**: Một đơn hàng có nhiều sản phẩm
* **Order (N) ←→ (1) OrderStatus**: Nhiều đơn hàng có cùng trạng thái

### Mô hình quan hệ (Relational Model)

**AspNetUsers**(Id, UserName, Email, PhoneNumber, PasswordHash, EmailConfirmed, NormalizedUserName, NormalizedEmail, SecurityStamp, ConcurrencyStamp, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnabled, AccessFailedCount)

**Genre**(Id, GenreName)

**Book**(Id, BookName, AuthorName, Price, Image, *GenreId*, Description)

* FK: GenreId → Genre(Id)

**Stock**(Id, *BookId*, Quantity)

* FK: BookId → Book(Id)

**ShoppingCart**(Id, *UserId*, IsDeleted)

* FK: UserId → AspNetUsers(Id)

**CartDetail**(Id, *ShoppingCartId*, *BookId*, Quantity, UnitPrice)

* FK: ShoppingCartId → ShoppingCart(Id)
* FK: BookId → Book(Id)

**Order**(Id, *UserId*, CreateDate, *OrderStatusId*, IsDeleted, Address, Email, MobileNumber, Name, PaymentMethod, IsPaid)

* FK: UserId → AspNetUsers(Id)
* FK: OrderStatusId → OrderStatus(Id)

**OrderDetail**(Id, *OrderId*, *BookId*, Quantity, UnitPrice)

* FK: OrderId → Order(Id)
* FK: BookId → Book(Id)

**OrderStatus**(Id, StatusId, StatusName)

**UserInteractions**(InteractionId, *UserId*, *BookId*, Rating, InteractionDate, Score)

* FK: UserId → AspNetUsers(Id)
* FK: BookId → Book(Id)

**AspNetRoles**(Id, Name, NormalizedName, ConcurrencyStamp)

**AspNetUserRoles**(*UserId*, *RoleId*)

* FK: UserId → AspNetUsers(Id)
* FK: RoleId → AspNetRoles(Id)

**AspNetRoleClaims**(Id, *RoleId*, ClaimType, ClaimValue)

* FK: RoleId → AspNetRoles(Id)

**AspNetUserClaims**(Id, *UserId*, ClaimType, ClaimValue)

* FK: UserId → AspNetUsers(Id)

**AspNetUserLogins**(LoginProvider, ProviderKey, ProviderDisplayName, *UserId*)

* FK: UserId → AspNetUsers(Id)

**AspNetUserTokens**(*UserId*, LoginProvider, Name, Value)

* FK: UserId → AspNetUsers(Id)

**\_\_EFMigrationsHistory**(MigrationId, ProductVersion)

### Chi tiết các bảng

#### Bảng AspNetUsers

Quản lý thông tin người dùng (ASP.NET Identity).

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | nvarchar(450) | PK, NOT NULL | ID người dùng (GUID) |
| UserName | nvarchar(256) | UNIQUE | Tên đăng nhập |
| Email | nvarchar(256) | UNIQUE | Email người dùng |
| NormalizedUserName | nvarchar(256) | NULL | Username chuẩn hóa |
| NormalizedEmail | nvarchar(256) | NULL | Email chuẩn hóa |
| EmailConfirmed | bit | NOT NULL | Đã xác thực email |
| PasswordHash | nvarchar(max) | NULL | Mật khẩu đã hash |
| SecurityStamp | nvarchar(max) | NULL | Stamp bảo mật |
| ConcurrencyStamp | nvarchar(max) | NULL | Stamp đồng thời |
| PhoneNumber | nvarchar(max) | NULL | Số điện thoại |
| PhoneNumberConfirmed | bit | NOT NULL | Đã xác thực SĐT |
| TwoFactorEnabled | bit | NOT NULL | Bật xác thực 2 yếu tố |
| LockoutEnd | datetimeoffset | NULL | Thời gian kết thúc khóa |
| LockoutEnabled | bit | NOT NULL | Cho phép khóa tài khoản |
| AccessFailedCount | int | NOT NULL | Số lần đăng nhập sai |

#### Bảng AspNetRoles

Quản lý vai trò người dùng.

| **Tên cột** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- |
| Id | PK, NOT NULL | ID vai trò |
| Name | UNIQUE | Tên vai trò (Admin, User) |
| NormalizedName | NULL | Tên vai trò chuẩn hóa |
| ConcurrencyStamp | NULL | Stamp đồng thời |

#### Bảng AspNetUserRoles

Liên kết người dùng với vai trò (Many-to-Many).

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| UserId | nvarchar(450) | PK, FK, NOT NULL | ID người dùng |
| RoleId | nvarchar(450) | PK, FK, NOT NULL | ID vai trò |

#### Bảng AspNetUserClaims

Lưu trữ claims của người dùng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID claim |
| UserId | nvarchar(450) | FK, NOT NULL | ID người dùng |
| ClaimType | nvarchar(max) | NULL | Loại claim |
| ClaimValue | nvarchar(max) | NULL | Giá trị claim |

#### Bảng AspNetRoleClaims

Lưu trữ claims của vai trò.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID claim |
| RoleId | nvarchar(450) | FK, NOT NULL | ID vai trò |
| ClaimType | nvarchar(max) | NULL | Loại claim |
| ClaimValue | nvarchar(max) | NULL | Giá trị claim |

#### Bảng AspNetUserLogins

Quản lý đăng nhập từ bên ngoài (Google, Facebook).

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| LoginProvider | nvarchar(450) | PK, NOT NULL | Nhà cung cấp đăng nhập |
| ProviderKey | nvarchar(450) | PK, NOT NULL | Khóa từ provider |
| ProviderDisplayName | nvarchar(max) | NULL | Tên hiển thị provider |
| UserId | nvarchar(450) | FK, NOT NULL | ID người dùng |

#### Bảng AspNetUserTokens

Lưu token xác thực của người dùng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| UserId | nvarchar(450) | PK, FK, NOT NULL | ID người dùng |
| LoginProvider | nvarchar(450) | PK, NOT NULL | Nhà cung cấp đăng nhập |
| Name | nvarchar(450) | PK, NOT NULL | Tên token |
| Value | nvarchar(max) | NULL | Giá trị token |

#### Bảng Genre

Lưu trữ thể loại sách.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID thể loại |
| GenreName | nvarchar(40) | NOT NULL, UNIQUE | Tên thể loại |

#### Bảng Book

Lưu trữ thông tin sách.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID sách |
| BookName | nvarchar(max) | NOT NULL | Tên sách |
| AuthorName | nvarchar(max) | NOT NULL | Tên tác giả |
| Price | decimal(18,2) | NOT NULL, CHECK (Price >= 0) | Giá bán |
| Image | nvarchar(max) | NULL | Đường dẫn hình ảnh |
| GenreId | int | FK, NOT NULL | ID thể loại |
| Description | nvarchar(max) | NULL | Mô tả sách |

#### Bảng Stock

Quản lý tồn kho.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID tồn kho |
| BookId | int | FK, NOT NULL, UNIQUE | ID sách |
| Quantity | int | NOT NULL, CHECK (Quantity >= 0) | Số lượng tồn |

#### Bảng ShoppingCart

Quản lý giỏ hàng của người dùng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID giỏ hàng |
| UserId | nvarchar(450) | FK, NOT NULL, UNIQUE | ID người dùng |
| IsDeleted | bit | NOT NULL, DEFAULT 0 | Đã xóa chưa |

#### Bảng CartDetail

Chi tiết sản phẩm trong giỏ hàng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID chi tiết giỏ |
| ShoppingCartId | int | FK, NOT NULL | ID giỏ hàng |
| BookId | int | FK, NOT NULL | ID sách |
| Quantity | int | NOT NULL, CHECK (Quantity > 0) | Số lượng |
| UnitPrice | decimal(18,2) | NOT NULL | Đơn giá |

Ràng buộc: UNIQUE(ShoppingCartId, BookId)

#### Bảng Order

Quản lý đơn hàng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID đơn hàng |
| UserId | nvarchar(450) | FK, NOT NULL | ID người dùng |
| CreateDate | datetime | NOT NULL, DEFAULT GETDATE() | Ngày tạo |
| OrderStatusId | int | FK, NOT NULL | ID trạng thái |
| IsDeleted | bit | NOT NULL, DEFAULT 0 | Đã xóa |
| Address | nvarchar(max) | NULL | Địa chỉ giao hàng |
| Email | nvarchar(256) | NULL | Email |
| MobileNumber | nvarchar(20) | NULL | Số điện thoại |
| Name | nvarchar(max) | NULL | Tên người nhận |
| PaymentMethod | nvarchar(50) | NULL | Phương thức thanh toán |
| IsPaid | bit | NOT NULL, DEFAULT 0 | Đã thanh toán |

#### Bảng OrderDetail

Chi tiết sản phẩm trong đơn hàng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID chi tiết đơn |
| OrderId | int | FK, NOT NULL | ID đơn hàng |
| BookId | int | FK, NOT NULL | ID sách |
| Quantity | int | NOT NULL, CHECK (Quantity > 0) | Số lượng |
| UnitPrice | decimal(18,2) | NOT NULL | Đơn giá |

#### Bảng OrderStatus

Trạng thái đơn hàng.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| Id | int | PK, IDENTITY, NOT NULL | ID trạng thái |
| StatusId | int | NOT NULL, UNIQUE | Mã trạng thái |
| StatusName | nvarchar(20) | NOT NULL | Tên trạng thái |

Giá trị mẫu: Pending, Shipped, Delivered, Cancelled

#### Bảng UserInteractions

Lưu hành vi tương tác của người dùng với sách (cho hệ thống gợi ý).

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| InteractionId | int | PK, IDENTITY, NOT NULL | ID tương tác |
| UserId | nvarchar(450) | FK, NOT NULL | ID người dùng |
| BookId | int | FK, NOT NULL | ID sách |
| Rating | int | NULL, CHECK (Rating BETWEEN 1 AND 5) | Đánh giá (1-5 sao) |
| InteractionDate | datetime | NOT NULL, DEFAULT GETDATE() | Ngày tương tác |
| Score | int | NULL | Điểm implicit feedback |

**Giải thích Score:**

* View: +1
* AddToCart: +3
* Purchase: +5
* Rating: -10 đến +10 (tùy số sao)

#### Bảng \_\_EFMigrationsHistory

Lưu lịch sử migration của Entity Framework.

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- |
| MigrationId | nvarchar(150) | PK, NOT NULL | ID migration |
| ProductVersion | nvarchar(32) | NOT NULL | Phiên bản EF Core |

# KIỂM THỬ

A screenshot of a computer

AI-generated content may be incorrect.

* + - * 1. Test API User-based Collaborative Filtering

**A screenshot of a computer

AI-generated content may be incorrect.**

* + - * 1. Test API Item-based Collaborative Filtering

**A screenshot of a computer

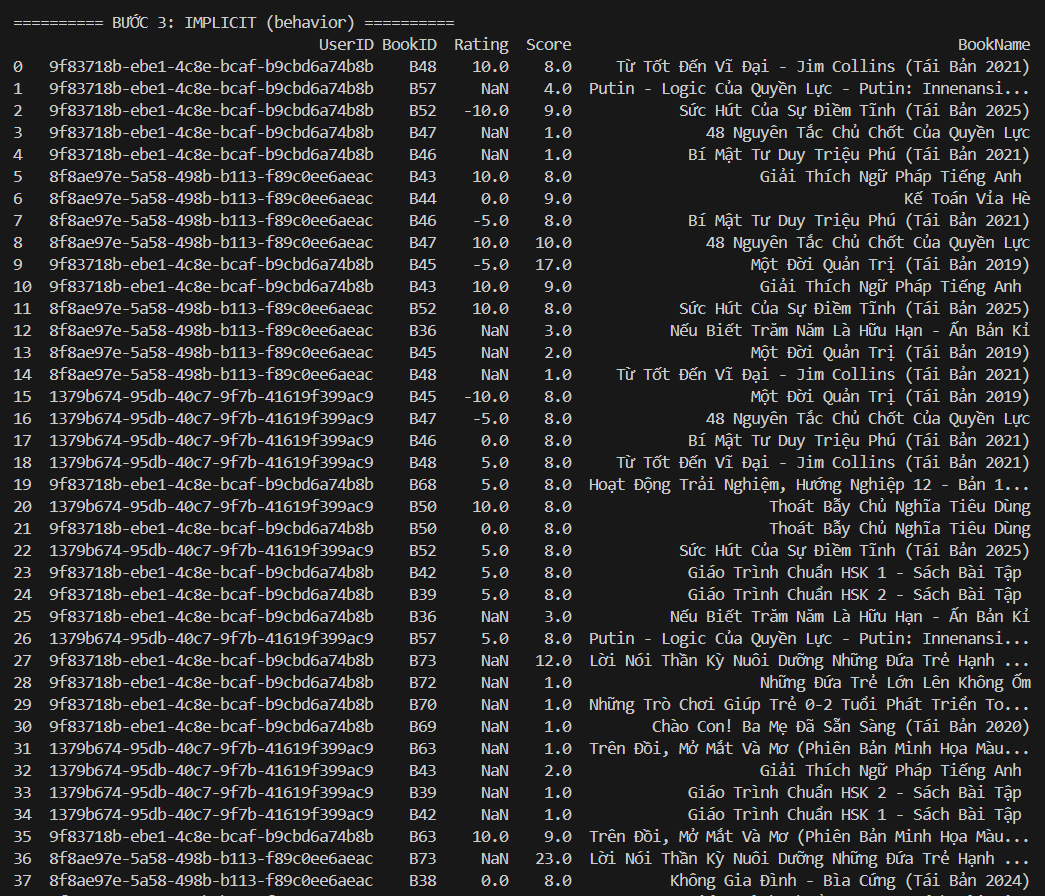
AI-generated content may be incorrect.**

* + - * 1. Test API Matrix Factorization

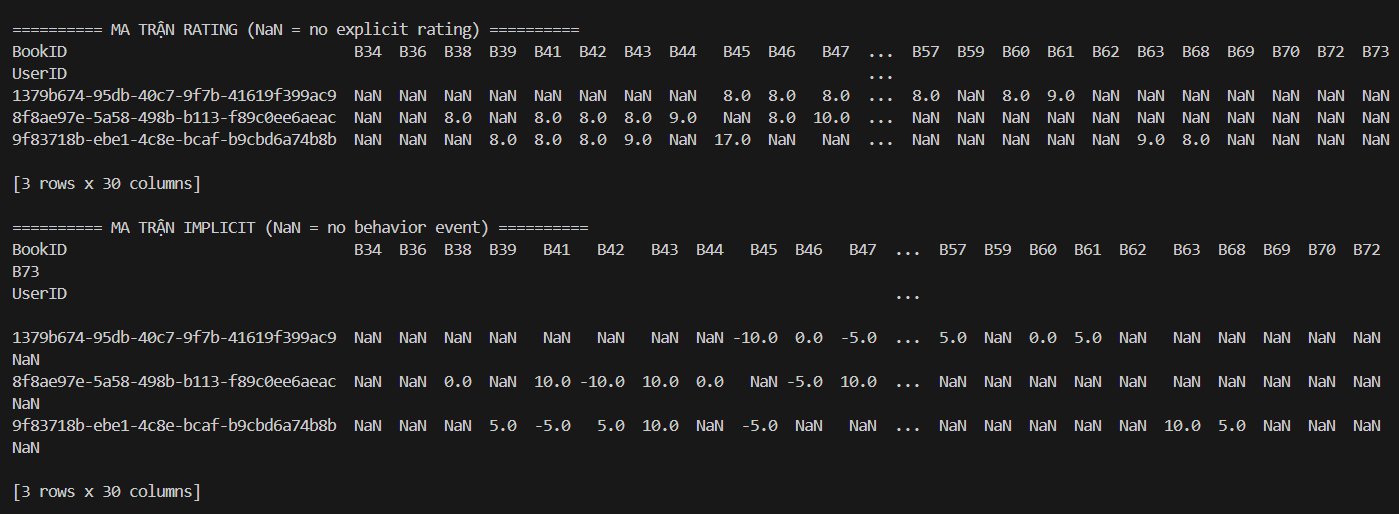
**A screenshot of a computer screen

AI-generated content may be incorrect.**

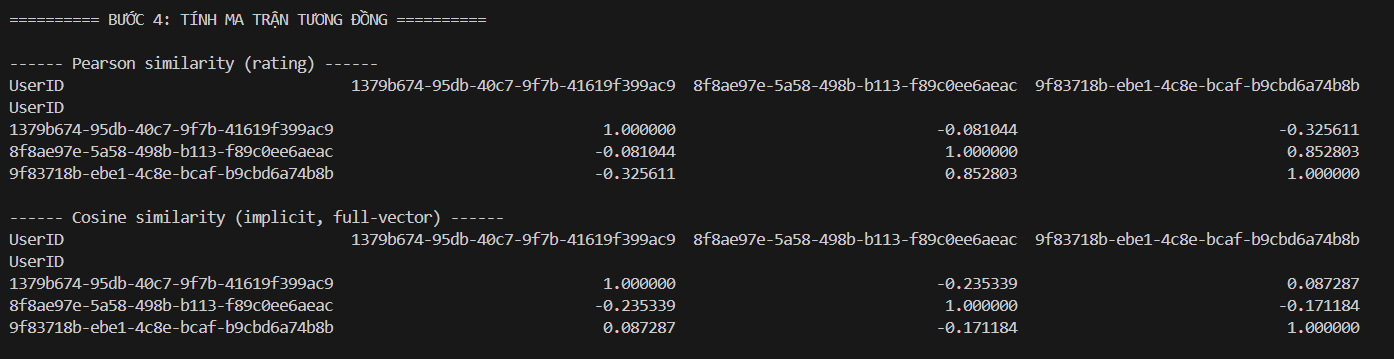
* + - * 1. Lấy dữ liệu từ trường Rating

****

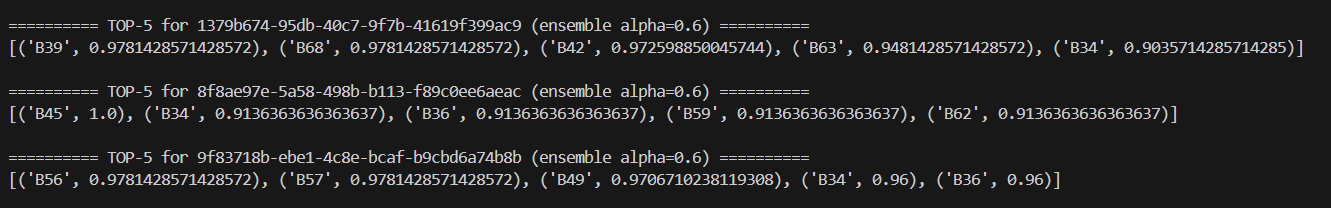
* + - * 1. Lấy dữ liệu từ trường Score

****

* + - * 1. Ma trận Rating và Implicit

****

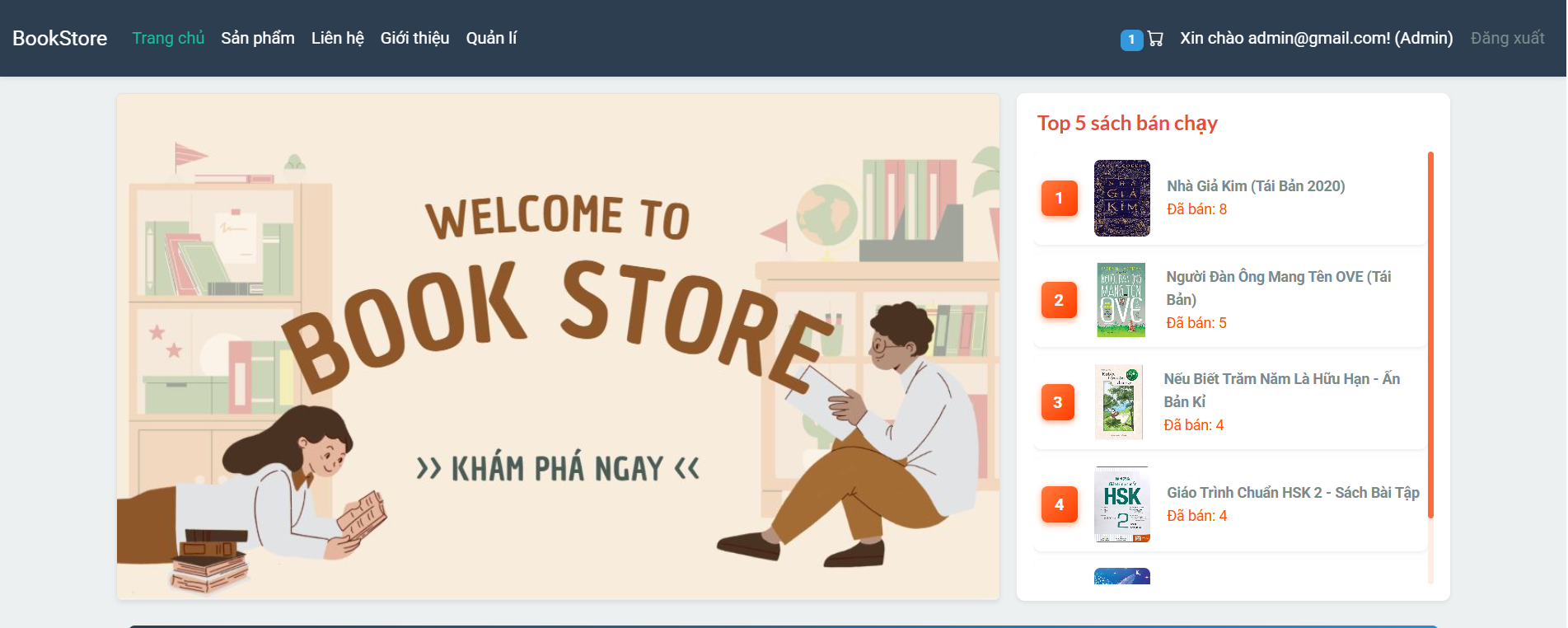
* + - * 1. Ma trận tương đồng

****

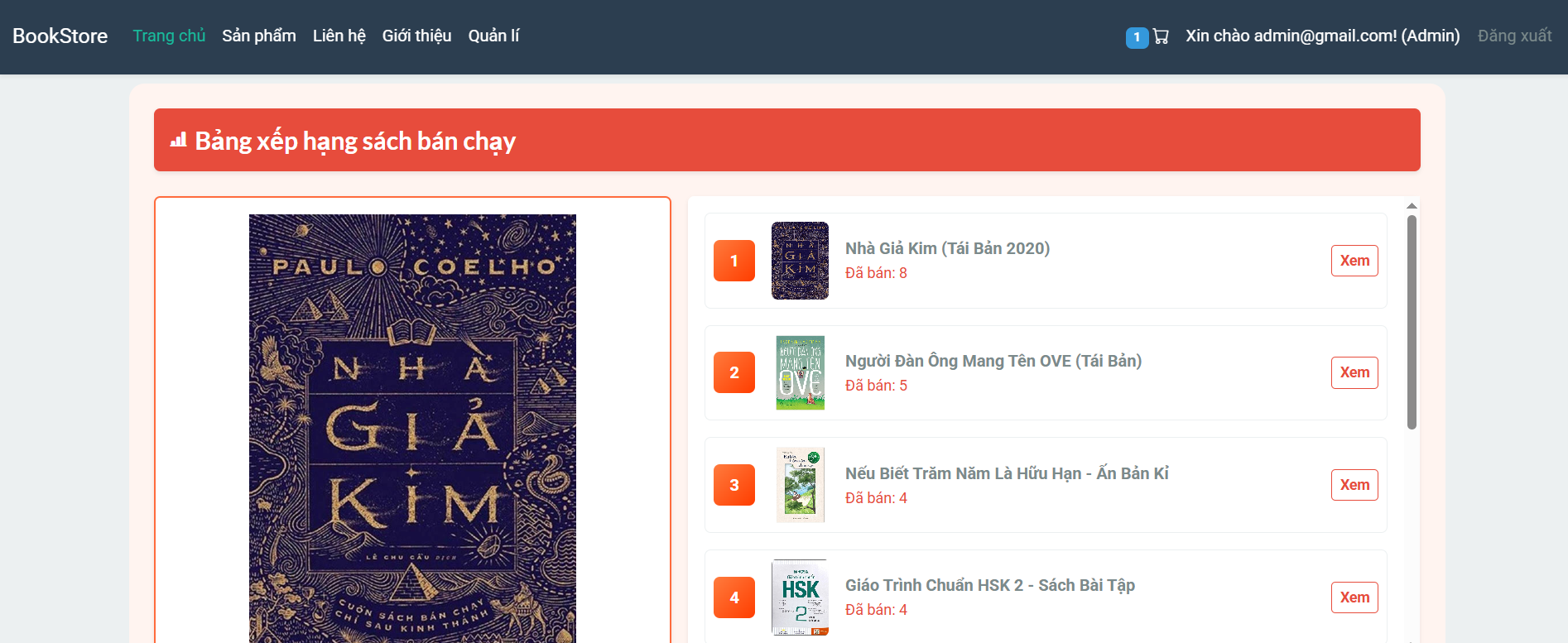
* + - * 1. Gợi ý sản phẩm



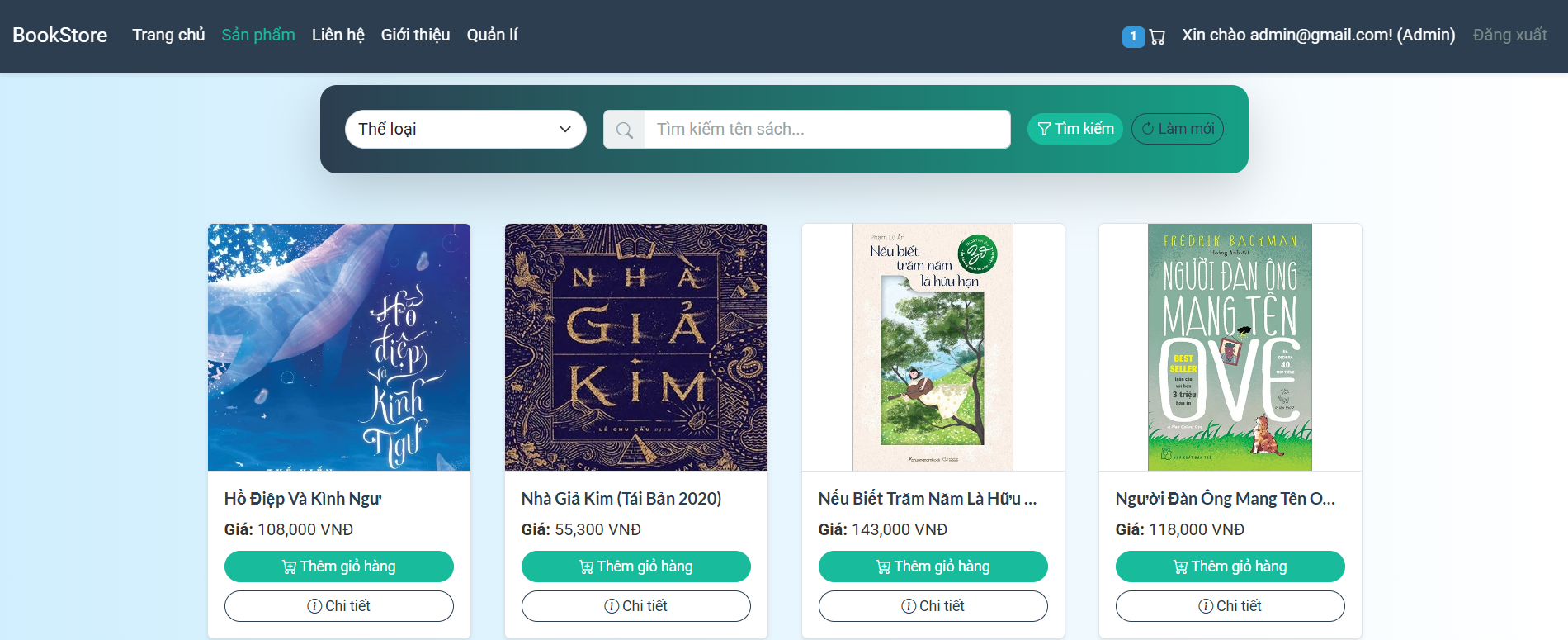
* + - * 1. Hiển thị sản phẩm gợi ý ra Website



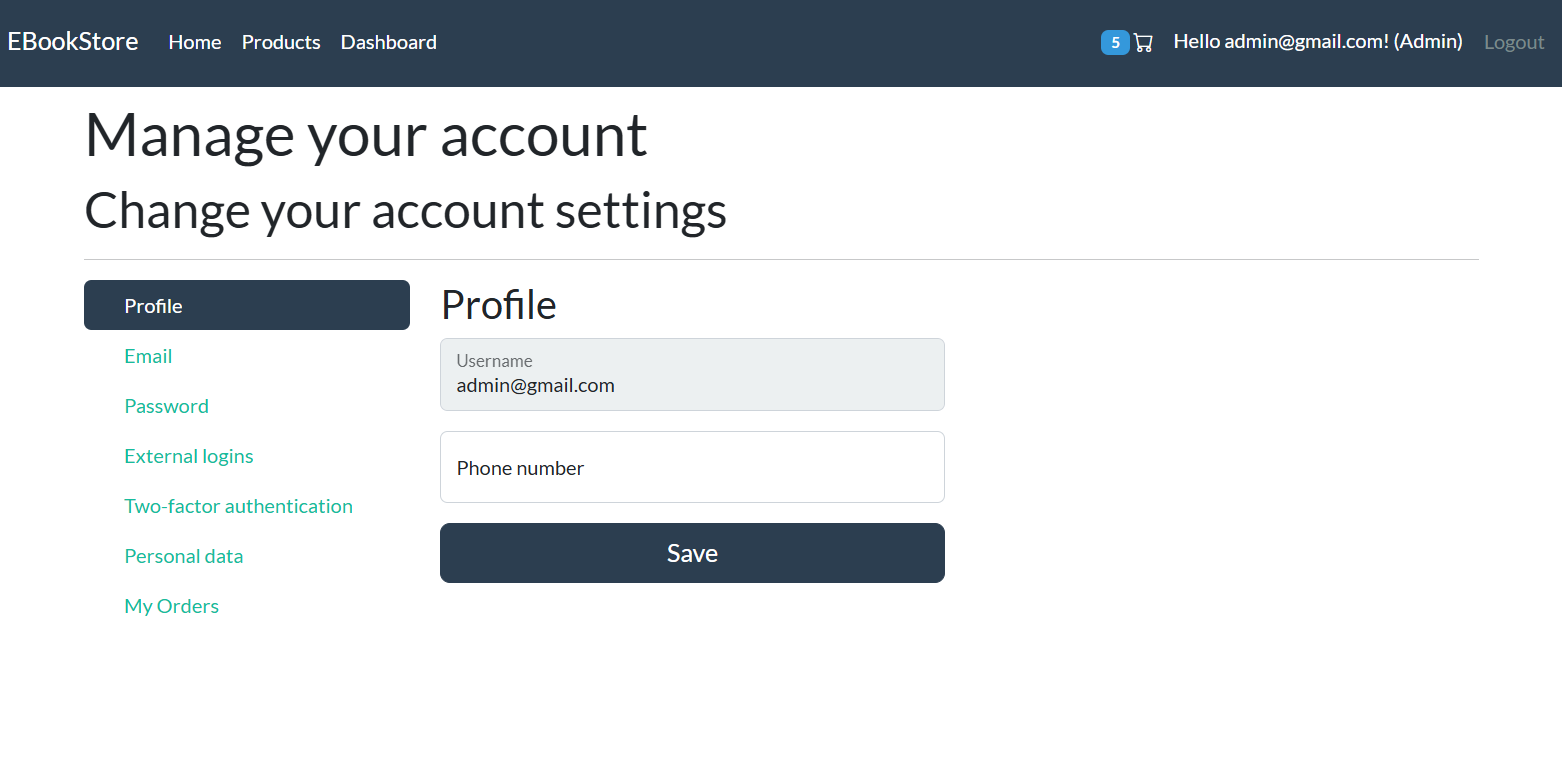
* + - * 1. Trang chủ website



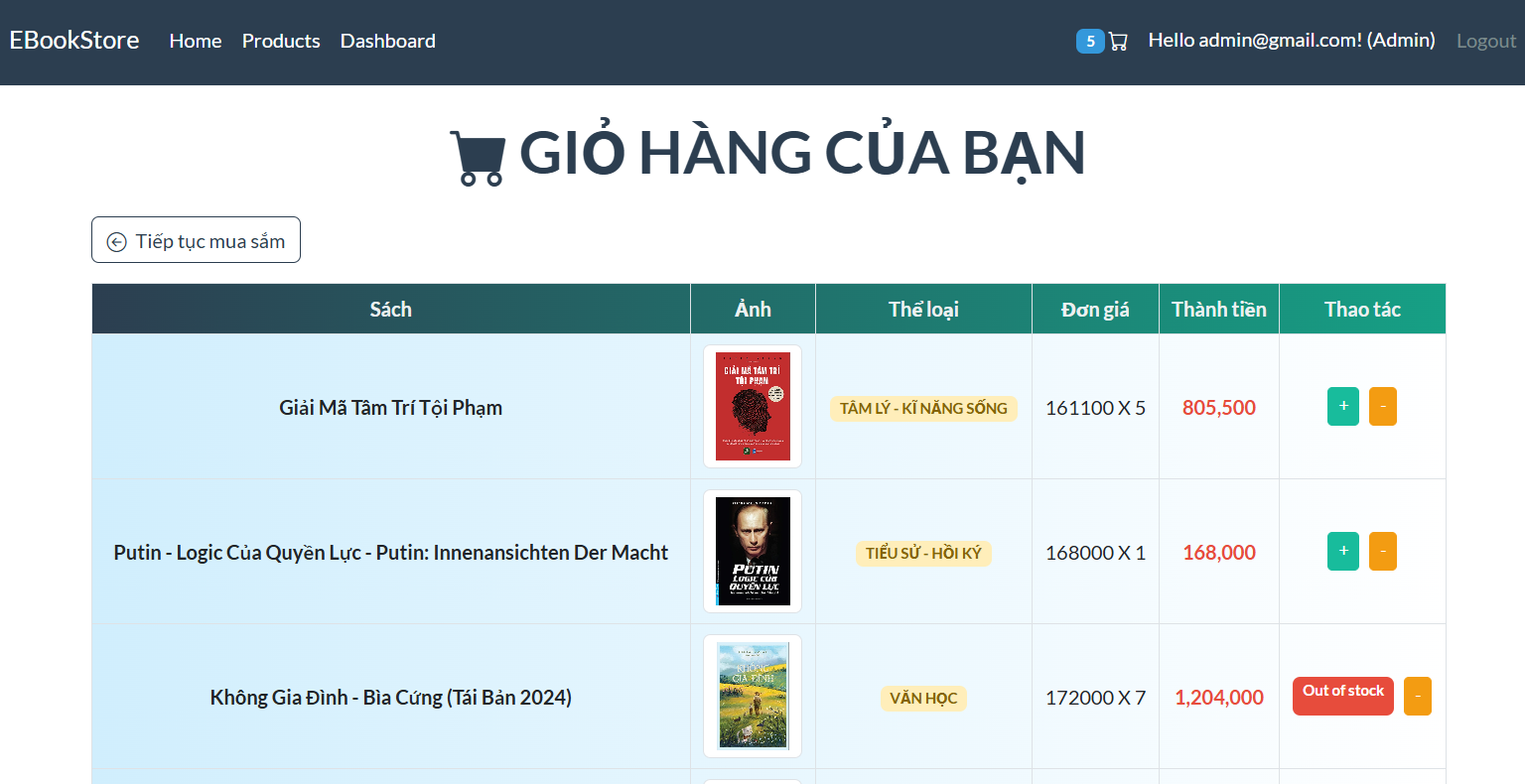
* + - * 1. Bảng xếp hạng sách bán chạy



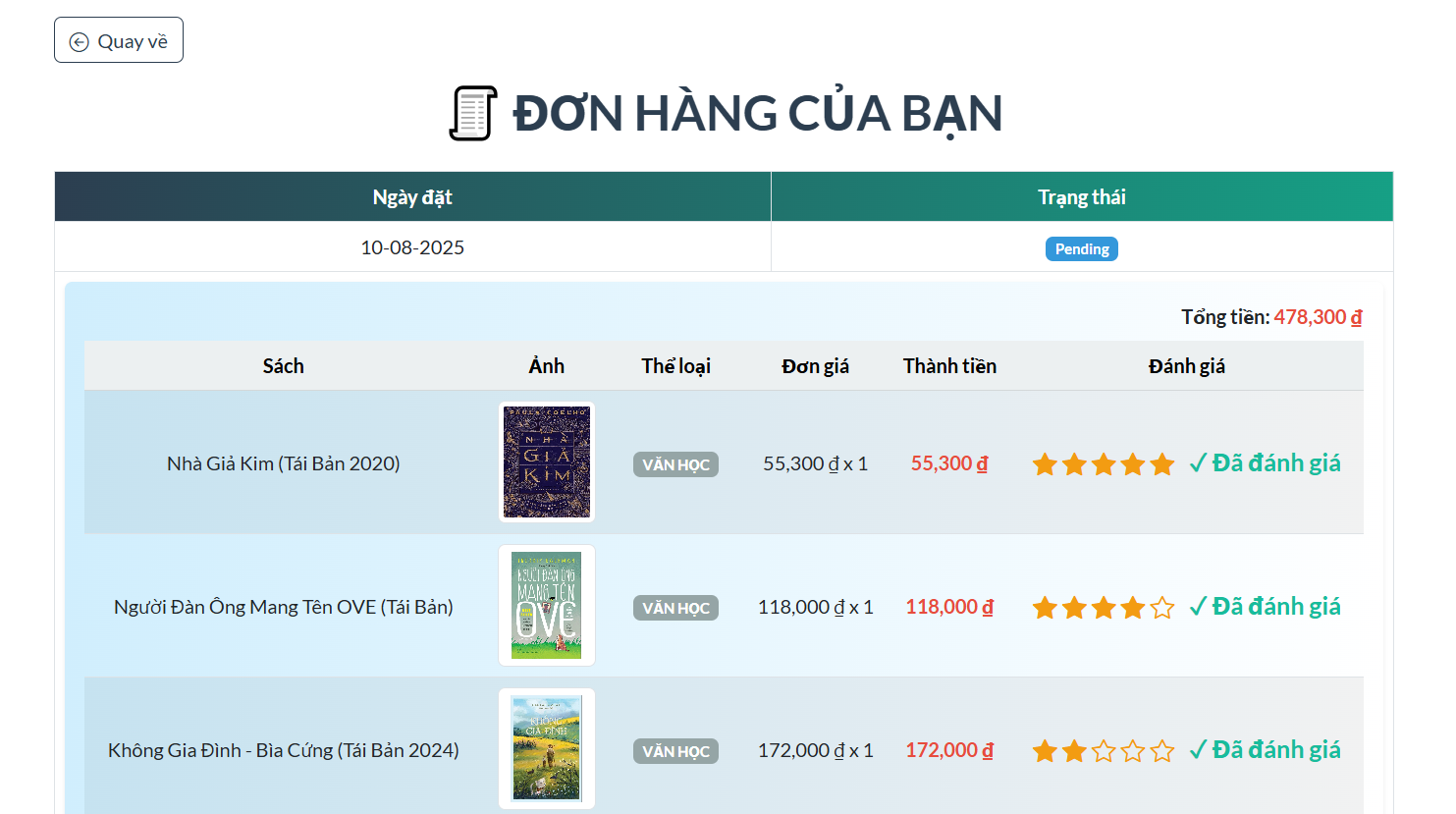
* + - * 1. Trang sản phẩm

****

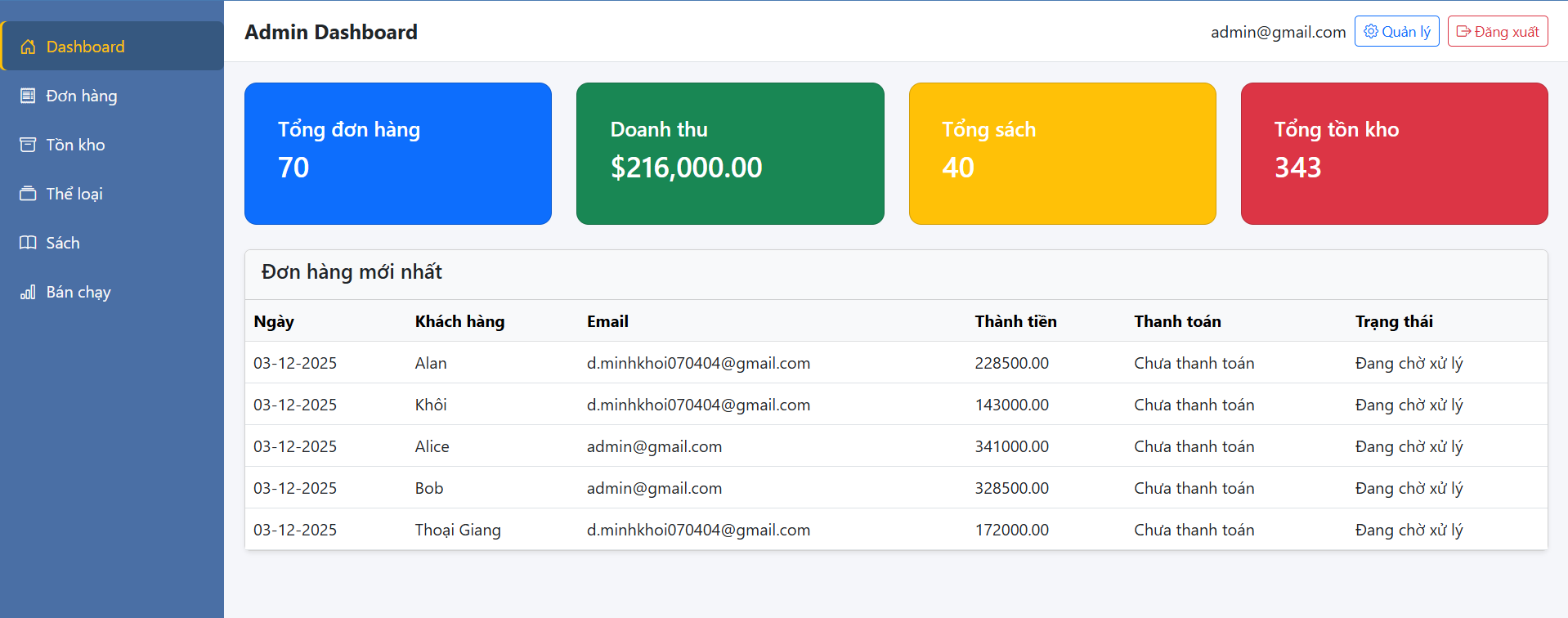
* + - * 1. Quản lí tài khoản

****

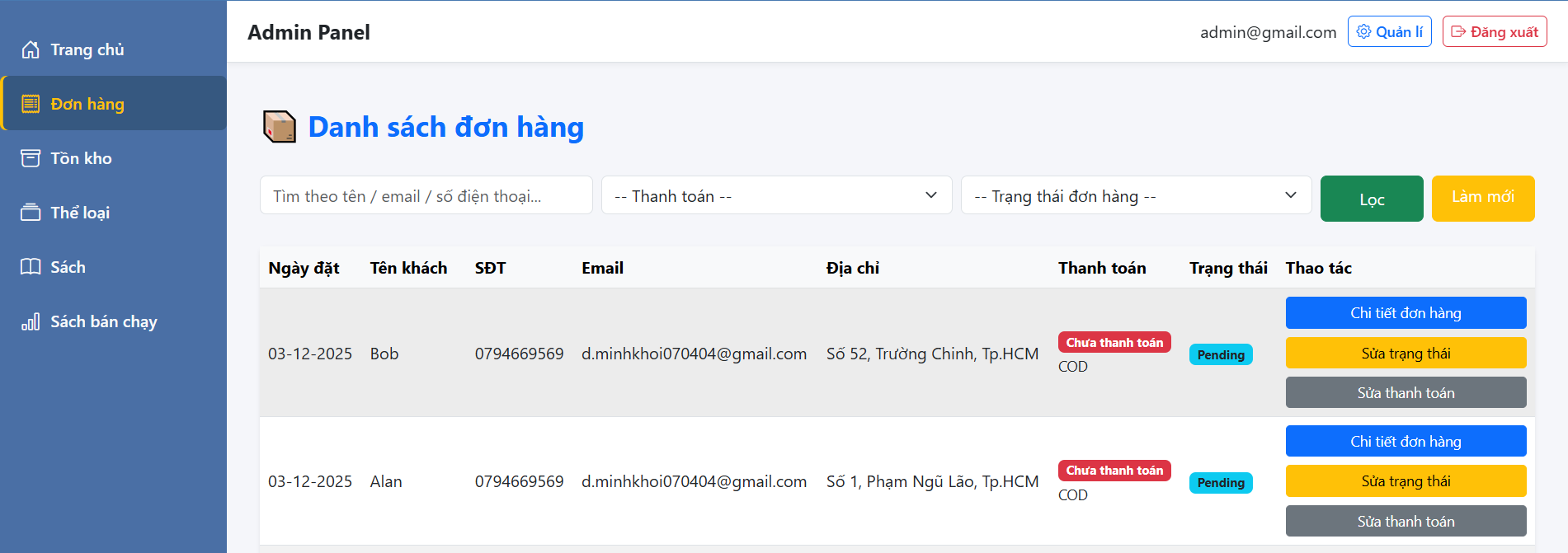
* + - * 1. Quản lí giỏ hàng

****

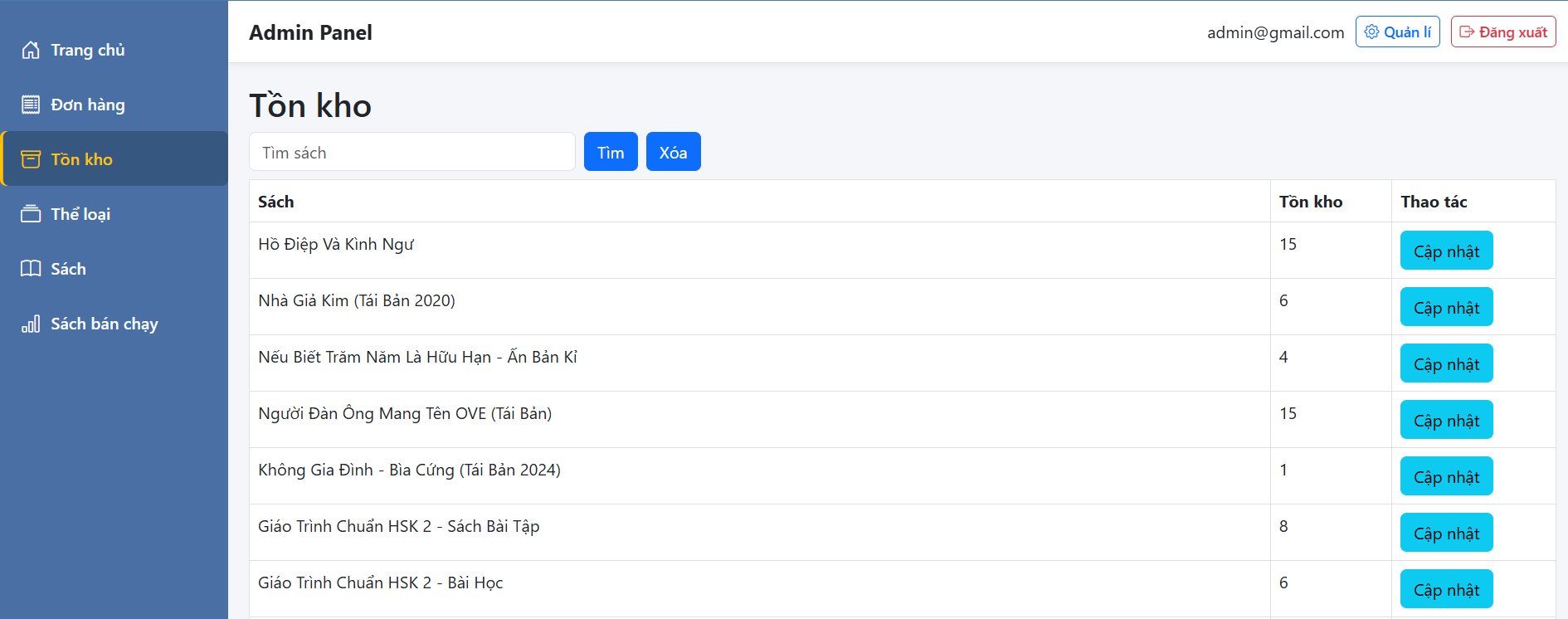
* + - * 1. Quản lí đơn hàng



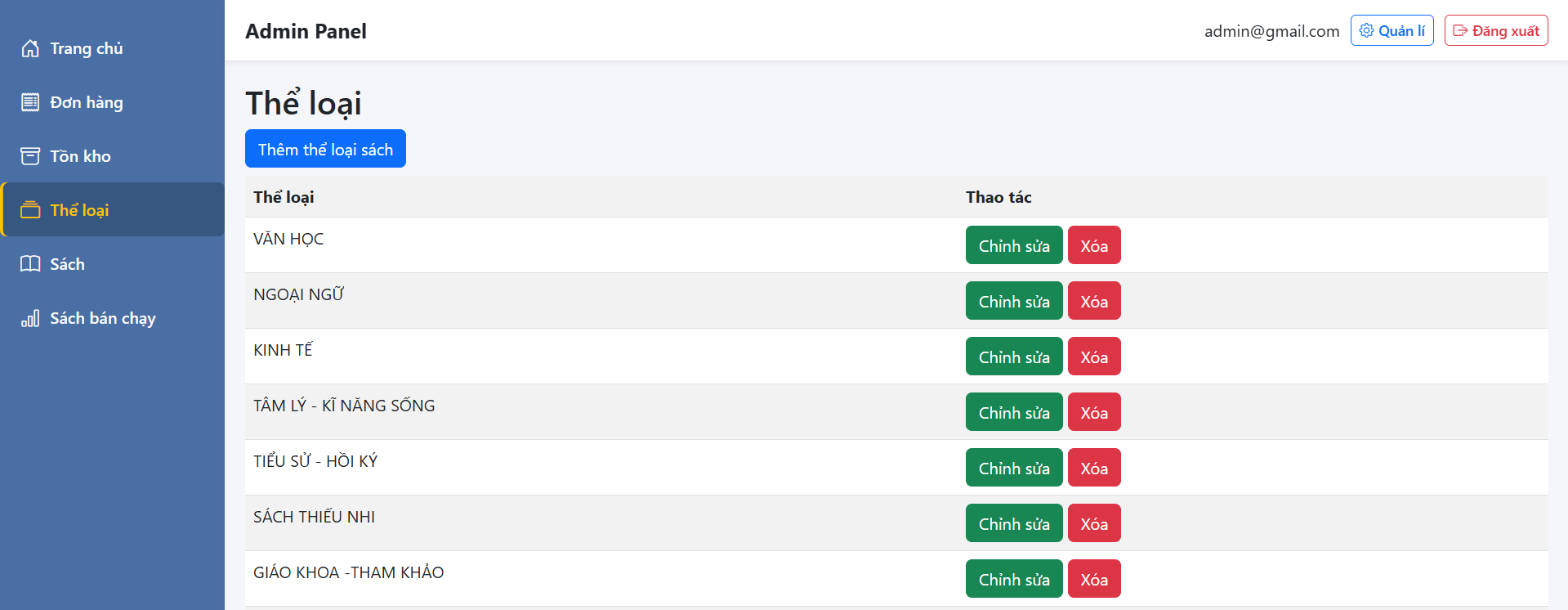
* + - * 1. Quản lí của admin



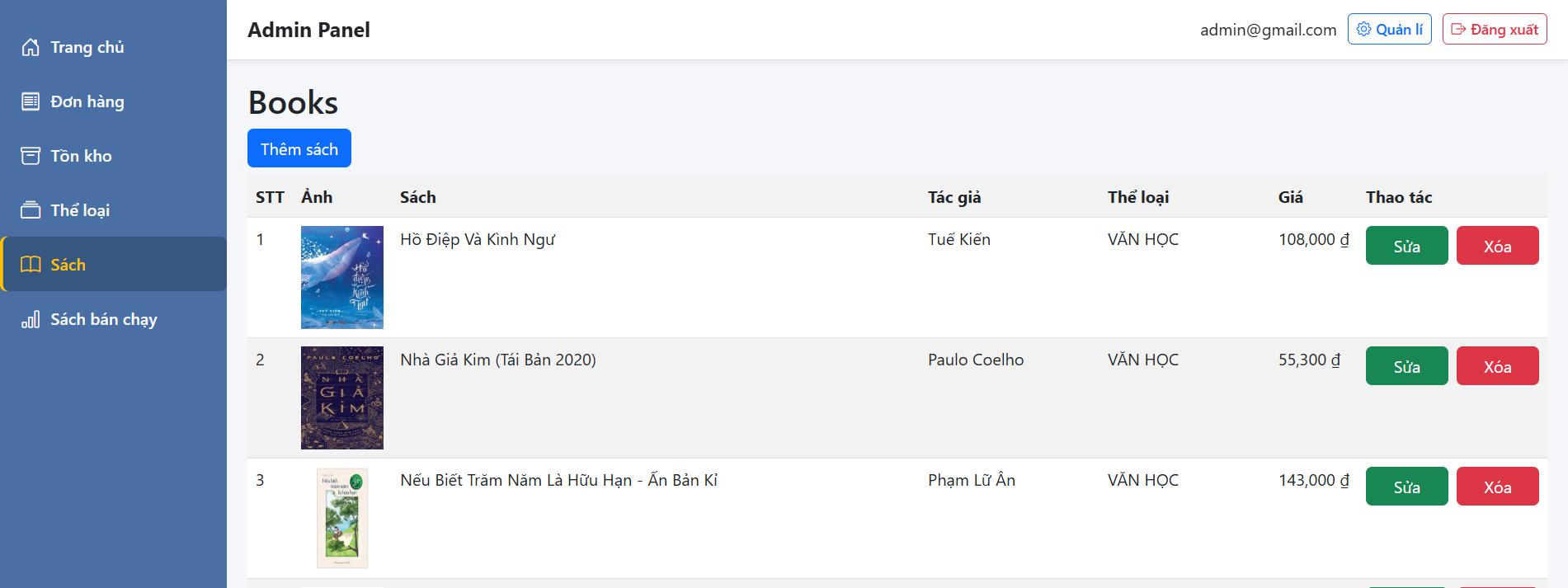
* + - * 1. Quản lí danh sách đơn hàng



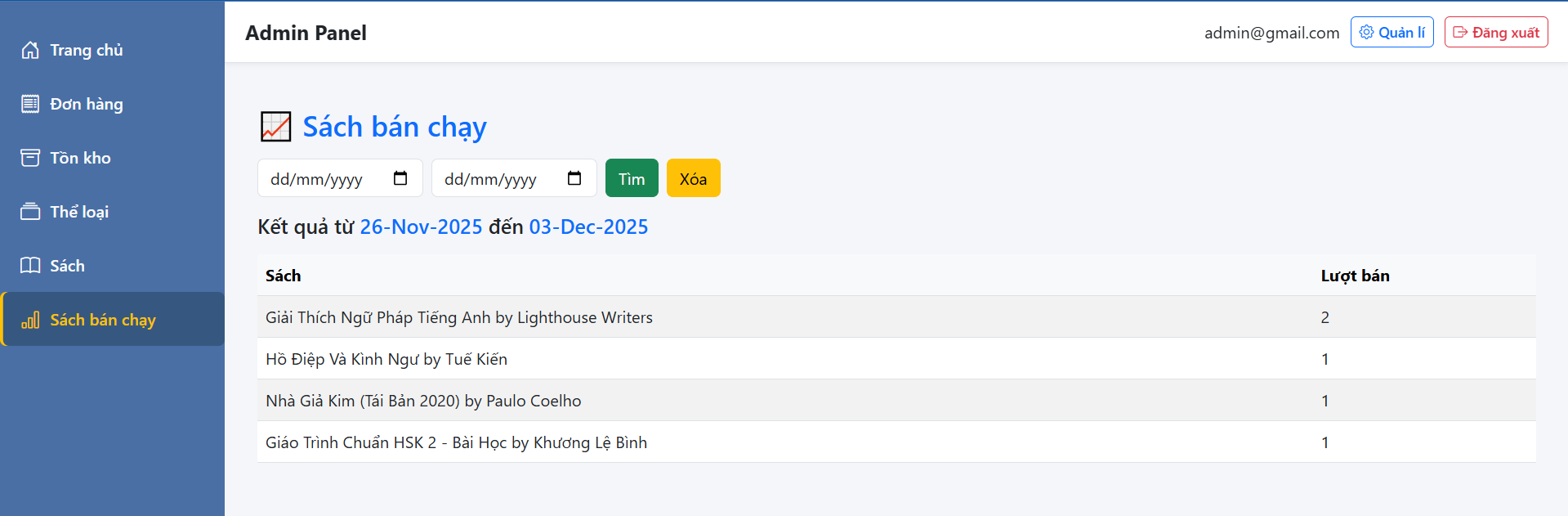
* + - * 1. Quản lí tồn kho



* + - * 1. Quản lí thể loại



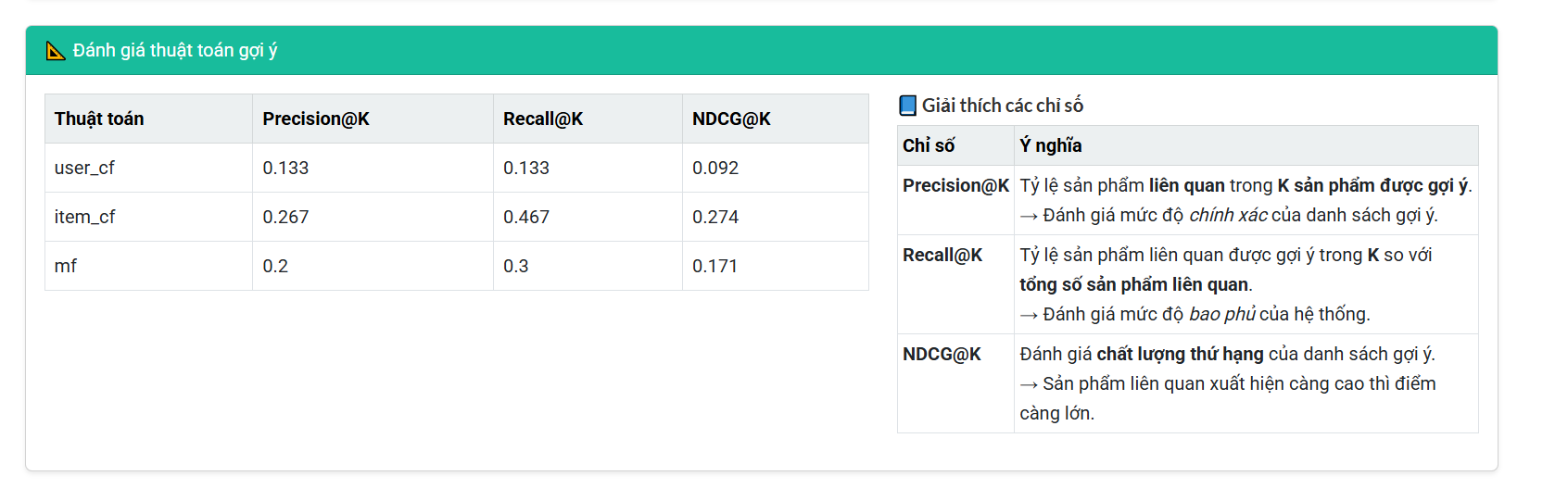
* + - * 1. Quản lí sách



* + - * 1. Quản lí sách bán chạy theo khoảng thời gian nhất định



* + - * 1. Quản lí hệ thống gợi ý



* + - * 1. Đánh giá thuật toán gợi ý

# 5. KẾT QUẢ, KẾT LUẬN/ HƯỚNG PHÁT TRIỂN

**5.1. Kết quả đạt được**

Sau quá trình xây dựng, hệ thống đã đạt được những kết quả chính:

***5.1.1 Website thương mại điện tử bán sách (ASP.NET MVC)***

* Hoàn thiện các chức năng **cơ bản** của một hệ thống e-commerce:
  + Quản lý tài khoản (đăng ký, đăng nhập, đăng nhập Google).
  + Duyệt, tìm kiếm, xem chi tiết sản phẩm.
  + Quản lý giỏ hàng, đơn hàng, lịch sử mua hàng.
  + Quản trị hệ thống: đơn hàng, tồn kho, thể loại, sách, thống kê sách bán chạy.
* Giao diện trực quan, hỗ trợ phân loại theo danh mục, hiển thị chi tiết rõ ràng.

***5.1.2 Hệ thống gợi ý sản phẩm (Recommendation System)***

* **Dữ liệu hành vi (UserInteractions)** được thu thập từ nhiều nguồn: xem chi tiết, thêm giỏ hàng, mua hàng, đánh giá.
* **3 mô hình gợi ý** đã được huấn luyện và tích hợp:
  + User-based Collaborative Filtering.
  + Item-based Collaborative Filtering.
  + Matrix Factorization (SGD).
* **API gợi ý (FastAPI)** hoạt động độc lập, cung cấp dữ liệu gợi ý cho ASP.NET MVC thông qua REST API.
* Trên website, người dùng được nhận các **sản phẩm cá nhân hóa**, cải thiện trải nghiệm và tăng khả năng mua hàng.

***5.1.3 Tích hợp & vận hành***

* Hệ thống ASP.NET gọi API Python để lấy kết quả gợi ý và render trực tiếp trên giao diện.
* Cơ sở dữ liệu được thiết kế hợp lý, đảm bảo quản lý tốt thông tin người dùng, sản phẩm, đơn hàng và dữ liệu huấn luyện.
* Thử nghiệm ban đầu cho thấy các gợi ý **sát với hành vi người dùng**, đặc biệt khi kết hợp explicit + implicit feedback.

***5.2. Kết luận***

Đề tài **“Xây dựng website bán sách tích hợp hệ thống gợi ý sản phẩm”** đã giải quyết thành công hai mục tiêu:

1. Xây dựng một nền tảng thương mại điện tử đầy đủ chức năng.
2. Tích hợp hệ thống gợi ý dựa trên **Collaborative Filtering và Matrix Factorization**, giúp website trở nên thông minh và thân thiện hơn với người dùng.

Hệ thống có khả năng **cá nhân hóa trải nghiệm mua sắm**, góp phần tăng sự hài lòng của khách hàng và hiệu quả kinh doanh.

***5.3. Hướng phát triển***

Trong tương lai, hệ thống có thể được mở rộng theo các hướng:

* **Tích hợp thêm thuật toán gợi ý nâng cao**: Deep Learning (Neural Collaborative Filtering, Autoencoders, Transformers).
* **Kết hợp thêm Content-based Filtering**: sử dụng metadata (thể loại, tác giả, từ khóa) để gợi ý chính xác hơn cho người dùng mới (cold-start problem).
* **Tối ưu hiệu năng hệ thống**: triển khai cache, tối ưu API, áp dụng mô hình huấn luyện online để cập nhật dữ liệu liên tục.
* **Phát triển trải nghiệm người dùng**: gợi ý theo ngữ cảnh (time-based, location-based), gợi ý theo nhóm người dùng.
* **Tích hợp thanh toán online qua VNPay, Momo,**…
* **Triển khai thực tế**: kết nối với cổng thanh toán online, triển khai hệ thống trên cloud (Azure/AWS/GCP) để phục vụ số lượng lớn người dùng.

Tài liệu tham khảo

* [1] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Procedia Computer Science,* vol. 49, pp. 136-146, 2015.
* [2] G. Gupta and R. Katarya, "Recommendation analysis on item-based and user-based collaborative filtering," in *2019 international conference on smart systems and inventive technology (ICSSIT)*, 2019: IEEE, pp. 1-4.
* [3] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR),* vol. 47, no. 1, pp. 1-45, 2014.

[1] [2] [3]