**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

--------------------------------------

**ĐỒ ÁN TỐT NGHIỆP**

NGÀNH HỆ THỐNG THÔNG TIN

**XÂY DỰNG HỆ THỐNG BÁN SÁCH TÍCH HỢP AI**

**CBHD: ThS. Nguyễn Thị Hương Lan**

**Sinh viên: Nguyễn Minh Tuân**

**Mã sinh viên: 2021606772**

**Hà Nội – 2025**

# **LỜI CẢM ƠN**

Lời đầu tiên, em muốn bày tỏ lòng biết ơn sâu sắc đến các thầy cô Trường Đại học Công nghiệp Hà Nội, đặc biệt là các thầy cô trong khoa Công nghệ thông tin đã tận tình dạy dỗ và hướng dẫn em trong suốt quãng thời gian học tập ở trường. Sự hiểu biết, tận tâm và kiến thức chuyên môn của các thầy, cô chính là nguồn động viên to lớn, là hành trang cho em trong quá trình chinh phục con đường học tập và sự nghiệp.

Đặc biệt, em xin gửi lời cảm ơn chân thành cô Nguyễn Thị Hương Lan, người đã giúp em vượt qua những thách thức và phát triển bản thân, cảm ơn cô đã tận tình dõi theo, giải đáp thắc mắc trong suốt quãng thời gian vừa qua. Những lời khuyên, định hướng và sự hỗ trợ của cô đã giúp em hoàn thành đề tài của mình một cách tốt nhất.

Học phần Đồ án tốt nghiệp là một học phần vô cùng quan trọng, đó là minh chứng cho những kiến thức đã có sau bốn năm học tập tại trường.Quá trình thực hiện đồ án không chỉ là thách thức còn là cơ hội để em có thể áp dụng những kiến thức đã học vào thực tế. Em đã cố gắng hết mình để có thể hoàn thiện đồ án, song em vẫn luôn trân trọng và mong nhận được những sự đóng góp ý kiến quý báu của các thầy cô để bài làm của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Minh Tuân

# **MỞ ĐÀU**

## **1. Lý do chọn đề tài:**

Trong thời đại công nghệ số phát triển mạnh mẽ, mua sắm trực tuyến đã trở thành xu hướng phổ biến và không thể thiếu trong đời sống hiện đại. Thay vì đến tận cửa hàng, người dùng ngày nay có xu hướng tìm kiếm, đánh giá và mua sắm các sản phẩm ngay trên nền tảng số, đặc biệt là với các mặt hàng như sách – một lĩnh vực đang từng bước chuyển dịch mạnh mẽ từ truyền thống sang thương mại điện tử.

Tuy nhiên, nhiều cửa hàng sách hiện nay vẫn còn gặp khó khăn trong việc quản lý bán hàng theo phương pháp thủ công như ghi chép đơn hàng, quản lý kho không đồng bộ, xử lý thanh toán chưa hiệu quả, hoặc không thể cá nhân hóa trải nghiệm mua sắm cho từng khách hàng. Điều này gây ra nhiều bất cập trong quá trình vận hành, đặc biệt trong các đợt khuyến mãi hoặc khi số lượng đơn hàng tăng đột biến.

Xuất phát từ thực tiễn đó, đề tài “Xây dựng hệ thống bán sách tích hợp AI sử dụng ngôn ngữ C# và nền tảng .NET” được lựa chọn với mục tiêu xây dựng một hệ thống thương mại điện tử hiện đại, thân thiện với người dùng, đồng thời cung cấp cho quản trị viên và nhân viên công cụ hỗ trợ quản lý hiệu quả. Hệ thống sẽ giúp người dùng dễ dàng tìm kiếm, xem chi tiết, đánh giá và đặt mua sách trực tuyến. Bên cạnh đó, việc tích hợp công nghệ AI còn mở ra hướng cá nhân hóa trải nghiệm mua sắm như đề xuất sách theo sở thích người dùng.

Việc lựa chọn ngôn ngữ C# cùng nền tảng ASP.NET Core MVC là sự lựa chọn hợp lý, đảm bảo tính ổn định, bảo mật và dễ dàng mở rộng hệ thống trong tương lai. Mô hình phân tầng logic (Core – Data – Web – API) giúp việc tổ chức mã nguồn rõ ràng, dễ bảo trì và phát triển thêm các tính năng thông minh như quản lý giỏ hàng, đơn hàng, người dùng, phân quyền và khuyến mãi một cách chuyên nghiệp.

## **2. Mục đích nghiên cứu**

* Xây dựng một hệ thống website thương mại điện tử bán sách hiện đại, hiệu quả và thân thiện, giúp khách hàng có thể dễ dàng tìm kiếm, tra cứu thông tin, so sánh và đặt mua các đầu sách phù hợp với nhu cầu cá nhân.
* Tích hợp các công nghệ AI nhằm cá nhân hóa trải nghiệm mua sắm, gợi ý sách theo hành vi và sở thích của người dùng, từ đó tăng sự hài lòng và khả năng quay lại của khách hàng.
* Cung cấp cho quản trị viên và nhân viên hệ thống các công cụ quản lý mạnh mẽ như quản lý danh mục sách, đơn hàng, tồn kho, doanh thu, khuyến mãi và người dùng; đồng thời hỗ trợ phân quyền truy cập rõ ràng giữa các vai trò trong hệ thống
* Tạo nền tảng thuận lợi cho việc mở rộng hệ thống trong tương lai, có thể tích hợp thêm các tính năng như quản lý vận chuyển, thanh toán online, đánh giá sản phẩm, hoặc đồng bộ dữ liệu với các nền tảng bên thứ ba.

## **3. Đối tượng nghiên cứu và phạm vi nghiên cứu**

* Tìm hiểu các tác vụ của người dùng trong hệ thống SmartBookStore, bao gồm khách hàng, quản lý cửa hàng và người quản trị hệ thống.
* Nghiên cứu và sử dụng các công nghệ web như HTML, CSS, JavaScript kết hợp với thư viện Bootstrap để xây dựng giao diện người dùng thân thiện, dễ sử dụng và có khả năng tương thích đa thiết bị.
* Sử dụng hệ quản trị cơ sở dữ liệu SQL Server để thiết kế và quản lý cơ sở dữ liệu, lưu trữ thông tin sách, người dùng, đơn hàng, giỏ hàng, khuyến mãi, và dữ liệu gợi ý từ AI.
* Tìm hiểu và áp dụng mô hình kiến trúc MVC (Model - View - Controller) trong ASP.NET Core MVC để phân tách các lớp chức năng, nâng cao khả năng bảo trì và mở rộng hệ thống.
* Xây dựng và phát triển website thương mại điện tử bán sách sử dụng framework ASP.NET Core MVC, tích hợp AI để gợi ý sản phẩm phù hợp cho người dùng, kết nối trực tiếp với cơ sở dữ liệu, đảm bảo luồng xử lý thống nhất và hiệu quả.

## **4. Phương pháp nghiên cứu**

* Tự học thông qua các tài liệu, khóa học và video hướng dẫn liên quan đến lập trình web, ngôn ngữ C#, ASP.NET Core MVC, thiết kế cơ sở dữ liệu và tích hợp AI.
* Tham khảo các website thương mại điện tử bán sách hiện có như Tiki, Fahasa, NewShop... để phân tích chức năng, giao diện và trải nghiệm người dùng.
* Tiếp thu kiến thức từ bài giảng của giảng viên hướng dẫn, vận dụng lý thuyết đã học vào thực tế dự án.
* Áp dụng các kiến thức đã học để xây dựng website thương mại điện tử bán sách – SmartBookStore, sử dụng framework ASP.NET Core MVC, kết hợp với AI đề xuất sản phẩm và cơ sở dữ liệu SQL Server để triển khai hệ thống quản lý và bán sách trực tuyến.

## **5. Kết quả dự kiến**

* Hiểu và phân tích được các nghiệp vụ chính của người dùng trong hệ thống website bán sách, bao gồm khách hàng, nhân viên và người quản trị.
* Nắm vững mô hình MVC (Model - View - Controller) và các thành phần trong ASP.NET Core MVC, như Routing, Authentication, Authorization, và cách tích hợp với cơ sở dữ liệu.
* Hoàn thiện một hệ thống website bán sách có thể đăng ký, đăng nhập, mua hàng, thanh toán, quản lý và thống kê, phục vụ bài toán thực tế trong thương mại điện tử.

## **6. Ý nghĩa khoa học – thực tiễn**

* Ứng dụng các công nghệ hiện đại như ASP.NET Core MVC, AI đề xuất sản phẩm, giúp sinh viên tiếp cận công nghệ mới và nâng cao trình độ chuyên môn trong lĩnh vực phát triển phần mềm.
* Tích hợp đa dạng công nghệ như HTML, CSS, JavaScript, Bootstrap, SQL Server và mô hình gợi ý AI, tạo nên một hệ thống hoàn chỉnh, có khả năng mở rộng và dễ bảo trì.
* Góp phần thúc đẩy chuyển đổi số trong hoạt động bán lẻ truyền thống, khắc phục những hạn chế về địa lý, thời gian và chi phí vận hành bằng mô hình thương mại điện tử.
* Cung cấp cơ sở để nghiên cứu và phát triển các hệ thống AI trong thương mại điện tử, từ đó mở rộng ứng dụng vào các lĩnh vực tương tự như bán hàng thời trang, thiết bị công nghệ, giáo dục,…

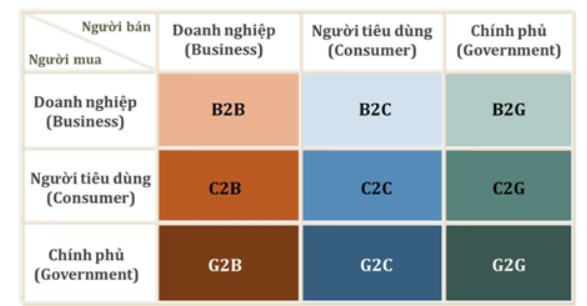
# **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

* 1. **Tổng quan hệ thống thương mại điện tử**
     1. **Khái niệm**

Thương mại điện tử (E-Commerce) là hình thức mua bán hàng hóa, dịch vụ và trao đổi thông tin thông qua các phương tiện điện tử như Internet. Hệ thống thương mại điện tử cho phép người tiêu dùng và doanh nghiệp thực hiện các giao dịch trực tuyến một cách nhanh chóng và thuận tiện mà không cần gặp mặt trực tiếp.

Theo Luật Giao dịch điện tử Việt Nam năm 2005, thương mại điện tử là hoạt động thương mại được thực hiện bằng phương tiện điện tử, trong đó các bên tham gia sử dụng thiết bị điện tử để ký kết hoặc thực hiện hợp đồng thương mại.

* + 1. **Các mô hình phổ biến**

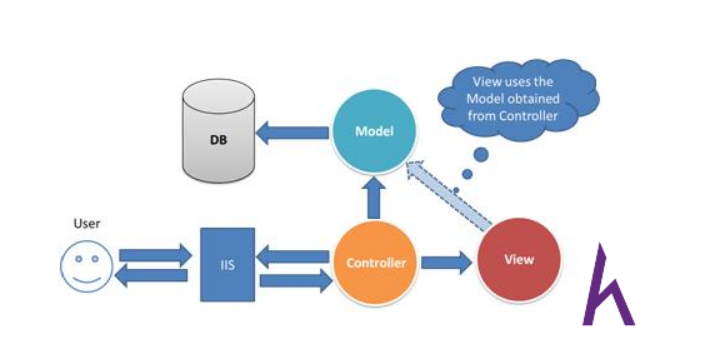
****

Hình 1.1. Các mô hình của hệ thống thương mại điện tử

* **B2C (Business to Consumer):** Giao dịch giữa doanh nghiệp và người tiêu dùng, ví dụ như các trang bán hàng online như Tiki, Lazada, Shopee.
* **B2B (Business to Business):** Giao dịch giữa hai doanh nghiệp, ví dụ như sàn thương mại điện tử Alibaba.
* **C2C (Consumer to Consumer):** Giao dịch giữa cá nhân với cá nhân, ví dụ như các giao dịch trên chợ điện tử như Chợ Tốt, eBay.
  + 1. **Vai trò và lợi ích của hệ thống thương mại điện tử**

Hệ thống thương mại điện tử mang lại nhiều lợi ích thiết thực cho cả người bán và người mua,bao gồm:

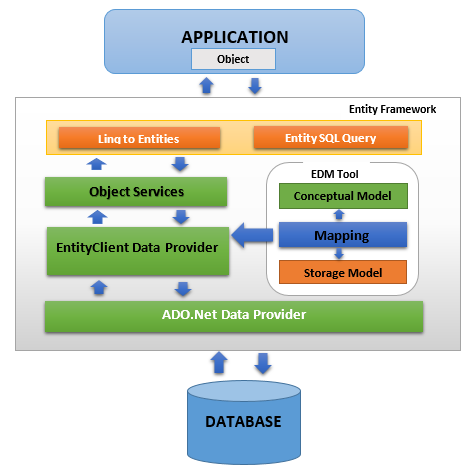
* Mở rộng phạm vi tiếp cận khách hàng, không bị giới hạn về mặt địa lý.
* Giảm chi phí vận hành nhờ tự động hóa quy trình bán hàng và quản lý.
* Cải thiện trải nghiệm người dùng với giao diện thân thiện, hỗ trợ tìm kiếm và thanh toán nhanh chóng.
* Tạo điều kiện cho doanh nghiệp dễ dàng phân tích hành vi khách hàng để đưa ra các chiến lược kinh doanh phù hợp.
  1. **Giới thiệu về ASP.NET Core**
     1. **Khái niệm**
* [ASP.NET](http://asp.net/) Core là một nền tảng mã nguồn mở, cross-platform framework cho việc xây dựng và kết nối các ứng dụng web hiện đại chẳng hạn như ứng dụng web, Internet of Thing, Mobile Backend - Web API
* [ASP.NET](http://asp.net/) Core cung cấp một kiến trúc để tối ưu hóa việc xây dựng các ứng dụng đám mây (microsoft azure) hoặc các ứng dụng độc lập. Bạn có thế phát triển ứng dụng [ASP.NET](http://asp.net/) Core đa nên tảng(Window, Mac, Linux), với mã nguồn mở tại [Github](https://github.com/aspnet/home).
  + 1. **Cơ chế hoạt động của ASP.NET Core**

****

*Hình 1.2. Cơ chế hoạt động của ASP.NET Core MVC*

* User gửi một yêu cầu tới server bằng cách truyền vào 1 URL trong browser.
* Yêu cầu đó được gửi tới Controller đầu tiên, Controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì Controller sẽ chuyển qua tầng Model.
* Tại tầng Model, dữ liệu được truy xuất từ database và sau đó truyền qua View thông qua Controller.
* Controller sẽ giúp dữ liệu được chuyển từ Model qua View.
* View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho. User thông qua tầng View.
  + 1. **Ưu điểm của ASP.NET Core**
* Hiệu năng cao: ASP.NET Core có hiệu năng vượt trội so với các framework web khác nhờ kiến trúc nhẹ và tối ưu hóa tốt.
* Đa nền tảng (Cross-platform): Có thể chạy trên Windows, macOS, và Linux.
* Mã nguồn mở: Dễ dàng truy cập và tùy chỉnh mã nguồn trên GitHub ([github.com/dotnet/aspnetcore](https://github.com/dotnet/aspnetcore)).
* Kiến trúc linh hoạt (Modular architecture): Cho phép thêm/bớt các thành phần theo nhu cầu của dự án.
* Hỗ trợ Dependency Injection tích hợp sẵn: Giúp giảm sự phụ thuộc và tăng khả năng kiểm thử.
* Hỗ trợ Razor Pages, MVC, API: Đa dạng phương thức phát triển website và dịch vụ web.
* Cập nhật nhanh và liên tục từ Microsoft: Bảo mật và tính năng luôn được cải tiến.
  1. **Giới thiệu về Entity Framework Core và mô hình Code First**
     1. **Khái niệm chung Entity Framework Core( EF Core)**

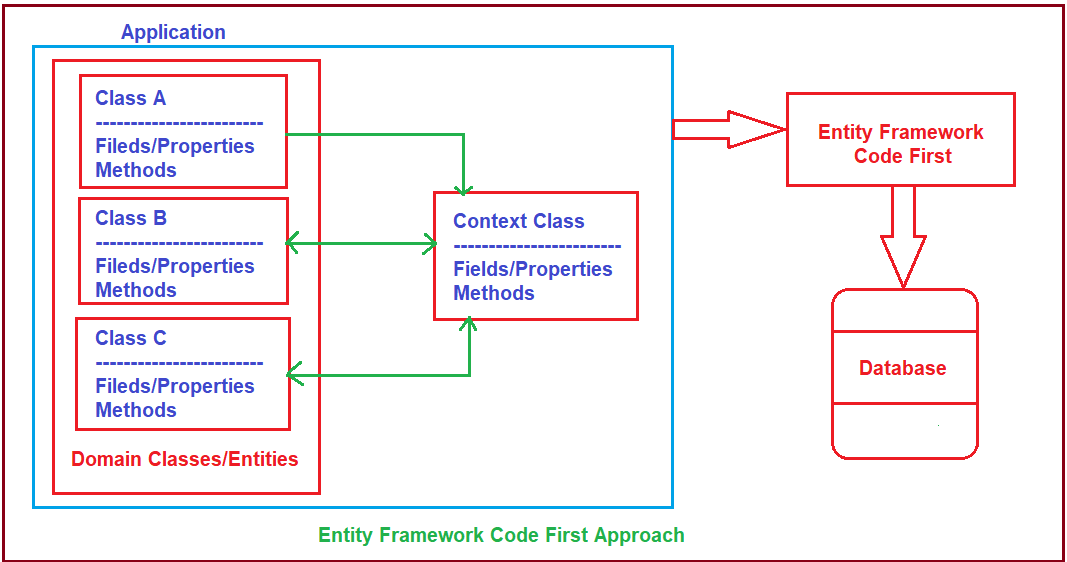
Entity Framework Core (EF Core) là một framework ORM (Object-Relational Mapper) mã nguồn mở, nhẹ, đa nền tảng, do Microsoft phát triển. Nó cho phép các nhà phát triển .NET tương tác với cơ sở dữ liệu bằng các đối tượng C# thay vì viết các truy vấn SQL trực tiếp, giúp đơn giản hóa và tăng tốc độ phát triển ứng dụng.



*Hình 1.3. Tổng quan quy trình hoạt động của EF Core*

* Ứng dụng gọi đến **DbContext** với các DbSet<T>.
* EF Core chuyển xử lý các thao tác CRUD bằng LINQ thành câu lệnh SQL.
* Truy vấn được gửi đến **cơ sở dữ liệu** (ví dụ: SQL Server).
* Kết quả truy vấn được EF Core ánh xạ lại thành các đối tượng C#.
* Khi gọi SaveChanges(), EF Core sẽ tạo và thực thi các lệnh INSERT, UPDATE, DELETE tương ứng.
  + 1. **Mô hình Code First là gì**

Mô hình Code First trong Entity Framework là một phương pháp phát triển, trong đó bạn bắt đầu với các lớp model (đối tượng) trong mã nguồn và sau đó Entity Framework tự động tạo ra cơ sở dữ liệu tương ứng dựa trên các lớp này. Nói cách khác, bạn định nghĩa cấu trúc dữ liệu thông qua các lớp C# (hoặc các ngôn ngữ khác trong .NET), và EF Core sẽ tạo ra cơ sở dữ liệu, bảng, quan hệ, và các ràng buộc dựa trên các lớp đó.



Hình 1.4. Mô hình tổng quan quá trình tạo và tương tác với database từ EF Core theo phương pháp Code First

1. Định nghĩa Domain model (các class C# như Book, User, Order) và DbContext.
2. Cấu hình model (tuỳ chọn) bằng Data Annotations hoặc Fluent API để đặt khóa, ràng buộc, quan hệ.
3. Tạo các file Migration khi dùng lệnh dotnet ef migrations add.
4. Áp dụng migration với dotnet ef database update → Hệ thống tạo hoặc cập nhật cấu trúc bảng trong CSDL.
5. Thao tác dữ liệu bằng LINQ và gọi SaveChanges() để lưu dữ liệu.
   * 1. **Lợi ích khi sử dụng EF Core với mô hình Code First trong phát triển ứng dụng ASP.NET Core**

* **Tăng tốc phát triển: EF Core với Code First giúp giảm thiểu công việc thủ công liên quan đến việc tạo và quản lý cơ sở dữ liệu. Developer có thể tập trung vào việc xây dựng logic nghiệp vụ và EF Core sẽ tự động tạo, cập nhật database schema và xử lý các thao tác CRUD (Create, Read, Update, Delete).**
* **Quản lý thay đổi dễ dàng:** Khi model dữ liệu thay đổi, EF Core Code First cho phép dễ dàng cập nhật database schema tương ứng. Điều này đặc biệt hữu ích trong các dự án có sự thay đổi thường xuyên về yêu cầu, giúp developer nhanh chóng thích ứng với các thay đổi mà không làm mất dữ liệu cũ.
* **Tính linh hoạt:** Code First cho phép developer tự do thay đổi cấu trúc model dữ liệu mà không bị ràng buộc bởi cơ sở dữ liệu hiện có. Điều này giúp tăng tính linh hoạt cho ứng dụng và dễ dàng thích ứng với các yêu cầu mới.
* **Mã nguồn sạch và dễ bảo trì:** Việc sử dụng LINQ (Language Integrated Query) trong EF Core giúp viết các truy vấn dữ liệu phức tạp một cách trực quan và dễ hiểu hơn. Điều này giúp cho mã nguồn trở nên sạch sẽ, dễ đọc và dễ bảo trì hơn.
* **Phù hợp với các dự án mới và thay đổi liên tục:** Code First rất phù hợp với các dự án mới hoặc các dự án có cấu trúc database thường xuyên thay đổi theo yêu cầu nghiệp vụ. Việc đồng bộ hóa cấu trúc database với mã nguồn giúp việc quản lý thay đổi trở nên dễ dàng hơn.
* **Tối ưu hóa hiệu suất:** EF Core cung cấp nhiều tính năng tối ưu hóa hiệu suất, như lazy loading, eager loading và caching, giúp cải thiện tốc độ truy vấn và hiệu năng của ứng dụng.
  1. **Giới thiệu về SQL Server**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phát triển bởi Microsoft. Nó cung cấp một nền tảng mạnh mẽ và đáng tin cậy để lưu trữ, truy vấn và quản lý dữ liệu trong môi trường doanh nghiệp.

Đặc điểm nổi bật của SQL Server:

* Tích hợp chặt chẽ với .NET: Cho phép các ứng dụng .NET truy xuất và xử lý dữ liệu một cách hiệu quả thông qua Entity Framework Core hoặc ADO.NET.
* Hỗ trợ ngôn ngữ T-SQL: Một phần mở rộng của SQL tiêu chuẩn, giúp thao tác dữ liệu linh hoạt, hiệu quả.
* Khả năng mở rộng cao: Hỗ trợ xử lý lượng lớn dữ liệu, nhiều người dùng đồng thời.
* Tính bảo mật cao: Có các cơ chế bảo mật như xác thực Windows, mã hóa dữ liệu, kiểm soát truy cập.
* Công cụ quản trị mạnh mẽ: SQL Server Management Studio (SSMS) giúp quản lý cơ sở dữ liệu dễ dàng thông qua giao diện trực quan.

Vai trò và lợi ích của SQL Server trong ứng dụng ASP.NET Core

* Lưu trữ dữ liệu ổn định: Là nền tảng chính để lưu trữ dữ liệu của các hệ thống web viết bằng ASP.NET Core.
* Tương thích tốt với EF Core: Hỗ trợ tốt cho các thao tác ORM (Object-Relational Mapping) khi sử dụng Entity Framework Core với mô hình Code First.
* Hiệu suất cao: Có khả năng tối ưu hóa truy vấn, cache dữ liệu, hỗ trợ lập chỉ mục giúp tăng tốc độ xử lý.
* Dễ triển khai và sao lưu: Có thể dễ dàng cấu hình, triển khai trên server cục bộ hoặc cloud (Azure SQL), sao lưu/khôi phục dữ liệu nhanh chóng.
  1. **Giới thiệu trí tuệ nhân tạo trong hệ thống gợi ý sản phẩm**
     1. **Khái niệm về trí tuệ nhân tạo (AI)**
* Trí tuệ nhân tạo (Artificial Intelligence - AI) là lĩnh vực của khoa học máy tính liên quan đến việc xây dựng các hệ thống có khả năng thực hiện các nhiệm vụ đòi hỏi trí tuệ của con người, như học hỏi, lập luận, nhận thức và ra quyết định.
* Trong thương mại điện tử, AI thường được sử dụng để cải thiện trải nghiệm người dùng, đặc biệt là thông qua hệ thống gợi ý.
  + 1. **Giới thiệu về hệ thống gợi ý**

Hệ thống gợi ý (Recommender systems hoặc Recommendation systems) là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp.

Hai loại hệ thống gợi ý phổ biến:

* Gợi ý dựa trên nội dung (Content-based filtering): Dựa vào các thuộc tính của sản phẩm và lịch sử của người dùng.
* Gợi ý dựa trên cộng tác (Collaborative filtering): Dựa vào hành vi của người dùng khác có hành vi tương tự.
  + 1. **Vai trò của AI trong hệ thống gợi ý**
* **Phân tích dữ liệu hành vi người dùng**: AI có thể học từ dữ liệu lớn (big data) để phát hiện xu hướng mua hàng, hành vi tìm kiếm.
* **Cá nhân hóa nội dung**: Đề xuất các sản phẩm phù hợp với từng người dùng cụ thể, giúp tăng tỷ lệ mua hàng.
* **Tối ưu hóa doanh thu**: Gợi ý đúng sản phẩm giúp cải thiện hiệu quả kinh doanh và giữ chân khách hàng.

# **CHƯƠNG 2.KHẢO SÁT HỆ THỐNG**

* 1. **Giới thiệu về hệ thống**

Trong bối cảnh chuyển đổi số mạnh mẽ hiện nay, thương mại điện tử đã trở thành một phần không thể thiếu trong đời sống hiện đại, đặc biệt là trong lĩnh vực kinh doanh sách. Việc mua sắm sách trực tuyến không chỉ giúp người dùng tiết kiệm thời gian, chi phí mà còn mang đến trải nghiệm tiện lợi và phong phú hơn bao giờ hết. Người đọc không còn phải đến trực tiếp hiệu sách để tìm kiếm tựa sách mong muốn mà có thể dễ dàng truy cập, lựa chọn, đặt hàng và nhận sách ngay tại nhà thông qua các nền tảng trực tuyến.

Trước thực tế đó, đề tài “Giới thiệu về hệ thống bán sách tích hợp bằng AI bằng .NET Core” được triển khai với mục tiêu xây dựng một nền tảng thương mại điện tử hiện đại chuyên về kinh doanh sách, kết hợp công nghệ trí tuệ nhân tạo để nâng cao trải nghiệm người dùng và hiệu quả quản lý. Hệ thống không chỉ cho phép khách hàng xem, tìm kiếm, đặt mua sách mà còn tích hợp các tính năng thông minh như gợi ý sách theo sở thích, quản lý giỏ hàng, xử lý đơn hàng và theo dõi lịch sử mua sắm.

Công nghệ AI (Artificial Intelligence) được tích hợp vào hệ thống đóng vai trò như một “trợ lý ảo” cho người dùng, đưa ra các đề xuất cá nhân hóa dựa trên hành vi duyệt web, lịch sử đặt hàng và xu hướng đọc. Điều này giúp người dùng nhanh chóng tìm được những cuốn sách phù hợp với sở thích hoặc nhu cầu học tập, nghiên cứu. Đồng thời, hệ thống cũng hỗ trợ người quản trị trong việc nắm bắt nhu cầu thị trường, thống kê bán hàng và tối ưu hoạt động tiếp thị.

Nền tảng được phát triển trên .NET Core, một công nghệ hiện đại của Microsoft, giúp hệ thống đạt hiệu suất cao, dễ mở rộng và tương thích đa nền tảng. Hệ thống sử dụng kiến trúc chia tầng rõ ràng (Data – Core – API – Web) để đảm bảo tính bảo trì và dễ dàng tích hợp các tính năng trong tương lai.

Với giao diện thân thiện, quy trình mua hàng mượt mà và khả năng tích hợp AI thông minh, hệ thống hướng tới việc trở thành một mô hình nền tảng bán sách điện tử ứng dụng AI tiêu biểu – vừa mang lại trải nghiệm tiện lợi cho khách hàng, vừa hỗ trợ doanh nghiệp trong việc tối ưu hoạt động kinh doanh trong thời đại số.

* 1. **Các yêu cầu của hệ thống**
     1. **Yêu cầu chức năng**
* Khách vãng lai (Người dùng chưa đăng ký):
  + Xem danh sách các sách đang bán.
* Xem chi tiết thông tin sách (giá, mô tả, bìa sách, số lượng còn...).
* Tìm kiếm sách theo từ khóa (tên sách, tác giả...).
* Lọc sách theo thể loại, nhà xuất bản.
* Xem đánh giá và bình luận (nếu có cấu hình public).
* Đăng ký tài khoản.
* Đăng nhập vào hệ thống.
* Nhận gợi ý sản phẩm chung (gợi ý mặc định cho người chưa đăng nhập - sử dụng AI nếu có dữ liệu lịch sử).
* Khách hàng đã đăng ký (Người dùng có tài khoản):
* Quản lý thông tin tài khoản (xem/sửa thông tin cá nhân, đổi mật khẩu).
* Xem danh mục sách và chi tiết sách.
* Thêm sách vào giỏ hàng.
* Cập nhật, xóa sách trong giỏ hàng.
* Tiến hành thanh toán đơn hàng (checkout).
* Chọn phương thức thanh toán.
* Xem lịch sử đơn hàng đã mua.
* Xem trạng thái đơn hàng.
* Đánh giá và bình luận sách đã mua.
* Nhận gợi ý sản phẩm cá nhân hóa từ hệ thống AI dựa trên hành vi, lịch sử mua hàng, tìm kiếm,...
* Tìm kiếm và lọc sách nâng cao.
* Nhận thông báo về khuyến mãi, đơn hàng,...
* Áp dụng mã khuyến mãi khi mua hàng (nếu có).
* Gửi yêu cầu hỗ trợ
* Người quản trị (Người quản lý hệ thống):
* Quản lý sách:
* Thêm mới sách.
* Sửa thông tin sách.
* Xóa sách.
* Cập nhật số lượng sách.
* Quản lý danh mục:
* Thêm/sửa/xóa thể loại sách.
* Quản lý nhà xuất bản, tác giả,...
* Quản lý người dùng:
* Xem danh sách người dùng.
* Cập nhật, khóa tài khoản người dùng.
* Quản lý đơn hàng:
* Xem danh sách đơn hàng.
* Cập nhật trạng thái đơn hàng (Đang xử lý, Đã giao, Hủy,...).
* Quản lý khuyến mãi:
* Tạo mã giảm giá.
* Cấu hình điều kiện sử dụng mã.
* Quản lý đánh giá:
* Duyệt, xóa đánh giá hoặc bình luận không phù hợp.
* Quản lý hệ thống AI gợi ý:
* Kiểm tra, giám sát kết quả gợi ý sản phẩm từ AI.
* Theo dõi thống kê độ chính xác, hiệu quả gợi ý (nếu có dashboard).
* Thống kê và báo cáo:
* Thống kê doanh thu theo ngày, tháng.
* Thống kê sách bán chạy.
* Thống kê người dùng hoạt động nhiều nhất.
* Nhân viên
  + Đăng nhập hệ thống
  + Quản lý đơn hàng
* Xem danh sách đơn hàng được phân công.
* Cập nhật, xử lý đơn hàng (xác nhận, đóng gói, giao hàng, hủy...).
* Cập nhật kho hàng
  + Cập nhật số lượng tồn kho.
  + Kiểm tra tình trạng sản phẩm.
* Theo dõi tình trạng đơn hàng
* Hỗ trợ khách hàng
* Xử lý đơn hàng
  + Tiếp nhận và thực hiện các đơn hàng mới.
* Xử lý yêu cầu hỗ trợ
  + Tiếp nhận, xử lý các yêu cầu hỗ trợ từ khách hàng qua hệ thống.
* Quản lý cửa hàng
  + Đăng nhập hệ thống.
  + Quản lý kho hàng
    - Xem, cập nhật tình trạng kho hàng.
    - Kiểm soát nhập - xuất sản phẩm.
  + Xem thống kê doanh thu
    - Theo ngày, tháng, quý, năm.
  + Quản lý nhân viên
    - Thêm mới, sửa, xóa thông tin nhân viên.
    - Phân công công việc.
  + Quản lý chương trình khuyến mãi
    - Tạo, sửa, xóa các chương trình khuyến mãi.
    - Cấu hình điều kiện áp dụng khuyến mãi.
    1. **Yêu cầu phi chức năng**

1. Yêu cầu về hiệu năng

* Hệ thống cần phản hồi nhanh với thời gian tải trang dưới 3 giây trong điều kiện mạng ổn định.
* Tìm kiếm sách và gợi ý sản phẩm bằng AI phải trả kết quả trong vòng 2-5 giây.
* Hệ thống có khả năng xử lý tối thiểu 100 người dùng đồng thời mà không bị gián đoạn.

1. Yêu cầu về khả năng mở rộng

* Dự án được xây dựng theo mô hình nhiều tầng (multi-layer: Web, API, Core, Data), thuận tiện cho việc nâng cấp, bảo trì và mở rộng sau này.
* Có thể tích hợp thêm các tính năng mới như chatbot AI, phân tích hành vi người dùng, thanh toán trực tuyến trong tương lai.

1. Yêu cầu về bảo mật

* Phân quyền truy cập:  
  Mỗi loại người dùng (khách chưa đăng nhập, khách đã đăng ký, quản trị viên) có các quyền truy cập rõ ràng và được kiểm soát nghiêm ngặt:
* Khách chưa đăng ký: chỉ được xem danh sách sách, xem chi tiết sản phẩm, tìm kiếm, và xem các chương trình khuyến mãi công khai.
* Khách đã đăng ký: được phép đăng nhập, thêm vào giỏ hàng, đặt hàng, xem và quản lý thông tin tài khoản, đánh giá sản phẩm.
* Quản trị viên: có quyền truy cập vào khu vực quản lý (Admin) bao gồm quản lý sản phẩm, người dùng, đơn hàng, thống kê, khuyến mãi và cấu hình hệ thống.
* Mã hóa mật khẩu:  
  Mật khẩu người dùng sẽ được mã hóa trước khi lưu trữ vào cơ sở dữ liệu nhằm đảm bảo không ai, kể cả quản trị viên, có thể xem được mật khẩu gốc.

1. Yêu cầu về khả năng sử dụng (tính thân thiện)

* Giao diện được thiết kế đơn giản, dễ hiểu, dễ sử dụng với cả người dùng mới và người dùng đã có kinh nghiệm.
* Các chức năng chính như tìm kiếm sách, thêm vào giỏ hàng, thanh toán... được bố trí trực quan, dễ thao tác trên cả máy tính và thiết bị di động.

1. Yêu cầu về độ tin cậy và tính toàn vẹn dữ liệu

* Dữ liệu khách hàng, đơn hàng và sản phẩm phải luôn được đảm bảo không bị mất mát, trùng lặp hoặc sai lệch do lỗi phần mềm.
* Hệ thống cần có cơ chế kiểm tra đầu vào hợp lệ (validation) để tránh lỗi dữ liệu đầu vào và giảm rủi ro bảo mật như SQL Injection.

1. Yêu cầu về sao lưu và khôi phục dữ liệu

* Cần có cơ chế sao lưu dữ liệu định kỳ để khôi phục trong trường hợp hệ thống gặp sự cố.
* Trong phiên bản đầu tiên, em sẽ mô phỏng hoặc ghi rõ trong tài liệu hướng dẫn các bước sao lưu/khôi phục thủ công.

1. Yêu cầu về công nghệ

* Hệ thống được phát triển bằng .NET Core theo mô hình MVC chia tầng (gồm các project: Web, API, Core, Data).
* Giao diện sử dụng Razor Pages hoặc Blazor. CSDL sử dụng SQL Server với mô hình Code First của Entity Framework Core.
* Công nghệ AI sử dụng mô hình gợi ý sản phẩm dựa trên các sách người dùng đã xem/mua.

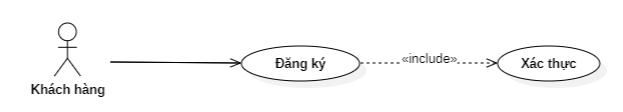
# **CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

* 1. **Phân tích hệ thống**
     1. **Mô hình use case**



Hình 2.1. Biểu đồ use case tổng quát

* Đặc tả use case
* Use case Đăng ký

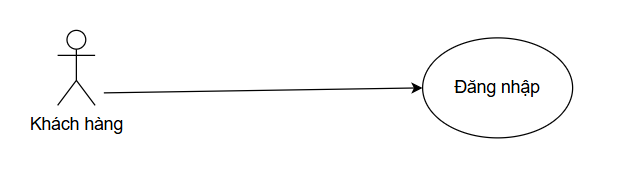


Hình 2.2 Đặc tả use case Đăng ký

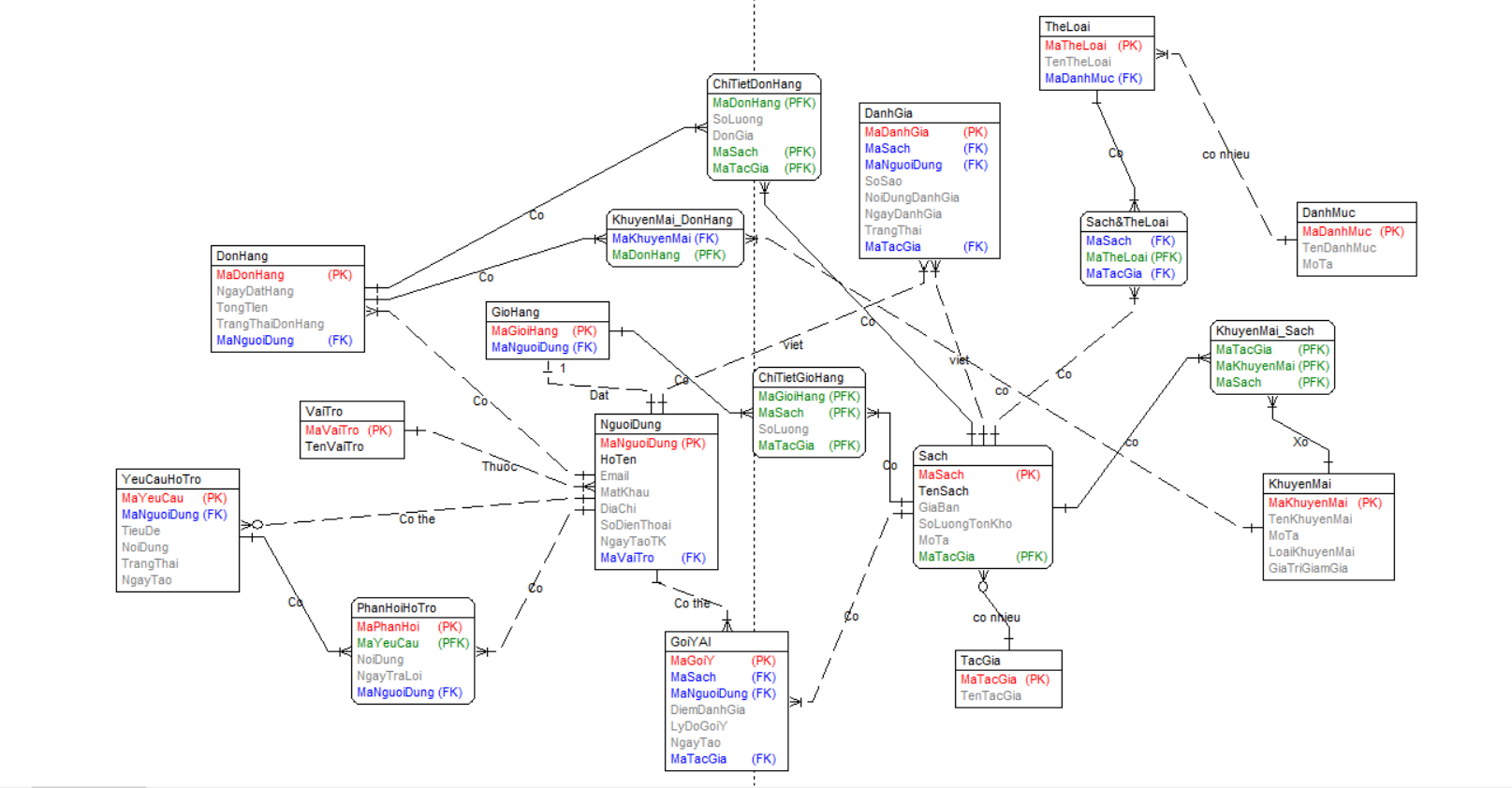
Bảng 2.1. Đặc tả use case Đăng ký

|  |  |
| --- | --- |
| Tên usecase | Đăng ký |
| Mô tả | Use case cho phép khách hàng đăng ký tài khoản tại website |
| Actor | Khách hàng |
| Điều kiện kích hoạt | Người dùng chọn chức năng đăng ký từ hệ thống |
| Tiền điều kiện | Người dùng chưa có tài khoản trên hệ thống |
| Hậu điều kiện | Đăng ký thành công |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Khách hàng | Chọn chức năng đăng ký | | 2 | Hệ thống | Hiển thị form đăng ký | | 3 | Khách hàng | Nhập thông tin đăng ký | | 4 | Khách hàng | Nhấn nút đăng ký | | 5 | Hệ thống | Kiểm tra tính hợp lệ và đầy đủ của các trường khách hàng nhập | | 6 | Hệ thống | Kiểm tra mật khẩu giống xác nhận mật khẩu | | 7 | Hệ thống | Kiểm tra email đã tồn tại hay chưa | | 8 | Hệ thống | Thông báo và chuyển đến trang chủ | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 5a | Hệ thống | Thông báo lỗi: “vui lòng điền đầy đủ thông tin” nếu nhập thiếu trong các trường trên | | 6a | Hệ thống | Thông báo lỗi: “mật khẩu không trùng nhau” nếu nhập mật khẩu và xác nhận mật khẩu khác nhau | | 7a | Hệ thống | Thông báo lỗi: “email đã tồn tại” nếu email này đã được đăng ký tài khoản | |

* Use case Đăng nhập



* + 1. **Mô hình hóa dữ liệu**

****

* + 1. **Phân tích use**
  1. **Thiết kế hệ thống**
     1. **Thiết kế cơ sở dữ liệu**
     2. **Thiết kế giao diện**

1. **CÀI ĐẶT VÀ KIỂM THỬ**