**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙘🕮🙚---



**ĐỒ ÁN CHUYÊN NGÀNH**

**ToyStore – Cửa hàng bán đồ chơi trực tuyến**

Giáo viên hướng dẫn: Lê Gia Công

Sinh viên thực hiện: 2212385 – Nguyễn Huỳnh Tiến Khải

2212379 – Lê Tiến Huy

2212338 – Lê Bình Duy Anh

*Đà Lạt,11 tháng 11 năm 2025*

**NHẬN XÉT**

**LỜI CẢM ƠN**

Đầu tiên chúng em xin được gửi lời cảm ơn tới thầy Lê Gia Công - người đã hướng dẫn và đóng góp nhiều ý kiến cho em trong suốt quá trình học tập và làm dự án. Với sự chỉ bảo của thầy, chúng em đã được trang bị đầy đủ kiến thức, kĩ năng cơ bản cần có, qua đó có được những định hướng cơ bản trong việc triển khai và thực hiện các yêu cầu trong quá trình làm dự án.

Chúng em xin chân thành cảm ơn.

Mục lục

[Chương 1. GIỚI THIỆU VÀ MỤC TIÊU 8](#_Toc215600110)

[1.1. Bối cảnh và vấn đề thực tiễn 8](#_Toc215600111)

[1.1.1. Bối cảnh thương mại điện tử 8](#_Toc215600112)

[1.1.2. Thị trường đồ chơi sưu tầm tại Việt Nam 8](#_Toc215600113)

[1.1.3. Vấn đề thực tiễn 9](#_Toc215600114)

[1.2. Lý do chọn đề tài 10](#_Toc215600115)

[1.2.1. Tính thực tiễn cao 10](#_Toc215600116)

[1.2.2. Ứng dụng kiến thức đã học 10](#_Toc215600117)

[1.3. Mục tiêu của đề tài 10](#_Toc215600118)

[1.3.1. Mục tiêu tổng quát 10](#_Toc215600119)

[1.3.2. Mục tiêu cụ thể 10](#_Toc215600120)

[Chương 2. PHÂN TÍCH YÊU CẦU 12](#_Toc215600121)

[2.1. Khái quát bài toán và hiện trạng 12](#_Toc215600122)

[2.1.1. Bài toán cần giải quyết 12](#_Toc215600123)

[2.1.2. Hiện trạng và hạn chế 13](#_Toc215600124)

[2.2. Các đối tượng sử dụng hệ thống 13](#_Toc215600125)

[2.2.1. Khách hàng (Customer/User) 13](#_Toc215600126)

[2.2.2. Nhân viên (Staff) 14](#_Toc215600127)

[2.2.3. Quản trị viên (Administrator) 14](#_Toc215600128)

[2.3. Yêu cầu chức năng 15](#_Toc215600129)

[2.3.1. Nhóm chức năng xác thực và quản lý tài khoản 15](#_Toc215600130)

[2.3.2. Nhóm chức năng quản lý sản phẩm 16](#_Toc215600131)

[2.3.3. Nhóm chức năng giỏ hàng 17](#_Toc215600132)

[2.3.4. Nhóm chức năng đặt hàng và thanh toán 17](#_Toc215600133)

[2.3.5. Nhóm chức năng quản lý đơn hàng 17](#_Toc215600134)

[2.3.6. Nhóm chức năng vận chuyển (GHN Integration) 18](#_Toc215600135)

[2.3.7. Nhóm chức năng voucher 18](#_Toc215600136)

[2.3.8. Nhóm chức năng đánh giá sản phẩm 19](#_Toc215600137)

[2.3.9. Nhóm chức năng thống kê (Admin) 19](#_Toc215600138)

[2.3.10. Nhóm chức năng khác 19](#_Toc215600139)

[2.4. Yêu cầu phi chức năng 19](#_Toc215600140)

[2.4.1. Hiệu năng 19](#_Toc215600141)

[2.4.2. Bảo mật 20](#_Toc215600142)

[2.4.3. Khả năng sử dụng (Usability) 20](#_Toc215600143)

[2.4.4. Độ tin cậy (Reliability) 21](#_Toc215600144)

[2.4.5. Khả năng mở rộng (Scalability) 21](#_Toc215600145)

[2.4.6. Bảo trì (Maintainability) 21](#_Toc215600146)

[2.5. Kịch bản sử dụng tiêu biểu 21](#_Toc215600147)

[2.5.1. Kịch bản 1: Học sinh mua mô hình Songoku 21](#_Toc215600148)

[2.5.2. Kịch bản 2: Sinh viên mua combo blind box Baby Three 22](#_Toc215600149)

[2.5.3. Kịch bản 3: Người chơi trading cards đặt booster Pokémon 23](#_Toc215600150)

[2.5.4. Kịch bản 4: Admin quản lý đơn hàng và thống kê 24](#_Toc215600151)

[Chương 3. THIẾT KẾ HỆ THỐNG 25](#_Toc215600152)

[3.1. Kiến trúc tổng thể hệ thống 25](#_Toc215600153)

[3.1.1. Lựa chọn kiến trúc 25](#_Toc215600154)

[3.1.2. Sơ đồ kiến trúc tổng thể 26](#_Toc215600155)

[3.1.3. Luồng dữ liệu 26](#_Toc215600156)

[3.2. Thiết kế cơ sở dữ liệu 27](#_Toc215600157)

[3.2.1. Sơ đồ ERD (Entity Relationship Diagram) 27](#_Toc215600158)

[3.2.2. Thiết kế các bảng chính 28](#_Toc215600159)

[3.2.3. Chuẩn hóa và ràng buộc 29](#_Toc215600160)

[3.3. Thiết kế luồng nghiệp vụ chính 29](#_Toc215600161)

[3.3.1. Luồng đăng ký và đăng nhập 29](#_Toc215600162)

[3.3.2. Luồng mua hàng (quan trọng nhất) 30](#_Toc215600163)

[3.3.3. Luồng quản lý trạng thái đơn hàng (State Pattern) 32](#_Toc215600164)

[3.4. Thiết kế luồng xác thực và phân quyền 34](#_Toc215600165)

[3.4.1. JWT Authentication Flow 34](#_Toc215600166)

[3.4.2. Role-Based Access Control (RBAC) 35](#_Toc215600167)

[3.5. Thiết kế các API chính 35](#_Toc215600168)

[3.5.1. Nhóm API Authentication 35](#_Toc215600169)

[3.5.2. Nhóm API Products 35](#_Toc215600170)

[3.5.3. Nhóm API Orders 35](#_Toc215600171)

[3.5.4. Thiết kế request/response mẫu 36](#_Toc215600172)

[3.6. Áp dụng Design Patterns 37](#_Toc215600173)

[3.7. Thiết kế giao diện 37](#_Toc215600174)

[3.7.1. Nguyên tắc thiết kế 37](#_Toc215600175)

[3.7.2. Bố cục và điều hướng 37](#_Toc215600176)

[3.7.3. Thiết kế các trang chính 38](#_Toc215600177)

[3.7.4. Màu sắc và kiểu chữ 39](#_Toc215600178)

[Chương 4. CÀI ĐẶT VÀ CÔNG NGHỆ 45](#_Toc215600179)

[4.1. Môi trường cài đặt 45](#_Toc215600180)

[4.1.1. Yêu cầu hệ thống 45](#_Toc215600181)

[4.1.2. Công cụ phát triển 45](#_Toc215600182)

[4.2. Công nghệ và framework sử dụng 45](#_Toc215600183)

[4.2.1. Backend 45](#_Toc215600184)

[4.2.2. Frontend 46](#_Toc215600185)

[4.2.3. Database 47](#_Toc215600186)

[4.2.4. Tích hợp bên thứ ba 47](#_Toc215600187)

[4.3. Lý do lựa chọn công nghệ 47](#_Toc215600188)

[4.3.1. Lựa chọn Node.js và Express 47](#_Toc215600189)

[4.3.2. Lựa chọn React 48](#_Toc215600190)

[4.3.3. Lựa chọn SQL Server 48](#_Toc215600191)

[4.3.4. Lựa chọn TailwindCSS 48](#_Toc215600192)

[4.4. Cấu trúc tổ chức code 48](#_Toc215600193)

[4.4.1. Cấu trúc Backend 48](#_Toc215600194)

[4.4.2. Cấu trúc Frontend 50](#_Toc215600195)

[4.5. Các đoạn code minh họa tiêu biểu 51](#_Toc215600196)

[4.5.1. Xác thực JWT (Middleware) 51](#_Toc215600197)

[4.5.2. Decorator Pattern - Tính giá đơn hàng 52](#_Toc215600198)

[4.5.3. State Pattern - Quản lý trạng thái đơn hàng 53](#_Toc215600199)

[4.5.4. Singleton Pattern - Logger 54](#_Toc215600200)

[4.5.5. React Service - API Call 56](#_Toc215600201)

[Chương 5. ĐÁNH GIÁ HỆ THỐNG 57](#_Toc215600202)

[5.1. Đánh giá mức độ đáp ứng yêu cầu chức năng 57](#_Toc215600203)

[5.1.1. Nhóm chức năng đã hoàn thành đầy đủ 57](#_Toc215600204)

[5.1.2. Điểm nổi bật trong cài đặt 58](#_Toc215600205)

[5.2. Đánh giá hiệu năng và khả năng mở rộng 58](#_Toc215600206)

[5.2.1. Hiệu năng thực tế 58](#_Toc215600207)

[5.2.2. Tối ưu đã thực hiện 58](#_Toc215600208)

[5.2.3. Khả năng mở rộng 59](#_Toc215600209)

[5.3. Đánh giá bảo mật 59](#_Toc215600210)

[5.3.1. Cơ chế bảo mật đã triển khai 59](#_Toc215600211)

[5.3.2. Điểm mạnh về bảo mật 59](#_Toc215600212)

[5.3.3. Hạn chế về bảo mật 60](#_Toc215600213)

[5.4. Đánh giá giao diện và trải nghiệm người dùng 60](#_Toc215600214)

[5.4.1. Đánh giá UI (User Interface) 60](#_Toc215600215)

[5.4.2. Đánh giá UX (User Experience) 60](#_Toc215600216)

[5.4.3. Đánh giá phù hợp với đối tượng người dùng 61](#_Toc215600217)

[5.5. Hạn chế và điểm cần cải thiện 61](#_Toc215600218)

[5.5.1. Hạn chế về chức năng 61](#_Toc215600219)

[5.5.2. Hạn chế về kỹ thuật 61](#_Toc215600220)

[5.5.3. Hạn chế về bảo mật 62](#_Toc215600221)

[5.5.4. Kế hoạch khắc phục 62](#_Toc215600222)

[Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 62](#_Toc215600223)

[6.1. Tóm tắt kết quả đạt được 62](#_Toc215600224)

[6.1.1. Về mặt chức năng 62](#_Toc215600225)

[6.1.2. Về mặt kỹ thuật 63](#_Toc215600226)

[6.1.3. Về mặt tích hợp 63](#_Toc215600227)

[6.2. Đóng góp của đề tài 64](#_Toc215600228)

[6.2.1. Đóng góp về mặt thực tiễn 64](#_Toc215600229)

[6.2.2. Đóng góp về mặt kỹ thuật 64](#_Toc215600230)

[6.2.3. Đóng góp về mặt học thuật 64](#_Toc215600231)

[6.3. Đánh giá mức độ hoàn thành 65](#_Toc215600232)

[6.3.1. So với mục tiêu ban đầu 65](#_Toc215600233)

[6.3.2. Hạn chế chưa hoàn thành 65](#_Toc215600234)

[6.4. Hướng phát triển 65](#_Toc215600235)

[6.4.1. Ngắn hạn 65](#_Toc215600236)

[6.4.2. Trung hạn 65](#_Toc215600237)

[6.4.3. Dài hạn 66](#_Toc215600238)

[6.5. Bài học kinh nghiệm 66](#_Toc215600239)

[6.5.1. Về kỹ thuật 66](#_Toc215600240)

[6.5.2. Về quy trình 66](#_Toc215600241)

[6.5.3. Về quản lý dự án 67](#_Toc215600242)

[TÀI LIỆU THAM KHẢO 67](#_Toc215600243)

# Chương 1. GIỚI THIỆU VÀ MỤC TIÊU

## 1.1. Bối cảnh và vấn đề thực tiễn

### 1.1.1. Bối cảnh thương mại điện tử

Trong những năm gần đây, thương mại điện tử tại Việt Nam đã có sự phát triển vượt bậc với giá trị thị trường tăng trưởng hàng năm trên 20%. Việc mua sắm trực tuyến không còn là xu hướng mà đã trở thành thói quen tiêu dùng của đại đa số người dân, đặc biệt là thế hệ trẻ. Sự phát triển của hạ tầng công nghệ, phương thức thanh toán điện tử và dịch vụ vận chuyển đã tạo điều kiện thuận lợi cho các cửa hàng trực tuyến hoạt động hiệu quả.

Theo đó, các ngành hàng truyền thống như sách, thời trang, điện tử đã chuyển đổi số thành công. Tuy nhiên, một phân khúc thị trường đang phát triển mạnh mẽ nhưng chưa được khai thác triệt để là thị trường đồ chơi dành cho lứa tuổi thanh thiếu niên và người lớn.

### 1.1.2. Thị trường đồ chơi sưu tầm tại Việt Nam

Khác với đồ chơi truyền thống dành cho trẻ em, thị trường đồ chơi sưu tầm (collectible toys) tại Việt Nam đang chứng kiến sự bùng nổ với các sản phẩm như:

* **Mô hình figure và anime**: Các nhân vật từ manga, anime như Son Goku (Dragon Ball), các sản phẩm Nendoroid, S.H. Figuarts thu hút đông đảo người hâm mộ là học sinh trung học phổ thông, sinh viên và người đi làm trẻ tuổi.
* **Art toys**: Các dòng sản phẩm như Molly, Baby Three, Sonny Angels đã tạo nên cơn sốt trong giới trẻ yêu thích nghệ thuật và sưu tầm. Đặc biệt, hình thức blind box (hộp mù) càng gia tăng sức hấp dẫn với yếu tố may rủi và bất ngờ.
* **EDC và Fidget toys**: Spinner, pop-it, các gadget công nghệ mini không chỉ là đồ chơi giải trí mà còn là phụ kiện hàng ngày của giới văn phòng và sinh viên.
* **LEGO và Building blocks**: Các bộ mô hình lắp ráp phức tạp thu hút không chỉ trẻ em mà còn cả người lớn với tính giải trí và thư giãn cao.
* **Trading cards**: Pokémon, Magic: The Gathering, Yu-Gi-Oh! tạo nên cộng đồng người chơi đông đảo với nhu cầu sưu tầm và thi đấu.
* **Board games**: Các trò chơi chiến thuật như Catan, Cờ tỷ phú, Avalon ngày càng phổ biến trong các nhóm bạn và gia đình.

Theo khảo sát, đối tượng khách hàng chính của thị trường này là học sinh cấp 2 và cấp 3 (12-18 tuổi), sinh viên (18-25 tuổi) và nhóm người đi làm trẻ tuổi (25-35 tuổi) có sở thích sưu tầm. Đây là nhóm có thu nhập ổn định hoặc tiêu dùng thông qua phụ huynh, am hiểu công nghệ và quen thuộc với mua sắm trực tuyến.

### 1.1.3. Vấn đề thực tiễn

Mặc dù nhu cầu ngày càng tăng, thị trường đồ chơi sưu tầm tại Việt Nam vẫn gặp nhiều thách thức:

**Về phía người mua:**

* Khó tiếp cận sản phẩm chính hãng với giá cả hợp lý do nguồn hàng thường được nhập khẩu thông qua nhiều trung gian.
* Thiếu kênh mua sắm uy tín, rõ ràng về nguồn gốc xuất xứ, dẫn đến nguy cơ cao mua phải hàng nhái, hàng kém chất lượng.
* Trải nghiệm mua sắm trực tuyến chưa tối ưu với giao diện website chung chung, thiếu hình ảnh chi tiết, mô tả sản phẩm không đầy đủ.
* Khó tra cứu thông tin đơn hàng, đặc biệt với khách vãng lai chưa có tài khoản.

**Về phía người bán:**

* Quản lý kho hàng thủ công kém hiệu quả, dễ xảy ra tình trạng sai lệch giữa tồn kho thực tế và ghi nhận trên hệ thống.
* Quy trình xử lý đơn hàng phức tạp, tốn thời gian do chưa có sự liên kết giữa các khâu từ đặt hàng, thanh toán đến vận chuyển.
* Thiếu công cụ phân tích dữ liệu để nắm bắt xu hướng tiêu dùng, các sản phẩm bán chạy, từ đó đưa ra quyết định nhập hàng phù hợp.
* Khó quản lý nhiều loại sản phẩm với đặc thù khác nhau (mô hình cần bảo quản cẩn thận, trading cards cần phân loại theo set, blind box cần quản lý theo series).

Để giải quyết những vấn đề trên, việc xây dựng một hệ thống quản lý bán hàng trực tuyến chuyên biệt cho ngành đồ chơi sưu tầm là cần thiết. Hệ thống cần đáp ứng được yêu cầu nghiệp vụ đặc thù của lĩnh vực này, đồng thời mang lại trải nghiệm tốt cho cả người mua và người bán.

## 1.2. Lý do chọn đề tài

### 1.2.1. Tính thực tiễn cao

Đề tài “Xây dựng hệ thống quản lý bán hàng trực tuyến cho cửa hàng đồ chơi ToyStore” được lựa chọn xuất phát từ nhu cầu thực tế của một cửa hàng kinh doanh đồ chơi cho lứa tuổi teen tại khu vực đô thị. Chủ cửa hàng gặp khó khăn trong việc quản lý nghiệp vụ bán hàng khi quy mô mở rộng và số lượng đơn hàng trực tuyến tăng cao.

Thông qua việc xây dựng hệ thống, sinh viên có cơ hội tiếp cận bài toán thực tế, hiểu rõ quy trình nghiệp vụ và yêu cầu cụ thể từ khách hàng, thay vì chỉ làm việc với các bài toán mang tính lý thuyết.

### 1.2.2. Ứng dụng kiến thức đã học

Đề tài yêu cầu tích hợp nhiều kiến thức chuyên ngành đã được học trong chương trình đào tạo:

* **Phân tích và thiết kế hệ thống**: Thu thập yêu cầu, phân tích nghiệp vụ, thiết kế use case, thiết kế cơ sở dữ liệu, xây dựng kiến trúc hệ thống.
* **Lập trình web**: Sử dụng công nghệ frontend (React) và backend (Node.js, Express) để xây dựng giao diện người dùng và API.
* **Cơ sở dữ liệu**: Thiết kế schema, tối ưu truy vấn, đảm bảo tính toàn vẹn dữ liệu trên SQL Server.
* **Kỹ thuật phần mềm**: Áp dụng design patterns (Decorator, Strategy, State, Singleton) để giải quyết các bài toán cụ thể trong hệ thống.
* **Bảo mật thông tin**: Xác thực người dùng bằng JWT, mã hóa mật khẩu, phân quyền truy cập, bảo vệ chống tấn công.
* **Tích hợp hệ thống**: Kết nối với các dịch vụ bên thứ ba như cổng thanh toán VNPay, dịch vụ vận chuyển Giao Hàng Nhanh (GHN), Google OAuth.

## 1.3. Mục tiêu của đề tài

### 1.3.1. Mục tiêu tổng quát

Xây dựng một hệ thống quản lý bán hàng trực tuyến hoàn chỉnh cho cửa hàng đồ chơi sưu tầm ToyStore, đáp ứng nhu cầu quản lý nghiệp vụ hiệu quả cho người bán và mang lại trải nghiệm mua sắm tốt cho khách hàng lứa tuổi teen trở lên.

### 1.3.2. Mục tiêu cụ thể

**Về chức năng:**

* Xây dựng hệ thống xác thực và phân quyền người dùng với ba vai trò: khách hàng, nhân viên và quản trị viên.
* Phát triển các tính năng quản lý sản phẩm đa dạng (mô hình, figure, art toy, EDC, LEGO, trading cards) với khả năng phân loại theo thương hiệu, loại sản phẩm, giá cả.
* Cung cấp giỏ hàng linh hoạt hỗ trợ cả khách đã đăng nhập và khách vãng lai.
* Tích hợp quy trình đặt hàng hoàn chỉnh từ chọn sản phẩm, tính toán giá (bao gồm VAT, voucher giảm giá, phí vận chuyển) đến thanh toán.
* Tích hợp cổng thanh toán VNPay cho phương thức thanh toán trực tuyến.
* Liên kết với dịch vụ vận chuyển Giao Hàng Nhanh (GHN) để tính phí ship tự động và theo dõi đơn hàng.
* Phát triển hệ thống đánh giá sản phẩm giúp khách hàng chia sẻ trải nghiệm và tham khảo ý kiến.
* Xây dựng module quản lý đơn hàng với quy trình trạng thái rõ ràng.
* Cung cấp công cụ thống kê doanh thu, phân tích sản phẩm bán chạy cho quản trị viên.

**Về kỹ thuật:**

* Thiết kế cơ sở dữ liệu chuẩn hóa, đảm bảo tính toàn vẹn và hiệu suất truy vấn.
* Áp dụng kiến trúc ba tầng (client-server) tách biệt frontend và backend để dễ bảo trì và mở rộng.
* Sử dụng design patterns phù hợp (Decorator cho tính giá đơn hàng, Strategy cho lọc sản phẩm, State cho quản lý trạng thái đơn hàng, Singleton cho các tiện ích dùng chung).
* Triển khai cơ chế bảo mật: mã hóa mật khẩu, xác thực JWT, phân quyền theo vai trò, rate limiting chống spam.
* Tối ưu hiệu năng giao diện người dùng với lazy loading, code splitting.

**Về trải nghiệm người dùng:**

* Thiết kế giao diện thân thiện, phù hợp với đối tượng khách hàng teen và giới trẻ yêu thích đồ chơi sưu tầm.
* Đảm bảo website responsive, hoạt động tốt trên cả desktop và mobile.
* Cung cấp chức năng tra cứu đơn hàng công khai cho khách vãng lai.
* Hiển thị thông tin sản phẩm đầy đủ với hình ảnh, mô tả chi tiết, đánh giá từ người mua.

# Chương 2. PHÂN TÍCH YÊU CẦU

## 2.1. Khái quát bài toán và hiện trạng

### 2.1.1. Bài toán cần giải quyết

Cửa hàng đồ chơi sưu tầm ToyStore hiện đang kinh doanh các sản phẩm như mô hình figure anime, art toys, blind box, EDC/Fidget, LEGO, trading cards và board games phục vụ đối tượng khách hàng chủ yếu là thanh thiếu niên, sinh viên và người trẻ yêu thích sưu tầm. Với sự phát triển của thị trường và nhu cầu mua sắm trực tuyến ngày càng tăng, cửa hàng cần một hệ thống quản lý toàn diện để:

* **Quản lý danh mục sản phẩm đa dạng**: Với hàng trăm sản phẩm thuộc nhiều loại khác nhau (figure, art toy, EDC, LEGO, trading cards), mỗi loại có đặc điểm riêng về thông tin mô tả, hình ảnh, thương hiệu, giá cả. Hệ thống cần hỗ trợ phân loại rõ ràng và hiển thị thông tin chi tiết cho từng dòng sản phẩm.
* **Theo dõi tồn kho chính xác**: Sản phẩm đồ chơi sưu tầm có đặc điểm về số lượng giới hạn, đặc biệt là các mặt hàng hiếm hoặc phiên bản giới hạn. Việc cập nhật tồn kho tự động khi có đơn hàng mới hoặc khi hủy đơn là yêu cầu bắt buộc để tránh tình trạng bán vượt quá khả năng cung ứng.
* **Xử lý đơn hàng hiệu quả**: Quy trình từ khi khách hàng đặt hàng, thanh toán, đóng gói đến giao hàng cần được quản lý chặt chẽ với trạng thái rõ ràng. Đặc biệt, việc tích hợp với đơn vị vận chuyển để tính phí ship tự động và theo dõi trạng thái giao hàng sẽ giúp tiết kiệm thời gian và tăng độ tin cậy.
* **Hỗ trợ nhiều hình thức thanh toán**: Khách hàng trẻ tuổi thường ưa chuộng thanh toán trực tuyến qua ví điện tử, chuyển khoản ngân hàng bên cạnh hình thức thanh toán khi nhận hàng (COD). Hệ thống cần tích hợp cổng thanh toán điện tử an toàn.
* **Tạo trải nghiệm mua sắm tốt**: Giao diện người dùng cần thân thiện, dễ sử dụng, có khả năng tìm kiếm và lọc sản phẩm linh hoạt. Hình ảnh sản phẩm cần hiển thị rõ nét, mô tả chi tiết để khách hàng có thể đánh giá trước khi mua.
* **Phân tích dữ liệu kinh doanh**: Quản trị viên cần các công cụ thống kê để theo dõi doanh thu, sản phẩm bán chạy, xu hướng tiêu dùng theo thời gian, từ đó đưa ra quyết định nhập hàng và chiến lược kinh doanh phù hợp.

### 2.1.2. Hiện trạng và hạn chế

Trước khi triển khai hệ thống, cửa hàng ToyStore vận hành theo phương thức truyền thống với nhiều hạn chế:

**Về quản lý sản phẩm:** - Thông tin sản phẩm được lưu trữ trên bảng tính Excel, khó tra cứu và dễ xảy ra sai sót khi cập nhật. - Không có cơ chế phân loại rõ ràng theo thương hiệu, loại sản phẩm, dẫn đến khó khăn trong việc tìm kiếm và báo cáo. - Hình ảnh sản phẩm lưu trữ rời rạc, không được tối ưu cho việc hiển thị trên web.

**Về xử lý đơn hàng:** - Nhận đơn qua Facebook, Zalo, điện thoại dẫn đến thông tin phân tán, dễ nhầm lẫn. - Tính toán giá thủ công bao gồm giá sản phẩm, phí vận chuyển, chiết khấu tốn thời gian và dễ sai sót. - Không có hệ thống theo dõi trạng thái đơn hàng, khách hàng phải liên hệ trực tiếp để hỏi tình trạng giao hàng.

**Về khách hàng:** - Khách hàng khó truy cập thông tin sản phẩm, phải hỏi qua tin nhắn hoặc điện thoại. - Không có kênh chính thức để xem lịch sử mua hàng hoặc theo dõi đơn hàng. - Thiếu cơ chế đánh giá sản phẩm, khiến khách hàng mới khó tham khảo ý kiến người mua trước.

**Về phân tích kinh doanh:** - Không có số liệu thống kê tự động, mọi báo cáo đều phải tổng hợp thủ công. - Khó xác định sản phẩm nào bán chạy, mùa nào có doanh thu cao để lập kế hoạch nhập hàng.

Những hạn chế trên ảnh hưởng trực tiếp đến hiệu quả kinh doanh và trải nghiệm khách hàng, đồng thời gây khó khăn khi cửa hàng muốn mở rộng quy mô.

## 2.2. Các đối tượng sử dụng hệ thống

Hệ thống ToyStore được thiết kế để phục vụ ba nhóm đối tượng chính với các nhu cầu và quyền hạn khác nhau:

### 2.2.1. Khách hàng (Customer/User)

**Đặc điểm:** - Là học sinh cấp 2, cấp 3 (12-18 tuổi), sinh viên (18-25 tuổi) và người trẻ đi làm (25-35 tuổi) có sở thích sưu tầm đồ chơi. - Am hiểu công nghệ, quen thuộc với việc mua sắm trực tuyến. - Quan tâm đến chất lượng sản phẩm, nguồn gốc xuất xứ và đánh giá từ người mua trước.

**Nhu cầu:** - Tìm kiếm và duyệt sản phẩm theo nhiều tiêu chí (loại sản phẩm, thương hiệu, khoảng giá). - Xem thông tin chi tiết sản phẩm với hình ảnh rõ nét, mô tả đầy đủ, đánh giá từ người mua khác. - Thêm sản phẩm vào giỏ hàng và quản lý đơn hàng. - Thanh toán linh hoạt qua nhiều hình thức (COD, chuyển khoản, VNPay). - Theo dõi trạng thái đơn hàng từ khi đặt hàng đến khi nhận được sản phẩm. - Đánh giá và chia sẻ trải nghiệm sau khi mua. - Quản lý thông tin cá nhân và địa chỉ giao hàng.

**Quyền hạn:** - Đăng ký, đăng nhập tài khoản (hỗ trợ cả Google OAuth). - Xem danh sách sản phẩm, chi tiết sản phẩm, đánh giá. - Quản lý giỏ hàng (thêm, xóa, cập nhật số lượng). - Đặt hàng, áp dụng mã voucher. - Xem lịch sử đơn hàng, chi tiết đơn hàng. - Hủy đơn hàng khi ở trạng thái “Chờ xử lý”. - Viết đánh giá cho sản phẩm đã mua. - Cập nhật thông tin cá nhân, quản lý địa chỉ giao hàng.

**Lưu ý:** Hệ thống cũng hỗ trợ khách vãng lai (guest) có thể duyệt sản phẩm, thêm vào giỏ hàng và đặt hàng mà không cần đăng ký tài khoản. Khách vãng lai có thể tra cứu đơn hàng bằng mã đơn hàng.

### 2.2.2. Nhân viên (Staff)

**Đặc điểm:** - Là nhân viên cửa hàng được phân công xử lý đơn hàng, hỗ trợ khách hàng. - Có kiến thức về sản phẩm và quy trình vận hành cửa hàng.

**Nhu cầu:** - Xem danh sách đơn hàng cần xử lý. - Cập nhật trạng thái đơn hàng (xác nhận, đóng gói, giao cho shipper). - Xem thông tin sản phẩm để tư vấn khách hàng. - Tra cứu thông tin khách hàng khi cần hỗ trợ.

**Quyền hạn:** - Đăng nhập vào hệ thống với tài khoản nhân viên. - Xem danh sách đơn hàng với bộ lọc theo trạng thái. - Xem chi tiết đơn hàng. - Cập nhật trạng thái đơn hàng (giới hạn trong các trạng thái được phép). - Xem danh sách sản phẩm và thông tin chi tiết. - **Không được phép:** Thêm/xóa/sửa sản phẩm, quản lý voucher, xem thống kê doanh thu, quản lý tài khoản.

### 2.2.3. Quản trị viên (Administrator)

**Đặc điểm:** - Là chủ cửa hàng hoặc người quản lý có trách nhiệm vận hành toàn bộ hệ thống. - Nắm rõ chiến lược kinh doanh và cần thông tin phân tích để ra quyết định.

**Nhu cầu:** - Quản lý toàn bộ sản phẩm, danh mục, thương hiệu. - Quản lý tồn kho, cập nhật giá, thêm sản phẩm mới. - Theo dõi và xử lý tất cả đơn hàng. - Quản lý voucher, chương trình khuyến mãi. - Quản lý tài khoản người dùng (khóa/mở khóa tài khoản vi phạm). - Quản lý nhân viên và phân quyền. - Xem thống kê doanh thu, phân tích sản phẩm bán chạy. - Quản lý banner, nội dung trang chủ. - Duyệt và quản lý đánh giá sản phẩm.

**Quyền hạn:** - Toàn quyền truy cập mọi chức năng của hệ thống. - CRUD (Create, Read, Update, Delete) đối với sản phẩm, danh mục, thương hiệu. - CRUD voucher và theo dõi lịch sử sử dụng. - Quản lý đơn hàng: xem, cập nhật trạng thái, hủy đơn, tạo đơn vận chuyển GHN. - Quản lý tài khoản người dùng và nhân viên. - Quản lý banner quảng cáo. - Xem các báo cáo thống kê: doanh thu theo thời gian, top sản phẩm bán chạy, top khách hàng. - Duyệt/ẩn đánh giá sản phẩm không phù hợp.

## 2.3. Yêu cầu chức năng

Ảnh có chứa bản phác thảo, biểu đồ, Bản vẽ kỹ thuật, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

Hình 2.3.1 - Sơ đồ: Use Case Diagram - Chức năng chính của hệ thống

### 2.3.1. Nhóm chức năng xác thực và quản lý tài khoản

**FR-1.1: Đăng ký tài khoản** - Người dùng có thể đăng ký tài khoản mới bằng cách cung cấp email, tên đăng nhập, mật khẩu và thông tin cơ bản (họ tên, số điện thoại). - Hệ thống phải kiểm tra tính hợp lệ của email (định dạng, không trùng lặp). - Mật khẩu phải được mã hóa trước khi lưu vào cơ sở dữ liệu. - Hệ thống áp dụng rate limiting (3 lần đăng ký/giờ) để chống spam.

**FR-1.2: Đăng nhập** - Người dùng có thể đăng nhập bằng tên đăng nhập/email và mật khẩu. - Hệ thống xác thực thông tin và cấp JWT token với thời hạn 7 ngày. - Hỗ trợ đăng nhập bằng tài khoản Google (OAuth 2.0). - Admin và Staff có endpoint đăng nhập riêng với xác thực vai trò. - Áp dụng rate limiting (5 lần đăng nhập/15 phút) để chống brute force.

**FR-1.3: Quản lý thông tin cá nhân** - Người dùng có thể xem và cập nhật thông tin cá nhân (họ tên, số điện thoại, email, ngày sinh, giới tính). - Có thể thêm, sửa, xóa địa chỉ giao hàng với đầy đủ thông tin (tỉnh/thành, quận/huyện, phường/xã). - Có thể đặt một địa chỉ làm mặc định.

**FR-1.4: Phân quyền người dùng** - Hệ thống hỗ trợ 3 vai trò: User, Staff, Admin. - Mỗi vai trò có quyền truy cập khác nhau vào các chức năng. - Admin có thể khóa/mở khóa tài khoản người dùng vi phạm.

### 2.3.2. Nhóm chức năng quản lý sản phẩm

**FR-2.1: Hiển thị danh sách sản phẩm (Public)** - Hiển thị danh sách sản phẩm với phân trang (có thể tùy chỉnh số sản phẩm/trang). - Mỗi sản phẩm hiển thị: hình ảnh, tên, giá, đánh giá trung bình, số lượng đánh giá. - Hỗ trợ tìm kiếm theo tên sản phẩm. - Lọc sản phẩm theo: - Loại sản phẩm (mô hình figure, art toy, EDC, LEGO, trading cards, board games) - Thương hiệu - Khoảng giá - Sắp xếp theo: mới nhất, bán chạy nhất, giá tăng dần, giá giảm dần (Strategy Pattern).

**FR-2.2: Xem chi tiết sản phẩm** - Hiển thị thông tin đầy đủ: tên sản phẩm, mô tả chi tiết, giá, số lượng tồn kho, loại sản phẩm, thương hiệu. - Hiển thị album hình ảnh (có thể có nhiều ảnh cho một sản phẩm). - Hiển thị đánh giá và nhận xét từ người mua, sắp xếp theo thời gian. - Hiển thị điểm đánh giá trung bình và tổng số đánh giá.

**FR-2.3: Quản lý sản phẩm (Admin)** - Admin có thể thêm sản phẩm mới với đầy đủ thông tin và upload hình ảnh. - Cập nhật thông tin sản phẩm (tên, mô tả, giá, tồn kho, loại, thương hiệu). - Xóa sản phẩm (soft delete - chỉ đánh dấu không hiển thị, không xóa vật lý). - Upload và quản lý nhiều hình ảnh cho một sản phẩm. - Hệ thống tự động resize và tối ưu hình ảnh khi upload.

**FR-2.4: Quản lý danh mục và thương hiệu (Admin)** - CRUD danh mục sản phẩm (loại đồ chơi). - CRUD thương hiệu. - Không cho phép xóa danh mục/thương hiệu đang có sản phẩm.

### 2.3.3. Nhóm chức năng giỏ hàng

**FR-3.1: Quản lý giỏ hàng cho người dùng đã đăng nhập** - Thêm sản phẩm vào giỏ hàng với số lượng tùy chọn. - Cập nhật số lượng sản phẩm trong giỏ. - Xóa sản phẩm khỏi giỏ hàng. - Xóa toàn bộ giỏ hàng. - Kiểm tra tồn kho trước khi thêm vào giỏ. - Giỏ hàng được lưu trong database, bền vững qua các phiên đăng nhập.

**FR-3.2: Giỏ hàng cho khách vãng lai** - Khách chưa đăng nhập có thể thêm sản phẩm vào giỏ hàng dựa trên SessionID. - Giỏ hàng được lưu tạm trong database với thời gian tồn tại giới hạn. - Khi khách đăng nhập, hệ thống có thể hợp nhất giỏ hàng vãng lai vào tài khoản.

**FR-3.3: Hiển thị giỏ hàng** - Hiển thị danh sách sản phẩm trong giỏ với hình ảnh, tên, giá, số lượng. - Tính tổng tiền tạm tính. - Cảnh báo nếu sản phẩm hết hàng hoặc không đủ tồn kho.

### 2.3.4. Nhóm chức năng đặt hàng và thanh toán

**FR-4.1: Tạo đơn hàng** - Người dùng nhập thông tin giao hàng (họ tên, số điện thoại, địa chỉ chi tiết). - Tích hợp API GHN để chọn tỉnh/thành, quận/huyện, phường/xã. - Tính phí vận chuyển tự động dựa trên địa chỉ giao hàng (GHN API). - Miễn phí vận chuyển nếu đơn hàng >= 500.000đ. - Cho phép nhập mã voucher giảm giá. - Hiển thị breakdown giá: - Tạm tính (tổng tiền sản phẩm) - VAT 10% - Giảm giá từ voucher - Phí vận chuyển - Thành tiền - Chọn phương thức thanh toán: COD hoặc VNPay. - Hệ thống tự động trừ tồn kho và xóa giỏ hàng sau khi đặt hàng thành công.

**FR-4.2: Thanh toán VNPay** - Tạo URL thanh toán VNPay cho đơn hàng. - Xử lý kết quả trả về từ VNPay (thành công/thất bại). - Nhận IPN (Instant Payment Notification) từ VNPay để xác nhận giao dịch. - Cập nhật trạng thái thanh toán của đơn hàng.

**FR-4.3: Áp dụng Decorator Pattern cho tính giá** - Base: Tính tổng tiền sản phẩm gốc. - Decorator 1: Thêm VAT 10%. - Decorator 2: Áp dụng voucher giảm giá (fixed amount hoặc percentage). - Decorator 3: Thêm phí vận chuyển (hoặc miễn phí nếu đủ điều kiện).

### 2.3.5. Nhóm chức năng quản lý đơn hàng

**FR-5.1: Xem lịch sử đơn hàng (User)** - Người dùng xem danh sách đơn hàng của mình với phân trang. - Lọc theo trạng thái đơn hàng. - Hiển thị: mã đơn hàng, ngày đặt, tổng tiền, trạng thái.

**FR-5.2: Xem chi tiết đơn hàng** - Hiển thị thông tin đầy đủ: danh sách sản phẩm, số lượng, giá, tổng tiền. - Thông tin giao hàng, phương thức thanh toán. - Trạng thái đơn hàng và lịch sử chuyển trạng thái. - Thông tin vận chuyển (mã vận đơn, trạng thái GHN nếu có).

**FR-5.3: Hủy đơn hàng (User)** - Người dùng có thể hủy đơn hàng khi ở trạng thái “Chờ xử lý”. - Hệ thống tự động hoàn lại tồn kho khi hủy đơn.

**FR-5.4: Tra cứu đơn hàng công khai** - Khách vãng lai có thể tra cứu đơn hàng bằng mã đơn hàng. - Hiển thị thông tin chi tiết và trạng thái đơn hàng.

**FR-5.5: Quản lý đơn hàng (Admin/Staff)** - Xem tất cả đơn hàng với bộ lọc theo trạng thái, ngày đặt, mã đơn hàng. - Cập nhật trạng thái đơn hàng theo quy trình (State Pattern): - Chờ thanh toán → Chờ xử lý → Đã xác nhận → Đang đóng gói → Sẵn sàng giao hàng → Đang giao hàng → Đã giao hàng → Hoàn thành - Có thể chuyển sang: Đã hủy, Giao hàng thất bại, Đang hoàn tiền, Đã hoàn tiền - Tạo đơn vận chuyển GHN khi chuyển sang “Đang đóng gói”. - In mã vận đơn, theo dõi trạng thái vận chuyển.

### 2.3.6. Nhóm chức năng vận chuyển (GHN Integration)

**FR-6.1: Lấy danh sách địa chỉ** - API lấy danh sách tỉnh/thành phố. - API lấy danh sách quận/huyện theo tỉnh. - API lấy danh sách phường/xã theo quận.

**FR-6.2: Tính phí vận chuyển** - Tính phí ship dựa trên địa chỉ giao hàng, trọng lượng, dịch vụ vận chuyển. - Hỗ trợ chế độ mock để test không cần API key thật.

**FR-6.3: Tạo đơn vận chuyển** - Tạo đơn hàng trên hệ thống GHN khi admin xác nhận đơn. - Nhận mã vận đơn từ GHN. - Lưu thông tin vận chuyển vào database.

**FR-6.4: Webhook từ GHN** - Nhận thông báo cập nhật trạng thái từ GHN. - Tự động cập nhật trạng thái đơn hàng trong hệ thống.

### 2.3.7. Nhóm chức năng voucher

**FR-7.1: Quản lý voucher (Admin)** - Tạo voucher mới với thông tin: mã voucher, loại giảm giá (fixed/percentage), giá trị, điều kiện áp dụng, số lượng, thời gian. - Cập nhật, vô hiệu hóa voucher. - Xem lịch sử sử dụng voucher.

**FR-7.2: Áp dụng voucher (User)** - Nhập mã voucher khi checkout. - Hệ thống validate voucher (còn hiệu lực, còn số lượng, đủ điều kiện). - Tính giảm giá tự động. - Lưu lịch sử sử dụng voucher.

**FR-7.3: Xem voucher khả dụng** - Người dùng xem danh sách voucher đang có hiệu lực. - Hiển thị điều kiện sử dụng, thời gian hết hạn.

### 2.3.8. Nhóm chức năng đánh giá sản phẩm

**FR-8.1: Viết đánh giá (User)** - Người dùng đã mua sản phẩm có thể viết đánh giá. - Đánh giá bao gồm: điểm số (1-5 sao), bình luận. - Một người dùng chỉ có thể đánh giá một sản phẩm một lần.

**FR-8.2: Xem danh sách sản phẩm có thể đánh giá** - Hiển thị danh sách sản phẩm từ các đơn hàng đã hoàn thành mà chưa đánh giá.

**FR-8.3: Quản lý đánh giá (Admin)** - Xem tất cả đánh giá. - Duyệt/ẩn đánh giá không phù hợp. - Xóa đánh giá vi phạm.

**FR-8.4: Hiển thị đánh giá** - Hiển thị đánh giá trên trang chi tiết sản phẩm. - Sắp xếp theo thời gian mới nhất. - Tính điểm trung bình và cập nhật vào thông tin sản phẩm.

### 2.3.9. Nhóm chức năng thống kê (Admin)

**FR-9.1: Thống kê tổng quan** - Tổng doanh thu trong khoảng thời gian. - Số lượng đơn hàng theo trạng thái. - Số lượng khách hàng mới. - Thống kê 7 ngày gần nhất.

**FR-9.2: Thống kê doanh thu** - Doanh thu theo ngày/tuần/tháng/năm. - Biểu đồ doanh thu theo thời gian. - Top khách hàng mua nhiều nhất.

**FR-9.3: Thống kê sản phẩm** - Top sản phẩm bán chạy nhất. - Sản phẩm theo loại, thương hiệu. - Sản phẩm tồn kho nhiều, tồn kho ít.

**FR-9.4: Thống kê phương thức thanh toán** - Tỷ lệ sử dụng COD/VNPay. - Biểu đồ tròn phân bổ phương thức thanh toán.

### 2.3.10. Nhóm chức năng khác

**FR-10.1: Quản lý banner (Admin)** - Upload banner quảng cáo cho trang chủ. - Thiết lập link, thứ tự hiển thị. - Kích hoạt/vô hiệu hóa banner.

**FR-10.2: Quản lý nhân viên (Admin)** - Tạo tài khoản nhân viên. - Phân quyền, khóa/mở khóa tài khoản nhân viên.

## 2.4. Yêu cầu phi chức năng

### 2.4.1. Hiệu năng

**NFR-1.1: Thời gian tải trang** - Trang chủ và danh sách sản phẩm phải tải hoàn tất trong vòng 3 giây với kết nối internet trung bình. - Trang chi tiết sản phẩm tải trong vòng 2 giây.

**NFR-1.2: Thời gian phản hồi API** - Các API tìm kiếm, lọc sản phẩm phải trả về kết quả trong vòng 1 giây. - API tạo đơn hàng phải hoàn tất trong vòng 3 giây.

**NFR-1.3: Khả năng chịu tải** - Hệ thống phải xử lý được tối thiểu 100 người dùng đồng thời mà không bị chậm đáng kể. - Database phải hỗ trợ tối thiểu 10,000 sản phẩm và 50,000 đơn hàng mà không ảnh hưởng hiệu năng.

### 2.4.2. Bảo mật

**NFR-2.1: Xác thực và ủy quyền** - Tất cả mật khẩu phải được mã hóa bằng bcrypt với salt rounds >= 10. - Sử dụng JWT token cho xác thực, token có thời gian sống 7 ngày. - Hỗ trợ OAuth 2.0 (Google) với client ID/Secret được bảo mật.

**NFR-2.2: Bảo vệ dữ liệu** - Thông tin thanh toán phải được bảo mật, chỉ lưu trữ thông tin cần thiết. - Kết nối với VNPay phải sử dụng HTTPS và HMAC SHA512 để bảo mật thông tin giao dịch.

**NFR-2.3: Chống tấn công** - Áp dụng rate limiting cho các endpoint nhạy cảm: - Login: 5 lần/15 phút - Register: 3 lần/giờ - Cart operations: 50 lần/10 phút - Order operations: 20 lần/giờ - Sử dụng Sequelize ORM để chống SQL Injection. - React tự động escape output để chống XSS.

**NFR-2.4: Phân quyền** - Kiểm tra quyền truy cập ở tầng middleware trước khi thực thi logic nghiệp vụ. - Các endpoint quản trị phải yêu cầu token admin hợp lệ.

### 2.4.3. Khả năng sử dụng (Usability)

**NFR-3.1: Giao diện người dùng** - Giao diện phải thân thiện, phù hợp với đối tượng khách hàng teen và giới trẻ. - Sử dụng màu sắc, typography hiện đại, không quá cầu kỳ. - Các nút bấm, liên kết phải rõ ràng, dễ nhận biết.

**NFR-3.2: Responsive Design** - Website phải hiển thị tốt trên các thiết bị: desktop, tablet, mobile. - Giao diện tự động điều chỉnh theo kích thước màn hình. - Các chức năng chính (tìm kiếm, giỏ hàng, checkout) phải dễ sử dụng trên mobile.

**NFR-3.3: Trải nghiệm mua sắm** - Quy trình mua hàng phải đơn giản, tối đa 3-4 bước từ giỏ hàng đến hoàn tất đơn. - Khách vãng lai có thể mua hàng mà không bắt buộc đăng ký. - Hiển thị thông báo rõ ràng khi có lỗi hoặc thành công.

**NFR-3.4: Hợp truy cập** - Website phải hoạt động tốt trên các trình duyệt phổ biến: Chrome, Firefox, Safari, Edge.

### 2.4.4. Độ tin cậy (Reliability)

**NFR-4.1: Thời gian hoạt động** - Hệ thống cần đảm bảo uptime tối thiểu 99% trong môi trường production.

**NFR-4.2: Xử lý lỗi** - Hệ thống phải có cơ chế xử lý lỗi graceful, không để lộ thông tin nhạy cảm khi có lỗi. - Logging đầy đủ các lỗi để dễ dàng debug và khắc phục.

**NFR-4.3: Toàn vẹn dữ liệu** - Sử dụng database transaction cho các thao tác như tạo đơn hàng (trừ tồn kho, tạo hóa đơn, xóa giỏ hàng). - Đảm bảo tính nhất quán của dữ liệu khi có nhiều người dùng đồng thời.

### 2.4.5. Khả năng mở rộng (Scalability)

**NFR-5.1: Kiến trúc** - Sử dụng kiến trúc ba tầng (client-server) để dễ dàng mở rộng từng tầng độc lập. - Frontend và Backend tách biệt, giao tiếp qua RESTful API.

**NFR-5.2: Database** - Thiết kế schema chuẩn hóa để dễ dàng thêm bảng, trường mới. - Sử dụng index cho các trường thường xuyên truy vấn.

**NFR-5.3: Code quality** - Áp dụng design patterns để code dễ bảo trì và mở rộng. - Tổ chức code theo module, tách biệt concern.

### 2.4.6. Bảo trì (Maintainability)

**NFR-6.1: Tài liệu** - Code phải có comment rõ ràng cho các phần logic phức tạp. - README file hướng dẫn cài đặt, cấu hình, chạy hệ thống.

**NFR-6.2: Logging** - Sử dụng Singleton Logger để ghi log nhất quán trong toàn hệ thống. - Log các sự kiện quan trọng: đăng nhập, tạo đơn hàng, lỗi hệ thống.

**NFR-6.3: Code organization** - Cấu trúc thư mục rõ ràng, phân tách theo chức năng (models, controllers, routes, services). - Đặt tên biến, hàm, file theo convention nhất quán.

## 2.5. Kịch bản sử dụng tiêu biểu

Để minh họa rõ hơn về cách hệ thống hoạt động, dưới đây là một số kịch bản sử dụng thực tế:

### 2.5.1. Kịch bản 1: Học sinh mua mô hình Songoku

**Đối tượng:** Nam, 16 tuổi, học sinh lớp 10, fan của Dragon Ball.

**Mục tiêu:** Tìm mua mô hình Son Goku phiên bản Super Saiyan.

**Các bước thực hiện:**

1. Nam truy cập website ToyStore qua điện thoại di động.
2. Trên thanh tìm kiếm, Nam gõ “Son Goku”.
3. Hệ thống hiển thị danh sách các mô hình liên quan đến Son Goku.
4. Nam lọc theo loại “Mô hình Figure” và thương hiệu “S.H. Figuarts”.
5. Nam chọn sản phẩm “S.H. Figuarts Son Goku Super Saiyan”.
6. Trang chi tiết hiển thị hình ảnh chi tiết, mô tả sản phẩm, giá 1.200.000đ, đánh giá 4.8/5 sao từ 15 người mua.
7. Nam đọc các đánh giá, thấy nhiều người khen chất lượng tốt.
8. Nam bấm “Thêm vào giỏ hàng”, chọn số lượng 1.
9. Hệ thống thông báo đã thêm thành công.
10. Nam bấm “Thanh toán”.
11. Vì chưa có tài khoản, Nam chọn “Tiếp tục với tư cách khách”.
12. Nam nhập thông tin: Họ tên, số điện thoại, chọn tỉnh/quận/phường (Hà Nội, Cầu Giấy, Dịch Vọng), nhập địa chỉ chi tiết.
13. Hệ thống tự động tính phí ship 30.000đ.
14. Tổng tiền hiển thị:
    * Tạm tính: 1.200.000đ
    * VAT 10%: 120.000đ
    * Phí ship: 30.000đ (miễn phí nếu >= 500K, nhưng tổng tạm tính + VAT < 500K nên vẫn tính)
    * Thành tiền: 1.350.000đ
15. Nam không có mã voucher, chọn thanh toán COD.
16. Hệ thống tạo đơn hàng, trừ tồn kho, gửi mã đơn hàng qua số điện thoại.
17. Nam nhận được mã đơn hàng: HD20241130001.
18. Nam có thể tra cứu đơn hàng bất cứ lúc nào bằng mã này mà không cần đăng nhập.

**Kết quả:** Đơn hàng được tạo thành công, Nam nhận được sản phẩm sau 2-3 ngày.

### 2.5.2. Kịch bản 2: Sinh viên mua combo blind box Baby Three

**Đối tượng:** Linh, 20 tuổi, sinh viên năm 2, yêu thích art toys.

**Mục tiêu:** Mua combo 3 blind box Baby Three để có cơ hội cao hơn trúng mẫu hiếm.

**Các bước thực hiện:**

1. Linh đã có tài khoản, đăng nhập vào hệ thống.
2. Linh vào danh mục “Art Toys” → Thương hiệu “Baby Three”.
3. Chọn sản phẩm “Baby Three Mystery Box Series 2” giá 150.000đ/box.
4. Thêm vào giỏ 3 box.
5. Vào giỏ hàng, kiểm tra tổng: 450.000đ.
6. Bấm thanh toán.
7. Hệ thống tự động load địa chỉ mặc định của Linh (đã lưu trước đó).
8. Linh nhập mã voucher “MEGA100K” (giảm 100.000đ cho đơn >= 400.000đ).
9. Hệ thống validate voucher thành công.
10. Tổng tiền sau khi áp dụng:
    * Tạm tính: 450.000đ
    * VAT 10%: 45.000đ
    * Giảm giá voucher: -100.000đ
    * Phí ship: 0đ (miễn phí vì đơn >= 500.000đ, nhưng sau giảm giá < 500K nên tính lại… chính xác là 30.000đ)
    * Thành tiền: 425.000đ
11. Linh chọn thanh toán VNPay.
12. Hệ thống chuyển đến trang VNPay.
13. Linh chọn ngân hàng và hoàn tất thanh toán.
14. VNPay redirect về website, hiển thị thanh toán thành công.
15. Linh vào “Lịch sử đơn hàng” để xem chi tiết.

**Kết quả:** Đơn hàng được thanh toán thành công, Linh nhận được 3 box sau 1 ngày.

### 2.5.3. Kịch bản 3: Người chơi trading cards đặt booster Pokémon

**Đối tượng:** Hưng, 24 tuổi, nhân viên văn phòng, chơi Pokémon TCG.

**Mục tiêu:** Đặt mua 1 display (36 booster packs) Pokémon Scarlet & Violet để mở bài.

**Các bước thực hiện:**

1. Hưng đăng nhập vào hệ thống.
2. Vào danh mục “Trading Cards” → “Pokémon”.
3. Tìm “Pokémon TCG Scarlet & Violet Booster Display” giá 4.500.000đ.
4. Xem chi tiết, đọc đánh giá từ người mua trước về tỷ lệ thẻ hiếm.
5. Thêm vào giỏ, thanh toán.
6. Chọn địa chỉ giao hàng (TP.HCM, Quận 1).
7. Không có mã voucher.
8. Tổng tiền:
   * Tạm tính: 4.500.000đ
   * VAT 10%: 450.000đ
   * Giảm giá: 0đ
   * Phí ship: 0đ (miễn phí vì đơn >= 500.000đ)
   * Thành tiền: 4.950.000đ
9. Chọn COD vì đơn giá trị cao muốn kiểm tra hàng trước.
10. Hoàn tất đặt hàng.
11. Admin xác nhận đơn → chuyển sang “Đang đóng gói”.
12. Admin tạo đơn GHN, nhận mã vận đơn.
13. Hưng nhận notification có mã vận đơn, theo dõi trên GHN.
14. Sau khi nhận hàng, Hưng vào “Sản phẩm có thể đánh giá”, viết review 5 sao, kèm hình ảnh các thẻ hiếm đã mở được.

**Kết quả:** Đơn hàng được giao thành công, Hưng hài lòng và viết đánh giá tích cực.

### 2.5.4. Kịch bản 4: Admin quản lý đơn hàng và thống kê

**Đối tượng:** Admin, chủ cửa hàng ToyStore.

**Mục tiêu:** Xử lý đơn hàng mới và xem báo cáo doanh thu tháng.

**Các bước thực hiện:**

1. Admin đăng nhập vào hệ thống với tài khoản quản trị.
2. Vào trang “Quản lý đơn hàng”, lọc theo trạng thái “Chờ xử lý”.
3. Có 5 đơn hàng mới cần xử lý.
4. Admin click vào đơn đầu tiên, xem chi tiết sản phẩm, địa chỉ giao hàng.
5. Xác nhận đơn → trạng thái chuyển sang “Đã xác nhận”.
6. Admin bấm “Tạo đơn GHN”.
7. Hệ thống gọi API GHN, tạo đơn vận chuyển, nhận mã vận đơn.
8. Trạng thái tự động chuyển sang “Đang đóng gói”.
9. Admin in mã vận đơn, dán lên kiện hàng.
10. Chuyển trạng thái sang “Sẵn sàng giao hàng”.
11. Shipper GHN đến lấy hàng, trạng thái tự động cập nhật sang “Đang giao hàng” (qua webhook).
12. Sau khi giao thành công, trạng thái chuyển sang “Đã giao hàng”.
13. 7 ngày sau không có khiếu nại, Admin chuyển sang “Hoàn thành”.
14. Admin vào trang “Thống kê”, chọn khoảng thời gian tháng 11/2024.
15. Hệ thống hiển thị:
    * Tổng doanh thu: 50.000.000đ
    * Số đơn hàng: 125 đơn
    * Top sản phẩm bán chạy: Molly blind box, Son Goku figure, Pokémon booster
    * Biểu đồ doanh thu theo ngày
    * Top khách hàng: 3 khách hàng mua nhiều nhất
16. Admin sử dụng thông tin này để quyết định nhập thêm Molly và Son Goku cho tháng sau.

**Kết quả:** Đơn hàng được xử lý nhanh chóng, admin có dữ liệu để ra quyết định kinh doanh.

# Chương 3. THIẾT KẾ HỆ THỐNG

## 3.1. Kiến trúc tổng thể hệ thống

### 3.1.1. Lựa chọn kiến trúc

Hệ thống ToyStore được thiết kế theo mô hình kiến trúc ba tầng (3-tier architecture) nhằm tách biệt rõ ràng giữa tầng giao diện, tầng xử lý nghiệp vụ và tầng dữ liệu. Lựa chọn này phù hợp với yêu cầu phi chức năng về khả năng mở rộng, bảo trì và bảo mật.

**Ba tầng chính:**

1. **Tầng trình bày (Presentation Layer)**: Ứng dụng React chạy trên trình duyệt web, cung cấp giao diện người dùng.
2. **Tầng xử lý nghiệp vụ (Business Logic Layer)**: API server Node.js/Express xử lý logic nghiệp vụ, xác thực, phân quyền.
3. **Tầng dữ liệu (Data Layer)**: SQL Server lưu trữ và quản lý dữ liệu.

### 3.1.2. Sơ đồ kiến trúc tổng thể

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.1.2.1 - Sơ đồ: Kiến trúc ba tầng chi tiết

### 3.1.3. Luồng dữ liệu

1. **Client → Frontend**: Người dùng tương tác với giao diện React
2. **Frontend → Backend**: Gửi HTTP request tới API endpoints
3. **Backend → Database**: Truy vấn/cập nhật dữ liệu qua Sequelize ORM
4. **Database → Backend**: Trả kết quả dưới dạng recordset
5. **Backend → Frontend**: Trả response dạng JSON
6. **Frontend → Client**: Hiển thị kết quả cho người dùng

## 3.2. Thiết kế cơ sở dữ liệu

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.2.1 – Sơ đồ Diagram 18 bảng dữ liệu

### 3.2.1. Sơ đồ ERD (Entity Relationship Diagram)

Hệ thống được nhóm thành 8 nhóm chức năng:

**Nhóm 1: Quản lý tài khoản (3 bảng)** - TaiKhoan (1) ←→ (1) KhachHang - TaiKhoan (1) ←→ (N) DiaChiGiaoHangUser

**Nhóm 2: Quản lý sản phẩm (4 bảng)** - SanPham (N) ←→ (1) LoaiSP - SanPham (N) ←→ (1) ThuongHieu  
- SanPham (1) ←→ (N) SanPhamHinhAnh

**Nhóm 3: Giỏ hàng (3 bảng)** - TaiKhoan (1) ←→ (1) GioHang (1) ←→ (N) GioHangChiTiet - GioHangKhachVangLai (khách vãng lai)

**Nhóm 4: Đơn hàng & thanh toán (4 bảng)** - KhachHang (1) ←→ (N) HoaDon - HoaDon (1) ←→ (N) ChiTietHoaDon - HoaDon (N) ←→ (1) PhuongThucThanhToan - HoaDon (1) ←→ (1) DiaChiGiaoHang

**Nhóm 5: Vận chuyển (1 bảng)** - HoaDon (1) ←→ (1) ThongTinVanChuyen

**Nhóm 6: Voucher (2 bảng)** - Voucher (1) ←→ (N) HoaDon - Voucher (1) ←→ (N) LichSuSuDungVoucher

**Nhóm 7: Đánh giá (1 bảng)** - SanPham (1) ←→ (N) DanhGiaSanPham - TaiKhoan (1) ←→ (N) DanhGiaSanPham

**Nhóm 8: Nội dung (1 bảng)** - Banner

### 3.2.2. Thiết kế các bảng chính

**Bảng TaiKhoan** (Authentication)

ID INT PRIMARY KEY IDENTITY  
TenDangNhap NVARCHAR(50) UNIQUE NOT NULL  
MatKhau NVARCHAR(255) NOT NULL -- bcrypt hashed  
Email NVARCHAR(100) UNIQUE  
VaiTro NVARCHAR(20) -- 'admin', 'staff', 'user'  
TrangThai BIT DEFAULT 1  
GoogleID NVARCHAR(255) – Oauth

**Bảng SanPham** (Products)

ID INT PRIMARY KEY IDENTITY  
Ten NVARCHAR(200) NOT NULL  
LoaiID INT FOREIGN KEY REFERENCES LoaiSP(ID)  
ThuongHieuID INT FOREIGN KEY REFERENCES ThuongHieu(ID)  
GiaBan DECIMAL(18,2) NOT NULL  
SoLuongTon INT DEFAULT 0  
MoTa NTEXT  
HinhAnhURL NVARCHAR(500)  
TrangThai BIT DEFAULT 1  
TongSoDanhGia INT DEFAULT 0  
DiemTrungBinh DECIMAL(3,2) DEFAULT 0.00

**Bảng HoaDon** (Orders) - Quan trọng nhất

ID INT PRIMARY KEY IDENTITY  
MaHD NVARCHAR(50) UNIQUE NOT NULL  
KhachHangID INT FOREIGN KEY  
NgayLap DATETIME DEFAULT GETDATE()  
PhuongThucThanhToanID INT FOREIGN KEY  
TrangThai NVARCHAR(50) -- State Pattern  
-- Các trường tính giá (Decorator Pattern)  
TienGoc DECIMAL(18,2) NOT NULL  
VoucherID INT FOREIGN KEY (nullable)  
GiamGia DECIMAL(18,2) DEFAULT 0  
TienShip DECIMAL(18,2) DEFAULT 0  
TyLeVAT DECIMAL(5,2) DEFAULT 0.10  
TienVAT DECIMAL(18,2) DEFAULT 0  
ThanhTien DECIMAL(18,2) NOT NULL  
GhiChu NTEXT

### 3.2.3. Chuẩn hóa và ràng buộc

Hệ thống áp dụng chuẩn hóa 3NF để đảm bảo: - Không có dư thừa dữ liệu - Tính toàn vẹn tham chiếu qua Foreign Keys - Hiệu suất truy vấn tối ưu với indexes

**Các ràng buộc chính:** - UNIQUE: TenDangNhap, Email, MaHD - NOT NULL: Các trường bắt buộc - FOREIGN KEY: Đảm bảo tham chiếu hợp lệ - CHECK: Giá trị hợp lệ (VD: GiaBan >= 0)

## 3.3. Thiết kế luồng nghiệp vụ chính

### 3.3.1. Luồng đăng ký và đăng nhập

**Đăng ký tài khoản:**

1. User nhập thông tin → Frontend validate  
2. Gửi POST /api/auth/register  
3. Backend: Check email/username trùng  
4. Hash password bằng bcrypt (salt=10)  
5. Insert vào TaiKhoan + KhachHang  
6. Trả về success hoặc error

**Đăng nhập:**

1. User nhập username/password  
2. POST /api/auth/login  
3. Backend: Query TaiKhoan  
4. Compare password với bcrypt  
5. Generate JWT token (expire 7 ngày)  
6. Trả token về client  
7. Client lưu token vào localStorage  
8. Các request sau gửi token qua header

### 3.3.2. Luồng mua hàng (quan trọng nhất)

**Bước 1: Thêm vào giỏ hàng**

1. User chọn sản phẩm, số lượng  
2. POST /api/cart/add {sanPhamId, soLuong}  
3. Backend: Check tồn kho  
4. Nếu đủ hàng: Insert/Update GioHangChiTiet  
5. Trả về giỏ hàng hiện tại

**Bước 2: Checkout và đặt hàng**

1. User vào /checkout  
2. Nhập địa chỉ giao hàng (GHN API)  
3. Tính phí ship tự động  
4. Nhập mã voucher (optional)  
5. Áp dụng Decorator Pattern tính giá:  
 - Base: Tổng tiền sản phẩm  
 - +VAT 10%  
 - -Voucher  
 - +Phí ship (free nếu >=500K)  
6. Chọn phương thức: COD/VNPay  
7. POST /api/orders/create

**Bước 3: Xử lý đặt hàng (Backend Transaction)**

BEGIN TRANSACTION  
 -- 1. Tạo HoaDon  
 INSERT INTO HoaDon (...)  
   
 -- 2. Tạo ChiTietHoaDon  
 INSERT INTO ChiTietHoaDon (...)  
   
 -- 3. Trừ tồn kho  
 UPDATE SanPham SET SoLuongTon = SoLuongTon - @soLuong  
   
 -- 4. Lưu địa chỉ giao hàng  
 INSERT INTO DiaChiGiaoHang (...)  
   
 -- 5. Lưu lịch sử voucher  
 INSERT INTO LichSuSuDungVoucher (...)  
   
 -- 6. Xóa giỏ hàng  
 DELETE FROM GioHangChiTiet WHERE GioHangID = @gioHangId  
   
COMMIT TRANSACTION

### 3.3.3. Luồng quản lý trạng thái đơn hàng (State Pattern)

Ảnh có chứa ảnh chụp màn hình, văn bản, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.3.3.1 - Luồng quản lý trạng thái đơn hàng

Chờ thanh toán (VNPay chưa thanh toán)  
 ↓  
Chờ xử lý (Đơn mới, chưa xác nhận)  
 ↓  
Đã xác nhận (Admin đã xác nhận)  
 ↓  
Đang đóng gói (Tạo đơn GHN, có mã vận đơn)  
 ↓  
Sẵn sàng giao hàng (Chờ shipper lấy)  
 ↓  
Đang giao hàng (Shipper đang giao)  
 ↓  
Đã giao hàng (Khách đã nhận)  
 ↓  
Hoàn thành (Không khiếu nại sau 7 ngày)

Mỗi trạng thái là một class riêng với: - canTransitionTo(): Kiểm tra có thể chuyển sang trạng thái nào - onEnter(): Hành động khi vào trạng thái mới - onExit(): Hành động trước khi rời trạng thái

## 3.4. Thiết kế luồng xác thực và phân quyền

### 3.4.1. JWT Authentication Flow

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

### 3.4.2. Role-Based Access Control (RBAC)

**Middleware chain:**

// Public route  
app.get('/api/products', productController.getAll);  
  
// User route (cần đăng nhập)  
app.post('/api/orders/create',   
 authMiddleware.verifyToken,   
 orderController.create);  
  
// Admin route  
app.post('/api/admin/products',   
 authMiddleware.verifyToken,  
 authMiddleware.requireAdmin,  
 adminProductController.create);

## 3.5. Thiết kế các API chính

### 3.5.1. Nhóm API Authentication

| Method | Endpoint | Mô tả | Auth |
| --- | --- | --- | --- |
| POST | /api/auth/register | Đăng ký tài khoản | Public |
| POST | /api/auth/login | Đăng nhập user | Public |
| POST | /api/auth/admin/login | Đăng nhập admin | Public |
| GET | /api/auth/google | OAuth Google | Public |

### 3.5.2. Nhóm API Products

| Method | Endpoint | Mô tả | Auth |
| --- | --- | --- | --- |
| GET | /api/products | Danh sách sản phẩm | Public |
| GET | /api/products/:id | Chi tiết sản phẩm | Public |
| POST | /api/admin/products | Thêm sản phẩm | Admin |
| PUT | /api/admin/products/:id | Sửa sản phẩm | Admin |
| DELETE | /api/admin/products/:id | Xóa sản phẩm | Admin |

### 3.5.3. Nhóm API Orders

| Method | Endpoint | Mô tả | Auth |
| --- | --- | --- | --- |
| POST | /api/orders/create | Tạo đơn hàng | User/Guest |
| GET | /api/orders/my-orders | Đơn hàng của tôi | User |
| GET | /api/orders/:id | Chi tiết đơn hàng | User |
| POST | /api/orders/:id/cancel | Hủy đơn hàng | User |
| PATCH | /api/admin/orders/:id/status | Cập nhật trạng thái | Admin |

### 3.5.4. Thiết kế request/response mẫu

**POST /api/orders/create**

Request:

{  
 "diaChiGiaoHang": {  
 "hoTen": "Nguyễn Văn A",  
 "soDienThoai": "0123456789",  
 "tinhThanh": "Hà Nội",  
 "quanHuyen": "Cầu Giấy",   
 "phuongXa": "Dịch Vọng",  
 "diaChi": "Số 1, Ngõ 2"  
 },  
 "phuongThucThanhToanId": 1,  
 "maVoucher": "MEGA100K",  
 "ghiChu": "Giao giờ hành chính"  
}

Response:

{  
 "success": true,  
 "message": "Đặt hàng thành công",  
 "data": {  
 "maHD": "HD20241130001",  
 "tienGoc": 450000,  
 "tienVAT": 45000,  
 "giamGia": 100000,  
 "tienShip": 30000,  
 "thanhTien": 425000  
 }  
}

## 3.6. Áp dụng Design Patterns

Hệ thống áp dụng 4 design patterns chính để giải quyết các vấn đề cụ thể:

**1. Decorator Pattern**: Tính giá đơn hàng linh hoạt (VAT, voucher, shipping)

**2. Strategy Pattern**: Sắp xếp sản phẩm theo nhiều tiêu chí (mới nhất, bán chạy, giá)

**3. State Pattern**: Quản lý 12 trạng thái đơn hàng với quy tắc chuyển đổi rõ ràng

**4. Singleton Pattern**: Đảm bảo chỉ 1 instance cho Logger, ConfigService, DBConnection

## 3.7. Thiết kế giao diện

### 3.7.1. Nguyên tắc thiết kế

Giao diện hệ thống được thiết kế theo hướng lấy người dùng làm trung tâm, đặc biệt ưu tiên trải nghiệm cho nhóm đối tượng teen và giới trẻ từ 13-25 tuổi. Toàn bộ giao diện được xây dựng theo nguyên tắc đơn giản, trực quan, dễ sử dụng mà không cần hướng dẫn phức tạp. Do đặc điểm đối tượng sử dụng smartphone nhiều, hệ thống áp dụng phương pháp thiết kế ưu tiên điện thoại trước, sau đó mở rộng lên máy tính bảng và máy tính để bàn.

Về phân cấp thị giác, các thông tin quan trọng như giá sản phẩm và nút mua hàng được làm nổi bật để người dùng dễ dàng nhận biết. Hình ảnh sản phẩm được ưu tiên hiển thị với kích thước lớn và chất lượng cao, vì đây là yếu tố quyết định hàng đầu trong quyết định mua sắm đồ chơi sưu tầm. Các nút kêu gọi hành động được thiết kế rõ ràng với màu sắc nổi bật và vị trí dễ tiếp cận.

Tính nhất quán được đảm bảo thông qua việc sử dụng TailwindCSS làm hệ thống thiết kế chuẩn. Tất cả các thành phần giao diện như nút bấm, thẻ sản phẩm, hộp thoại đều được xây dựng dưới dạng component tái sử dụng, đảm bảo trải nghiệm thống nhất trên toàn bộ hệ thống.

### 3.7.2. Bố cục và điều hướng

Phần đầu trang được chia thành ba khu vực chính: logo thương hiệu đặt bên trái giúp tăng nhận diện, thanh tìm kiếm nằm giữa để người dùng dễ dàng tiếp cận, và menu người dùng đặt bên phải hiển thị thông tin tài khoản và giỏ hàng. Thanh điều hướng chính bao gồm các mục Trang chủ, Sản phẩm, Giỏ hàng và Đơn hàng, được cố định khi người dùng cuộn trang để luôn dễ dàng truy cập.

Khu vực nội dung chính sử dụng bố cục lưới tự động điều chỉnh theo kích thước màn hình. Trên điện thoại, sản phẩm hiển thị 1 cột để tối ưu không gian. Máy tính bảng hiển thị 2-3 cột, trong khi màn hình lớn có thể hiển thị tối đa 4 cột. Thanh lọc sản phẩm được đặt bên trái trên desktop, nhưng chuyển thành ngăn kéo trên di động. Hệ thống đường dẫn điều hướng giúp người dùng luôn biết vị trí hiện tại trong cấu trúc trang.

Phần chân trang chứa các thông tin hữu ích như liên hệ, chính sách bảo hành đổi trả, liên kết đến các trang mạng xã hội, cùng với form đăng ký nhận tin khuyến mãi qua email.

### 3.7.3. Thiết kế các trang chính

**Trang chủ** được thiết kế để thu hút ngay từ cái nhìn đầu tiên với băng chuyền hình ảnh quảng cáo các sản phẩm nổi bật hoặc chương trình khuyến mãi. Bên dưới là khu vực hiển thị sản phẩm mới nhất theo dạng lưới 4 cột, tiếp theo là phần sản phẩm bán chạy nhất và các danh mục nổi bật như Mô hình Anime, Art Toys, Trading Cards.

**Trang danh sách sản phẩm** tối ưu hóa trải nghiệm tìm kiếm và so sánh. Thanh lọc cho phép người dùng thu hẹp kết quả theo loại sản phẩm, thương hiệu và khoảng giá. Chức năng sắp xếp linh hoạt theo mới nhất, bán chạy, hoặc giá tăng giảm dần giúp người dùng nhanh chóng tìm thấy sản phẩm phù hợp. Mỗi thẻ sản phẩm hiển thị đầy đủ thông tin cần thiết gồm ảnh, tên, giá và điểm đánh giá. Phân trang được thiết kế rõ ràng để dễ dàng duyệt qua nhiều trang sản phẩm.

**Trang chi tiết sản phẩm** cung cấp đầy đủ thông tin để hỗ trợ quyết định mua hàng. Bộ sưu tập ảnh cho phép xem nhiều góc độ và có chức năng phóng to chi tiết. Thông tin sản phẩm được trình bày rõ ràng bao gồm giá, mô tả chi tiết, thông số kỹ thuật. Nút thêm vào giỏ hàng được thiết kế nổi bật với màu sắc tương phản. Khu vực đánh giá hiển thị điểm số trung bình và các bình luận từ người mua trước đó, tạo niềm tin cho khách hàng mới. Cuối trang gợi ý các sản phẩm liên quan để tăng cơ hội bán hàng.

**Trang giỏ hàng** hiển thị bảng danh sách sản phẩm đã chọn với đầy đủ thông tin ảnh, tên, số lượng có thể điều chỉnh, và giá từng mục. Khu vực tóm tắt bên phải tính toán chi tiết từ tạm tính, thuế VAT, phí vận chuyển đến tổng cộng cuối cùng. Ô nhập mã giảm giá được đặt ở vị trí dễ thấy. Nút thanh toán có màu nổi bật để khuyến khích người dùng hoàn tất giao dịch.

**Trang thanh toán** được chia thành nhiều bước rõ ràng gồm nhập địa chỉ giao hàng, chọn phương thức thanh toán, và xác nhận đơn hàng. Thanh tiến trình ở trên cùng giúp người dùng biết đang ở bước nào. Tóm tắt đơn hàng được cố định bên phải để người dùng luôn theo dõi được tổng tiền trong suốt quá trình.

**Trang quản trị** dành cho admin và staff sử dụng bố cục với menu bên trái chứa các mục Sản phẩm, Đơn hàng, Người dùng, Thống kê. Dashboard hiển thị các thẻ thống kê nhanh về doanh thu, số đơn hàng, số người dùng. Biểu đồ doanh thu theo thời gian giúp theo dõi xu hướng kinh doanh. Bảng đơn hàng mới nhất cho phép xử lý nhanh các đơn cần chú ý.

### 3.7.4. Màu sắc và kiểu chữ

Bảng màu được chọn lọc kỹ lưỡng để phù hợp với đối tượng teen. Màu chính là xanh dương #3B82F6 được sử dụng cho các nút bấm và liên kết quan trọng, tạo cảm giác hiện đại và tin cậy. Màu phụ xanh lá #10B981 dùng cho các trạng thái thành công như thêm vào giỏ hàng, thanh toán hoàn tất. Màu nhấn mạnh vàng cam #F59E0B được dùng tiết chế cho các điểm nhấn đặc biệt như khuyến mãi. Màu trung tính xám #6B7280 dùng cho văn bản thông thường. Nền trang sử dụng xám rất nhạt #F9FAFB để giảm độ chói mà vẫn sạch sẽ.

Về kiểu chữ, hệ thống sử dụng phông hệ thống mặc định để đảm bảo hiển thị nhanh và nhất quán trên mọi thiết bị. Tiêu đề sử dụng kiểu đậm với kích thước lớn từ 2xl đến 4xl tùy mức độ quan trọng. Nội dung chính dùng kiểu bình thường kích thước chuẩn để dễ đọc. Văn bản phụ và ghi chú sử dụng kích thước nhỏ hơn.

Biểu tượng được chọn từ bộ Lucide React, các icon nhẹ dạng SVG dễ tùy chỉnh màu sắc và kích thước. Hệ thống sử dụng ba kích thước chuẩn là 16px, 20px và 24px tùy theo vị trí sử dụng.

Ảnh có chứa văn bản, ảnh chụp màn hình, Quảng cáo trực tuyến

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.1 – Trang chủ

Ảnh có chứa văn bản, đồ chơi, ảnh chụp màn hình, phim hoạt hình

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.2 – Trang sản phẩm

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.3 – Trang đơn hàng, chi tiết đơn hàng

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.4 – Trang thanh toán

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.5 – Trang thanh toán thất bại, thành công (vnpay)

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.6 – Trang chủ admin

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.7 – Trang quản lý đơn hàng admin

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.

Hình 3.7.8 – Trang báo cáo admin

# Chương 4. CÀI ĐẶT VÀ CÔNG NGHỆ

## 4.1. Môi trường cài đặt

### 4.1.1. Yêu cầu hệ thống

**Máy chủ Backend:** - Hệ điều hành: Windows 10/11, Linux (Ubuntu 20.04+), macOS 10.15+ - Node.js: phiên bản 18.x trở lên

**Máy chủ Database:** - Microsoft SQL Server 2019 trở lên

**Máy trạm phát triển Frontend:** - Hệ điều hành: Windows 10/11, Linux, macOS - Node.js: phiên bản 18.x trở lên - Trình duyệt: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+

### 4.1.2. Công cụ phát triển

* **IDE/Code Editor**: Visual Studio Code
* **Version Control**: Git 2.30+
* **API Testing**: Postman
* **Database Management**: SQL Server Management Studio (SSMS

## 4.2. Công nghệ và framework sử dụng

### 4.2.1. Backend

**Runtime và Framework chính:**

* **Node.js 18.x**: Môi trường chạy JavaScript phía server, hiệu năng cao với kiến trúc event-driven non-blocking I/O, phù hợp cho ứng dụng real-time.
* **Express 4.21.2**: Web framework nhẹ, linh hoạt, cung cấp các tính năng mạnh mẽ để xây dựng RESTful API. Express có cộng đồng lớn và nhiều middleware hỗ trợ.

**ORM và Database:**

* **Sequelize 6.37.7**: ORM cho Node.js hỗ trợ nhiều DBMS (SQL Server, PostgreSQL, MySQL). Sequelize giúp tương tác với database thông qua JavaScript object thay vì viết SQL thuần, đồng thời cung cấp tính năng migration và validation.
* **mssql 10.0.4**: Driver kết nối SQL Server cho Node.js.

**Authentication và Security:**

* **jsonwebtoken 9.0.2**: Tạo và xác thực JWT token cho stateless authentication.
* **bcrypt/bcryptjs**: Mã hóa mật khẩu với thuật toán bcrypt (salt rounds = 10).
* **Passport.js**: Middleware authentication hỗ trợ nhiều strategy, sử dụng passport-google-oauth20 cho Google OAuth.
* **express-rate-limit 8.1.0**: Middleware giới hạn số lượng request để chống spam và brute force.

**File Processing:**

* **Multer 2.0.2**: Middleware xử lý multipart/form-data cho upload file.
* **Sharp 0.34.5**: Thư viện xử lý ảnh hiệu năng cao (resize, optimize, convert format).

**Real-time và Utilities:**

* **Socket.io 4.8.1**: Thư viện real-time bidirectional communication cho cập nhật giỏ hàng, thông báo đơn hàng.
* **Decimal.js 10.6.0**: Tính toán số thập phân chính xác, tránh lỗi floating-point khi tính tiền.
* **moment**: Xử lý ngày tháng và múi giờ.
* **node-cron 4.2.1**: Lập lịch chạy tác vụ định kỳ (cleanup sessions, update statistics).

### 4.2.2. Frontend

**Core Framework:**

* **React 18.2.0**: Thư viện JavaScript để xây dựng giao diện người dùng dựa trên component. React sử dụng Virtual DOM để tối ưu hiệu năng rendering.
* **React Router DOM 6.8.1**: Thư viện routing cho ứng dụng single-page application (SPA), hỗ trợ lazy loading và protected routes.

**HTTP Client và State Management:**

* **Axios 1.12.2**: HTTP client hỗ trợ Promise, interceptors để xử lý request/response globally (thêm token, handle errors).
* **React Context API**: Quản lý state toàn cục (AuthContext cho thông tin user, token).

**UI và Styling:**

* **TailwindCSS 3.4.18**: Utility-first CSS framework, giúp xây dựng giao diện nhanh với classes có sẵn, hỗ trợ responsive design và dark mode.
* **Lucide React 0.548.0**: Thư viện icons SVG nhẹ và đẹp.

**Charts và Visualization:**

* **Recharts 2.5.0**: Thư viện vẽ biểu đồ dựa trên React và D3.js, dùng cho dashboard thống kê.

**Other:**

* **Socket.io Client 4.8.1**: Client-side socket.io cho real-time updates.
* **Swiper 12.0.3**: Slider/carousel component hiện đại cho banner và gallery.

### 4.2.3. Database

**DBMS:**

* **Microsoft SQL Server 2019+**: Hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, hỗ trợ transaction ACID, stored procedures, triggers và full-text search.

**Tính năng sử dụng:** - Transaction để đảm bảo tính toàn vẹn dữ liệu - Foreign Keys để ràng buộc tham chiếu - Indexes để tối ưu query performance - Default values và computed columns

### 4.2.4. Tích hợp bên thứ ba

**VNPay Payment Gateway:** - API thanh toán trực tuyến phổ biến tại Việt Nam - Hỗ trợ đa dạng ngân hàng và ví điện tử - Bảo mật với HMAC SHA512

**Giao Hàng Nhanh (GHN) API:** - Tính phí vận chuyển tự động - Tạo đơn vận chuyển và lấy mã vận đơn - Webhook cập nhật trạng thái đơn hàng - Hỗ trợ mock mode để test

**Google OAuth 2.0:** - Đăng nhập nhanh bằng tài khoản Google - Tăng trải nghiệm người dùng

## 4.3. Lý do lựa chọn công nghệ

### 4.3.1. Lựa chọn Node.js và Express

**Ưu điểm:** - **Ngôn ngữ thống nhất**: Cả frontend và backend đều dùng JavaScript, giảm context switching cho developer. - **Hiệu năng cao**: Event-driven non-blocking I/O phù hợp cho ứng dụng có nhiều I/O operations (database queries, API calls). - **Cộng đồng lớn**: NPM có hàng triệu packages, dễ tìm giải pháp cho mọi vấn đề. - **Real-time**: Tích hợp tốt với Socket.io cho các tính năng real-time.

**Phù hợp với dự án:** - Hệ thống bán hàng cần xử lý nhiều request đồng thời (xem sản phẩm, đặt hàng). - Cần real-time updates (giỏ hàng, thông báo đơn hàng). - Team developer quen với JavaScript.

### 4.3.2. Lựa chọn React

**Ưu điểm:** - **Component-based**: Tái sử dụng code, dễ bảo trì. - **Virtual DOM**: Hiệu năng rendering tốt. - **Ecosystem phong phú**: Nhiều thư viện hỗ trợ (routing, state management, UI components). - **Declarative**: Code dễ đọc, dễ debug.

**Phù hợp với dự án:** - Giao diện người dùng phức tạp với nhiều trang (38 pages, 48 components). - Cần responsive design cho đối tượng khách hàng teen dùng nhiều thiết bị.

### 4.3.3. Lựa chọn SQL Server

**Ưu điểm:** - **ACID compliance**: Đảm bảo tính toàn vẹn dữ liệu trong transactions. - **Transaction hỗ trợ tốt**: Quan trọng cho nghiệp vụ đặt hàng (trừ tồn kho, tạo hóa đơn phải đồng bộ). - **Performance**: Index, query optimization tốt. - **Enterprise features**: Backup/restore, replication, high availability.

**Phù hợp với dự án:** - Nghiệp vụ yêu cầu transaction chặt chẽ (đặt hàng, thanh toán). - Dữ liệu có quan hệ phức tạp (21 bảng với nhiều foreign keys).

### 4.3.4. Lựa chọn TailwindCSS

**Ưu điểm:** - **Utility-first**: Xây dựng UI nhanh mà không cần viết CSS từ đầu. - **Customizable**: Dễ dàng tùy chỉnh theme, colors, spacing. - **Responsive**: Built-in responsive modifiers (sm:, md:, lg:). - **Production optimization**: PurgeCSS loại bỏ CSS không dùng.

**Phù hợp với dự án:** - Cần giao diện hiện đại, thân thiện với teen. - Yêu cầu responsive cho mobile, tablet, desktop.

## 4.4. Cấu trúc tổ chức code

### 4.4.1. Cấu trúc Backend

backend/  
├── config/  
│ └── database.js # Database connection config  
├── controllers/ # 21 controllers (business logic)  
│ ├── auth.controller.js  
│ ├── product.controller.js  
│ ├── order.controller.js  
│ └── ...  
├── decorators/ # Decorator Pattern  
│ ├── OrderPriceDecorator.js  
│ ├── VATDecorator.js  
│ └── VoucherDecorator.js  
├── models/ # 21 Sequelize models  
│ ├── index.js # Model associations  
│ ├── TaiKhoan.js  
│ ├── SanPham.js  
│ └── ...  
├── routes/ # 21 route files  
│ ├── auth.routes.js  
│ ├── product.routes.js  
│ └── ...  
├── middlewares/ # Middleware functions  
│ └── auth.middleware.js # JWT verification  
├── strategies/ # Strategy Pattern  
│ ├── ProductFilterStrategy.js  
│ ├── BestSellerStrategy.js  
│ └── ...  
├── states/ # State Pattern  
│ └── OrderState.js # 12 order states  
├── utils/ # Singleton utilities  
│ ├── Logger.js  
│ ├── ConfigService.js  
│ └── DBConnection.js  
├── uploads/ # Uploaded files  
├── .env # Environment variables  
├── server.js # Entry point  
└── package.json

**Nguyên tắc tổ chức:** - **Separation of Concerns**: Tách biệt routes, controllers, models. - **Single Responsibility**: Mỗi file chỉ làm một việc cụ thể. - **Design Patterns**: Đặt riêng thư mục cho decorators, strategies, states.

### 4.4.2. Cấu trúc Frontend

frontend/src/  
├── components/ # 48 components  
│ ├── common/ # UI common (Navbar, Footer)  
│ ├── product/ # Product components  
│ ├── order/ # Order components  
│ └── admin/ # Admin components  
├── pages/ # 38 pages  
│ ├── Homepage.js  
│ ├── ProductList.js  
│ ├── CheckoutPage.js  
│ ├── admin/ # Admin pages  
│ └── ...  
├── services/ # 15 API service modules  
│ ├── authService.js  
│ ├── productService.js  
│ ├── orderService.js  
│ └── ...  
├── contexts/ # React Context  
│ └── AuthContext.js  
├── layouts/ # Layout wrappers  
│ ├── MainLayout.js  
│ └── AdminLayout.js  
├── config.js # API base URL  
├── App.js # Router configuration  
└── index.js # Entry point

**Nguyên tắc tổ chức:** - **Component hierarchy**: Tổ chức components theo tính năng. - **Reusability**: Components tái sử dụng đặt trong common. - **Service layer**: Tách biệt logic gọi API vào services.

## 4.5. Các đoạn code minh họa tiêu biểu

### 4.5.1. Xác thực JWT (Middleware)

// backend/middlewares/auth.middleware.js  
const jwt = require('jsonwebtoken');  
const Logger = require('../utils/Logger');  
  
const verifyToken = (req, res, next) => {  
 try {  
 const token = req.headers.authorization?.split(' ')[1];  
   
 if (!token) {  
 return res.status(401).json({   
 message: 'Không tìm thấy token xác thực'   
 });  
 }  
   
 const decoded = jwt.verify(token, process.env.JWT\_SECRET);  
 req.user = decoded; // {id, username, role}  
 next();  
 } catch (error) {  
 Logger.getInstance().logError('JWT verification failed', error);  
 return res.status(401).json({   
 message: 'Token không hợp lệ hoặc đã hết hạn'   
 });  
 }  
};  
  
const requireAdmin = (req, res, next) => {  
 if (req.user.role !== 'admin') {  
 return res.status(403).json({   
 message: 'Không có quyền truy cập'   
 });  
 }  
 next();  
};  
  
module.exports = { verifyToken, requireAdmin };

### 4.5.2. Decorator Pattern - Tính giá đơn hàng

// backend/decorators/VATDecorator.js  
const Decimal = require('decimal.js');  
const OrderPriceDecorator = require('./OrderPriceDecorator');  
  
class VATDecorator extends OrderPriceDecorator {  
 constructor(calculator, vatRate = 0.10) {  
 super(calculator);  
 this.vatRate = new Decimal(vatRate);  
 }  
   
 calculate() {  
 const basePrice = this.calculator.calculate();  
 const vatAmount = basePrice.times(this.vatRate);  
 return basePrice.plus(vatAmount);  
 }  
   
 getDetails() {  
 const details = this.calculator.getDetails();  
 const basePrice = new Decimal(  
 details.thanhTien || details.tongTienSanPham  
 );  
 const vatAmount = basePrice.times(this.vatRate);  
   
 return {  
 ...details,  
 tyLeVAT: this.vatRate.toFixed(2),  
 tienVAT: vatAmount.toFixed(2),  
 thanhTien: this.calculate().toFixed(2)  
 };  
 }  
}  
  
module.exports = VATDecorator;

### 4.5.3. State Pattern - Quản lý trạng thái đơn hàng

// backend/states/OrderState.js (trích đoạn)  
class PendingState extends OrderState {  
 getName() {  
 return 'Chờ xử lý';  
 }  
   
 getAllowedTransitions() {  
 return ['Đã xác nhận', 'Đã hủy'];  
 }  
   
 async onEnter(previousState, transaction) {  
 await super.onEnter(previousState, transaction);  
 console.log('📧 Gửi email xác nhận đơn hàng');  
 // Logic gửi email cho khách hàng  
 }  
   
 canCustomerCancel() {  
 return true; // Khách có thể hủy ở trạng thái này  
 }  
}  
  
class OrderStateContext {  
 async transitionTo(newStateName, transaction, additionalData = {}) {  
 const currentStateName = this.currentState.getName();  
   
 if (!this.currentState.canTransitionTo(newStateName)) {  
 throw new Error(  
 `Không thể chuyển từ "${currentStateName}" sang "${newStateName}"`  
 );  
 }  
   
 const newState = this.getStateInstance(newStateName);  
   
 // Lifecycle hooks  
 await this.currentState.onExit(newStateName, transaction);  
   
 // Update database  
 this.order.TrangThai = newStateName;  
 await this.order.save({ transaction });  
   
 await newState.onEnter(currentStateName, transaction);  
   
 this.currentState = newState;  
 return this.order;  
 }  
}

### 4.5.4. Singleton Pattern - Logger

// backend/utils/Logger.js  
class Logger {  
 static instance = null;  
   
 constructor() {  
 if (Logger.instance) {  
 return Logger.instance;  
 }  
 Logger.instance = this;  
 }  
   
 static getInstance() {  
 if (!Logger.instance) {  
 Logger.instance = new Logger();  
 }  
 return Logger.instance;  
 }  
   
 info(message, data = {}) {  
 console.log(`[INFO] ${new Date().toISOString()} - ${message}`, data);  
 }  
   
 logError(message, error) {  
 console.error(`[ERROR] ${new Date().toISOString()} - ${message}`, {  
 message: error.message,  
 stack: error.stack  
 });  
 }  
}  
  
module.exports = Logger;

### 4.5.5. React Service - API Call

// frontend/src/services/orderService.js  
import axios from 'axios';  
import config from '../config';  
  
const API\_URL = `${config.API\_BASE\_URL}/orders`;  
  
const orderService = {  
 async createOrder(orderData) {  
 const token = localStorage.getItem('token');  
 const response = await axios.post(`${API\_URL}/create`, orderData, {  
 headers: {  
 'Authorization': `Bearer ${token}`,  
 'Content-Type': 'application/json'  
 }  
 });  
 return response.data;  
 },  
   
 async getMyOrders(page = 1, limit = 10) {  
 const token = localStorage.getItem('token');  
 const response = await axios.get(`${API\_URL}/my-orders`, {  
 params: { page, limit },  
 headers: { 'Authorization': `Bearer ${token}` }  
 });  
 return response.data;  
 }  
};  
  
export default orderService;

# Chương 5. ĐÁNH GIÁ HỆ THỐNG

## 5.1. Đánh giá mức độ đáp ứng yêu cầu chức năng

### 5.1.1. Nhóm chức năng đã hoàn thành đầy đủ

Hệ thống đã thực hiện được tất cả 10 nhóm yêu cầu chức năng đã đề ra trong Chương 2:

**Xác thực và quản lý tài khoản (100%):** - Đăng ký, đăng nhập với JWT token (expire 7 ngày) - Đăng nhập Google OAuth hoạt động ổn định - Phân quyền 3 vai trò (admin, staff, user) chính xác - Rate limiting chống brute force hiệu quả

**Quản lý sản phẩm (100%):** - CRUD sản phẩm với upload nhiều ảnh - Tìm kiếm, lọc theo loại, thương hiệu, giá - Sắp xếp linh hoạt với Strategy Pattern (mới nhất, bán chạy, giá) - Quản lý danh mục và thương hiệu đầy đủ

**Giỏ hàng (100%):** - Hỗ trợ cả user đăng nhập và khách vãng lai - Thao tác thêm, sửa, xóa hoạt động tốt - Validation tồn kho chính xác

**Đặt hàng và thanh toán (100%):** - Tích hợp GHN cho địa chỉ và phí ship thành công - Decorator Pattern tính giá chính xác (VAT, voucher, shipping) - VNPay integration hoạt động tốt (sandbox mode) - COD hỗ trợ đầy đủ

**Quản lý đơn hàng (100%):** - State Pattern quản lý 12 trạng thái rõ ràng - Admin có thể xác nhận, tạo đơn GHN, cập nhật trạng thái - User có thể xem lịch sử, hủy đơn khi ở trạng thái cho phép - Tra cứu đơn hàng công khai cho khách vãng lai

**Vận chuyển GHN (100%):** - Lấy địa chỉ (tỉnh/quận/phường) chính xác - Tính phí ship tự động - Tạo đơn vận chuyển và nhận mã vận đơn - Mock mode hỗ trợ test

**Voucher (100%):** - CRUD voucher với điều kiện (giá trị tối thiểu, số lượng, thời gian) - Validate và áp dụng voucher đúng logic - Lưu lịch sử sử dụng

**Đánh giá sản phẩm (100%):** - User đã mua có thể đánh giá - Tính điểm trung bình tự động - Admin duyệt/ẩn đánh giá

**Thống kê (100%):** - Dashboard tổng quan với các chỉ số quan trọng - Biểu đồ doanh thu theo thời gian - Top sản phẩm bán chạy, top khách hàng - Thống kê phương thức thanh toán

**Banner và quản lý người dùng (100%):** - CRUD banner hoạt động ổn - Quản lý users, khóa/mở khóa tài khoản - Quản lý nhân viên và phân quyền

### 5.1.2. Điểm nổi bật trong cài đặt

**Áp dụng Design Patterns hiệu quả:** - Decorator Pattern giúp tính giá đơn hàng linh hoạt, dễ thêm các thành phần mới (discount, tax) mà không sửa code cũ - State Pattern quản lý vòng đời đơn hàng rõ ràng, mỗi trạng thái có quy tắc chuyển đổi riêng, tránh logic rối - Strategy Pattern cho phép thay đổi thuật toán sắp xếp/lọc sản phẩm dễ dàng - Singleton Pattern đảm bảo Logger, ConfigService, DBConnection có duy nhất một instance

**Transaction đảm bảo toàn vẹn dữ liệu:** - Khi tạo đơn hàng, các thao tác (tạo hóa đơn, trừ tồn kho, xóa giỏ hàng, lưu voucher) được thực hiện trong một transaction - Nếu một bước lỗi, toàn bộ rollback, không tồn tại tình trạng dữ liệu không nhất quán

**Tích hợp bên thứ ba ổn định:** - VNPay: xử lý return URL và IPN đầy đủ - GHN: tạo đơn vận chuyển, webhook cập nhật trạng thái tự động - Google OAuth: đăng nhập nhanh, lưu GoogleID

## 5.2. Đánh giá hiệu năng và khả năng mở rộng

### 5.2.1. Hiệu năng thực tế

**Thời gian tải trang:** - Trang chủ: ~2 giây (đạt yêu cầu < 3s) - Danh sách sản phẩm: ~2.5 giây với phân trang - Chi tiết sản phẩm: ~1.8 giây (đạt yêu cầu < 2s)

**Thời gian phản hồi API:** - GET /api/products: ~500ms cho 20 sản phẩm (đạt yêu cầu < 1s) - POST /api/orders/create: ~2 giây bao gồm transaction (đạt yêu cầu < 3s) - API tìm kiếm: ~700ms (đạt yêu cầu)

**Khả năng chịu tải:** - Hệ thống đã test với 50 concurrent users, xử lý ổn định - Database với 1000 sản phẩm, 500 đơn hàng hoạt động tốt - Chưa test với 100+ concurrent users do giới hạn môi trường dev

### 5.2.2. Tối ưu đã thực hiện

**Frontend:** - Lazy loading components với React.lazy() giảm initial bundle size - Code splitting theo routes - Phân trang cho danh sách sản phẩm, đơn hàng - TailwindCSS PurgeCSS loại bỏ CSS không dùng

**Backend:** - Sequelize ORM tự động optimize queries - Eager loading cho associations giảm N+1 queries - Rate limiting chống spam, bảo vệ server

**Database:** - Indexes trên các cột thường query (TenDangNhap, Email, MaHD) - Foreign keys đảm bảo referential integrity

### 5.2.3. Khả năng mở rộng

**Kiến trúc ba tầng:** - Frontend và Backend tách biệt hoàn toàn, dễ scale độc lập - Có thể deploy frontend lên CDN (Vercel, Netlify) - Backend có thể scale horizontal với load balancer

**Database:** - Schema chuẩn hóa 3NF dễ thêm bảng mới - Sequelize migrations hỗ trợ version control schema

**Code organization:** - Design patterns giúp thêm tính năng mới dễ dàng - Modular structure dễ bảo trì

## 5.3. Đánh giá bảo mật

### 5.3.1. Cơ chế bảo mật đã triển khai

**Authentication:** - Mật khẩu được hash bằng bcrypt (salt rounds = 10), không lưu plaintext - JWT token với secret key từ .env, expire 7 ngày - Google OAuth với Client ID/Secret bảo mật

**Authorization:** - Middleware verifyToken kiểm tra token trước mọi protected route - requireAdmin, requireStaff đảm bảo phân quyền chính xác - Frontend cũng check role để ẩn UI không được phép

**Rate Limiting:** - Login: 5 lần/15 phút (chống brute force) - Register: 3 lần/giờ (chống spam) - Cart, Order operations có rate limit riêng

**Input Validation:** - Sequelize ORM tự động escape SQL, chống SQL Injection - React auto-escape output, chống XSS - File upload: whitelist extensions (jpg, png, gif, webp), max size 10MB

**Data Protection:** - Kết nối VNPay dùng HTTPS và HMAC SHA512 - Thông tin thanh toán không lưu trữ, chỉ lưu mã giao dịch - CORS cấu hình chỉ cho phép origin từ frontend domain

### 5.3.2. Điểm mạnh về bảo mật

* **Stateless authentication**: JWT token giúp backend không cần lưu session, dễ scale
* **Phân quyền rõ ràng**: 3 vai trò với quyền hạn khác nhau, check ở cả frontend và backend
* **Rate limiting**: Bảo vệ tốt khỏi brute force, spam, DDoS
* **Environment variables**: Tất cả thông tin nhạy cảm (DB password, JWT secret, API keys) trong .env, không commit lên Git

### 5.3.3. Hạn chế về bảo mật

* **Chưa có refresh token**: JWT expire 7 ngày hơi dài, nếu bị đánh cắp token có thể bị lợi dụng lâu. Nên implement refresh token để shorten access token lifetime
* **Chưa có 2FA**: Tài khoản admin nên có xác thực 2 yếu tố để bảo mật cao hơn
* **Logging chưa chi tiết**: Chưa log đầy đủ các hành vi đáng ngờ (failed login attempts, privilege escalation)
* **File upload validation**: Chỉ check extension, chưa check MIME type và scan virus

## 5.4. Đánh giá giao diện và trải nghiệm người dùng

### 5.4.1. Đánh giá UI (User Interface)

**Điểm mạnh:** - **Responsive design**: Giao diện tự động điều chỉnh đẹp trên desktop, tablet, mobile nhờ TailwindCSS - **Màu sắc hiện đại**: Phù hợp với đối tượng teen, sử dụng màu sắc tươi sáng, gradient - **Icons rõ ràng**: Lucide React icons dễ nhận biết - **Consistency**: Các components tái sử dụng đảm bảo UI nhất quán

**Điểm cần cải thiện:** - **Typography**: Chưa sử dụng custom font từ Google Fonts, vẫn dùng system font - **Animations**: Thiếu micro-animations để tăng tính sống động - **Loading states**: Một số trang chưa có skeleton loading, chỉ có spinner đơn giản

### 5.4.2. Đánh giá UX (User Experience)

**Luồng mua hàng:** - **Thuận tiện**: Khách vãng lai có thể mua hàng mà không cần đăng ký - **Rõ ràng**: Các bước checkout hiển thị progress, dễ theo dõi - **Tính giá minh bạch**: Breakdown giá (tạm tính, VAT, voucher, ship) rõ ràng - **Tra cứu đơn hàng**: Khách vãng lai tra cứu đơn bằng mã dễ dàng

**Tìm kiếm và lọc:** - **Tìm kiếm nhanh**: Search bar ở trên navbar, dễ tiếp cận - **Lọc linh hoạt**: Theo loại, thương hiệu, giá - **Sắp xếp đa dạng**: Mới nhất, bán chạy, giá tăng/giảm

**Thông báo:** - **Toast notification**: Hiển thị feedback khi thêm giỏ hàng, đặt hàng thành công/lỗi - **Validation messages**: Lỗi form hiển thị rõ ràng

**Điểm cần cải thiện:** - **Breadcrumb**: Chưa có breadcrumb navigation để người dùng biết đang ở đâu - **Số lượng tồn kho**: Chưa hiển thị “Còn X sản phẩm” để tạo urgency - **Wishlist**: Chưa có tính năng yêu thích/lưu sản phẩm - **Product comparison**: Chưa có so sánh sản phẩm

### 5.4.3. Đánh giá phù hợp với đối tượng người dùng

Hệ thống được thiết kế cho khách hàng teen và giới trẻ yêu thích đồ chơi sưu tầm:

**Phù hợp:** - Giao diện hiện đại, không cứng nhắc - Hình ảnh sản phẩm lớn, rõ nét - Đánh giá sản phẩm giúp teen tham khảo ý kiến người khác - Mobile responsive quan trọng vì teen dùng điện thoại nhiều

**Có thể cải thiện:** - Thêm tính năng “Sản phẩm liên quan” để gợi ý - Blog/News về đồ chơi mới, tin tức anime, gaming - Social features: share sản phẩm lên Facebook, Instagram

## 5.5. Hạn chế và điểm cần cải thiện

### 5.5.1. Hạn chế về chức năng

**Chưa có tính năng nâng cao:** - **Recommendation system**: Chưa có gợi ý sản phẩm dựa trên lịch sử mua hàng, sở thích - **Wishlist**: Chưa có danh sách yêu thích - **Product comparison**: Chưa so sánh sản phẩm - **Chatbot**: Chưa có hỗ trợ tự động qua chatbot - **Loyalty program**: Chưa có tích điểm thành viên

**Quản lý tồn kho:** - Chưa có cảnh báo tồn kho thấp cho admin - Chưa có báo cáo nhập/xuất kho chi tiết - Chưa có dự đoán nhu cầu nhập hàng

**Báo cáo thống kê:** - Dashboard thống kê còn đơn giản - Chưa có export báo cáo ra Excel/PDF - Chưa có phân tích xu hướng tiêu dùng theo mùa

### 5.5.2. Hạn chế về kỹ thuật

**Performance:** - Frontend bundle size còn lớn (~2MB), chưa optimize tốt nhất - Backend chưa implement caching (Redis) cho queries thường xuyên - Database chưa có stored procedures cho logic phức tạp

**Testing:** - Chưa có unit tests cho backend controllers, services - Chưa có integration tests cho API - Frontend chưa có tests cho components

**Deployment:** - Chưa có CI/CD pipeline tự động - Chưa có monitoring, logging tập trung (ELK stack) - Chưa có alerting khi có lỗi nghiêm trọng

**Scalability:** - Backend chưa stateless hoàn toàn (file upload lưu local) - Chưa có load balancer cho multiple instances - Database chưa có replication, backup tự động

### 5.5.3. Hạn chế về bảo mật

* Chưa có refresh token
* Chưa có 2FA
* Logging chưa chi tiết
* File upload validation chưa đầy đủ

### 5.5.4. Kế hoạch khắc phục

**Ngắn hạn:** - Thêm wishlist, product comparison - Implement caching với Redis - Viết unit tests cho critical paths - Thêm stored procedures cho queries phức tạp - Improve file upload validation

**Trung hạn:** - Xây dựng recommendation system đơn giản (collaborative filtering) - Implement refresh token - Thêm 2FA cho admin - Setup CI/CD pipeline - Centralized logging với ELK stack

**Dài hạn:** - Chatbot AI hỗ trợ khách hàng - Loyalty program với tích điểm - Mobile app (React Native) - Analytics và dự đoán xu hướng - Microservices architecture nếu scale lớn

# Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 6.1. Tóm tắt kết quả đạt được

### 6.1.1. Về mặt chức năng

Đồ án đã xây dựng thành công hệ thống quản lý bán hàng trực tuyến ToyStore cho cửa hàng đồ chơi sưu tầm phục vụ đối tượng teen và người trẻ. Hệ thống đáp ứng đầy đủ 10 nhóm yêu cầu chức năng đã đề ra:

* **Xác thực và phân quyền**: Hỗ trợ đăng ký, đăng nhập local và Google OAuth với 3 vai trò (user, staff, admin), bảo mật bằng JWT và bcrypt.
* **Quản lý sản phẩm đa dạng**: CRUD sản phẩm với upload nhiều ảnh, phân loại theo danh mục và thương hiệu phù hợp với các dòng đồ chơi như mô hình anime, art toys, EDC, LEGO, trading cards.
* **Giỏ hàng linh hoạt**: Hỗ trợ cả user đã đăng nhập và khách vãng lai, lưu trữ bền vững và kiểm tra tồn kho tự động.
* **Quy trình đặt hàng hoàn chỉnh**: Tích hợp GHN cho địa chỉ và phí vận chuyển, áp dụng voucher giảm giá, tính giá chính xác với Decorator Pattern (VAT, voucher, shipping).
* **Thanh toán đa dạng**: Hỗ trợ COD và VNPay với xử lý return URL và IPN đầy đủ.
* **Quản lý đơn hàng thông minh**: Sử dụng State Pattern để quản lý 12 trạng thái đơn hàng với quy tắc chuyển đổi rõ ràng, tích hợp GHN để tạo đơn vận chuyển và theo dõi trạng thái.
* **Đánh giá sản phẩm**: User có thể đánh giá sản phẩm đã mua, tính điểm trung bình tự động, admin duyệt đánh giá.
* **Thống kê và báo cáo**: Dashboard cho admin với biểu đồ doanh thu, top sản phẩm bán chạy, phân tích phương thức thanh toán.

### 6.1.2. Về mặt kỹ thuật

**Kiến trúc hệ thống:** - Áp dụng kiến trúc ba tầng (Presentation - Business Logic - Data) tách biệt rõ ràng, dễ bảo trì và mở rộng. - Frontend React và Backend Node.js/Express giao tiếp qua RESTful API, có thể deploy độc lập. - Database SQL Server với 21 bảng chuẩn hóa 3NF đảm bảo tính toàn vẹn dữ liệu.

**Design Patterns:** - Decorator Pattern: Tính giá đơn hàng linh hoạt, dễ thêm các thành phần mới. - Strategy Pattern: Sắp xếp/lọc sản phẩm theo nhiều tiêu chí. - State Pattern: Quản lý vòng đời đơn hàng với 12 trạng thái. - Singleton Pattern: Logger, ConfigService, DBConnection đảm bảo single instance.

**Bảo mật:** - Mật khẩu hash bằng bcrypt, JWT authentication, rate limiting chống brute force. - Sequelize ORM chống SQL Injection, React auto-escape chống XSS. - CORS, HTTPS cho VNPay, environment variables cho thông tin nhạy cảm.

**Performance:** - Lazy loading và code splitting trên frontend giảm initial load time. - Database indexes, eager loading giảm N+1 queries. - Thời gian tải trang và API response đều đạt yêu cầu.

### 6.1.3. Về mặt tích hợp

Hệ thống tích hợp thành công với 3 dịch vụ bên thứ ba: - **VNPay**: Thanh toán trực tuyến với sandbox mode, xử lý return và IPN. - **Giao Hàng Nhanh (GHN)**: Lấy địa chỉ, tính phí ship, tạo đơn vận chuyển, webhook cập nhật trạng thái. - **Google OAuth**: Đăng nhập nhanh, tăng trải nghiệm người dùng.

## 6.2. Đóng góp của đề tài

### 6.2.1. Đóng góp về mặt thực tiễn

**Giải quyết bài toán thực tế:** - Hệ thống giải quyết được các vấn đề của cửa hàng đồ chơi sưu tầm: quản lý sản phẩm đa dạng, xử lý đơn hàng hiệu quả, theo dõi tồn kho chính xác. - Tự động hóa quy trình từ đặt hàng đến giao hàng, giảm thời gian xử lý và sai sót. - Cung cấp trải nghiệm mua sắm tốt cho khách hàng teen với giao diện thân thiện, responsive.

**Tạo nền tảng cho mở rộng:** - Kiến trúc modular và áp dụng design patterns giúp dễ dàng thêm tính năng mới trong tương lai. - Database schema chuẩn hóa cho phép thêm mô hình sản phẩm mới, phương thức thanh toán mới. - Tích hợp API bên thứ ba mở ra khả năng kết nối với nhiều dịch vụ khác.

### 6.2.2. Đóng góp về mặt kỹ thuật

**Áp dụng Design Patterns trong thực tế:** - Đồ án minh chứng cách áp dụng 4 design patterns (Decorator, Strategy, State, Singleton) vào bài toán thực tế, không chỉ lý thuyết. - Decorator Pattern cho thấy cách tính giá đơn hàng linh hoạt mà không vi phạm Open/Closed Principle. - State Pattern quản lý vòng đời đơn hàng phức tạp một cách rõ ràng, dễ bảo trì.

**Transaction và Data Integrity:** - Minh họa cách sử dụng database transaction để đảm bảo tính toàn vẹn dữ liệu khi tạo đơn hàng (tạo hóa đơn, trừ tồn kho, xóa giỏ hàng phải thành công đồng thời hoặc rollback toàn bộ).

**Full-stack Development:** - Đồ án là một ví dụ hoàn chỉnh về ứng dụng full-stack với React, Node.js, Express, SQL Server. - Kết hợp cả frontend, backend, database, và tích hợp bên thứ ba.

### 6.2.3. Đóng góp về mặt học thuật

Đồ án cung cấp tài liệu tham khảo cho sinh viên về: - Cách phân tích yêu cầu và thiết kế hệ thống thương mại điện tử. - Áp dụng design patterns trong dự án thực tế. - Tích hợp API bên thứ ba (payment gateway, shipping service, OAuth). - Xây dựng hệ thống bảo mật với JWT, bcrypt, rate limiting.

## 6.3. Đánh giá mức độ hoàn thành

### 6.3.1. So với mục tiêu ban đầu

**Mục tiêu đã đạt (100%):** - Xây dựng hệ thống quản lý bán hàng đầy đủ chức năng. - Áp dụng kiến trúc ba tầng và design patterns. - Tích hợp VNPay, GHN, Google OAuth thành công. - Đảm bảo bảo mật với JWT, bcrypt, rate limiting. - Giao diện responsive phù hợp với đối tượng teen. - Phân quyền rõ ràng cho 3 vai trò.

**Vượt mục tiêu:** - Hỗ trợ giỏ hàng cho khách vãng lai (không yêu cầu ban đầu). - State Pattern quản lý 12 trạng thái đơn hàng (chi tiết hơn kế hoạch). - Real-time updates với Socket.io.

### 6.3.2. Hạn chế chưa hoàn thành

Một số tính năng nâng cao chưa triển khai do giới hạn thời gian: - Recommendation system (gợi ý sản phẩm thông minh) - Chatbot AI hỗ trợ khách hàng - Loyalty program (tích điểm thành viên) - Unit tests và integration tests - CI/CD pipeline tự động

## 6.4. Hướng phát triển

### 6.4.1. Ngắn hạn

**Cải thiện chức năng hiện tại:** - Thêm wishlist cho user lưu sản phẩm yêu thích - Product comparison để so sánh nhiều sản phẩm - Breadcrumb navigation để người dùng biết vị trí hiện tại - Hiển thị số lượng tồn kho trên trang sản phẩm

**Tối ưu hiệu năng:** - Implement Redis caching cho queries thường xuyên (danh sách sản phẩm, categories) - Optimize frontend bundle size xuống dưới 1MB - Thêm stored procedures cho logic tính toán phức tạp

**Bảo mật:** - Implement refresh token để shorten access token lifetime - Improve file upload validation (check MIME type, scan virus) - Thêm detailed logging cho security events

### 6.4.2. Trung hạn

**Tính năng nâng cao:** - **Recommendation System**: Sử dụng collaborative filtering để gợi ý sản phẩm dựa trên lịch sử mua hàng và hành vi người dùng tương tự. - **Email Marketing**: Gửi email thông báo khuyến mãi, newsletter về sản phẩm mới, tin tức anime/gaming. - **Advanced Analytics**: Phân tích xu hướng tiêu dùng theo mùa, dự đoán nhu cầu nhập hàng.

**DevOps:** - Setup CI/CD pipeline với GitHub Actions hoặc GitLab CI - Implement centralized logging với ELK stack (Elasticsearch, Logstash, Kibana) - Monitoring với Prometheus và Grafana - Automated backup và disaster recovery

**Testing:** - Viết unit tests cho critical paths (authentication, order creation, payment) - Integration tests cho API endpoints - End-to-end tests cho user flows (đăng ký, mua hàng, thanh toán)

### 6.4.3. Dài hạn

**Mở rộng nền tảng:** - **Mobile App**: Phát triển ứng dụng mobile bằng React Native để tiếp cận khách hàng teen trên điện thoại tốt hơn. - **Loyalty Program**: Hệ thống tích điểm thành viên với các tier (Bronze, Silver, Gold) và ưu đãi riêng. - **Chatbot AI**: Sử dụng GPT-4 hoặc Claude để xây dựng chatbot tư vấn sản phẩm, trả lời câu hỏi thường gặp.

**Scale hệ thống:** - Chuyển sang microservices architecture nếu traffic tăng cao (Service riêng cho Products, Orders, Users, Payments). - Deploy lên cloud (AWS, Azure, Google Cloud) với auto-scaling. - CDN cho static assets và images. - Database replication và sharding nếu data lớn.

**Tính năng cộng đồng:** - Forum/Community cho người sưu tầm chia sẻ, trao đổi. - Marketplace cho user bán lại đồ chơi sưu tầm cũ (second-hand market). - Social features: follow users, share collections, like/comment sản phẩm.

## 6.5. Bài học kinh nghiệm

### 6.5.1. Về kỹ thuật

* **Design Patterns quan trọng**: Áp dụng đúng design pattern giúp code dễ bảo trì và mở rộng. State Pattern giúp quản lý đơn hàng rõ ràng hơn nhiều so với if-else rối.
* **Transaction là must-have**: Với nghiệp vụ tài chính, transaction là bắt buộc để đảm bảo data integrity.
* **Testing sớm**: Nên viết tests từ đầu, không để sau mới viết sẽ tốn thời gian hơn.

### 6.5.2. Về quy trình

* **Phân tích kỹ yêu cầu**: Hiểu rõ nghiệp vụ từ đầu giúp thiết kế database và API tốt hơn, tránh refactor nhiều.
* **Tích hợp bên thứ ba cần mock**: Nên có mock mode cho các API bên thứ ba (VNPay, GHN) để test dễ dàng.
* **Documentation**: Viết README, API docs từ đầu giúp onboarding và debug nhanh hơn.

### 6.5.3. Về quản lý dự án

* **Chia nhỏ tasks**: Chia dự án thành các task nhỏ (theo feature) giúp theo dõi tiến độ dễ dàng.
* **Version control tốt**: Commit thường xuyên với message rõ ràng giúp rollback khi cần.
* **Deploy sớm**: Deploy lên staging environment sớm để phát hiện vấn đề production.

# TÀI LIỆU THAM KHẢO

[1] React Team. (2024). *React - A JavaScript library for building user interfaces*.

[2] OpenJS Foundation. (2024). *Node.js - JavaScript runtime built on Chrome’s V8*.

[3 ] Express Team. (2024). *Express - Fast, unopinionated, minimalist web framework for Node.js*.

[4] Sequelize Team. (2024). *Sequelize - A promise-based Node.js ORM for SQL databases*.

[5] Tailwind Labs. (2024). *Tailwind CSS - A utility-first CSS framework*.

[6] Auth0. (2024). *Introduction to JSON Web Tokens*.

[7] npm. (2024). *bcrypt - A library to help you hash passwords*.

[8] VNPAY. (2024). *Tài liệu tích hợp cổng thanh toán VNPay*.

[9] GHN. (2024). *GHN API Documentation for Developers*.

[10] Osmani, A. (2023). *Learning JavaScript Design Patterns*.