TRƯỜNG ĐẠI HỌC CẦN THƠ

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

****

**NIÊN LUẬN CƠ SỞ**

**MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU**

**ĐỀ TÀI**

**PHÁT TRIỂN ỨNG DỤNG WEB BÁN LAP TOP**

**Sinh viên thực hiện: Ngô Gia Hạo**

**MSSV: B2110964**

**Khóa: 47**

***Cần Thơ, 4/2025***

# **LỜI CẢM ƠN**

Cần Thơ, ngày ... tháng ... năm 20...

Sinh viên thực hiện

# 

# **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

**..............................................................................................................................**

Cần Thơ, ngày ... tháng ... năm 20...

Giảng viên hướng dẫn

Mục lục

[LỜI CẢM ƠN 2](#__RefHeading___Toc289_853921268)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 3](#__RefHeading___Toc291_853921268)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 6](#__RefHeading___Toc1255_853921268)

[1. ĐẶT VẤN ĐỀ 6](#__RefHeading___Toc299_853921268)

[2. MỤC TIÊU CỦA ĐỀ TÀI 7](#__RefHeading___Toc301_853921268)

[3. PHẠM VI CỦA ĐỀ TÀI 8](#__RefHeading___Toc303_853921268)

[4. CÁC CÔNG NGHÊ SỬ DỤNG 8](#__RefHeading___Toc959_2901295804)

[5. BỐ CỤC NIÊN LUẬN 11](#__RefHeading___Toc311_853921268)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 12](#__RefHeading___Toc315_853921268)

[1. PHÂN TÍCH YÊU CẦU 12](#__RefHeading___Toc961_2901295804)

[2. THIẾT KẾ KIẾN TRÚC HỆ THỐNG. 16](#__RefHeading___Toc963_2901295804)

**TÓM TẮT**

Đề tài: Xây dựng một ứng dụng web thương mại điện tử chuyên bán laptop, cung cấp trải nghiệm mua sắm thuận tiện, tích hợp tình năng báo mật (JWT) và có tính mở rộng cao.

# 

# 

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

## ĐẶT VẤN ĐỀ

Trong thời đại công nghệ 4.0, nhu cầu sử dụng laptop và các thiết bị công nghệ ngày càng cao, đặc biệt sau giai đoạn đại dịch COVID-19 khi xu hướng làm việc từ xa và học trực tuyến ngày càng trở nên phổ biến**.** Tuy nhiên việc mua sắm các sản phẩm công nghệ, vẫn gặp nhiều khó khăn.

Dưới đây là các vấn đề:

+ **Hạn chế của các cửa hàng truyền thống**: Các của hàng bán lẻ laptop và thiết bị công nghệ truyền thống có giờ mở cửa cố định trong ngày, diện tích trưng bày không đủ để hiển thị đầy đủ sản phẩm và thông số kỹ thuật chi tiết. **Cuối cùng** là không thể phục vụ số lượng lớn khách hàng khi có nhu cầu mua sắm ở một thời điểm tăng cao, dẫn đến mất doanh thu và khách hàng sẽ chuyển qua các cữa hàng khác với khả năng phục vụ tốt hơn.

+ **Thiếu thông tin so sánh**: Khách hàng khó so sánh thông số kỹ thuật, giá cả giữa các dòng laptop và thiết bị công nghệ khác nhau một cách trực quan và hiệu quả.

+ **Quy trình mua hàng phức tạp:**  Việc tìm hiểu, tư vấn và mua sắm các sản phẩm công nghệ thường mất nhiều thời gian, đặc biệt khi cần tư vấn chuyên sâu về cấu hình phù hợp.

+ **Nhu cầu mua sắm online tăng cao:** Sau đại dịch COVID-19 xu hướng mua sắm các sản phẩm online tăng cao nhưng nhiều doanh nghiệp vừa và nhỏ chưa có nền tảng bán hàng online chuyên nghiệp. Việc xây dựng nhanh một ứng dụng bán hàng online với chi phí thấp và có tính mở rộng cao nếu tình hình kinh doanh thuận lợi và ứng dụng cần thêm, cải thiện dịch vụ là một bước đi sáng suốt.

## **MỤC TIÊU CỦA ĐỀ TÀI**

**2.1 Mục tiêu chính**

Xây dựng một sản phẩm thực tế web thương mại điện tử với đầy đủ các chức năng cơ bản của một ứng dụng web hiện đại, bao gồm:

* + Hệ thống xác thực và phân quyền với JWT tokens.
  + Quản lý sản phẩm và danh mục với khả năng CRUD đầy đủ.
  + Hệ thống giỏ hàng cho phép thêm, sửa, xóa sản phẩm.
  + Quy trình đặt hàng từ giỏ hàng đến thanh toán.
  + Tích hợp thanh toán với cổng thanh toán SePay QR Code.
  + Admin panel để quản lý toàn bộ hệ thống.
  + Dashboard thống kê với các chỉ số quan trọng.

**2.2 Mục tiêu kỹ thuật**

* + **Backend Development:**
  + Thiết kế và triển khai API với chuẩn RESTful sử dụng NestJS.
  + Áp dụng kiến trúc modular.
  + Implement authentication và authorization với JWT strategy
  + Thiết kế database với PostgreSQL và TypeORM.
  + Sử dụng Validation Pipe cho Data Transfer Object (DTO) và error handling.
    - **Frontend Development:**
  + Xây dựng Single Page Application (SPA) với React và TypeScript.
  + Implement responsive design với TailwindCSS và ShadcnUI components.
  + Quản lý state với Zustand theo pattern phân chia theo domain.
  + Tích hợp API với error handling và loading states.

**2.3 Mục tiêu học tập**

* + - Thực hành nguyên lý thiết kế phần mềm OOP, SOLID.
    - Hiểu về full-stask development workflow.
    - Thực hành database design.
    - Học cách tích hợp các dịch vụ bên thứ ba (payment gateway, email service).
    - Học cách thiết kế kiến trúc hệ thống dễ dàng maintain và mở rộng.
    - Cũng cố kiến thức về NestJS Framework, ReactJS.
    - Phát triển kỹ năng debugging.

## PHẠM VI CỦA ĐỀ TÀI

Nhằm đảm bảo có một sản phẩm có thể áp dụng vào thực tế, dưới đây là một số tính năng cốt lõi sẽ triển khai mà một ứng dụng bán hàng cần có

**Những chức năng sẽ triển khai**

* + - **Dành cho người dùng:**
  + Đăng ký đăng nhập với email và mật khẩu.
  + Quản lý thông tin cá nhân (profile, địa chỉ, số diện thoại).
  + Duyệt và tím kiếm sản phẩm với bộ lọc.
  + Viết đánh giá sản phẩm sau khi mua hàng.
  + Xem đánh giá sản phẩm.
  + Xem chi tiết sản phẩm với hình ảnh và thông tin đây đủ.
  + Thêm sản phẩm vào giỏ hàng và quản lý giỏ hàng.
  + Đặt hàng với thông tin giao hàng.
  + Thanh toán qua Sepay QR Code hoặc thanh toán khi nhận hàng (COD).
  + Theo dõi trạng thái đơn hàng và lịch sử các đơn hàng.
    - **Dành cho quản trị viên:**
  + Dashboard thống kê tổng quan (tổng người dùng, đơn hàng, doanh thu, sản phẩm).
  + Quản lý người dùng (view, tìm kiếm người dùng theo tên hoặc email, khóa/mở tài khoảng).
  + Quản lý sản phẩm (CRUD, tìm kiếm sản phẩm).
  + Quản lý danh mục sản phẩm (CRUD, tìm kiếm danh mục).
  + Quản lý đơn hàng (view, tìm kiếm, cập nhật trạng thái đơn hàng).
  + Xem chi tiết đơn hàng với thông tin giao hàng.

## CÁC CÔNG NGHÊ SỬ DỤNG

Phần này sẽ liệt kê các cộng nghệ được sử dụng kèm theo thông tin version tại thời điểm đang làm đồ án.

**4.1 Công nghệ cho Backend.**

* **Framework và Runtime:**
  + NestJS 11.0.7: Framework Backend của NodeJS phù hợp cho việc xây dựng API dễ bảo trì và mở rộng.
  + NodeJS 22.15.0: Môi trường chạy JavaScript runtime.
  + TypeScript 5.7.3: Ngôn ngữ mở rộng của JavaScript, hỗ trợ kiểm tra kiểu tĩnh, giúp phát hiện lỗi lúc biên dịch, cải thiện khả năng debug.
* **Database và ORM:**
  + PostgreSQL 14.18: Hệ thống quản lý cơ sở dữ liệu quan hệ**.**
  + TypeORM 0.3.24: Thư viện TypeScript hỗ trợ tương tác với Database dễ dàng**.**
* **Bảo mật và xác thực:**
  + JWT (JSON Web Token)**.**
  + Passport.js với JET Strategy: dùng để xây dưng middleware cho logic xác thực request gửi lên Backend**.**
  + Thư viện bcrypt 6.0.0: dùng để hashing mật khẩu**.**
  + Class-validator và Class-transformer: 2 thư viện dùng để validation các input đâu vào và chuyển đổi kiểu dữ liệu các trường**.**
* **Dịch vụ bên thứ 3:**
  + SePaSePay API: Dịch vụ API dành cho tính năng thanh toán.
  + Mailtrap API: Dành cho tính năng đăng ký tài khoảng, sẽ gửi mail chào mừng khi đăng ký thành công. Gói miễn phí chỉ cho phép gửi nhận mail trong sandbox.
  + Axios 1.9.0: Thư viện hỗ trợ gọi các API bên ngoài ứng dụng.

**4.2. Công nghệ cho Frontend.**

* **Framework:**
  + React 19: Thư viện làm UI phổ biến nhất với kiến trúc component.
  + TypeScript: Ngôn ngữ mở rộng của Javasript.
  + Vite 6.3.5: Cộng cụ xậy dựng base ứng dụng React nhanh.
* **Styling và UI Components:**
  + TailwindCSS 4.1.7: CSS framework nhẹ, không cần viết style riêng như CSS thuần.
  + ShadcnUi: Thư viện components được build sẵn.
  + Lucide React 0.511.0: Thư viện icon.
* **Quản lý State:**
  + Zustand 5.0.5: Thư viện quản lý state cho ứng dụng React**.**
  + React Hook Form: 7.56.4: Thư viện tối ưu hiệu suất cho form kèm validation**.**
  + Zod 3.25.28: Thư viện tạo validation schema cho form**.**
* **Routing và Navigation:**
  + React Router Dom 7.56.4: Thư viện định nghĩa routing cho ứng dụng React.

**4.3. Workspace cho SharedType.**

* Monorepo: công cụ quản lý nhiều app trong một repository duy nhất, giúp chia sẽ tài nguyên giữa các app sử dụng các cộng nghệ khác nhau**.**
* PNPM >= 8.0.0: Trính quán lí gói nhanh hơn NPM mặc định.
* Shared Types Package: Package định nghĩa type chung cho Frontend và Backend, tránh boilerplate type.

**4.4. Lý do lựa chọn công nghệ.**

* **NestJS đươc chọn vì:**
  + Dễ học nếu đã biết NodeJS kèm Express.
  + Cú pháp dễ đọc
  + Sử dụng cùng ngôn ngữ lập trình với Frontend giúp giảm thiểu thời gian xây dựng ứng dụng fullstask vì không cần phải học 2 ngôn ngữ riêng biệt.
  + Kiến trúc modular tự nhiên, dễ scale và maintain.
  + Dependency injection, giảm thiểu sự phụ thuộc giữa các class, tối ưu hiệu suất do chỉ cần tạo một instance rồi sử dụng ở nhiều nơi và quản lý ở một nơi (class định nghĩa service), giúp tách biệt sự quan tâm.
  + Hệ sinh thái hỗ trợ các middleware được xây dựng sẵn cho bảo mật như Guards, Interceptors và Pipes.
* **React với TypeScript vì:**
  + Nhờ kiến trúc component-based giúp tái sử dụng code hiệu quả.
  + Sử dụng Virture DOM và thuật toán đối chiếu giúp tối ưu hiệu suất hơn khi thao tác với DOM thật.
  + Hệ sinh thái lớn, có nhiều công cụ hỗ trợ.
  + TypeScript giúp debug dễ dàng hơn.
* **Quản lý state với Zustand vì:**
  + Zustand nhẹ và dễ học hơn Redux
  + Không cần setup context rườm rà, không cần phải wrapper app component, dùng càng nhiều context thì càng phải bọc nhiều, nhìn càng rối.
  + Với Zustand chúng ta chỉ cần định nghĩa stores và có thể gọi state, funtion được viết trong store ở mọi nơi trong pham vi ứng dụng.
* **PostgresSQL vì:**
  + Hỗ trợ kiểu dữ liệu quan hệ chặc chẽ.
  + Phù hợp làm ứng dung thương mại có tính năng giao dịch do tuân thủ ACID.
  + Mãnh mẽ, dễ thao tác khi dùng với TypeORM.

## BỐ CỤC NIÊN LUẬN

Bố cục niên luận được chia thành 6 chương chính, mỗi chương sẽ tập trung vào một phía cạnh cụ thể của quá trình phát triển.

**Chương 1 - Tổng quan đề tài:** Chương này sẽ giới thiệu bối cảnh, lý do chọn đề tài, mục tiêu và pham vi công nghệ sử dụng.

**Chương 2 - Phân tích và thiết kế hệ thống:** Chương này sẽ phân tích các yêu cầu chi tiết, thiết kế kiến trúc tổng thể, thiết kế database schema với sơ đồ ERD và lập kế hoạch cho việc thực hiện.

**Chương 3 - Triển khai Backend (NestJS):** Chương này sẽ đi vào phát triển Backend, giải thích cách triển khai từng module, luồng các thực, thiết kế API, tích hợp database và logic nghiệp vụ.

**Chương 4 - Triển khai Frontend (React):** Chương này sẽ đi vào xây dựng user interface, triển khai quản lý state, kiến trúc component, tích hợp API, tối ưu hóa.

**Chương 5 - Các tính năng nâng cao:** Chương này sẽ trình bày về monorepo setup, tích hợp các dịch vụ bên thứ 3.

**Chương 6 - Kết luận và hướng phát triển:** Tổng kết kết quả đạt được, bài học kình nghiệm, những thách thức và lộ trình cải tiến trong tương lai.

Bố cục này được thiết kế để người đọc có thể hiểu được toàn bộ quá trình phát triển từ ý tưởng ban đầu đến sản phẩm hoàn thiện.

# 

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **1. PHÂN TÍCH YÊU CẦU**

Hệ thống Web được thiết kế để phục vụ cho 2 người người dùng chính, và một người dùng phụ. Người dùng cao hơn sẽ kế thừa các tính năng của người dùng thấp hơn.

**1.1. Về yêu cầu chức năng:**

**1.1.1. Chức năng dành cho người dùng thông thường:**

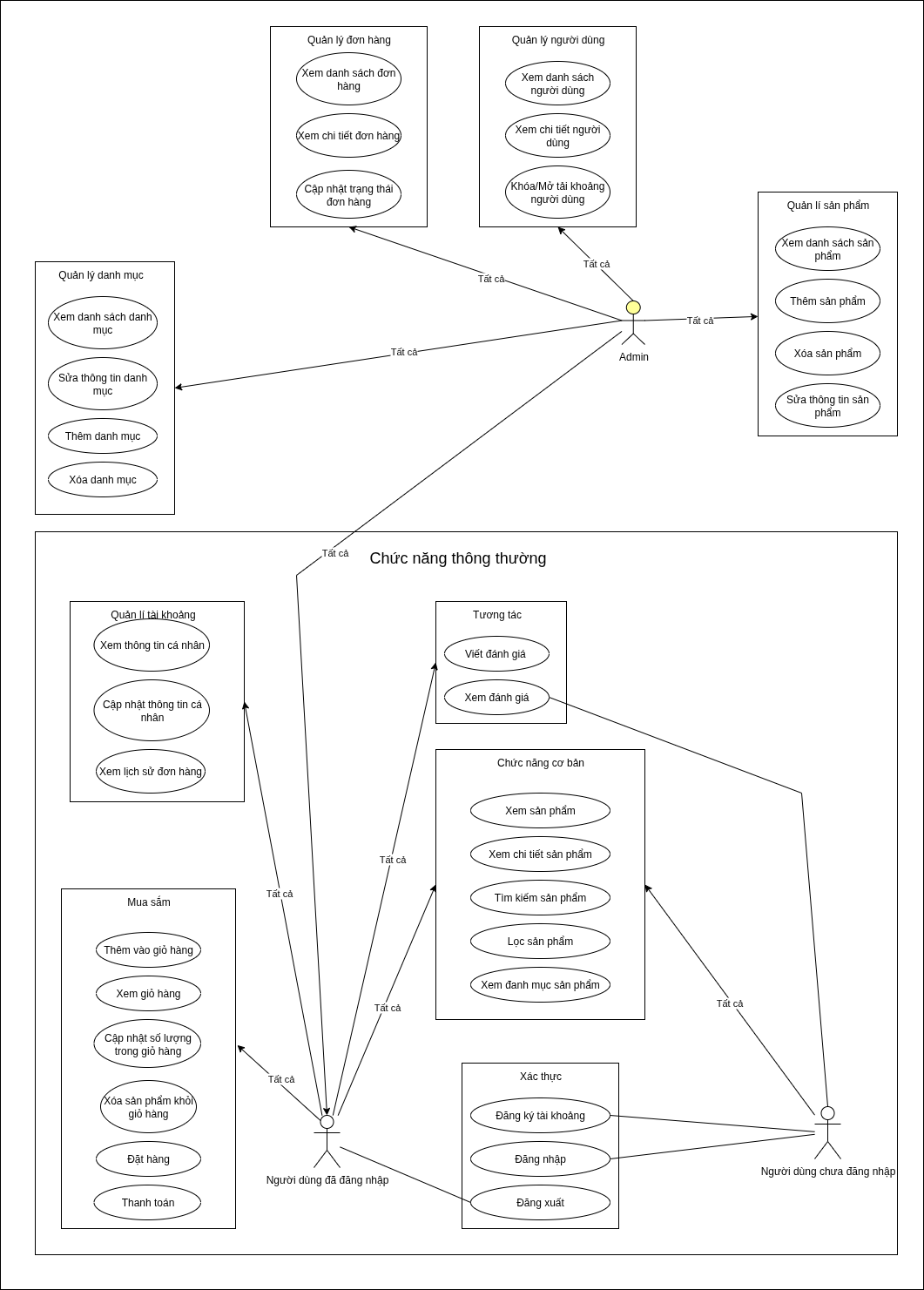
* **Quản lý tài khoảng cả nhân:**
  + Đăng ký tài khoảng mới với email và username duy nhất.
  + Đăng nhập/Đăng xuất sử dụng JWT cho xác thực.
  + Quản lý thông tin cá nhân: cập nhật địa chỉ, số điện thoại, username.
* **Duyệt và tìm kiếm sản phẩm:**
  + Xem danh sách sản phẩm với phân trang.
  + Lọc sản phẩm theo danh mục, khoảng giá tiền.
  + Tìm kiếm sản phẩm theo tên.
  + Xem chi tiết sản phẩm bao gồm mô tả, giá, bình luận, số lượng sản phẩm còn lại.
* **Quản lý giỏ hàng:**
  + Thêm sản phẩm vào giỏ hàng với số lượng tùy chọn (giới hạn bởi số lượng sản phẩm còn trong kho).
  + Cập nhật số lượng hoặc xóa sản phẩm khỏi giỏ hàng (Không thể xóa sản phẩm nếu đang có đơn hàng).
  + Xem tổng quan giỏ hàng kèm theo tổng tiền phải thanh toán.
* **Đặt hàng và thanh toán:**
  + Tạo đơn hàng từ giỏ hàng
  + Nhập thông tin giao hàng như địa chi, số điện thoại, ghi chú thêm.
  + Thanh toán qua SePay QR code hoặc thanh toán khi nhận hàng (COD).
  + Theo dõi trạng thái đơn hàng và thanh toán.
* **Lịch sử mua hàng:**
  + Xem danh sách đơn hàng đã đạt, kém trạng thái đơn hàng.

**1.1.2. Chức năng dánh cho quản trị viên (Admin):**

* **Dashboard tổng quan:** 
  + Xem thống kê tổng quan: tổng số người dùng, sản phẩm, đơn hàng.
  + Doanh thu theo thời gian.
  + Báo cáo số người dùng mới, đơn hàng mới, đơn hàng theo trạng thái.
* **Quản lí người dùng:** 
  + Xem danh sách người dùng với phân trang.
  + Tìm kiếm người dùng theo email, username.
  + Xem thông tìn người dùng.
  + Khóa người dùng.
* **Quản lý sản phẩm:** 
  + Thêm sản phẩm với đầy đủ thông tin.
  + Cập nhật thông tin sản phẩm (tên, giá, mô tả, hình ảnh, số lượng, danh mục).
  + Xóa mềm sản phẩm (soft delete).
  + Quản lí trạng thái hoạt động của sản phẩm.
* **Quản lý danh mục:**
  + CRUD danh mục sản phẩm.
  + Phân loại sản phẩm theo danh mục.
* **Quản lý đơn hàng:** 
  + Xem danh sách tất cả đơn hàng với khả năng lọc theo trạng thái.
  + Cập nhật trạng thái đơn hàng (Pending → Processing → Shipped → Delivered).
  + Xem chi tiết đơn hàng bao gồm thông tin giao hàng và sản phẩm.

**1.2. Về yêu cầu phi chức năng**

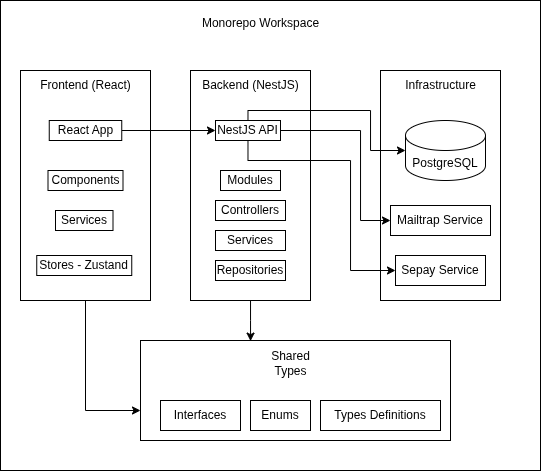
* **Bảo mật:**
  + Mã hóa mật khảu bằng bcrypt.
  + Xác thực dựa trên JWT dùng access token, refresh token.
  + Phân quyền rõ ràng giữa Admin và User.
  + Bảo vệ các API endpoint với NestJS Guards.
* **Khả năng mở rộng:**
  + Kiến trúc modular theo domain.
  + Separation of concerns rõ ràng.
  + Dễ thêm tính năng mới theo pattern có sẵn.
* **Về UX:**
  + Giao diện responsive với TailwindCSS + ShadcnUi.
  + Có loading states và error handling



***Hình 1 - Sơ đồ usecase tỗng quan các tính năng***

## **2. THIẾT KẾ KIẾN TRÚC HỆ THỐNG.**

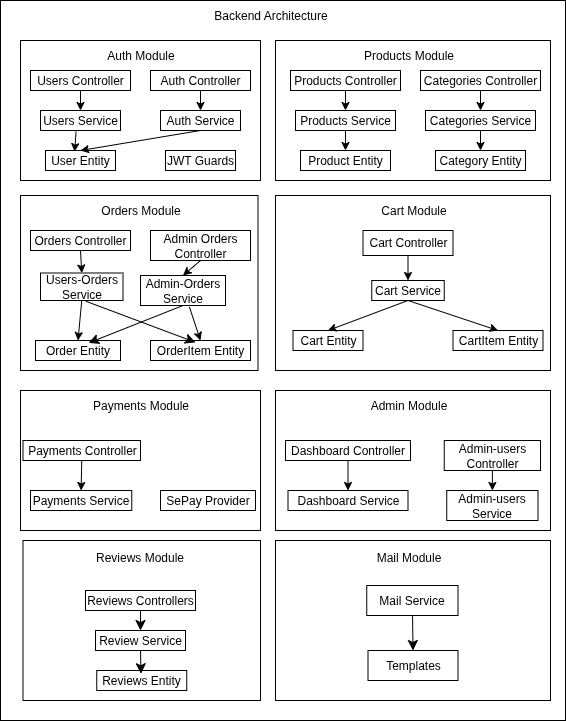
**2.1. Kiến trúc tổng quan.**

Hệ thống được thiết kế theo mô hình Client – Server sử dụng Monorepo cho phép chia sẽ kiểu dữ liệu giữa Frontend và Backend, giúp đồng bộ kiểu dữ liệu từ Backend đến Frontend hiệu quả, tránh, boilerplate code, lỗi định nghĩa thiếu các trường bên trong Interface hoặc các Type Definition khác.

Hình 2 -Sơ đồ kiến trúc tổng quan

**2.2. Kiến trúc Backend (NestJS)**

Backend được tổ chức theo kiến trúc modular phân chia theo domain, tuân thủ các nguyên tắc SOLID đẻ đảm bảo code dễ bảo trì mà mở rộng.

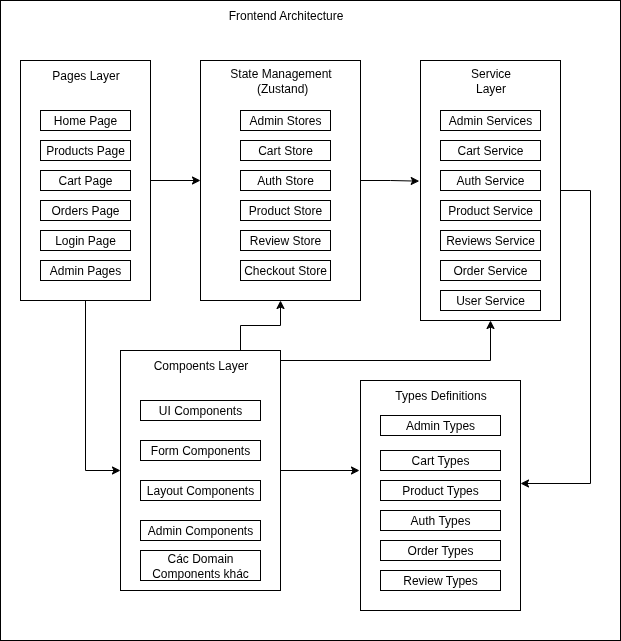
***Hình 3 - Sơ đồ kiến trúc tổng quan của Backend***

**Nguyên lý thiết kế được áp dung:**

1. S**ingle Responsibillity Principle (SRP)**: Mỗi module chỉ chịu trách nhiệm cho một domain cụ thể (Auth, Products, Orders, Cart, Payments, Admin).
2. **Dependency Inversion Principle (DIP):** Controller chỉ phụ thuộc vào abstractions (Services), không phụ thuộc vào implementations cụ thể.
3. **Open/Closed Principle (OCP):** Các module có thể mở rộng thêm chức năng mà không cần sửa đổi code hiện tại.
4. **Separation of Concerns:** Tách biệt rõ ràng giữa Controller (xử lý HTTP requests), Service (business logic), và Entity (data model).

**2.3. Kiến trúc Frontend (React):**

Frontend được tổ chức theo kiến trúc compoent-based và quản lý state tập trung sử dụng Zustand



***Hình 4 - Sơ đồ tổng quan kiến trúc Frontend***

**Đặc điểm của kiến trúc Frontend:**

* 1. **Kiến trúc Layered:** Phân chia rõ ràng thành các lớp Pages, Components, Services và Stores.
  2. **Quản lý state:** Sử dụng Zustand với cách tổ chúc stores theo domain giúp dễ quản lý, tách biệt các mối quan tâm.
  3. **An toàn kiểu dữ liệu:** TypeScript được sử dụng toàn bộ với type definition được chia sẽ từ shared-types package.
  4. **Components tái sử dụng:** Các UI components được xây dựng với ShadcnUI có thể tái sử dụng lại ở nhiều nơi, có thể dùng Tailwind customize lại khi cần, tăng tính linh hoạt khi sử dụng.

## **3. THIẾT KẾ CƠ SỞ DỮ LIỆU**

**3.1 Mô tả các thực thể chính**

Cơ sở dữ liệu được thiết kế với các thực thể chính như sau: