TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP. HCM

Khoa Công Nghệ Thông Tin

**CHUYÊN ĐỀ ỨNG DỤNG LẬP TRÌNH WEB 2**

****

**ĐỀ TÀI:**

**Trang Web bán thiết bị di động**

**SVTH: Trần Văn Toàn**

**MSSV: 2122110162**

**SVTH: Đoàn Ngọc Hải**

**MSSV: 2122110176**

*TPHCM, tháng 05 năm 2025*

# **NHẬN XÉT**

# **LỜI CẢM ƠN**

Em xin chân thành cảm ơn **Thầy Nguyễn Minh Tuấn**, giảng viên khoa **Công Nghệ Thông Tin** – **Trường Cao đẳng Công Thương TP.HCM**, đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề tài này.

Nhờ sự chỉ dẫn tận tâm của Thầy, em đã có thêm nhiều kiến thức thực tế và kinh nghiệm quý báu trong việc xây dựng và triển khai ứng dụng web sử dụng **Java Spring Boot** cho backend và **ReactJS** cho frontend. Đây là những hành trang quan trọng giúp em vững vàng hơn trong con đường học tập và làm việc sau này.

Em xin gửi lời cảm ơn sâu sắc đến Thầy cùng toàn thể quý Thầy Cô trong khoa CNTT đã luôn tạo điều kiện tốt nhất để em hoàn thành bài báo cáo này.

**Mục lục**

[**NHẬN XÉT** 2](#_Toc197977985)

[**LỜI CẢM ƠN** 3](#_Toc197977986)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 5](#_Toc197977987)

[**1.1. Giới thiệu** 5](#_Toc197977988)

[**1.2. Các công nghệ chính** 5](#_Toc197977989)

[**1.2.1. Frontend** 5](#_Toc197977990)

[**1.2.2. Backend** 5](#_Toc197977991)

[**1.3. Kiến trúc ứng dụng** 5](#_Toc197977992)

[**1.3.1. Kiến trúc MVC (Model-View-Controller)** 5](#_Toc197977993)

[**1.3.2. Kiến trúc RESTful API** 6](#_Toc197977994)

[**1.4. Các mô hình và thuật toán** 6](#_Toc197977995)

[**1.4.1. JWT (JSON Web Token)** 6](#_Toc197977996)

[**1.4.2. Mô hình phân quyền RBAC (Role-Based Access Control)** 7](#_Toc197977997)

[**CHƯƠNG 2: FRONTEND** 8](#_Toc197977998)

[**2.1. Cấu trúc thư mục** 8](#_Toc197977999)

[**2.2. Giao diện người dùng** 9](#_Toc197978000)

[**2.2.1. Trang chủ** 9](#_Toc197978001)

[**2.2.2. Giỏ hàng và thanh toán** 10](#_Toc197978002)

[**2.2.3. Trang quản trị (Admin Dashboard)** 10](#_Toc197978003)

[**2.3. Các thành phần chính** 11](#_Toc197978004)

[**2.3.1. Components** 11](#_Toc197978005)

[**2.3.2. Redux và State Management** 11](#_Toc197978006)

[**2.3.3. API Integration** 11](#_Toc197978007)

[**CHƯƠNG 3: BACKEND** 12](#_Toc197978008)

[**3.1. Cấu trúc thư mục** 12](#_Toc197978009)

[**3.2. Mô hình dữ liệu** 12](#_Toc197978010)

[**3.3. RESTful API** 13](#_Toc197978011)

[**3.3.1. Authentication API** 13](#_Toc197978012)

[**3.4. Bảo mật và xác thực** 13](#_Toc197978013)

[**3.4.1. Spring Security** 13](#_Toc197978014)

[**3.4.2. JWT Implementation** 14](#_Toc197978015)

[**CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 15](#_Toc197978016)

[**4.1. Kết quả đạt được** 15](#_Toc197978017)

[**4.2. Hạn chế và bài học kinh nghiệm** 15](#_Toc197978018)

[**4.3. Hướng phát triển** 15](#_Toc197978019)

[**4.4. Kết luận** 16](#_Toc197978020)

[**TÀI LIỆU THAM KHẢO** 17](#_Toc197978021)

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

## **1.1. Giới thiệu**

Đồ án này tập trung vào việc xây dựng một ứng dụng web bán điện thoại hoàn chỉnh, tích hợp giao diện người dùng (Frontend) sử dụng ReactJS và phần máy chủ (Backend) xây dựng bằng Java Spring Boot. Hệ thống được thiết kế theo kiến trúc phân lớp, với sự tách biệt rõ ràng giữa giao diện người dùng và logic nghiệp vụ, đảm bảo tính mô-đun, dễ bảo trì và mở rộng.

## **1.2. Các công nghệ chính**

### **1.2.1. Frontend**

* **ReactJS**: Thư viện JavaScript xây dựng giao diện người dùng với các ưu điểm như tốc độ nhanh, tái sử dụng component và Virtual DOM
* **Redux**: Quản lý state tập trung, giúp kiểm soát luồng dữ liệu hiệu quả
* **Axios**: Thư viện HTTP client giao tiếp với API, hỗ trợ Promise
* **SCSS/SASS**: Ngôn ngữ tiền xử lý CSS, cung cấp các tính năng như biến, mixin, nested rules
* **Bootstrap**: Framework CSS phát triển giao diện responsive với hệ thống grid và các component UI có sẵn

### **1.2.2. Backend**

* **Java Spring Boot**: Framework phát triển ứng dụng Java nhanh chóng với tính năng auto-configuration
* **Spring Security**: Quản lý bảo mật và xác thực với các cơ chế bảo vệ mạnh mẽ
* **Spring Data JPA**: Truy xuất và quản lý dữ liệu với interface đơn giản
* **JWT (JSON Web Token)**: Xác thực và ủy quyền phi trạng thái (stateless)
* **MySQL**: Hệ quản trị cơ sở dữ liệu quan hệ phổ biến, đáng tin cậy

## **1.3. Kiến trúc ứng dụng**

### **1.3.1. Kiến trúc MVC (Model-View-Controller)**

MVC tách ứng dụng thành ba thành phần logic:

* **Model**: Đại diện cho dữ liệu và logic nghiệp vụ
* **View**: Hiển thị dữ liệu cho người dùng
* **Controller**: Xử lý sự kiện từ người dùng và cập nhật Model và View

Ứng dụng của chúng tôi áp dụng MVC theo cách hiện đại:

* Frontend (ReactJS) đóng vai trò View và một phần Controller
* Backend (Java Spring Boot) đóng vai trò Model và Controller chính

### **1.3.2. Kiến trúc RESTful API**

Backend được thiết kế theo nguyên tắc REST (Representational State Transfer) với các ưu điểm:

* Sử dụng các phương thức HTTP (GET, POST, PUT, DELETE) phù hợp với mục đích
* Stateless: Mỗi yêu cầu từ client chứa đầy đủ thông tin cần thiết
* Caching: Hỗ trợ lưu cache để tăng hiệu suất
* Interface thống nhất: API được thiết kế nhất quán

## **1.4. Các mô hình và thuật toán**

### **1.4.1. JWT (JSON Web Token)**

JWT là chuẩn mở (RFC 7519) định nghĩa cách truyền thông tin an toàn dưới dạng JSON. Cấu trúc JWT:

* **Header**: Chứa loại token và thuật toán hashing
* **Payload**: Chứa các claims (thông tin)
* **Signature**: Đảm bảo tính toàn vẹn của token

Ưu điểm của JWT trong hệ thống:

* Giảm tải cho database: Không cần lưu trữ phiên
* Khả năng mở rộng tốt: Hoạt động tốt trong hệ thống phân tán
* Linh hoạt: Có thể truyền qua nhiều môi trường khác nhau

### **1.4.2. Mô hình phân quyền RBAC (Role-Based Access Control)**

RBAC là phương pháp hạn chế quyền truy cập hệ thống dựa trên vai trò của người dùng:

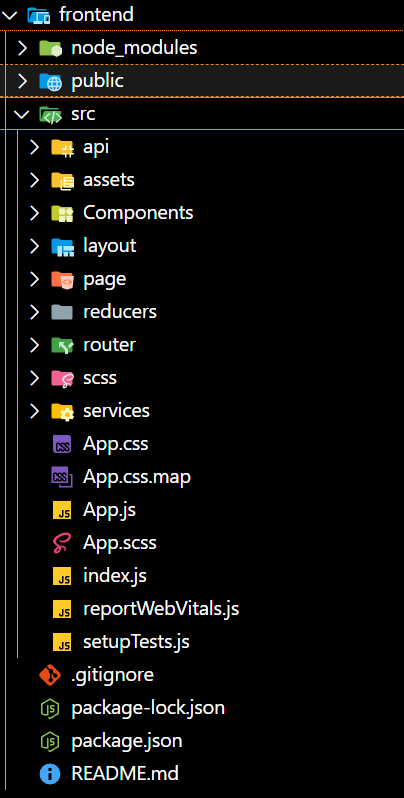
* **User (Người dùng)**: End-user, Admin, Manager
* **Role (Vai trò)**: Tập hợp các quyền
* **Permission (Quyền)**: Khả năng thực hiện một hành động cụ thể

Trong hệ thống của chúng tôi, các vai trò chính:

* **Khách hàng**: Xem sản phẩm, thêm vào giỏ hàng, thanh toán
* **Admin**: Quản lý sản phẩm, người dùng, đơn hàng và thống kê

# **CHƯƠNG 2: FRONTEND**

## **2.1. Cấu trúc thư mục**

Cấu trúc thư mục frontend được tổ chức theo tính năng và chức năng, giúp dễ dàng mở rộng và bảo trì:

frontend/

├── src/

│ ├── api/ # Modules gọi API

│ ├── assets/ # Tài nguyên tĩnh (hình ảnh, font)

│ ├── Components/ # Components tái sử dụng

│ ├── layout/ # Layout (header, footer, sidebar)

│ ├── page/ # Các trang chính

│ ├── reducers/ # Redux reducers

│ ├── router/ # Cấu hình định tuyến

│ ├── scss/ # SCSS stylesheets

│ ├── services/ # Services nghiệp vụ

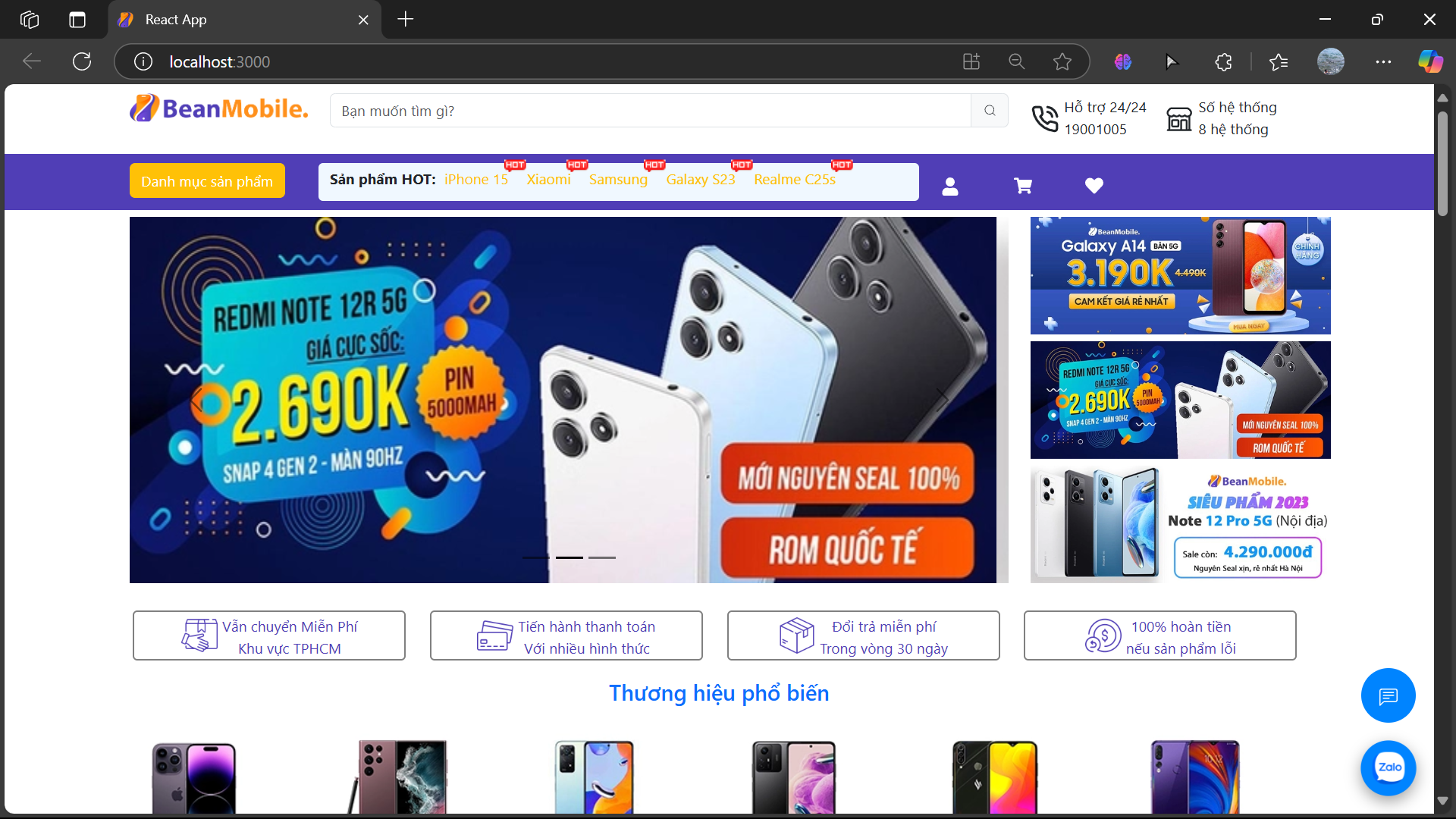
│ └── App.js # Component gốc  
**Hình 1: Cấu trúc thư mục frontend**

## **2.2. Giao diện người dùng**

### **2.2.1. Trang chủ**

Trang chủ hiển thị các sản phẩm nổi bật, banner quảng cáo và danh mục sản phẩm. Thiết kế hiện đại, tối ưu trải nghiệm người dùng với:

* Slider banner chính với các ưu đãi
* Grid sản phẩm với bộ lọc
* Thanh tìm kiếm nhanh

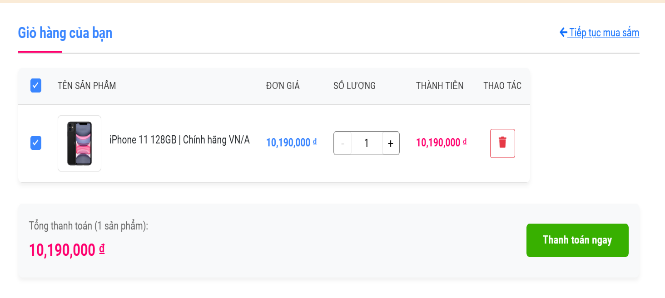
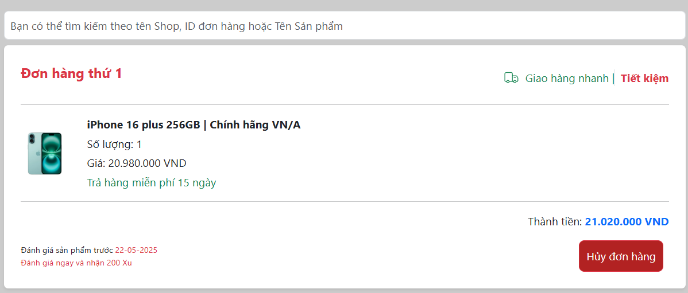


**Hình 2: Giao diện trang chủ**

### **2.2.2. Giỏ hàng và thanh toán**

Quy trình giỏ hàng và thanh toán được thiết kế trực quan:

* Hiển thị sản phẩm trong giỏ hàng
* Tính toán tổng tiền và thuế
* Form nhập thông tin giao hàng
* Các phương thức thanh toán

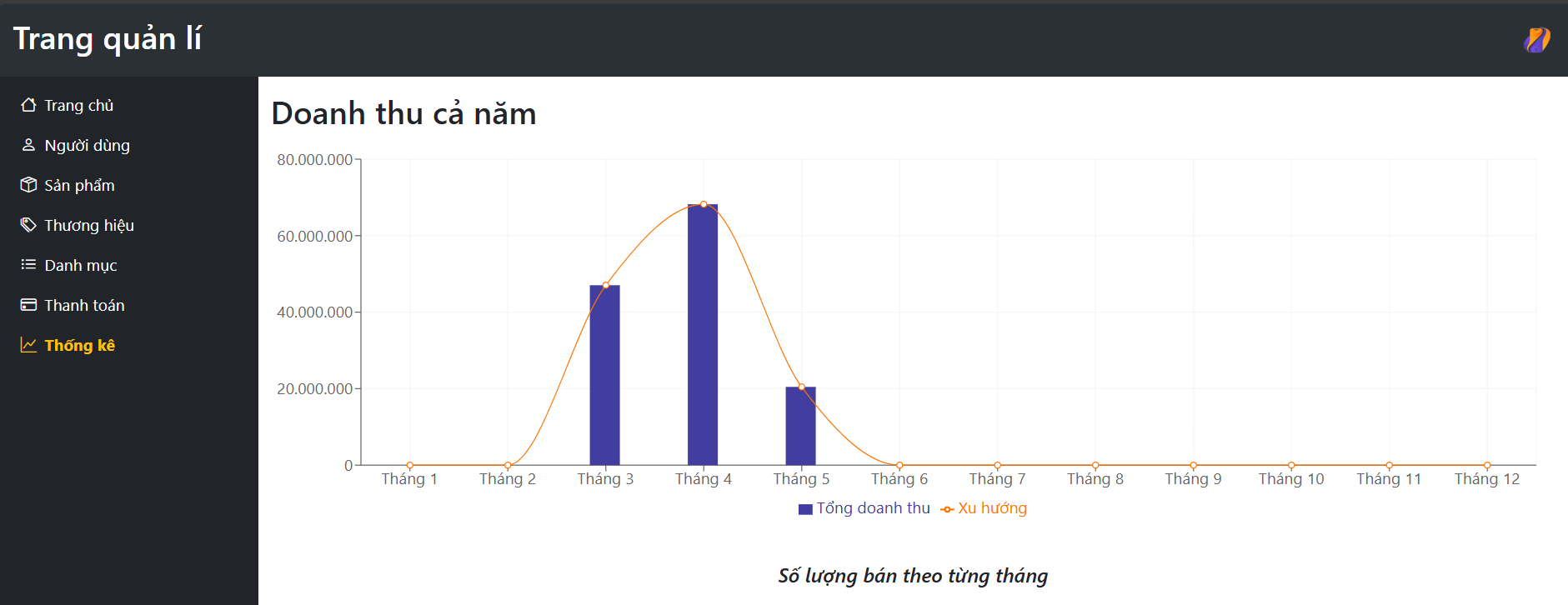
****

**Hình 3: Trang giỏ hàng và thanh toán**

### **2.2.3. Trang quản trị (Admin Dashboard)**

Giao diện quản trị hiện đại với các tính năng:

* Thống kê doanh số, đơn hàng
* Quản lý sản phẩm (thêm, sửa, xóa)
* Quản lý người dùng
* Quản lý đơn hàng



**Hình 4: Dashboard quản trị**

## **2.3. Các thành phần chính**

### **2.3.1. Components**

Components là các khối xây dựng UI tái sử dụng. Dự án sử dụng hệ thống components phân cấp:

**Common Components**:

* **ProductCard**: Hiển thị thông tin sản phẩm dạng card
* **Button**: Các loại nút với nhiều style
* **Modal**: Cửa sổ pop-up đa mục đích
* **Rating**: Hiển thị đánh giá sao
* **Pagination**: Phân trang

### **2.3.2. Redux và State Management**

Redux được sử dụng để quản lý state của ứng dụng với luồng dữ liệu một chiều. Các store chính:

* **authSlice**: Quản lý trạng thái đăng nhập
* **cartSlice**: Quản lý giỏ hàng
* **productSlice**: Quản lý sản phẩm
* **orderSlice**: Quản lý đơn hàng

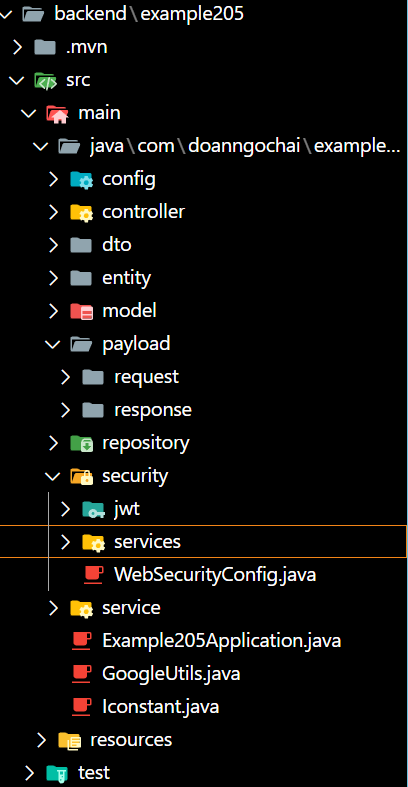
### **2.3.3. API Integration**

Ứng dụng sử dụng Axios để giao tiếp với backend API. Các API được tổ chức

theo module:

# **CHƯƠNG 3: BACKEND**

## **3.1. Cấu trúc thư mục**

Backend được tổ chức theo mô hình chuẩn của Spring Boot, giúp dễ dàng mở rộng và bảo trì:

backend/example205/

├── src/main/java/com/doanngochai/example205/

│ ├── config/ # Cấu hình ứng dụng

│ ├── controller/ # Xử lý request/response

│ ├── dto/ # Data Transfer Objects

│ ├── entity/ # Các entity JPA

│ ├── repository/ # JPA repositories

│ ├── security/ # Cấu hình bảo mật

│ │ ├── jwt/ # JWT implementation

│ │ └── services/ # Security services

│ └── service/ # Business logic

**Hình 5: Cấu trúc thư mục backend**

## **3.2. Mô hình dữ liệu**

Cơ sở dữ liệu được thiết kế với các entity chính:

* **User**: Người dùng hệ thống
* **Product**: Sản phẩm
* **Category**: Danh mục sản phẩm
* **Order**: Đơn hàng
* **OrderItem**: Chi tiết đơn hàng
* **Review**: Đánh giá sản phẩm

## **3.3. RESTful API**

Backend cung cấp RESTful API với các endpoint chính:

### **3.3.1. Authentication API**

POST /api/auth/register - Đăng ký tài khoản

POST /api/auth/login - Đăng nhập

POST /api/auth/refresh-token - Làm mới token

## **3.4. Bảo mật và xác thực**

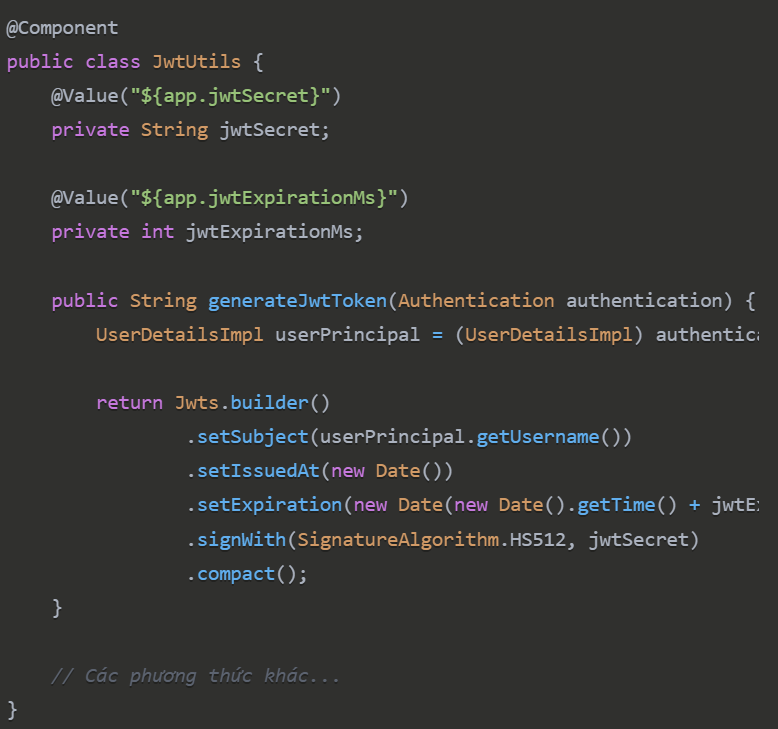
### **3.4.1. Spring Security**

Spring Security được cấu hình để bảo vệ các endpoint:

**Hình 6: Cơ chế xác thực và bảo mật trong ứng dụng(minh họa)**

### **3.4.2. JWT Implementation**

JWT được triển khai để xác thực và phân quyền:



**Hình 7: Luồng xác thực với JWT**

# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **4.1. Kết quả đạt được**

Ứng dụng web bán điện thoại đã được xây dựng thành công với:

* Giao diện người dùng hiện đại, trực quan và responsive
* Hệ thống backend ổn định, bảo mật và dễ mở rộng
* Tích hợp đầy đủ các chức năng cơ bản của một website thương mại điện tử
* Trải nghiệm mua sắm trực tuyến mượt mà và thuận tiện

## **4.2. Hạn chế và bài học kinh nghiệm**

Trong quá trình phát triển, chúng tôi gặp một số thách thức:

* Tối ưu hiệu suất với số lượng sản phẩm lớn
* Đồng bộ state giữa các component phức tạp
* Xử lý bảo mật và ngăn chặn các lỗ hổng tiềm ẩn

Bài học rút ra:

* Cần lập kế hoạch chi tiết trước khi bắt đầu
* Thiết kế API một cách cẩn thận và nhất quán
* Tổ chức code theo module để dễ bảo trì

## **4.3. Hướng phát triển**

Dự án có thể phát triển thêm với các tính năng nâng cao:

1. **Trải nghiệm người dùng**:
   * Tìm kiếm bằng giọng nói và hình ảnh
   * Chatbot hỗ trợ khách hàng
   * AR/VR để xem trước sản phẩm
2. **Tính năng kinh doanh**:
   * Hệ thống khuyến mãi và mã giảm giá thông minh
   * Tích hợp nhiều phương thức thanh toán
   * Chương trình khách hàng thân thiết
3. **Kỹ thuật**:
   * Triển khai Progressive Web App (PWA)
   * Tối ưu hóa SEO
   * Phân tích dữ liệu người dùng và ML cho đề xuất sản phẩm

## **4.4. Kết luận**

Ứng dụng web bán điện thoại đã được xây dựng thành công với kiến trúc hiện đại, tận dụng sức mạnh của ReactJS và Spring Boot. Dự án không chỉ đáp ứng các yêu cầu chức năng cơ bản mà còn cung cấp nền tảng vững chắc để phát triển thêm các tính năng trong tương lai.

# **TÀI LIỆU THAM KHẢO**

1. Spring Boot Documentation: <https://docs.spring.io/spring-boot/docs/current/reference/html/>
2. Baeldung - Spring Boot Tutorials: https://www.baeldung.com/spring-boot
3. Java Persistence API (JPA) Reference: <https://docs.oracle.com/javaee/7/tutorial/persistence-intro.htm>
4. Spring Security Official Guide: <https://spring.io/guides/topicals/spring-security-architecture/>
5. REST API Design Best Practices: <https://restfulapi.net/>
6. React Official Documentation: <https://react.dev/>
7. React Router Documentation: https://reactrouter.com/en/main
8. MDN Web Docs - JavaScript & Web APIs: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
9. Axios (HTTP Client) Documentation: <https://axios-http.com/>
10. Tailwind CSS: <https://tailwindcss.com/docs>
11. Visual Studio Code: <https://code.visualstudio.com/>
12. Postman API Tool: <https://www.postman.com/>
13. Git & GitHub Guides: <https://docs.github.com/en/get-started>
14. MySQL Official Documentation: <https://dev.mysql.com/doc/>