**CHƯƠNG 1: MÔ TẢ ĐỀ TÀI THỰC HIỆN**

**1.1 Giới thiệu về đề tài**

Website bán cây cảnh là một nền tảng thương mại điện tử được thiết kế để kết nối người mua và người bán các loại cây cảnh khác nhau. Trong bối cảnh nhu cầu trang trí không gian sống bằng cây xanh ngày càng tăng, việc xây dựng một website như vậy không chỉ đáp ứng nhu cầu thị trường mà còn tạo điều kiện thuận lợi cho người dùng giao dịch trực tuyến một cách nhanh chóng và hiệu quả.

**1.2 Mục tiêu của dự án**

Dự án hướng đến việc:

* Cung cấp một giao diện thân thiện, dễ sử dụng cho cả người mua và người bán.
* Quản lý hàng tồn kho cây cảnh một cách khoa học.
* Đảm bảo quy trình thanh toán trực tuyến an toàn và thuận tiện.
* Hỗ trợ người dùng tiếp cận thông tin chi tiết về các loại cây cảnh (giá cả, nguồn gốc, cách chăm sóc).

**1.3 Phạm vi của dự án**

Website sẽ bao gồm các tính năng chính như:

* **Đăng ký/Đăng nhập**: Cho phép người dùng tạo tài khoản và đăng nhập.
* **Danh sách sản phẩm**: Hiển thị các loại cây cảnh kèm thông tin chi tiết.
* **Tìm kiếm**: Tìm kiếm cây cảnh theo danh mục, giá cả hoặc từ khóa.
* **Giỏ hàng**: Quản lý sản phẩm mà người dùng muốn mua.
* **Thanh toán**: Tích hợp cổng thanh toán trực tuyến.
* **Quản lý đơn hàng**: Theo dõi trạng thái đơn hàng.
* **Quản lý người dùng**: Dành cho quản trị viên để quản lý tài khoản và sản phẩm.

**CHƯƠNG 2: VẼ SƠ ĐỒ USE CASE, DIAGRAM…**

**2.1 Sơ đồ Use Case**

Sơ đồ Use Case mô tả các tác nhân (actors) và các trường hợp sử dụng (use cases) trong hệ thống:

* **Tác nhân**:
  + **Người dùng (User)**: Khách hàng mua cây cảnh.
  + **Quản trị viên (Admin)**: Người quản lý hệ thống.
* **Các trường hợp sử dụng chính**:
  + Đăng ký tài khoản.
  + Đăng nhập.
  + Xem danh sách cây cảnh.
  + Thêm sản phẩm vào giỏ hàng.
  + Thanh toán.
  + Quản lý đơn hàng (dành cho người dùng và quản trị viên).
  + Quản lý sản phẩm (dành cho quản trị viên).

*(Gợi ý: Bạn có thể sử dụng công cụ như Draw.io hoặc Visio để vẽ sơ đồ Use Case dựa trên các tác nhân và trường hợp này.)*

**2.2 Class Diagram**

Sơ đồ Class mô tả cấu trúc các lớp trong hệ thống:

* **User**: Lưu trữ thông tin người dùng (ID, tên, email, mật khẩu).
* **Product**: Thông tin cây cảnh (ID, tên, giá, mô tả, số lượng tồn kho).
* **Order**: Thông tin đơn hàng (ID, ID người dùng, danh sách sản phẩm, tổng tiền, trạng thái).
* **Cart**: Giỏ hàng (ID người dùng, danh sách sản phẩm).
* **Payment**: Thanh toán (ID đơn hàng, phương thức thanh toán, trạng thái).

*(Gợi ý: Vẽ sơ đồ với các mối quan hệ giữa các lớp, ví dụ: User có nhiều Order, Order chứa nhiều Product.)*

**2.3 Sequence Diagram**

Sơ đồ Sequence mô tả luồng tương tác giữa các đối tượng cho một số trường hợp sử dụng chính:

* **Đăng nhập**: Người dùng nhập thông tin → Hệ thống kiểm tra → Trả về kết quả thành công/thất bại.
* **Thêm vào giỏ hàng**: Người dùng chọn sản phẩm → Gửi yêu cầu → Hệ thống cập nhật giỏ hàng.
* **Thanh toán**: Người dùng xác nhận đơn hàng → Hệ thống xử lý thanh toán → Cập nhật trạng thái đơn hàng.

*(Gợi ý: Vẽ luồng tương tác giữa các thành phần như giao diện ReactJS, API SpringBoot, và cơ sở dữ liệu MySQL.)*

**CHƯƠNG 3: TRÌNH BÀY CÔNG NGHỆ THỰC HIỆN**

**3.1 MySQL**

* **Vai trò**: Cơ sở dữ liệu quan hệ dùng để lưu trữ thông tin hệ thống.
* **Ứng dụng**: Lưu trữ bảng Users, Products, Orders, Payments với các mối quan hệ rõ ràng (1-n, n-n).
* **Lợi ích**: Hỗ trợ truy vấn nhanh, quản lý dữ liệu hiệu quả.

**3.2 SpringBoot**

* **Vai trò**: Framework Java để xây dựng backend.
* **Ứng dụng**: Tạo các API RESTful (ví dụ: /api/products, /api/orders) để giao tiếp với frontend và cơ sở dữ liệu.
* **Lợi ích**: Dễ tích hợp với MySQL qua Spring Data JPA, hỗ trợ bảo mật và quản lý API.

**3.3 MicroService**

* **Vai trò**: Kiến trúc dịch vụ vi mô chia nhỏ hệ thống thành các dịch vụ độc lập.
* **Ứng dụng**:
  + Dịch vụ người dùng (User Service): Quản lý đăng ký, đăng nhập.
  + Dịch vụ sản phẩm (Product Service): Quản lý danh sách cây cảnh.
  + Dịch vụ đơn hàng (Order Service): Xử lý đơn hàng và thanh toán.
* **Lợi ích**: Dễ mở rộng, bảo trì, và triển khai từng phần.

**3.4 ReactJS**

* **Vai trò**: Thư viện JavaScript để xây dựng giao diện người dùng.
* **Ứng dụng**: Tạo các thành phần (components) như trang chủ, danh sách sản phẩm, giỏ hàng, trang thanh toán.
* **Lợi ích**: Giao diện động, tương tác cao, tối ưu trải nghiệm người dùng.

**CHƯƠNG 4: TRÌNH BÀY PROJECT ĐÃ HOÀN THÀNH**

**4.1 Hình ảnh giao diện**

* **Trang chủ**: Hiển thị danh sách cây cảnh nổi bật.
* **Trang sản phẩm**: Chi tiết từng cây cảnh (hình ảnh, giá, mô tả).
* **Giỏ hàng**: Danh sách sản phẩm đã chọn và tổng tiền.
* **Trang thanh toán**: Form nhập thông tin thanh toán.
* **Trang quản trị**: Bảng điều khiển để thêm/sửa/xóa sản phẩm.

*(Gợi ý: Chụp ảnh màn hình từ project thực tế của bạn để minh họa.)*

**4.2 Cấu trúc cơ sở dữ liệu**

* **Bảng Users**: id, username, password, email.
* **Bảng Products**: id, name, price, description, stock.
* **Bảng Orders**: id, user\_id, total, status.
* **Bảng Order\_Items**: order\_id, product\_id, quantity (bảng trung gian giữa Orders và Products).
* **Truy vấn mẫu**:

sql

CollapseWrapCopy

SELECT p.name, p.price FROM Products p WHERE p.stock > 0;

**4.3 Một số đoạn mã nguồn minh họa**

* **API lấy danh sách sản phẩm (SpringBoot)**:

java

CollapseWrapCopy

@RestController

@RequestMapping("/api/products")

public class ProductController {

@Autowired

private ProductService productService;

@GetMapping

public List<Product> getAllProducts() {

return productService.findAll();

}

}

* **Component ReactJS hiển thị sản phẩm**:

jsx

CollapseWrapCopy

function ProductList({ products }) {

return (

<div>

{products.map(product => (

<div key={product.id}>

<h3>{product.name}</h3>

<p>{product.price} VND</p>

</div>

))}

</div>

);

}

**CHƯƠNG 5: HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI**

**5.1 Mở rộng tính năng**

* Thêm tính năng đánh giá sản phẩm và gợi ý cây cảnh dựa trên sở thích người dùng.
* Tích hợp mạng xã hội để chia sẻ sản phẩm.
* Hỗ trợ đa ngôn ngữ (tiếng Anh, tiếng Việt).

**5.2 Cải thiện hiệu suất**

* Sử dụng caching (Redis) để tăng tốc độ truy xuất dữ liệu.
* Áp dụng load balancing để xử lý lượng truy cập lớn.

**5.3 Tăng cường bảo mật**

* Sử dụng HTTPS và mã hóa dữ liệu nhạy cảm.
* Thêm xác thực hai yếu tố (2FA) cho tài khoản người dùng.

**5.4 Phát triển ứng dụng di động**

* Xây dựng ứng dụng mobile (dùng React Native) để mở rộng phạm vi tiếp cận.