

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



BÁO CÁO GIỮA KỲ

Phát triển hệ thống thương mại điện tử

Dự án: Xây dựng website bán đồ gỗ nội thất

Giảng viên hướng dẫn : Kim Ngọc Bách

Nhóm lớp : 01

Nhóm BTL : 11

Danh sách thành viên:

Nguyễn Nhật Minh	B21DCCN529
Trần Thị Hiền	B21DCCN337
Hà Trần Thế Anh	B21DCCN145

Hà Nội – 2025

MỤC LỤC

MỤC LỤC.....	2
1. Tổng quan dự án.....	3
a. Tổng quan.....	3
b. Mô hình kinh doanh.....	3
c. Mô hình doanh thu.....	3
2. Cơ sở lý thuyết, công nghệ áp dụng.....	4
a. Cơ sở lý thuyết.....	4
b. Công nghệ áp dụng.....	4
3. Mô tả hệ thống.....	6
a, Phía client:.....	6
b, Phía Admin:.....	7
4. Kế hoạch thực thi, triển khai dự án.....	8
4.1. Các giai đoạn thực thi dự án.....	8
4.2. Phân chia công việc.....	9

1. Tổng quan dự án

a. Tổng quan

Tên dự án: Xây dựng website bán đồ gỗ - nội thất

Mô tả: Dự án là một hệ thống thương mại điện tử được phát triển để cung cấp các sản phẩm đồ gỗ - nội thất trực tuyến.

- Hệ thống này bao gồm các chức năng cơ bản như: quản lý sản phẩm, quản lý đơn hàng, và tích hợp thanh toán trực tuyến. Ngoài ra còn có các tính năng khác phục vụ cho công tác thương mại như: tài khoản, giỏ hàng, danh mục mặt hàng, thanh toán khi nhận hàng, chat box để trao đổi giữa doanh nghiệp và người tiêu dùng.... Người dùng có thể duyệt qua các sản phẩm, trao đổi với người bán, thêm vào giỏ hàng, và thực hiện thanh toán trực tuyến hoặc khi nhận được hàng.

Các mặt hàng cơ bản: bàn ghế gỗ, tủ gỗ, bàn thờ, giường, v.v.

b. Mô hình kinh doanh

Dự án sử dụng mô hình kinh doanh B2C (Business-to-Consumer)

Đặc điểm cơ bản của mô hình B2C:

- Đối tượng khách hàng: Người tiêu dùng cá nhân.
- Quy trình mua bán: Đơn giản và trực tiếp, người tiêu dùng có thể chọn sản phẩm và mua ngay trên trang web.
- Sản phẩm: Các mặt hàng được liệt kê với thông tin chi tiết như tên, giá cả, và hình ảnh.

c. Mô hình doanh thu

Mô hình doanh thu: Bán hàng trực tiếp (Sales Revenue Model)

Chi tiết:

- Nguồn doanh thu chính: Doanh thu được tạo ra từ việc bán sản phẩm trực tiếp cho người tiêu dùng trên trang web.
- Phương thức thanh toán: Tích hợp các cổng thanh toán trực tuyến để người tiêu dùng có thể thanh toán một cách thuận tiện và an toàn.
- Chiến lược giá: Định giá sản phẩm dựa trên chi phí, giá trị thị trường và cạnh tranh để tối ưu hóa lợi nhuận.

2. Cơ sở lý thuyết, công nghệ áp dụng

a. Cơ sở lý thuyết

Kiến trúc MVC (Model-View-Controller):

Mô hình kiến trúc phần mềm giúp tách biệt các thành phần của ứng dụng:

- **Model:** Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.
- **View:** Quản lý giao diện người dùng và hiển thị dữ liệu từ Model.
- **Controller:** Xử lý các yêu cầu từ người dùng, tương tác với Model và cập nhật View.

Tách biệt các thành phần giúp dễ dàng thay đổi và nâng cấp mà không ảnh hưởng đến toàn bộ dự án.

Nguyên lý RESTful API:

- **Stateless:** Mỗi yêu cầu từ client đến server đều độc lập và không lưu trạng thái.
- **Resource-based:** Mỗi tài nguyên (resource) được định danh bằng một URI duy nhất.
- **HTTP Methods:** Sử dụng các phương thức HTTP như GET, POST, PUT, DELETE để thao tác với tài nguyên.

RESTful API cho phép các ứng dụng dễ dàng tương tác và trao đổi dữ liệu. Điều này đặc biệt hữu ích trong các ứng dụng web và mobile, nơi cần tích hợp nhiều dịch vụ và hệ thống khác nhau. RESTful API cũng giúp cải thiện khả năng mở rộng và bảo trì của hệ thống, vì các tài nguyên và phương thức có thể được phát triển và quản lý độc lập.

b. Công nghệ áp dụng

Ngôn ngữ và nền tảng:

- **Node.js:** Nền tảng chạy JavaScript phía server, cho phép xây dựng các ứng dụng mạng hiệu năng cao và mở rộng dễ dàng.
- **React.js:** Thư viện JavaScript để xây dựng giao diện người dùng, giúp tạo ra các ứng dụng web tương tác và hiệu quả.
- **Express.js:** Framework web nhẹ và linh hoạt cho Node.js, cung cấp một tập hợp tính năng mạnh mẽ để xây dựng các API và ứng dụng web.
- **Socket.IO:** Thư viện JavaScript hỗ trợ giao tiếp hai chiều giữa client và server, sử dụng WebSocket và các phương thức fallback như long-polling. Socket.IO giúp xây dựng các ứng dụng thời gian thực, như chat box, một cách hiệu quả và ổn định.

Quản lý giao diện người dùng (View engine):

- **Pug:** Template engine cho Node.js, cho phép viết HTML một cách ngắn gọn và dễ đọc, hỗ trợ tạo giao diện động một cách hiệu quả.

- **Bootstrap**: Thư viện CSS phổ biến, cung cấp các thành phần giao diện đáp ứng và tiện dụng, giúp thiết kế giao diện người dùng nhất quán và thân thiện trên nhiều thiết bị.

Cơ sở dữ liệu:

- **MongoDB**: Hệ quản trị cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng document linh hoạt, phù hợp cho các ứng dụng yêu cầu mở rộng và hiệu năng cao.
- **Mongoose**: Thư viện ODM (Object Data Modeling) cho Node.js, cung cấp các schema để mô hình hóa dữ liệu và tương tác với MongoDB một cách dễ dàng và hiệu quả.
- **Mongoose-slug-updater**: Plugin cho Mongoose, tự động tạo và cập nhật slug cho các document, giúp tạo URL thân thiện và dễ đọc.

Upload và lưu trữ hình ảnh:

- **Multer**: Middleware cho Express.js, hỗ trợ xử lý multipart/form-data, chủ yếu được sử dụng để upload files, bao gồm hình ảnh.
- **Cloudinary**: Nền tảng quản lý và lưu trữ hình ảnh trên đám mây, cung cấp các API mạnh mẽ để upload, lưu trữ, quản lý và tối ưu hóa hình ảnh.
- **Streamifier**: Thư viện giúp chuyển đổi buffer thành luồng (stream), hỗ trợ quá trình upload dữ liệu lên các dịch vụ như Cloudinary một cách hiệu quả.

Công cụ hỗ trợ server:

- **dotenv**: Thư viện giúp quản lý các biến môi trường trong ứng dụng Node.js thông qua file .env, tăng tính bảo mật và linh hoạt trong cấu hình ứng dụng.
- **method-override**: Middleware cho phép sử dụng các phương thức HTTP như PUT hoặc DELETE trong các form HTML, vốn chỉ hỗ trợ GET và POST.
- **body-parser**: Middleware giúp phân tích (parse) dữ liệu từ body của request, hỗ trợ các định dạng như JSON, URL-encoded, giúp xử lý dữ liệu gửi từ client một cách dễ dàng.
- **express-flash**: Middleware cung cấp cơ chế gửi thông báo tạm thời (flash messages) giữa các request, hữu ích cho việc thông báo kết quả sau khi xử lý form hoặc thao tác người dùng.

Triển khai và thiết lập:

- **Nodemon**: Công cụ giám sát các thay đổi trong mã nguồn và tự động khởi động lại server, giúp tăng hiệu suất trong quá trình phát triển.
- **Quản lý biến môi trường và file tĩnh**: Sử dụng file .env để cấu hình các biến môi trường, và thiết lập thư mục public để phục vụ các file tĩnh như CSS, JavaScript và hình ảnh.
- **Vercel**: Nền tảng đám mây giúp triển khai ứng dụng dễ dàng và hiệu quả. Vercel tích hợp Git, tự động build và deploy phiên bản mới, sử dụng CDN toàn cầu để tối ưu hóa

hiệu suất và hỗ trợ serverless functions, giúp quản lý và mở rộng hệ thống một cách linh hoạt.

3. Mô tả hệ thống

a. Phía client:

Trang chủ:

- Hiện thị article, các sản phẩm nổi bật, chương trình khuyến mãi và thông tin về brand.
- Thanh tìm kiếm nâng cao.
- Menu điều hướng đến các danh mục sản phẩm.
- Botchat (dùng để giao tiếp, giải đáp mọi thắc mắc giữa client và admin liên quan đến hệ thống và các sản phẩm).

Danh mục sản phẩm:

- Thanh tìm kiếm sản phẩm.
- Bộ lọc sản phẩm nâng cao(lọc theo đánh giá, theo giá,...).
- Phân loại sản phẩm theo nhóm danh mục sản phẩm.
- Hỗ trợ bộ lọc và tìm kiếm nâng cao theo giá, đánh giá, danh mục sản phẩm,...
- Xem chi tiết danh mục sản phẩm và các sản phẩm thuộc danh mục đó.
- Xem chi tiết một sản phẩm(Thông tin, đánh giá sản phẩm).

Giỏ hàng:

- Xem thông tin giỏ hàng.
- Thêm sản phẩm vào giỏ hàng.
- Xóa sản phẩm khỏi giỏ hàng.
- Sửa sản phẩm trong giỏ hàng.
- Cập nhật số lượng sản phẩm trong giỏ hàng.

Thanh toán:

- Xem hóa đơn.
- Tiến hành thanh toán.
- Thanh toán.

Chính sách:

- Xem hóa đơn.
- Đánh giá sản phẩm.
- Yêu cầu hoàn trả.
- Xem trạng thái hoàn trả.

Theo dõi cập nhập trạng thái đơn hàng.

- Xem trạng thái đơn hàng.
- Lịch sử đơn hàng

Tài khoản:

- Đăng ký tài khoản người dùng.
- Đăng nhập tài khoản người dùng.
- Thông tin tài khoản người dùng.
- Đăng xuất tài khoản người dùng.
- Quên mật khẩu.
- Nhập OTP, xác nhận gmail đặt lại mật khẩu.
- Đặt lại mật khẩu.

b. Phía Admin:

Dashboard: Thống kê website: Thống kê trạng thái đơn hàng, số lượng user,...

Sản phẩm:

- Xem danh sách sản phẩm.
- Tìm kiếm sản phẩm.
- Bộ lọc nâng cao(sắp xếp theo các tiêu chí như trạng thái, giá, theo thứ tự tiêu đề, theo vị trí,...).
- Phân trang danh sách sản phẩm.
- Chỉnh sửa vị trí xuất hiện, giới hạn sản phẩm tối đa xuất hiện trong một trang phía client.
- Xem chi tiết sản phẩm.
- Thêm sản phẩm mới.
- Sửa, xóa sản phẩm.
- Thay đổi,xóa, cập nhập trạng thái 1 hoặc nhiều sản phẩm(multi box).

Danh mục sản phẩm:

- Xem danh sách danh mục sản phẩm.
- Tạo danh mục sản phẩm.
- Thêm, sửa,xóa danh mục sản phẩm.
- Cập nhập trạng thái danh mục sản phẩm.

Quản lý đơn hàng:

- Xem thông tin đơn hàng.
- Xác nhận đơn hàng.
- Xem danh sách yêu cầu hoàn hàng.
- Xử lý yêu cầu hoàn hàng.
- Xem, cập nhập trạng thái đơn hàng.

Phân quyền:

- Tạo danh sách các quyền.
- Sửa, xóa các quyền.

Quản lý botchat:

- Xem các thắc mắc từ khách hàng.
- Trả lời khách hàng.

Quản lý article:

- Thêm, sửa xóa các bài đăng trên trang chủ client.

Tài khoản admin:

- Đăng nhập tài khoản admin.
- Đăng xuất tài khoản admin.
- Xem thông tin chi tiết tài khoản.
- Chỉnh sửa thông tin tài khoản.

4. Kế hoạch thực thi, triển khai dự án

a. Các giai đoạn thực thi dự án

Giai đoạn	Thời gian thực hiện (tính từ ngày bắt đầu làm dự án)
1. Lập kế hoạch và phân tích yêu cầu	Tuần 1
2. Thiết kế CSDL	Tuần 1
3. Thiết kế giao diện người dùng	Tuần 2 -3
3.1 Thiết kế layout tổng thể	
3.2 Thiết kế trang chủ	
3.3 Thiết kế trang danh mục, sản phẩm	
3.4 Thiết kế trang giỏ hàng, thanh toán	
4. Xây dựng chức năng người dùng	Tuần 4
4.1. Đăng ký, đăng nhập, quên mật khẩu	
4.2. Xem, sửa thông tin tài khoản, lịch sử đơn hàng	
4.3. Giỏ hàng, thanh toán (ngân hàng, ví điện tử, COD), trả hàng/hoàn tiền.	
5. Xây dựng chức năng quản lý sản phẩm	Tuần 5
5.1. Xem danh sách, thêm/sửa/xóa sản phẩm	
5.2. Tìm kiếm sản phẩm	
5.3. Danh mục sản phẩm và bộ lọc nâng cao	
6. Xây dựng chức năng quản lý đơn hàng	Tuần 6
7. Xây dựng trang chủ, bài viết, khuyến mãi	Tuần 7
8. Tính năng chat (chatbox)	Tuần 7
9. Xây dựng chức năng admin (dashboard, phân quyền, ...)	Tuần 7
10. Kiểm thử và sửa lỗi	Tuần 8
11. Triển khai website lên Vercel	Tuần 8

b. Phân chia công việc

Nguyễn Nhật Minh: Thiết kế, xây dựng frontend, hỗ trợ xây dựng backend.

Hà Trần Thế Anh: Thiết kế, xây dựng frontend, hỗ trợ xây dựng backend.

Trần Thị Hiền: Thiết kế, xây dựng backend, CSDL. Hỗ trợ xây dựng frontend.

Các công việc chung: Lập kế hoạch và phân tích yêu cầu, tìm hiểu công nghệ sử dụng, tính năng chat, kiểm thử, sửa lỗi, triển khai trên nền tảng Vercel.

Quản lý mã nguồn: Github

Nền tảng triển khai online: Vercel