**Tài liệu triển khai**

# **Website với CMS vs E-commerce**

## **1.1 CMS (Hệ quản trị nội dung)**

**CMS** là viết tắt của **Content Management System** (Hệ quản trị nội dung). Đây là một phần mềm hoặc nền tảng cho phép người dùng tạo, quản lý và xuất bản nội dung trên website mà không cần kiến thức về lập trình. CMS cung cấp một giao diện dễ sử dụng, giúp quản trị viên có thể thêm, sửa, xóa nội dung, quản lý người dùng, và nhiều chức năng khác mà không cần phải viết mã.

## **1.2 Các tính năng chính của CMS:**

* **Quản lý nội dung:** Tạo và chỉnh sửa bài viết, trang, hình ảnh, video, và các loại nội dung khác.
* **Giao diện người dùng thân thiện:** Cung cấp giao diện trực quan, dễ sử dụng với các công cụ soạn thảo WYSIWYG (What You See Is What You Get).
* **Quản lý người dùng:** Thiết lập các quyền hạn và vai trò khác nhau cho người dùng như quản trị viên, biên tập viên, và tác giả.
* **Mở rộng tính năng:** Hỗ trợ các plugin và module để mở rộng tính năng, bao gồm SEO, bảo mật, và nhiều tính năng khác.
* **Quản lý tài nguyên:** Hệ thống quản lý tập tin, hình ảnh, và tài liệu.

**Ví dụ về CMS:** WordPress, Joomla, Drupal, và Magento.

## **1.3 WordPress**

### **1.3.1 Giới thiệu**

WordPress là một hệ quản trị nội dung (CMS) phổ biến, và XAMPP là một gói phần mềm miễn phí bao gồm Apache, MySQL, PHP, và Perl, giúp dễ dàng thiết lập môi trường máy chủ cục bộ trên máy tính của bạn. Hướng dẫn này sẽ mô tả chi tiết cách cài đặt và cấu hình WordPress sử dụng XAMPP.

### **1.3.2 Tải và giải nén WordPress**

1. Truy cập trang tải WordPress.
2. Tải xuống phiên bản mới nhất.
3. Giải nén tập tin tải về vào thư mục web server (ví dụ: /var/www/html trên Linux hoặc C:\xampp\htdocs trên Windows).

### **1.3.3 Cấu hình xampp**

A screenshot of a computer

Description automatically generated

### **1.3.4 Tạo cơ sở dữ liệu**

* + - Mở trình duyệt và truy cập http://localhost:8089/phpmyadmin/.
    - Nhấp vào New để tạo một cơ sở dữ liệu mới.
    - Đặt tên cơ sở dữ liệu (ví dụ: wordpress) và nhấn Create.

### **1.3.5 Cấu hình WordPress**

1. Mở trình duyệt và truy cập http://localhost/wordpress/.
2. Chọn ngôn ngữ và nhấn Continue.
3. Nhấn Let's go! để bắt đầu cấu hình.
4. Nhập thông tin cơ sở dữ liệu:
   * Database Name: wordpress
   * Username: root
   * Password: (để trống, nếu bạn chưa thiết lập mật khẩu cho MySQL)
   * Database Host: localhost
   * Table Prefix: wp\_
5. Nhấn Submit, sau đó nhấn Run the installation.

# **2. E-commerce (Thương mại điện tử)**

**E-commerce** là viết tắt của **Electronic Commerce** (Thương mại điện tử). Đây là việc mua bán hàng hóa và dịch vụ qua Internet. E-commerce bao gồm việc sử dụng các công nghệ mạng để thực hiện các giao dịch mua bán, thanh toán trực tuyến, và quản lý kho hàng. Mục tiêu của E-commerce là tạo ra một môi trường kinh doanh trực tuyến thuận tiện, an toàn và hiệu quả.

## **2.1 Các tính năng chính của E-commerce:**

* **Quản lý sản phẩm:** Thêm, sửa, xóa sản phẩm, quản lý danh mục, và các thuộc tính sản phẩm như giá, mô tả, và hình ảnh.
* **Giỏ hàng và thanh toán:** Tích hợp các giải pháp thanh toán trực tuyến như PayPal, Stripe, và các cổng thanh toán khác để khách hàng có thể mua hàng một cách dễ dàng.
* **Quản lý đơn hàng:** Theo dõi trạng thái đơn hàng, xử lý thanh toán, và quản lý vận chuyển.
* **Tùy chỉnh giao diện:** Cho phép thiết kế cửa hàng trực tuyến đẹp mắt và dễ sử dụng, cải thiện trải nghiệm người dùng.
* **Báo cáo và phân tích:** Cung cấp các công cụ phân tích và báo cáo doanh thu, khách hàng, và các chỉ số kinh doanh khác.

**Ví dụ về các nền tảng E-commerce:** WooCommerce, Shopify, Magento, và BigCommerce.

## **2.2 Cài đặt WooCommerce**

### **2.2.1. Cài đặt plugin WooCommerce**

1. Đăng nhập vào bảng điều khiển WordPress.
2. Truy cập Plugins > Add New.
3. Tìm kiếm WooCommerce và nhấn Install Now.
4. Sau khi cài đặt, nhấn Activate.

### **2.2.2. Thiết lập WooCommerce**

1. Sau khi kích hoạt WooCommerce, hệ thống sẽ đưa bạn qua các bước cài đặt.

### Thiết lập các thông tin cơ bản như vị trí cửa hàng, đơn vị tiền tệ, và phương thức thanh toán.

### **2.2.3. Cấu hình cửa hàng trực tuyến**

#### **2.2.3.1. Thêm sản phẩm**

1. Truy cập Products > Add New.
2. Nhập thông tin sản phẩm bao gồm tên, mô tả, giá, và hình ảnh.
3. Chọn danh mục sản phẩm và nhấn Publish.

#### **2.2.3.2. Cấu hình phương thức thanh toán**

1. Truy cập WooCommerce > Settings > Payments.
2. Kích hoạt và cấu hình các phương thức thanh toán như PayPal, Stripe, hoặc thanh toán khi nhận hàng.

#### **2.2.3.3 Cấu hình phương thức vận chuyển**

1. Truy cập WooCommerce > Settings > Shipping.
2. Thêm các khu vực vận chuyển và thiết lập phí vận chuyển tương ứng.

#### **2.2.3.4. Tùy chỉnh giao diện**

1. Truy cập Appearance > Themes.
2. Chọn và cài đặt một giao diện phù hợp với cửa hàng của bạn.
3. Tùy chỉnh giao diện thông qua Customize.

# **3. Tài liệu Triển khai Nginx trên Windows**

## **3.1 Giới thiệu**

Nginx là một web server hiệu suất cao được sử dụng phổ biến cho các trang web lớn và ứng dụng web. Mặc dù Nginx thường được triển khai trên các hệ điều hành Linux, nó cũng hỗ trợ chạy trên Windows. Hướng dẫn này sẽ chi tiết cách cài đặt và cấu hình Nginx trên hệ điều hành Windows.

## **3.2 Tải và cài đặt Nginx**

### **3.3.1. Tải Nginx**

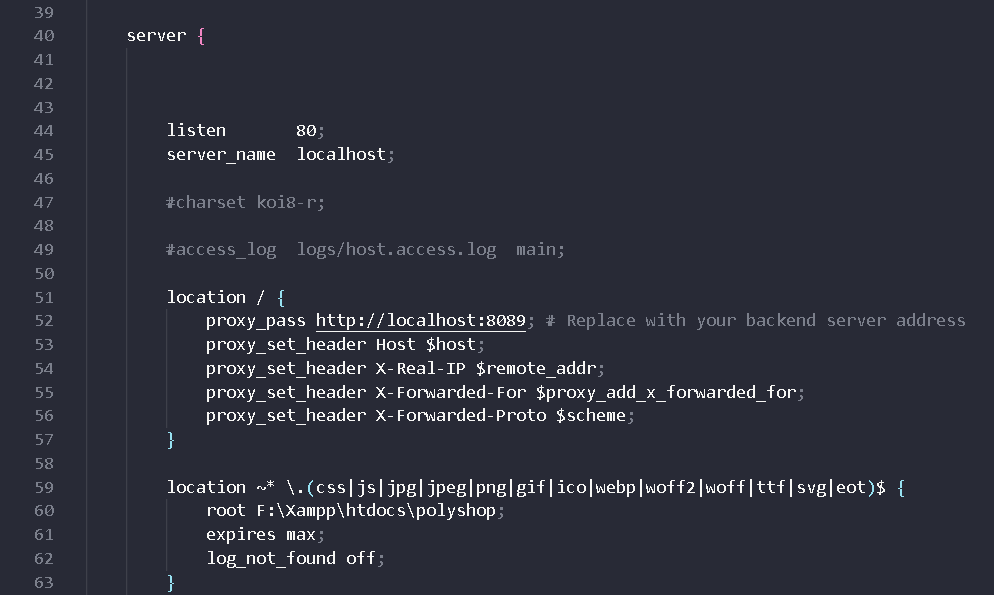
1. Truy cập trang tải Nginx chính thức.
2. Tìm phần "Stable version" và nhấp vào liên kết để tải xuống tệp nginx-<phiên bản>-win32.zip hoặc nginx-<phiên bản>-win64.zip tùy thuộc vào kiến trúc hệ điều hành của bạn.

### **3.3.2. Giải nén Nginx**

1. Giải nén tệp zip đã tải về vào thư mục bạn muốn cài đặt Nginx (ví dụ: C:\nginx).

## **3.3. Cấu hình Nginx**

1. Truy cập vào thư mục C:\tools\nginx.
2. Mở tập tin conf\nginx.conf bằng trình soạn thảo văn bản (ví dụ: Notepad).

****

**Giải thích:**

1. **server { ... }**
   * Định nghĩa một khối server mới trong cấu hình Nginx.
2. **listen 80;**
   * Lắng nghe trên cổng 80, cổng mặc định cho HTTP.
3. **server\_name localhost;**
   * Xác định tên server mà Nginx sẽ phản hồi. Trong trường hợp này, nó là localhost.
4. **location / { ... }**
   * Định nghĩa một khối location cho đường dẫn gốc (/). Tất cả các yêu cầu đến đường dẫn gốc sẽ được xử lý bởi khối này.
5. **proxy\_pass** [**http://localhost:8089**](http://localhost:8089)**;**
   * Chuyển tiếp (proxy) các yêu cầu đến server backend chạy trên localhost cổng 8089.
6. **proxy\_set\_header Host $host;**
   * Thiết lập header Host với giá trị của biến $host (tên host của yêu cầu gốc).
7. **proxy\_set\_header X-Real-IP $remote\_addr;**
   * Thiết lập header X-Real-IP với địa chỉ IP thực của client.
8. **proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**
   * Thiết lập header X-Forwarded-For với danh sách các địa chỉ IP của các proxy mà yêu cầu đã đi qua, cùng với địa chỉ IP của client thực.
9. **proxy\_set\_header X-Forwarded-Proto $scheme;**
   * Thiết lập header X-Forwarded-Proto với giao thức được sử dụng (HTTP hoặc HTTPS).
10. ***location ~ .(css|js|jpg|jpeg|png|gif|ico|webp|woff2|woff|ttf|svg|eot)$ { ... }*\***
    * Định nghĩa một khối location cho các file tĩnh có đuôi mở rộng là .css, .js, .jpg, .jpeg, .png, .gif, .ico, .webp, .woff2, .woff, .ttf, .svg, hoặc .eot.
11. **root F:\Xampp\htdocs\polyshop;**
    * Chỉ định thư mục gốc nơi các file tĩnh được lưu trữ. Trong trường hợp này là F:\Xampp\htdocs\polyshop.
12. **expires max;**
    * Thiết lập thời gian hết hạn cho các file tĩnh ở mức tối đa, giúp cải thiện hiệu suất bằng cách cho phép trình duyệt lưu trữ cache các file này trong thời gian dài.
13. **log\_not\_found off;**
    * Tắt ghi log cho các yêu cầu không tìm thấy (404) đối với các file tĩnh. Điều này giúp giảm tải ghi log cho Nginx.

# **4. Mô tả về CI/CD**

## **4.1 Giới thiệu**

CI/CD là viết tắt của Continuous Integration (Tích hợp liên tục) và Continuous Deployment/Delivery (Triển khai/Phân phối liên tục). Đây là một tập hợp các phương pháp và công cụ được sử dụng trong phát triển phần mềm nhằm tự động hóa các bước từ việc viết mã cho đến khi triển khai phần mềm. CI/CD giúp tăng cường hiệu suất, độ tin cậy và chất lượng của phần mềm thông qua các quy trình tự động và liên tục.

### **4.1.1 Continuous Integration (CI)**

**Continuous Integration (CI)** là một phương pháp trong đó các lập trình viên thường xuyên tích hợp mã nguồn của họ vào một nhánh chính của kho mã. Mục tiêu chính của CI là phát hiện lỗi sớm nhất có thể bằng cách tích hợp thường xuyên và chạy các kiểm tra tự động (test).

**Các bước chính trong CI:**

1. **Commit thường xuyên:** Lập trình viên commit mã thường xuyên vào kho mã chung (thường là Git).
2. **Xây dựng tự động:** Một hệ thống CI sẽ tự động kiểm tra, xây dựng và chạy các test mỗi khi có commit mới.
3. **Phản hồi nhanh chóng:** Nếu có lỗi, hệ thống CI sẽ thông báo ngay cho lập trình viên để sửa chữa kịp thời.

### **4.1.2 Continuous Delivery (CD)**

**Continuous Delivery (CD)** là phương pháp mở rộng của CI, trong đó phần mềm được xây dựng, kiểm tra và chuẩn bị để triển khai vào môi trường sản xuất bất kỳ lúc nào. CD đảm bảo rằng phần mềm luôn ở trạng thái sẵn sàng triển khai, nhưng việc triển khai vẫn được thực hiện thủ công.

### **4.1.3 Các công cụ CI/CD phổ biến**

1. **Jenkins:** Một trong những công cụ CI/CD phổ biến nhất, mã nguồn mở và rất linh hoạt.
2. **GitLab CI/CD:** Tích hợp sẵn trong GitLab, cung cấp giải pháp CI/CD toàn diện.
3. **CircleCI:** Công cụ CI/CD mạnh mẽ và dễ sử dụng, hỗ trợ tích hợp với nhiều nền tảng.
4. **Travis CI:** Công cụ CI/CD phổ biến, đặc biệt được ưa chuộng trong các dự án mã nguồn mở.
5. **Azure DevOps:** Dịch vụ CI/CD của Microsoft, tích hợp tốt với các công cụ của Microsoft và nhiều công cụ khác.

# **4.2 Tài liệu Triển khai CI/CD bằng GitHub Actions**

## **4.2.1 Giới thiệu**

GitHub Actions là một nền tảng CI/CD mạnh mẽ và linh hoạt được tích hợp trực tiếp vào GitHub. Nó cho phép bạn tự động hóa các quy trình xây dựng, kiểm tra và triển khai phần mềm của mình mỗi khi có thay đổi mã nguồn.

## **4.2.2 Cấu hình CI/CD bằng GitHub Actions**

#### **4.2.2.1 Tạo file workflow**

1. Trong repository của bạn, tạo một thư mục .github/workflows nếu chưa có.
2. Trong thư mục workflows, tạo một file mới với phần mở rộng .yml hoặc .yaml. Ví dụ, ci-cd-pipeline.yml.

#### **4.2.2.2 Cấu trúc cơ bản của file workflow**

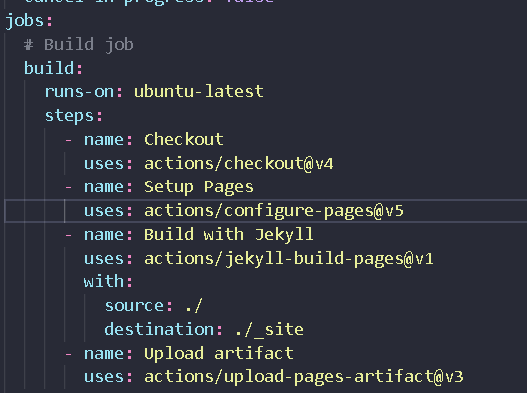
File workflow sử dụng định dạng YAML. Dưới đây là cấu trúc cơ bản của một file workflow:

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **name:** Tên của workflow, được hiển thị trong tab Actions.
2. **on:**
   * **push:** Kích hoạt workflow khi có thay đổi được push lên nhánh main.
   * **workflow\_dispatch:** Cho phép kích hoạt workflow thủ công từ tab Actions trên GitHub.
3. **permissions:** Thiết lập quyền cho GITHUB\_TOKEN:
   * contents: read: Quyền đọc nội dung repository.
   * pages: write: Quyền ghi lên GitHub Pages.
   * id-token: write: Quyền ghi token định danh.
4. **concurrency:**
   * **group:** Nhóm các lần chạy workflow vào một nhóm có tên "pages".
   * **cancel-in-progress:** false nghĩa là không hủy bỏ các lần chạy đang tiến hành, nhằm đảm bảo các lần triển khai sản xuất được hoàn thành.



**Giải thích:**

1. **jobs:** Chứa các công việc sẽ được thực hiện trong workflow.
2. **build:** Định nghĩa công việc xây dựng trang web.
   * **runs-on:** Xác định hệ điều hành của máy chủ runner, ở đây là Ubuntu mới nhất (ubuntu-latest).
   * **steps:** Các bước thực hiện trong công việc build:
     + **Checkout:** Sử dụng action actions/checkout@v4 để lấy mã nguồn từ repository.
     + **Setup Pages:** Sử dụng action actions/configure-pages@v5 để thiết lập GitHub Pages.
     + **Build with Jekyll:** Sử dụng action actions/jekyll-build-pages@v1 để xây dựng trang Jekyll từ thư mục nguồn (./) và lưu kết quả vào thư mục đích (./\_site).
     + **Upload artifact:** Sử dụng action actions/upload-pages-artifact@v3 để tải lên kết quả xây dựng (artifact) để sử dụng trong công việc tiếp theo.

A screenshot of a computer program

Description automatically generated

**Giải thích:**

1. **deploy:** Định nghĩa công việc triển khai trang web.
   * **environment:** Thiết lập môi trường triển khai:
     + **name:** Tên của môi trường, ở đây là github-pages.
     + **url:** URL của trang được triển khai, lấy từ output của bước triển khai (${{ steps.deployment.outputs.page\_url }}).
   * **runs-on:** Xác định hệ điều hành của máy chủ runner, ở đây là Ubuntu mới nhất (ubuntu-latest).
   * **needs:** Xác định rằng công việc deploy cần hoàn thành công việc build trước khi chạy.
   * **steps:** Các bước thực hiện trong công việc deploy:
     + **Deploy to GitHub Pages:** Sử dụng action actions/deploy-pages@v4 để triển khai kết quả xây dựng lên GitHub Pages. Bước này có id là deployment để sử dụng cho việc lấy output URL.