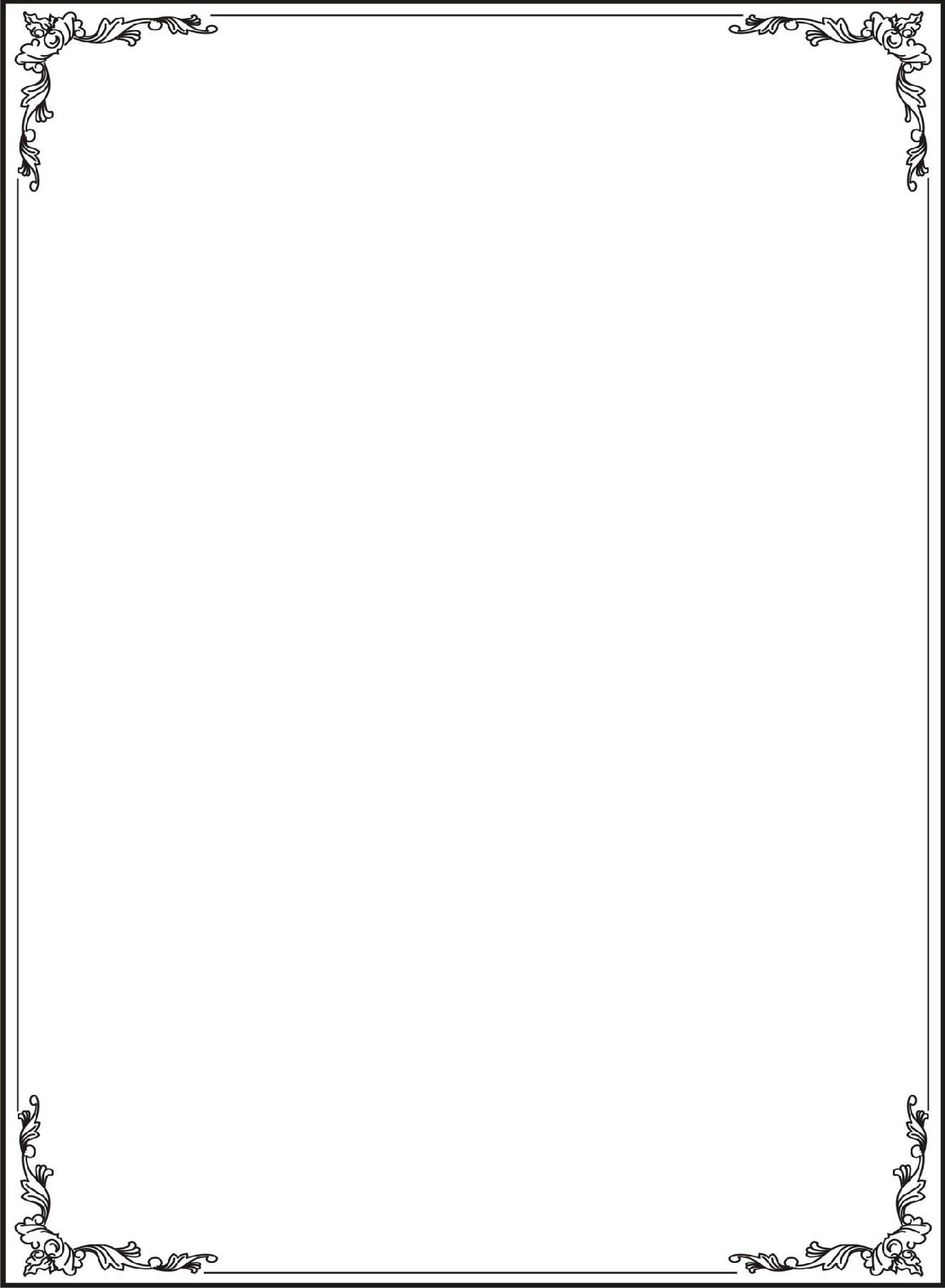
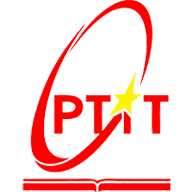
**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

****

**CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI: TRIỂN KHAI PHẦN MỀM**

**Giảng viên hướng dẫn : ThS. Lê Hà Thanh**

**Lớp : D20CQCNPM01-N**

**Sinh viên thực hiện : Phạm Ngọc Bảo – N20DCCN006**

**Hồ Chí Minh, ngày 20 tháng 06 năm 2024**

**Mục lục**

[I. Triển khai phần mềm 1](#_Toc169623730)

[1. Giới thiệu 1](#_Toc169623731)

[2. Yêu cầu hệ thống 1](#_Toc169623732)

[2.1 Phần cứng 1](#_Toc169623733)

[2.2 Phần mềm 1](#_Toc169623734)

[3. Chuẩn bị 1](#_Toc169623735)

[4. Thông số thiết lập server 4](#_Toc169623736)

[5. Thiết lập cấu hình Reverse Proxy 4](#_Toc169623737)

[6. Triển khai ứng dụng 5](#_Toc169623738)

[7. Bảo trì và Hỗ trợ 10](#_Toc169623739)

[8. Ghi Chú 11](#_Toc169623740)

[II. Continuous Integration and Continous Deployment (CI/CD) 12](#_Toc169623741)

[1. CI/CD là gì? 12](#_Toc169623742)

[2. Jenkins là gì? 12](#_Toc169623743)

[3. Thực thi 13](#_Toc169623744)

[III. Kết luận 20](#_Toc169623745)

# Triển khai phần mềm

## Giới thiệu

Tài liệu này hướng dẫn triển khai và cài đặt một website gồm hai thành phần: CMS (Content Management System) và E-Commerce mã nguồn mở. Các công cụ và công nghệ sử dụng bao gồm cơ chế Reverse Proxy, Jenkins.

Tên phần mềm: Trang web bán nhạc cụ BShop

Phiên bản: 1.0.0

Ngày phát hành: 20/06/2024

Người soạn thảo: Phạm Ngọc Bảo

Email liên hệ: phamngocbao2104@gmail.com

## Yêu cầu hệ thống

### 2.1 Phần cứng

- Bộ xử lý: Ryzen 4800H hoặc cao hơn

- Ram: tối thiểu 8GB

- Ổ cứng: Tối thiểu 20GB dung lượng trống

- Mạng: Kết nối internet ổn định

### 2.2 Phần mềm

* Máy chủ: Window.
* Phần mềm: Docker, Nginx, MySQL.
* Công cụ CI/CD: Jenkins.
* Phần mềm CMS: WordPress.
* E-Commerce: WooCommerce.

## Chuẩn bị

a. Cài đặt Docker

 Tải Docker Desktop:

Truy cập trang web chính thức của Docker tại https://www.docker.com/products/docker-desktop.

Tải phiên bản Docker Desktop dành cho Windows.

 Cài đặt Docker Desktop:

Mở file cài đặt bạn vừa tải về.

Làm theo hướng dẫn trên màn hình để hoàn tất quá trình cài đặt.

Sau khi cài đặt, khởi động Docker Desktop. Docker sẽ yêu cầu bạn đăng nhập hoặc tạo tài khoản Docker (nếu bạn chưa có).

 Kiểm tra Docker:

Mở Command Prompt hoặc PowerShell.

Gõ lệnh docker --version để kiểm tra xem Docker đã được cài đặt thành công hay chưa.

b. Cài đặt Jenkins

- Bước 1: Cài đặt Java

Jenkins yêu cầu Java để chạy, vì vậy bạn cần cài đặt Java Development Kit (JDK) trước.

**+ Tải JDK**:

Truy cập trang web của Oracle hoặc OpenJDK để tải bản cài đặt JDK. Bạn có thể tải OpenJDK tại [AdoptOpenJDK](https://adoptopenjdk.net/).

**+ Cài đặt JDK**:

Chạy file cài đặt JDK bạn vừa tải về và làm theo hướng dẫn để cài đặt.

Sau khi cài đặt, hãy thêm biến môi trường JAVA\_HOME trỏ đến thư mục cài đặt JDK. Ví dụ: C:\Program Files\Java\jdk-xx.

**+ Kiểm tra cài đặt Java**:

Mở Command Prompt và chạy lệnh sau để kiểm tra phiên bản Java:

java -version

- Bước 2: Tải Jenkins

**+ Tải Jenkins**:

Truy cập trang web chính thức của Jenkins tại https://www.jenkins.io/download/.

Chọn phiên bản Windows và tải file .msi.

- Bước 3: Cài đặt Jenkins

**+ Chạy file cài đặt Jenkins**:

Mở file .msi bạn vừa tải về và làm theo hướng dẫn trên màn hình để cài đặt Jenkins.

Trong quá trình cài đặt, bạn sẽ được yêu cầu nhập đường dẫn đến thư mục JDK (JAVA\_HOME).

**+ Hoàn tất cài đặt**:

Sau khi cài đặt hoàn tất, Jenkins sẽ tự động khởi động dưới dạng một dịch vụ Windows.

c. Cài đặt Visual Studio Code

- Bước 1: Tải xuống Visual Studio Code

**+ Truy cập trang web chính thức của Visual Studio Code**:

Địa chỉ trang web: <https://code.visualstudio.com/>

Trang web này cung cấp phiên bản VS Code cho Windows, macOS và Linux. Chọn phiên bản phù hợp với hệ điều hành của bạn (Windows trong trường hợp này).

**+ Tải xuống bản cài đặt**:

Sau khi trang web hiển thị, bạn sẽ thấy nút "Download for Windows" (hoặc tương tự). Nhấn vào nút này để tải xuống file cài đặt của Visual Studio Code (.exe).

- Bước 2: Cài đặt Visual Studio Code

**+ Chạy file cài đặt**:

Mở file VSCodeSetup-xxx.exe (trong đó xxx là số phiên bản) mà bạn đã tải xuống.

Hộp thoại cài đặt sẽ hiển thị. Nhấn "Next" để tiếp tục.

**+ Chọn tùy chọn cài đặt**:

Chọn các tùy chọn cài đặt mà bạn muốn, ví dụ như tạo biểu tượng trên Desktop hay thêm vào PATH (nếu có tùy chọn này).

Nhấn "Next" để tiếp tục.

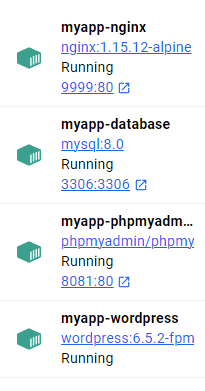
**+ Cài đặt**:

Chờ đợi quá trình cài đặt hoàn tất. Điều này có thể mất vài phút tùy vào tốc độ của máy tính của bạn.

**+ Hoàn tất cài đặt**:

Khi cài đặt hoàn tất, bạn có thể chọn "Finish" để đóng hộp thoại cài đặt.

## Thông số thiết lập server



## Thiết lập cấu hình Reverse Proxy

server{

    listen 80 default\_server;

    root /var/www/html;

    location / {

        index index.php index.html;

    }

    location ~ \.php$ {

      try\_files $uri =404;

      fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

      fastcgi\_pass wordpress:9000;

      fastcgi\_index index.php;

      include fastcgi\_params;

      fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

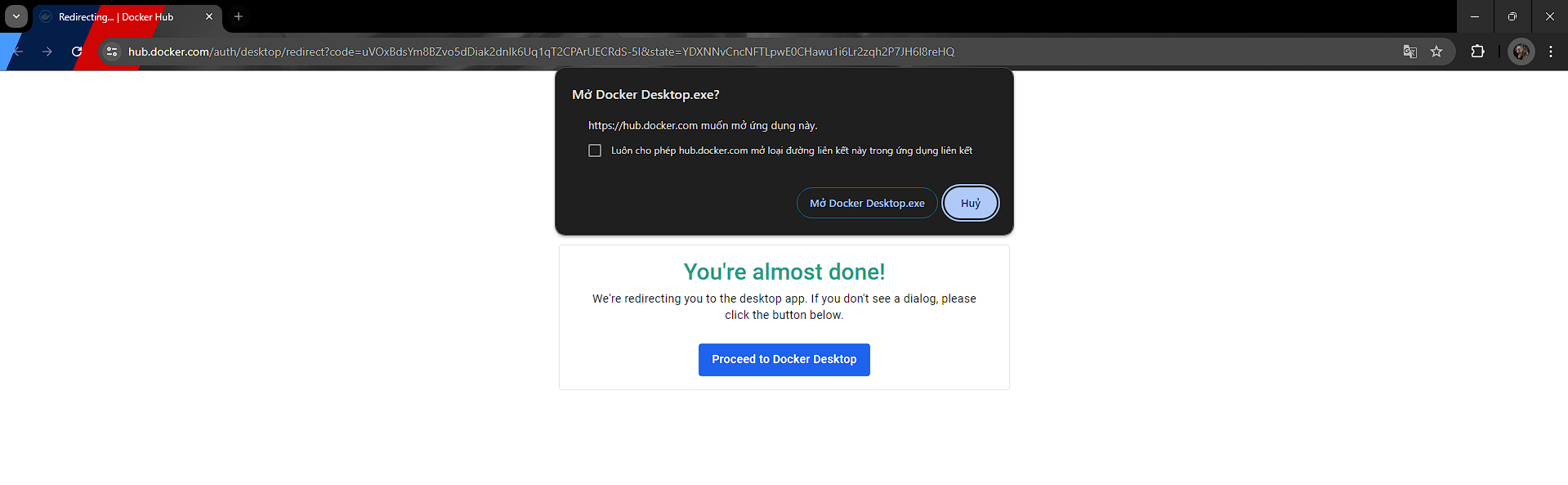
      fastcgi\_param PATH\_INFO $fastcgi\_path\_info;

  }

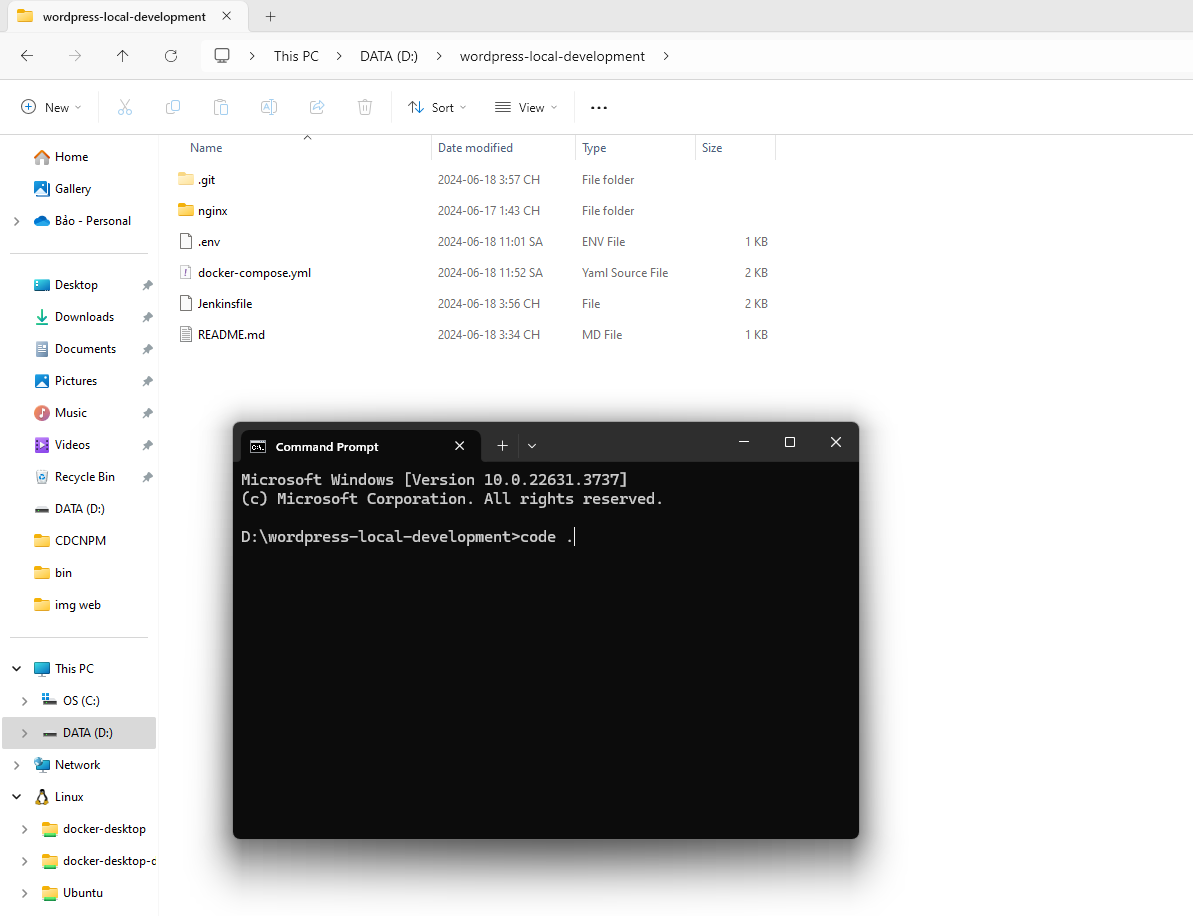
}

## Triển khai ứng dụng

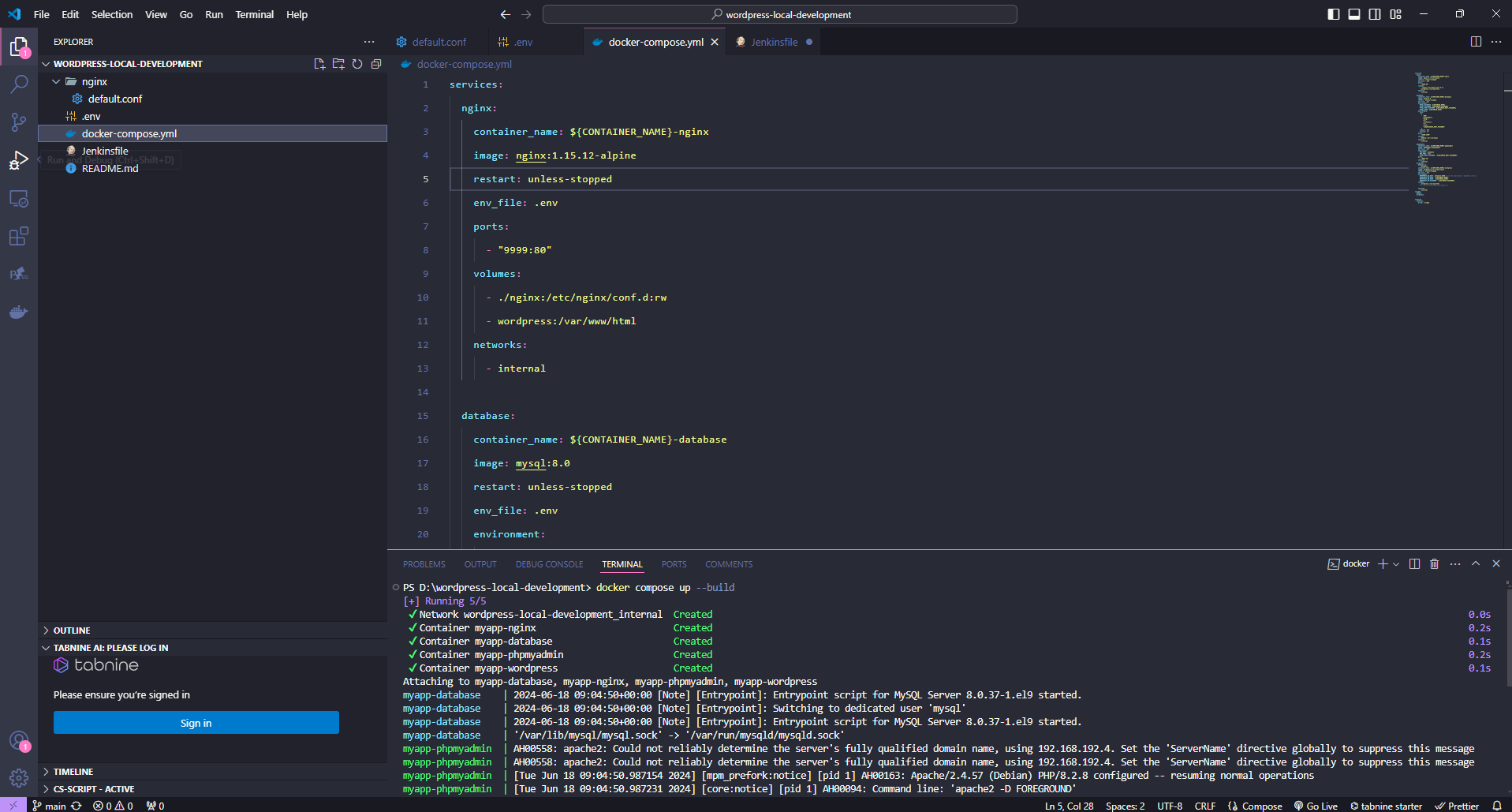
* 1. Khởi chạy Docker và đăng nhập



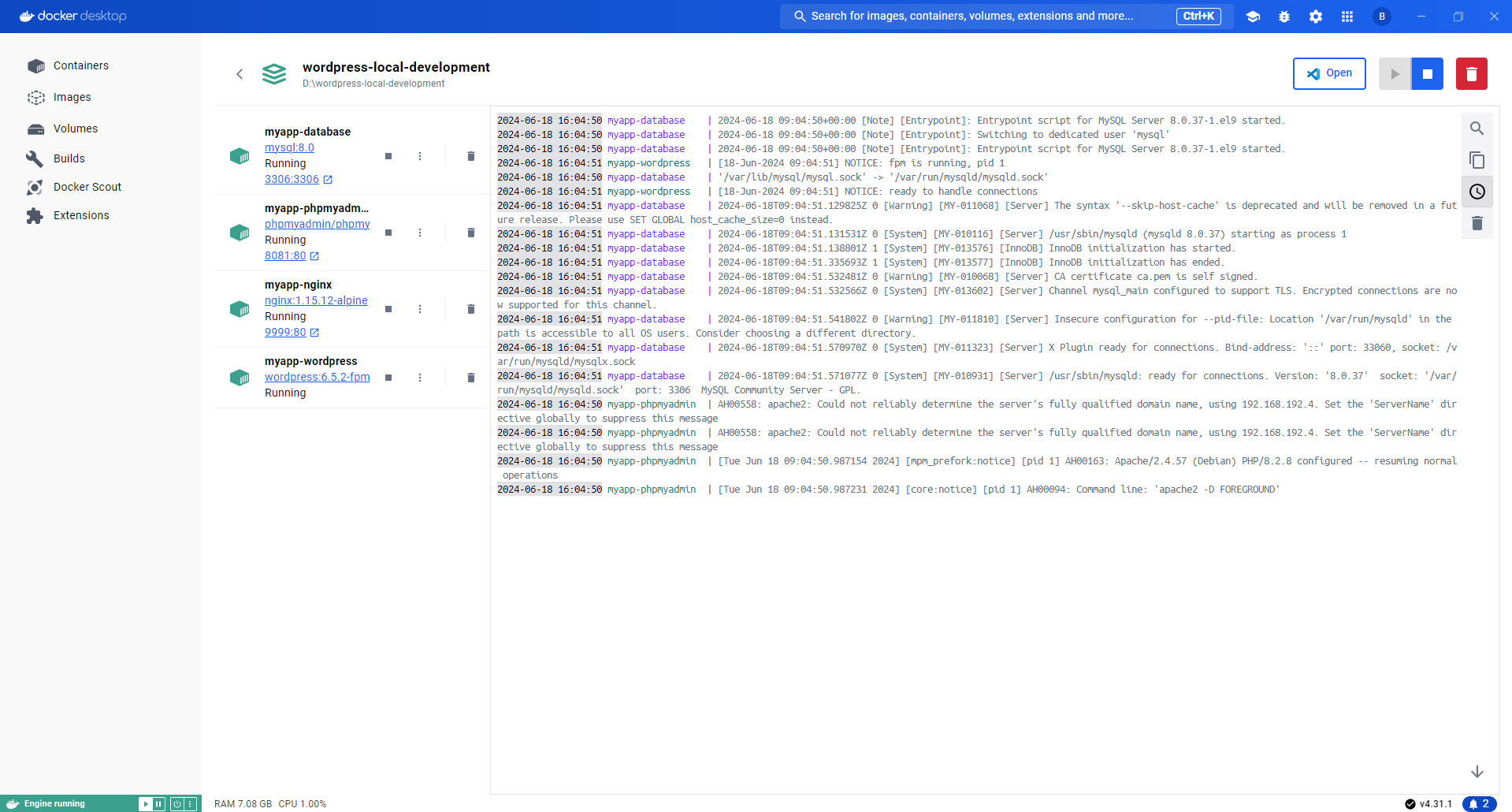
* 1. Tải mã nguồn từ repository (tải file .zip) và giải nén: <https://github.com/BaoPham76/wordpress-local-development.git>
  2. Vào thư mục wordpress-local-development vừa giải nén và nháy chuột phải chọn Open with Terminal
  3. Chạy lệnh code . để mở code trong Visual Studio Code



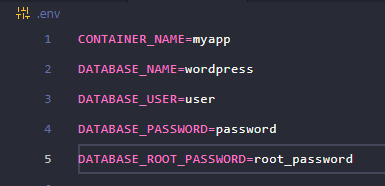
* 1. Mở Terminal trong Visual Studo Code bằng tổ hợp phím Ctrl + `. Sau đó chạy lệnh docker compose up –build để khởi động và xây dựng lại các container dựa trên cấu hình đã được định nghĩa trong tệp docker-compose.yml



* 1. Kết quả trên docker



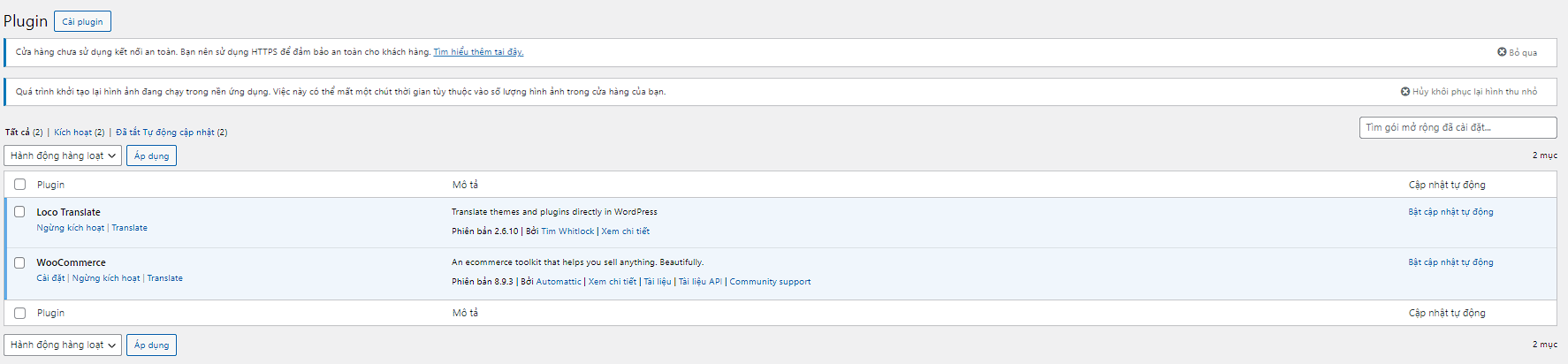
* 1. File cấu hình .env

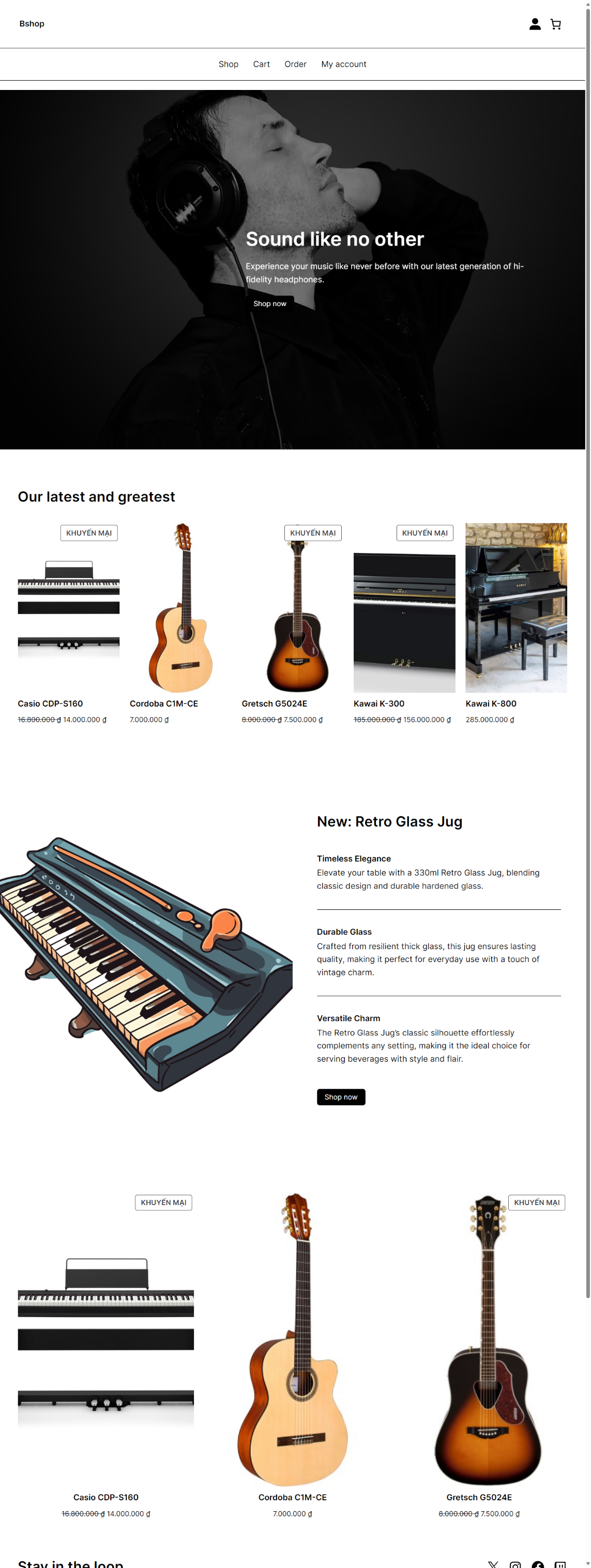


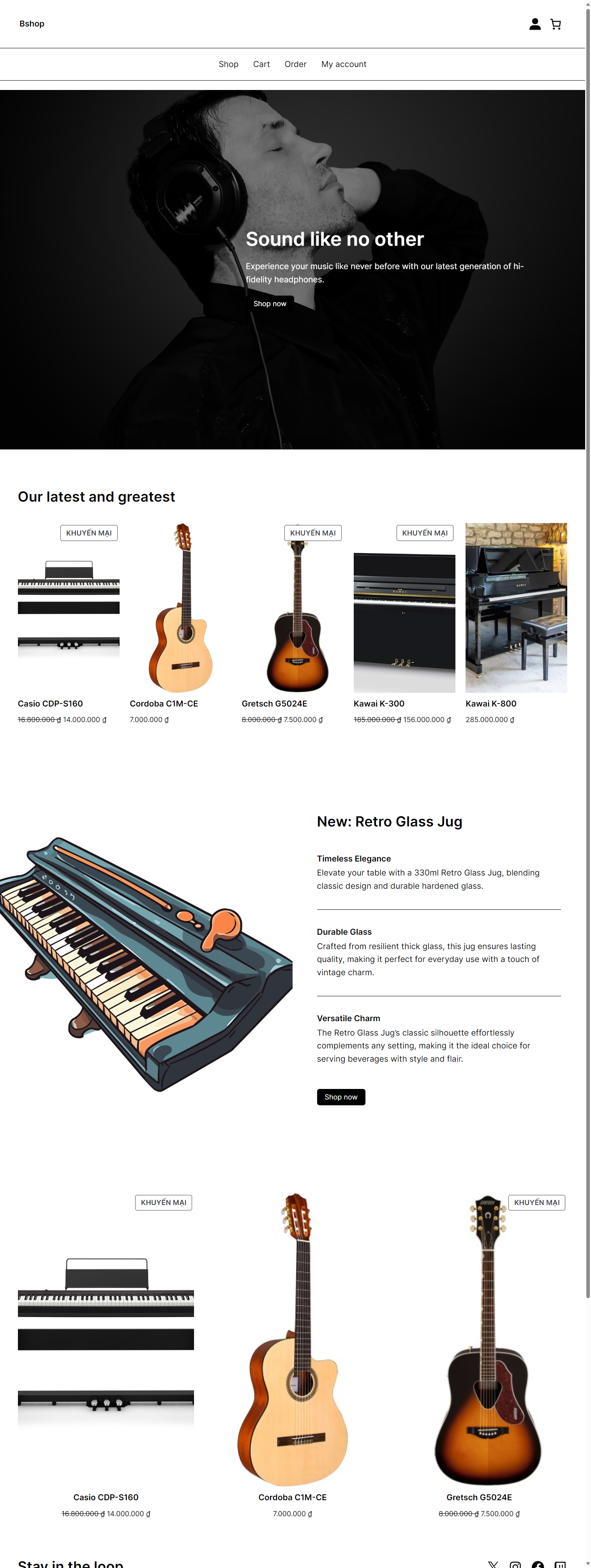
* 1. Triển khai

Bước 1: Cài đặt và cấu hình Wordpress



Bước 2: Cài đặt các plugin cần thiết  


Bước 3: Cấu hình website và chạy trang web  




## Bảo trì và Hỗ trợ

* Email hỗ trợ: [phamngocbao2104@gmail.com](mailto:phamngocbao2104@gmail.com)
* Số điện thoại hỗ trợ: 0365597196
* Thời gian hỗ trợ: 24/7

## Ghi Chú

* Đảm bảo backup dữ liệu trước khi triển khai bất kì bản cập nhật nào.
* Thường xuyên kiểm tra và cập nhật các thành phần phụ thuộc để đảm bảo tính bảo mật và hiệu năng của hệ thống.

# Continuous Integration and Continous Deployment (CI/CD)

## CI/CD là gì?

CI/CD được viết tắt của Continuous Integration / Continuous Deployment. Đó là quá trình tự động hoá việc kiểm tra ( test ) mã nguồn, xây dựng ( build ) và triển khai ( deploy ) ứng dụng của mình lên các môi trường như Dev, Staging hoặc Production.

Cách hoạt động:

Trong phần này, chúng ta sẽ giải thích chi tiết hơn về các khái niệm sau CI và CD:  
**CI (Continuous Integration):** là một quá trình tự động hóa trong phát triển phần mềm, cho phép các thành viên trong team kiểm tra và hợp nhất mã nguồn của họ vào kho chung một cách tự động và định kỳ. Mỗi lần được hợp nhất thì được xây dựng 1 cách tự động để phát hiện lỗi phát sinh 1 cách nhanh nhất có thể.

Quy trình của CI được diễn ra như sau:

* **Tích hợp mã nguồn:** Các lập trình viên commit và push code của mình lên repository.
* **Kích hoạt CI:** CI server giám sát repository và kiểm tra liên tục sự xem có sự thay đổi nào hay không. Khi phát ra sự thay đổi, CI server bắt đầu quá trình CI.
* **Build code**: Với code mới nhất trên repository, CI server sẽ build mã nguồn này thành một phần mềm hoàn chỉnh.
* **Chạy các test case:** Sau khi quá trình build hoàn tất, công cụ CI server bắt đầu chạy kiển tra đơn vị (unit test), kiểm tra tích hợp (integration test), kiểm tra giao diện (UI test) hoặc các kiểm tra khác tùy thuộc vào yêu cầu của dự án.
* **Đưa ra thông báo:** Nếu có lỗi trong quá trình kiểm tra, CI server sẽ đưa ra thông báo lỗi cho các lập trình viên biết. Các thông báo này thường được gửi qua email, chat...
* **Hợp nhất mã nguồn:** Nếu không có lỗi, CI server sẽ tự động hợp nhất mã nguồn của các lập trình viên vào một phiên bản mới nhất của phần mềm. Phiên bản mới này được lưu trữ trong kho chung của dự án và có thể được triển khai vào các môi trường khác nhau của dự án.
* **CD ( Continuos Deployment ):** Là quá trình triển khai tự động sau khi quá trình CI được diễn ra thành công. Quá trình CD sẽ được kích hoạt để triển khai ứng dụng của chúng ta vào các môi trường của dự án. Các gói phần mềm đã build thành công được triển khai bằng cách sử dụng các công cụ tự động hoặc được triển khai thủ công bằng tay. Điều này đảm bảo rằng phần mềm được cài đặt đúng cách và hoạt động tốt trên các môi trường.

## Jenkins là gì?

Jenkins là một công cụ liên tục hội nhập (continuous integration - CI) phổ biến được sử dụng trong phát triển phần mềm. Nó cho phép các nhà phát triển tự động hóa các quy trình xây dựng, kiểm thử và triển khai phần mềm. Jenkins có khả năng tích hợp với nhiều công cụ khác và cung cấp giao diện đồ họa và các plugin để mở rộng tính năng của nó. Công cụ này giúp tăng tốc độ phát triển phần mềm, cải thiện chất lượng mã và giảm thiểu các lỗi tích hợp.

## Thực thi

File script Jenkinsfile:

pipeline {

    agent any

    environment {

        CONTAINER\_NAME = "myapp"

        DATABASE\_NAME = "wordpress"

        DATABASE\_PASSWORD = "password"

        DATABASE\_ROOT\_PASSWORD = "root\_password"

        DATABASE\_USER = "user"

    }

    stages {

        stage('Checkout') {

            steps {

                // Lấy mã nguồn từ repository

                git credentialsId: '938e6ad1-045b-4eee-a581-a5c12e67a672', url: 'https://github.com/BaoPham76/wordpress-local-development.git', branch: 'main'

            }

        }

        stage('Build and Deploy') {

            steps {

                script {

                    // Xây dựng và triển khai bằng Docker Compose

                    bat 'docker-compose down'

                    // Chỉ khởi động lại dịch vụ WordPress

                    bat 'docker-compose up --build'

                }

            }

        }

        stage('Post-deploy Cleanup') {

            steps {

                // Bất kỳ bước dọn dẹp nào sau triển khai, nếu cần

                bat 'docker system prune -f'

            }

        }

    }

    post {

        always {

            // Luôn luôn lưu trữ log

            archiveArtifacts artifacts: '\*\*/logs/\*\*', allowEmptyArchive: true

        }

    }

}

**- Bước 1: Truy cập Jenkins**

 Sau khi cài đặt, Jenkins sẽ chạy trên http://localhost:8888.

 Mở trình duyệt và truy cập địa chỉ trên.

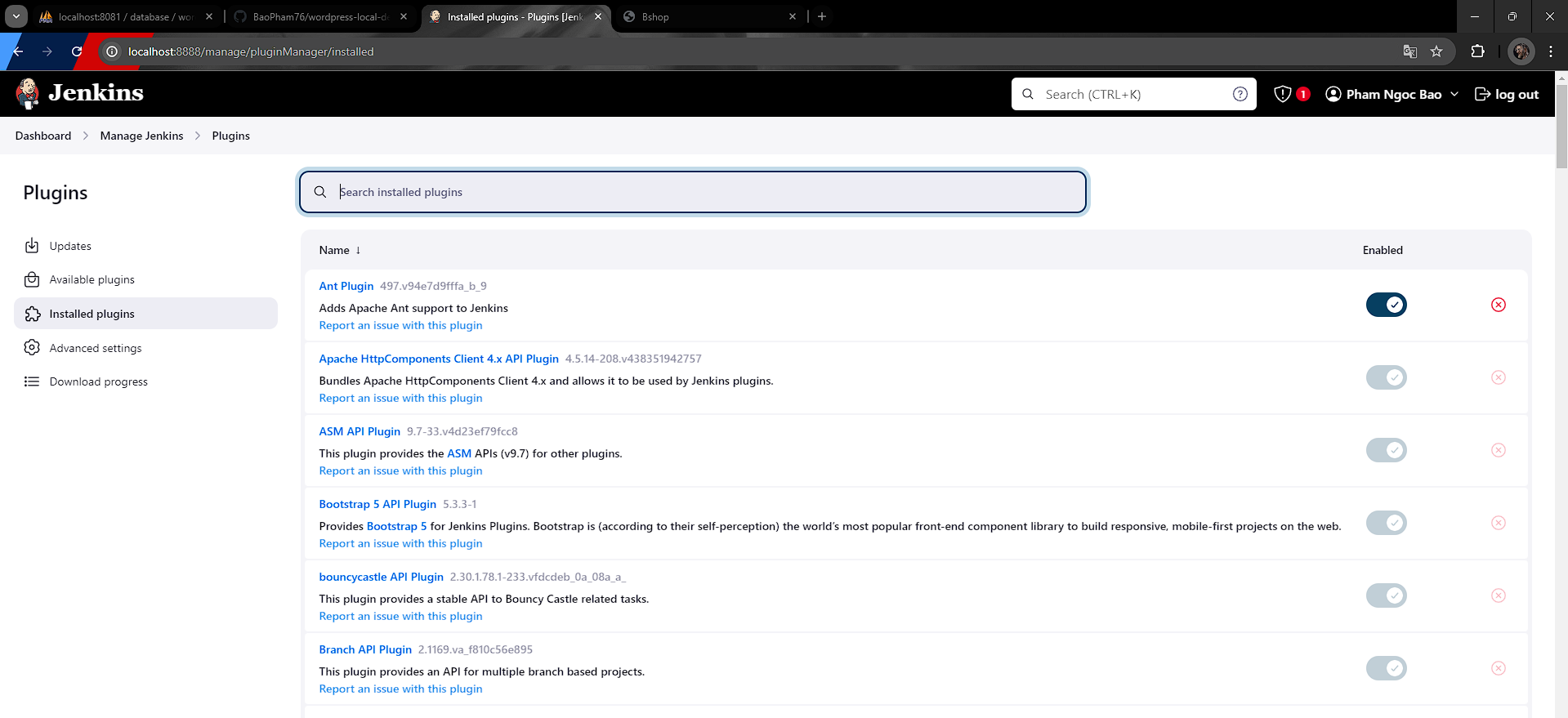
 Nhập mã unlock được cung cấp tại C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword.

 Thiết lập tài khoản admin và cài đặt các plugin đề xuất.

**- Bước 2: Cài đặt các Plugin Jenkins**

Truy cập Jenkins Dashboard, chọn "Manage Jenkins" -> "Manage Plugins".

Tìm và cài đặt các plugin sau: Docker Pipeline, Git, SSH Agent.



**- Bước 3: Thêm thông tin xác thực vào Jenkins**

 Đi đến "Manage Jenkins":

Trên Jenkins Dashboard, nhấn vào "Manage Jenkins".

 Chọn "Manage Credentials":

Trong trang "Manage Jenkins", nhấn vào "Manage Credentials".

 Thêm thông tin xác thực mới:

Chọn "Jenkins" (hoặc domain thích hợp nếu bạn đã tạo các domain).

Chọn "Global credentials (unrestricted)".

Nhấn "Add Credentials".

 Điền thông tin xác thực:

Kind: Chọn loại thông tin xác thực phù hợp (ví dụ: "Username with password", "SSH Username with private key", "Secret text").

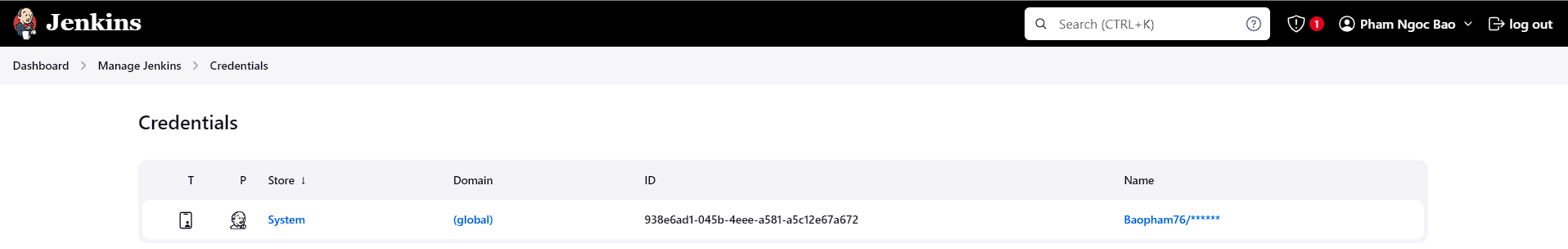
Username: Tên người dùng GitHub của bạn (nếu cần).

Password: Mật khẩu GitHub của bạn hoặc token cá nhân (personal access token).

ID: Bạn có thể để trống để Jenkins tự tạo ID hoặc bạn có thể điền vào một ID dễ nhớ (ví dụ: github-credentials).

Description: Mô tả thông tin xác thực để dễ nhận biết sau này.

Nhấn "OK" để lưu lại thông tin xác thực.



**- Bước 4: Thiết lập Project trên Jenkins**

a. Tạo một Pipeline Job

Truy cập Jenkins Dashboard và chọn "New Item".

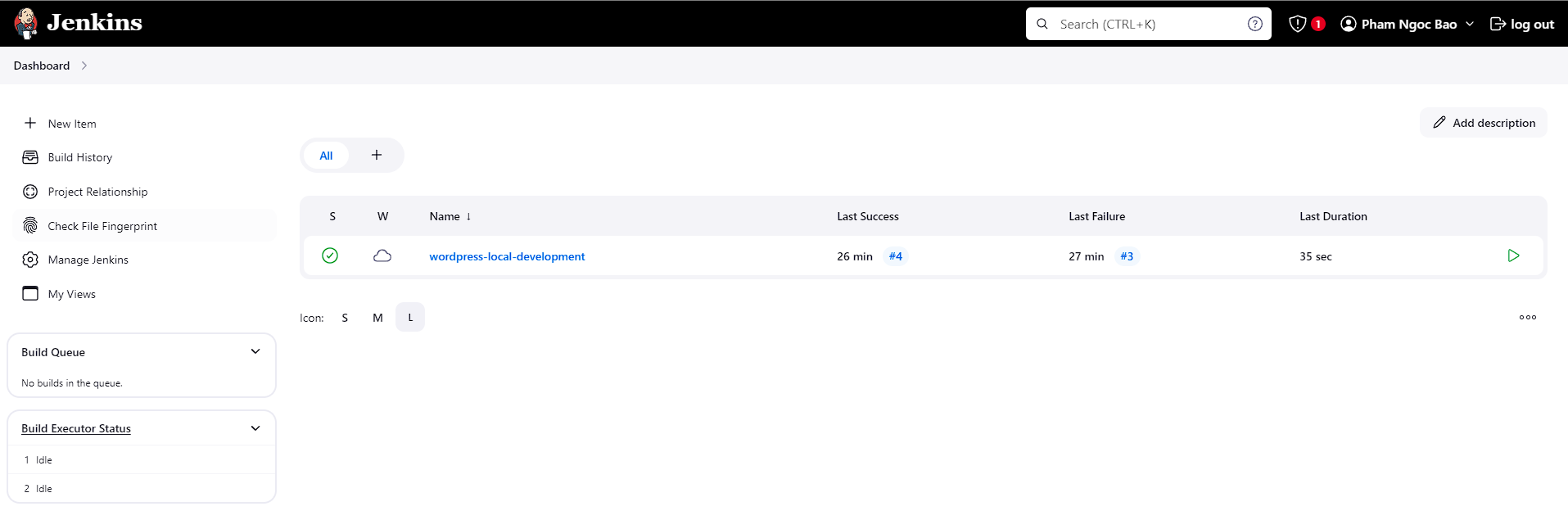
Đặt tên cho Job là “wordpress-local-development” và chọn "Pipeline".

b. Cấu hình Pipeline Job

Trong phần "Pipeline", chọn "Pipeline script from SCM".

Chọn Git và nhập URL của repository Git <https://github.com/BaoPham76/wordpress-local-development.git>

Cấu hình thông tin xác thực (credentials).



**- Bước 5: Viết Pipeline Script**

Trong repository, tạo một file Jenkinsfile để định nghĩa các bước CI/CD.

Sau khi tạo thông tin xác thực, bạn sẽ có một ID (credentialsId) để sử dụng trong Jenkinsfile.

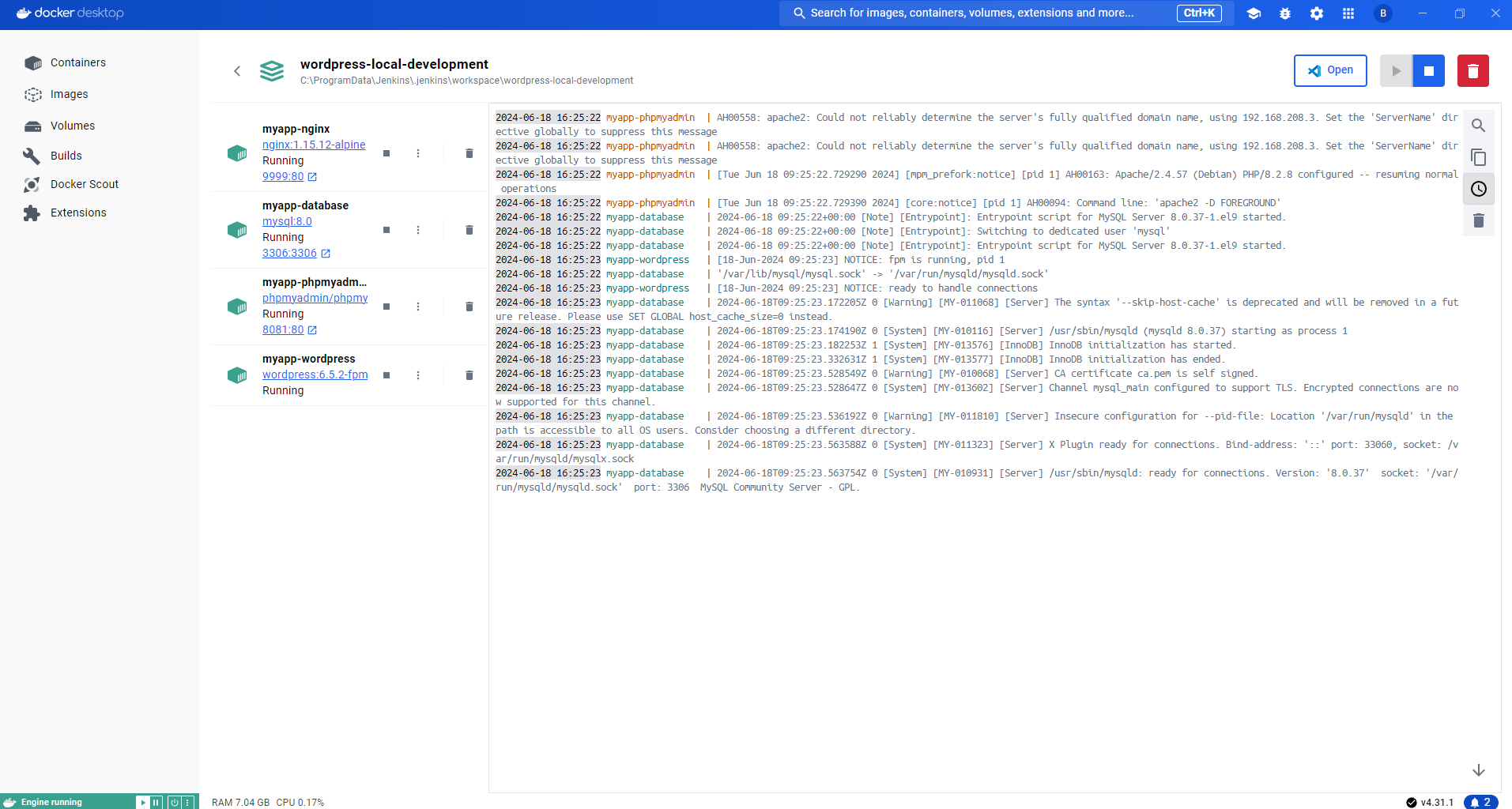
**- Bước 6: Chạy và kiểm tra Pipeline**

**Kích hoạt Job trong Jenkins**:

Truy cập Jenkins Dashboard, chọn Job của bạn và nhấn "Build Now".

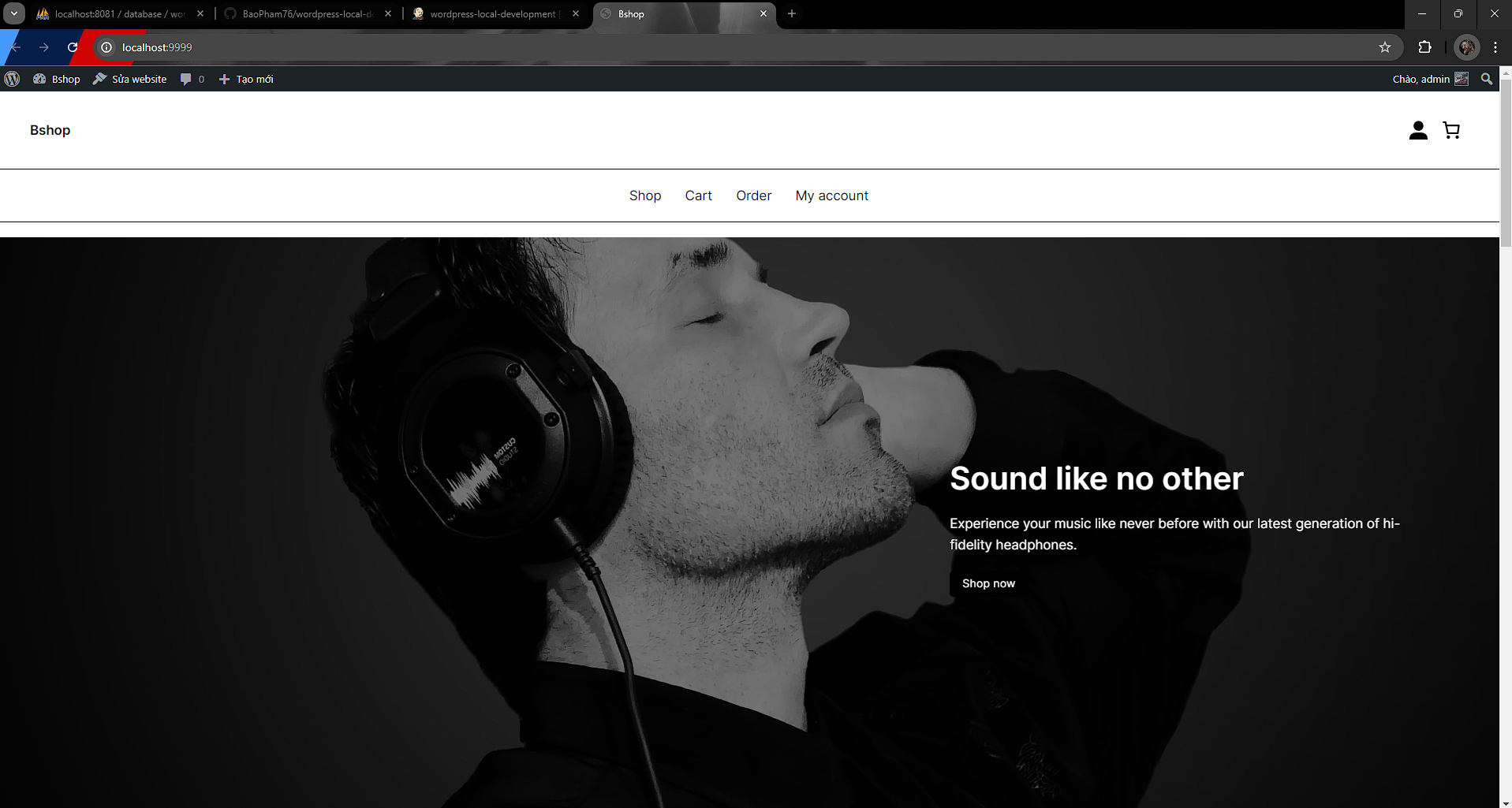
Kiểm tra log của Job để đảm bảo mọi thứ hoạt động đúng.

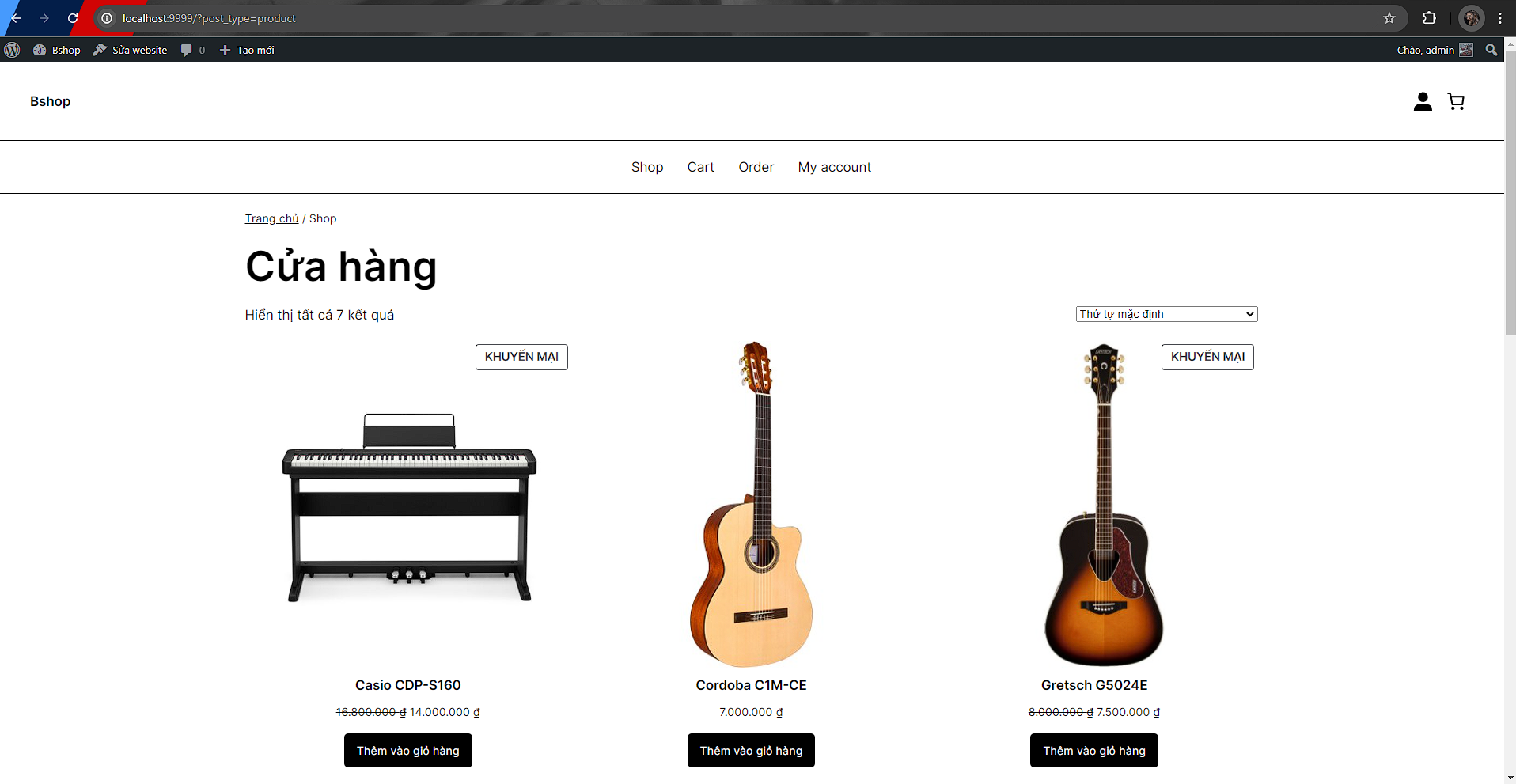




**- Bước 7: Kiểm tra ứng dụng**

Mở trình duyệt và truy cập http://localhost:9999 để kiểm tra ứng dụng WordPress của bạn.

Kết quả: 





# Kết luận

Bài báo cáo này đã giới thiệu quá trình triển khai và cài đặt một website tích hợp hai thành phần quan trọng là hệ thống quản lý nội dung (CMS) và nền tảng thương mại điện tử (E-commerce) dựa trên mã nguồn mở. Để đáp ứng yêu cầu này, em đã lựa chọn và triển khai CMS Wordpress và nền tảng E-commerce WooCommerce.

Để tối ưu hóa hiệu suất và bảo mật của hệ thống, em đã áp dụng cơ chế Reverse Proxy. Cụ thể, em sử dụng Nginx làm Reverse Proxy để bảo vệ hệ thống và tăng cường hiệu suất cho website.

Việc triển khai và cấu hình các thành phần này đã được thực hiện theo quy trình CI/CD (Continuous Integration/Continuous Deployment). Em đã chọn Jenkins làm công cụ CI/CD để tự động hóa quy trình triển khai, từ việc kiểm tra mã nguồn, xây dựng, đóng gói và triển khai tự động lên máy chủ sản phẩm. Jenkins đã giúp chúng tôi đạt được mục tiêu triển khai liên tục (Continuous Deployment) một cách hiệu quả và đáng tin cậy.

Qua quá trình này, em đã nhận thấy sự quan trọng của việc sử dụng CI/CD trong việc cung cấp các thay đổi và cập nhật nhanh chóng, đồng thời đảm bảo tính ổn định và hiệu suất của hệ thống website. Sự kết hợp giữa các công nghệ mã nguồn mở và các công cụ hiện đại như Reverse Proxy và Jenkins đã tạo nên một nền tảng mạnh mẽ cho việc phát triển và vận hành website.

Tóm lại, việc triển khai website với hai thành phần CMS và E-commerce mã nguồn mở, kết hợp với cơ chế Reverse Proxy và triển khai bằng CI/CD đã mang lại những kết quả tích cực, đáp ứng được yêu cầu về tính linh hoạt, bảo mật và hiệu suất của hệ thống. Đây là một bước tiến quan trọng trong việc áp dụng các công nghệ mới để nâng cao trải nghiệm người dùng và hiệu quả kinh doanh của website.