

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN 2
---o0o---



BÁO CÁO GIỮA KỲ & CUỐI KỲ
TÀI LIỆU HƯỚNG DẪN THIẾT LẬP VÀ TRIỂN KHAI PHẦN MỀM

Môn học: Chuyên đề Hệ thống thông tin

Giảng viên hướng dẫn: Lê Hà Thanh

Sinh viên thực hiện: Nguyễn Đức Huy

MSSV: N20DCCN021

Lớp: D20CQCNHT01-N

TP.HCM, ngày 17 tháng 06 năm 2024

Mục lục

I.	Giới thiệu tài liệu	2
II.	Công nghệ sử dụng	2
III.	Yêu cầu hệ thống	2
1.	Yêu cầu phần cứng	2
2.	Yêu cầu phần mềm	2
IV.	Triển khai phần mềm	2
1.	IDE/Text Editor.....	2
2.	Hệ quản trị phiên bản - Git	2
3.	Docker Desktop.....	2
4.	Pull sản phẩm về máy	3
5.	Sử dụng Visual Studio Code để triển khai phần mềm.....	4
5.1.	Mở Folder trong Visual Studio Code	4
5.2.	Giải thích file trong dự án	4
5.3.	Khởi chạy dự án	6
6.	Triển khai CMS Wordpress	9
7.	Triển khai cài đặt E-Commerce (WooCommerce).....	13
V.	Triển khai CI/CD	18

I. Giới thiệu tài liệu

- Tài liệu sử dụng cho:
- + Cài đặt website có hai thành phần CMS và E-Commerce mã nguồn mở dùng cơ chế Reverse Proxy.
- + Triển khai website theo cách thức CI/CD.

II. Công nghệ sử dụng

- Hệ quản trị nội dung mã nguồn mở (CMS - web content management system): Wordpress
- Cơ sở dữ liệu: MySQL
- Ngôn ngữ lập trình: PHP
- Reverse Proxy Server: Nginx
- Run-time Enviroment: Docker Desktop.
- Git GUI: SourceTree
- CI/CD: GitHub Actions
- Text Editor: Visual Studio Code.

III. Yêu cầu hệ thống

1. Yêu cầu phần cứng

- Hệ điều hành: Window 10/11 64-bit: Home or Pro phiên bản 21H2 hoặc cao hơn.
- Ram: Tối thiểu 4GB.
- Dung lượng: còn trống ít nhất 3GB.

2. Yêu cầu phần mềm

- IDE/Text Editor: Visual Studio, Visual Studio Code
- Run-time Enviroment: Docker Desktop.
- Hệ quản trị phiên bản: Git hoặc Git GUI (Github Desktop, SourceTree, ...)

IV. Triển khai phần mềm

1. IDE/Text Editor

- Ta sử dụng Visual Studio Code để dễ dàng cài đặt.
- Tải Visual Studio Code: <https://code.visualstudio.com/download>

2. Hệ quản trị phiên bản - Git

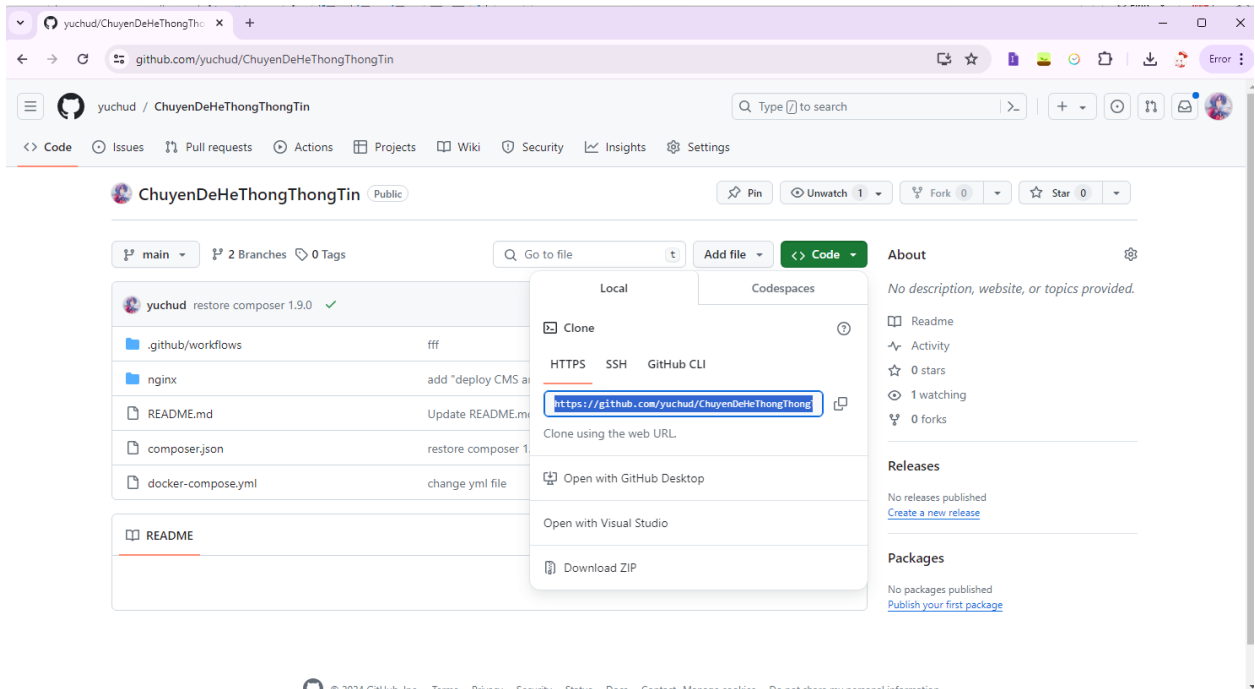
- Có thể sử dụng các câu lệnh của Git.
- Hoặc sử dụng Git GUI như Github Desktop, SourceTree
- Tải SourceTree: <https://www.sourcetreeapp.com/>

3. Docker Desktop

- Yêu cầu phải bật Virtual Machine Platform hoặc Hyper-V, Xem hướng dẫn tại: <https://funix.edu.vn/chia-se-kien-thuc/cach-bat-hyper-v-trong-windows-11/>
- Cài đặt Docker Desktop: <https://docs.docker.com/desktop/install/windows-install/>

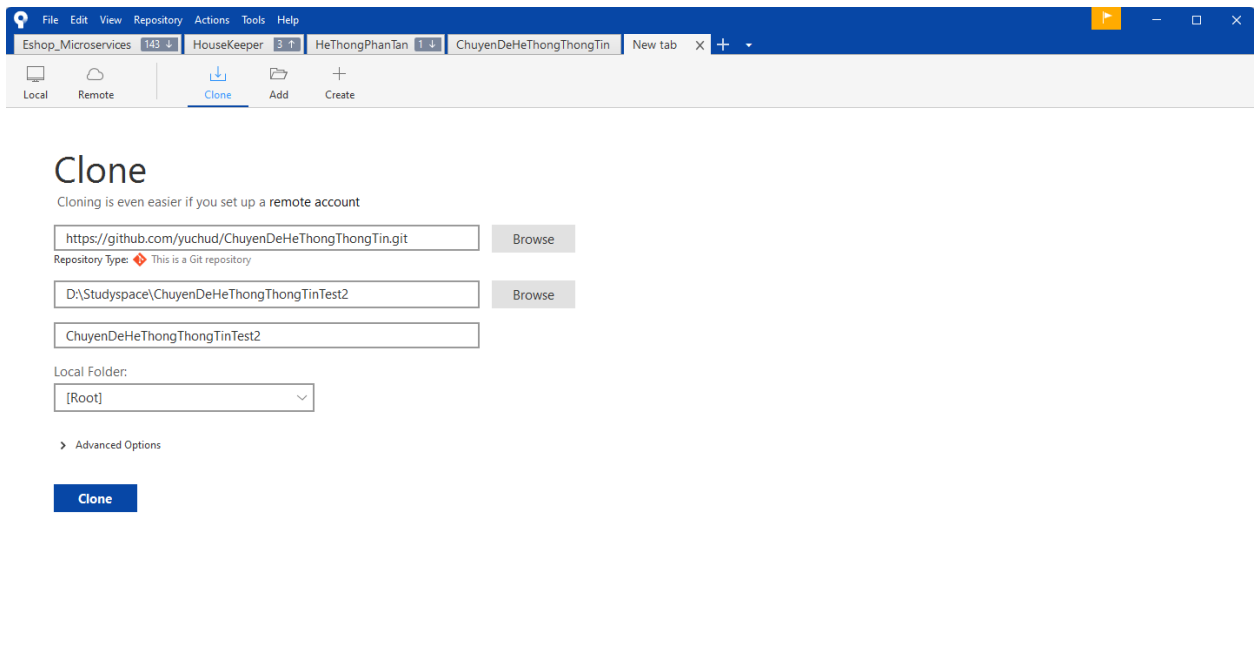
4. Pull sản phẩm về máy

- Truy cập trang web Github sau: <https://github.com/yuchud/ChuyenDeHeThongThongTin>
- Ấn vào **<> Code** và copy link có đuôi .git



Ảnh 1. Truy cập trang Github của dự án

- Mở SourceTree, Nhấn vào nút “clone”



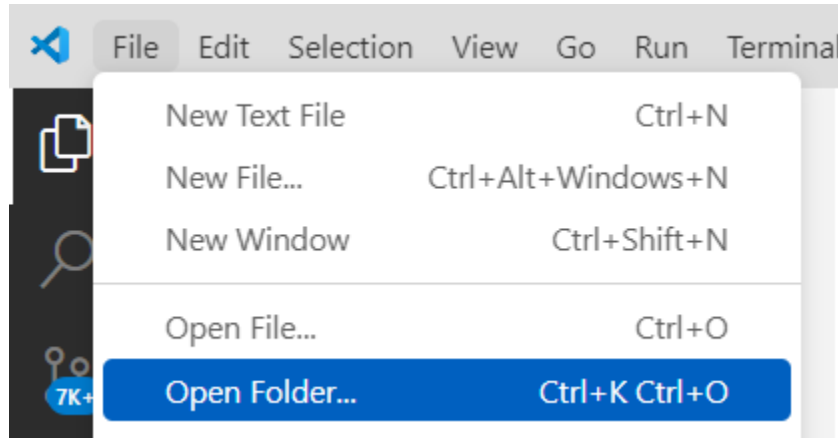
Ảnh 2. Pull dự án bằng SourceTree

- Nhập link có đuôi .git vào ô đầu tiên
- Ô thứ hai ta chọn thư mục muốn lưu dự án trong máy.
- Sau đó ta tiến hành clone.

5. Sử dụng Visual Studio Code để triển khai phần mềm

5.1. Mở Folder trong Visual Studio Code

- Bật Visual Studio Code, ấn File -> Open Folder rồi chọn Folder đã chọn trong máy (Ô thứ 2 của SourceTree ở trên)



Ảnh 3. Mở Folder dự án trong Visual Studio Code

5.2. Giải thích file trong dự án

a. Docker-compose.yml

- Dùng để cấu hình (config) các dịch vụ (services) của dự án
- Tìm hiểu thêm tại: <https://viblo.asia/p/docker-compose-la-gi-kien-thuc-co-ban-ve-docker-compose-1VgZv8d75Aw>

```
7  # Sử dụng phiên bản mới nhất của docker-compose
8  version: '3'
```

Ảnh 4. Phiên bản của docker-compose.yml

```

10 # Danh sách tất cả services (containers) cần chạy
11 # bao gồm wordpress, db, nginx
12 services:
13     # Service đầu tiên (wordpress)
14     wordpress:
15         # Sử dụng phiên bản mới nhất của wordpress từ Docker Hub
16         # Docker Hub wordpress: https://hub.docker.com/\_/wordpress
17         image: wordpress:latest
18         # Luôn khởi động lại container nếu container bị dừng
19         restart: always
20         # Mở cổng 8080 của máy host và map vào cổng 80 của container
21         ports:
22             - 8080:80
23         # Cấu hình biến môi trường cho wordpress
24         environment:
25             WORDPRESS_DB_HOST: db
26             WORDPRESS_DB_USER: wordpress
27             WORDPRESS_DB_PASSWORD: password
28             WORDPRESS_DB_NAME: wordpress
29         # Mount thư mục hiện tại vào thư mục /var/www/html của container
30         volumes:
31             - ./wordpress:/var/www/html

```

Ảnh 5. Service đầu tiên

```

33 # Service thứ hai (db)
34 db:
35     # Sử dụng phiên bản mới nhất của mysql từ Docker Hub
36     # Docker Hub mysql: https://hub.docker.com/\_/mysql
37     image: mysql:latest
38     # Luôn khởi động lại container nếu container bị dừng
39     restart: always
40     # Cấu hình biến môi trường cho mysql
41     environment:
42         MYSQL_DATABASE: wordpress
43         MYSQL_USER: wordpress
44         MYSQL_PASSWORD: password
45         MYSQL_ROOT_PASSWORD: root_password
46     # Mount thư mục hiện tại vào thư mục /var/lib/mysql của container
47     volumes:
48         - ./mysql:/var/lib/mysql

```

Ảnh 6. Service thứ hai

```

50 # Service thứ ba (nginx)
51 nginx:
52 # Sử dụng phiên bản mới nhất của nginx từ Docker Hub
53 # Docker Hub nginx: https://hub.docker.com/\_/nginx
54 image: nginx:latest
55 # Luôn khởi động lại container nếu container bị dừng
56 restart: always
57 # Mở cổng 80 của máy host và map vào cổng 80 của container
58 ports:
59 - 80:80
60 # Mount thư mục hiện tại vào thư mục /etc/nginx/conf.d của container
61 volumes:
62 - ./nginx:/etc/nginx/conf.d
63 - ./wordpress:/var/www/html

```

Ảnh 7. Service thứ ba

b. nginx.conf

- cho phép cấu hình nginx server.

```

1 # Xác định máy chủ web sẽ chạy trên cổng 80
2 server {
3     # Xác định cổng (port)
4     listen 80;
5     # Xác định tên miền hoặc địa chỉ IP của máy chủ web
6     server_name localhost; # yourdomain.com
7
8     # Xác định thư mục chứa mã nguồn của website
9     location / {
10         # Khi nginx nhận yêu cầu, nó sẽ chuyển yêu cầu đến dịch vụ "wordpress trên cổng 80
11         # wordpress là tên của dịch vụ trong file docker-compose.yml
12         proxy_pass http://wordpress:80;
13
14         #
15         proxy_set_header Host $host;
16
17         # Xác định địa chỉ IP của người dùng
18         proxy_set_header X-Real-IP $remote_addr;
19
20         #
21         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
22     }
23 }

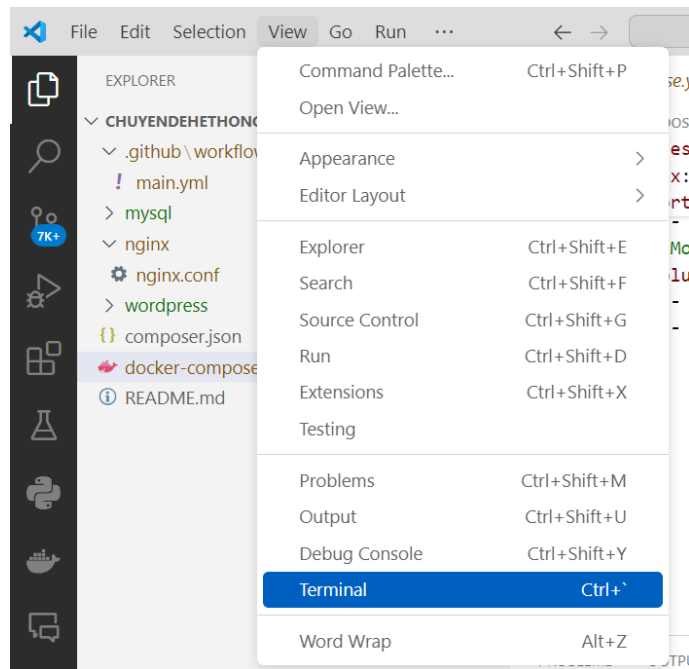
```

Ảnh 8. nginx.conf

5.3. Khởi chạy dự án

- Khởi động Docker Desktop.

- Mở bảng CMD bằng tổ hợp phím “Ctrl + `” hoặc vào View -> Terminal



Ảnh 9. Khởi chạy Terminal

- Sau đó nhập “docker-compose up” cho phép khởi động các containers.

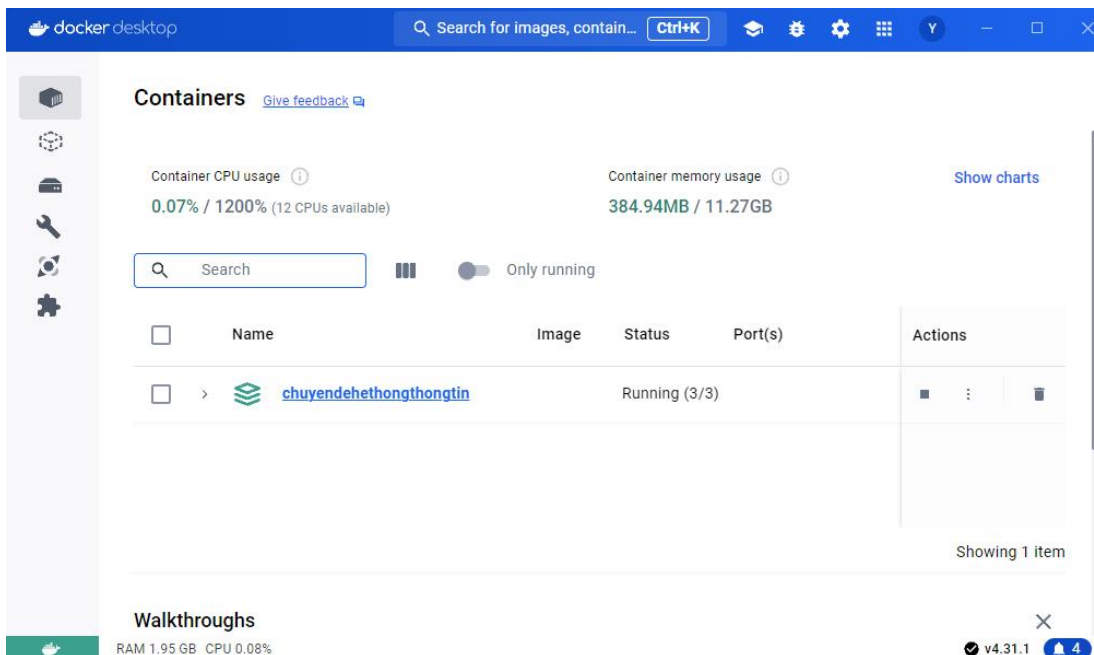
```
PS D:\Studyspace\ChuyenDeHeThongThongTin> docker-compose up
```

Ảnh 10. Thực thi docker-compose up

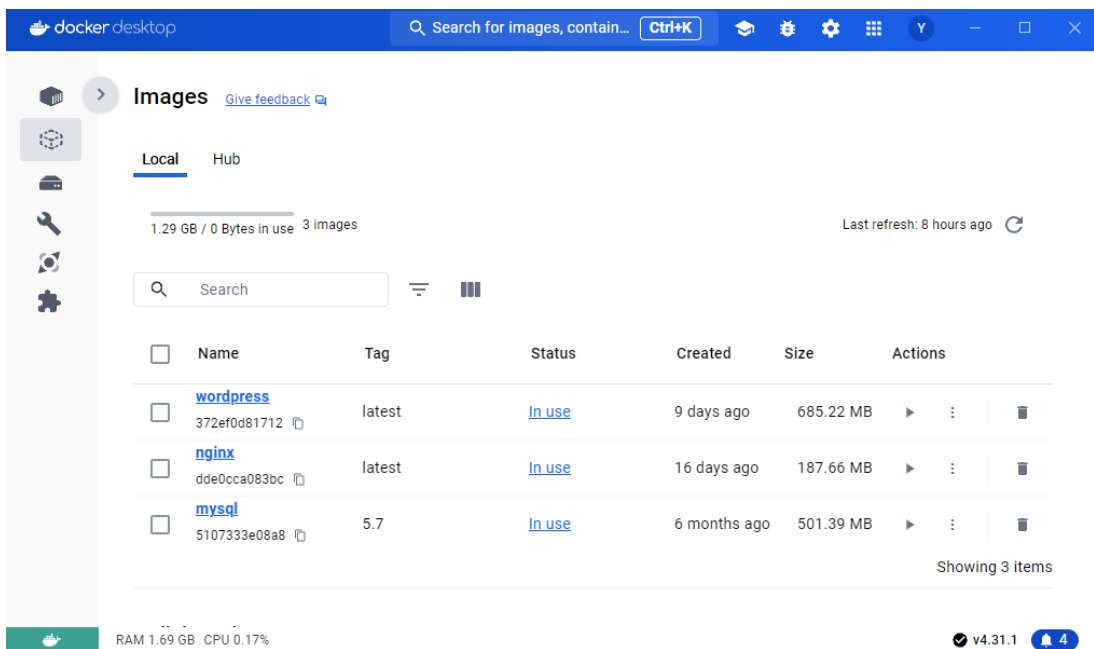
```
[+] Running 3/0
✓ Container chuyendehethongthongtin-nginx-1    Running    0.0s
✓ Container chuyendehethongthongtin-db-1       Running    0.0s
✓ Container chuyendehethongthongtin-wordpress-1 Running    0.0s
Attaching to db-1, nginx-1, wordpress-1
```

Ảnh 11. Các Containers và Images đã được khởi chạy

- Vào Docker Desktop để xem các Containers và Images.

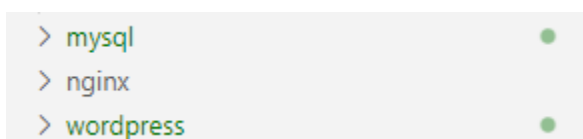


Ảnh 12. Docker Containers



Ảnh 13. Các Docker Images

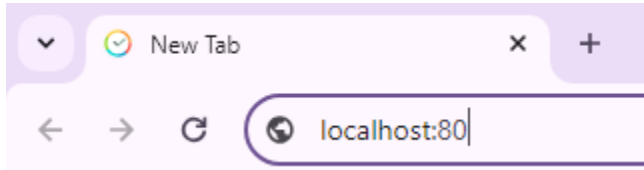
- Lúc này, trong Visual Studio Code, sẽ có thêm 2 Folder là “mysql” và “wordpress” gồm các file phục vụ cho cơ sở dữ liệu Mysql và CMS Wordpress.



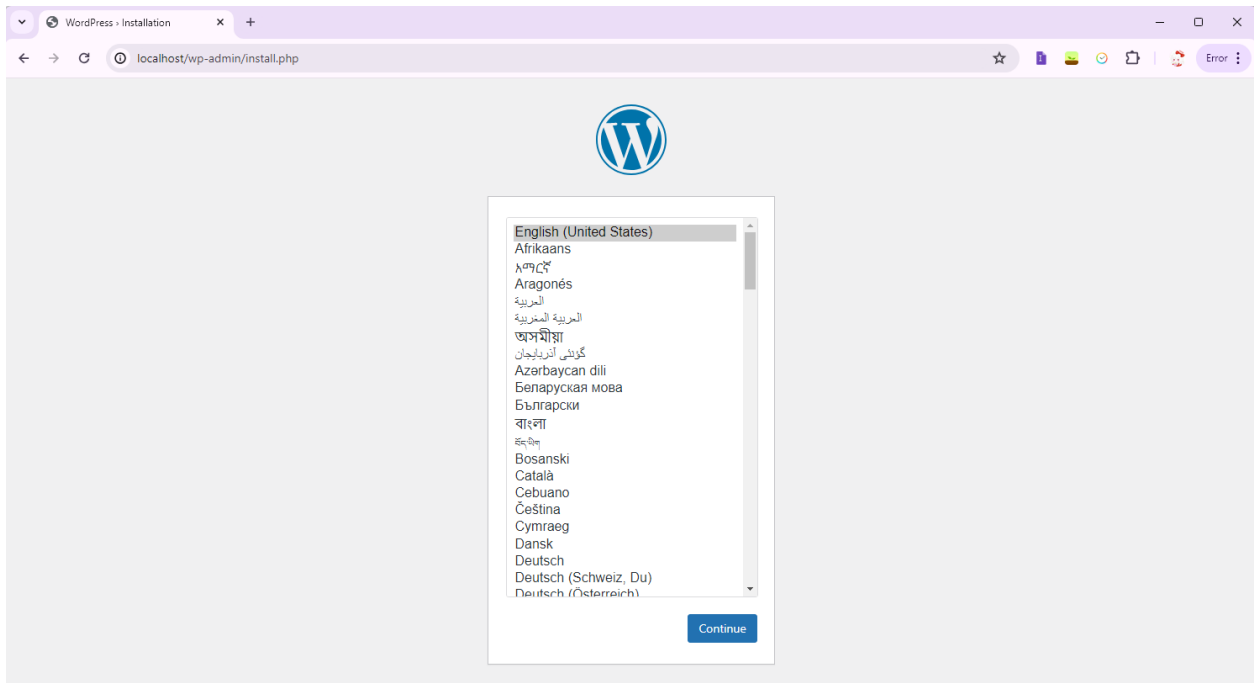
- Địa chỉ truy cập trang web: <http://localhost:80>
- Địa chỉ truy cập trang web cho quản trị viên: <http://localhost/wp-admin/>

6. Triển khai CMS Wordpress

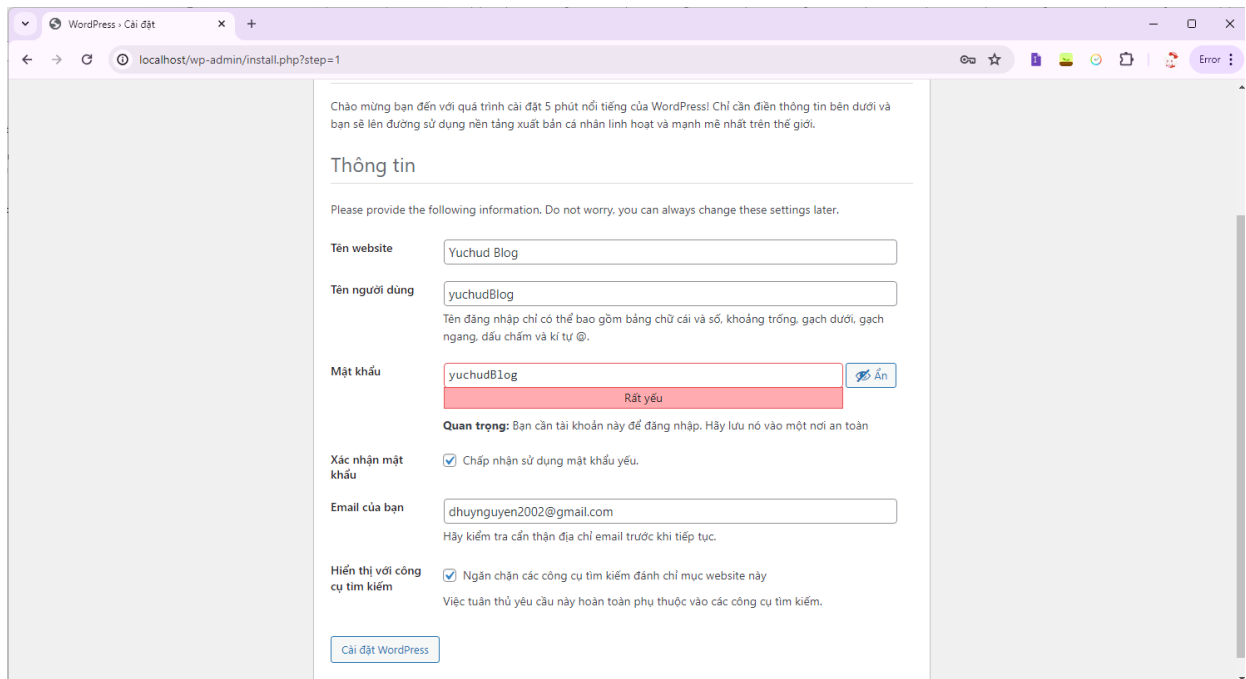
- Trên trình duyệt, truy cập [trang web](#) hoặc nhập “localhost:80” để truy cập trang web



- Lúc này, sẽ hiển thị một trang để thiết lập ngôn ngữ cho trang web.



- Ta đặt tài khoản và mật khẩu để đăng nhập vào khu vực Quản trị của Wordpress.



The screenshot shows the first step of the WordPress installation process in a web browser. The address bar indicates the URL is localhost/wp-admin/install.php?step=1. The page has a light gray background with a white central form area. At the top of the form, there is a welcome message in Vietnamese and English. Below this, the heading "Thông tin" (Information) is followed by a prompt to provide information. The form contains several input fields: "Tên website" (Website name) with the value "Yuchud Blog", "Tên người dùng" (Username) with the value "yuchudBlog", and "Mật khẩu" (Password) with the value "yuchudBlog". The password field has a red highlight and the text "Rất yếu" (Very weak). There are also checkboxes for "Xác nhận mật khẩu" (Confirm password), "Hiển thị với công cụ tìm kiếm" (Show to search engines), and "Ngăn chặn các công cụ tìm kiếm đánh chỉ mục website này" (Prevent search engines from indexing this website). A "Cài đặt WordPress" (Install WordPress) button is at the bottom of the form.

Chào mừng bạn đến với quá trình cài đặt 5 phút nổi tiếng của WordPress! Chỉ cần điền thông tin bên dưới và bạn sẽ lên đường sử dụng nền tảng xuất bản cá nhân linh hoạt và mạnh mẽ nhất trên thế giới.

Thông tin

Please provide the following information. Do not worry, you can always change these settings later.

Tên website: Yuchud Blog

Tên người dùng: yuchudBlog

Tên đăng nhập chỉ có thể bao gồm bảng chữ cái và số, khoảng trống, gạch dưới, gạch ngang, dấu chấm và kí tự @.

Mật khẩu: yuchudBlog Rất yếu

Quan trọng: Bạn cần tài khoản này để đăng nhập. Hãy lưu nó vào một nơi an toàn

Xác nhận mật khẩu: ☒ Chấp nhận sử dụng mật khẩu yếu.

Email của bạn: dhuynghuyen2002@gmail.com

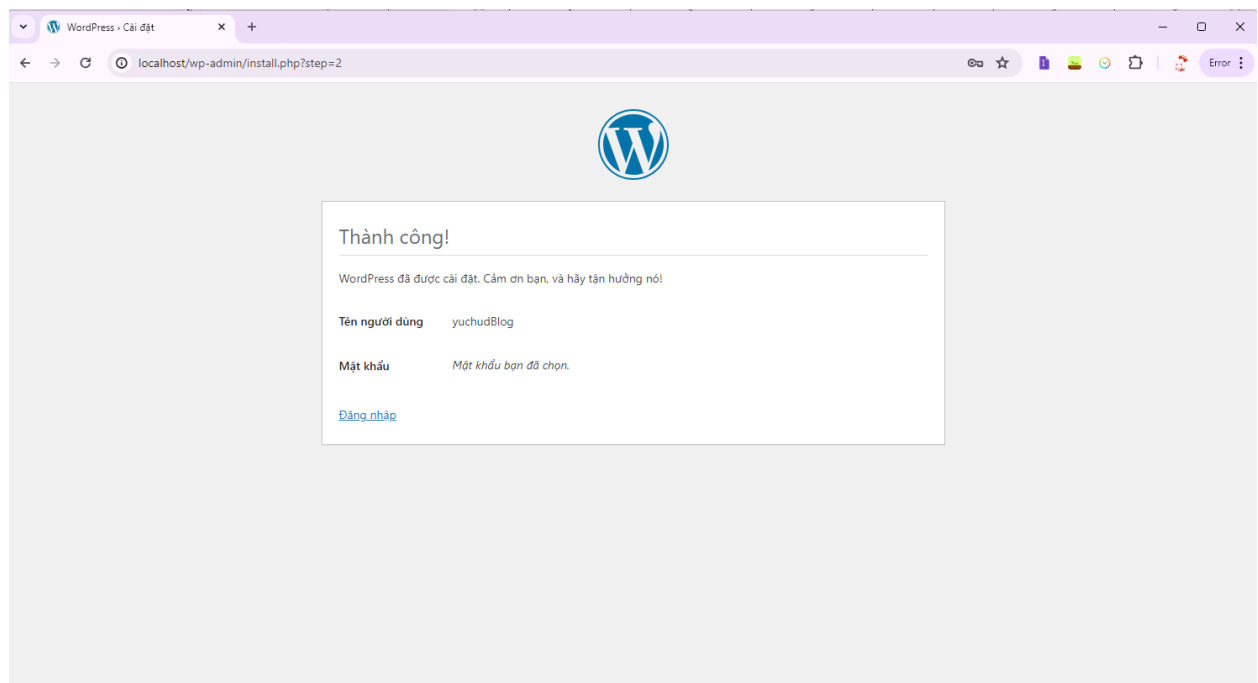
Hãy kiểm tra cẩn thận địa chỉ email trước khi tiếp tục.

Hiển thị với công cụ tìm kiếm: ☒ Ngăn chặn các công cụ tìm kiếm đánh chỉ mục website này

Việc tuân thủ yêu cầu này hoàn toàn phụ thuộc vào các công cụ tìm kiếm.

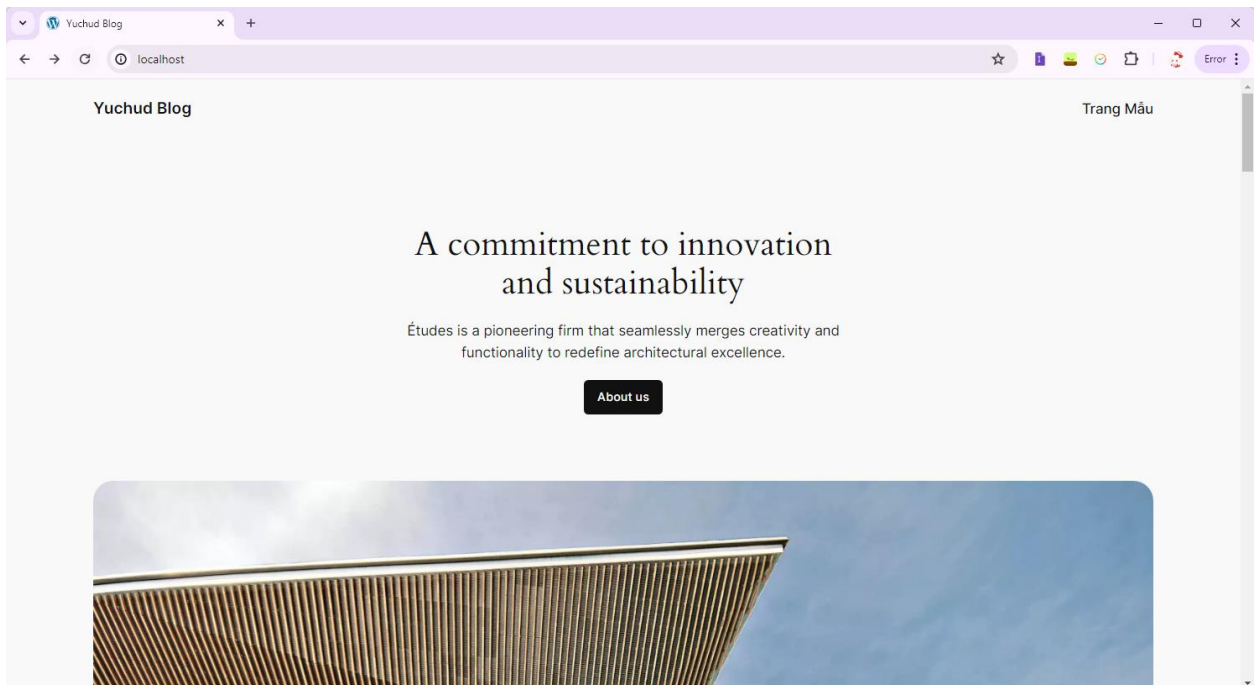
[Cài đặt WordPress](#)

Ảnh 14. Nhập thông tin trên Wordpress



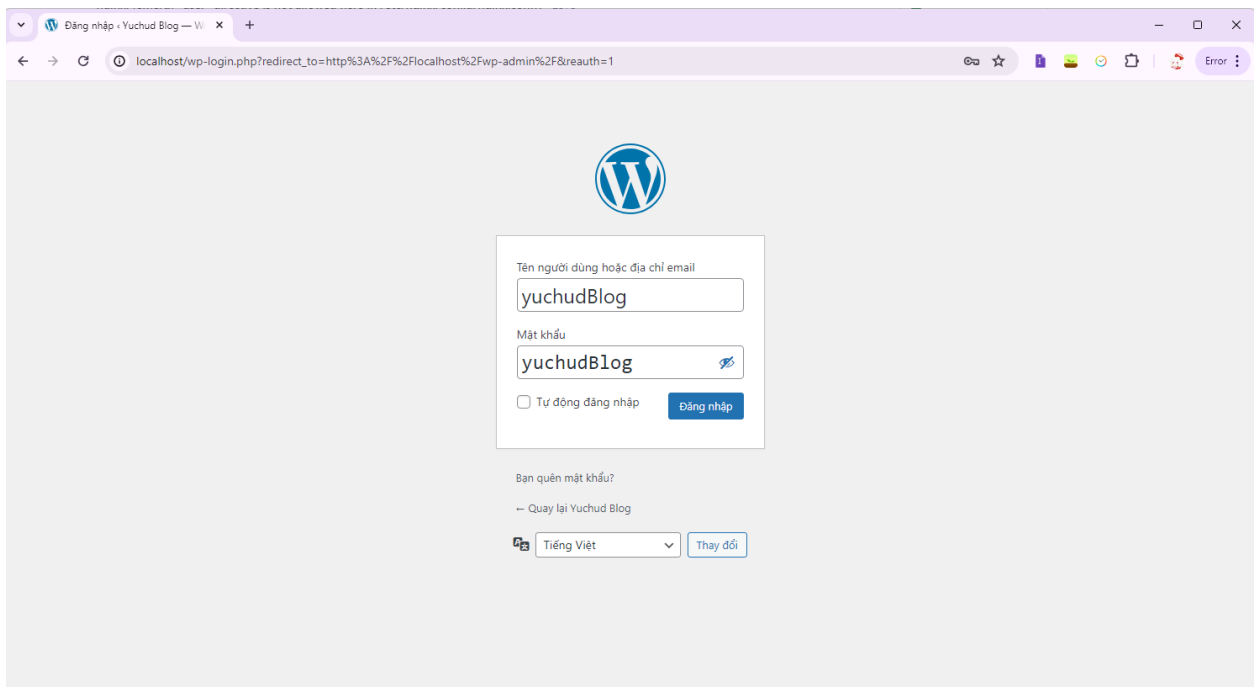
Ảnh 15. Tạo trang web thành công

- Truy cập [trang web](#) hoặc nhập “localhost:80”, ta sẽ thấy trang web đã tạo

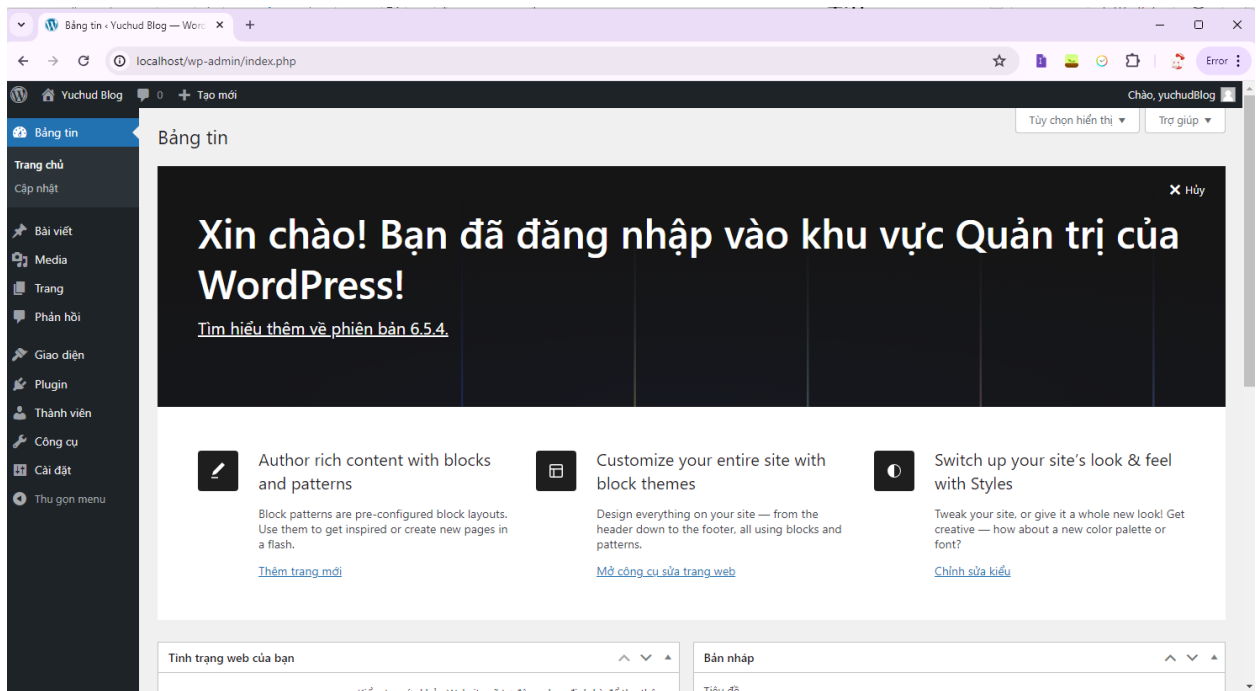


Ảnh 16. Trang blog được tạo bởi WordPress

- Truy cập [Trang Web cho admin](#) hoặc nhập địa chỉ “localhost:80/wp-admin/” để vào trang web cho admin.

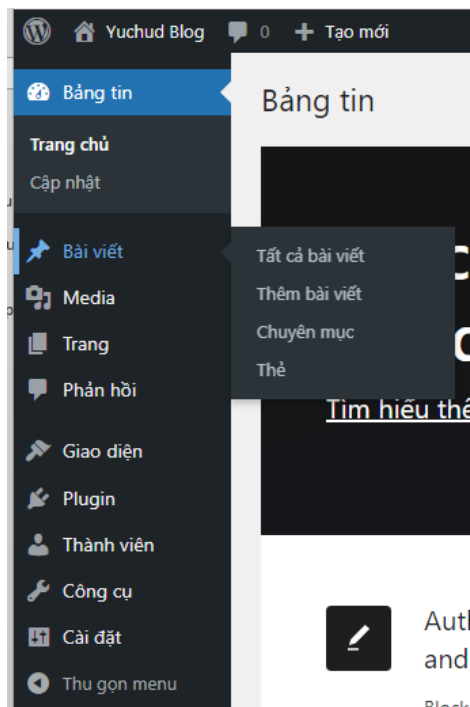


Ảnh 17. Đăng nhập vào khu vực Quản trị của WordPress



Ảnh 18. Khu vực quản trị của WordPress

- Thanh bên trái là thanh công cụ giúp quản lý trang web, ví dụ như việc quản lý bài viết hay thêm bài viết.

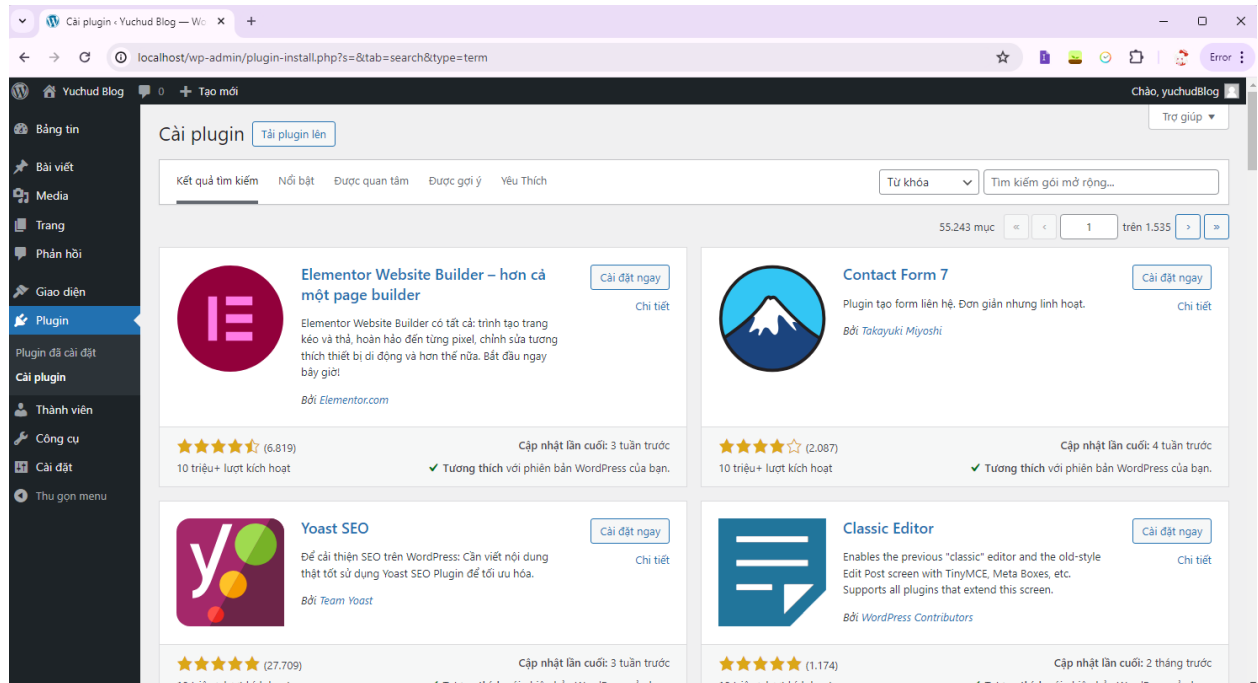


Ảnh 19. Quản lý các bài viết

- Lúc này, ta đã tạo được một trang web viết Blog bằng Wordpress

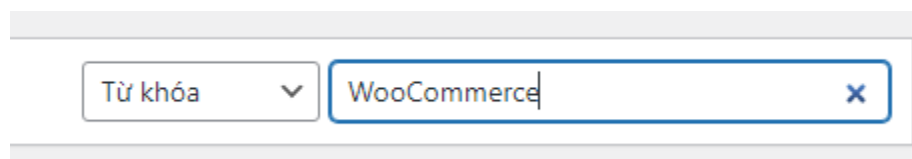
7. Triển khai cài đặt E-Commerce (WooCommerce)

- Wordpress cho phép cài đặt nhiều tiện ích phục vụ cho trang web
- Tại trang quản trị, trong thanh công cụ bên trái chọn “Plugin” Sau đó chọn “Cài plugin”

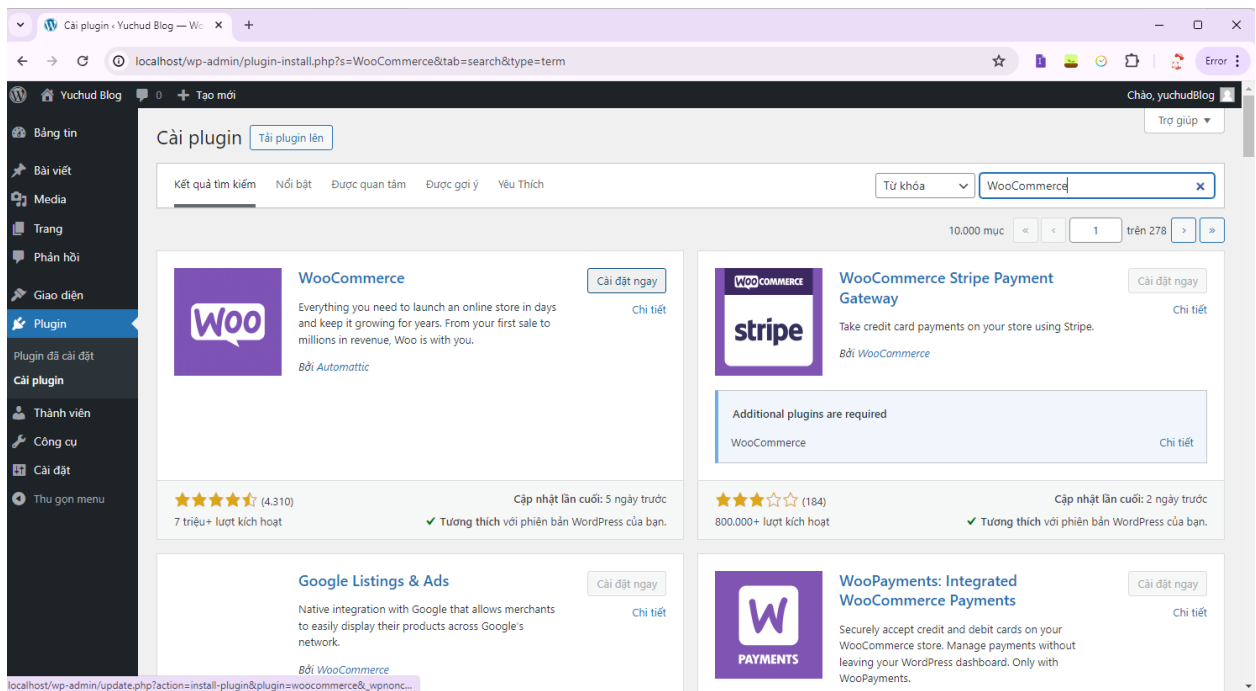


Ảnh 20. Các gói tiện ích của Wordpress

- Nhập từ khóa “WooCommerce” để tìm kiếm

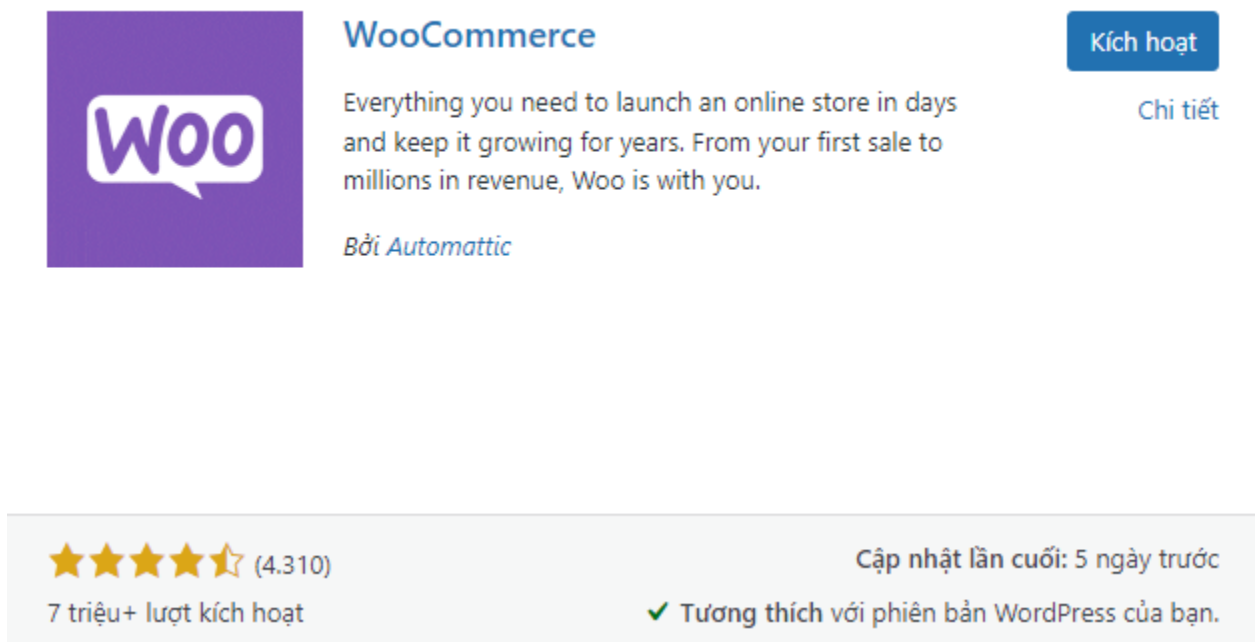


Ảnh 21. Tìm WooCommerce

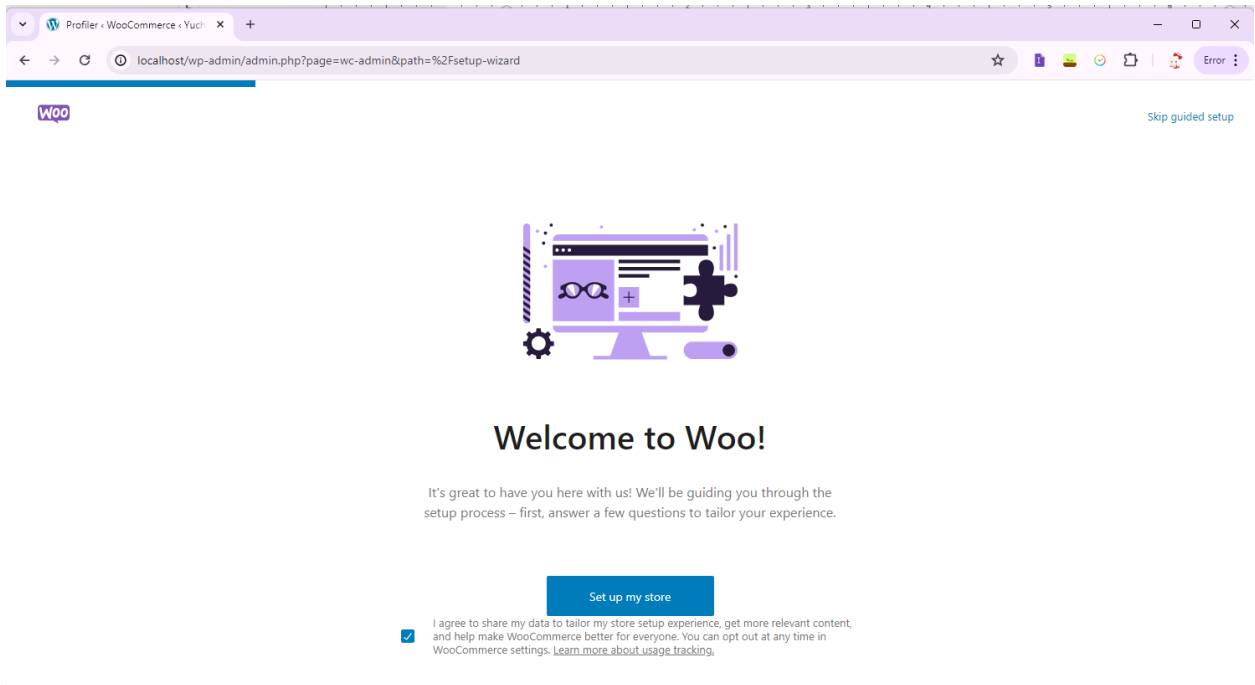


Ảnh 22. Cài đặt WooCommerce

- Ta tiến hành cài đặt ” WooCommerce”, Sau khi cài đặt xong nhấn “kích hoạt” để áp dụng vào trang Web.

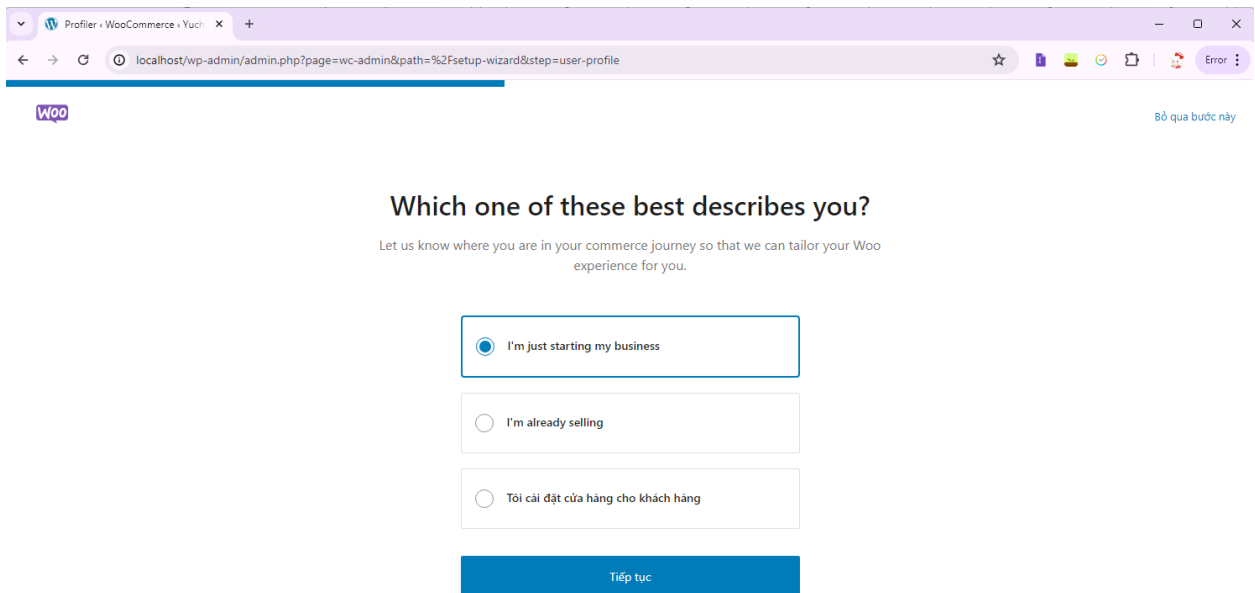


Ảnh 23. Kích hoạt WooCommerce



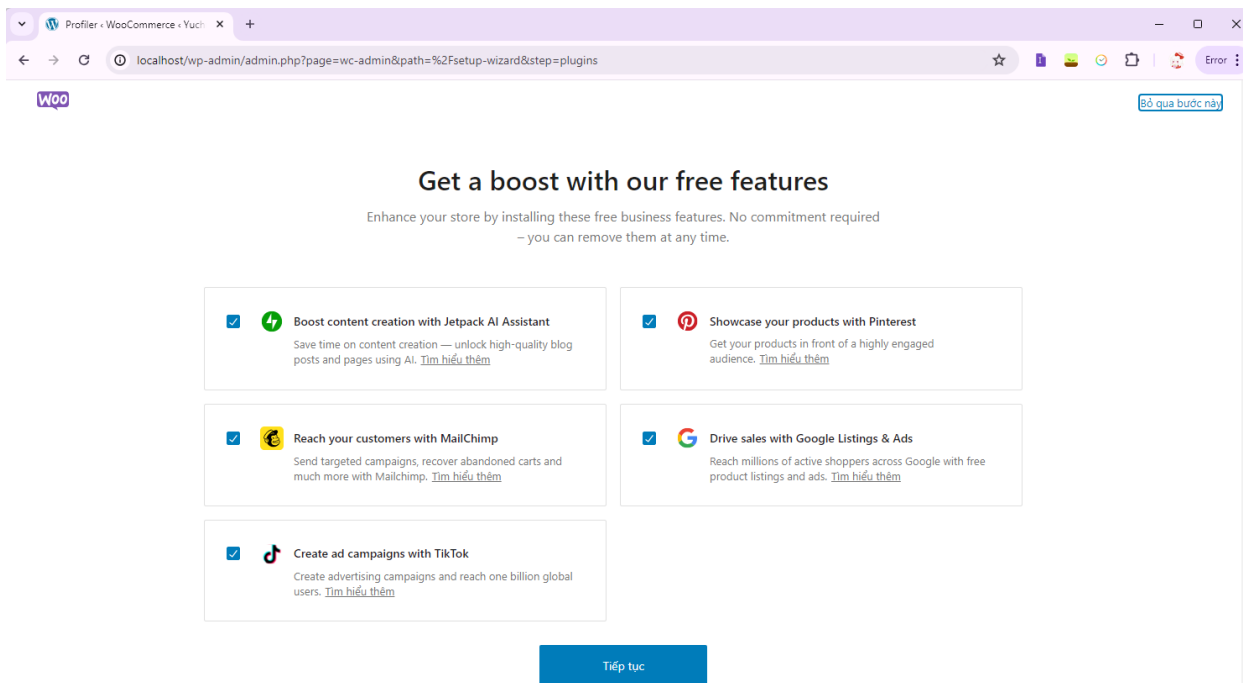
Ảnh 24. Tiến hành thiết lập cửa hàng cho bản thân

- Lúc này, ta trả lời các câu hỏi theo nhu cầu của bản thân để xây dựng trang web



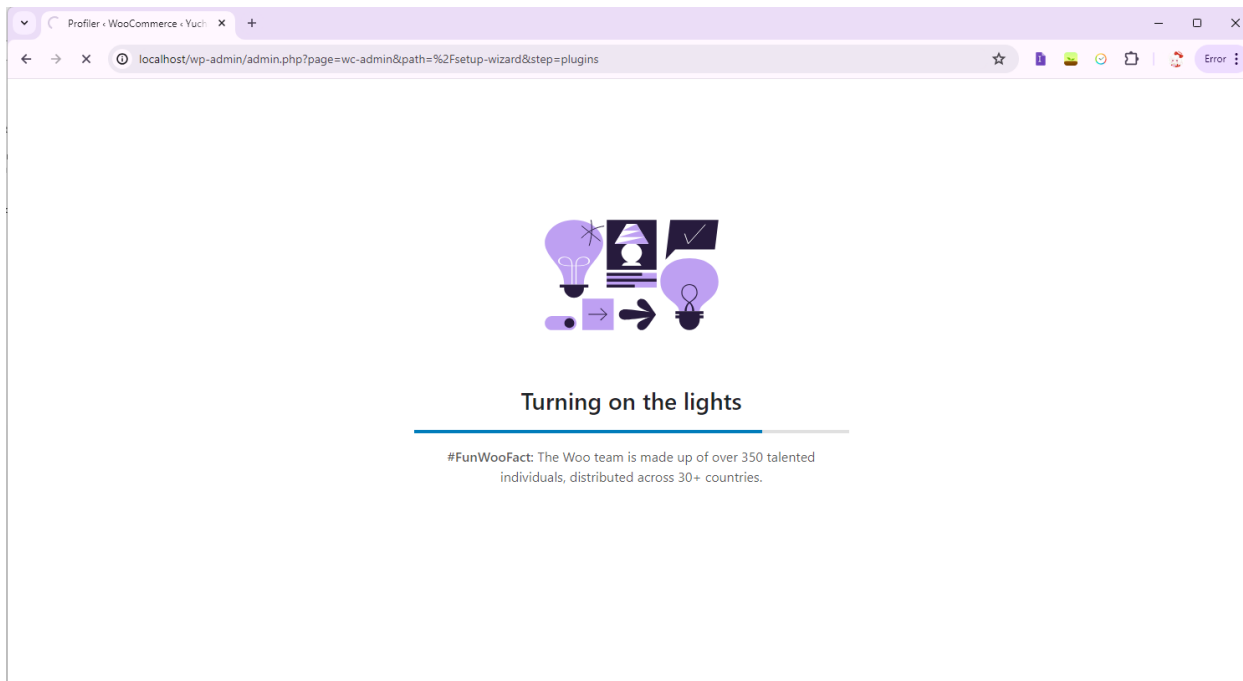
Ảnh 25. Trả lời các câu hỏi dựa trên nhu cầu bản thân

- Ta có thể thêm các tính năng hỗ trợ hoặc là bỏ qua.



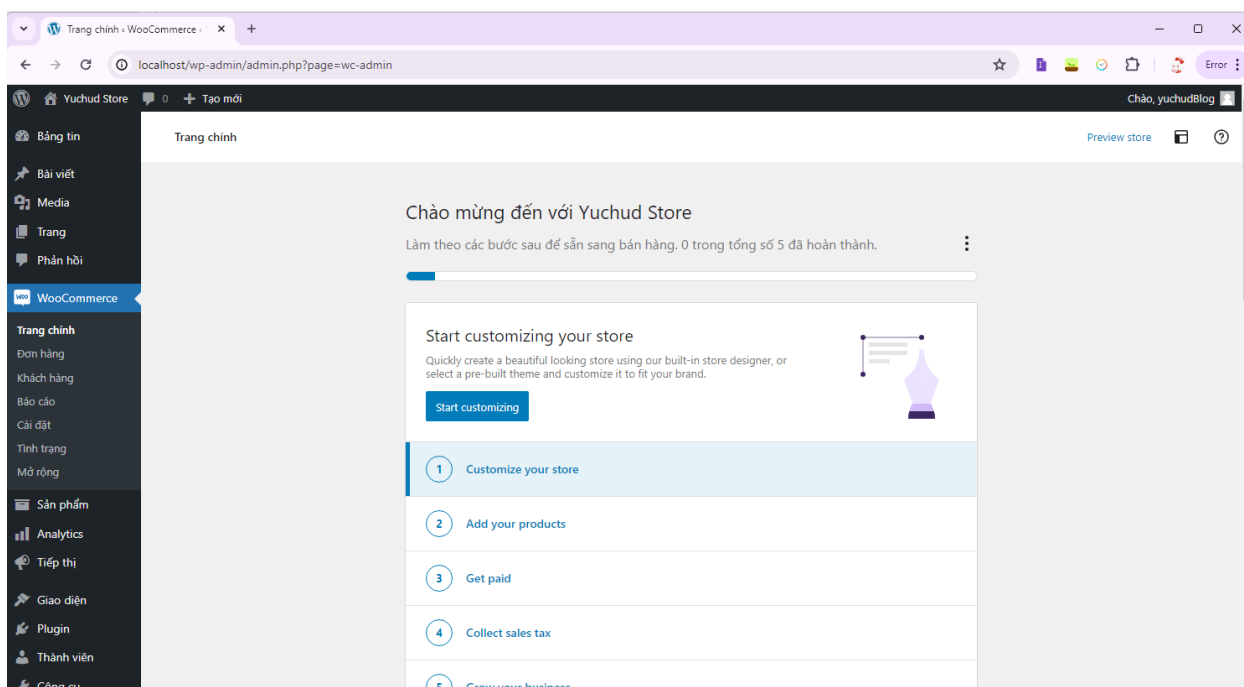
Ảnh 26. Có thể thêm tiện ích cho cửa hàng nếu có nhu cầu

- Ta chỉ cần đợi hoàn tất



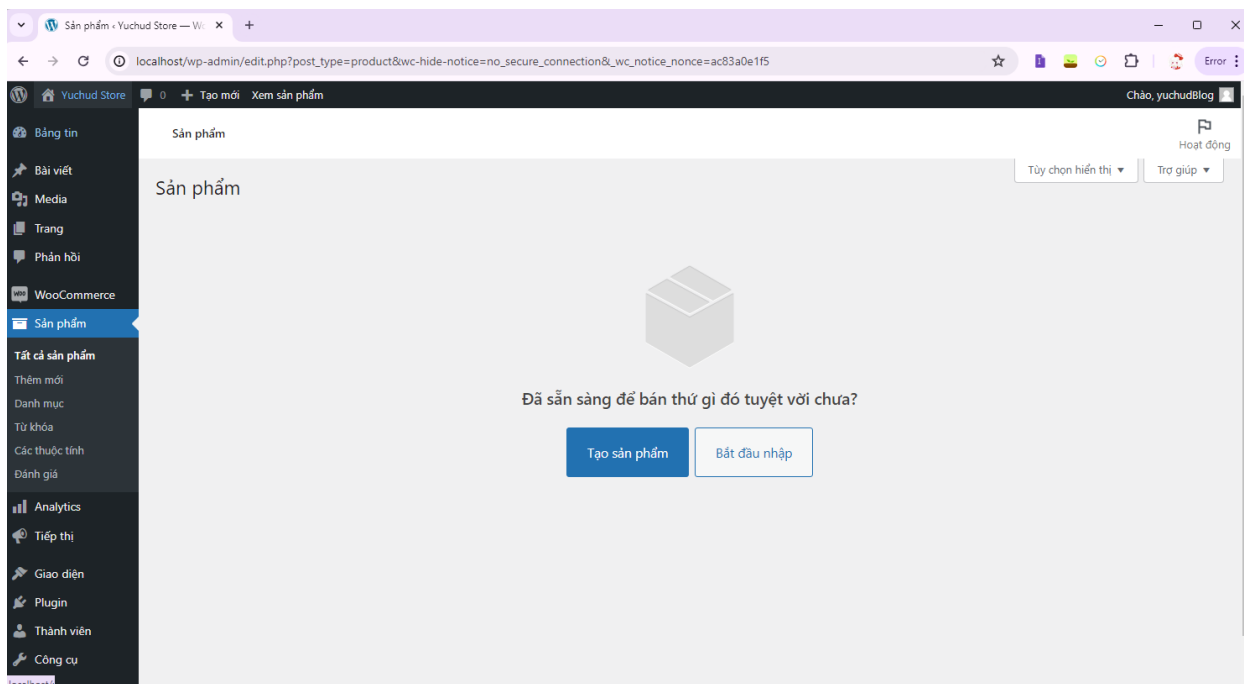
Ảnh 27. Chờ hoàn tất

- Tại thanh công cụ bên trái có hiển thị “WooCommerce”, ta có thể thiết lập cửa hàng của bản thân.



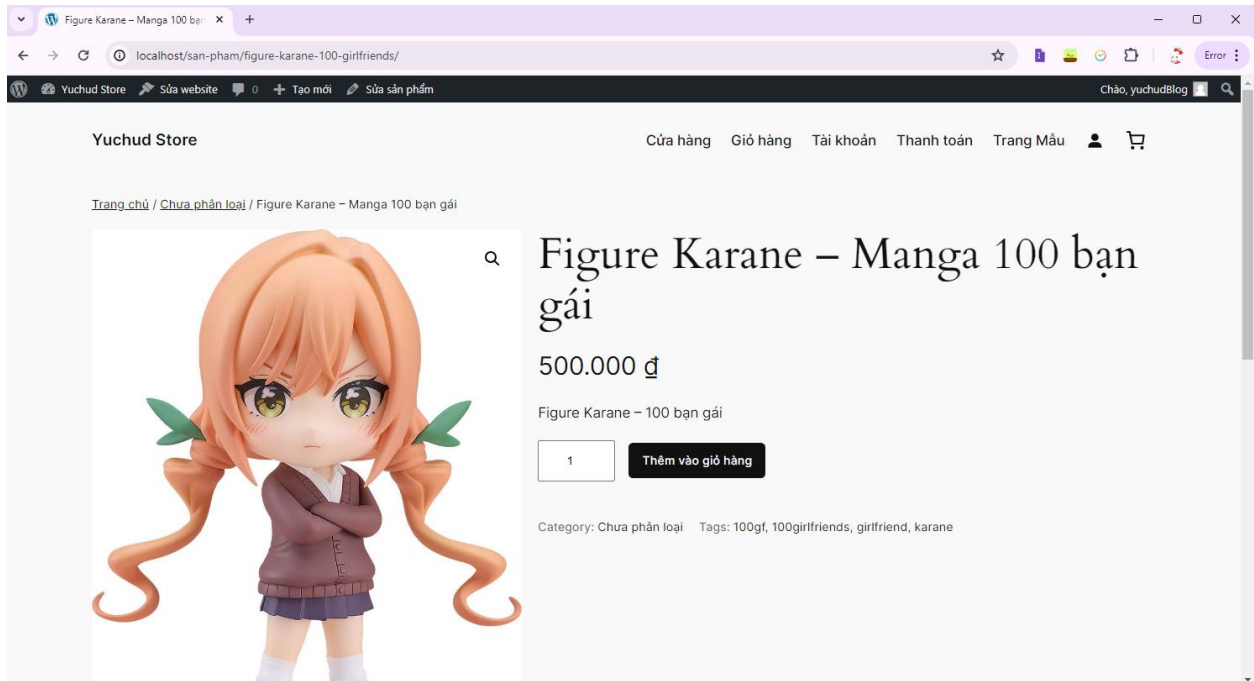
Ảnh 28. Thiết lập thông tin cho cửa hàng cá nhân

- Tại thanh công cụ, nhấn vào “Sản phẩm” để quản lý sản phẩm và thêm sản phẩm.



Ảnh 29. Quản lý các sản phẩm trên Website

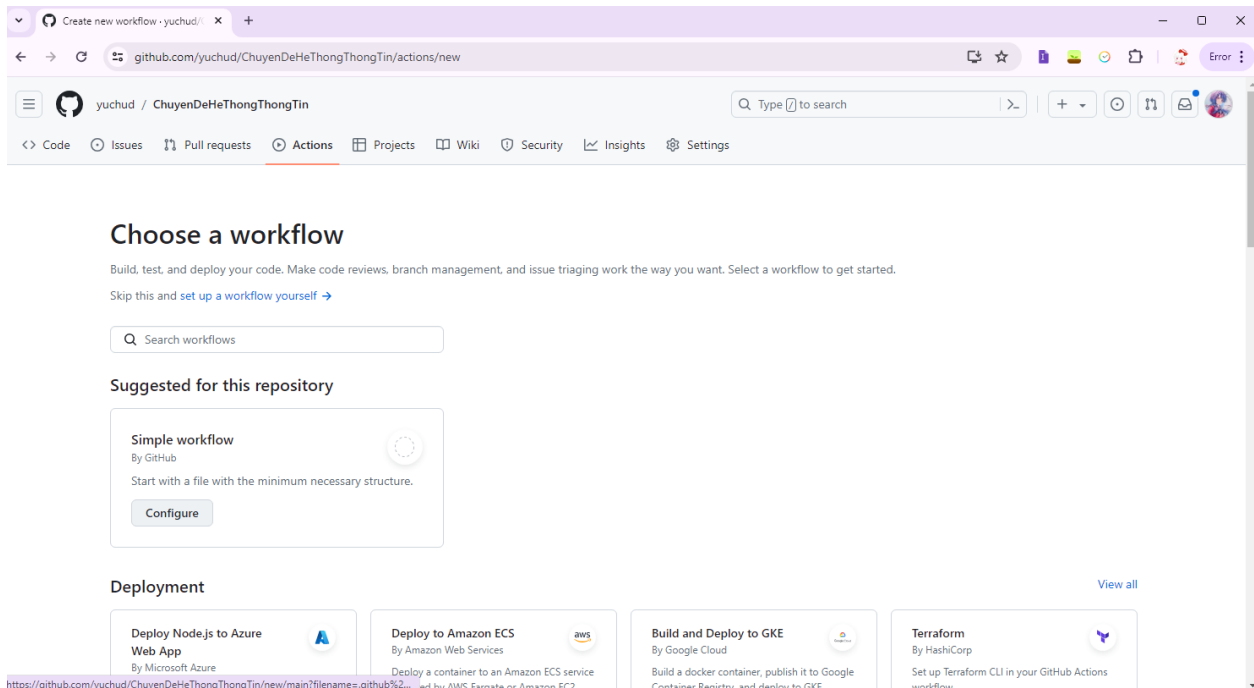
- Khi thêm sản phẩm thành công, sản phẩm sẽ được thêm lên cửa hàng của trang Web.



Ảnh 30. Khi thêm sản phẩm, sẽ được hiển thị trên cửa hàng

V. Triển khai CI/CD

- Sử dụng Github Actions của trang Github.



- Khi ta Configure, Github sẽ tạo một file có đuôi .yml trong thư mục .github/workflows.
- Trong Dự án này, file main.yml đã được config sẵn.

```
! main.yml M X
.github > workflows > ! main.yml
1  # File của Github Actions
2  name: CI/CD Pipeline
3
4  # Chạy khi có sự kiện push hoặc pull request trên nhánh main hoặc CI/CD_Test
5  on:
6    push:
7      branches: [ main, CI/CD_Test ]
8    pull_request:
9      branches: [ main, CI/CD_Test ]
10
11 # Các job cần thực hiện trên Github Actions
```

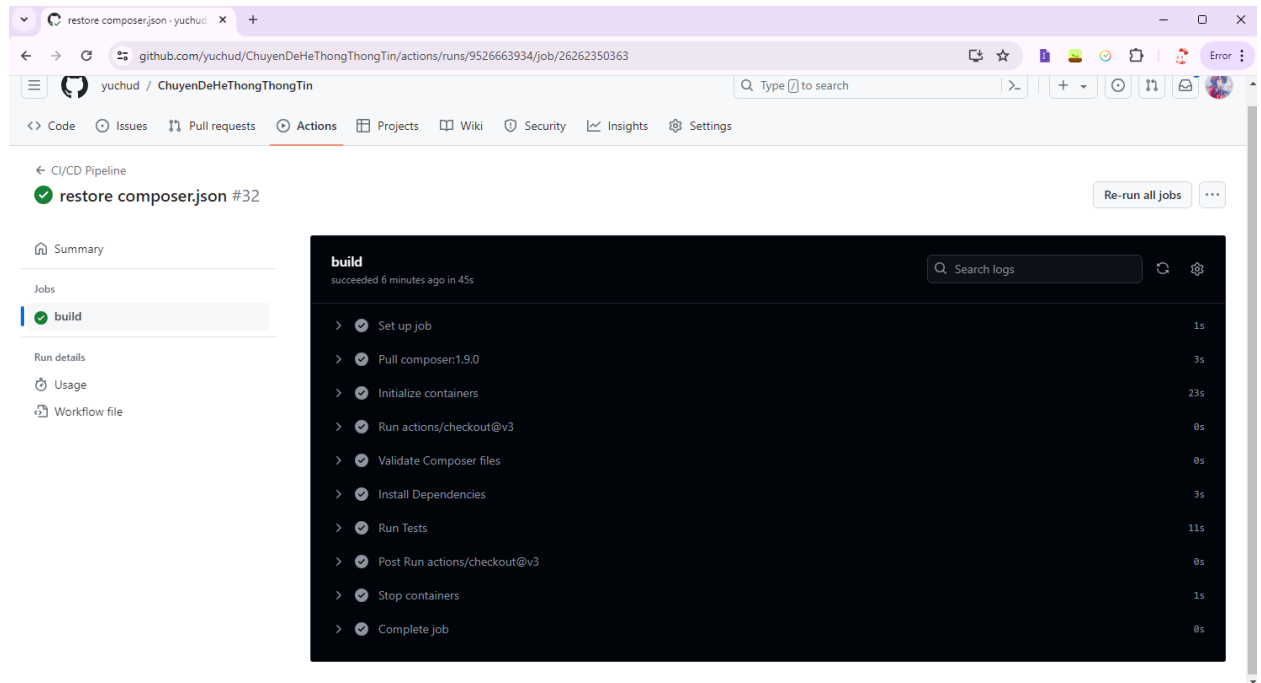
Ảnh 31. file main.yml

- Đồng thời sẽ có thêm file `composer.json` chứa các dependencies để hỗ trợ trong cho việc chạy của Github Actions có thể sử dụng Docker.

```
{ } composer.json ×
{ } composer.json > { } extra
1  {
2      "name": "vendor/myproject",
3      "description": "My project",
4      "require": {
5          "php": "^7.4",
6          "ext-json": "*"
7      },
8      "config": {
9          "platform": {
10             "php": "7.4"
11         }
12     },
13     "extra": {
14         "github": {
15             "actions": {
16                 "check-docker": {
17                     "runs": {
18                         "using": "composite",
19                         "steps": [
20                             {
21                                 "name": "Check Docker",
22                                 "run": "docker --version"
23                             }
24                         ]
25                     }
26                 }
27             }
28         }
29     }
30 }
```

Ảnh 32. `Composer.json`

- Trong file main.yml, ta đã thiết lập chạy Github Actions khi có “pull request” hoặc “push” từ nhánh “main” hoặc “CICD_Test”.
- Nên từ nhánh main, ta thực hiện push, Github Actions sẽ được khởi chạy.

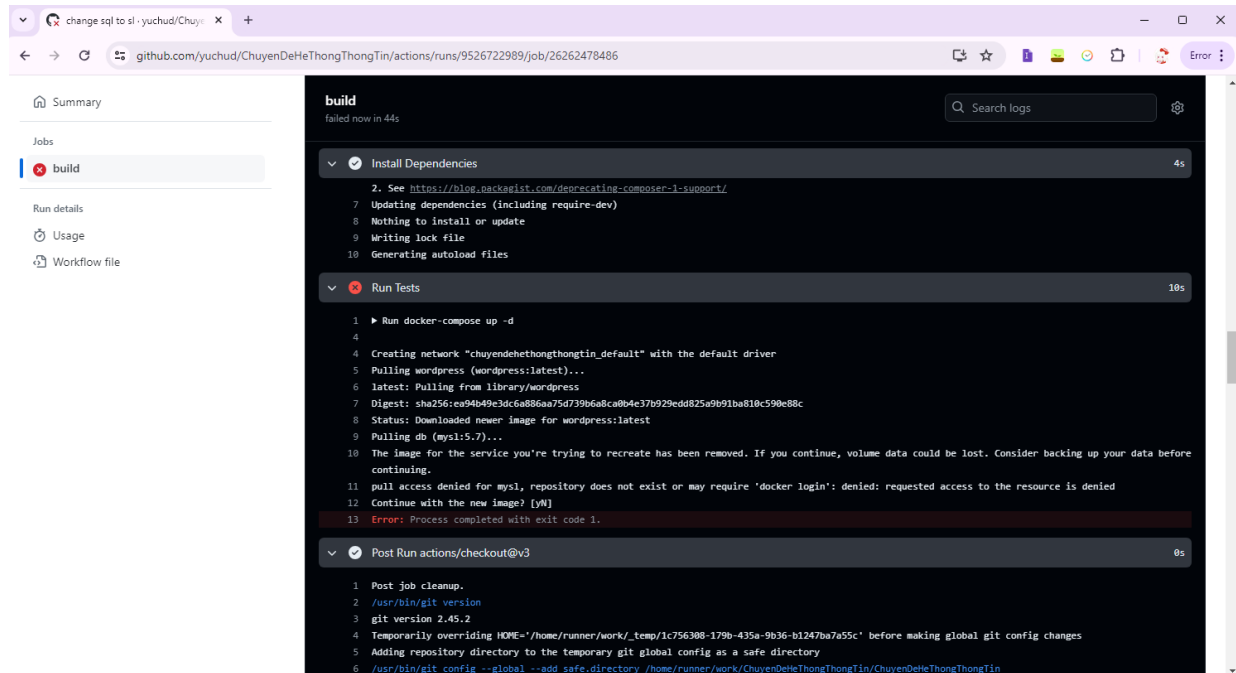


Ảnh 33. Kiểm tra thành công

- Trong trường hợp ta sửa sai, ví như ta sửa “mysql” thành “mysl” trong file “docker-compose.yml”, khi push, github sẽ báo lỗi.

```
--
17 db:
18   image: mysl:5.7
19   restart: always
```

Ảnh 34. Thay đổi để tạo lỗi sai



Ảnh 35. Kiểm tra thất bại