



**Conceive Design Implement Operate** 



THỰC HỌC – THỰC NGHIỆP

#### FRONT-END FRAMEWORK

GIỚI THIỆU FRAMEWORK NUXT -TRIỂN KHAI DỰ ÁN VUE





- Hiểu được cơ bản Framework Nuxt.js
- Biết được sự khác nhau cơ bản giữa vue.js và nuxt.js
- Nắm được khái niệm CSR và SSR
- Biết cách triển khai dự án trên nền tảng Netlify







#### Phần 1

- Giới thiệu framework Nuxt.js
- Lợi ích khi sử dụng Nuxt.js
- So sánh Nuxt.Js và Vue.Js
- ❖ SSR và CSR
- Cài đặt và sử dụng Nuxt.js
- Routing: Pages và Navigation

#### Phần 2

- Chuẩn bị
- Đẩy code lên github
- \* Kết nối github với netlify
- Hoàn tất





# PHÂN 1 GIỚI THIỆU FRAMEWORK NUXT



- Nuxt.js là một framework dựa trên Vue.js, giúp xây dựng các ứng dụng Vue có khả năng render từ phía server (SSR) và tạo các trang web tĩnh dễ dàng hơn.
- Server-Side Rendering (SSR): Render nội dung trên server trước khi gửi đến client.
- Static Site Generation (SSG): Tạo trang web tĩnh từ nội dung động.
- Single Page Application (SPA):
  H\u00f6 tr\u00f3 c\u00e1c \u00fcr ung d\u00fcn trang.





# Lợi ích khi sử dụng Nuxt.js

- 1. Server-Side Rendering (SSR)
- 2. Static Site Generation (SSG)
- 3. Routing Tự động
- 4. Tích hợp Vuex
- 5. Quản lý Middleware
- 6. Plugin và Module
- 7. Quản lý Cấu hình dễ dàng
- 8. Hỗ trợ cho API
- 9. SEO và Meta Tags
- 10. Tối ưu hóa Tốc độ





# SO SÁNH NUXT.JS VÀ VUE.JS

Tiêu chí	Vue.js	Nuxt.js
Kiểu framework	Front-end	Full-stack (dựa trên Vue.js)
Routing	Cấu hình thủ công	Tự động
SSR	Không hỗ trợ sẵn	Hỗ trợ mặc định
SEO	Khó tối ưu	Tối ưu nhờ SSR và SSG
Cấu trúc thư mục	Tự do	Chuẩn hóa (pages, components, etc.)
Ứng dụng phù hợp	Các website dạng quản lý phần mềm, private	các website liên quan đến SEO như bán hàng, tin tức
Hiệu suất	Phụ thuộc vào SPA	Nhanh hơn nhờ SSR và SSG



# SERVER SIDE RENDERING (SSR) LÀ GÌ?



- SSR (Server-Side Rendering): là quá trình render trang web trên máy chủ và gửi HTML đã render sẵn về cho trình duyệt người dùng.
- CSR (Client-Side Rendering): là quá trình render trang web diễn ra trên trình duyệt, JavaScript tải và dựng trang sau khi nhận từ server.

#### **CLIENT SIDE RENDERING**

#### CSR hoạt động như thế nào:

- Khi truy cập trang web, trình duyệt tải file HTML rỗng
- sau đó tải và chạy JavaScript để render nội dung, có thể làm trang chậm hơn trong lần tải đầu tiên.
- Bot google cũng sẽ không đọc được nội dung

#### Ví dụ:

```
<!-- index.html của Vue.js -->
<!DOCTYPEhtml>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Vue App</title>
</head>
<body>
<!-- Element này sẽ được Vue render sau khi JavaScript chạy -->
<div id="app"></div>
<!-- JavaScript tải và thực thi -->
<script src="app.js"></script>
</body>
</html>
```

#### Sau khi JavaScript chay:

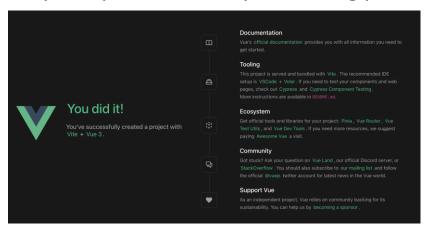
```
<divid="app">
<h1>Trang chủ</h1>
Chào mừng bạn đến với trang của chúng tôi!
</div>
```

#### Kết quả ban đầu code của trang:

```
<div id="app"></div><!-- Chưa có nội dung -->
```

#### **CLIENT SIDE RENDERING**

Tuy kết quả hiển thị đầy đủ, nhưng phải chờ JS chạy thì mới render nội dung.



ctrl + u để kiểm tra mã nguồn, bot google sẽ không tìm thấy nội dung nếu chỉ có mỗi div id=app



### **SERVER SIDE RENDERING**

#### SSR hoạt động như thế nào:

- Khi người dùng truy cập trang, nội dung đã được render trên server và gửi về trình duyệt dưới dạng HTML hoàn chỉnh
- Sau đó, JavaScript tải về để trang trở nên tương tác, giúp cải thiện tốc độ tải trang ban đầu và SEO.

#### Ví dụ:

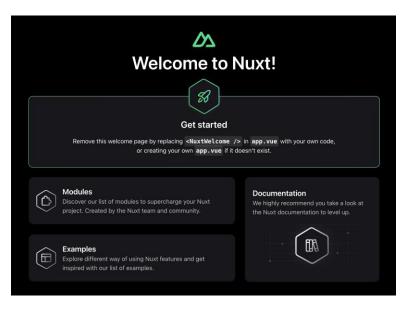
```
<!-- index.html được render trên server bởi Nuxt.js -->
<!DOCTYPEhtml>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <title>Nuxt App</title>
</head>
<body>
<!-- Nôi dung đã được server render -->
<div id=" nuxt">
        <div id="app">
                <h1>Trang chủ</h1>
                Chào mừng bạn đến với trang của chúng tôi!
        </div>
</div>
<!-- JavaScript tải sau để trang có thể tương tác -->
<script src="app.js"></script>
</body>
</html>
```

#### Kết quả ban đầu code của trang:

```
<divid="app">
<h1>Trang chủ</h1>
Chào mừng bạn đến với trang của chúng tôi!
</div>
<!-- Đã có nội dung đầy đủ ngay từ đầu -->
```

#### **SERVER SIDE RENDERING**

#### Kết quả hiển thị



#### ctrl + u để xem code

# TẠI SAO LẠI SỬ DỤNG SSR?

- Tải trang nhanh: Cung cấp HTML đã render, người dùng thấy nội dung ngay lập tức.
- **2. Tối ưu SEO**: Nội dung dễ dàng được crawl, cải thiện thứ hạng tìm kiếm.
- 3. Trải nghiệm người dùng tốt: Giảm thời gian chờ, tăng tính tương tác.
- **4. Quản lý trạng thái hiệu quả:** Dữ liệu được tải đồng bộ từ server.
- 5. Bảo mật: Ấn thông tin nhạy cảm, xử lý dữ liệu trên server.
- **6. Khả năng tương thích:** Hỗ trợ trình duyệt cũ không tốt về JavaScript.





Bước 1: Cài đặt Node.js

Bước 2: Tạo dự án Nuxt

- Mở terminal (Command Prompt hoặc PowerShell trên Windows, Terminal trên macOS hoặc Linux).
- 2. Chạy lệnh sau để tạo một dự án Nuxt mới:

npx nuxi@latest init <project-name>

Bước 3: Di chuyển vào thư mục dự án

cd <project-name>

Bước 4: Cài đặt các package và chạy dự án

npm i npm run dev

Bước 5: Truy cập http://localhost:3000 trong trình duyệt.



# **CẤU TRÚC THƯ MỤC**

- > 🔜 .nuxt
- > Fr components
- > 📜 layouts
- > middleware
- > node\_modules
- > to pages
- > **plugins**
- > m public
- > 📴 server
- > == store
  - .gitignore
  - ▼ app.vue
  - nuxt.config.ts
  - package-lock.json
  - package.json
  - README.md
  - tsconfig.json

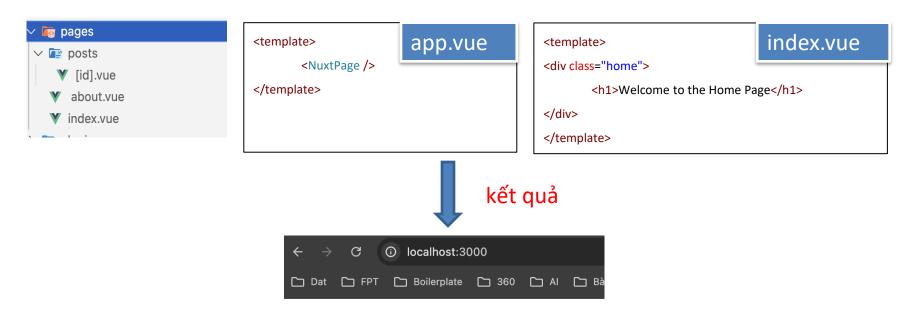
- public: Thư mục này chứa các file tĩnh mà sẽ được phục vụ trực tiếp bởi server.
- > server: Thư mục này chứa các file cấu hình và mã nguồn cho server.
- > app.vue: File Vue chính của ứng dụng.
- nuxt.config.ts: File cấu hình chính cho Nuxt.js.
- components: Chứa các thành phần Vue.js có thể tái sử dụng.
- layouts: Chứa các layout cho ứng dụng của bạn.
- middleware: Chứa các middleware để xử lý các yêu cầu trước khi chúng đến các trang hoặc API.
- pages: Chứa các trang của ứng dụng, mỗi file trong thư mục này sẽ tương ứng với một route.
- plugins: Chứa các plugin để thêm các chức năng vào ứng dụng của bạn.
- store: Chứa các module Vuex để quản lý trạng thái của ứng dụng.

Lưu ý: các thư mục có màu đỏ là được bổ sung thêm



**Routing** trong Nuxt.js dựa trên **vue-router** và tự động tạo các route từ từng component được tạo trong thư mục pages/, dựa trên tên file của chúng.

Hệ thống routing dựa trên file này sử dụng quy ước đặt tên để tạo ra các route động và lồng nhau:



Welcome to the Home Page





- Component < NuxtLink > liên kết các trang trong ứng dụng Nuxt.js.
- Thay thế thẻ <a> thông thường, cho phép chuyển trang mà không tải lại toàn bộ trang
- ☐ Giúp tăng tốc độ điều hướng.

```
<template>
                                                          app.vue
<div>
       <header>
       <nav>
              <NuxtLink to="/">Trang chủ</NuxtLink>
                     <NuxtLink to="/about">Giới thiệu</NuxtLink>
                     <NuxtLink to="/posts/2">Bài viết</NuxtLink>
              </nav>
       </header>
       <NuxtPage />
</div>
</template>
```



Thực hiện cài đặt nuxt.js và tạo các page demo



# PHAN 2: TRIÊN KHAI DỰ ÁN





**Netlify** là một nền tảng cung cấp dịch vụ triển khai website tĩnh và ứng dụng web với quy trình tự động hoá từ lúc xây dựng đến khi phát hành. Netlify hỗ trợ việc xây dựng, quản lý và phân phối các trang web hiện đại dựa trên kiến trúc JAMstack (JavaScript, APIs, Markup), giúp phát triển web trở nên nhanh chóng và dễ dàng hơn.





Để triển khai một dự án Vue.js lên **Netlify**, bạn có thể làm theo các bước sau:

#### Bước 1: Chuẩn bị dự án Vue.js

Trước khi deploy, đảm bảo rằng dự án Vue.js của bạn đã sẵn sàng và có thể build thành phiên bản production. Nếu bạn chưa tạo một dự án Vue, có thể tạo bằng lệnh sau:

#### npm create vue@latest

Cài đặt các dependencies và kiểm tra ứng dụng chạy thành công:

npm install npm run dev



# ĐẨY CODE LÊN GITHUB

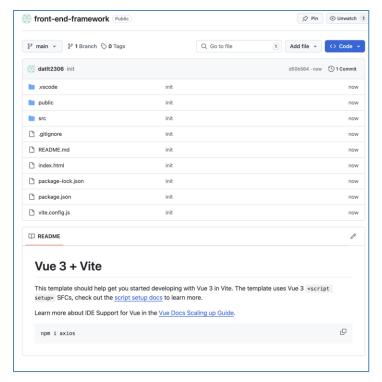
#### Bước 2: Đẩy code lên GitHub:

Khởi tạo Git trong thư mục dự án của bạn (nếu chưa có):

git init git add . git commit -m "Initial commit"

Thêm remote repository và đẩy code lên GitHub:

git remote add origin <URL-repo-của-bạn> git branch -M main git push -u origin main



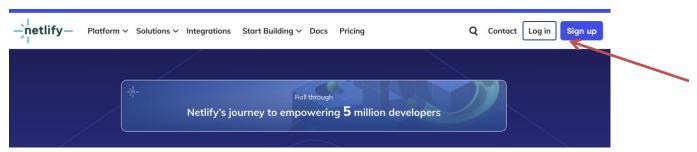


## KÉT NŐI NETLIFY VỚI GITHUB

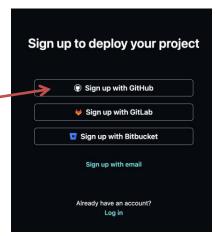
Bước 3: Kết nối Netlify với GitHub

Đăng nhập Netlify:

Truy cập <u>Netlify.com</u> và đăng nhập hoặc tạo tài khoản nếu bạn chưa có.



Bạn có thể sử dụng tài khoản github để đăng ký mới tài khoản Netlify

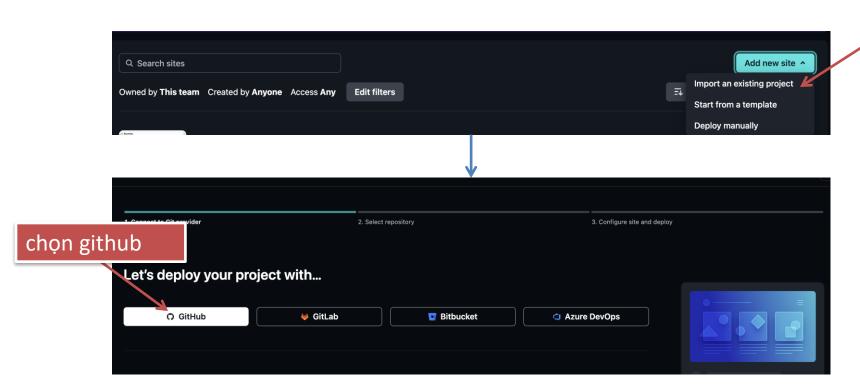




# KÉT NŐI NETLIFY VỚI GITHUB

> Tạo site mới từ GitHub:

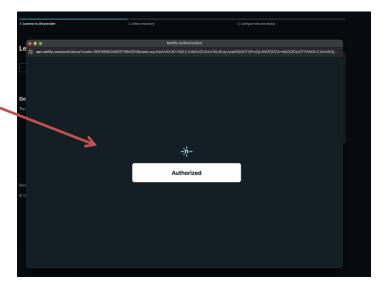
Sau khi đăng nhập, chọn New site from Git trên dashboard Netlify.



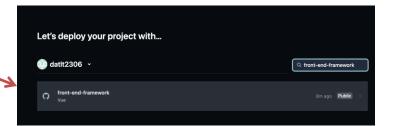


# KÉT NŐI NETLIFY VỚI GITHUB

Cấp quyền cho Netlify để truy
 cập các repository của bạn.



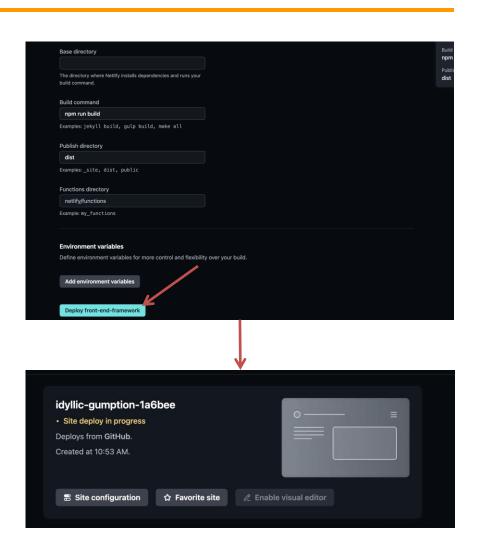
Chọn repository Vue.js





## **C**ẤU HÌNH THÔNG TIN BUILD

- Netlify sẽ hiển thị một bảng để cấu hình thông tin build.
- Ở phần Build command, nhập:
  npm run build
- Ở phần Publish directory nhập: dist
- Nhấn Deploy site để bắt đầu quá trình build và deploy.

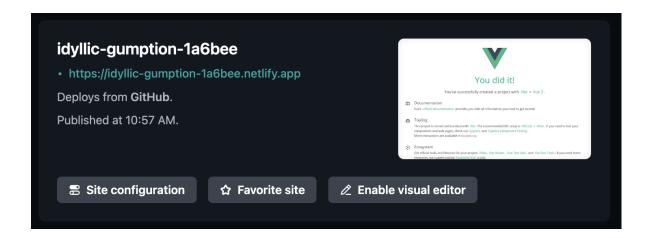




# HOÀN TẤT QUÁ TRÌNH TRIỂN KHAI

#### Bước 4: Hoàn tất

- Netlify sẽ tự động build dự án của bạn. Sau khi quá trình build hoàn thành,
   trang web của bạn sẽ được xuất bản và có thể truy cập bằng một URL do
   Netlify cung cấp
- (ví dụ: https://your-site-name.netlify.app).





# THỰC HÀNH LẠI CÁC BƯỚC HƯỚNG DẪN TRIỂN KHAI MỘT PROJECT TỪ GITHUB LÊN NETLIFY





- ✓ Phần 1
  - Giới thiệu framework Nuxt.js
  - Lợi ích khi sử dụng Nuxt.js
  - So sánh Nuxt.Js và Vue.Js
  - **♦**SSR và CSR
  - Cài đặt và sử dụng Nuxt.js
  - Routing: Pages và Navigation
- ✓ Phần 2
  - ❖Chuẩn bị
  - Đẩy code lên github
  - Két női github với netlify
  - ❖ Hoàn tất



