

ĐỒ ÁN MÔN HỌC

XÂY DỰNG WEBSITE BLOG CÁ NHÂN CHIA SẺ KIẾN THỨC LẬP TRÌNH MẠNG

Ngành: **CÔNG NGHỆ THÔNG TIN**
Môn : **LẬP TRÌNH MẠNG MÁY TÍNH**

Giảng viên hướng dẫn: **Ts. Nguyễn Quang Trung**

Sinh viên thực hiện: **Phạm Ngọc Trai 2280603326**

TP.Hồ Chí Minh, tháng 12 năm 2025

LỜI MỞ ĐẦU

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc tự học và chia sẻ kiến thức đã trở thành một kỹ năng quan trọng đối với sinh viên ngành Công nghệ thông tin. Đặc biệt, trong lĩnh vực **lập trình mạng**, kiến thức không chỉ dừng lại ở lý thuyết mà còn cần được củng cố thông qua thực hành, nghiên cứu tài liệu và trình bày lại theo cách dễ hiểu.

Xuất phát từ yêu cầu của học phần, em thực hiện đồ án **xây dựng Blog cá nhân chia sẻ kiến thức về lập trình mạng**, tập trung vào các nội dung đã học liên quan đến **Java** và **JavaScript**. Blog không chỉ đóng vai trò là nơi tổng hợp và hệ thống hóa kiến thức, mà còn là môi trường để rèn luyện kỹ năng trình bày, tư duy logic, cũng như khả năng triển khai một sản phẩm web hoàn chỉnh theo hướng thực tế.

Trang Blog được xây dựng theo mô hình **Static Site Generator (SSG)**, sử dụng GitHub Repository để quản lý mã nguồn và triển khai website. Nội dung Blog bao gồm trang giới thiệu hồ sơ cá nhân (Profile) và các bài viết chuyên đề về Java & JavaScript, được trình bày theo phong cách **tối giản, rõ ràng, dễ đọc**, phù hợp với mục tiêu chia sẻ kiến thức học thuật.

Thông qua đồ án này, em không chỉ củng cố lại các kiến thức đã học trong môn Lập trình mạng, mà còn nâng cao kỹ năng tự nghiên cứu, tổ chức nội dung và triển khai một sản phẩm công nghệ hoàn chỉnh. Đây cũng là bước chuẩn bị cần thiết cho các học phần chuyên ngành và định hướng phát triển cá nhân trong lĩnh vực công nghệ thông tin sau này.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Phạm Ngọc Trai

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án học phần **xây dựng Blog cá nhân chia sẻ kiến thức về lập trình mạng**, em đã nhận được sự hướng dẫn, hỗ trợ và tạo điều kiện từ giảng viên bộ môn cũng như từ nhà trường.

Em xin gửi lời cảm ơn chân thành đến **Tiến Sĩ Nguyễn Quang Trung** đã tận tình hướng dẫn, định hướng nội dung và góp ý trong suốt quá trình học tập và thực hiện đồ án, giúp em hiểu rõ hơn về các kiến thức liên quan đến lập trình mạng cũng như cách áp dụng chúng vào thực tế.

Bên cạnh đó, em cũng xin cảm ơn **nha truong va khoa** đã tạo môi trường học tập thuận lợi, cung cấp nền tảng kiến thức cần thiết để em có thể hoàn thành đồ án này một cách đầy đủ và đúng yêu cầu đề ra.

Mặc dù đã cố gắng hoàn thiện đồ án trong khả năng của mình, nhưng do thời gian và kinh nghiệm còn hạn chế, báo cáo không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ giảng viên để có thể cải thiện và hoàn thiện hơn trong quá trình học tập và nghiên cứu sau này.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 26 tháng 12 năm 2025

MỤC LỤC

LỜI MỞ ĐẦU	ii
LỜI CẢM ƠN	iii
MỤC LỤC	iv
DANH MỤC HÌNH	v
CHƯƠNG 1: TỔNG QUAN.....	1
1.1 ĐẶT VẤN ĐỀ VÀ LÝ DO CHỌN ĐỀ TÀI	1
1.2 MỤC TIÊU VÀ YÊU CẦU ĐỀ TÀI.....	1
1.2.1 Yêu cầu về Cấu trúc và Chức năng (Site Architecture)	1
1.2.2 Yêu cầu về Nội dung (Content Strategy)	2
1.2.3 Yêu cầu Phi chức năng (Non-functional):.....	2
1.3 PHẠM VI NGHIÊN CỨU.....	2
1.3.1 Công nghệ xây dựng Website:.....	2
1.3.2 Hệ thống hóa kiến thức Lập trình.....	3
CHƯƠNG 2: Lê Quốc Trọng_Blog	4
2.1 TRANG HOME.....	4
2.2 TRANG GIỚI THIỆU	4
2.3 Trang Blog	5
2.3.1 Bài post về Java&JavaScrpit	6
2.2.1 Trang Liên Hệ	7

DANH MỤC HÌNH

Hình 2.1: Trang home	4
Hình 2.2: Trang giới thiệu.	4
Hình 2.3: Trang Blog.....	5
Hình 2.4: Bài post về Java&JavaScript.	6
Hình 2.5: Trang Liên Hệ.....	7

CHƯƠNG 1: TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ VÀ LÝ DO CHỌN ĐỀ TÀI

Trong bức tranh toàn cảnh của ngành công nghiệp phần mềm hiện đại, Java và JavaScript luôn giữ vững vị thế là hai trụ cột công nghệ quan trọng nhất. Nếu như Java được xem là "xương sống" vững chắc cho các hệ thống Backend doanh nghiệp nhờ tính bảo mật và đa luồng mạnh mẽ, thì JavaScript lại là "ngôn ngữ của Web", thống trị mảng Frontend và mang lại trải nghiệm tương tác cho người dùng.

Đối với sinh viên và lập trình viên, việc làm chủ song song cả hai ngôn ngữ này là bước đệm thiết yếu để hướng tới lộ trình Full-stack Developer. Tuy nhiên, khối lượng kiến thức của hai ngôn ngữ này là không lồ và liên tục cập nhật. Việc học tập thụ động qua sách vở thường không hiệu quả bằng phương pháp "Learning in Public" – tức là vừa học, vừa ghi chép và chia sẻ lại kiến thức đó.

Xuất phát từ nhu cầu xây dựng một kho lưu trữ kiến thức cá nhân, đồng thời rèn luyện kỹ năng xây dựng giao diện web, tôi quyết định thực hiện đề tài: "Thiết kế và Xây dựng Blog cá nhân chia sẻ kiến thức lập trình Java & JavaScript".

Đề tài này không chỉ dừng lại ở việc tạo ra một website tin tức thông thường, mà còn hướng đến việc xây dựng một không gian học thuật số hóa, nơi các kiến thức về cú pháp, thuật toán và các mẫu thiết kế (Design Patterns) của Java/JavaScript được trình bày khoa học, dễ tra cứu và tiếp cận.

1.2 MỤC TIÊU VÀ YÊU CẦU ĐỀ TÀI

1.2.1 Yêu cầu về Cấu trúc và Chức năng (Site Architecture)

- Trang chủ (Home):** Là "bộ mặt" của website, hiển thị tóm tắt thông tin nổi bật, các bài viết mới nhất và định hướng điều hướng rõ ràng cho người truy cập.
- Trang Blog:** Đây là trang lõi của hệ thống, nơi danh sách các bài viết được hiển thị theo dạng lưới (Grid) hoặc danh sách (List). Cần hỗ trợ tính năng phân loại theo chủ đề (Category):

Java Core, JS ES6, Frameworks...) để người đọc dễ dàng lọc nội dung.

- **Trang Giới thiệu (About Me):** Trình bày hồ sơ năng lực (Portfolio), kinh nghiệm học tập và định hướng nghề nghiệp của tác giả. Đây là phần quan trọng để xây dựng thương hiệu cá nhân.
- **Trang Liên hệ (Contact):** Cung cấp biểu mẫu (Form) hoặc thông tin kết nối (Email, LinkedIn, GitHub) để độc giả hoặc nhà tuyển dụng có thể tương tác trực tiếp với tác giả.

1.2.2 Yêu cầu về Nội dung (Content Strategy)

- **Java:** Các bài viết về lập trình hướng đối tượng (OOP), xử lý ngoại lệ, Java Collections và đa luồng.
- **JavaScript:** Các bài viết về thao tác DOM, xử lý bất đồng bộ (Async/Await, Promises) và các cú pháp hiện đại (ES6+).
- **Trình bày:** Các đoạn mã nguồn (Code Block) phải được hiển thị rõ ràng (Syntax Highlighting), dễ đọc và dễ sao chép.

1.2.3 Yêu cầu Phi chức năng (Non-functional):

- **Giao diện (UI/UX):** Thiết kế tối giản, hiện đại, tập trung vào trải nghiệm đọc.
- **Responsive:** Hiển thị tốt trên mọi thiết bị từ Desktop, Tablet đến Mobile.
- **Tốc độ:** Tối ưu hóa thời gian tải trang để giảm tỷ lệ thoát trang (Bounce rate).

1.3 PHẠM VI NGHIÊN CỨU

1.3.1 Công nghệ xây dựng Website:

- Tìm hiểu và ứng dụng HTML5, CSS3 để xây dựng bộ khung giao diện.
- Sử dụng Static Site Generator (Hugo) để quản lý nội dung và tạo trang tĩnh nhằm tối ưu bảo mật và tốc độ.

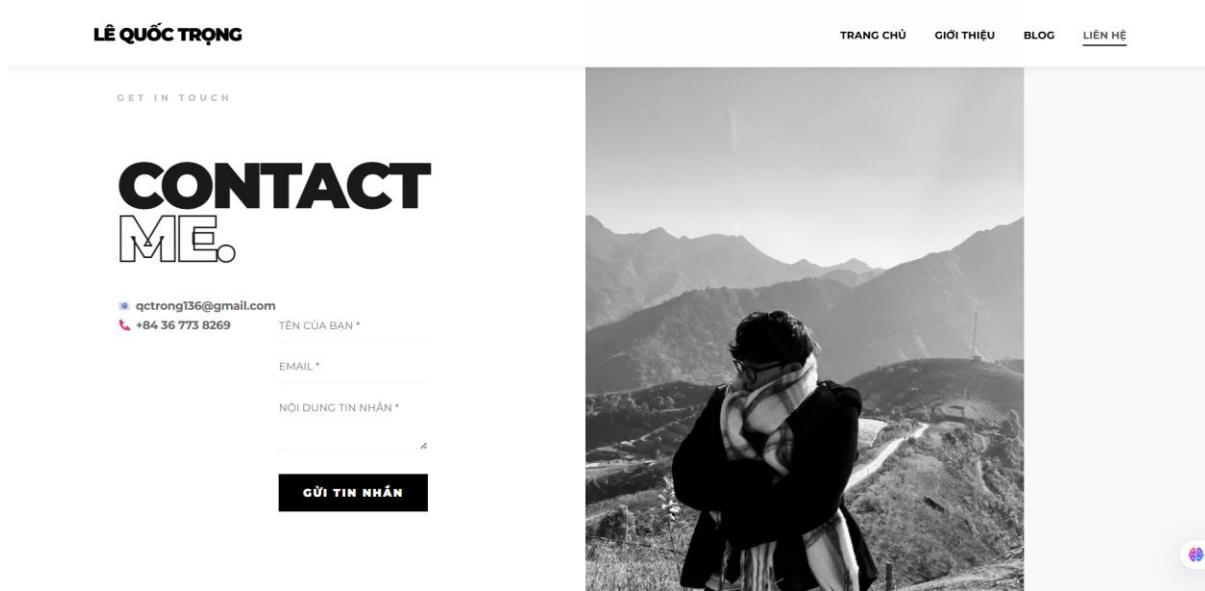
-
- Tích hợp Markdown để soạn thảo bài viết một cách nhanh chóng và chuẩn hóa.
 - Triển khai (Deploy) website lên môi trường Internet (GitHub Pages/Netlify).

1.3.2 Hệ thống hóa kiến thức Lập trình

- Nghiên cứu tài liệu chính thống (Documentation) của Oracle (cho Java) và MDN Web Docs (cho JavaScript).
- Biên soạn lại các khái niệm phức tạp thành các bài hướng dẫn (Tutorials) dễ hiểu, đi kèm ví dụ thực tế.
- Phân tích sự khác biệt và điểm tương đồng giữa Java (Static typing) và JavaScript (Dynamic typing) trong quá trình phát triển ứng dụng.

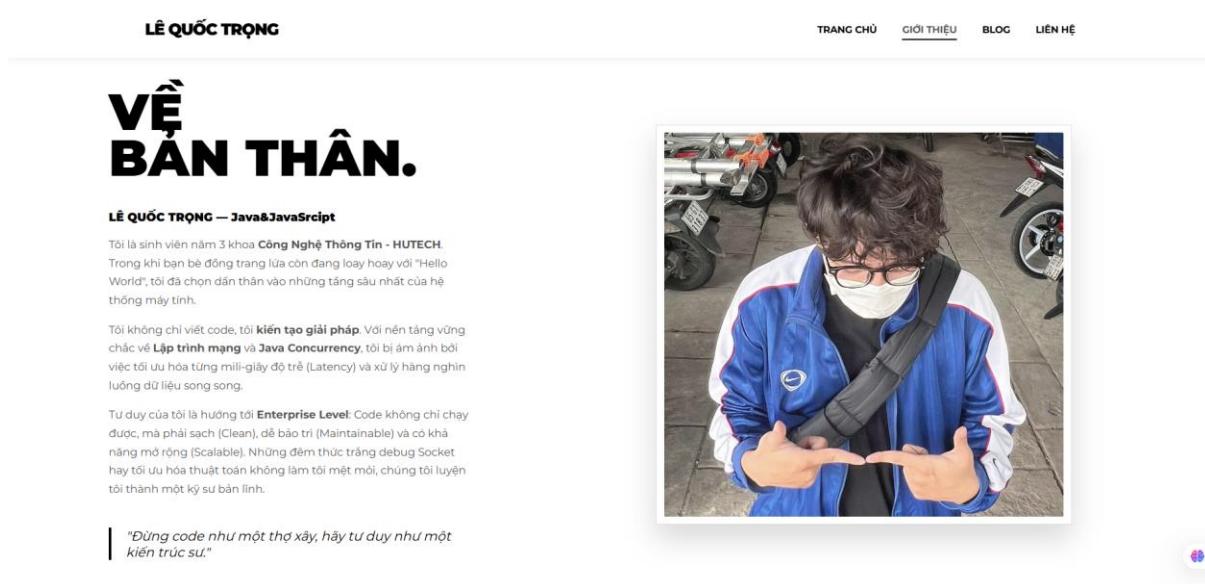
CHƯƠNG 2: Lê Quốc Trọng_Blog

2.1 TRANG HOME



Hình 2.1: Trang home

2.2 TRANG GIỚI THIỆU



Hình 2.2: Trang giới thiệu.

2.3 Trang Blog

The screenshot shows a blog page with the title "Java&JavaScript." at the top. Below the title, there is a list of ten posts, each with a date, a title, a brief description, and a "Xem bài ..." link.

- 15 Dec 2025: **Bài 1: Làm sao biết máy mình đang dùng IP gì trong Java?**
Hướng dẫn sử dụng class InetAddress để lấy thông tin IP và tên máy tính (localhost).
[Xem bài ...](#)
- 16 Dec 2025: **Bài 2: Port là gì? Tại sao Server hay dùng cổng 8080?**
Hiểu về khái niệm Port (Cổng) để tránh xung đột khi chạy phần mềm.
[Xem bài ...](#)
- 17 Dec 2025: **Bài 3: Callback Hell trong JavaScript và cách né tránh**
Viết code JS lồng nhau quá nhiều sẽ tạo ra thảm họa.
[Xem bài ...](#)
- 18 Dec 2025: **Bài 4: Hướng dẫn gửi File ảnh qua Socket trong Java**
Gửi text thì dễ, gửi file nhị phân (ảnh, nhạc) thì phải dùng FileInputStream.
[Xem bài ...](#)
- 19 Dec 2025: **Bài 5: WebSocket - Công nghệ sau các ứng dụng Chat Realtime**
Tại sao Facebook, Zalo lại dùng WebSocket thay vì HTTP thông thường?
[Xem bài ...](#)
- 20 Dec 2025: **Bài 6: Phân biệt Blocking IO và Non-Blocking IO**
Khai niệm quan trọng để tối ưu hiệu năng Server.
[Xem bài ...](#)
- 21 Dec 2025: **Bài 7: Xử lý dữ liệu JSON trong JavaScript**
Hai hàm quan trọng nhất khi làm việc với API.
[Xem bài ...](#)
- 22 Dec 2025: **Bài 8: Xử lý ngoại lệ (Exception) khi lập trình mạng**
Mạng rớt, Server sập... làm sao để chương trình không bị Crash?
[Xem bài ...](#)
- 23 Dec 2025: **Bài 9: Đa luồng (Multithreading) - Cách để Server tiếp 100 khách cùng lúc**
Nếu không có Đa luồng, Server của bạn chỉ phục vụ được đúng 1 người, người thứ 2 sẽ bị treo. Tại sao?
[Xem bài ...](#)
- 23 Dec 2025: **Bài 10: Tổng kết và Đánh giá kết quả Đỗ án môn Lập trình mạng**
Báo cáo tổng kết quá trình thực hiện, các kết quả đạt được và kiến thức tích lũy từ đồ án xây dựng Blog cá nhân.
[Xem bài ...](#)

Hình 2.3: Trang Blog.

2.3.1 Bài post về Java&JavaScrpit

LÊ QUỐC TRỌNG

TRANG CHỦ GIỚI THIỆU BLOG LIÊN HỆ

15 JANUARY 2025 — [JAVA NETWORK LẬP TRÌNH MẠNG]

Bài 1: Làm sao biết máy mình đang dùng IP gì trong Java?

Trong lập trình mạng, "địa chỉ nhà" (IP Address) là thông tin cơ bản nhất để các thiết bị có thể tìm thấy và giao tiếp với nhau. Java cung cấp một class cực kỳ mạnh mẽ để xử lý việc này là `java.net.InetAddress`.

Bài viết này sẽ hướng dẫn bạn cách lấy **Tên máy** và **Địa chỉ IP** của chính máy tính bạn đang sử dụng (localhost).

1. Class InetAddress là gì?

`InetAddress` là class đại diện cho một địa chỉ giao thức Internet (IP). Nó có thể xử lý cả IPv4 và IPv6. Điểm đặc biệt là class này **không có Public Constructor**, nghĩa là bạn không thể tạo đối tượng bằng từ khóa `new`. Thay vào đó, bạn phải dùng các phương thức tĩnh (static methods) như:

```
getLocalHost(): Lấy thông tin máy hiện tại.  
getByName(String host): Lấy thông tin từ tên miền (ví dụ: "google.com").  
getAllByName(String host): Lấy tất cả IP của một tên miền.
```

2. Code thực hành: Lấy IP của máy (localhost)

Dưới đây là đoạn code đơn giản để in ra tên máy và địa chỉ IP hiện tại.

```
import java.net.InetAddress;  
  
public class MyIP {  
    public static void main(String[] args) {  
        try {  
            // Lấy đối tượng InetAddress đại diện cho máy local  
            InetAddress myIP = InetAddress.getLocalHost();  
  
            // In tên máy (Host Name)  
            System.out.println("Tên máy: " + myIP.getHostName());  
  
            // In địa chỉ IP (Host Address)  
            System.out.println("Địa chỉ IP: " + myIP.getHostAddress());  
  
        } catch (Exception e) {  
            // In ra lỗi nếu không lấy được thông tin (ví dụ: lỗi mạng)  
            e.printStackTrace();  
        }  
    }  
}
```

Trong lập trình mạng, "địa chỉ nhà" (IP Address) là thông tin cơ bản nhất để các thiết bị có thể tìm thấy và giao tiếp với nhau. Java cung cấp một class cực kỳ mạnh mẽ để xử lý việc này là `java.net.InetAddress`.

Bài viết này sẽ hướng dẫn bạn cách lấy **Tên máy** và **Địa chỉ IP** của chính máy tính bạn đang sử dụng (localhost).

1. Class InetAddress là gì?

`InetAddress` là class đại diện cho một địa chỉ giao thức Internet (IP). Nó có thể xử lý cả IPv4 và IPv6. Điểm đặc biệt là class này **không có Public Constructor**, nghĩa là bạn không thể tạo đối tượng bằng từ khóa `new`. Thay vào đó, bạn phải dùng các phương thức tĩnh (static methods) như:

```
getLocalHost(): Lấy thông tin máy hiện tại.  
getByName(String host): Lấy thông tin từ tên miền (ví dụ: "google.com").  
getAllByName(String host): Lấy tất cả IP của một tên miền.
```

2. Code thực hành: Lấy IP của máy (localhost)

Dưới đây là đoạn code đơn giản để in ra tên máy và địa chỉ IP hiện tại.

```
import java.net.InetAddress;  
  
public class MyIP {  
    public static void main(String[] args) {  
        try {  
            // Lấy đối tượng InetAddress đại diện cho máy local  
            InetAddress myIP = InetAddress.getLocalHost();  
  
            // In tên máy (Host Name)  
            System.out.println("Tên máy: " + myIP.getHostName());  
  
            // In địa chỉ IP (Host Address)  
            System.out.println("Địa chỉ IP: " + myIP.getHostAddress());  
  
        } catch (Exception e) {  
            // In ra lỗi nếu không lấy được thông tin (ví dụ: lỗi mạng)  
            e.printStackTrace();  
        }  
    }  
}
```

Hình 2.4: Bài post về Java&JavaScrpit.

2.2.1 Trang Liên Hệ

LÊ QUỐC TRỌNG

TRANG CHỦ GIỚI THIỆU BLOG LIÊN HỆ

GET IN TOUCH

CONTACT ME.

qctrong136@gmail.com
 +84 36 773 8269

TÊN CỦA BẠN *

EMAIL *

NỘI DUNG TIN NHẮN *



A small circular icon with a blue and red design is located in the bottom right corner of the image area.

Hình 2.5: Trang Liên Hệ.