TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

LẬP TRÌNH MẠNG

Giảng viên hướng dẫn: ThS. Nguyễn Quang Trung

Sinh viên thực hiện: Trương Hải Nam

MSSV: 2280602042 Lớp: 22DTHE5

TP. Hồ Chí Minh, 2025

Mục Lục

Các chứng chỉ Cisco

Networking Academy

Certificate of Course Completion

Truong Hai Nam

has successfully achieved student level credential for completing the **Networking Basics** *course.*

The student was able to proficiently:

- *Explain important concepts in network communication, network types, components, and connections.*
- *Explain the importance of standards and protocols in network communications.*
- *Explain how communication occurs on Ethernet networks.*
- *Explain the features of an IP address and IPv4 addresses are used in network communication.*
- *Explain features of IPv6 addressing.*
- *Explain how routers connect networks together.*
- *Use various tools to test and troubleshoot network connectivity.*
- *Configure an integrated wireless router and wireless client to connect securely to the internet.*

Verified

Networking Basics

Scan to Verify

Issued on: Nov 20, 2025

Lynn Bloomer

Lynn Bloomer
Director, Cisco Networking Academy

Networking Academy

JS INSTITUTE
Open Education & Development Group

Statement of Achievement

Truong Hai Nam

*has successfully achieved student level credential for completing the **JavaScript Essentials 1** course, provided by Cisco Networking Academy in collaboration with OpenEDG JavaScript Institute.*

The graduate is able to proficiently:

- *Understand the syntax of the core JavaScript language that allows for working with variables, operators, flow control, and functions.*
- *Understand the basics of the JavaScript data types system, distinguishing between primitive and complex types.*
- *Think algorithmically and can analyze a problem using a programmatic conceptual apparatus.*
- *Choose a data type adequate to the problem being solved and use suitable flow control means.*
- *Design, develop, and improve very simple JavaScript programs.*
- *Interpret and handle basic exceptions related to errors in program execution.*
- *Understand a programmer's work in the software development process and the role of fundamental development tools.*
- *Understand how a program is interpreted and executed in an actual computer environment, local or remote.*

Scan to Verify

Issued on: Nov 21, 2025

Lynn Bloomer
Director, Cisco Networking Academy

Networking Academy

JS INSTITUTE
Open Education & Development Group

Statement of Achievement

Truong Hai Nam

*has successfully achieved student level credential for completing the **JavaScript Essentials 2** course, provided by Cisco Networking Academy in collaboration with OpenEDG JavaScript Institute.*

The graduate has studied:

- *Techniques for constructing and modifying objects, including the use of prototypes and inheritance.*
- *Methods for defining and encapsulating class properties and managing array data, including JSON conversion.*
- *Utilization of the Math object and regular expressions for mathematical and string operations.*
- *Advanced function techniques and asynchronous programming, including callbacks and iterators.*
- *Problem analysis and program development using algorithmic thinking and object-oriented principles.*

Scan to Verify

Issued on: Nov 23, 2025

Lynn Bloomer

Lynn Bloomer
Director, Cisco Networking Academy

LỜI MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, việc khẳng định giá trị cá nhân và xây dựng thương hiệu số đã trở thành yếu tố then chốt đối với mỗi sinh viên ngành Công nghệ thông tin. Một hồ sơ năng lực trực tuyến (Portfolio) không chỉ đơn thuần là công cụ trưng bày các sản phẩm phần mềm, mà còn là minh chứng rõ nét cho quá trình rèn luyện, tư duy logic và khả năng làm chủ công nghệ của người thực hiện. Bên cạnh đó, việc tích hợp một trang Blog chia sẻ tri thức chuyên sâu còn giúp hệ thống hóa lại những kiến thức đã học, tạo ra một không gian giao lưu học thuật giá trị cho cộng đồng. Đặc biệt, trong phạm vi môn học Mạng máy tính, việc xây dựng một nền tảng web hội tụ đủ các yếu tố về hiệu suất, bảo mật và khả năng truyền tải dữ liệu là một thử thách thực tế giúp củng cố lý thuyết về các giao thức mạng cốt lõi.

Ứng dụng được phát triển dựa trên các công nghệ web hiện đại bao gồm HTML5, CSS3 cho giao diện, JavaScript cho các hiệu ứng tương tác, và định hướng tích hợp các công nghệ Backend như Java, Node.js để quản lý dữ liệu. Toàn bộ hệ thống được tổ chức theo cấu trúc phân khối khoa học, từ trang Kỹ năng chuyên môn, Danh mục dự án đa nền tảng (Desktop, Mobile, Web) cho đến chuyên mục Kiến thức mạng nâng cao. Các nội dung về giao thức TCP/IP, HTTP/3 hay bảo mật TLS 1.3 được trình bày trực quan thông qua thiết kế Accordion, giúp người xem dễ dàng tiếp cận những kiến thức mạng phức tạp một cách gọn gàng và logic nhất. Đề tài không chỉ mang ý nghĩa học thuật trong việc áp dụng kiến thức lập trình mạng vào thực tế, mà còn là một sản phẩm hoàn thiện giúp tác giả kết nối hiệu quả với các nhà tuyển dụng trong tương lai.

Mặc dù đã dành nhiều thời gian nghiên cứu và đầu tư tâm huyết cho dự án, song do kiến thức và kinh nghiệm thực tế còn những hạn chế nhất định, đồ án chắc chắn không tránh khỏi những thiếu sót. Tác giả rất mong nhận được những ý kiến đóng góp và nhận xét quý báu từ quý Thầy/Cô để đề tài được hoàn thiện hơn, cũng như rút ra được những bài học kinh nghiệm cho quá trình phát triển sự nghiệp sau này.

Xin chân thành cảm ơn!

Sinh viên thực hiện Trương Hải Nam

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn đến quý thầy cô Khoa Công Nghệ Thông Tin HUTECH đã nhiệt tình hỗ trợ cho chúng em những kiến thức về Công Nghệ Thông Tin để thực hiện đồ án này.

Bằng cách riêng, chúng em xin cảm ơn ThS. Nguyễn Quang Trung – người trực tiếp giúp đỡ và tạo điều kiện, cung cấp cho chúng em những kiến thức và các công nghệ cần thiết để hoàn thành đồ án này.

Cuối cùng, chúng em xin gửi lời cảm ơn đến gia đình, các anh chị và các bạn đã hỗ trợ cho chúng em rất nhiều trong suốt quá trình học tập, nghiên cứu và thực hiện đề tài đồ án một cách hoàn chỉnh. Chúng em xin kính chúc Ban Giám Hiệu nhà trường cùng quý Thầy Cô sức khoẻ, luôn vui vẻ và đạt nhiều thành công trong công việc.

TP. Hồ Chí Minh, ngày 27 tháng 12 năm 2025

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

1.1.1. Bối cảnh thị trường lao động CNTT và sự cần thiết của thương hiệu cá nhân

Trong thập kỷ qua, cuộc Cách mạng Công nghiệp 4.0 đã làm thay đổi hoàn toàn diện mạo của thị trường lao động toàn cầu, đặc biệt là trong lĩnh vực Công nghệ thông tin (CNTT). Với sự bùng nổ của các công nghệ mới như AI, Cloud Computing và Big Data, nhu cầu tuyển dụng lập trình viên tăng cao, nhưng đi kèm với đó là sự sàng lọc vô cùng khắt khe.

Đối với các sinh viên mới tốt nghiệp hoặc lập trình viên trẻ, một bản CV truyền thống dưới dạng PDF hay bản in cứng đã trở nên quá cũ kỹ và hạn chế về mặt không gian để trình diễn năng lực. Nhà tuyển dụng hiện nay không chỉ quan tâm đến bằng cấp mà họ muốn thấy **"sản phẩm thực tế"** và **"tư duy thực tế"**. Website Portfolio đóng vai trò là một "văn phòng làm việc trực tuyến", nơi tác giả có thể trưng bày toàn bộ hành trình học tập, các dự án đã thực hiện và minh chứng rõ ràng nhất cho kỹ năng Frontend của mình.

1.1.2. Nhu cầu hệ thống hóa kiến thức thông qua Blog công nghệ

Trong quá trình học tập tại giảng đường, lượng kiến thức về các môn học cốt lõi như **Lập trình hướng đối tượng (Java)**, **Lập trình JavaScript** và **Mạng máy tính** là rất lớn và có tính trừu tượng cao. Thực tế cho thấy, sinh viên thường gặp khó khăn trong việc kết nối các khái niệm lý thuyết với ứng dụng thực tiễn.

Việc xây dựng một chuyên mục Blog tích hợp ngay trên Portfolio không chỉ giúp chia sẻ giá trị cho cộng đồng mà còn là phương pháp **"Learning by Teaching" (Học bằng cách dạy lại)**. Khi viết ra các bài blog về Socket TCP/UDP, Đa luồng trong Java hay Cơ chế Hoisting trong JS, người viết buộc phải nghiên cứu sâu hơn, hệ thống hóa lại tư duy một

cách mạch lạc nhất. Đây chính là cách tốt nhất để biến kiến thức của nhân loại thành tri thức của bản thân.

1.1.3. Tính cấp thiết của đề tài

Từ những thực tế trên, việc nghiên cứu và xây dựng một nền tảng Web vừa làm Portfolio giới thiệu bản thân, vừa làm Blog chia sẻ tri thức là một nhu cầu cực kỳ cấp thiết. Đề tài không chỉ dừng lại ở việc tạo ra một giao diện đẹp mà còn là bài toán về **tối ưu hóa trải nghiệm người dùng (UX)** và **quản trị nội dung khoa học**. Vì vậy, em đã chọn đề tài: **"Xây dựng Website Portfolio cá nhân tích hợp hệ thống Blog chia sẻ kiến thức công nghệ"** để thực hiện đồ án của mình.

1.2. Mục tiêu của đề tài

1.2.1. Mục tiêu về tri thức và kỹ năng

- **Nâng cao năng lực Frontend:** Vận dụng các công nghệ HTML5, CSS3 chuẩn SEO và JavaScript ES6+ để hiện thực hóa giao diện từ bản vẽ phác thảo đến mã nguồn thực tế.
- **Tối ưu hóa đa thiết bị:** Áp dụng kỹ thuật Responsive Web Design (RWD) để đảm bảo website hoạt động hoàn hảo trên mọi kích thước màn hình (Mobile, Tablet, Desktop).
- **Nghiên cứu về trải nghiệm người dùng (UX):** Thiết kế cấu trúc điều hướng thông minh, giúp người đọc dễ dàng tìm kiếm thông tin và bài viết.

1.2.2. Mục tiêu về sản phẩm hệ thống

- **Hoàn thiện bộ nhận diện cá nhân:** Xây dựng đầy đủ các trang: Trang chủ (Home), Kỹ năng (Skills), Dự án (Projects), Liên hệ (Contact).
- **Xây dựng kho nội dung số:** Thiết lập hệ thống 09 bài viết chuyên sâu bám sát chương trình đào tạo chuyên ngành, đảm bảo tính chính xác về mặt kỹ thuật và phong cách trình bày blog hiện đại.

- **Triển khai vận hành:** Quản lý dự án chuyên nghiệp bằng Git, thực hiện quy trình đẩy mã nguồn lên GitHub và duy trì khả năng mở rộng hệ thống trong tương lai.

1.3. Đối tượng và phạm vi nghiên cứu

1.3.1. Đối tượng nghiên cứu

- **Về công nghệ:** Nghiên cứu sâu về các thành phần cấu tạo nên Web tĩnh, các thư viện hỗ trợ (JQuery, FontAwesome) và các quy chuẩn CSS hiện đại.
- **Về nội dung:** Tập trung vào 3 mảng kiến thức chính:
    1. JavaScript căn bản và nâng cao.
    2. Lập trình Java Core và ứng dụng giao diện Swing.
    3. Giao thức truyền tải dữ liệu trong Mạng máy tính.

1.3.2. Phạm vi nghiên cứu

- **Giới hạn kỹ thuật:** Đồ án tập trung vào phía **Client-side (Frontend)**. Sử dụng các ngôn ngữ lập trình kịch bản để xử lý tương tác, chưa bao gồm các hệ quản trị nội dung (CMS) phía Server-side.
- **Giới hạn nội dung:** Bài viết tập trung chia sẻ kinh nghiệm đồ án và các kiến thức khó trong chương trình đại học mà sinh viên thường gặp lỗi.
- **Môi trường thực nghiệm:** Dự án được kiểm thử trên các trình duyệt phổ biến nhất hiện nay (Google Chrome, Microsoft Edge, Safari) và được triển khai lưu trữ công khai trên GitHub.

1.4. Phương pháp nghiên cứu và thực hiện

Để đạt được các mục tiêu đề ra, đồ án sử dụng kết hợp các nhóm phương pháp sau:

- **Phương pháp thu thập và tổng hợp tài liệu:**
    - Nghiên cứu tài liệu kỹ thuật từ các nguồn uy tín như MDN Web Docs, W3Schools, và các diễn đàn công nghệ quốc tế (Stack Overflow).

    - Tham khảo các mẫu thiết kế Portfolio từ các nền tảng nghệ thuật số như Pinterest, Behance để định hình phong cách cá nhân.
- **Phương pháp phân tích hệ thống:**
    - Phân tích nhu cầu của người dùng mục tiêu (Nhà tuyển dụng, cộng đồng sinh viên).
    - Thiết kế sơ đồ phân rã chức năng và luồng tương tác giữa các trang trong website.
- **Phương pháp thực nghiệm (Coding & Testing):**
    - Áp dụng quy trình phát triển phần mềm theo từng giai đoạn (Vòng đời phát triển phần mềm - SDLC).
    - Thực hiện việc mã hóa (Coding), sau đó tiến hành kiểm thử trên nhiều môi trường khác nhau để phát hiện và khắc phục các lỗi về giao diện (Bug CSS) và logic (Bug JavaScript).

1.5. Ý nghĩa của đề tài

1.5.1. Ý nghĩa đối với cá nhân

- Là minh chứng thực tế cho quá trình tự học và rèn luyện của sinh viên trong suốt 4 năm đại học.
- Giúp nâng cao khả năng viết lách, tư duy phản biện và kỹ năng giải thích các vấn đề kỹ thuật phức tạp một cách đơn giản.

1.5.2. Ý nghĩa đối với cộng đồng và nhà tuyển dụng

- **Với cộng đồng:** Cung cấp các bài viết chất lượng về Java và Mạng máy tính, giúp các sinh viên khóa sau có thêm nguồn tài liệu tham khảo thực tế từ chính trải nghiệm của người đi trước.
- **Với nhà tuyển dụng:** Giúp họ có cái nhìn trực quan, nhanh chóng về năng lực chuyên môn của ứng viên mà không cần thông qua các bài kiểm tra căn bản.

1.6. Cấu trúc của đồ án

Báo cáo đồ án bao gồm 05 chương với nội dung tóm tắt như sau:

- **Chương 1: Giới thiệu đề tài.** Trình bày tổng quan về lý do, mục tiêu, đối tượng và phương pháp nghiên cứu.
- **Chương 2: Cơ sở lý thuyết.** Giới thiệu các kiến thức nền tảng về Web (HTML, CSS, JS) và các giao thức mạng (TCP, UDP, HTTP) được sử dụng trong dự án.
- **Chương 3: Phân tích và Thiết kế hệ thống.** Chi tiết về Sitemap, Wireframe và cách phối màu, chọn font chữ cho Portfolio.
- **Chương 4: Xây dựng hệ thống và Demo giao diện.** Trình bày quá trình hiện thực hóa mã nguồn và các hình ảnh chụp màn hình thực tế của sản phẩm.
- **Chương 5: Kết luận và Hướng phát triển.** Tổng kết các kết quả đạt được, tự đánh giá ưu khuyết điểm và đề xuất lộ trình nâng cấp hệ thống lên Web động (Fullstack).

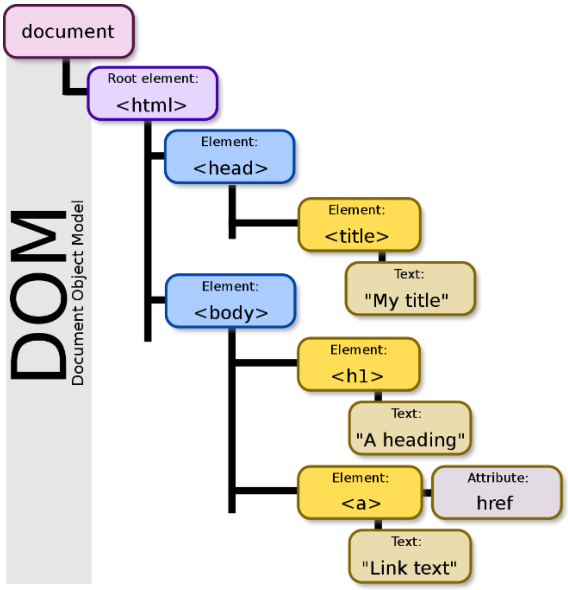
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về công nghệ phát triển ứng dụng Web (Frontend)

2.1.1. Ngôn ngữ cấu trúc HTML5 (HyperText Markup Language 5)

HTML5 là phiên bản mới nhất của ngôn ngữ đánh dấu siêu văn bản, được thiết kế để thay thế các tiêu chuẩn cũ (HTML4, XHTML) nhằm đáp ứng nhu cầu truyền tải đa phương tiện và ứng dụng web phức tạp.

- **Sự tiến hóa của HTML5:** HTML5 giới thiệu các API mạnh mẽ như Web Storage (thay thế Cookie cho việc lưu trữ lớn), Web Workers (xử lý đa luồng trên trình duyệt), và hỗ trợ gốc cho đa phương tiện (<video>, <audio>) mà không cần Plugin bên thứ ba như Flash.
- **HTML5 Semantic Web (Web ngữ nghĩa):** Đây là một bước tiến lớn giúp máy tính (Search Engines, Screen Readers) hiểu được nội dung trang web. Thay vì lạm dụng thẻ , HTML5 sử dụng các thẻ đặc thù:
    - <header>, <footer>: Xác định vùng đầu và cuối trang.
    - <nav>: Chứa các liên kết điều hướng.
    - <article>, <section>: Phân đoạn nội dung có tính độc lập và logic.
- **Cấu trúc cây DOM (Document Object Model):** DOM là giao diện lập trình cho tài liệu HTML. Khi trình duyệt tải trang, nó chuyển đổi HTML thành một cấu trúc cây các đối tượng. Mỗi node trong cây đại diện cho một phần tử, thuộc tính hoặc văn bản. Việc nắm vững DOM cho phép JavaScript truy cập, sửa đổi cấu trúc, phong cách và nội dung trang web một cách động.

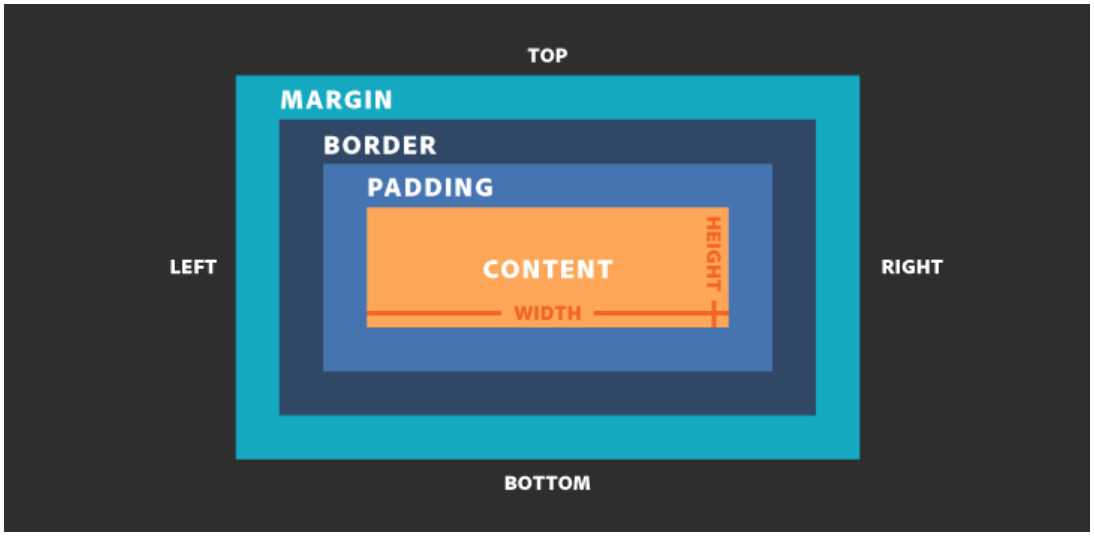


Hình 2.1. Cấu trúc cây đối tượng tài liệu (DOM Tree).

2.1.2. Ngôn ngữ định dạng CSS3 (Cascading Style Sheets)

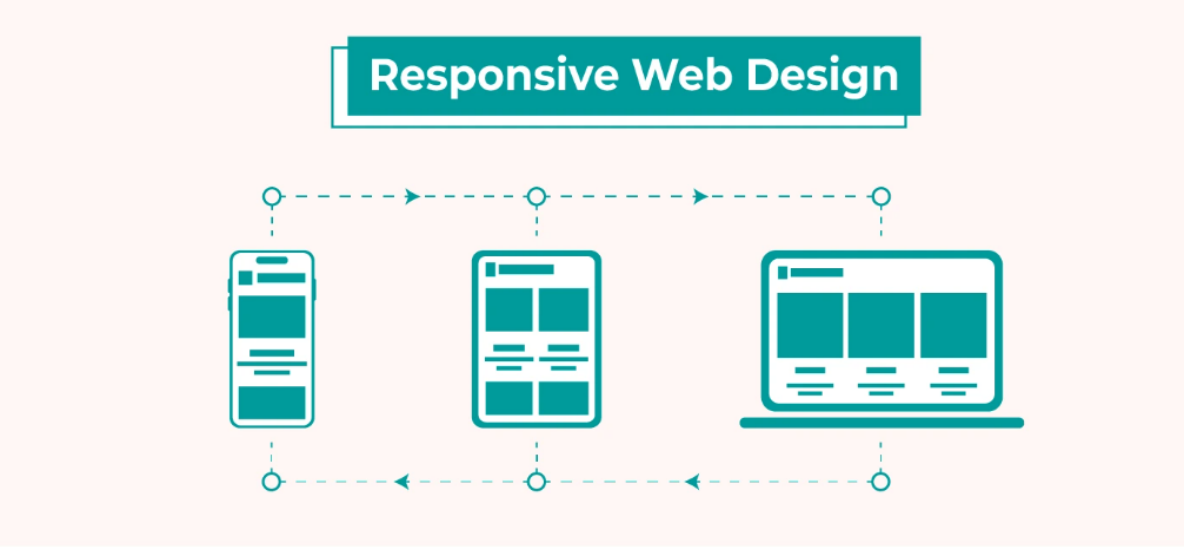
CSS3 không chỉ đơn thuần là trang trí, mà là một ngôn ngữ định dạng kiểu phân cấp cho phép tách biệt hoàn toàn nội dung (HTML) và hình thức trình bày.

- **Mô hình hộp (The Box Model):** Đây là nguyên lý cốt lõi của thiết kế giao diện Web. Mỗi phần tử được bao bọc bởi một hình hộp bao gồm:
    - **Content:** Nội dung thực tế (chữ, ảnh).
    - **Padding:** Khoảng đệm giữa nội dung và viền.
    - **Border:** Đường viền bao quanh phần đệm.
    - **Margin:** Khoảng cách lề ngoài giữa các phần tử với nhau.



Hình 2.2. Thành phần cấu tạo của CSS Box Model.

- **Bố cục nâng cao với Flexbox và CSS Grid:**
  - **Flexbox (Flexible Box):** Tập trung vào việc sắp xếp các phần tử dọc theo một chiều duy nhất (dòng hoặc cột). Nó cực kỳ hiệu quả trong việc căn giữa phần tử hoặc phân bổ không gian linh hoạt.
  - **CSS Grid:** Hệ thống bố cục hai chiều (hàng và cột), cho phép xây dựng các layout phức tạp như trang báo hoặc Portfolio một cách dễ dàng và sạch sẽ.
- **Nguyên lý Responsive Web Design (RWD):** Sử dụng **Media Queries** để phát hiện các thông số của thiết bị (độ phân giải, chiều rộng màn hình) để áp dụng các bộ quy tắc CSS khác nhau. Điều này đảm bảo website Portfolio hiển thị tốt trên cả máy tính để bàn (Desktop), máy tính bảng (Tablet) và điện thoại di động (Smartphone).



Hình 2.3. Minh họa thiết kế web đáp ứng (Responsive Design) trên đa thiết bị.

2.1.3. Ngôn ngữ lập trình JavaScript (ES6+)

JavaScript là ngôn ngữ lập trình kịch bản phía máy khách (Client-side), đóng vai trò tạo ra các tương tác động.

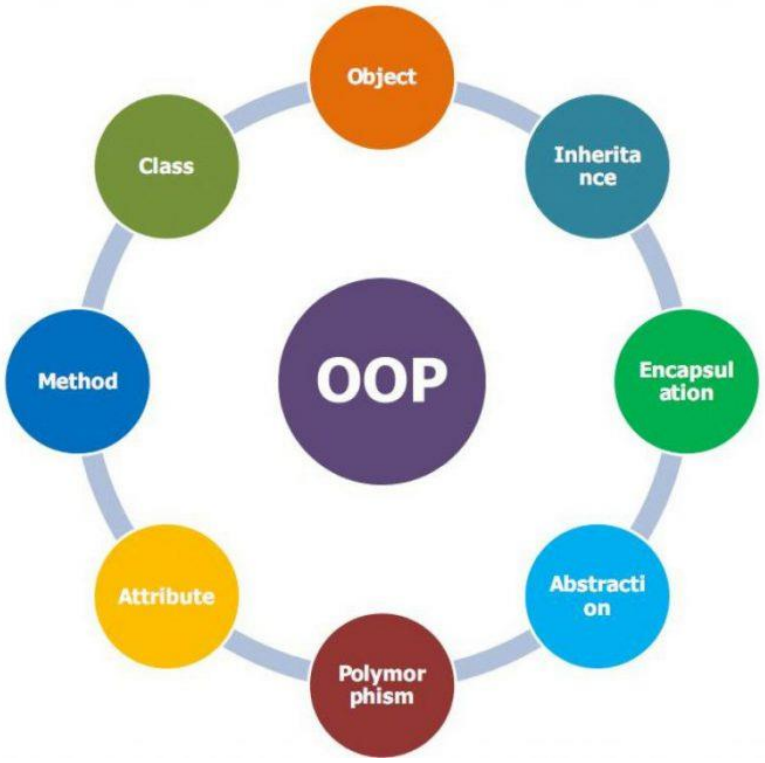
- **Cơ chế thực thi Single-threaded và Event Loop:** JavaScript chạy trên một luồng duy nhất (Single-thread), nghĩa là nó chỉ xử lý một lệnh tại một thời điểm. Tuy nhiên, nhờ vào **Event Loop** (vòng lặp sự kiện) và **Callback Queue**, JavaScript có thể xử lý các tác vụ bất đồng bộ (như gọi API hoặc Timer) mà không làm tắc nghẽn (Block) giao diện người dùng.
- **Các cải tiến từ ECMAScript 2015 (ES6):**
    - **Let/Const:** Quản lý phạm vi biến (Scope) theo khối lệnh, tránh các lỗi tiềm ẩn của var.
    - **Arrow Functions:** Rút gọn cú pháp viết hàm và giữ được ngữ cảnh của từ khóa this.
    - **Template Literals:** Cho phép nhúng biến vào chuỗi bằng cú pháp ${variable}, tăng khả năng đọc mã.

2.2. Lập trình hướng đối tượng và Quản trị dữ liệu với Java

2.2.1. 4 trụ cột của Lập trình hướng đối tượng (OOP)

Java là ngôn ngữ thuần hướng đối tượng, mọi kiến thức trong đồ án đều xoay quanh 4 nguyên lý:

- **Tính đóng gói (Encapsulation):** Bảo vệ dữ liệu bằng cách che giấu các thuộc tính bên trong lớp qua từ khóa private và chỉ cho phép truy cập qua getter/setter.
- **Tính kế thừa (Inheritance):** Cho phép các lớp con (như Smartphone) sử dụng lại các thuộc tính và phương thức của lớp cha (Phone), tối ưu hóa việc tái sử dụng mã.
- **Tính đa hình (Polymorphism):** Cho phép một đối tượng có nhiều hình thái khác nhau thông qua việc ghi đè (Overriding) hoặc nạp chồng (Overloading) phương thức.
- **Tính trừu tượng (Abstraction):** Loại bỏ các chi tiết triển khai rườm rà, chỉ tập trung vào các hành vi cốt lõi thông qua Interface hoặc Abstract Class.

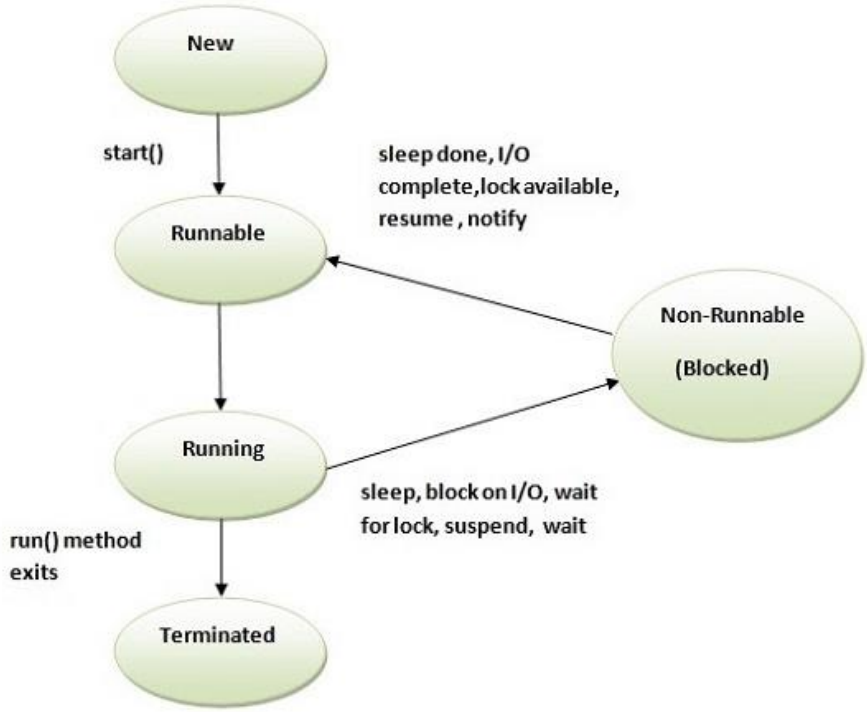


Hình 2.4. Bốn đặc tính cơ bản của lập trình hướng đối tượng.

2.2.2. Kiến trúc đa luồng (Multithreading)

Trong Java, đa luồng cho phép thực thi đồng thời nhiều đoạn mã, giúp tối ưu hóa tài nguyên CPU.

- **Chu kỳ sống của luồng (Thread Lifecycle):**
  - **New:** Luồng được tạo nhưng chưa chạy.
  - **Runnable:** Sẵn sàng thực thi.
  - **Running:** Đang được CPU xử lý.
  - **Waiting/Blocked:** Tạm dừng để chờ tài nguyên hoặc sự kiện.
  - **Terminated:** Luồng kết thúc công việc.



Hình 2.5. Sơ đồ các trạng thái trong vòng đời của luồng (Thread Lifecycle).

- **Đồng bộ hóa (Synchronization):** Đảm bảo tại một thời điểm chỉ có một luồng được truy cập vào tài nguyên dùng chung, ngăn chặn lỗi **Race Condition**.

2.2.3. Công nghệ JDBC (Java Database Connectivity)

JDBC là một bộ API cung cấp khả năng kết nối và truy vấn các hệ quản trị cơ sở dữ liệu (DBMS).

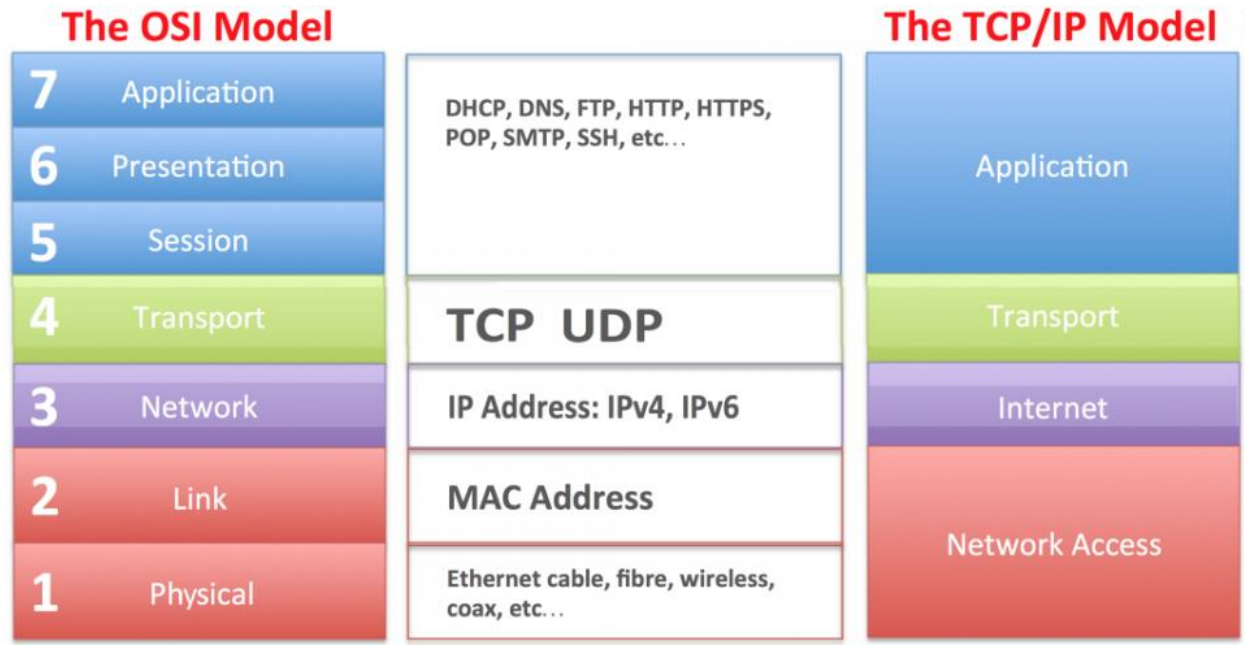
- **Quy trình làm việc:** Nạp Driver -> Thiết lập Connection -> Tạo Statement -> Thực thi SQL -> Xử lý ResultSet -> Đóng kết nối.
- **Transaction Management:** Đảm bảo tính toàn vẹn dữ liệu (ACID). Một giao dịch gồm nhiều lệnh SQL phải thành công tất cả (commit) hoặc không có lệnh nào được thực thi (rollback) nếu có lỗi xảy ra.

2.3. Hệ thống mạng và Giao thức truyền thông

2.3.1. Mô hình tham chiếu OSI và TCP/IP

Để website vận hành, dữ liệu phải đi qua các tầng mạng:

- **OSI 7 Tầng:** Phân chia lý thuyết từ tầng Vật lý đến tầng Ứng dụng.
- **TCP/IP 4 Tầng:** Mô hình thực tế của Internet, tập trung vào tầng Truy cập mạng, tầng Internet (IP), tầng Giao vận (TCP/UDP) và tầng Ứng dụng (HTTP, FTP).



Hình 2.6. So sánh mô hình tham chiếu OSI và mô hình TCP/IP.

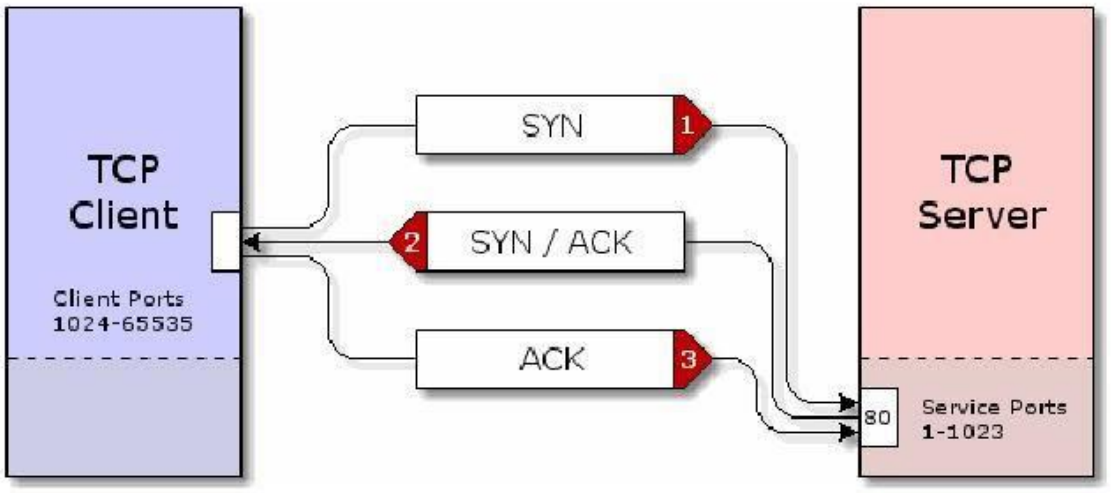
2.3.2. Lập trình Socket

Socket là giao diện lập trình cho phép hai tiến trình trên mạng giao tiếp với nhau.

- **Cơ chế:** Server mở một cổng (Port) và lắng nghe. Client khởi tạo một kết nối đến địa chỉ IP và Port đó. Khi kết nối được thiết lập, cả hai có thể truyền dữ liệu qua các luồng I/O (Input/Output Streams).

2.3.3. So sánh giao thức TCP và UDP

- **TCP (Transmission Control Protocol):** Giao thức hướng kết nối. Nó yêu cầu **bắt tay 3 bước (3-way handshake)**: SYN -> SYN-ACK -> ACK để thiết lập phiên làm việc. Đảm bảo dữ liệu đến nơi đầy đủ và đúng thứ tự.
- **UDP (User Datagram Protocol):** Giao thức không hướng kết nối. Dữ liệu được gửi đi nhanh chóng mà không cần kiểm tra phản hồi. Thích hợp cho các ứng dụng Real-time như Game hoặc Livestream.



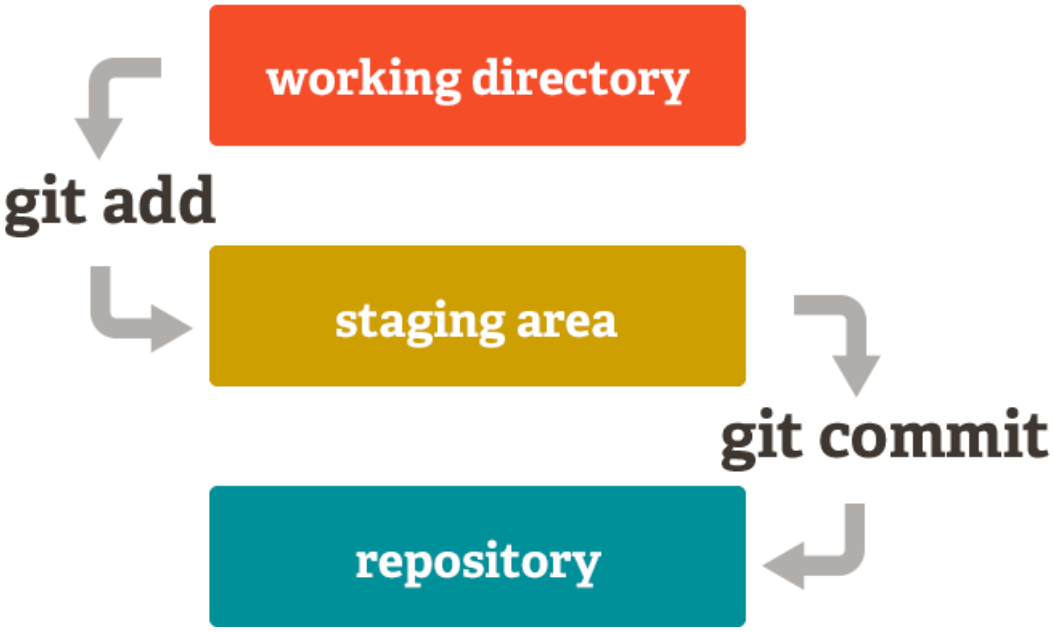
Hình 2.7. Quy trình thiết lập kết nối thông qua bắt tay 3 bước (TCP 3-way Handshake).

2.4. Quy trình phát triển và Công cụ quản trị

2.4.1. Hệ thống quản trị phiên bản Git

Git là hệ thống quản lý phiên bản phân tán (DVCS). Nó giúp lập trình viên lưu lại các "ảnh chụp" (Snapshots) của dự án.

- **Working Directory:** Thư mục làm việc trên máy tính.
- **Staging Area:** Nơi đánh dấu các thay đổi chuẩn bị cho việc lưu trữ.
- **Local Repository:** Kho lưu trữ chính thức trên máy cá nhân.
- **Remote Repository:** Kho lưu trữ trên đám mây (GitHub).



Hình 2.8. Luồng làm việc và các phân vùng quản lý của Git.

2.4.2. Nền tảng GitHub

Git là một hệ thống quản lý phiên bản phân tán, cho phép theo dõi lịch sử thay đổi của mã nguồn một cách hiệu quả. GitHub là nền tảng lưu trữ mã nguồn trực tuyến dựa trên Git, cung cấp giao diện thân thiện và nhiều tính năng hỗ trợ phát triển phần mềm.

Trong đề tài này, GitHub được sử dụng để:

- Lưu trữ toàn bộ mã nguồn website Portfolio.
- Quản lý các phiên bản phát triển.
- Chia sẻ sản phẩm với giảng viên và người đánh giá.

2.4.3. Vai trò của GitHub trong Website Portfolio

Việc sử dụng GitHub không chỉ mang ý nghĩa kỹ thuật mà còn mang tính học thuật và nghề nghiệp. GitHub giúp sinh viên:

- Rèn luyện kỹ năng làm việc với hệ thống quản lý mã nguồn.

- Thể hiện tính chuyên nghiệp trong quá trình phát triển phần mềm.
- Dễ dàng mở rộng và bảo trì website trong tương lai.

Ngoài ra, GitHub còn đóng vai trò như một minh chứng cho quá trình học tập và làm việc nghiêm túc của sinh viên trong lĩnh vực Công nghệ phần mềm.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG WEBSITE PORTFOLIO CÁ NHÂN

3.1. Phân tích đặc tả yêu cầu hệ thống

3.1.1. Mục tiêu xây dựng hệ thống

Trong bối cảnh thị trường lao động công nghệ thông tin ngày càng cạnh tranh, Website Portfolio cá nhân không chỉ đơn thuần là nơi lưu trữ thông tin mà còn đóng vai trò là một **"Định danh số" (Digital Identity)**. Khác với các hệ thống thương mại điện tử hay quản lý doanh nghiệp tập trung vào nghiệp vụ xử lý dữ liệu, Portfolio tập trung tối đa vào **Trải nghiệm thị giác (Visual Experience)** và **Khả năng trình diễn năng lực (Competency Showcase)**.

Mục tiêu cốt lõi của hệ thống bao gồm:

- **Xây dựng thương hiệu cá nhân (Personal Branding):** Tạo lập một hình ảnh lập trình viên chuyên nghiệp, chỉn chu thông qua giao diện hiện đại (Modern UI) và bố cục khoa học.
- **Minh chứng năng lực thực tế:** Thay vì liệt kê kỹ năng trên giấy, website cho phép trình diễn trực quan các dự án (Projects), mã nguồn (Source code) và tư duy giải quyết vấn đề.
- **Chia sẻ và hệ thống hóa tri thức:** Tích hợp module Blog để chia sẻ các kiến thức chuyên sâu về Java, JavaScript và Network, qua đó khẳng định sự hiểu biết sâu sắc của tác giả.
- **Tối ưu hóa khả năng tiếp cận:** Tạo ra một kênh liên lạc trực tuyến (Online Presence) hoạt động 24/7, giúp nhà tuyển dụng dễ dàng tiếp cận hồ sơ ứng viên bất cứ lúc nào.

3.1.2. Phân tích đối tượng sử dụng (Stakeholders)

Hệ thống được thiết kế để phục vụ các nhóm đối tượng với nhu cầu cụ thể sau:

- **Nhà tuyển dụng (Recruiters/HR):** Nhóm đối tượng quan trọng nhất. Họ cần tìm kiếm thông tin nhanh chóng, đánh giá năng lực qua các dự án thực tế và xem xét phong cách code (Coding style).
- **Giảng viên & Hội đồng đánh giá:** Sử dụng website làm thước đo để chấm điểm đồ án, đánh giá khả năng áp dụng lý thuyết vào thực tiễn và tư duy thiết kế hệ thống tổng thể.
- **Cộng đồng lập trình viên (Peers):** Truy cập để tham khảo kiến thức từ các bài Blog, xem các đoạn mã mẫu (Code snippets) và trao đổi kinh nghiệm.
- **Quản trị viên (Admin - Chính sinh viên):** Sử dụng hệ thống như một kho lưu trữ (Repository) để cập nhật liên tục quá trình phát triển bản thân (Lifelong Learning).

3.1.3. Yêu cầu chức năng (Functional Requirements)

Dựa trên phân tích nhu cầu, hệ thống được chia thành các nhóm chức năng trọng yếu:

- **Nhóm chức năng Giới thiệu (Identity Core):**
  - Hiển thị **Hero Section** ấn tượng với thông điệp định hướng nghề nghiệp (Slogan) và nút kêu gọi hành động (CTA).
  - Hiển thị Hồ sơ năng lực (Profile) và Biểu đồ kỹ năng (Skill Bars) trực quan.
- **Nhóm chức năng Trình diễn (Showcase Core):**
  - Hiển thị danh sách dự án dạng lưới (Grid Layout) hoặc thẻ (Card).
  - Xem chi tiết dự án: Mô tả, Công nghệ sử dụng (Tech Stack), Link Demo, Link GitHub.
- **Nhóm chức năng Nội dung (Content Core):**
  - Hệ thống Blog phân loại theo chủ đề (Java, Network, Life).
  - Trình đọc bài viết chi tiết hỗ trợ định dạng mã nguồn (Syntax Highlighting).
- **Nhóm chức năng Tương tác:**
  - Form liên hệ trực tuyến.
  - Thư viện chứng chỉ (Certificates) với tính năng xem ảnh chi tiết (Lightbox).

3.1.4. Yêu cầu phi chức năng (Non-functional Requirements)

Để đảm bảo chất lượng phần mềm, hệ thống tuân thủ các tiêu chuẩn kỹ thuật:

- **Hiệu năng (Performance):** Tối ưu hóa điểm số Google Lighthouse, thời gian tải trang đầu tiên (First Contentful Paint) dưới 1.5 giây.
- **Tính đáp ứng (Responsiveness):** Giao diện tự động thích nghi hoàn hảo trên mọi thiết bị (Mobile, Tablet, Desktop, Laptop) theo nguyên lý Mobile-First.
- **Tính khả dụng (Usability):** Tuân thủ các nguyên tắc thiết kế UX, đảm bảo độ tương phản màu sắc, kích thước phông chữ dễ đọc.
- **Tính bảo trì (Maintainability):** Mã nguồn HTML/CSS/JS được tổ chức theo module, dễ dàng mở rộng tính năng mới.
- **Chuẩn SEO:** Cấu trúc HTML Semantic thân thiện với các bộ máy tìm kiếm.

3.2. Kiến trúc thông tin và Sơ đồ Sitemap

3.2.1. Kiến trúc thông tin (Information Architecture)

Website được tổ chức theo mô hình **Cấu trúc phẳng (Flat Hierarchy)**. Điều này giúp giảm thiểu độ sâu của các liên kết, cho phép người dùng truy cập bất kỳ nội dung quan trọng nào chỉ trong tối đa 2 lần nhấp chuột (2-click rule).

3.2.2. Sơ đồ Sitemap chi tiết

Sơ đồ Sitemap dưới đây minh họa luồng điều hướng của hệ thống:

- **Trang chủ (Home):** Trung tâm điều hướng.
  - Giới thiệu ngắn (About Teaser)
  - Dự án nổi bật (Featured Projects)
  - Bài viết mới nhất (Latest Posts)
- **Kỹ năng (Skills):**
  - Technical Skills (Java, JS, SQL...)
  - Soft Skills & Tools (Git, Teamwork...)
- **Dự án (Projects):**
  - Dự án Web
  - Dự án Desktop App
- **Blog Công nghệ:**
  - Chuyên mục Java/JDBC
  - Chuyên mục Network/Socket
  - Chuyên mục Frontend
- **Liên hệ (Contact):** Thông tin kết nối và Form gửi tin nhắn.

3.3. Thiết kế Hệ thống thiết kế (Design System) và Giao diện

3.3.1. Hệ thống thiết kế (Design System)

Trước khi đi vào thiết kế chi tiết từng trang, một hệ thống thiết kế nhất quán được xây dựng để đảm bảo tính đồng bộ (Consistency):

- **Bảng màu (Color Palette):**
  - Màu chủ đạo (Primary): Sử dụng tông màu đỏ cam (#f56a6a) thể hiện sự nhiệt huyết, năng động và sáng tạo.
  - Màu trung tính (Neutral): Trắng (#fff) và Xám đậm (#3d4449) tạo sự tương phản cao, mang lại cảm giác chuyên nghiệp, dễ đọc (Readability).
- **Typography (Kiểu chữ):** Sử dụng dòng font Sans-serif (như Roboto hoặc Open Sans) hiện đại, tối giản, phù hợp cho việc hiển thị trên màn hình kỹ thuật số.
- **Hệ thống lưới (Grid System):** Sử dụng lưới 12 cột (12-column grid) để căn chỉnh bố cục, đảm bảo sự cân đối về thị giác.

3.3.2. Thiết kế chi tiết Trang chủ (Dashboard)

Được ví như "mặt tiền" của ngôi nhà số, trang chủ được thiết kế theo cấu trúc **Sidebar Layout**:

- **Sidebar (Thanh bên):** Cố định bên trái (hoặc ẩn trong menu Burger trên mobile), chứa Avatar, Menu điều hướng và Social Links. Giúp người dùng luôn định vị được vị trí của mình.

- **Main Content (Nội dung chính):** Hiển thị Banner lớn (Hero Image) và tóm tắt các nội dung quan trọng nhất.

3.3.3. Thiết kế trang Hồ sơ & Kỹ năng

Sử dụng phương pháp **Trực quan hóa dữ liệu (Data Visualization)**:

- Các kỹ năng không liệt kê dạng văn bản thuần túy mà được thể hiện qua các thanh tiến trình (Progress Bars) hoặc biểu đồ tròn.
- Giúp nhà tuyển dụng đánh giá nhanh mức độ thành thạo công nghệ của ứng viên (ví dụ: Java: 80%, HTML/CSS: 90%).

3.3.4. Thiết kế trang Dự án (Portfolio Showcase)

Áp dụng mẫu thiết kế **Card UI (Giao diện dạng thẻ)**:

- Mỗi dự án là một thẻ bao gồm: Hình ảnh thumbnail chất lượng cao, Tên dự án, Công nghệ sử dụng (Tags) và Nút xem chi tiết.
- Bố cục dạng lưới (Grid) giúp hiển thị được nhiều dự án cùng lúc nhưng vẫn giữ được sự thoáng đãng (White space).

3.3.5. Thiết kế trang Blog chuyên sâu

Được tối ưu hóa cho **Trải nghiệm đọc (Reading Experience)**:

- Sử dụng độ rộng dòng chữ (Line length) tiêu chuẩn (50-75 ký tự/dòng) để mắt không bị mỏi.
- Tích hợp các khối hiển thị mã nguồn (Code Blocks) với màu sắc cú pháp (Syntax highlighting) chuyên biệt cho Java/JS, phục vụ đối tượng độc giả là lập trình viên.

3.3.6. Thiết kế trang Liên hệ

Tối giản hóa các trường thông tin, tập trung vào tính kết nối nhanh. Tích hợp bản đồ hoặc các icon mạng xã hội lớn để khuyến khích tương tác.

3.4. Áp dụng các nguyên tắc UX/UI trong thiết kế

3.4.1. Nguyên tắc Trải nghiệm người dùng (UX)

- **Luật Jakob:** Xây dựng giao diện quen thuộc, hoạt động theo cách người dùng mong đợi (ví dụ: Logo luôn trỏ về trang chủ, Menu luôn ở vị trí dễ thấy).
- **Luật Fitts:** Các nút bấm quan trọng (CTA) như "Xem dự án", "Liên hệ" được thiết kế kích thước lớn và đặt ở vị trí thuận tiện cho thao tác chạm/click.

- **Tốc độ phản hồi:** Sử dụng kỹ thuật tải không đồng bộ hoặc Lazy Loading cho hình ảnh để giảm thời gian chờ đợi.

3.4.2. Nguyên tắc Giao diện người dùng (UI)

- **Phân cấp thị giác (Visual Hierarchy):** Sử dụng kích thước chữ (H1, H2, H3) và màu sắc đậm nhạt để dẫn dắt mắt người xem tập trung vào các nội dung quan trọng nhất.
- **Không gian âm (Negative Space):** Sử dụng hợp lý các khoảng trắng (White space) giúp giao diện "dễ thở", sang trọng và không gây rối mắt.
- **Tính nhất quán (Consistency):** Đảm bảo sự đồng bộ về kiểu nút bấm, hiệu ứng hover, font chữ trên toàn bộ 9-10 trang của website.

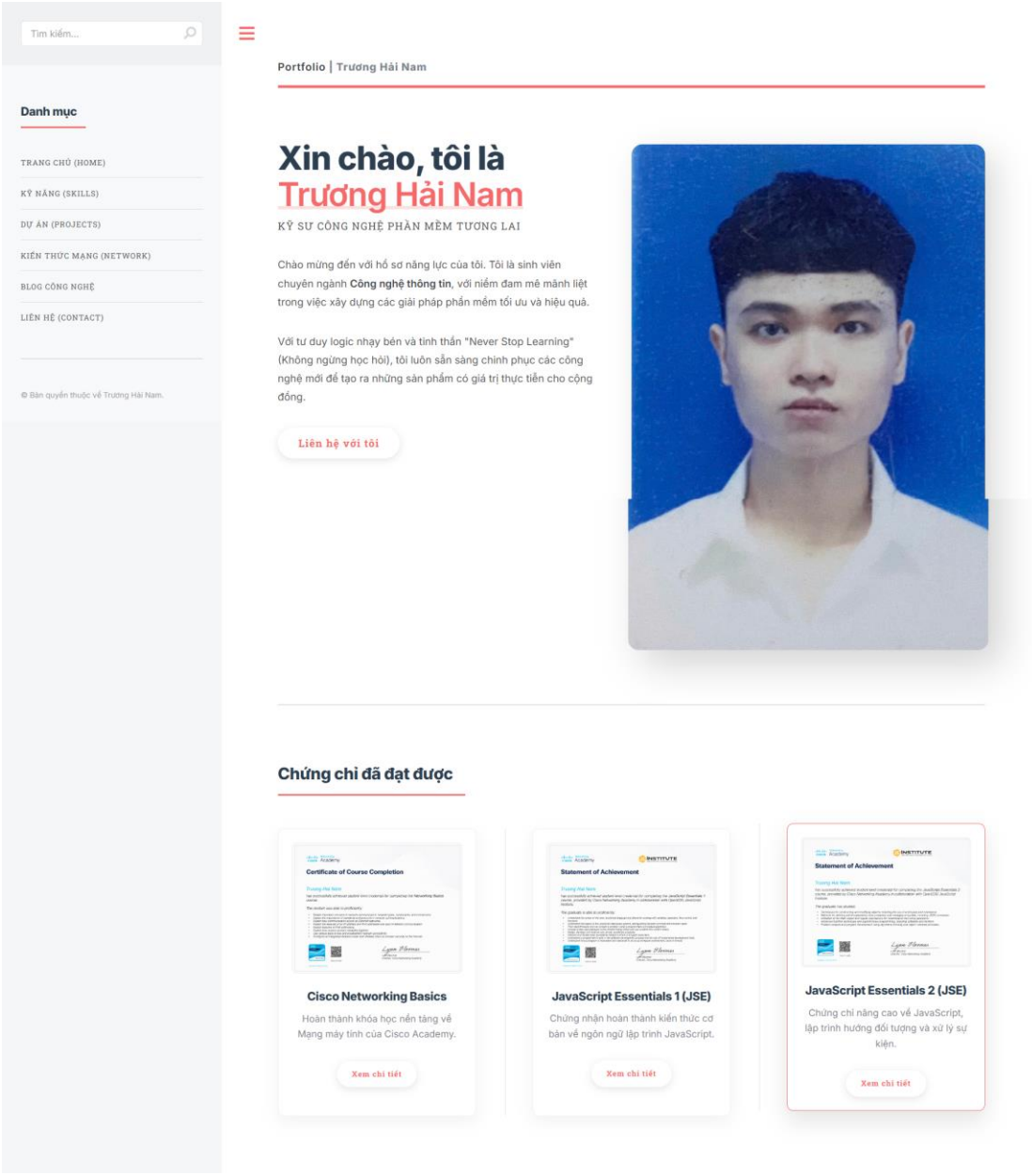
CHƯƠNG 4: XÂY DỰNG HỆ THỐNG VÀ DEMO GIAO DIỆN WEBSITE PORTFOLIO

4.1. Giới thiệu chung về quá trình xây dựng giao diện

Sau khi hoàn thành quá trình phân tích yêu cầu và thiết kế hệ thống ở Chương 3, bước tiếp theo là giai đoạn hiện thực hóa (Implementation) để xây dựng giao diện Website Portfolio cá nhân. Đây là giai đoạn chuyển đổi các bản vẽ Wireframe và tư duy thiết kế thành mã nguồn HTML, CSS và JavaScript chạy thực tế.

Giao diện đóng vai trò then chốt trong việc định hình thương hiệu cá nhân (Personal Branding). Một giao diện tốt không chỉ cần đẹp về mặt thẩm mỹ mà còn phải đảm bảo trải nghiệm người dùng (UX), tốc độ tải trang nhanh và khả năng tương thích đa thiết bị. Trong chương này, nội dung sẽ tập trung trình bày các hình ảnh thực nghiệm của hệ thống sau khi hoàn thiện, kèm theo các đánh giá chi tiết về bố cục và chức năng.

4.2. Giao diện Trang chủ (Home Page)



Hình 4.1. Giao diện Trang chủ với bố cục Sidebar và Main Content

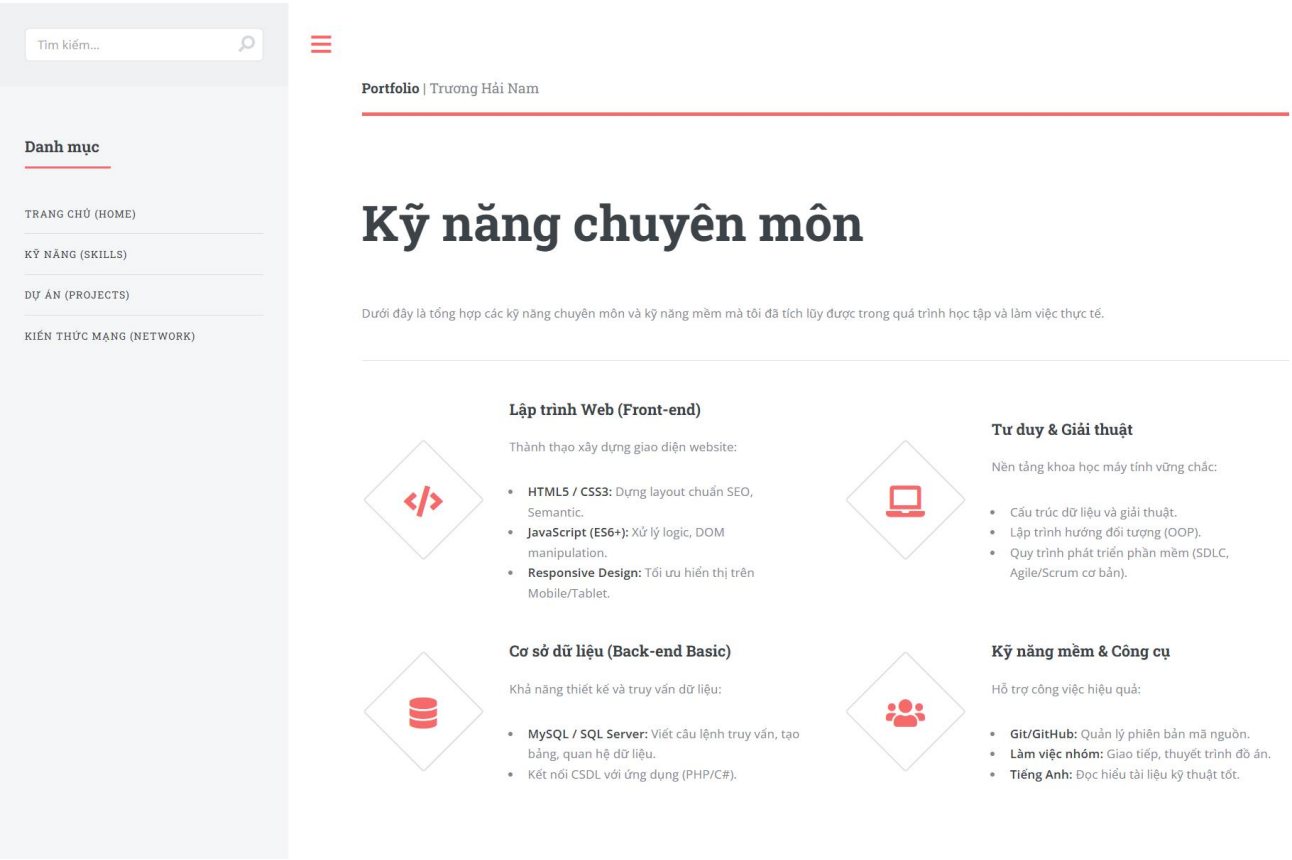
Mô tả chi tiết: Trang chủ được thiết kế theo bố cục **Hai cột bất đối xứng (Asymmetrical Two-Column Layout)**:

- **Thanh Sidebar (Cột bên):** Được cố định dọc theo chiều cao màn hình. Đây là trung tâm điều hướng chứa:
    - **Thanh tìm kiếm (Search Bar):** Cho phép người dùng lọc bài viết nhanh chóng.

- **Menu điều hướng chính:** Các liên kết đến trang Kỹ năng, Dự án, Liên hệ.
- **Danh sách bài viết mới (Blog):** Gợi ý các bài viết vừa cập nhật để giữ chân người đọc.

- **Khu vực Nội dung chính (Main Content):** Hiển thị Banner giới thiệu (Intro Header) với tên tác giả, các liên kết mạng xã hội (Facebook, GitHub, Email) sử dụng bộ icon *FontAwesome*, tạo điểm nhấn chuyên nghiệp ngay từ cái nhìn đầu tiên.

4.3. Demo giao diện Trang Kỹ năng (Skills Page)



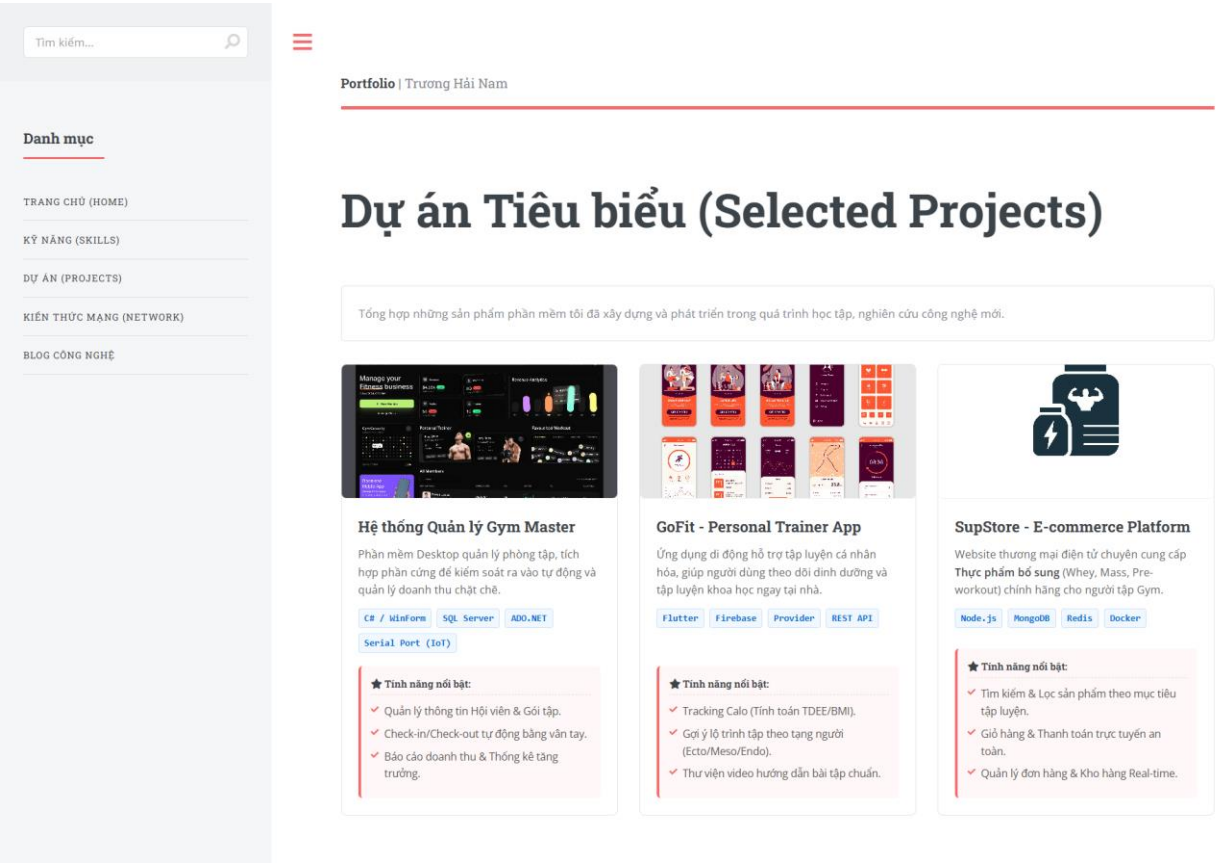
Hình 4.2. Giao diện Trang Kỹ năng với bố cục lưới (Grid Layout) phân chia theo nhóm.

Mô tả chi tiết: Trang Kỹ năng được thiết kế theo phong cách tối giản (Minimalism), tập trung tối đa vào nội dung văn bản và khả năng đọc (Readability):

- **Tiêu đề chính (Main Heading):** Dòng chữ "Kỹ năng chuyên môn" được trình bày nổi bật với phông chữ có chân (Serif) cỡ lớn, tạo cảm giác trang trọng và chuyên nghiệp.
- **Bố cục nội dung (Content Layout):** Sử dụng hệ thống Lưới (Grid System) để chia các nhóm kỹ năng thành 2 cột cân đối, giúp tận dụng không gian màn hình hiệu quả.

- **Phân nhóm kỹ năng:** Nội dung được chia thành 4 khối rõ ràng:
    - **Lập trình Web (Front-end):** Các công nghệ xây dựng giao diện (HTML5, CSS3, JS).
    - **Cơ sở dữ liệu (Back-end Basic):** Khả năng thiết kế và truy vấn dữ liệu (MySQL, SQL Server).
    - **Tư duy & Giải thuật:** Nền tảng khoa học máy tính (OOP, Cấu trúc dữ liệu).
    - **Kỹ năng mềm & Công cụ:** Các kỹ năng bổ trợ (Git, Tiếng Anh, Làm việc nhóm).
- **Yếu tố đồ họa (Visual Elements):** Mỗi nhóm kỹ năng đi kèm một biểu tượng (Icon) nằm trong khung hình thoi (Diamond shape), tạo điểm nhấn thị giác và giúp người xem nhanh chóng nhận diện loại kỹ năng.

4.4. Demo giao diện Trang Dự án (Projects Page)

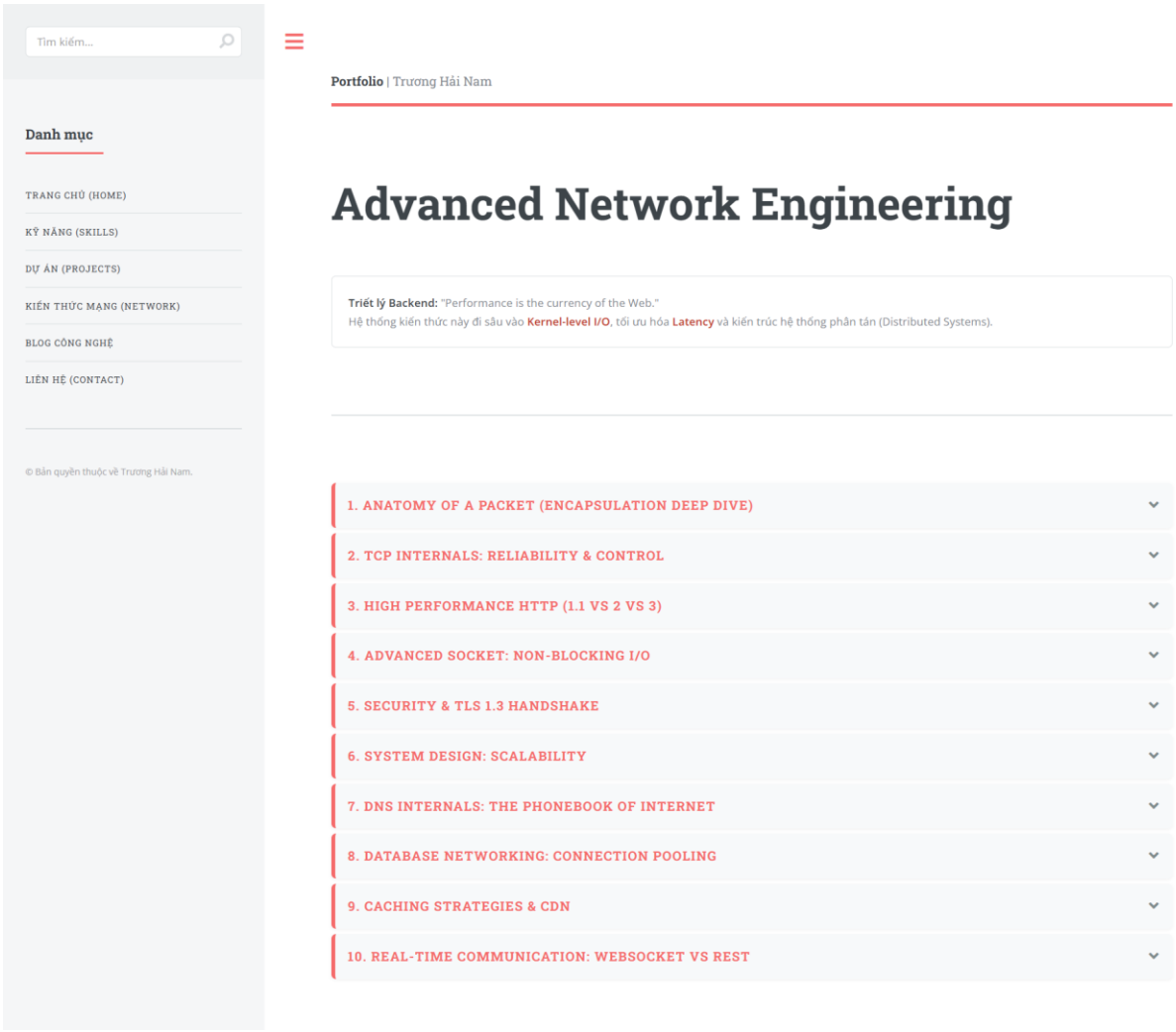


Hình 4.3. Giao diện Trang Dự án Tiêu biểu với thiết kế dạng thẻ (Card Layout).

Mô tả chi tiết: Trang Dự án là nơi trưng bày các sản phẩm phần mềm thực tế, được thiết kế để làm nổi bật năng lực kỹ thuật và phạm vi công nghệ (Tech Stack) đa dạng của tác giả:

- **Bố cục (Layout):** Sử dụng thiết kế dạng Lưới (Grid System) chia thành 3 cột đều nhau, giúp trình bày các dự án một cách khoa học và dễ so sánh.
- **Thành phần Thẻ Dự án (Project Card):** Mỗi dự án được gói gọn trong một thẻ (Card) độc lập với cấu trúc thông tin nhất quán:
    - **Hình ảnh minh họa (Thumbnail):** Ảnh chụp màn hình ứng dụng hoặc biểu tượng đại diện giúp người xem hình dung trực quan về sản phẩm.
    - **Thông tin tổng quan:** Tên dự án và mô tả ngắn gọn mục đích sử dụng.
    - **Công nghệ sử dụng (Tech Stack):** Sử dụng các nhãn (Tags/Badges) màu xanh dương để liệt kê các công nghệ lõi (ví dụ: C#, Flutter, Node.js, Docker...). Điều này giúp người xem nhanh chóng nắm bắt kỹ năng kỹ thuật áp dụng trong từng dự án.
    - **Tính năng nổi bật (Key Features):** Khu vực được làm nổi bật với nền màu hồng nhạt, liệt kê các chức năng chính của phần mềm, chứng minh độ phức tạp và tính hoàn thiện của sản phẩm.
- **Đa dạng nền tảng:** Các dự án mẫu thể hiện khả năng lập trình trên nhiều môi trường khác nhau: Ứng dụng Desktop (Gym Master), Ứng dụng Di động (GoFit), và Web App/Backend (SupStore).

4.5. Demo giao diện Trang Kiến thức mạng (Network Knowledge Page)



Hình 4.4. Giao diện chuyên sâu về Kỹ thuật Mạng với thiết kế Accordion.

Mô tả chi tiết: Đây là trang thể hiện chiều sâu kiến thức chuyên môn của tác giả, tập trung vào các khái niệm nâng cao trong lập trình mạng và hệ thống phân tán. Giao diện được tối ưu hóa cho việc đọc và tra cứu tài liệu kỹ thuật:

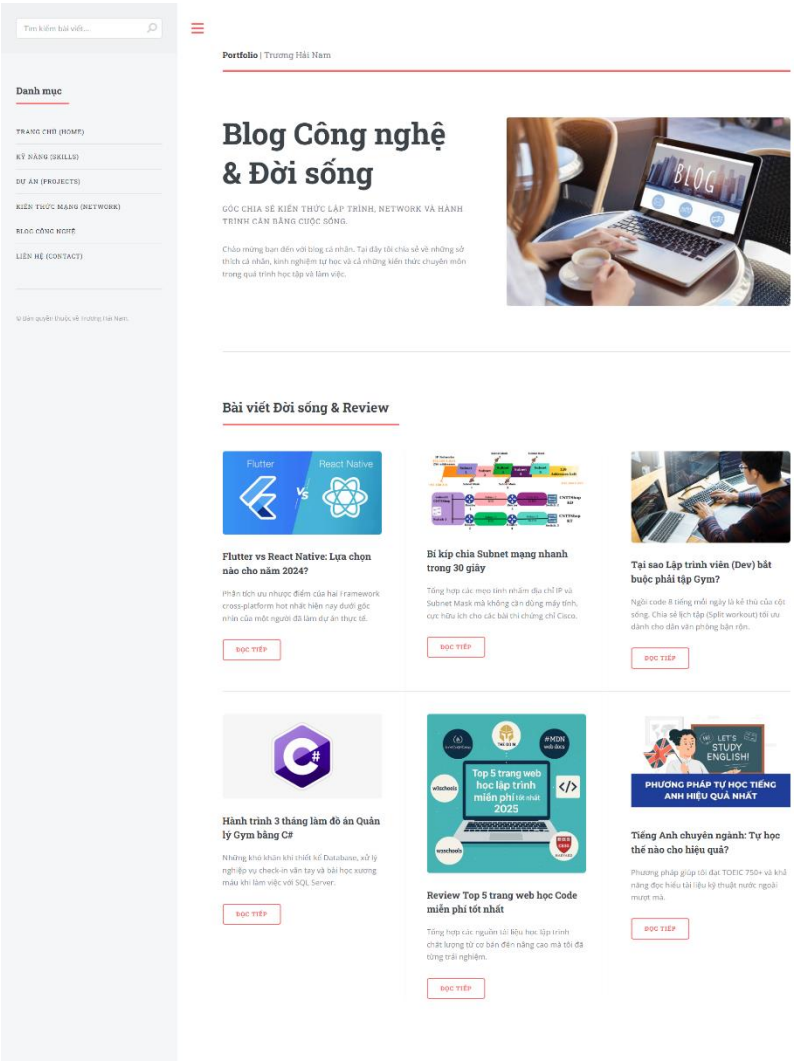
- **Triết lý thiết kế (Design Philosophy):**
  - Phần đầu trang nổi bật với trích dẫn (Quote Block) về triết lý Backend: *"Performance is the currency of the Web"*. Điều này định hướng người xem rằng nội dung bên dưới sẽ tập trung vào hiệu năng, tối ưu hóa Latency và kiến trúc hệ thống.
- **Thành phần UI chính (UI Components):**
  - **Danh sách dạng Accordion (Collapsible List):** Thay vì hiển thị toàn bộ văn bản dài dòng, trang web sử dụng cơ chế "Bấm để mở rộng". Người dùng chỉ cần nhấp vào tiêu đề (ví dụ: *2. TCP Internals*) để xem nội dung chi tiết bên trong. Cách này giúp giao diện luôn gọn gàng, sạch sẽ.

- **Dấu hiệu trực quan (Visual Cues):** Mỗi mục đều có một thanh màu đỏ đậm (Accent Color) ở lề trái và mũi tên chỉ xuống (Chevron icon) ở bên phải, giúp người dùng dễ dàng nhận biết đây là các phần tử có thể tương tác được.

- **Nội dung chuyên sâu:**
  - Danh mục bao quát các chủ đề cốt lõi của môn Mạng máy tính như: Mô hình OSI/TCP-IP (Anatomy of a Packet), Giao thức HTTP (1.1 vs 2 vs 3), Bảo mật (TLS 1.3 Handshake) và Thiết kế hệ thống (Scalability, Load Balancing).

4.6. Demo giao diện Trang Blog Công nghệ & Đời sống

4.6.1. Giao diện Bài viết Đời sống & Review

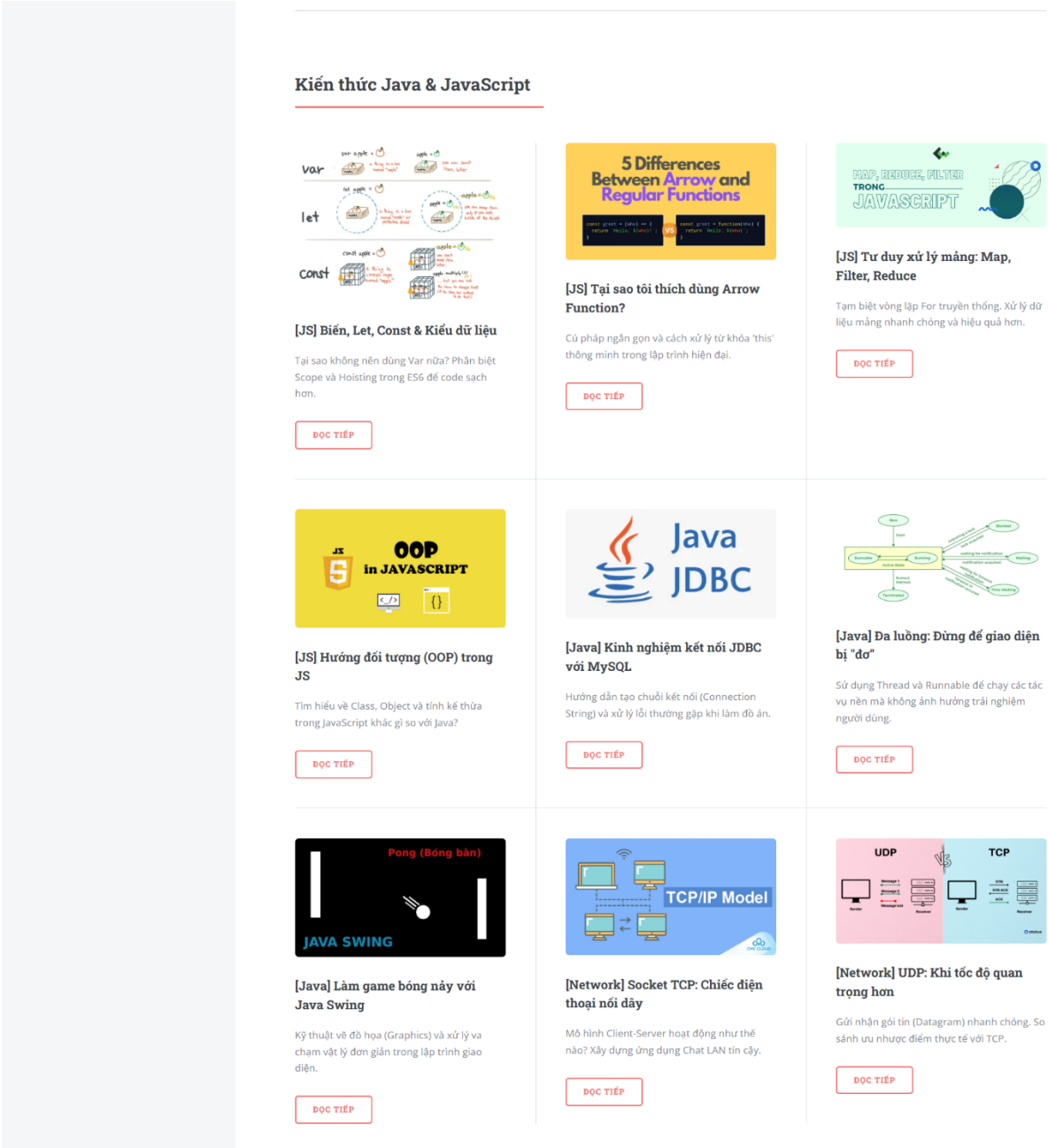


Hình 4.5. Giao diện trang Blog bài viết đời sống và review

Mô tả chi tiết: Trang Blog được xây dựng như một không gian chia sẻ kiến thức cá nhân, kết hợp giữa yếu tố kỹ thuật chuyên sâu và trải nghiệm đời sống thực tế của một lập trình viên:

- **Phần giới thiệu (Intro Section):** Sử dụng hình ảnh minh họa chất lượng cao kết hợp với lời chào dẫn dắt, tạo cảm giác thân thiện và gần gũi với độc giả.
- **Khu vực Bài viết Đời sống & Review:** Đây là phần nội dung trọng tâm của trang, được trình bày dưới dạng **Lưới thẻ (Card Grid Layout)** với 3 cột, giúp hiển thị tối đa thông tin trong một khung hình:
    - **Cấu trúc mỗi thẻ bài viết:** Bao gồm ảnh đại diện (Thumbnail) bắt mắt, tiêu đề bài viết sử dụng font chữ đậm, đoạn mô tả ngắn (Excerpt) để tóm tắt nội dung và nút "Đọc tiếp" được thiết kế tối giản.
    - **Sự đa dạng về chủ đề:** Các bài viết được phân loại rõ ràng, đáp ứng nhiều nhu cầu đọc khác nhau:
        - **Kỹ thuật & Xu hướng:** So sánh công nghệ (Flutter vs React Native), thủ thuật mạng (Chia Subnet nhanh).
        - **Kỹ năng mềm & Học tập:** Phương pháp học tiếng Anh chuyên ngành, Review các nền tảng học Code.
        - **Đời sống & Sức khỏe:** Tầm quan trọng của việc tập Gym đối với lập trình viên, chia sẻ hành trình thực hiện đồ án cá nhân.
- **Tính tương tác (Interaction):** Các thẻ bài viết có hiệu ứng đổ bóng nhẹ, tạo chiều sâu cho giao diện và tăng trải nghiệm người dùng khi tương tác trên trình duyệt.

4.6.2. Khu vực Kiến thức Java & JavaScript



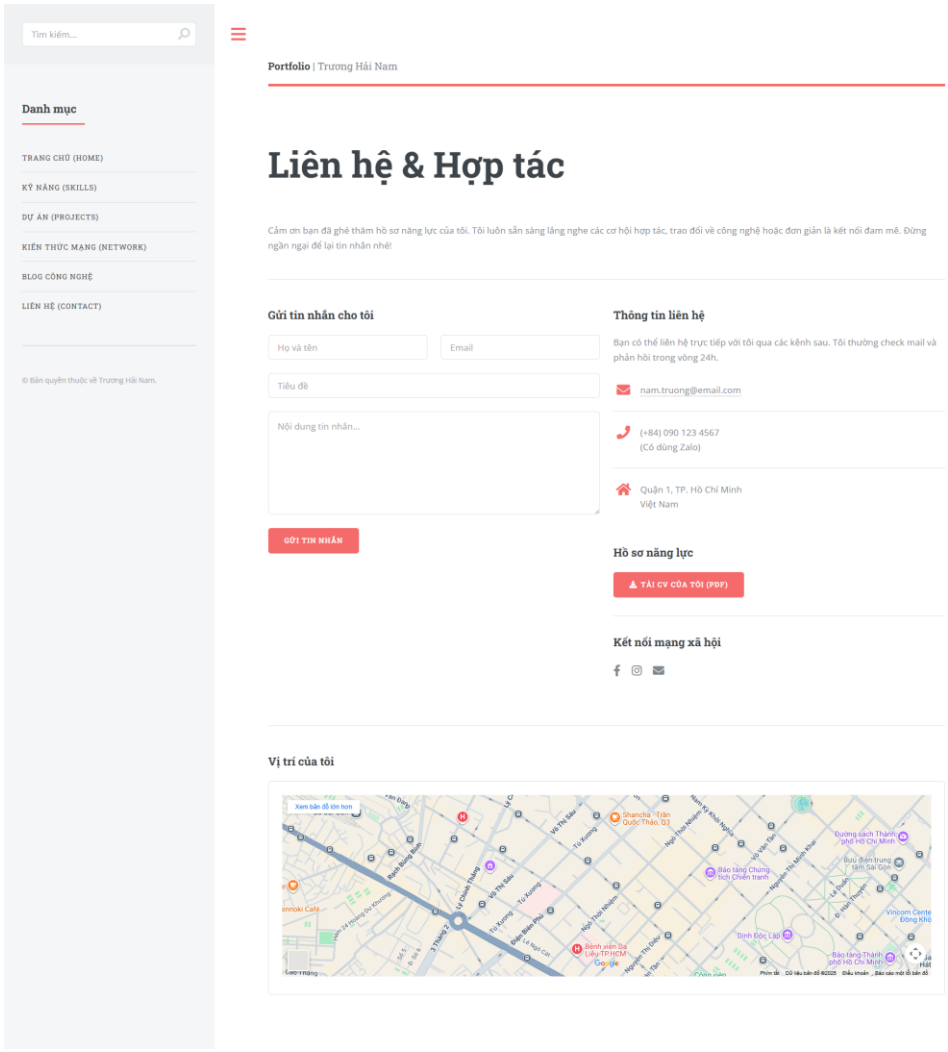
Hình 4.6. Giao diện chuyên mục chia sẻ kiến thức lập trình chuyên sâu.

Mô tả chi tiết: Đây là phân khúc quan trọng nhất của trang Blog, tập trung vào việc hệ thống hóa và chia sẻ các kiến thức kỹ thuật từ cơ bản đến nâng cao, giúp khẳng định năng lực chuyên môn của tác giả:

- **Bố cục và Hình ảnh minh họa:**

    - Sử dụng cấu trúc **Lưới (Grid)** đồng nhất với phần Đời sống, giúp duy trì sự nhất quán về mặt thị giác cho toàn bộ website.
    - Mỗi bài viết đi kèm một hình ảnh minh họa (Thumbnail) được thiết kế riêng biệt, chứa các biểu đồ, đoạn mã hoặc sơ đồ tư duy (Mindmap), giúp người đọc dễ dàng nắm bắt chủ đề ngay từ cái nhìn đầu tiên.
- **Hệ thống nội dung kỹ thuật đa dạng:** Các bài viết được đầu tư kỹ lưỡng về mặt nội dung, bao quát nhiều mảng quan trọng trong phát triển phần mềm:
    - **JavaScript hiện đại (ES6+):** Đi sâu vào các khái niệm cốt lõi như sự khác biệt giữa Var/Let/Const, Arrow Functions, các phương thức xử lý mảng (Map, Filter, Reduce) và lập trình hướng đối tượng (OOP) trong JS.
    - **Lập trình Java chuyên sâu:** Chia sẻ kinh nghiệm thực tế về kết nối cơ sở dữ liệu (JDBC), xử lý đa luồng (Multi-threading) để tránh hiện tượng treo giao diện, và lập trình đồ họa cơ bản với Java Swing thông qua dự án Game bóng bàn (Pong).
    - **Lập trình mạng (Network Programming):** Giải thích trực quan các mô hình Client-Server, sự khác biệt giữa giao thức TCP và UDP thông qua các ví dụ thực tế.
- **Giá trị học thuật:** Cách trình bày nội dung theo dạng "Note-taking" kết hợp với nút "Đọc tiếp" thôi thúc sự tò mò, biến trang blog thành một thư viện nhỏ hỗ trợ việc tự học và ôn tập kiến thức hiệu quả.

4.7. Demo giao diện Trang Liên hệ (Contact Page)



Hình 4.7. Giao diện Trang Liên hệ và Hợp tác với tích hợp bản đồ và biểu mẫu tương tác.

Mô tả chi tiết: Trang Liên hệ được thiết kế với mục tiêu tối ưu hóa khả năng kết nối giữa tác giả và người xem (nhà tuyển dụng, đối tác hoặc bạn bè). Giao diện được phân chia khoa học thành các khối chức năng rõ rệt:

- **Biểu mẫu tương tác (Contact Form):** Khu vực "Gửi tin nhắn cho tôi" cho phép người dùng để lại thông tin liên hệ và nội dung trao đổi trực tiếp trên website. Các trường dữ liệu (Họ tên, Email, Tiêu đề, Nội dung) được bố trí thoáng đãng, đi kèm nút gửi màu đỏ nổi bật tạo điểm nhấn hành động.
- **Thông tin liên lạc đa kênh:** Cung cấp đầy đủ các phương thức kết nối nhanh bao gồm Email, số điện thoại cá nhân (hỗ trợ Zalo) và địa chỉ lưu trú tại Quận 1, TP. Hồ Chí Minh. Việc công khai các thông tin này giúp tăng độ tin cậy và tính chuyên nghiệp cho hồ sơ.

- **Hồ sơ năng lực (CV):** Một nút kêu gọi hành động (CTA) "Tải CV của tôi (PDF)" được đặt ở vị trí trung tâm, giúp người xem dễ dàng lưu trữ hồ sơ của tác giả về máy cá nhân một cách nhanh chóng.
- **Liên kết mạng xã hội:** Các biểu tượng mạng xã hội (Facebook, Instagram, Email) được đặt ở phần cuối của khối thông tin, tạo thêm nhiều lựa chọn kết nối và tìm hiểu thêm về phong cách sống của tác giả.
- **Tích hợp Bản đồ thực tế (Location Map):** Phía dưới cùng là một bản đồ nhúng (Google Maps) hiển thị vị trí địa lý cụ thể. Tính năng này không chỉ hoàn thiện về mặt thẩm mỹ mà còn minh chứng cho khả năng tích hợp các dịch vụ bên thứ ba (Third-party API) vào dự án web.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận về kết quả đạt được

Sau quá trình nghiên cứu, thiết kế và triển khai, đồ án xây dựng website Portfolio cá nhân đã hoàn thành các mục tiêu đề ra với những kết quả cụ thể sau:

- **Về mặt giao diện và trải nghiệm người dùng (UI/UX):** Đồ án đã xây dựng được một hệ thống giao diện nhất quán, sử dụng phong cách thiết kế tối giản (Minimalism) với bố cục Sidebar cố định giúp tối ưu hóa việc điều phối giữa các trang. Hệ thống màu sắc (Đỏ, Trắng, Xám) và phông chữ được lựa chọn kỹ lưỡng để tạo sự chuyên nghiệp cho một hồ sơ kỹ thuật.
- **Về mặt nội dung chuyên môn: * Hệ thống kỹ năng:** Đồ án đã phân loại và trình bày rõ ràng các kỹ năng từ Front-end (HTML/CSS/JS), Cơ sở dữ liệu (MySQL/SQL Server) đến tư duy lập trình (OOP/Agile).
    - **Kho lưu trữ kiến thức mạng:** Xây dựng được chuyên mục "Advanced Network Engineering" với 10 chủ đề chuyên sâu, bao quát các kiến thức trọng tâm của môn Mạng máy tính như mô hình OSI, giao thức TCP/IP, HTTP/3, DNS và bảo mật TLS 1.3.
    - **Quản lý dự án:** Trình bày được các dự án tiêu biểu đa nền tảng, bao gồm ứng dụng Desktop (C#/WinForm), ứng dụng di động (Flutter) và nền tảng thương mại điện tử (Node.js/Docker).
    - **Hệ thống Blog:** Xây dựng được kho bài viết đa dạng, từ hướng dẫn kỹ thuật Java/JavaScript chuyên sâu đến các bài chia sẻ kinh nghiệm học tập và đời sống.
- **Về mặt tương tác:** Tích hợp thành công các tính năng tương tác thực tế như biểu mẫu gửi tin nhắn (Contact Form), bản đồ nhúng trực tuyến (Google Maps) và liên kết hồ sơ năng lực (CV).

5.2. Đánh giá ưu điểm và hạn chế

5.2.1. Ưu điểm

- **Tính thực tiễn cao:** Đồ án không chỉ dừng lại ở mức độ lý thuyết mà đã tạo ra một sản phẩm hoàn thiện, có thể ứng dụng ngay vào việc quảng bá hồ sơ cá nhân.
- **Kiến thức chuyên sâu:** Nội dung blog và phần kiến thức mạng thể hiện sự đầu tư nghiên cứu kỹ lưỡng về các giao thức mạng và cơ chế hoạt động của hệ thống.
- **Cấu trúc mã nguồn:** Website được tổ chức theo cấu trúc linh hoạt, dễ dàng mở rộng và bảo trì các mục nội dung mới.

5.2.2. Hạn chế

- Hệ thống hiện tại chủ yếu tập trung vào giao diện người dùng (Front-end), các bài viết vẫn được quản lý thông qua mã nguồn tĩnh, chưa có hệ quản trị nội dung (CMS) để đăng bài tự động.
- Chưa triển khai các tính năng tương tác thời gian thực như hệ thống bình luận hoặc thông báo đẩy (Push Notification).

5.3. Hướng phát triển tương lai

Để hoàn thiện và nâng cao giá trị của dự án, các hướng phát triển tiếp theo của đồ án bao gồm:

- **Xây dựng hệ quản trị nội dung (Full-stack CMS):** Chuyển đổi từ website tĩnh sang website động bằng cách sử dụng Node.js hoặc Java để xây dựng hệ thống Admin, cho phép quản lý dự án và bài viết thông qua giao diện quản trị.
- **Tối ưu hóa hiệu năng và bảo mật:**
    - Triển khai giao thức HTTPS toàn diện và cấu hình các chính sách bảo mật cho máy chủ.
    - Sử dụng các kỹ thuật Caching và CDN để tối ưu hóa tốc độ tải trang toàn cầu.
- **Ứng dụng công nghệ mới:** * Tích hợp Socket.io để xây dựng tính năng trò chuyện trực tuyến (Chatbot) hỗ trợ khách ghé thăm trang web.
    - Sử dụng Docker để đóng gói toàn bộ ứng dụng, giúp việc triển khai (Deployment) lên các nền tảng Cloud như AWS, Google Cloud trở nên linh hoạt và nhanh chóng hơn.
- **Cải thiện SEO (Search Engine Optimization):** Tối ưu hóa các thẻ meta, cấu trúc dữ liệu và hình ảnh để tăng khả năng hiển thị của website trên các công cụ tìm kiếm.