

ĐỒ ÁN MÔN HỌC

HỌC PHẦN: LẬP TRÌNH MẠNG MÁY TÍNH

Chủ đề: XÂY DỰNG WEBSITE BLOG CÁ NHÂN
“IMANU’S LAB

Giảng viên hướng dẫn: TS.Nguyễn Quang Chung

Sinh viên thực hiện:

Họ và tên	MSSV	Lớp
Nguyễn Phạm Hoài Hiếu	2180604319	21DTHB4

Nhóm:Overflow
Lớp:23DTHB7

Thành Phố Hồ Chí Minh, 2025

LỜI CẢM ƠN

Lời nói đầu tiên em xin chân thành cảm ơn Trường đại học Công nghệ TP.HCM nói chung và Khoa Công nghệ Thông tin nói riêng đã tạo điều kiện cho chúng em học tập và thực hiện thành công bản báo cáo đồ án môn học này.

Chúng em xin chân thành cảm ơn thầy đã dành thời gian và tận tâm giúp đỡ nhóm trong quá trình làm báo cáo đồ án. Thầy đã trang bị cho nhóm em những kiến thức cần thiết và hướng dẫn cách tiếp cận vấn đề một cách khoa học và có hệ thống. Thời gian làm báo cáo đồ án không phải lúc nào cũng dễ dàng, và nhóm đã gặp phải nhiều khó khăn và thách thức. Nhưng nhờ sự hướng dẫn tận tâm của thầy, nhóm đã có thể vượt qua những khó khăn đó và hoàn thành đồ án một cách thành công. Nhóm em xin bày tỏ lòng biết ơn đối với những góp ý và phản hồi chân thành từ thầy. Những ý kiến đó không chỉ giúp nhóm cải thiện báo cáo đồ án mà còn giúp nhóm hiểu rõ hơn về chủ đề và phát triển khả năng nghiên cứu của mình.

Nhóm em không thể quên nhắc đến sự đóng góp của thầy trong việc mở rộng tầm nhìn của em. Thầy đã giới thiệu những nguồn tài liệu quan trọng và đưa ra những ý tưởng mới mẻ, giúp nhóm em có cái nhìn toàn diện hơn về đề tài. Cuối cùng, em thay mặt nhóm xin gửi lời cảm ơn sâu sắc và biết ơn thầy đã tạo điều kiện cho chúng em để hoàn thành đồ án một cách tốt nhất. Em rất biết ơn vì sự đồng hành và sự tin tưởng của thầy. Kính chúc thầy sức khỏe dồi dào và thành công trong sự nghiệp giảng dạy.

Trân trọng gửi đến thầy, ngày 29 tháng 10 năm 2023.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI	5
1.1. LÝ DO CHỌN ĐỀ TÀI	5
1.2. MỤC TIÊU & PHẠM VI	5
1.3. PHƯƠNG PHÁP NGHIÊN CỨU	6
CHƯƠNG 2: TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG	7
2.1. TỔNG QUAN VỀ LẬP TRÌNH WEB TĨNH	7
2.2. NGÔN NGỮ VÀ CÔNG CỤ SỬ DỤNG	7
2.2.1. HTML5 (HyperText Markup Language)	7
2.2.2. CSS3 (Cascading Style Sheets)	7
2.2.3. JavaScript (ES6+)	7
2.2.4. Thư viện p5.js	7
2.2.5. Git & GitHub	8
2.2.6. GitHub Pages	8
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	9
3.1. PHÂN TÍCH YÊU CẦU	9
3.1.1. Yêu cầu chức năng (Functional)	9
3.1.2. Yêu cầu phi chức năng (Non-functional)	9
3.2. THIẾT KẾ GIAO DIỆN (UI/UX)	9
3.2.1. Bố cục các trang chính	9
3.2.2. Bảng màu và font chữ	17
3.3. THIẾT KẾ CẤU TRÚC WEBSITE	18
CHƯƠNG 4. XÂY DỰNG VÀ TRIỂN KHAI WEBSITE	19
4.1. TRIỂN KHAI FRONT-END	19
4.1.1. Xây dựng cấu trúc HTML	19
4.1.2. Định dạng giao diện với CSS	20
4.1.3. Xử lý logic với JavaScript	25
4.2. QUẢN LÝ NỘI DUNG(BLOG & GALLERY)	26
4.3. TRIỂN KHAI TRÊN GITHUB PAGE	28
4.3.1. Push code lên GitHub	29
4.3.2. Thiết lập GitHub Pages	30

4.3.3. Kiểm thử và hoàn thiện	31
CHƯƠNG 5. KẾT QUẢ VÀ ĐÁNH GIÁ	33
5.1. TRÌNH BÀY KẾT QUẢ	34
5.2. ĐÁNH GIÁ	35
5.2.1. Ưu điểm	35
5.2.2. Hạn chế	35
CHƯƠNG 6. TÀI LIỆU THAM KHẢO	36
PHỤ LỤC	37

DANH MỤC HÌNH

Hình 3.1.1: Giao diện trang chủ (Home)	11
Hình 3.1.2: Giao diện trang chủ (Home)	11
Hình 3.1.3: Giao diện trang chủ (Home)	12
Hình 3.1.1: Giao diện trang chủ (Home)	12
Hình 3.2.1: Giao diện trang Blog	13
Hình 3.2.2: Giao diện trang Blog	13
Hình 3.2.3: Giao diện trang Blog	14
Hình 3.3.1: Giao diện trang Gallery	14
Hình 3.3.2: Giao diện trang Gallery	15
Hình 3.4.1: Giao diện Modal xem chi tiết	16
Hình 3.4.2: Giao diện Modal xem chi tiết	17
Hình 3.4.3: Giao diện Modal xem chi tiết	18
Hình 3.5: Sơ đồ cấu trúc thư mục dự án	19
Hình 4.1: Cấu trúc HTML thẻ <nav>	20
Hình 4.2: Cấu trúc HTML thẻ <hero>	20
Hình 4.3: Cấu trúc HTML bộ lọc bài viết	21
Hình 4.4: Cấu trúc HTML lưới bài viết	21
Hình 4.5: Định dạng cho blog trong CSS	22
Hình 4.6: Định dạng cho blog trong CSS	23
Hình 4.7: Mã CSS cho Responsive Design (Media Query) trong style.css phần 1	24
Hình 4.8: Mã CSS cho Responsive Design (Media Query) trong style.css phần 2	25
Hình 4.9: Hàm initializeCardAnimations IntersectionObserver trong script.js	26
Hình 4.10: Hàm draw() và class Particle trong hero-art.js	27
Hình 4.11: Cấu trúc mảng dữ liệu blogPosts trong blog.js	28
Hình 4.12: Cấu trúc mảng dữ liệu galleryItems trong blog.js	29
Hình 4.13: Logic Hiển thị Modal (Tập: blog.js)	29
Hình 4.14: Minh họa cách push project lên Github	31
Hình 4.15: Cấu hình GitHub Pages để triển khai từ nhánh 'main'	32
Hình 5.1: Giao diện trang chủ hoàn thiện với hiệu ứng particle	34

<i>Hình 5.2: Chức năng lọc bài viết trang Blog</i>	34
<i>Hình 5.3: Chức năng xem chi tiết Gallery (Modal)</i>	35

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. LÝ DO CHỌN ĐỀ TÀI

Trong kỷ nguyên số hóa hiện nay, Internet đã trở thành một phần không thể thiếu trong cuộc sống. Hàng ngày, hàng tỷ thiết bị trên toàn thế giới kết nối và giao tiếp với nhau thông qua mạng máy tính. Đằng sau những ứng dụng web, mobile app, hay dịch vụ trực tuyến mà chúng ta sử dụng hàng ngày là một hệ thống phức tạp của các giao thức mạng, kiến trúc client-server, và các kỹ thuật lập trình mạng.

Xuất phát từ nhu cầu cá nhân và mong muốn đóng góp cho cộng đồng học tập, em quyết định chọn đề tài “Xây dựng Website Blog Cá Nhân - Imanu's Lab” với các lý do sau:

- Ứng dụng kiến thức Lập Trình Mạng vào thực tế.
- Chia sẻ kiến thức với cộng đồng.
- Xây dựng kỹ năng Full-stack Development.
- Tạo portfolio cá nhân.

1.2. MỤC TIÊU & PHẠM VI

Mục tiêu:

1. Về mặt kỹ thuật: Xây dựng thành công một website blog cá nhân hoàn chỉnh gồm 3 trang (Home, Blog, Gallery) bằng công nghệ Front-end.
2. Về mặt chức năng: Triển khai các tính năng tương tác hiện đại như hiệu ứng particle nền (p5.js), animation khi cuộn trang (IntersectionObserver), lọc bài viết, tìm kiếm nội dung, và xem chi tiết bằng modal.
3. **Về mặt nội dung:** Tổ chức và hiển thị một cách khoa học các bài viết kỹ thuật (Java, JavaScript) và các tác phẩm nghệ thuật (Gallery).

Trong phạm vi:

1. Website được xây dựng hoàn toàn bằng công nghệ Front-end (HTML, CSS, JavaScript);
2. Dữ liệu nội dung (bài viết, hình ảnh) được quản lý tĩnh (hard-coded) trong các tệp JavaScript (blog.js, gallery.js);
3. Triển khai website công khai qua dịch vụ GitHub Pages.

Ngoài phạm vi:

1. Không xây dựng hệ thống Back-end;

2. Không sử dụng cơ sở dữ liệu (Database);
3. Không bao gồm các chức năng phức tạp như: hệ thống đăng nhập/đăng ký, bình luận, quản lý nội dung (CMS).

1.3. PHƯƠNG PHÁP NGHIÊN CỨU

1. **Nghiên cứu:** Tìm hiểu các công nghệ Front-end phù hợp (HTML, CSS, JS) và thư viện đồ họa (p5.js) để thực hiện hiệu ứng nền.
2. **Phân tích & Thiết kế:** Phân tích các yêu cầu chức năng và phi chức năng. Thiết kế bố cục (layout) và trải nghiệm người dùng (UX) cho cả 3 trang.
3. **Triển khai (Coding):** Tiến hành xây dựng code HTML, CSS và JavaScript cho từng thành phần (component) và từng trang.
4. **Quản lý & Triển khai:** Sử dụng Git để quản lý phiên bản mã nguồn và triển khai sản phẩm cuối cùng lên GitHub Pages.

CHƯƠNG 2: TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG

2.1. TỔNG QUAN VỀ LẬP TRÌNH WEB TĨNH

Nội dung: Giải thích ngắn gọn web tĩnh là gì (các tệp HTML, CSS, JS được gửi thẳng từ server đến trình duyệt mà không cần xử lý phía máy chủ). Nêu các ưu điểm mà dự án này có được:

- Tốc độ: Tải trang cực kỳ nhanh.
- Bảo mật: Giảm thiểu rủi ro tấn công vì không có cơ sở dữ liệu hay back-end.
- Chi phí: Dễ dàng triển khai và lưu trữ miễn phí qua GitHub Pages.

2.2. NGÔN NGỮ VÀ CÔNG CỤ SỬ DỤNG

2.2.1. HTML5 (HyperText Markup Language)

Vai trò: Đóng vai trò là “khung xương”, xây dựng cấu trúc ngữ nghĩa cho 3 trang chính của website (index.html, blog.html, gallery.html).

Ứng dụng: Sử dụng các thẻ HTML ngữ nghĩa (Semantic HTML) như `<nav>`, `<header>`, `<section>`, `<article>`, `<footer>` để tăng tính rõ ràng cho code và hỗ trợ SEO.

2.2.2. CSS3 (Cascading Style Sheets)

Vai trò: Chịu trách nhiệm tạo giao diện, định dạng bố cục và trang điểm” cho website.

Ứng dụng: Sử dụng các kỹ thuật layout hiện đại như Flexbox và Grid để sắp xếp các thẻ bài viết (blog card) và ảnh (gallery card). Sử dụng Media Queries để đảm bảo website có thiết kế đáp ứng (Responsive Design), hiển thị tốt trên cả máy tính và điện thoại.

2.2.3. JavaScript (ES6+)

Vai trò: Là “bộ não” xử lý toàn bộ logic và tính tương tác cho website.

Ứng dụng:

- Xử lý DOM: Ẩn/hiện menu, điều khiển pop-up (Modal) khi xem chi tiết.
- Sự kiện: Xử lý các sự kiện click, cuộn trang.
- Logic nghiệp vụ: Viết các hàm lọc bài viết (filterPosts) và tìm kiếm (performSearch) trong blog.js.
- Hiệu ứng: Sử dụng IntersectionObserver (script.js) để phát hiện khi người dùng cuộn đến một thành phần và kích hoạt hiệu ứng xuất hiện (animation).

2.2.4. Thư viện p5.js

Vai trò: Là một thư viện JavaScript chuyên về đồ họa sáng tạo (creative coding).

Ứng dụng: Đây là công nghệ đặc biệt của dự án, được sử dụng trong tệp hero-art.js để vẽ hiệu ứng nền gồm các hạt (particle) chuyển động và tương tác với chuột, thể hiện rõ chủ đề “Art Flow” của website.

2.2.5. Git & GitHub

Git: Hệ thống quản lý phiên bản phân tán, giúp theo dõi, lưu trữ các thay đổi của mã nguồn.

GitHub: Nền tảng lưu trữ (hosting) các kho mã nguồn Git, đóng vai trò là nơi sao lưu code và là nền tảng để triển khai website.

2.2.6. GitHub Pages

Vai trò: Dịch vụ miễn phí từ GitHub cho phép triển khai (deploy) các website tĩnh trực tiếp từ kho mã nguồn.

Ứng dụng: Giúp đưa website "Imanu's Lab" lên Internet để mọi người có thể truy cập.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. PHÂN TÍCH YÊU CẦU

3.1.1. Yêu cầu chức năng (Functional)

Hệ thống (chung): Website phải có 3 trang riêng biệt: Trang chủ, Blog, và Gallery.

Điều hướng: Thanh điều hướng (Navbar) phải cố định và cho phép chuyển trang.

Trang chủ: Phải hiển thị được:

1. Thông tin giới thiệu cá nhân (Hero section) và hiệu ứng particle;
2. Phần “About me”;
3. Danh sách các kỹ năng (Skills).

Trang Blog & Gallery:

1. Thông tin giới thiệu cá nhân (Hero section) và hiệu ứng particle;
2. Phần “About me”.
3. Danh sách các kỹ năng (Skills).

Chức năng Lọc (Blog): Phải cho phép người dùng lọc bài viết theo các danh mục: “All”, “Java”, “JavaScript”, “Chứng chỉ”.

Chức năng Tìm kiếm: Phải có ô tìm kiếm cho phép tìm kiếm *cả* nội dung bài viết và mô tả gallery. Ngoài ra còn có phần Kết luận, Tài liệu tham khảo, và Phụ lục (source code, screenshots).

Liên hệ: Hiển thị một form liên hệ (chỉ cần giao diện).

3.1.2. Yêu cầu phi chức năng (Non-functional)

Thiết kế: Giao diện phải hiện đại, tối giản, chuyên nghiệp.

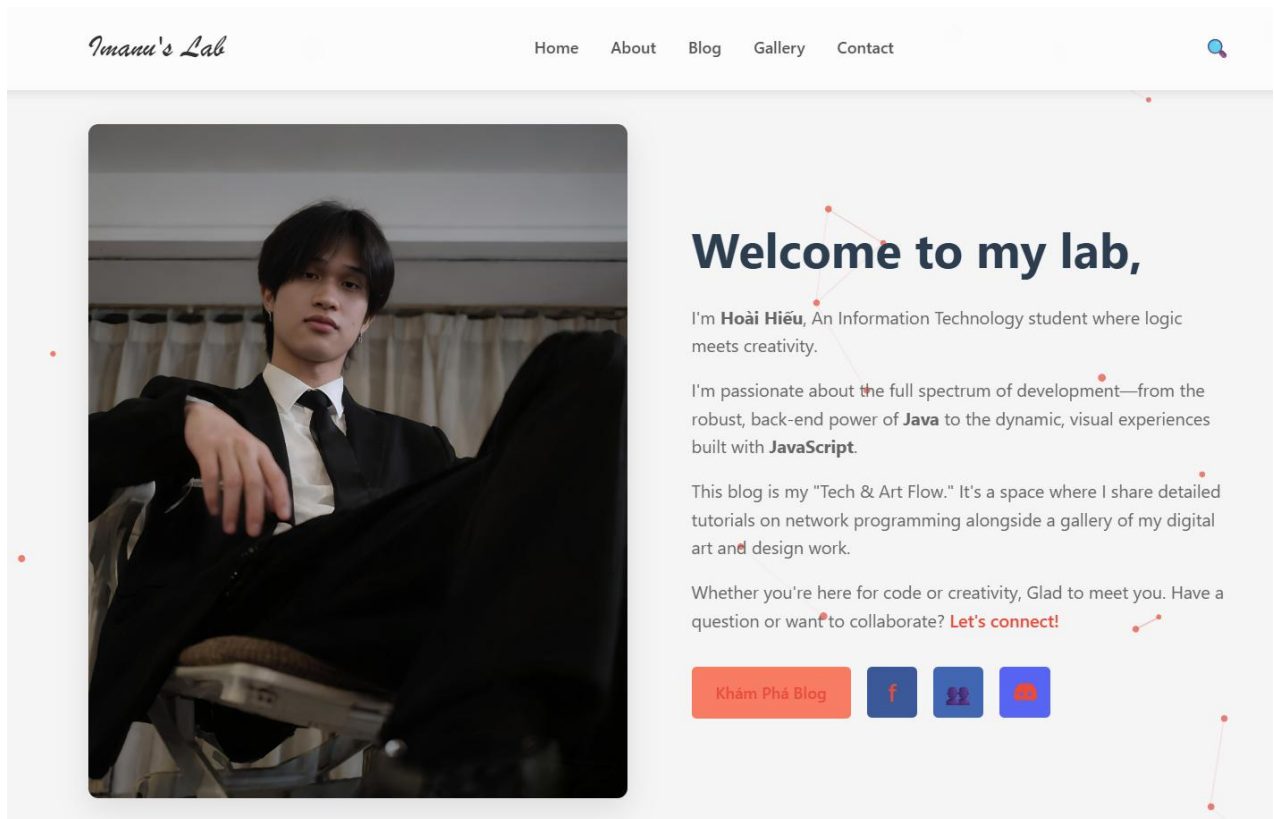
Tương thích: Website phải hiển thị tốt trên các trình duyệt phổ biến (Chrome, Firefox) và các kích thước màn hình (máy tính, điện thoại).

Hiệu suất: Tốc độ tải trang phải nhanh (do là web tĩnh).

Bảo trì: Cấu trúc code phải rõ ràng, chia tách các tệp JavaScript theo đúng chức năng (script.js, blog.js, gallery.js, hero-art.js) để dễ dàng bảo trì và nâng cấp.

3.2. THIẾT KẾ GIAO DIỆN (UI/UX)

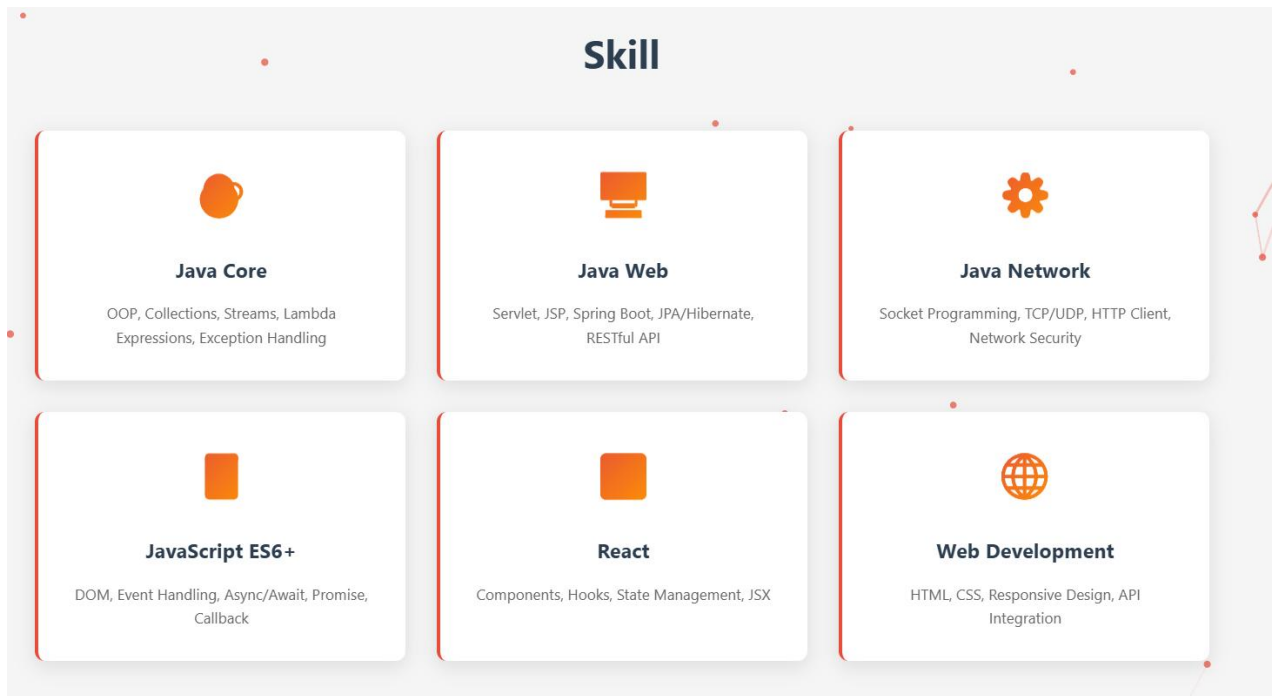
3.2.1. Bố cục các trang chính



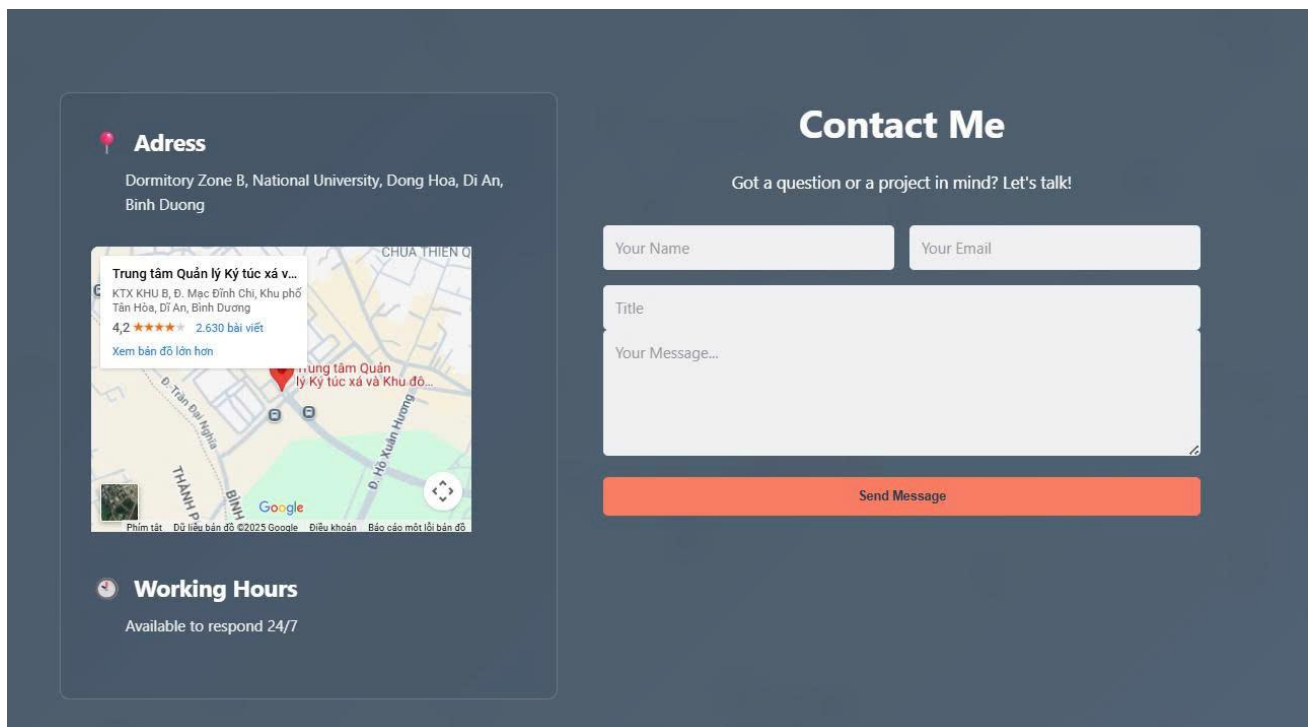
Hình 3.1.1: Giao diện trang chủ (Home)



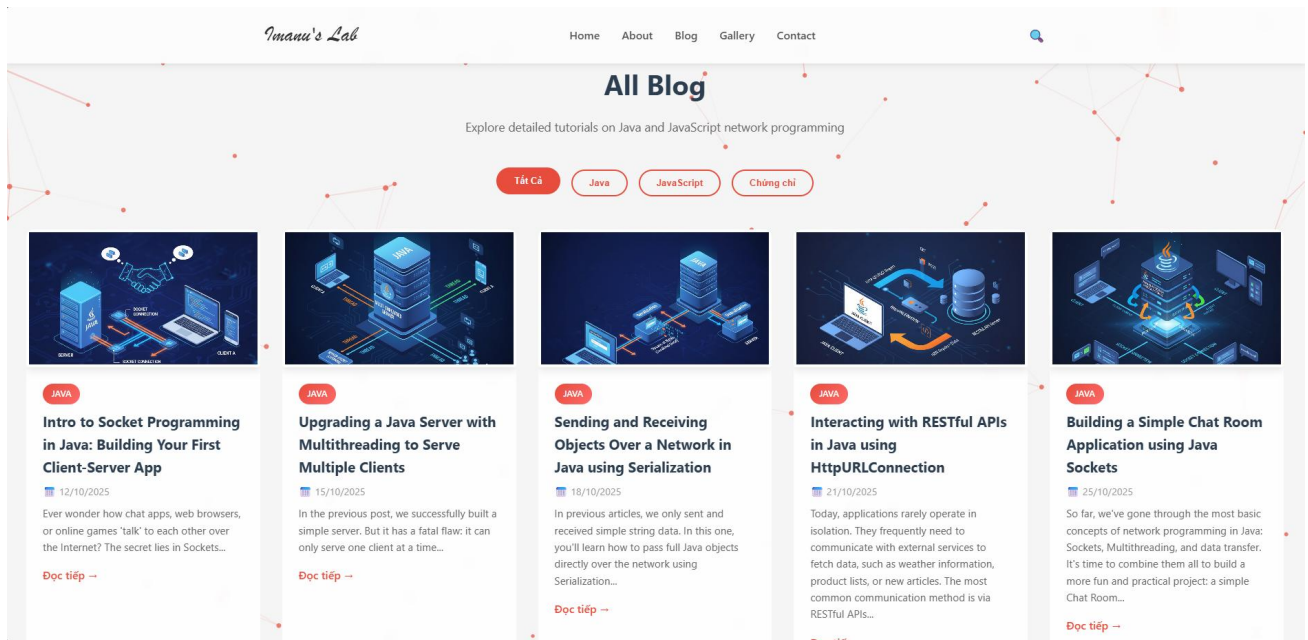
Hình 3.1.2: Giao diện trang chủ (Home)



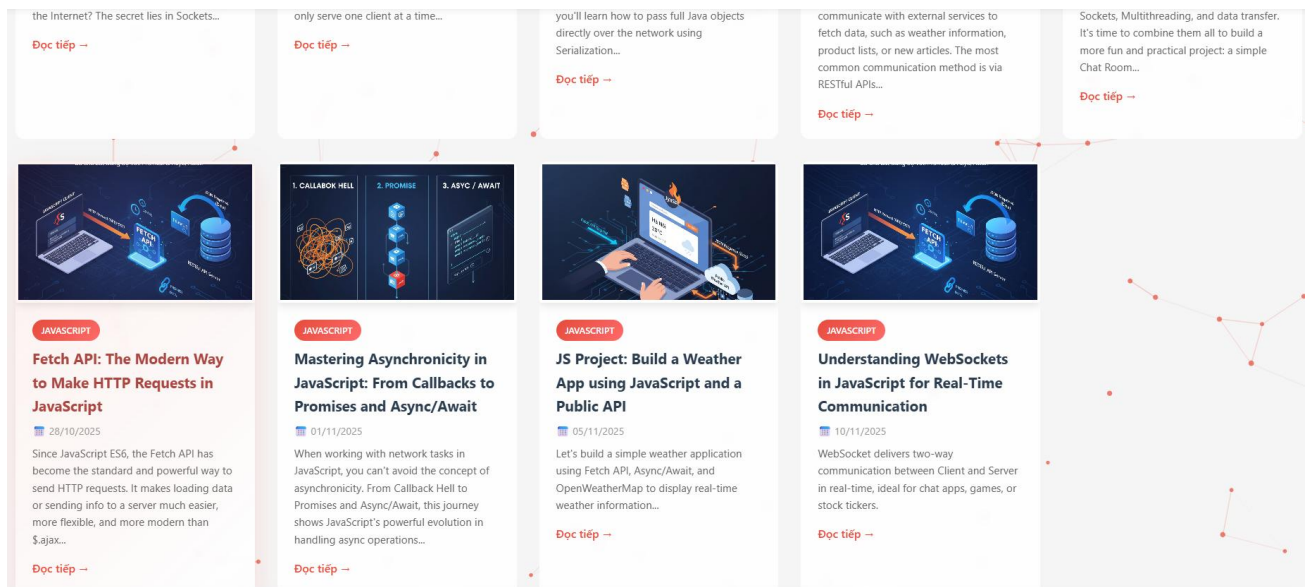
Hình 3.1.3: Giao diện trang chủ (Home)



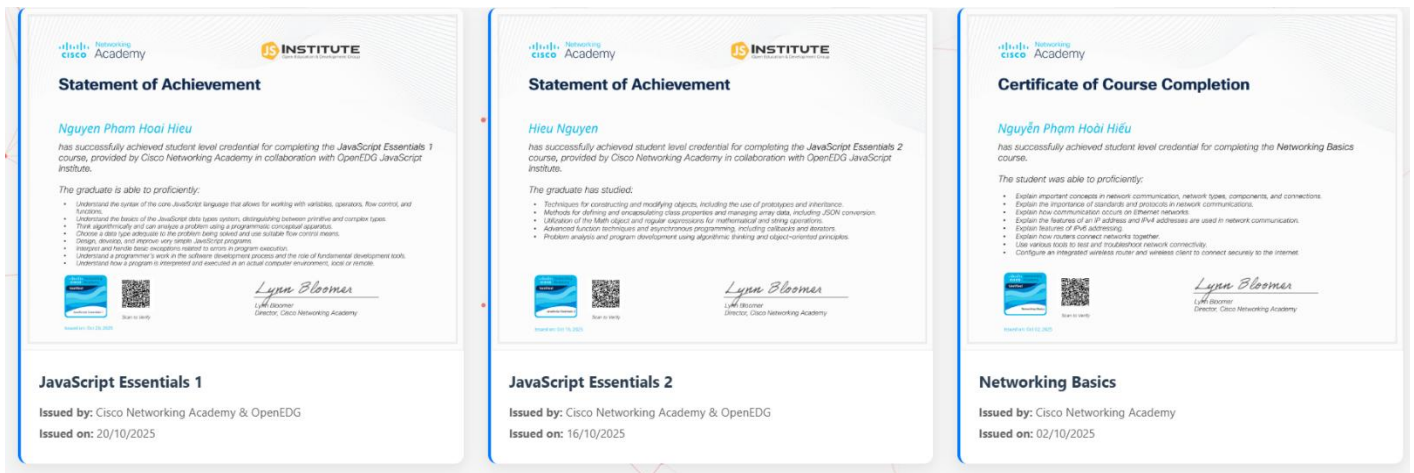
Hình 3.1.4: Giao diện trang chủ (Home)



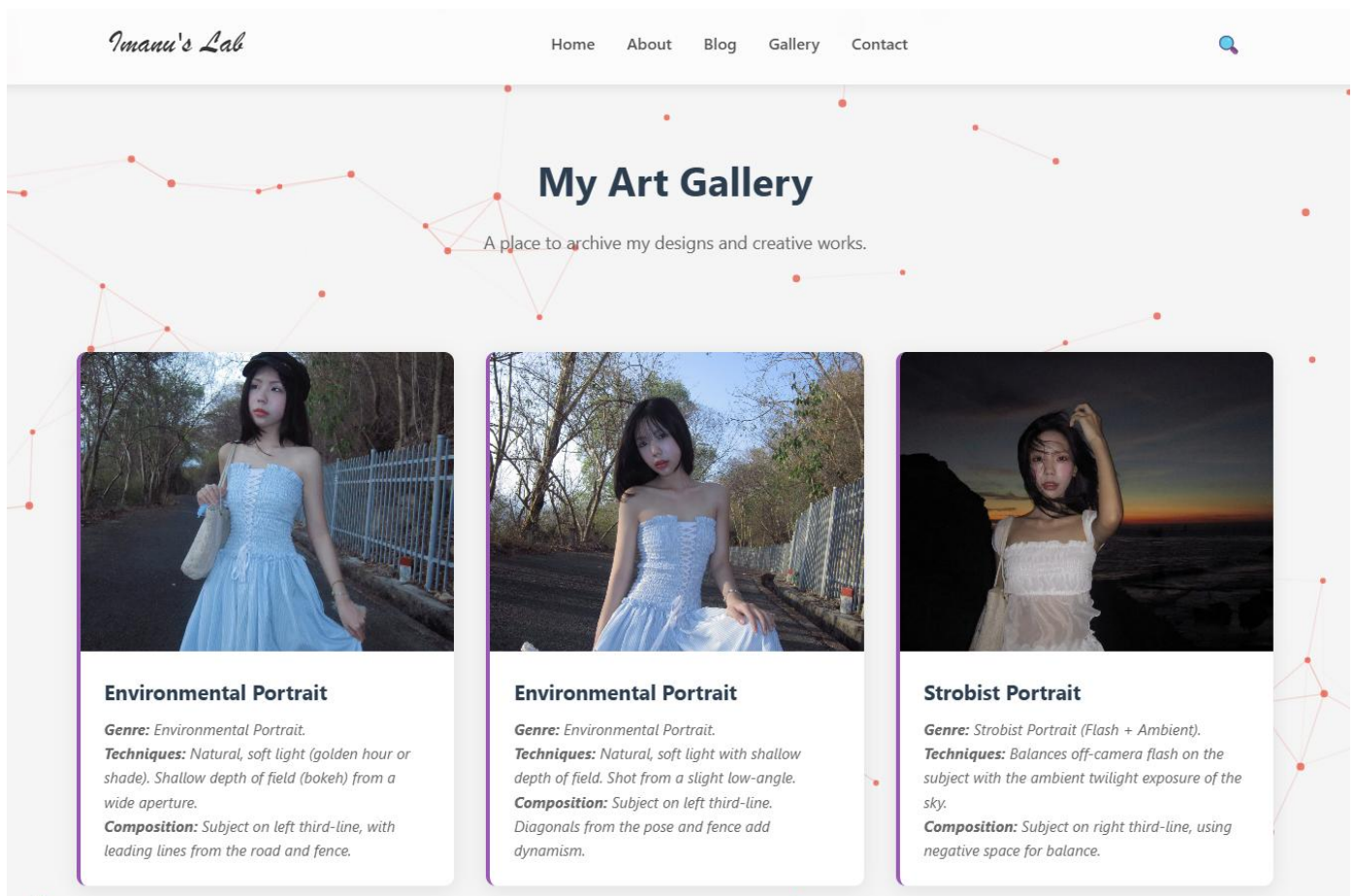
Hình 3.2.1: Giao diện trang Blog



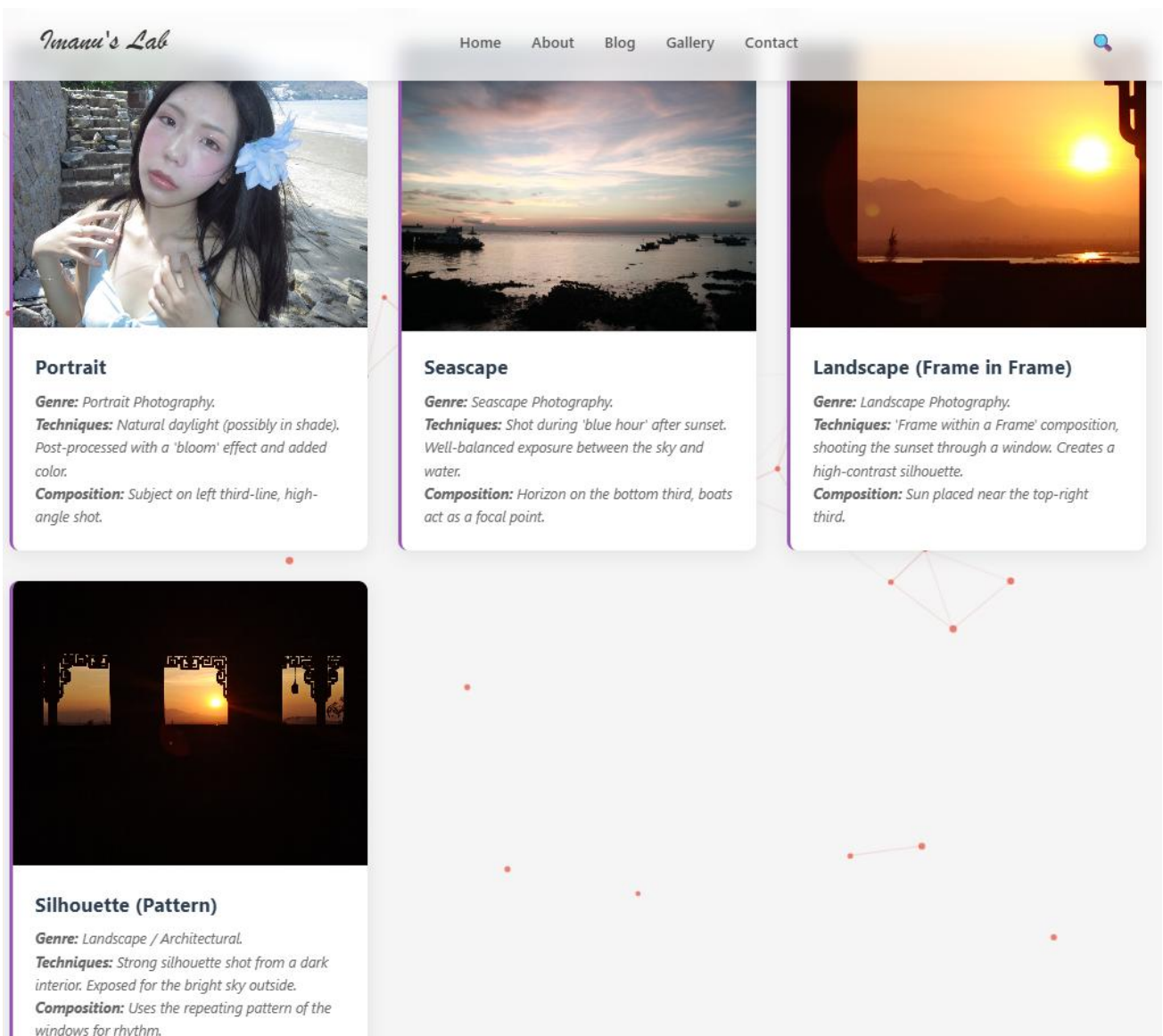
Hình 3.2.2: Giao diện trang Blog



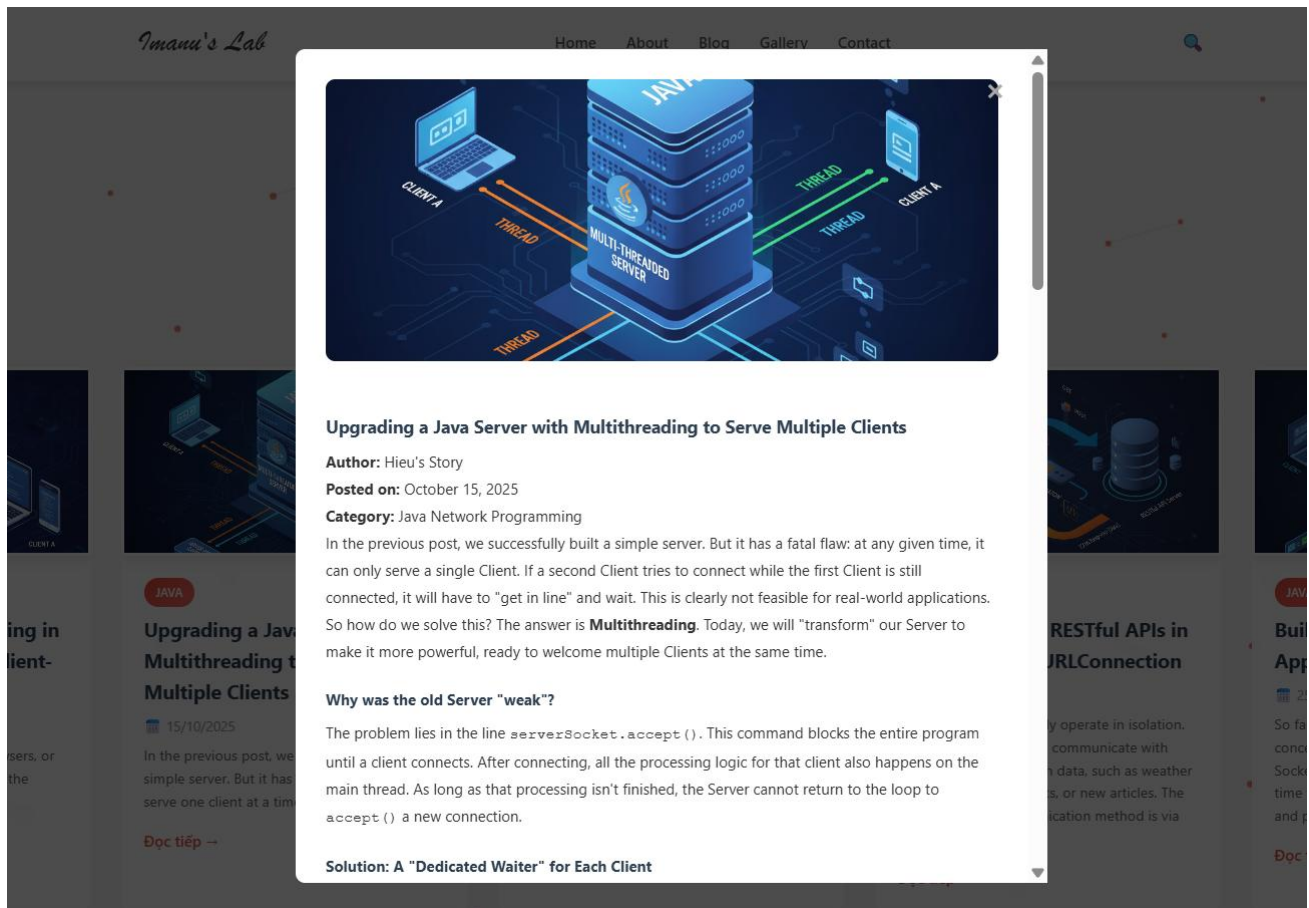
Hình 3.2.3: Giao diện trang Blog



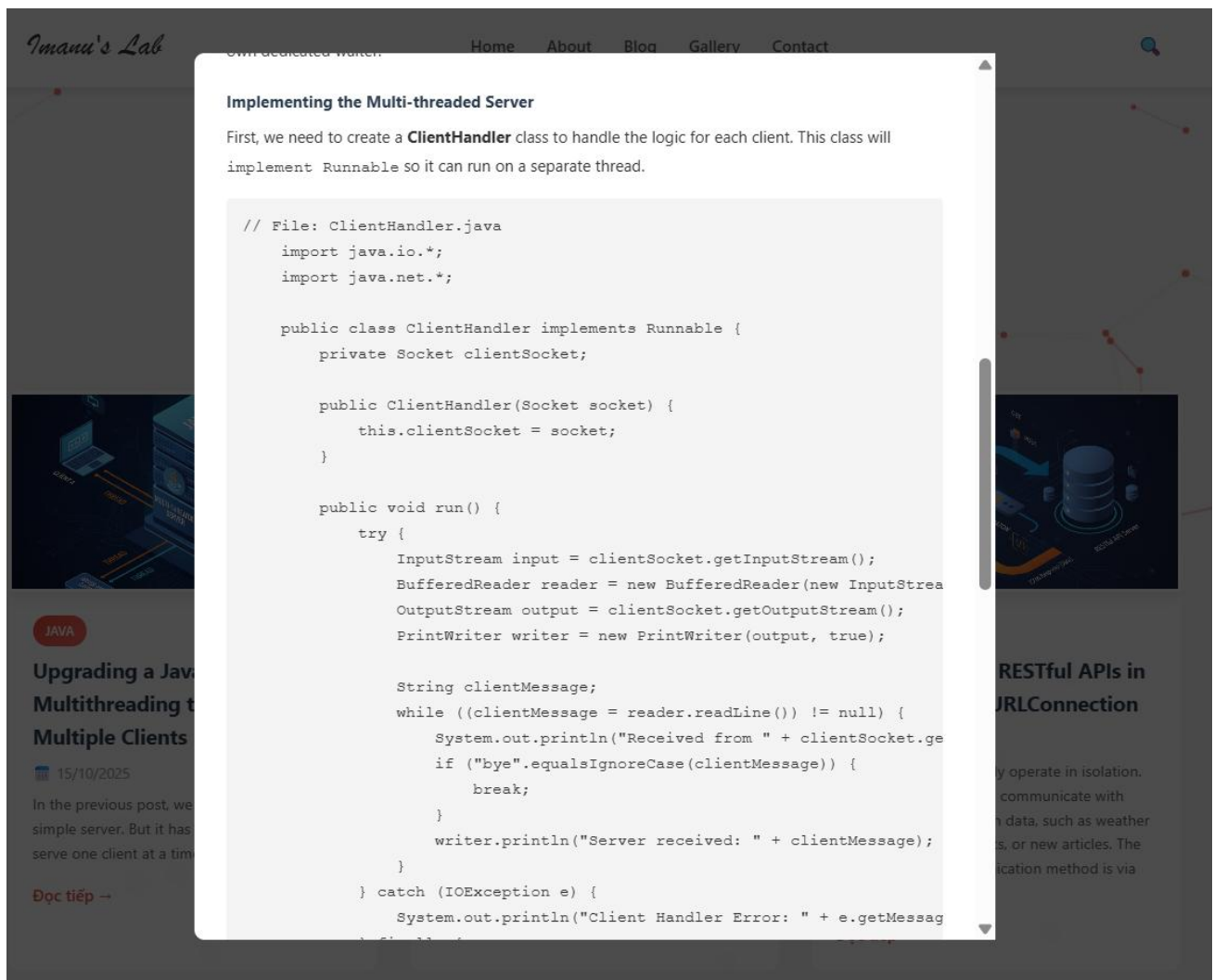
Hình 3.3.1: Giao diện trang Gallery



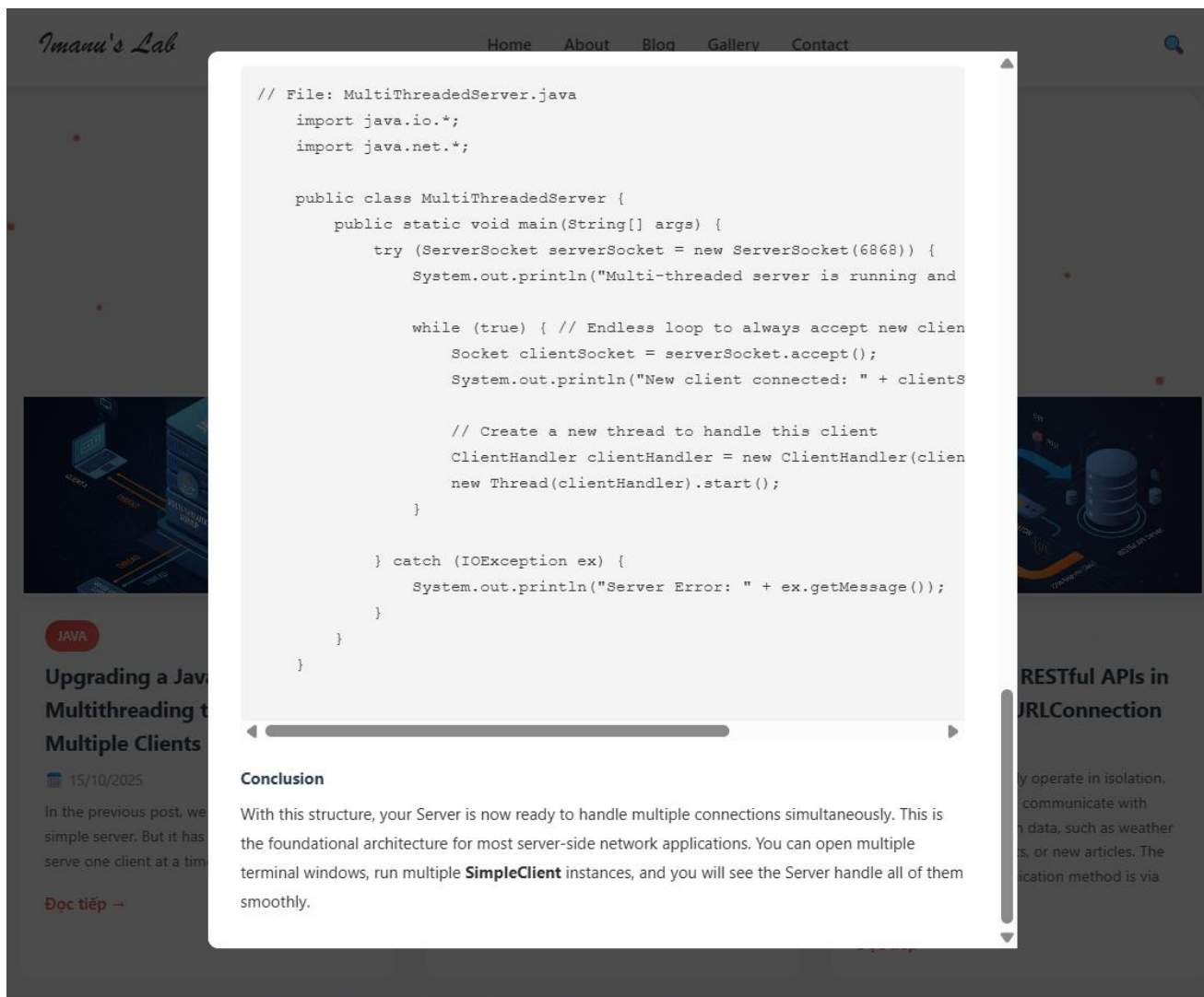
Hình 3.3.2: Giao diện trang Gallery



Hình 3.4.1: Giao diện Modal xem chi tiết



Hình 3.4.2: Giao diện Modal xem chi tiết



Hình 3.4.3: Giao diện Modal xem chi tiết

3.2.2. Bảng màu và Font chữ

Font chữ:

- Font chính (cho nội dung, tiêu đề phụ): Segoe UI, Tahoma, Geneva, Verdana, sans-serif.
- Font Logo ('Imanu's Lab'): Brush Script MT, cursive (hoặc Dancing Script từ Google Fonts).

Bảng màu chủ đạo:

- **Màu Cam san hô (Coral Red):** #e74c3c (Dùng cho các nút, tiêu đề, và hiệu ứng particle).
- **Màu Xanh than (Dark Blue):** #2c3e50 (Dùng cho các tiêu đề chính, nền footer).
- **Màu Nền (Background):** #f5f5f5 (Màu xám rất nhạt).

- **Màu Chữ (Text):** #333 (Màu đen/xám đậm cho nội dung).

3.3. THIẾT KẾ CẤU TRÚC WEBSITE

Sơ đồ cấu trúc thư mục



Hình 3.5: Sơ đồ cấu trúc thư mục dự án

CHƯƠNG 4. XÂY DỰNG VÀ TRIỂN KHAI WEBSITE

4.1. TRIỂN KHAI FRONT-END

4.1.1. Xây dựng cấu trúc HTML

```
<header>
  <nav>
    <div class="logo">Imanu's Lab</div>
    <ul class="nav-menu">
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About</a></li>
      <li><a href="blog.html">Blog</a></li>
      <li><a href="gallery.html">Gallery</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
    <div class="nav-icons">
      <button id="open-search-btn">🔍</button>
    </div>
  </nav>
</header>
```

Hình 4.1: Cấu trúc HTML thẻ <nav>

```
<section class="hero" id="home">
  <div class="hero-image">
    
  </div>
  <div class="hero-content">
    <h1>Welcome to my lab,</h1>
    <p>I'm <strong>Hoài Hiếu</strong>, An Information Technology student where logic meets creativity.</p>
    <p>I'm passionate about the full spectrum of development—from the robust, back-end power of <strong>Java</strong> to the dynamic, visual experiences built with <strong>JavaScript</strong>.</p>
    <p>This blog is my "Tech & Art Flow." It's a space where I share detailed tutorials on network programming alongside a gallery of my digital art and design work.</p>
    <p>Whether you're here for code or creativity, Glad to meet you. Have a question or want to collaborate?<br>
    <a href="#contact">Let's connect!</a></p>
    <div class="social-buttons">
      <a href="blog.html" class="btn btn-primary">Khám Phá Blog</a> <a href="https://www.facebook.com/imanutetsuya/" class="btn btn-social btn-facebook" target="_blank" rel="noopener noreferrer">fb</a>
      <a href="#" class="btn btn-social btn-group">📱</a>
      <a href="https://discord.gg/Sz3hnmVet" class="btn btn-social btn-discord" target="_blank" rel="noopener noreferrer">Discord</a>
    </div>
  </div>
</section>
```

Hình 4.2: Cấu trúc HTML thẻ <hero>

```

<div class="blog-filters">
  <button class="filter-btn active" data-filter="all">Tất Cả</button>
  <button class="filter-btn" data-filter="java">Java</button>
  <button class="filter-btn" data-filter="javascript">JavaScript</button>
  <button class="filter-btn" data-filter="certificates">Chứng chỉ</button>
</div>

```

Hình 4.3: Cấu trúc HTML bộ lọc bài viết

```

<div class="blog-grid" id="blog-container">
  </div>
<div id="certificates-container" class="certificates-grid">
  <div class="certificate-card">
    
    <div class="certificate-info">
      <h4>JavaScript Essentials 1</h4>
      <p><strong>Issued by:</strong> Cisco Networking Academy & OpenEDG</p>
      <p><strong>Issued on:</strong> 20/10/2025</p>
    </div>
  </div>
</div>

```

Hình 4.4: Cấu trúc HTML lưới bài viết

4.1.2. Định dạng giao diện với CSS


```

.blog-grid {
  grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));
  gap: 2rem;
}

.blog-card {
  background: ■ rgba(255, 255, 255, 0.9); /* Hơi trong suốt */
  backdrop-filter: blur(5px);
  border-radius: 12px;
  overflow: hidden;
}

.blog-card:nth-child(1) { animation-delay: 0.2s; }
.blog-card:nth-child(2) { animation-delay: 0.4s; }
.blog-card:nth-child(3) { animation-delay: 0.6s; }
.blog-card:nth-child(4) { animation-delay: 0.2s; }
.blog-card:nth-child(5) { animation-delay: 0.4s; }
.blog-card:nth-child(6) { animation-delay: 0.6s; }
.blog-card:nth-child(7) { animation-delay: 0.2s; }
.blog-card:nth-child(8) { animation-delay: 0.4s; }
.blog-card:nth-child(9) { animation-delay: 0.6s; }

.blog-card:hover {
  transform: translateY(-10px) scale(1.02);
  box-shadow: 0 15px 40px ■ rgba(231, 76, 60, 0.15);
  border-left-color: ■ #c0392b;
}

```

Hình 4.5: Định dạng cho blog trong CSS

```

.blog-image {
  width: 100%;
  height: 200px;
  object-fit: cover;
  /* 1. Thêm một đường viền trắng mỏng giống như viền ảnh */
  border: 4px solid white;

  /* 2. Thêm hiệu ứng đổ bóng tinh tế */
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

  /* 3. Thêm một viền ngoài cũng mỏng (tùy chọn) */
  border-radius: 4px; /* Hơi bo góc một chút */

  /* 4. Đảm bảo ảnh không bị méo */
  display: block;
}

.blog-content {
  padding: 1.5rem;
}

.blog-category {
  display: inline-block;
  background: linear-gradient(135deg, #e74c3c, #ff6b6b);
  color: white;
  padding: 0.3rem 0.8rem;
  border-radius: 15px;
  font-size: 0.8rem;
  font-weight: 600;
  margin-bottom: 0.5rem;
  animation: slideInRight 0.8s ease 0.3s both;
}

```

Hình 4.6: Định dạng cho blog trong CSS


```

@media (max-width: 1024px) {
  .about-content {
    grid-template-columns: 1fr;
  }

  .about-stats {
    grid-template-columns: repeat(3, 1fr);
  }
}

@media (max-width: 768px) {
  nav {
    flex-wrap: wrap;
    padding: 1rem;
  }

  .Logo {
    font-size: 1.2rem;
  }

  .nav-menu {
    flex-wrap: nowrap;
    overflow-x: auto;
    gap: 1rem;
    order: 3;
    width: 100%;
    -ms-overflow-style: none; /* IE/Edge */
    scrollbar-width: none; /* Firefox */
  }

  .nav-menu::-webkit-scrollbar {
    display: none; /* Chrome, Safari và Opera */
  }

  .hero {
    background: transparent;
    position: relative;
    z-index: 1;
    grid-template-columns: 1fr;
    gap: 2rem;
  }
}

```

Hình 4.7: Mã CSS cho Responsive Design (Media Query) trong style.css phần 1

```

.hero-content h1 {
  font-size: 2rem;
}

.hero-content p {
  font-size: 1rem;
}

.about-section h2,
.skills-section h2,
.blog-section h2,
.contact-section h2 {
  font-size: 2rem;
}

.about-stats {
  grid-template-columns: 1fr;
}

.form-row {
  grid-template-columns: 1fr;
}

#subject {
  grid-column: auto;
}

.footer-container {
  grid-template-columns: 1fr;
  gap: 2rem;
}

.blog-filters {
  gap: 0.5rem;
}

.filter-btn {
  padding: 0.5rem 1rem;
  font-size: 0.9rem;
}

.blog-grid {
  grid-template-columns: 1fr;
}

```

Hình 4.8: Mã CSS cho Responsive Design (Media Query) trong style.css phần 2

4.1.3. Xử lý logic với JavaScript

```
// ===== HIỆU ỨNG THẺ KHI CUỘN =====  
function initializeCardAnimations() {  
    const observer = new IntersectionObserver((entries) => {  
        entries.forEach(entry => {  
            if (entry.isIntersecting) {  
                entry.target.style.opacity = '1';  
                entry.target.style.transform = 'translateY(0)';  
                observer.unobserve(entry.target);  
            }  
        });  
    }, { threshold: 0.1 });  
  
    document.querySelectorAll('.blog-card, .skill-card').forEach(el => {  
        observer.observe(el);  
    });  
}
```

Hình 4.9: Hàm `initializeCardAnimations` sử dụng `IntersectionObserver` trong `script.js`

```

// Lớp Particle (Hạt)
class Particle {
  constructor() {
    // Vị trí ngẫu nhiên ban đầu
    this.x = random(width);
    this.y = random(height);
    // Vận tốc ngẫu nhiên
    this.vx = random(-0.8, 0.8);
    this.vy = random(-0.8, 0.8);
    // Kích thước lớn hơn để dễ nhìn
    this.size = random(5, 8);
    // Màu đỏ cam theo theme website - TĂNG ALPHA
    this.color = color(231, 76, 60, 180);
  }

  // Cập nhật vị trí
  update() {
    this.x += this.vx;
    this.y += this.vy;

    // Đảo chiều nếu chạm cạnh
    if (this.x < 0 || this.x > width) {
      this.vx *= -1;
    }
    if (this.y < 0 || this.y > height) {
      this.vy *= -1;
    }
  }

  // Vẽ hạt
  show() {
    noStroke();
    fill(this.color);
    circle(this.x, this.y, this.size);
  }
}

```

Hình 4.10: Hàm draw() và class Particle trong hero-art.js

4.2. QUẢN LÝ NỘI DUNG (BLOG & GALLERY)

Nội dung của website không được lưu trong cơ sở dữ liệu mà được quản lý tĩnh bằng cách lưu trữ trực tiếp dưới dạng các mảng (Array) đối tượng (Object) trong các tệp JavaScript.

- Nội dung các bài viết blog được lưu trong mảng const blogPosts tại tệp blog.js.

- Nội dung các hình ảnh gallery được lưu trong mảng const galleryItems tại tệp gallery.js.

JavaScript sẽ đọc các mảng này và tự động render (vẽ) ra giao diện HTML tương ứng.

```
// ===== HIỂN THỊ CÁC BÀI VIẾT BLOG =====
function renderBlogPosts() {
  const container = document.getElementById('blog-container');
  if (!container) return;
  container.innerHTML = '';

  blogPosts.forEach(post => {
    const card = createBlogCard(post);
    container.appendChild(card);
  });
}

function createBlogCard(post) {
  const card = document.createElement('div');
  card.className = 'blog-card';
  card.setAttribute('data-category', post.category);

  card.innerHTML = `
    
    <div class="blog-content">
      <span class="blog-category">${post.category.toUpperCase()}</span>
      <h3>${post.title}</h3>
      <div class="blog-date">📅 ${post.date}</div>
      <p class="blog-excerpt">${post.excerpt}</p>
      <a href="#" class="read-more">Đọc tiếp →</a>
    </div>
  `;

  // Cải tiến: Nhấp vào bất kỳ đâu trên thẻ cũng sẽ mở bài viết
  card.addEventListener('click', (e) => {
    e.preventDefault(); // Ngăn hành vi mặc định của thẻ <a>
    showPostDetail(post);
  });

  return card;
}
```

Hình 4.11: Cấu trúc mảng dữ liệu blogPosts trong blog.js


```
const blogPosts = [
  {
    id: 1,
    title: "Intro to Socket Programming in Java: Building Your First Client-Server App",
    category: "java",
    date: "12/10/2025",
    excerpt: "Ever wonder how chat apps, web browsers, or online games 'talk' to each other over the Internet? The secret lies in Sockets...",
    content: `<h3>Intro to Socket Programming in Java: Building Your First Client-Server App</h3>
    <p><strong>Author:</strong> Hieu's Story</p>
    <p><strong>Posted on:</strong> October 12, 2025</p>
    <p><strong>Category:</strong> Java Network Programming</p>

    <p>Ever wonder how chat apps, web browsers, or online games "talk" to each other over the Internet? The secret lies in one of the most fundamen

    <p>In this article, we will build a super-simple Client-Server application together so you can understand the essence of the problem.</p>

    <h4>What is a Socket? Imagine a Phone Call</h4>
    <p>The easiest way to understand Sockets is to associate it with a phone call:</p>

    <p><strong>ServerSocket (Server-side):</strong> Like a person sitting and waiting for the phone to ring. They have a phone number (Port) and ar

    <p><strong>Socket (Client-side):</strong> Like the person who wants to make a call. They need to know the recipient's phone number (IP Address

    <p><strong>Communication Stream:</strong> When the call is connected, a two-way communication channel is formed. Both can talk and listen throu

    <h4>Let's get to the code: Building the Server</h4>
    <p>Our server will perform a simple task: listen for a connection on port 6868, receive a message from the client, print it to the screen, and

    <pre><code class="language-java"> File: SimpleServer.java
import java.io.*;
import java.net.*;

public class SimpleServer {
    public static void main(String[] args) {
```

Hình 4.12: Cấu trúc mảng dữ liệu galleryItems trong blog.js

```
function showPostDetail(post) {
  const modal = document.createElement('div');
  modal.className = 'modal';
  modal.innerHTML = `
    <div class="modal-content">
      <span class="modal-close">&times;</span>
      
        ${post.content}
      </div>
    </div>
  `;

  document.body.appendChild(modal);
  document.body.style.overflow = 'hidden'; // Ngăn cuộn trang nền

  // Chức năng đóng modal
  const closeModal = () => {
    modal.remove();
    document.body.style.overflow = 'auto';
  };

  modal.querySelector('.modal-close').addEventListener('click', closeModal);
  modal.addEventListener('click', (e) => {
    if (e.target === modal) {
      closeModal();
    }
  });
  document.addEventListener('keydown', (e) => {
    if (e.key === 'Escape') {
      closeModal();
    }
  }, { once: true });
}
```

Hình 4.13. Logic Hiển thị Modal (Tập: blog.js)

4.3. TRIỂN KHAI TRÊN GITHUB PAGES

Sau khi website “Imanu’s Lab” đã được hoàn thiện và kiểm thử đầy đủ ở môi trường local (máy cá nhân), bước cuối cùng là triển khai (deploy) website lên Internet để mọi người có thể truy cập công khai.

Nền tảng được chọn là **GitHub Pages** vì đây là dịch vụ miễn phí, hiệu suất cao (thông qua CDN), tối ưu cho website tĩnh và được tích hợp thẳng vào kho lưu trữ mã nguồn GitHub, giúp quy trình cập nhật vô cùng đơn giản.

Nội dung các bài viết blog được lưu trong mảng `const blogPosts` tại tệp `blog.js`.

4.3.1. Push code lên GitHub

Đầu tiên, toàn bộ mã nguồn của dự án (các tệp `index.html`, `style.css`, `blog.js`, `gallery.js`...) cần được đưa từ máy local lên kho lưu trữ (repository) trên GitHub. Quá trình này được thực hiện bằng các lệnh Git cơ bản trong terminal.

Các bước thực hiện:

1. **Khởi tạo Git:** `git init` (Nếu là lần đầu).
2. **Thêm file:** `git add .` để thêm tất cả các file vào “khu vực chờ”.
3. **Tạo phiên bản lưu:** `git commit -m “Hoan thien chuc nang, chuan bi deploy”` để lưu lại phiên bản code.
4. **Kết nối remote:** `git remote add origin <URL_REPO_CUA_BAN.git>` (Nếu là lần đầu).
5. **Đẩy code:** `git push -u origin main` để đẩy toàn bộ code lên nhánh main của kho lưu trữ GitHub.

```

GIGABYTE@Imanu MINGW64 /d/Imanu/Study/Demo2
$ git init
Initialized empty Git repository in D:/Imanu/Study/Demo2/.git/

GIGABYTE@Imanu MINGW64 /d/Imanu/Study/Demo2 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .vscode/
        blog.html
        blog.js
        gallery.html
        gallery.js
        hero-art.js
        image/
        index.html
        script.js
        style.css

nothing added to commit but untracked files present (use "git add" to track)

GIGABYTE@Imanu MINGW64 /d/Imanu/Study/Demo2 (master)
$ git add .
i
GIGABYTE@Imanu MINGW64 /d/Imanu/Study/Demo2 (master)
$ git commit -m "push"
[master (root-commit) flf0f1d] push
31 files changed, 3136 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 blog.html
create mode 100644 blog.js
create mode 100644 gallery.html
create mode 100644 gallery.js
create mode 100644 hero-art.js
create mode 100644 image/Avt.jpg
create mode 100644 image/img1.jpg
create mode 100644 image/img10.jpg
create mode 100644 image/img11.jpg
create mode 100644 image/img12.jpg
create mode 100644 image/img13.jpg
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 16 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (34/34), 25.33 MiB | 2.68 MiB/s, done.
Total 34 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ImanuBae/HoaiHieuBlog.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

GIGABYTE@Imanu MINGW64 /d/Imanu/Study/Demo2 (main)

```

Hình 4.14. Minh họa cách push project lên Github

4.3.2. Thiết lập GitHub Pages

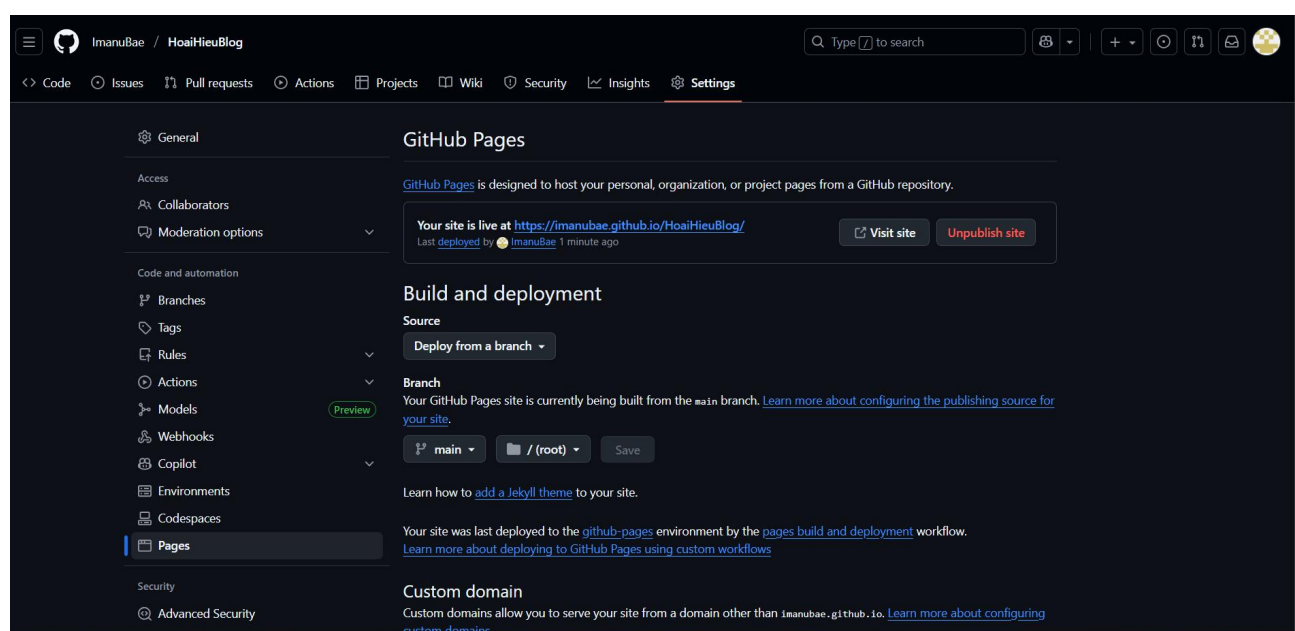
Đầu tiên, toàn bộ mã nguồn của dự án (các tệp index.html, style.css, blog.js, gallery.js...) cần được đưa từ máy local lên kho lưu trữ (repository) trên GitHub. Quá trình này được thực hiện bằng các lệnh Git cơ bản trong terminal.

Các bước thực hiện:

Khi mã nguồn đã có mặt trên GitHub, việc kích hoạt website rất đơn giản:

1. Truy cập kho lưu trữ (repository) của dự án trên website GitHub;
2. Vào tab Settings (Cài đặt) của kho lưu trữ;
3. Chọn mục Pages ở menu;
4. Trong phần “Build and deployment”, tại mục “Source”, chọn là “Deploy from a branch”;
5. Trong mục “Branch”, chọn nhánh chứa code website của bạn (thường là main), giữ nguyên thư mục / (root) và nhấn Save.

Sau khi nhấn Save, GitHub sẽ mất khoảng 1-2 phút để build và triển khai website. Khi hoàn tất, một đường dẫn công khai (ví dụ: [https://\[tên-user\].github.io/Imanu-Lab/](https://[tên-user].github.io/Imanu-Lab/)) sẽ xuất hiện ngay tại màn hình này.



Hình 4.15: Cấu hình GitHub Pages để triển khai từ nhánh 'main'

4.3.3. Kiểm thử và hoàn thiện

Bước cuối cùng là truy cập vào đường dẫn URL công khai mà GitHub Pages vừa cung cấp để kiểm thử website trên môi trường Internet thực tế.

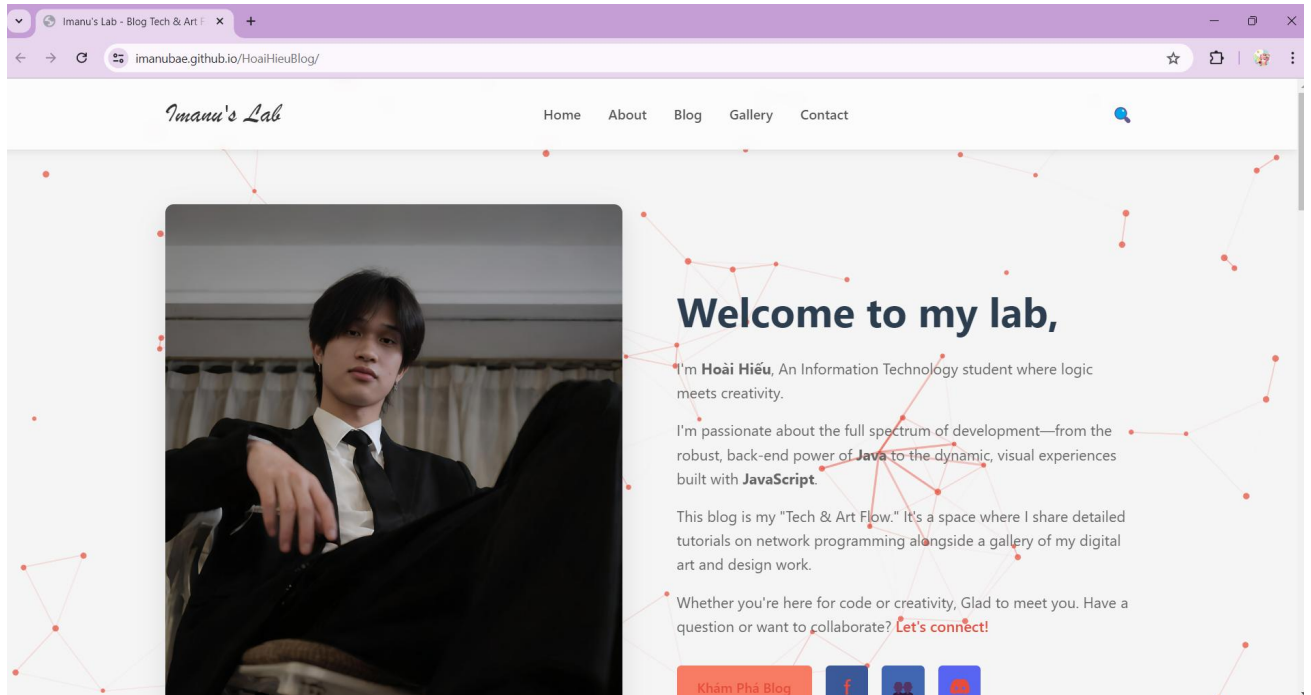
Các mục cần kiểm tra:

1. **Tải tài nguyên:** Đảm bảo tất cả hình ảnh (.jpg, .png), tệp CSS và JavaScript (.css, .js) đã được tải thành công (Không có lỗi 404 trong Console của trình duyệt).
Điều hướng: Thanh điều hướng (Navbar) phải cố định và cho phép chuyển trang.
2. **Đường dẫn:** Kiểm tra các liên kết giữa 3 trang (Home, Blog, Gallery) hoạt động chính xác.
3. **Chức năng:** Kiểm tra lại toàn bộ các chức năng JavaScript (hiệu ứng p5.js, animation cuộn, lọc bài viết, tìm kiếm, và hiển thị modal) để đảm bảo chúng hoạt động mượt mà như trên môi trường local.
4. **Responsive:** Mở website trên nhiều trình duyệt và kích thước màn hình khác nhau (đặc biệt là điện thoại) để đảm bảo giao diện không bị vỡ.
5. Sau khi kiểm thử, website “Imanu’s Lab” đã hoạt động ổn định, nhanh chóng và có thể truy cập được từ bất kỳ đâu có kết nối Internet.

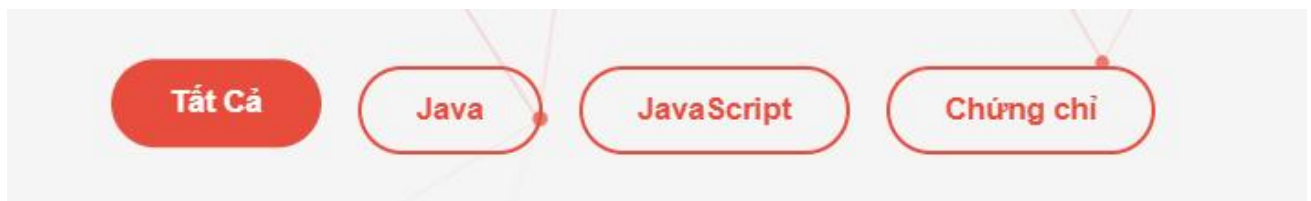
CHƯƠNG 5. KẾT QUẢ VÀ ĐÁNH GIÁ

5.1. TRÌNH BÀY KẾT QUẢ

Sau quá trình thiết kế và triển khai, website “Imanu’s Lab” đã hoàn thiện và hoạt động đúng với các mục tiêu đề ra.



Hình 5.1: Giao diện trang chủ hoàn thiện với hiệu ứng particle



Hình 5.2: Chức năng lọc bài viết trang Blog



Environmental Portrait

Genre: Environmental Portrait.

Techniques: Natural, soft light (golden hour or shade). Shallow depth of field (bokeh) from a wide aperture.

Composition: Subject on left third-line, with leading lines from the road and fence.

Hình 5.3: Chức năng xem chi tiết Gallery (Modal)

5.2. ĐÁNH GIÁ

5.2.1. Ưu điểm

- **Giao diện:** Website có thiết kế hiện đại, chuyên nghiệp, thu hút người xem với hiệu ứng p5.js độc đáo, thể hiện rõ chủ đề “Tech & Art.”
- **Hiệu suất:** Tốc độ tải trang rất nhanh do là website tĩnh.
- **Tổ chức code:** Mã nguồn được tổ chức tốt, chia thành các tệp JavaScript riêng biệt cho từng chức năng, dễ dàng bảo trì và nâng cấp.
- **Chức năng:** Các tính năng cốt lõi (lọc, tìm kiếm, xem modal) hoạt động mượt mà và chính xác.
- **Chi phí:** Triển khai miễn phí và dễ dàng qua GitHub Pages.

5.2.2. Hạn chế

- **Quản lý nội dung:** Việc thêm/sửa/xóa bài viết phải thực hiện thủ công bằng cách sửa đổi các mảng JavaScript. Điều này bất tiện và đòi hỏi người quản trị phải biết code.
- **Form liên hệ:** Form liên hệ ở trang chủ chỉ là giao diện (UI), chưa thể gửi email thực tế do thiếu Back-end xử lý.
- **Tài nguyên:** Hiệu ứng p5.js (dù đã tối ưu) vẫn có thể tiêu tốn tài nguyên (CPU/RAM) trên các thiết bị di động cấu hình quá thấp.

5.3. Kết luận và Hướng phát triển

Kết luận: Đồ án đã hoàn thành mục tiêu đề ra là xây dựng một website portfolio cá nhân bằng công nghệ Front-end. Qua quá trình thực hiện, em đã củng cố vững chắc kiến thức về HTML, CSS, và đặc biệt là JavaScript (xử lý DOM, sự kiện, ES6+). Em cũng đã học hỏi thêm được công nghệ mới (p5.js) và quy trình làm việc chuyên nghiệp (sử dụng Git, GitHub, và deploy sản phẩm).

Hướng phát triển:

- **Xây dựng Back-end:** Sử dụng Node.js (Express) và Cơ sở dữ liệu (MongoDB hoặc MySQL) để chuyển đổi website từ tĩnh sang động.
- **Phát triển API:** Xây dựng các API RESTful để quản lý blogPosts và galleryItems thay vì lưu trong file .js.
- **Xây dựng CMS:** Tạo một trang quản trị (Admin) cho phép thêm/sửa/xóa bài viết thông qua giao diện mà không cần can thiệp vào code.
- **Hoàn thiện Form:** Tích hợp Back-end để form liên hệ có thể gửi email thực sự.

TÀI LIỆU THAM KHẢO

1. MDN Web Docs. (2025). *JavaScript Guide*. <https://developer.mozilla.org/>
2. W3Schools. (2025). *CSS Tutorial*. <https://www.w3schools.com/css/>
3. p5.js Foundation. (2025). *p5.js Reference*. <https://p5js.org/reference/>
4. Hải Dương.(2022).*Duong's Story*. <https://duongstory.com/>

PHỤ LỤC



Statement of Achievement

Nguyen Pham Hoai Hieu

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: Oct 20, 2025

Lynn Bloomer

Lynn Bloomer
Director, Cisco Networking Academy

Phụ lục A: Chứng chỉ JavaScript Essentials 1

Statement of Achievement

Hieu Nguyen

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.*



Issued on: Oct 16, 2025



Scan to Verify

Lynn Bloomer

Lynn Bloomer
Director, Cisco Networking Academy

Phụ lục B: Chứng chỉ JavaScript Essentials 2

Certificate of Course Completion

Nguyễn Phạm Hoài Hiếu

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.*



Scan to Verify

Issued on: Oct 02, 2025

Lynn Bloomer

Lynn Bloomer
Director, Cisco Networking Academy

Phụ lục C: Chứng chỉ Networking Basics