

# Tài liệu giải thích code - Portfolio Website

---

## Mục lục

1. [Giới thiệu](#)
2. [Cấu trúc HTML](#)
3. [Định dạng CSS](#)
4. [Chức năng JavaScript](#)

## 1. Giới thiệu

Website portfolio này được phát triển để giới thiệu thông tin cá nhân, kinh nghiệm làm việc, học tập, các dự án và bài viết. Website có thiết kế thích ứng (responsive) và hỗ trợ chế độ sáng/tối (light/dark mode). Tài liệu này cung cấp giải thích chi tiết về các thành phần code của website.

## 2. Cấu trúc HTML

### 2.1. Cấu trúc cơ bản

File `index.html` là trang chính của website, có cấu trúc tiêu chuẩn HTML5 với các phần:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Thẻ meta và các file liên kết -->
</head>
<body>
  <div class="container">
    <!-- Nội dung trang web -->
  </div>
</body>
</html>
```

### 2.2. Phần Head

Phần `<head>` chứa:

- Thẻ meta với thông tin về bộ ký tự và viewport
- Tiêu đề trang web
- Liên kết đến file CSS chính và font chữ từ Google Fonts
- Liên kết đến thư viện Font Awesome để sử dụng các biểu tượng
- Script ngăn chặn hiện tượng nhấp nháy khi chuyển chế độ sáng/tối

### 2.3. Phần Navigation

```

<nav>
  <ul class="nav-links">
    <li><a href="index.html" class="active">home</a></li>
    <li><a href="projects.html">projects</a></li>
    <li><a href="blog.html">blog</a></li>
    <li><a href="contact.html">contact</a></li>
  </ul>
  <div class="theme-toggles">
    <button id="theme-toggle"><i class="fa-solid fa-moon"></i>
  </button>
  </div>
</nav>

```

Phần này tạo ra thanh điều hướng với các liên kết đến các trang chính và nút chuyển đổi chế độ sáng/tối.

## 2.4. Phần Hero

```

<section class="hero">
  <div class="hero-content">
    <h1>hi viet here 🙋</h1>
    <p class="bio">19-year-old software developer from Viet Nam 🇻🇳 </p>
    <p class="bio-detail">I like to develop full-stack, self-hosting
    homelabs and get coding advice from my !!!</p>

    <div class="social-links">
      <a href="resume.pdf" class="btn resume-btn">Resume <i
class="fa-regular fa-file-lines"></i></a>
      <a href="https://linkedin.com/in/vietds" class="social-icon">
<i class="fa-brands fa-linkedin-in"></i></a>
      <a href="https://github.com/VietDSK6" class="social-icon"><i
class="fa-brands fa-github"></i></a>
      <a href="mailto:vietdsptit@gmail.com" class="social-icon"><i
class="fa-regular fa-envelope"></i></a>
    </div>
  </div>
  <div class="profile-image">
    
  </div>
</section>

```

Phần Hero hiển thị thông tin giới thiệu chính, bao gồm tên, mô tả ngắn, liên kết đến CV và các mạng xã hội, cùng với hình ảnh đại diện.

## 2.5. Phần Tabs (Kinh nghiệm và Học vấn)

```

<section class="tabs-section">
  <div class="tabs">
    <button class="tab active" data-tab="work">Work</button>

```

```

        <button class="tab" data-tab="education">Education</button>
    </div>

    <div class="tab-content active" id="work-content">
        <!-- Nội dung về kinh nghiệm làm việc -->
    </div>

    <div class="tab-content" id="education-content">
        <!-- Nội dung về quá trình học tập -->
    </div>
</section>

```

Phần này sử dụng hệ thống tab để chuyển đổi giữa thông tin về kinh nghiệm làm việc và quá trình học tập.

## 2.6. Phần Dự án nổi bật

```

<section class="featured-projects">
    <div class="section-header">
        <h2>featured projects</h2>
        <a href="projects.html" class="view-more">view more <i class="fa-solid fa-arrow-right"></i></a>
    </div>

    <div class="projects-grid">
        <!-- Các thẻ project-card -->
    </div>
</section>

```

Hiển thị các dự án nổi bật dưới dạng lưới với hình ảnh, tên dự án, mô tả và các công nghệ được sử dụng.

## 2.7. Phần Bài viết gần đây

```

<section class="recent-posts">
    <div class="section-header">
        <h2>recent posts</h2>
        <a href="blog.html" class="view-more">view more <i class="fa-solid fa-arrow-right"></i></a>
    </div>

    <div class="posts-list">
        <!-- Các thẻ post-item -->
    </div>
</section>

```

Hiển thị các bài viết blog gần đây với tiêu đề, mô tả ngắn và ngày đăng.

## 2.8. Phần Footer

```
<footer>
  <div class="copyright">© 2025 vietdosy.me</div>
  <div class="footer-social">
    <a href="https://linkedin.com/in/vietds" class="social-icon"><i
class="fa-brands fa-linkedin-in"></i></a>
    <a href="https://github.com/VietDSK6" class="social-icon"><i
class="fa-brands fa-github"></i></a>
    <a href="mailto:vietdsptit@gmail.com" class="social-icon"><i
class="fa-regular fa-envelope"></i></a>
  </div>
</footer>
```

Hiển thị thông tin bản quyền và các liên kết mạng xã hội ở cuối trang.

## 3. Định dạng CSS

### 3.1. Biến CSS

File `styles.css` bắt đầu với việc định nghĩa các biến CSS trong `:root` để dễ dàng quản lý và thay đổi:

```
:root {
  /* Colors */
  --bg-color: #030713;
  --text-color: #ffffff;
  --text-secondary: #a1a1aa;
  --border-color: #27272a;
  --card-bg: #18181b;
  --accent-color: #3b82f6;
  --hover-color: #2563eb;
  --tag-bg: #27272a;
  --tag-text: #ffffff;

  /* Spacing */
  --spacing-xs: 5px;
  --spacing-sm: 10px;
  --spacing-md: 20px;
  --spacing-lg: 30px;
  --spacing-xl: 60px;

  /* Border Radius */
  --radius-sm: 4px;
  --radius-md: 8px;
  --radius-lg: 12px;
}
```

Các biến này giúp duy trì tính nhất quán trong toàn bộ trang web và giúp việc thay đổi chế độ sáng/tối dễ dàng hơn.

### 3.2. Reset và Cài đặt cơ bản

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: 'Inter', -apple-system, BlinkMacSystemFont, 'Segoe UI',
  Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
  background-color: var(--bg-color);
  color: var(--text-color);
  line-height: 1.6;
}

.container {
  max-width: 800px;
  margin: 0 auto;
  padding: 0 20px;
}
```

Phần này thiết lập reset CSS cơ bản, cài đặt font chữ, màu nền, màu văn bản và container chính có chiều rộng giới hạn.

### 3.3. Định dạng Navigation

```
nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 30px 0px;
  position: sticky;
  top: 0;
  z-index: 1000;
  background-color: var(--bg-color);
  backdrop-filter: blur(5px);
}

.nav-links {
  display: flex;
  list-style: none;
  gap: 30px;
}

.nav-links a {
  color: var(--text-secondary);
  text-decoration: none;
  font-size: 14px;
  transition: color 0.3s;
}
```

```
.nav-links a:hover, .nav-links a.active {  
  color: var(--text-color);  
}
```

Thanh điều hướng sử dụng flexbox để sắp xếp các liên kết và nút chuyển đổi chế độ sáng/tối. Navigation được cài đặt sticky để nó luôn hiển thị ở trên cùng khi cuộn trang.

### 3.4. Định dạng Hero Section

```
.hero {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 80px 0 60px;  
  gap: 40px;  
}  
  
.hero-content {  
  flex: 1;  
}  
  
h1 {  
  font-size: 3rem;  
  font-weight: 700;  
  margin-bottom: 20px;  
}  
  
.bio, .bio-detail {  
  font-size: 0.9rem;  
  color: var(--text-secondary);  
  margin-bottom: 10px;  
}  
  
.profile-image {  
  flex: 0 0 200px;  
}  
  
.profile-image img {  
  width: 75%;  
  height: auto;  
  border-radius: 12px;  
  object-fit: cover;  
}
```

Phần Hero sử dụng flexbox để tạo layout 2 cột: một cột cho nội dung và một cột cho hình ảnh. Kích thước font chữ lớn được áp dụng cho tiêu đề.

### 3.5. Định dạng Tabs và Timeline

```
.tabs {
  display: flex;
  margin-bottom: 30px;
  background-color: var(--card-bg);
  border-radius: 8px;
  overflow: hidden;
}

.tab {
  padding: 12px 30px;
  background-color: transparent;
  color: var(--text-secondary);
  border: none;
  cursor: pointer;
  font-size: 14px;
  font-weight: 500;
  transition: all 0.3s ease;
  width: 100%;
  text-align: center;
}

.tab.active {
  color: var(--text-color);
  background-color: var(--border-color);
}

.tab-content {
  display: none;
  position: relative;
  border: 1px solid var(--border-color);
  border-radius: 12px;
  padding: 20px;
  margin-bottom: 20px;
}

.tab-content.active {
  display: block;
}

.tab-content:before {
  content: "";
  position: absolute;
  top: 25px;
  bottom: 25px;
  left: 25px;
  width: 1px;
  background-color: var(--border-color);
  z-index: 1;
}
```

Phần tab sử dụng lớp `.active` để hiển thị tab đang được chọn. CSS sử dụng pseudo-element `:before` để tạo đường timeline dọc.

### 3.6. Định dạng Projects Grid

```
.projects-grid {  
  display: grid;  
  grid-template-columns: repeat(2, 1fr);  
  gap: 20px;  
}  
  
.project-card {  
  background-color: transparent;  
  border: 1px solid var(--border-color);  
  border-radius: 12px;  
  overflow: hidden;  
}  
  
.project-image {  
  width: 100%;  
  height: 250px;  
  overflow: hidden;  
}  
  
.project-image img {  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
}
```

Phần hiển thị dự án sử dụng CSS Grid để tạo lưới 2 cột. Mỗi thẻ dự án có viền và góc bo tròn.

### 3.7. Responsive Design

```
@media (max-width: 768px) {  
  .hero {  
    flex-direction: column-reverse;  
    text-align: center;  
    padding: 40px 0;  
  }  
  
  .social-links {  
    justify-content: center;  
  }  
  
  .profile-image {  
    flex: 0 0 250px;  
  }  
  
  .projects-grid {  
    grid-template-columns: 1fr;  
  }  
}
```



```
.post-item {
  flex-direction: column;
  gap: 10px;
}

.post-date {
  text-align: left;
}

footer {
  flex-direction: column;
  gap: 20px;
}

/* Timeline responsive adjustments */
.tabs {
  justify-content: center;
}

.tab-content:before {
  left: 20px;
}

.experience-logo {
  flex: 0 0 40px;
  height: 40px;
}
}
```

Media query được sử dụng để điều chỉnh layout khi xem trên thiết bị có màn hình nhỏ hơn 768px. Các thay đổi chính bao gồm chuyển layout từ ngang sang dọc và điều chỉnh kích thước một số phần tử.

## 4. Chức năng JavaScript

### 4.1. Tổng quan

File `main.js` chứa các chức năng JavaScript để xử lý:

- Chuyển đổi chế độ sáng/tối
- Chuyển đổi tabs
- Xử lý form liên hệ

### 4.2. Khởi tạo nhanh chế độ giao diện

```
// Apply theme immediately to prevent flash
(function() {
  // Get theme from localStorage or default to user preference
  const savedTheme = localStorage.getItem('theme');
  const prefersDark = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches;
  // Use saved theme if exists, otherwise use system preference
```

```
const isDarkMode = savedTheme ? savedTheme === 'dark' : prefersDark;

// Apply theme before page renders
applyThemeEarly(isDarkMode);
})();
```

Đoạn code này được thực thi ngay lập tức (IIFE - Immediately Invoked Function Expression) để áp dụng chế độ giao diện trước khi trang web được render hoàn toàn, tránh hiện tượng nhấp nháy khi tải trang.

#### 4.3. Hàm áp dụng chế độ giao diện sớm

```
// Apply theme immediately without waiting for DOM elements
function applyThemeEarly(isDarkMode) {
  if (isDarkMode) {
    // Dark theme colors
    document.documentElement.style.setProperty('--bg-color',
'#0a0a0f');
    document.documentElement.style.setProperty('--card-bg',
'#18181b');
    document.documentElement.style.setProperty('--text-color',
'ffffff');
    document.documentElement.style.setProperty('--text-secondary',
'#a1a1aa');
    document.documentElement.style.setProperty('--border-color',
'#27272a');
    document.documentElement.style.setProperty('--tag-bg', '#27272a');
    document.documentElement.style.setProperty('--tag-text',
'ffffff');
  } else {
    // Light theme colors
    document.documentElement.style.setProperty('--bg-color',
'#f8fafc');
    document.documentElement.style.setProperty('--card-bg',
'ffffff');
    document.documentElement.style.setProperty('--text-color',
'#0f172a');
    document.documentElement.style.setProperty('--text-secondary',
'#64748b');
    document.documentElement.style.setProperty('--border-color',
'#e2e8f0');
    document.documentElement.style.setProperty('--tag-bg', '#e2e8f0');
    document.documentElement.style.setProperty('--tag-text',
'#0f172a');
  }

  // Store the current mode
  localStorage.setItem('theme', isDarkMode ? 'dark' : 'light');
}
```

Hàm này thay đổi các giá trị của biến CSS để áp dụng chế độ sáng hoặc tối và lưu trạng thái hiện tại vào localStorage.

#### 4.4. Khởi tạo các thành phần sau khi tải trang

```
// Wait for DOM to be fully loaded before executing code
document.addEventListener('DOMContentLoaded', function() {
  // Initialize all components
  initThemeToggle();
  initTabSwitching();
  initContactForm();
});
```

Sự kiện `DOMContentLoaded` được sử dụng để đảm bảo rằng DOM đã được tải hoàn toàn trước khi thực thi mã JavaScript. Sau đó, ba hàm khởi tạo được gọi.

#### 4.5. Khởi tạo nút chuyển đổi chế độ giao diện

```
function initThemeToggle() {
  const themeToggle = document.getElementById('theme-toggle');
  if (!themeToggle) return;
  // Retrieve theme mode from localStorage or default to dark mode
  let isDarkMode = localStorage.getItem('theme') === 'dark';

  // Apply the saved theme mode on page load
  applyTheme(isDarkMode);

  themeToggle.addEventListener('click', () => {
    isDarkMode = !isDarkMode; // Toggle the theme state
    applyTheme(isDarkMode);

    // Save the current theme mode to localStorage
    localStorage.setItem('theme', isDarkMode ? 'dark' : 'light');
  });
}
```

Hàm này kết nối sự kiện click vào nút chuyển đổi chế độ giao diện, đảo ngược trạng thái chế độ hiện tại, áp dụng chế độ mới và lưu trạng thái vào localStorage.

#### 4.6. Hàm áp dụng chế độ giao diện

```
function applyTheme(isDarkMode) {
  if (isDarkMode) {
    // Switch to dark mode
    document.documentElement.style.setProperty('--bg-color',
    '#0a0a0f');
    document.documentElement.style.setProperty('--card-bg',
```

```

'#18181b');
    document.documentElement.style.setProperty('--text-color',
'#ffffff');
    document.documentElement.style.setProperty('--text-secondary',
'#a1a1aa');
    document.documentElement.style.setProperty('--border-color',
'#27272a');
    document.documentElement.style.setProperty('--tag-bg', '#27272a');
    document.documentElement.style.setProperty('--tag-text',
'#ffffff');

    // Change icon to moon
    const themeToggle = document.getElementById('theme-toggle');
    if (themeToggle) themeToggle.innerHTML = '<i class="fa-solid fa-
moon"></i>';
  } else {
    // Switch to light mode
    document.documentElement.style.setProperty('--bg-color',
'#f8fafc');
    document.documentElement.style.setProperty('--card-bg',
'#ffffff');
    document.documentElement.style.setProperty('--text-color',
'#0f172a');
    document.documentElement.style.setProperty('--text-secondary',
'#64748b');
    document.documentElement.style.setProperty('--border-color',
'#e2e8f0');
    document.documentElement.style.setProperty('--tag-bg', '#e2e8f0');
    document.documentElement.style.setProperty('--tag-text',
'#0f172a');

    // Change icon to sun
    const themeToggle = document.getElementById('theme-toggle');
    if (themeToggle) themeToggle.innerHTML = '<i class="fa-solid fa-
sun"></i>';
  }
}

```

Tương tự như `applyThemeEarly`, hàm này thay đổi các giá trị của biến CSS để áp dụng chế độ sáng hoặc tối, nhưng còn thay đổi biểu tượng nút chuyển đổi.

#### 4.7. Khởi tạo chuyển đổi tabs

```

function initTabSwitching() {
  const tabs = document.querySelectorAll('.tab');
  const tabContents = document.querySelectorAll('.tab-content');

  if (tabs.length === 0 || tabContents.length === 0) return;

  tabs.forEach(tab => {
    tab.addEventListener('click', () => {

```

```

        const tabId = tab.getAttribute('data-tab');

        // Remove active class from all tabs and contents
        tabs.forEach(t => t.classList.remove('active'));
        tabContents.forEach(content =>
content.classList.remove('active'));

        // Add active class to clicked tab and corresponding content
        tab.classList.add('active');
        document.getElementById(`${tabId}-
content`).classList.add('active');
    });
});
}

```

Hàm này xử lý chuyển đổi giữa các tab. Khi một tab được nhấp vào, lớp `active` được xóa khỏi tất cả các tab và nội dung, sau đó được thêm vào tab đã nhấp và nội dung tương ứng.

#### 4.8. Khởi tạo form liên hệ

```

function initContactForm() {
    const contactForm = document.getElementById('contact-form');

    if (!contactForm) return;

    contactForm.addEventListener('submit', function(e) {
        e.preventDefault();

        // Get form values
        const name = document.getElementById('name').value;
        const email = document.getElementById('email').value;
        const message = document.getElementById('message').value;

        // Show loading state
        const submitBtn = contactForm.querySelector('.minimal-submit-
btn');
        const originalBtnText = submitBtn.innerHTML;
        submitBtn.innerHTML = '<i class="fa-solid fa-spinner fa-spin"></i>
Sending...';
        submitBtn.disabled = true;

        // Simulate API call with timeout
        setTimeout(function() {
            // Reset the form
            contactForm.reset();

            // Show success message
            const formContainer = contactForm.parentElement;
            const successMessage = document.createElement('div');
            successMessage.className = 'success-message';
            successMessage.innerHTML = `

```

```
        <i class="fa-solid fa-check-circle"></i>
        <h3>Message Sent!</h3>
        <p>Thanks for reaching out, ${name}! I'll get back to you
as soon as possible.</p>
    `;

    // Replace form with success message
    formContainer.replaceChild(successMessage, contactForm);

    // Restore form after 5 seconds (optional)
    setTimeout(function() {
        formContainer.replaceChild(contactForm, successMessage);
        submitBtn.innerHTML = originalBtnText;
        submitBtn.disabled = false;
    }, 5000);

    }, 1500);
});
}
```

Hàm này xử lý sự kiện khi form liên hệ được gửi đi. Nó ngăn chặn hành vi gửi mặc định, hiển thị trạng thái đang tải, và sau một khoảng thời gian (1,5 giây), hiển thị thông báo thành công và khôi phục form sau 5 giây.