**Bước 4: Làm theo hướng dẫn từng bước**

1. **Tạo cấu trúc thư mục**
   * Tạo một thư mục mới cho dự án
   * Tạo các file: index.html, styles.css, và script.js
2. **Copy code vào các file**
   * Copy các đoạn code tôi đã cung cấp vào từng file tương ứng
3. **Hiểu về cấu trúc HTML5**
   * Các thẻ semantic như <header>, <nav>, <main>, <section>, <footer> giúp đặt tên và sắp xếp nội dung một cách có ý nghĩa
   * Các thuộc tính như id và class giúp định danh các phần tử để tạo kiểu và tương tác
   * Cấu trúc form với các thẻ <form>, <label>, <input>, <textarea> để thu thập thông tin từ người dùng
4. **Hiểu về CSS3**
   * **CSS Variables**: Sử dụng --primary-color để tạo biến màu chính
   * **Flexbox**: Sử dụng display: flex để tạo layout linh hoạt
   * **Grid**: Sử dụng display: grid để tạo layout lưới
   * **Responsive Design**: Sử dụng media queries @media screen and (max-width: 768px) để tạo giao diện thích ứng với các thiết bị khác nhau
   * **Animations**: Sử dụng @keyframes và animation để tạo hiệu ứng chuyển động
   * **Transitions**: Sử dụng transition để tạo hiệu ứng chuyển đổi mượt mà
   * **Transform**: Sử dụng transform để thay đổi vị trí, kích thước của phần tử
   * **Box Shadow**: Sử dụng box-shadow để tạo bóng đổ
   * **Border Radius**: Sử dụng border-radius để bo tròn góc
5. **Hiểu về JavaScript**
   * **DOM Manipulation**: Truy cập và thay đổi các phần tử HTML
   * **Event Handling**: Xử lý sự kiện như click, scroll
   * **Form Validation**: Kiểm tra dữ liệu nhập vào form
   * **Animation Control**: Điều khiển các hiệu ứng chuyển động
   * **Intersection Observer**: Theo dõi khi một phần tử xuất hiện trong viewport
   * **Smooth Scrolling**: Cuộn mượt đến các phần của trang
6. **Thử nghiệm và điều chỉnh**
   * Mở file index.html trong trình duyệt để xem kết quả
   * Điều chỉnh CSS và JavaScript để hiểu cách chúng hoạt động
   * Thay đổi các giá trị CSS để thấy sự thay đổi trong thiết kế

**Điểm quan trọng để lưu ý:**

1. **HTML5 Semantic Tags**:
   * <header>, <nav>, <main>, <section>, <footer>: Giúp tạo cấu trúc trang web có ý nghĩa
2. **CSS3 Features**:
   * **Variables**: --primary-color
   * **Flexbox**: display: flex
   * **Grid**: display: grid
   * **Media Queries**: @media screen and (max-width: 768px)
   * **Animations**: @keyframes fillBar
   * **Transitions**: transition: background-color 0.3s ease
   * **Transform**: transform: translateY(-10px)
3. **JavaScript Concepts**:
   * **DOM Selection**: document.querySelector()
   * **Event Listeners**: addEventListener()
   * **Arrow Functions**: const navSlide = () => {}
   * **Form Validation**: Kiểm tra dữ liệu nhập vào
   * **IntersectionObserver**: Theo dõi phần tử xuất hiện trong viewport

**Bài tập mở rộng để thực hành thêm:**

1. **Thêm dark mode**:
   * Tạo toggle switch để chuyển đổi giữa light mode và dark mode
   * Sử dụng CSS variables để thay đổi màu sắc
2. **Thêm hiệu ứng parallax**:
   * Tạo hiệu ứng parallax cho hero section
   * Sử dụng JavaScript để thay đổi vị trí các phần tử khi cuộn
3. **Tạo trang chi tiết dự án**:
   * Tạo trang mới để hiển thị chi tiết của mỗi dự án
   * Thực hành sử dụng JavaScript để truyền dữ liệu giữa các trang
4. **Thêm gallery ảnh**:
   * Tạo gallery ảnh với lightbox
   * Sử dụng JavaScript để xử lý sự kiện click và hiển thị ảnh

Bạn có câu hỏi gì về bài tập này không? Tôi có thể giải thích thêm về bất kỳ phần nào bạn cần hiểu rõ hơn.

**GALAXY IMAGE**

**Giải thích code:**

1. **HTML**:
   * Tạo một grid gallery với các hình ảnh thumbnail
   * Mỗi gallery-item có thuộc tính data-src lưu đường dẫn đến ảnh có độ phân giải cao hơn
   * Tạo một lightbox modal với nút đóng, nút next/prev và phần hiển thị caption
2. **CSS**:
   * Sử dụng CSS Grid để tạo layout cho gallery
   * Thêm hiệu ứng hover cho các hình ảnh
   * Tạo modal lightbox với background tối
   * Sử dụng animation fade để tạo hiệu ứng mềm mại khi chuyển ảnh
   * Thêm responsive styling cho các thiết bị nhỏ hơn
3. **JavaScript**:
   * Tạo hàm initGallery() để khởi tạo và xử lý tất cả các chức năng của gallery
   * Xử lý sự kiện click vào hình ảnh để mở lightbox
   * Xử lý việc đóng lightbox khi click vào nút đóng hoặc bên ngoài ảnh
   * Thêm tính năng chuyển qua lại giữa các ảnh với nút next/prev
   * Thêm điều hướng bằng bàn phím (phím mũi tên trái/phải để chuyển ảnh, ESC để đóng)

**Chức năng của Gallery và Lightbox:**

1. **Hiển thị thumbnail**: Hiển thị các hình ảnh nhỏ trong layout grid
2. **Hiệu ứng hover**: Khi di chuột qua ảnh, ảnh sẽ phóng to nhẹ nhàng
3. **Mở lightbox**: Click vào ảnh để mở ảnh lớn hơn trong lightbox
4. **Điều hướng ảnh**: Dùng nút mũi tên hoặc phím mũi tên trái/phải để xem ảnh khác
5. **Đóng lightbox**: Click vào dấu X, nhấn ESC, hoặc click vào khu vực bên ngoài ảnh để đóng
6. **Hiển thị caption**: Hiển thị mô tả ảnh dựa trên thuộc tính alt

**Các bước tích hợp vào dự án Portfolio:**

1. Thêm section gallery vào index.html trước section contact
2. Thêm CSS vào file styles.css
3. Thêm JavaScript vào file script.js và gọi hàm initGallery() trong event DOMContentLoaded

Bạn có thể thay đổi ảnh trong gallery bằng cách thay đổi đường dẫn trong thuộc tính data-src và src của các phần tử gallery-item. Bạn cũng có thể điều chỉnh số lượng ảnh và layout theo nhu cầu của mình.