

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**



**BÀI TẬP LỚN**  
**MÔN: CÔNG NGHỆ WEB VÀ ỨNG DỤNG**

**XÂY DỰNG BACKEND CHO WEBSITE BÁN MÔ HÌNH**

**SINH VIÊN: ĐỖ MINH HOÀNG**  
**MÃ LỚP: 124211**  
**GIÁO VIÊN GIẢNG DẠY: NGUYỄN HỮU ĐÔNG**

**HƯNG YÊN – 2024**

## NHẬN XÉT

**Nhận xét của giảng viên hướng dẫn:**

This image shows a full page of primary-ruled paper. It features approximately 20 horizontal rows, each defined by two parallel dotted lines. The lines are evenly spaced and extend across the entire width of the page, providing a guide for handwriting practice. There is no text or other markings on the paper.

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký và ghi rõ họ tên)

## LỜI CAM ĐOAN

Em xin cam đoan bài tập lớn “Xây dựng BackEnd cho website bán mô hình” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Nguyễn Hữu Đông

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong bài tập lớn và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

*Hưng Yên, ngày ... tháng ... năm 202...*

Sinh viên

.....

## LỜI CẢM ƠN

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Nguyễn Hữu Đông đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các thầy giáo về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI .....	9
1.1 LÝ DO CHỌN ĐỀ TÀI .....	9
1.2 MỤC TIÊU CỦA ĐỀ TÀI.....	10
1.2.1 Mục tiêu tổng quát .....	10
1.2.2 Mục tiêu cụ thể.....	10
1.3 GIỚI HẠN VÀ PHẠM VI CỦA ĐỀ TÀI.....	10
1.3.1 Đối tượng nghiên cứu .....	10
1.3.2 Phạm vi nghiên cứu.....	10
1.4 PHƯƠNG PHÁP TIẾP CẬN.....	10
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	11
2.1 QUY TRÌNH PHÁT TRIỂN PHẦN MỀM .....	11
2.2 THIẾT KẾ GIAO DIỆN WEB VỚI HTML, CSS .....	14
2.2.1 HTML .....	14
2.2.2 CSS.....	17
2.3 LẬP TRÌNH PHÍA FRONT-END .....	19
2.4 LẬP TRÌNH PHÍA BACK-END.....	20
2.4.1 ASP.NET CORE .....	20
2.4.2 Lập trình phía back-end .....	21
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	23
3.1 PHÁT BIỂU BÀI TOÁN.....	23
3.2 ĐẶC TẢ YÊU CẦU PHẦN MỀM .....	23
3.2.1 Các chức năng.....	23
3.2.2 Các yêu cầu phi chức năng.....	24
3.3 THIẾT KẾ CƠ SỞ DỮ LIỆU .....	25
3.4 THIẾT KẾ API PHÍA BE.....	29
3.5 THIẾT KẾ GIAO DIỆN.....	30
3.5.1 Thiết kế giao diện đăng nhập đăng kí .....	30

3.5.2 Thiết kế giao diện trang chủ.....	32
3.5.3 Thiết kế giao diện trang chi tiết sản phẩm .....	34
3.5.4 Thiết kế giao diện trang danh mục sản phẩm .....	35
3.5.5 Thiết kế giao diện bán hàng .....	37
3.5.6 Thiết kế giao diện trang thanh toán.....	38
3.5.7 Thiết kế giao diện thông tin người dùng .....	39
3.5.8 Thiết kế giao diện trang Admin .....	41
CHƯƠNG 4: TRIỂN KHAI WEBSITE .....	44
4.1 TRIỂN KHAI CÁC CHỨC NĂNG CHO PHÂN HỆ NGƯỜI DÙNG .....	44
4.2 KIỂM THỬ VÀ TRIỂN KHAI ỨNG DỤNG.....	45
4.2.1 Kiểm thử.....	45
4.2.2 Đóng gói ứng dụng.....	45
4.2.3 Triển khai ứng dụng .....	46
KẾT LUẬN .....	47
4.3 1. KẾT QUẢ ĐẠT ĐƯỢC .....	47
4.4 2. NHỮNG HẠN CHẾ.....	47
4.5 3. HƯỚNG PHÁT TRIỂN .....	47
TÀI LIỆU THAM KHẢO.....	48

## DANH MỤC HÌNH ẢNH

<i>HÌNH 2.1.1: QUY TRÌNH PHÁT TRIỂN.....</i>	<i>12</i>
<i>HÌNH 2.2.1: CẤU TRÚC CỦA 1 VẤN BẢN HTML.....</i>	<i>16</i>
<i>HÌNH 3.1: CƠ SỞ DỮ LIỆU CHO CỬA HÀNG BÁN MÔ HÌNH .....</i>	<i>25</i>
<i>HÌNH 3.2: API PHÍA BACKEND CHO WEBSITE BÁN MÔ HÌNH.....</i>	<i>29</i>
<i>HÌNH 3.3.1: GIAO DIỆN ĐĂNG NHẬP .....</i>	<i>30</i>
<i>HÌNH 3.3.2: GIAO DIỆN TRANG ĐĂNG KÍ.....</i>	<i>31</i>
<i>HÌNH 3.3.3: GIAO DIỆN HEADER MENU VÀ BANNER.....</i>	<i>32</i>
<i>HÌNH 3.3.4: SẢN PHẨM TRANG CHỦ .....</i>	<i>32</i>
<i>HÌNH 3.3.5: MENU DANH MỤC SẢN PHẨM .....</i>	<i>32</i>
<i>HÌNH 3.3.6: GIAO DIỆN TRANG CHI TIẾT SẢN PHẨM.....</i>	<i>34</i>
<i>HÌNH 3.3.7: GIAO DIỆN TRANG DANH MỤC SẢN PHẨM.....</i>	<i>35</i>
<i>HÌNH 3.3.8: THIẾT KẾ GIAO DIỆN GIỎ HÀNG .....</i>	<i>37</i>
<i>HÌNH 3.3.9: GIAO DIỆN TRANG THANH TOÁN.....</i>	<i>38</i>
<i>HÌNH 3.3.10: GIAO DIỆN THÔNG TIN NGƯỜI DÙNG.....</i>	<i>39</i>
<i>HÌNH 3.3.11: QUẢN LÝ NGƯỜI DÙNG .....</i>	<i>41</i>
<i>HÌNH 3.3.12: QUẢN LÝ SẢN PHẨM.....</i>	<i>41</i>
<i>HÌNH 3.3.13: FORM THÊM SẢN PHẨM .....</i>	<i>42</i>

## **DANH MỤC BẢNG BIỂU**

<i>Bảng 1: Chức năng của phân hệ quản trị nội dung.....</i>	<i>23</i>
<i>Bảng 2: Chức năng của phân hệ người dùng .....</i>	<i>24</i>



## CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1 Lý do chọn đề tài

Chúng ta có thể nói rằng thế kỷ 21 đã và đang chứng kiến sự phát triển mạnh mẽ của ngành Công nghệ thông tin. Công nghệ thông tin (CNTT) đã và đang thay đổi thế giới. Ngày nay, khi mà công nghệ và nhất là internet đang ngày càng phát triển, những dịch vụ, hoạt động mà trước đây chúng ta phải thực hiện, phải giao dịch, trao đổi bằng cách gặp mặt trực tiếp đang được dần thay thế bởi internet, và mua hàng online là một trong số đó.

Việc mua hàng online trong vài năm trở lại đây đang là xu hướng của toàn thế giới, nhất là trong bối cảnh đại dịch Covid-19, khi mà mọi người không thể đến trực tiếp các cửa hàng thì việc mua sắm qua mạng là một cách vô cùng hiệu quả.

Đã có rất nhiều những lĩnh vực, những thị trường sản phẩm đã rất thành công trong việc mua bán qua hình thức trực tuyến. Tiêu biểu trong số đó phải nhắc đến thời trang và các vật dụng nhu yếu phẩm, v.v... Chúng ta có thể thấy rằng với việc mà số lượng người mua hàng online ngày càng tăng thì thị trường thời trang online sẽ càng có nhiều cơ hội để phát triển. Và để đáp ứng được nhu cầu của thị trường mua hàng online nói chung cũng như thời trang online nói riêng, những website thời trang trực tuyến đã được ra đời giống như một cầu nối kết nối người mua hàng với những người bán. Việc tạo ra những trang web thời trang trực tuyến giúp cho người bán tiếp cận được với nhiều khách hàng hơn, điều mà hình thức mua hàng trực tiếp khó có thể làm được.

Một ưu điểm nữa của các trang web thời trang trực tuyến đó chính là người bán có thể kinh doanh mọi lúc, mọi nơi. Họ có thể thực hiện các giao dịch bán hàng từ xa một cách linh hoạt, thuận tiện cho khách hàng của mình.

Nắm bắt được xu hướng trên, em đã tạo nên một trang web về việc bán hàng online, giúp cho người dùng cảm thấy thuận tiện hơn trong việc kinh doanh, đồng thời giúp khách hàng mua sắm một cách tiện lợi, tránh được những bất tiện khi phải mua hàng trực tiếp tại cửa hàng. Vì vậy em quyết định chọn đề tài “Xây dựng BackEnd cho website bán mô hình” làm bài tập lớn.

## **1.2 Mục tiêu của đề tài**

### **1.2.1 Mục tiêu tổng quát**

Xây dựng BackEnd cho website bán mô hình.

### **1.2.2 Mục tiêu cụ thể**

- Xây dựng BackEnd cho Website bán mô hình có giao diện thân thiện với người dùng giúp cho người dùng dễ dàng thao tác tìm kiếm mua sắm sản phẩm. Đồng thời giúp cho nhân viên, người quản lý dễ dàng quản lý.
- Nghiên cứu các công nghệ phát triển website và áp dụng được vào đề tài của mình
- Giải quyết tối ưu hóa quá trình quản lý bán hàng và sản phẩm

## **1.3 Giới hạn và phạm vi của đề tài**

### **1.3.1 Đối tượng nghiên cứu**

- Đối tượng nghiên cứu: Website thương mại.
- Khách thể nghiên cứu: Các nhân viên quản lý và nhân viên bán hàng.

### **1.3.2 Phạm vi nghiên cứu**

- Phạm vi không gian: Tại các website bán hàng.
- Phạm vi thời gian: Kéo dài từ tháng 3 năm 2024 đến tháng 5 năm 2024.

## **1.4 Phương pháp tiếp cận**

- Khảo sát thực trạng các website thương mại.
- Thu thập các yêu cầu từ phía người dùng.
- Phân tích thiết kế website theo yêu cầu người dùng.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Quy trình phát triển phần mềm

Quy trình phát triển phần mềm (Software Process) là một tập hợp các hoạt động có cấu trúc nhằm phát triển hoặc tiến hóa một sản phẩm phần mềm. Trong thực tế, ứng với mỗi một dự án phần mềm cụ thể cũng như một loại sản phẩm phần mềm cụ thể mà các công ty sản xuất phần mềm có thể áp dụng các quy trình sản xuất phần mềm khác nhau. Mặc dù tồn tại nhiều quy trình sản xuất phần mềm khác nhau, những tất cả các quy trình này phải bao gồm bốn hoạt động kỹ nghệ phần mềm cơ bản:

**1. Đặc tả phần mềm (Software Specification):** Hoạt động này được thực hiện bởi các kỹ sư phần mềm làm việc với khách hàng để xác định được các chức năng của phần mềm, các nghiệp vụ và các ràng buộc trên các chức năng của phần mềm, các nghiệp vụ và các ràng buộc trên các chức năng của phần mềm.

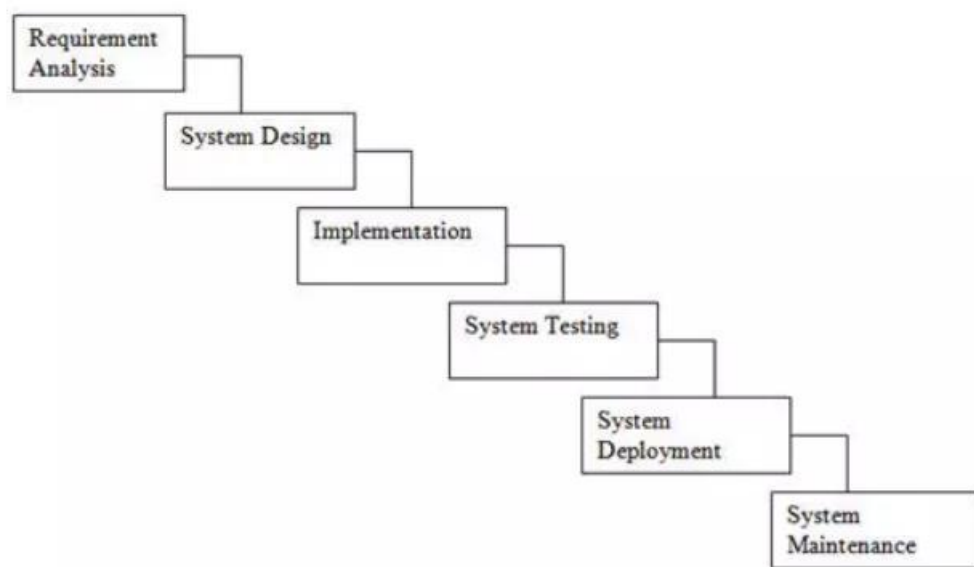
**2. Phát triển phần mềm (Software Development):** Kỹ sư phần mềm thực hiện thiết kế và lập trình phần mềm theo bản đặc tả của phần mềm.

**3. Thử nghiệm phần mềm (Software Evolution):** Khi phần mềm được sửa đổi để đáp ứng sự thay đổi yêu cầu của khách hàng và yêu cầu của thị trường.

Mỗi một hoạt động có thể là những hoạt động phức tạp chúng có thể gồm những hoạt động nhỏ hơn như trong phát triển phần mềm có thể bao gồm: thiết kế, lập trình, kiểm thử. Ngoài ra các quy trình cũng bao gồm các hoạt động khác như quản lý cấu hình phần mềm và lập kế hoạch dự án. Tùy theo từng mô hình phát triển phần mềm mà các hoạt động này được tổ chức thực hiện theo các cách khác nhau. Các quy trình phần mềm thực tế ngày nay là một chuỗi xen kẽ các hoạt động kỹ thuật, cộng tác, và quản lý với mục tiêu chung là đặc tả, thiết kế, triển khai và kiểm thử một hệ thống phần mềm.

Quy trình phát triển phần mềm em sử dụng trong quá trình thực hiện đồ án này là dựa theo mô hình thác nước.

Mô hình thác nước (Waterfall Model) là mô hình quy trình phát triển phần mềm đầu tiên được đề xuất bắt nguồn từ các mô hình quy trình kỹ nghệ được sử dụng trong các kỹ nghệ hệ thống quân sự lớn (Royce 1970). Mô hình thác nước gồm các pha như Hình ... được thực hiện một cách tuần tự như dòng chảy thác nước từ pha này sang pha khác. Đây là mô hình đầu tiên thể hiện quy trình phát triển phần mềm có kế hoạch. Về nguyên tắc, chúng ta cần lập kế hoạch và lên lịch trình tất cả các hoạt động quy trình trước khi phát triển phần mềm.



Hình 2.1.1: Quy trình phát triển

Các giai đoạn của mô hình thác nước phản ánh trực tiếp các hoạt động phát triển phần mềm cơ bản gồm:

#### 4 Định nghĩa và phân tích yêu cầu(Requirements analysis and definition)

Pha này sẽ xác định những “yêu cầu” (“What”) liên quan đến chức năng và phi chức năng mà hệ thống phần mềm cần có. Giai đoạn này cần sự tham gia tích cực của khách hàng và đầu ra là một tài liệu đặc tả yêu cầu phần mềm (Software Requirement Specification – SRS document). Trong đó, tài liệu bao gồm tập hợp các yêu cầu đã được duyệt (reviewed) và nghiệm thu (approved) bởi những người có trách nhiệm đối với dự án (từ phía khách hàng). Tài liệu SRS là đầu vào của tất cả các hoạt động tiếp theo cho đến cuối dự án phần mềm.

## **5 Thiết kế hệ thống và phần mềm (System and software design)**

Pha này là pha phân bố các yêu cầu cho hệ thống phần cứng hoặc phần mềm. Thiết kế hệ thống sẽ thiết lập kiến trúc tổng thể hệ thống. Thiết kế phần mềm sẽ xác định “làm thế nào” (“How”) để hệ thống phần mềm đáp ứng những “yêu cầu” (“What”) mà khách hàng yêu cầu trong tài liệu SRS. Đây chính là cầu nối giữa “đòi hỏi” (“What”) và mã (Code) được thực hiện đáp ứng yêu cầu đó.

## **6 Triển khai cài đặt và kiểm thử đơn vị (Implementation and unit testing)**

Pha này là pha thực hiện “làm thế nào” (“How”) được chỉ ra trong giai đoạn “Thiết kế hệ thống và phần mềm” thành mã nguồn triển khai và thực hiện kiểm thử mức đơn vị để xác minh rằng mỗi đơn vị mã nguồn đáp ứng được đặc tả của nó.

## **7 Kiểm thử tích hợp và kiểm thử hệ thống (Integration and system testing)**

Pha này sẽ tiến hành kiểm thử mã nguồn đã được triển khai. Đầu tiên là các đơn vị chương trình riêng lẻ. các hệ thống con được tích hợp dần và kiểm thử tích hợp. Sau đó là kiểm thử toàn bộ hệ thống. Cuối cùng kiểm thử chấp nhận được thực hiện bởi khách hàng để xác định hệ thống phần mềm có đáp ứng yêu cầu của họ hay không. Sau khi kiểm thử toàn diện hệ thống, phần mềm được chuyển giao cho khách hàng.

## **8 Vận hành và bảo trì (Operation and Maintenance)**

Đây là pha có vòng đời dài nhất. Hệ thống được cài đặt, cấu hình và đưa vào sử dụng thực tế. Pha này sửa chữa những lỗi của phần mềm không được phát hiện trong các giai đoạn trước của vòng đời và phát triển những thay đổi mới được khách hàng yêu cầu.

## 2.2 Thiết kế giao diện web với HTML, CSS

### 2.2.1 HTML

**Ngôn ngữ HTML (HyperText Markup Language - ngôn ngữ siêu văn bản) là một trong các loại ngôn ngữ được sử dụng trong lập trình web.** Khi truy cập một trang web cụ thể là click vào các đường link, bạn sẽ được dẫn tới nhiều trang các nhau, và các trang này được gọi là một tài liệu HTML (tập tin HTML).

Một trang HTML như vậy được cấu thành bởi nhiều phần tử HTML nhỏ và được quy định bằng các thẻ tag. Bạn có thể phân biệt một trang web được viết bằng ngôn ngữ HTML hay PHP thông qua đường link của nó. Ở cuối các trang HTML thường hay có đuôi là .HTML hoặc .HTM.

HTML là ngôn ngữ lập trình web được đánh giá là đơn giản. Mọi trang web, mọi trình duyệt web đều có thể hiển thị tốt ngôn ngữ HTML. Hiện nay, phiên bản mới nhất của HTML là HTML5 với nhiều tính năng tốt và chất lượng hơn so với các phiên bản HTML cũ.

#### **Ưu điểm và nhược điểm của HTML**

- Ưu điểm của HTML
  - + Có nhiều tài nguyên hỗ trợ với cộng đồng người dùng vô cùng lớn
  - + Có thể hoạt động mượt mà trên hầu hết mọi trình duyệt hiện nay
  - + Học HTML khá đơn giản
  - + Các markup sử dụng trong HTML thường ngắn gọn, có độ đồng nhất cao
  - + Sử dụng mã nguồn mở, hoàn toàn miễn phí
  - + HTML là chuẩn web được vận hành bởi W3C
  - + Dễ dàng để tích hợp với các loại ngôn ngữ backend (ví dụ như: PHP, Node.js, ...)

- Nhược điểm của HTML
  - + Chỉ được áp dụng chủ yếu cho web tĩnh. Nếu muốn tạo các tính năng động, lập trình viên phải dùng thêm JavaScript hoặc ngôn ngữ backend của bên thứ 3 (ví dụ như: PHP)
  - + Mỗi trang HTML cần được tạo riêng biệt, ngay cả khi có nhiều yếu tố trùng lặp như header, footer
  - + Khó để kiểm soát cách đọc và hiển thị file HTML của trình duyệt (ví dụ, một số trình duyệt cũ không render được tag mới. Do đó, dù trong HTML document có sử dụng các tag này thì trình duyệt cũng không đọc được)
  - + Một vài trình duyệt còn chậm cập nhật để hỗ trợ tính năng mới của HTML.

### **Cấu trúc một văn bản HTML**

Mỗi trang HTML chứa một bộ các **tag** (cũng được gọi là **elements**). Mỗi thẻ sẽ có những tác dụng nhất định, giúp xây dựng nên một cấu trúc hoàn chỉnh cho Website. Bạn có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc cây thư mục bao gồm section, paragraph, heading, và những khối nội dung khác.

Hầu hết các HTML elements đều có tag mở và tag đóng với cấu trúc như `<tag></tag>`.

Để biết bố cục HTML của một trang web như thế nào, bạn có thể xem code ví dụ của một trang HTML được cấu trúc như thế nào:

```
<!DOCTYPE html>

<html>

  <head>

    <title>Page Title</title>

  </head>

  <body>

    <h1>The Main Heading</h1>

    <h2>A catchy subheading</h2>

    <p>First paragraph</p>

  </body>

</html>
```

Hình 2.2.1: Cấu trúc của 1 văn bản html

Trong đó:

**<!DOCTYPE html>**: khai báo kiểu dữ liệu hiển thị

**<html>** và **</html>**: cặp thẻ bắt buộc, element cấp cao nhất, có nhiệm vụ đóng gói tất cả nội dung của trang HTML

**<head>** và **</head>**: khai báo các thông tin meta của trang web như: tiêu đề trang, charset

**<title>** và **</title>**: cặp thẻ nằm bên trong thẻ <head>, dùng để khai báo tiêu đề của trang



**<body>** và **</body>**: cặp thẻ dùng để đóng gói tất cả các nội dung sẽ hiển thị trên trang

**<h1></h1>**, **<h2></h2>**: định dạng dữ liệu dạng heading. Thông thường có 6 cấp độ heading trong HTML, trải dài từ **<h1>** tới **<h6>**. Trong đó, **<h1>** là cấp độ heading cao nhất và **<h6>** là cấp độ heading thấp nhất.

**<p>** và **</p>**: cặp thẻ chứa các đoạn văn bản của trang web

### 2.2.2 CSS

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng, ... thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc...

CSS được phát triển bởi **W3C** (World Wide Web Consortium) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

#### **Bố cục và cấu trúc một đoạn CSS**

- Bố cục của một đoạn CSS
- Bố cục CSS thường chủ yếu dựa vào hình hộp và mỗi hộp đều chiếm những khoảng trống trên trang của bạn với các thuộc tính như:

- + **Padding:** Gồm không gian xung quanh nội dung (ví dụ: xung quanh đoạn văn bản).
- + **Border:** Là đường viền nằm ngay bên ngoài phần đệm.
- + **Margin:** Là khoảng cách xung quanh bên ngoài của phần tử.
- Cấu trúc của một đoạn CSS
  - + Một đoạn CSS bao gồm các phần như thế này:

```
vùng chọn {      thuộc tính : giá trị;      thuộc tính: giá trị;      ..... }
```

Nghĩa là nó sẽ được khai báo bằng vùng chọn, sau đó các thuộc tính và giá trị sẽ nằm bên trong cặp dấu ngoặc nhọn { }. Mỗi thuộc tính sẽ luôn có một giá trị riêng, giá trị có thể là dạng số, hoặc các tên giá trị trong danh sách có sẵn của CSS. Phần giá trị và thuộc tính phải được cách nhau bằng dấu hai chấm, và mỗi một dòng khai báo thuộc tính sẽ luôn có dấu chấm phẩy ở cuối. Một vùng chọn có thể sử dụng không giới hạn thuộc tính.

### Tại sao sử dụng CSS?

Đây là ba lợi ích chính của CSS:

#### 1. Giải quyết một vấn đề lớn

Trước khi có CSS, các thẻ như phông chữ, màu sắc, kiểu nền, các sắp xếp phần tử, đường viền và kích thước phải được lặp lại trên mọi trang web. Đây là một quá trình rất dài tốn thời gian và công sức. Ví dụ: Nếu bạn đang phát triển một trang web lớn nơi phông chữ và thông tin màu được thêm vào mỗi trang, nó sẽ trở thành một quá trình dài và tốn kém. CSS đã được tạo ra để giải quyết vấn đề này. Đó là một khuyến cáo của W3C.

Nhờ CSS mà source code của trang Web sẽ được tổ chức gọn gàng hơn, trật tự hơn. Nội dung trang web sẽ được tách bạch hơn trong việc định dạng hiển thị. Từ đó, quá trình cập nhập nội dung sẽ dễ dàng hơn và có thể hạn chế tối thiểu làm rối cho mã HTML.

## 2. Tiết kiệm rất nhiều thời gian

Định nghĩa kiểu CSS được lưu trong các tệp CSS bên ngoài vì vậy có thể thay đổi toàn bộ trang web bằng cách thay đổi chỉ một tệp. Sử dụng CSS sẽ giúp bạn không cần thực hiện lặp lại các mô tả cho từng thành phần. Từ đó, bạn có thể tiết kiệm được tối đa thời gian làm việc với nó, làm code ngắn lại giúp kiểm soát dễ dàng hơn các lỗi không đáng có.

CSS tạo ra nhiều style khác nhau nên có thể được áp dụng với nhiều trang web, từ đó giảm tránh việc lặp lại các định dạng của các trang web giống nhau.

## 3. Cung cấp thêm các thuộc tính

CSS cung cấp các thuộc tính chi tiết hơn HTML để định nghĩa giao diện của trang web. CSS giúp người dùng nhiều styles trên một trang web HTML nên khả năng điều chỉnh trang của bạn trở nên vô hạn.

### 2.3 Lập trình phía front-end

Khi truy cập vào một trang web, bạn sẽ thấy giao diện và nội dung website, cũng như các nút bấm, banner, đồ họa, ... Như vậy, frontend có nghĩa là tất cả những gì mà bạn, những người khách truy cập website nhìn thấy trên một website.

Nói tóm lại, frontend là cách gọi quy trình sử dụng các ngôn ngữ lập trình như HTML, CSS, JavaScript để thiết kế và xây dựng giao diện cho các trang web hoặc các ứng dụng web để người dùng có thể xem và tương tác trực tiếp trên đó.

Trong thực tế người truy cập vào các trang web sử dụng rất nhiều loại thiết bị khác nhau với kích thước và độ phân giải khác nhau, chính vì vậy mà lập trình viên frontend phải xem xét hết tất cả các khía cạnh này khi thiết kế trang web. Họ cần phải đảm bảo trang web xuất hiện chính xác trên các trình duyệt khác nhau, hệ điều hành khác nhau và các thiết bị khác nhau.

### Công việc của lập trình viên frontend

Giao diện, tính năng, font chữ, đồ họa, nút đăng nhập thử điện tử và các ảnh động là những ấn tượng đầu tiên của những người khách truy cập về một website.

Tùy vào tính chất mà những website khác nhau sẽ có những tính năng cũng như những cách thức điều hướng khác nhau. Tất cả những gì mà bạn nhìn thấy trên trình duyệt đã được xử lý qua ngôn ngữ HTML, một loại ngôn ngữ sử dụng trong lập trình, trong khi bố cục và màu sắc được thiết lập bởi CSS. Nói chính xác hơn, tất cả những gì bạn nhìn thấy được trên website chính là công việc của một lập trình viên frontend.

Nhiệm vụ của một lập trình viên frontend là phải sắp xếp gọn gàng và đảm bảo cho quá trình tải trang không bị gián đoạn. Đồng thời, đảm bảo các website phải luôn linh hoạt, thay đổi giao diện đáp ứng cho từng thiết bị cụ thể, từ desktop, smartphone cho đến máy tính bảng...

Thông qua những gì mà bạn nhìn thấy trên website thể hiện sự tinh tế của một người lập trình frontend.

## **2.4 Lập trình phía back-end**

### **2.4.1 ASP.NET CORE**

ASP.NET Core là một framework phát triển ứng dụng web mã nguồn mở do Microsoft phát triển. Nó là phiên bản tiếp theo của ASP.NET, được thiết kế để chạy trên nền tảng đa nền tảng, hỗ trợ Windows, macOS và Linux. Dưới đây là một số điểm quan trọng trong việc giới thiệu ASP.NET Core:

ASP.NET Core hỗ trợ chạy trên nhiều hệ điều hành như Windows, macOS và Linux. Điều này mang lại sự linh hoạt cho các nhà phát triển khi triển khai ứng dụng trên các môi trường khác nhau.

ASP.NET Core được tối ưu hóa để đạt hiệu suất cao. Nó có thể xử lý số lượng lớn các yêu cầu mà không ảnh hưởng đến hiệu suất.

ASP.NET Core sử dụng mô hình middleware, cho phép bạn thêm, loại bỏ hoặc thay đổi các thành phần trung tâm một cách dễ dàng. Điều này tăng tính linh hoạt của ứng dụng.

ASP.NET Core tích hợp sẵn hệ thống Dependency Injection, giúp quản lý và tổ chức các thành phần phụ thuộc của ứng dụng một cách hiệu quả.

ASP.NET Core được thiết kế để dễ dàng tích hợp với các dịch vụ đám mây như Azure. Nó cung cấp các tính năng như cân bằng tải và kiểm soát phiên bản để hỗ trợ việc triển khai và quản lý ứng dụng trên môi trường đám mây.

ASP.NET Core là mã nguồn mở, cho phép cộng đồng phát triển đóng góp, kiểm tra mã nguồn, và thậm chí tạo ra các biến thể tùy chỉnh của framework.

ASP.NET Core tích hợp logging, giúp theo dõi và ghi lại thông tin từ ứng dụng một cách dễ dàng.

ASP.NET Core cung cấp các công cụ mạnh mẽ cho việc xây dựng và triển khai các dịch vụ web RESTful.

Bạn có thể sử dụng cả Visual Studio và Visual Studio Code để phát triển ứng dụng ASP.NET Core. Visual Studio cung cấp một số tính năng phong phú hỗ trợ cho việc phát triển ứng dụng web.

ASP.NET Core sử dụng mô hình MVC giúp tách biệt logic dữ liệu, giao diện và quản lý điều hướng.

ASP.NET Core là một lựa chọn phổ biến cho việc phát triển ứng dụng web đa nền tảng và hiệu quả. Nó kết hợp các tính năng mạnh mẽ, linh hoạt, và hiện đại để giúp nhà phát triển xây dựng ứng dụng web chất lượng cao.

#### **2.4.2 Lập trình phía back-end**

Lập trình phía back-end là quá trình xây dựng và duy trì phần của một ứng dụng web hoặc dịch vụ mà người dùng không được trực tiếp. Back-end thường xử lý các nhiệm vụ như xử lý dữ liệu, tương tác với cơ sở dữ liệu, xác thực người dùng, và cung cấp dữ liệu cho phía front-end.

- Công việc của lập trình viên back-end:

##### **1. Xây dựng API (Application Programming Interface):**

- Phát triển và thiết kế các API để tương tác với phía front-end hoặc các ứng dụng khác.
- Sử dụng các phong cách kiến trúc như RESTful để xây dựng các API.

2. Xử lý dữ liệu:

- Xử lý dữ liệu từ nguồn cung cấp dữ liệu, API bên ngoài hoặc các dịch vụ khác.
- Thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên cơ sở dữ liệu.

3. Quản lý cơ sở dữ liệu:

- Thiết kế và triển khai cơ sở dữ liệu.
- Tối ưu hóa câu truy vấn để cải thiện hiệu suất.

4. Authentication và Authorization:

- Xây dựng hệ thống xác thực và ủy quyền để bảo vệ tài nguyên của ứng dụng.
- Sử dụng các phương thức như JWT, OAuth để đảm bảo an toàn thông tin người dùng.

5. Xử lý logic kinh doanh:

- Triển khai logic kinh doanh của ứng dụng, đảm bảo tính logic và chính xác.
- Tích hợp các yêu cầu kinh doanh vào mã nguồn.

6. Tối ưu hóa hiệu suất:

- Theo dõi và tối ưu hóa hiệu suất hệ thống, đảm bảo ứng dụng chạy mượt mà và hiệu quả.
- Điều chỉnh các cấu hình máy chủ và cơ sở dữ liệu để tối ưu hóa hiệu suất.

7. Bảo mật:

- Triển khai các biện pháp bảo mật để ngăn chặn tấn công như SQL injection, ...

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1 Phát biểu bài toán

Xây dựng BackEnd cho website bán là một bài toán quan trọng. Đây là một số phát biểu cơ bản về bài toán này:

- Mục tiêu chính: Xây dựng BackEnd cho website bán hàng phục vụ cho việc mua sắm mô hình online hướng đến nhiều đối tượng người dùng. Website bao gồm những chức năng cơ bản như: xem sản phẩm, tìm kiếm sản phẩm, đặt hàng, quản lý sản phẩm, thống kê, báo cáo, ...
- Tính năng cơ bản:
  - + Xem sản phẩm
  - + Tìm kiếm sản phẩm
  - + Quản lý giỏ hàng
  - + Quản lý sản phẩm
  - + Quản lý khách hàng

### 3.2 Đặc tả yêu cầu phần mềm

#### 3.2.1 Các chức năng

##### a) Chức năng của phân hệ quản trị nội dung

Bảng 1: Chức năng của phân hệ quản trị nội dung

TT	Tên yêu cầu	Mô tả yêu cầu
1	Quản lý người dùng	Là chức năng giúp nhân viên bán hàng quản lý người dùng
2	Quản lý sản phẩm	Là chức năng giúp nhân viên bán hàng quản lý thông tin của sản phẩm

**b) Chức năng của phân hệ người dùng**

*Bảng 2: Chức năng của phân hệ người dùng*

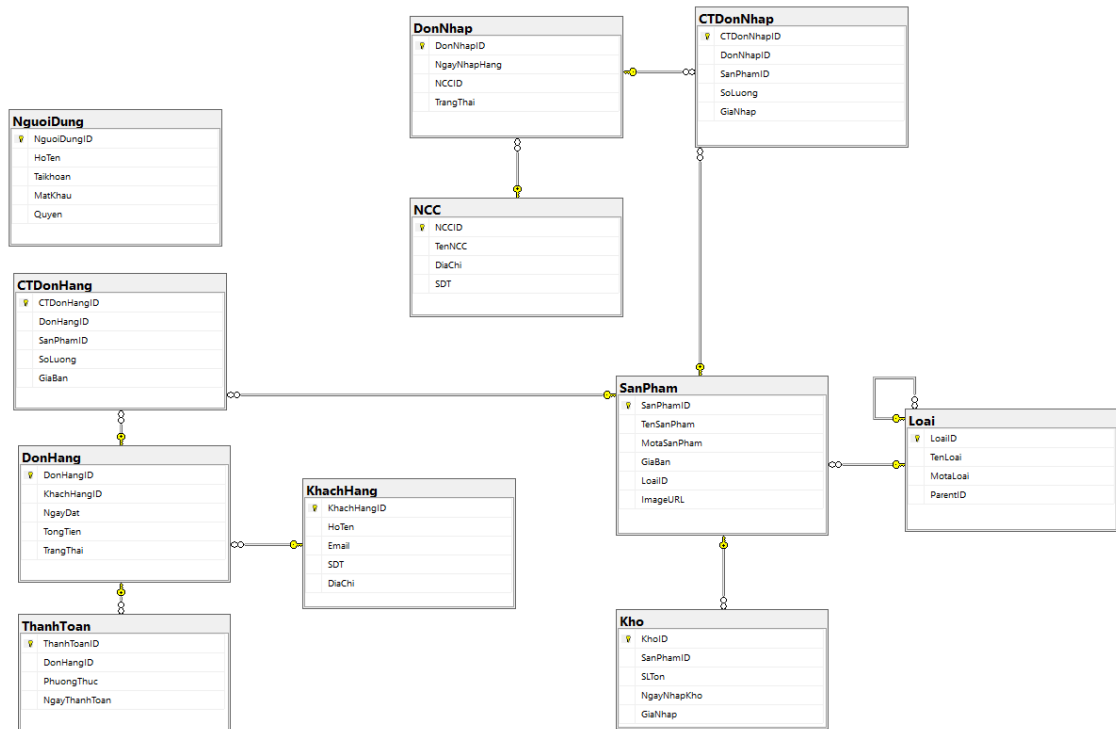
TT	Tên yêu cầu	Mô tả yêu cầu
1	Xem sản phẩm	Là chức năng giúp khách hàng xem được thông tin của sản phẩm
2	Quản lý giỏ hàng	Là chức năng giúp khách hàng quản lý được các sản phẩm giỏ hàng của mình
3	Tìm kiếm sản phẩm	Là chức năng cho phép người dùng tìm kiếm thông tin của sản phẩm

**3.2.2 Các yêu cầu phi chức năng**

1. Yêu cầu về tốc độ việc thực hiện các chức năng khi có nhiều người dùng cùng thao tác trong 1 thời điểm.
2. Giao diện người dùng cần thân thiện với người dùng, dễ dàng thao tác.
3. Thiết kế các chức của website cần phải đầy đủ theo yêu cầu.
4. Tính tương thích: cần tương thích với nhiều loại trình duyệt khác nhau, nhiều loại phiên bản trình duyệt khác nhau và độ phân giải cần thay đổi 1 cách linh hoạt trên các thiết bị khác nhau.
5. Bảo mật thông tin khách hàng, thông tin sản phẩm, ...



### 3.3 Thiết kế cơ sở dữ liệu



Hình 3.1: Cơ sở dữ liệu cho cửa hàng bán mô hình

Cơ sở dữ liệu cho cửa hàng bán mô hình gồm có các bảng, phương thức và chức năng của các phương thức như sau:

- **Bảng Loai**
- **Mục đích:** Lưu thông tin về các loại sản phẩm.
- **Phương thức:**
  - LoaiID: Mã loại sản phẩm (khóa chính, tự tăng).
  - TenLoai: Tên loại sản phẩm (ví dụ: "Anime Figures").
  - MotaLoai: Mô tả về loại sản phẩm.
  - ParentID: Mã loại sản phẩm cha (cho phép tạo cấu trúc phân cấp giữa các loại sản phẩm, ví dụ, các loại con thuộc nhóm "Anime Figures").

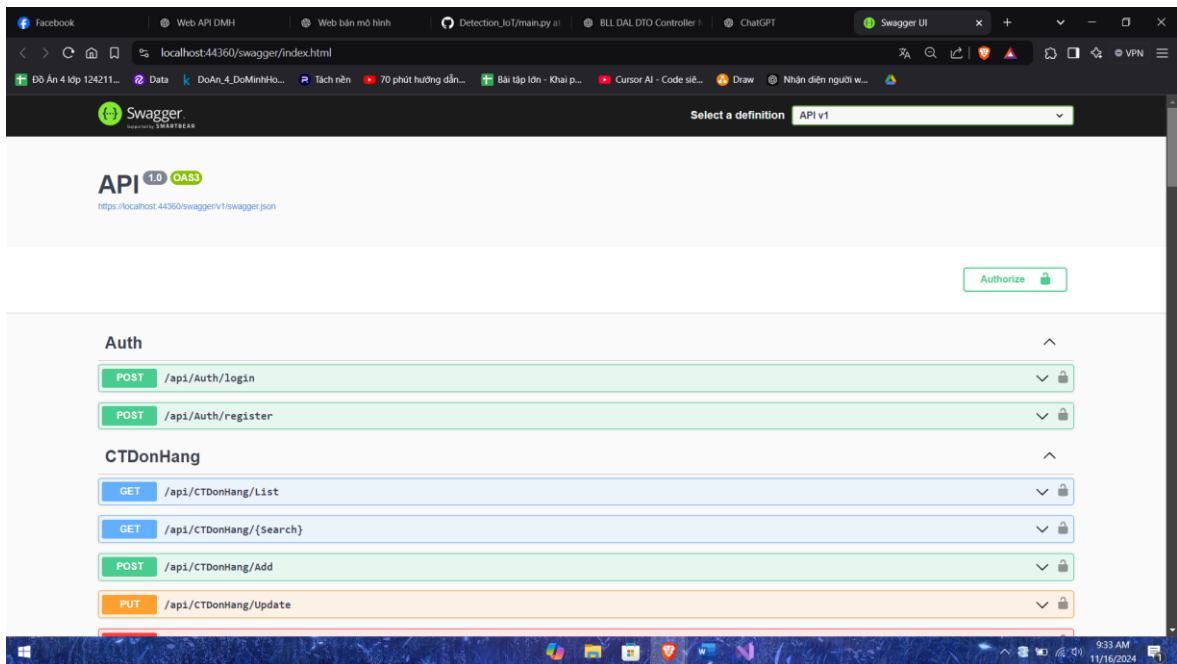
- **Bảng SanPham**
- **Mục đích:** Lưu thông tin chi tiết về từng sản phẩm.
- **Cột:**
  - SanPhamID: Mã sản phẩm (khóa chính, tự tăng).
  - TenSanPham: Tên sản phẩm (ví dụ: "Nendoroid Naruto").
  - MotaSanPham: Mô tả chi tiết về sản phẩm.
  - GiaBan: Giá bán của sản phẩm.
  - LoaiID: Mã loại sản phẩm (khóa ngoại tham chiếu từ bảng Loai).
  - ImageURL: Đường dẫn hình ảnh của sản phẩm.
- **Bảng Kho**
- **Mục đích:** Quản lý số lượng tồn kho của từng sản phẩm.
- **Phương thức:**
  - KhoID: Mã kho (khóa chính, tự tăng).
  - SanPhamID: Mã sản phẩm (khóa ngoại tham chiếu từ bảng SanPham).
  - SLTon: Số lượng tồn kho.
  - NgayNhapKho: Ngày nhập kho sản phẩm.
  - GiaNhap: Giá nhập của sản phẩm.
- **Bảng NCC (Nhà Cung Cấp)**
- **Mục đích:** Quản lý thông tin về nhà cung cấp.
- **Phương thức:**
  - NCCID: Mã nhà cung cấp (khóa chính, tự tăng).
  - TenNCC: Tên nhà cung cấp.
  - DiaChi: Địa chỉ nhà cung cấp.
  - SDT: Số điện thoại của nhà cung cấp.

- **Bảng DonNhap (Đơn Nhập)**
- **Mục đích:** Quản lý thông tin về các đơn nhập kho.
- **Cột:**
  - DonNhapID: Mã đơn nhập (khóa chính, tự tăng).
  - NgayNhapHang: Ngày nhập hàng vào kho.
  - NCCID: Mã nhà cung cấp (khóa ngoại tham chiếu từ bảng NCC).
  - TrangThai: Trạng thái đơn nhập (ví dụ: "Hoàn thành").
- **Bảng CTDonNhap (Chi Tiết Đơn Nhập)**
- **Mục đích:** Quản lý chi tiết sản phẩm trong mỗi đơn nhập.
- **Cột:**
  - CTDonNhapID: Mã chi tiết đơn nhập (khóa chính, tự tăng).
  - DonNhapID: Mã đơn nhập (khóa ngoại tham chiếu từ bảng DonNhap).
  - SanPhamID: Mã sản phẩm (khóa ngoại tham chiếu từ bảng SanPham).
  - SoLuong: Số lượng sản phẩm nhập vào.
  - GiaNhap: Giá nhập của sản phẩm trong đơn nhập.
- **Bảng KhachHang (Khách Hàng)**
- **Mục đích:** Lưu thông tin về khách hàng.
- **Cột:**
  - KhachHangID: Mã khách hàng (khóa chính, tự tăng).
  - HoTen: Họ tên khách hàng.
  - Email: Email của khách hàng (duy nhất).
  - SDT: Số điện thoại của khách hàng.
  - DiaChi: Địa chỉ của khách hàng.

- **Bảng DonHang (Đơn Hàng)**
- **Mục đích:** Quản lý thông tin về các đơn hàng của khách hàng.
- **Cột:**
  - DonHangID: Mã đơn hàng (khóa chính, tự tăng).
  - KhachHangID: Mã khách hàng (khóa ngoại tham chiếu từ bảng KhachHang).
  - NgayDat: Ngày đặt hàng.
  - TongTien: Tổng tiền của đơn hàng.
  - TrangThai: Trạng thái của đơn hàng (ví dụ: "Đang xử lý", "Đã giao").
- **Bảng CTDonHang (Chi Tiết Đơn Hàng)**
- **Mục đích:** Quản lý chi tiết các sản phẩm trong đơn hàng.
- **Cột:**
  - CTDonHangID: Mã chi tiết đơn hàng (khóa chính, tự tăng).
  - DonHangID: Mã đơn hàng (khóa ngoại tham chiếu từ bảng DonHang).
  - SanPhamID: Mã sản phẩm (khóa ngoại tham chiếu từ bảng SanPham).
  - SoLuong: Số lượng sản phẩm trong đơn hàng.
  - GiaBan: Giá bán của sản phẩm trong đơn hàng.
- **Bảng ThanhToan (Thanh Toán)**
- **Mục đích:** Quản lý thông tin thanh toán của đơn hàng.
- **Cột:**
  - ThanhToanID: Mã thanh toán (khóa chính, tự tăng).
  - DonHangID: Mã đơn hàng (khóa ngoại tham chiếu từ bảng DonHang).
  - PhuongThuc: Phương thức thanh toán (ví dụ: "Thẻ tín dụng", "Tiền mặt", "Chuyển khoản").
  - NgayThanhToan: Ngày thanh toán.

- **Bảng NgườiDung (Người Dùng)**
- **Mục đích:** Quản lý thông tin người dùng hệ thống.
- **Cột:**
  - **NguoIDungID:** Mã người dùng (khóa chính, tự tăng).
  - **HoTen:** Họ tên người dùng.
  - **Taikhuan:** Tài khoản đăng nhập của người dùng (duy nhất).
  - **MatKhou:** Mật khẩu của người dùng.
  - **Quyen:** Quyền hạn của người dùng (ví dụ: "Quản trị viên", "Khách hàng").

### 3.4 Thiết kế API phía BE



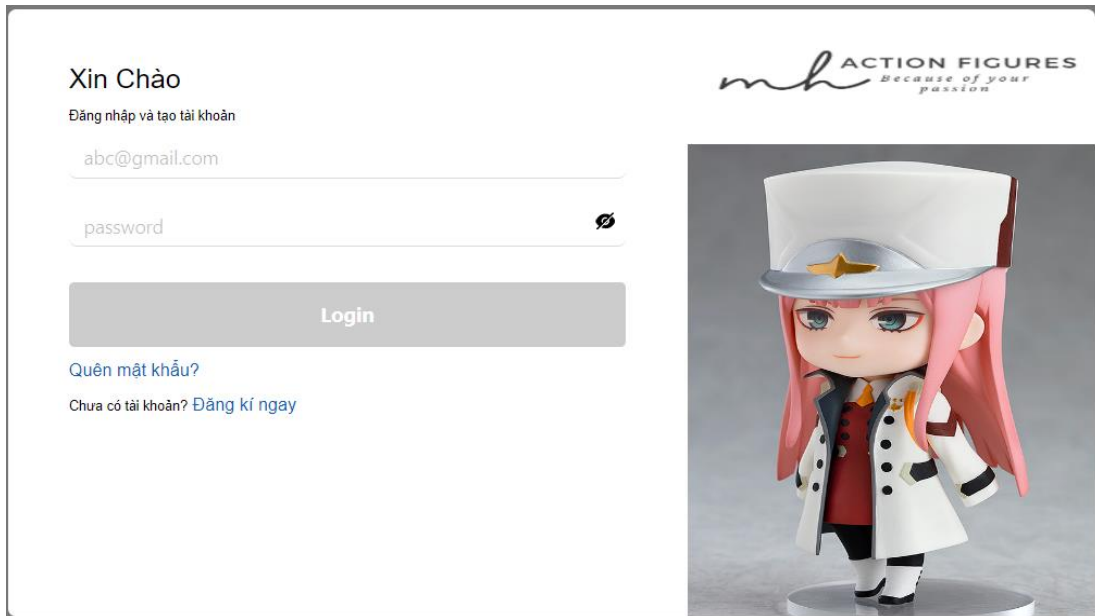
Hình 3.2: API phía BackEnd cho website bán mô hình

Api gồm các nghiệp vụ như Login, register, getAll, Search, Add, Update, Delete. Đây đủ các nghiệp vụ cho người dùng.

Ở đây cũng đã được phân quyền sử dụng: Admin sẽ có toàn bộ quyền còn người dùng chỉ có thể xem hoặc tìm kiếm sản phẩm

### 3.5 Thiết kế giao diện

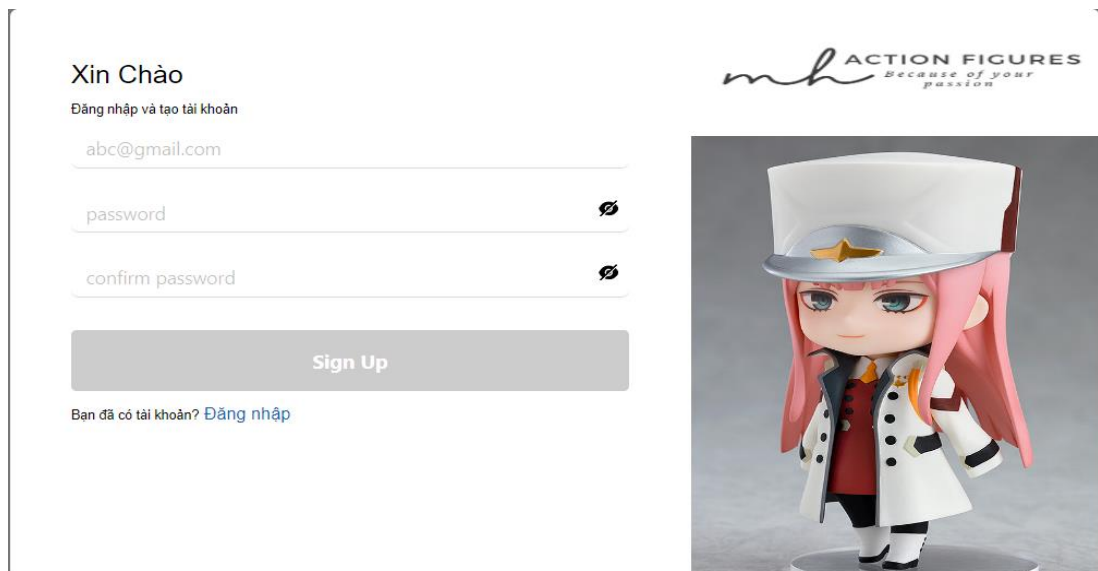
#### 3.5.1 Thiết kế giao diện đăng nhập đăng kí



Hình 3.5.1: Giao diện đăng nhập

- Các chức năng của trang đăng nhập bao gồm:
  - Nhập thông tin tài khoản, mật khẩu đã đăng ký
  - Đăng nhập
- Phía front-end
  - + Bố cục trang đăng nhập được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như nhập thông tin tài khoản và mật khẩu, đăng nhập vào hệ thống. Các thẻ `<div>`, `<span>`, `<button>` được sử dụng để tạo cấu trúc tổng thể cho trang
  - + CSS được sử dụng để định dạng và bố trí các phần của trang đăng nhập theo thiết kế. Các thuộc tính CSS như `display`, `flex`, `align-items`, `justify-content` được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng
  - + Sử dụng JavaScript, React để lập trình các chức năng
- Phía back-end

- + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.
- + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
- + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business

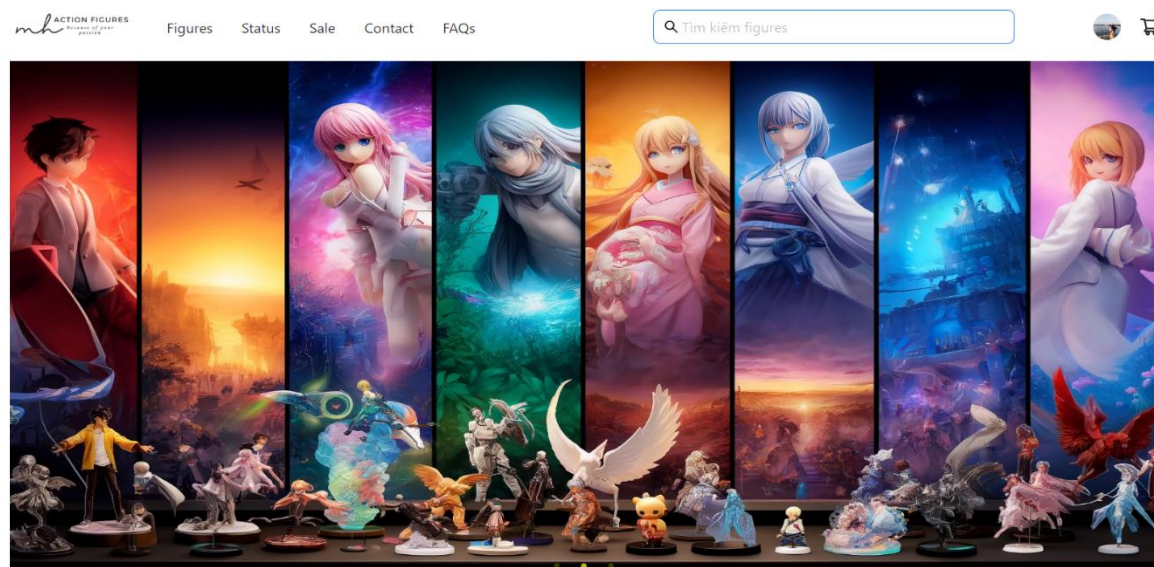


The image shows a registration form for a website called 'ACTION FIGURES'. The form is titled 'Xin Chào' (Hello) and has a subtitle 'Đăng nhập và tạo tài khoản' (Login and create account). It contains three input fields: 'abc@gmail.com' for email, 'password' for the password, and 'confirm password' for the confirmation password. Each password field has an eye icon to toggle visibility. Below the fields is a 'Sign Up' button. At the bottom, there is a link that says 'Bạn đã có tài khoản? Đăng nhập' (Do you have an account? Login). To the right of the form is a small image of a pink-haired anime-style action figure wearing a white uniform and a white hat with a gold star.

Hình 3.5.2: Giao diện trang đăng kí

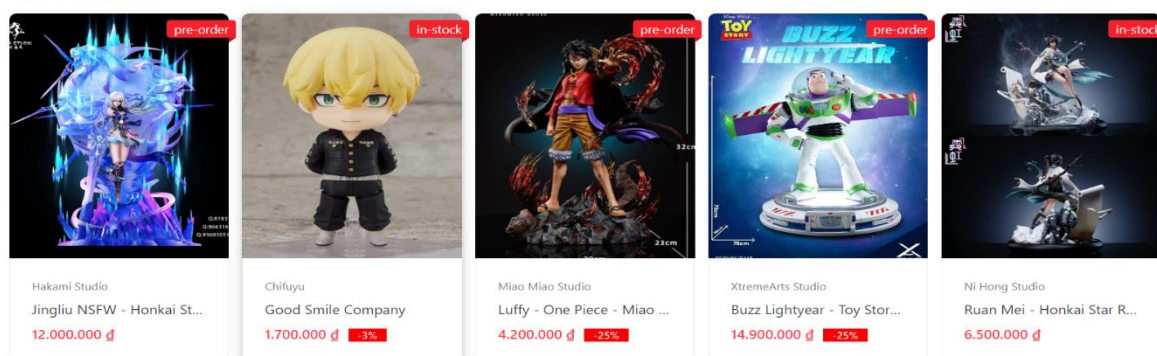
Trang đăng kí sẽ tương tự với trang đăng nhập nhưng sẽ thay đổi cách hoạt động ở BE. Thay vì log user ta sẽ 'createUser'

### 3.5.2 Thiết kế giao diện trang chủ

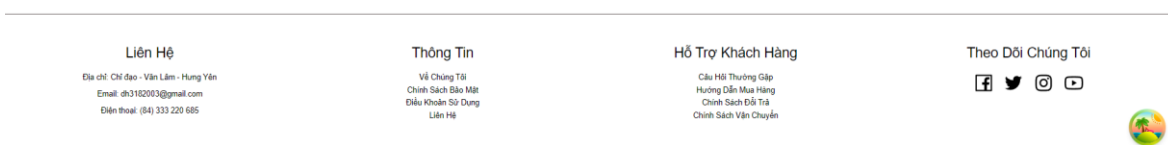


Hình 3.5.3: Giao diện header menu và banner

#### Tất cả sản phẩm



Hình 3.5.4: Sản phẩm trang chủ



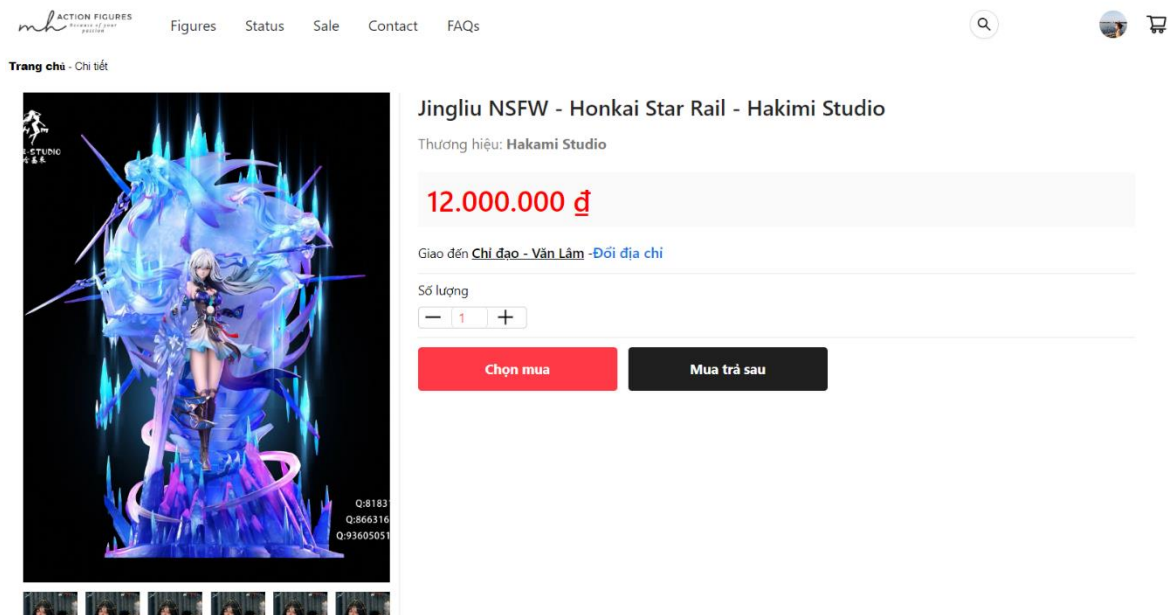
Hình 3.5.5: Menu danh mục sản phẩm

- Các chức năng của trang chủ bao gồm:
  - + Hiển thị danh sách sản phẩm theo dạng thẻ.
  - + Tìm kiếm và lọc sản phẩm theo từ khóa.
  - + Phân loại sản phẩm theo từng loại.
  - + Nút "Xem thêm" để tải thêm sản phẩm khi người dùng nhấn vào.



- Phía font-end
  - + Bố cục trang chủ được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như header, slider, danh sách sản phẩm và footer. Các thẻ `<div>`, `<header>`, `<footer>`, `<section>` được sử dụng để tạo cấu trúc tổng thể cho trang.
  - + CSS được sử dụng để định dạng và bố trí các phần của trang Home theo thiết kế. Các thuộc tính CSS như `display`, `flex`, `align-items`, `justify-content` được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng. Ví dụ, sử dụng `styled-components` để tạo các component CSS tái sử dụng
  - + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu sản phẩm từ server được thực hiện bằng cách sử dụng `React Query` để gọi API và cập nhật giao diện theo dữ liệu nhận được
- Phía back-end
  - + Lớp `DataAccess` xử lý các thao tác trên cơ sở dữ liệu, bao gồm các phương thức để lấy dữ liệu sản phẩm và loại sản phẩm từ cơ sở dữ liệu. Các thao tác này được triển khai bằng cách sử dụng các framework như `ExpressJS` và các thư viện kết nối cơ sở dữ liệu như `Mongoose`
  - + Lớp `Business` chứa các logic xử lý nghiệp vụ của ứng dụng. Các phương thức trong lớp này đảm bảo rằng dữ liệu lấy từ cơ sở dữ liệu được xử lý đúng cách trước khi trả về cho tầng `Controller` hoặc `View`.
  - + Lớp `Controller` chứa các action để xử lý các yêu cầu từ phía người dùng và gọi các phương thức trong tầng `Business` để thực hiện các thao tác cần thiết.

### 3.5.3 Thiết kế giao diện trang chi tiết sản phẩm



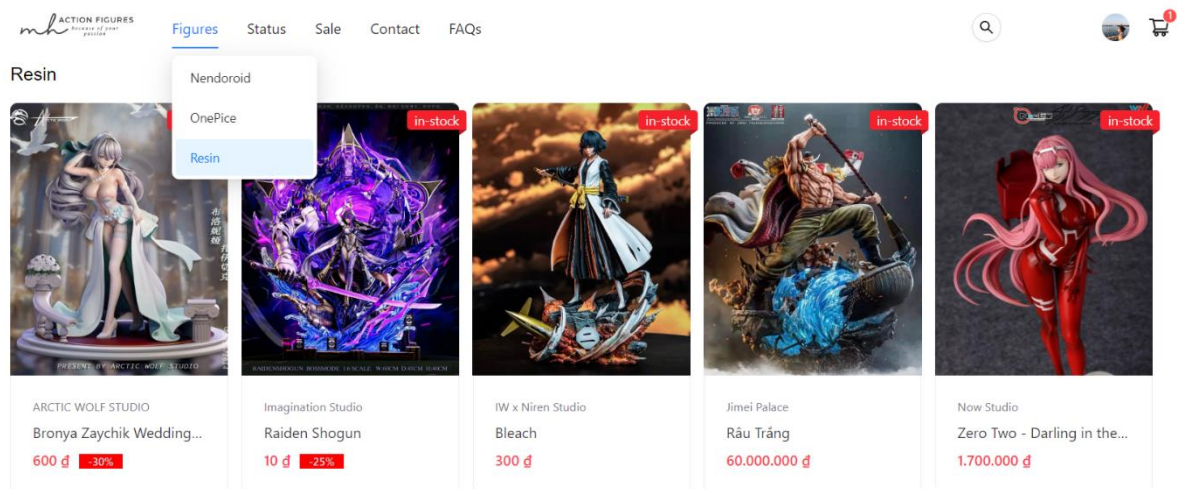
Hình 3.5.6: Giao diện trang chi tiết sản phẩm

- Các chức năng của trang chi tiết sản phẩm bao gồm:
  - + Hiển thị hình ảnh sản phẩm, thông tin sản phẩm
  - + Thông tin giao hàng, lựa chọn số lượng sản phẩm
  - + Thêm vào giỏ hàng.
- Phía font-end
  - + Bố cục trang chi tiết sản phẩm được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như hình ảnh sản phẩm, thông tin sản phẩm, đánh giá và lựa chọn số lượng
  - + CSS được sử dụng để định dạng và bố trí các phần của trang chi tiết sản phẩm theo thiết kế. Các thuộc tính CSS như display, flex, align-items, justify-content được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng
  - + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu chi tiết sản phẩm từ server được thực hiện bằng

cách sử dụng React Query để gọi API và cập nhật giao diện theo dữ liệu nhận được

- Phía backend
  - + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.
  - + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
  - + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business

### 3.5.4 Thiết kế giao diện trang danh mục sản phẩm

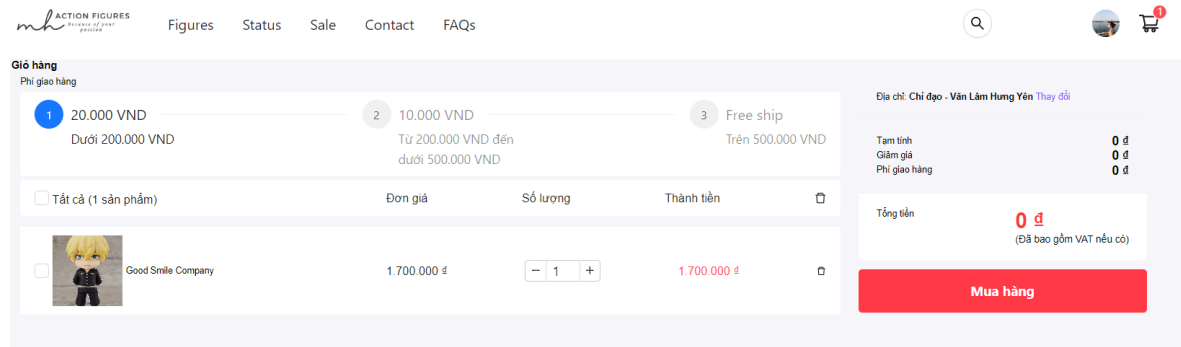


Hình 3.5.7: Giao diện trang danh mục sản phẩm

- Các chức năng của trang danh mục sản phẩm bao gồm:
  - + Hiển thị hình ảnh sản phẩm, thông tin sản phẩm theo loại, danh mục
- Phía font-end

- + Bố cục trang chi tiết sản phẩm được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như hình ảnh sản phẩm, thông tin sản phẩm, đánh giá và lựa chọn số lượng
- + CSS được sử dụng để định dạng và bố trí các phần của trang chi tiết sản phẩm theo thiết kế. Các thuộc tính CSS như display, flex, align-items, justify-content được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng
- + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu chi tiết sản phẩm từ server được thực hiện bằng cách sử dụng React Query để gọi API và cập nhật giao diện theo dữ liệu nhận được
- Phía backend
  - + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.
  - + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
  - + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business

### 3.5.5 Thiết kế giao diện bán hàng



Hình 3.5.8: Thiết kế giao diện giỏ hàng

- Các chức năng của trang giỏ hàng bao gồm:
  - + Hiển thị sản phẩm trong giỏ hàng
  - + Tùy chọn chỉnh sửa giỏ hàng
  - + Hiện thị thông tin thanh toán
  - + Nút "Mua hàng" để tiến hành thanh toán và đặt hàng.
- Phía front-end
  - + Bố cục trang chi tiết sản phẩm được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như hình ảnh sản phẩm, thông tin sản phẩm, đánh giá và lựa chọn số lượng
  - + CSS được sử dụng để định dạng và bố trí các phần của trang chi tiết sản phẩm theo thiết kế. Các thuộc tính CSS như display, flex, align-items, justify-content được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng
  - + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu chi tiết sản phẩm từ server được thực hiện bằng cách sử dụng React Query để gọi API và cập nhật giao diện theo dữ liệu nhận được
- Phía backend
  - + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.

- + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
- + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business

### 3.5.6 Thiết kế giao diện trang thanh toán

Thanh toán

Chọn phương thức giao hàng

☒ FAST Giao hàng tiết kiệm

☐ GO\_JEK Giao hàng tiết kiệm

Chọn phương thức thanh toán

☒ Thanh toán tiền mặt khi nhận hàng

☐ Thanh toán tiền bằng paypal

Địa chỉ: Chí đạo - Văn Lâm Hưng Yên [Thay đổi](#)

Tạm tính	10 đ
Giảm giá	2.5 đ
Phí giao hàng	20.000 đ

Tổng tiền **20.007.5 đ**  
(Đã bao gồm VAT nếu có)

[Đặt hàng](#)

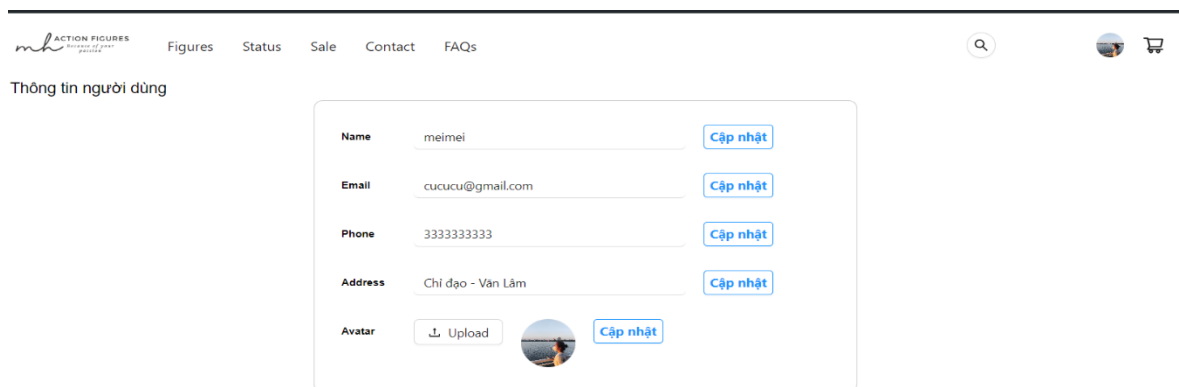
Hình 3.5.9: Giao diện trang thanh toán

- Các chức năng của trang thanh toán bao gồm:
  - + Hiện thị phương thức thanh toán
  - + Hiện thị các phương thức vận chuyển
  - + Hiện thị thông tin đơn hàng: giá tiền, giảm giá, tổng tiền
  - + Hiện thị thông tin thanh toán
  - + Nút "Đặt hàng" để tiến hành thanh toán và đặt hàng.
- Phía front-end
  - + Bố cục trang chi tiết sản phẩm được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như hình ảnh sản phẩm, thông tin sản phẩm, đánh giá và lựa chọn số lượng
  - + CSS được sử dụng để định dạng và bố trí các phần của trang chi tiết sản phẩm theo thiết kế. Các thuộc tính CSS như display, flex, align-items,

justify-content được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng

- + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu chi tiết sản phẩm từ server được thực hiện bằng cách sử dụng React Query để gọi API và cập nhật giao diện theo dữ liệu nhận được
- Phía backend
  - + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.
  - + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
  - + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business

### 3.5.7 Thiết kế giao diện thông tin người dùng



The screenshot shows a web interface for a user profile. At the top, there's a navigation bar with the 'Action Figures' logo and links for 'Figures', 'Status', 'Sale', 'Contact', and 'FAQs'. A search icon and a shopping cart icon are also present. Below the navigation bar, the page title is 'Thông tin người dùng'. The main content area contains a form with the following fields:

- Name:** melmei, with a 'Cập nhật' button.
- Email:** cucucu@gmail.com, with a 'Cập nhật' button.
- Phone:** 3333333333, with a 'Cập nhật' button.
- Address:** Chi đạo - Văn Lâm, with a 'Cập nhật' button.
- Avatar:** Includes an 'Upload' button, a preview image of a person, and a 'Cập nhật' button.

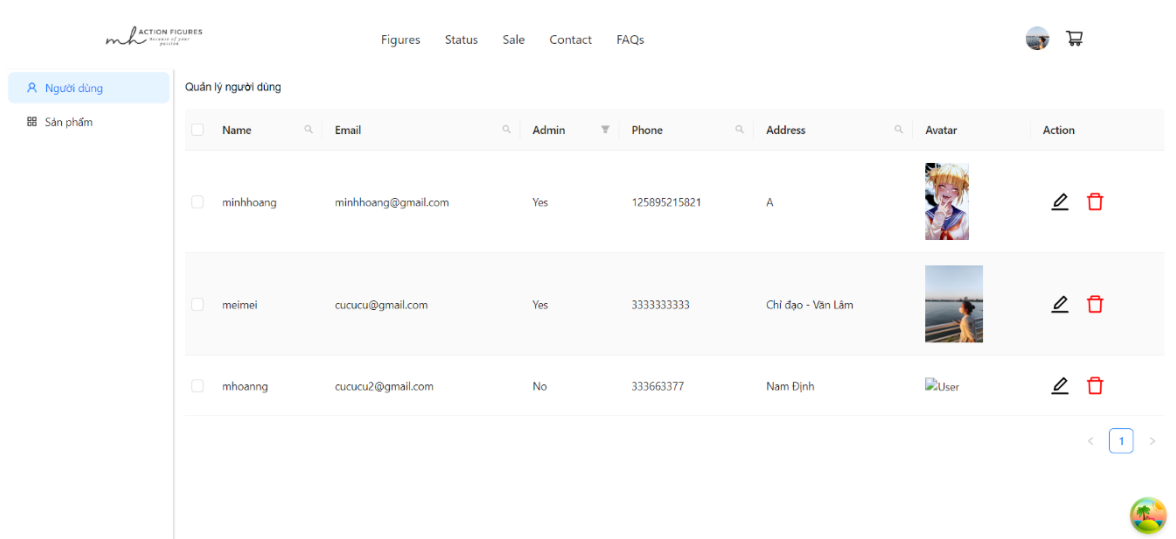
Hình 3.5.10: Giao diện thông tin người dùng

- Các chức năng của trang thông tin người dùng bao gồm:
  - + Hiển thị thông tin của người dùng
  - + Tùy chọn chỉnh sửa thông tin của người dùng

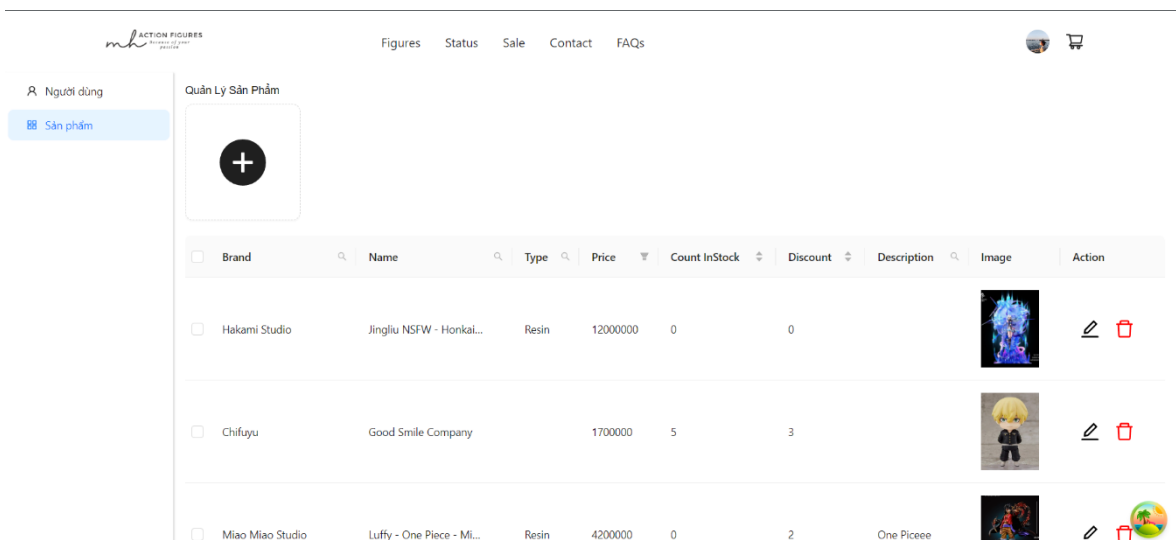
- Phía font-end
  - + Bố cục trang chi tiết sản phẩm được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như hình ảnh sản phẩm, thông tin sản phẩm, đánh giá và lựa chọn số lượng
  - + CSS được sử dụng để định dạng và bố trí các phần của trang chi tiết sản phẩm theo thiết kế. Các thuộc tính CSS như display, flex, align-items, justify-content được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng
  - + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu chi tiết sản phẩm từ server được thực hiện bằng cách sử dụng React Query để gọi API và cập nhật giao diện theo dữ liệu nhận được
- Phía backend
  - + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.
  - + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
  - + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business



### 3.5.8 Thiết kế giao diện trang Admin



Hình 3.5.11: Quản lý người dùng



Hình 3.5.12: Quản lý sản phẩm

The image shows a web form titled "Thêm Sản Phẩm" (Add Product) with a close button (X) in the top right corner. The form contains the following fields:

- \* Brand:
- \* Name:
- \* Type:
- \* Price:
- \* Count InStock:
- Description:
- Discount:
- Image:
- 

*Hình 3.5.13: Form thêm sản phẩm*

- Các chức năng của trang admin bao gồm:
  - + Chức năng quản lý người dùng: Thêm, sửa, xóa
  - + Chức năng quản lý sản phẩm: Thêm, sửa, xóa
- Phía front-end
  - + Bố cục trang chi tiết sản phẩm được xây dựng bằng cách sử dụng các thẻ HTML cơ bản để phân chia các phần như hình ảnh sản phẩm, thông tin sản phẩm, đánh giá và lựa chọn số lượng
  - + CSS được sử dụng để định dạng và bố trí các phần của trang chi tiết sản phẩm theo thiết kế. Các thuộc tính CSS như display, flex, align-items, justify-content được sử dụng để sắp xếp các phần tử và tạo khoảng cách giữa chúng

- + ReactJS được sử dụng để xây dựng các component và quản lý trạng thái của trang. Ví dụ, việc tải dữ liệu chi tiết sản phẩm từ server được thực hiện bằng cách sử dụng React Query để gọi API và cập nhật giao diện theo dữ liệu nhận được
- Phía backend
  - + Tầng DataAccess chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu, bao gồm các thao tác CRUD (Create, Read, Update, Delete). Tầng này sử dụng các mô hình (models) để đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu.
  - + Tầng Business chứa các logic xử lý nghiệp vụ của ứng dụng. Tầng này nhận dữ liệu từ tầng DataAccess và xử lý các nghiệp vụ cần thiết trước khi trả về kết quả cho tầng Controller hoặc View
  - + Tầng Controller chịu trách nhiệm nhận các yêu cầu từ phía người dùng, gọi các phương thức xử lý ở tầng Business, và trả về kết quả cho người dùng. Tầng này sử dụng các action để xử lý các yêu cầu HTTP và gọi các phương thức từ tầng Business

## CHƯƠNG 4: TRIỂN KHAI WEBSITE

### 4.1 Triển khai các chức năng cho phân hệ người dùng

Để xây dựng được các chức năng của trang người dùng theo thiết kế đã được trình bày ở chương 3, đã sử dụng HTML, CSS, và JavaScript, JQuery để thiết kế giao diện và thao tác dữ liệu các trang theo yêu cầu. Tiếp theo, bài tập lớn sẽ trình bày các kỹ thuật được sử dụng để xây dựng các trang.

Về mặt cấu tạo html, trang chủ gồm 2 phần chính là head và body. Phần phần head dùng cặp thẻ `<head></head>`. Phần head là phần cấu hình của một trang web, nơi khai báo các thư viện, link, các thẻ SEO(`<meta>`)

Phần body gồm các thẻ để tạo nên cấu trúc của trang web dùng cặp thẻ `<body></body>`. Thân trang web(body) lại được chia thành 3 phần header, main, footer, các thẻ này đều được hỗ trợ trong phiên bản HTML5. Header sử dụng cặp thẻ `<header></header>`. Phần này chứa navigation bar(`<nav>`) và tools bar. Navigation bar là cặp thẻ `<nav></nav>` bên trong có chứa thẻ `<ul></ul>` và tập các thẻ `<li></li>` để định dạng các phần tử của menu.

Bên trong các thẻ `<li>` còn có các menu con khác các menu con này cũng được định dạng giống như trên. Khi click chuột vào các item, trang web sẽ chuyển hướng, như vậy bên trong nội dung các thẻ li còn chứa các cặp thẻ `<a></a>` và thuộc tính href chứa các đường dẫn chuyển hướng. Các thẻ `<a>` được định dạng mặc định là chữ màu xanh và có gạch chân, để bỏ chúng đi sử dụng 2 thuộc tính là color và text-decoration.

Sau menu là phần banner giới thiệu sản phẩm mới cũng như là những sản phẩm nổi bật. Danh sách sản phẩm được bao bọc bởi thẻ div có class là banner và để trang web có thể responsive cần gán class là row. Đi kèm theo là phần sản phẩm mới mỗi sản phẩm được bao bọc bởi div được gán class spm. Spm có các class như col-`<number>` để có thể responsive.

Tiếp theo sẽ đến phần hợp tác cùng với các thương hiệu khác tạo nên sự mới mẻ trong từng sản phẩm. Danh sách được bao bọc bởi thẻ div có class là relationship

và để trang web có thể responsive cần gán class là row và col-<number> để có thể responsive.

Phần tiếp theo là danh mục các loại mô hình mà trang web bán gồm nhiều loại khác nhau gồm: Resin, OnePiece,... Danh sách sản phẩm được bao bọc bởi thẻ div có class là list-group và để trang web có thể responsive cần gán class là row. Đi kèm theo là phần sản phẩm mới mỗi sản phẩm được bao bọc bởi div được gán class pro-list. Pro-list có các class như col-<number> để có thể responsive.

## **4.2 Kiểm thử và triển khai ứng dụng**

### **4.2.1 Kiểm thử**

Trong giai đoạn này, toàn bộ các chức năng của website bán mô hình đã được kiểm thử để đảm bảo rằng hệ thống hoạt động đúng theo yêu cầu. Quá trình kiểm thử bao gồm kiểm tra từng chức năng, phát hiện và sửa lỗi để đảm bảo tính ổn định và hiệu suất của ứng dụng.

### **4.2.2 Đóng gói ứng dụng**

#### **1. Tạo build cho frontend:**

- Sử dụng công cụ build như Webpack hoặc Create React App để tạo build cho ứng dụng React.
- Chạy lệnh build để tạo thư mục chứa các file tĩnh của ứng dụng.

#### **2. Đóng gói backend:**

- Đảm bảo rằng tất cả các thư viện và phụ thuộc cần thiết được cài đặt và cấu hình.
- Sử dụng các công cụ như Docker để đóng gói ứng dụng Node.js thành một Docker image, giúp dễ dàng triển khai trên các môi trường khác nhau.

#### **3. Tạo Docker Compose file:**

- Sử dụng Docker Compose để quản lý và chạy các container cho frontend, backend và cơ sở dữ liệu.

- Định nghĩa các dịch vụ và cấu hình mạng, môi trường trong file Docker Compose.

#### 4. Chạy Docker Compose:

- Sử dụng Docker Compose để khởi động các container và kiểm tra xem tất cả các dịch vụ có hoạt động đúng không.

### 4.2.3 Triển khai ứng dụng

Điều kiện và môi trường triển khai ứng dụng bao gồm:

#### 1. Phần cứng:

- Máy chủ có cấu hình tối thiểu:
  - CPU: 2 nhân
  - RAM: 4GB
  - Ổ cứng: 50GB
- Máy chủ có khả năng chạy Docker.

#### 2. Phần mềm:

- Hệ điều hành: window 7 trở nên hoặc MacOS .
- Docker: Phiên bản Docker Engine mới nhất.
- Docker Compose: Phiên bản mới nhất.
- Node.js: Phiên bản 14 trở lên.
- MongoDB: Phiên bản 4.4 trở lên.

#### 3. Quy trình triển khai:

- Cài đặt Docker và Docker Compose trên máy chủ.
- Clone mã nguồn của ứng dụng từ Git repository.
- Tạo file cấu hình để cấu hình các biến môi trường cho ứng dụng.
- Chạy lệnh Docker Compose để khởi động các container.
- Kiểm tra và đảm bảo rằng các container frontend, backend và MongoDB đang chạy và kết nối thành công.

## KẾT LUẬN

### 4.3 1. Kết quả đạt được

Có thêm các kiến thức về HTML, CSS, JavaScript và Reactjs.

Nắm được kiến thức cơ bản về lập trình backend, xử lý dữ liệu, và quản lý server.

Có kỹ năng trong việc tương tác giữa frontend và backend thông qua các API.

### 4.4 2. Những hạn chế

Tuy nhiên sản phẩm còn những hạn chế như các chức năng trên trang quản trị chưa hoàn thành.

Các đường link gán, các chức năng của trang web vẫn còn thiếu, chức năng thống kê, hóa đơn cho phân hệ quản trị chưa được hoàn thiện.

### 4.5 3. Hướng phát triển

Tìm kiếm thêm các kiến thức để xây dựng thêm các chức năng còn thiếu để hoàn thiện website đáp ứng được với nhu cầu thực tế

## **TÀI LIỆU THAM KHẢO**

- [1] Khoa công nghệ thông tin, Giáo trình công nghệ Web và Ứng dụng
- [2] Trang web: <http://www.w3school.com>
- [3] Trang web: <https://fkhadra.github.io/react-toastify/introduction/>
- [4] Trang web: <https://vi.legacy.reactjs.org/tutorial/tutorial.html>