

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



**BÁO CÁO CUỐI KỲ**

**MÔN HỌC: LẬP TRÌNH WEB**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG**

**BÁN GIÀY SNEAKER ONLINE**

**GVHD: ThS. Mai Anh Thơ**

**SVTH: Đặng Minh Nhật – 21110842**

**Đặng Phú Quý – 21110849**

**Trần Quốc Phương - 21110847**

**Mã lớp học: WEPR330479\_23\_1\_07CLC (ST6)**

*TP. Hồ Chí Minh, tháng 12 năm 2023*

# MỤC LỤC

<b>CHƯƠNG I: KHẢO SÁT HIỆN TRẠNG .....</b>	<b>7</b>
1.1. Lí do chọn đề tài.....	7
1.2. Tính cấp thiết của đề tài .....	7
1.3. Khảo sát các ứng dụng tương tự .....	8
1.4. Mục tiêu cần đạt được.....	9
<b>CHƯƠNG 2: THIẾT KẾ ỨNG DỤNG .....</b>	<b>10</b>
2.1. Bảng phân công.....	10
2.2. Thiết kế lớp .....	11
2.2.1. Mô hình thiết kế.....	11
2.2.2. Business layer .....	12
2.2.3. Controller layer .....	27
2.2.4. Data layer .....	29
2.2.5. Util .....	39
2.3. Thiết kế cơ sở dữ liệu.....	40
2.3.1. Hình vẽ.....	40
2.3.2. Bảng mô tả các Table trong cơ sở dữ liệu .....	41
2.3.3. Bảng các Fields trong 1 Table .....	42
2.4. Thiết kế giao diện.....	46
2.4.1. Giao diện trang chủ.....	46
2.4.2. Giao diện trang xem sản phẩm .....	48
2.4.3. Giao diện chi tiết sản phẩm .....	49
2.4.4. Giao diện trang giỏ hàng.....	50
2.4.5. Giao diện trang thanh toán.....	51
2.4.6. Giao diện trang thanh toán vnpay .....	52
2.4.7. Giao diện trang hóa đơn.....	53
2.4.8. Giao diện đăng nhập và đăng ký tài khoản.....	54
2.4.9. Giao diện trang quản lý tài khoản.....	55
2.4.10. Giao diện trang thêm mới tài khoản .....	56

2.4.11.	Giao diện trang chỉnh sửa tài khoản .....	57
2.4.12.	Giao diện trang quản lý sản phẩm .....	58
2.4.13.	Giao diện trang thêm mới sản phẩm.....	59
2.4.14.	Giao diện trang chỉnh sửa sản phẩm.....	60
2.4.15.	Giao diện trang quản lý hóa đơn.....	61
2.4.16.	Giao diện trang xem hóa đơn.....	62
2.4.17.	Giao diện trang quản lý khuyến mãi.....	63
2.4.18.	Giao diện thêm mới mã giảm giá .....	64
2.4.19.	Giao diện chỉnh sửa mã giảm giá .....	65
2.4.20.	Giao diện quản lý danh mục thương hiệu.....	66
2.4.21.	Giao diện chỉnh sửa thương hiệu .....	67
2.4.22.	Giao diện trang thống kê sản phẩm bán được .....	68
2.4.23.	Giao diện trang thống kê số tiền mua hàng của người dùng .....	69
2.4.24.	Giao diện trang đặt lại mật khẩu.....	70
<b>CHƯƠNG 3:</b>	<b>TRIỂN KHAI ỨNG DỤNG .....</b>	<b>71</b>
3.1.	Render .....	71
3.2.	Triển khai cơ sở dữ liệu .....	71
3.3.	Triển khai ứng dụng web .....	72
<b>CHƯƠNG 4:</b>	<b>KIỂM THỬ .....</b>	<b>76</b>
<b>CHƯƠNG 5:</b>	<b>KẾT LUẬN .....</b>	<b>79</b>
5.1.	Kết quả đạt được .....	79
5.2.	Đánh giá .....	79
5.2.1.	Ưu điểm .....	79
5.2.2.	Khuyết điểm.....	79
5.3.	Hướng phát triển .....	79
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>80</b>

## DANH MỤC HÌNH ẢNH

Hình 1. Class diagram của bài toán .....	11
Hình 2. Các mối quan hệ giữa các bảng trong cơ sở dữ liệu .....	40
Hình 3. Giao diện trang chủ .....	46
Hình 4. Giao diện trang xem sản phẩm .....	48
Hình 5. Giao diện trang chi tiết sản phẩm .....	49
Hình 6. Giao diện trang giỏ hàng .....	50
Hình 7. Giao diện trang thanh toán .....	51
Hình 8. Giao diện trang thanh toán vnpay .....	53
Hình 9. Giao diện trang hóa đơn .....	53
Hình 10. Giao diện đăng nhập và đăng ký tài khoản .....	54
Hình 11. Giao diện trang quản lý tài khoản .....	55
Hình 12. Giao diện trang thêm mới tài khoản .....	56
Hình 13. Giao diện trang chỉnh sửa tài khoản .....	57
Hình 14. Giao diện trang quản lý sản phẩm .....	58
Hình 15. Giao diện trang thêm mới sản phẩm .....	59
Hình 16. Giao diện trang chỉnh sửa sản phẩm .....	60
Hình 17. Giao diện trang quản lý hóa đơn .....	61
Hình 18. Giao diện trang xem hóa đơn .....	62
Hình 19. Giao diện trang quản lý khuyến mãi .....	63
Hình 20. Giao diện thêm mới mã giảm giá .....	64
Hình 21. Giao diện trang chỉnh sửa mã giảm giá .....	65
Hình 22. Giao diện trang quản lý danh mục thương hiệu .....	66
Hình 23. Giao diện chỉnh sửa thương hiệu .....	67
Hình 24. Giao diện trang thống kê sản phẩm bán được .....	68
Hình 25. Giao diện trang thống kê số tiền mua hàng của người dùng .....	69
Hình 26. Giao diện trang đặt lại mật khẩu .....	70

## DANH MỤC CÁC BẢNG

<b>Bảng 1. Bảng phân công.....</b>	<b>10</b>
<b>Bảng 2. Các lớp Bussiness Layer.....</b>	<b>12</b>
<b>Bảng 3. Các phương thức của lớp User .....</b>	<b>14</b>
<b>Bảng 4. Các phương thức của lớp CategoryShoes .....</b>	<b>15</b>
<b>Bảng 5. Các phương thức của lớp Shoes .....</b>	<b>17</b>
<b>Bảng 6. Các phương thức của lớp ShoeSize.....</b>	<b>18</b>
<b>Bảng 7. Các phương thức của lớp sImage.....</b>	<b>18</b>
<b>Bảng 8. Các phương thức của lớp LineItem .....</b>	<b>19</b>
<b>Bảng 9. Các phương thức của lớp Cart.....</b>	<b>20</b>
<b>Bảng 10. Các phương thức của lớp PromotionCode.....</b>	<b>21</b>
<b>Bảng 11. Các phương thức của lớp ShippingInfo .....</b>	<b>23</b>
<b>Bảng 12. Các phương thức của lớp Invoice .....</b>	<b>25</b>
<b>Bảng 13. Các phương thức của lớp UserSpending.....</b>	<b>26</b>
<b>Bảng 14. Các phương thức của lớp SoldShoes.....</b>	<b>26</b>
<b>Bảng 15. Các lớp của Controller Layer.....</b>	<b>28</b>
<b>Bảng 16. Các lớp của Data Layer.....</b>	<b>29</b>
<b>Bảng 17. Các phương thức của lớp DBUtil.....</b>	<b>30</b>
<b>Bảng 18. Các phương thức của lớp UserDB .....</b>	<b>31</b>
<b>Bảng 19. Các phương thức của lớp CategoryDB.....</b>	<b>32</b>
<b>Bảng 20. Các phương thức của lớp ShoesDB.....</b>	<b>35</b>
<b>Bảng 21. Các phương thức của lớp LineItemDB.....</b>	<b>36</b>
<b>Bảng 22. Các phương thức của lớp CartDB .....</b>	<b>37</b>
<b>Bảng 23. Các phương thức của lớp PromotionDB .....</b>	<b>37</b>
<b>Bảng 24. Các phương thức của lớp InvoiceDB .....</b>	<b>38</b>
<b>Bảng 25. Các phương thức của lớp ReportDB .....</b>	<b>38</b>
<b>Bảng 26. Các lớp của Util.....</b>	<b>39</b>
<b>Bảng 27. Các phương thức của lớp CookieUtil .....</b>	<b>39</b>
<b>Bảng 28. Các phương thức của lớp EmailUtil .....</b>	<b>39</b>

<b>Bảng 29. Các bảng trong cơ sở dữ liệu .....</b>	<b>41</b>
<b>Bảng 30. Các Fields của bảng shipinfo .....</b>	<b>42</b>
<b>Bảng 31. Các Fields của bảng cart .....</b>	<b>42</b>
<b>Bảng 32. Các Fields của bảng cart_lineitem .....</b>	<b>42</b>
<b>Bảng 33. Các Fields của bảng category .....</b>	<b>42</b>
<b>Bảng 34. Các Fields của bảng image.....</b>	<b>43</b>
<b>Bảng 35. Các Fields của bảng invoice .....</b>	<b>43</b>
<b>Bảng 36. Các Fields của bảng invoice_lineitem .....</b>	<b>43</b>
<b>Bảng 37. Các Fields của bảng lineitem .....</b>	<b>44</b>
<b>Bảng 38. Các Fields của bảng promotion .....</b>	<b>44</b>
<b>Bảng 39. Các Fields của bảng shoes.....</b>	<b>44</b>
<b>Bảng 40. Bảng các Fields của shoes_image .....</b>	<b>45</b>
<b>Bảng 41. Bảng shoes_shoesize .....</b>	<b>45</b>
<b>Bảng 42. Các Fields của bảng shoesize .....</b>	<b>45</b>
<b>Bảng 43. Các Fields của bảng user_table .....</b>	<b>45</b>
<b>Bảng 44. Các tình huống kiểm thử.....</b>	<b>78</b>

# CHƯƠNG I: KHẢO SÁT HIỆN TRẠNG

## 1.1. Lí do chọn đề tài

Trong thời đại số hóa ngày càng phát triển, việc tạo ra một website bán giày online không chỉ là một cơ hội kinh doanh mà còn là một hành trình đầy hứng khởi trong ngành thời trang giày thể thao. Điều này là lý do thú vị để nhóm em chọn đề tài này cho dự án của nhóm. Thị trường giày thể thao online đang trở nên hấp dẫn hơn bao giờ hết. Sự đa dạng mẫu mã, hình dáng, màu sắc từ các thương hiệu cung cấp một cơ hội rộng lớn để tạo ra một nền tảng thương mại điện tử với sự linh hoạt và đa dạng về sản phẩm. Điểm mạnh của website bán giày online là khả năng tiếp cận một lượng lớn khách hàng trực tuyến. Sự thuận tiện trong việc mua sắm từ xa cùng với khả năng tùy chọn sản phẩm và mức giá đa dạng tạo điều kiện thuận lợi cho khách hàng.

Ngày nay, các website cũng đòi hỏi sự sáng tạo để thu hút và giữ chân khách hàng. Việc thiết kế website, chiến lược quảng cáo và các chiến dịch marketing mang tính sáng tạo có thể là yếu tố quyết định để tạo ra sự khác biệt và thành công trong môi trường cạnh tranh.

Tóm lại, việc chọn đề tài xây dựng một website bán giày online không chỉ đáp ứng yêu cầu môn học mà còn là cơ hội để nhóm em vận dụng những kiến thức của môn học để xây dựng một website bán hàng mang tính thiết thực nhất.

## 1.2. Tính cấp thiết của đề tài

Website chuyên về bán giày online không thể phủ nhận trong bối cảnh thị trường mua sắm ngày càng chuyển dịch từ cửa hàng truyền thống sang môi trường mua sắm trực tuyến. Đây không chỉ là một cơ hội kinh doanh mà còn là một bước quan trọng để thích ứng và tận dụng những tiềm năng phát triển mà thị trường này đem lại.

Việc tạo ra một nền tảng bán hàng giày trực tuyến mang lại sự tiện lợi tối đa cho người tiêu dùng. Khách hàng có thể lựa chọn và mua sắm sản phẩm từ bất kỳ đâu và bất kỳ khi nào mà họ muốn, thay vì phải dành thời gian và công sức để đến cửa hàng truyền thống. Một trong những ưu điểm đáng chú ý là khả năng cạnh tranh với các thương hiệu lớn. Với một website bán giày chất lượng, các doanh nghiệp mới có cơ hội tiếp cận và cạnh tranh trực tiếp với các

đổi thủ lớn hơn, thu hút và giữ chân khách hàng một cách hiệu quả. Website bán giày cũng mang lại trải nghiệm mua sắm tốt hơn, không chỉ thông qua việc cung cấp thông tin chi tiết về sản phẩm mà còn qua việc tạo sự tương tác và tin cậy thông qua đánh giá từ người dùng trước đó.

Ngoài ra, việc quản lý và cập nhật thông tin sản phẩm trên website trở nên dễ dàng hơn bao giờ hết. Điều này cho phép doanh nghiệp linh hoạt thay đổi chiến lược bán hàng, điều chỉnh giá cả hay thực hiện các chiến dịch khuyến mãi một cách nhanh chóng và linh hoạt để phản ứng với diễn biến thị trường.

Cuối cùng, việc phát triển một website bán giày online mở ra cơ hội kinh doanh toàn cầu, mở rộng thị trường và tiếp cận khách hàng trên toàn thế giới. Điều này không chỉ làm tăng trưởng doanh số bán hàng mà còn là bước đi đưa doanh nghiệp vào môi trường cạnh tranh rộng lớn hơn.

### **1.3. Khảo sát các ứng dụng tương tự**

- Website Adidas: <https://www.adidas.com.vn/vi>
  - Giao diện: Cầu kì, rối mắt.
  - Chức năng: Đây đủ các chức năng cơ bản phục vụ cho việc mua sắm của người dùng.
  - Tốc độ phản hồi: Nhanh.
  - Thích ứng với thiết bị điện thoại động: Có.
- Website Nike: <https://www.nike.com/vn/>
  - Giao diện: Đơn giản, dễ nhìn, thân thiện với người dùng.
  - Chức năng: Đây đủ các chức năng cơ bản phục vụ cho việc mua sắm của người dùng.
  - Tốc độ phản hồi: Nhanh.
  - Thích ứng với thiết bị điện thoại di động: Có.
- Website Converse: <https://www.converse.vn/>
  - Giao diện: Đẹp, đơn giản, dễ nhìn, thân thiện với người dùng.
  - Chức năng: Đây đủ các chức năng cơ bản phục vụ cho việc mua sắm của người dùng.
  - Tốc độ phản hồi: Nhanh.
  - Thích ứng với thiết bị điện thoại di động: Có.



#### **1.4. Mục tiêu cần đạt được**

Dựa trên các kết quả khảo sát trên, nhóm em nhận thấy các yêu cầu cơ bản của một website bán hàng online cần đạt được như sau:

- Website phải đẹp, đơn giản, dễ nhìn, thân thiện với người dùng.
- Đầy đủ các chức năng cơ bản của một website bán hàng online như xem sản phẩm, mua hàng, thanh toán để phục vụ cho khách hàng.
- Tốc độ phản hồi nhanh chóng.
- Thích ứng với thiết bị điện thoại di động.

Ngoài ra, website bán hàng online còn phải có các chức năng để quản lý khách hàng và các sản phẩm được đăng bán trên website:

- Quản lý việc thêm, sửa, xóa sản phẩm.
- Phân loại các sản phẩm theo thương hiệu, giá tiền.
- Quản lý các đơn hàng, hóa đơn mua sắm của khách hàng.
- Quản lý tài khoản của khách hàng mua sắm.
- Gửi hóa đơn xác nhận đến mail khách hàng sau khi thanh toán.
- Đưa ra những chiến dịch khuyến mãi cho khách hàng.
- Gửi thông báo khuyến mãi đến mail khách hàng mỗi khi có chương trình khuyến mãi.
- Thống kê các sản phẩm bán được, chi tiêu của khách hàng.

## CHƯƠNG 2: THIẾT KẾ ỨNG DỤNG

### 2.1. Bảng phân công

Bảng phân công công việc

TT	Tên SV	Mô tả khái quát công việc SV thực hiện trong đề án	Ước tính phần trăm đóng góp
1	Đặng Phú Quý	<ul style="list-style-type: none"><li>- Xây dựng giao diện.</li><li>- Xây dựng chức năng đăng nhập, đăng xuất.</li><li>- Xây dựng các chức năng phần admin.</li><li>- Viết báo cáo</li></ul>	100%
2	Trần Quốc Phương	<ul style="list-style-type: none"><li>- Xây dựng giao diện</li><li>- Xây dựng các chức năng trang chủ.</li><li>- Xây dựng các chức năng của trang danh mục và chi tiết sản phẩm.</li><li>- Viết báo cáo</li></ul>	100%
3	Đặng Minh Nhật	<ul style="list-style-type: none"><li>- Xây dựng giao diện.</li><li>- Xây dựng các chức năng giỏ hàng.</li><li>- Xây dựng các chức năng thanh toán.</li><li>- Viết báo cáo</li></ul>	100%

**Bảng 1. Bảng phân công**

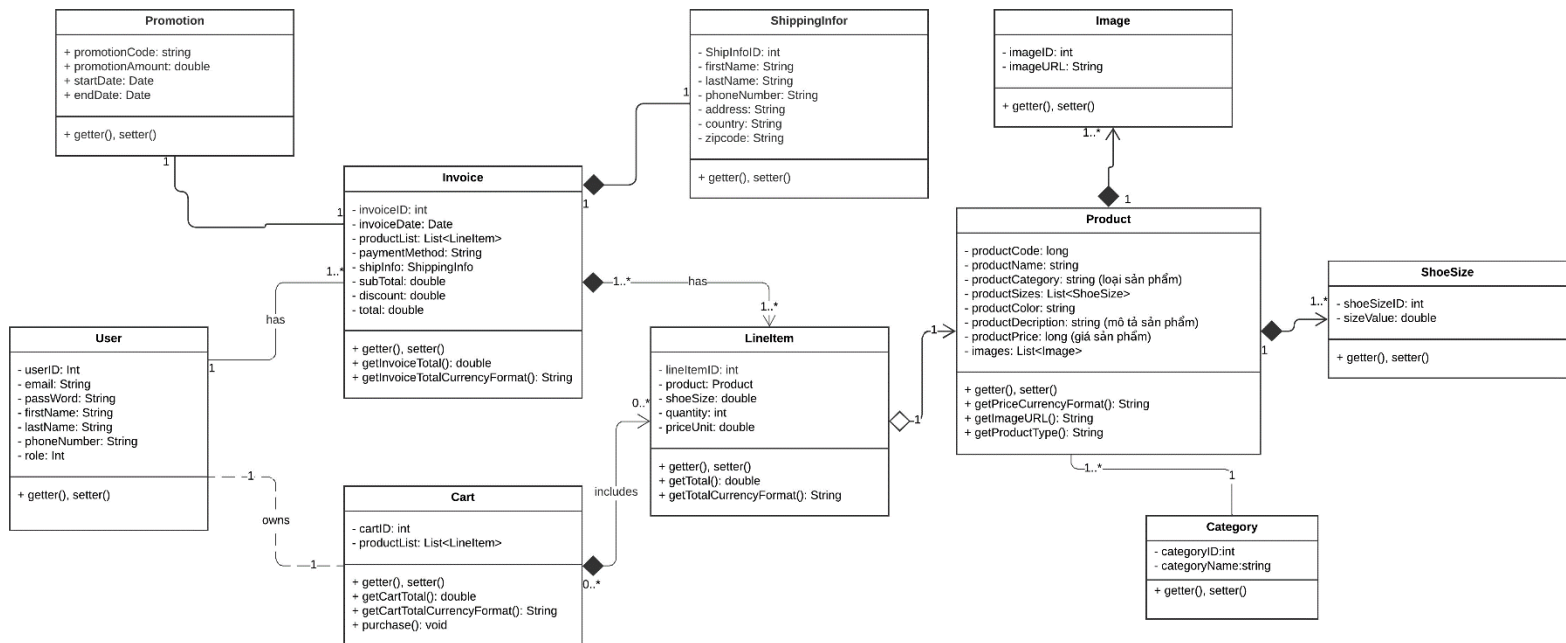
## 2.2. Thiết kế lớp

### 2.2.1. Mô hình thiết kế

Trong việc thiết kế hệ thống, mô hình MVC (Model-View-Controller) là một mô hình phổ biến để phát triển các ứng dụng web, giúp tách biệt logic xử lý dữ liệu (Model), giao diện người dùng (View), và xử lý các tương tác người dùng (Controller). Sử dụng Java Servlet trong mô hình MVC giúp xử lý các yêu cầu HTTP và phản hồi đến người dùng, làm cầu nối giữa tầng View và Model.

Mô hình MVC cung cấp cấu trúc rõ ràng, giúp tách biệt các phần khác nhau của ứng dụng, làm cho việc bảo trì và mở rộng ứng dụng trở nên dễ dàng hơn. Việc sử dụng Java Servlet trong mô hình này giúp xử lý các yêu cầu và phản hồi một cách hiệu quả, tận dụng lợi thế của Java trong việc xây dựng các ứng dụng web đa năng và mạnh mẽ.

### Class diagram:



Hình 1. Class diagram của bài toán

### 2.2.2. Business layer

Chứa các lớp biểu diễn logic nghiệp vụ, như quản lý thông tin sản phẩm, người dùng, đơn hàng. Mỗi lớp được thiết kế với mục đích cụ thể, giúp quản lý và xử lý dữ liệu liên quan đến nghiệp vụ cụ thể.

TT	Tên lớp	Mục đích
1	User	Lưu trữ thông tin cá nhân của người dùng.
2	CategoryShoes	Danh mục sản phẩm để phân loại sản phẩm
3	Shoes	Biểu diễn thông tin chi tiết về sản phẩm giày và mối quan hệ với các đối tượng liên quan.
4	ShoeSize	Biểu diễn thông tin kích thước giày
5	sImage	Lưu trữ thông tin về hình ảnh
6	LineItem	Đại diện cho một sản phẩm được mua trong giỏ hàng.
7	Cart	Quản lý thông tin về giỏ hàng của người dùng trong ứng dụng web.
8	PromotionCode	Quản lý thông tin về các chương trình khuyến mãi.
9	ShippingInfo	Quản lý thông tin vận chuyển của đơn hàng.
10	Invoice	Quản lý mối quan hệ giữa hoá đơn và các đối tượng khác, và cung cấp các phương thức để tính toán giá trị của hóa đơn.
11	UserSpending	Giữ thông tin về số tiền đã chi tiêu của một người dùng cụ thể để theo dõi và thống kê hoạt động mua sắm của họ
12	SoldShoes	Giữ thông tin về số lượng đôi giày đã bán của một sản phẩm giày cụ thể để theo dõi doanh số bán hàng.

**Bảng 2. Các lớp Bussiness Layer**

### Phương thức của lớp User

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getId() Input: Không có Output: Id	Trả về giá trị của thuộc tính Id.	User.java (60)
2	setId(Long id) Input: Id Output: Không có	Thiết lập giá trị cho thuộc tính Id.	User.java (64)
3	getEmail() Input: Không có Output: Email	Trả về giá trị của thuộc tính Email.	User.java (68)
4	setEmail(String email) Input: Email Output: Không có	Thiết lập giá trị cho thuộc tính Email.	User.java (72)
5	getPassword() Input: Không có Output: Password	Trả về giá trị của thuộc tính Password.	User.java (76)
6	setPassword(String password) Input: Password Output: Không có	Thiết lập giá trị cho thuộc tính Password.	User.java (80)
7	getFirstName() Input: Không có Output: FirstName	Trả về giá trị của thuộc tính FirstName.	User.java (84)
8	setFirstName(String firstName) Input: FirstName Output: Không có	Thiết lập giá trị cho thuộc tính FirstName.	User.java (88)
9	getLastName() Input: Không có Output: LastName	Trả về giá trị của thuộc tính LastName.	User.java (92)
10	setLastName(String lastName) Input: LastName Output: Không có	Thiết lập giá trị cho thuộc tính LastName.	User.java (96)

11	getPhoneNumber() Input: Không có Output: PhoneNumber	Trả về giá trị của thuộc tính PhoneNumber.	User.java (100)
12	setPhoneNumber(String phoneNumber) Input: PhoneNumber Output: Không có	Thiết lập giá trị cho thuộc tính PhoneNumber.	User.java (104)
13	getIsAdmin() Input: Không có Output: IsAdmin	Trả về giá trị của thuộc tính IsAdmin.	User.java (108)
14	setIsAdmin(int isAdmin) Input: IsAdmin Output: Không có	Thiết lập giá trị cho thuộc tính IsAdmin.	User.java (112)
15	getCart() Input: Không có Output: Cart	Trả về giá trị của thuộc tính Cart.	User.java (116)
16	setCart(Cart cart) Input: Cart Output: Không có	Thiết lập giá trị cho thuộc tính Cart.	User.java (120)
17	getFullName() Input: Không có Output: FullName	Trả về chuỗi chứa tên đầy đủ của một user.	User.java (124)

**Bảng 3. Các phương thức của lớp User**

**Phương thức lớp CategoryShoes**

<b>TT</b>	<b>Phương thức</b>	<b>Mục đích</b>	<b>Tên file, số thứ tự dòng chứa khai báo.</b>
1	getCategoryID() Input: Không có Output: categoryID	Trả về giá trị của thuộc tính categoryID.	CategoryID.java (35)
2	setCategoryID(int categoryID) Input: categoryID Output: Không có	Thiết lập giá trị cho thuộc tính categoryID.	CategoryID.java (39)
3	getCategoryName() Input: Không có Output: categoryName	Trả về giá trị của thuộc tính categoryName.	CategoryID.java (43)
4	setCategoryName(String categoryName) Input: categoryName Output: Không có	Thiết lập giá trị cho thuộc tính categoryName.	CategoryID.java (47)

**Bảng 4. Các phương thức của lớp CategoryShoes**

### Phương thức của lớp Shoes

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getProductID() Input: Không có Output: productID	Trả về giá trị của thuộc tính productID.	Shoes.java (88)
2	setProductID(int productID) Input: productID Output: Không có	Thiết lập giá trị cho thuộc tính productID.	Shoes.java (92)
3	getProductName() Input: Không có Output: productID	Trả về giá trị của thuộc tính productName.	Shoes.java (96)
4	setProductName(String productName) Input: productID Output: Không có	Thiết lập giá trị cho thuộc tính productName.	Shoes.java (100)
5	getSizes() Input: Không có Output: sizes	Trả về giá trị của thuộc tính sizes.	Shoes.java (111)
6	setSizes(List<ShoeSize> sizes) Input: sizes Output: Không có	Thiết lập giá trị cho thuộc tính sizes.	Shoes.java (115)
7	getProductColor() Input: Không có Output: productColor	Trả về giá trị của thuộc tính productColor.	Shoes.java (119)
8	setProductColor(String productColor) Input: productColor Output: Không có	Thiết lập giá trị cho thuộc tính productColor.	Shoes.java (123)
9	getProductDecription() Input: Không có Output: productDecription	Trả về giá trị của thuộc tính productDecription.	Shoes.java (127)
10	setProductDecription(String productDecription) Input: productDecription	Thiết lập giá trị cho thuộc tính productDecription.	Shoes.java (131)



	Output: Không có		
11	getProductPrice() Input: Không có Output: productPrice	Trả về giá trị của thuộc tính productPrice.	Shoes.java (135)
12	setProductPrice(Long productPrice) Input: productPrice Output: Không có	Thiết lập giá trị cho thuộc tính productPrice.	Shoes.java (139)
13	getCategory() Input: Không có Output: category	Trả về giá trị của thuộc tính category.	Shoes.java (143)
14	setCategory(CategoryShoes category) Input: category Output: Không có	Thiết lập giá trị cho thuộc tính category.	Shoes.java (147)
15	getImages() Input: Không có Output: images	Trả về giá trị của thuộc tính images.	Shoes.java (151)
16	setImages(List<Image> images) Input: images Output: Không có	Thiết lập giá trị cho thuộc tính images.	Shoes.java (155)
17	getProductSizeString() Input: Không có Output: String	Chuỗi gồm các size hiện có ngăn cách nhau bởi dấu “/”.	Shoes.java (159)

**Bảng 5. Các phương thức của lớp Shoes**

### Phương thức của lớp ShoeSize

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getSizeId() Input: Không có Output: sizeId	Trả về giá trị của thuộc tính sizeId.	ShoeSize.java (37)
2	setSizeId(int sizeId) Input: sizeId Output: Không có	Thiết lập giá trị cho thuộc tính sizeId.	ShoeSize.java (41)
3	getSizeValue() Input: Không có Output: sizeValue	Trả về giá trị của thuộc tính sizeValue.	ShoeSize.java (45)
4	setSizeValue(double sizeValue) Input: sizeValue Output: Không có	Thiết lập giá trị cho thuộc tính sizeValue.	ShoeSize.java (49)

**Bảng 6. Các phương thức của lớp ShoeSize**

### Phương thức của lớp sImage

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getImageID() Input: Không có Output: ImageID	Trả về giá trị của thuộc tính ImageID.	sImage.java (39)
2	setImageID(Long imageID) Input: ImageID Output: Không có	Thiết lập giá trị cho thuộc tính ImageID.	sImage.java (43)
3	getImageURL() Input: Không có Output: ImageURL	Trả về giá trị của thuộc tính ImageURL.	sImage.java (47)
4	setImageUrl(String imageURL) Input: ImageURL Output: Không có	Thiết lập giá trị cho thuộc tính ImageURL.	sImage.java (51)

**Bảng 7. Các phương thức của lớp sImage**

### Phương thức của lớp LineItem

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getLineItemID() Input: Không có Output: LineItemID	Trả về giá trị của thuộc tính LineItemID.	LineItem.java (45)
2	getProduct() Input: Không có Output: Product	Trả về giá trị của thuộc tính Product.	LineItem.java (49)
3	setProduct(Shoes product) Input: product Output: Không có	Thiết lập giá trị cho thuộc tính product.	LineItem.java (53)
4	getQuantity() Input: Không có Output: Quantity	Trả về giá trị của thuộc tính Quantity.	LineItem.java (57)
5	setQuantity(int quantity) Input: quantity Output: Không có	Thiết lập giá trị cho thuộc tính quantity.	LineItem.java (61)
6	getShoeSize() Input: Không có Output: ShoeSize	Trả về giá trị của thuộc tính ShoeSize.	LineItem.java (65)
7	setShoeSize(double shoeSize) Input: shoeSize Output: Không có	Thiết lập giá trị cho thuộc tính shoeSize.	LineItem.java (69)
8	getPriceUnit() Input: Không có Output: PriceUnit	Trả về giá trị của thuộc tính PriceUnit.	LineItem.java (73)
9	setPriceUnit(double priceUnit) Input: PriceUnit Output: Không có	Thiết lập giá trị cho thuộc tính PriceUnit.	LineItem.java (77)
10	getAmount()	Trả về giá trị của một LineItem bằng đơn giá * số lượng.	LineItem.java (81)

**Bảng 8. Các phương thức của lớp LineItem**

### Phương thức của lớp Cart

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getCartID() Input: Không có Output: CartID	Trả về giá trị của thuộc tính CartID.	Cart.java (39)
2	setCartID(Long cartID) Input: CartID Output: Không có	Thiết lập giá trị cho thuộc tính CartID.	Cart.java (43)
3	getProductList() Input: Không có Output: ProductList	Trả về giá trị của thuộc tính ProductList.	Cart.java (47)
4	setProductList(List<LineItem> productList) Input: ProductList Output: Không có	Thiết lập giá trị cho thuộc tính ProductList.	Cart.java (51)
5	totalCart() Input: Không có Output: double	Tính tổng giá trị của giỏ hàng.	Cart.java (55)
6	addItem(LineItem item) Input: item Output: Không có	Thêm một mục vào giỏ hàng. Nếu mục đã tồn tại, tăng số lượng của nó; nếu không, thêm mới.	Cart.java (64)
7	updateItem(LineItem item, String change) Input: item Output: Không có	Cập nhật số lượng của một mục trong giỏ hàng	Cart.java (79)
8	removeItem(LineItem item) Input: item Output: Không có	Loại bỏ một mục khỏi giỏ hàng	Cart.java (101)

**Bảng 9. Các phương thức của lớp Cart**

**Phương thức của lớp PromotionCode**

<b>TT</b>	<b>Phương thức</b>	<b>Mục đích</b>	<b>Tên file, số thứ tự dòng chứa khai báo.</b>
1	getPromotionCode() Input: Không có Output: PromotionCode	Trả về giá trị của thuộc tính PromotionCode.	PromotionCode.java (37)
2	setPromotionCode(String promotionCode) Input: PromotionCode Output: Không có	Thiết lập giá trị cho thuộc tính PromotionCode.	PromotionCode.java (41)
3	getPromotionAmount() Input: Không có Output: PromotionAmount	Trả về giá trị của thuộc tính PromotionAmount.	PromotionCode.java (45)
4	setPromotionAmount(double promotionAmount) Input: PromotionAmount Output: Không có	Thiết lập giá trị cho thuộc tính PromotionAmount.	PromotionCode.java (49)
5	getStartDate() Input: Không có Output: StartDate	Trả về giá trị của thuộc tính StartDate.	PromotionCode.java (53)
6	setStartDate(Date startDate) Input: StartDate Output: Không có	Thiết lập giá trị cho thuộc tính StartDate.	PromotionCode.java (57)
7	getEndDate() Input: Không có Output: EndDate	Trả về giá trị của thuộc tính EndDate.	PromotionCode.java (61)
8	setEndDate(Date endDate) Input: EndDate Output: Không có	Thiết lập giá trị cho thuộc tính EndDate.	PromotionCode.java (65)

**Bảng 10. Các phương thức của lớp PromotionCode**

### Phương thức của lớp ShippingInfo

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getShipInfoID() Input: Không có Output: ShipInfoID	Trả về giá trị của thuộc tính ShipInfoID.	ShippingInfo.java (28)
2	setShipInfoID(Long shipInfoID) Input: ShipInfoID Output: Không có	Thiết lập giá trị cho thuộc tính ShipInfoID.	ShippingInfo.java (32)
3	getFirstName() Input: Không có Output: FirstName	Trả về giá trị của thuộc tính FirstName.	ShippingInfo.java (55)
4	setFirstName(String firstName) Input: FirstName Output: Không có	Thiết lập giá trị cho thuộc tính FirstName.	ShippingInfo.java (59)
5	getLastName() Input: Không có Output: LastName	Trả về giá trị của thuộc tính LastName.	ShippingInfo.java (63)
6	setLastName(String lastName) Input: LastName Output: Không có	Thiết lập giá trị cho thuộc tính LastName.	ShippingInfo.java (67)
7	getPhoneNumber() Input: Không có Output: PhoneNumber	Trả về giá trị của thuộc tính PhoneNumber.	ShippingInfo.java (71)
8	setPhoneNumber(String phoneNumber) Input: PhoneNumber Output: Không có	Thiết lập giá trị cho thuộc tính PhoneNumber.	ShippingInfo.java (75)
9	getAddress() Input: Không có Output: Address	Trả về giá trị của thuộc tính Address.	ShippingInfo.java (79)
10	setAddress(String address) Input: Address Output: Không có	Thiết lập giá trị cho thuộc tính Address.	ShippingInfo.java (83)

11	getCountry() Input: Không có Output: Country	Trả về giá trị của thuộc tính Country.	ShippingInfo.java (87)
12	setCountry(String country) Input: Country Output: Không có	Thiết lập giá trị cho thuộc tính Country.	ShippingInfo.java (91)
13	getZipcode() Input: Không có Output: Zipcode	Trả về giá trị của thuộc tính Zipcode.	ShippingInfo.java (95)
14	setZipcode(String zipcode) Input: Zipcode Output: Không có	Thiết lập giá trị cho thuộc tính Zipcode.	ShippingInfo.java (99)
15	toString() Input: Không có Output: String	Trả về chuỗi gồm address, country và zipcode.	ShippingInfo.java (104)

**Bảng 11. Các phương thức của lớp ShippingInfo**

### Phương thức của lớp Invoice

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getUser() Input: Không có Output: User	Trả về giá trị của thuộc tính User.	Invoice.java (82)
2	setUser(User user) Input: User Output: Không có	Thiết lập giá trị cho thuộc tính User.	Invoice.java (86)
3	getInvoiceID() Input: Không có Output: InvoiceID	Trả về giá trị của thuộc tính InvoiceID.	Invoice.java (90)
4	getInvoiceDate() Input: Không có Output: InvoiceDate	Trả về giá trị của thuộc tính InvoiceDate.	Invoice.java (94)
5	setInvoiceDate(Date invoiceDate) Input: InvoiceDate Output: Không có	Thiết lập giá trị cho thuộc tính InvoiceDate.	Invoice.java (98)
6	getProductList() Input: Không có Output: ProductList	Trả về giá trị của thuộc tính ProductList.	Invoice.java (102)
7	setProductList(List<LineItem> productList) Input: ProductList Output: Không có	Thiết lập giá trị cho thuộc tính ProductList.	Invoice.java (106)
8	getPaymentMethod() Input: Không có Output: PaymentMethod	Trả về giá trị của thuộc tính PaymentMethod.	Invoice.java (110)
9	setPaymentMethod(String paymentMethod) Input: PaymentMethod Output: Không có	Thiết lập giá trị cho thuộc tính PaymentMethod.	Invoice.java (114)



10	getAddress() Input: Không có Output: ShippingInfo	Trả về giá trị của thuộc tính Address là thông tin giao hàng.	Invoice.java (118)
11	setAddress(ShippingInfo address) Input: ShippingInfo Output: Không có	Thiết lập giá trị cho thuộc tính Address là thông tin giao hàng..	Invoice.java (122)
12	getPromotion() Input: Không có Output: Promotion	Trả về giá trị của thuộc tính Promotion.	Invoice.java (126)
13	setPromotion(PromotionCode promotion) Input: Promotion Output: Không có	Thiết lập giá trị cho thuộc tính Promotion.	Invoice.java (130)
14	getSubTotal() Input: Không có Output: SubTotal	Trả về giá trị của thuộc tính SubTotal.	Invoice.java (134)
15	setSubTotal(double subTotal) Input: SubTotal Output: Không có	Thiết lập giá trị cho thuộc tính SubTotal.	Invoice.java (138)
16	getDiscount() Input: Không có Output: Discount	Trả về giá trị của thuộc tính Discount.	Invoice.java (142)
17	setDiscount(double discount) Input: Discount Output: Không có	Thiết lập giá trị cho thuộc tính Discount.	Invoice.java (146)
18	getTotal() Input: Không có Output: Total	Trả về giá trị của thuộc tính Total.	Invoice.java (150)
19	setTotal(double total) Input: Total Output: Không có	Thiết lập giá trị cho thuộc tính Total.	Invoice.java (154)
20	sendEmailForCustomer(Invoice invoice)	Gửi email cho khách hàng về chi tiết đơn hàng họ đã mua.	Invoice.java (158)

**Bảng 12. Các phương thức của lớp Invoice**

### Phương thức của lớp UserSpending

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getUser() Input: Không có Output: User	Trả về giá trị của thuộc tính User.	UserSpending.java (23)
2	setUser(User user) Input: User Output: Không có	Thiết lập giá trị cho thuộc tính User.	UserSpending.java (27)
3	getSpending() Input: Không có Output: Spending	Trả về giá trị của thuộc tính Spending.	UserSpending.java (31)
4	setSpending(double spending) Input: Spending Output: Không có	Thiết lập giá trị cho thuộc tính Spending.	UserSpending.java (35)

**Bảng 13. Các phương thức của lớp UserSpending**

### Phương thức của lớp SoldShoes

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getShoes() Input: Không có Output: shoes	Trả về giá trị của thuộc tính shoes.	SoldShoes.java (23)
2	setShoes(Shoes shoes) Input: shoes Output: Không có	Thiết lập giá trị cho thuộc tính shoes.	SoldShoes.java (27)
3	getCount() Input: Không có Output: count	Trả về giá trị của thuộc tính count.	SoldShoes.java (31)
4	setCount(long count) Input: count Output: Không có	Thiết lập giá trị cho thuộc tính count.	SoldShoes.java (35)

**Bảng 14. Các phương thức của lớp SoldShoes**

### 2.2.3. Controller layer

Chứa các lớp xử lý các yêu cầu từ người dùng, điều hướng và gọi các hàm xử lý nghiệp vụ. Các lớp như AddController, DeleteController đều có chức năng cụ thể nhằm xử lý các loại yêu cầu khác nhau từ giao diện người dùng.

TT	Tên lớp	Mục đích
1	LoginController	Quản lý các hoạt động liên quan đến đăng nhập, đăng ký, quên mật khẩu, chỉnh sửa thông tin người dùng và đăng xuất trong ứng dụng web.
2	LoadAdminController	Xử lý các yêu cầu liên quan đến việc hiển thị thông tin và quản lý dữ liệu trong giao diện quản trị của ứng dụng web.
3	SearchingAdmin	Xử lý các chức năng tìm kiếm và thống kê trong trang quản lý của admin.
4	AddController	Xử lý logic điều hướng và xử lý dữ liệu liên quan đến thêm người dùng, danh mục, mã khuyến mãi, và sản phẩm trong mục admin.
5	DeleteController	Xử lý việc xóa dữ liệu người dùng, danh mục, khuyến mãi và sản phẩm trong mục admin.
6	EditController	Xử lý các yêu cầu liên quan đến việc chỉnh sửa thông tin người dùng, danh mục, khuyến mãi và sản phẩm trong mục admin.
7	ExportController	Xử lý yêu cầu xuất báo cáo dưới dạng file Excel (.xls)
8	LoadDetailInvoice	Xử lý yêu cầu hiển thị thông tin chi tiết của một hóa đơn trong giao diện quản trị.
9	LoadEditController	Xử lý yêu cầu hiển thị thông tin chi tiết của một hóa đơn khi thực hiện chỉnh sửa thông tin liên quan đến sản phẩm, người dùng, danh mục và khuyến mãi trong giao diện quản trị.
10	LoadMoreController	Xử lý các yêu cầu liên quan đến việc tải thêm dữ liệu sản phẩm trên trang chính của ứng dụng.

11	LoadingIndex	Chuẩn bị dữ liệu cần thiết cho trang index (trang chính) của ứng dụng web.
12	SearchingShop	Dùng để tìm kiếm các sản phẩm trong cửa hàng bằng cách nhập dữ liệu vào thanh tìm kiếm trên taskbar theo yêu cầu của khách hàng.
13	LoadProductShop	Dùng để load dữ liệu từ dưới cơ sở dữ liệu lên trang web hiển thị các sản phẩm theo từng tiêu chí yêu cầu của khách hàng.
14	LoadProductDetail	Dùng để load dữ liệu thông tin của một sản phẩm lên trang sản phẩm chi tiết để khách hàng xem chi tiết hơn.
15	SendEmailController	Xử lý yêu cầu liên quan đến việc gửi email từ hệ thống đến một danh sách người dùng được chọn.
16	CartController	Xử lý các yêu cầu của người dùng đối với giỏ hàng như: thêm vào giỏ hàng, tăng giảm số lượng mua của 1 sản phẩm, xóa sản phẩm. Xử lý các yêu cầu chuyển hướng của người dùng từ các trang khác đến giỏ hàng.
17	PaymentController	Xử lý các yêu cầu thanh toán của người dùng, thanh toán bằng tiền mặt hoặc thanh toán bằng tài khoản ngân hàng (vnpay).
18	CheckoutOnlineController	Xử lý hóa đơn khi người dùng yêu cầu thanh toán bằng tài khoản ngân hàng (vnpay)
19	ajaxServlet	Xử lý việc thanh toán bằng tài khoản ngân hàng, lấy dữ liệu và chuyển hướng sang trang thanh toán của vnpay.

**Bảng 15. Các lớp của Controller Layer**

#### 2.2.4. Data layer

Chứa các lớp liên quan đến việc truy xuất và thao tác dữ liệu. Các lớp này tương tác trực tiếp với cơ sở dữ liệu, như CategoryDB, UserDB, thực hiện các thao tác CRUD (Create, Read, Update, Delete).

TT	Tên lớp	Mục đích
1	DBUtil	Tạo các EntityManager để tương tác với cơ sở dữ liệu thông qua JPA.
2	UserDB	Thực hiện các thao tác liên quan đến cơ sở dữ liệu đối với đối tượng User trong ứng dụng.
3	CategoryDB	Quản lý các thao tác cơ sở dữ liệu liên quan đến các danh mục giày.
4	ShoesDB	Thực hiện các thao tác liên quan đến cơ sở dữ liệu đối với đối tượng Shoes trong ứng dụng.
5	LineItemDB	Thực hiện các thao tác liên quan đến cơ sở dữ liệu đối với đối tượng LineItem trong ứng dụng.
6	CartDB	Thực hiện các thao tác liên quan đến cơ sở dữ liệu cho đối tượng Cart trong ứng dụng.
7	PromotionDB	Thực hiện các thao tác liên quan đến cơ sở dữ liệu đối với đối tượng Promotion trong ứng dụng.
8	InvoiceDB	Thực hiện các thao tác liên quan đến cơ sở dữ liệu đối với đối tượng Invoice trong ứng dụng.
9	ReportDB	Tạo báo cáo về các hóa đơn trong một khoảng thời gian cụ thể.
10	Config	Phục vụ cho việc thanh toán thông qua vnpay. (Cung cấp dữ liệu cho lớp ajaxServlet).

**Bảng 16. Các lớp của Data Layer**

### Phương thức của lớp DBUtil

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getEmFactory() Input: Không có Output: EntityManagerFactory	Được sử dụng để tạo và quản lý các EntityManager, làm cầu nối giữa ứng dụng Java và cơ sở dữ liệu.	DBUtil.java (10)

**Bảng 17. Các phương thức của lớp DBUtil**

### Phương thức của lớp UserDB

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	insert(User user) Input: user Output: Không có	Chèn một đối tượng User mới vào cơ sở dữ liệu.	UserDB.java (25)
2	update(User user) Input: user Output: Không có	Cập nhật thông tin của một đối tượng User đã tồn tại trong cơ sở dữ liệu.	UserDB.java (39)
3	selectUser(String email) Input: email Output: User	Trả về thông tin người dùng dựa trên địa chỉ email.	UserDB.java (54)
4	selectAccount(String email, String password) Input: email, password Output: User	Xác minh đăng nhập của người dùng bằng cách kiểm tra địa chỉ email và mật khẩu.	UserDB.java (72)
5	emailExists(String email) Input: email Output: True/False	Kiểm tra xem một địa chỉ email đã tồn tại trong cơ sở dữ liệu hay chưa.	UserDB.java (91)
6	accountExists(String email, String password) Input: email, password Output: True/False	Kiểm tra xem một tài khoản với địa chỉ email và mật khẩu đã tồn tại trong cơ sở dữ liệu hay chưa.	UserDB.java (96)

7	selectUsers() Input: Không có Output: List<User>	Trả về danh sách tất cả người dùng trong cơ sở dữ liệu.	UserDB.java (101)
8	deleteUserById(long userID) Input: userID Output: Không có	Xóa một người dùng khỏi cơ sở dữ liệu dựa trên ID của họ.	UserDB.java (117)
9	selectUserById(long userID) Input: userID Output: User	Ấy thông tin của một người dùng dựa trên ID của họ.	UserDB.java (138)
10	getUserByPage(int pageNumber, int pageSize) Input: pageNumber, pageSize Output: List<User>	Lấy danh sách người theo số trang và kích thước trang cụ thể.	UserDB.java (149)
11	getUserCount() Input: Không có Output: long	Đếm số lượng người dùng hiện có trong cơ sở dữ liệu.	UserDB.java (167)
12	searchUserByEmail(String email) Input: email Output: List<User>	Tìm kiếm người dùng dựa trên địa chỉ email.	UserDB.java (179)
13	getUsersSpending() Input: Không có Output: List<UserSpending>	Lấy danh sách người dùng và tổng số tiền mà họ đã chi tiêu, sắp xếp theo tổng số tiền giảm dần.	UserDB.java (193)
14	getUserSpendingCountByInterval(String startDate, String endDate) Input: startDate, endDate Output: List<UserSpending>	Lấy danh sách người dùng và tổng số tiền mà họ đã chi tiêu trong một khoảng thời gian cụ thể, sắp xếp theo tổng số tiền giảm dần.	UserDB.java (207)

**Bảng 18. Các phương thức của lớp UserDB**

### Phương thức của lớp CategoryDB

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	selectCategories() Input: Không có Output: List<CategoryShoes>	Lấy danh sách tất cả các danh mục (CategoryShoes) từ cơ sở dữ liệu.	CategoryDB.java (21)
2	getCategoryShoesById(int categoryId) Input: categoryId Output: CategoryShoes	Lấy đối tượng CategoryShoes từ cơ sở dữ liệu dựa trên ID của danh mục.	CategoryDB.java (37)
3	insertCategory(CategoryShoes category) Input: category Output: Không có	Thêm đối tượng CategoryShoes mới vào cơ sở dữ liệu.	CategoryDB.java (46)
4	updateCategory(CategoryShoes category) Input: category Output: Không có	Cập nhật thông tin của một đối tượng CategoryShoes đã tồn tại trong cơ sở dữ liệu.	CategoryDB.java (60)
5	deleteCategory(CategoryShoes category) Input: category Output: Không có	Xóa đối tượng CategoryShoes khỏi cơ sở dữ liệu.	CategoryDB.java (75)

**Bảng 19. Các phương thức của lớp CategoryDB**



### Phương thức của lớp ShoesDB

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	insertShoes(Shoes shoes) Input: shoes Output: Không có	Chèn thông tin về giày (Shoes) mới vào cơ sở dữ liệu.	ShoesDB.java (25)
2	updateShoes(Shoes shoes) Input: shoes Output: Không có	Cập nhật thông tin về giày (Shoes) trong cơ sở dữ liệu.	ShoesDB.java (39)
3	deleteShoes(Shoes shoes) Input: shoes Output: Không có	Xóa thông tin về giày (Shoes) khỏi cơ sở dữ liệu.	ShoesDB.java (54)
4	selectShoes() Input: Không có Output: List<Shoes>	Truy vấn và trả về tất cả thông tin về giày từ cơ sở dữ liệu.	ShoesDB.java (70)
5	selectShoesByCategory(CategoryShoes category) Input: category Output: List<Shoes>	Truy vấn và trả về danh sách giày thuộc một danh mục cụ thể.	ShoesDB.java (86)
6	deleteShoesById(int shoesId) Input: shoesId Output: Không có	Xóa thông tin về giày theo ID từ cơ sở dữ liệu.	ShoesDB.java (100)
7	selectShoesById(int shoesId) Input: shoesId Output: Shoes	Truy vấn và trả về thông tin về giày dựa trên ID.	ShoesDB.java (122)
8	getShoesByPageCategory(CategoryShoes category, int pageNumber, int pageSize) Input: category, pageNumber, pageSize Output: List<Shoes>	Truy vấn và trả về danh sách giày theo danh mục, trang và kích thước trang.	ShoesDB.java (138)
9	getShoesByPageSize(double inputSize, int pageNumber, int pageSize) Input: inputSize, pageNumber, pageSize Output: List<Shoes>	Truy vấn và trả về danh sách giày theo size giày và kích thước trang.	ShoesDB.java (175)

10	getShoesByPage(int pageNumber, int pageSize) Input: pageNumber, pageSize Output: List<Shoes>	Truy vấn và trả về danh sách giày theo trang và kích thước trang.	ShoesDB.java (194)
11	getProductCount() Input: Không có Output: long	Truy vấn và trả về tổng số lượng sản phẩm giày có trong cơ sở dữ liệu.	ShoesDB.java (212)
12	searchShoesByName(String shoeName) Input: shoeName Output: List<Shoes>	Truy vấn và trả về danh sách giày dựa trên tên giày tương tự với tên đã cho.	ShoesDB.java (224)
13	getShoesBySize(double inputSize) Input: inputSize Output: List<Shoes>	Truy vấn và trả về danh sách giày có kích thước (inputSize).	ShoesDB.java (248)
14	getTopProducts(int top) Input: top Output: List<Shoes>	Truy vấn và trả về danh sách các giày mới nhất theo số lượng được yêu cầu.	ShoesDB.java (262)
15	getTopNewProductNext(int top, int offset) Input: top, offset Output: List<Shoes>	Truy vấn và trả về danh sách giày mới nhất theo số lượng và vị trí bắt đầu.	ShoesDB.java (281)
16	getTopBestSellingShoes(int top) Input: top Output: List<Shoes>	Truy vấn và trả về danh sách giày bán chạy nhất theo số lượng được yêu cầu.	ShoesDB.java (301)
17	getTopBestSellingShoesNext(int top, int offset) Input: top, offset Output: List<Shoes>	Truy vấn và trả về danh sách giày bán chạy nhất theo số lượng và vị trí bắt đầu.	ShoesDB.java (327)
18	getTopNewBrand(int top, int categoryID) Input: top, categoryID Output: List<Shoes>	Truy vấn và trả về danh sách giày mới nhất của một danh mục cụ thể.	ShoesDB.java (359)

19	getTopNewBrandNext(int top, int offset, int categoryID) Input: top, offset, categoryID Output: List<Shoes>	Truy vấn và trả về danh sách giày mới nhất của một danh mục cụ thể theo số lượng và vị trí bắt đầu.	ShoesDB.java (379)
20	getSellingShoesCount() Input: Không có Output: List<SoldShoes>	Truy vấn và trả về danh sách số lượng giày đã bán theo sản phẩm.	ShoesDB.java (400)
21	getSellingShoesCountByInterval(String startDate, String endDate) Input: startDate, endDate Output: List<SoldShoes>	Truy vấn và trả về danh sách số lượng giày đã bán trong khoảng thời gian đã cho.	ShoesDB.java (416)
22	getShoesByPrice(Long minPrice, Long maxPrice) Input: minPrice, maxPrice Output: List<Shoes>	Truy vấn và trả về danh sách giày trong khoảng giá đã cho.	ShoesDB.java (442)
23	getShoesByPagePrice(Long minPrice, Long maxPrice, int pageNumber, int pageSize) Input: minPrice, maxPrice, pageNumber, pageSize Output: List<Shoes>	Truy vấn và trả về danh sách giày trong khoảng giá đã cho theo trang và kích thước trang.	ShoesDB.java (459)

**Bảng 20. Các phương thức của lớp ShoesDB**

### Phương thức của lớp LineItemDB

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	insertLineItem(LineItem item) Input: item Output: Không có	Chèn một đối tượng LineItem mới vào cơ sở dữ liệu.	LineItemDB.java (22)
2	updateLineItem(LineItem item) Input: item Output: Không có	Cập nhật thông tin của một đối tượng LineItem đã tồn tại trong cơ sở dữ liệu.	LineItemDB.java (36)
3	deleteLineItem(LineItem item) Input: item Output: Không có	Xóa một đối tượng LineItem khỏi cơ sở dữ liệu.	LineItemDB.java (51)
4	selectLineItemById(Long itemID) Input: itemID Output: LineItem	Lấy thông tin của một LineItem dựa trên ID của nó từ cơ sở dữ liệu.	LineItemDB.java (67)

**Bảng 21. Các phương thức của lớp LineItemDB**

### Phương thức của lớp CartDB

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	insertCart(Cart cart) Input: cart Output: Không có	Chèn một đối tượng Cart mới vào cơ sở dữ liệu.	CartDB.java (22)
2	updateCart(Cart cart) Input: cart Output: Không có	Cập nhật thông tin của một đối tượng Cart đã tồn tại trong cơ sở dữ liệu.	CartDB.java (36)
3	deleteCart(Cart cart) Input: cart Output: Không có	Xóa một đối tượng Cart khỏi cơ sở dữ liệu	CartDB.java (51)
4	deleteCartById(Long cartID) Input: cartID Output: Không có	Xóa một đối tượng Cart khỏi cơ sở dữ liệu dựa trên ID của nó.	CartDB.java (67)

5	selectCartById(Long cartID) Input: cartID Output: Không có	Lấy thông tin của một đối tượng Cart dựa trên ID của nó từ cơ sở dữ liệu.	CartDB.java (89)
---	--	---	------------------

**Bảng 22. Các phương thức của lớp CartDB**

**Phương thức của lớp PromotionDB**

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	selectPromotions() Input: Không có Output: List<PromotionCode>	Lấy danh sách tất cả các mã khuyến mãi từ cơ sở dữ liệu.	PromotionDB.java (20)
2	insert(PromotionCode promotion) Input: promotion Output: Không có	Thêm đối tượng PromotionCode mới vào cơ sở dữ liệu.	PromotionDB.java (35)
3	update(PromotionCode promotion) Input: promotion Output: Không có	Cập nhật thông tin của một đối tượng PromotionCode đã tồn tại trong cơ sở dữ liệu.	PromotionDB.java (49)
4	selectPromotionByCode (String promotionCode) Input: promotionCode Output: PromotionCode	Tìm kiếm và lấy một đối tượng PromotionCode dựa trên mã khuyến mãi từ cơ sở dữ liệu.	PromotionDB.java (63)
5	deletePromotionByCode (String promotionCode) Input: promotionCode Output: Không có	Xóa một đối tượng PromotionCode khỏi cơ sở dữ liệu dựa trên mã khuyến mãi.	PromotionDB.java (74)

**Bảng 23. Các phương thức của lớp PromotionDB**

**Phương thức của lớp InvoiceDB**

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	insert(Invoice invoice) Input: invoice Output: Không có	Chèn một đối tượng Invoice mới vào cơ sở dữ liệu.	InvoiceDB.java (25)
2	selectsInvoices() Input: Không có Output: List<Invoice>	Trả về danh sách tất cả hóa đơn từ cơ sở dữ liệu hoặc null nếu không có dữ liệu.	InvoiceDB.java (39)
3	getInvoiceById(long invoiceID) Input: invoiceID Output: Invoice	Lấy thông tin của một hóa đơn dựa trên ID của nó từ cơ sở dữ liệu.	InvoiceDB.java (55)
4	getInvoiceByInterval(String startDate, String endDate) Input: startDate, endDate Output: List<Invoice>	Lấy một hóa đơn dựa trên người dùng và mã khuyến mãi cụ thể từ cơ sở dữ liệu.	InvoiceDB.java (71)
5	getInvoiceByUserPromotion(User user, String promotionCode) Input: user, promotionCode Output: Invoice	Lấy một hóa đơn dựa trên người dùng và mã khuyến mãi cụ thể từ cơ sở dữ liệu.	InvoiceDB.java (96)

**Bảng 24. Các phương thức của lớp InvoiceDB****Phương thức của lớp ReportDB**

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getInvoiceReport(String startDate, String endDate) Input: startDate, endDate Output: Workbook	Truy vấn và tạo báo cáo hóa đơn dựa trên khoảng thời gian đã cho. Tích hợp dữ liệu từ cơ sở dữ liệu (danh sách hóa đơn) vào một bảng Excel để xuất báo cáo.	ReportDB.java (17)

**Bảng 25. Các phương thức của lớp ReportDB**

### 2.2.5. Util

TT	Tên lớp	Mục đích
1	CookieUtil	Lấy giá trị của một cookie cụ thể thông qua tên của nó từ một mảng các cookie.
2	EmailUtil	Cung cấp các phương thức hỗ trợ việc gửi email trong ứng dụng.

**Bảng 26. Các lớp của Util**

#### Phương thức của lớp CookieUtil

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	getCookieValue(Cookie[] cookies, String cookieName) Input: cookies, cookieName Output: String	Trích xuất giá trị của một cookie cụ thể từ danh sách các cookie có sẵn.	CookieUtil.java (7)

**Bảng 27. Các phương thức của lớp CookieUtil**

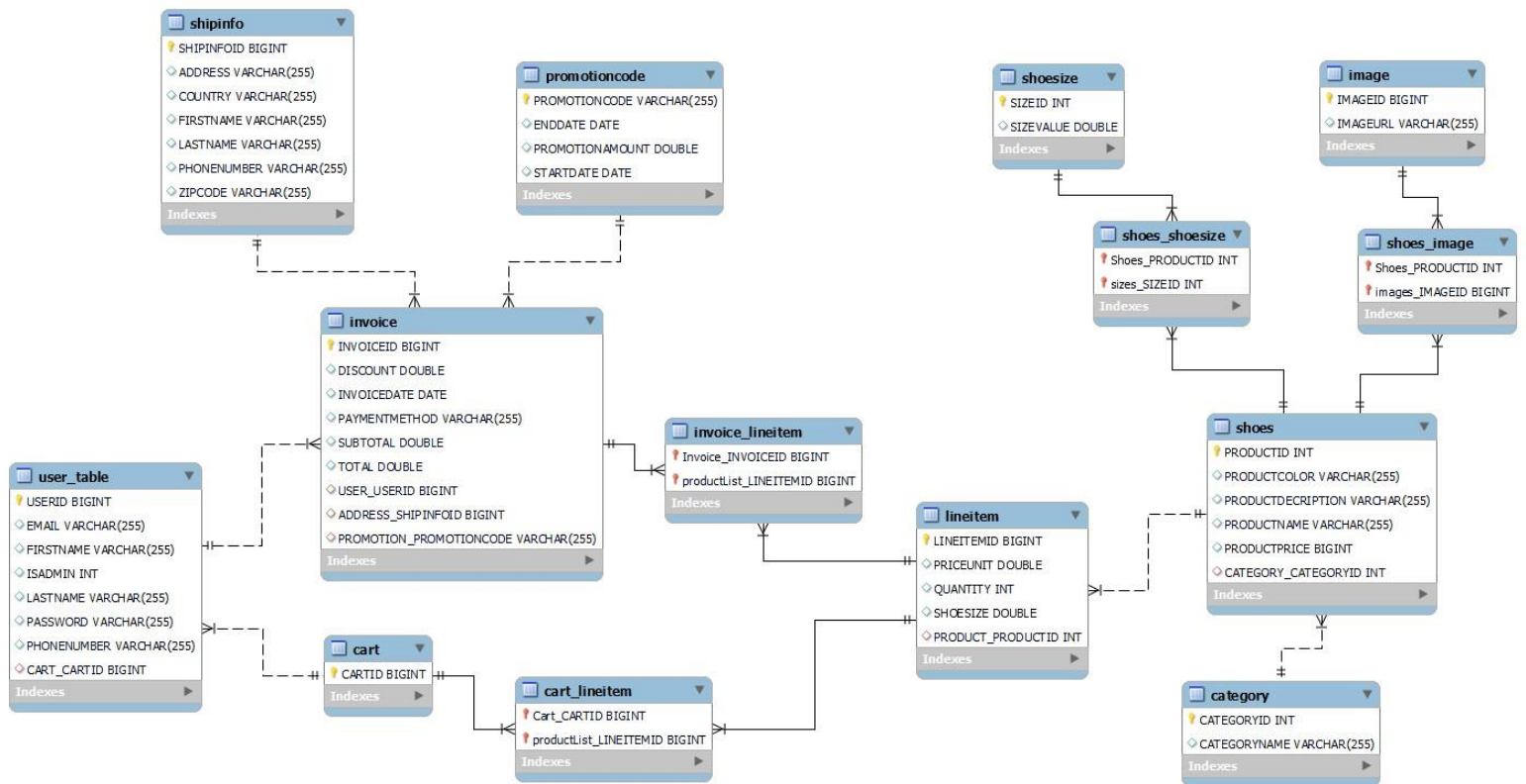
#### Phương thức của lớp EmailUtil

TT	Phương thức	Mục đích	Tên file, số thứ tự dòng chứa khai báo.
1	sendEmail(String to, String tieuDe, String noiDung) Input: to, tieuDe, noiDung Output: Không có	Gửi email từ địa chỉ email được định nghĩa trong class đến địa chỉ email được chỉ định với tiêu đề và nội dung cụ thể.	EmailUtil.java (29)

**Bảng 28. Các phương thức của lớp EmailUtil**

## 2.3. Thiết kế cơ sở dữ liệu

### 2.3.1. Hình vẽ



Hình 2. Các mối quan hệ giữa các bảng trong cơ sở dữ liệu



### 2.3.2. Bảng mô tả các Table trong cơ sở dữ liệu

TT	Tên bảng	Mục đích
1	shipinfo	Lưu trữ thông tin địa chỉ và thông tin khách hàng trên hoá đơn của người dùng khi mua hàng.
2	cart	Lưu lại các sản phẩm mà người dùng đã bỏ vào giỏ hàng.
3	cart_lineitem	Lưu trữ các mối quan hệ giữa giỏ hàng (cart) và các mục sản phẩm (lineitem). Nó cho phép theo dõi và quản lý các sản phẩm được thêm vào giỏ hàng.
4	category	Lưu trữ thông tin về các nhóm sản phẩm trong hệ thống. Các danh mục này được sử dụng để phân loại và tổ chức sản phẩm, giúp dễ dàng tìm kiếm và duyệt qua các sản phẩm cùng loại.
5	image	Lưu trữ URL hình ảnh của sản phẩm.
6	invoice	Lưu trữ thông tin về các hóa đơn (invoices) trong hệ thống.
7	invoice_lineitem	Được sử dụng để xác định mối quan hệ giữa một hóa đơn cụ thể và các mục sản phẩm liên quan đến hóa đơn đó.
8	lineitem	Lưu trữ chi tiết về các mục sản phẩm được thêm vào một hóa đơn (Invoice) hoặc giỏ hàng (Cart)
9	promotioncode	Lưu trữ thông tin về các mã khuyến mãi trong hệ thống. Các mã khuyến mãi này có thể được áp dụng cho các hóa đơn (invoices) để cung cấp giảm giá cho người mua hàng.
10	shoes	Lưu trữ thông tin cơ bản về sản phẩm giày, bao gồm tên sản phẩm, mô tả, màu sắc, giá và danh mục của sản phẩm.
11	shoes_image	Được sử dụng để xác định mối quan hệ giữa sản phẩm giày và các hình ảnh của nó.
12	shoes_shoesize	Được sử dụng để xác định mối quan hệ giữa một sản phẩm giày cụ thể và các kích cỡ giày mà sản phẩm đó có sẵn.
13	shoesize	Lưu trữ thông tin về các kích cỡ giày khả dụng trong hệ thống. Các kích cỡ giày này có thể được liên kết với các sản phẩm giày trong bảng khác, giúp xác định những kích cỡ giày nào có sẵn cho từng sản phẩm.
14	user_table	Lưu trữ thông tin cá nhân của người dùng trong hệ thống, bao gồm thông tin liên hệ, quyền truy cập (nếu có), và thông tin đăng nhập.

**Bảng 29. Các bảng trong cơ sở dữ liệu**

### 2.3.3. Bảng các Fields trong 1 Table

**Bảng shipinfo**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	SHIPINFOID	BIGINT	Mã định danh duy nhất cho mỗi thông tin nhận hàng.
2	ADDRESS	VARCHAR(255)	Địa chỉ giao hàng.
3	COUNTRY	VARCHAR(255)	Quốc gia.
4	FIRSTNAME	VARCHAR(255)	Tên của người nhận.
5	LASTNAME	VARCHAR(255)	Họ của người nhận.
6	PHONENUMBER	VARCHAR(255)	Số điện thoại của người nhận.
7	ZIPCODE	VARCHAR(255)	Mã bưu chính của đơn hàng.

**Bảng 30. Các Fields của bảng shipinfo**

**Bảng cart**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	CARTID	BIGINT	Mã định danh duy nhất cho mỗi giỏ hàng

**Bảng 31. Các Fields của bảng cart**

**Bảng cart\_lineitem**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	Cart_CARTID	BIGINT	Dùng để tham chiếu khóa ngoại
2	productList_LINEITEMID	BIGINT	Dùng để tham chiếu khóa ngoại

**Bảng 32. Các Fields của bảng cart\_lineitem**

**Bảng category**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	CATEGORYID	INT	Mã định danh duy nhất cho danh mục sản phẩm
2	CATEGORYNAME	VARCHAR(255)	Tên danh mục sản phẩm

**Bảng 33. Các Fields của bảng category**

### Bảng image

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	IMAGEID	BIGINT	Mã định danh duy nhất cho mỗi hình ảnh
2	IMAGEURL	VARCHAR(255)	Đường dẫn đến nơi lưu trữ, nguồn cung cấp hình ảnh

**Bảng 34. Các Fields của bảng image**

### Bảng invoice

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	INVOICEID	BIGINT	Mã định danh duy nhất cho mỗi hóa đơn
2	INVOICEDATE	DATE	Ngày tạo hóa đơn
3	PAYMENTMETHOD	VARCHAR(255)	Phương thức thanh toán
4	ADDRESS_ADDRESSID	BIGINT	Khóa ngoại liên quan đến đối tượng Address
5	SUBTOTAL	DOUBLE	Tổng giá trị của các sản phẩm trong hóa đơn khi chưa áp dụng khuyến mại
6	DISCOUNT	DOUBLE	Khuyến mại giảm giá
7	TOTAL	DOUBLE	Tổng giá trị cuối cùng của hóa đơn sau áp dụng khuyến mại (nếu có)
8	USER_USERID	BIGINT	Khóa ngoại liên quan đến đối tượng User
9	PROMOTION_PROMOTIONCODE	VARCHAR(255)	Khóa ngoại liên quan đến đối tượng Promotion

**Bảng 35. Các Fields của bảng invoice**

### Bảng invoice\_lineitem

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	Invoice_INVOICEID	BIGINT	Dùng để tham chiếu khóa ngoại
2	productList_LINEITEMID	BIGINT	Dùng để tham chiếu khóa ngoại

**Bảng 36. Các Fields của bảng invoice\_lineitem**

**Bảng lineitem**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	LINEITEMID	BIGINT	Mã định danh duy nhất cho mỗi mục chi tiết đơn hàng
2	QUANTITY	INT	Số lượng sản phẩm tương ứng trong mục chi tiết
3	PRICEUNIT	BIGINT	Giá của một đơn vị sản phẩm trong mục chi tiết
4	PRODUCT_PRODUCTID	INT	Sử dụng các thuộc tính ở bảng Product
5	SHOESIZE	DOUBLE	Kích cỡ của sản phẩm tương ứng cho mỗi mục chi tiết đơn hàng

**Bảng 37. Các Fields của bảng lineitem****Bảng promotion**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	PROMOTIONCODE	VARCHAR(255)	Mã định danh duy nhất cho mỗi chương trình khuyến mại
2	PROMOTIONAMOUNT	DOUBLE	Phần trăm giảm giá áp dụng cho chương trình khuyến mại
3	STARTDATE	DATE	Ngày bắt đầu khuyến mại
4	ENDDATE	DATE	Ngày kết thúc khuyến mại

**Bảng 38. Các Fields của bảng promotion****Bảng shoes**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	PRODUCTID	INT	Mã định danh duy nhất cho mỗi sản phẩm
2	PRODUCTNAME	VARCHAR(255)	Tên của sản phẩm
3	CATEGORY_CATEGORYID	INT	Danh mục sản phẩm
4	PRODUCTCOLOR	VARCHAR(255)	Màu sắc của sản phẩm
5	PRODUCTDESCRIPTION	VARCHAR(255)	Mô tả chi tiết sản phẩm
6	PRODUCTPRICE	DOUBLE	Giá cả của sản phẩm

**Bảng 39. Các Fields của bảng shoes**

**Bảng shoes\_image**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	Shoes_PRODUCTID	INT	Dùng để tham chiếu khóa ngoại
2	Images_IMAGEID	BIGINT	Dùng để tham chiếu khóa ngoại

**Bảng 40. Bảng các Fields của shoes\_image****Bảng shoes\_shoesize**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	Shoes_PRODUCTID	INT	Xác định sản phẩm trong danh sách giày.
2	sizes_SIZEID	INT	Xác định kích thước của sản phẩm giày.

**Bảng 41. Bảng shoes\_shoesize****Bảng shoesize**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	SIZEID	INT	Mã định danh duy nhất cho mỗi kích thước.
2	SIZEVALUE	DOUBLE	Thông tin về giá trị cụ thể của kích thước.

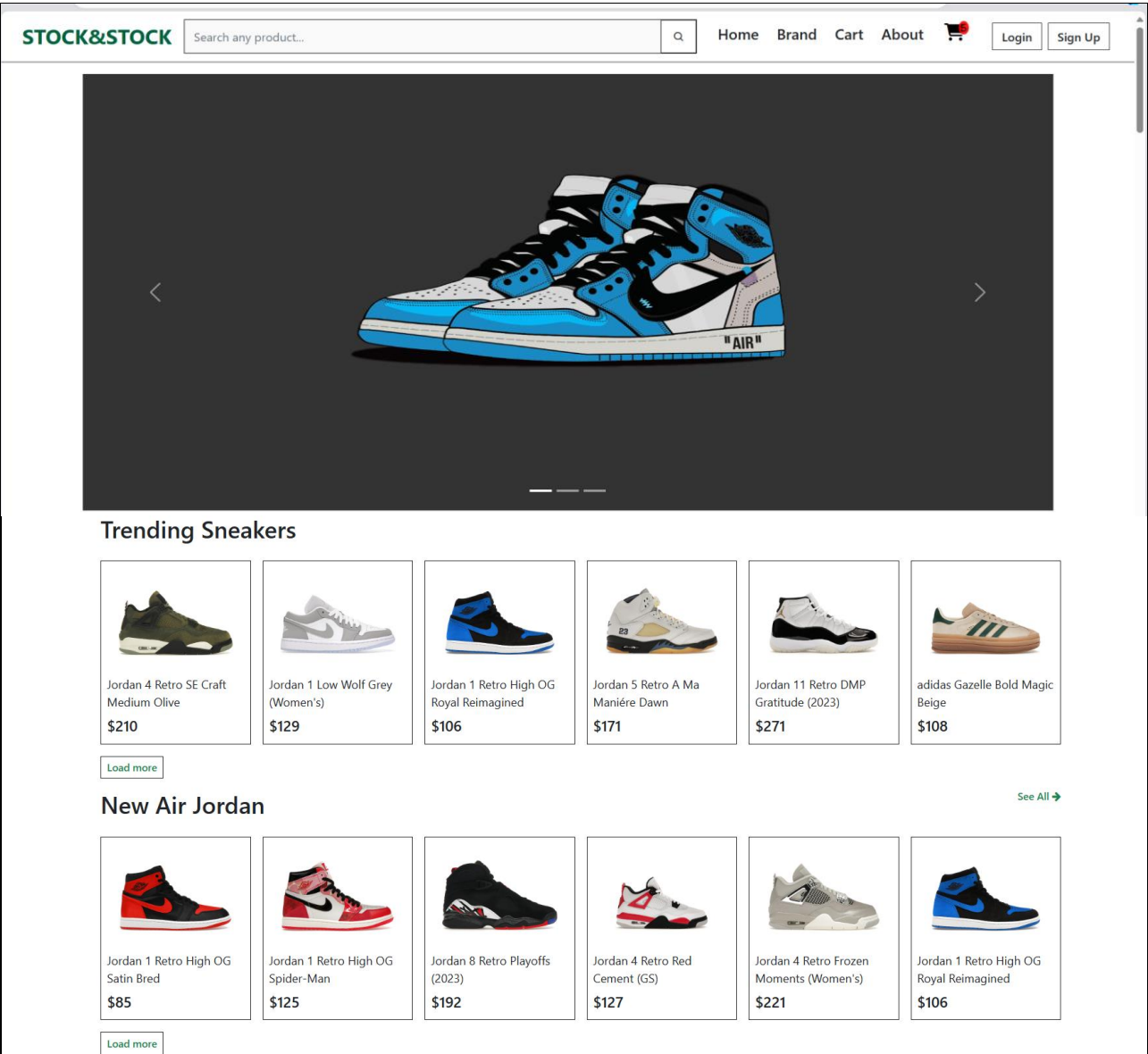
**Bảng 42. Các Fields của bảng shoesize****Bảng user\_table**

TT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	USERID	BIGINT	Mã định danh duy nhất cho mỗi khách hàng
2	EMAIL	VARCHAR(255)	Địa chỉ email của khách hàng
3	PASSWORD	VARCHAR(255)	Mật khẩu
4	FIRSTNAME	VARCHAR(255)	Tên người dùng
5	LASTNAME	VARCHAR(255)	Họ người dùng
6	PHONENUMBER	VARCHAR(255)	Số điện thoại
7	ISADMIN	INT	Vai trò của người dùng
8	CART_CARTID	BIGINT	Khóa ngoại liên quan đến đối tượng Cart

**Bảng 43. Các Fields của bảng user\_table**

2.4. Thiết kế giao diện

2.4.1. Giao diện trang chủ

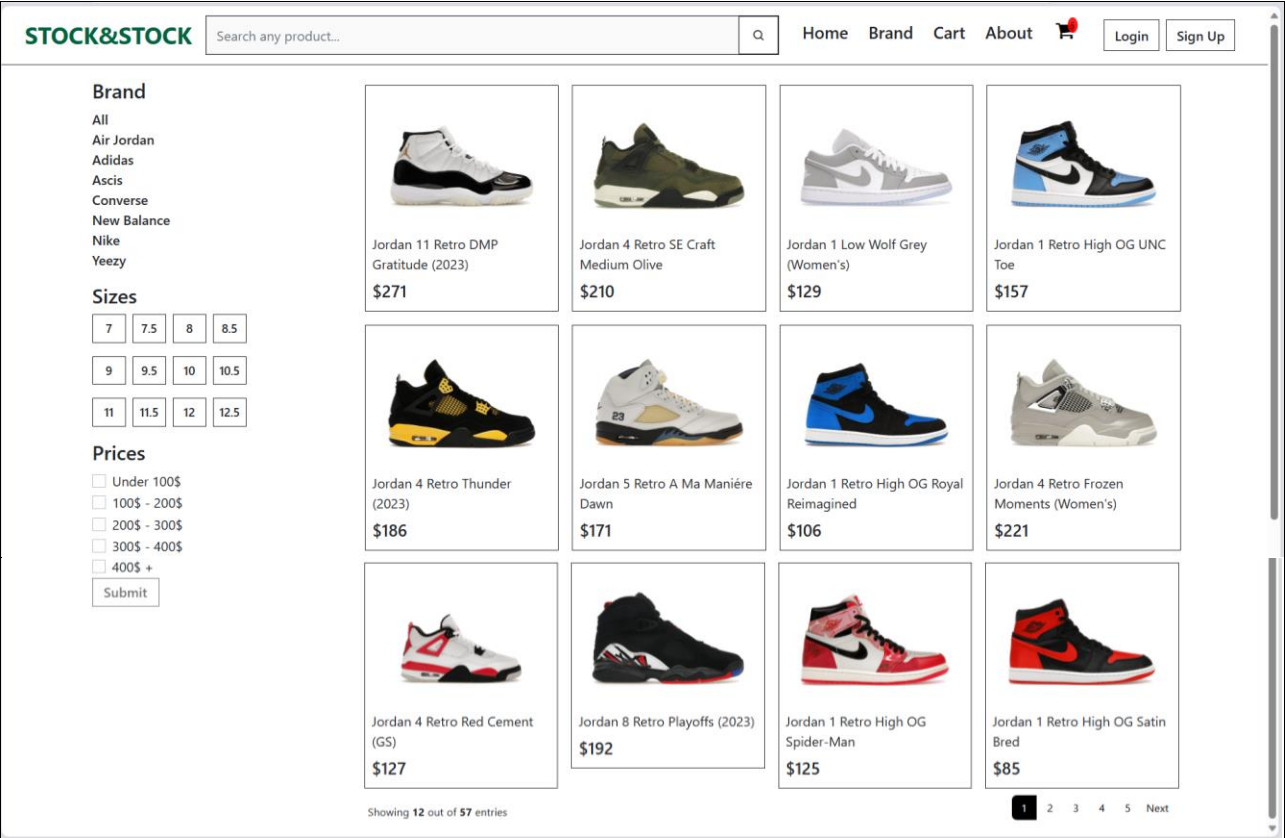


Hình 3. Giao diện trang chủ

Mục đích	Giải thích	Người thiết kế
Hiển thị logo của website, thanh tìm kiếm sản phẩm, nút đăng nhập và đăng ký.	<ul style="list-style-type: none"><li>• Khi người dùng vào trang web, giao diện này sẽ hiện ra.</li><li>• Người dùng có thể tìm kiếm sản phẩm theo tên ở thanh tìm kiếm.</li><li>• Người dùng có thể đăng nhập vào trang web</li></ul>	Đặng Minh Nhật và Đặng Phú Quý

<p>Hiển thị các sản phẩm bán chạy, sản phẩm mới đề xuất cho khách hàng, các mẫu giày của các hãng khác nhau.</p>	<p>để mua hàng, nếu chưa có tài khoản thì có thể đăng ký.</p> <ul style="list-style-type: none"> <li>• Người dùng có thể bấm “Load more” để xem thêm sản phẩm ở các mục khác nhau.</li> <li>• Người dùng có thể chọn “See all” để xem tất cả sản phẩm của hãng đó.</li> </ul>	
--	---	--

2.4.2. Giao diện trang xem sản phẩm

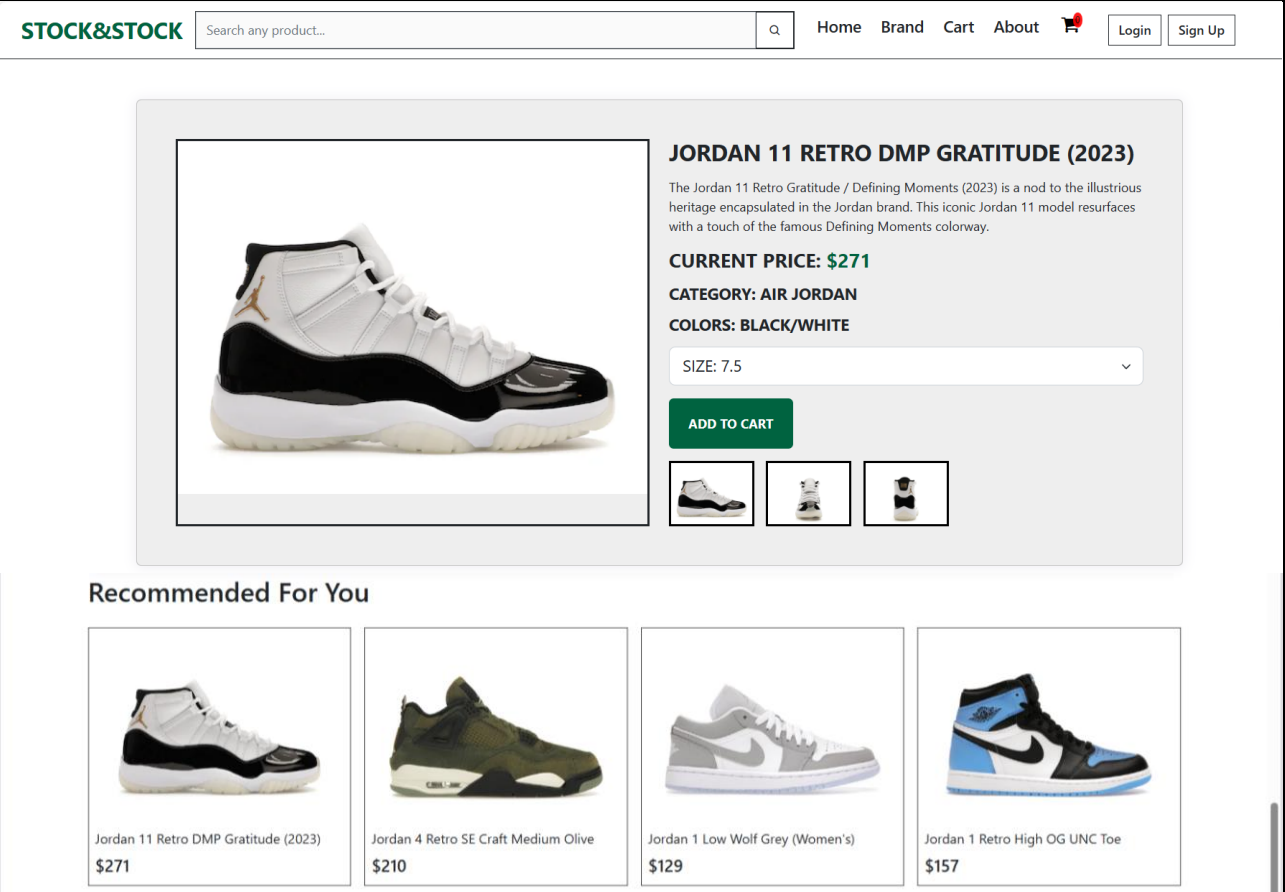


Hình 4. Giao diện trang xem sản phẩm

Mục đích	Giải thích	Người thiết kế
Hiển thị các sản phẩm của cửa hàng cho khách hàng xem hoặc tìm kiếm sản phẩm theo nhu cầu mình mong muốn.	<ul style="list-style-type: none"><li>• Khi người dùng chọn vào mục Brand trên thanh taskbar thì lúc này trang sẽ load lên toàn bộ sản phẩm của cửa hàng.</li><li>• Bên trái trang web là các tùy chọn lọc sản phẩm theo nhu cầu của khách: lọc theo hãng, lọc theo kích cỡ và lọc theo giá tiền.</li><li>• Bên phải trang web là danh sách các sản phẩm bao gồm 12 sản phẩm. Phía dưới danh sách là mục phân trang để qua trang mới để hiển thị thêm sản phẩm.</li></ul>	Đặng Minh Nhật và Trần Quốc Phương



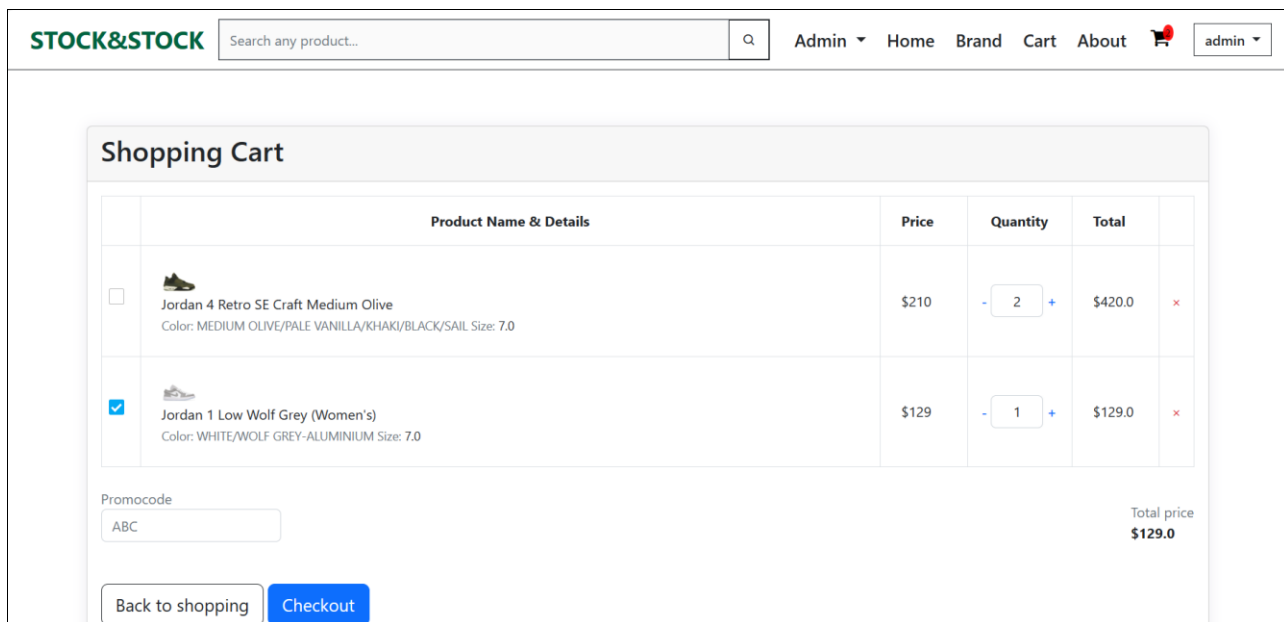
2.4.3. Giao diện chi tiết sản phẩm



Hình 5. Giao diện trang chi tiết sản phẩm

Mục đích	Giải thích	Người thiết kế
Hiện thị thông tin chi tiết của một sản phẩm khi người dùng nhấp chọn vào sản phẩm đó.	<ul style="list-style-type: none"><li>Phần trên là hình ảnh to hiển thị rõ sản phẩm, cùng theo đó là thông tin của sản phẩm bao gồm: tên, mô tả, giá tiền, loại, màu và kích cỡ mà sản phẩm đang có. Dưới thông tin là nút thêm giỏ hàng “ADD TO CART” và các hình ảnh khác của sản phẩm.</li><li>Phía dưới là loạt các sản phẩm có cùng loại với sản phẩm mà khách hàng đã chọn.</li></ul>	Đặng Minh Nhật và Trần Quốc Phương

### 2.4.4. Giao diện trang giỏ hàng



**Hình 6. Giao diện trang giỏ hàng**

Mục đích	Giải thích	Người thiết kế
Hiển thị những sản phẩm mà người dùng đã thêm vào giỏ hàng, cho phép người dùng nhập mã khuyến mãi để thanh toán hoặc tiếp tục mua sắm	<ul style="list-style-type: none"> <li>• Người dùng có thể tích vào những sản phẩm muốn thanh toán, “Total price” sẽ hiển thị tổng giá trị của những sản phẩm đó.</li> <li>• Người dùng nhập mã khuyến mãi vào “Promocode” để nhận được khuyến mãi.</li> <li>• Nếu người dùng muốn thanh toán thì bấm vào nút “Checkout”, nếu muốn tiếp tục mua sắm thì bấm vào nút “Back to shopping”.</li> </ul>	Đặng Minh Nhật

2.4.5. Giao diện trang thanh toán

STOCK&STOCK

Search any product...

Q

Admin

Home

Brand

Cart

About

admin

Billing address

First name

admin

Last name

Email

abc1@gmail.com

Phone Number

1234567890

Address

1 Vo Van Ngan, Thanh pho Thu Duc

Country

Viet Nam

Zip

Payment

☒ Cash in hand

☐ Bank transfer

Continue to checkout

Your cart

Jordan 4 Retro SE Craft Medium Olive

\$420.0

Size: 7.0, Quantity: 2

Jordan 1 Low Wolf Grey (Women's)

\$129.0

Size: 7.0, Quantity: 1

Promo code

NULL

-\$0.0

Total (USD)

\$549.0

Hình 7. Giao diện trang thanh toán

Mục đích	Giải thích	Người thiết kế
Hiển thị những món hàng được chọn để thanh toán ở bên “Your cart” và nơi để điền thông tin khách hàng, chọn phương thức thanh toán	<ul style="list-style-type: none"><li>Khách hàng kiểm tra lại thông tin những món hàng đã chọn để thanh toán cũng như mã khuyến mãi đã nhập, kiểm tra tổng tiền hiện ở phần “Total (USD)”.</li><li>Khách hàng điền các thông tin như: First name, Last name, Phone Number, Address, Country, Zip</li><li>Khách hàng chọn phương thức thanh toán bằng tiền mặt hoặc theo phương thức trực tuyến (thông qua vnpay).</li></ul>	Đặng Minh Nhật

## 2.4.6. Giao diện trang thanh toán vnpay

### VNPAY DEMO

---

#### Tạo mới đơn hàng

**Số tiền (\$)**

129.0

Chọn phương thức thanh toán

Chuyển hướng sang Cổng VNPay chọn phương thức thanh toán

☒ **Cổng thanh toán VNPAYQR**

Chọn ngôn ngữ giao diện thanh toán:


☒ **Tiếng việt**

☐ Tiếng anh

Thanh toán

---

© VNPAY 2020



Giao dịch hết hạn sau **14 : 52**

#### Thông tin đơn hàng

---

Số tiền thanh toán  
**2.967.000VND**

Giá trị đơn hàng  
**2.967.000VND**

Phí giao dịch  
**0VND**

Mã đơn hàng  
**83463484**


Nhà cung cấp  
**Công ty CTT HTT1**

#### Thanh toán qua Ngân hàng NCB

---

Thẻ nội địa


Số thẻ



Tên chủ thẻ

Ngày phát hành ⓘ


Mã khuyến mại


 Chọn hoặc nhập mã



[Điều kiện sử dụng dịch vụ](#)

Hủy thanh toán

Tiếp tục


 [1900.5555.77](tel:1900.5555.77)

 [hotrovnipay@vnpay.vn](mailto:hotrovnipay@vnpay.vn)



Hình 8. Giao diện trang thanh toán vnpay		
Mục đích	Giải thích	Người thiết kế
Hiển thị tạo đơn thanh toán bằng vnpay.	<ul style="list-style-type: none"> <li>Khách hàng kiểm tra số tiền thanh toán và giá trị đơn hàng.</li> <li>Khách hàng điền các thông tin tài khoản ngân hàng để tiến hành thanh toán.</li> </ul>	VNPAY

#### 2.4.7. Giao diện trang hóa đơn

<div> <div>STOCK&amp;STOCK</div> <div>Search any product...</div> <div>Q</div> <div>Admin ▾ Home Brand Cart About </div> <div>admin ▾</div> </div>																							
<div>Thank for your purchase</div> <div> <div>Name: Dang Nhat</div> <div>Invoice #1604</div> <div>Date: 12/12/2023</div> <div>Address: 1 Vo Van Ngan, Thanh pho Thu Duc, Viet Nam, 70000</div> </div> <table> <tr> <td>Jordan 4 Retro SE Craft Medium Olive</td><td>Quantity: 2</td><td>\$420.0</td></tr> <tr> <td>Size: 7.0</td><td></td><td></td></tr> <tr> <td>Jordan 1 Low Wolf Grey (Women's)</td><td>Quantity: 1</td><td>\$129.0</td></tr> <tr> <td>Size: 7.0</td><td></td><td></td></tr> <tr> <td colspan="2"></td><td>Sub total: \$549.0</td></tr> <tr> <td colspan="2"></td><td>Discount: -\$0.0</td></tr> <tr> <td colspan="2"></td><td><b>Total: \$549.0</b></td></tr> </table>			Jordan 4 Retro SE Craft Medium Olive	Quantity: 2	\$420.0	Size: 7.0			Jordan 1 Low Wolf Grey (Women's)	Quantity: 1	\$129.0	Size: 7.0					Sub total: \$549.0			Discount: -\$0.0			<b>Total: \$549.0</b>
Jordan 4 Retro SE Craft Medium Olive	Quantity: 2	\$420.0																					
Size: 7.0																							
Jordan 1 Low Wolf Grey (Women's)	Quantity: 1	\$129.0																					
Size: 7.0																							
		Sub total: \$549.0																					
		Discount: -\$0.0																					
		<b>Total: \$549.0</b>																					

Hình 9. Giao diện trang hóa đơn		
Mục đích	Giải thích	Người thiết kế
Hiển thị thông tin khách hàng để khách hàng xác nhận lại thông tin mua hàng, các sản phẩm đã mua, số tiền giảm giá và tổng tiền của đơn hàng.	<ul style="list-style-type: none"> <li>Cho khách hàng kiểm tra lại thông tin mua hàng như: tên, mã hóa đơn, ngày mua hàng, địa chỉ nhận hàng</li> <li>Cho khách hàng kiểm tra lại những món hàng đã mua như: size giày, số lượng, giá tiền.</li> <li>Cho khách hàng kiểm tra số tiền được giảm và tổng tiền của đơn hàng.</li> </ul>	Đặng Minh Nhật

#### 2.4.8. Giao diện đăng nhập và đăng ký tài khoản

## STOCK&STOCK

Sign Up
Log In

Email address\*  
 nhatdzfbk@gmail.com

We'll never share your email with anyone else.

Password\*  
 .....

☐ Remember me

Log In

[Forget Password?](#)

Continue with Google

Continue with Facebook

Need an account? [Sign Up](#)

## STOCK&STOCK

Sign Up
Log In

First Name\*

Last Name\*

Email Address\*  
 abc1@gmail.com

Phone Number\*

Password\*  
 .....

At least 8 characters, 1 uppercase letter, 1 number & 1 symbol

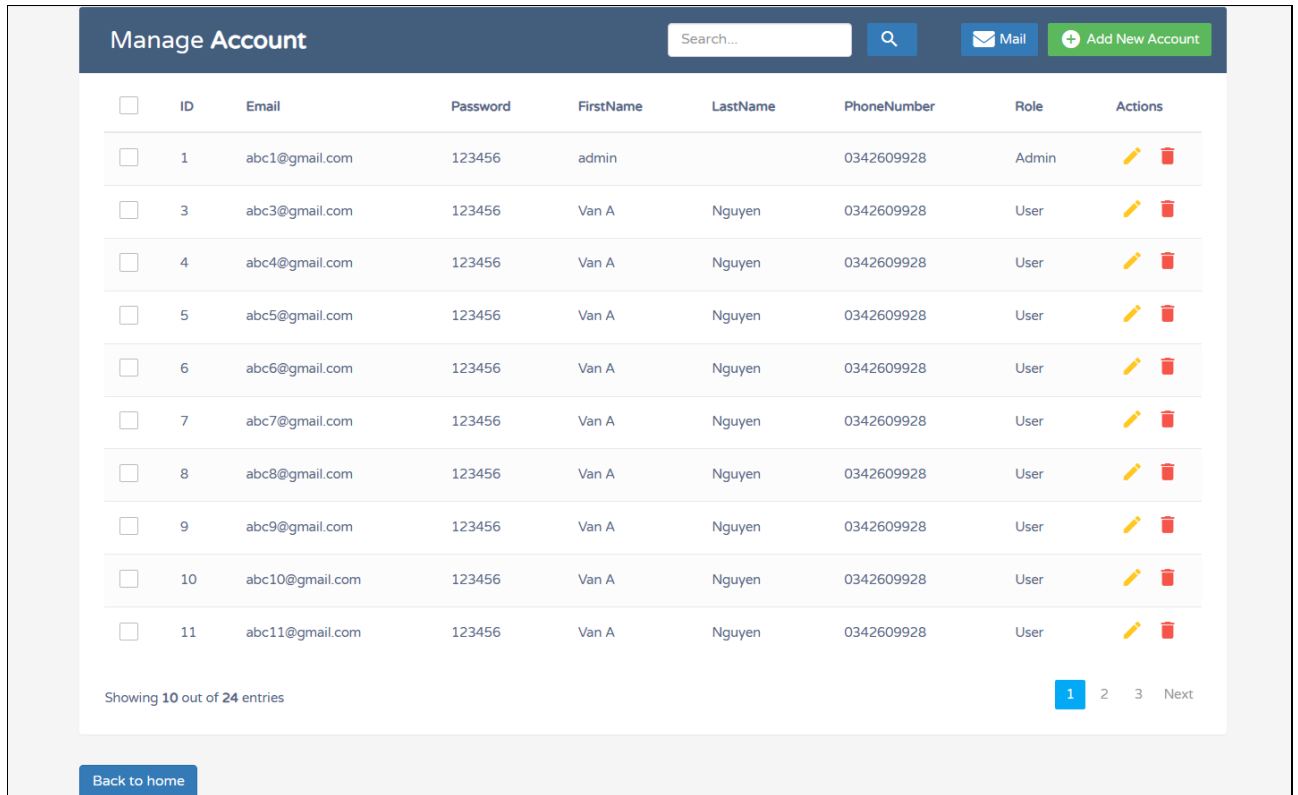
Sign Up

Already have an account? [Log In](#)

**Hình 10. Giao diện đăng nhập và đăng ký tài khoản**

Mục đích	Giải thích	Người thiết kế
Để người dùng có thể đăng nhập và đăng ký tài khoản. Bên cạnh đó người dùng có thể lưu lại mật khẩu hoặc khôi phục mật khẩu.	<ul style="list-style-type: none"> <li>• Nếu đã có tài khoản, ấn phần Log In và điền email và mật khẩu để đăng nhập.</li> <li>• Tích chọn lưu mật khẩu để lưu mật khẩu lại cho những lần đăng nhập sau.</li> <li>• Nếu chưa có tài khoản, ấn phần Sign Up và điền các thông tin cần thiết để tạo mới một tài khoản.</li> <li>• Ở phần Log In khi người dùng quên mật khẩu có thể ấn vào “Forget password” để tiến hành khôi phục lại mật khẩu.</li> </ul>	Đặng Minh Nhật

### 2.4.9. Giao diện trang quản lý tài khoản



**Hình 11. Giao diện trang quản lý tài khoản**

Mục đích	Giải thích	Người thiết kế
Giúp quản trị viên có thể quản lý được danh sách người dùng của hệ thống. Thực hiện các thao tác thêm, sửa, xóa và tìm kiếm thông tin với tài khoản người dùng.	<ul style="list-style-type: none"> <li>• Để thêm mới người dùng ấn vào nút “Add New Account”.</li> <li>• Để sửa tài khoản người dùng, tại tài khoản tương ứng chọn vào biểu tượng edit ở cột Action.</li> <li>• Để xóa tài khoản người dùng ấn vào biểu tượng xóa ở cột Action.</li> <li>• Để tìm kiếm tài khoản người dùng dựa trên email nhập vào thanh tìm kiếm chuỗi ký tự có liên quan và ấn vào biểu tượng tìm kiếm.</li> <li>• Để gửi mail cho người dùng, tích chọn một hay nhiều người dùng và ấn vào biểu tượng email ở đầu trang để gửi email cho người dùng.</li> </ul>	Đặng Phú Quý

#### 2.4.10. Giao diện trang thêm mới tài khoản

The image shows a web application interface for managing accounts. A modal window titled 'New Account' is open, allowing the creation of a new user. The modal contains the following fields:

- Email:
- Password:
- FirstName:
- LastName:
- PhoneNumber:
- Role: User (dropdown menu)

At the bottom of the modal are two buttons: 'Cancel' and 'Add'.

The background interface, titled 'Manage Account', displays a table of existing accounts. The table has columns for ID, Email, Phone Number, Role, and Actions. The data rows are as follows:

ID	Email	Phone Number	Role	Actions
1	abc1@gmail.com	342609928	Admin	
3	abc3@gmail.com	342609928	User	
4	abc4@gmail.com	342609928	User	
5	abc5@gmail.com	342609928	User	
6	abc6@gmail.com	342609928	User	
7	abc7@gmail.com	342609928	User	
8	abc8@gmail.com	342609928	User	
9	abc9@gmail.com	342609928	User	
10	abc10@gmail.com	342609928	User	
12	abc12@gmail.com	0342609928	User	

**Hình 12. Giao diện trang thêm mới tài khoản**

Mục đích	Giải thích	Người thiết kế
Cho phép quản trị viên tạo mới một tài khoản người dùng.	<ul style="list-style-type: none"> <li>Điền vào các thông tin cần thiết cho một tài khoản người dùng để tạo mới một tài khoản. Tài khoản này có thể là người dùng hoặc quản trị viên.</li> </ul>	Đặng Phú Quý



#### 2.4.11. Giao diện trang chỉnh sửa tài khoản

Edit User

Edit User

ID

3

Email

abc3@gmail.com

Password

123456

FirstName

Van A

LastName

Nguyen

PhoneNumber

0342609928

Role
















User

editUser

**Hình 13. Giao diện trang chỉnh sửa tài khoản**

Mục đích	Giải thích	Người thiết kế
Dùng để chỉnh sửa thông tin tài khoản người dùng dành cho quản trị viên	<ul style="list-style-type: none"><li>• Cập nhật các thông tin tại các trường để tiến hành thay đổi thông tin cho tài khoản.</li><li>• Chỉ có thể thay đổi thông tin về mật khẩu, tên, họ, số điện thoại và loại tài khoản.</li></ul>	Đặng Phú Quý

## 2.4.12. Giao diện trang quản lý sản phẩm

Manage Product							
				Search....	Q	+ Add New Product	
ID	Name	Image	Brand	Color	Size	Price	Actions
51	Jordan 11 Retro DMP Gratitude (2023)		Air Jordan	Black/White	7.5 / 8.0 / 8.5 / 9.0 / 9.5 / 10.0 / 10.5 / 11.0 / 11.5 / 12.0 / 12.5	\$271	 
67	Jordan 4 Retro SE Craft Medium Olive		Air Jordan	MEDIUM OLIVE/PALE VANILLA/KHAKI/BLACK/SAIL	7.0 / 7.5 / 8.0 / 8.5 / 9.0 / 9.5 / 10.0 / 10.5 / 11.0 / 11.5 / 12.0 / 12.5	\$210	 
83	Jordan 1 Low Wolf Grey (Women's)		Air Jordan	WHITE/WOLF GREY-ALUMINIUM	7.0 / 7.5 / 8.0 / 8.5 / 9.0 / 9.5 / 10.0 / 10.5 / 11.0 / 11.5 / 12.0 / 12.5	\$129	 
99	Jordan 1 Retro High OG UNC Toe		Air Jordan	UNIVERSITY BLUE/BLACK/WHITE	7.0 / 7.5 / 8.0 / 8.5 / 9.0 / 9.5 / 10.0 / 10.5 / 11.0 / 11.5 / 12.0 / 12.5	\$157	 
115	Jordan 4 Retro Thunder (2022)		Air Jordan	BLACK/TOUR YELLOW	7.0 / 7.5 / 8.0 / 8.5 / 9.0 / 9.5 / 10.0 / 10.5 / 11.0 / 11.5 / 12.0 / 12.5	\$186	 

**Hình 14. Giao diện trang quản lý sản phẩm**

Mục đích	Giải thích	Người thiết kế
Quản lý sản phẩm của hệ thống. Tại đây có thể tiến hành thêm mới, sửa, xoá và tìm kiếm sản phẩm.	<ul style="list-style-type: none"> <li>• Ấn vào “Add New Product” để thêm mới sản phẩm.</li> <li>• Ấn vào biểu tượng edit của sản phẩm tương ứng để sửa thông tin sản phẩm.</li> <li>• Ấn vào biểu tượng xoá của sản phẩm tương ứng để xoá sản phẩm.</li> <li>• Nhập vào từ khoá liên quan đến tên của sản phẩm vào thanh tìm kiếm và ấn biểu tượng search để tiến hành tìm kiếm sản phẩm.</li> </ul>	Đặng Phú Quý

### 2.4.13. Giao diện trang thêm mới sản phẩm

**Hình 15. Giao diện trang thêm mới sản phẩm**

Mục đích	Giải thích	Người thiết kế
Thêm mới một sản phẩm cho cửa hàng.	<ul style="list-style-type: none"> <li>•Điền đầy đủ các thông tin vào trường của sản phẩm như tên, link ảnh, màu sắc, ...</li> <li>•Tại phần size, mặc định sẽ được tích chọn đầy đủ, người quản trị có thể bỏ tích để chọn đúng size của sản phẩm hiện có.</li> </ul>	Đặng Phú Quý

#### 2.4.14. Giao diện trang chỉnh sửa sản phẩm

Edit Product

Edit Product

ID

51

Name

Jordan 11 Retro DMP Gratitude (2023)

Color

Black/White

Size

☐ 7.0

☒ 7.5

☒ 8.0

☒ 8.5

☒ 9.0

☒ 9.5

☒ 10.0

☒ 10.5

☒ 11.0

☒ 11.5

☒ 12.0

☒ 12.5

Image1

https://images.stockx.com/360/Air-Jordan-11-Retro-DMP-Defining-Moments-20

Image2

https://images.stockx.com/360/Air-Jordan-11-Retro-DMP-Defining-Moments-20

Image3

https://images.stockx.com/360/Air-Jordan-11-Retro-DMP-Defining-Moments-20

Price

271

Description

The Jordan 11 Retro Gratitude / Defining Moments (2023) is a nod to the illustrious heritage encapsulated in the Jordan brand. This iconic Jordan 11

Category

Air Jordan

editProduct

**Hình 16. Giao diện trang chỉnh sửa sản phẩm**

Mục đích	Giải thích	Người thiết kế
Chỉnh sửa thông tin của một sản phẩm đang có trong cửa hàng.	<ul style="list-style-type: none"> <li>Cập nhật lại thông tin ở các trường cần chỉnh sửa và submit để thực hiện chỉnh sửa.</li> </ul>	Đặng Phú Quý

### 2.4.15. Giao diện trang quản lý hóa đơn


Manage Invoice							
				02/01/2023	12/12/2023		Export Report
Invoice ID	Customer	Address	Payment Method	Date	SubTotal	Discount	Total
1602	admin	1 Vo Van Ngan, Thanh pho Thu Duc, Viet Nam, 70000	Cash in hand	12-12-2023	549.0	0.0	549.0
1551	Dang Nhat	nhat, nhat, nhat	Bank transfer	08-12-2023	129.0	0.0	129.0
1534	Dang Nhat	nhat, nhat, nhat	Cash in hand	08-12-2023	258.0	20.0	238.0
1452	admin	nhat, nhat, nhat	Cash in hand	08-12-2023	210.0	20.0	190.0
1406	Tran Phuong	nhat, nhat, nhat	Cash in hand	08-12-2023	210.0	20.0	190.0
1401	Dang Nhat	nhat, nhat, nhat	Bank transfer	08-12-2023	210.0	0.0	210.0
1307	admin	nhat, nhat, nhat	Cash in hand	08-12-2023	129.0	0.0	129.0
1302	Dang Nhat	nhat, nhat, nhat	Cash in hand	08-12-2023	210.0	0.0	210.0
1252	Dang Nhat	nhat, nhat, nhat	Cash in hand	08-12-2023	157.0	0.0	157.0

**Hình 17. Giao diện trang quản lý hóa đơn**

Mục đích	Giải thích	Người thiết kế
Quản lý hoá đơn của cửa hàng. Tại đây người quản trị có thể xem được toàn bộ hoá đơn của cửa hàng. Có thể chọn xem hoá đơn trong khoảng thời gian chỉ định. Xuất báo cáo theo khoảng thời gian.	<ul style="list-style-type: none"> <li>Mỗi dòng của bảng hiện thị các thông tin của một hoá đơn như id, tên khách hàng, địa chỉ giao hàng, phương thức thanh toán, ngày tạo hoá đơn, tổng tiền, ...</li> <li>Để xem chi tiết một hoá đơn gồm những món hàng gì có thể ấn vào biểu tượng view ở dòng chứa hoá đơn tương ứng.</li> <li>Nhập vào khoảng thời gian mong muốn để xem các hoá đơn đã tạo trong khoảng thời gian ấy.</li> <li>Ấn “Export Report” để xuất ra báo cáo dưới dạng file excel (.xls) về tổng hợp các hoá đơn trong khoảng thời gian chỉ định.</li> </ul>	Đặng Phú Quý

## 2.4.16. Giao diện trang xem hóa đơn

[Invoice](#) >> **ID: #1602**



To: [Nhat Dang](#)  
 Address: 1 Vo Van Ngan, Thanh  
 pho Thu Duc, Viet Nam, 70000  
 Phone: 0938951674

Invoice  
 ID: #1602  
 Creation Date: 12-12-2023  
 Payment Method: Cash in hand

#	Description	Qty	Unit Price	Amount
1	Jordan 4 Retro SE Craft Medium Olive	2	210.0	420.0
2	Jordan 1 Low Wolf Grey (Women's)	1	129.0	129.0

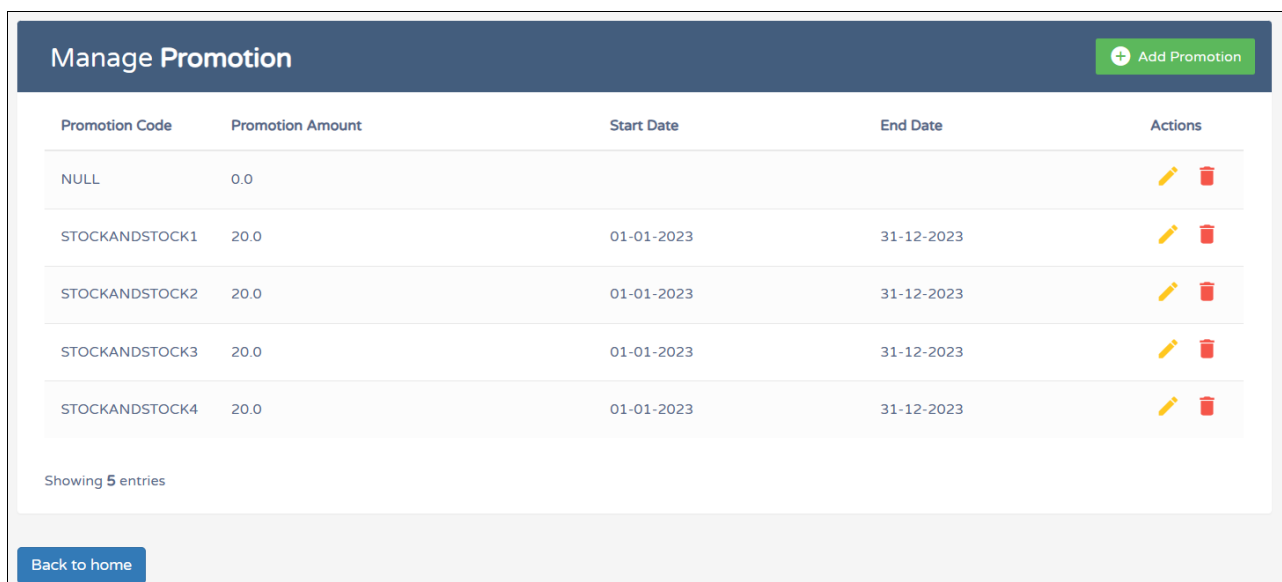
SubTotal \$549.0  
 Discount \$0.0  
**Total Amount \$549.0**

Thank you for your purchase
 [Return](#)

**Hình 18. Giao diện trang xem hóa đơn**

Mục đích	Giải thích	Người thiết kế
Xem chi tiết một hoá đơn mua hàng của người dùng	<ul style="list-style-type: none"> <li>Gồm có các thông tin chi tiết về một hoá đơn như tên người nhận, địa chỉ giao hàng, id, ngày tạo hoá đơn, phương thức thanh toán, ... và chi tiết các món hàng đã mua.</li> </ul>	Đặng Phú Quý

### 2.4.17. Giao diện trang quản lý khuyến mãi



**Hình 19. Giao diện trang quản lý khuyến mãi**

Mục đích	Giải thích	Người thiết kế
Quản lý mã giảm giá của cửa hàng. Tại đây có thể thực hiện việc thêm xoá, sửa thông tin liên quan đến mã giảm giá.	<ul style="list-style-type: none"> <li>• Ấn “Add Promotion” để thêm mã giảm giá mới.</li> <li>• Ấn vào biểu tượng edit để chỉnh sửa thông tin của mã giảm giá tương ứng.</li> <li>• Ấn vào biểu tượng delete để xoá mã giảm giá ở hàng tương ứng.</li> </ul>	Đặng Phú Quý

#### 2.4.18. Giao diện thêm mới mã giảm giá

The screenshot displays a web application interface for managing promotions. A modal window titled 'New Promotion' is open in the center, allowing users to add a new promotion. The modal contains the following fields:

- Promotion Code:** A text input field.
- Promotion Amount:** A text input field.
- Start Date:** A date picker with a calendar icon, showing the format 'mm/dd/yyyy'.
- End Date:** A date picker with a calendar icon, showing the format 'mm/dd/yyyy'.

At the bottom of the modal are two buttons: 'Cancel' and 'Add'. The background interface, titled 'Manage Promotion', shows a table of existing promotions with columns for 'Promotion Code' and 'Promotion Amount'. The table lists five entries, including a NULL entry and four entries with codes like 'STOCKANDSTOCK1' and amounts of '20.0'. A '+ Add Promotion' button is visible in the top right corner of the background interface.

**Hình 20. Giao diện thêm mới mã giảm giá**

Mục đích	Giải thích	Người thiết kế
Thêm mới một mã giảm giá.	<ul style="list-style-type: none"> <li>Thêm các thông tin cần thiết cho một mã giảm giá bao gồm mã giảm giá, số lượng giảm, ngày bắt đầu và ngày kết thúc.</li> </ul>	Đặng Phú Quý



#### 2.4.19. Giao diện chỉnh sửa mã giảm giá

Edit Promotion

Edit Promotion

Promotion Code

STOCKANDSTOCK1

Promotion Amount

20.0

Start Date

01/01/2023

End Date

12/31/2023

editPromotion

**Hình 21. Giao diện trang chỉnh sửa mã giảm giá**

Mục đích	Giải thích	Người thiết kế
Chỉnh sửa thông tin của một mã giảm giá.	<ul style="list-style-type: none"><li>•Cập nhật các thông tin cần chỉnh sửa của một mã giảm giá và lưu lại.</li><li>•Thông tin có thể chỉnh sửa bao gồm tiền giảm, ngày bắt đầu và kết thúc.</li></ul>	Đặng Phú Quý

#### 2.4.20. Giao diện quản lý danh mục thương hiệu



**Hình 22. Giao diện trang quản lý danh mục thương hiệu**

Mục đích	Giải thích	Người thiết kế
Quản lý các danh mục sản phẩm của cửa hàng. Gồm thêm, sửa, xoá danh mục sản phẩm.	<ul style="list-style-type: none"> <li>• Ấn vào “Add Category” để thêm danh mục sản phẩm mới.</li> <li>• Ấn vào biểu tượng edit của danh mục cùng hàng tương ứng để chỉnh sửa thông tin danh mục.</li> <li>• Ấn vào biểu tượng xoá để xoá danh mục cùng hàng tương ứng.</li> </ul>	Đặng Phú Quý

#### 2.4.21. Giao diện chỉnh sửa thương hiệu

Edit Category

Edit Category

Category ID

51

Category Name





Air Jordan

editCate

**Hình 23. Giao diện chỉnh sửa thương hiệu**

Mục đích	Giải thích	Người thiết kế
Chỉnh sửa thông tin của một danh mục.	•Người quản trị có thể cập nhật lại tên của danh mục.	Đặng Phú Quý

#### 2.4.22. Giao diện trang thống kê sản phẩm bán được

Product Statistic							mm/dd/yyyy	mm/dd/yyyy	Q
No.	ID	Name	Image	Brand	Price	Sold Total			
1	67	Jordan 4 Retro SE Craft Medium Olive		Air Jordan	\$210	8			
2	83	Jordan 1 Low Wolf Grey (Women's)		Air Jordan	\$129	6			
3	147	Jordan 1 Retro High OG Royal Reimagined		Air Jordan	\$106	3			
9	99	Jordan 1 Retro High OG UNC Toe		Air Jordan	\$157	1			
						Total Count:	28		
Showing 9 entries									
<a href="#">Back to home</a>									

**Hình 24. Giao diện trang thống kê sản phẩm bán được**

Mục đích	Giải thích	Người thiết kế
Thống kê số lượng sản đã bán của mỗi sản phẩm trong cửa hàng.	<ul style="list-style-type: none"> <li>Thống kê số lượng sản của mỗi sản phẩm và xếp theo thứ tự giảm, nếu sản phẩm không xuất hiện thì có nghĩa là sản phẩm chưa có lượt mua nào.</li> <li>Có thể thực hiện thống kê theo khoảng thời gian chỉ định.</li> </ul>	Đặng Phú Quý

2.4.23. Giao diện trang thống kê số tiền mua hàng của người dùng

User Statistic

mm/dd/yyyy

mm/dd/yyyy

Q

No.	ID	Name	Email	Phone	Spending
1	1	admin	abc1@gmail.com	0342609928	\$3008.0
2	1001	Dang Nhat	nhatdzfbk@gmail.com	1234567890	\$1154.0
3	1101	hehe hihi	haha@gmail.com	123	\$210.0
4	1403	Tran Phuong	phuong@gmail.com	1234567890	\$190.0
				Total Spending:	\$4562.0

Showing 4 entries

Back to home

Hình 25. Giao diện trang thống kê số tiền mua hàng của người dùng

Mục đích	Giải thích	Người thiết kế
Thông kê chi tiêu của một người dùng.	<ul style="list-style-type: none"><li>•Chi tiêu của mỗi người dùng được thống kê và sắp xếp giảm dần.</li><li>•Có thể thực hiện thống kê trên khoảng thời gian chỉ định.</li></ul>	Đặng Phú Quý

#### 2.4.24. Giao diện trang đặt lại mật khẩu

STOCK&STOCK

Reset Password

Please enter the email address that is associated with your StockX account.

Email Address

Reset Password

STOCK&STOCK

Reset Password

Please check your email address to get the verification code.

Verification code

Reset Password

**Hình 26. Giao diện trang đặt lại mật khẩu**

Mục đích	Giải thích	Người thiết kế
Để người dùng có thể khôi phục lại tài khoản khi quên mật khẩu.	<ul style="list-style-type: none"><li>•Người dùng nhập vào email đã đăng ký tài khoản. Sau đó mã xác nhận sẽ được gửi về email để xác minh người dùng đúng là chủ tài khoản.</li><li>•Nếu xác minh thành công, mật khẩu mới sẽ được cấp và gửi về email đã đăng ký tài khoản.</li><li>•Người dùng có thể dùng mật khẩu được gửi về email để có thể đăng nhập vào lại tài khoản sau đó đổi lại mật khẩu của mình.</li></ul>	Đặng Phú Quý

## CHƯƠNG 3: TRIỂN KHAI ỨNG DỤNG

### 3.1. Render

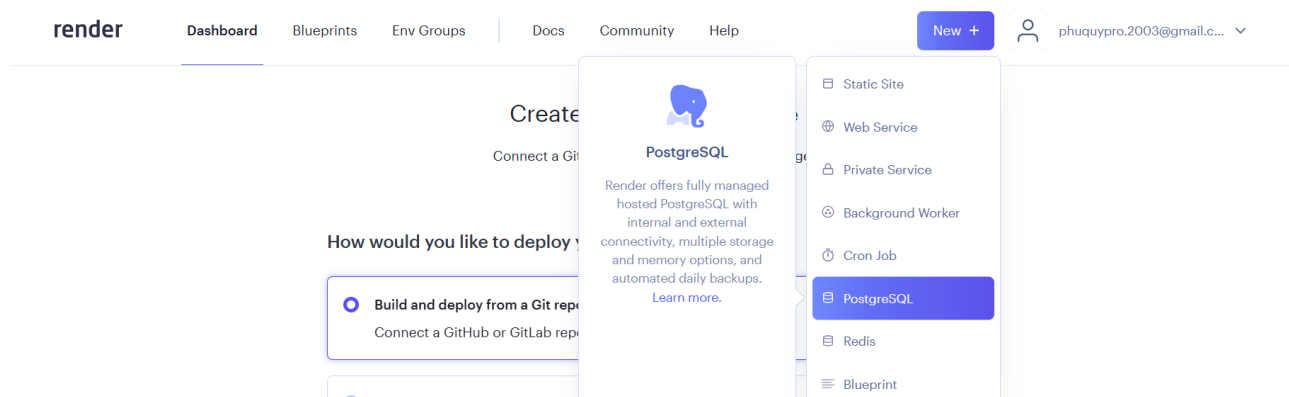
Render.com là một nền tảng đám mây (cloud platform) cung cấp các dịch vụ điện toán đám mây dễ sử dụng cho việc triển khai và quản lý các ứng dụng web và dịch vụ. Render tập trung vào việc giúp các nhà phát triển và doanh nghiệp triển khai ứng dụng của họ một cách nhanh chóng, dễ dàng và hiệu quả.

### 3.2. Triển khai cơ sở dữ liệu

Để triển khai và sử dụng cơ sở dữ liệu PostgreSQL trên Render, cần thực hiện các bước sau:

#### Bước 1: Tạo Dịch Vụ PostgreSQL trên Render

##### - Tạo dịch vụ PostgreSQL



##### - Cấu hình PostgreSQL

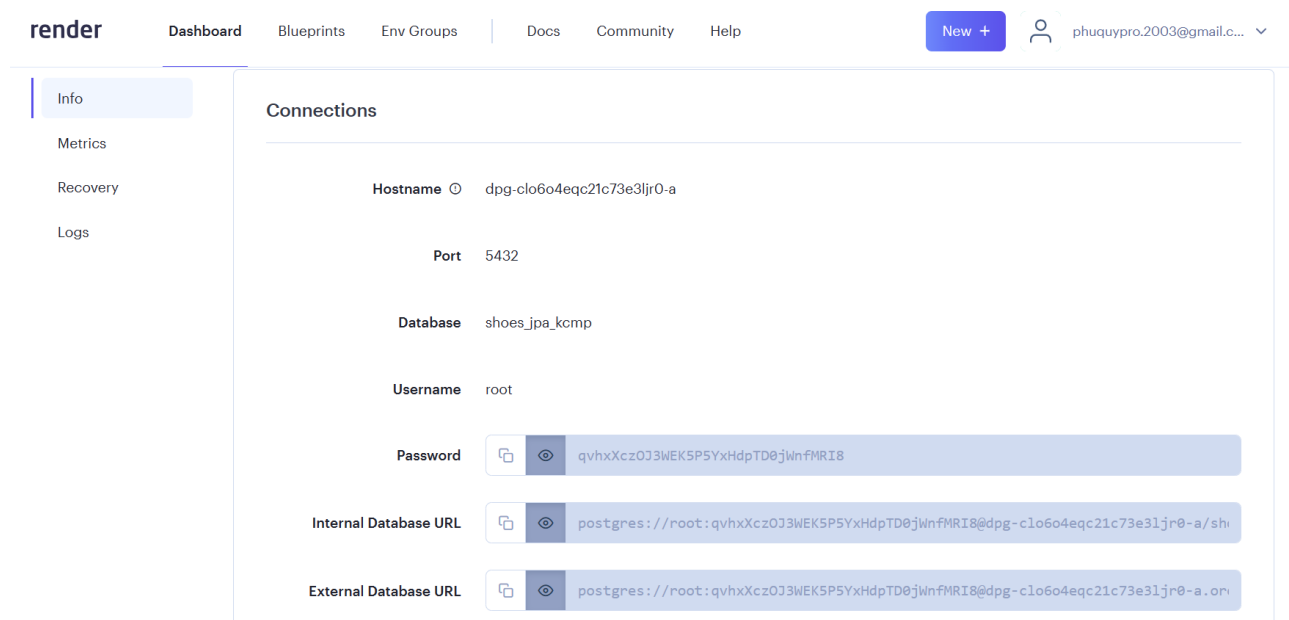
#### New PostgreSQL

[Read the docs](#)

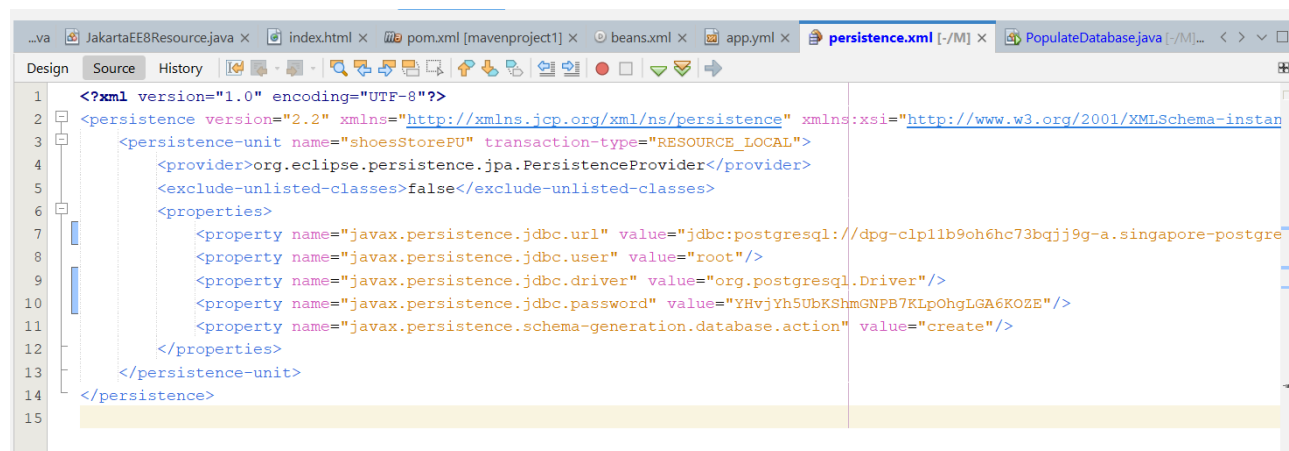
<b>Name</b> A unique name for your PostgreSQL instance.	StockAndStock
<b>Database</b> <small>Optional</small> The PostgreSQL `dbname`	shoes_jpa
<b>User</b> <small>Optional</small>	root
<b>Region</b> The <a href="#">region</a> where your PostgreSQL instance runs. Services must be in the same region to communicate privately and you currently have services running in <b>Oregon</b> .	Oregon (US West)
<b>PostgreSQL Version</b>	15

## Bước 2: Thiết Lập Kết Nối Đến PostgreSQL

### - Lấy thông tin kết nối



- Trong ứng dụng, cấu hình thông tin kết nối với PostgreSQL (ta sẽ sử dụng External Database URL để thiết lập kết nối)



### 3.3. Triển khai ứng dụng web

Demo triển khai trên render: <https://stockandstock.onrender.com/StockAndStock>

Để triển khai dự án website Java Servlet sử dụng Tomcat Server lên Render cần thực hiện các bước sau:

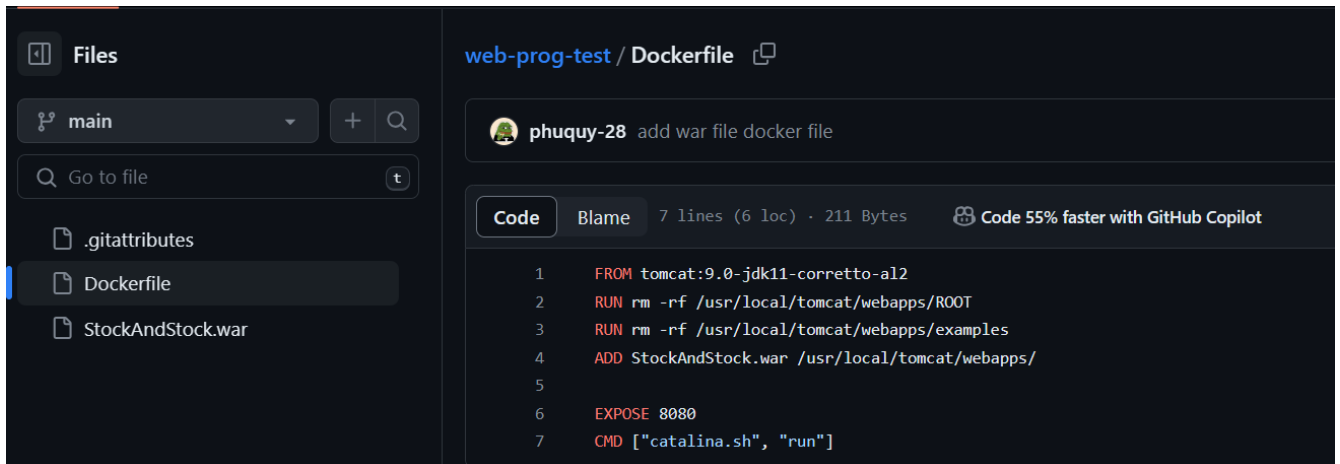


## Bước 1: Chuẩn bị ứng dụng Java servlet

- Đảm bảo ứng dụng Java Servlet của bạn tuân theo các tiêu chuẩn và yêu cầu của Tomcat. Cần phải có tệp WAR chứa ứng dụng.

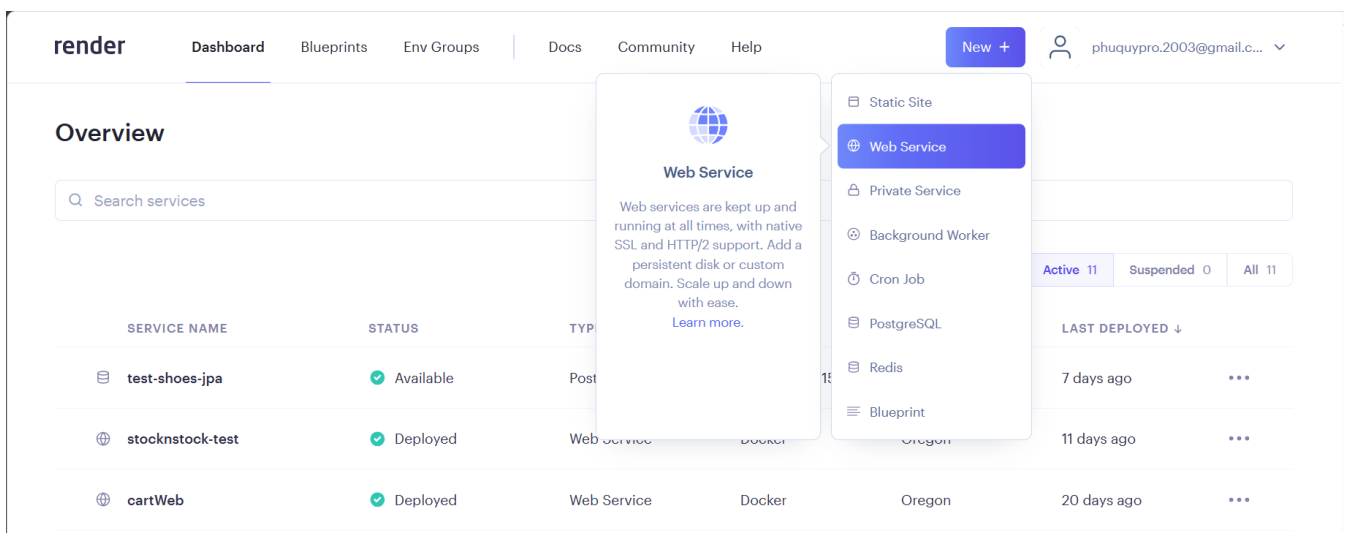
## Bước 2: Tạo Tệp “Dockerfile”:

- Tạo một tệp Dockerfile trong thư mục gốc của dự án



## Bước 3: Đăng Ký và Tạo Dự Án Render.com:

- Đăng nhập vào Render.com thông qua tài khoản github có repository chứa dự án cần triển khai. Tạo một Web Service



- Chọn “Build and deploy from a Git repository”

## Create a new **Web Service**

Connect a Git repository, or use an existing image.

### How would you like to deploy your web service?

☒ **Build and deploy from a Git repository**

Connect a GitHub or GitLab repository.

☐ **Deploy an existing image from a registry** **ADVANCED**

Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.



Next

- Chọn Repository chứa ứng dụng web và dockerfile đã tạo từ trước.


## Create a new **Web Service**

Connect your Git repository or use an existing public repository URL.



### Connect a repository

 phquy-28 / test-gae-prj  • 2 days ago



Connect

 dmnhatit / AI\_Pacman • 5 days ago

Connect



 phquy-28 / web-prog-test  • 11 days ago



Connect



 dmnhatit / pretest4  • 16 days ago


Connect

### GitHub


 @phquy-28  • 21 repos

 @dmnhatit  • 3 repos

 @khangbuiquoc203  • 3 repos

 [Configure account](#)

### GitLab

 [Connect account](#)



## CHƯƠNG 4: KIỂM THỬ

TT	Tình huống dự kiến	Mục đích	Giải thích
1	<p><i>Tìm kiếm sản phẩm theo tên và lọc theo kích cỡ, giá, loại sản phẩm.</i></p> <p><b>Dữ liệu đầu vào:</b></p> <p>Chuỗi ký tự liên quan đến tên sản phẩm. Hoặc chọn vào kích cỡ hoặc khoảng giá, danh mục có sẵn.</p> <p><b>Kết quả dự kiến:</b></p> <p>Hiện thị ra các sản phẩm phù với tên đã tìm kiếm hoặc giá, kích cỡ, danh mục đã áp dụng bộ lọc.</p>	Giúp người dùng nhanh chóng tìm được sản phẩm phù hợp với các tiêu chí cụ thể.	Tình huống này phản ánh một yêu cầu phổ biến khi người dùng muốn tìm kiếm sản phẩm chính xác và lọc kết quả theo các tiêu chí cụ thể như thương hiệu, kích cỡ và giá.
4	<p><i>Thêm bớt số lượng của một sản phẩm trong giỏ hàng thông qua nút (-)(+)(×)</i></p> <p><b>Dữ liệu đầu vào:</b></p> <p>Số lần click vào dấu (-)(+)(×) tại cột số lượng của sản phẩm đó trong giỏ hàng.</p> <p><b>Kết quả dự kiến:</b></p> <p>Cứ mỗi lượt click vào dấu (+) thì số lượng của sản phẩm đó sẽ được tăng lên một đơn vị.</p> <p>Nếu số lượng đang là 1 và có một lần click vào dấu (-) thì sản phẩm sẽ bị xoá ra khỏi giỏ hàng.</p> <p>Nếu click vào dấu (×) thì sản phẩm đó sẽ bị xoá khỏi giỏ hàng mà không quan tâm đến số lượng hiện có.</p>	Mục đích của quyết định này là cung cấp sự linh hoạt và thuận tiện cho người dùng khi họ muốn điều chỉnh số lượng sản phẩm mà họ đang mua mà không cần phải thực hiện các bước phức tạp.	Trường hợp này phản ánh quá trình quản lý số lượng sản phẩm trong giỏ hàng trực tuyến, nơi người dùng có khả năng linh hoạt điều chỉnh số lượng của từng sản phẩm một cách thuận tiện.

5	<p><i>Tích chọn một hoặc nhiều sản phẩm trong giỏ hàng để thanh toán.</i></p> <p><b>Dữ liệu đầu vào:</b></p> <p>Các sản phẩm được tích chọn trong trong giỏ hàng</p> <p><b>Kết quả dự kiến:</b></p> <p>Các sản phẩm được tích chọn sẽ được chuyển sang trang thanh toán.</p>	<p>Cung cấp khả năng lựa chọn, quản lý và thanh toán cho các sản phẩm cụ thể một cách thuận tiện trong quá trình mua sắm trực tuyến.</p>	<p>Trường hợp này tập trung vào việc tối ưu hóa quy trình thanh toán và quản lý giỏ hàng để đáp ứng nhu cầu linh hoạt và thuận tiện của người dùng trong quá trình mua sắm trực tuyến.</p>
6	<p><i>Nhập vào một mã giảm giá hợp lệ và không hợp lệ.</i></p> <p><b>Dữ liệu đầu vào:</b></p> <p>Một mã giảm giá hợp lệ, còn thời hạn và chưa được người dùng đó sử dụng lần nào</p> <p>Một mã giảm giá sai hoặc không còn thời hạn hoặc đã được sử dụng trước đó bởi người dùng.</p> <p><b>Kết quả dự kiến:</b></p> <p>Mã giảm giá hợp lệ sẽ được chuyển đến trang thanh toán và hiển thị số tiền giảm.</p> <p>Mã giảm giá không hợp lệ sẽ không thể chuyển trang và xuất hiện thông báo về vi phạm.</p>	<p>Mục đích chính là cung cấp khả năng áp dụng ưu đãi giảm giá cho đơn hàng và xử lý các trường hợp khi mã không hợp lệ.</p>	<p>Trường hợp này phản ánh quy trình quản lý mã giảm giá trong quá trình thanh toán, mang lại trải nghiệm linh hoạt và minh bạch cho người dùng cũng như đảm bảo tính chính xác của giao dịch.</p>
7	<p><i>Khi tiến hành thanh toán, nếu giao dịch bị hủy, sản phẩm trong giỏ hàng sẽ không bị mất đi.</i></p> <p><b>Dữ liệu đầu vào:</b></p>	<p>Bảo vệ thông tin giỏ hàng của người dùng và đảm bảo rằng họ không mất đi các sản phẩm đã chọn khi giao dịch bị hủy.</p>	<p>Phản ánh cam kết của hệ thống để bảo vệ thông tin giỏ hàng và sản phẩm đã chọn của người dùng trong trường hợp giao dịch không thành công. Điều này nhấn mạnh vào</p>

	<p>Sản phẩm trong giỏ hàng được tích chọn để thanh toán và được điều hướng tới trang thanh toán.</p> <p><b>Kết quả dự kiến:</b></p> <p>Trở lại trang giỏ hàng, các sản phẩm vừa huỷ giao dịch vẫn được lưu lại.</p>		<p>việc cung cấp một trải nghiệm mua sắm trực tuyến mượt mà và không làm mất lòng tin của người dùng.</p>
8	<p><i>Tạo tài một tài khoản mới.</i></p> <p><b>Dữ liệu đầu vào:</b></p> <p>Thông tin một tài khoản chưa được đăng ký.</p> <p>Thông tin của một tài khoản đã được đăng ký.</p> <p><b>Kết quả dự kiến:</b></p> <p>Nếu thông tin chưa được đăng ký trước đó, sẽ tạo được tài khoản và đăng nhập bình thường.</p> <p>Nếu thông tin đã được đăng ký từ trước sẽ xuất hiện cảnh báo lỗi.</p>	<p>Cho phép người dùng đăng ký và tạo một tài khoản mới trên trang web hoặc ứng dụng. Quy trình này nhằm xác nhận tính duy nhất của thông tin tài khoản và cung cấp quyền truy cập đối với người dùng mới.</p>	<p>Phản ánh quy trình quan trọng trong việc quản lý người dùng và đảm bảo tính duy nhất của thông tin tài khoản trên nền tảng trực tuyến.</p>
9	<p><i>Khôi phục lại mật khẩu của tài khoản người dùng.</i></p> <p><b>Dữ liệu đầu vào:</b></p> <p>Email của tài khoản đã đăng ký.</p> <p>Mã xác nhận được gửi về email đã đăng ký đó.</p> <p><b>Kết quả dự kiến:</b></p> <p>Mật khẩu mới được đặt lại cho tài khoản và gửi mật khẩu về email đã đăng ký.</p>	<p>Cung cấp khả năng cho người dùng lấy lại quyền truy cập vào tài khoản của họ khi họ quên mật khẩu. Mục đích chính là bảo vệ thông tin cá nhân của người dùng và cung cấp quy trình an toàn để khôi phục mật khẩu.</p>	<p>Trường hợp này quan trọng để đảm bảo tính bảo mật và thuận tiện trong quá trình quản lý mật khẩu và thông tin tài khoản người dùng trên các nền tảng trực tuyến.</p>

**Bảng 44. Các tình huống kiểm thử**

## CHƯƠNG 5: KẾT LUẬN

### 5.1. Kết quả đạt được

Xây dựng được website bán giày online với các chức năng cơ bản theo mô hình MVC. Các chức năng bao gồm: Quản lý sản phẩm, quản lý khách hàng, đăng tải các sản phẩm để khách hàng lựa chọn và mua sắm, hỗ trợ khách hàng lưu lại sản phẩm vào giỏ hàng, hỗ trợ khách hàng thanh toán những sản phẩm mà khách hàng chọn từ giỏ hàng, hỗ trợ khách hàng lựa chọn thanh toán bằng tiền mặt hoặc thẻ tín dụng, cung cấp cho khách hàng những chương trình khuyến mãi, nhập mã khuyến mãi để được giảm giá.

Triển khai thành công cơ sở dữ liệu và ứng dụng lên nền tảng Render và thực hiện được các chức năng đã nói trên.

### 5.2. Đánh giá

#### 5.2.1. Ưu điểm

- Giao diện đơn giản, dễ nhìn, thân thiện với người dùng.
- Có đầy đủ chức năng để khách hàng có thể truy cập và mua sắm.
- Có lưu trữ lại thông tin mua sắm của khách hàng.
- Hỗ trợ khách hàng mua sắm theo nhu cầu của họ như tên giày, thương hiệu, giá tiền cũng như là size giày.

#### 5.2.2. Khuyết điểm

- Chưa tối ưu được tốc độ của hệ thống.
- Tốc độ load của trang web khi triển khai trên Render chậm.

### 5.3. Hướng phát triển

- Cải tiến giao diện đẹp hơn, thu hút người dùng hơn.
- Tối ưu hóa các chức năng hiện tại của trang web.
- Thêm các chức năng mới mà những trang web khác chưa có.
- Tối ưu hóa tốc độ của trang web.

## TÀI LIỆU THAM KHẢO

[1] Murach, J., & Urban, M. (2014). *Murach's Java Servlets and JSP*. Mike Murach & Associates.

[2] Tham khảo giao diện website: <https://stockx.com/>