

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH HỆ THÔNG THÔNG TIN

Đề tài

Xây dựng website quản lý cửa hàng thiết bị
thể thao

Sinh viên: Nguyễn Tân Tài

Mã số: B1906342

Khóa: K45

Cần Thơ, 11/2023

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH HỆ THÔNG THÔNG TIN

Đề tài

Xây dựng website quản lý cửa hàng thiết bị
thể thao

Người hướng dẫn
Phạm Thị Ngọc Diễm

Sinh viên: Nguyễn Tân Tài
Mã số: B1906342
Khóa: K45

Cần Thơ, 11/2023

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA HỆ THÔNG THÔNG TIN

XÁC NHẬN CHỈNH SỬA LUẬN VĂN
THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh):

Xây dựng website quản lý cửa hàng thiết bị thể thao

Building a website for managing a sporting goods store

Họ tên sinh viên: Nguyễn Tân Tài

MASV: B1906342

Mã lớp: DI1995A1

Đã báo cáo tại hội đồng ngành: Hệ Thông Tin

Ngày báo cáo: 15/5/2024

Hội đồng báo cáo gồm:

ThS. Phan Tân Tài

ThS. Phạm Ngọc Quyền

TS. Phạm Thị Ngọc Diễm

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày 15 tháng 5 năm 2024

Giáo viên hướng dẫn

(Ký và ghi họ tên)

Phạm Thị Ngọc Diễm

LỜI CẢM ƠN

Trước hết, em xin chân thành cảm ơn cô Phạm Thị Ngọc Diễm – một người cô tận tâm, luôn sẵn lòng giúp đỡ và đồng hành cùng em trong suốt quá trình thực hiện báo cáo niêm luận. Không chỉ là sự hướng dẫn về mặt chuyên môn, cô còn là nguồn động viên tinh thần, giúp em vượt qua những khó khăn, mệt mỏi trong quá trình nghiên cứu.

Sự kiên nhẫn, tâm huyết và trí tuệ của cô đã giúp em mở rộng tầm mắt, định hình được hướng đi đúng đắn cho niêm luận của mình. Các buổi thảo luận, những lời khuyên và nhận xét chân thành từ cô đã tạo nên một nền tảng vững chắc cho em, giúp em tránh được nhiều sai lầm có thể gặp phải.

Em cũng xin bày tỏ lòng biết ơn sâu sắc với những tài liệu, nguồn thông tin mà cô đã giới thiệu. Chúng đã trở thành nguồn tham khảo quý giá, giúp em có cái nhìn sâu rộng hơn về đề tài mình đang nghiên cứu.

Với tất cả lòng biết ơn, em kính chúc cô luôn mạnh khỏe, tràn đầy niềm vui và tiếp tục đem lại ánh sáng kiến thức, sự hướng dẫn tận tâm cho thế hệ sinh viên sau này.

Trân trọng và kính mến,

Nguyễn Tấn Tài

TÓM TẮT

Trong bối cảnh ngày nay, sự phổ cập của internet đã tạo ra cơ hội mới và thách thức đối với các doanh nghiệp trực tuyến, đặc biệt là trong lĩnh vực bán thiết bị thể thao. Để tạo ra trải nghiệm mua sắm tốt nhất cho người dùng và đồng thời nâng cao hiệu suất kinh doanh, việc áp dụng công nghệ tiên tiến đã trở thành một xu hướng quan trọng.

Để tài trợ trung vào tối ưu hóa trang web bán thiết bị thể thao bằng cách tích hợp mô hình VGG-16 để cải thiện quá trình tìm kiếm sản phẩm dựa trên hình ảnh. Sử dụng cosine similarity, để tài đánh giá và tối ưu hóa sự tương đồng giữa các sản phẩm, nâng cao trải nghiệm mua sắm.

Ngoài ra, để tài mở rộng chức năng trang web bằng cách tích hợp chatbot, cung cấp thông tin sản phẩm và hỗ trợ người dùng trong quá trình mua sắm. Kết hợp giữa tìm kiếm ảnh thông minh và chatbot mang lại trải nghiệm mua sắm độc đáo và thuận tiện.

Từ khoá: Thiết bị thể thao, VGG-16, TFLearn, độ đo tương tự cosin, SVM

ABSTRACT

In the contemporary era, the ubiquity of the internet has presented both new opportunities and challenges for online businesses, particularly in the realm of store management website sport. The pursuit of delivering the optimal shopping experience for users while concurrently enhancing business efficiency has underscored the significance of integrating advanced technologies.

This study focuses on optimizing a store management website sport e-commerce website by incorporating the VGG-16 model to refine the product search process based on images. Employing cosine similarity, the research evaluates and optimizes the similarity between products, thereby elevating the shopping experience.

Furthermore, the study extends the website's functionality by integrating a chatbot, offering product information, and assisting users throughout the shopping journey. The synergy between intelligent image search and the chatbot culminates in a distinctive and convenient shopping experience.

Keywords: Selling interior decorations, VGG-16, TFLearn, cosine similarity, SVM

MỤC LỤC

LỜI CẢM ƠN.....	i
TÓM TẮT	ii
ABSTRACT.....	iii
MỤC LỤC.....	iv
DANH MỤC HÌNH.....	vii
DANH MỤC PHỤ LỤC HÌNH.....	xii
DANH MỤC BẢNG.....	xiii
DANH MỤC TỪ CHUYÊN NGÀNH	xv
CHƯƠNG 1. GIỚI THIỆU	1
1.1. Đặt vấn đề.....	1
1.2. Nghiên cứu liên quan	1
1.2.1. Nhận dạng hình ảnh	1
1.2.2. TensorFlow và TFLearn.....	2
1.2.3. Hệ thống gợi ý	2
1.2.4. Phân loại văn bản	3
1.3. Mục tiêu đề tài.....	4
1.4. Đối tượng và phạm vi đề tài	4
1.4.1. Đối tượng nghiên cứu.....	4
1.4.2. Phạm vi nghiên cứu.....	5
CHƯƠNG 2. MÔ TẢ BÀI TOÁN	6
2.1. Mô tả chi tiết đề tài.....	6
2.1.1. Chức năng chính.....	6
2.1.2. Nhóm người dùng	7
2.1.3. Đặc điểm nổi bật	9
2.1.4. Giải pháp khắc phục	9
2.2. Hướng tiếp cận giải quyết của đề tài	9
2.2.1. Phương pháp tiếp cận	9
2.2.2. Công nghệ và ngôn ngữ lập trình	9
2.2.3. Frameworks và thư viện.....	10
2.3. Cơ sở lý thuyết	10
2.3.1. Deep Learning (Mô hình học sâu)	10
2.3.2. Mạng nơ-ron tích chập CNN.....	11
2.3.3. Mạng nơ-ron sâu DNN.....	13
2.3.4. Mô hình VGG-16	16
2.3.5. Thư viện TFlearn.....	19
2.3.6. Độ tương tự cosin	21

2.3.7. SVM - Support Vector Machine	24
CHƯƠNG 3. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP	30
3.1. Kiến trúc tổng quát hệ thống.....	30
3.1.1. Mô hình Client-Server.....	30
3.1.2. Mô hình giao tiếp	30
3.1.3. Cơ sở dữ liệu (CSDL)	31
3.1.4. Kiến trúc tổng quát.....	31
3.1.5. Xây dựng các mô hình dữ liệu	31
3.1.6. Xây dựng các mô hình AI	36
3.2. Giải pháp cài đặt.....	54
3.2.1. Môi trường cài đặt.....	54
3.2.2. Cài đặt các công cụ cần thiết.....	54
3.2.3. Cài đặt Laravel	54
3.2.4. Cài đặt Vue.js và TypeScript	54
3.2.5. Cài đặt và cấu hình MySQL, Apache, PHPMyAdmin	55
3.2.6. Phát triển và tích hợp thư viện	55
3.2.7. Tham số và cấu hình	55
3.3. Thiết kế giao diện.....	55
3.3.1. Giao diện người dùng.....	55
3.3.2. Giao diện quản trị.....	77
CHƯƠNG 4. KIỂM THỦ VÀ ĐÁNH GIÁ	92
4.1. Mục tiêu kiểm thử	92
4.2. Kịch bản kiểm thử	95
4.3. Kết quả kiểm thử	100
4.4. Kết luận chung	105
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	107
5.1. Kết luận	107
5.1.1. Đề tài đã đạt được	107
5.1.2. Hạn chế của đề tài	107
5.2. Hướng phát triển	107
5.2.1. Tối ưu hóa trải nghiệm di động.....	107
5.2.2. Hỗ trợ đa ngôn ngữ và đa tiền tệ.....	107
5.2.3. Phát triển hệ thống Big Data	108
5.2.4. Tăng cường tính bảo mật	108
TÀI LIỆU THAM KHẢO	109
PHỤ LỤC	112
7.1. Mô tả dữ liệu mức luận lý	112
7.2. Cách đào tạo mô hình tìm kiếm hình ảnh tương tự.....	125
7.3. Cách đào tạo mô hình Chatbot	134

7.4. Cách đào tạo mô hình Hệ thống gợi ý.....	141
7.5. Cách đào tạo mô hình Nhận dạng cảm xúc.....	147

DANH MỤC HÌNH

Hình 2. 1. Ví dụ tích chập ảnh ma trận 5x5 với bộ lọc 3x3	11
Hình 2. 2. Ví dụ cách trượt stride = 2 và dùng đệm padding = 0	12
Hình 2. 3. Ví dụ tầng pooling sử max pooling	13
Hình 2. 4. Tầng kết nối đầy đủ nhận đầu vào dữ liệu làm phẳng.....	13
Hình 2. 5. Mô phỏng thuật toán Gradient Descent.....	14
Hình 2. 6. Mô phỏng và so sánh thuật toán SGD và GD	15
Hình 2. 7. Minh họa thuật toán RMSProp.....	15
Hình 2. 8. Mô phỏng thuật toán Adam.....	16
Hình 2. 9. Kiến trúc VGG-16	17
Hình 2. 10. Kiến trúc TFlearn	20
Hình 2. 11. Hàm mất mát và độ chính xác của TFlearn.....	20
Hình 2. 12. Các layers của TFLearn.....	21
Hình 2. 13. Công thức độ tương tự cosin	22
Hình 2. 14. Siêu phẳng (hyper-plane)	24
Hình 2. 15. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 1).....	25
Hình 2. 16. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 2).....	25
Hình 2. 17. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 3).....	26
Hình 2. 18. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 4).....	27
Hình 2. 19. Cách phân lớp cho siêu phẳng ngữ cảnh 4	27
Hình 2. 20. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 5).....	28
Hình 2. 21. Siêu phẳng bị biến đổi (ngữ cảnh 5).....	28
Hình 2. 22. Ví dụ về margin trong SVM.....	29
Hình 3. 1. Mô hình client-server	30
Hình 3. 2. Mô hình CDM	32
Hình 3. 3. Mô hình DFD	35
Hình 3. 4. Mô phỏng sử dụng Google Colaboratory.....	36
Hình 3. 5. Cây thư mục dữ liệu huấn luyện.....	37
Hình 3. 6. Quá trình huấn luyện mô hình tìm kiếm hình ảnh.....	38

Hình 3. 7. Kết quả lấy ra ngẫu nhiên 1 nhóm các ảnh tương đồng	39
Hình 3. 8. Kết quả tìm kiếm ảnh tương đồng khi upload một hình ảnh chỉ định.....	39
Hình 3. 9. Mẫu cấu trúc intent gồm tag, câu hỏi và câu trả lời tương ứng.....	42
Hình 3. 10. Mô hình của model Chatbot	42
Hình 3. 11. Kết quả quá trình đào tạo mô hình chatbot	43
Hình 3. 12. Kết quả sau khi đào tạo mô hình và được lưu trữ lại	43
Hình 3. 13. Tổng quan về mô hình và số lượng dữ liệu training	44
Hình 3. 14. Dataset nhận dạng cảm xúc thu thập được	45
Hình 3. 15. Chi tiết số lượng dataset nhận dạng cảm xúc	45
Hình 3. 16. Cấu trúc dataset mô hình phân tích cảm xúc xấu	46
Hình 3. 17. Dữ liệu mô hình phân tích cảm xúc sau khi được xử lí	47
Hình 3. 18. Kết quả huấn luyện mô hình phân tích cảm xúc C = 3.0	48
Hình 3. 19. Biểu ROC của mô hình phân tích cảm xúc	49
Hình 3. 20. Kết quả của mô hình phân tích cảm xúc ở lần train thứ nhất.....	50
Hình 3. 21. Kết quả của mô hình phân tích cảm xúc ở lần train thứ hai.....	50
Hình 3. 22. Kết quả của mô hình phân tích cảm xúc cho những đánh giá dài	51
Hình 3. 23. Kết quả của mô hình phân tích cảm xúc chê bai	51
Hình 3. 24. Lưu trữ lại mô hình phân tích cảm xúc	51
Hình 3. 25. 5 dòng đầu tiên trong dataset.....	52
Hình 3. 26. Tiền xử lí dữ liệu hệ thống gợi ý	53
Hình 3. 27. Các dữ liệu đã vector hoá được lưu lại của hệ thống gợi ý	54
Hình 3. 28. Giao diện trang đăng nhập.....	56
Hình 3. 29. Popup báo lỗi “Email này chưa đăng ký”	57
Hình 3. 30. Popup báo lỗi “Mật khẩu không chính xác”	57
Hình 3. 31. Giao diện trang đăng ký tài khoản.....	58
Hình 3. 32. Thông báo lỗi khi thông tin đăng ký không hợp lệ	59
Hình 3. 33. Giao diện trang chủ của website.....	60
Hình 3. 34. Giao diện danh sách sản phẩm đang bán chạy	60
Hình 3. 35. Giao diện danh sách tất cả các sản phẩm	61
Hình 3. 36. Bộ lọc sản phẩm theo giá	61

Hình 3. 37. Nút tìm tải ảnh lên để tìm kiếm sản phẩm có ảnh tương đồng.....	61
Hình 3. 38. Giao diện khi tìm kiếm sản phẩm bằng từ khóa.....	62
Hình 3. 39. Giao diện khi chọn vào 1 danh mục chỉ định	62
Hình 3. 40. Giao diện khi xem một sản phẩm chi tiết.....	63
Hình 3. 41. Giao diện xem chi tiết một hình ảnh của sản phẩm	63
Hình 3. 42. Giao diện xem chi tiết hình ảnh khi xoay qua phải và thu nhỏ	64
Hình 3. 43. Giao diện thông tin chi tiết sản phẩm và khuyến mãi	64
Hình 3. 44. Giao diện phần đánh giá sản phẩm.....	65
Hình 3. 45. Giao diện phân tích cảm xúc đánh giá	65
Hình 3. 46. Giao diện phần bình luận.....	66
Hình 3. 47. Giao diện sửa đổi bình luận.....	66
Hình 3. 48. Giao diện các sản phẩm cùng danh mục	67
Hình 3. 49. Giao diện các sản phẩm được gợi ý	67
Hình 3. 50. Giao diện tìm ảnh tương tự khi nhấn vào các sản phẩm có liên quan...	68
Hình 3. 51. Giao diện phần footer của trang web.....	68
Hình 3. 52. Giao diện trang giỏ hàng	69
Hình 3. 53. Giao diện trang giỏ hàng đang xử lí	69
Hình 3. 54. Giao diện danh sách các giỏ hàng đã được xử lý và thanh toán	70
Hình 3. 55. Giao diện phần đánh giá mức độ hài lòng cho sản phẩm đã mua	70
Hình 3. 56. Giao diện cảm ơn khách hàng đã đánh giá sản phẩm	71
Hình 3. 57. Giao diện danh sách đơn hàng đang được vận chuyển	71
Hình 3. 58. Giao diện danh sách các đơn hàng đã bị huỷ	72
Hình 3. 59. Giao diện hóa đơn của 1 đơn hàng	72
Hình 3. 60. Giao diện trang thanh toán	73
Hình 3. 61. Giao diện thông báo hoàn tất thanh toán	73
Hình 3. 62. Giao diện cập nhật thông tin cá nhân của người dùng	74
Hình 3. 63. Giao diện trang giới thiệu	74
Hình 3. 64. Giao diện trang liên hệ 1	75
Hình 3. 65. Giao diện trang liên hệ 2	76
Hình 3. 66. Giao diện gợi ý khi người dùng dùng tìm kiếm	76

Hình 3. 67. Giao diện Dashboard của hệ thống.....	77
Hình 3. 68. Biểu đồ thống kê tình trạng, số lượng các đơn hàng.....	77
Hình 3. 69. Biểu đồ thống kê doanh thu của hệ thống theo thời gian	78
Hình 3. 70. Biểu đồ thống kê lượng người mua sản phẩm theo thời gian	78
Hình 3. 71. Biểu đồ thống kê Top các sản phẩm được mua gần đây	79
Hình 3. 72. Giao diện quản lí các sản phẩm đang có trên hệ thống	79
Hình 3. 73. Biểu mẫu tạo một sản phẩm mới	80
Hình 3. 74. Form cập nhật một sản phẩm đang tồn tại.....	81
Hình 3. 75. Form nhập thêm hàng cho sản phẩm đang tồn tại.....	81
Hình 3. 76. Giao diện thêm khuyến mãi cho sản phẩm đang tồn tại.....	82
Hình 3. 77. Giao diện danh sách lịch sử các thay đổi giá của sản phẩm.....	82
Hình 3. 78. Giao diện quản lí danh mục.....	83
Hình 3. 79. Form thêm một danh mục mới	83
Hình 3. 80. Form cập nhật một danh mục đang tồn tại	84
Hình 3. 81. Giao diện quản lí các khuyến mãi	84
Hình 3. 82. Form tạo một khuyến mãi mới	85
Hình 3. 83. Giao diện cập nhật một sản phẩm đang tồn tại	86
Hình 3. 84. Giao diện quản lý các đơn đặt hàng	87
Hình 3. 85. Giao diện danh sách các lịch sử trạng thái của đơn hàng.....	87
Hình 3. 86. Giao diện quản lý khách hàng	88
Hình 3. 87. Giao diện thay đổi cấp độ của người bán hàng	89
Hình 3. 88. Giao diện quản lý danh sách người bán hàng.....	90
Hình 3. 89. Giao diện thay đổi cấp độ người dùng dành cho người quản trị	91

DANH MỤC PHỤ LỤC HÌNH

Phụ lục hình 1 Kết nối Google Colab với Drive	125
Phụ lục hình 2 Import các thư viện có liên quan đến keras.....	125
Phụ lục hình 3 Load mạng nơ-ron của VGG16.....	125
Phụ lục hình 4 Hàm tiền xử lí hình ảnh load_image	126
Phụ lục hình 5 Xem thử ngẫu nhiên vector dữ liệu của 1 ảnh trong dataset	127
Phụ lục hình 6 Tra cứu các lớp dự đoán từ model.....	128
Phụ lục hình 7 Khởi tạo model feature_extractor.....	128
Phụ lục hình 8 Kiến trúc model feature_extractor.....	129
Phụ lục hình 9 Copy thư mục dataset ra thư mục làm việc colab	129
Phụ lục hình 10 Quét qua thư mục data để lấy các ảnh đại diện của sản phẩm	130
Phụ lục hình 11 Cây thư mục dataset	130
Phụ lục hình 12 Trích xuất và lưu trữ đặc trưng của các hình ảnh.....	131
Phụ lục hình 13 Khởi tạo đối tượng PCA.....	132
Phụ lục hình 14 Chuyển vector đặc trưng ban đầu sang không gian mới	132
Phụ lục hình 15 Tìm kiếm các ảnh tương tự của một ảnh được chỉ định.....	133
Phụ lục hình 16 Kết quả sau khi tìm ảnh tương tự của một hình ảnh	134
Phụ lục hình 17 Lưu dữ liệu mô hình tìm kiếm ảnh tương tự vào file pkl	134
Phụ lục hình 18 Thiết lập NLTK	135
Phụ lục hình 19 Import các thư viện có liên quan đến Chatbot	135
Phụ lục hình 20 Dồn các file intents thành một file intents.js	136
Phụ lục hình 21 Tokenize các từ trong các đoạn dữ liệu trong intents	137
Phụ lục hình 22 Tạo dữ liệu để training	138
Phụ lục hình 23 Đào tạo mô hình Chatbot	139
Phụ lục hình 24 Lưu trữ lại mô hình Chatbot.....	139
Phụ lục hình 25 Hàm clean_up_sentence và bow của mô hình Chatbot.....	140
Phụ lục hình 26 Hàm classify, getRandomResponseOfATag, responses của mô hình Chatbot	141
Phụ lục hình 27 Kiểm nghiệm mô hình Chatbot	141
Phụ lục hình 28 Cấu trúc File dataset csv cho hệ thống gợi ý	142
Phụ lục hình 29 Import các thư viện phụ thuộc của hệ thống gợi ý	143

Phụ lục hình 30 Cài đặt pyvi	143
Phụ lục hình 31 Load ra các stop word và tạo ra hàm để tiền xử lí dữ liệu	143
Phụ lục hình 32 Tiền xử lí dữ liệu cho hệ thống gợi ý	144
Phụ lục hình 33 Chọn ra các đặc trưng liên quan hệ thống gợi ý.....	144
Phụ lục hình 34 Tính TF-IDF cho các đặc trưng hệ thống gợi ý	145
Phụ lục hình 35 Dùng phép tính độ tương tự cosine Hệ thống gợi ý	145
Phụ lục hình 36 Lấy danh sách tên các sản phẩm – Hệ thống gợi ý	146
Phụ lục hình 37 Lưu lại các thông tin của mô hình – Hệ thống gợi ý.....	146
Phụ lục hình 38 Load ra các dữ liệu đã lưu từ file pkl ra lại	147
Phụ lục hình 39 Kiểm nghiệm lại mô hình – Hệ thống gợi ý.....	147
Phụ lục hình 40 Load ra các stop word – Mô hình nhận dạng cảm xúc.....	148
Phụ lục hình 41 Hàm tiền xử lí dữ liệu – Mô hình nhận dạng cảm xúc.....	148
Phụ lục hình 42 Import các thư viện cần thiết – Mô hình nhận dạng cảm xúc	149
Phụ lục hình 43 Tạo ra một hàm có thể load được data từ folder dataset	149
Phụ lục hình 44 Chia tập dữ liệu thành tập huấn luyện và kiểm tra.....	149
Phụ lục hình 45 Vector hoá văn bản.....	150
Phụ lục hình 46 Huấn luyện mô hình SVM	150
Phụ lục hình 47 Dự đoán và đánh giá mô hình	151
Phụ lục hình 48 Lưu trữ mô hình SVM.....	151
Phụ lục hình 49 Load ra lại mô hình đã lưu trước đó	152
Phụ lục hình 50 Kiểm nghiệm lại mô hình Nhận dạng cảm xúc.....	152

DANH MỤC BẢNG

Bảng 3. 1: Ràng buộc tham chiếu.....	34
Bảng 3. 2. Bảng mô tả các intent của dữ liệu đào tạo chatbot.....	40
Bảng 3. 3. Kết quả thu được sau các lần đào tạo mô hình chatbot.....	43
Bảng 3. 4. Kết quả sau các lần train mô hình thay đổi siêu tham số C	48
Bảng 4. 1. Các chức năng kiểm thử.....	92
Bảng 4. 2. Kịch bản kiểm thử	95
Bảng 4. 3. Kết quả kiểm thử	100
Bảng 7. 1. Mô hình dữ liệu mức vật lý bảng User_type.....	112
Bảng 7. 2. Mô hình dữ liệu mức vật lý bảng Category	112
Bảng 7. 3. Mô hình dữ liệu mức vật lý bảng Product.....	112
Bảng 7. 4. Mô hình dữ liệu mức vật lý bảng Users	113
Bảng 7. 5. Mô tả dữ liệu mức vật lý bảng Material.....	114
Bảng 7. 6. Mô tả dữ liệu mức vật lý bảng Color	114
Bảng 7. 7. Mô tả dữ liệu mức vật lý bảng Dimensions	114
Bảng 7. 8. Mô tả dữ liệu mức vật lý bảng Supplier.....	115
Bảng 7. 9. Mô tả dữ liệu mức vật lý bảng Sale_Off.....	115
Bảng 7. 10. Mô tả dữ liệu mức vật lý bảng Detail_Product_Material	116
Bảng 7. 11. Mô tả dữ liệu mức vật lý bảng Detail_Product_Color.....	116
Bảng 7. 12. Mô tả dữ liệu mức vật lý bảng Detail_SaleOf_Product.....	117
Bảng 7. 13. Mô tả dữ liệu mức vật lý bảng Import_History_Detail.....	117
Bảng 7. 14. Mô tả dữ liệu mức vật lý bảng Import History	118
Bảng 7. 15. Mô tả dữ liệu mức vật lý bảng Detail_Product_Image	118
Bảng 7. 16. Mô tả dữ liệu mức vật lý bảng Product_Price_History.....	118
Bảng 7. 17. Mô tả dữ liệu mức vật lý bảng Cart_Status.....	119
Bảng 7. 18. Mô tả dữ liệu mức vật lý bảng Shoppingcart	119
Bảng 7. 19. Mô tả dữ liệu mức vật lý bảng Cart_Detail.....	120
Bảng 7. 20. Mô tả dữ liệu mức vật lý bảng Ship Method	120

Bảng 7. 21. Mô tả dữ liệu mức vật lý bảng MethodOfPayment.....	121
Bảng 7. 22. Mô tả dữ liệu mức vật lý bảng Order	121
Bảng 7. 23. Mô tả dữ liệu mức vật lý bảng Order_Detail	122
Bảng 7. 24. Mô tả dữ liệu mức vật lý bảng Bill_Status	122
Bảng 7. 25. Mô tả dữ liệu mức vật lý bảng Bill	122
Bảng 7. 26. Mô tả dữ liệu mức vật lý bảng Bill_Status_History	123
Bảng 7. 27. Mô tả dữ liệu mức vật lý bảng Comment.....	124

DANH MỤC TỪ CHUYÊN NGÀNH

Viết tắt	Giải thích
API	Application Programming Interface – Giao diện lập trình ứng dụng
AI	Artificial Intelligence - Trí tuệ nhân tạo
AlexNet	AlexNet là một kiến trúc mạng nơ-ron tích chập (CNN)
B2C	Business-to-Consumer – Mô hình kinh doanh cung cấp sản phẩm & dịch vụ đến người dùng cuối cùng.
B2B	Business-to-Business: Trong mô hình này, một doanh nghiệp cung cấp sản phẩm hoặc dịch vụ cho một doanh nghiệp khác
CSS	Cascading Style Sheets - Ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML)
CDM	Conceptual Data Model – Mô hình dữ liệu mức quan niệm
CSDL	Cơ sở dữ liệu
CNN	Mô hình mạng nơ-ron tích chập (Convolution Neural Network)
DNN	Mô hình mạng nơ-ron sâu (Deep neural network)
DMF	Kỹ thuật phân rã ma trận sâu (Deep Matrix Factorization)
DFD	Data Flow Diagram - Sơ đồ luồng dữ liệu
GoogLeNet	GoogLeNet là một kiến trúc mạng nơ-ron tích chập (CNN) sâu 22 lớp
HTML	HTML - ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web

ILSVRC	ImageNet Large Scale Visual Recognition Challenge - một cuộc thi phần mềm hàng năm do dự án ImageNet tổ chức
JSON	JavaScript Object Notation – là một định dạng dữ liệu dựa trên văn bản (text-based data format) để trao đổi dữ liệu giữa các ứng dụng
JS	JavaScript - ngôn ngữ lập trình thông dịch với khả năng hướng đến đối tượng
LDM	Mô hình dữ liệu mức luận lý
LeNet	LeNet là một kiến trúc mạng nơ-ron tích chập (CNN)
MySQL	Hệ quản trị cơ sở dữ liệu tự do mã nguồn mở
NPM	Node package manager – là một công cụ tạo và quản lý các thư viện lập trình JavaScript cho Node.js
NLP	Xử lý ngôn ngữ tự nhiên (Natural Language Processing)
NN	Neural Network – mạng nơ-ron
PHP	Hypertext Preprocessor - Ngôn ngữ lập trình kịch bản, mã nguồn mở
Pip	Trình quản lý gói tiêu chuẩn cho Python
SEO	Search Engine Optimization - Tối ưu hóa công cụ tìm kiếm là quá trình tăng chất lượng và lưu lượng truy cập website bằng cách tăng khả năng hiển thị của website hoặc webpage
SPA	Single Page Application - Ứng dụng web chỉ có một trang duy nhất
SVM	Support Vector Machine - là một mô hình phân loại hoạt động bằng việc xây dựng một siêu phẳng (hyperplane)
TF-IDF	Term frequency-inverse document frequency, là một thống kê số học nhằm

	phản ánh tầm quan trọng của một từ đối với một văn bản trong một tập hợp hay một ngũ liệu văn bản
VueJS	Framework linh động dùng để xây dựng giao diện người dùng mã nguồn mở của JavaScript
VAT	Value-Added Tax - thuế giá trị gia tăng
VGG16	VGG16 là một mô hình mạng nơ-ron tích chập (CNN) được sử dụng để nhận dạng hình ảnh

CHƯƠNG 1. GIỚI THIỆU

1.1. Đặt vấn đề

Trong thời đại ngày nay, sự phát triển của công nghệ và mạng internet đã mở ra những cơ hội mới cho việc kinh doanh trực tuyến. Trong lĩnh vực thiết bị thể thao, việc tạo ra một trang web bán hàng hiệu quả không chỉ đòi hỏi sự sáng tạo trong thiết kế sản phẩm mà còn đặt ra thách thức về cách tối ưu hóa trải nghiệm người dùng và cách tìm kiếm sản phẩm.

Mặc dù có nhiều trang web bán thiết bị thể thao đa dạng, nhưng vẫn đề thường gặp là khả năng tìm kiếm sản phẩm một cách nhanh chóng và chính xác. Người dùng thường phải dành nhiều thời gian để duyệt qua các trang web, điều này có thể dẫn đến sự mệt mỏi và giảm trải nghiệm của họ.

Với những thách thức này, nghiên cứu và áp dụng các phương pháp tiên tiến, như machine learning và data analytics, có thể giúp tối ưu hóa trang web, tạo ra trải nghiệm mua sắm tốt hơn và cung cấp các gợi ý sản phẩm phù hợp. Điều này không chỉ mang lại lợi ích cho người dùng mà còn giúp doanh nghiệp nâng cao khả năng cạnh tranh và tăng cường doanh số bán hàng.

Do đó, đề tài này đặt ra câu hỏi: "Làm thế nào chúng ta có thể áp dụng công nghệ hiện đại để cải thiện trải nghiệm mua sắm trực tuyến trong lĩnh vực thiết bị thể thao và tối ưu hóa quá trình tìm kiếm sản phẩm cho người dùng?" Đồng thời, nó hướng dẫn đến việc khám phá và phát triển các giải pháp sáng tạo để giải quyết những thách thức này, tạo ra một không gian trực tuyến độc đáo và hấp dẫn cho khách hàng.

1.2. Nghiên cứu liên quan

1.2.1. Nhận dạng hình ảnh

Hầu hết các đột phá về học sâu đều liên quan tới mạng thần kinh tích chập (CNNs), nó cũng là một lĩnh vực chính trong mạng lưới thần kinh sâu (DNNs). Các phương pháp CNN hiện đại có thể nhận dạng hình ảnh chính xác hơn con người. Thủ thách nhận dạng hình ảnh quy mô lớn ImageNet (ILSVRC) [1] là một cuộc thi học thuật mang tính đại diện cao dành cho các giải pháp thị giác máy dựa trên dữ liệu hình ảnh do ImageNet cung cấp. Bộ dữ liệu chính bao gồm hơn 14 triệu hình ảnh được đánh dấu, với một tập hợp con nhỏ hơn được lấy mẫu cho ILSVRC hàng năm. Kết quả ILSVRC tốt nhất trước năm 2012 đạt tỉ lệ lỗi 26%. Tuy nhiên, mô hình CNN dựa trên AlexNet [2] đã giảm tỉ lệ lỗi xuống 16,4% vào năm 2012, giành chức vô địch. Các nghiên cứu sau đó đã sử dụng nhiều cách tiếp cận CNN khác nhau, một số đã trở thành những kiến trúc CNN chính và nổi tiếng.

Nhiều sản phẩm hiện nay sử dụng công nghệ deep learning, hầu hết liên quan đến nhận dạng hình ảnh. Những đột phá phần cứng gần đây và những cải tiến đáng kể đã mang lại sự tập trung mạnh mẽ vào học sâu, với các mô hình CNN trở thành phương pháp phổ

biến nhất để nhận dạng hình ảnh, phân đoạn hình ảnh [3] [4] [5] và nhận dạng đối tượng. Nhận dạng hình ảnh đề cập đến quá trình máy được huấn luyện sử dụng CNN để trích xuất các tính năng quan trọng từ bộ dữ liệu hình ảnh lớn, kết hợp chúng thành bản đồ tính năng và thực hiện nhận dạng bằng cách kết nối các nơ-ron.

Sự phát triển của deep learning và machine learning gần đây có nghĩa là hiệu suất của phương pháp xử lý hình ảnh truyền thống để nhận dạng hình ảnh không còn có thể so sánh được với các phương pháp tiếp cận dựa trên mạng thần kinh (NN). Do đó, nhiều nghiên cứu đã đề xuất các thuật toán học sâu được tối ưu hóa để cải thiện độ chính xác nhận dạng hình ảnh cho các tình huống nhận dạng khác nhau. CNN là phương pháp nổi bật nhất để nhận dạng hình ảnh, cải thiện độ chính xác của nhận dạng bằng cách tăng độ sâu lớp ẩn và thu được nhiều tham số đặc trưng hơn một cách hiệu quả. Các ứng dụng nhận dạng hình ảnh thành công bao gồm nhận dạng khuôn mặt, vật thể và biển số xe,... Các hệ thống nhận dạng đã được đào tạo trước bằng cách sử dụng hình ảnh LeNet [6], AlexNet [2], GoogLeNet [7] và VGG16 [8].

1.2.2. TensorFlow và TFLearn

TensorFlow, một thư viện tính toán số đa năng được Google mở nguồn vào tháng 11 năm 2015, có kiến trúc và triển khai linh hoạt cho phép người dùng tập trung vào việc xây dựng biểu đồ tính toán và triển khai mô hình mà không cần nỗ lực nhiều trên các nền tảng không đồng nhất như thiết bị di động, hàng trăm máy hoặc hàng nghìn máy. của các thiết bị tính toán. TensorFlow nhìn chung rất dễ sử dụng theo nghĩa là hầu hết những người tìm kiếm lại trong lĩnh vực nghiên cứu mà không có kinh nghiệm sử dụng thư viện này đều có thể hiểu được điều gì đang xảy ra ngay sau các khối mã. TensorFlow cung cấp nền tảng tốt để xây dựng các hình dạng khác nhau của ứng dụng học máy. [9]

TFLearn là một giao diện được đơn giản hóa cho TensorFlow, để giúp mọi người bắt đầu phân tích dự đoán và khai thác dữ liệu. Nó giúp làm tròn quá trình chuyển đổi từ thế giới học máy đơn tuyển trong Scikit-learn sang thế giới cởi mở hơn trong việc xây dựng các hình dạng khác nhau của mô hình học máy. Người dùng có thể bắt đầu bằng cách sử dụng kiểu phù hợp/dự đoán quen thuộc và chuyển sang sử dụng API TensorFlow để cảm thấy thoải mái hơn. Nó tương thích với Scikit-learn nên người dùng cũng có thể hưởng lợi từ các tính năng của Scikit-learn như GridSearch và Pipeline.

1.2.3. Hệ thống gợi ý

Hệ thống đề xuất (Recommender systems) đã trở nên vô cùng phổ biến và được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm phim, sách, bài báo nghiên cứu và các thẻ xã hội. Có ba loại thuật toán đề xuất cơ bản: lọc hợp tác (collaborative filtering), lọc dựa trên nội dung (content-based filtering) và đề xuất kết hợp (hybrid recommendation).

Phương pháp lọc hợp tác dựa trên việc thu thập và phân tích một lượng lớn thông tin về hành vi, hoạt động hoặc sở thích của người dùng và dự đoán những gì người dùng sẽ thích dựa trên sự tương đồng với người dùng khác [10]. Thuật toán được đề xuất không

cần kiến thức chuyên sâu và tác động để xuất sỹ trở nên tốt hơn theo thời gian với sự quan tâm của người dùng, nhưng có nhược điểm về độ dày dữ liệu và các vấn đề khác. Có hai loại kỹ thuật lọc hợp tác: lọc dựa trên sản phẩm (Item-based) [11] và lọc dựa trên người dùng (User-based) [12] collaborative filtering[CF]. Sự tương đồng giữa các sản phẩm hoặc người dùng có thể được tính bằng cách sử dụng các độ đo tương đồng khác nhau như cosine, Euclidean, jaccard [13].

Phương pháp lọc dựa trên nội dung [11] [14] dựa trên mô tả của sản phẩm và hồ sơ ưa thích của người dùng. Những thuật toán này cố gắng để xuất những sản phẩm giống những sản phẩm mà người dùng đã thích trong quá khứ.

Đề xuất kết hợp [10] [14] là sự kết hợp giữa lọc hợp tác và lọc dựa trên nội dung. Những phương pháp này cũng có thể được sử dụng để vượt qua một số vấn đề phổ biến trong hệ thống đề xuất như vấn đề khởi động lạnh và vấn đề độ dày dữ liệu [15].

Hầu hết các nghiên cứu sử dụng hàm Tương đồng Vector Cosine để tính độ tương đồng giữa các đánh giá sách để đề xuất sách, để xem xét việc người dùng có các bảng đánh giá khác nhau. Một số người thường đánh giá cao trong khi người khác thường đánh giá thấp [13].

1.2.4. Phân loại văn bản

Phân loại văn bản – Text categorization (TC – còn được gọi là phân loại văn bản hoặc tìm chủ đề) là nhiệm vụ tự động sắp xếp một tập hợp tài liệu thành các danh mục (hoặc lớp hoặc chủ đề) từ một tập hợp được xác định trước [16]. Khái niệm phân loại văn bản là việc phân loại các tài liệu thành một số cố định các danh mục hoặc lớp được xác định trước [17]. Phân loại tài liệu có thể được định nghĩa là nhiệm vụ tự động phân loại các bộ sưu tập tài liệu điện tử thành các lớp được chú thích dựa trên nội dung của chúng [18]. Mỗi tài liệu có thể thuộc chính xác một, nhiều hoặc không có danh mục nào cả. Sử dụng học máy, mục tiêu là học các bộ phân loại từ các ví dụ thực hiện các bài tập danh mục một cách tự động. Đây là một vấn đề học tập có giám sát. Vì các danh mục có thể trùng lặp nên mỗi danh mục được coi là một vấn đề phân loại nhị phân riêng biệt [17]. Việc phân loại giúp xác định chính xác danh mục tên miền đang sử dụng, một tệp văn bản nhất định liên quan đến [19].

SVM lần đầu tiên được đề xuất bởi Vapnik [20] [21] và nhóm tại Phòng thí nghiệm AT&T Bell đã đề xuất một kỹ thuật mới Máy vectơ hỗ trợ [SVM] cung cấp một nhóm các phương pháp học có giám sát có thể được áp dụng để phân loại hoặc hồi quy [21] [22] [23]. Học có giám sát là một cách tiếp cận rất phổ biến, trong đó thuật toán Phân loại văn bản (TC) học các mẫu phân loại từ một tập hợp các ví dụ được gắn nhãn, cho một số lượng đủ lớn các ví dụ được gắn nhãn (tập huấn luyện) và nhiệm vụ là xây dựng mô hình TC. Sau đó có thể sử dụng mô hình TC để dự đoán danh mục (lớp) của các ví dụ mới chưa thấy (bộ kiểm tra) [24]. SVM học cách xây dựng siêu phẳng N chiều làm bì mặt quyết định sao cho biên độ phân tách giữa các ví dụ dương và âm được tối đa hóa [25]. Để triển khai SVM, Phần mềm có tên LIBSVM của Chung Chang và Chih-Jen Lin đã được sử dụng. LIBSVM là phần mềm tích hợp để hỗ trợ phân loại vectơ, hồi quy và

ước tính phân phối [SVM một lớp]. Nó hỗ trợ phân loại nhiều lớp [22] [26]. Mục tiêu của việc sử dụng LIBSVM là xác định các kết quả tích cực để bộ phân loại có thể dự đoán chính xác dữ liệu chưa biết [tức là dữ liệu thử nghiệm]. Các giá trị từ tập thử nghiệm được đưa vào công cụ LIBSVM để đào tạo và dự đoán tập dữ liệu cũng như quá trình phân tích được thực hiện [22].

1.3. Mục tiêu đề tài

Để đảm bảo rằng website bán hàng "Bán thiết bị thể thao" đáp ứng nhu cầu của người tiêu dùng và tối ưu hóa hiệu suất kinh doanh, đề tài này đặt ra một số mục tiêu trọng tâm như sau:

- **Phân tích thị trường:** Hiểu rõ nhu cầu và xu hướng của người tiêu dùng trong lĩnh vực thể thao, từ đó định hình các sản phẩm chính và phân đoạn thị trường mục tiêu.
- **Tối ưu trải nghiệm người dùng:** Thiết kế giao diện website dễ sử dụng, thân thiện, giúp người dùng dễ dàng tìm kiếm, so sánh và mua sắm sản phẩm.
- **Nâng cao hiệu suất kinh doanh:** Xây dựng hệ thống quản lý hàng tồn kho, phân tích dữ liệu khách hàng, và tối ưu hóa chiến lược tiếp thị, giúp gia tăng doanh số và lợi nhuận.
- **Bảo mật thông tin:** Phát triển các giải pháp bảo mật để đảm bảo an toàn cho thông tin khách hàng và giao dịch trực tuyến.
- **Mô hình kinh doanh:** Xác định mô hình kinh doanh phù hợp, liệu có nên áp dụng mô hình B2C, B2B hay kết hợp cả hai, và làm thế nào để tích hợp dịch vụ hậu mãi, chăm sóc khách hàng.
- **Tích hợp công nghệ:** Khám phá và áp dụng các công nghệ mới như trí tuệ nhân tạo, phân tích dữ liệu lớn để nâng cao chất lượng dịch vụ và tạo ra sự khác biệt trên thị trường.

Qua việc xác định và tập trung giải quyết những vấn đề trên, đề tài mong muốn mang đến một giải pháp hoàn thiện cho website bán hàng, góp phần thúc đẩy sự phát triển của ngành Hệ thống thông tin trong thời đại số.

1.4. Đối tượng và phạm vi đề tài

1.4.1. Đối tượng nghiên cứu

Đề tài này tập trung nghiên cứu vào hai nhóm đối tượng chính:

Người sử dụng trang web: Đối tượng chính là những người truy cập và sử dụng trang web về thiết bị thể thao. Nghiên cứu sẽ khám phá nhu cầu, thói quen mua sắm, và mong đợi của họ để hiểu rõ hơn về cách họ tương tác với trang web và mong muốn gặp phải những trải nghiệm nào khi tìm kiếm sản phẩm.

Người quan trọng nghiệp vụ: Đối tượng này bao gồm các chuyên gia nghiên cứu và phát triển các giải pháp tối ưu hóa trang web. Nghiên cứu sẽ tìm hiểu về các ưu tiên kinh doanh, chiến lược tiếp thị, và cách họ có thể tận dụng công nghệ để cải thiện trải nghiệm người dùng và quy trình tìm kiếm.

1.4.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài này được xác định như sau:

Thời gian: Nghiên cứu sẽ diễn ra trong khoảng [thời gian dự kiến], tập trung vào việc phát triển và triển khai các giải pháp tối ưu hóa trang web.

Khía cạnh công nghệ: Đề tài sẽ tập trung vào sử dụng machine learning và data analytics để cải thiện quá trình tìm kiếm sản phẩm và đề xuất gợi ý cho người dùng.

Phạm vi địa lý: Phạm vi nghiên cứu sẽ tập trung vào thị trường ở Việt Nam hoặc có thể được mở rộng ra để bao gồm một số thị trường quốc tế.

Bằng cách xác định rõ đối tượng và phạm vi nghiên cứu, đề tài này nhằm mục đích đảm bảo sự tập trung và hiệu quả trong quá trình nghiên cứu và triển khai giải pháp.

CHƯƠNG 2. MÔ TẢ BÀI TOÁN

2.1. Mô tả chi tiết đề tài

Website bán hàng thiết bị thể thao được thiết kế với mục tiêu chủ đạo là cung cấp trải nghiệm mua sắm trực tuyến tối ưu cho khách hàng, đặc biệt là những người quan tâm đến các dụng cụ hỗ trợ tập thể thao. Hệ thống quản lý đơn hàng và giỏ hàng được xây dựng để đáp ứng đầy đủ nhu cầu thông tin của người mua hàng và người bán hàng.

Người mua hàng sẽ được cung cấp các thông tin chi tiết về đơn đặt hàng, bao gồm mã đơn hàng, ngày tạo đơn hàng, phí VAT, tổng giá trị sản phẩm, tổng phí VAT, tổng giá trị đơn hàng trước khi áp dụng khuyến mãi, và tổng giá trị đơn hàng sau khi áp dụng khuyến mãi. Ngoài ra, họ cũng có thể quản lý chi tiết giỏ hàng với mã sản phẩm, màu sắc, chất liệu, mã giỏ hàng, và số lượng của mỗi sản phẩm trong giỏ.

Đối với người bán hàng, họ có khả năng quản lý thông tin người dùng như email, username, số điện thoại, địa chỉ, mật khẩu, và thời gian tạo tài khoản. Họ cũng có thể cập nhật trạng thái của đơn đặt hàng, bao gồm đang vận chuyển, đã thanh toán, chưa thanh toán, hoặc đơn đã bị huỷ. Các số liệu thống kê như doanh thu, số lượng đơn đặt hàng chưa xử lý, số lượng khách hàng đăng ký, tổng số sản phẩm tồn kho, và xu hướng mua sắm theo thời gian cũng được cung cấp để hỗ trợ quản lý kinh doanh. Danh mục sản phẩm và thông tin khuyến mãi cũng được quản lý một cách hiệu quả, bao gồm mã danh mục, tên danh mục, biểu tượng danh mục, mã khuyến mãi, tên khuyến mãi, ngày bắt đầu và kết thúc khuyến mãi.

Người quản trị website sẽ có quyền truy cập toàn diện, quản lý thông tin người dùng, theo dõi và xác nhận đơn đặt hàng, và theo dõi các báo cáo thống kê về doanh thu và xu hướng mua sắm. Họ cũng có thể quản lý danh mục sản phẩm và chiến lược khuyến mãi, đồng thời kiểm soát quyền truy cập để bảo vệ an toàn dữ liệu trang web. Điều này giúp duy trì một môi trường mua sắm trực tuyến hiệu quả và an toàn.

Dưới đây là mô tả chi tiết về hệ thống:

2.1.1. Chức năng chính

Trang chính: Hiển thị danh mục sản phẩm, các sản phẩm nổi bật, khuyến mãi, và sản phẩm mới nhất.

Tìm kiếm sản phẩm: Cho phép người dùng tìm kiếm sản phẩm dựa trên tên, danh mục và tìm kiếm hình ảnh tương tự dựa vào trí tuệ nhân tạo.

Lọc sản phẩm: Cho phép người dùng sử dụng bộ lọc để lọc giá sản phẩm từ thấp lên cao hoặc từ cao xuống thấp

Thêm sản phẩm vào giỏ hàng: Người dùng có thể chọn sản phẩm ưng ý và đưa vào giỏ hàng. Sau đó người dùng có thể truy cập vào giỏ hàng để chỉnh sửa số lượng hoặc xóa sản phẩm đi khỏi giỏ hàng.

Thanh toán: Người dùng tiến hành thanh toán giờ hàng đã chọn hoặc một sản phẩm chỉ định một cách an toàn và thuận tiện. Bên cạnh đó còn có các tùy chọn như hình thức thanh toán giúp người dùng có nhiều lựa chọn phù hợp với những hoàn cảnh khác nhau.

Đánh giá và bình luận: Khách hàng có thể để lại đánh giá và bình luận về sản phẩm mà họ đã mua.

Quản lý tài khoản: Người dùng có thể tạo và quản lý tài khoản cá nhân, theo dõi đơn hàng và lịch sử mua sắm.

Hỗ trợ trực tuyến: Tích hợp chức năng chat trực tuyến giúp khách hàng liên hệ và được hỗ trợ nhanh chóng từ nhân viên kinh doanh.

2.1.2. Nhóm người dùng

Hệ thống sẽ có 4 nhóm người dùng bao gồm: người dùng khách hàng chưa đăng nhập, người dùng khách hàng đã đăng nhập và người dùng là chủ cửa hàng, người dùng là quản trị viên. Sau đây là chi tiết các chức năng của từng nhóm người dùng

- Người dùng khách hàng chưa đăng nhập
 - Xem danh sách sản phẩm.
 - Xem chi tiết một sản phẩm.
 - Xem trang liên hệ, giới thiệu của cửa hàng.
 - Xem bình luận và đánh giá của người khác về sản phẩm nào đó
 - Đăng ký tài khoản: người dùng điền đầy đủ thông tin cần thiết để tạo tài khoản.
 - Đăng nhập: người dùng đăng nhập vào trang web bằng email mà mật khẩu mà người dùng đã đăng ký trước đó.
 - Tìm kiếm sản phẩm: người dùng được phép tìm kiếm sản phẩm theo từ khoá, tìm kiếm bằng hình ảnh hoặc tìm kiếm các sản phẩm trong danh mục mong muốn.
 - Lọc sản phẩm: người dùng có thể lọc sản phẩm theo giá từ thấp lên cao hoặc từ cao xuống thấp.
- Người dùng khách hàng đã đăng nhập
 - Sử dụng được hết các chức năng của người dùng chưa đăng nhập.
 - Bình luận vào một sản phẩm nào đó, xoá hoặc cập nhật bình luận thuộc sở hữu của mình.
 - Thêm vào sản phẩm vào giỏ hàng.

- Xoá sản phẩm khỏi giỏ hàng.
- Tăng giảm số lượng của sản phẩm đã đặt vào giỏ hàng.
- Thanh toán giỏ hàng
- Xem các đơn hàng và thông tin chi tiết của đơn hàng
- Đăng xuất tài khoản

➤ Người dùng là chủ cửa hàng

- Sử dụng được hết các chức năng của người dùng chưa đăng nhập và người dùng đã đăng nhập.
- Thêm sản phẩm mới vào hệ thống, cập nhật và xoá sản phẩm đã tồn tại.
- Thêm danh mục mới vào hệ thống, cập nhật và xoá danh mục đã tồn tại.
- Thêm một khuyến mãi mới vào hệ thống, cập nhật hoặc xoá khuyến mãi đã tồn tại.
- Thêm/cập nhật một khuyến mãi đã tồn tại vào cho một sản phẩm đang tồn tại.
- Nhập thêm hàng cho sản phẩm đang tồn tại.
- Xuất danh sách các thông kê trên hệ thống ra file excel.
- Xác nhận đơn hàng, huỷ đơn hàng, đánh dấu đơn hàng đang vận chuyển.
- Quản lý danh sách nhóm người dùng đã đăng ký trên hệ thống.
- Tìm kiếm tên sản phẩm, mã sản phẩm, sắp xếp thứ tự mã sản phẩm, tên sản phẩm, tên danh mục theo nhóm alphabet.
- Ở phần danh sách các khuyến mãi có thể sắp xếp thứ tự mã khuyến mãi, phần trăm khuyến mãi, ngày bắt đầu, ngày kết thúc khuyến mãi theo nhóm alphabet.
- Ở phần danh sách các đơn hàng có thể tìm kiếm theo tên đơn hàng, sắp xếp ngày đặt đơn, tình trạng, tên khách hàng theo thứ tự alphabet.
- Có thể xem được các danh sách các đơn hàng đã được thanh toán, chưa được thanh toán, đơn hàng đang vận chuyển, đơn hàng đã huỷ và tất cả đơn hàng.

➤ Người dùng là quản trị viên

- Sử dụng được hết các chức năng của người dùng chưa đăng nhập và người dùng đã đăng nhập.
- Quản lý danh sách nhóm người dùng bán hàng.

2.1.3. Đặc điểm nổi bật

Giao diện thân thiện: Thiết kế đẹp mắt, trực quan và dễ dàng sử dụng trên mọi thiết bị. Vì sử dụng VueJS nên website ở dạng SPA (Single Page Application), nên khi chuyển giữa các trang sẽ tạo cảm giác mượt mà và linh hoạt cho người dùng.

Tối ưu hóa SEO: Giúp website dễ dàng được tìm thấy trên các công cụ tìm kiếm.

Bảo mật cao: Sử dụng các giải pháp bảo mật đảm bảo an toàn cho thông tin của khách hàng và giao dịch trực tuyến.

2.1.4. Giải pháp khắc phục

Hệ thống quản lý tồn kho: Tự động cập nhật số lượng hàng tồn kho, giúp ngăn chặn tình trạng hết hàng hoặc quá tải.

Phân tích dữ liệu: Tích hợp công nghệ phân tích dữ liệu giúp người bán nắm bắt xu hướng và ưu tiên chiến lược tiếp thị một cách hiệu quả.

Hệ thống khuyến mãi linh hoạt: Cho phép quản trị viên tạo ra các chương trình khuyến mãi dễ dàng, thu hút khách hàng và tăng doanh số bán hàng.

Với những chức năng và đặc điểm trên, website này hứa hẹn sẽ mang lại một trải nghiệm mua sắm trực tuyến tuyệt vời cho người tiêu dùng, đồng thời giúp doanh nghiệp nắm bắt cơ hội trong thị trường thiết bị thể thao đang phát triển.

2.2. Hướng tiếp cận giải quyết của đề tài

2.2.1. Phương pháp tiếp cận

Mô hình hướng đối tượng (Object-Oriented Approach): Đối với việc phát triển hệ thống web bán hàng, việc sử dụng mô hình hướng đối tượng là lựa chọn phổ biến và hiệu quả. Qua mô hình này có thể dễ dàng mô tả, thiết kế và triển khai các chức năng của hệ thống dưới dạng các đối tượng, thuộc tính và phương thức.

Mô hình CDM (Conceptual Data Model): Để xây dựng cơ sở dữ liệu cho website, em sẽ sử dụng mô hình CDM. Mô hình này giúp chúng ta biểu diễn cấu trúc dữ liệu và mối quan hệ giữa các thực thể trong hệ thống.

2.2.2. Công nghệ và ngôn ngữ lập trình

Ứng dụng Web: Sử dụng PHP với framework Laravel cho phía server-side. Laravel là một framework mạnh mẽ, bảo mật và phổ biến, giúp xây dựng ứng dụng web nhanh chóng và hiệu quả.

Ứng dụng Frontend

- **HTML,CSS, Typescript:** Những ngôn ngữ cơ bản như HTML để định nghĩa cấu trúc trang, CSS để thiết kế giao diện, và Typescript để tạo ra trải nghiệm người dùng tương tác và mang lại lợi ích về kiểm tra kiểu và bảo mật code
- **Vue.js 3 Composition API:** VueJS là một framework Javascript để sử dụng và linh hoạt, sử dụng VueJS xây dựng giao diện người dùng một cách hiệu quả, dễ bảo trì và mở rộng.

Cơ sở dữ liệu: MySQL sẽ được sử dụng như là hệ quản trị cơ sở dữ liệu quan hệ. MySQL là một lựa chọn phổ biến cho các ứng dụng web bởi vì nó ổn định, hiệu quả và dễ sử dụng.

Công cụ hỗ trợ:

- **Docker:** Tích hợp Docker vào cơ cấu phát triển của dự án cũng đóng một vai trò quan trọng, nó giúp cho ứng dụng web luôn đồng nhất, cô lập và dễ quản lý trong quá trình phát triển và triển khai
- **Git:** Sử dụng Git để quản lý mã nguồn, theo dõi sự thay đổi.

Qua việc tích hợp những công nghệ và ngôn ngữ lập trình này, đề tài sẽ mang sự ổn định, linh hoạt, và hiệu suất cao cho hệ thống web bán hàng.

2.2.3. Frameworks và thư viện

Laravel: Framework PHP phổ biến cho phát triển ứng dụng web.

Vue.js: Framework JavaScript được sử dụng để xây dựng giao diện người dùng động và tương tác.

TypeScript: Ngôn ngữ lập trình mở rộng từ JavaScript, cung cấp kiểm tra kiểu tĩnh và tính năng lập trình hướng đối tượng.

Bằng việc kết hợp các phương pháp tiếp cận và công nghệ đã nêu trên, đề tài này hứa hẹn sẽ xây dựng được một hệ thống bán hàng trực tuyến hiện đại, bảo mật và thân thiện với người dùng.

2.3. Cơ sở lí thuyết

2.3.1. Deep Learning (Mô hình học sâu)

Mô hình học sâu là một nhánh của mô hình máy học, tập trung vào việc giải quyết vấn đề liên quan đến mạng thần kinh, mô hình học sâu cũng được hiểu như là một bộ não của máy tính cho những hệ thống thông minh khi cho nó tự học hỏi với nhiều tập dữ liệu khác nhau, mô hình có khả năng ghi nhớ tốt sẽ được chọn tích hợp vào các hệ thống nhằm phục vụ cho con người áp dụng vào các công trình thực tiễn với mục đích ý nghĩa khác nhau. Mỗi mô hình thuộc mô hình học sâu sẽ có những ứng dụng và đáp ứng các yêu cầu khác nhau. Ví dụ các ứng dụng thực tiễn mà mô hình học sâu mang lại như: Hệ thống lái xe tự động, trợ lý ảo, nhận diện hình ảnh, trích xuất văn bản từ âm thanh và chuẩn đoán liên quan đến lĩnh vực y tế, v.v...

Các bài toán trong lĩnh vực thị giác máy tính (Computer Vision) như bài toán phân lớp, bài toán định vị đối tượng, bài toán phát hiện đối tượng và bài toán phân vùng ảnh.

Bài toán Classification (phân lớp ảnh): Đây là bài toán dự đoán nhãn của một đối tượng khi cho đầu vào là một ảnh chứa một đối tượng và đầu ra là đối tượng trong ảnh đó thuộc nhãn nào (hay nói cách khác đầu ra là dự đoán nhãn cho đối tượng).

Bài toán Object localization (định vị đối tượng): là một bài toán xác định vị trí của đối tượng, đối tượng phát hiện được sẽ có một hộp hình chữ nhật bao quanh đối tượng đó.

Bài toán Object Detection (phát hiện đối tượng): là sự kết hợp của 2 bài toán phân lớp ảnh và bài toán định vị đối tượng, nhưng có sự khác biệt hơn là bài toán phát hiện đối tượng không chỉ đơn thuần là phát hiện và tạo hộp giới hạn cho một đối tượng đang được quan tâm mà bài toán còn phân biệt được nhiều đối tượng khác nhau, mỗi đối tượng được phát hiện sẽ có tên của đối tượng đó và hộp giới hạn tương ứng.

Bài toán Image Segmentation (phân vùng ảnh): bài toán phân vùng ảnh có tác dụng phân vùng các đối tượng được quan tâm ra thành nhiều phần khác nhau.

2.3.2. Mạng nơ-ron tích chập CNN

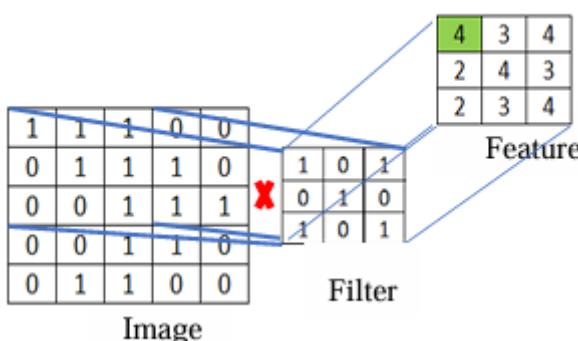
Hiện nay có rất nhiều hệ thống thông minh được ra đời phục vụ vào đời sống xã hội ngày càng nhiều nhờ vào tính hiệu quả, tốc độ tính toán và độ chính xác của mạng CNN mang lại.

Về mặt kỹ thuật mạng CNN bao gồm các thành phần: Lớp tích chập (Convolution layer), Lớp tổng hợp (Pooling Layer) và lớp kết nối đầy đủ (Fully Connected Layer)

Lớp tích chập (Convolution Layer)

Đây là lớp đầu tiên cũng là lớp quan trọng nhất trong mô hình mạng CNN là thành phần đảm nhiệm việc phát hiện và tính mọi phép toán.

Lớp tích chập nhận đầu vào là một ma trận hình ảnh sau đó ma trận này được áp dụng với nhiều bộ lọc khác nhau. Sử dụng phép tích chập giữa ảnh đầu vào với các bộ lọc để tạo ra bản đồ đặc trưng.

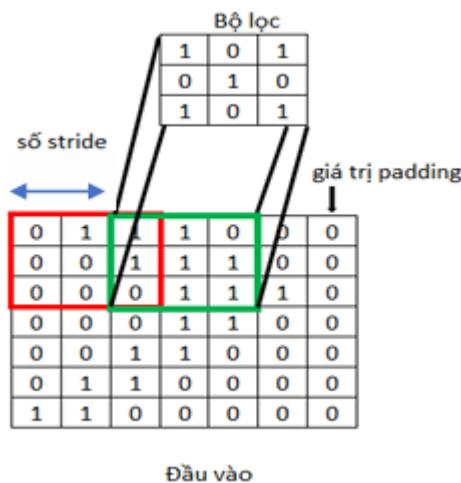


Hình 2. 1. Ví dụ tích chập ảnh ma trận 5x5 với bộ lọc 3x3

Như ví dụ **Hình 2. 1** thể hiện tính tích chập nhận đầu vào là một ma trận ảnh kết hợp với bộ lọc kích thước 3×3 . Hai ma trận này tiến hành thực hiện phép tích chập từ trái sang phải từ trên xuống dưới một khoảng trượt (stride), theo như ví dụ **Hình 2. 1** khoảng cách trượt là 1. Mỗi lần thực hiện tích chập sẽ tạo ra 1 giá trị trong bản đồ đặc trưng. Quá trình tích chập hoàn thành sau khi nhận được một ma trận bản đồ đặc trưng hoàn chỉnh và làm đầu vào cho tầng tiếp theo.

Stride: Là khoảng cách dịch chuyển của bộ lọc sau mỗi lần tích chập trên từng vị trí ảnh. Ví dụ Stride = 2 thì mỗi lần tính tích chập bộ lọc tiến hành dịch chuyển sang phải 2 ô, việc dịch chuyển xuống dưới cũng tương tự

Padding: có giá trị 0 hoặc 1 được dùng làm bộ đệm cho ma trận đầu vào, padding được sử dụng phổ biến nhất là 0. Ví dụ về sử dụng khoảng trượt và padding được mô phỏng dưới **Hình 2. 2**

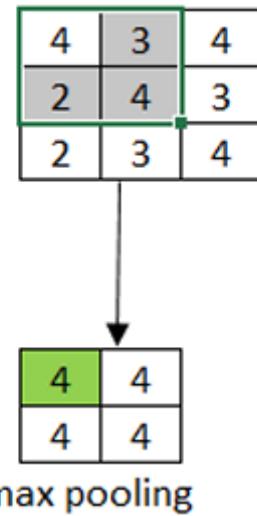


Hình 2. 2. Ví dụ cách trượt stride = 2 và dùng đệm padding = 0

Hàm Relu: là một hàm tuyến tính được sử dụng nhiều nhất khi đào tạo mô hình. Có chức năng loại bỏ những giá trị âm và giữ lại các giá trị dương.

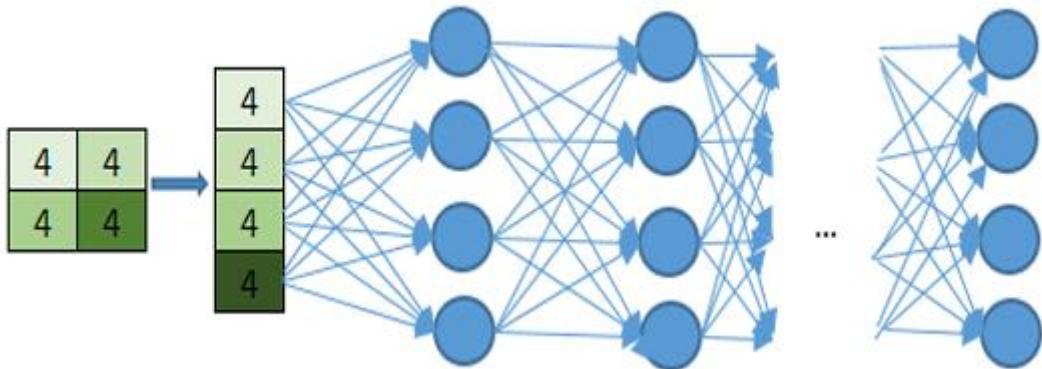
Lớp tổng hợp (Pooling Layer): Lớp tổng hợp được sử dụng sau tầng tích chập, tầng này làm đầu vào cho tầng tổng hợp có tác dụng làm giảm kích thước ma trận đầu vào. Việc giảm kích thước ở tầng này vẫn bảo toàn được các đặc trưng của dữ liệu đầu vào. Dạng tầng tổng hợp thường xuyên sử dụng là: Max Poling và Avg Pooling

Max Pooling: trả về giá trị lớn nhất trong khu vực ảnh được chọn, ví dụ **Hình 2. 3** về max pooling tích chập bộ lọc (2,2)



Hình 2. 3. Ví dụ tầng pooling sử max pooling

Lớp kết nối đầy đủ (Fully Connected Layer): Lớp kết nối đầy đủ có chức năng tạo ra mô hình sau và phân loại đầu ra khi quá trình các ma trận làm phẳng kết nối đến các mạng neuron đã được học hỏi với nhau (xác thực chéo với nhau). Các ma trận được làm phẳng kết nối đến các mạng neuron được mô phỏng như **Hình 2. 4**



Hình 2. 4. Tầng kết nối đầy đủ nhận đầu vào dữ liệu làm phẳng

2.3.3. Mạng nơ-ron sâu DNN

DNN là một loại mạng nơ-ron nhân tạo trong lĩnh vực trí tuệ nhân tạo. DNN được xây dựng dựa trên kiến trúc mạng lưới thần kinh truyền thống, có nhiều tầng ẩn giữa tầng đầu vào và tầng đầu ra.

Cấu trúc của DNN gồm nhiều tầng ẩn, mỗi tầng ẩn chứa nhiều nơ-ron. Mỗi nơ-ron trong mạng lưới thần kinh sẽ nhận đầu vào từ các nơ-ron xung quanh và tính toán giá trị đầu ra dựa trên trọng số và hàm kích hoạt.

Quá trình huấn luyện DNN bao gồm hai giai đoạn chính: tiền xử lý và lan truyền ngược. Trong giai đoạn tiền xử lý, dữ liệu đầu vào sẽ được chuẩn hóa và chuyển thành các đặc trưng phù hợp để đưa vào mạng lưới. Sau đó, quá trình lan truyền ngược được sử dụng

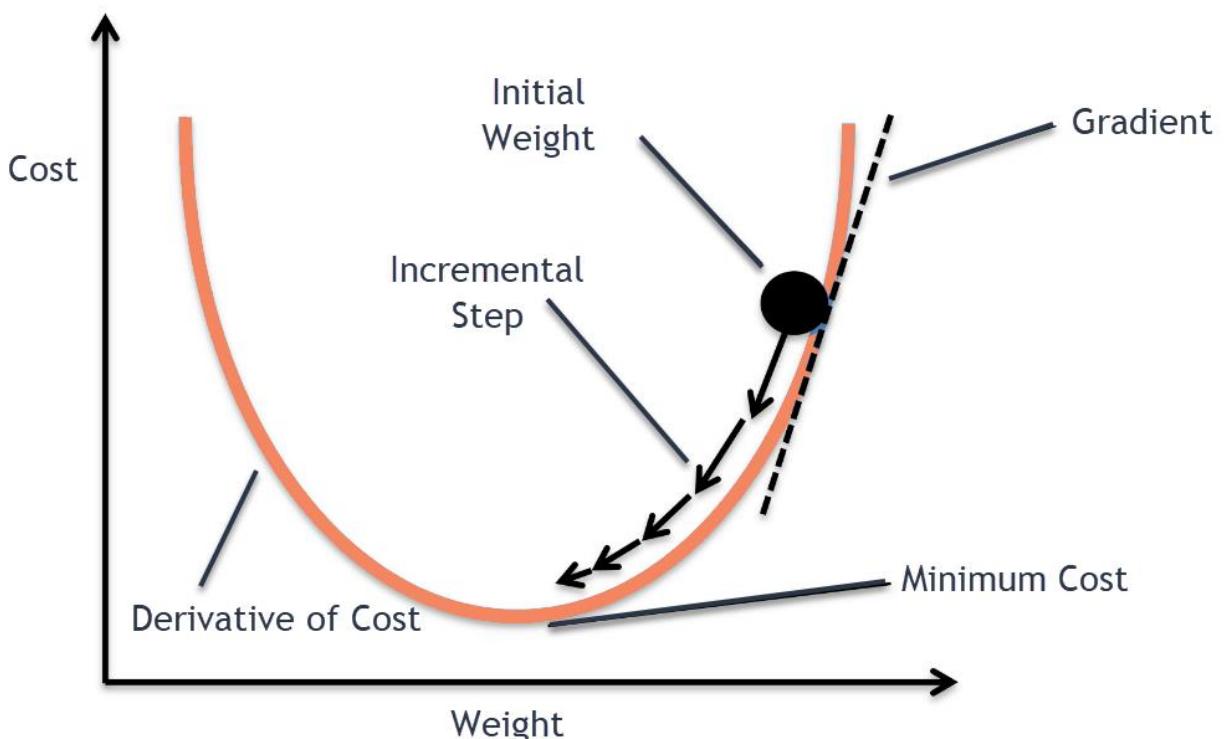
để tối ưu hóa trọng số của mạng lưới thông qua việc tính toán độ lỗi và cập nhật trọng số theo hướng giảm thiểu sai số.

DNN được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng hình ảnh, nhận dạng giọng nói, dịch máy, và nhiều ứng dụng khác trong trí tuệ nhân tạo.

Các thuật toán được sử dụng trong DNN:

Gradient Descent (GD): là thuật toán tìm tối ưu chung cho các hàm số. Ý tưởng chung của GD là điều chỉnh các tham số để lặp đi lặp lại thông qua mỗi dữ liệu huấn luyện để giảm thiểu hàm chi phí.

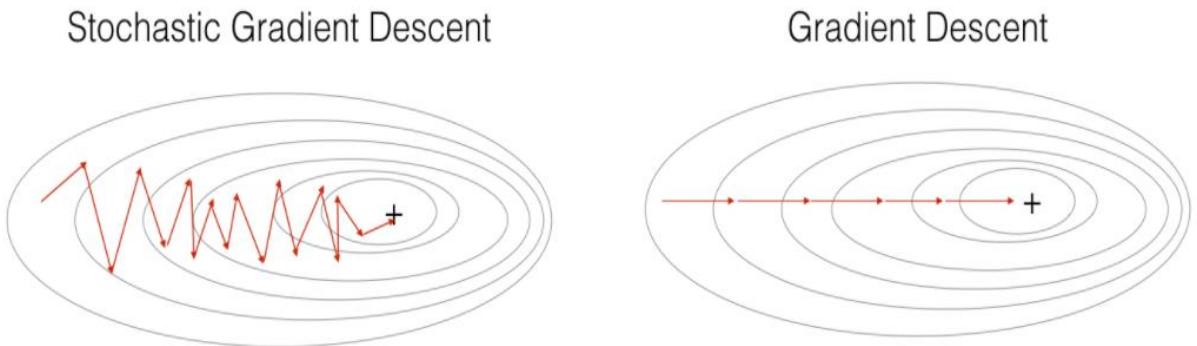
Giả sử một người bị lạc trên 1 ngọn núi và trong sương mù dày đặc, người đó chỉ có thể cảm thấy độ dốc của mặt đất dưới chân họ. Một cách tốt nhất để nhanh chóng xuống chân núi là xuống dốc theo hướng dốc nhất **Hình 2.5**. Đây chính là ý tưởng của Gradient Descent thực hiện, tại mỗi điểm của hàm số, nó sẽ xác định độ dốc sau đó đi ngược lại với hướng của độ dốc đến khi nào độ dốc tại chỗ đó bằng 0 (cực tiểu)



Hình 2.5. Mô phỏng thuật toán Gradient Descent

Backpropagation: là bản chất của việc huấn luyện mạng lưới thần kinh. Đó là thực hành tinh chỉnh các trọng số của mạng lưới thần kinh dựa trên tỉ lệ lỗi (tức là măt mát) thu được trong kỉ nguyên trước đó (tức là lặp lại.) Việc điều chỉnh các trọng số đúng cách sẽ đảm bảo tỉ lệ lỗi thấp hơn, làm cho mô hình trở nên đáng tin cậy bằng cách tăng độ tin cậy của nó.

Stochastic Gradient Descent (SGD): Stochastic là 1 biến thể của Gradient Descent . Thay vì sau mỗi epoch chúng ta sẽ cập nhật trọng số (Weight) 1 lần thì trong mỗi epoch có N điểm dữ liệu chúng ta sẽ cập nhật trọng số N lần. Nhìn vào 1 mặt , SGD sẽ làm giảm đi tốc độ của 1 epoch. Tuy nhiên nhìn theo 1 hướng khác, SGD sẽ hội tụ rất nhanh chỉ sau vài epoch. Công thức SGD cũng tương tự như GD nhưng thực hiện trên từng điểm dữ liệu. Để so sánh sự khác biệt giữa SGD và GD có thể xem **Hình 2. 6**



Hình 2. 6. Mô phỏng và so sánh thuật toán SGD và GD

Nhìn vào **Hình 2. 6**, ta thấy SGD có đường đi khá là zig zắc , không mượt nhu GD. Để hiểu điều đó vì 1 điểm dữ liệu không thể đại diện cho toàn bộ dữ liệu.

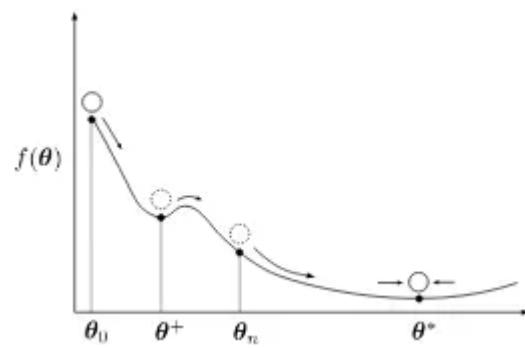
RMSProp: RMSprop giải quyết vấn đề tỉ lệ học giảm dần của Adagrad bằng cách chia tỉ lệ học cho trung bình của bình phương gradient như **Hình 2. 7**

$$E[g^2]_t = 0,9E[g^2]_{t-1} + 0,1g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Hình 2. 7. Minh họa thuật toán RMSProp

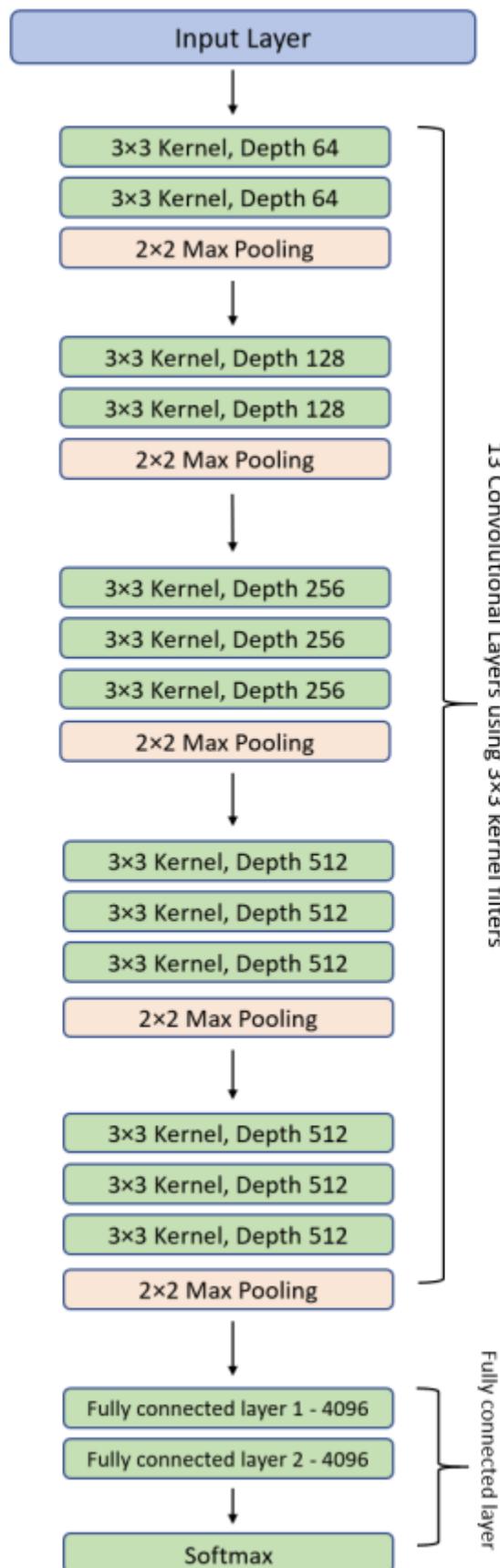
Adam: Adam là sự kết hợp của Momentum và RMSprop. Nếu giải thích theo hiện tượng vật lí thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như 1 quả cầu rất nặng có ma sát như **Hình 2. 8**, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn.



Hình 2. 8. Mô phỏng thuật toán Adam

2.3.4. Mô hình VGG-16

Mô hình VGG-16 – một trong những mô hình CNN nổi tiếng được đào tạo trên tập dữ liệu ImageNet để nhận diện đối tượng trong hình ảnh. Mô hình này đã được tích hợp sẵn trong thư viện Keras. Ở **Hình 2. 9** là thể hiện của kiến trúc VGG-16.



Hình 2.9. Kiến trúc VGG-16

Ở kiến trúc trên có thể thấy rằng có 13 lớp tích chập, 2 lớp được kết nối đầy đủ và 1 bộ phân loại SoftMax VGG-16 - Karen Simonyan và Andrew Zisserman đã giới thiệu kiến trúc VGG-16 vào năm 2014 trong bài báo của họ Mạng lưới chuyển đổi rất sâu để nhận dạng hình ảnh quy mô lớn. Karen và Andrew đã tạo ra một mạng 16 lớp bao gồm các lớp tích chập và được kết nối đầy đủ. Chỉ sử dụng các lớp chập 3×3 xếp chồng lên nhau để đơn giản. [27]

Cấu trúc chính xác của VGG-16 ở **Hình 2.9** được thể hiện chính xác như sau:

- Lớp tích chập thứ nhất và thứ hai bao gồm 64 bộ lọc kernel tính năng và kích thước của bộ lọc là 3×3 . Khi hình ảnh đầu vào (hình ảnh RGB có độ sâu 3) được chuyển vào lớp chập thứ nhất và thứ hai, kích thước sẽ thay đổi thành $224 \times 224 \times 64$. Sau đó, kết quả đầu ra được chuyển đến lớp gộp tối đa với bước nhảy là 2.
- Lớp tích chập thứ ba và thứ tư gồm 124 bộ lọc kernel tính năng và kích thước của bộ lọc là 3×3 . Tiếp theo hai lớp này là lớp gộp tối đa với bước 2 và kết quả đầu ra sẽ giảm xuống $56 \times 56 \times 128$.
- Lớp thứ năm, thứ sáu và thứ bảy là các lớp chập có kích thước hạt nhân 3×3 . Cả ba đều sử dụng 256 bản đồ đặc trưng. Tiếp theo các lớp này là lớp gộp tối đa với bước 2.
- Lớp thứ tám đến lớp mười ba là hai bộ lớp chập với kích thước hạt nhân 3×3 . Tất cả các bộ lớp chập này có 512 bộ lọc hạt nhân. Tiếp theo các lớp này là lớp gộp tối đa với bước tiến là 1.
- Mười bốn và mười lăm lớp là các lớp ẩn được kết nối đầy đủ gồm 4096 đơn vị, theo sau là lớp đầu ra softmax (Lớp thứ mười sáu) gồm 1000 đơn vị.

Ứng dụng độ đo cosine trong mô hình

Trong việc tìm kiếm ảnh tương đồng với mô hình VGG16, việc sử dụng độ đo cosine là một phương pháp quan trọng và hiệu quả. Độ đo cosine giúp đánh giá sự tương đồng giữa các vector biểu diễn hình ảnh dựa trên góc giữa chúng, thay vì dựa trên độ dài của vector.

Trong ngữ cảnh này, các hình ảnh được biểu diễn dưới dạng vector trong không gian nhiều chiều, và việc sử dụng cosine giúp đo lường sự tương đồng hình học giữa chúng. Nếu hai vector biểu diễn hình ảnh gần nhau trong không gian vector, tức là chúng có góc giữa chúng nhỏ, thì giá trị cosine sẽ gần 1, thể hiện mức độ tương đồng cao. Ngược lại, nếu chúng ở xa nhau, có góc giữa lớn, giá trị cosine sẽ giảm xuống, thể hiện sự khác biệt giữa chúng.

Do đó, việc sử dụng cosine trong mô hình tìm kiếm ảnh tương đồng giúp cải thiện khả năng đo lường sự tương đồng không gian giữa các hình ảnh, làm cho kết quả tìm kiếm trở nên chính xác và hiệu quả hơn.

2.3.5. Thư viện TFlearn

TFlearn là một thư viện deep learning mô-đun và minh bạch được xây dựng dựa trên Tensorflow. Nó được thiết kế để cung cấp API cấp cao hơn cho TensorFlow nhằm tạo điều kiện thuận lợi và tăng tốc độ thử nghiệm, trong khi vẫn hoàn toàn minh bạch và tương thích với nó. [9]

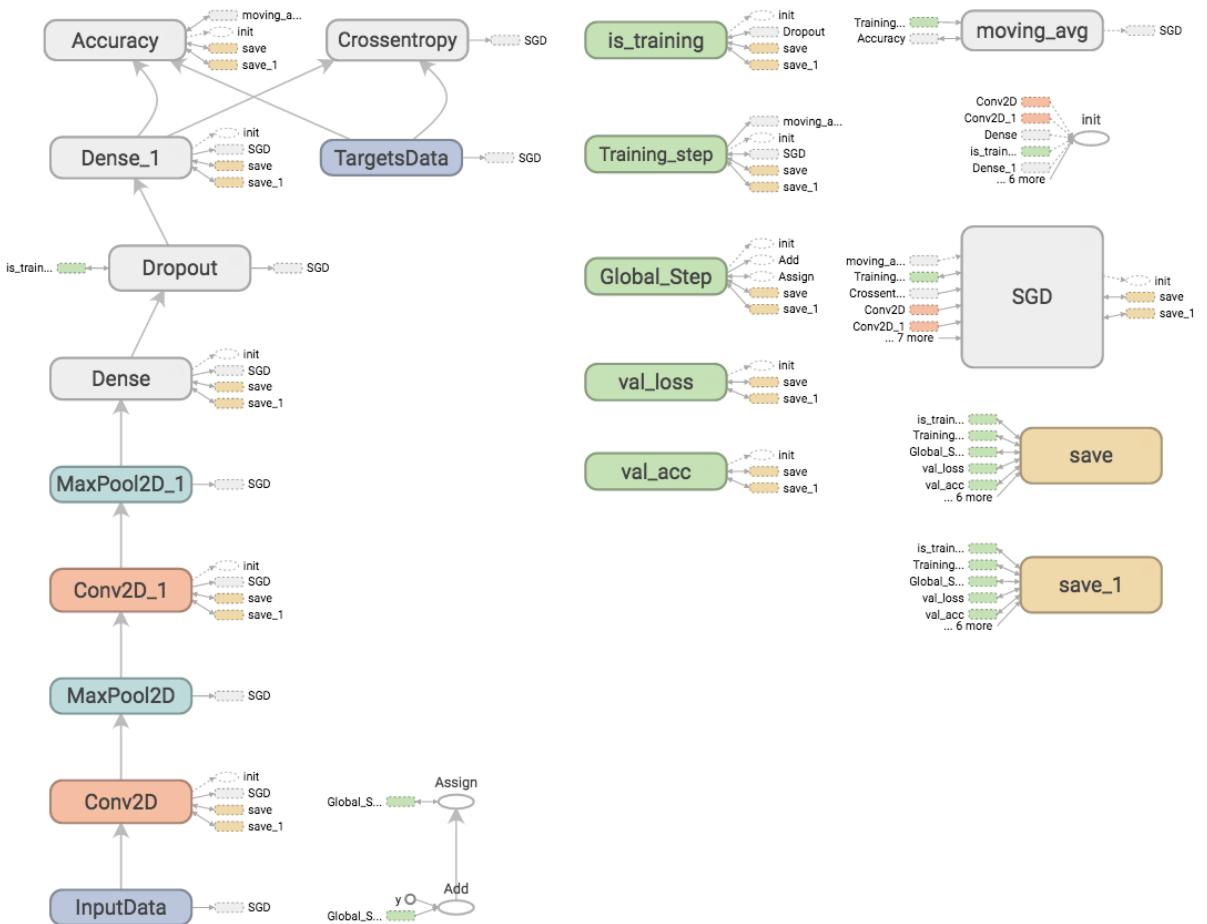
Các tính năng của TFLearn bao gồm:

- API cấp cao dễ sử dụng và dễ hiểu để triển khai mạng lưới thần kinh sâu, kèm theo hướng dẫn và ví dụ.
- Tạo mẫu nhanh thông qua các lớp mạng thần kinh tích hợp có tính mô-đun cao, bộ điều chỉnh, bộ tối ưu hóa, số liệu...
- Hoàn toàn minh bạch trên Tensorflow. Tất cả các chức năng đều được xây dựng trên tensor và có thể được sử dụng độc lập với TFLearn.
- Chức năng trợ giúp mạnh mẽ để huấn luyện bất kỳ biểu đồ TensorFlow nào, với sự hỗ trợ của nhiều đầu vào, đầu ra và trình tối ưu hóa.
- Trực quan hóa biểu đồ dễ dàng và đẹp mắt, với các chi tiết về trọng lượng, độ dốc, kích hoạt và hơn thế nữa...

Vị trí thiết bị dễ dàng để sử dụng nhiều CPU/GPU.

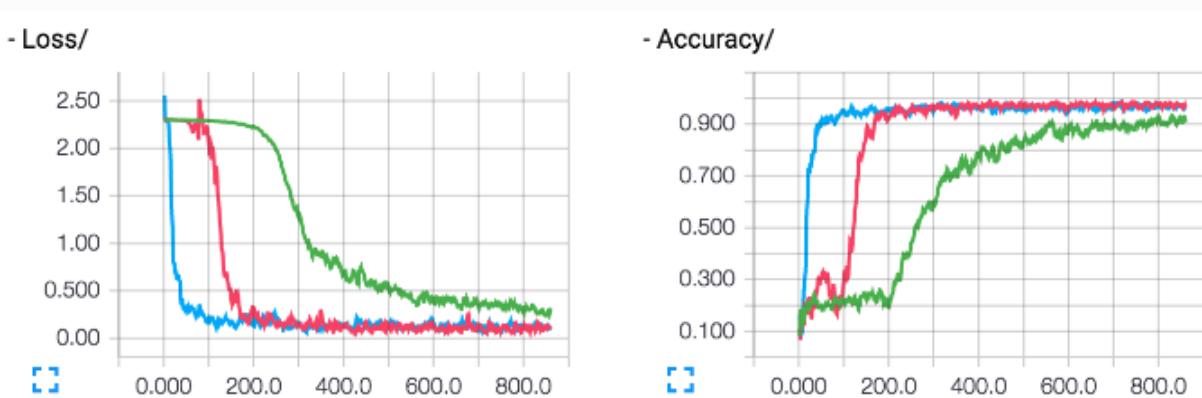
API cấp cao hiện hỗ trợ hầu hết các mô hình học sâu gần đây, chẳng hạn như Convolutions, LSTM, BiRNN, BatchNorm, PRELU, Residual Networks, Generative Network... Trong tương lai, TFLearn cũng nhằm mục đích luôn cập nhật với kỹ thuật học sâu mới nhất.

Để trực quan hơn về mô hình này, sau đây là các mô hình trực quan hóa của mô đun thư viện học sâu TFlearn như kiến trúc ở **Hình 2. 10**



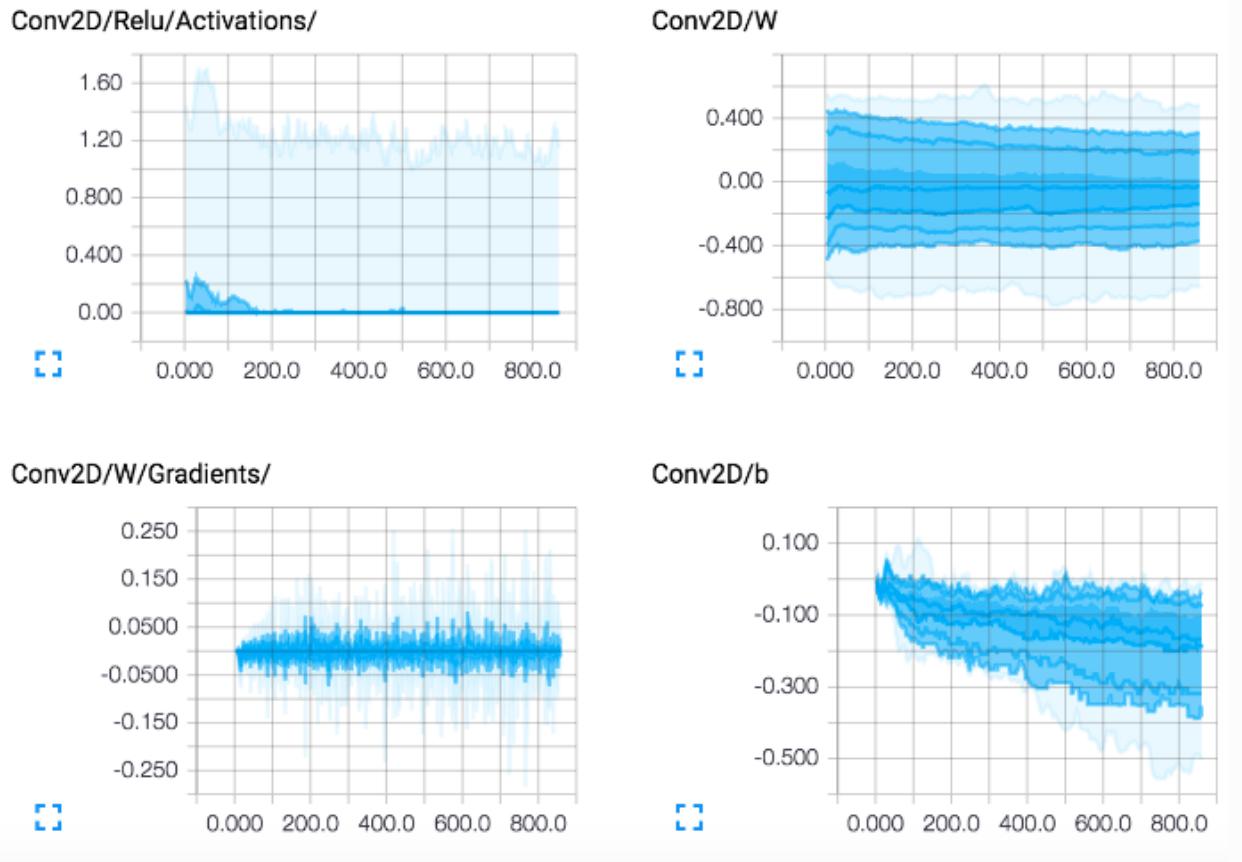
Hình 2. 10. Kiến trúc TFlearn

Hàm mất mát và độ chính xác như **Hình 2. 11**



Hình 2. 11. Hàm mất mát và độ chính xác của TFlearn

Các lớp (layers) gồm Conv2D/Relu/Activations, Conv2D/W, Conv2D/W/Gradients, Conv2D/b được thể hiện qua **Hình 2. 12**



Hình 2. 12. Các layers của TFLearn

2.3.6. Độ tương tự cosin

Trong cuộc sống hiện nay, việc áp dụng độ đo cosin đang đóng một vai trò quan trọng trong nhiều lĩnh vực khác nhau. Một ứng dụng quan trọng của độ đo cosin là trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), nơi nó được sử dụng để đánh giá sự tương tự giữa các văn bản. Điều này có thể thấy trong các ứng dụng như tìm kiếm web, phân loại văn bản và hệ thống gợi ý.

Ngoài ra, độ đo cosin có một vai trò quan trọng trong lĩnh vực lọc cộng tác, giúp đánh giá sự tương tự giữa người dùng hoặc sản phẩm. Điều này đặc biệt hữu ích trong việc đề xuất sản phẩm hoặc nội dung dựa trên sự tương tự với người dùng khác, tạo ra trải nghiệm cá nhân hóa và tăng cường tương tác.

Các lĩnh vực khác như xử lý ảnh, thị trường tài chính và y học cũng sử dụng độ đo cosin để đo lường sự tương tự trong các ngữ cảnh cụ thể của chúng. Trong xử lý ảnh, nó có thể so sánh đặc trưng của các hình ảnh, trong thị trường tài chính giúp đánh giá sự tương tự giữa cổ phiếu hoặc quỹ đầu tư, và trong lĩnh vực y học, có thể được áp dụng để so sánh gen của các bệnh nhân.

Độ đo cosin, với khả năng đo lường sự tương tự trong không gian nhiều chiều, đóng vai trò quan trọng trong việc cung cấp thông tin giá trị và hỗ trợ quyết định trong nhiều lĩnh vực ứng dụng khác nhau.

Độ tương tự cosin [10] là một cách đo độ tương tự (measure of similarity) giữa hai vectơ khác không của một không gian tích vô hướng. Độ tương tự này được định nghĩa bằng giá trị cosine của góc giữa hai vectơ, và cũng là tích vô hướng của cùng các vectơ đơn vị để cả hai đều có chiều dài 1. Giá trị cosine của 0° là 1, và bé hơn 1 với bất kỳ góc nào trong khoảng các radian $(0, \pi]$.

Độ tương tự cosin là một thắc định có tính định hướng chứ không phải về độ lớn (to nhỏ): hai vectơ cùng hướng có độ tương tự cosin là 1, hai vectơ vuông góc nhau (hay có hướng 90°) có độ tương tự là 0, và hai vectơ đối nhau theo đường kính có độ tương tự (hay 180°) là -1. Độ tương tự cosin đặc biệt được sử dụng trong không gian dương với kết quả được giới hạn chặt chẽ trong biên độ $[0,1]$. Cái tên "độ tương tự cosin" bắt nguồn từ thuật ngữ "cosin có hướng": trong trường hợp này, các vectơ đơn vị có độ "tương tự" tối đa nếu chúng song song và "khác nhau" cực đại nếu chúng là trực giao (vuông góc). Điều này tương tự với cosin, có giá trị lớn nhất khi các phân đoạn tạo thành một góc bằng 0, và giá trị bằng 0 (không liên quan) khi các đoạn thẳng vuông góc.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

Hình 2. 13. Công thức độ tương tự cosin

Ở công thức **Hình 2. 13**, A_i và B_i là thành phần của vector A và B tương ứng.

Độ tương tự có giá trị -1 có nghĩa là trái nghĩa hoàn toàn, với giá trị 1 nghĩa là giống nhau hoàn toàn, với 0 có nghĩa là trực giao hay tương quan (decorrelation), trong khi các giá trị ở giữa biểu thị sự giống nhau hoặc không giống nhau ở mức trung gian.

Đối với so khớp chuỗi xấp xỉ (approximate string matching), các vectơ thuộc tính A và B thường là các vectơ tf-idf của các tài liệu. Độ tương tự cosine có thể xem là một phương thức bình thường hóa độ dài tài liệu trong quá trình so sánh.

Trong trường hợp truy hồi thông tin, độ tương tự cosine của hai tài liệu sẽ từ 0 đến 1, do các tần số thuật ngữ (dùng các trọng lượng tf-idf) không thể là giá trị âm. Do đó, góc giữa các vectơ tần số thuật ngữ không thể lớn hơn 90° .

Kỹ thuật TF-IDF

TF-IDF là viết tắt của từ Term frequency invert document frequency. Nó là một kỹ thuật feature extraction dùng trong text mining và information retrieval. Trước khi có tf-idf người ta dùng one-hot-encoding để embedding words sang vector. Nhưng kỹ thuật này gặp một số hạn chế là :

- Những từ thường xuyên xuất hiện sẽ không có nhiều thông tin nhưng vẫn có tỉ trọng(weight) ngang với các từ khác. Ví dụ stop word chẳng hạn, khi phân tích về cửa hàng nào đó thì từ "cửa hàng" xuất hiện ở tất cả document. Điều cần làm là giảm tỉ trọng về mặt thông tin nó xuống vì thông tin không mang nhiều giá trị.
- Những từ hiếm(rare word) hoặc từ khoá (key word) không có sự khác biệt về tỉ trọng thông tin

Để khắc phục hạn chế này tf-idf đã ra đời. Tf-idf bao gồm 2 thành phần là tf (term frequency) và idf (inverse document frequency).

$$tf(w, d) = \frac{\text{Tổng số từ } w \text{ trong tài liệu } d}{\text{Tổng số từ trong tài liệu}}$$

tf đo lường tỉ trọng tần suất từ w có trong tài liệu d. Vì document thường có chiều dài khác nhau nên để bình thường hóa (normalization) phải chia nó cho tổng số từ trong tài liệu d.

$$idf = tf * \frac{N}{M}$$

Trong đó:

- N là tổng số tài liệu trong dataset.
- M là tổng tài liệu có từ w xuất hiện.

Tỉ số $\frac{N}{M}$ được xem là inverse document frequency. Nếu một từ xuất hiện nhiều ở các tài liệu thì tỉ số này sẽ gần 1. Và ngược lại một từ ít xuất hiện hơn tỉ số này sẽ cao hơn 1. Điều này giúp giảm tỉ trọng của những từ thường xuyên xuất hiện và tăng tỉ trọng những từ ít xuất hiện trong document hơn (lưu ý N luôn lớn hơn hoặc bằng M).

Một vấn đề là khi N rất lớn mà M rất nhỏ thì tỉ số này rất lớn cho nên cần log transform để giảm giá trị tỉ số $\frac{N}{M}$ tránh gây khó khăn trong việc tính toán. Khi đó công thức idf cuối cùng sẽ là:

$$idf = tf * \log\left(\frac{N}{M}\right)$$

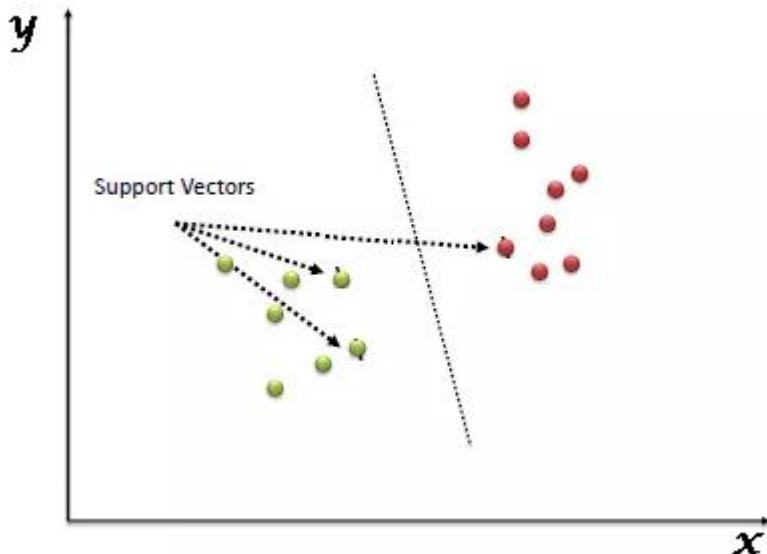
Ví dụ: một tài liệu có 100 từ, chứa từ “gấu” 3 lần và có tổng 10.000 tài liệu trong đó có 1000 tài liệu có chứa từ “gấu” thì lúc này tf và idf sẽ được tính như sau:

$$tf = \frac{3}{100} = 0.03$$

$$idf(gấu) = 0.03 * \log\left(\frac{10000}{1000}\right) = 0.06$$

2.3.7. SVM - Support Vector Machine

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đề quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, đồ thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó thực hiện tìm "siêu phẳng" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt như **Hình 2. 14**

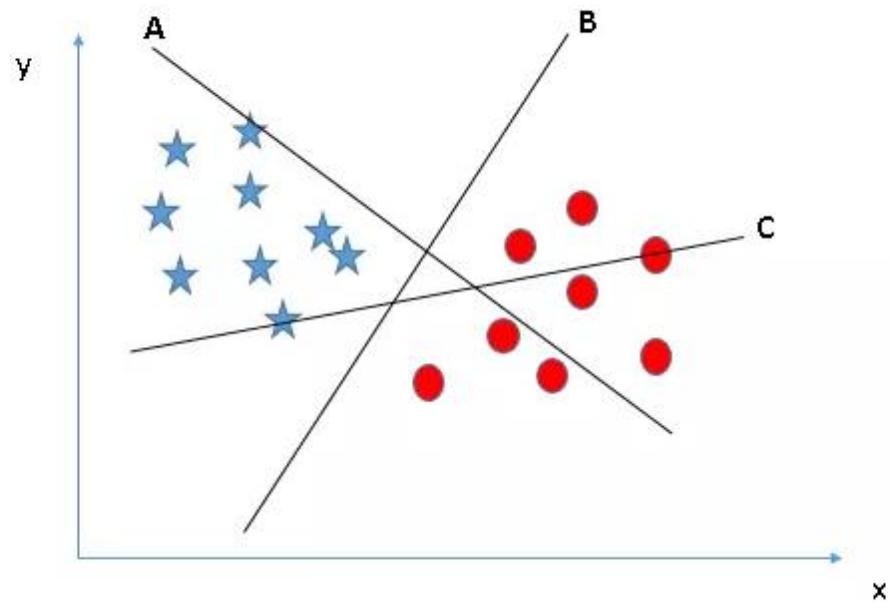


Hình 2. 14. Siêu phẳng (hyper-plane)

Support Vectors hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới để chia hai lớp tốt nhất.

Xác định siêu phẳng phù hợp (ngữ cảnh 1)

Ở **Hình 2. 15**, có 3 đường hyper-plane (A,B và C). Bây giờ đường nào là hyper-plane đúng cho nhóm ngôi sao và hình tròn.

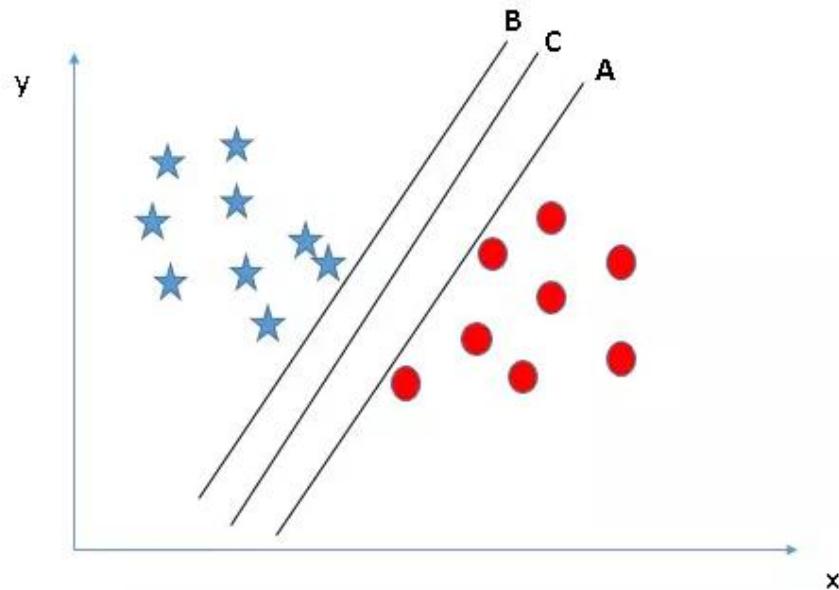


Hình 2. 15. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 1)

Quy tắc số một để chọn 1 hyper-lane, chọn một hyper-plane để phân chia hai lớp tốt nhất. Trong ví dụ này chính là đường B.

Xác định siêu phẳng phù hợp (ngữ cảnh 2)

Ở **Hình 2. 16** cũng có 3 đường hyper-plane (A,B và C), theo quy tắc số 1, chúng đều thỏa mãn.

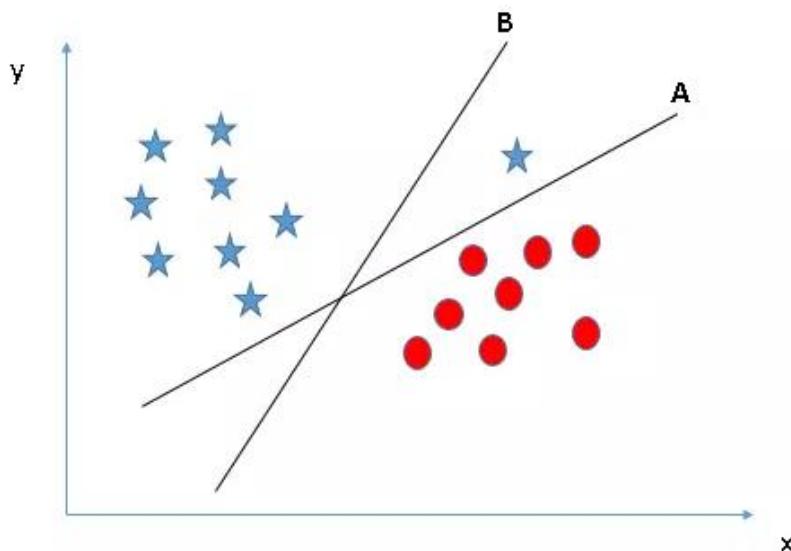


Hình 2. 16. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 2)

Quy tắc thứ hai chính là xác định khoảng cách lớn nhất từ điểm gần nhất của một lớp nào đó đến đường hyper-plane. Khoảng cách này được gọi là "Margin", ở **Hình 2. 16** có thể nhìn thấy khoảng cách margin lớn nhất đây là đường C. Cần nhớ nếu chọn làm hyper-plane có margin thấp hơn thì sau này khi dữ liệu tăng lên thì sẽ sinh ra nguy cơ cao về việc xác định nhầm lớp cho dữ liệu.

Xác định siêu phẳng phù hợp (ngữ cảnh 3)

Sử dụng các nguyên tắc đã nêu trên để chọn ra hyper-plane cho trường hợp ở **Hình 2. 17** sau:

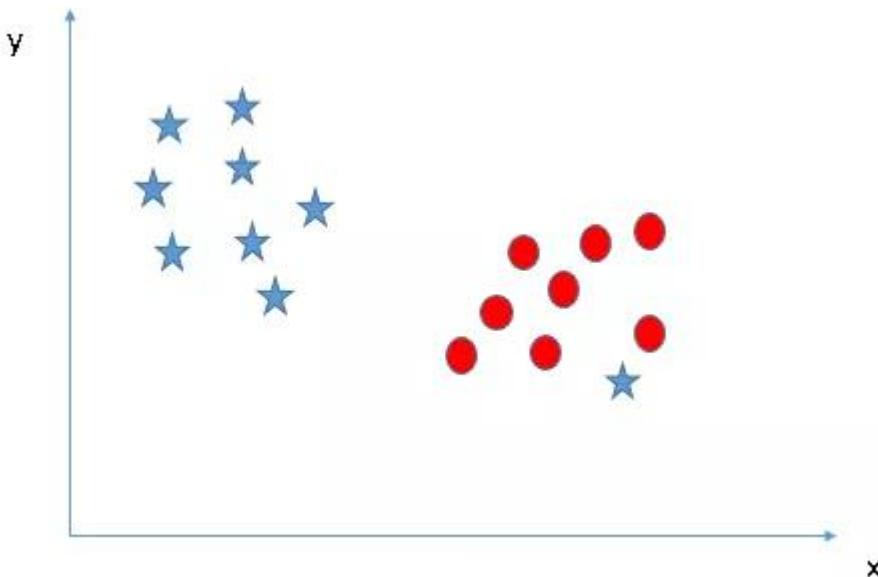


Hình 2. 17. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 3)

Có thể lần đầu nhìn sẽ chọn làm đường B bởi vì nó có margin cao hơn đường A, nhưng đây sẽ không đúng bởi vì nguyên tắc đầu tiên sẽ là nguyên tắc số 1 cần chọn hyper-plane để phân chia các lớp thành riêng biệt. Vì vậy đường A mới là lựa chọn chính xác.

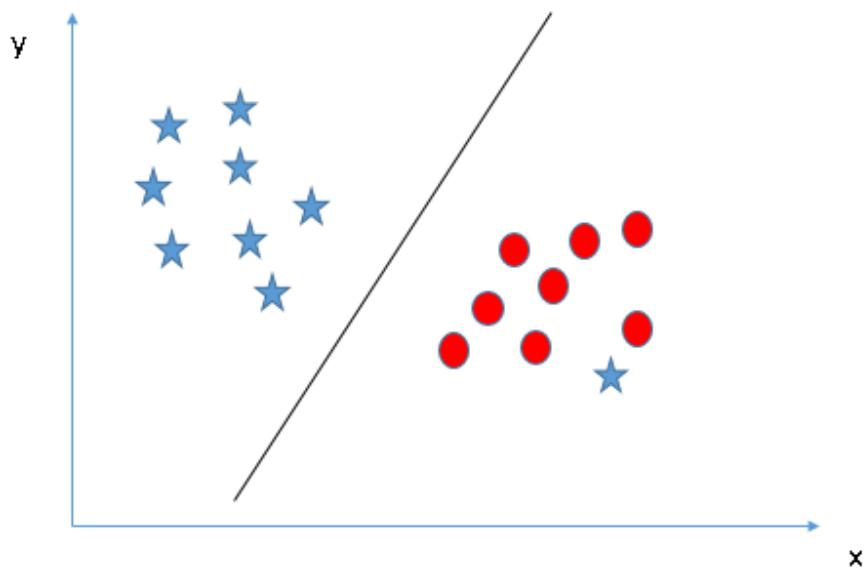
Xác định siêu phẳng phù hợp (ngữ cảnh 4)

Tiếp theo ở ngữ cảnh 4 này **Hình 2. 18**, làm sao để phân loại 2 lớp riêng biệt dựa vào các quy tắc trên.



Hình 2. 18. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 4)

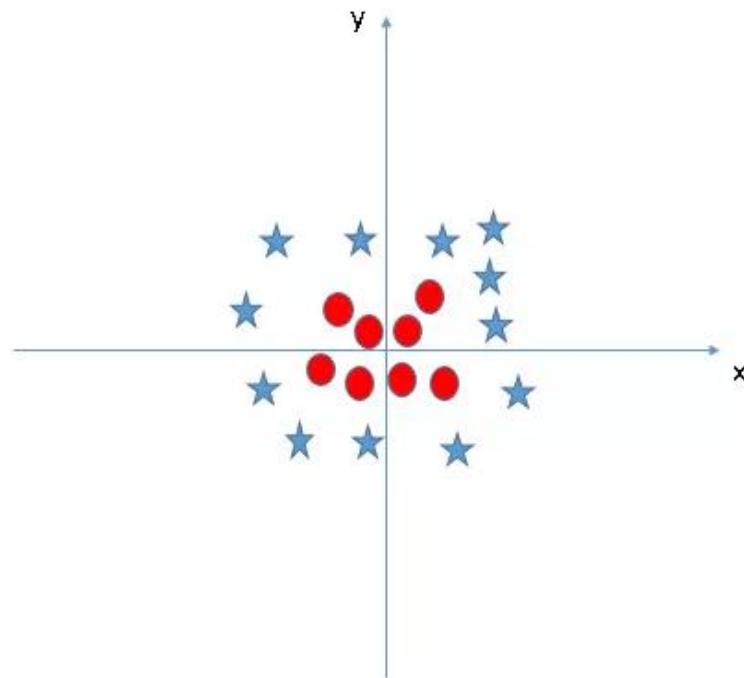
Ở trong ngữ cảnh **Hình 2. 18** bắt buộc phải chấp nhận một ngôi sao ở bên ngoài cuối được xem như một ngôi sao phía ngoài hơn, SVM có tính năng cho phép bỏ qua các ngoại lệ và tìm ra hyper-plane có biên giới tối đa. Do đó có thể nói, SVM có khả năng mạnh trong việc chấp nhận ngoại lệ. Như vậy **Hình 2. 19** là cách phân lớp cho ngữ cảnh này.



Hình 2. 19. Cách phân lớp cho siêu phẳng ngữ cảnh 4

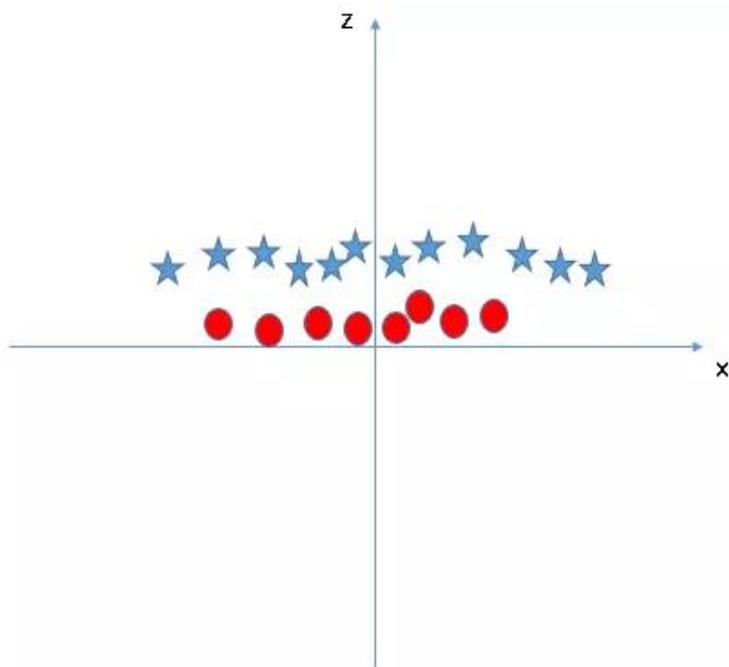
Xác định siêu phẳng phù hợp (ngữ cảnh 5)

Trong trường hợp **Hình 2. 20**, không thể tìm ra 1 đường hyper-plane tương đối để chia các lớp, vậy làm thế nào để SVM phân tách dữ liệu thành hai lớp riêng biệt? Cho đến bây giờ chỉ nhìn vào các đường tuyếntính hyper-plane.



Hình 2. 20. Ví dụ về xác định siêu phẳng phù hợp (ngữ cảnh 5)

SVM có thể giải quyết vấn đề này, Khá đơn giản, nó sẽ được giải quyết bằng việc thêm một tính năng, Ở đây sẽ thêm tính năng $z = x^2 + y^2$. Bây giờ dữ liệu sẽ được biến đổi theo trục x và z như **Hình 2. 21**

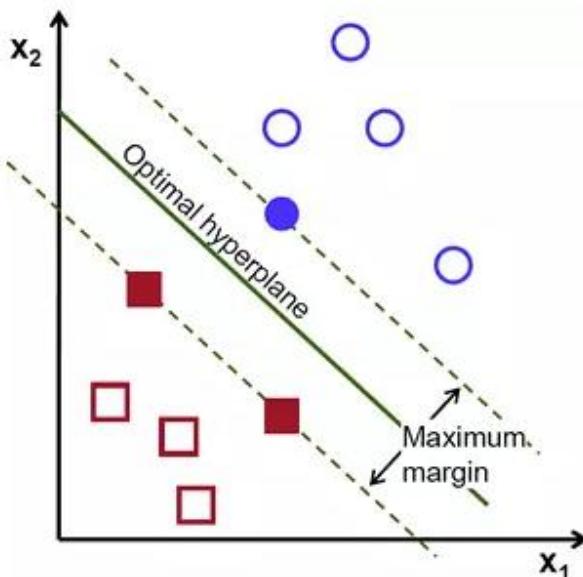


Hình 2. 21. Siêu phẳng bị biến đổi (ngữ cảnh 5)

Trong sơ đồ trên, các điểm cần xem xét là:

- Tất cả dữ liệu trên trục z sẽ là số dương vì nó là tổng bình phương x và y
- Trên biểu đồ các điểm tròn đỏ xuất hiện gần trục x và y hơn vì thế z sẽ nhỏ hơn => nằm gần trục x hơn trong đồ thị (z, x) Trong SVM, rất dễ dàng để có một siêu phẳng tuyến tính (linear hyper-plane) để chia thành hai lớp, Nhưng một câu hỏi sẽ nảy sinh, đây là có cần phải thêm một tính năng phân chia này bằng tay hay không. Câu trả lời là không, bởi vì SVM có một kỹ thuật được gọi là kernel trick (kỹ thuật hạt nhân), đây là tính năng có không gian đầu vào có chiều sâu thẩm và biến đổi nó thành không gian có chiều cao hơn, tức là nó không phân chia các vấn đề thành các vấn đề riêng biệt, các tính năng này được gọi là kernel. Nói một cách đơn giản nó thực hiện một số biến đổi dữ liệu phức tạp, sau đó tìm ra quá trình tách dữ liệu dựa trên các nhãn hoặc đầu ra mà chúng ra đã xác định trước.

Margin



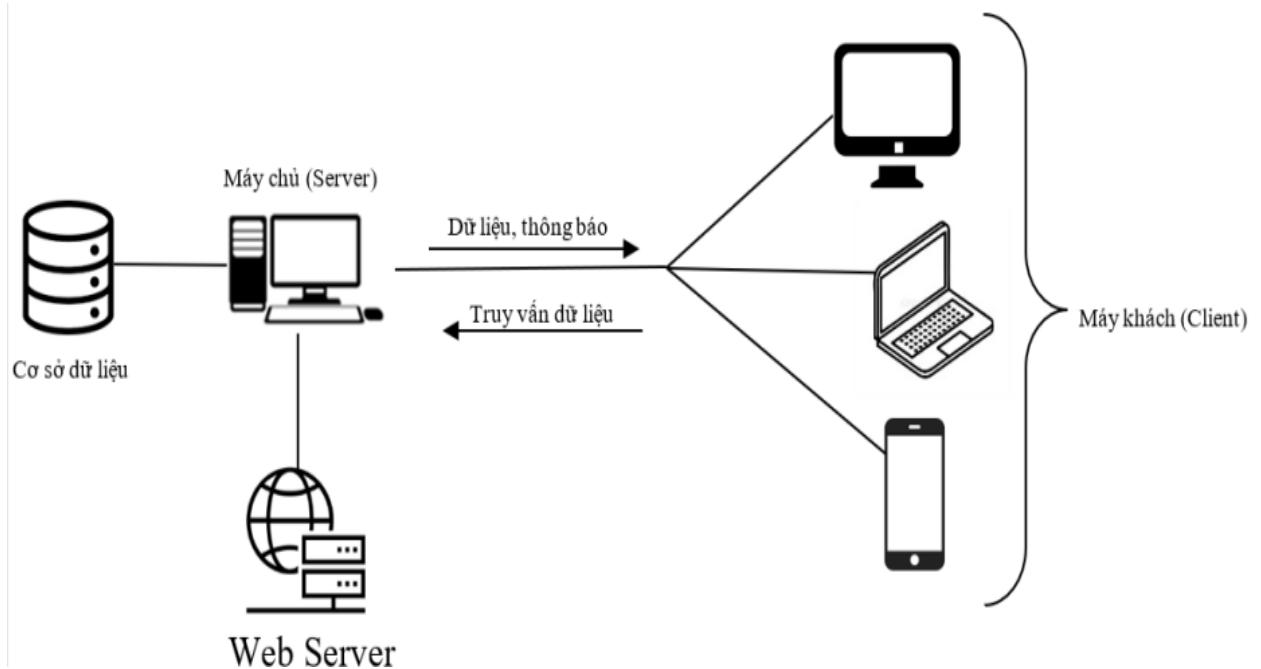
Hình 2. 22. Ví dụ về margin trong SVM

Margin là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp. Trong ví dụ quả táo quả lê đặt trên mặt bán ở **Hình 2. 22**, margin chính là khoảng cách giữa cây que và hai quả táo và lê gần nó nhất. Điều quan trọng ở đây đó là phương pháp SVM luôn cố gắng cực đại hóa margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với 2 quả táo và lê. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào

CHƯƠNG 3. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

3.1. Kiến trúc tổng quát hệ thống

Hệ thống website được thiết kế dựa trên mô hình client-server được biểu diễn ở **Hình 3. 1**, kết hợp với một số công nghệ và tiêu chuẩn hiện đại.



Hình 3. 1. Mô hình client-server

3.1.1. Mô hình Client-Server

Client (Khách hàng): Gồm các thiết bị người dùng cuối như máy tính cá nhân, điện thoại di động, máy tính bảng. Ở phía client, người dùng sẽ truy cập website thông qua trình duyệt web và tương tác với giao diện do Vue.js cung cấp. Tất cả các yêu cầu từ phía người dùng đều được gửi đến server để xử lý.

Server (Máy chủ): Máy chủ sẽ xử lý yêu cầu từ phía client, truy vấn CSDL nếu cần và gửi phản hồi trở lại client. Ở phía server, Laravel đóng vai trò xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu.

3.1.2. Mô hình giao tiếp

API (Application Programming Interface): Được xây dựng dựa trên Laravel, API cung cấp các điểm cuối (endpoints) để client gọi và nhận dữ liệu. API sẽ truy vấn CSDL và trả dữ liệu về dưới dạng JSON cho client.

SPA (Single Page Application): Sử dụng Vue.js, SPA giúp tạo ra một trải nghiệm người dùng mượt mà, với việc tải dữ liệu mà không cần làm mới trang. SPA tương tác với API để lấy và gửi dữ liệu.

3.1.3. Cơ sở dữ liệu (CSDL)

MySQL: Là hệ thống quản lý cơ sở dữ liệu quan hệ, chịu trách nhiệm lưu trữ tất cả dữ liệu của hệ thống, bao gồm thông tin sản phẩm, thông tin khách hàng, đơn hàng và nhiều loại dữ liệu khác.

3.1.4. Kiến trúc tổng quát

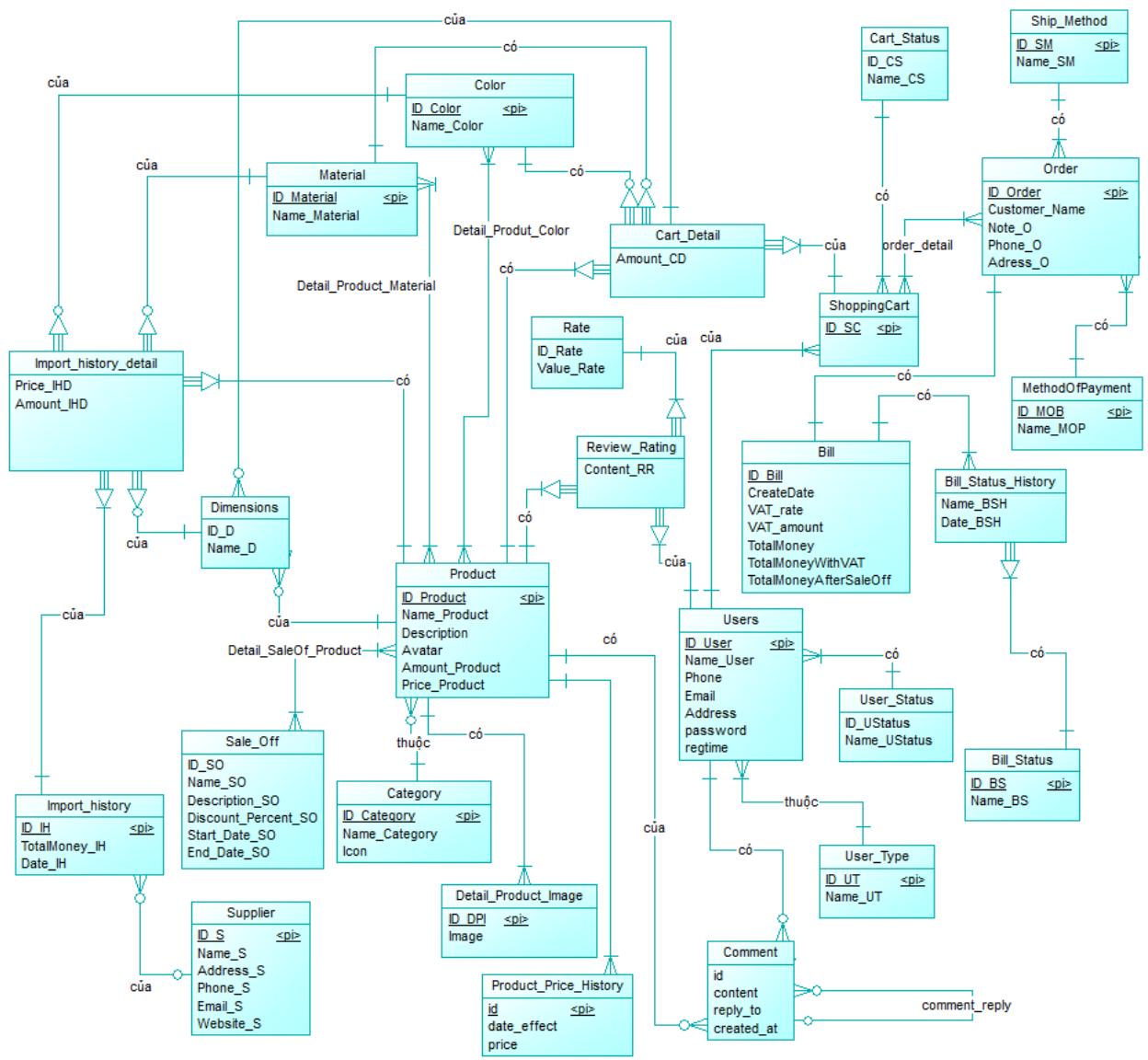
- Người dùng tương tác với website thông qua trình duyệt web.
- Trình duyệt gửi yêu cầu đến server thông qua API.
- Server xử lý yêu cầu, truy vấn dữ liệu từ MySQL nếu cần.
- Server gửi phản hồi trả lại client dưới dạng dữ liệu (thường là JSON).
- Vue.js ở phía client sẽ cập nhật giao diện dựa trên dữ liệu nhận được từ server.

Với kiến trúc tổng quát trên, hệ thống đảm bảo khả năng mở rộng, bảo mật và tốc độ xử lý nhanh chóng, đáp ứng nhu cầu của người dùng.

3.1.5. Xây dựng các mô hình dữ liệu

3.1.5.1. Mô hình dữ liệu mức quan niệm (CDM)

Để thấy được khía cạnh trừu tượng và trực quan của dữ liệu trong đề tài này thì mô hình CDM ở **Hình 3.2** sau đây sẽ là minh họa.



Hình 3.2. Mô hình CDM

3.1.5.2. Mô hình dữ liệu mức luận lý LDM

Từ mô hình dữ liệu mức quan niệm được minh họa ở **Hình 3. 2**, ta có được mô hình dữ liệu mức luận lý như sau:

Điễn giải:

- Khóa chính: in đậm, gạch chân.
- Khóa ngoại: in nghiêng, dấu thăng.

User_Type(**ID_UT**, Name_UT)

Category(**ID_Category**, Name_Categoty, Icon)

Product(**ID_Product**, Name_Product, Description, Avatar, Amount_Product, Price_Product, #*ID_Category*)

User(**ID_User**, Name_User, Phone, Email, Address, password, regtime, #*ID_UT*)

Material(**ID_Material**, Name_Material)

Color(**ID_Color**, Name_Color)

Dimensions(**ID_D**, Name_D, #*ID_Product*)

Supplier(**ID_S**, Name_S, Address_S, Phone_S, Email_S, Website_S)

Sale_Off(**ID_SO**, Name_SO, Discount_Percent_SO, Start_Date_SO, End_Date_SO)

Detail_Product_Material(#*ID_Product*, #*ID_Material*)

Detail_Product_Color(#*ID_Product*, #*ID_Color*)

Detail_SaleOf_Product(#*ID_Product*, #*ID_SO*)

Import_history_detail(Price_IHD, Amount_IHD, #*ID_D*, #*ID_IH*, #*ID_Material*, #*ID_Color*)

Import_history(**ID_IH**, TotalMoney_IH, Date_IH, #*ID_S*)

Detail_Product_Image(**ID_DPI**, Image, #*ID_Product*)

Product_Price_History(**id**, date_effect, price, #*ID_Product*, #*ID_User*)

Cart_Status(**ID_CS**, Name_CS)

ShoppingCart(**ID_SC**, #*ID_CS*, #*ID_User*)

Cart_Detail(Amount_CD, #*ID_Product*, #*ID_Color*, #*ID_Material*, #*ID_SC*)

Ship_Method(**ID_SM**, Name_SM)

MethodOfPayment(**ID_MOB**, Name_MOP)

Order(**ID_Order**, Customer_Name, Note_O, Phone_O, Adress_O, #ID_SM,
 #ID_MOB)
 order_detail(#ID_SC, #ID_Order)
 Bill_Status(**ID_BS**, Name_BS)
 Bill(**ID_Bill**, CreateDate, VAT_rate, VAT_amount, TotalMoney,
 TotalMoneyWithVAT, TotalMoneyAfterSaleOff, #ID_Order)
 Bill_Status_History(Date_BSH, #ID_Bill, #ID_BS, #ID_User)
 Comment(**id**, content, reply_to, created_at, #id)

Từ mô hình mức luận lý trên, để thấy được sự tham chiếu giữa các bảng với nhau thì sẽ được thể hiện ở **Bảng 3. 1** sau đây.

Bảng 3. 1: Ràng buộc tham chiếu

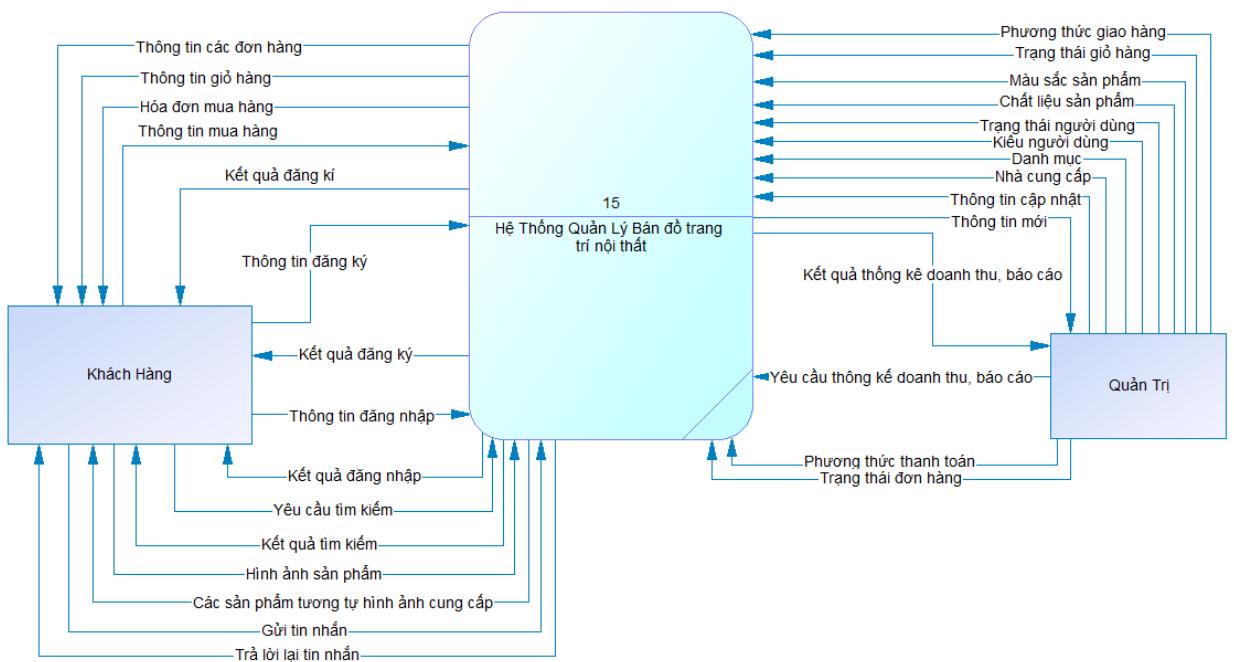
STT	Bảng con (Reference table)	Bảng cha (Primary table)
1	Product(ID_Category)	→ Category(ID_Category)
2	User(ID_UT)	→ User_Type(ID_UT)
3	Dimensions(ID_Product)	→ Product(ID_Product)
4	Detail_Product_Material(ID_Product)	→ Product(ID_Product)
5	Detail_Product_Material(ID_Material)	→ Material(ID_Material)
6	Detail_Product_Color(ID_Product)	→ Product(ID_Product)
7	Detail_Product_Color(ID_Color)	→ Color(ID_Color)
8	Detail_SaleOf_Product(ID_Product)	→ Product(ID_Product)
9	Detail_SaleOf_Product(ID_SO)	→ Sale_Off(ID_SO)
10	Import_history_detail(ID_D)	→ Dimensions(ID_D)
11	Import_history_detail(ID_IH)	→ Import_history(ID_IH)
12	Import_history_detail(ID_Material)	→ Material(ID_Material)
13	Import_history_detail(ID_Color)	→ Color(ID_Color)
14	Import_history(ID_S)	→ Supplier(ID_S)
15	Detail_Product_Image(ID_Product)	→ Product(ID_Product)
16	Product_Price_History(ID_Product)	→ Product(ID_Product)
17	ShoppingCart(ID_CS)	→ Cart_Status(ID_CS)

18	ShoppingCart(ID_User)	→ User(ID_User)
19	Cart_Detail(ID_Product)	→ Product(ID_Product)
20	Cart_Detail(ID_Color)	→ Color(ID_Color)
21	Cart_Detail(ID_Material)	→ Material(ID_Material)
22	Cart_Detail(ID_SC)	→ ShoppingCart(ID_SC)
23	Order(ID_SM)	→ Ship_Method(ID_SM)
24	Order(ID_MOB)	→ MethodOfPayment(ID_MOB)
25	order_detail(ID_SC)	→ ShoppingCart(ID_SC)
26	order_detail(ID_Order)	→ Order(ID_Order)
27	Bill(ID_Order)	→ Order(ID_Order)
28	Bill_Status_History(ID_Bill)	→ Bill(ID_Bill)
29	Bill_Status_History(ID_BS)	→ Bill_Status(ID_BS)
30	Comment(id)	→ Comment(id)

Chi tiết hơn về kiểu dữ liệu ở các bảng có thể xem phần phụ lục Mô tả dữ liệu mức luận lý

3.1.5.3. Mô hình DFD

Sơ đồ luồng dữ liệu để thể hiện ở **Hình 3. 3**

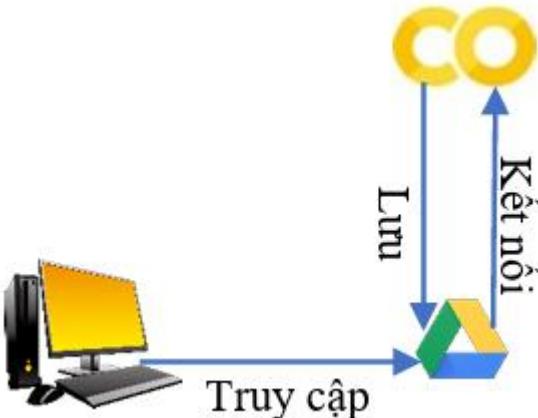


Hình 3. 3. Mô hình DFD

3.1.6. Xây dựng các mô hình AI

3.1.6.1. Môi trường đào tạo các mô hình

Sử dụng Google Colaboratory sản phẩm của Google hỗ trợ viết ngôn ngữ python, hỗ trợ việc đánh giá thực nghiệm các mô hình máy học. Cách sử dụng môi trường Google Colaboratory như **Hình 3. 4**



Hình 3. 4. Mô phỏng sử dụng Google Colaboratory

Sử dụng thư viện hỗ trợ máy học Numpy, keras, sklearn, nltk, tflearn, pickle

Sử dụng thư viện Tensorflow hỗ trợ việc tính toán và lấy kết quả đánh giá trong quá trình đào tạo mô hình.

3.1.6.2. Mô hình tìm kiếm hình ảnh tương tự

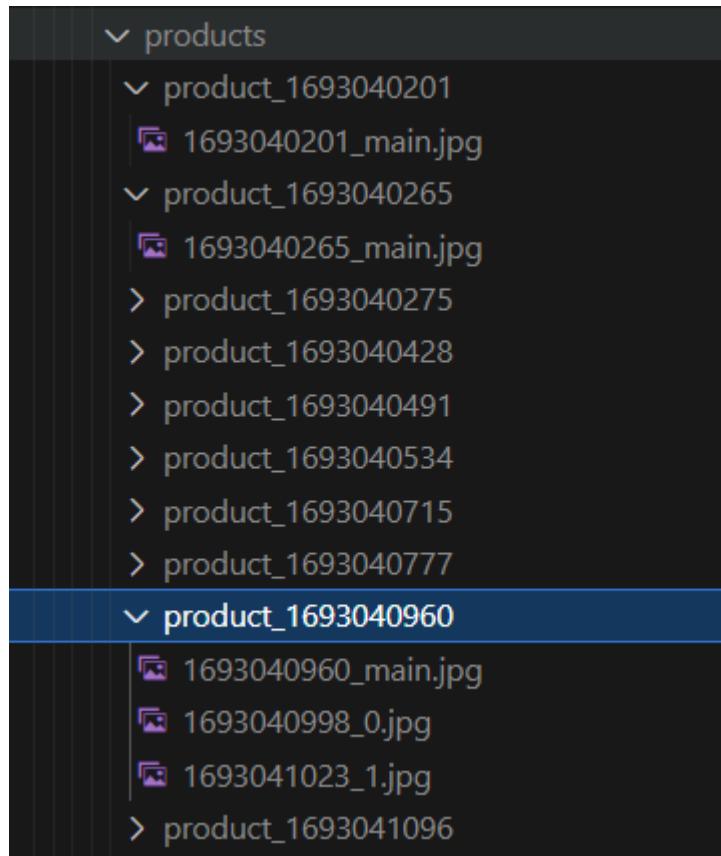
Để nâng cao trải nghiệm mua sắm trực tuyến và cung cấp cho người dùng khả năng tìm kiếm sản phẩm một cách linh hoạt, hệ thống đã tích hợp một mô hình tìm kiếm hình ảnh tương tự dựa trên trí tuệ nhân tạo.

3.1.6.2.1. Mô hình sử dụng

Mô hình tìm kiếm ảnh tương tự sử dụng mô hình VGG-16. Ở mô hình này đã sử dụng tầng fc2 làm tầng cuối cùng để trích xuất đặc trưng, đây là một tầng dày đặc (fully connected layer) có số lượng neuron tương đối lớn, được thiết kế để biểu diễn các đặc trưng cao cấp của hình ảnh.

3.1.6.2.2. Dữ liệu huấn luyện

Mô hình của hệ thống đã được huấn luyện trên một bộ dữ liệu của các sản phẩm đang có trên hệ thống chứa khoảng 130-200 hình ảnh đại diện sản phẩm. Sự đa dạng các sản phẩm trong 14 danh mục sản phẩm, dữ liệu huấn luyện sẽ có cấu trúc như **Hình 3. 5**, trong đó chỉ lấy các ảnh đại diện của sản phẩm để huấn luyện cho mô hình, ảnh đại diện sẽ có dạng là <timestamp>_main.jpg trong mỗi thư mục như **Hình 3. 5**



Hình 3. 5. Cây thư mục dữ liệu huấn luyện

3.1.6.2.3. Tiền xử lý dữ liệu

Trước khi đưa dữ liệu vào mô hình cần phải thực hiện một chuỗi các bước tiền xử lý để đảm bảo rằng dữ liệu là phù hợp và có thể được sử dụng hiệu quả trong quá trình huấn luyện. Bước tiền xử lý bao gồm:

- **Tải ảnh:** Xây dựng một hàm tải ảnh để đảm bảo rằng tất cả các hình ảnh từ tập dữ liệu được đưa vào mô hình một cách hiệu quả.
- **Chuyển đổi dữ liệu:** Ảnh sau khi tải được chuyển đổi thành định dạng mà mô hình có thể xử lý, thông qua việc sử dụng thư viện Keras để chuyển đổi ảnh thành mảng các giá trị pixel.
- **Tiền xử lý thêm:** thực hiện các bước tiền xử lý bổ sung như chuẩn hóa giá trị pixel để đảm bảo rằng dữ liệu là ổn định và thích hợp cho quá trình huấn luyện.

3.1.6.2.4. Xây dựng kiến trúc mô hình

Mô hình sử dụng là một biến thể của mô hình VGG16, một mô hình mạng nơ-ron tiên tiến đã được chứng minh hiệu suất cao trong nhiều nhiệm vụ thị giác máy tính. Kiến trúc mô hình bao gồm:

- **Lớp Convolutional:** Được sử dụng để trích xuất đặc trưng từ ảnh đầu vào.
- **Lớp Pooling:** Giảm kích thước của đặc trưng để giảm chiều của dữ liệu và giảm độ phức tạp tính toán.
- **Lớp Fully-Connected:** Được sử dụng để thực hiện phân loại và xuất ra dự đoán.

3.1.6.2.5. Quá trình huấn luyện

Quá trình huấn luyện được thực hiện bằng cách trích xuất ra các đặc trưng của hình ảnh như **Hình 3. 6**, sau đó lưu ở theo 1 cặp giá trị đặc trưng (feature) và tên của thư mục chứa hình ảnh, tên thư mục được thể hiện như các thư mục con trong thư mục products ở **Hình 3. 5**

```
# Tạo một từ điển rỗng để lưu trữ đặc trưng và tên thư mục
features_dict = {}
for i, image_path in enumerate(images):
    # Giả sử image_path có định dạng: '/content/drive/MyDrive/products_TrangTri'
    folder_name = image_path.split('/')[-2]

    img, x = load_image(image_path);
    feat = feat_extractor.predict(x)[0]
    # Lưu trữ đặc trưng và tên thư mục vào từ điển
    features_dict[image_path] = {'feature': feat, 'folder_name': folder_name}

1/1 [=====] - 1s 909ms/step
1/1 [=====] - 1s 918ms/step
1/1 [=====] - 1s 955ms/step
1/1 [=====] - 1s 895ms/step
1/1 [=====] - 1s 832ms/step
1/1 [=====] - 1s 820ms/step
1/1 [=====] - 1s 797ms/step
1/1 [=====] - 1s 512ms/step
1/1 [=====] - 0s 496ms/step
1/1 [=====] - 0s 492ms/step
1/1 [=====] - 1s 519ms/step
1/1 [=====] - 1s 525ms/step
1/1 [=====] - 1s 532ms/step
1/1 [=====] - 1s 505ms/step
1/1 [=====] - 0s 494ms/step
1/1 [=====] - 1s 507ms/step
1/1 [=====] - 0s 489ms/step
1/1 [=====] - 0s 498ms/step
1/1 [=====] - 1s 501ms/step
```

Hình 3. 6. Quá trình huấn luyện mô hình tìm kiếm hình ảnh

3.1.6.2.6. Đánh giá và lưu trữ mô hình

Sau khi hoàn thành số lượng trích xuất các đặc trưng cho toàn bộ hình ảnh, tiếp theo là đánh giá hiệu suất của mô hình trên tập kiểm thử để đảm bảo rằng nó đáp ứng đúng yêu cầu của hệ thống. Nếu hiệu suất đạt đến một ngưỡng xác định, Lưu trữ mô hình để sử dụng trong quá trình triển khai trên trang web.

Sau khi huấn luyện xong mô hình, đây là kết quả của việc lấy ra thử ngẫu nhiên một nhóm các tấm ảnh có sự tương đồng nhau như **Hình 3. 7** và kết quả của quá trình huấn luyện được lưu lại ở một file duy nhất có tên là data.pkl

```

# load all the similarity results as thumbnails of height 100
thumbs = []
for idx in idx_closest:
    img = image.load_img(images[idx])
    img = img.resize((int(img.width * 100 / img.height), 100))
    thumbs.append(img)

# concatenate the images into a single image
concat_image = np.concatenate([np.asarray(t) for t in thumbs], axis=1)

# show the image
plt.figure(figsize = (16,12))
plt.imshow(concat_image)

```

<matplotlib.image.AxesImage at 0x7bae3ef79bd0>

Hình 3. 7. Kết quả lấy ra ngẫu nhiên 1 nhóm các ảnh tương đồng

Sau đó thử tìm kiếm các ảnh tương đồng bằng một hình ảnh được upload lên khác

Text(0.5, 1.0, 'result images - Folders: product104, product62, product63, product87, product123')
query image



result images - Folders: product104, product62, product63, product87, product123



Hình 3. 8. Kết quả tìm kiếm ảnh tương đồng khi upload một hình ảnh chỉ định

Kết quả ở **Hình 3. 8** có thể thấy rằng nó đã cho ra các hình ảnh tương đồng và bên cạnh đó cũng lấy ra được tên các thư mục mà lúc trích xuất hình ảnh đã lưu lại, có được những thông tin này có thể dễ dàng xử lý để truy vấn ở database lấy các thông tin của sản phẩm.

3.1.6.3. Mô hình chatbot

Với chatbot, hệ thống sẽ cung cấp thông tin sản phẩm và hỗ trợ người dùng trong quá trình mua sắm một cách linh hoạt và tiện lợi.

3.1.6.3.1. Mô hình sử dụng

Mô hình chatbot được sử dụng là một mô hình học máy dựa trên mạng nơ-ron sử dụng thư viện Tensorflow và TFlearn. Mô hình được huấn luyện để thực hiện tác vụ phân loại và tạo ra câu trả lời cho các mẫu đầu vào.

Mô hình nơ-ron có ba tầng ẩn (fully connected layers) với hàm kích hoạt softmax ở tầng đầu ra để thực hiện phân loại nhiều lớp. Cụ thể:

- Mạng nơ-ron có một tầng đầu vào với số lượng nơ-ron bằng số chiều của vector đặc trưng đầu vào (tổng số từ trong từ vựng).
- Hai tầng ẩn với 10 nơ-ron mỗi tầng được thêm vào, và một tầng đầu ra có số lượng nơ-ron bằng số lớp.
- Hàm kích hoạt softmax được sử dụng ở tầng đầu ra để đưa ra xác suất của từng lớp.

Mô hình được huấn luyện bằng cách sử dụng hàm mất mát "categorical_crossentropy" và tối ưu hóa bằng thuật toán Adam.

3.1.6.3.2. Dữ liệu huấn luyện

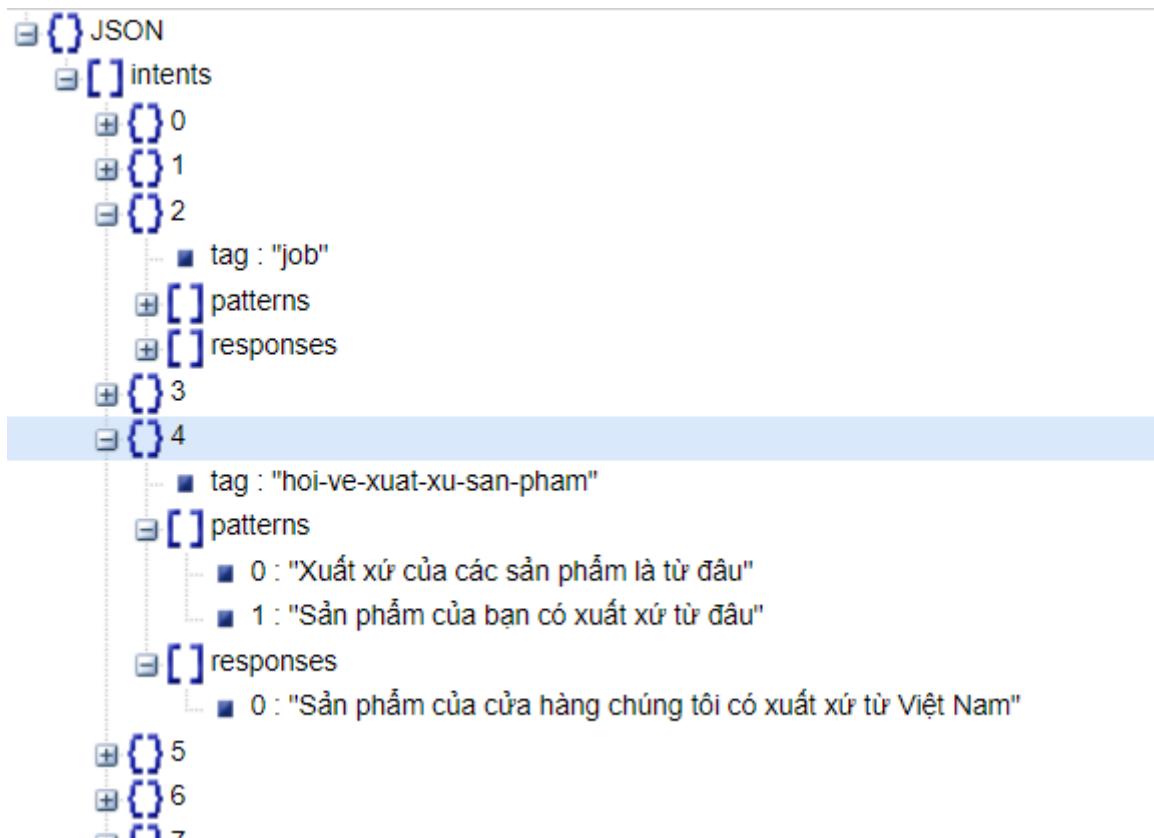
Dữ liệu huấn luyện cho mô hình chatbot có chứa các mẫu tag, câu hỏi và câu trả lời tương ứng như **Bảng 3. 2** và **Hình 3. 9**

Bảng 3. 2. Bảng mô tả các intent của dữ liệu đào tạo chatbot

STT	Tag	Giải thích
1	cam-thay-ban-than-nhu-the-nao	Hỏi về chatbot cảm thấy bản thân nó như thế nào
2	disparage	Chê bai chatbot
3	goodbye	Nói lời tạm biệt tới chatbot
4	greeting	Chào hỏi chatbot
5	hoi-tham	Hỏi thăm chatbot
6	hoi-ve-buon-ban-cai-gi	Hỏi về cửa hàng đang bán mặt hàng gì
7	hoi-ve-chat-lieu-san-pham	Hỏi về các chất liệu của sản phẩm ở cửa hàng
8	hoi-ve-chinh-sach-bao-hanh	Hỏi về chính sách bảo hành

9	hoi-ve-chuong-trinh-khuyen-mai	Hỏi về chương trình khuyến mãi
10	hoi-ve-kiem-tra-tinh-trang-don-hang	Hỏi về cách kiểm tra tình trạng đơn hàng
11	hoi-ve-mua-tuy-chon	Hỏi về có thể mua một sản phẩm với các yêu cầu tùy chọn
12	hoi-ve-nguo-i-than	Hỏi về người thân của chatbot
13	hoi-ve-chi-phi-van-chuyen	Hỏi về chi phí vận chuyển
14	hoi-ve-phuong-thuc-thanh-toan	Hỏi về phương thức thanh toán
15	hoi-ve-san-pham-giam-gia	Hỏi về sản phẩm giảm giá
16	hoi-ve-san-pham-pho-bien	Hỏi về sản phẩm phổ biến
17	hoi-ve-thoi-gian-giao-hang	Hỏi về thời gian giao hàng
18	hoi-ve-tu-van-sap-xep	Hỏi về cách trang trí sản phẩm sao cho đẹp
19	hoi-ve-xuat-xu-san-pham	Hỏi về xuất xứ của sản phẩm
20	introduce	Yêu cầu giới thiệu bản thân
21	job	Hỏi về các công việc chatbot có thể hỗ trợ
22	ok	Xử lý lại khi khách hàng nhắn các câu trả lời ngắn như ok, ừ,...
23	thanks	Khi khách hàng cảm ơn chatbot
24	what-your-name	Hỏi tên của chatbot

Các dữ liệu được thu thập và tiếp tục được cập nhật thêm rất nhiều khi triển khai mô hình và sử dụng trong quá trình kiểm thử.



Hình 3. 9. Mẫu cấu trúc intent gồm tag, câu hỏi và câu trả lời tương ứng

3.1.6.3.3. Tiền xử lí dữ liệu

Dữ liệu được tiền xử lí lại như xoá bỏ các kí tự đặc biệt, tokenize mỗi từ trong các câu, xoá đi các từ trùng lặp.

3.1.6.3.4. Xây dựng kiến trúc mô hình

Mô hình chatbot được xây dựng sử dụng mạng nơ-ron đầy đủ (DNN) với thư viện TensorFlow và TFLearn. Mô hình này dựa trên kiến trúc mạng nơ-ron, bao gồm một lớp input để biểu diễn túi từ (bag of words) từ câu hỏi đầu vào, hai lớp ẩn để học các đặc trưng, và một lớp output với hàm kích hoạt softmax để dự đoán nhãn của câu trả lời như **Hình 3. 10**.

```

# Building neural network
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, len(train_y[0]), activation='softmax')
net = tflearn.regression(net, optimizer='adam', loss='categorical_crossentropy')

```

Hình 3. 10. Mô hình của model Chatbot

3.1.6.3.5. Quá trình huấn luyện

Quá trình huấn luyện được thực hiện trên tập dữ liệu lớn được chia thành tập huấn luyện và tập validation. Chúng tôi sử dụng trình tối ưu hóa Adam để cập nhật trọng số của

mô hình và hàm mất mát categorical crossentropy để đo lường sự sai lệch giữa dự đoán và nhãn thực tế.

Quá trình này chia thành các epochs, mỗi epoch đại diện cho một lần duyệt qua toàn bộ tập huấn luyện. Các chỉ số hiệu suất như độ chính xác trên tập huấn luyện và tập validation được theo dõi để đánh giá sự tiến triển của mô hình.

Mô hình được huấn luyện qua rất nhiều lần từ 100 epoch, 150 epoch, 300 epoch **Bảng 3.3** và cuối cùng, sau nhiều lần điều chỉnh tham số, mô hình cho kết quả tốt nhất khi train với 250 epoch và với batch_size = 8.

Bảng 3.3. Kết quả thu được sau các lần đào tạo mô hình chatbot

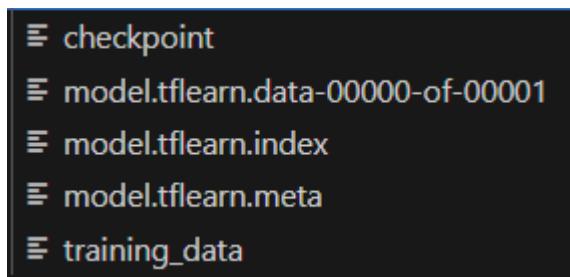
Lần train	epoch	loss	accuracy	total loss
1	100	0.06300	0.9927	0.06300
2	150	0.01436	1.0	0.01436
3	250	0.00054	1.0	0.0054

3.1.6.3.6. Đánh giá và lưu trữ mô hình

Sau quá trình đào tạo kết quả của mô hình đào tạo như **Hình 3.11** có hiệu suất tốt. Cả hai giá trị loss và accuracy đều rất thấp, với loss giảm xuống dưới 0.001 và độ chính xác (accuracy) đạt 100%. Điều này ngụ ý rằng mô hình đã học được mẫu dữ liệu đào tạo khá tốt và đang dự đoán chính xác trên tập dữ liệu này và sau đó cũng được lưu trữ lại như **Hình 3.12**.

```
Training Step: 8749 | total loss: 0.00058 | time: 0.104s
| Adam | epoch: 250 | loss: 0.00058 - acc: 1.0000 -- iter: 272/279
Training Step: 8750 | total loss: 0.00054 | time: 0.109s
| Adam | epoch: 250 | loss: 0.00054 - acc: 1.0000 -- iter: 279/279
--
```

Hình 3.11. Kết quả quá trình đào tạo mô hình chatbot



Hình 3.12. Kết quả sau khi đào tạo mô hình và được lưu trữ lại

3.1.6.4. Mô hình phân tích cảm xúc đánh giá

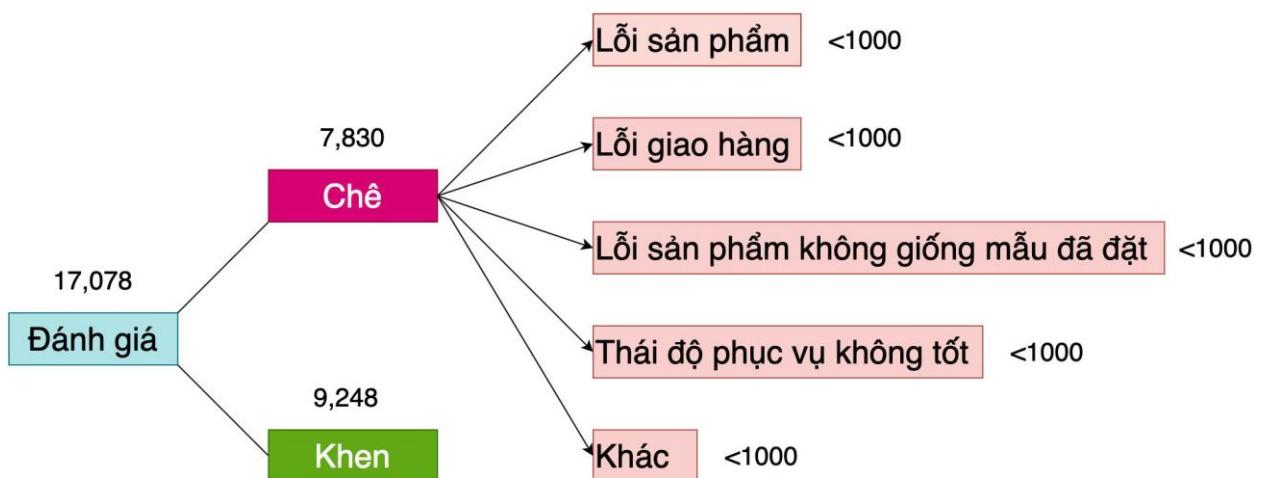
3.1.6.4.1. Mô hình sử dụng

Mô hình phân tích cảm xúc sử dụng mô hình học máy SVM (Support Vector Machine) với kernel tuyến tính (linear) để phân loại văn bản thành hai nhãn ‘good’ và ‘bad’. Mô hình SVM tìm ra ranh giới tuyến tính (linear boundary) giữa các điểm dữ liệu của hai nhãn để tối ưu hóa việc phân loại, bên cạnh đó, nhằm nâng cao hơn cho tính năng phân tích cảm xúc, hệ thống cũng phân loại chi tiết hơn với các đánh giá là bad nhằm mục đích hiểu rõ được khách hàng hơn. Kernel được sử dụng để ánh xạ dữ liệu từ không gian đặc trưng ban đầu sang không gian cao chiều hơn nếu cần thiết. Mô hình chỉ có một lớp, đó là lớp đầu ra, được xác định bởi hàm quyết định của SVM

3.1.6.4.2. Dữ liệu huấn luyện

Dữ liệu huấn luyện cho mô hình phân tích cảm xúc được thu thập với dữ liệu thực tế trên Shopee, có 2 mô hình, mô hình thứ nhất có 2 nhãn khen (good) và chê (bad). Mô hình thứ hai là mô hình phân tích chi tiết hơn về nhãn bad mà dựa vào mô hình một đã đánh giá như **Hình 3. 13**, ở mô hình thứ hai này được chia ra thành 5 nhãn, bao gồm:

- Lỗi sản phẩm
- Lỗi giao hàng
- Lỗi sản phẩm không giống mẫu đã đặt
- Thái độ phục vụ không tốt
- Khác (không nằm trong các nguyên nhân trên)

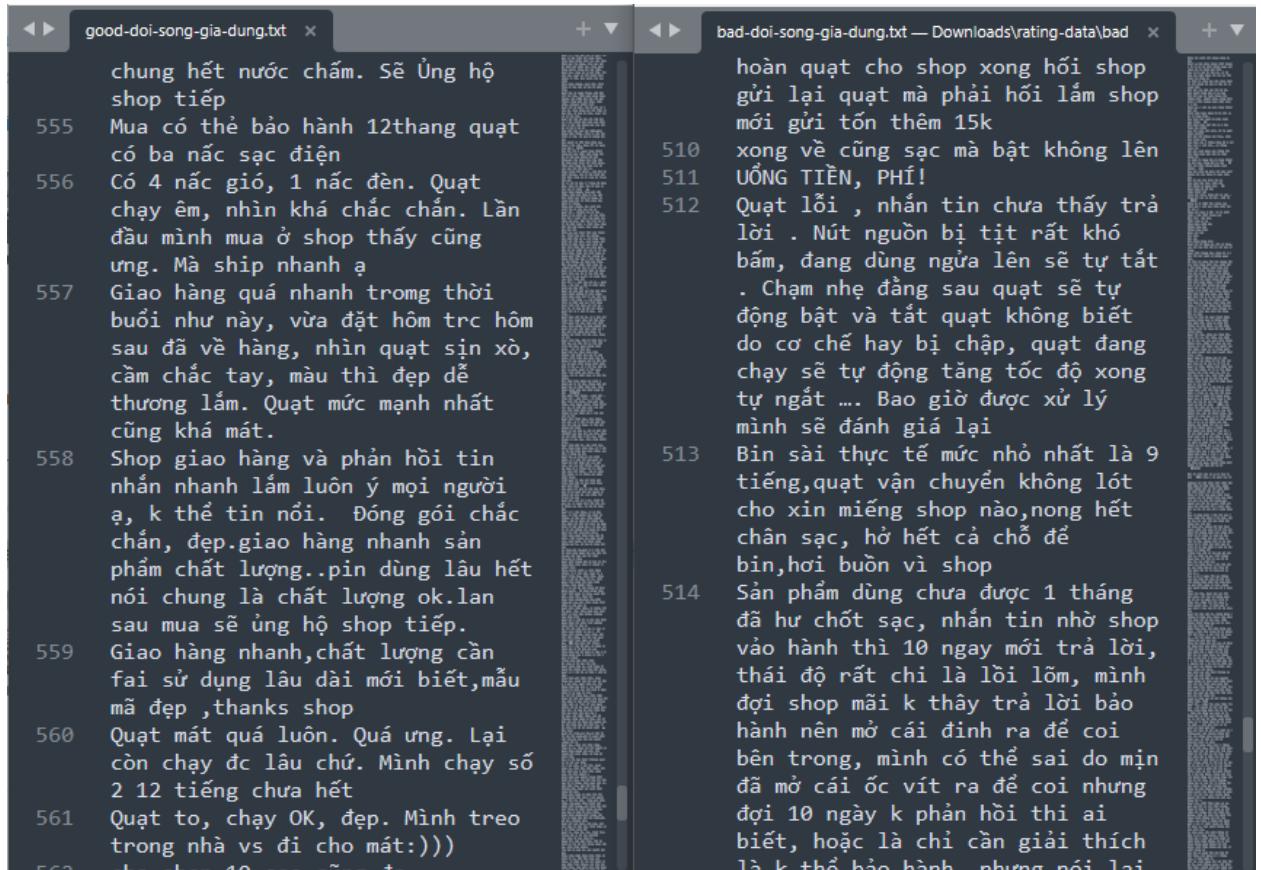


Hình 3. 13. Tổng quan về mô hình và số lượng dữ liệu training

Cụ thể, khi mô hình một với nhãn good thì sẽ lấy các đánh giá 5 sao của mọi người và với nhãn bad thì sẽ lấy các đánh giá 1 sao của mọi người. Sau khi thu thập và chọn lọc lại những đánh giá rõ ràng về mặt nội dung thì số lượng dataset thu thập được là 1258 đánh giá, trong đó có 665 đánh giá good và 593 đánh giá bad. Tuy nhiên khi mô hình được huấn luyện thì kết quả không được tốt như **Hình 3. 21** nên đã thu thập thêm dữ liệu cho mô hình để cải thiện khả năng phân tích của mô hình, lần thu thập dữ liệu thứ hai đã thu được 9,248

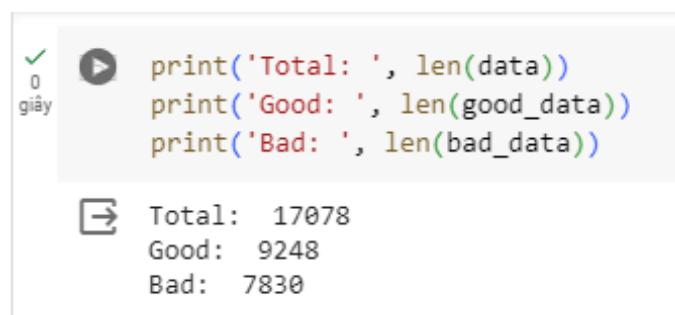
đánh giá good và 7,830 đánh giá bad thể hiện ở **Hình 3. 14** và **Hình 3. 15** lúc này mô hình đã có tổng là 17,078 đánh giá.

Với mô hình thứ hai, có 5 nhãn như **Hình 3. 16**, việc gán nhãn trở nên khó, phức tạp và mất nhiều thời gian hơn, nên mỗi nhãn chỉ có dữ liệu khoảng 1000 đánh giá trở xuống.



The image shows two side-by-side text editor windows. The left window is titled 'good-doi-song-gia-dung.txt' and contains reviews with IDs 555 through 562. The right window is titled 'bad-doi-song-gia-dung.txt' and contains reviews with IDs 510 through 514. Both windows display Vietnamese text comments from customers.

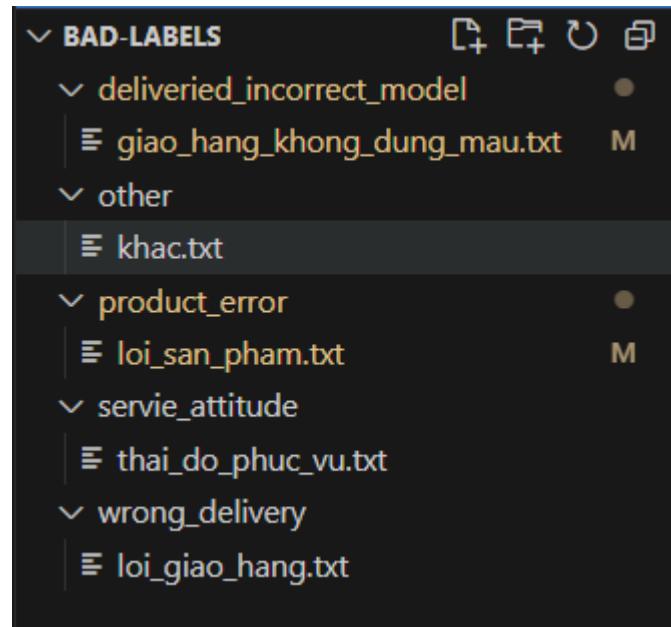
Hình 3. 14. Dataset nhận dạng cảm xúc thu thập được



```
✓ 0 giây
  ➔ print('Total: ', len(data))
      print('Good: ', len(good_data))
      print('Bad: ', len(bad_data))

  ➔ Total: 17078
      Good: 9248
      Bad: 7830
```

Hình 3. 15. Chi tiết số lượng dataset nhận dạng cảm xúc



Hình 3. 16. Cấu trúc dataset mô hình phân tích cảm xúc xấu

3.1.6.4.3. Tiền xử lí dữ liệu

Dữ liệu thô thu thập được từ **Hình 3. 14** sẽ chưa đủ tiêu chuẩn cho mô hình này, nên cần phải xử lí lại dữ liệu cho sạch. Cụ thể các bước tiền xử lí sẽ như sau:

- Đưa tất cả về chữ thường (lower)
- Tokenize các dữ liệu
- Loại bỏ các stopword và các dấu câu, kí tự đặc biệt

Sau bước tiền xử lí lại dữ liệu, lúc này dữ liệu sẽ trở nên tối ưu như **Hình 3. 17** cho bước huấn luyện mô hình tiếp theo, nhãn good sẽ là 1 và bad sẽ là 0.

```
[{'text': 'màu đẹp dày_dặn cầm nhẹ_tay form đẹp giặt ko biết chất_lượng thế_nào nhìn tổng_thể hàng ổn nhé đáng đồng_tiền nhu_cầu ủng_hộ tiếp_ạ',
 'label': 1},
 {'text': 'hình_ảnh kh liên_quan lầm giá rẻ áo xuất_sắc luôn đẹp hơn mong_đợi mấy bạn mua nha shop sao',
 'label': 1},
 {'text': 'hàng chất_liệu dày_dặn đường may kỹ thừa đúng mô_tả màu_sắc giống ảnh form áo_rộng đẹp',
 'label': 1},
 {'text': 'hàng chất_lượng nhẹ mua ủng_hộ shop nha mng ơi shop làm_việc tâm uy_tín chất_lượng nhẹ cảm_on shop k làm mik thất_vọng',
 'label': 1},
 {'text': 'chất vài dày_dặn k xù màu chuẩn k nước nha giặt máy oke nè chất vài mềm_mại mặc sướng lầm nhẹ',
 'label': 1},
 {'text': 'quá xứng_đáng ạ cách đóng_gói hàng biết shop chu_đáo tâm r ạ màu_sắc y hình chất_liệu dày_dặn lầm nèeee mng mua tặng ngy tặng bạn hết bài lunn mùi điểmm',
 'label': 1},
 {'text': 'sp đẹp giao hàng nhanh chọn size nhầm rộng tí sao giá rẻ hợp_lý làm biển chụp ảnh lấy ảnh đại nhé',
 'label': 1},
 {'text': 'mình thích hàng tốt đóng_gói cẩn_thận cảm_on shop lầm mang đến mình trải nghiệm tốt',
 'label': 1},
 {'text': 'sp nha phù_hợp mấy bạn sinh_viện hoặc nv văn_phòng diện ok lầm nè tác_phong lịch_sự luôn vài ok mát nha mặc thoảiái_mái',
 'label': 1},
 {'text': 'mua shop hàng khá ok mức giá sản_phẩm sales mua tiếp giao hàng nhanh_chóng ngày hàng',
 'label': 1},
 {'text': 'đẹp quá trời luôn ý vài đinh thật_sự nói thật lần đầu mua áo shopee dc chất vài ưng +hết cu đống mua mua rồi in dùi dẩn cuả luôn cái thừa nha mua ủng_hộ shop tiền ~'}
```

Hình 3. 17. Dữ liệu mô hình phân tích cảm xúc sau khi được xử lí

Dữ liệu sẽ được chia ra thành 2 tập dữ liệu test và train, trong đó 80% cho dữ liệu train và 20% cho dữ liệu test. Phàn này là phàn quan trọng để đảm bảo mô hình có khả năng tổng quát hoá và không overfitting.

3.1.6.4.4. Xây dựng kiến trúc mô hình

Mô hình SVM với kernel tuyến tính (linear) bằng cách sử dụng thư viện SVC từ Scikit-learn.

Kiến trúc của mô hình được thay đổi nhiều lần trong quá trình kiểm thử, cụ thể là ở lần đầu tiên với tập dữ liệu là 665 đánh giá good và 593 đánh giá bad, mô hình phân tích không tốt cho các đánh giá ngắn như ‘tốt’, ‘5 sao’, ‘tuyệt vời’,... thì kết quả phân tích bị sai như **Hình 3. 20**. Sau đó mô hình đã được bổ sung thêm dữ liệu với tổng là 17,078 đánh giá và lần cuối cùng thay đổi siêu tham số C của mô hình SVM thành 0.3.

3.1.6.4.5. Quá trình huấn luyện

Mô hình được huấn luyện sau khi chúng được vector hoá. Quá trình huấn luyện diễn ra khá nhanh do dữ liệu không được lớn lắm.

Mô hình đã được huấn luyện qua nhiều lần thay đổi tập dữ liệu và tham số như **Bảng 3. 4**

Bảng 3. 4. Kết quả sau các lần train mô hình thay đổi siêu tham số C

Lần train	Siêu tham số C	Label	Accuracy	Percision	Recall	F1-Score
1	1.0	bad	0.950	0.94	0.97	0.95
		good		0.96	0.93	0.95
2	2.0	bad	0.952	0.93	0.97	0.95
		good		0.97	0.93	0.95
3	3.0	bad	0.946	0.93	0.97	0.95
		good		0.97	0.92	0.94

3.1.6.4.6. Đánh giá và lưu trữ mô hình

Mô hình sau huấn luyện, sau nhiều lần thử ở **Bảng 3. 4** thì kết quả tốt nhất khi siêu tham số C là 3.0, kết quả chi tiết của mô hình khi tham số là 3.0 đã có sự cải thiện đáng kể như **Hình 3. 18**.

```

Accuracy: 0.8905152224824356
Classification Report:
precision    recall   f1-score   support
          0       0.87      0.89      0.88     1526
          1       0.91      0.89      0.90     1890
accuracy                           0.89     3416
macro avg       0.89      0.89      0.89     3416
weighted avg    0.89      0.89      0.89     3416

```

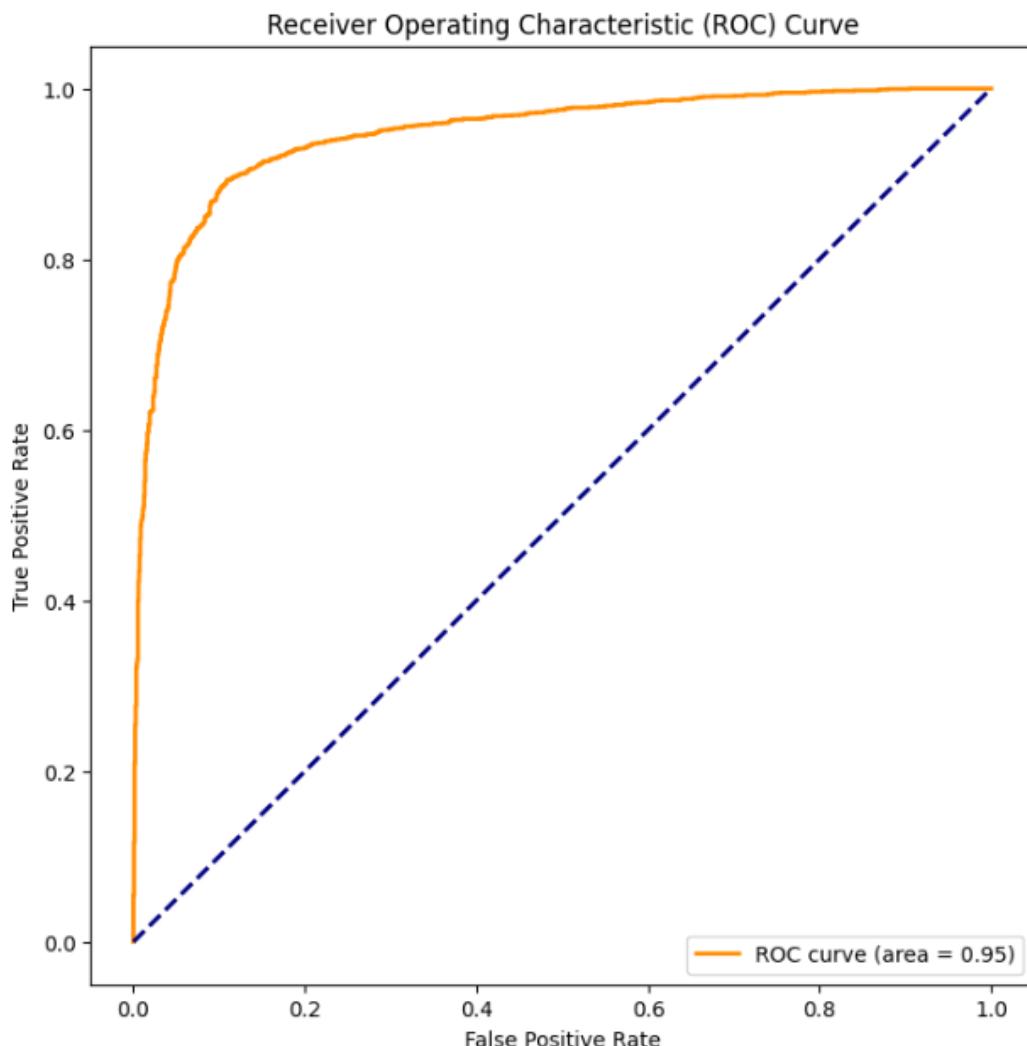
Hình 3. 18. Kết quả huấn luyện mô hình phân tích cảm xúc C = 3.0

Mô hình phân loại được huấn luyện đã đạt được một độ chính xác (accuracy) đáng kể, lên tới khoảng 89%. Điều này chỉ ra rằng mô hình của bạn có khả năng đưa ra dự đoán chính xác cho đa số các trường hợp trong tập dữ liệu kiểm thử.

Precision, Recall và F1-Score

- Đối với lớp 0 (negative):
 - Precision: 87% - Đây là tỉ lệ các trường hợp được dự đoán là negative mà thực sự là negative. Mô hình có khả năng dự đoán negative một cách chính xác.
 - Recall: 89% - Tỉ lệ các trường hợp negative thực sự mà mô hình đã dự đoán đúng. Độ chính xác tỉ lệ khá cao và đáng tin cậy.

- F1-Score: 88% - Trung bình điều hòa giữa precision và recall. Giá trị này cho biết sự cân bằng giữa độ chính xác và độ phủ của mô hình đối với lớp negative.
- Đối với lớp 1 (positive):
 - Precision: 91% - Đạt được tỉ lệ dự đoán positive chính xác, đối với các trường hợp thực sự là positive.
 - Recall: 89% - Tỉ lệ các trường hợp positive thực sự mà mô hình đã dự đoán đúng. Mô hình có khả năng phát hiện positive một cách hiệu quả.
 - F1-Score: 90% - Sự cân bằng giữa precision và recall cho lớp positive, cho thấy mô hình hiệu quả trong việc phân loại các trường hợp positive.



Hình 3. 19. Biểu ROC của mô hình phân tích cảm xúc

Dựa vào biểu đồ ROC ở **Hình 3. 19** có thể thấy rằng "TOC curve (area = 0.95)" có nghĩa là diện tích dưới đường ROC của mô hình là 0.95, điều này là một dấu hiệu tích cực về hiệu suất của mô hình. Mô hình có khả năng phân loại giữa các trường hợp tích cực và tiêu cực một cách hiệu quả, với tỉ lệ đúng và tỉ lệ sai được thể hiện trên biểu đồ ROC.

Nhưng mô hình không phân tích tốt cho các trường hợp đầu vào là những câu ngắn như **Hình 3. 20**

```
... tốt
=> 0
tốt lắm
=> 1
tệ lắm
=> 0
5 sao
=> 0
tuyệt vời
=> 0
sản phẩm tuyệt vời
=> 0
sản phẩm rất tốt
=> 1
```

Hình 3. 20. Kết quả của mô hình phân tích cảm xúc ở lần train thứ nhất

```
5 sao
=> 1
tốt
=> 1
tốt lắm
=> 1
tệ lắm
=> 0
5 sao
=> 1
tuyệt vời
=> 1
sản phẩm tuyệt vời
=> 1
sản phẩm rất tốt
=> 1
```

Hình 3. 21. Kết quả của mô hình phân tích cảm xúc ở lần train thứ hai

Ở kết quả lần đào tạo thứ hai **Hình 3. 21** thì đã ổn hơn rất nhiều so với lần đào tạo đầu tiên.

Tiếp theo sẽ thử cho mô hình phân tích với dữ liệu thực tế, các đánh giá ở **Hình 3. 22** được lấy ở một sản phẩm có đánh giá 4 sao trên Shopee.

```

*** Chưa sử dụng nhưng thấy cũng ô kê la Máy có thể đánh bông trứng hoặc kem
=> 0
Shipper nhiệt tình giáo hàng nhanh hoi hàng cần thận Sẽ ủng hộ tiếp
=> 1
Hình ảnh chỉ mang tính chất minh họa Máy đánh cafe ngon. Đẹp. Dùng pin mình
=> 1
Máy đẹp mik đúng thử rồi máy rất mạnh nha lần sau ghé shop tiếp
=> 1
Chăm sóc khác hàng tốt giao hàng nhanh hơn .chất lượng tốt.lần sau nên tặng
=> 1
Máy chạy đc ổn, m chưa đánh trứng nên ko biết đc ko
=> 0
Mình dùng r mọi ng ạ , mua thử lm cfe trứng nma loại này yếu lắm k đánh đc t
=> 0
giao hàng thì quá lâu k đc tặng kèm bin hơi buồn mn cân nhắc khi mua
=> 1
Mk lúc bật ko thì nhìn có vẻ mạnh nhưng khi mk dùng thử thì thấy ní rất yếu
=> 0

```

Hình 3. 22. Kết quả của mô hình phân tích cảm xúc cho những đánh giá dài

Qua kết quả phân tích của **Hình 3. 22** có thể thấy mô hình phân tích khá ổn, tuy nhiên do dữ liệu đào tạo quá ít nên không thể mong muốn mô hình phân tích chính tuyệt đối được. Mô hình cần phải được thu thập nhiều hơn và do các đánh giá của người Việt đôi khi quá đa dạng, như ví dụ ở **Hình 3. 22** có thể thấy các từ viết tắt như ‘cfe’, ‘k’, ‘mk’, ‘ô kê la’,...

Sau khi sử dụng mô hình một để phân tích được các đánh giá xấu, tiếp tục có thể phân tích các đánh giá xấu này để biết chi tiết hơn về nguyên nhân như

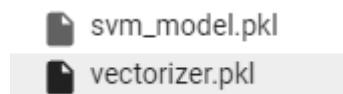
```

sản phẩm bị bể, hư hết rồi
=> product_error
nhắn tin không kèm trả lời
=> servie_attitude

```

Hình 3. 23. Kết quả của mô hình phân tích cảm xúc chê bai

Sau đó mô hình và vectorizer cũng được lưu trữ lại để sử dụng cho hệ thống bán hàng như **Hình 3. 24**



Hình 3. 24. Lưu trữ lại mô hình phân tích cảm xúc

3.1.6.5. Hệ thống gợi ý

3.1.6.5.1. Mô hình sử dụng

Hệ thống gợi ý sẽ áp dụng phương pháp xử lý ngôn ngữ tự nhiên và tính toán đại số tuyến tính đại số tuyến tính, cụ thể mô hình sẽ:

- Tạo ma trận TF-IDF

- Tính độ tương đồng bằng độ đo tương tự cosin dựa trên ma trận TF-IDF

3.1.6.5.2. Dữ liệu đầu vào

Với mục tiêu là gợi ý các sản phẩm có liên quan cho một sản phẩm mà người dùng tìm kiếm hoặc một sản phẩm chỉ định thì dữ liệu đầu vào sẽ là các sản phẩm, danh mục đang có trên hệ thống khoảng 100 sản phẩm từ 14 danh mục, dataset sẽ là 1 file csv có dạng như **Hình 3. 25**

index	id_product		name_product	category
0	0	130	Tranh hổ phách "Thiên đường"	Tranh
1	1	129	Bức tranh "The Falcon On The Hunt"	Tranh
2	2	128	Tranh hổ phách "Sư tử vua thú dữ"	Tranh
3	3	127	Tranh hổ phách "Liễu băng nước"	Tranh
4	4	126	Tranh hổ phách "Chuyện tình"	Tranh

Hình 3. 25. 5 dòng đầu tiên trong dataset

3.1.6.5.3. Tiền xử lý dữ liệu

Các dữ liệu sản phẩm chưa được chuẩn vì có những kí tự như \t hoặc cá khoảng trắng dư thừa, kí tự đặc biệt,... nên cần phải được xử lý lại để đặt được kết quả tốt khi sử dụng mô hình.

- Đưa tất cả về chữ thường (lower)
- Tokenize các dữ liệu
- Loại bỏ các stopword và các dấu câu, kí tự đặc biệt

Chọn lọc lại các đặc trưng quan trọng như tên sản phẩm (name_product) và danh mục (category) để sử dụng trong mô hình. Sau đó xử lí 2 đặc trưng này bằng cách ghép chuỗi lại với nhau có dạng <name_product><khoảng trắng><category> như **Hình 3. 26**

```

[ ] # selecting the relevant features for recommendation

selected_features = ['name_product', 'category']
print(selected_features)

['name_product', 'category']

[ ] # replacing the null values with null string

for feature in selected_features:
    products_data[feature] = products_data[feature].fillna('')

[ ] # combining all the 2 selected features

combined_features = products_data['name_product']+''+products_data['category']

[ ] print(combined_features)

0           tranh hổ_phách thiên_đường tranh
1           bức tranh the falcon on the hunt tranh
2           tranh hổ_phách sư_tử vua thú_dữ tranh
3           tranh hổ_phách liễu_bằng nước tranh
4           tranh hổ_phách tình tranh
...
90          ghế lưng lưới cao chân xoay ghế
91          đồng_hồ treo tường kim trôi cao_cấp hono đồng_hồ
92          đèn led trang_trí bóng tròn dài 40 bóng tranh b...
93          gấu bông mèo béo gấu bông sang trọng đồ_chơi t...
94          tranh cát chuyển_động hình_tròn tranh cát chày...
Length: 95, dtype: object

```

Hình 3. 26. Tiền xử lí dữ liệu hệ thống gợi ý

3.1.6.5.4. Chuyển đổi dữ liệu văn bản

Chuyển các dòng văn bản đã được tiền xử lý trước đó thành dạng vector bằng cách sử dụng “TfidfVectorizer” để chuyển văn bản thành vector đặc trưng TF-IDF, làm nền tảng cho việc tính toán độ tương đồng cosin.

```

vectorizer = TfidfVectorizer()

[ ] feature_vectors = vectorizer.fit_transform(combined_features)

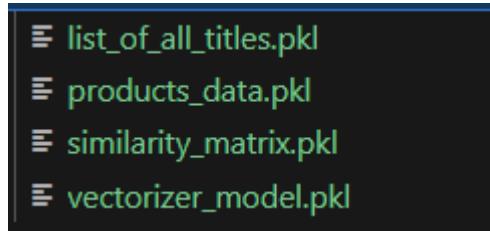
[ ] print(feature_vectors)

[ ] (0, 304)      0.46116056367434016
[ ] (0, 234)      0.46116056367434016
[ ] (0, 198)      0.34256047400282247
[ ] (0, 137)      0.34256047400282247
[ ] (0, 253)      0.5830664733330607
[ ] (1, 127)      0.34226312284735455
[ ] (1, 196)      0.31377410057215277
[ ] (1, 93)       0.34226312284735455

```

3.1.6.5.5. Đánh giá và lưu trữ mô hình

Sau khi đã chuyển đổi các dữ liệu thành các vector thì lúc này sẽ lưu trữ lại các dữ liệu này để sau đó nạp vào hệ thống bán hàng để sử dụng **Hình 3. 27.**



Hình 3. 27. Các dữ liệu đã vector hoá được lưu lại của hệ thống gợi ý

3.2. Giải pháp cài đặt

Để triển khai hệ thống cần tiến hành và sử dụng các công nghệ cụ thể sau:

3.2.1. Môi trường cài đặt

Hệ điều hành: macOS Ventura 13.4 (hoặc bất kỳ hệ điều hành Linux phổ biến khác).

Máy chủ web: Nginx hoặc Apache.

Phiên bản PHP: PHP 8 trở lên.

Phiên bản VueJS: VueJS 3.

Phiên bản Python: 3.10 hoặc 3.12.

Phiên bản TensorFlow: 2.10 hoặc tensorflow-macos

Code Editor: Visual Studio Code.

3.2.2. Cài đặt các công cụ cần thiết

Homebrew: Trước tiên, cài đặt Homebrew, hệ thống quản lý gói:

```
/bin/bash -c "$(curl -fsSL
```

```
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Composer: Quản lý gói phụ thuộc cho PHP:

```
brew install composer
```

3.2.3. Cài đặt Laravel

Cài đặt Laravel bằng lệnh composer như sau:

```
composer global require laravel/installer
```

3.2.4. Cài đặt Vue.js và TypeScript

Node.js và NPM:

```
brew install node
```

Cài đặt Vue.js với TypeScript:

```
npm install vue  
npm install vue-router
```

3.2.5. Cài đặt và cấu hình MySQL, Apache, PHPMyAdmin

Ở đây do đề tài này có sử dụng Docker nên việc cài đặt các thành phần như MySQL, máy chủ Apache, PHPMyAdmin, PHP, Xdebug,... đều trở nên đơn giản và nhanh hơn nhiều so với cách cài đặt thủ công, truyền thống.

Sau khi cài đặt, tạo một cơ sở dữ liệu mới cho dự án và cấu hình thông tin kết nối trong tệp .env của Laravel.

3.2.6. Phát triển và tích hợp thư viện

Sử dụng Composer để cài đặt các gói phụ thuộc cho Laravel.

Sử dụng NPM (hoặc Yarn) để cài đặt và quản lý các gói phụ thuộc cho Vue.js và Typescript.

Sử dụng Pip để cài đặt các gói phụ thuộc của python cho mô hình AI.

3.2.7. Tham số và cấu hình

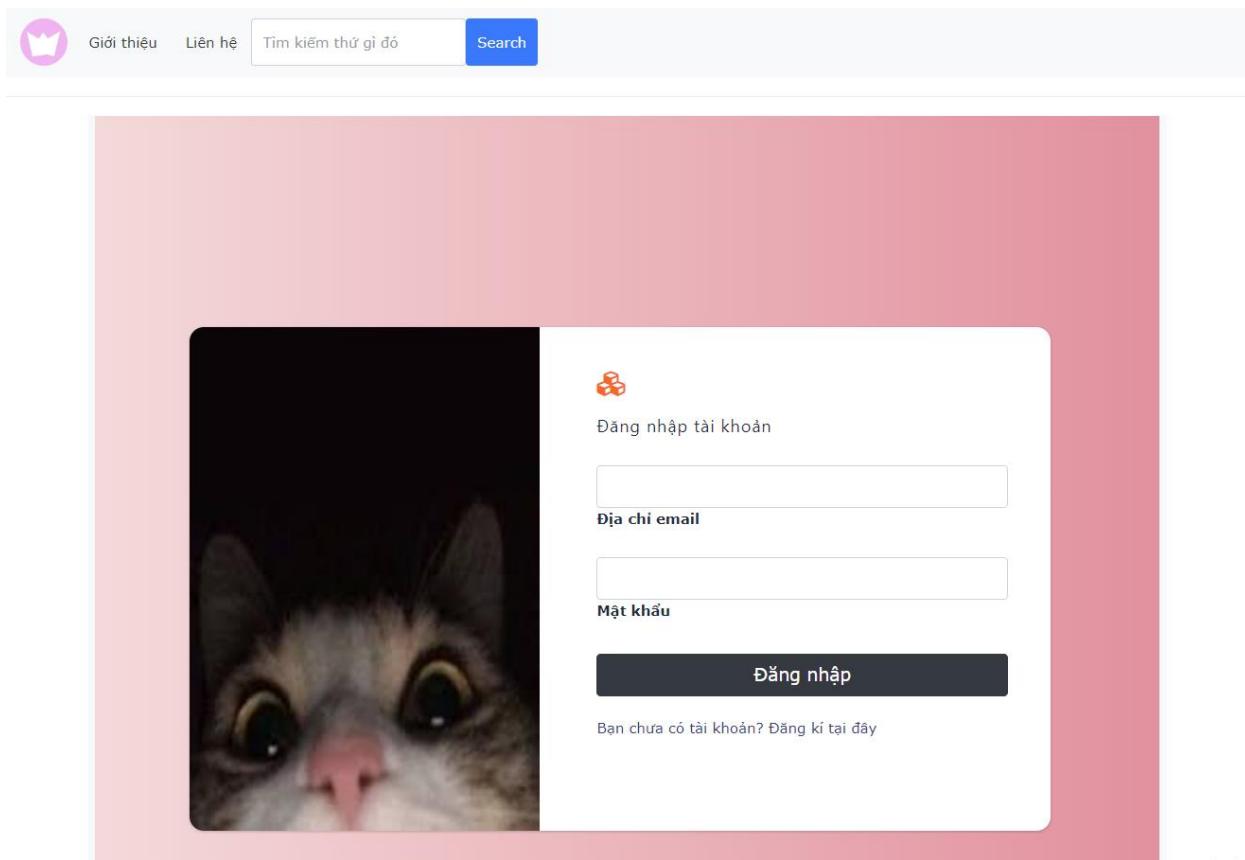
Tất cả các thông tin cấu hình như thông tin kết nối CSDL, API keys, và các tham số khác được lưu trữ trong tệp .env của Laravel.

Kết hợp những hướng dẫn trên, đề tài sẽ có một hệ thống hoạt động mượt mà, tận dụng tối đa các công nghệ và thư viện tiên tiến.

3.3. Thiết kế giao diện

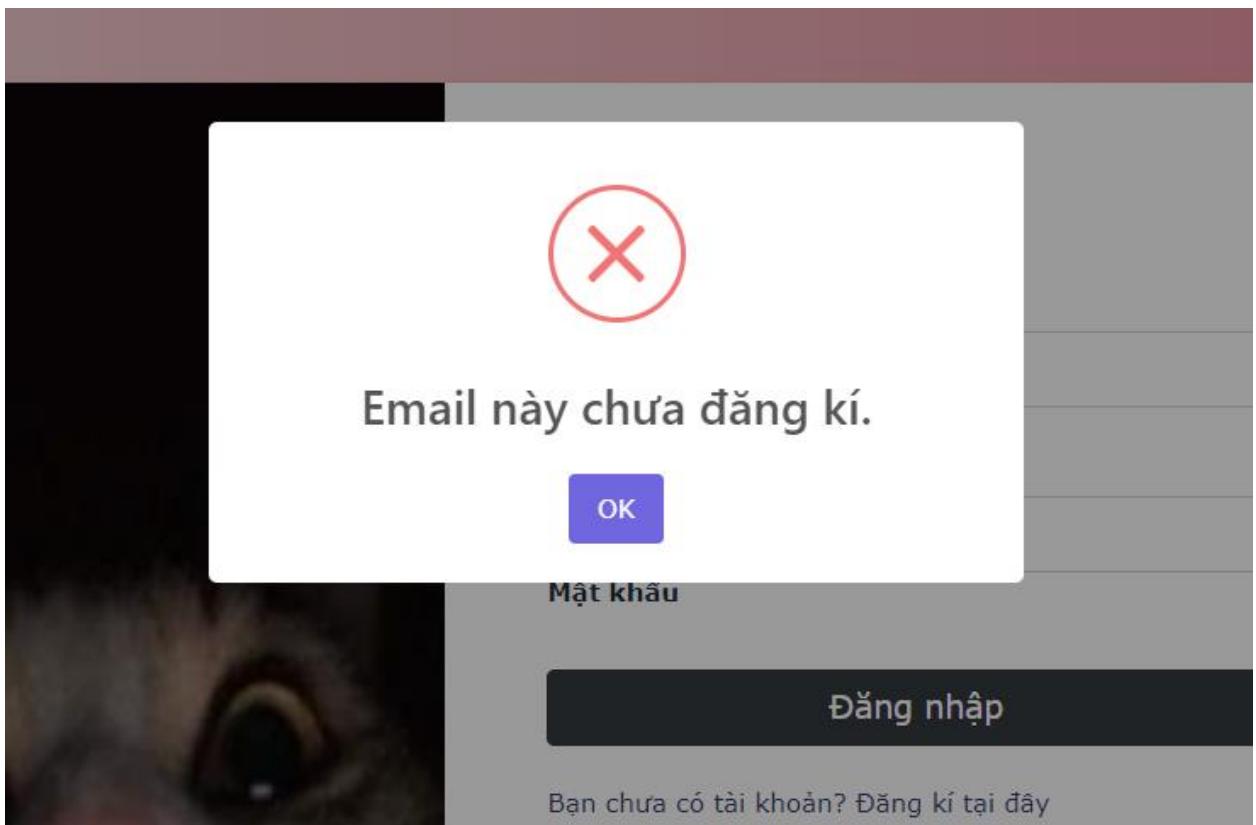
3.3.1. Giao diện người dùng

Giao diện đăng nhập website như **Hình 3. 28**. Trên **Hình 3. 28**, người dùng sẽ nhập vào email và mật khẩu của tài khoản. Khi ấn nút đăng nhập website sẽ thực hiện đăng nhập. Nếu email đã được đăng ký trước đó, website sẽ chuyển người dùng vào giao diện trang chủ như **Hình 3. 33**.

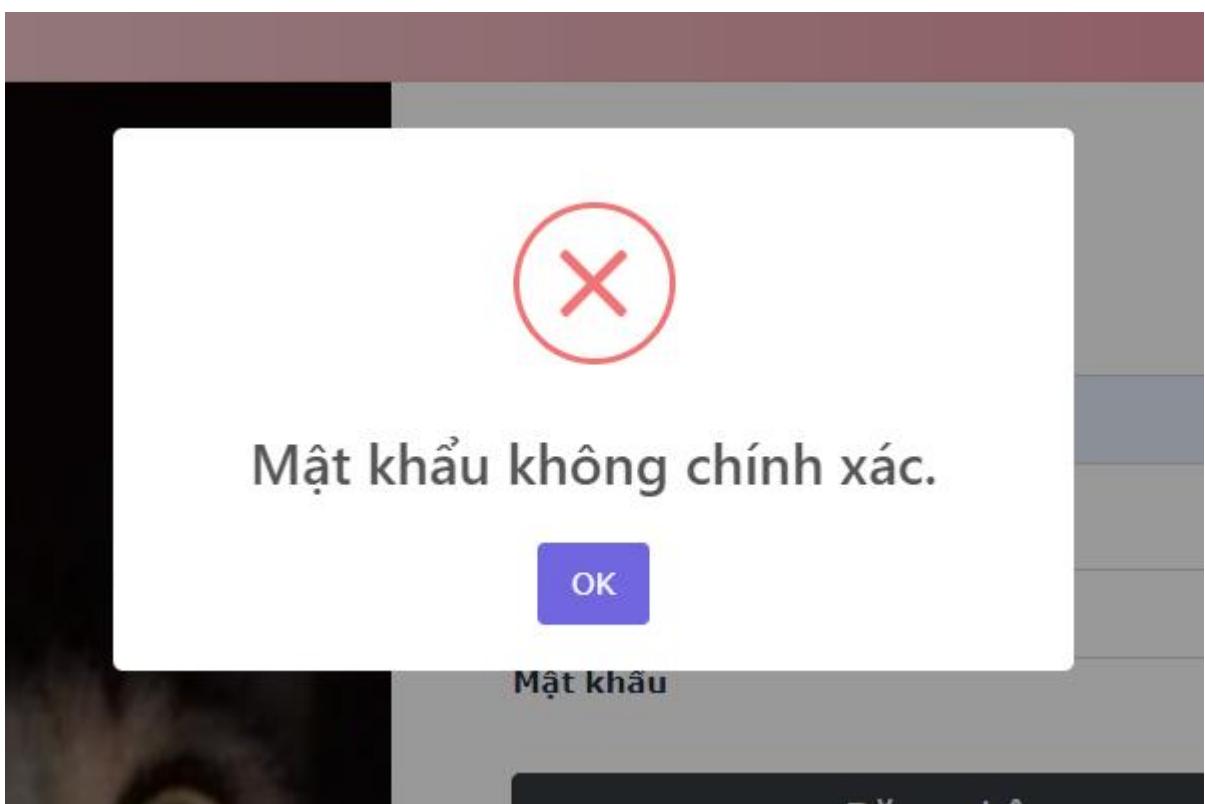


Hình 3. 28. Giao diện trang đăng nhập

Nếu email chưa đăng ký hệ thống sẽ báo lỗi cho người dùng biết rằng email này chưa được đăng ký như **Hình 3. 29** hoặc nếu như người dùng nhập vào email đã được đăng ký nhưng mà sai mật khẩu, hệ thống lúc này sẽ báo lỗi tương ứng cho trường hợp sai mật khẩu như **Hình 3. 30**.

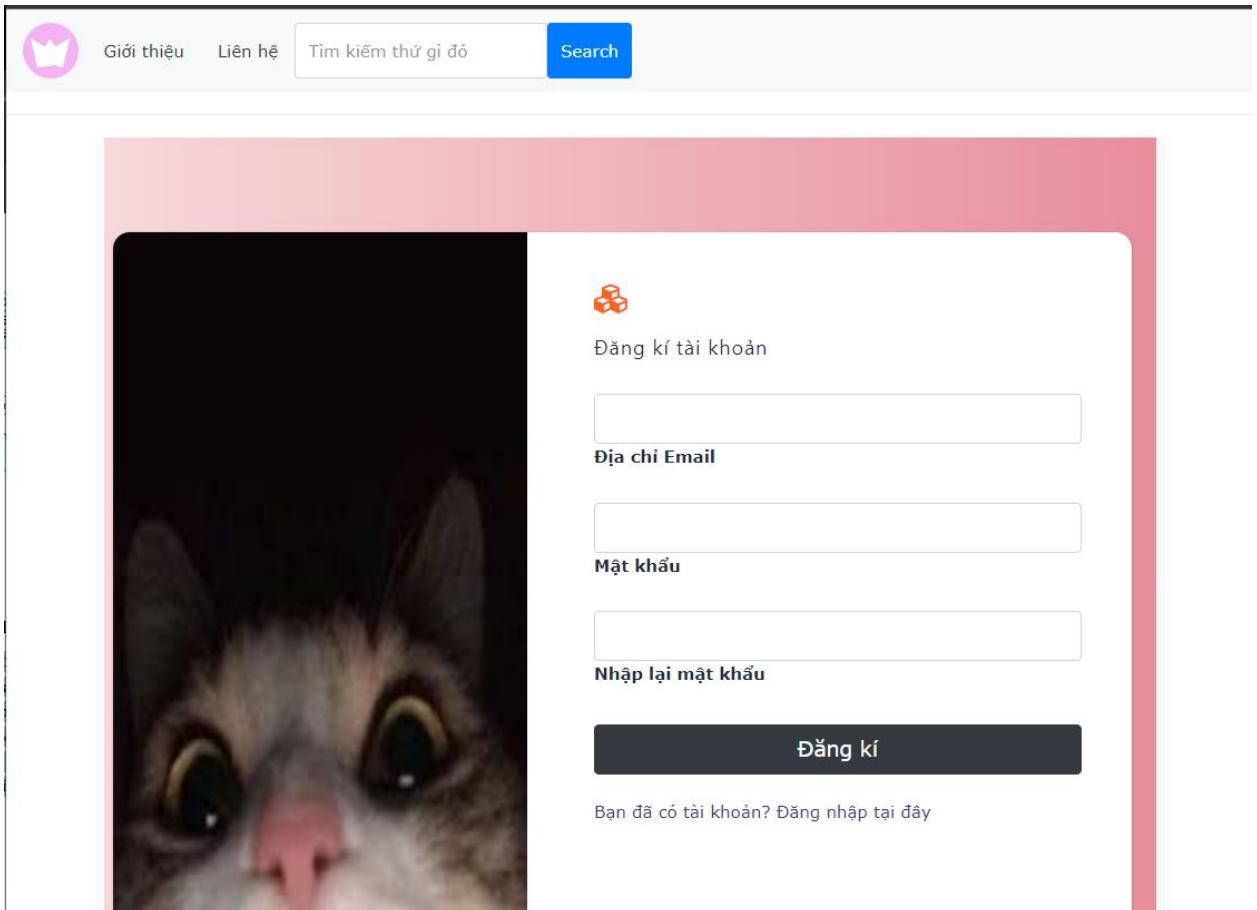


Hình 3. 29. Popup báo lỗi “Email này chưa đăng ký”



Hình 3. 30. Popup báo lỗi “Mật khẩu không chính xác”

Trong trường hợp mà người dùng nhận được thông báo trên thì người dùng có thể sẽ chuyển hướng sang trang đăng ký như **Hình 3. 31**. Tại đây người dùng sẽ nhập vào các thông tin như email, mật khẩu, nhập lại mật khẩu và sau đó nhấn nút “Đăng ký” để thực hiện quá trình đăng ký.



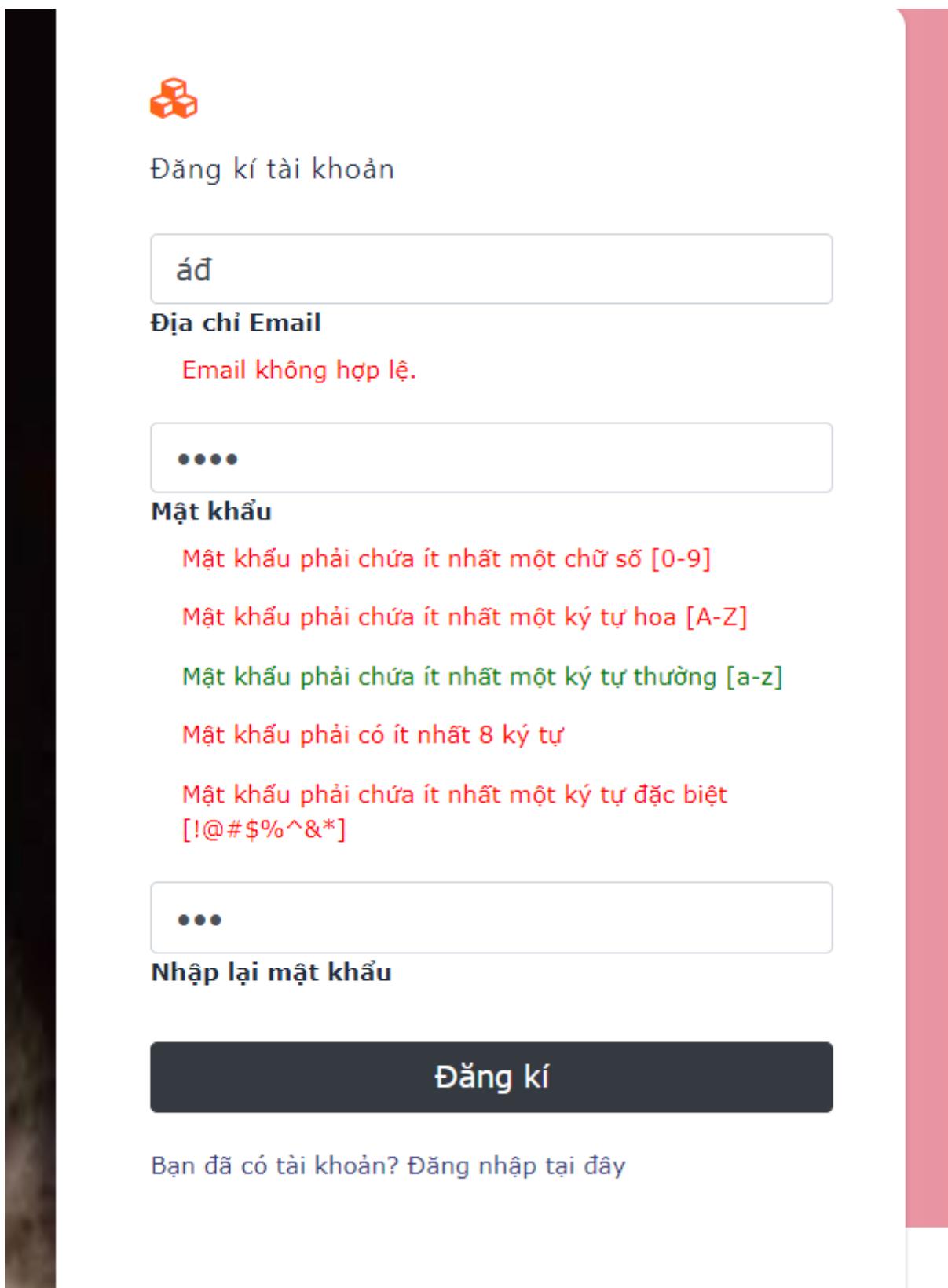
Hình 3. 31. Giao diện trang đăng ký tài khoản

Trong quá trình đăng ký hệ thống sẽ kiểm tra đầu vào như:

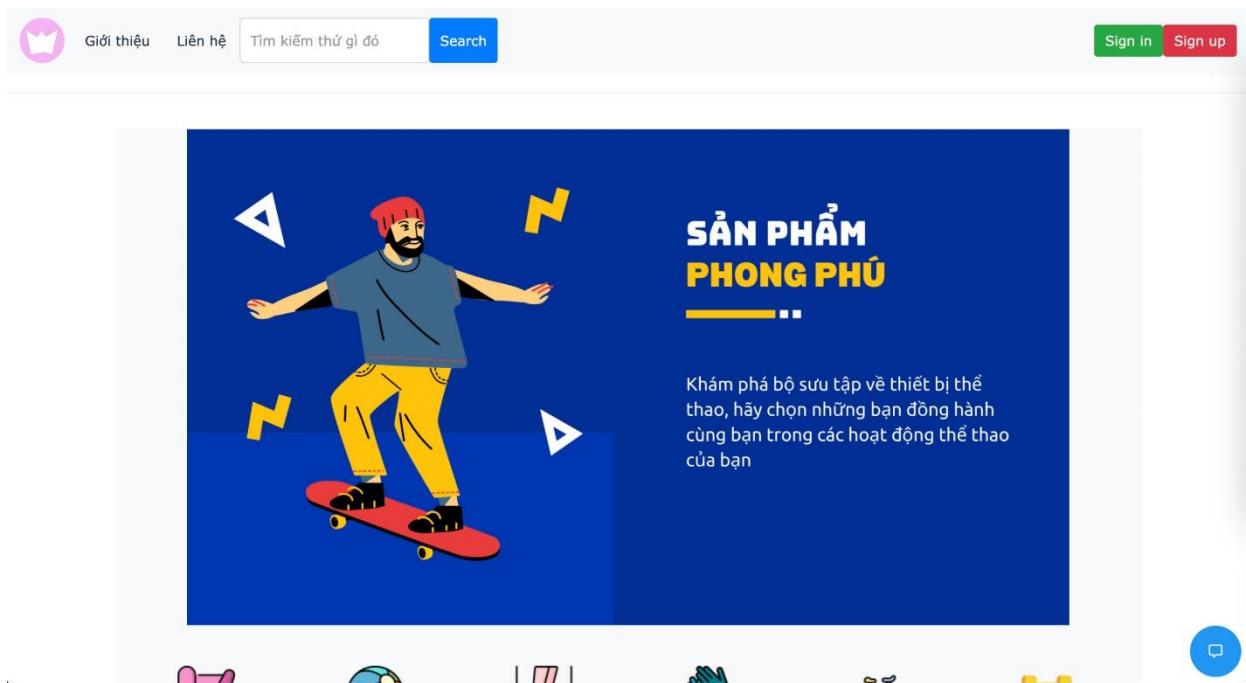
- Email có hợp lệ không
- Mật khẩu và mật khẩu nhập lại có khớp nhau hay không
- Email đã có được đăng ký trước đó hay chưa
- Mật khẩu có đảm bảo an toàn hay không
 - Mật khẩu phải chứa ít nhất 1 chữ số
 - Mật khẩu phải chứa ít nhất 1 từ viết hoa
 - Mật khẩu phải chứa ít nhất 1 từ viết thường
 - Mật khẩu phải có 8 ký tự hoặc dài hơn
 - Mật khẩu phải chứa ít nhất 1 ký tự đặc biệt là [!@#\$%^&*]

Nếu 1 trong những điều trên不合 lệ thì hệ thống sẽ thông báo lỗi ngay lập tức **Hình 3. 32**, ngược lại nếu như các thông tin đăng ký đã phù hợp thì quá trình đăng ký sẽ thành công và tự động chuyển hướng người dùng tới trang đăng nhập để người dùng thực hiện

đăng nhập tài khoản vừa đăng ký. Khi người dùng đăng nhập thành công sẽ chuyển hướng đến trang chủ như **Hình 3. 33**



Hình 3. 32. Thông báo lỗi khi thông tin đăng ký không hợp lệ

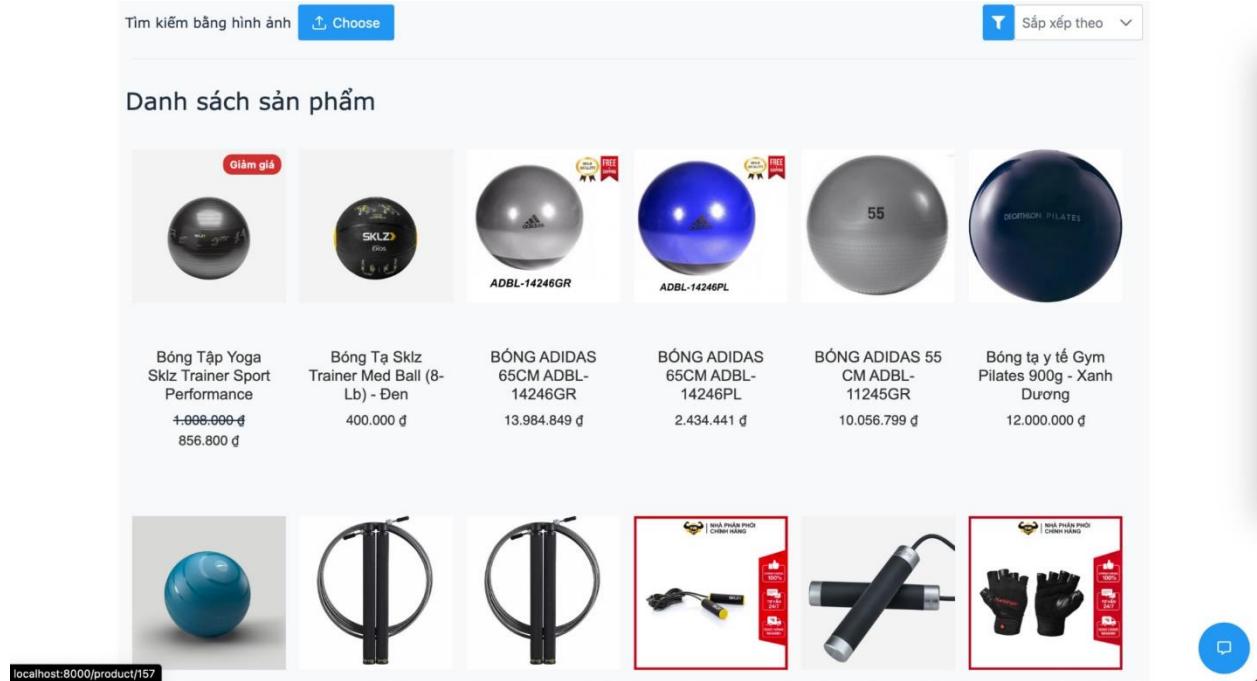


Hình 3. 33. Giao diện trang chủ của website

Tại trang chủ của website, sẽ hiển thị các danh sách của sản phẩm, chẳng hạn như danh sách các sản phẩm đang bán chạy như **Hình 3. 34** và danh sách tất cả các sản phẩm như **Hình 3. 35**

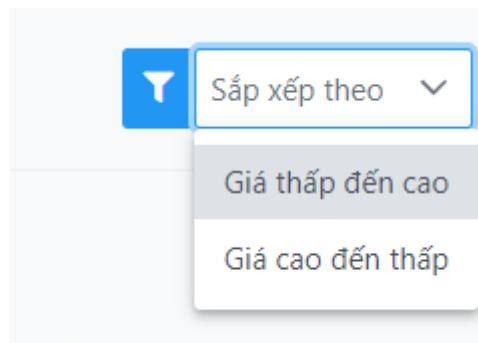
The screenshot shows a product listing page. At the top, it says 'Sản phẩm bán chạy NEW'. Below this, there are three product cards. The first card on the left shows a black exercise ball with the text 'Bóng Tập Yoga Sklz Trainer Sport Performance 4.008.000đ'. The middle card shows a treadmill with the text 'TRỢ GIÁ - MÁY CHẠY BỘ KINGSPORT KUNGFU ĐƠN NĂNG'. The third card on the right shows a blue Adidas exercise ball with the text 'BÓNG ADIDAS 65CM ADBL-14246PL \$2.434.441 đ'. Each product card includes a small image, a title, a price, and a 'FREE' badge.

Hình 3. 34. Giao diện danh sách sản phẩm đang bán chạy

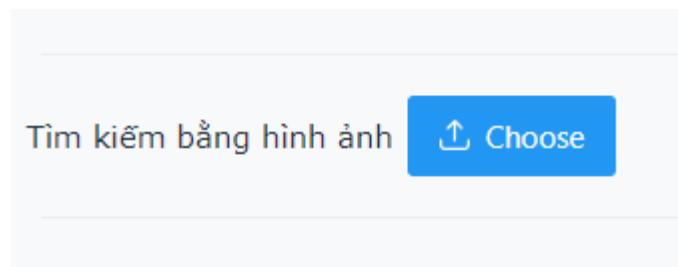


Hình 3. 35. Giao diện danh sách tất cả các sản phẩm

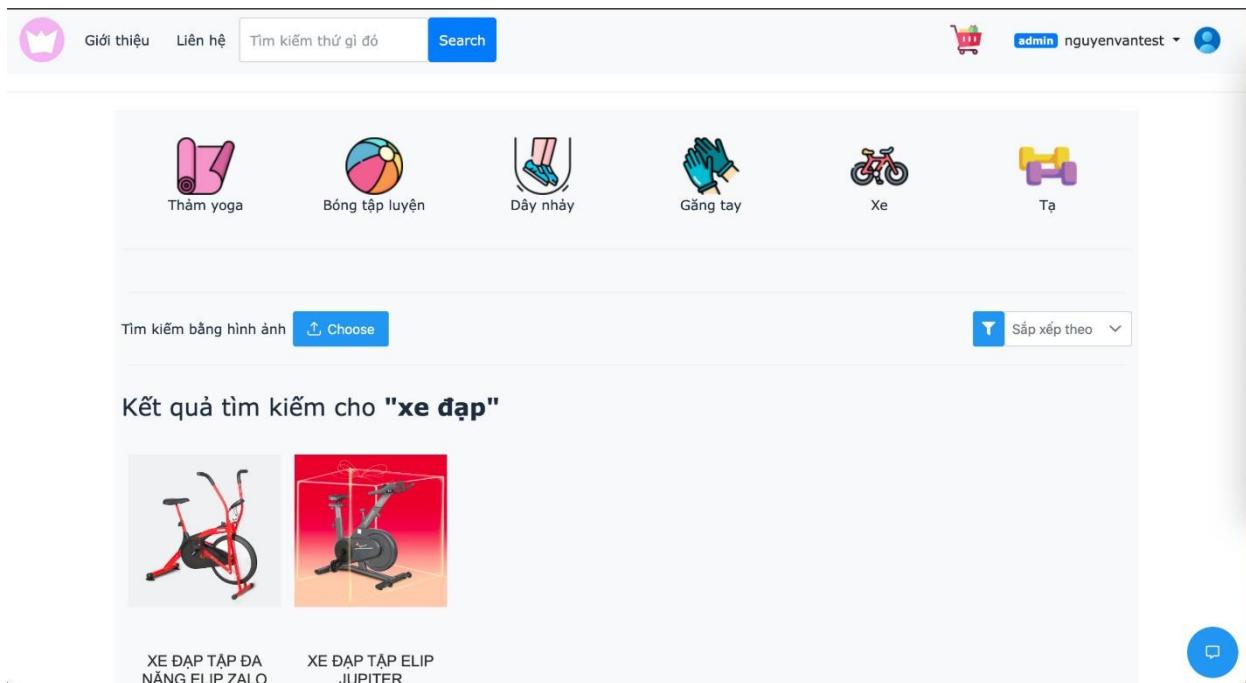
Ở phần giao diện ở **Hình 3. 35** sẽ có các tùy chọn lọc sản phẩm theo giá như **Hình 3. 36** và nút tìm kiếm bằng hình ảnh tương đồng như **Hình 3. 37** hoặc có thể tìm kiếm sản phẩm với 1 từ khóa nào đó như **Hình 3. 38**



Hình 3. 36. Bộ lọc sản phẩm theo giá

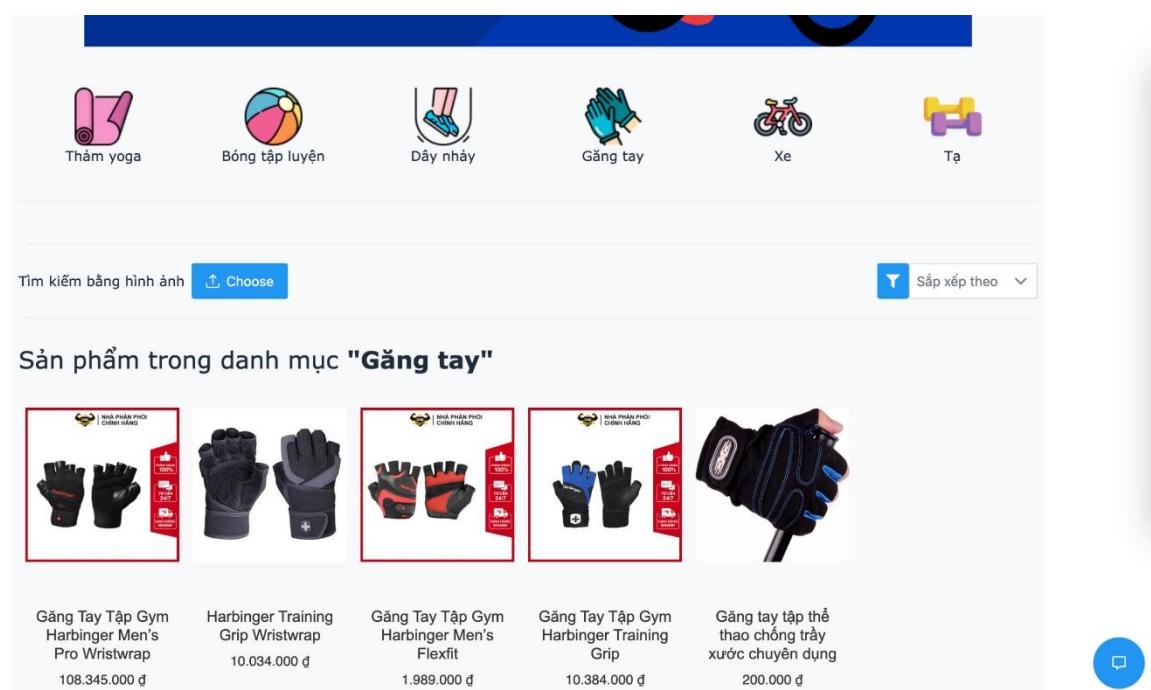


Hình 3. 37. Nút tìm tải ảnh lên để tìm kiếm sản phẩm có ảnh tương đồng



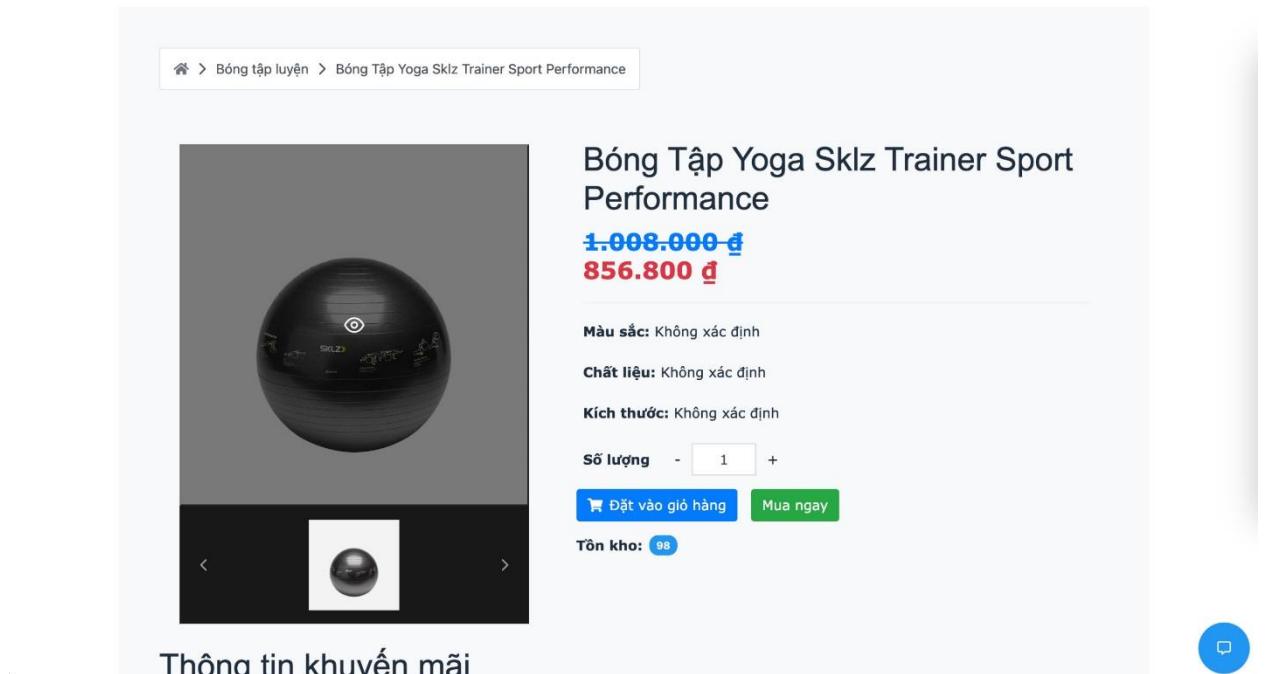
Hình 3. 38. Giao diện khi tìm kiếm sản phẩm bằng từ khóa

Nếu người dùng chọn vào 1 danh mục cụ thể lúc này sẽ có giao diện như **Hình 3. 39.**



Hình 3. 39. Giao diện khi chọn vào 1 danh mục chỉ định

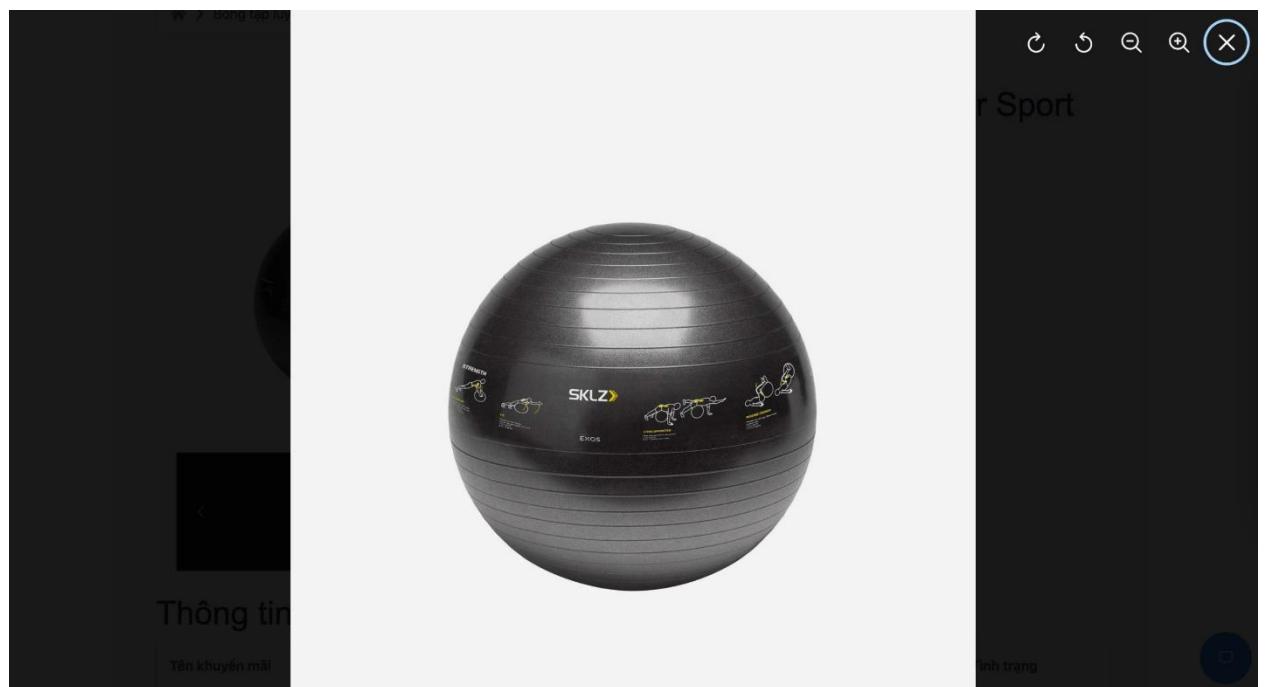
Nếu người dùng chọn vào một sản phẩm để xem chi tiết giao diện sẽ như **Hình 3. 40**



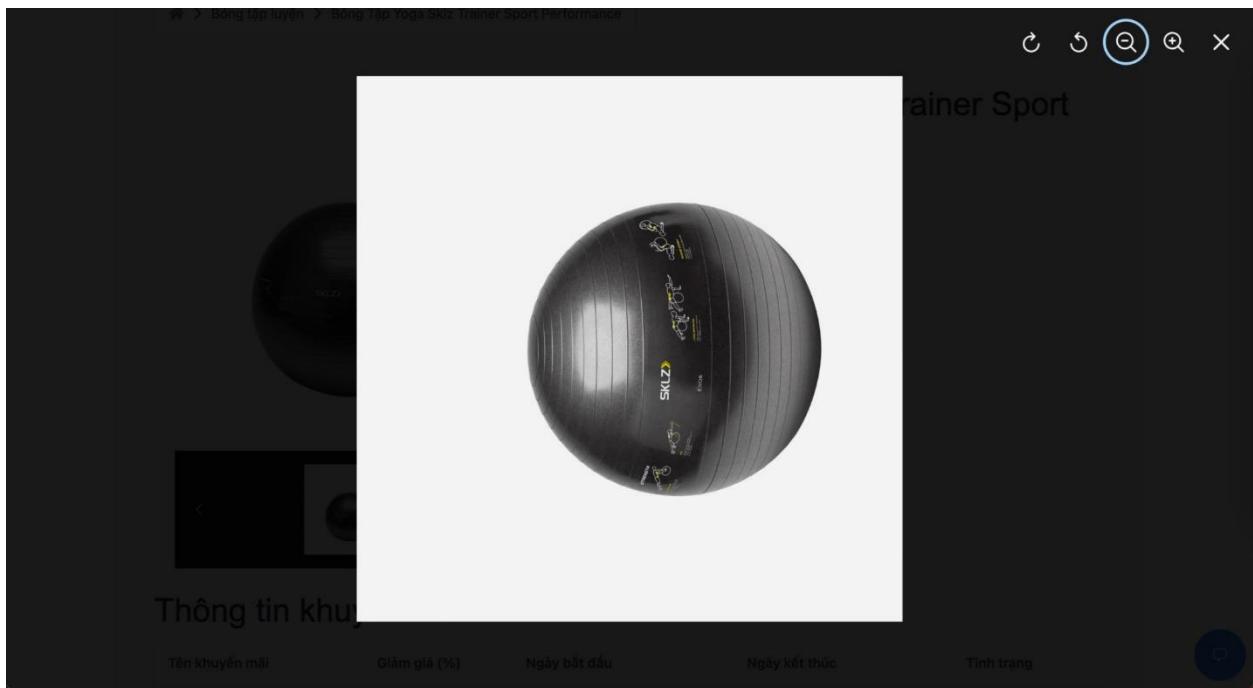
Thông tin khuyến mãi

Hình 3. 40. Giao diện khi xem một sản phẩm chi tiết

Khi người dùng nhấp vào một tấm ảnh nào đó (ảnh đại diện hoặc ảnh chi tiết sản phẩm) ở **Hình 3. 40** thì lúc này hệ thống sẽ phóng to hình ảnh lên để người dùng có thể xem chi tiết phẩm hơn và bên cạnh đó người dùng cũng có thể thực hiện các hành động như xoay hình ảnh qua trái, qua phải, phóng to, thu nhỏ ở phần menu phía góc trên bên phải như **Hình 3. 41**, **Hình 3. 42**



Hình 3. 41. Giao diện xem chi tiết một hình ảnh của sản phẩm



Hình 3.42. Giao diện xem chi tiết hình ảnh khi xoay qua phải và thu nhỏ

Tiếp theo phía dưới sẽ là chi tiết thông tin của sản phẩm và khuyến mãi của sản phẩm như **Hình 3.43**

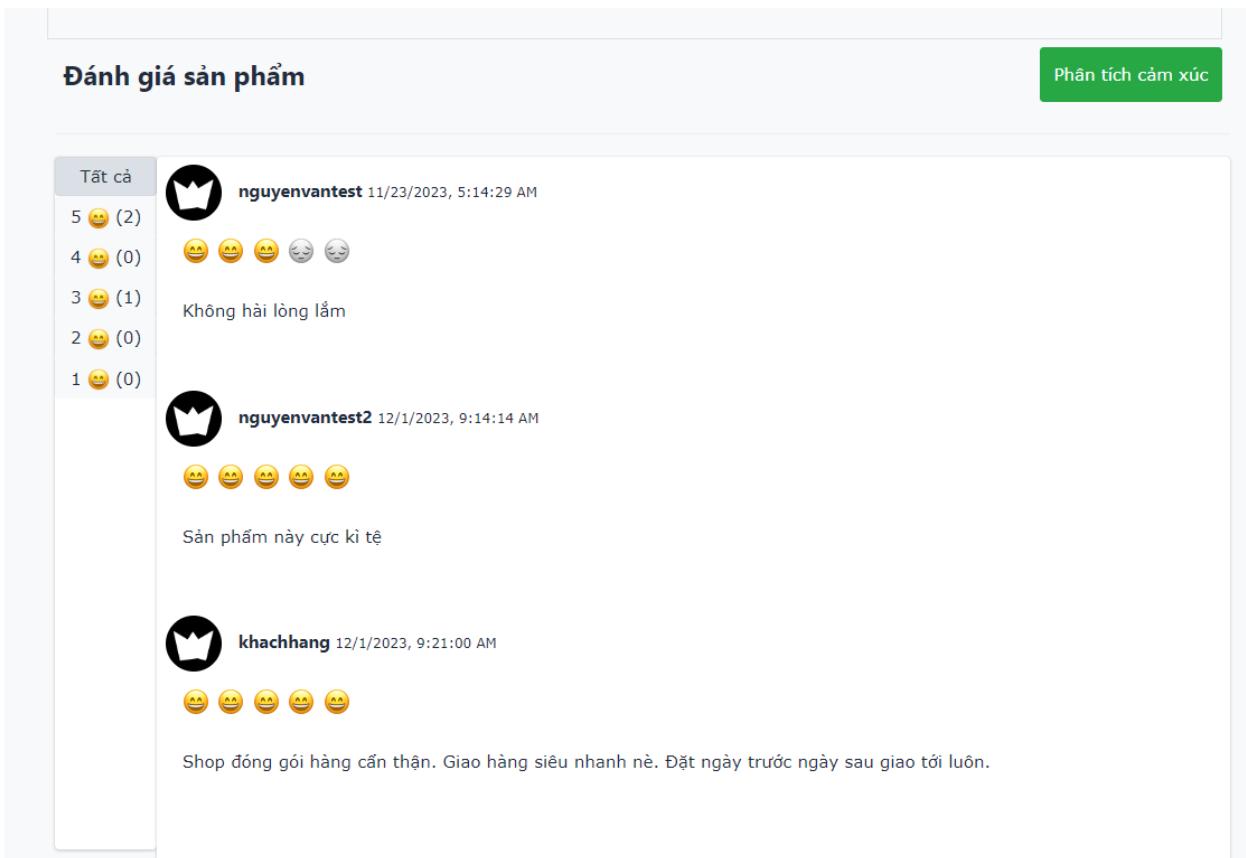
The screenshot shows the 'Thông tin khuyến mãi' (Promotional Information) section of the product detail page. It contains a table with five columns: 'Tên khuyến mãi', 'Giảm giá (%)', 'Ngày bắt đầu', 'Ngày kết thúc', and 'Tình trạng'. The table has three rows of data:

Tên khuyến mãi	Giảm giá (%)	Ngày bắt đầu	Ngày kết thúc	Tình trạng
noel 2022	10	2022-11-01 07:00:00	2022-12-31 07:00:00	Đã áp dụng
Ngày quốc tế phụ nữ	15	2024-02-29 07:00:00	2024-09-10 07:00:00	Đang áp dụng
noel	10	2024-11-22 00:00:00	2024-12-12 00:00:00	Chưa áp dụng

Below this section is a button labeled 'Thông tin sản phẩm' (Product Information). Further down the page, there is another section titled 'Thông tin sản phẩm' (Product Information) which contains a detailed description of the product: 'Bóng Tập Yoga SKLZ Trainer Sport Performance là một công cụ linh hoạt được thiết kế để nâng cao sự linh hoạt của bạn trong thực hành Yoga và cả chế độ tập luyện sức khỏe tổng thể. Được làm từ các vật liệu chất lượng cao, bóng yoga này mang lại độ bền và ổn định, làm cho nó lý tưởng cho các bài tập khác nhau, bao gồm cung cống cơ bụng, tập trung vào cân bằng và tăng cường sự linh hoạt. Bề mặt có độ săn chắc cung cấp độ bám, đảm bảo việc sử dụng an toàn và thoải mái trong suốt quá trình tập luyện. Dù bạn là người mới bắt đầu hay là một người yêu thích Yoga có kinh nghiệm, Bóng Tập Yoga SKLZ Trainer Sport Performance đều phục vụ cho mọi cấp độ sức khỏe, giúp bạn đạt được mục tiêu sức khỏe một cách hiệu quả. Hãy tích hợp nó vào chế độ tập luyện của bạn để trải nghiệm sự ổn định, sức mạnh và sự linh hoạt được cải thiện, nâng cao các buổi tập Yoga của bạn lên một tầm cao mới.'

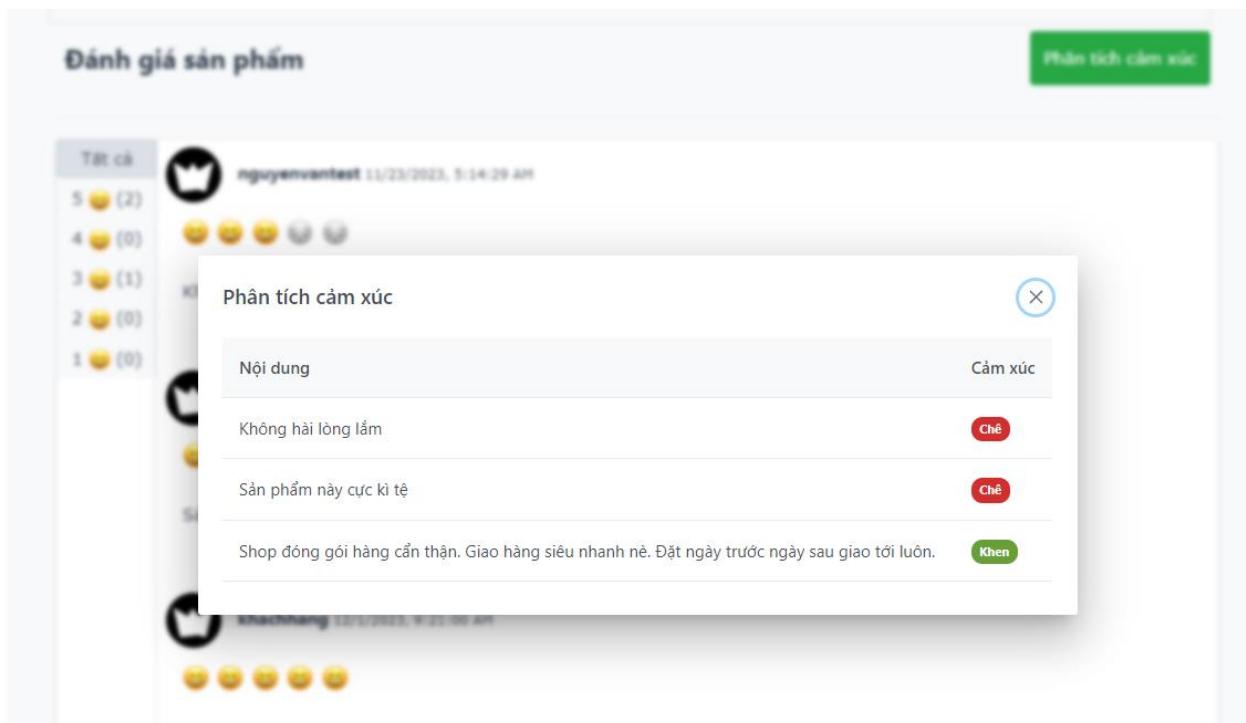
Hình 3.43. Giao diện thông tin chi tiết sản phẩm và khuyến mãi

Phía dưới phần chi tiết khuyến mãi sẽ là phần đánh giá của những người đã mua sản phẩm với các thông tin như mức độ hài lòng và nội dung đánh giá được thể hiện qua **Hình 3.44**



Hình 3. 44. Giao diện phần đánh giá sản phẩm

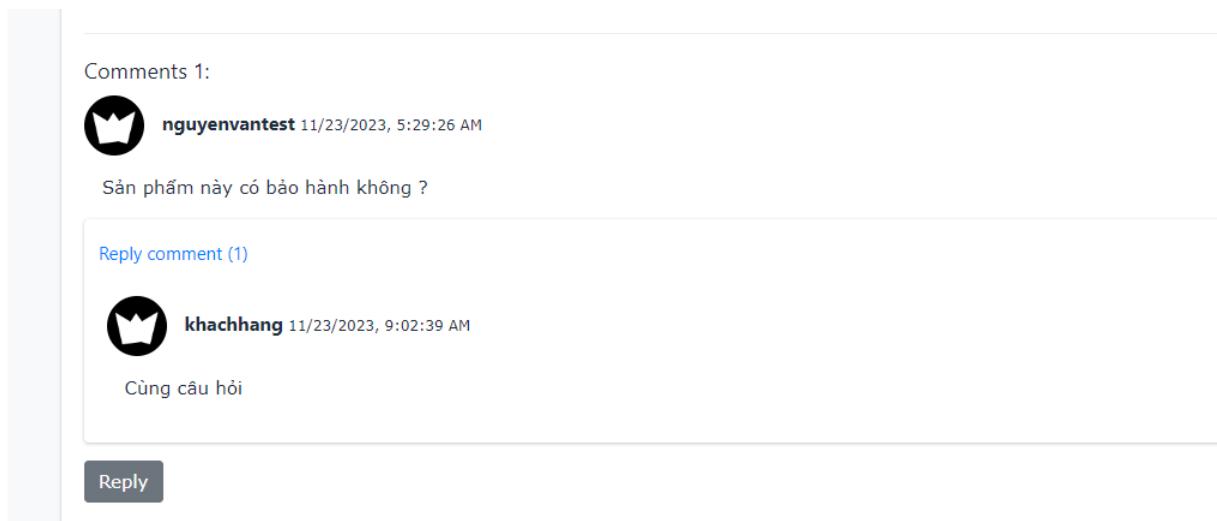
Khi nhấp vào nút “Phân tích cảm xúc”, lúc này sẽ hiển thị ra các kết quả phân tích đánh giá của người dùng một cách rõ ràng hơn như **Hình 3. 45**



Hình 3. 45. Giao diện phân tích cảm xúc đánh giá

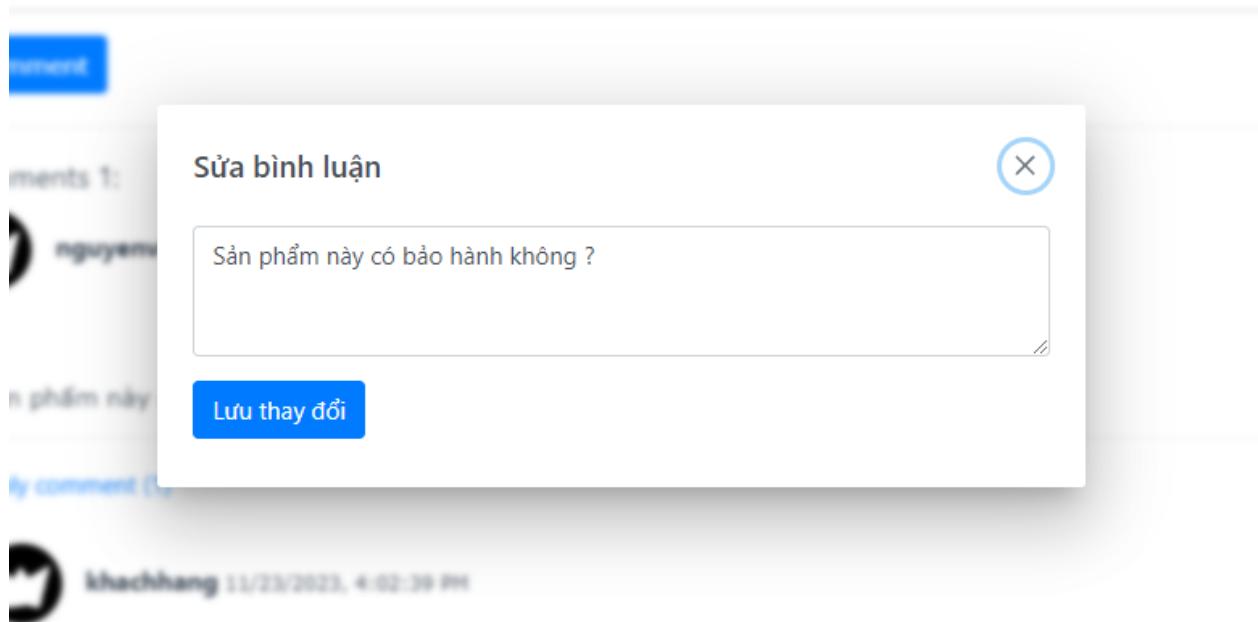
Tiếp theo, giao diện phía dưới cùng sẽ là phần bình luận của mọi người như **Hình 3.**

46



Hình 3. 46. Giao diện phần bình luận

Khi người dùng đã bình luận và muốn sửa bình luận vừa thực hiện, họ sẽ chọn vào menu chọn “Edit” để mở popup sửa đổi bình luận như **Hình 3. 47**



Hình 3. 47. Giao diện sửa đổi bình luận

Phía dưới phần bình luận sẽ là danh sách các sản phẩm có cùng danh mục như **Hình 3. 48** và các sản phẩm được gợi ý như **Hình 3. 49**

Các sản phẩm có cùng danh mục



Bóng Tập Yoga
Sklz Trainer Sport
Performance
1.008.000 đ

Bóng Tạ Sklz
Trainer Med Ball
(8-Lb) - Đen
400.000 đ

BÓNG ADIDAS
65CM ADBL-
14246GR
13.984.849 đ

BÓNG ADIDAS
65CM ADBL-
14246PL
2.434.441 đ

BÓNG ADIDAS
55 CM ADBL-
11245GR
10.056.799 đ

Bóng tạ y tế Gym
Pilates 900g -
Xanh Dương
12.000.000 đ

Hình 3. 48. Giao diện các sản phẩm cùng danh mục

Các sản phẩm được gợi ý



GHẾ TẠ ĐA
NĂNG
KINGSPORT BK-
599
9.000.000 đ

XE ĐẠP TẬP
ELIP JUPITER
1.000.000 đ

GIÀN TẠ ĐA
NĂNG
KINGSPORT BK-
1999 NEW
3.000.000 đ

TRỢ GIÁ - MÁY
CHẠY BỘ
KINGSPORT
KUNGFU ĐƠN
NĂNG
4.000.000 đ

GIÀN TẠ ĐA
NĂNG
KINGSPORT BK-
1998 NEW
120.000.000 đ

XE ĐẠP TẬP ĐA
NĂNG ELIP
ZALO
43.000.000 đ

Hình 3. 49. Giao diện các sản phẩm được gợi ý

Tiếp theo sẽ là giao diện tìm kiếm hình ảnh tương tự dựa vào một hình ảnh của các hình ảnh có liên quan như **Hình 3. 50**

Nhấn vào hình ảnh để tìm kiếm hình ảnh tương tự



Kết quả ảnh tương tự đã chọn:



Bóng tạ y tế Gym
Pilates 900g -
Xanh Dương
12.000.000 ₫

BÓNG ADIDAS
55 CM ADBL-
11245GR
10.056.799 ₫

BÓNG ADIDAS
65CM ADBL-
14246GR
13.984.849 ₫

Bóng Tạ Sklz
Trainer Med Ball
(8-Lb) - Đen
400.000 ₫

Hình 3. 50. Giao diện tìm ảnh tương tự khi nhấn vào các sản phẩm có liên quan
Phần footer của các trang sẽ có dạng như **Hình 3. 51**

Về chúng tôi

Chúng tôi cung cấp các thiết bị thể thao chất lượng cao với giá cả hợp lý. Chúng tôi cam kết mang lại sự hài lòng cho khách hàng.

localhost:8000/product/159

Liên hệ

Địa chỉ: Số 123, đường ABC, quận XYZ, TP. Cần Thơ

Điện thoại: 0123456789

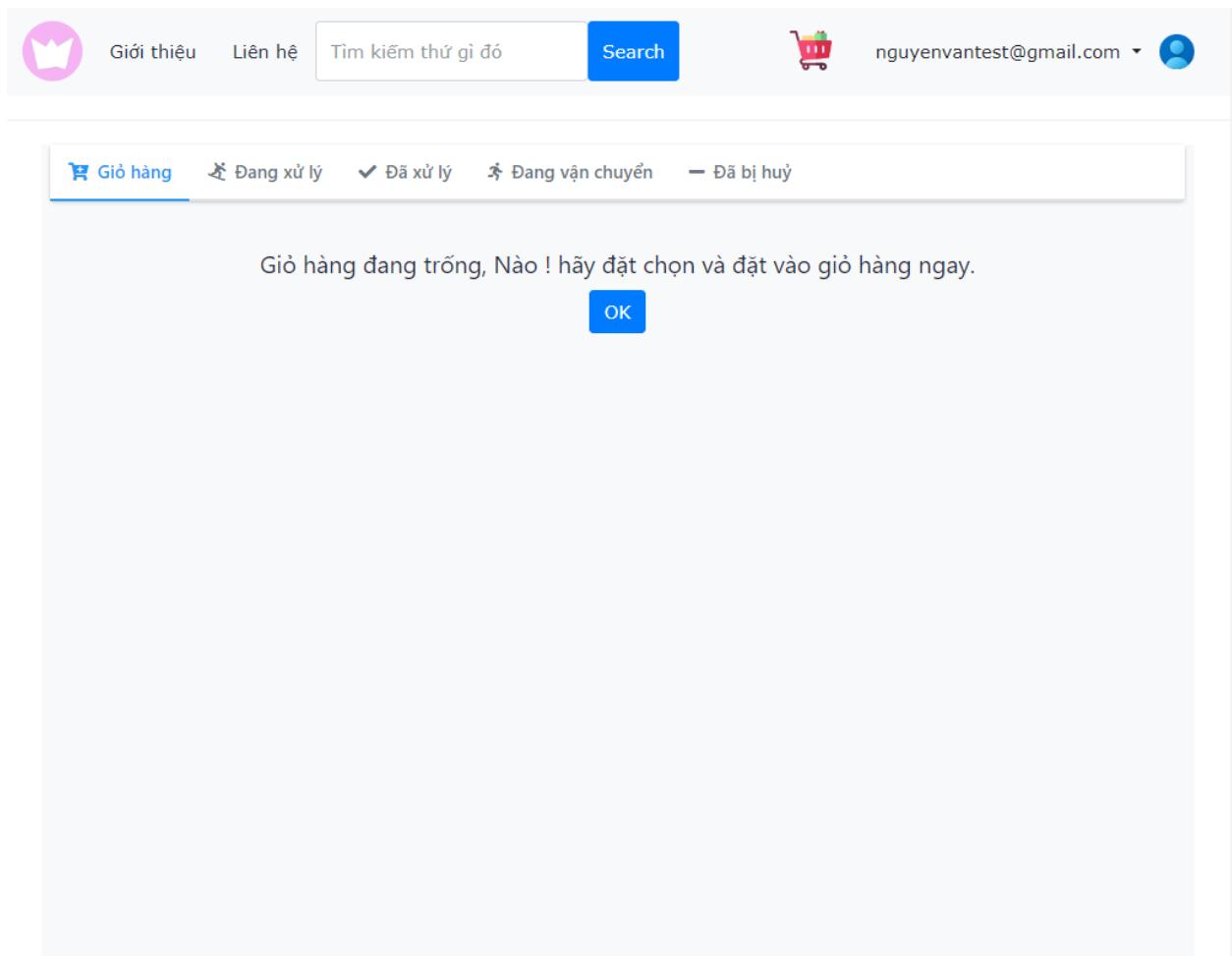
Email: info@example.com

Theo dõi chúng tôi

Facebook
 Twitter
 Instagram

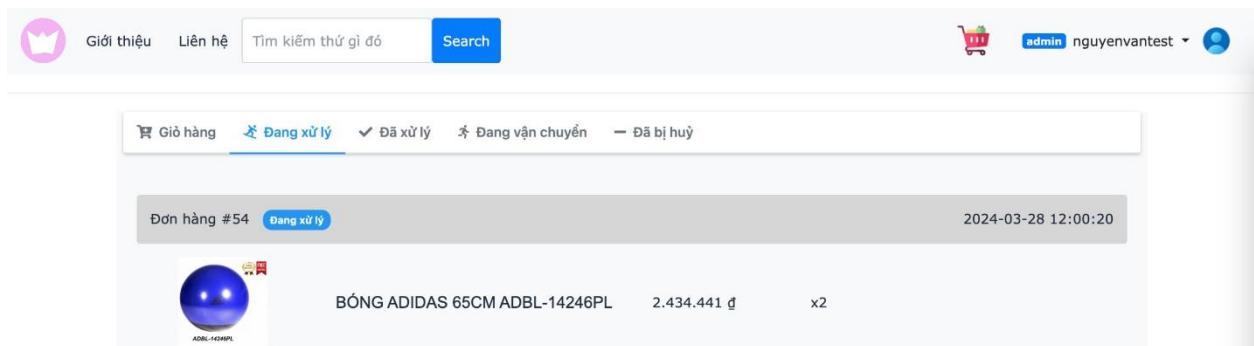
Hình 3. 51. Giao diện phần footer của trang web

Giả sử khi người dùng đã chọn được một sản phẩm yêu thích, lúc này họ có thể chọn sản phẩm đó vào giỏ hàng và sau đây là giao diện của trang giỏ hàng như **Hình 3. 52**



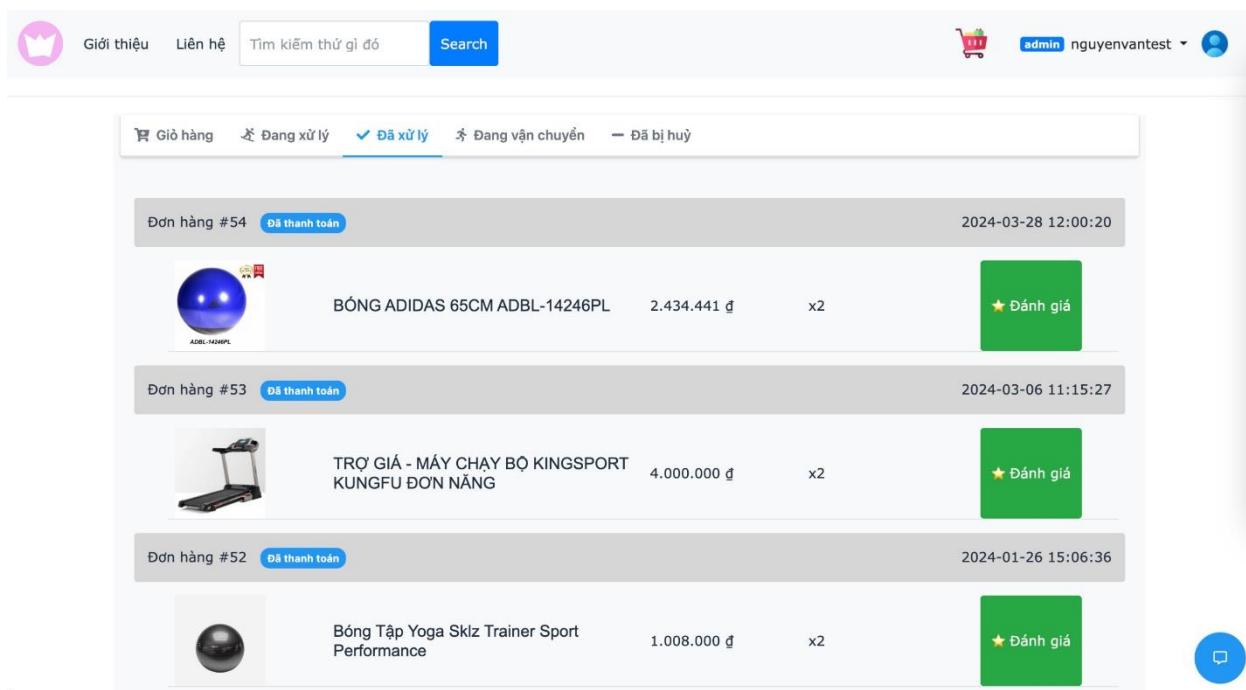
Hình 3. 52. Giao diện trang giỏ hàng

Tiếp theo khi giỏ hàng đã được thanh toán thì sẽ nằm ở trang giỏ hàng đang xử lý như **Hình 3. 53**



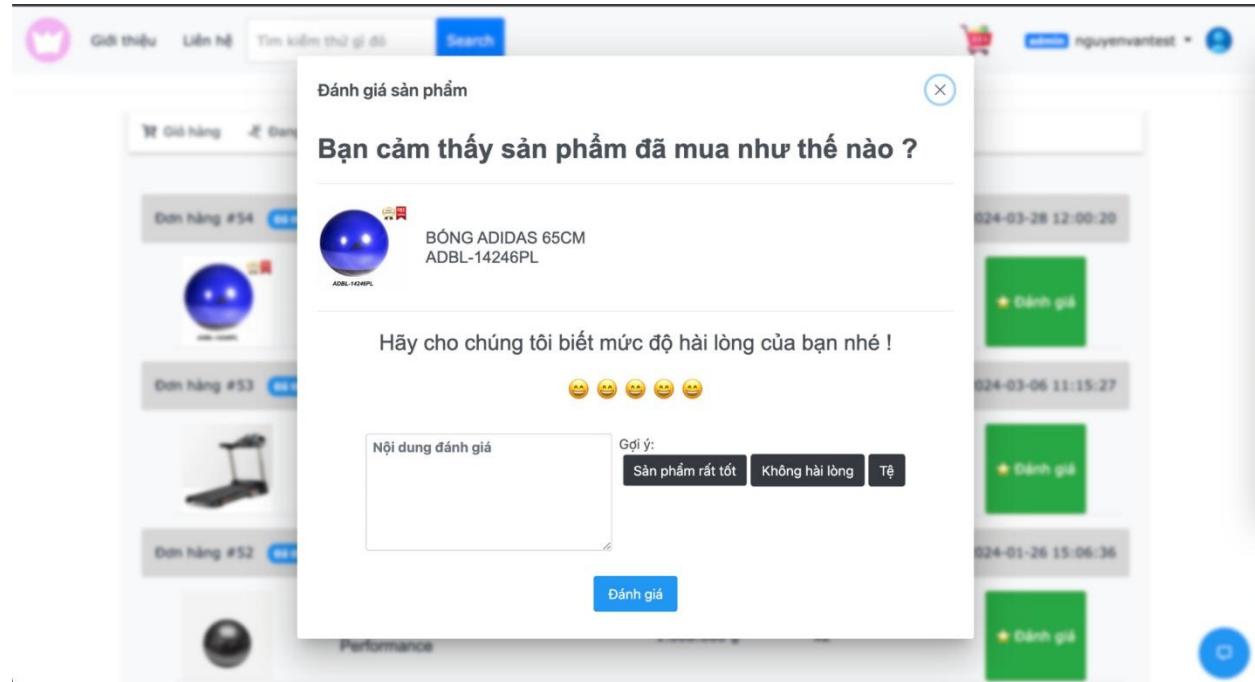
Hình 3. 53. Giao diện trang giỏ hàng đang xử lí

Kế tiếp sẽ là tab “Đã xử lý”, tại đây sẽ hiển thị danh sách các giỏ hàng đã xử lý như **Hình 3. 54**



Hình 3. 54. Giao diện danh sách các giò hàng đã được xử lý và thanh toán

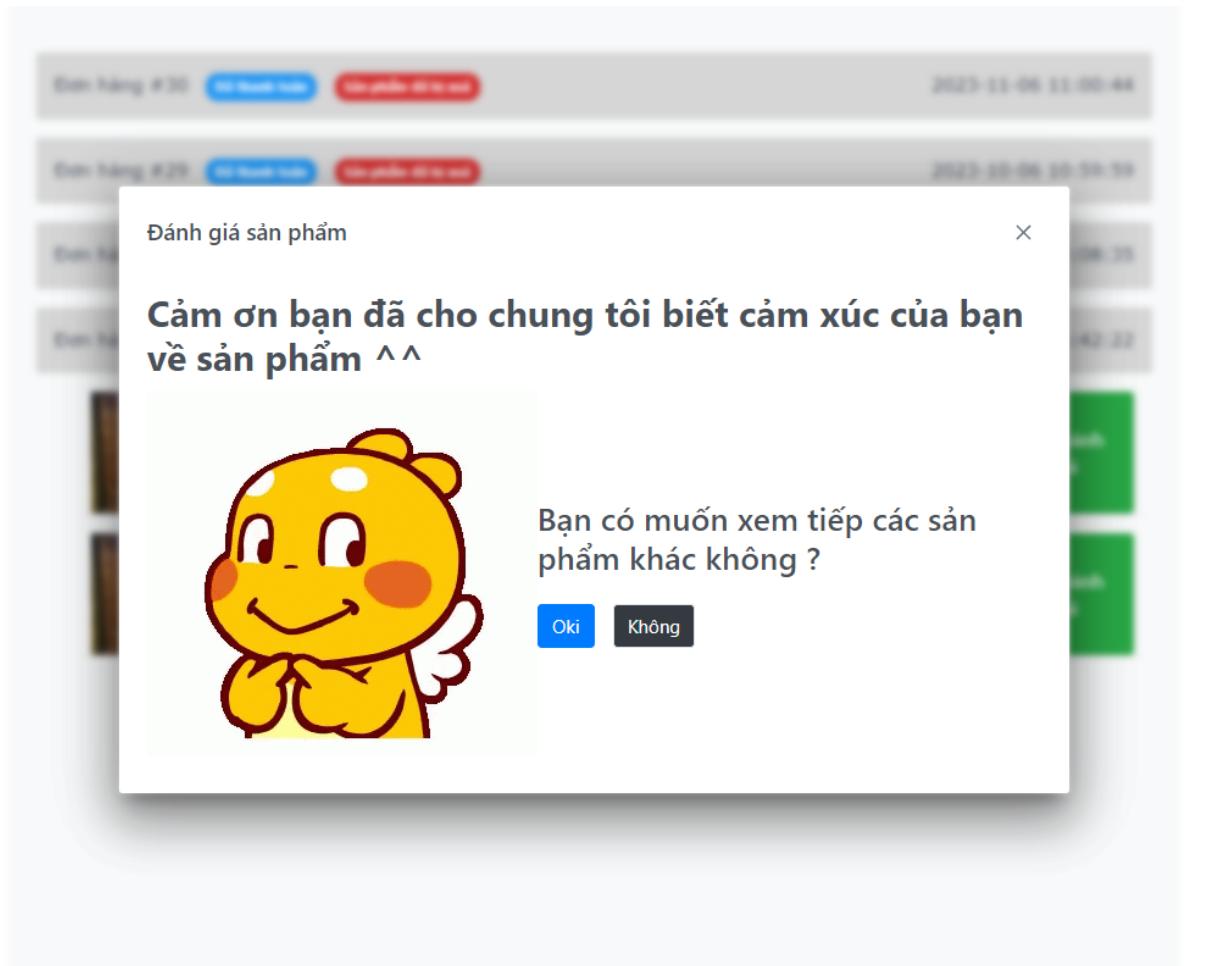
Ở phần tab “Đã xử lý” như **Hình 3. 54** hiển thị danh sách các đơn hàng đã được xử lý và giao hàng thành công đến khách hàng, lúc này khách hàng có thể nhấn vào nút “Đánh giá” để có thể thực hiện đánh giá mức độ hài lòng của phẩm thông qua các biểu tượng cảm xúc và để lại nội dung của đánh giá như **Hình 3. 55**



Hình 3. 55. Giao diện phản đánh giá mức độ hài lòng cho sản phẩm đã mua

Khi người dùng điền đầy đủ thông tin về phản đánh giá và nhấn vào nút đánh giá, lúc này hệ thống sẽ thực hiện quá trình đánh giá và trả về kết quả cho người dùng, nếu như mà quá trình đánh giá diễn ra thành công thì người dùng sẽ được chuyển sang giao diện cảm ơn như **Hình 3. 56**, lúc này hệ thống sẽ hỏi người dùng muốn tiếp tục mua sắm hay không,

nếu người dùng nhấn vào nút “Oki” thì sẽ điều hướng người dùng tới trang chủ để người dùng tiếp tục mua sắm và nếu người dùng nhấn vào nút “Không” thì sẽ tắt đi popup đánh giá và giữ nguyên người dùng ở trang giỏ hàng hiện tại.



Hình 3. 56. Giao diện cảm ơn khách hàng đã đánh giá sản phẩm

Còn ở tab “Đang vận chuyển” sẽ hiển thị danh sách các giỏ hàng đã được xác nhận đơn và đang trong quá trình vận chuyển đến khách hàng được thể hiện qua **Hình 3. 57**



Hình 3. 57. Giao diện danh sách đơn hàng đang được vận chuyển

Cuối cùng sẽ là tab “Đã bị huỷ”, tại đây sẽ hiển thị danh sách các đơn hàng đã bị huỷ như **Hình 3. 58**

Hình 3. 58. Giao diện danh sách các đơn hàng đã bị huỷ

Ở giao diện các danh sách ở giỏ hàng, người dùng có thể chọn vào mã 1 đơn hàng bất kì lúc này hệ thống sẽ chuyển sang giao diện chi tiết hóa đơn đã lập như **Hình 3. 59**

Hình 3. 59. Giao diện hóa đơn của 1 đơn hàng

Trở lại giao diện giỏ hàng ở **Hình 3. 52**, tại đây người dùng có thể sẽ chọn vào nút thanh toán các giỏ hàng đã thêm, lúc này hệ thống sẽ đưa người dùng đến giao diện như **Hình 3. 60**

The screenshot shows a payment page with the following sections:

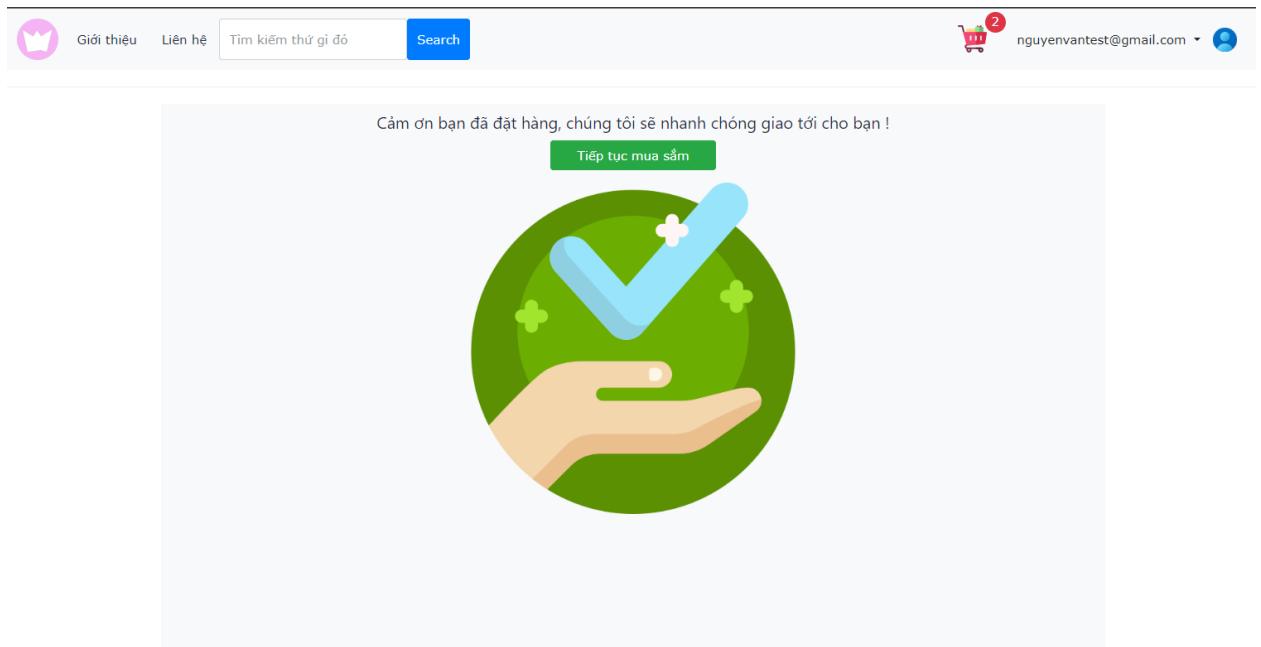
- Thông tin thanh toán** (Payment Information):
 - Full Name:
 - Address:
 - Số điện thoại:
 - Ghi chú (nếu có):
- Phương thức thanh toán** (Payment Methods):
 - Thanh toán khi nhận hàng
 - Thanh toán bằng chuyển khoản [Sắp ra mắt](#)
- Phương thức vận chuyển** (Shipping Method):
 - Giao hàng nhanh
- Các sản phẩm:**

Tên sản phẩm	Màu sắc	Chất liệu	Kích thước	Số lượng	Tổng tiền
Bóng Tập Yoga Sklz Trainer Sport Performance				1	1.008.000 ₫ 856.800 ₫

- Tổng tiền đơn hàng**: **1.008.000 ₫**
- Tổng tiền đơn hàng sau khi áp dụng khuyến mãi **856.800 ₫**
- VAT **8%**
- Tổng phí VAT **80.640 ₫**
- Tổng tiền thanh toán **937.440 ₫**
- [Đặt hàng](#)

Hình 3. 60. Giao diện trang thanh toán

Khi người dùng điền đầy đủ các thông tin thanh toán và nhấn vào nút “Đặt hàng”, lúc này hệ thống sẽ xử lý các thông tin đó để tạo hóa đơn cho khách hàng, nếu không có bất kỳ lỗi nào xảy ra sẽ đưa người dùng đến trang hoàn tất thanh toán như **Hình 3. 61**, tại đây nếu người dùng chọn vào nút “Tiếp tục mua sắm”, hệ thống sẽ chuyển hướng người dùng tới trang chủ để cho phép người dùng tiếp tục mua sắm.



Hình 3. 61. Giao diện thông báo hoàn tất thanh toán

Người dùng cũng có thể cập nhật các thông tin cá nhân của mình như email, họ tên, số điện thoại, địa chỉ như **Hình 3. 62**

Thông tin cá nhân

Email
nguyenvantest@gmail.com

Họ tên
Nguyễn Văn Test

Số điện thoại
0123456789

Địa chỉ
Tân Châu - An Giang

Lưu

Hình 3. 62. Giao diện cập nhật thông tin cá nhân của người dùng
Giao diện trang giới thiệu thông tin website được thể hiện ở **Hình 3. 63**

LỊCH SỬ HÌNH THÀNH.

Ra đời từ ý tưởng tạo nên sự khác biệt, LazyShop đã giữ vững và phát triển trở thành vị trí hàng đầu trong thị trường thể thao Việt Nam. Đến nay, LazyShop đã có nhiều cửa hàng quy mô và chuyên nghiệp tại các thành phố lớn là Hà Nội và TP.HCM, Cần Thơ.

2023: Khẳng định thương hiệu bền vững với những bước phát triển mới dù đổi mới những khía cạnh. Hệ thống cửa hàng mở rộng, với showroom mới tại Cần Thơ.

2022: Đây là năm mang dấu ấn đặc biệt của thương hiệu LazyShop – Đánh dấu chặng đường phát triển 4 năm. Với sự xuất hiện của 2 cửa hàng tại Võ Văn Kiệt (Cần Thơ) và Trung tâm thể thao Cái Khê (Cái Khê, Cần Thơ)

2021: Khẳng định thương hiệu với sự xuất hiện của 2 cửa hàng mới tại ngã tư trung tâm Cần Thơ (CMT8-Nguyễn Văn Cừ).

Hình 3. 63. Giao diện trang giới thiệu

Giao diện trang liên hệ của website sẽ như **Hình 3. 64, Hình 3. 65**

The screenshot shows a website's contact page. At the top, there is a navigation bar with a logo (a pink crown icon), links for "Giới thiệu" and "Liên hệ", a search bar containing the placeholder "Tìm kiếm thứ gì đó", and two buttons: "Search" (blue) and "Sign in" (green) and "Sign up" (red). Below the navigation, the page title "LIÊN HỆ" is displayed in a large, bold, black font. A sub-headline "Luôn lắng nghe ý kiến từ khách hàng" (Always listen to customer feedback) follows. A section titled "THÔNG TIN LIÊN HỆ" (Contact Information) includes a location pin icon followed by the address "Phường Xuân Khách, Quận Ninh Kiều, TP Cần Thơ" and a phone icon followed by the number "0342 123 123". The main visual element is a large, high-quality photograph of a modern living room. It features a light-colored sofa with several pillows (grey, white, and yellow), a wooden coffee table, and a side table holding a potted plant. In the background, there are abstract wall art pieces and a window with white curtains. An inset image in the bottom-left corner shows a close-up of a wooden cabinet door.

Hình 3. 64. Giao diện trang liên hệ 1



GỬI TIN NHẮN CHO CHÚNG TÔI

Họ tên (*)

Số điện thoại (*)

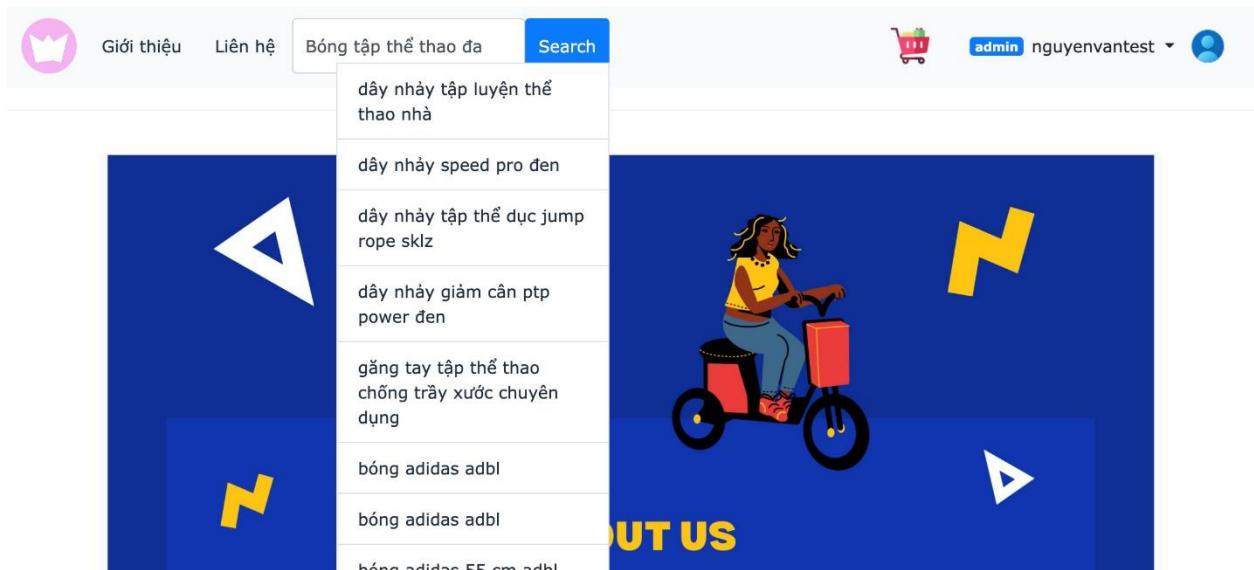
Email

Nội dung

GỬI TIN NHẮN

Hình 3. 65. Giao diện trang liên hệ 2

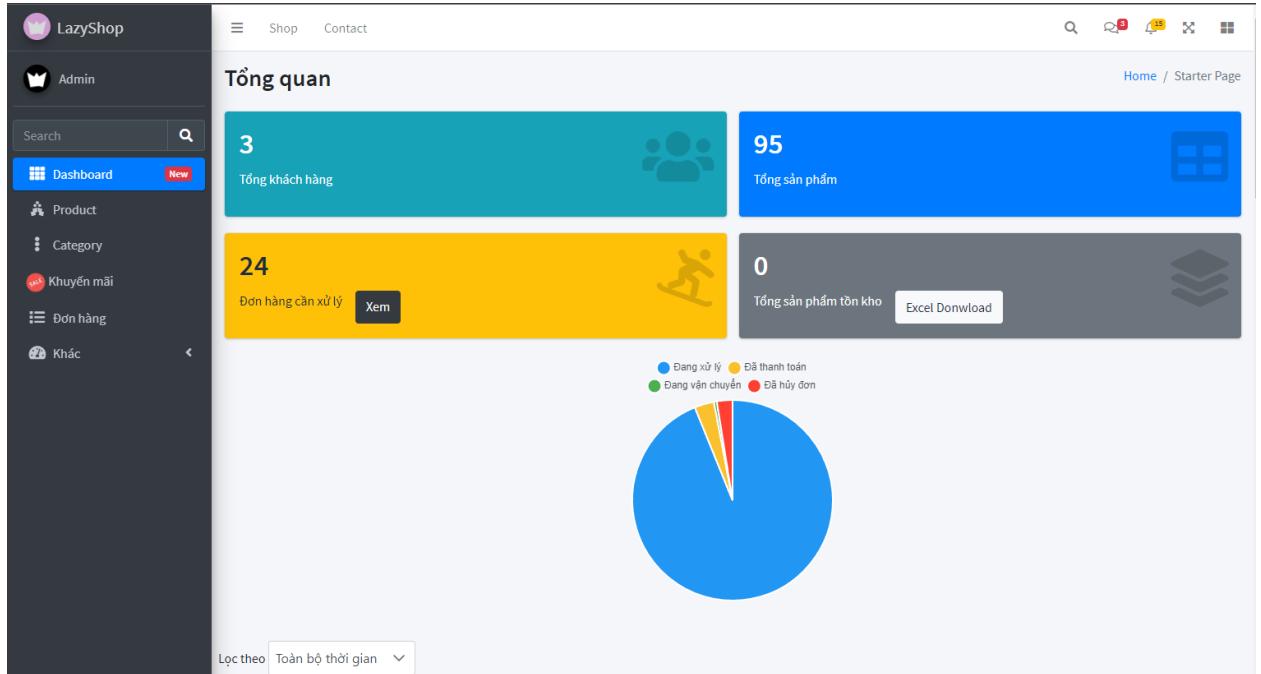
Người dùng có thể tìm kiếm và hệ thống sẽ gợi ý cho người dùng dựa vào những từ khoá đã nhập vào như **Hình 3. 66**



Hình 3. 66. Giao diện gợi ý khi người dùng tìm kiếm

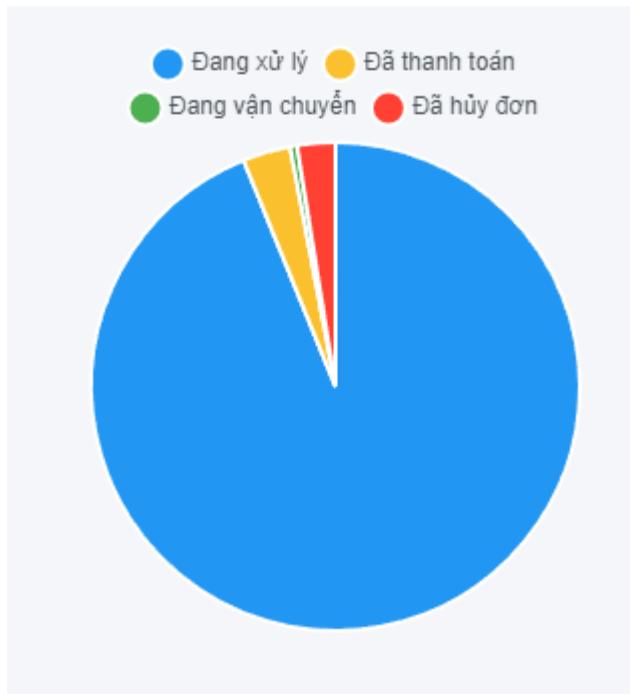
3.3.2. Giao diện quản trị

Ở giao diện quản trị của admin hoặc người bán hàng thì sẽ có giao diện dashboard trông như **Hình 3. 67**

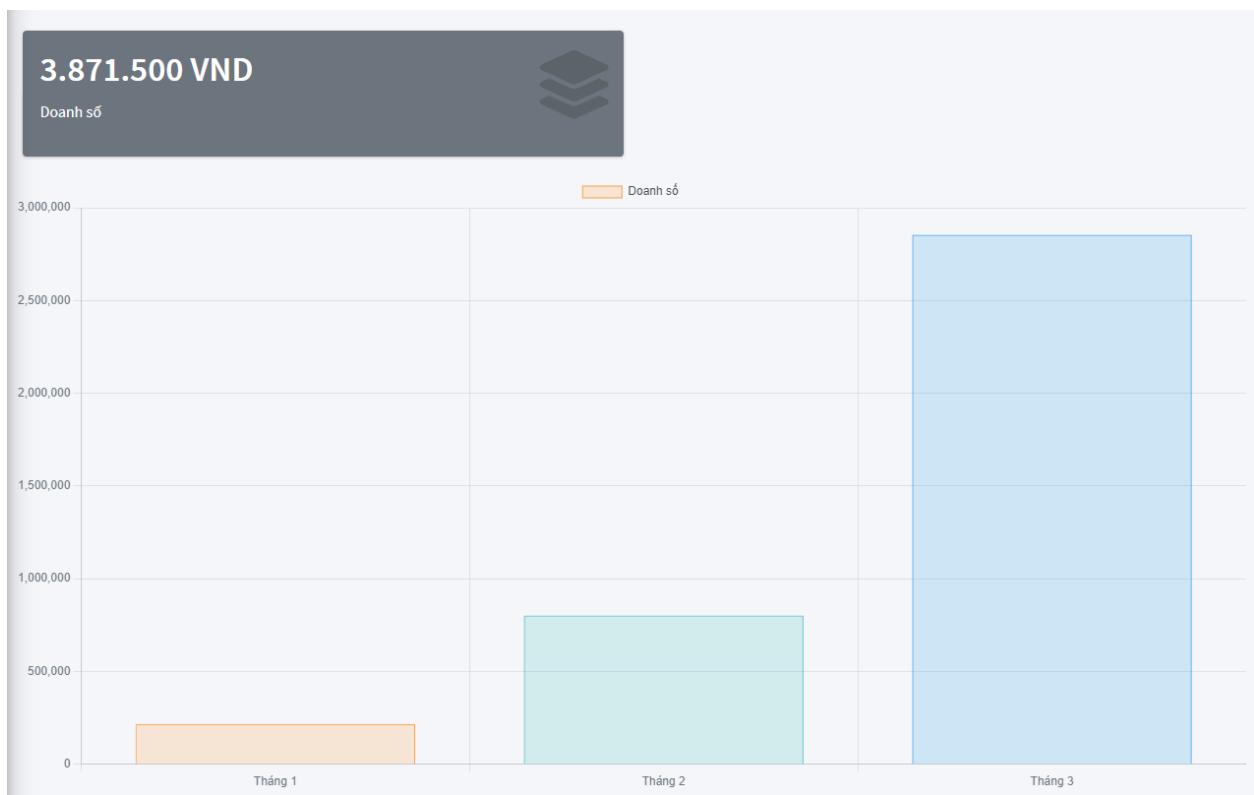


Hình 3. 67. Giao diện Dashboard của hệ thống

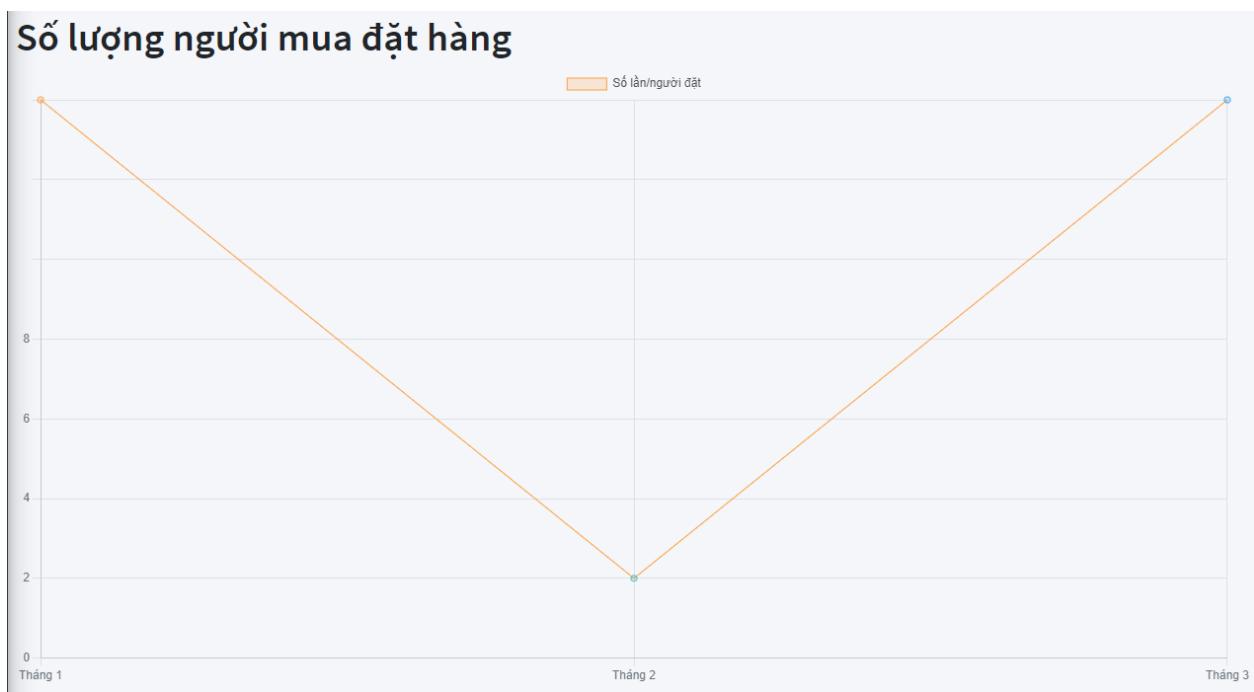
Tại trang dashboard này sẽ hiển thị các thông tin và thống kê trong toàn hệ thống như thống kê tình trạng, số lượng các đơn hàng như **Hình 3. 68**, thống kê doanh thu như **Hình 3. 69**, thống kê lượng người mua sản phẩm theo thời gian như **Hình 3. 70**, thống kê Top các sản phẩm được mua gần đây như **Hình 3. 71**



Hình 3. 68. Biểu đồ thống kê tình trạng, số lượng các đơn hàng.



Hình 3. 69. Biểu đồ thống kê doanh thu của hệ thống theo thời gian



Hình 3. 70. Biểu đồ thống kê lượng người mua sản phẩm theo thời gian

Top các sản phẩm được mua gần đây

Tên sản phẩm	Mã sản phẩm
1 BÓNG ADIDAS 65CM ADBL-14246PL	#156
2 TRỢ GIÁ - MÁY CHẠY BỘ KINGSPORT KUNGFU ĐƠN NĂNG	#137
3 Bóng Tập Yoga Sklz Trainer Sport Performance	#159

Hình 3. 71. Biểu đồ thống kê Top các sản phẩm được mua gần đây

Giao diện quản lí các sản phẩm của hệ thống như **Hình 3. 72**

The screenshot shows the LazyShop admin dashboard under the 'Product' section. On the left sidebar, there are links for Category, Khuyến mãi (Promotions), Đơn hàng (Orders), Khách hàng (Customers), Người bán (Sellers), and Khác (Others). The main area displays a table titled 'Danh sách các sản phẩm:' (List of products) with the following columns: Code ↑↓, Ảnh sản phẩm (Product image), Tên sản phẩm (Product name), Màu sắc (Color), Chất liệu (Material), Mô tả (Description), Giá ↑↓, Danh mục ↑↓ (Category), and Hành động (Actions). Two products are listed:

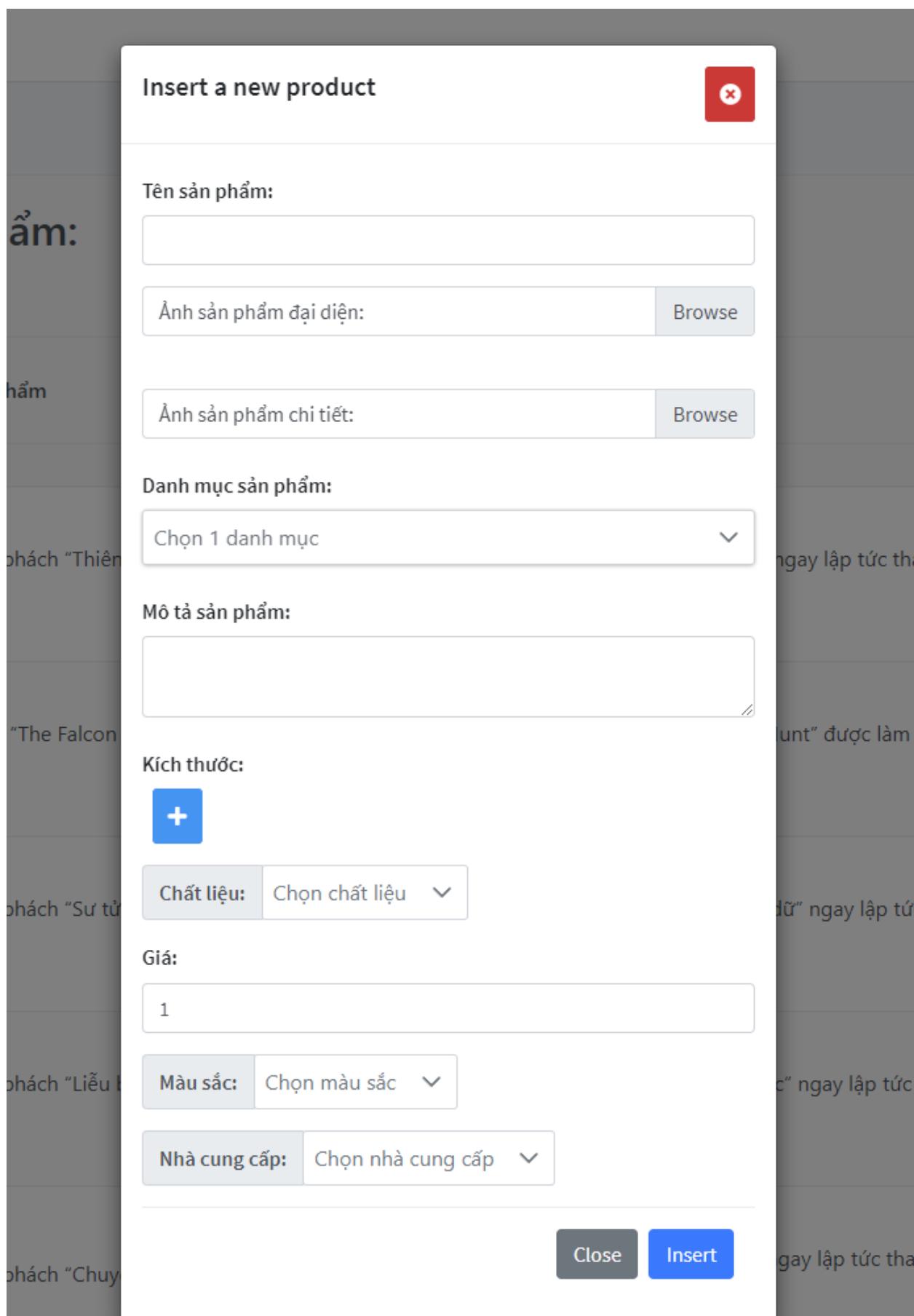
Code ↑↓	Ảnh sản phẩm	Tên sản phẩm	Màu sắc	Chất liệu	Mô tả	Giá ↑↓	Danh mục ↑↓	Hành động
159		Bóng Tập Yoga Sklz Trainer Sport Performance			Bóng Tập Yoga SKLZ Trainer Sport Performance là một công cụ linh	1.008.000 đ	Bóng tập luyện	<button>Update</button> <button>Import</button> <button>Sale-off</button> <button>Delete</button> <button>Price History</button>
158		Bóng Tạ Sklz Trainer Med Ball 10 kg	kjk			400.000 đ	Bóng tập luyện	<button>Update</button> <button>Import</button> <button>Sale-off</button>

Hình 3. 72. Giao diện quản lí các sản phẩm đang có trên hệ thống

Tại đây có các nhiều tùy chọn hành động cho các sản phẩm ở các nút trong **Hình 3.**

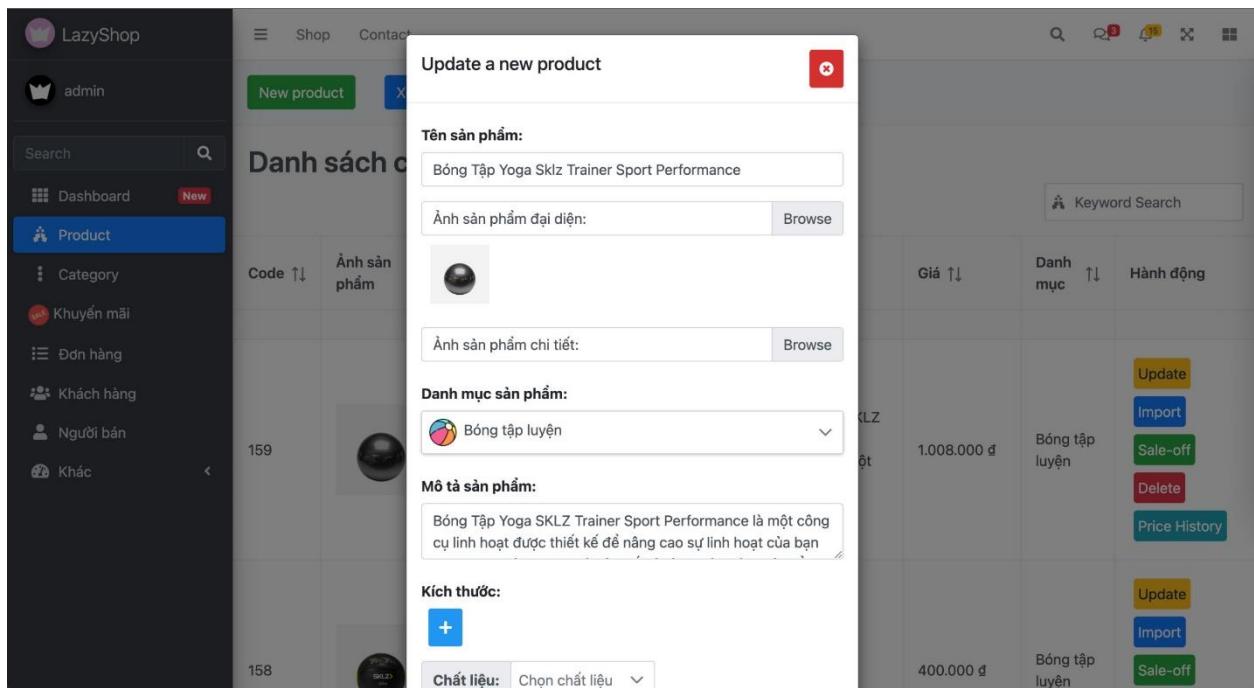
72

Giao diện thêm mới sản phẩm như **Hình 3. 73.**

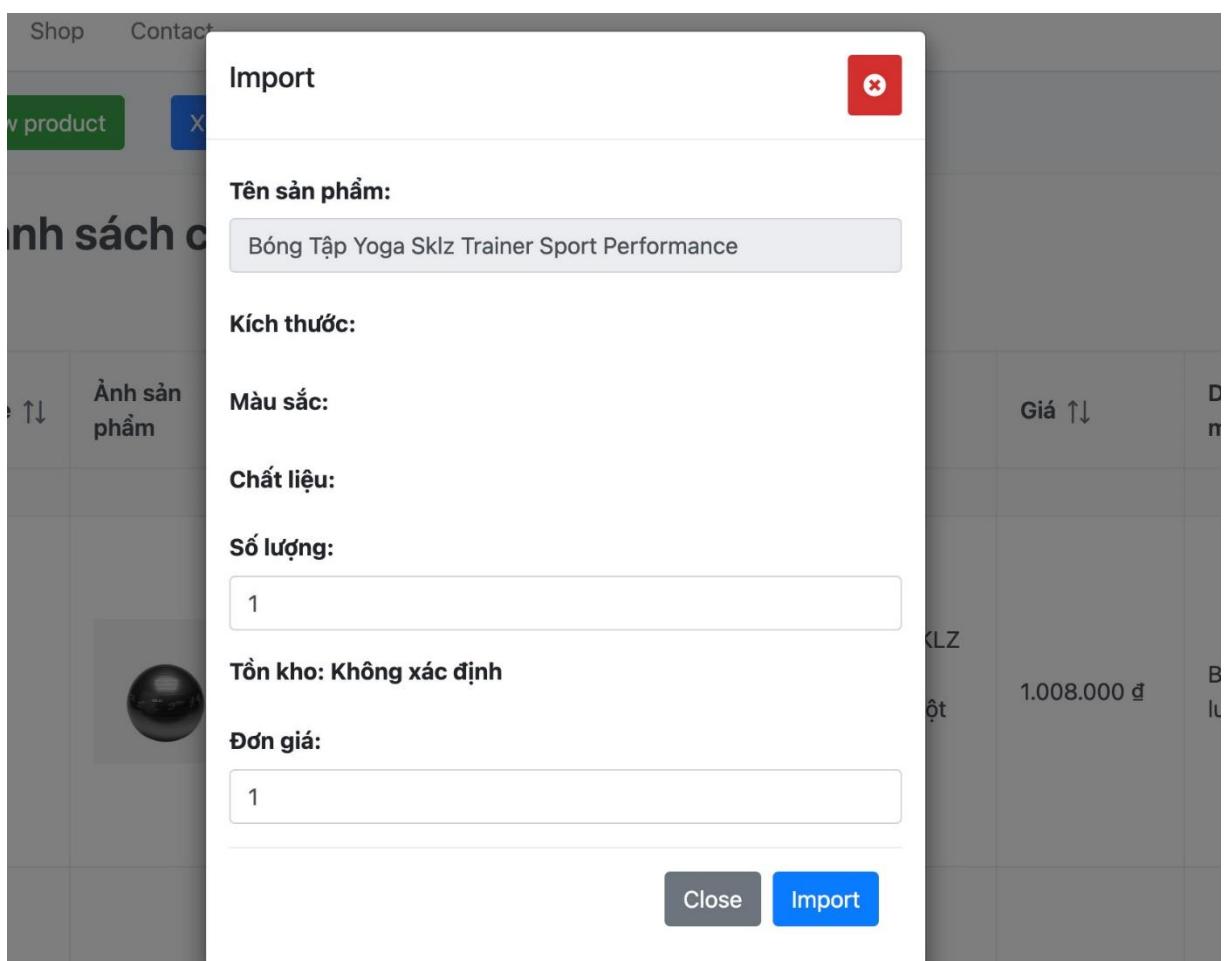


Hình 3. 73. Biểu mẫu tạo một sản phẩm mới

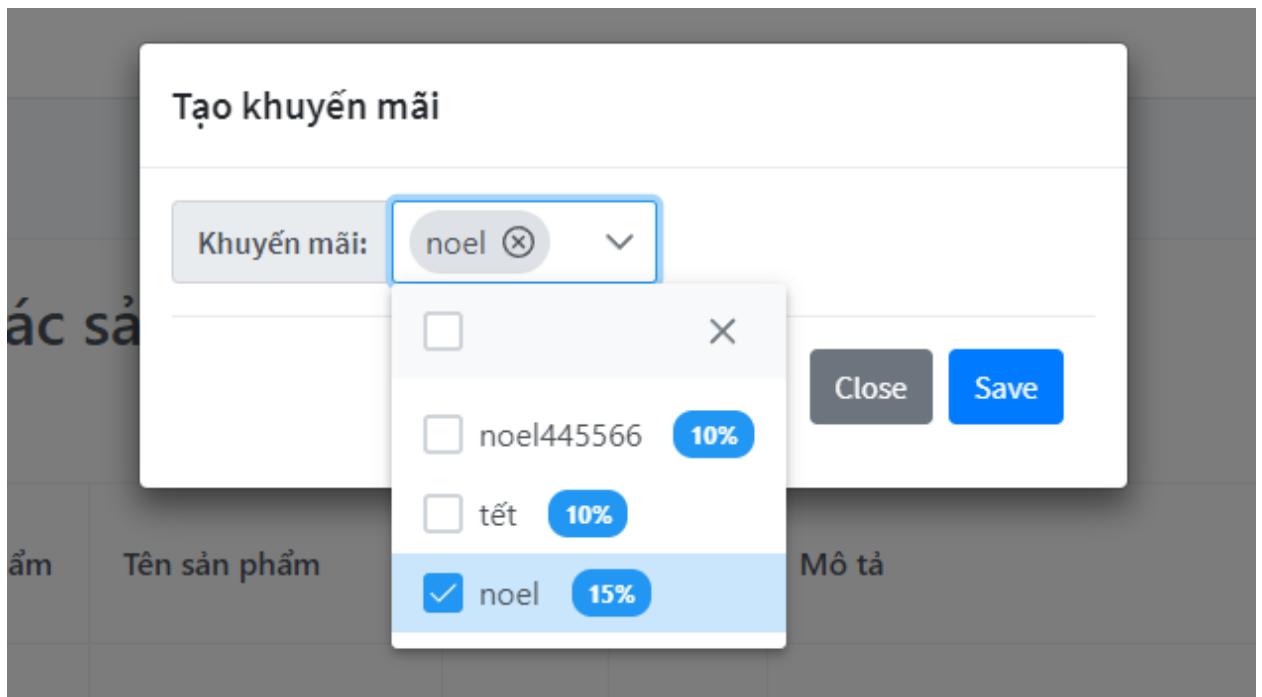
Giao diện cập nhật một sản phẩm đang tồn tại như **Hình 3. 74**



Hình 3. 74. Form cập nhật một sản phẩm đang tồn tại
Giao diện nhập hàng cho một sản phẩm đang tồn tại như **Hình 3. 75**



Hình 3. 75. Form nhập thêm hàng cho sản phẩm đang tồn tại
Giao diện tạo khuyến mãi cho một sản phẩm đang tồn tại như **Hình 3. 76**



Hình 3. 76. Giao diện thêm khuyến mãi cho sản phẩm đang tồn tại
Giao diện xem danh sách lịch sử các thay đổi giá của sản phẩm như **Hình 3. 77**

Lịch sử giá của sản phẩm #130			
Giá	Thời gian thay đổi	Người thay đổi	Mã người thay đổi
1600000	2023-11-25 14:09:50	Nguyễn Văn Huy	6
1593728	2023-11-25 05:58:19	Nguyễn Văn Test	5

Hình 3. 77. Giao diện danh sách lịch sử các thay đổi giá của sản phẩm
Tiếp theo, đến phần danh mục, giao diện sẽ như **Hình 3. 78**

Biểu tượng	Tên danh mục	Hành động
	Thảm yoga	<button>Update</button> <button>Delete</button>
	Bóng tập luyện	<button>Update</button> <button>Delete</button>
	Dây nhảy	<button>Update</button> <button>Delete</button>
	Găng tay	<button>Update</button> <button>Delete</button>

Hình 3. 78. Giao diện quản lí danh mục

Giao diện thêm một danh mục mới như **Hình 3. 79**

Thêm 1 danh mục mới

Biểu tượng:

Choose File No file chosen

Tên danh mục:

Close Insert

Hình 3. 79. Form thêm một danh mục mới

Giao diện cập nhật một danh mục đang tồn tại như **Hình 3. 80**

Cập nhật danh mục

Biểu tượng:

sả

Choose File No file chosen



Tên danh mục:

Bàn trang điểm

Close **Update**

Hình 3. 80. Form cập nhật một danh mục đang tồn tại
Tiếp theo là đến giao diện của phần khuyến mãi được thể hiện ở **Hình 3. 81**

Danh sách các khuyến mãi:

Code ↑↓	Tên khuyến mãi	Phần trăm ↑↓	Ngày bắt đầu ↑↓	Ngày kết thúc ↑↓	Hành động
5	noel445566	10	2023-11-09 00:00:00	2023-11-18 00:00:00	Update Delete
3	tết	10	2024-01-31 17:00:00	2024-02-28 17:00:00	Update Delete
2	noel	15	2023-11-13 17:00:00	2023-11-29 17:00:00	Update Delete

Tổng cộng 3 khuyến mãi.

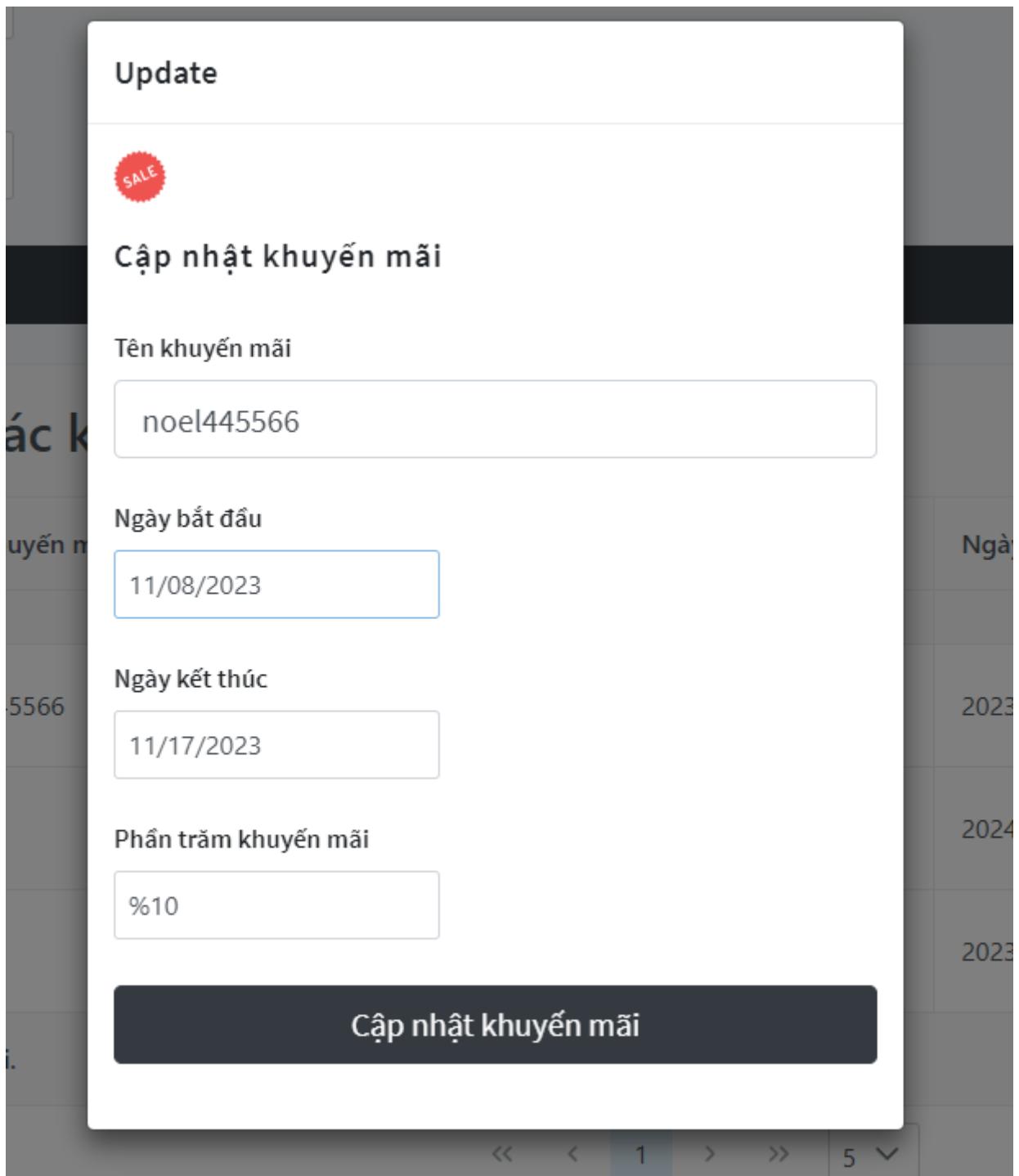
<< < 1 > >> 5 ▾

Hình 3. 81. Giao diện quản lý các khuyến mãi
Giao diện thêm một khuyến mãi mới như **Hình 3. 82**

The image shows a user interface for creating a new promotion. At the top left, there is a small red circular badge with the word "SALE". Below it, the title "Tạo khuyến mãi" (Create promotion) is displayed in bold black font. The form consists of several input fields and a main action button. The first field is "Tên khuyến mãi" (Promotion name), which is currently empty. The second field is "Ngày bắt đầu" (Start date), containing the value "11/16/2023". The third field is "Ngày kết thúc" (End date), which is also empty. The fourth field is "Phần trăm khuyến mãi" (Discount percentage), showing "%10". At the bottom right of the form area is a large, dark blue rectangular button with the text "Tạo khuyến mãi" in white.

Hình 3. 82. Form tạo một khuyến mãi mới

Giao diện cập nhật một khuyến mãi đang tồn tại như **Hình 3. 83**



Hình 3. 83. Giao diện cập nhật một sản phẩm đang tồn tại

Tiếp theo là giao diện quản lý các đơn đặt hàng như **Hình 3. 84**, có các hành động ở các nút như thay đổi trạng thái đơn đặt hàng thành đã thành toán, chưa thành toán, đang vận chuyển và huỷ đơn, bên cạnh đó còn có nút “Lịch sử thay đổi”, khi nhấn vào nút này có thể thấy được các thay đổi của đơn hàng trong quá khứ đến hiện tại như **Hình 3. 85**.

Đơn đặt hàng	Ngày ↑↓	Tình trạng ↑↓	Giao hàng đến	Khách hàng ↑↓	Tổng	Hành động
#31	2023-11-13 04:43:39	Đang xử lý	Ninh Kiều Cần Thơ	Nguyễn Tấn Tài	398.520 ₫	Đã thanh toán Chưa thanh toán Đang vận chuyển Hủy đơn Lịch sử thay đổi
#27	2023-09-04 15:06:40	Đang xử lý	ertet	Nguyễn Tấn Tài	398.520 ₫	Đã thanh toán Chưa thanh toán Đang vận chuyển Hủy đơn Lịch sử thay đổi

Hình 3. 84. Giao diện quản lý các đơn đặt hàng

Trạng thái	Thời gian thay đổi	Người thay đổi	Mã người thay đổi
Đã thanh toán	2023-11-25 14:04:21	Nguyễn Văn Huy	6
Đang vận chuyển	2023-11-25 14:04:19	Nguyễn Văn Huy	6
Đang xử lý	2023-11-25 14:04:16	Nguyễn Văn Huy	6

Hình 3. 85. Giao diện danh sách các lịch sử trạng thái của đơn hàng

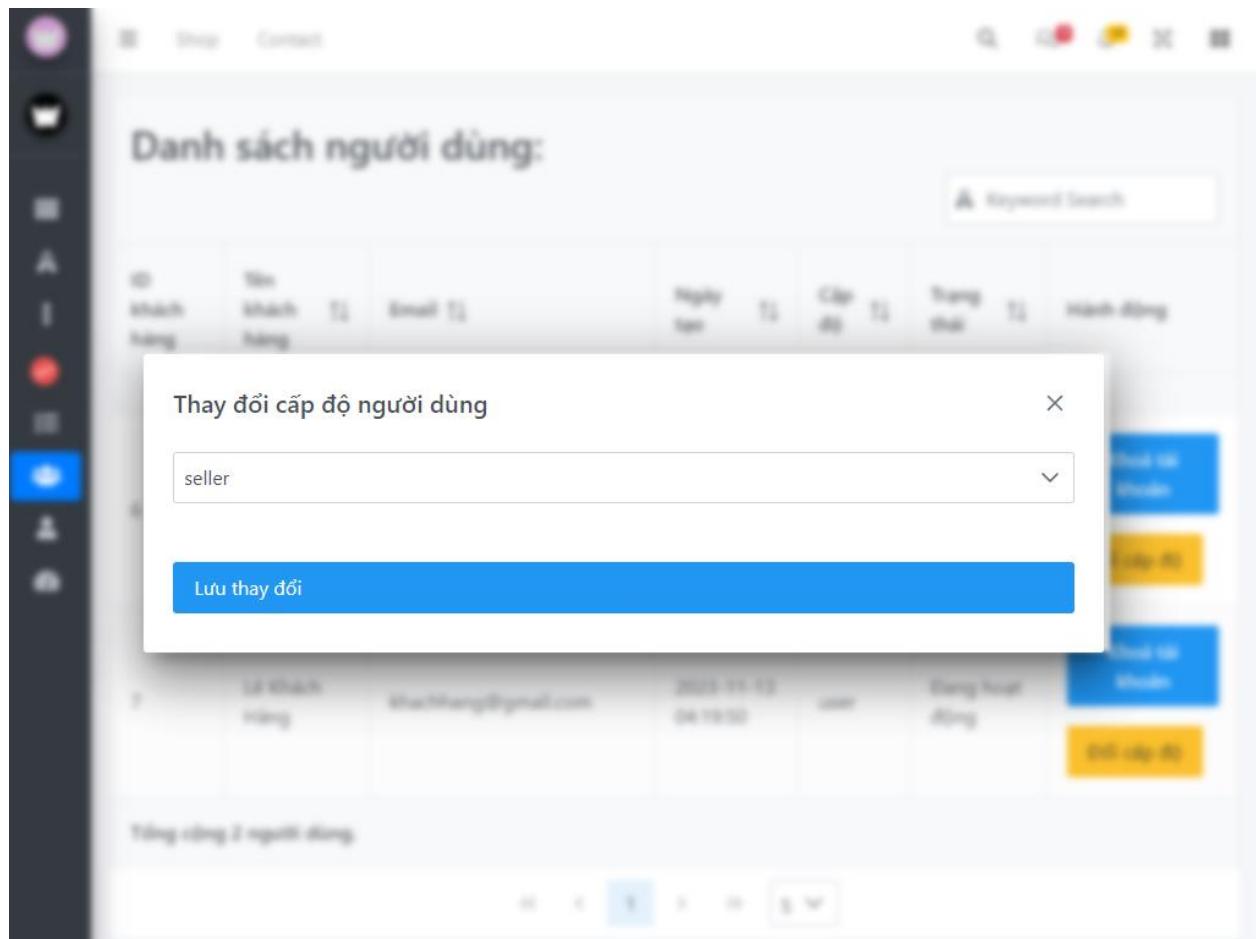
Với người bán hàng thì việc quản lý khách hàng cũng là một phần quan trọng và giao diện quản lý khách hàng được thể hiện ở **Hình 3. 86**

The screenshot shows a user management interface with a sidebar containing icons for navigation and settings. The main area is titled "Danh sách người dùng:" (List of users) and features a search bar labeled "Keyword Search". A table lists users with the following columns:

ID khách hàng	Tên khách hàng ↑↓	Email ↑↓	Ngày tạo ↑↓	Cập độ ↑↓	Trạng thái ↑↓	Hành động
6	Nguyễn Văn Huy	nguyenvantest2@gmail.com	2023-08-29 21:31:27	seller	Đã bị khoá	Mở tài khoản Đổi cấp độ
7	Lê Khách Hàng	khachhang@gmail.com	2023-11-13 11:19:50	user	Đang hoạt	Khoá tài khoản Đổi cấp độ
8	Chưa cung	asdsad@yahoo.com	2023-11-23 13:11:19	user	Đang hoạt	Khoá tài khoản Đổi cấp độ
9	Chưa cung	nguyenvantest2000@gmail.com	2023-11-23 13:11:19	user	Đang hoạt	Khoá tài khoản

Hình 3. 86. Giao diện quản lý khách hàng

Tại giao diện quản lý khách hàng ở **Hình 3. 86**, người bán có thể xem các thông tin của người dùng như ID khách hàng, tên khách hàng, email, ngày tạo, trạng thái của tài khoản (đã bị khoá hoặc đang hoạt động). Bên cạnh đó còn có thể chỉ định 1 tài khoản của khách hàng khoá tài khoản hoặc mở tài khoản đó bằng cách nhấn nút “Mở tài khoản” hoặc “Khoá tài khoản”. Bên cạnh đó người bán cũng có thể đổi cấp độ của người dùng (seller hoặc user) bằng cách nhấn vào nút “Đổi cấp độ” sẽ mở ra giao diện để thực hiện đổi cấp độ như **Hình 3. 87**



Hình 3. 87. Giao diện thay đổi cấp độ của người bán hàng

Ở cấp độ của người quản trị cũng có chức năng quản lý tương tự như người bán hàng nhưng mà họ có quyền hạn cao hơn, người quản trị có thể đổi ở cấp độ admin, seller, user. Giao diện quản lý người dùng như **Hình 3. 88** và các cấp độ mà người quản trị có thể thay đổi như **Hình 3. 89**

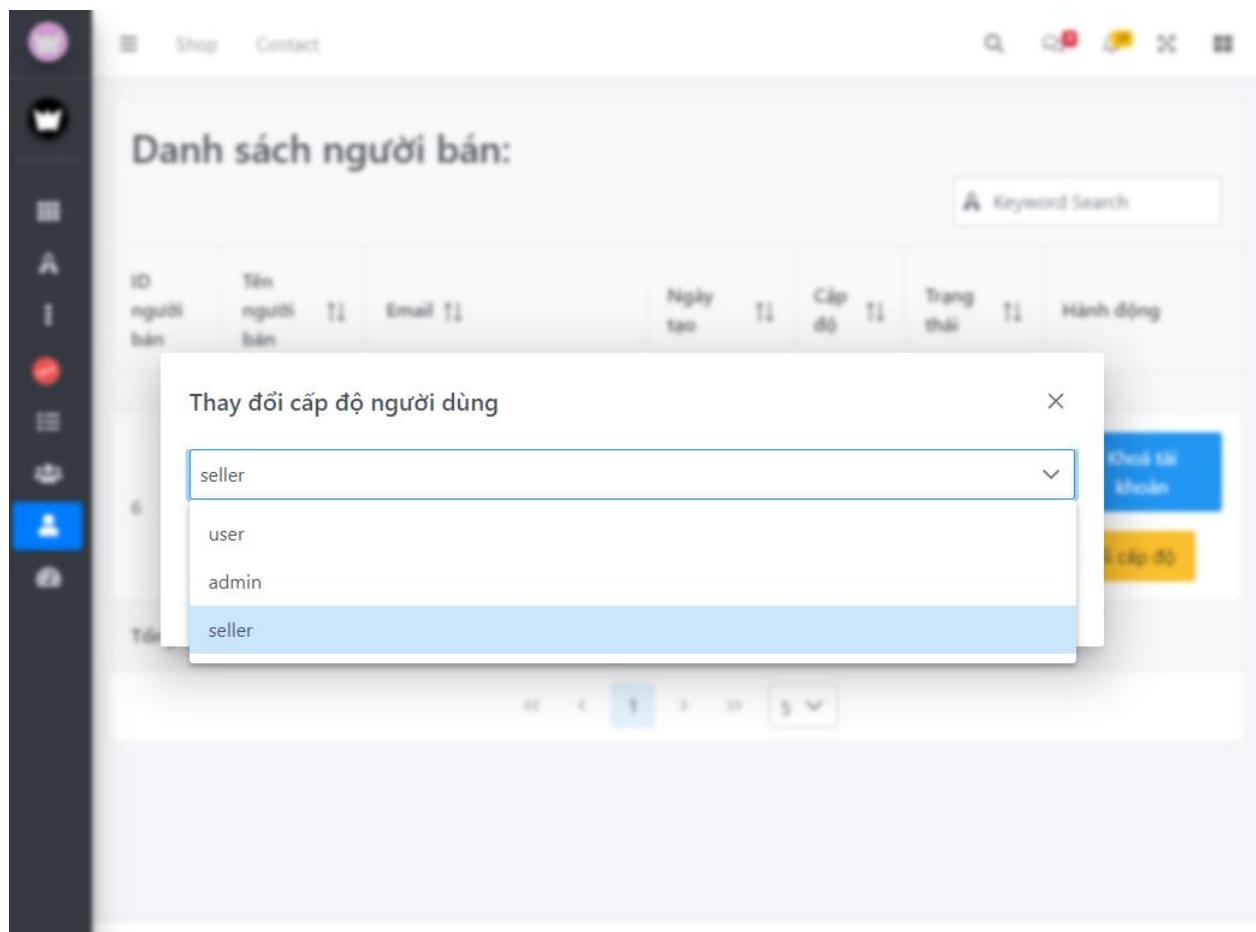
The screenshot shows a user interface for managing sellers. On the left is a vertical sidebar with icons for navigation. The main area has a header with 'Shop' and 'Contact' buttons, a search bar, and notification icons. Below the header is a title 'Danh sách người bán:' and a 'Keyword Search' input field. A table lists one seller with columns for ID, name, email, creation date, status, and actions. The table includes sorting arrows for each column. At the bottom, it says 'Tổng cộng 1 người dùng.' and shows a page navigation bar with a single page selected.

ID người bán	Tên người bán ↑↓	Email ↑↓	Ngày tạo ↑↓	Cập độ ↑↓	Trạng thái ↑↓	Hành động
6	Nguyễn Văn Huy	nguyenvantest2@gmail.com	2023-08-29 14:31:27	seller	Đang hoạt động	<button>Khoá tài khoản</button> <button>Đổi cấp độ</button>

Tổng cộng 1 người dùng.

<< < 1 > >> 5 ▾

Hình 3.88. Giao diện quản lý danh sách người bán hàng



Hình 3. 89. Giao diện thay đổi cấp độ người dùng dành cho người quản trị

CHƯƠNG 4. KIỂM THỬ VÀ ĐÁNH GIÁ

4.1. Mục tiêu kiểm thử

Mục tiêu kiểm thử là kiểm tra các giải thuật và chức năng hoạt động chính xác đúng mục đích. Tìm những lỗi tồn đọng trong thời gian phát triển phần mềm chưa nhận ra.

Các chức năng kiểm thử được thể hiện ở **Bảng 4. 1**

Bảng 4. 1. Các chức năng kiểm thử

STT	Mã chức năng	Tên chức năng
1	DN	Đăng nhập
2	DK	Đăng ký
3	GH_01	Lấy số lượng giỏ hàng
4	GH_02	Lấy danh sách giỏ hàng
5	GH_03	Xoá 1 sản phẩm trong giỏ hàng
6	GH_04	Thêm 1 sản phẩm vào giỏ hàng
7	GH_05	Cập nhật số lượng sản phẩm trong giỏ hàng
8	PL	Hiển thị danh sách sản phẩm
9	CMT_01	Bình luận một sản phẩm
10	CMT_02	Trả lời một bình luận
11	CMT_03	Xoá bình luận đã bình luận
12	CMT_04	Cập nhật bình luận
13	CMT_05	Hiển thị danh sách bình luận
14	RATE_01	Hiển thị danh sách các đánh giá
15	RATE_02	Đánh giá một sản phẩm
16	BILL_1	Thanh toán và tạo hoá đơn giỏ hàng
17	BILL_2	Xem thông tin đơn hàng đã thanh toán
18	BILL_3	Cập nhật trạng thái đơn hàng

19	BILL_4	Hiển thị danh sách các đơn hàng
20	PM_01	Thêm một sản phẩm mới
21	PM_02	Cập nhật một sản phẩm
22	PM_03	Xoá một sản phẩm
23	PM_04	Nhập thêm hàng cho sản phẩm
24	PM_05	Thêm khuyến mãi cho sản phẩm
25	PM_06	Tìm kiếm sản phẩm
26	CM_01	Thêm một danh mục mới
27	CM_02	Cập nhật một danh mục
28	CM_03	Xoá một danh mục
29	CM_04	Tìm kiếm danh mục
30	SALE_01	Tạo một khuyến mãi mới
31	SALE_02	Cập nhật một khuyến mãi
32	SALE_03	Xoá một khuyến mãi
33	UM_01	Hiển thị danh sách các người dùng
34	UM_02	Khoá tài khoản người dùng
35	UM_03	Đổi cấp độ người dùng
36	UM_04	Tìm kiếm người dùng
37	USER_01	Cập nhật thông tin cá nhân
38	USER_02	Đăng xuất tài khoản
39	TK	Thống kê dữ liệu trên toàn hệ thống
40	SEARCH	Tìm kiếm sản phẩm
41	CHAT	Nhắn tin với chatbot
42	ANA	Phân tích cảm xúc đánh giá

Kiểm tra giao diện đăng nhập (DN): tính năng đăng nhập, kiểm tra thông tin đăng nhập có hợp lệ hay không.

Kiểm tra giao diện đăng ký (DK): tính năng đăng ký, kiểm tra thông tin đăng ký có hợp lệ hay không.

Kiểm tra giao diện giỏ hàng (GH_01, GH_02, GH_03, GH_04, GH_05): các tính năng như lấy số lượng giỏ hàng, danh sách giỏ hàng, xoá 1 sản phẩm trong giỏ hàng, thêm 1 sản phẩm trong giỏ hàng, cập nhật số lượng sản phẩm trong giỏ hàng, đánh giá sản phẩm đã mua.

Kiểm tra giao diện danh sách sản phẩm (PL): các tính năng như hiển thị danh sách sản phẩm theo nhiều loại hiển thị khác nhau.

Kiểm tra giao diện danh sách bình luận (CMT_01, CMT_2, CMT_3, CMT_4, CMT_5): các tính năng như bình luận một sản phẩm, trả lời một bình luận, xoá bình luận, xoá trả lời bình luận, cập nhật bình luận, cập nhật trả lời bình luận, hiển thị danh sách bình luận.

Kiểm tra giao diện đánh giá sản phẩm (RATE_01, RATE_02): các tính năng như hiển thi danh sách các đánh giá và đánh giá một sản phẩm.

Kiểm tra giao diện thanh toán và hoá đơn (BILL_01, BILL_02, BILL_03, BILL_04): tính năng thanh toán giỏ hàng, xem thông tin chi tiết của hoá đơn, cập nhật hoá đơn, hiển thị hoá đơn.

Kiểm tra giao diện quản lý sản phẩm (PM_01, PM_02, PM_03, PM_04, PM_05, PM_06): các tính năng như thêm một sản phẩm mới, cập nhật một sản phẩm, xoá một sản phẩm, nhập thêm hàng cho sản phẩm, thêm khuyến mãi cho sản phẩm, tìm kiếm sản phẩm.

Kiểm tra giao diện quản lý danh mục (CM_01, CM_02, CM_03, CM_04): các tính năng như thêm một danh mục mới, cập nhật một danh mục, xoá một danh mục, tìm kiếm danh mục.

Kiểm tra giao diện quản lý khuyến mãi (SALE_01, SALE_02, SALE_03): các tính năng như tạo một khuyến mãi mới, cập nhật một khuyến mãi, xoá một khuyến mãi.

Kiểm tra giao diện quản lý người dùng (UM_01, UM_02, UM_03, UM_04): các tính năng như hiển thị danh sách các người dùng, khoá tài khoản người dùng, đổi cấp độ người dùng, tìm kiếm người dùng.

Kiểm tra giao diện người dùng (USER_01, USER_02): các tính năng như cập nhật thông tin người dùng, đăng xuất tài khoản.

Kiểm tra giao diện thống kê (TK): các thống kê dữ liệu trên toàn bộ hệ thống.

Kiểm tra giao diện tìm kiếm sản phẩm (SEARCH): các tính năng tìm kiếm sản phẩm.

Kiểm tra giao diện nhắn tin (CHAT): tính năng nhắn tin với chatbot.

Kiểm tra giao diện phân tích (ANA): tính năng phân tích cảm xúc đánh giá.

4.2. Kịch bản kiểm thử

Dựa theo chức năng muốn kiểm thử, kịch bản để kiểm thử được thể hiện ở **Bảng 4.2**

Bảng 4.2. Kịch bản kiểm thử

STT	Mã chức năng	Mã kịch bản	Kịch bản kiểm thử
1	DN	DN_01	Đăng nhập website với địa chỉ email và mật khẩu có tồn tại trên hệ thống
		DN_02	Đăng nhập website với địa chỉ email không tồn tại và mật khẩu đúng
		DN_03	Đăng nhập website với địa chỉ email tồn tại và mật khẩu sai
		DN_04	Đăng nhập website với các thông tin tài khoản hợp lệ nhưng mà trạng thái tài khoản đã bị khoá
2	DK	DK_01	Đăng ký tài khoản khi email hợp lệ, mật khẩu và mật khẩu nhập lại khớp nhau và hợp lệ
		DK_02	Đăng ký tài khoản khi email không hợp lệ, mật khẩu và mật khẩu nhập lại khớp nhau và hợp lệ
		DK_03	Đăng ký tài khoản khi email hợp lệ, mật khẩu và mật khẩu nhập lại hợp lệ nhưng khác nhau
		DK_04	Đăng ký tài khoản khi email hợp lệ, mật khẩu và mật khẩu nhập lại khớp nhau nhưng không hợp lệ
3	GH_01	GH_01	Lấy số lượng giỏ hàng đang có của người dùng đã đăng nhập
4	GH_02	GH_02_01	Lấy danh sách giỏ hàng của một người dùng
		GH_02_02	Lấy danh sách giỏ hàng đang trong quá trình xử lý của một khách hàng

		GH_02_03	Lấy danh sách giỏ hàng đã thanh toán của một khách hàng
		GH_02_04	Lấy danh sách giỏ hàng đang vận chuyển của một khách hàng
		GH_02_05	Lấy danh sách giỏ hàng đã bị huỷ của một khách hàng
5	GH_03	GH_03_01	Xoá 1 sản phẩm trong giỏ hàng
6	GH_04	GH_04_01	Thêm 1 sản phẩm vào giỏ hàng và chọn đầy đủ các màu sắc, chất liệu, kích thước (nếu có)
		GH_04_02	Thêm 1 sản phẩm vào giỏ hàng và chưa chọn đầy đủ các màu sắc, chất liệu, kích thước (nếu có)
7	GH_05	GH_05_01	Cập nhật số lượng một sản phẩm trong giỏ hàng
8	PL	PL_01	Hiển thị danh sách tất cả các sản phẩm
		PL_02	Hiển thị danh sách các sản phẩm bán chạy
		PL_03	Hiển thị top 10 sản phẩm được mua nhiều nhất
		PL_04	Hiển thị danh sách các sản phẩm có liên quan đến 1 sản phẩm
		PL_05	Hiển thị danh sách các sản phẩm có cùng danh mục với sản phẩm chỉ định
9	CMT_01	CMT_01_01	Bình luận vào một sản phẩm
10	CMT_02	CMT_02_01	Trả lời một bình luận
11	CMT_03	CMT_03_01	Xoá một bình luận đã bình luận
12	CMT_04	CMT_04_01	Cập nhật lại một bình luận đã bình luận
13	CMT_05	CMT_05_01	Hiển thị danh sách các bình luận của một sản phẩm
14	RATE_01	RATE_01	Hiển thị các đánh giá 5 sao của sản phẩm
		RATE_02	Hiển thị các đánh giá 4 sao của sản phẩm

		RATE_03	Hiển thị các đánh giá 3 sao của sản phẩm
		RATE_04	Hiển thị các đánh giá 2 sao một sản phẩm
		RATE_05	Hiển thị các đánh giá 1 sao của sản phẩm
		RATE_06	Hiển thị tất cả các đánh giá của sản phẩm
15	RATE_02	RATE_02	Đánh giá một sản phẩm
16	BILL_01	BILL_01_01	Thanh toán và tạo hóa đơn một giỏ hàng
17	BILL_02	BILL_02_01	Xem thông tin chi tiết một đơn hàng
18	BILL_03	BILL_03_01	Cập nhật trạng thái đơn hàng thành đã thanh toán
		BILL_03_02	Cập nhật trạng thái đơn hàng thành chưa thanh toán
		BILL_03_03	Cập nhật trạng thái đơn hàng thành đang vận chuyển
		BILL_03_04	Cập nhật trạng thái đơn hàng thành đã bị huỷ
19	BILL_04	BILL_04_01	Hiển thị danh sách tất cả các đơn hàng
		BILL_04_02	Hiển thị danh sách các đơn hàng đang xử lý
		BILL_04_03	Hiển thị danh sách các đơn hàng đã thanh toán
		BILL_04_04	Hiển thị danh sách các đơn hàng đang vận chuyển
		BILL_04_05	Hiển thị danh sách các đơn hàng đã bị huỷ
20	PM_01	PM_01_01	Thêm một sản phẩm mới
21	PM_02	PM_02_01	Cập nhật một sản phẩm
22	PM_03	PM_03_01	Xoá một sản phẩm
23	PM_04	PM_04_01	Nhập thêm hàng cho sản phẩm
24	PM_05	PM_05_01	Thêm khuyến mãi cho sản phẩm
25	PM_06	PM_06_01	Tìm kiếm sản phẩm trong danh sách

26	CM_01	CM_01_01	Thêm một danh mục mới
27	CM_02	CM_02	Cập nhật một danh mục
28	CM_03	CM_03	Xoá một danh mục
29	CM_04	CM_04	Tìm kiếm danh mục
30	SALE_01	SALE_01_01	Tạo một khuyến mãi mới
31	SALE_02	SALE_02_01	Cập nhật một khuyến mãi
32	SALE_03	SALE_03_01	Xoá một khuyến mãi
33	UM_01	UM_01_01	Hiển thị danh sách người dùng là khách hàng
		UM_01_02	Hiển thị danh sách người dùng là người bán
34	UM_02	UM_02_01	Khoá tài khoản người dùng là khách hàng
		UM_02_02	Khoá tài khoản người dùng là người bán
35	UM_03	UM_03_01	Đổi cấp độ người dùng là khách hàng
		UM_03_02	Đổi cấp độ người dùng là người bán
36	UM_04	UM_04_01	Tìm kiếm người dùng là khách hàng
		UM_04_02	Tìm kiếm người dùng là người bán
37	USER_01	USER_01_01	Cập nhật thông tin cá nhân
38	USER_02	USER_02_01	Đăng xuất tài khoản
39	TK	TK_01	Thống kê số lượng người dùng
		TK_02	Thống kê tổng số sản phẩm
		TK_03	Thống kê tổng số đơn hàng cần xử lý
		TK_04	Thống kê tổng số sản phẩm tồn kho
		TK_05	Thống kê doanh số theo thời gian
		TK_06	Thống kê người mua hàng theo thời gian
		TK_07	Thống kê các sản phẩm được mua gần đây
40	SEARCH	SEARCH_01	Tìm kiếm sản phẩm bằng từ khoá
		SEARCH_02	Tìm kiếm sản phẩm bằng hình ảnh

		SEARCH_03	Tìm kiếm sản phẩm thuộc danh mục
41	CHAT	CHAT_01	Nhắn tin hỏi về sản phẩm phổ biến
		CHAT_02	Nhắn tin hỏi về buôn bán cái gì
		CHAT_03	Nhắn tin hỏi về chất liệu các sản phẩm
		CHAT_04	Nhắn tin hỏi về chính sách bảo hành
		CHAT_05	Nhắn tin hỏi về chương trình khuyến mãi
		CHAT_06	Nhắn tin hỏi về kiểm tra tình trạng đơn hàng
		CHAT_07	Nhắn tin hỏi về các tuỳ chọn mua hàng
		CHAT_08	Nhắn tin hỏi về chi phí vận chuyển
		CHAT_09	Nhắn tin hỏi về phương thức thanh toán
		CHAT_10	Nhắn tin hỏi về sản phẩm giảm giá
		CHAT_11	Nhắn tin hỏi về thời gian giao hàng
		CHAT_12	Nhắn tin hỏi về tư vấn sắp xếp
		CHAT_13	Nhắn tin hỏi về xuất xứ sản phẩm
		CHAT_14	Nhắn tin hỏi về giới thiệu bản thân
		CHAT_15	Nhắn tin hỏi về các việc chatbot có thể làm
		CHAT_16	Nhắn tin khi cảm ơn chatbot
		CHAT_17	Nhắn tin khi hỏi tên của chatbot
		CHAT_18	Nhắn tin khi tạm biệt chatbot
		CHAT_19	Nhắn tin khi hỏi thăm chatbot
		CHAT_20	Nhắn tin khi chê bai chatbot
		CHAT_21	Nhắn tin khi hỏi chatbot tự nhận xét về nó
		CHAT_22	Nhắn tin khi chào hỏi chatbot
		CHAT_23	Nhắn tin khi hỏi về người thân
		CHAT_24	Nhắn tin các câu ngắn như ừ, ok
		CHAT_25	Nhắn tin không liên quan
42	ANA	ANA_01	Phân tích cảm xúc đánh giá

4.3. Kết quả kiểm thử

Sau khi kiểm thử theo kịch bản **Bảng 4. 2, Bảng 4. 3** là kết quả của kiểm thử:

Bảng 4. 3. Kết quả kiểm thử

STT	Mã kịch bản	Kết quả	Thành công/Thất bại
1	DN_01	Đăng nhập thành công và website chuyển đến trang chủ của cửa hàng	Thành công
2	DN_02	Báo lỗi email chưa được đăng ký	Thành công
3	DN_03	Báo lỗi mật khẩu không chính xác	Thành công
4	DN_04	Báo lỗi tài khoản đã bị khoá	Thành công
5	DK_01	Đăng ký thành công và website chuyển đến trang đăng nhập	Thành công
6	DK_02	Báo lỗi email không hợp lệ	Thành công
7	DK_03	Báo lỗi mật khẩu và mật khẩu nhập lại không khớp nhau	Thành công
8	DK_04	Báo lỗi mật khẩu không hợp lệ	Thành công
9	GH_01	Đã hiển thị số lượng giỏ hàng lên giao diện	Thành công
10	GH_02_01	Đã hiển thị danh sách giỏ hàng lên giao diện	Thành công
11	GH_02_02	Đã hiển thị danh sách giỏ hàng đang trong quá trình xử lý của một khách hàng lên giao diện	Thành công
12	GH_02_03	Đã hiển thị danh sách giỏ hàng đã thanh toán của một khách hàng lên giao diện	Thành công
13	GH_02_04	Đã hiển thị danh sách giỏ hàng đang vận chuyển của một khách hàng lên giao diện	Thành công
14	GH_02_05	Hiển thị danh sách giỏ hàng đã bị huỷ của một khách hàng lên giao diện	Thành công
15	GH_03_01	Xoá thành công 1 sản phẩm trong giỏ hàng	Thành công
16	GH_04_01	Thêm thành công 1 sản phẩm vào giỏ hàng và chọn đầy đủ các màu sắc, chất liệu, kích thước (nếu có)	Thành công

17	GH_04_02	Thêm thành công 1 sản phẩm vào giỏ hàng và chưa chọn đầy đủ các màu sắc, chất liệu, kích thước (nếu có)	Thành công
18	GH_05_01	Cập nhật thành công số lượng một sản phẩm trong giỏ hàng	Thành công
19	PL_01	Đã hiển thị danh sách tất cả các sản phẩm lên giao diện	Thành công
20	PL_02	Đã hiển thị danh sách các sản phẩm bán chạy lên giao diện	Thành công
21	PL_03	Đã hiển thị top 10 sản phẩm được mua nhiều nhất lên giao diện	Thành công
22	PL_04	Đã hiển thị danh sách các sản phẩm có liên quan đến 1 sản phẩm lên giao diện	Thành công
23	PL_05	Đã hiển thị danh sách các sản phẩm có cùng danh mục với sản phẩm chỉ định	Thành công
24	CMT_01_01	Bình luận vào một sản phẩm thành công	Thành công
25	CMT_02_01	Trả lời một bình luận thành công	Thành công
26	CMT_03_01	Xoá một bình luận đã bình luận thành công	Thành công
27	CMT_04_01	Bình luận đã bình luận thành công	Thành công
28	CMT_05_01	Đã hiển thị danh sách các bình luận của một sản phẩm lên giao diện	Thành công
29	RATE_01	Đã hiển thị các đánh giá 5 sao của sản phẩm lên giao diện	Thành công
30	RATE_02	Đã hiển thị các đánh giá 4 sao của sản phẩm lên giao diện	Thành công
31	RATE_03	Đã hiển thị các đánh giá 3 sao của sản phẩm lên giao diện	Thành công
32	RATE_04	Đã hiển thị các đánh giá 2 sao một sản phẩm lên giao diện	Thành công
33	RATE_05	Đã hiển thị các đánh giá 1 sao của sản phẩm lên giao diện	Thành công

34	RATE_06	Đã hiển thị tất cả các đánh giá của sản phẩm lên giao diện	Thành công
35	RATE_02	Đánh giá một sản phẩm thành công	Thành công
36	BILL_01_01	Thanh toán và tạo hoá đơn một giỏ hàng thành công	Thành công
37	BILL_02_01	Thông tin chi tiết một đơn hàng đã hiển thị lên giao diện	Thành công
38	BILL_03_01	Trạng thái đơn hàng đã đổi thành đã thanh toán	Thành công
39	BILL_03_02	Trạng thái đơn hàng đã đổi thành chưa thanh toán	Thành công
40	BILL_03_03	Trạng thái đơn hàng đã đổi thành đang vận chuyển	Thành công
41	BILL_03_04	Trạng thái đơn hàng đã đổi thành đã bị huỷ	Thành công
42	BILL_04_01	Hiển thị danh sách tất cả các đơn hàng lên giao diện	Thành công
43	BILL_04_02	Hiển thị danh sách các đơn hàng đang xử lý lên giao diện	Thành công
44	BILL_04_03	Hiển thị danh sách các đơn hàng đã thanh toán lên giao diện	Thành công
45	BILL_04_04	Hiển thị danh sách các đơn hàng đang vận chuyển lên giao diện	Thành công
46	BILL_04_05	Hiển thị danh sách các đơn hàng đã bị huỷ lên giao diện	Thành công
47	PM_01_01	Sản phẩm mới đã được thêm	Thành công
48	PM_02_01	Sản phẩm đã được cập nhật	Thành công
49	PM_03_01	Sản phẩm đã bị xoá	Thành công
50	PM_04_01	Sản phẩm đã có thêm hàng	Thành công
51	PM_05_01	Khuyến mãi đã được thêm cho sản phẩm	Thành công
52	PM_06_01	Kết quả tìm kiếm sản phẩm trong danh sách đã được hiển thị	Thành công

53	CM_01_01	Danh mục mới đã được thêm	Thành công
54	CM_02	Danh mục đã được cập nhật	Thành công
55	CM_03	Danh mục đã bị xoá	Thành công
56	CM_04	Kết quả tìm kiếm danh mục đã hiển thị	Thành công
57	SALE_01_01	Khuyến mãi mới đã được tạo	Thành công
58	SALE_02_01	Thông tin khuyến mãi đã được thay đổi	Thành công
59	SALE_03_01	Khuyến mãi đã bị xoá	Thành công
60	UM_01_01	Hiển thị danh sách người dùng là khách hàng lên giao diện	Thành công
61	UM_01_02	Hiển thị danh sách người dùng là người bán lên giao diện	Thành công
62	UM_02_01	Tài khoản người dùng là khách hàng đã bị khoá	Thành công
63	UM_02_02	Tài khoản người dùng là người bán đã bị khoá	Thành công
64	UM_03_01	Cấp độ người dùng là khách hàng đã bị thay đổi	Thành công
65	UM_03_02	Cấp độ người dùng là người bán đã bị thay đổi	Thành công
66	UM_04_01	Kết quả tìm kiếm người dùng là khách hàng được hiển thị	Thành công
67	UM_04_02	Kết quả tìm kiếm người dùng là người bán được hiển thị	Thành công
68	USER_01_01	Thông tin cá nhân đã được thay đổi	Thành công
69	USER_02_01	Tài khoản đã bị đăng xuất	Thành công
70	TK_01	Đã hiển thị thống kê số lượng người dùng	Thành công
71	TK_02	Đã hiển thị thống kê tổng số sản phẩm	Thành công
72	TK_03	Đã hiển thị thống kê tổng số đơn hàng cần xử lý	Thành công
73	TK_04	Đã hiển thị thống kê tổng số sản phẩm tồn kho	Thành công

74	TK_05	Đã hiển thị thống kê doanh số theo thời gian	Thành công
75	TK_06	Đã hiển thị thống kê người mua hàng theo thời gian	Thành công
76	TK_07	Đã hiển thị thống kê các sản phẩm được mua gần đây	Thành công
77	SEARCH_01	Các sản phẩm có từ khoá phù hợp được hiển thị	Thành công
78	SEARCH_02	Các sản phẩm có hình ảnh tương đồng được hiển thị	Thành công
79	SEARCH_03	Các sản phẩm có danh mục được chọn được hiển thị	Thành công
80	CHAT_01	Trả lời về sản phẩm phổ biến	Thành công
81	CHAT_02	Trả lời về buôn bán cái gì	Thành công
82	CHAT_03	Trả lời về chất liệu các sản phẩm	Thành công
83	CHAT_04	Trả lời về chính sách bảo hành	Thành công
84	CHAT_05	Trả lời về chương trình khuyến mãi	Thành công
85	CHAT_06	Trả lời và chuyển hướng người dùng đến trang quản lý đơn hàng	Thành công
86	CHAT_07	Trả lời về các tùy chọn mua hàng	Thành công
87	CHAT_08	Trả lời về chi phí vận chuyển	Thành công
88	CHAT_09	Trả lời về phương thức thanh toán	Thành công
89	CHAT_10	Trả lời về sản phẩm giảm giá	Thành công
90	CHAT_11	Trả lời về thời gian giao hàng	Thành công
91	CHAT_12	Trả lời về tư vấn sắp xếp	Thành công
92	CHAT_13	Trả lời về xuất xứ sản phẩm	Thành công
93	CHAT_14	Trả lời về giới thiệu bản thân	Thành công
94	CHAT_15	Trả lời về các việc chatbot có thể làm	Thành công

95	CHAT_16	Trả lời cảm ơn lại khách hàng	Thành công
96	CHAT_17	Trả lời tên của chatbot	Thành công
97	CHAT_18	Trả lời tạm biệt lại khách hàng	Thành công
98	CHAT_19	Trả lời về lời hỏi thăm của khách hàng	Thành công
99	CHAT_20	Trả lời với sự xin lỗi tới khách hàng	Thành công
100	CHAT_21	Trả lời nhận xét về bản thân chatbot	Thành công
101	CHAT_22	Chào hỏi lại khách hàng	Thành công
102	CHAT_23	Xử lý và trả lời lại về thông tin người thân	Thành công
103	CHAT_24	Trả lời lại các câu trả lời ngắn phù hợp	Thành công
104	CHAT_25	Trả lời lại với người dùng các vấn đề có thể tư vấn	Thành công
105	ANA_01	Danh sách và kết quả các phân tích được xuất hiện	Thành công

4.4. Kết luận chung

Trong quá trình thực hiện kiểm thử. Dưới đây là một số điểm quan trọng:

Hoàn Thiện Mục Tiêu Kiểm Thử

Hệ thống đã đạt được mục tiêu kiểm thử bằng cách xác định rõ các chức năng và yêu cầu cần được kiểm thử.

Quá trình kiểm thử đã tập trung vào các khía cạnh quan trọng như giao diện người dùng, tính năng mua sắm, thanh toán, thống kê, quản lý khách hàng, quản lý người bán, quản lý danh mục, quản lý khuyến mãi và quản lý đơn hàng.

Hiệu Suất Ổn Định

Hệ thống đã được kiểm thử đối với tải cao để đảm bảo rằng nó có thể xử lý một lượng lớn người dùng cùng một lúc mà vẫn duy trì hiệu suất ổn định.

Được thực hiện các biện pháp tối ưu hóa để giảm thiểu thời gian phản hồi và tăng trải nghiệm người dùng.

Phản Hồi Tích Cực từ Khách Hàng Kiểm Thử

Kịch bản kiểm thử đã đưa ra những phản hồi tích cực về việc sửa lỗi, cải thiện tính năng, và tối ưu hóa trải nghiệm người dùng.

Các điều chỉnh và cải tiến đã được thực hiện dựa trên kết quả kiểm thử để đảm bảo rằng hệ thống đáp ứng đúng nhu cầu của người dùng.

Triển Khai Thành Công

Sau quá trình kiểm thử kỹ lưỡng, website đã được triển khai thành công và đã nhận phản hồi tích cực từ người dùng.

Hệ thống đã chứng minh sự ổn định và khả năng hoạt động hiệu quả trên môi trường sản xuất.

Kết Luận Tổng Quan

Hệ thống thông qua quá trình phát triển và kiểm thử, website quản lý bán thiết bị thể thao đã đáp ứng được mục tiêu đề ra và mang lại trải nghiệm tích cực cho người dùng. Các kết quả từ quá trình này sẽ là cơ sở để tiếp tục cải thiện và phát triển sản phẩm trong tương lai.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Hệ thống website bán hàng đã được thiết kế và phát triển thành công, cung cấp một giải pháp trực tuyến hiệu quả cho việc mua bán thiết bị thể thao.

5.1.1. Đề tài đã đạt được

Tạo ra một giao diện người dùng thân thiện và dễ dàng sử dụng, giúp người dùng tìm kiếm, lựa chọn và mua sắm sản phẩm một cách mượt mà.

Phát triển một hệ thống quản lý sản phẩm cho phép người quản trị dễ dàng thêm, sửa, xóa sản phẩm và cập nhật thông tin.

Triển khai hệ thống giỏ hàng và thanh toán trực tuyến giúp tối ưu hóa quy trình mua sắm và tăng cường trải nghiệm người dùng.

Tích hợp hệ thống phản hồi và đánh giá sản phẩm từ người dùng, giúp tăng tính tương tác và độ tin cậy cho platform.

5.1.2. Hạn chế của đề tài

Tối ưu hóa trải nghiệm trên di động: Mặc dù website hoạt động tốt trên máy tính để bàn, nhưng việc tối ưu hóa trên các thiết bị di động như điện thoại và máy tính bảng chưa đạt được mức tối ưu nhất.

Hỗ trợ đa ngôn ngữ và tiền tệ: Hệ thống hiện tại chủ yếu hỗ trợ một ngôn ngữ và một đơn vị tiền tệ, điều này có thể giới hạn khả năng mở rộng tới các thị trường quốc tế.

Mặc dù đề tài đã đạt được nhiều kết quả tích cực, nhưng vẫn còn một số hạn chế cần được xem xét và phát triển thêm trong tương lai.

5.2. Hướng phát triển

Dựa trên những kết quả và hạn chế của đề tài, dưới đây là một số hướng phát triển tiếp theo cho hệ thống website:

5.2.1. Tối ưu hóa trải nghiệm di động

Phát triển ứng dụng di động: Xây dựng ứng dụng dành riêng cho các thiết bị iOS và Android, giúp tăng cường trải nghiệm và tương tác của người dùng trên di động.

Responsive Design: Cải tiến thiết kế web hiện tại để đảm bảo hoạt động mượt mà trên mọi kích cỡ màn hình và thiết bị.

5.2.2. Hỗ trợ đa ngôn ngữ và đa tiền tệ

Tích hợp hỗ trợ đa ngôn ngữ vào website, giúp mở rộng tới khách hàng quốc tế.

Cung cấp chức năng chuyển đổi tiền tệ dựa trên tỷ giá thị trường, giúp người mua hàng từ nhiều quốc gia có trải nghiệm mua sắm thuận tiện hơn.

5.2.3. Phát triển hệ thống Big Data

Sử dụng Big Data để phân tích hành vi và xu hướng mua sắm của khách hàng, giúp đưa ra quyết định kinh doanh hiệu quả hơn.

5.2.4. Tăng cường tính bảo mật

Phát triển và tích hợp thêm các lớp bảo mật cho hệ thống thanh toán, đảm bảo an toàn cho thông tin cá nhân và tài chính của khách hàng.

Tích hợp chứng thực hai yếu tố (2FA) để tăng cường bảo mật cho tài khoản người dùng.

Những hướng phát triển trên sẽ giúp hệ thống website trở nên hoàn thiện hơn, đáp ứng nhiều nhu cầu của khách hàng và đối tác, và mở rộng thị trường tiềm năng.

TÀI LIỆU THAM KHẢO

- [1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein and e. al, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision.*, pp. 211-252, 2015.
- [2] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (Adv. Neural Inf. Process. Syst.)*, pp. 1097-1105, 2012.
- [3] N. Paluru, A. Dayal, H. Jenssen, T. Sakinis, L. Cenkeramaddi, J. Prakash and P. Yalavarthy, "Anam-Net: Anamorphic depth embedding-based lightweight CNN for segmentation of anomalies in COVID-19 chest CT images," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 932-946, 2021.
- [4] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang and M.-M. Cheng, "Jcs: An explainable COVID-19 diagnosis system by joint classification and segmentation," pp. 3113-3126, 2021.
- [5] M. Hesamian, W. Jia, X. He and P. Kennedy, "Deep learning techniques for medical image segmentation: Achievements and challenges," *Journal of Digital Imaging*, pp. 582-596, 2019.
- [6] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 2278-2324, 1998.
- [7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.
- [8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.
- [9] Y. R. I. L. Tom Hope, Learning TensorFlow: A Guide to Building Deep Learning Systems, O'Reilly Media, 2017.
- [10] B. Z. Y. w. Y. z. Q. W. Yonghong Tian, College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm, 2019, pp. 490-494.

- [11] B. K. V. H. Praveena Mathew, Book Recommendation System through content based and collaborative filtering method, IEEE, 2016.
- [12] A. D, Algorithms and Methods in Recommender Systems, Berlin, Germany: Berlin Institute of Technology, 2011.
- [13] A. R. v. K. Deeba, "Online Book Recommendation System using Collaborative Filtering (With Jaccard Similarity)," *Journal of Physics: Conference Series*, 2019.
- [14] S. G. D. M. Manisha Chandak, "Introducing Hybrid Technique for Optimization of Book Recommender System," 2015.
- [15] D. B. J. (. Scholar, "Online book recommendation system by using collaborative filtering and association mining," 2016.
- [16] F. Sebastiani, "Text Categorization," 2004.
- [17] T. Joachims, "Proceedings of the 10th European Conference on Machine Learning," 1998.
- [18] L. H. L. V. K. a. R. R. Dino Isa, "Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine," vol. 20, no. 9, 2008.
- [19] A. a. a. S. Atika Mustafa, "Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization," 2009.
- [20] V. Vapnik, The Nature of Statistical Learning Theory, New York, 1995.
- [21] S. S. Ya Gao, An empirical evaluation of linear and nonlinear kernels for text classification using Support Vector Machines, 2010.
- [22] K. P. Bhasker Pant, DiRiboPred: A Web Tool for Classification and Prediction of Ribonucleases, 2010.
- [23] S. P. H.-M. J. D. K. S. Y. B. Hyun-Chul Kim, Constructing support vector machine ensemble, 2005.
- [24] G. K. Abdelwadood Moh'd Mesleh, Support Vector Machine Text Classification System: Using Ant Colony Optimization Based Feature Subset Selection, 2008.
- [25] S. sahay, Support Vector Machines and Document Classification, 2004.

- [26] C.-C. & L. C.-J. Chang, LIBSVM: a library for support vector machine, 2003.
- [27] T. S., Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications, 2019.
- [28] C. Macrae, Vue.js: Up and Running, O'Reilly Media, Inc., 2018.
- [29] M. Stauffer, Laravel: Up and Running, 2016.
- [30] M. Zandstra, PHP Objects, Patterns, and Practice, Apress, 2016.
- [31] J. Lockhart, Modern PHP: New Features and Good Practices, O'Reilly Media, 2015.
- [32] N. Poulton, Docker Deep Dive, Pearson IT Certification, 2017.
- [33] J. N. v. S. Kuenzli, Docker in Action, Manning Publications, 2016.
- [34] B. Cherny, Programming TypeScript, O'Reilly Media, 2019.
- [35] N. Rozentals, Mastering TypeScript, Packt Publishing, 2017.
- [36] a. W. Y. C. D. A. I. E-Commerce Website Optimization: Why 95% of Your Website Visitors Don't Buy, Dan Croxen-John và Johann van Tonder, Kogan Page, 2020.
- [37] C. Henderson, Building Scalable Web Sites, O'Reilly Media, 2006.
- [38] C. G. v. M. Love, The Interior Design Reference & Specification Book: Everything Interior Designers Need to Know Every Day, Rockport Publishers, 2018.
- [39] E. K. K. K. P. R. N.JayaLakshmi, BOOK RECOMMENDATION SYSTEM, 2022.

PHỤ LỤC

7.1. Mô tả dữ liệu mức luận lý

Bảng 7. 1. Mô hình dữ liệu mức vật lý bảng User_type

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_UT	INT(11)	X			X		Mã kiểu người dùng
Name_UT	VARCHAR(100)					X	Tên kiểu người dùng

Bảng 7. 2. Mô hình dữ liệu mức vật lý bảng Category

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Category	INT(11)	X			X		Mã danh mục
Name_Category	VARCHAR(50)					X	Tên danh mục
Icon	VARCHAR(100)					X	Biểu tượng danh mục

Bảng 7. 3. Mô hình dữ liệu mức vật lý bảng Product

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Product	INT(11)	X			X		Mã sản phẩm
ID_Category	INT(11)		X			X	Mã danh mục
Name_Product	VARCHAR(50)					X	Tên sản phẩm
Description	TEXT						Mô tả sản phẩm
Price	DOUBLE					X	Giá sản phẩm

Avatar	VARCHAR(100)					X	Ảnh đại diện sản phẩm
Amount_Product	INT(11)			0			Số lượng sản phẩm
ID_S	INT(11)					X	Mã nhà cung cấp

Bảng 7.4. Mô hình dữ liệu mức vật lý bảng Users

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_User	INT(11)	X			X		Mã người dùng
Name_User	VARCHAR(100)						Tên người dùng
Phone	CHAR(20)						Số điện thoại người dùng
Email	CHAR(100)					X	Địa chỉ email người dùng
Address	TEXT						Địa chỉ nơi ở người dùng
password	VARCHAR(65)					X	Mật khẩu người dùng
regtime	TIMESTAMP			CURRENT TIMESTAMP			Ngày tạo tài khoản

ID_UT	INT(11)		X			X	Mã loại người dùng
ID_UStatus	INT(11)		X			X	Mã trạng thái người dùng

Bảng 7. 5. Mô tả dữ liệu mức vật lý bảng Material

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Material	INT(11)	X			X		Mã chất liệu
Name_Material	VARCHAR(100)					X	Tên chất liệu

Bảng 7. 6. Mô tả dữ liệu mức vật lý bảng Color

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Color	INT(11)	X			X		Mã màu sắc
Name_Color	VARCHAR(100)					X	Tên màu sắc

Bảng 7. 7. Mô tả dữ liệu mức vật lý bảng Dimensions

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_D	INT(11)	X			X		Mã kích thước

Name_D	VARCHAR(100)					X	Tên kích thước
--------	--------------	--	--	--	--	---	----------------

Bảng 7. 8. Mô tả dữ liệu mức vật lý bảng Supplier

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_S	INT(11)	X			X		Mã kích thước
Name_S	VARCHAR(200)					X	Tên kích thước
Address_S	VARCHAR(1000)						Địa chỉ nhà cung cấp
Phone_S	CHAR(15)						Số điện thoại nhà cung cấp
Email_S	CHAR(100)						Địa chỉ email nhà cung cấp
Website_S	CHAR(100)						Website nhà cung cấp

Bảng 7. 9. Mô tả dữ liệu mức vật lý bảng Sale_Off

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_SO	INT(11)	X			X		Mã khuyến mãi

Name_SO	VARCHAR(255)					X	Tên khuyễn mãi
Discount_Percent_SO	DOUBLE					X	Tỉ lệ giảm giá khuyến mãi
Start_Date_SO	TIMESTAMP					X	Ngày bắt đầu khuyến mãi
End_Date_SO	TIMESTAMP					X	Ngày kết thúc khuyến mãi

Bảng 7. 10. Mô tả dữ liệu mức vật lý bảng Detail_Product_Material

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Material	INT(11)		X			X	Mã chất liệu
ID_Product	INT(11)		X			X	Mã sản phẩm

Bảng 7. 11. Mô tả dữ liệu mức vật lý bảng Detail_Product_Color

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Color	INT(11)		X			X	Mã màu sắc
ID_Product	INT(11)		X			X	Mã sản phẩm

Bảng 7. 12. Mô tả dữ liệu mức vật lý bảng Detail_SaleOf_Product

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_SO	INT(11)		X			X	Mã khuyến mãi
ID_Product	INT(11)		X			X	Mã sản phẩm

Bảng 7. 13. Mô tả dữ liệu mức vật lý bảng Import_History_Detail

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
id	BIGINT(20)	X			X		Mã khuyến mãi
Price_IDH	DOUBLE(20,2)					X	Mã sản phẩm
Amount_IDH	INT(11)					X	Số lượng nhập hàng
ID_Color	INT(11)		X			X	Mã màu sắc
ID_Material	INT(11)		X			X	Mã chất liệu
ID_D	INT(11)		X			X	Mã kích thước
ID_Product	INT(11)		X			X	Mã sản phẩm
ID_IH	INT(11)		X			X	Mã nhập hàng
create_at	TIMESTAMP					X	Thời gian nhập hàng
update_at	TIMESTAMP						Thời gian cập nhật hàng

Bảng 7. 14. Mô tả dữ liệu mức vật lý bảng Import History

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_IH	INT(11)		X			X	Mã nhập hàng
TotalMoney_IH	INT(11)					X	Tổng tiền nhập hàng
DATE_IH	TIMESTAMP			CURRENT TIMESTAMP			Thời gian nhập hàng

Bảng 7. 15. Mô tả dữ liệu mức vật lý bảng Detail_Product_Image

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_DPI	INT(11)	X			X		Mã chi tiết hình ảnh sản phẩm
ID_Product	INT(11)		X			X	Mã sản phẩm
Image	VARCHAR(100)					X	Đường dẫn hình ảnh

Bảng 7. 16. Mô tả dữ liệu mức vật lý bảng Product_Price_History

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
id	INT(11)	X			X		Mã lịch sử nhập hàng

price	INT(11)					X	Giá nhập
ID_Product	INT(11)		X			X	Mã sản phẩm
ID_User	INT(11)		X			X	Mã người thay đổi
date_effect	TIMESTAMP					X	Ngày hiệu lực
created_at	TIMESTAMP					X	Thời gian nhập

Bảng 7. 17. Mô tả dữ liệu mức vật lý bảng Cart_Status

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_CS	INT(11)	X			X		Mã trạng thái giờ hàng
Name_CS	VARCHAR(255)					X	Tên trạng thái giờ hàng

Bảng 7. 18. Mô tả dữ liệu mức vật lý bảng ShoppingCart

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_SC	INT(11)	X			X		Mã giờ hàng
ID_User	INT(11)		X			X	Mã người dùng
ID_CS	INT(11)		X			X	Mã trạng thái giờ hàng

Bảng 7.19. Mô tả dữ liệu mức vật lý bảng Cart_Detail

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_SC	INT(11)		X			X	Mã giỏ hàng
ID_Product	INT(11)		X			X	Mã sản phẩm
Amount_CD	INT(11)					X	Số lượng sản phẩm
Created_at	TIMESTAMP					X	Thời gian tạo chi tiết giỏ hàng
Updated_at	TIMESTAMP					X	Thời gian cập nhật chi tiết giỏ hàng
ID_Color	INT(11)		X			X	Mã màu sắc
ID_Material	INT(11)		X			X	Mã chất liệu
ID_D	INT(11)		X			X	Mã kích thước

Bảng 7.20. Mô tả dữ liệu mức vật lý bảng Ship Method

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_SM	INT(11)	X			X		Mã phương thức vận chuyển
Name_SM	VARCHAR(255)					X	Tên phương thức vận chuyển

Bảng 7. 21. Mô tả dữ liệu mức vật lý bảng MethodOfPayment

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_MOP	INT(11)	X			X		Mã phương thức thanh toán
Name_MOP	VARCHAR(255)					X	Tên phương thức thanh toán

Bảng 7. 22. Mô tả dữ liệu mức vật lý bảng Order

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Order	INT(11)	X			X		Mã đơn đặt hàng
ID_SM	INT(11)		X			X	Mã phương thức vận chuyển
ID_MOP	INT(11)		X			X	Mã phương thức thanh toán
Customer_Name	VARCHAR(100)					X	Tên khách hàng
Note_O	TEXT						Ghi chú của khách hàng
Phone_O	CHAR(20)					X	Số điện thoại của khách hàng

Address_O	TEXT					X	Địa chỉ giao hàng
-----------	------	--	--	--	--	---	-------------------

Bảng 7. 23. Mô tả dữ liệu mức vật lý bảng Order_Detail

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Order	INT(11)		X			X	Mã đơn đặt hàng
ID_SC	INT(11)		X			X	Mã giỏ hàng

Bảng 7. 24. Mô tả dữ liệu mức vật lý bảng Bill_Status

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_BS	INT(11)	X			X		Mã trạng thái đơn hàng
Name_BS	VARCHAR(255)					X	Tên trạng thái đơn hàng

Bảng 7. 25. Mô tả dữ liệu mức vật lý bảng Bill

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Bill	INT(11)	X			X		Mã trạng thái đơn hàng

CreateDate	TIMESTAMP				X	Ngày tạo hóa đơn
TotalMoney	INT(11)				X	Tổng tiền hóa đơn
VAT_rate	INT(11)				X	Tỉ lệ phí VAT
VAT_amount	INT(11)				X	Số tiền VAT
TotalMoneyCheckout	FLOAT				X	Tổng tiền sản phẩm
ID_BS	INT(11)		X		X	Mã trạng thái đơn hàng
ID_Order	INT(11)		X		X	Mã đơn đặt hàng
TotalMoneyAfterSaleOff	FLOAT				X	Tổng tiền sau khi áp dụng khuyến mãi

Bảng 7. 26. Mô tả dữ liệu mức vật lý bảng Bill_Status_History

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_BILL	INT(11)		X			X	Mã trạng

							thái đơn hàng
ID_BS	INT(11)		X			X	Mã trạng thái đơn hàng
ID_User	INT(11)		X			X	Mã người lập hoá đơn
Date_BSH	TIMESTAMP			CURRENT TIMESTAMP		X	Ngày lập hoá đơn

Bảng 7.27. Mô tả dữ liệu mức vật lý bảng Comment

Tên thuộc tính	Kiểu dữ liệu	Khoá chính	Khoá ngoại	Giá trị mặc định	Identity	NN	Diễn giải
id	BIGINT(20)	X				X	Mã bình luận
content	TEXT					X	Nội dung bình luận
reply_to	BIG_INT(20)	X					Mã bình luận cần trả lời
ID_Product	INT(11)	X				X	Mã phẩm
ID_User	INT(11)	X				X	Mã người dùng

created_at	TIMESTAMP			CURRENT TIMESTAMP		X	Ngày tạo bình luận
updated_at	TIMESTAMP						Ngày cập nhật bình luận

7.2. Cách đào tạo mô hình tìm kiếm hình ảnh tương tự

❖ Chuẩn bị dữ liệu

Tạo thư mục data chứa tất cả file ảnh

❖ Đào tạo mô hình

Bước 1: Kết nối Google Colab với Drive như **Phụ lục hình 1**

```
[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Phụ lục hình 1 Kết nối Google Colab với Drive

Bước 2: Import các thư viện liên quan đến keras mà có sử dụng cho mô hình như **Phụ lục hình 2**

```
▶ import os
import keras
import keras.utils as image

from keras.applications.imagenet_utils import decode_predictions, preprocess_input
from keras.models import Model
```

Phụ lục hình 2 Import các thư viện có liên quan đến keras

Bước 3: Load một mạng nơ-ron đã được train trước đó của VGG16, đi kèm với keras như **Phụ lục hình 3**

```
[ ] from tensorflow.keras import applications
model = keras.applications.VGG16(weights='imagenet', include_top=True)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
553467096/553467096 [=====] - 6s 0us/step
```

Phụ lục hình 3 Load mạng nơ-ron của VGG16

Bước 4: Để cho một image đầu vào đưa vào network, nó phải được tiền xử lí trước thành các vector đặc trưng với kích thước chính xác của nó. Để làm được điều này, hàm load_image như **Phụ lục hình 4** dưới đây sẽ xử lí các bước tiền xử lí như: Load một file hình ảnh từ hệ thống và biến nó thành vector đầu vào có kích thước chính xác và được qui định bởi VGG16, cụ thể là ảnh màu có kích thước 224x224.

```
[ ] import numpy as np
import matplotlib.pyplot as plt

def load_image(path):
    img = image.load_img(path, target_size=model.input_shape[1:3])
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    return img, x
```

Phụ lục hình 4 Hàm tiền xử lí hình ảnh load_image

Bước 5: Tiếp theo, thử load một ảnh từ thư mục chứa các tập hình ảnh và có thể xem được vector dữ liệu của nó trông như nào ở **Phụ lục hình 5**

```

import os
import random

def get_all_image_paths(folder):
    image_paths = []
    for root, dirs, files in os.walk(folder):
        for file in files:
            if file.lower().endswith('.jpg', '.png', '.jpeg'):
                image_paths.append(os.path.join(root, file))
    return image_paths

# Đường dẫn đến thư mục chứa hình ảnh và thư mục con
image_folder = '/content/drive/MyDrive/dataset/Shop_Sport'
# Lấy danh sách đầy đủ các tệp hình ảnh
all_image_paths = get_all_image_paths(image_folder)
# Chọn ngẫu nhiên một tệp hình ảnh
random_image_path = random.choice(all_image_paths)
# Sử dụng hàm load_image với đường dẫn được chọn ngẫu nhiên
img, x = load_image(random_image_path)
print("shape of x: ", x.shape)
print("data type: ", x.dtype)
plt.imshow(img)

```

shape of x: (1, 224, 224, 3)
data type: float32
<matplotlib.image.AxesImage at 0x7f5df1401780>



Phụ lục hình 5 Xem thử ngẫu nhiên vector dữ liệu của 1 ảnh trong dataset

Kích cỡ của ảnh trên **Phụ lục hình 5** có kích thước [1, 224, 224, 3]. Lý do nó có thêm 1 chiều để cho mạng có thể lấy các batches của các hình ảnh để xử lý chúng một cách đồng thời. Ví dụ, 10 hình ảnh có thể được truyền qua mạng nếu như x có hình dạng là [10, 224, 224, 3].

Để có thể lấy các lớp dự đoán từ model này, phải chuyển tiếp x qua model và sau đó decode_predictions được tích hợp sẵn để tra cứu các tên lớp như

```

▶ # forward the image through the network
predictions = model.predict(x)

# print out the
for _, pred, prob in decode_predictions(predictions)[0]:
    print("predicted %s with probability %0.3f" % (pred, prob))

⇒ 1/1 [=====] - 1s 1s/step
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet\_class\_index.json
35363/35363 [=====] - 0s 0us/step
predicted knee_pad with probability 0.996
predicted neck_brace with probability 0.001
predicted punching_bag with probability 0.001
predicted muzzle with probability 0.000
predicted bulletproof_vest with probability 0.000

```

Phụ lục hình 6 Tra cứu các lớp dự đoán từ model

Bước 6: Khởi tạo một model mới gọi là feature_extractor như **Phụ lục hình 7** tham chiếu tới đầu vào và đầu ra mong muốn của mô hình VGG16. Do đó, đầu ra của feature_extractor là lớp ngay trước phân loại, lớp kết nối đầy đủ 4096 neuron cuối cùng. Nó trông giống như một bản sao, nhưng bên trong, tất cả Keras làm là tạo một con trỏ đến từng lớp này và không thực sự sao chép bất cứ điều gì. Do đó, đầu ra "prediction" từ feat_extractor sẽ chỉ là lớp fc2 từ model.

Nếu chạy hàm summary() một lần nữa, sẽ thấy kiến trúc của feat_extractor giống hệt với mô hình gốc, ngoại trừ lớp cuối cùng đã được loại bỏ. Không chỉ kiến trúc giống nhau, mà cả hai cũng có cùng trọng số như **Phụ lục hình 8**

```

▶ feat_extractor = Model(inputs=model.input, outputs=model.get_layer("fc2").output)
feat_extractor.summary()

```

Phụ lục hình 7 Khởi tạo model feature_extractor

Layer (type)	Output Shape	Param #
<hr/>		
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
<hr/>		
Total params: 134260544 (512.16 MB)		
Trainable params: 134260544 (512.16 MB)		
Non-trainable params: 0 (0.00 Byte)		

Phụ lục hình 8 Kiến trúc model feature_extractor

Bước 7: Copy thư mục chứa các hình ảnh sẽ train ra thư mục ở colab có đường dẫn /content/data như **Phụ lục hình 9**

```
[ ] cp -r /content/drive/MyDrive/dataset/Shop_Sport /content/data
```

Phụ lục hình 9 Copy thư mục dataset ra thư mục làm việc colab

Bước 8: Quét qua tất cả các thư mục con và chỉ lấy ra 1 hình ảnh có chứa tên file, có chứa chữ main như **Phụ lục hình 10**, vì đây là các hình ảnh đại diện của sản phẩm. Và cấu trúc của thư mục chứa các tập dataset có dạng như **Phụ lục hình 11**

```

[ ] import os

images_path = '/content/data'
image_extensions = ['.jpg', '.png', '.jpeg'] # case-insensitive (upper/lower doesn't matter)
max_num_images = 10000

# Lưu trữ đường dẫn của các hình ảnh duy nhất từ các thư mục con chứa từ "main"
unique_images = set()

for dp, dn, filenames in os.walk(images_path):
    # Kiểm tra xem thư mục có chứa từ "main" không

        for f in filenames:
            if "main" in f.lower():
                if os.path.splitext(f)[1].lower() in image_extensions:
                    image_path = os.path.join(dp, f)
                    unique_images.add(image_path)
            else:
                print('1 folder not contains "main"')

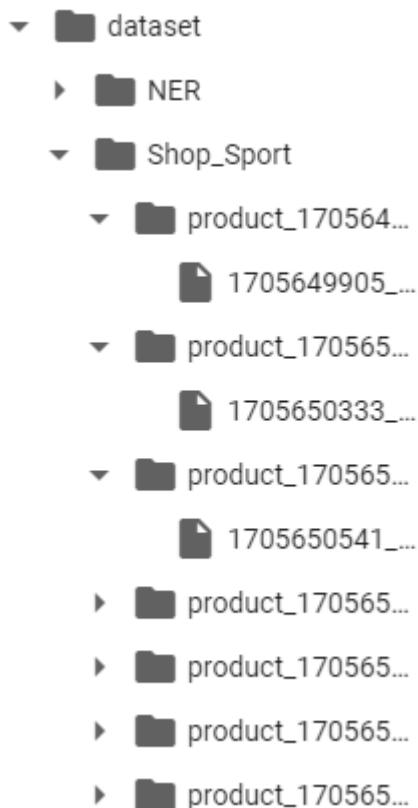
# Chuyển set thành list để tiếp tục sử dụng
unique_images = list(unique_images)

# Chọn ngẫu nhiên max_num_images hình ảnh nếu có nhiều hơn
if max_num_images < len(unique_images):
    unique_images = random.sample(unique_images, max_num_images)

print("keeping %d images to analyze" % len(unique_images))
images = unique_images

```

Phụ lục hình 10 Quét qua thư mục data để lấy các ảnh đại diện của sản phẩm



Phụ lục hình 11 Cây thư mục dataset

Bước 9: Ở bước này sẽ tiến hành lưu trữ các đặc trưng của tập hình ảnh đã được chọn ra lưu ở biến `images`, điều quan trọng là ở bước này phải tạo ra một biến `features_dict`, đây là một từ điển để lưu trữ lại cặp dữ liệu feature và `folder_name` (tên thư mục), để khi predict một hình ảnh, có thể dựa vào từ điển này có thể truy ngược ra lại được thông tin hình ảnh như **Phụ lục hình 12**.

```
# Tạo một từ điển rỗng để lưu trữ đặc trưng và tên thư mục
features_dict = {}

for i, image_path in enumerate(images):
    # Giả sử image_path có định dạng: '/content/drive/MyDrive/products_TrangTriNoiThat/folderName/imageName.jpg'
    folder_name = image_path.split('/')[-2]

    img, x = load_image(image_path);
    feat = feat_extractor.predict(x)[0]
    # Lưu trữ đặc trưng và tên thư mục vào từ điển
    features_dict[image_path] = {'feature': feat, 'folder_name': folder_name}

1/1 [=====] - 1s 614ms/step
1/1 [=====] - 1s 615ms/step
1/1 [=====] - 1s 594ms/step
1/1 [=====] - 1s 611ms/step
1/1 [=====] - 1s 613ms/step
1/1 [=====] - 1s 593ms/step
1/1 [=====] - 1s 613ms/step
1/1 [=====] - 1s 604ms/step
1/1 [=====] - 1s 593ms/step
1/1 [=====] - 1s 618ms/step
1/1 [=====] - 1s 599ms/step
1/1 [=====] - 1s 607ms/step
1/1 [=====] - 1s 638ms/step
1/1 [=====] - 1s 1s/step
1/1 [=====] - 1s 1s/step
1/1 [=====] - 1s 994ms/step
1/1 [=====] - 1s 688ms/step
1/1 [=====] - 1s 615ms/step
1/1 [=====] - 1s 594ms/step
1/1 [=====] - 1s 615ms/step
1/1 [=====] - 1s 629ms/step
1/1 [=====] - 1s 623ms/step
```

Phụ lục hình 12 Trích xuất và lưu trữ đặc trưng của các hình ảnh

Thực hiện thêm một bước nữa trước khi sử dụng mô hình này, đó là thực hiện phân tích thành phần chính (PCA) để giảm số chiều của vectơ đặc trưng xuống còn 22. Áp dụng PCA vì hai lý do:

1) Vectơ đặc trưng 4096-bit có thể có một số dư thừa trong đó, sao cho nhiều phần tử trong vectơ có mối tương quan cao hoặc tương tự nhau. Điều này sẽ làm lệch các so sánh tương tự đối với các tính năng được thể hiện quá mức đó.

2) Việc vận hành trên 4096 phần tử là không hiệu quả cả về yêu cầu không gian/bộ nhớ và tốc độ xử lý, và sẽ tốt hơn có thể giảm độ dài của các vectơ này nhưng vẫn duy trì cách biểu diễn hiệu quả tương tự. PCA cho phép thực hiện điều này bằng cách giảm số chiều của các vectơ đặc trưng từ 4096 xuống ít hơn nhiều, nhưng vẫn duy trì cách biểu diễn vẫn trung thực với dữ liệu gốc, bằng cách duy trì khoảng cách giữa các điểm tương đối. Do đó, PCA giảm lượng dư thừa trong các đặc trưng, tăng tốc độ tính toán trên chúng và giảm lượng bộ nhớ mà chúng chiếm dụng.

Bước 10: Khởi tạo một đối tượng PCA, chọn giữ lại n_components như **Phụ lục hình 13**, với n_components là số lượng bằng với số lượng ảnh đại diện của sản phẩm. Nếu giá trị này lớn hơn sẽ bị lỗi.

```
[ ] from sklearn.decomposition import PCA  
  
# features = np.array(features)  
features_matrix = np.array([value['feature'] for value in features_dict.values()])  
  
# pca = PCA(n_components=300)  
pca = PCA(n_components=22)  
pca.fit(features_matrix)
```

Phụ lục hình 13 Khởi tạo đối tượng PCA

Đối tượng pca lưu trữ ma trận biến đổi thực tế phù hợp với bước trước đó. Bây giờ có thể sử dụng nó để chuyển đổi bất kỳ vectơ đặc trưng ban đầu nào (có độ dài 4096) thành vectơ đặc trưng 300 chiều rút gọn trong không gian thành phần chính được PCA tìm thấy.

Bước 11: Lấy các vectơ đặc trưng ban đầu và chuyển đổi chúng sang không gian mới như **Phụ lục hình 14**.

```
▶ pca_features = pca.transform(features_matrix)
```

Phụ lục hình 14 Chuyển vector đặc trưng ban đầu sang không gian mới

Bước 12: Tiến hành predict một ảnh chỉ định để kiểm tra kết quả của mô hình như **Phụ lục hình 15**

```

✓ [27] from scipy.spatial import distance
5   # load image and extract features
giây urlImg = '/content/download.png';
      new_image, x = load_image(urlImg)
      new_features = feat_extractor.predict(x)
      new_pca_features = pca.transform(new_features)[0]
      distances = [distance.cosine(new_pca_features, feat) for feat in pca_features]
      idx_closest, folder_names_closest = get_closest_images(urlImg, 5)
      results_image = get_concatenated_images(idx_closest, 200)

      # display the results
      plt.figure(figsize = (5,5))
      plt.imshow(new_image)
      plt.title("query image")

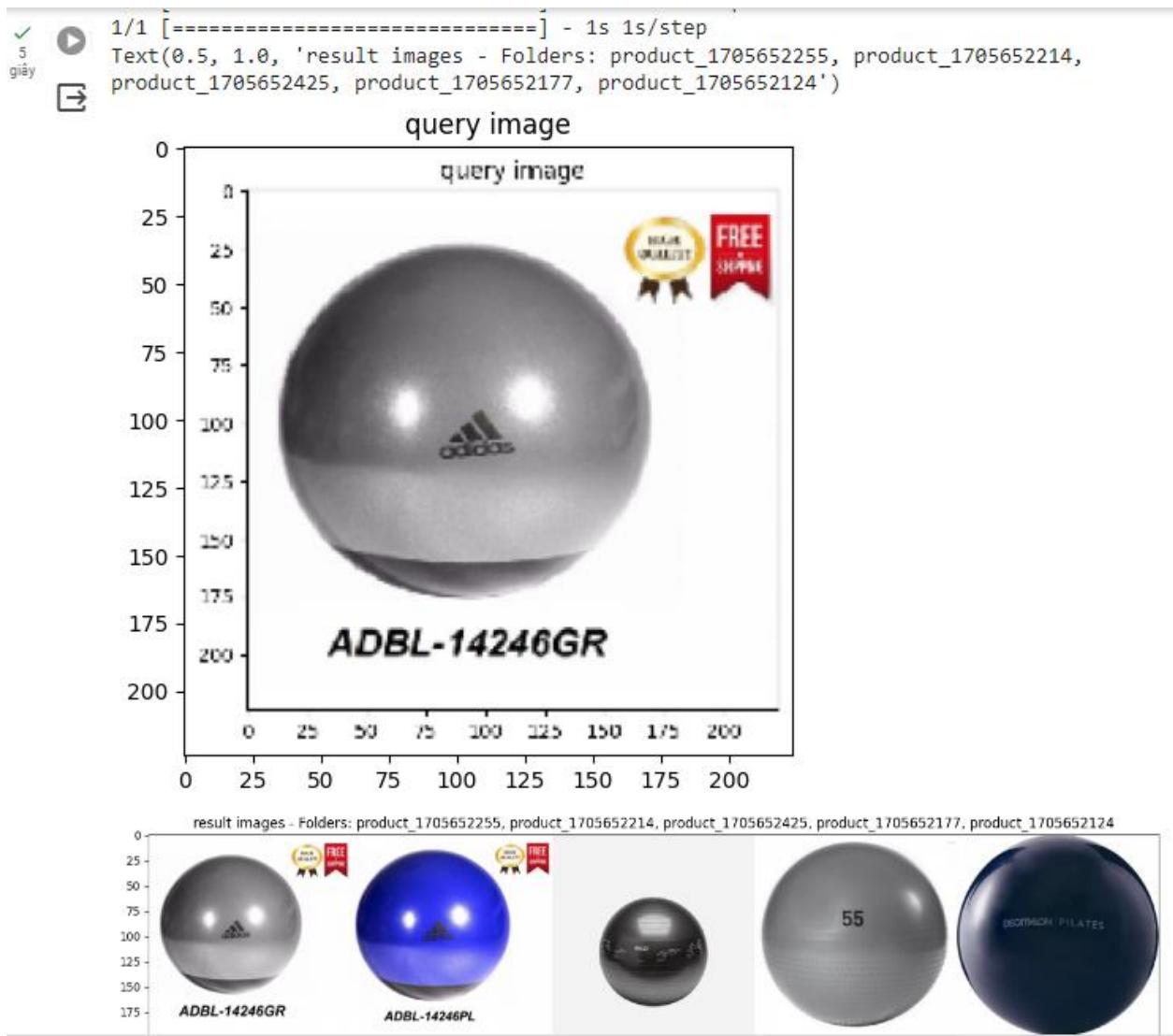
      # display the resulting images
      plt.figure(figsize = (16,12))
      plt.imshow(results_image)
      plt.title(f"result images - Folders: {', '.join(folder_names_closest)}")

```

1/1 [=====] - 1s 993ms/step
 1/1 [=====] - 1s 1s/step
 Text(0.5, 1.0, 'result images - Folders: product_1705652255, product_1705652214,
 product_1705652425, product_1705652177, product_1705652124')

Phụ lục hình 15 Tìm kiếm các ảnh tương tự của một ảnh được chỉ định

Thu được kết quả như **Phụ lục hình 16**



Phụ lục hình 16 Kết quả sau khi tìm ảnh tương tự của một hình ảnh

Từ kết quả của **Phụ lục hình 16** có thể thấy rằng, mô hình thực hiện khá tốt, cho ra các ảnh tương tự với ảnh chỉ định.

Bước cuối cùng: tiến hành lưu các giá trị của mô hình vào một file data.pkl để có thể tái sử dụng được cho hệ thống như **Phụ lục hình 17**

```
[ ] import pickle
pickle.dump([images, pca_features, pca, features_dict], open('/content/data.pkl', 'wb'))
```

Phụ lục hình 17 Lưu dữ liệu mô hình tìm kiếm ảnh tương tự vào file pkl

7.3. Cách đào tạo mô hình Chatbot

❖ Chuẩn bị dữ liệu

Tạo thư mục data, chứa các file, mỗi file chứa tất cả các intent cho Chatbot

❖ Đào tạo mô hình

Bước 1: Kết nối Google Colab với Drive **Phụ lục hình 1**

Bước 2: Thiết lập NLTK để mã hóa và bắt nguồn bằng thuật toán bắt nguồn Lancaster như **Phụ lục hình 18**

```
▶ import nltk
  nltk.download('punkt')
  from nltk.stem.lancaster import LancasterStemmer
  stemmer = LancasterStemmer()

[ ] [nltk_data] Downloading package punkt to /root/nltk_data...
[ ] [nltk_data]  Unzipping tokenizers/punkt.zip.
```

Phụ lục hình 18 Thiết lập NLTK

Bước 3: Import vào các thư viện có liên quan như **Phụ lục hình 19**

```
[ ] import tensorflow as tf
  import numpy as np
  import tflearn
  import random
  import json
  import os
```

Phụ lục hình 19 Import các thư viên có liên quan đến Chatbot

Bước 4: Thực hiện xử lí các file dataset, dồn lại thành 1 file intents.json để huấn luyện cho mô hình như **Phụ lục hình 20**

```
[ ] directoryRoot = '/content/drive/MyDrive/Data Chatbot/SportShop/'

[ ] intents_directory = directoryRoot + 'intents'
[ ] intents_data = {'intents': []}

[ ] # Loop through each subdirectory in 'intents'
[ ] for intent_folder in os.listdir(intents_directory):
[ ]     intent_path = os.path.join(intents_directory, intent_folder)

[ ]     # Check if it's a directory
[ ]     if os.path.isdir(intent_path):
[ ]         patterns_path = os.path.join(intent_path, 'patterns.txt')
[ ]         responses_path = os.path.join(intent_path, 'responses.txt')

[ ]         # Read patterns and responses from files
[ ]         with open(patterns_path, 'r', encoding='utf-8') as patterns_file:
[ ]             patterns = [line.strip() for line in patterns_file.readlines()]

[ ]         with open(responses_path, 'r', encoding='utf-8') as responses_file:
[ ]             responses = [line.strip() for line in responses_file.readlines()]

[ ]         # Add intent data to the intents_data dictionary
[ ]         intent_data = {
[ ]             'tag': intent_folder,
[ ]             'patterns': patterns,
[ ]             'responses': responses
[ ]         }

[ ]         intents_data['intents'].append(intent_data)

[ ]     # Save the intents data to intents.json
[ ]     output_json_path = 'intents.json'
[ ]     with open(output_json_path, 'w', encoding='utf-8') as output_json_file:
[ ]         json.dump(intents_data, output_json_file, ensure_ascii=False, indent=2)

[ ]     print(f"Conversion completed. Data saved to {output_json_path}")


```

Phụ lục hình 20 Dồn các file intents thành một file intents.js

Bước 5: Thực hiện tokenize mỗi từ trong các đoạn mẫu dữ liệu trong các intent như

Phụ lục hình 21

```

▶ words = []
classes = []
documents = []
ignore = ['?']
# loop through each sentence in the intent's patterns
for intent in intents['intents']:
    for pattern in intent['patterns']:
        # tokenize each and every word in the sentence
        w = nltk.word_tokenize(pattern)
        # add word to the words list
        words.extend(w)
        # add word(s) to documents
        documents.append((w, intent['tag']))
        # add tags to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])

[ ] # Perform stemming and lower each word as well as remove duplicates
words = [stemmer.stem(w.lower()) for w in words if w not in ignore]
words = sorted(list(set(words)))

# remove duplicate classes
classes = sorted(list(set(classes)))

print (len(documents), "documents")
print (len(classes), "classes", classes)
print (len(words), "unique stemmed words", words)

279 documents
23 classes ['cam-thay-ban-than-nhu-the-nao', 'disparage', 'goodbye', 'greeting', 'hoi-tham',
437 unique stemmed words ['!', ',', '.', 'ai', 'an', 'anh', 'ba', 'bao', 'biên', 'biển', 'bi

```

Phụ lục hình 21 Tokenize các từ trong các đoạn dữ liệu trong intents

Bước 6: Xử lí dữ liệu và phân chia ra thành 2 tập dữ liệu train_x và train_y như **Phụ lục hình 22**

```

▶ # create training data
training = []
output = []
# create an empty array for output
output_empty = [0] * len(classes)

# create training set, bag of words for each sentence
for doc in documents:
    # initialize bag of words
    bag = []
    # list of tokenized words for the pattern
    pattern_words = doc[0]
    # stemming each word
    pattern_words = [stemmer.stem(word.lower()) for word in pattern_words]
    # create bag of words array
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    # output is '1' for current tag and '0' for rest of other tags
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1

    training.append([bag, output_row])

# shuffling features and turning it into np.array
random.shuffle(training)
training = np.array(training)

# creating training lists
train_x = list(training[:,0])
train_y = list(training[:,1])

```

Phụ lục hình 22 Tạo dữ liệu để training

Bước 7: Tiến hành train mô hình như **Phụ lục hình 23**

```

▶ # resetting underlying graph data
tf.compat.v1.reset_default_graph()

# Building neural network
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, 10)
net = tflearn.fully_connected(net, len(train_y[0]), activation='softmax')
net = tflearn.regression(net, optimizer='adam', loss='categorical_crossentropy')

# Defining model and setting up tensorboard
model = tflearn.DNN(net, tensorboard_dir='tflearn_logs')

# Start training
model.fit(train_x, train_y, n_epoch=250, batch_size=8, show_metric=True)
model.save(directoryRoot + 'model.tflearn')

→ Training Step: 5249 | total loss: 0.01526 | time: 0.244s
| Adam | epoch: 150 | loss: 0.01526 - acc: 1.0000 -- iter: 272/279
Training Step: 5250 | total loss: 0.01436 | time: 0.251s
| Adam | epoch: 150 | loss: 0.01436 - acc: 1.0000 -- iter: 279/279
--
```

Phụ lục hình 23 Đào tạo mô hình Chatbot

Bước 8: Lưu trữ lại mô hình như **Phụ lục hình 24**

```
[ ] import pickle
pickle.dump( {'words':words, 'classes':classes, 'train_x':train_x, 'train_y':train_y}, open( directoryRoot + "training_data", "wb" ) )
```

Phụ lục hình 24 Lưu trữ lại mô hình Chatbot

Bước 9: Tạo ra các hàm giúp kiểm nghiệm mô hình như **Phụ lục hình 25, Phụ lục hình 26**

```
[ ] def clean_up_sentence(sentence):
    # tokenizing the pattern
    sentence_words = nltk.word_tokenize(sentence)
    # stemming each word
    sentence_words = [stemmer.stem(word.lower()) for word in sentence_words]
    return sentence_words

# returning bag of words array: 0 or 1 for each word in the bag that exists in the sentence
def bow(sentence, words, show_details=False):
    # tokenizing the pattern
    sentence_words = clean_up_sentence(sentence)
    # generating bag of words
    bag = [0]*len(words)
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                bag[i] = 1
            if show_details:
                print ("found in bag: %s" % w)

    return(np.array(bag))
```

Phụ lục hình 25 Hàm clean_up_sentence và bow của mô hình Chatbot

```

▶ ERROR_THRESHOLD = 0.50
context = {}
def classify(sentence):
    # generate probabilities from the model
    results = model.predict([bow(sentence, words)])[0]

    # filter out predictions below a threshold
    results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD]

    # sort by strength of probability
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append((classes[r[0]], r[1]))
    # return tuple of intent and probability
    return return_list
def getRandomResponseOfATag(self, tag):
    # Danh sách intents

    job_intents = [intent for intent in self.intents['intents'] if intent['tag'] == tag]

    if job_intents:
        # Lấy ngẫu nhiên một câu trả lời từ danh sách responses
        random_response = random.choice(job_intents[0]['responses'])
        return random_response
def response(sentence, userID='1', show_details=False):
    results = classify(sentence)
    if results:
        while results:
            for i in intents['intents']:
                if 'unknown' == results[0][0]:
                    traloi = 'Tôi không hiểu yêu cầu của bạn, ' + getRandomResponseOfATag('job')
                    return {"message": traloi, "intent": 'unknown'}
                elif i['tag'] == results[0][0]:
                    if 'context_set' in i:
                        if show_details: print ('context:', i['context_set'])
                        context(userID) = i['context_set']
                    if not 'context_filter' in i or \
                        (userID in context and 'context_filter' in i and i['context_filter'] == context(userID)):
                        if show_details: print ('tag:', i['tag'])
                        return {"message": random.choice(i['responses']), "intent": results[0][0]}
            results.pop(0)

```

Phụ lục hình 26 Hàm classify, getRandomResponseOfATag, responses của mô hình Chatbot

Bước 10: Kiểm nghiệm mô hình như **Phụ lục hình 27**

```
[ ] response("trang trí sản phẩm sao cho đẹp")
{'message': 'Hiện tại, tôi chỉ có thể trả lời cho bạn các thông tin về sản phẩm, chính sách và dịch vụ, vận chuyển và thanh toán, tư vấn và hướng dẫn.', 'intent': 'job'}
```

Phụ lục hình 27 Kiểm nghiệm mô hình Chatbot

7.4. Cách đào tạo mô hình Hệ thống gợi ý

❖ Chuẩn bị dữ liệu

Một file DanhSachSanPham.csv có dạng như **Phụ lục hình 28**

DanhSachSanPham

index	id_product	name_product	category
0	159	Bóng Tập Yoga Sklz Trainer Sport Performance	Bóng tập luyện
1	158	Bóng Tạ Sklz Trainer Med Ball (8-Lb) - Đen	Bóng tập luyện
2	157	BÓNG ADIDAS 65CM ADBL-14246GR	Bóng tập luyện
3	156	BÓNG ADIDAS 65CM ADBL-14246PL	Bóng tập luyện
4	155	BÓNG ADIDAS 55 CM ADBL-11245GR	Bóng tập luyện
5	153	Bóng tạ y tế Gym Pilates 900g - Xanh Dương	Bóng tập luyện
6	151	Bóng thể dục cỡ L - Xanh dương	Bóng tập luyện
7	150	Dây Nhảy Tập Luyện Thể Thao Tại Nhà	Dây nhảy
8	149	Dây nhảy Speed Pro - Đen	Dây nhảy
9	148	Dây Nhảy Tập Thể Dục JUMP ROPE SKLZ	Dây nhảy
10	147	Dây Nhảy Giảm Cân Ptp Power - Đen	Dây nhảy
11	146	Găng Tay Tập Gym Harbinger Men's Pro Wristwrap	Găng tay
12	145	Harbinger Training Grip Wristwrap	Găng tay
13	143	Găng Tay Tập Gym Harbinger Men's Flexfit	Găng tay
14	141	Găng Tay Tập Gym Harbinger Training Grip	Găng tay
15	140	Găng tay tập thể thao chống trầy xước chuyên dụng	Găng tay
16	139	XE ĐẠP TẬP ĐA NĂNG ELIP ZALO	Xe
17	138	GIÀN TẠ ĐA NĂNG KINGSPORT BK-1998 NEW	Tạ
18	137	TRỢ GIÁ - MÁY CHẠY BỘ KINGSPORT KUNGFU ĐƠN NĂNG	Xe
19	135	GIÀN TẠ ĐA NĂNG KINGSPORT BK-1999 NEW	Tạ
20	134	XE ĐẠP TẬP ELIP JUPITER	Xe
21	133	GHẾ TẠ ĐA NĂNG KINGSPORT BK-599	Tạ

Phụ lục hình 28 Cấu trúc File dataset csv cho hệ thống gợi ý

❖ Đào tạo mô hình

Bước 1: Kết nối Google Colab với Drive **Phụ lục hình 1**

Bước 2: Import các thư viện phụ thuộc như **Phụ lục hình 29**

```
[ ] import numpy as np
import pandas as pd
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

Phụ lục hình 29 Import các thư viện phụ thuộc của hệ thống gợi ý

Bước 3: Cài đặt thư viện pyvi để xử lí dữ liệu như **Phụ lục hình 30**

▶ pip install pyvi

Phụ lục hình 30 Cài đặt pyvi

Bước 4: Load ra các stop word và tạo ra hàm để tiền xử lí dữ liệu như **Phụ lục hình 31**

```
▶ from pyvi import ViTokenizer
from string import punctuation
stop_word = []
with open("/content/drive/MyDrive/stop_word.txt",encoding="utf-8") as f :
    text = f.read()
    for word in text.split() :
        stop_word.append(word)
    f.close()
punc = list(punctuation)
stop_word = stop_word + punc
print(stop_word)
def pre_data_progress(text):
    # Đưa chuỗi về chữ thường và trim
    text_lower = text.lower().strip()
    # Tokenize
    text_token = ViTokenizer.tokenize(text_lower)
    # Loại bỏ stop_word và punctuation
    sentence = ""

    for word in text_token.split(" "):
        if (word not in stop_word):

            if ("_" in word) or (word.isalpha() == True) or word.isdigit():
                sentence += word + " "
    return sentence.strip()
```

✉ ['bị', 'bởi', 'cà', 'các', 'cái', 'cần', 'càng', 'chì', 'chiếc', 'cho', 'chứ',

Phụ lục hình 31 Load ra các stop word và tạo ra hàm để tiền xử lí dữ liệu

Bước 5: Tiến hành tiền xử lí dữ liệu **Phụ lục hình 32**

```
[ ] # loading the data from the csv file to apandas dataframe
products_data = pd.read_csv('/content/DanhSachSanPham.csv')
# Vì có khả năng các dữ liệu như tên sản phẩm có các kí tự như khoảng trắng ở đầu hoặc các kí tự \t nên sẽ xoá đi
# Định nghĩa hàm trim cho áp dụng cho mỗi phần tử
def trim_whitespace(element):
    if isinstance(element, str):
        return pre_data_progress(element.strip())
    return element

# Áp dụng hàm trim cho tất cả các phần tử trong DataFrame
trimmed_products_data = products_data.applymap(trim_whitespace)

# In ra DataFrame sau khi đã trim
print(trimmed_products_data)
products_data = trimmed_products_data;
```

index	id_product	name_product
0	159	bóng tập yoga sklz trainer sport performance
1	158	bóng tạ sklz trainer med ball 8 lb đen
2	157	bóng adidas adbl
3	156	bóng adidas adbl
4	155	bóng adidas 55 cm adbl
5	153	bóng tạ y_tế gym pilates xanh dương
6	151	bóng thể_dục cỡ l xanh dương
7	150	dây nhảy tập_luyện thể_thao nhà
8	149	dây nhảy speed pro đen
9	148	dây nhảy tập_thể_dục jump rope sklz
10	147	dây nhảy giảm cân ptp power đen
11	146	găng_tay tập gym harbinger men s pro wristwrap
12	145	harbinger training grip wristwrap
13	143	găng_tay tập gym harbinger men s flexfit
14	141	găng_tay tập gym harbinger training grip
15	140	găng_tay tập thể_thao chống trầy_xước chuyên_dụng
16	139	xe_đạp tập đa_năng elip zalo
17	138	piàn ta đa_năng kinectsport hk 1998 new

Phụ lục hình 32 Tiết xử lí dữ liệu cho hệ thống gợi ý

Bước 6: Chọn ra các đặc trưng liên quan đến hệ thống gợi ý, giả sử, với nhu cầu gợi ý cho người dùng các bài viết có liên quan, như vậy sử dụng tên sản phẩm và tên danh mục của sản phẩm đó là đủ, có trường khác sẽ bỏ qua như **Phụ lục hình 33**.

```
[ ] # selecting the relevant features for recommendation

selected_features = ['name_product', 'category']
print(selected_features)

['name_product', 'category']

[ ] # replacing the null values with null string

for feature in selected_features:
    products_data[feature] = products_data[feature].fillna('')

▶ # combining all the 2 selected features
|
combined_features = products_data['name_product']+ ' '+products_data['category']
```

Phụ lục hình 33 Chọn ra các đặc trưng liên quan hệ thống gợi ý

Bước 7: Thực hiện tính TF-IDF cho các đặc trưng đã lấy ra như **Phụ lục hình 34**

```
[ ] # converting the text data to feature vectors
vectorizer = TfidfVectorizer()

[ ] feature_vectors = vectorizer.fit_transform(combined_features)

▶ print(feature_vectors)

⇒ (0, 56)      0.22056621642139265
    (0, 39)      0.3917233423477475
    (0, 47)      0.3917233423477475
    (0, 50)      0.3455832682292513
    (0, 45)      0.31284636465838517
    (0, 63)      0.3917233423477475
```

Phụ lục hình 34 Tính TF-IDF cho các đặc trưng hệ thống gọi ý

Bước 8: Dùng phép tính độ tương tự cosine cho các đặc trưng như **Phụ lục hình 35**

```
[ ] # getting the similarity scores using cosine similarity
similarity = cosine_similarity(feature_vectors)

▶ print(similarity)

⇒ [[1.          0.30825641 0.53384099 ... 0.          0.41816769]
   [0.30825641 1.          0.24108673 ... 0.          0.          0.25403315]
   [0.53384099 0.24108673 1.          ... 0.          0.          0.32704812]
   ...
   [0.          0.          0.          ... 1.          0.          0.          ]
   [0.          0.          0.          ... 0.          1.          0.          ]
   [0.41816769 0.25403315 0.32704812 ... 0.          0.          1.          ]]
```



```
[ ] print(similarity.shape)
```

(95, 95)

Phụ lục hình 35 Dùng phép tính độ tương tự cosine Hệ thống gọi ý

Bước 9: Lấy ra danh sách các tên sản phẩm lưu vào một biến như **Phụ lục hình 36**

```

▶ # creating a list with all the movie names given in the dataset

# In danh sách sau khi đã trim
list_of_all_titles = products_data['name_product'].tolist()
print(list_of_all_titles)

['bóng tập yoga sklz trainer sport performance', 'bóng tạ sklz trainer med ball 8 lb đen',

```

Phụ lục hình 36

Lấy danh sách tên các sản phẩm – Hệ thống gợi ý

Bước 10: Lưu lại các thông tin của mô hình vào file pkl như **Phụ lục hình 37**

```

▶ import pickle
import os
import zipfile

# Tạo thư mục /data nếu nó không tồn tại
data_folder = './data'
if not os.path.exists(data_folder):
    os.makedirs(data_folder)

# Lưu trữ mô hình
with open(os.path.join(data_folder, 'vectorizer_model.pkl'), 'wb') as f:
    pickle.dump(vectorizer, f)

with open(os.path.join(data_folder, 'similarity_matrix.pkl'), 'wb') as f:
    pickle.dump(similarity, f)

with open(os.path.join(data_folder, 'list_of_all_titles.pkl'), 'wb') as f:
    pickle.dump(list_of_all_titles, f)

# Lưu trữ DataFrame
products_data.to_pickle(os.path.join(data_folder, 'products_data.pkl'))

# Zip các file trong thư mục /data
with zipfile.ZipFile( './data_files.zip', 'w') as zip_file:
    for foldername, subfolders, filenames in os.walk(data_folder):
        for filename in filenames:
            file_path = os.path.join(foldername, filename)
            arcname = os.path.relpath(file_path, data_folder)
            zip_file.write(file_path, arcname=arcname)

```

Phụ lục hình 37

Lưu lại các thông tin của mô hình – Hệ thống gợi ý

Bước 11: Load các mô hình đã được lưu ra lại như **Phụ lục hình 38** và kiểm nghiệm mô hình như **Phụ lục hình 39**

```
[ ] import pickle
import pandas as pd

# Load mô hình
with open(os.path.join(data_folder, 'vectorizer_model.pkl'), 'rb') as f:
    vectorizer = pickle.load(f)

with open(os.path.join(data_folder, 'similarity_matrix.pkl'), 'rb') as f:
    similarity = pickle.load(f)

with open(os.path.join(data_folder, 'list_of_all_titles.pkl'), 'rb') as f:
    list_of_all_titles = pickle.load(f)

# Load DataFrame
products_data = pd.read_pickle(os.path.join(data_folder, 'products_data.pkl'))
```

Phụ lục hình 38 Load ra các dữ liệu đã lưu từ file pkl ra lại



```
product_name = input(' Enter your product name : ')
product_name = pre_data_progress(product_name)
print('=>', product_name)
list_of_all_titles = products_data['name_product'].tolist()

find_close_match = difflib.get_close_matches(product_name, list_of_all_titles)

if find_close_match == []:
    print("Không tìm thấy kết quả phù hợp")
else:
    close_match = find_close_match[0]

index_of_the_movie = products_data[products_data.name_product == close_match]['index'].values[0]

similarity_score = list(enumerate(similarity[index_of_the_movie]))

sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)

print('Movies suggested for you : \n')

i = 1

for i, movie in enumerate(sorted_similar_movies[:30], start=1):
    index = movie[0]
    title_from_index = products_data.at[index, 'name_product']
    id_product = products_data.at[index, 'id_product']
    print(i, '.', id_product, '.', title_from_index)
```

Phụ lục hình 39 Kiểm nghiệm lại mô hình – Hệ thống gợi ý

7.5. Cách đào tạo mô hình Nhận dạng cảm xúc

❖ Chuẩn bị dữ liệu

Tạo thư mục chứa 2 thư mục con tên là good và bad, lần lượt trong 2 thư mục này chứa các file có nội dung tốt và xấu về các đánh giá thu thập được

❖ Đào tạo mô hình

Bước 1: Kết nối Google Colab với Drive **Phụ lục hình 1**

Bước 2: Cài đặt thư viện pyvi để xử lí data như **Phụ lục hình 30**

Bước 3: Load ra các stop word như **Phụ lục hình 40**

```
▶ from string import punctuation
stop_word = []
with open("/content/drive/MyDrive/stop_word.txt",encoding="utf-8") as f :
    text = f.read()
    for word in text.split() :
        stop_word.append(word)
    f.close()
punc = list(punctuation)
stop_word = stop_word + punc
print(stop_word)
```

```
➡ ['bị', 'bởi', 'cà', 'các', 'cái', 'cần', 'càng', 'chỉ', 'chiếc', 'cho', 'chứ',
```

Phụ lục hình 40 Load ra các stop word – Mô hình nhận dạng cảm xúc

Bước 4: Tạo ra hàm để tiền xử lí dữ liệu như **Phụ lục hình 41**

```
▶ from pyvi import ViTokenizer
def pre_data_progress(text):
    # Đưa chuỗi về chữ thường và trim
    text_lower = text.lower().strip()
    # Tokenize
    text_token = ViTokenizer.tokenize(text_lower)
    # Loại bỏ stop_word và punctuation
    sentence = ""

    for word in text_token.split(" "):
        if (word not in stop_word):

            if ("_" in word) or (word.isalpha() == True) or word.isdigit():
                sentence += word + " "
    return sentence.strip()
```

Phụ lục hình 41 Hàm tiền xử lí dữ liệu – Mô hình nhận dạng cảm xúc

Bước 5: Import vào các thư viện phụ thuộc cần thiết như **Phụ lục hình 42**

```
[ ] # Import các thư viện cần thiết
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

import os
```

Phụ lục hình 42 Import các thư viện cần thiết – Mô hình nhận dạng cảm xúc

Bước 6: Tạo ra một hàm có thể load được data từ folder dataset mà ta đã chuẩn bị trước đó như **Phụ lục hình 43**

```
def load_data_from_folder(folder_path, label):
    data = []
    for filename in os.listdir(folder_path):
        file_path = os.path.join(folder_path, filename)
        if os.path.isfile(file_path):
            with open(file_path, 'r', encoding='utf-8') as file:
                print('- file: ', file_path)
                lines = file.readlines()
            data.extend([{"text": pre_data_progress(line), "label": label} for line in lines])
    return data

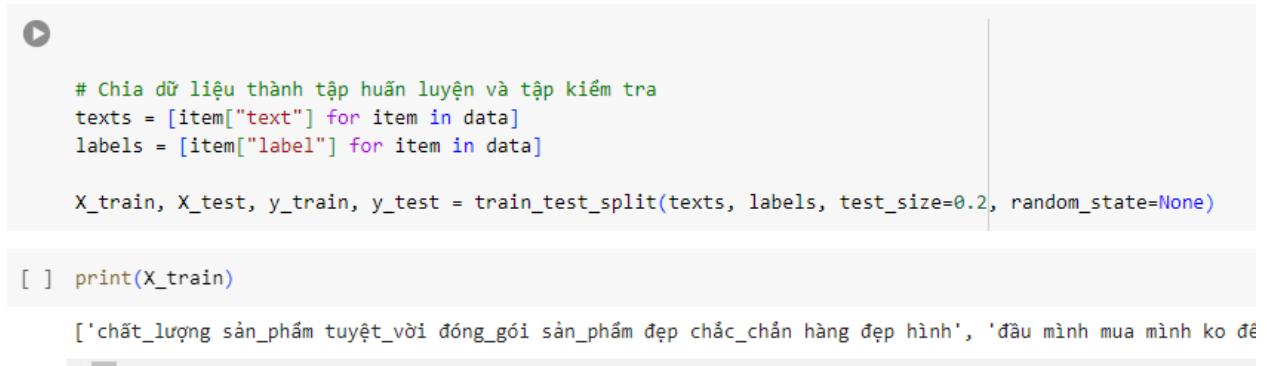
# Đường dẫn thư mục chứa các file dữ liệu
badFolderData = '/content/drive/MyDrive/Dataset-Rating/bad'
goodFolderData = '/content/drive/MyDrive/Dataset-Rating/good'

# Load dữ liệu từ thư mục
good_data = load_data_from_folder(goodFolderData, 1)
bad_data = load_data_from_folder(badFolderData, 0)

# Kết hợp các tập dữ liệu thành một tập dữ liệu chung
data = good_data + bad_data
print(data)
```

Phụ lục hình 43 Tạo ra một hàm có thể load được data từ folder dataset

Bước 7: Chia tập dữ liệu thành tập huấn luyện và kiểm tra như **Phụ lục hình 44**



```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
texts = [item["text"] for item in data]
labels = [item["label"] for item in data]

X_train, X_test, y_train, y_test = train_test_split(texts, labels, test_size=0.2, random_state=None)

[ ] print(X_train)

['chất_lượng_sản_phẩm_tuyệt_vời đóng_gói_sản_phẩm đẹp chắc_chắn hàng đẹp hình', 'đầu_mình_mua_mình_ko_đẽ
```

Phụ lục hình 44 Chia tập dữ liệu thành tập huấn luyện và kiểm tra

Ở **Phụ lục hình 44**, random_state=42 random_state trong hàm train_test_split của scikit-learn là một tham số để kiểm soát việc tạo ngẫu nhiên các tập dữ liệu. Nó là một giá trị số nguyên, và nếu bạn cung cấp cùng một giá trị cho random_state, mỗi lần bạn chạy mã, quá trình tạo ngẫu nhiên sẽ tạo ra cùng một kết quả. Nếu muốn mỗi lần chạy là 1 kết quả ngẫu nhiên thì để random_state=None, con số 42 không có ý nghĩa gì cả, chỉ là mỗi lần chạy nó check nếu vẫn giữ nguyên giá trị này thì random sẽ không thay đổi thôi.

CountVectorizer là một công cụ trong thư viện scikit-learn được sử dụng để chuyển đổi một tập hợp các văn bản thành một ma trận các đặc trưng dựa trên đếm số lần xuất hiện của từng từ trong mỗi văn bản. Điều này tạo ra một biểu diễn số lượng từ vựng trong tập dữ liệu và đếm số lần xuất hiện của từng từ trong mỗi văn bản.

Bước 8: Vector hoá văn bản như **Phụ lục hình 45**

```
▶ # Vector hóa văn bản
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
X_train_vectorized

⇒ <13662x9574 sparse matrix of type '<class 'numpy.int64'>' with 160148 stored elements in Compressed Sparse Row format>
```

Phụ lục hình 45 Vector hoá văn bản

Ở **Phụ lục hình 45**, CountVectorizer là một công cụ trong thư viện scikit-learn được sử dụng để chuyển đổi một tập hợp các văn bản thành một ma trận các đặc trưng dựa trên đếm số lần xuất hiện của từng từ trong mỗi văn bản. Điều này tạo ra một biểu diễn số lượng từ vựng trong tập dữ liệu và đếm số lần xuất hiện của từng từ trong mỗi văn bản.

fit_transform được sử dụng để học từ vựng từ tập huấn luyện (X_train) và sau đó chuyển đổi các văn bản trong tập huấn luyện thành ma trận đặc trưng. Mỗi hàng của ma trận biểu diễn một văn bản, và mỗi cột biểu diễn một từ trong từ vựng. Các giá trị trong ma trận thể hiện số lần xuất hiện của từng từ trong mỗi văn bản.

Bước 9: Huấn luyện mô hình SVM như **Phụ lục hình 46**

```
# Huấn luyện mô hình SVM
clf = SVC(kernel='linear', C=1.0)
clf.fit(X_train_vectorized, y_train)
```

Phụ lục hình 46 Huấn luyện mô hình SVM

Bước 10: Dự đoán và đánh giá mô hình như **Phụ lục hình 47**

```

# Dự đoán trên tập kiểm tra
y_pred = clf.predict(X_test_vectorized)

# Đánh giá mô hình
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print(f'Classification Report:\n{report}')

```

```

> Accuracy: 0.8905152224824356
> Classification Report:
      precision    recall   f1-score   support
          0       0.87     0.89     0.88     1526
          1       0.91     0.89     0.90     1890

      accuracy                           0.89     3416
      macro avg       0.89     0.89     0.89     3416
      weighted avg    0.89     0.89     0.89     3416

```

Phụ lục hình 47 Dự đoán và đánh giá mô hình

Bước 11: Lưu trữ mô hình như **Phụ lục hình 48**

```

▶ import pickle

# Lưu trữ mô hình vào file
with open('svm_model.pkl', 'wb') as model_file:
    pickle.dump(clf, model_file)

# Lưu trữ vectorizer nếu cần thiết
with open('vectorizer.pkl', 'wb') as vectorizer_file:
    pickle.dump(vectorizer, vectorizer_file)

```

Phụ lục hình 48 Lưu trữ mô hình SVM

Bước 12: Load ra lại mô hình đã lưu trước đó như **Phụ lục hình 49**

```

▶ # Đọc mô hình từ file
with open('svm_model.pkl', 'rb') as model_file:
    loaded_model = pickle.load(model_file)

# Đọc vectorizer nếu cần thiết
with open('vectorizer.pkl', 'rb') as vectorizer_file:
    loaded_vectorizer = pickle.load(vectorizer_file)

```

Phụ lục hình 49 Load ra lại mô hình đã lưu trước đó

Bước 13: Kiểm nghiệm mô hình như **Phụ lục hình 50**

```

comment = 'không hài lòng lắm'
comment = pre_data_progress(comment)
print(comment)
new_text_vectorized = vectorizer.transform([comment])
prediction = clf.predict(new_text_vectorized)

print(f'=> {prediction[0]}')

comment = '5 sao'
comment = pre_data_progress(comment)
print(comment)
new_text_vectorized = vectorizer.transform([comment])
prediction = clf.predict(new_text_vectorized)

print(f'=> {prediction[0]}')

comment = '5 sao cho shop'
comment = pre_data_progress(comment)
print(comment)
new_text_vectorized = vectorizer.transform([comment])
prediction = clf.predict(new_text_vectorized)

print(f'=> {prediction[0]}')

```

⇒ không hài lòng lắm
=> 0

Phụ lục hình 50 Kiểm nghiệm lại mô hình Nhận dạng cảm xúc