

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC  
HỌC PHẦN: HỌC MÁY CƠ BẢN**

**ĐỀ TÀI: XÂY DỰNG APP QUẢN LÝ CỬA HÀNG THỜI TRANG  
STYLO TÍCH HỢP TÌM KIẾM VÀ GỢI Ý SẢN PHẨM**

Giảng viên hướng dẫn: TRẦN THỊ DUNG

Sinh viên thực hiện: ĐẶNG THỊ KIM THẢO - 6351071066

LÊ HOÀNG PHÚC - 6351071057

VÕ THÀNH HOÀNG PHÚC - 6351071056

Lớp: CÔNG NGHỆ THÔNG TIN

Khoá: 63

Tp. Hồ Chí Minh, năm 2025

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN  
HỌC PHẦN: HỌC MÁY CƠ BẢN**

**ĐỀ TÀI: XÂY DỰNG APP QUẢN LÝ CỬA HÀNG THỜI TRANG  
STYLO TÍCH HỢP TÌM KIẾM VÀ GỢI Ý SẢN PHẨM**

Giảng viên hướng dẫn: **TRẦN THỊ DUNG**

Sinh viên thực hiện: **ĐẶNG THỊ KIM THẢO - 6351071066**

**LÊ HOÀNG PHÚC - 6351071057**

**VÕ THÀNH HOÀNG PHÚC - 6351071056**

Lớp: **CÔNG NGHỆ THÔNG TIN**

Khoá: **63**

Tp. Hồ Chí Minh, năm 2025

**NHIỆM VỤ THIẾT KẾ BÁO CÁO ĐỒ ÁN**  
**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

-----\*\*\*-----

**Mã sinh viên:**

6351071066

6351071057

6351071056

**Khóa:** 63

**Họ tên SV:**

Đặng Thị Kim Thảo

Lê Hoàng Phúc

Võ Thành Hoàng Phúc

**Lớp:** Công Nghệ Thông Tin

**1. Tên đề tài báo cáo đồ án**

***XÂY DỰNG APP QUẢN LÝ CỬA HÀNG THỜI TRANG STYLO TÍCH HỢP TÌM KIẾM VÀ GỢI Ý SẢN PHẨM.***

**2. Mục đích, yêu cầu**

**a. Mục đích:**

- Xây dựng hệ thống tìm kiếm sản phẩm bằng hình ảnh:
  - Ứng dụng mô hình Convolutional Neural Network (CNN) để trích xuất đặc trưng (embedding) từ hình ảnh sản phẩm.
  - Sử dụng Cosine Similarity để đo mức độ tương đồng giữa các sản phẩm dựa trên đặc trưng hình ảnh.
  - Hỗ trợ người dùng tìm kiếm sản phẩm tương tự thông qua hình ảnh đầu vào.
- Xây dựng hệ thống gợi ý sản phẩm thông minh cho cửa hàng thời trang:
  - Áp dụng phương pháp Content-Based Filtering (CBF) để gợi ý sản phẩm dựa trên nội dung mô tả sản phẩm.

- Sử dụng kỹ thuật TF-IDF (Term Frequency – Inverse Document Frequency) để biểu diễn đặc trưng văn bản của sản phẩm.
- Kết hợp Cosine Similarity nhằm xác định mức độ tương đồng giữa các sản phẩm và đề xuất danh sách sản phẩm liên quan khi người dùng xem chi tiết một sản phẩm.

#### **b. Yêu cầu:**

- Thu thập và tiền xử lý dữ liệu hình ảnh và dữ liệu mô tả sản phẩm thời trang.
- Cài đặt và huấn luyện mô hình CNN để trích xuất đặc trưng hình ảnh sản phẩm.
- Xây dựng hệ thống tìm kiếm sản phẩm bằng hình ảnh dựa trên Cosine Similarity.
- Xây dựng hệ thống gợi ý sản phẩm dựa trên Content-Based Filtering sử dụng TF-IDF và Cosine Similarity.
- Đánh giá và trực quan hóa kết quả tìm kiếm và gợi ý sản phẩm.
- Mô phỏng quy trình tích hợp các chức năng tìm kiếm và gợi ý vào ứng dụng di động Stylo.

### **3. Nội dung và phạm vi đề tài**

#### **a. Nội dung:**

- Tìm hiểu cơ sở lý thuyết về:
  - Mô hình Convolutional Neural Network (CNN) trong xử lý ảnh.
  - Phương pháp Content-Based Filtering trong hệ thống gợi ý.
  - Kỹ thuật TF-IDF và Cosine Similarity
- Phân tích bài toán tìm kiếm sản phẩm bằng hình ảnh trong lĩnh vực thời trang
- Xây dựng bộ dữ liệu gồm hình ảnh và thông tin mô tả sản phẩm thời trang.
- Huấn luyện mô hình CNN để trích xuất embedding từ hình ảnh sản phẩm.
- Xây dựng hệ thống tìm kiếm sản phẩm tương tự dựa trên hình ảnh.
- Xây dựng hệ thống gợi ý sản phẩm dựa trên nội dung mô tả sản phẩm.
- Đánh giá hiệu quả của hệ thống tìm kiếm và gợi ý sản phẩm.

- Thảo luận khả năng mở rộng và ứng dụng thực tế của hệ thống trong thương mại điện tử

#### **b. Phạm vi đề tài:**

- Dữ liệu sử dụng có quy mô nhỏ đến trung bình, phù hợp với giới hạn tài nguyên của Google Colab và tính chất thử nghiệm của đề án.
- Tích hợp mô hình sau khi train xong vào ứng dụng thực tế.
- Sử dụng các mô hình CNN có sẵn để giảm chi phí huấn luyện.

### **4. Công nghệ, công cụ và ngôn ngữ lập trình**

#### **a. Công nghệ:**

Flutter, ASP.NET Core Web API, FastAPI, SQL Server, TensorFlow, FAISS.

#### **b. Công cụ:**

Google Colab, Visual Studio Code, Visual Studio, Postman, GitHub

#### **c. Ngôn ngữ lập trình:**

Python, C#, Dart.

### **5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

#### **a. Kết quả dự kiến:**

- Xây dựng thành công ứng dụng di động Stylo hỗ trợ bán hàng thời trang, tích hợp chức năng tìm kiếm sản phẩm bằng hình ảnh và gợi ý sản phẩm thông minh.
- Hoàn thiện hệ thống tìm kiếm sản phẩm dựa trên hình ảnh với độ chính xác và hiệu năng phù hợp trong phạm vi đề án cuối kì.
- Hoàn thiện hệ thống gợi ý sản phẩm dựa trên nội dung mô tả sản phẩm, hỗ trợ người dùng khám phá các sản phẩm liên quan.
- Mô phỏng kiến trúc hệ thống thực tế, đảm bảo khả năng mở rộng và bảo trì.
- Báo cáo chi tiết quá trình xây dựng hệ thống, huấn luyện mô hình và đánh giá.

**b. Ứng dụng:**

- Người dùng có thể sử dụng hình ảnh sản phẩm để tìm kiếm các sản phẩm tương tự trong hệ thống, góp phần nâng cao trải nghiệm mua sắm.
- Khi người dùng xem chi tiết một sản phẩm, hệ thống tự động đề xuất các sản phẩm liên quan dựa trên nội dung mô tả, giúp người dùng khám phá thêm các mặt hàng phù hợp.
- Việc tích hợp chức năng tìm kiếm và gợi ý sản phẩm thông minh giúp tăng khả năng tiếp cận sản phẩm, hỗ trợ quyết định mua hàng và góp phần cải thiện doanh thu cho cửa hàng.

**6. Giáo viên và cán bộ hướng dẫn**

Họ tên: TRẦN THỊ DUNG

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao thông  
Vận tải phân hiệu tại TP. Hồ Chí Minh.

Điện thoại:

Email:

**Ngày 04 tháng 09 năm 2025**  
**Trưởng BM Công nghệ Thông tin**

**Đã giao nhiệm vụ TKBCDA**  
**Giáo viên hướng dẫn**

**ThS. Trần Phong Nhã**

**ThS. Trần Thị Dung**

Đã nhận nhiệm vụ TKBCDA

Nhóm sinh viên:

ĐẶNG THỊ KIM THẢO

LÊ HOÀNG PHÚC

VÕ THÀNH HOÀNG PHÚC

Điện thoại:

Ký tên:

Email:

[6351071066@st.utc2.edu.vn](mailto:6351071066@st.utc2.edu.vn)

[6351071057@st.utc2.edu.vn](mailto:6351071057@st.utc2.edu.vn)

[6351071056@st.utc2.edu.vn](mailto:6351071056@st.utc2.edu.vn)

## LỜI CẢM ƠN

Qua thời gian học tập và rèn luyện tại Trường Đại học Giao thông Vận tải phân hiệu tại TP. Hồ Chí Minh, đến nay chúng em đã được trang bị những kỹ năng, kiến thức cơ bản để có thể hoàn thành được bài tập cuối kì do giảng viên giao.

Chúng em cảm ơn tập thể các thầy cô giáo Bộ môn Công Nghệ Thông Tin và các thầy cô thỉnh giảng đã giảng dạy, quan tâm và không ngần ngại dành thời gian để chỉ bài và giải đáp những thắc mắc của chúng em trong những tiết học và cả những lúc ngoài giờ.

Và chúng em cảm ơn cô Thạc sĩ Trần Thị Dung đã luôn quan tâm nhiệt tình hướng dẫn, giúp đỡ chúng em trong quá trình triển khai và thực hiện bài tập cuối kì. Cô cũng luôn nhắc nhở, giúp đỡ mỗi khi chúng em gặp khó khăn, nhờ vậy mà chúng em đã hoàn thành bài tập cuối kì của nhóm mình đúng thời hạn được giao. Nếu không có sự hướng dẫn của cô thì có lẽ chúng em đã khó có thể thực hiện được bài tập đúng theo mong muốn của mình.

Chúng em đã bỏ ra nhiều thời gian để tìm hiểu và trang bị thêm kiến thức nhằm phục vụ cho việc thực hiện ý tưởng, nhưng chắc chắn rằng chúng em sẽ không thể tránh khỏi những sai sót không đáng có vì kiến thức còn hạn chế. Chúng em hi vọng rằng sẽ nhận được những lời góp ý quý báu của cô để có thể hoàn thiện ý tưởng của nhóm một cách tốt nhất có thể.

***TP. Hồ Chí Minh, ngày 25 tháng 12 năm 2025***

**Sinh viên thực hiện**

Đặng Thị Kim Thảo

Lê Hoàng Phúc

Võ Thành Hoàng Phúc



## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Tp. Hồ Chí Minh, ngày 25 tháng 12 năm 2025*  
Giáo viên hướng dẫn

**ThS. Trần Thị Dung**

# MỤC LỤC

NHIỆM VỤ THIẾT KẾ BÁO CÁO ĐỒ ÁN.....	i
LỜI CẢM ƠN .....	vi
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN .....	vii
MỤC LỤC.....	viii
DANH MỤC HÌNH ẢNH VÀ BẢNG BIỂU .....	xi
CHƯƠNG 1: MỞ ĐẦU .....	1
1.1 Giới thiệu chung về đề tài.....	1
1.1.1 Bối cảnh và sự phát triển của thương mại điện tử và nhu cầu cá nhân hóa .....	1
1.1.2 Sự cần thiết của việc ứng dụng trí tuệ nhân tạo và thuật toán tiên tiến.....	2
1.2 Mục tiêu nghiên cứu .....	3
1.2.1 Mục tiêu tổng quát.....	3
1.2.2 Mục tiêu cụ thể .....	3
1.3 Phạm vi nghiên cứu .....	3
1.3.1 Đối tượng áp dụng.....	3
1.3.2 Phạm vi chức năng .....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	5
2.1 Tổng quan về ngôn ngữ lập trình Python và FastApi.....	5
2.1.1 Giới thiệu python.....	5
2.1.2 FastAPI Framework .....	7
2.2 Google Colab .....	8
2.3 Flutter .....	10
2.4 Thuật toán Content-Based Filtering.....	16
2.5 Mô hình Convolutional Neural Network.....	17
CHƯƠNG 3. GIỚI THIỆU BỘ DỮ LIỆU .....	19

3.1 Tổng quan .....	19
3.2 Đặc điểm của dữ liệu .....	21
3.2.1 Mô tả thuộc tính id .....	21
3.2.2 Mô tả thuộc tính year.....	22
3.2.3 Mô tả thuộc tính gender.....	23
3.2.4 Mô tả thuộc tính masterCategory .....	23
3.2.5 Mô tả thuộc tính subCategory .....	24
3.2.6 Mô tả thuộc tính articleType .....	24
3.2.7 Mô tả thuộc tính baseColour .....	25
3.2.8 Mô tả thuộc tính season.....	25
3.2.9 Mô tả thuộc tính usage .....	26
3.2.10 Mô tả thuộc tính productDisplayName .....	26
CHƯƠNG 4: THỰC NGHIỆM .....	27
4.1 Thực nghiệm .....	27
4.1.1 Phương pháp xây dựng và huấn luyện mô hình CBF .....	27
4.1.2 Xây dựng và huấn luyện mô hình CNN .....	31
4.2 Kết quả đạt được .....	36
CHƯƠNG 5. XÂY DỰNG CHƯƠNG TRÌNH .....	39
5.1 Giao diện đăng ký tài khoản .....	39
5.2 Giao diện đăng nhập tài khoản .....	40
5.3 Giao diện trang chủ.....	41
5.4 Giao diện gợi ý sản phẩm .....	42
5.5 Giao diện tìm kiếm sản phẩm bằng hình ảnh .....	43
KẾT KUẬN.....	44
Kết quả đạt được .....	44
Hạn chế .....	44

Hướng phát triển .....	45
PHỤ LỤC .....	46
Phụ lục 1: Hướng dẫn cài đặt.....	46
Phụ lục 2: Hướng dẫn sử dụng.....	49
TÀI LIỆU THAM KHẢO.....	52

## DANH MỤC HÌNH ẢNH VÀ BẢNG BIỂU

Hình 2.1 Ảnh minh họa về Python .....	5
Hình 2.2 Ảnh minh họa về google colab .....	8
Hình 2.3 Các lớp kiến trúc .....	12
Hình 2.4 Cấu trúc của 1 ứng dụng .....	14
Hình 3.1 Mô tả thuộc tính id .....	21
Hình 3.2 Mô tả thuộc tính year .....	22
Hình 3.3 Mô tả thuộc tính gender .....	23
Hình 3.4 Mô tả thuộc tính masterCategory .....	23
Hình 3.5 Mô tả thuộc tính subCategory .....	24
Hình 3.6 Mô tả thuộc tính articleType .....	24
Hình 3.7 Mô tả thuộc tính baseColour .....	25
Hình 3.8 Mô tả thuộc tính season .....	25
Hình 3.9 Mô tả thuộc tính usage .....	26
Hình 3.10 Mô tả thuộc tính productDisplayName .....	26
Hình 4.1 Thông tin dữ liệu thiếu ban đầu .....	29
Hình 4.2 Kết quả đã xử lý các cột có dữ liệu thiếu .....	30
Hình 4.3 Xóa các cột không cần thiết .....	31
Hình 4.4 Kết quả sau khi xóa các cột không cần thiết .....	31
Hình 4.5 Ảnh sản phẩm trước khi tiền xử lý .....	32
Hình 4.6 Trạng thái dữ liệu hình ảnh trước khi tiền xử lý .....	33
Hình 4.7 Ảnh sản phẩm sau khi tiền xử lý .....	33
Hình 4.8 Biểu đồ so sánh hiệu năng 3 mô hình .....	34
Hình 4.9 Ma trận tương quan .....	34

<i>Hình 4.10 Biểu đồ phân bố điểm.....</i>	<i>35</i>
<i>Hình 4.11 Kết quả sau khi huấn luyện CBF.....</i>	<i>36</i>
<i>Hình 4.12 Kết quả sau khi huấn luyện mô hình CNN.....</i>	<i>37</i>
<i>Hình 5.1 Giao diện màn hình đăng ký.....</i>	<i>39</i>
<i>Hình 5.2 Giao diện màn hình đăng nhập.....</i>	<i>40</i>
<i>Hình 5.3 Giao diện trang chủ.....</i>	<i>41</i>
<i>Hình 5.4 Giao diện màn hình gợi ý sản phẩm.....</i>	<i>42</i>
<i>Hình 5.5 Giao diện tìm kiếm sản phẩm bằng hình ảnh.....</i>	<i>43</i>
<i>Bảng 3.1 Các đặc trưng của tập dữ liệu.....</i>	<i>20</i>

## BẢNG PHÂN CÔNG CÔNG VIỆC

Tên thành viên	Công việc được giao	Đánh giá	Ký tên
Đặng Thị Kim Thảo	Xây dựng giao diện chương trình. Huấn luyện và tích hợp mô hình CNN cho chức năng tìm kiếm sản phẩm bằng hình ảnh.	35%	
Lê Hoàng Phúc	Nghiên cứu và tìm hiểu cơ sở lý thuyết cho toàn bộ chương trình. Khảo sát yêu cầu, phân tích nghiệp vụ và thiết kế hệ thống cho toàn bộ chương trình.	30%	
Võ Thành Hoàng Phúc	Xây dựng Backend cho toàn bộ chương trình. Huấn luyện và tích hợp mô hình CBF cho chức năng gợi ý sản phẩm.	35%	

# CHƯƠNG 1: MỞ ĐẦU

## 1.1 Giới thiệu chung về đề tài

### 1.1.1 Bối cảnh và sự phát triển của thương mại điện tử và nhu cầu cá nhân hóa

Trong bối cảnh chuyển đổi số diễn ra mạnh mẽ trên toàn thế giới, thương mại điện tử trở thành một trong những lĩnh vực trọng điểm có tốc độ tăng trưởng nhanh nhất. Người tiêu dùng hiện đại ngày càng có xu hướng ưu tiên các nền tảng trực tuyến thay vì mua sắm truyền thống. Với sự phát triển của hạ tầng Internet, thiết bị di động và các hệ thống thanh toán điện tử, hành vi mua hàng của người dùng đã thay đổi sâu sắc, tạo nên một thị trường thương mại điện tử rộng lớn và cạnh tranh khốc liệt. [1]

Riêng tại Việt Nam, giai đoạn 2020–2024 đã chứng kiến sự bùng nổ của các nền tảng TMĐT như Shopee, Lazada, Tiki,... kéo theo việc hàng nghìn cửa hàng thời trang, phụ kiện chuyển sang bán hàng trực tuyến để tiếp cận khách hàng hiệu quả hơn. Các cửa hàng thời trang vừa và nhỏ cũng bắt đầu nhận thấy rằng nếu chỉ kinh doanh theo kiểu truyền thống thì khó cạnh tranh và khó duy trì lượng khách ổn định. Việc xây dựng một hệ thống bán hàng trực tuyến chuyên nghiệp không chỉ giúp mở rộng thị trường mà còn giúp giảm chi phí vận hành, tối ưu hóa quy trình xử lý đơn hàng và cung cấp trải nghiệm mua sắm tốt hơn. [2]

Tuy nhiên, cùng với sự phát triển mạnh mẽ đó, hành vi người dùng cũng trở nên phức tạp và khó dự đoán hơn. Khách hàng mong muốn được tương tác trong một môi trường đem đến cảm giác riêng tư và hiểu họ. Khái niệm cá nhân hóa trải nghiệm ngày càng quan trọng: người mua không muốn xem hàng loạt sản phẩm không liên quan mà muốn nhìn thấy những mặt hàng phù hợp đúng nhu cầu, đúng phong cách, đúng kích cỡ và sở thích của mình.

Đây chính là lý do mà các hệ thống thương mại điện tử hiện đại không chỉ dừng lại ở việc hiển thị sản phẩm, mà còn phải biết học hỏi từ hành vi khách hàng để đưa ra các gợi ý thông minh. Những nền tảng lớn coi AI và phân tích dữ liệu là chìa khóa để giữ chân khách hàng và tăng tỉ lệ mua hàng.

Trong khi đó, đa số cửa hàng thời trang vừa và nhỏ lại chưa có điều kiện triển khai các hệ thống hiện đại như vậy. Việc quản lý đơn hàng, sản phẩm, kho hàng, theo dõi doanh thu hay phân tích hành vi khách hàng hầu hết vẫn thực hiện thủ công hoặc sử dụng các



phần mềm rời rạc. Điều này tạo ra sự chậm trễ, sai sót và không có khả năng hỗ trợ ra quyết định dựa trên dữ liệu. [3]

Từ thực tế đó, nhóm quyết định xây dựng ứng dụng di động Stylo, vừa đáp ứng được các chức năng cơ bản của một hệ thống thương mại điện tử, vừa kết hợp thêm các yếu tố thông minh giúp nâng cao trải nghiệm người dùng và tối ưu hoạt động kinh doanh cho cửa hàng.

### **1.1.2 Sự cần thiết của việc ứng dụng trí tuệ nhân tạo và thuật toán tiên tiến**

Cùng với sự phát triển của công nghệ dữ liệu và học máy, Trí tuệ nhân tạo đã trở thành công nghệ cốt lõi trong thương mại điện tử toàn cầu. Các hệ thống AI hiện nay có khả năng phân tích lượng dữ liệu khổng lồ, học từ các mẫu hành vi của người dùng và đưa ra quyết định hoặc dự đoán với độ chính xác cao. Việc ứng dụng AI trong TMĐT không còn là xu hướng mà đã trở thành tiêu chuẩn để cạnh tranh. Một số lý do khiến việc tích hợp AI vào hệ thống bán hàng thời trang trở nên vô cùng cần thiết:

Cá nhân hóa trải nghiệm mua sắm, trí tuệ nhân tạo giúp hệ thống hiểu rõ từng khách hàng hơn thông qua: Sản phẩm họ xem và tương tác nhiều nhất, thói quen mua sắm theo thời điểm, màu sắc, kích cỡ, phong cách thời trang họ ưu tiên, các đánh giá và phản hồi của họ. Từ đó, hệ thống đưa ra danh sách sản phẩm gợi ý phù hợp, giúp: Tăng khả năng khách hàng đưa sản phẩm vào giỏ, rút ngắn thời gian tìm kiếm, tăng mức độ hài lòng, gia tăng doanh số.

Tự động hóa quy trình vận hành: Thay vì quản trị viên phải phân tích thủ công doanh thu, sản phẩm nổi bật hay tỉ lệ chuyển đổi, AI có thể: Tự động phân tích dữ liệu bán hàng, phát hiện xu hướng sản phẩm sắp hot, đề xuất nhập hàng, phát hiện bất thường trong hành vi mua hàng. Điều này giúp quản trị viên tiết kiệm thời gian và ra quyết định chính xác hơn. [4]

Tìm kiếm thông minh bằng hình ảnh: Nhiều khách hàng không biết phải mô tả sản phẩm bằng từ khóa như thế nào. Vì vậy, tìm kiếm bằng ảnh giúp: Người dùng chỉ cần chụp lại sản phẩm mình thích, hệ thống sử dụng CNN để tìm sản phẩm giống nhất trong cửa hàng, tăng hiệu quả tiếp cận sản phẩm ngay cả khi khách không biết cách gọi tên. [5]

## **1.2 Mục tiêu nghiên cứu**

### **1.2.1 Mục tiêu tổng quát**

Mục tiêu chính của đề tài là xây dựng một hệ thống bán hàng thời trang trực tuyến hoạt động trên nền tảng di động, thân thiện với người dùng, tích hợp đầy đủ chức năng từ tìm kiếm sản phẩm, xem chi tiết, giỏ hàng đến đặt hàng và đánh giá sản phẩm. Hệ thống đồng thời phải hỗ trợ quản trị viên quản lý sản phẩm, kho hàng, đơn hàng, người dùng và thống kê doanh thu một cách trực quan và dễ sử dụng.

Bên cạnh đó, đề tài hướng tới việc tích hợp các mô hình trí tuệ nhân tạo nhằm mang đến trải nghiệm cá nhân hóa, tăng hiệu quả kinh doanh.

### **1.2.2 Mục tiêu cụ thể**

#### **a. Mục tiêu về mặt kĩ thuật và kiến trúc hệ thống**

- Xây dựng backend theo kiến trúc ASP.NET Core Web API đảm bảo bảo mật, hiệu suất và dễ mở rộng.
- Thiết kế cơ sở dữ liệu SQL Server có cấu trúc rõ ràng, hỗ trợ truy vấn tối ưu.
- Phát triển frontend bằng Flutter với giao diện hiện đại, dễ dùng, đồng bộ trên Android.
- Tích hợp các mô hình AI: Gợi ý sản phẩm (CBF), tìm kiếm bằng ảnh (CNN)

#### **b. Mục tiêu về mặt chức năng và nghiệp vụ**

- Cung cấp đầy đủ quy trình mua sắm.
- Mang lại cho khách hàng trải nghiệm cá nhân hóa thông minh hơn.
- Tạo môi trường quản lý hiệu quả cho quản trị viên.
- Hỗ trợ thống kê chính xác để chủ cửa hàng ra quyết định.
- Đảm bảo hệ thống hoạt động mượt, an toàn, ổn định.

## **1.3 Phạm vi nghiên cứu**

### **1.3.1 Đối tượng áp dụng**

Hệ thống được xây dựng hướng tới: Khách hàng mua sắm thời trang trên thiết bị di động, quản trị viên cửa hàng, bộ phận xử lý dữ liệu và AI của hệ thống. Hệ thống tập trung vào quy mô cửa hàng vừa và nhỏ nhưng có khả năng mở rộng về lâu dài.

### **1.3.2 Phạm vi chức năng**

#### **a. Chức năng cơ bản của khách hàng**

- Đăng ký/đăng nhập.
- Xem sản phẩm theo danh mục.
- Tìm kiếm từ khóa, bộ lọc, hình ảnh.
- Xem chi tiết, đánh giá, lựa chọn kích cỡ – màu sắc.
- Thêm vào giỏ hàng, đặt hàng.
- Sản phẩm yêu thích.
- Gợi ý sản phẩm cá nhân hóa.

#### **b. Chức năng quản trị viên**

- Quản lý sản phẩm.
- Quản lý tồn kho.
- Quản lý đơn hàng.
- Quản lý người dùng.
- Dashboard thống kê.

#### **c. Chức năng ứng dụng của trí tuệ nhân tạo**

- Gợi ý sản phẩm thông minh.
- Tìm kiếm bằng hình ảnh.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Tổng quan về ngôn ngữ lập trình Python và FastApi

#### 2.1.1 Giới thiệu python

Python là một ngôn ngữ lập trình thông dịch, dễ đọc và dễ hiểu. Nền tảng nổi tiếng với cú pháp đơn giản và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. Điển hình như phát triển website, phân tích dữ liệu, trí tuệ nhân tạo và nhiều ứng dụng khác. Python có cú pháp linh hoạt và cấu trúc dữ liệu mạnh mẽ, công nghệ được hỗ trợ bởi một cộng đồng lớn. Điều này đã mang đến các thư viện và framework phong phú mà người dùng có thể sử dụng để xây dựng các ứng dụng phức tạp. Python cũng là một trong những ngôn ngữ phổ biến cho người mới học lập trình nhờ vào tính linh hoạt của nó.



Hình 2.1 Ảnh minh họa về Python

Những lợi ích của python:

- Cú pháp đơn giản và dễ đọc, phù hợp cho người mới học lập trình và cũng dễ dàng cho những người có kinh nghiệm.
- Python được sử dụng trong nhiều lĩnh vực như phát triển website, phân tích dữ liệu, trí tuệ nhân tạo, khoa học dữ liệu và nhiều ứng dụng khác. Điều này làm cho

Python trở thành một ngôn ngữ lập trình linh hoạt và tiện lợi.

- Cộng đồng Python khá lớn và luôn hoạt động tích cực nhằm cung cấp nhiều thư viện, framework hữu ích. Người dùng sẽ được nhận sự hỗ trợ từ cộng đồng thông qua tài liệu, diễn đàn và các nguồn thông tin trực tuyến.
- Python cung cấp nhiều thư viện mạnh mẽ cho phân tích dữ liệu như Pandas, NumPy, và Matplotlib. Bộ ngôn ngữ giúp người dùng xử lý và thể hiện dữ liệu một cách hiệu quả.
- Python có thể chạy trên nhiều hệ điều hành khác nhau và trở thành một trong những ngôn ngữ lập trình chéo phổ biến.
- Ngôn ngữ Python là mã nguồn mở và miễn phí, cho phép người dùng tự do sử dụng, phân phối và thay đổi phiên bản của nó.

Phương thức hoạt động chính của python

- Python có nhiều framework phát triển web phổ biến, ví dụ như Django, Flask, Pyramid và Fast API. Đây là các công cụ tối ưu giúp lập trình viên phát triển ứng dụng web từ phía máy chủ một cách dễ dàng và hiệu quả.
- Các framework này cung cấp nhiều tính năng như xử lý URL, tương tác cơ sở dữ liệu, quản lý phiên và tạo giao diện người dùng. Django được xem là một trong những framework phát triển web phổ biến nhất trong cộng đồng Python. Nền tảng cung cấp sẵn các tính năng linh hoạt và nhiều công cụ hữu ích giúp cho việc xây dựng ứng dụng web phức tạp trở nên thuận lợi hơn.
- Flask là một framework nhỏ gọn và linh hoạt hơn nên nó sẽ phù hợp cho việc xây dựng ứng dụng web từ nhỏ đến trung bình. Công nghệ cho phép người dùng tùy chỉnh nhiều hơn trên một hệ thống. Pyramid cũng là tiện ích cung cấp chế độ hoạt động mạnh mẽ, đặc biệt là việc xây dựng ứng dụng lớn và phức tạp.
- Fast API là một framework mới nhưng có tốc độ làm việc khá nhanh. Nền tảng được sử dụng để xây dựng các ứng dụng web hiệu suất cao và API. Tùy thuộc vào yêu cầu cụ thể của dự án mà người phát triển nên lựa chọn framework phù hợp với ngôn ngữ Python.

Ứng dụng trong khoa học dữ liệu: Ngôn ngữ Python được sử dụng phổ biến trong

lĩnh vực khoa học dữ liệu và máy học. Cách sử dụng thư viện và framework mạnh mẽ trong Python giúp cho việc phân tích dữ liệu hoặc triển khai mô hình lập trình trở nên dễ dàng hơn. [6]

Dưới đây là một số thư viện quan trọng mà người làm việc trong lĩnh vực này thường sử dụng:

- NumPy: Thư viện mạnh mẽ cho các phép toán trên mảng đa chiều và ma trận, cung cấp các hàm để thao tác dữ liệu số học một cách hiệu quả
- Pandas: Pandas cung cấp cấu trúc dữ liệu và công cụ phân tích dữ liệu mạnh mẽ, đặc biệt là trong việc làm việc với dữ liệu có cấu trúc như bảng và chuỗi thời gian
- Matplotlib và Seaborn: Đây là các thư viện hỗ trợ vẽ đồ thị và trực quan hóa dữ liệu một cách dễ dàng và mạnh mẽ
- Scikit-learn: Scikit-learn là một trong những thư viện học máy phổ biến nhất trong Python, cung cấp nhiều thuật toán lập trình và công cụ cho tiền xử lý dữ liệu, đánh giá mô hình và tinh chỉnh tham số
- TensorFlow và PyTorch: Đây là hai framework phổ biến cho việc triển khai mô hình học sâu (deep learning). Cả hai đều cung cấp API mạnh mẽ để xây dựng và huấn luyện mạng nơ-ron

Những thư viện này cung cấp hệ sinh thái để phân tích dữ liệu, xây dựng mô hình và triển khai ứng dụng trong lĩnh vực khoa học dữ liệu. Python được xem là ngôn ngữ lập trình hàng đầu trong lĩnh vực này. [7]

### **2.1.2 FastAPI Framework**

FastAPI là một framework phát triển web hiện đại được xây dựng trên nền tảng Python, chuyên dùng để xây dựng các Web API hiệu năng cao. FastAPI được thiết kế nhằm tối ưu tốc độ xử lý, đồng thời đảm bảo tính đơn giản, rõ ràng và dễ bảo trì trong quá trình phát triển ứng dụng. [8]

FastAPI tận dụng các tính năng mới của Python để tự động kiểm tra dữ liệu đầu vào, giúp giảm lỗi và tăng độ an toàn cho hệ thống. Nhờ đó, quá trình phát triển API trở nên nhanh chóng và hiệu quả hơn so với nhiều framework truyền thống.

Đặc điểm nổi bật:

- Hiệu năng cao
- Tự động sinh tài liệu API
- Kiểm tra và xác thực dữ liệu tự động
- Hỗ trợ bất đồng bộ

Vai trò:

- Triển khai dịch vụ tìm kiếm sản phẩm bằng hình ảnh
- Triển khai dịch vụ gợi ý sản phẩm thông minh

Lý do chọn:

- Phù hợp với các hệ thống có tích hợp mô hình học máy
- Dễ triển khai, dễ kiểm thử và dễ tích hợp
- Hỗ trợ tốt cho việc xây dựng API phục vụ mô hình học máy
- Có cộng đồng phát triển mạnh và tài liệu đầy đủ

## 2.2 Google Colab

Google Colab là một dịch vụ miễn phí từ Google cho phép bạn viết và chia sẻ mã Python. Người dùng có thể lưu trữ và chạy Notebook trên đám mây thông qua trình duyệt mà không cần kích hoạt cấu hình bộ đệm phức tạp cho máy tính cá nhân.



*Hình 2.2 Ảnh minh họa về google colab*

Nền tảng cũng cung cấp GPU và TPU để huấn luyện mô hình học và thực hiện các yêu cầu tính toán lớn. Điều này có tác dụng mở rộng sức mạnh xử lý trong điện toán và giảm thời gian chờ đợi khi thực hiện các nhiệm vụ tính toán nặng nề.

Google Colab được phát triển bởi nhóm Google Research nhằm hỗ trợ quá trình nghiên cứu và phát triển trong lĩnh vực học máy tính và khoa học dữ liệu. Công nghệ xuất phát từ nền tảng Jupyter Notebook phổ biến. Sau đó, Colab được kết hợp thêm khả năng tính toán trên tiện ích đám mây vô cùng mạnh mẽ của Google. [9]

Từ đó, quá trình phát triển Google Colab tập trung vào mục tiêu cung cấp một môi trường lập trình linh hoạt và mạnh mẽ. Nền tảng hỗ trợ đa dạng công nghệ tính toán như Python, CUDA, Tensorflow. Kèm theo đó có nhiều thư viện khoa học máy tính và khoa học dữ liệu khác. Đồng thời, dịch vụ còn cung cấp GPU và TPU để tăng cường sức mạnh tính toán.

Các chức năng chính của google colab

- Sử dụng Jupyter Notebooks trực tuyến: Google Colab cho phép tạo và chạy các Jupyter Notebooks trực tuyến mà không cần cài đặt môi trường phát triển phức tạp trên máy tính cá nhân. Giao diện sử dụng tương tự như Jupyter Notebook truyền thống với các cell cho phép thực thi mã Python hoặc viết markdown để tạo nội dung hướng dẫn.
- Khả năng chia sẻ và cộng tác: Người dùng có thể chia sẻ notebook với những người khác để cùng làm việc trên cùng một notebook, tạo điều kiện thuận lợi cho việc học tập và làm việc nhóm. Các tính năng như bình luận và chế độ chỉnh sửa đồng thời giúp tăng tính tương tác và hiệu quả của quá trình cộng tác.
- Dùng GPU và TPU miễn phí: Google Colab cung cấp truy cập miễn phí đến GPU và TPU, đặc biệt hữu ích cho các tác vụ tính toán nặng về mặt số học, đặc biệt là trong lĩnh vực học máy và deep learning. Việc sử dụng các card GPU hoặc TPU có sẵn giúp tăng tốc độ xử lý và huấn luyện mô hình so với việc sử dụng CPU thông thường.
- Lưu trữ dữ liệu trên Google Drive và tích hợp Google Cloud: Người dùng có thể truy cập và lưu trữ dữ liệu trực tiếp từ Google Drive, tạo điều kiện thuận lợi cho việc làm việc với tập tin dữ liệu lớn. Tích hợp với Google Cloud Platform (GCP)



cũng cho phép sử dụng các dịch vụ như BigQuery, Cloud Storage, và các API khác từ GCP trong quá trình làm việc

Ứng dụng của google colab:

- Học máy và khoa học dữ liệu: Google Colab là một công cụ mạnh mẽ cho việc học máy và nghiên cứu khoa học dữ liệu. Với khả năng sử dụng miễn phí GPU và TPU, người dùng có thể xây dựng, huấn luyện và kiểm định các mô hình máy học một cách hiệu quả. Google Colab cung cấp môi trường Jupyter Notebook trực tuyến, cho phép viết và chạy code Python, đồng thời cung cấp các thư viện phổ biến như TensorFlow, Keras, và scikit-learn.
- Phát triển ứng dụng AI và ML: Với sức mạnh của Colab's GPU và TPU, các nhà phát triển có thể tận dụng để phát triển ứng dụng trí tuệ nhân tạo (AI) và máy học (ML). Điều này bao gồm việc xây dựng và đánh giá các mô hình học máy, tạo ứng dụng dự đoán, xử lý ngôn ngữ tự nhiên, nhận diện hình ảnh và nhiều ứng dụng AI khác.
- Nghiên cứu và phân tích dữ liệu: Các nhà nghiên cứu và chuyên gia phân tích dữ liệu thường sử dụng Google Colab để thực hiện phân tích số liệu, xử lý dữ liệu lớn, và thực hiện các thí nghiệm khoa học. Colab cung cấp khả năng tích hợp dữ liệu từ Google Drive hoặc các nguồn khác, cho phép họ thực hiện các phân tích phức tạp mà không cần tải về dữ liệu về máy cục bộ. [10]
- Giáo dục và đào tạo: Google Colab cũng được sử dụng rộng rãi trong lĩnh vực giáo dục và đào tạo. Nó cung cấp môi trường lập trình Python trực tuyến mà không yêu cầu cài đặt, giúp sinh viên và giáo viên dễ dàng tiếp cận và chia sẻ các notebook học tập. Điều này làm tăng tính tương tác và khả năng học tập cộng đồng trong việc chia sẻ kiến thức và dự án.

## 2.3 Flutter

Flutter là một bộ công cụ giao diện người dùng đa nền tảng được thiết kế để cho phép tái sử dụng mã trên các hệ điều hành như iOS, Android, web và máy tính để bàn, đồng thời cho phép các ứng dụng giao tiếp trực tiếp với các dịch vụ nền tảng bên dưới. Mục tiêu là giúp các nhà phát triển tạo ra các ứng dụng hiệu suất cao, hoạt động mượt mà trên các nền tảng khác nhau, chấp nhận sự khác biệt khi cần thiết trong khi vẫn chia sẻ

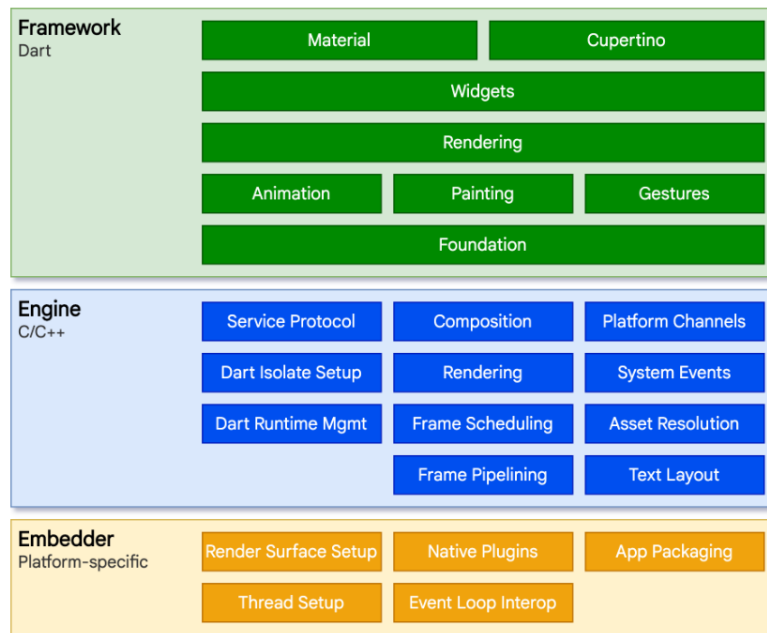
càng nhiều mã càng tốt.

Trong quá trình phát triển, ứng dụng Flutter chạy trên máy ảo (VM) cung cấp tính năng tải lại nóng (hot reload) có trạng thái mà không cần biên dịch lại toàn bộ. Khi phát hành, ứng dụng Flutter được biên dịch trực tiếp thành mã máy, có thể là lệnh Intel x64 hoặc ARM, hoặc thành JavaScript nếu nhắm mục tiêu đến web. Framework này là mã nguồn mở, với giấy phép BSD cho phép sử dụng rộng rãi, và có một hệ sinh thái phát triển mạnh mẽ gồm các gói bên thứ ba bổ sung chức năng cho thư viện cốt lõi.

Tổng quan này được chia thành nhiều phần:

- Mô hình lớp : Các thành phần cấu tạo nên Flutter.
- Giao diện người dùng tương tác : Một khái niệm cốt lõi trong phát triển giao diện người dùng Flutter.
- Giới thiệu về widget : Các khối xây dựng cơ bản của giao diện người dùng Flutter.
- Quá trình hiển thị : Flutter chuyển đổi mã giao diện người dùng thành các điểm ảnh như thế nào.
- Tổng quan về các trình nhúng nền tảng : Mã cho phép hệ điều hành di động và máy tính để bàn chạy các ứng dụng Flutter.
- Tích hợp Flutter với các mã nguồn khác : Thông tin về các kỹ thuật khác nhau có sẵn cho ứng dụng Flutter.
- Hỗ trợ cho web : Kết luận về các đặc điểm của Flutter trong môi trường web.

Flutter được thiết kế như một hệ thống phân lớp, có khả năng mở rộng. Nó tồn tại dưới dạng một chuỗi các thư viện độc lập, mỗi thư viện phụ thuộc vào lớp nền tảng bên dưới. Không có lớp nào có quyền truy cập đặc quyền vào lớp bên dưới, và mọi phần của khung phần mềm đều được thiết kế để tùy chọn và có thể thay thế.



*Hình 2.3 Các lớp kiến trúc*

Đối với hệ điều hành nền tảng, các ứng dụng Flutter được đóng gói theo cùng một cách như bất kỳ ứng dụng gốc nào khác. Một trình nhúng dành riêng cho nền tảng cung cấp điểm truy cập; phối hợp với hệ điều hành nền tảng để truy cập các dịch vụ như bề mặt hiển thị, khả năng truy cập và nhập liệu; và quản lý vòng lặp sự kiện thông báo. Trình nhúng được viết bằng ngôn ngữ phù hợp với nền tảng: hiện tại là Java và C++ cho Android, Swift và Objective-C/Objective-C++ cho iOS và macOS, và C++ cho Windows và Linux. Sử dụng trình nhúng, mã Flutter có thể được tích hợp vào một ứng dụng hiện có dưới dạng một mô-đun, hoặc mã đó có thể là toàn bộ nội dung của ứng dụng. Flutter bao gồm một số trình nhúng cho các nền tảng mục tiêu phổ biến, nhưng cũng tồn tại các trình nhúng khác. [11]

Cốt lõi của Flutter là công cụ Flutter (Flutter engine), chủ yếu được viết bằng C++ và hỗ trợ các thành phần cơ bản cần thiết để vận hành tất cả các ứng dụng Flutter. Công cụ này chịu trách nhiệm tạo ảnh bitmap cho các cảnh ghép nối mỗi khi cần vẽ một khung hình mới. Nó cung cấp triển khai cấp thấp của API cốt lõi của Flutter, bao gồm bố cục văn bản đồ họa, nhập/xuất tệp và mạng, môi trường chạy Dart và chuỗi công cụ biên dịch.

Công cụ này được tích hợp vào khung Flutter thông qua thư viện `dart:ui`, thư viện này bao bọc mã C++ cơ bản trong các lớp Dart. Thư viện này cung cấp các thành phần cơ

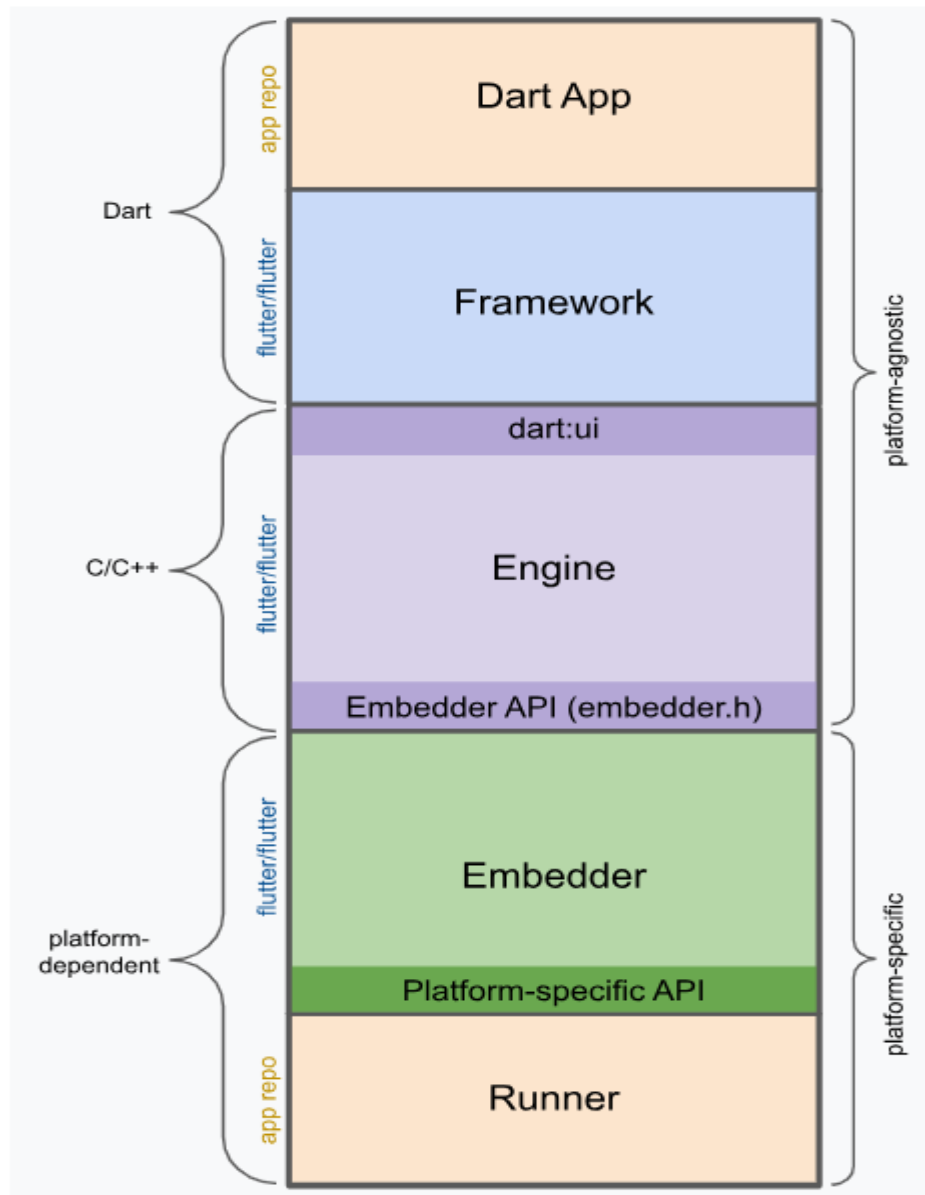
bản nhất, chẳng hạn như các lớp điều khiển hệ thống nhập liệu, đồ họa và hiển thị văn bản.

Thông thường, các nhà phát triển tương tác với Flutter thông qua framework Flutter, một framework hiện đại, phản ứng nhanh được viết bằng ngôn ngữ Dart. Nó bao gồm một bộ thư viện nền tảng, bố cục và cơ bản phong phú, được cấu thành từ nhiều lớp.

Từ dưới lên trên, ta có:

- Các lớp nền tảng cơ bản và các dịch vụ khối xây dựng như hoạt hình, vẽ và cử chỉ cung cấp các khái niệm trừu tượng thường được sử dụng trên nền tảng cơ bản đó.
- Lớp hiển thị cung cấp một lớp trừu tượng để xử lý bố cục. Với lớp này, bạn có thể xây dựng một cây các đối tượng có thể hiển thị. Bạn có thể thao tác các đối tượng này một cách linh hoạt, với cây tự động cập nhật bố cục để phản ánh những thay đổi của bạn.
- Lớp widget là một sự trừu tượng hóa về thành phần. Mỗi đối tượng hiển thị trong lớp hiển thị đều có một lớp tương ứng trong lớp widget. Ngoài ra, lớp widget cho phép bạn định nghĩa các tổ hợp lớp mà bạn có thể tái sử dụng. Đây là lớp mà mô hình lập trình phản ứng được giới thiệu.
- Các thư viện Material và Cupertino cung cấp bộ điều khiển toàn diện sử dụng các thành phần cơ bản của lớp widget để triển khai ngôn ngữ thiết kế Material hoặc iOS.

Sơ đồ sau đây cung cấp cái nhìn tổng quan về các thành phần tạo nên một ứng dụng Flutter thông thường được tạo ra bởi flutter create. Nó cho thấy vị trí của Flutter Engine trong cấu trúc này, làm nổi bật các ranh giới API và xác định các kho lưu trữ nơi các thành phần riêng lẻ được đặt. Chú thích bên dưới làm rõ một số thuật ngữ thường được sử dụng để mô tả các thành phần của một ứng dụng Flutter.



Hình 2.4 Cấu trúc của 1 ứng dụng

- Ứng dụng Dart: Sắp xếp các widget thành giao diện người dùng mong muốn, thực thi logic nghiệp vụ, thuộc sở hữu của nhà phát triển ứng dụng.
- Khung phần mềm: Cung cấp API cấp cao hơn để xây dựng các ứng dụng chất lượng cao (ví dụ: widget, kiểm thử va chạm, phát hiện cử chỉ, khả năng truy cập, nhập văn bản). Tổng hợp cây widget của ứng dụng thành một khung cảnh.

- Công cụ: Chịu trách nhiệm chuyển đổi các cảnh ghép thành ảnh bitmap. Cung cấp triển khai cấp thấp các API cốt lõi của Flutter (ví dụ: đồ họa, bố cục văn bản, môi trường chạy Dart). Nó cung cấp chức năng của mình cho framework bằng cách sử dụng API dart:ui. Tích hợp với một nền tảng cụ thể bằng cách sử dụng API Embedder của Engine.
- Trình nhúng: Phối hợp với hệ điều hành cơ bản để truy cập các dịch vụ như hiển thị bề mặt, khả năng truy cập và nhập liệu, quản lý vòng lặp sự kiện, cung cấp API dành riêng cho từng nền tảng để tích hợp Embedder vào các ứng dụng.
- Người chạy bộ: Công cụ này kết hợp các thành phần được cung cấp bởi API dành riêng cho nền tảng của Embedder thành một gói ứng dụng có thể chạy trên nền tảng mục tiêu. Một phần của mẫu ứng dụng được tạo bởi flutter create, thuộc sở hữu của nhà phát triển ứng dụng.

Trong trường hợp có thể, số lượng khái niệm thiết kế được giữ ở mức tối thiểu trong khi vẫn cho phép tổng vốn từ vựng lớn. Ví dụ, trong lớp widget, Flutter sử dụng cùng một khái niệm cốt lõi (Widget) để biểu diễn việc vẽ lên màn hình, bố cục (định vị và kích thước), tương tác người dùng, quản lý trạng thái, chủ đề, hoạt ảnh và điều hướng. Trong lớp hoạt ảnh, một cặp khái niệm, Animations và Tweens, bao phủ hầu hết không gian thiết kế. Trong lớp hiển thị, RenderObjects được sử dụng để mô tả bố cục, vẽ, kiểm tra va chạm và khả năng truy cập. Trong mỗi trường hợp này, vốn từ vựng tương ứng cuối cùng đều rất lớn: có hàng trăm widget và đối tượng hiển thị, và hàng chục loại hoạt ảnh và chuyển động trung gian.

Cấu trúc phân cấp lớp được thiết kế đơn giản và rộng rãi nhằm tối đa hóa số lượng kết hợp có thể, tập trung vào các widget nhỏ, có thể kết hợp với nhau, mỗi widget thực hiện tốt một chức năng duy nhất. Các tính năng cốt lõi được trừu tượng hóa, ngay cả các tính năng cơ bản như khoảng cách lề và căn chỉnh cũng được triển khai dưới dạng các thành phần riêng biệt thay vì được tích hợp vào lõi. (Điều này cũng trái ngược với các API truyền thống hơn, nơi các tính năng như khoảng cách lề được tích hợp vào lõi chung của mọi thành phần bố cục.) Vì vậy, ví dụ, để căn giữa một widget, thay vì điều chỉnh một thuộc tính giả định, bạn sẽ bọc nó trong một Center widget khác.

Có các widget dành cho khoảng cách lề, căn chỉnh, hàng, cột và lưới. Các widget bố cục này không có hình ảnh trực quan riêng. Thay vào đó, mục đích duy nhất của chúng là kiểm soát một khía cạnh nào đó trong bố cục của một widget khác. Flutter cũng bao gồm các widget tiện ích tận dụng cách tiếp cận theo thành phần này. [12]

## 2.4 Thuật toán Content-Based Filtering

Khái niệm: Thuật toán Content – Based Filtering (CBF) là thuật toán gợi ý sản phẩm dựa trên đặc điểm của sản phẩm và lịch sử hành vi của người dùng.

CBF phân tích: Thuộc tính sản phẩm: loại sản phẩm, kiểu, màu sắc, chất liệu, thương hiệu..., mô tả sản phẩm, từ khóa liên quan, lịch sử xem hoặc mua của người dùng. Sau đó tính độ tương đồng giữa: Sản phẩm mà người dùng tương tác và các sản phẩm khác trong cửa hàng. Kết quả là hệ thống đưa ra danh sách gợi ý có mức độ phù hợp cao.

Thuật toán gợi ý dựa trên nội dung CBF là một trong những phương pháp nền tảng của hệ thống gợi ý và được áp dụng rộng rãi trong thương mại điện tử, đặc biệt khi hệ thống cần hiểu rõ đặc điểm của từng sản phẩm để phục vụ mục tiêu cá nhân hóa trải nghiệm người dùng. Thuật toán này hoạt động dựa trên nguyên tắc phân tích nội dung sản phẩm, tức là xem xét những đặc trưng mô tả sản phẩm như màu sắc, chất liệu, kiểu dáng, phong cách, danh mục, mô tả chi tiết hoặc các thuộc tính liên quan khác, sau đó chuyển các đặc trưng này thành dạng vector số hóa bằng các kỹ thuật như Bag of Words, TF-IDF hoặc Word Embedding. Mỗi sản phẩm vì thế được biểu diễn như một điểm dữ liệu trong không gian nhiều chiều, mang theo toàn bộ “dấu hiệu nhận dạng” về nội dung của chính sản phẩm đó.

Trong khi đó, để đưa ra các gợi ý phù hợp cho người dùng, thuật toán cần xác định hồ sơ sở thích của họ. Hồ sơ này được hình thành từ lịch sử tương tác của người dùng, bao gồm các sản phẩm mà họ đã xem, đã nhấn yêu thích hoặc đã mua trong quá khứ. Những sản phẩm này cũng được chuyển thành vector đặc trưng, và hồ sơ người dùng chính là trung bình trọng số của các vector đó, phản ánh xu hướng phong cách hoặc sở thích đặc trưng của từng cá nhân. Sau khi có hồ sơ người dùng và đặc trưng của toàn bộ sản phẩm trong hệ thống, CBF sử dụng độ đo tương đồng – phổ biến nhất là Cosine Similarity – để tính toán mức độ giống nhau giữa hồ sơ người dùng với từng sản phẩm còn lại trong kho dữ liệu. Các sản phẩm có điểm tương đồng cao nhất sẽ được đề xuất cho người dùng.

Việc áp dụng CBF trong lĩnh vực thời trang mang lại nhiều hiệu quả vì sản phẩm thời trang thường có mô tả phong phú và mang tính phân loại cao. Khi người dùng xem một chiếc áo thun unisex form rộng, hệ thống có thể dễ dàng phân tích các từ khóa quan trọng như form rộng, unisex, streetwear, cotton, và tìm ra các sản phẩm tương tự trong hệ thống có đặc trưng kỹ thuật gần giống, từ đó tạo ra danh sách gợi ý mang tính cá nhân hóa cao. Điều này giúp người dùng tiếp cận nhanh chóng những sản phẩm phù hợp với phong cách của mình mà không cần mất thời gian tìm kiếm thủ công. Ngoài ra, CBF còn có ưu điểm lớn ở khả năng hoạt động hiệu quả ngay cả khi hệ thống mới triển khai và lượng người dùng chưa nhiều, bởi thuật toán không phụ thuộc vào dữ liệu của người dùng khác như các phương pháp Collaborative Filtering. Chính vì vậy, việc áp dụng CBF trong đề tài mang ý nghĩa quan trọng, giúp hệ thống xây dựng khả năng hiểu nội dung sản phẩm và tự động đưa ra những gợi ý thông minh, phù hợp với sở thích của từng khách hàng dựa trên hành vi tương tác của họ. [13]

## **2.5 Mô hình Convolutional Neural Network**

Mô hình Convolutional Neural Network (CNN) là một trong những kiến trúc mạng nơ-ron sâu mang tính đột phá, được thiết kế chuyên biệt để xử lý dữ liệu hình ảnh và tự động học các đặc trưng trực quan mà không cần con người phải lập trình thủ công. CNN được ứng dụng rộng rãi trong nhiều lĩnh vực như nhận diện khuôn mặt, phân loại ảnh, phát hiện vật thể và đặc biệt phù hợp với bài toán tìm kiếm sản phẩm bằng hình ảnh trong thương mại điện tử. Điểm mạnh của CNN nằm ở khả năng trích xuất đặc trưng một cách tự động và hiệu quả. Khi một hình ảnh được đưa vào mô hình, các lớp tích chập sẽ thực hiện việc “quét” trên ảnh bằng các bộ lọc (filter) nhỏ để phát hiện ra những đặc trưng cơ bản như đường biên, cạnh, họa tiết hay sự thay đổi ánh sáng. Khi đi sâu vào mô hình, các lớp tiếp theo tiếp tục học các đặc trưng phức tạp hơn như form dáng quần áo, cấu trúc họa tiết, kiểu cổ áo, chất liệu vải hoặc phong cách tổng thể của trang phục.

Sau quá trình trích xuất đặc trưng, CNN tạo ra một vector đặc trưng đại diện cho hình ảnh đầu vào. Vector này có thể được dùng để so sánh với các vector hình ảnh sản phẩm có sẵn trong cơ sở dữ liệu để tìm ra sản phẩm có mức độ tương đồng cao nhất. Việc so sánh mục tiêu này thường sử dụng các thước đo như Euclidean Distance hoặc Cosine



Similarity. Sản phẩm có vector gần nhất với ảnh input sẽ được xem là sản phẩm giống nhất và hiển thị cho người dùng. Điều này đặc biệt hữu ích trong ngành thời trang – nơi mà hình ảnh đóng vai trò quan trọng hơn nhiều so với mô tả bằng văn bản. Người dùng thường khó mô tả bằng từ khóa các chi tiết như “kiểu sọc caro mảnh pha màu pastel” hoặc dáng váy xếp tầng nhẹ, có bèo trước ngực, nhưng CNN có thể nhận ra các đặc điểm này một cách tự nhiên chỉ dựa trên ảnh. [14]

Việc áp dụng CNN vào hệ thống tìm kiếm của ứng dụng Stylo không chỉ giúp nâng cao trải nghiệm người dùng mà còn tạo ra sự tiện lợi vượt trội. Người dùng có thể bắt gặp một kiểu quần áo yêu thích khi xem video, đọc bài đăng trên mạng xã hội hay nhìn thấy ngoài đời thật và chỉ cần chụp hoặc tải lên hình ảnh đó, hệ thống sẽ chủ động tìm kiếm các sản phẩm tương tự trong cửa hàng. Điều này giúp rút ngắn quá trình tìm kiếm và làm tăng khả năng tiếp cận sản phẩm mong muốn một cách nhanh chóng. Không chỉ vậy, CNN còn mang lại tính năng hiện đại và tạo nên lợi thế cạnh tranh đáng kể cho ứng dụng so với các hệ thống bán hàng truyền thống.

Tổng quan lại, mô hình CNN và thuật toán CBF đóng vai trò quan trọng trong việc tạo nên hệ thống gợi ý và tìm kiếm thông minh của ứng dụng Stylo. CNN giúp hệ thống hiểu được hình ảnh sản phẩm, còn CBF giúp hệ thống hiểu được nội dung và mối quan hệ giữa các sản phẩm bên trong ứng dụng. Sự kết hợp này giúp tạo nên một nền tảng bán hàng thời trang trực tuyến vừa trực quan, vừa cá nhân hóa, đáp ứng tốt xu hướng công nghệ hiện đại và mang lại trải nghiệm tối ưu cho người dùng.

## CHƯƠNG 3. GIỚI THIỆU BỘ DỮ LIỆU

### 3.1 Tổng quan

- Bộ dữ liệu được sử dụng trong đề án có tên là Fashion Product Images, được cung cấp công khai trên nền tảng Kaggle – một kho dữ liệu lớn và uy tín, thường xuyên được sử dụng trong các nghiên cứu và bài toán thực tế liên quan đến học máy và trí tuệ nhân tạo. Bộ dữ liệu này được xây dựng nhằm phục vụ cho các bài toán phân loại, tìm kiếm và gợi ý sản phẩm trong lĩnh vực thời trang, do đó rất phù hợp với mục tiêu nghiên cứu của đề tài. Với quy mô lớn, dữ liệu đa dạng và có tính thực tế cao, Fashion Product Images cho phép mô phỏng tương đối chính xác hành vi tìm kiếm và mua sắm của người dùng trong các hệ thống thương mại điện tử hiện đại.
- Về cấu trúc, bộ dữ liệu bao gồm hai thành phần chính là dữ liệu hình ảnh và dữ liệu mô tả (metadata). Phần dữ liệu hình ảnh chứa hàng chục nghìn ảnh sản phẩm thời trang với chất lượng tương đối tốt, được chụp trên nền trắng, giúp làm nổi bật sản phẩm và giảm nhiễu trong quá trình xử lý ảnh. Các hình ảnh này đại diện cho nhiều loại sản phẩm khác nhau như quần áo, giày dép, phụ kiện dành cho cả nam và nữ, với nhiều kiểu dáng, màu sắc và phong cách khác nhau. Điều này giúp mô hình học sâu, đặc biệt là các mô hình CNN, có đủ dữ liệu để học được các đặc trưng thị giác quan trọng như hình dạng, màu sắc, hoa văn và cấu trúc của sản phẩm.
- Bên cạnh dữ liệu hình ảnh, bộ dữ liệu còn cung cấp thông tin mô tả chi tiết cho từng sản phẩm dưới dạng các thuộc tính có cấu trúc, bao gồm mã sản phẩm, tên sản phẩm, danh mục, nhóm đối tượng sử dụng (nam, nữ, trẻ em), màu sắc chủ đạo, loại sản phẩm và một số thông tin phân loại khác. Các đặc trưng này đóng vai trò quan trọng trong các bài toán gợi ý dựa trên nội dung (Content-Based Filtering), giúp hệ thống phân tích sự tương đồng giữa các sản phẩm dựa trên thuộc tính thay vì chỉ dựa vào hành vi người dùng. Việc kết hợp cả dữ liệu hình ảnh và dữ liệu mô tả giúp tăng độ chính xác và tính linh hoạt cho hệ thống gợi ý.
- Bộ dữ liệu Fashion Product Images có đầy đủ các yếu tố cần thiết để triển khai và đánh giá các thuật toán trong đề án, từ việc xây dựng mô hình tìm kiếm sản phẩm bằng hình ảnh cho đến hệ thống gợi ý sản phẩm thời trang. Tính đa dạng, quy mô lớn và cấu trúc rõ ràng của bộ dữ liệu không chỉ giúp quá trình huấn luyện mô hình

đạt hiệu quả cao mà còn góp phần đảm bảo tính thực tiễn và khả năng mở rộng của hệ thống trong các ứng dụng thương mại điện tử thực tế. [15]

- Link dataset: <https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-small>
- Số lượng thông của dữ liệu là 44, 446 dòng.
- Các đặc trưng của tập dữ liệu:

Tên đặc trưng	Kiểu dữ liệu	Mô tả
id	int	ID sản phẩm duy nhất.
gender	Object	Giới tính của đối tượng sử dụng sản phẩm.
masterCategory	Object	Danh mục sản phẩm cấp cao.
subCategory	Object	Danh mục sản phẩm phụ.
articleType	Object	Loại mặt hàng cụ thể.
baseColour	Object	Màu cơ bản của sản phẩm.
season	Object	Mùa được thiết kế cho sản phẩm.
year	int	Năm sản phẩm được phát hành hoặc liệt kê.
usage	Object	Mục đích sử dụng của sản phẩm.
productDisplayName	Object	Tên đầy đủ, dễ đọc của sản phẩm.

*Bảng 3.1 Các đặc trưng của tập dữ liệu*

## 3.2 Đặc điểm của dữ liệu

### 3.2.1 Mô tả thuộc tính id

	id
count	44424.000000
mean	29696.334301
std	17049.490518
min	1163.000000
25%	14768.750000
50%	28618.500000
75%	44683.250000
max	60000.000000

Hình 3.1 Mô tả thuộc tính id

- Số lượng dữ liệu (count): Có 44,424 mẫu dữ liệu trong thuộc tính.
- Giá trị trung bình (mean): Giá trị trung bình của ID là 29,696.33
- Độ lệch chuẩn (std): Độ lệch chuẩn là 17,049.49, cho thấy độ biến động rất lớn trong các giá trị ID. Điều này là hoàn toàn bình thường và được mong đợi đối với một thuộc tính định danh (ID) có phạm vi rộng
- Giá trị nhỏ nhất (min): ID nhỏ nhất là 1,163.00
- Tứ phân vị thứ nhất (25%): 25% giá trị ID nhỏ hơn hoặc bằng 14,768.75
- Trung vị (50%): 50% giá trị ID nhỏ hơn hoặc bằng 28,618.50. Đây là giá trị nằm chính giữa khi sắp xếp dữ liệu theo thứ tự tăng dần
- Tứ phân vị thứ ba (75%): 75% giá trị ID nhỏ hơn hoặc bằng 44,683.25
- Giá trị lớn nhất (max): Giá trị lớn nhất của ID là 60,000.00

### 3.2.2 Mô tả thuộc tính year

year	
count	44423.000000
mean	2012.806497
std	2.126480
min	2007.000000
25%	2011.000000
50%	2012.000000
75%	2015.000000
max	2019.000000

Hình 3.2 Mô tả thuộc tính year

- Số lượng dữ liệu (count): Có 44,423 mẫu dữ liệu trong thuộc tính
- Giá trị trung bình (mean): Giá trị trung bình của year là 2012.81
- Độ lệch chuẩn (std): Độ lệch chuẩn là 2.13, cho thấy độ biến động rất thấp. Các giá trị năm rất tập trung và đồng nhất xung quanh giá trị trung bình
- Giá trị nhỏ nhất (min): year nhỏ nhất là 2007
- Tứ phân vị thứ nhất (25%): 25% giá trị year nhỏ hơn hoặc bằng 2011
- Trung vị (50%): 50% giá trị year nhỏ hơn hoặc bằng 2012.00. Đây là năm nằm giữa khi sắp xếp dữ liệu
- Tứ phân vị thứ ba (75%): 75% giá trị year nhỏ hơn hoặc bằng 2015.00
- Giá trị lớn nhất (max): Giá trị lớn nhất của year là 2019.00

### 3.2.3 Mô tả thuộc tính gender

gender	
count	44424
unique	5
top	Men
freq	22147

Hình 3.3 Mô tả thuộc tính gender

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 5
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Men
- Số lần giá trị phổ biến nhất xuất hiện: 22,147 lần

### 3.2.4 Mô tả thuộc tính masterCategory

masterCategory	
count	44424
unique	7
top	Apparel
freq	21397

Hình 3.4 Mô tả thuộc tính masterCategory

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 7
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Apparel
- Số lần giá trị phổ biến nhất xuất hiện: 21397 lần

### 3.2.5 Mô tả thuộc tính subCategory

subCategory	
count	44424
unique	45
top	Topwear
freq	15402

Hình 3.5 Mô tả thuộc tính subCategory

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 45
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Topwear
- Số lần giá trị phổ biến nhất xuất hiện: 15402 lần

### 3.2.6 Mô tả thuộc tính articleType

articleType	
count	44424
unique	143
top	Tshirts
freq	7067

Hình 3.6 Mô tả thuộc tính articleType

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 143
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Tshirts
- Số lần giá trị phổ biến nhất xuất hiện: 7076 lần

### 3.2.7 Mô tả thuộc tính baseColour

baseColour	
count	44409
unique	46
top	Black
freq	9728

Hình 3.7 Mô tả thuộc tính baseColour

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 46
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Black
- Số lần giá trị phổ biến nhất xuất hiện: 9728 lần

### 3.2.8 Mô tả thuộc tính season

season	
count	44403
unique	4
top	Summer
freq	21472

Hình 3.8 Mô tả thuộc tính season

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 4
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Summer
- Số lần giá trị phổ biến nhất xuất hiện: 21472 lần



### 3.2.9 Mô tả thuộc tính usage

usage	
count	44107
unique	8
top	Casual
freq	34406

Hình 3.9 Mô tả thuộc tính usage

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 8
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Casual
- Số lần giá trị phổ biến nhất xuất hiện: 34406 lần

### 3.2.10 Mô tả thuộc tính productDisplayName

productDisplayName	
count	44417
unique	31121
top	Lucera Women Silver Earrings
freq	82

Hình 3.10 Mô tả thuộc tính productDisplayName

- Tổng số mẫu dữ liệu có trong thuộc tính: 44,424 mẫu
- Số lượng giá trị duy nhất của thuộc tính: 31121
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: Lucera Women Silver Earrings
- Số lần giá trị phổ biến nhất xuất hiện: 82 lần

## CHƯƠNG 4: THỰC NGHIỆM

### 4.1 Thực nghiệm

#### 4.1.1 Phương pháp xây dựng và huấn luyện mô hình CBF

##### a. Phương pháp xây dựng

Trong phạm vi đề án Học Máy Cơ Bản, việc sử dụng kỹ thuật Term Frequency - Inverse Document Frequency (TF-IDF) kết hợp cùng phép đo Cosine Similarity hoàn toàn được xác lập là một quá trình huấn luyện mô hình dựa trên các cơ sở sau:

- Học không giám sát: Khác với học có giám sát cần nhãn, mô hình CBF trong đề án này tự tìm ra các cấu trúc ẩn và mối quan hệ giữa các sản phẩm thông qua dữ liệu thuộc tính.
- Quá trình học đặc trưng: Khi thực hiện `tfidf.fit()`, máy tính thực sự đang thực hiện quá trình học bộ từ điển và tính toán trọng số IDF để xác định tầm quan trọng của từng thuộc tính.
- Mô hình hóa không gian Vector: Kết quả của quá trình huấn luyện là một ma trận vector đa chiều. Đây chính là tri thức của mô hình, cho phép hệ thống biểu diễn bất kỳ sản phẩm nào dưới dạng các tọa độ toán học để so sánh.

Một trong những điểm đặc thù của hệ thống gợi ý dựa trên nội dung được triển khai trong đề án này là việc sử dụng toàn bộ tập dữ liệu (44,446 bản ghi) thay vì chia nhỏ thành các tập huấn luyện/kiểm tra.

Việc chia tập dữ liệu truyền thống không được áp dụng trong Pipeline này vì các lý do kỹ thuật sau:

- Tính chất Lazy Learning: Thuật toán CBF không tối ưu hóa các trọng số thông qua các vòng lặp mà dựa trên phép tính tương đồng hình học trực tiếp. Do đó, việc chia tập Test để đo độ lỗi không mang ý nghĩa toán học trong bối cảnh này.
- Bảo toàn không gian tham chiếu: Trong hệ thống gợi ý, mỗi sản phẩm đều đóng vai trò là một ứng viên. Nếu chia dữ liệu và loại bỏ 20% khỏi quá trình tính toán, hệ thống sẽ bị mất đi 20% cơ hội tìm thấy các sản

phẩm tương đồng nhất, làm giảm hiệu quả thực tế của ứng dụng.

- Đánh giá nội tại: Thay vì dùng Accuracy/Precision chúng em sử dụng chỉ số Intra-list Diversity (ILD) để đánh giá chất lượng mô hình dựa trên sự phân bố của các vector trong danh sách kết quả.

Việc sử dụng toàn bộ quy mô dữ liệu giúp xây dựng một Corpus toàn diện nhất. Điều này đảm bảo trọng số TF-IDF được tính toán khách quan trên toàn hệ thống, giúp mô hình nhận diện chính xác độ hiếm và độ đặc trưng của từng sản phẩm.

Mô hình triển khai không chỉ đơn thuần là các công thức toán học mà là sự kết hợp của nhiều kỹ thuật cốt lõi trong học máy:

- Tiền xử lý dữ liệu: Xử lý dữ liệu khuyết thiếu bằng Mode Imputation.
- Kỹ nghệ đặc trưng (Feature Engineering): Xây dựng "Metadata Soup" để tối ưu hóa không gian đặc trưng đầu vào.
- Giảm chiều dữ liệu: Tối ưu hóa mô hình thông qua số `max_features = 3000` để tăng hiệu suất tính toán.
- Đo lường độ tương đồng: Áp dụng Cosine Similarity - một nền tảng toán học quan trọng trong các thuật toán láng giềng gần nhất.

## b. Tiền xử lí dữ liệu

- Xử lí giá trị thiếu: nhóm dữ liệu đầy đủ : Các cột id, gender, masterCategory, subCategory, và articleType hoàn toàn sạch, không cần xử lý thêm. Nhóm dữ liệu đang thiếu cần xử lí productDisplayName: thiếu 7 giá trị, baseColour: thiếu 15 giá trị, usage: thiếu 317 giá trị.

	<b>0</b>
<b>id</b>	0
<b>gender</b>	0
<b>masterCategory</b>	0
<b>subCategory</b>	0
<b>articleType</b>	0
<b>baseColour</b>	15
<b>usage</b>	317
<b>productDisplayName</b>	7

**dtype: int64**

*Hình 4.1 Thông tin dữ liệu thiếu ban đầu*

- Sau khi đã xử lí giá trị thiếu xong thì ta dễ dàng quan sát được ảnh bên dưới không còn giá trị ở cột nào thiếu. Ta thấy sinh ra thêm 1 cột soup ở đây là do cột đó kết hợp các thuộc tính gender, masterCategory, subCategory, articleType, productDisplayName, baseColour và usage.

	0
id	0
gender	0
masterCategory	0
subCategory	0
articleType	0
baseColour	0
usage	0
productDisplayName	0
soup	0

**dtype:** int64

*Hình 4.2 Kết quả đã xử lí các cột có dữ liệu thiếu*

- Xóa các cột không cần thiết:

	id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName
0	15970	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	2011.0	Casual	Turtle Check Men Navy Blue Shirt
1	39386	Men	Apparel	Bottomwear	Jeans	Blue	Summer	2012.0	Casual	Peter England Men Party Blue Jeans
2	59263	Women	Accessories	Watches	Watches	Silver	Winter	2016.0	Casual	Titan Women Silver Watch
3	21379	Men	Apparel	Bottomwear	Track Pants	Black	Fall	2011.0	Casual	Manchester United Men Solid Black Track Pants
4	53759	Men	Apparel	Topwear	Tshirts	Grey	Summer	2012.0	Casual	Puma Men Grey T-shirt

*Hình 4.3 Xóa các cột không cần thiết*

- Kết quả bên dưới cho thấy đã xóa 2 cột year và seson lí do xóa 2 cột này là vì cột year tính theo thời điểm năm sản xuất hoặc năm ra mắt thường không phản ánh sở thích cốt lõi của người dùng. Một chiếc áo phông từ năm 2011 vẫn có thể tương đồng về kiểu dáng với năm 2025. Nếu giữ lại, mô hình có thể hiểu nhầm rằng người dùng chỉ thích các sản phẩm của một năm nhất định, làm giảm khả năng gợi ý các sản phẩm mới hoặc cổ điển. Cột seson tính nhất thời Mặc dù mùa ảnh hưởng đến việc mua sắm, nhưng trong hệ thống gợi ý dựa trên nội dung (Content-based filtering), chúng ta tập trung vào đặc điểm định danh của món đồ hơn là thời gian mặc nó.

	id	gender	masterCategory	subCategory	articleType	baseColour	usage	productDisplayName
0	15970	Men	Apparel	Topwear	Shirts	Navy Blue	Casual	Turtle Check Men Navy Blue Shirt
1	39386	Men	Apparel	Bottomwear	Jeans	Blue	Casual	Peter England Men Party Blue Jeans
2	59263	Women	Accessories	Watches	Watches	Silver	Casual	Titan Women Silver Watch
3	21379	Men	Apparel	Bottomwear	Track Pants	Black	Casual	Manchester United Men Solid Black Track Pants
4	53759	Men	Apparel	Topwear	Tshirts	Grey	Casual	Puma Men Grey T-shirt

*Hình 4.4 Kết quả sau khi xóa các cột không cần thiết*

#### 4.1.2 Xây dựng và huấn luyện mô hình CNN

##### a. Phương pháp xây dựng

Kiến trúc mô hình chúng em sử dụng phương pháp Transfer Learning với mô hình ResNet50 đã được huấn luyện sẵn trên tập dữ liệu ImageNet. ResNet50 được cấu hình với tham số `include_top=False` và `pooling='avg'`. Loại bỏ lớp phân loại cuối cùng để biến mạng CNN thành một bộ trích xuất đặc trưng. Đầu ra thu được là một vector đặc trưng phẳng (global average pooled feature vector) đại diện cho nội dung hình ảnh thay vì nhãn lớp. Tìm kiếm tương đồng với FAISS khởi tạo index

chúng em sử dụng thư viện FAISS (Facebook AI Similarity Search) với IndexFlatL2. Thuật toán sử dụng khoảng cách L2 (Euclidean Distance) để đo lường mức độ giống nhau giữa các vector. Khoảng cách giữa hai vector càng nhỏ thì hai hình ảnh càng tương đồng về đặc điểm thị giác (màu sắc, kiểu dáng, hoa văn). FAISS cho phép thực hiện truy vấn hàng triệu vector trong thời gian thực, điều mà các vòng lặp truyền thống không làm được.

Quy trình tìm kiếm thực tế:

- FAISS cho phép thực hiện truy vấn hàng triệu vector trong thời gian thực, điều mà các vòng lặp truyền thống không làm được.
- Vector này được đưa vào `index.Search()` của FAISS để tìm ra top k láng giềng gần nhất.
- Hệ thống trả về file và hiển thị các hình ảnh có độ tương đồng cao nhất lên màn hình.

#### **b. Tiền xử lý**

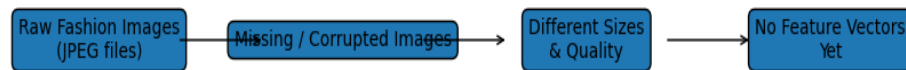
- Xử lý dữ liệu cho chức năng tìm kiếm hình ảnh: trước khi tiến hành tiền xử lý, tập dữ liệu hình ảnh tồn tại một số vấn đề sau: Hình ảnh có kích thước không đồng nhất, một số ảnh bị lỗi, hỏng hoặc không thể đọc được, dữ liệu ảnh chưa được chuẩn hóa để làm đầu vào cho mô hình học sâu, chưa tồn tại biểu diễn số (feature vector) cho việc so sánh độ tương đồng.

Before preprocessing  
(Raw image)



*Hình 4.5 Ảnh sản phẩm trước khi tiền xử lý*

- Các bước tiền xử lý dữ liệu hình ảnh: Lọc ảnh lỗi hệ thống tiến hành kiểm tra và loại bỏ các hình ảnh không tồn tại hoặc không thể mở được nhằm đảm bảo chỉ các ảnh hợp lệ mới được đưa vào pipeline xử lý. Chuẩn hóa kích thước ảnh tất cả các hình ảnh hợp lệ được resize về kích thước cố định  $224 \times 224$  pixel, phù hợp với yêu cầu đầu vào của các mô hình CNN pretrained như EfficientNet, ResNet và DenseNet. Chuẩn hóa giá trị pixel giá trị pixel của ảnh được chuẩn hóa theo hàm preprocess\_input tương ứng với từng mô hình CNN. Bước này giúp phân bố dữ liệu đầu vào phù hợp với quá trình huấn luyện ban đầu của mô hình pretrained, từ đó cải thiện chất lượng trích xuất đặc trưng.



*Hình 4.6 Trạng thái dữ liệu hình ảnh trước khi tiền xử lý*

- Sau khi hoàn tất các bước tiền xử lý, dữ liệu hình ảnh đạt được các đặc điểm sau: Tất cả hình ảnh có kích thước đồng nhất, giá trị pixel đã được chuẩn hóa, ảnh sẵn sàng làm đầu vào cho mô hình CNN để trích xuất đặc trưng, đảm bảo tính nhất quán trong toàn bộ tập dữ liệu.

After preprocessing  
(Resize + Normalize)



*Hình 4.7 Ảnh sản phẩm sau khi tiền xử lý*

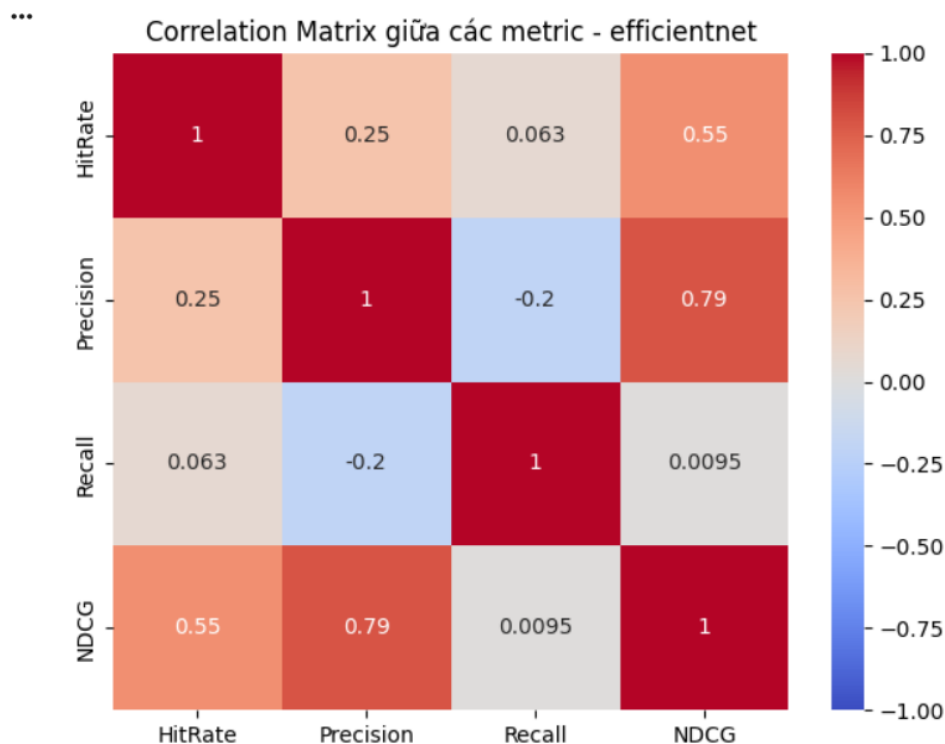


- Quy trình tìm kiếm hình ảnh tương tự dựa trên mạng CNN:



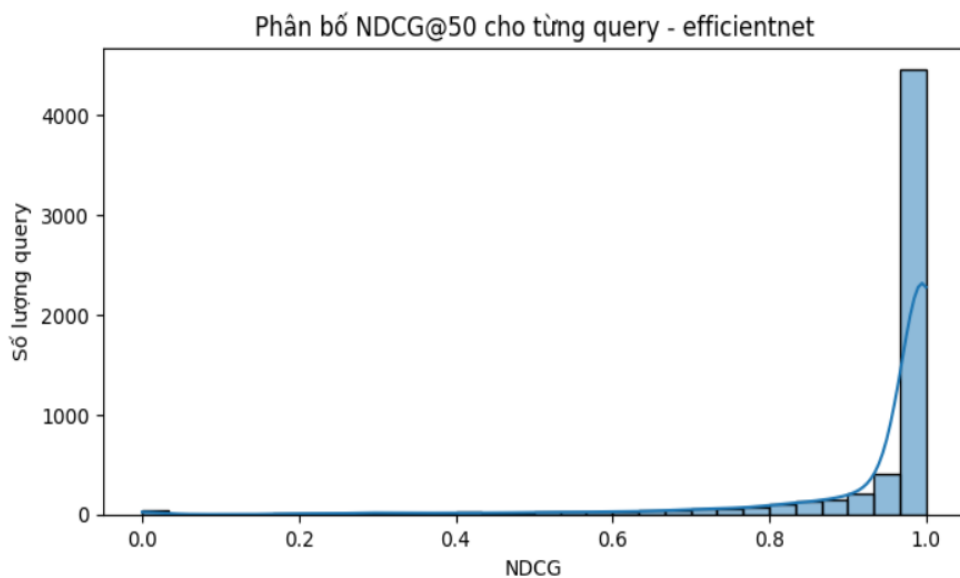
Hình 4.8 Biểu đồ so sánh hiệu năng 3 mô hình

- Ma trận tương quan giữa các độ đo đánh giá mô hình gợi ý: Ma trận này chứa đựng các đặc điểm cốt lõi của sản phẩm như kiểu dáng, màu sắc, chất liệu và cấu trúc không gian đã được mô hình ResNet50 trích xuất. Mỗi hàng trong ma trận là một tọa độ trong không gian 2048 chiều (nếu bạn dùng ResNet50 không có lớp top). Những sản phẩm nhìn giống nhau sẽ có các tọa độ nằm gần nhau trong ma trận này. Khi bạn đưa vào một ảnh mới, FAISS sẽ chuyển ảnh đó thành một vector và tính khoảng cách L2 (Euclidean) hoặc Cosine Similarity giữa vector đó với tất cả các hàng trong ma trận này. FAISS thực hiện các phép toán ma trận cực nhanh trên ma trận này để lọc ra các dòng (các sản phẩm) có khoảng cách nhỏ nhất, tức là những sản phẩm có hình ảnh tương đồng nhất.



Hình 4.9 Ma trận tương quan

- Biểu đồ phân bố điểm cho từng truy vấn, sản phẩm được gợi ý càng giống với ảnh gốc. Đây là mục tiêu của hệ thống tìm kiếm sản phẩm bằng hình ảnh.



Hình 4.10 Biểu đồ phân bố điểm

## 4.2 Kết quả đạt được

- Kết quả sau khi huấn luyện mô hình CBF:

```
=====
BÁO CÁO GỢI Ý ĐA DẠNG CHO SẢN PHẨM: 15970
=====

  id masterCategory subCategory productDisplayName
26396 Apparel Topwear Turtle Men Check Navy Blue Shirt
33554 Apparel Topwear Wrangler Men Exeter Check Navy Blue Shirt
16013 Accessories Scarves Turtle Check Men Navy Blue Scarf
51396 Apparel Bottomwear Myntra Men Navy Blue Check Shorts
27628 Apparel Bottomwear Scullers Men Check Navy Blue Shorts
54519 Footwear Shoes F Sports Men Navy Blue Vito Shoes
25447 Footwear Shoes Nike Men Casual Navy Blue Shoes
23100 Accessories Accessories Turtle Men Formal Navy Blue Tie
8373 Accessories Bags Fastrack Men Navy Blue Backpack
31415 Apparel Innerwear Spykar Men Navy Blue Trevor Brief

>> Tổng số Sub-Categories: 7
>> Chỉ số Độ Đa dạng (Diversity Score): 0.5022

-----
PHÂN BỐ DANH MỤC:
- Topwear: 2 sản phẩm (20.0%)
- Bottomwear: 2 sản phẩm (20.0%)
- Shoes: 2 sản phẩm (20.0%)
- Scarves: 1 sản phẩm (10.0%)
- Accessories: 1 sản phẩm (10.0%)
- Bags: 1 sản phẩm (10.0%)
- Innerwear: 1 sản phẩm (10.0%)
```

*Hình 4.11 Kết quả sau khi huấn luyện CBF*

Trong quá trình phát triển mô hình, chúng tôi đã thực hiện các điều chỉnh về kỹ thuật và cấu trúc hàm gợi ý để quan sát sự thay đổi của Diversity qua từng giai đoạn:

- Giai đoạn 1 (Nguyên bản): Sử dụng TF-IDF trên tên sản phẩm. Chỉ số Diversity đạt 0.2691. Danh sách gợi ý bị lặp lại nhiều sản phẩm tương đồng về tên gọi.
- Giai đoạn 2 (Cải tiến Metadata): Sử dụng Metadata Soup (kết hợp tất cả thuộc tính: Gender, Category, Colour, Usage). Chỉ số tăng lên 0.3543.
- Giai đoạn 3 (Tối ưu hóa Production): Áp dụng chiến thuật giới hạn số lượng sản phẩm trên mỗi danh mục phụ và điều chỉnh max\_features=3000. Chỉ số đạt mức ổn định 0.5022.

Mặc dù về mặt lý thuyết, độ đa dạng có thể đẩy lên mức cao hơn ( $> 0.9$ ), nhưng qua thực nghiệm triển khai hệ thống, chúng tôi nhận thấy mức 0.5022 là "điểm ngọt" (Sweet Spot) tối ưu vì các lý do sau:

- Tính liên quan chỉ số này đảm bảo các sản phẩm gợi ý vẫn bám sát nhu cầu gốc của người dùng. Như kết quả thực nghiệm cho thấy, dù đa dạng về chủng loại (7 Sub-categories) nhưng tất cả sản phẩm đều giữ được đặc tính navy blue và men, đảm bảo tính thẩm mỹ của bộ sưu tập.
  - Hiệu năng hệ thống : việc giới hạn số lượng đặc trưng văn bản xuống 3000 giúp nén dung lượng ma trận TF-IDF, giảm độ trễ truy vấn API đáng kể. Điều này phục vụ mục tiêu phản hồi thời gian thực ( $< 200\text{ms}$ ) trên môi trường Mobile App thông qua FastAPI và ASP.NET Core.
  - Tính thực tiễn: kết quả mang lại một danh sách gợi ý phong phú từ Áo, Quần đến Giày dép và Phụ kiện, chuyển đổi từ một công cụ tìm kiếm đơn thuần sang một trợ lý tư vấn thời trang chuyên nghiệp.
- Kết quả sau khi huấn luyện mô hình CNN:

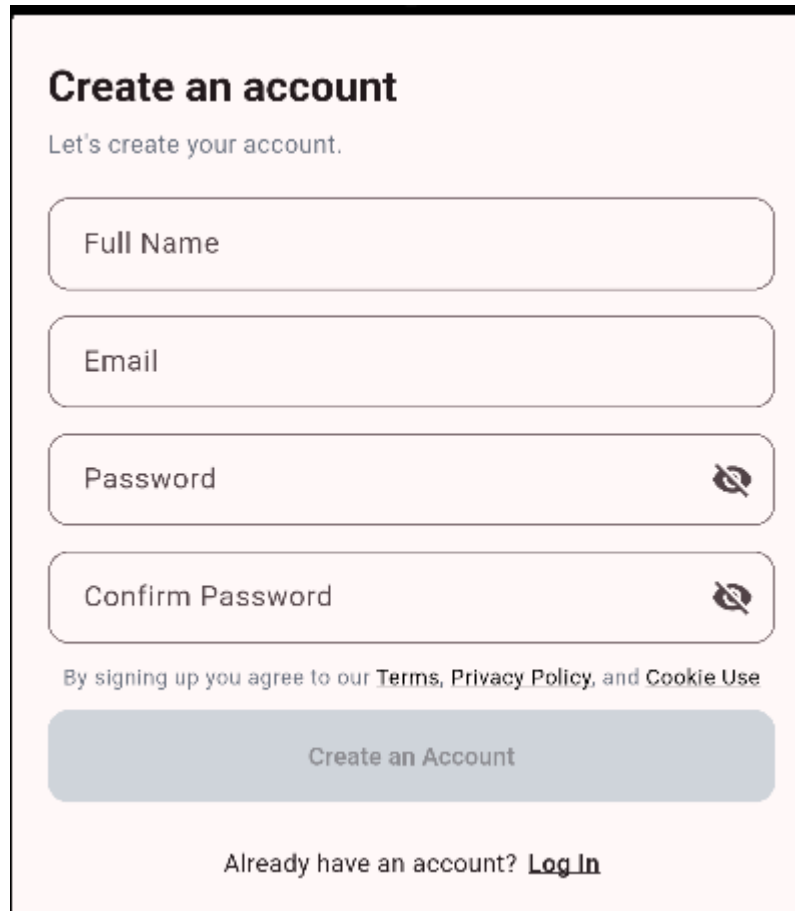
*Hình 4.12 Kết quả sau khi huấn luyện mô hình CNN*

Mô hình CNN kết hợp với thư viện FAISS trong bài toán tìm kiếm sản phẩm bằng hình ảnh đã mang lại những kết quả khả quan về cả độ chính xác lẫn hiệu năng xử lý. Mô hình đã thành công trong việc chuyển đổi hơn 44,446 hình ảnh sản phẩm từ dạng pixel sang các vector đặc trưng (embeddings) có kích thước cố định. Nhờ kỹ thuật Transfer Learning, mô hình ResNet50 có khả năng nhận diện tốt các đặc điểm phức tạp của sản phẩm thời trang như hoa văn, kiểu dáng cổ áo, chiều dài tay áo và chất liệu bề mặt. Việc sử dụng lớp GlobalAveragePooling2D giúp giảm số lượng tham số nhưng vẫn giữ được các thông tin quan trọng nhất của ảnh, giúp vector đặc trưng gọn nhẹ và hiệu quả. Với việc sử dụng IndexFlatL2 từ thư viện FAISS, hệ thống có khả năng so sánh vector của ảnh truy vấn với toàn bộ 44,446 vector trong kho dữ liệu gần như tức thì. Kết quả trả về cho thấy các sản phẩm có khoảng cách Euclidean nhỏ nhất đều có sự tương đồng rất cao về mặt thị giác với ảnh gốc. Phương pháp này cho thấy sự ổn định cao; ngay cả khi số lượng sản phẩm tăng lên, tốc độ truy xuất vẫn được duy trì ở mức nhanh chóng nhờ vào cấu trúc index tối ưu của FAISS. Khi thực hiện truy vấn với một sản phẩm cụ thể (ví dụ: một chiếc đồng hồ hoặc áo thun), hệ thống trả về chính xác các sản phẩm cùng loại, cùng màu sắc và kiểu dáng tương đồng trong top 6 kết quả. Kết quả tìm kiếm hình ảnh có sự tương quan chặt chẽ với các thuộc tính văn bản đã tiền xử lý như gender, articleType và baseColour, chứng minh rằng mô hình CNN đã học được các thuộc tính này một cách tự nhiên từ hình ảnh.

## CHƯƠNG 5. XÂY DỰNG CHƯƠNG TRÌNH

### 5.1 Giao diện đăng ký tài khoản

Đăng kí tài khoản: Người dùng điền đầy đủ thông tin vào vào form đăng kí sau đó ấn nút Create an Account khi đó sẽ nhận được thông báo tạo tài khoản thành công.



**Create an account**

Let's create your account.

Full Name

Email

Password

Confirm Password

By signing up you agree to our [Terms](#), [Privacy Policy](#), and [Cookie Use](#)

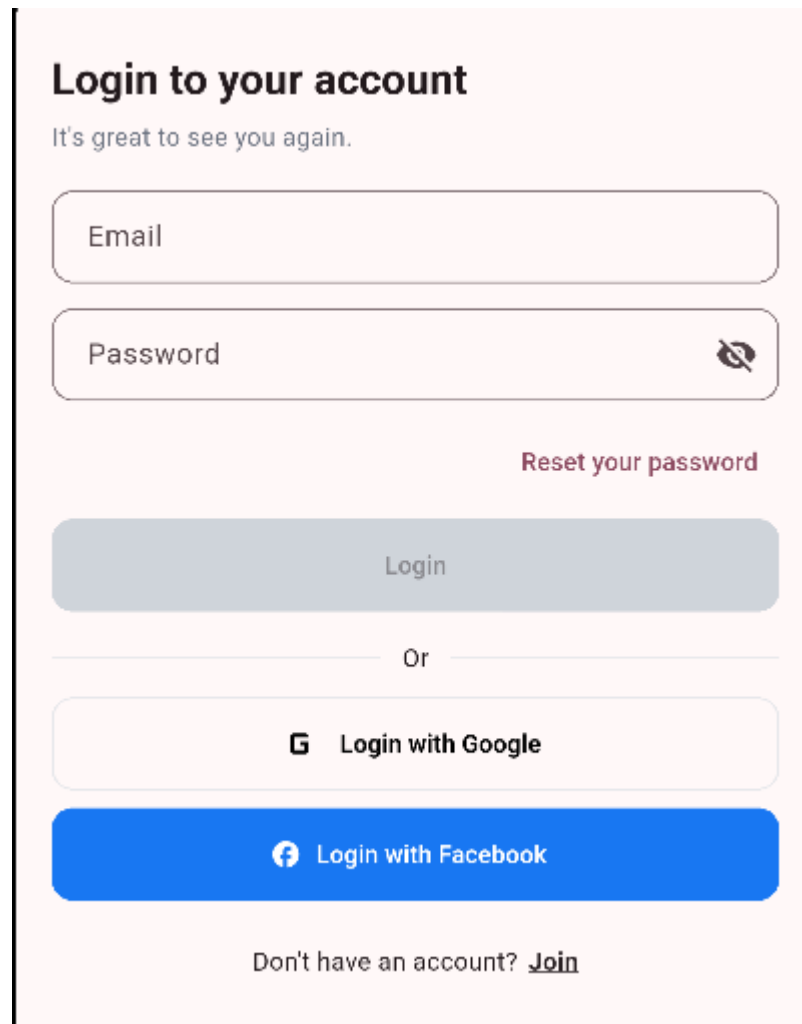
Create an Account

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

Hình 5.1 Giao diện màn hình đăng ký

## 5.2 Giao diện đăng nhập tài khoản

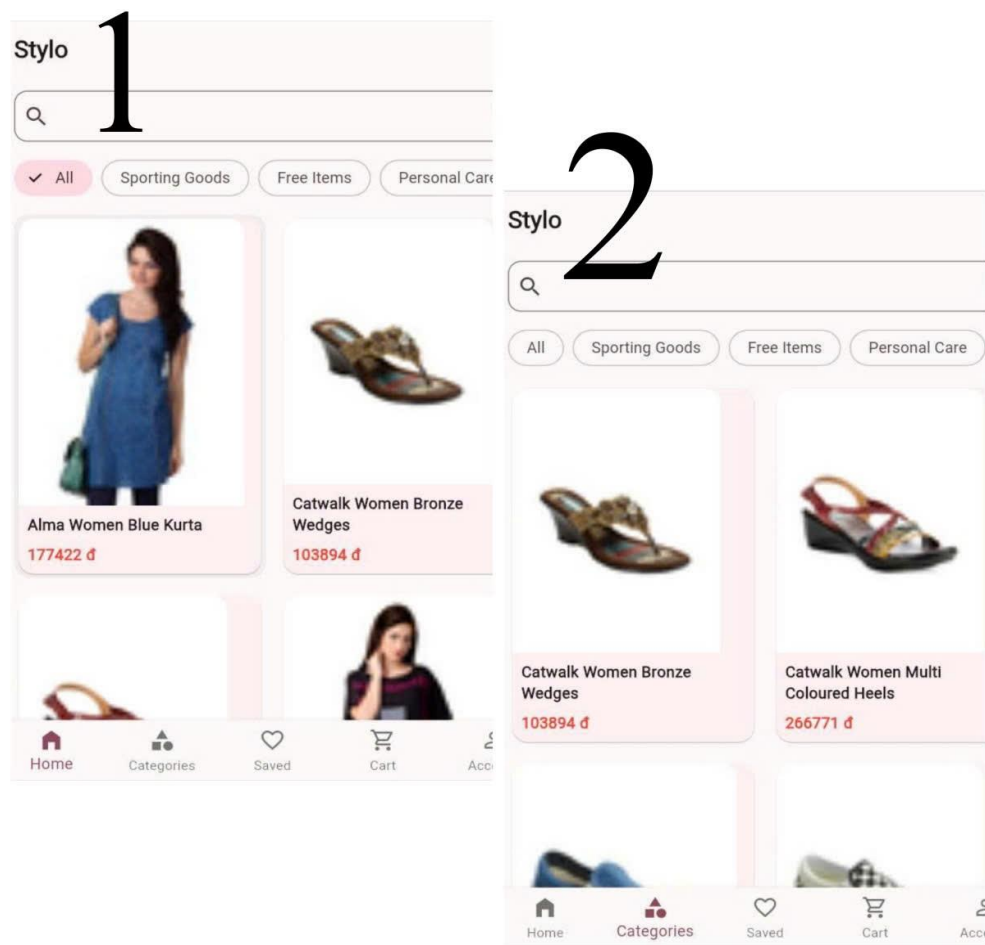
Người dùng điền đầy đủ thông tin email và password đã đăng kí trước đó, sau đó ấn nút Login là sẽ đăng nhập được vào hệ thống.

A login interface mockup on a light pink background. At the top, the heading "Login to your account" is in bold black text, followed by the subtext "It's great to see you again." in a smaller, lighter font. Below this are two input fields: "Email" and "Password". The "Password" field has a small eye icon on the right to toggle visibility. To the right of the password field is a link "Reset your password" in a reddish-pink color. Below the inputs is a large, rounded grey button labeled "Login". Underneath the button is a horizontal line with the word "Or" in the center. Below the line are two more buttons: a white one with a Google "G" logo and the text "Login with Google", and a blue one with a Facebook "f" logo and the text "Login with Facebook". At the bottom, there is a link "Don't have an account? Join" where "Join" is underlined.

*Hình 5.2 Giao diện màn hình đăng nhập*

### 5.3 Giao diện trang chủ

Đây là giao diện chính khi người dùng mở ứng dụng (hình 1). Bao gồm: Thanh tìm kiếm ở phía trên, các nút lọc phân loại nhanh (hình 2): lọc ra các sản phẩm có trong phân loại đó, khu vực hiển thị sản phẩm: Hiển thị danh sách các sản phẩm nổi bật với hình ảnh, tên và giá, thanh điều hướng dưới cùng (Bottom Navigation Bar): Trang chủ, Phân loại, Yêu thích, Giỏ hàng, Tài khoản

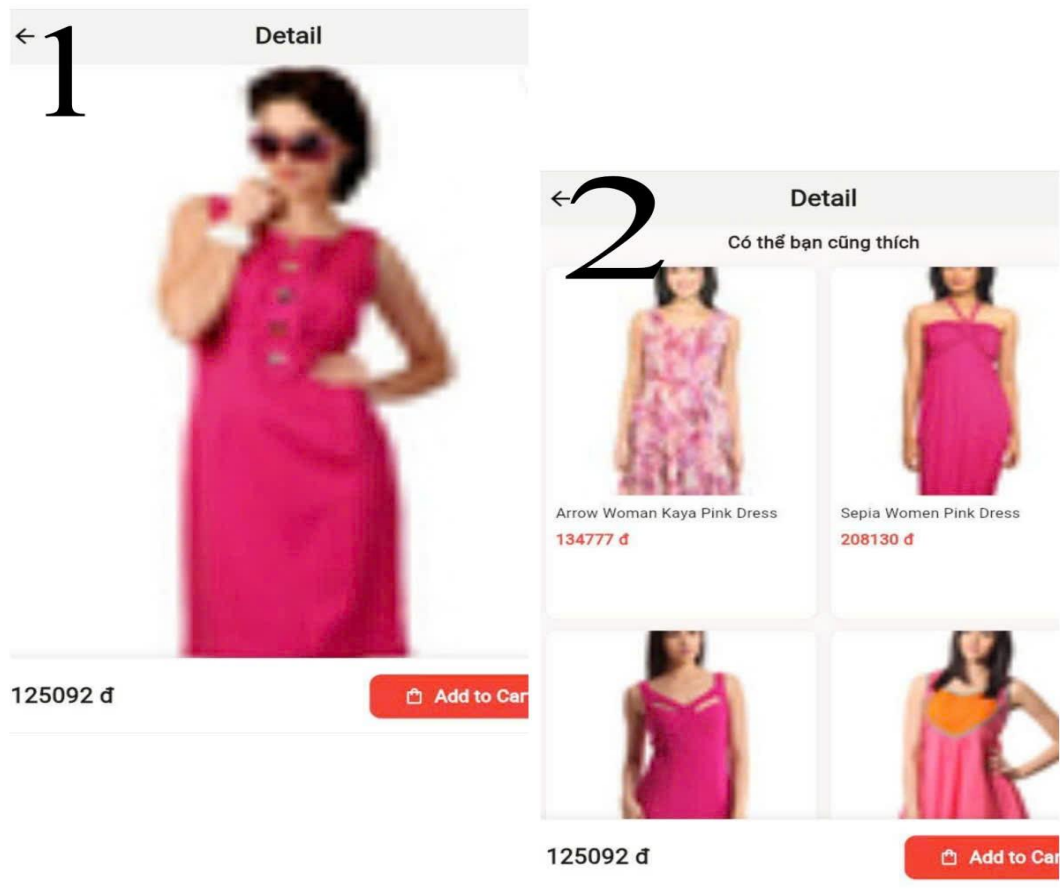


Hình 5.3 Giao diện trang chủ



#### 5.4 Giao diện gợi ý sản phẩm

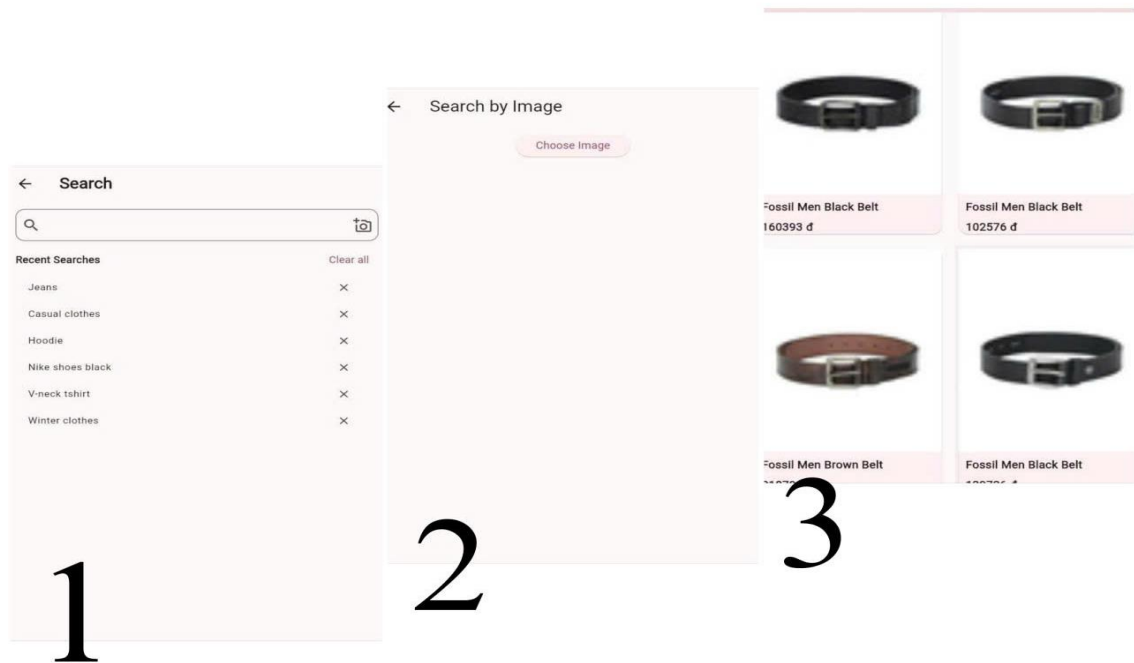
Giao diện chức năng gợi ý sản phẩm: Trang chi tiết sản phẩm (hình 1): Hiển thị tất cả thông tin của sản phẩm đó bao gồm tên sản phẩm, giá sản phẩm, mô tả, đánh giá sản phẩm. Ngoài ra có thể thêm sản phẩm vào mục yêu thích và thêm vào giỏ hàng. Khi lướt xuống là trang gợi ý sản phẩm liên quan hình 2), trang này sẽ hiển thị 10 sản phẩm liên quan nhất về sản phẩm đó dựa vào các đặc tính liên quan.



Hình 5.4 Giao diện màn hình gợi ý sản phẩm

## 5.5 Giao diện tìm kiếm sản phẩm bằng hình ảnh

Giao diện chức năng tìm kiếm sản phẩm bằng hình ảnh: Trang tìm kiếm (hình 1) bao gồm tìm kiếm theo từ khóa và tìm kiếm theo hình ảnh. Tìm kiếm theo từ khóa sẽ được hiển thị các từ khóa đã được tìm kiếm gần đây. Người dùng có thể xóa từ khóa hoặc xóa toàn bộ lịch sử tìm kiếm. Biểu tượng máy ảnh trong ô tìm kiếm cho phép chuyển sang chức năng tìm kiếm bằng hình ảnh (hình 2). Sau khi chuyển sang trang tìm kiếm bằng hình ảnh có nút chọn ảnh, người dùng sẽ được chọn ảnh từ thư viện. Sau khi người dùng tải ảnh lên, hệ thống hiển thị danh sách sản phẩm tương tự như hình ảnh người dùng cung cấp (hình 3).



Hình 5.5 Giao diện tìm kiếm sản phẩm bằng hình ảnh

## KẾT KUẬN

### Kết quả đạt được

Qua quá trình nghiên cứu và triển khai, đề tài đã đạt được những kết quả quan trọng trong việc xây dựng hệ thống gợi ý và tìm kiếm thông minh dành cho cửa hàng thời trang. Thuật toán lọc cộng tác nội dung CBF đã được áp dụng thành công để phân tích các thuộc tính đặc trưng của sản phẩm như loại trang phục, màu sắc, chất liệu, phong cách..., từ đó tạo ra các gợi ý phù hợp với sản phẩm mà người dùng đang quan tâm. Mô hình cho thấy khả năng hoạt động ổn định, dễ dàng mở rộng, và đặc biệt phù hợp trong bối cảnh dữ liệu người dùng còn hạn chế. Song song đó, mô hình tìm kiếm sản phẩm bằng hình ảnh dựa trên CNN kết hợp FAISS đã giúp hệ thống có thể nhận diện và truy vấn các sản phẩm tương tự chỉ từ một ảnh đầu vào. CNN đảm nhiệm trích xuất đặc trưng hình ảnh, trong khi FAISS tăng tốc độ tìm kiếm vector, giúp người dùng có được kết quả tương đồng chỉ trong thời gian rất ngắn. Thực nghiệm cho thấy mô hình có khả năng phân biệt tốt các loại quần áo, xử lý tương đối hiệu quả các biến động về góc chụp và bố cục, qua đó mang lại trải nghiệm mua sắm trực quan và tiện lợi hơn. Nhìn chung, cả hai hướng tiếp cận đều chứng minh được tính khả thi và mang lại giá trị thực tiễn trong việc nâng cao chất lượng tương tác giữa người dùng và hệ thống thương mại điện tử.

### Hạn chế

Mặc dù đạt được nhiều kết quả tích cực, đề tài vẫn tồn tại một số hạn chế cần được nhìn nhận rõ. Đối với thuật toán CBF, chất lượng gợi ý phụ thuộc nhiều vào mức độ chi tiết và nhất quán của dữ liệu mô tả sản phẩm; khi thông tin về sản phẩm chưa đầy đủ hoặc không đồng nhất, mô hình dễ tạo ra các gợi ý đơn điệu và thiếu đa dạng. Ngoài ra, CBF chỉ dựa vào đặc điểm sản phẩm mà chưa xem xét đầy đủ hành vi và sở thích cá nhân theo thời gian của người dùng nên chưa mang lại tính cá nhân hóa sâu. Với mô hình tìm kiếm bằng hình ảnh, độ chính xác vẫn chưa thực sự ổn định khi ảnh đầu vào bị mờ, thiếu sáng, chứa nhiều vật thể gây nhiễu hoặc không thể hiện rõ chi tiết thời trang. Tập dữ liệu để huấn luyện CNN còn hạn chế về số lượng và tính đa dạng nên mô hình đôi khi chưa nhận diện tốt những phong cách thời trang đặc thù hoặc sản phẩm ít xuất hiện. Bên cạnh đó, hệ thống mới dừng

lại ở mức mô phỏng thử nghiệm mà chưa được tối ưu toàn diện về hiệu năng, khả năng tích hợp với giao diện web thực tế hoặc khả năng mở rộng trên quy mô lớn.

### **Hướng phát triển**

Trong thời gian tới, đề tài có thể được mở rộng theo nhiều hướng nhằm nâng cao chất lượng và khả năng ứng dụng thực tế. Hệ thống gợi ý dựa trên CBF có thể kết hợp thêm dữ liệu hành vi như lịch sử mua sắm, thời gian xem sản phẩm, tần suất truy cập hay đánh giá của người dùng để chuyển sang hướng hybrid recommender giúp tăng mức độ cá nhân hóa. Bộ dữ liệu mô tả sản phẩm cũng cần được mở rộng và chuẩn hóa để thuật toán khai thác tốt hơn sự khác biệt giữa các loại trang phục. Đối với tìm kiếm hình ảnh, mô hình CNN có thể được thay thế hoặc bổ sung bởi các kiến trúc tiên tiến hơn như EfficientNet, Vision Transformer (ViT) hoặc CLIP nhằm cải thiện khả năng hiểu ngữ nghĩa và phân loại hình ảnh trong nhiều điều kiện phức tạp. Tập dữ liệu huấn luyện cũng cần mở rộng cả về số lượng lẫn độ đa dạng, bao gồm nhiều môi trường chụp, độ sáng, kiểu dáng và phong cách để mô hình có tính tổng quát cao hơn. Hệ thống truy vấn FAISS có thể được tối ưu để hoạt động trên môi trường sản phẩm thực tế với số lượng vector lớn hơn. Cuối cùng, việc triển khai mô hình vào một nền tảng web có giao diện trực quan, tốc độ phản hồi nhanh và trải nghiệm người dùng tốt sẽ giúp hoàn thiện chức năng gợi ý sản phẩm thông minh, hướng tới một giải pháp hỗ trợ mua sắm hiện đại, trực quan và hiệu quả cho các cửa hàng thời trang.

## PHỤ LỤC

### Phụ lục 1: Hướng dẫn cài đặt

#### a. Tổng quan hệ thống

- Hệ thống Stylo được xây dựng theo mô hình Client – Server, bao gồm các thành phần chính:
  - Frontend (Mobile App): Ứng dụng di động viết bằng Flutter.
  - Backend chính: ASP.NET Core Web API – xử lý nghiệp vụ.
  - API gợi ý sản phẩm: FastAPI – Content-Based Filtering.
  - API tìm kiếm ảnh tương tự: FastAPI – CNN + FAISS.
- Các thành phần này hoạt động độc lập và giao tiếp thông qua RESTful API.

#### b. Yêu cầu môi trường

- Phần mềm cần cài đặt
  - Python  $\geq 3.9$
  - .NET SDK  $\geq 6.0$
  - Flutter SDK  $\geq 3.x$
  - Android Studio (hoặc thiết bị Android thật)
  - SQL Server
  - Git
- Phần cứng khuyến nghị
  - RAM  $\geq 8GB$
  - Ổ cứng trống  $\geq 20GB$
  - CPU hỗ trợ ảo hóa (khuyến nghị)

#### c. Cài đặt Backend chính (ASP.NET Core Web API)

Bước 1: Clone source code

- Git clone <https://github.com/PhucHoang200/stylo-backend.git>

- cd stylo-backend

Bước 2: Cấu hình cơ sở dữ liệu

- Mở file appsettings.json
- Cập nhật chuỗi kết nối SQL Server: "ConnectionStrings":  
{"DefaultConnection":  
"Server=.;Database=StyloDB;Trusted\_Connection=True;" }

Bước 3: Khởi tạo cơ sở dữ liệu

- Mở project bằng Visual Studio
- Chạy lệnh Migration (nếu có) hoặc chạy script SQL khởi tạo CSDL
- Đảm bảo database được tạo thành công trong SQL Server

Bước 4: Chạy backend

#### **d. Cài đặt API gợi ý sản phẩm (FastAPI – Recommendation Service)**

Bước 1: Clone source code

- Git clone <https://github.com/PhucHoang200/recommend-service-fastapi.git>
- Cd [recommend-service-fastapi](#)

Bước 2: Tạo môi trường ảo: python -m venv venv

- Kích hoạt môi trường:
  - Windows: venv\Scripts\activate
  - Linux/Mac: source venv/bin/activate

Bước 3: Cài đặt thư viện: pip install -r requirements.txt

Bước 4: Chạy API: uvicorn app.main:app --reload

#### **e. Cài đặt API tìm kiếm ảnh tương tự (FastAPI – Image Similarity)**

Bước 1: Clone source code

- Git clone [https://github.com/KimThao26042004/Image\\_Sim\\_Api.git](https://github.com/KimThao26042004/Image_Sim_Api.git)
- cd Image\_Sim\_Api

Bước 2: Tạo môi trường ảo: python -m venv venv

- Kích hoạt môi trường:
  - Windows: `venv\Scripts\activate`
  - Linux/Mac: `source venv/bin/activate`

Bước 3: Cài đặt thư viện: `pip install -r requirements.txt`

Bước 4: Chuẩn bị model

- Đảm bảo đã có: Mô hình CNN đã train, FAISS index, thư mục ảnh sản phẩm
- Kiểm tra đường dẫn model trong file cấu hình API

Bước 5: Chạy API: `uvicorn app:app --reload`

#### **f. Cài đặt Frontend (Flutter Mobile App)**

Bước 1: Clone source code

- Git clone <https://github.com/KimThao26042004/Stylo.git>
- `cd Stylo`

Bước 2: Cấu hình API

- Mở file cấu hình (ví dụ: `constants.dart`, `api_service.dart`)
- Cập nhật địa chỉ API:
  - `const String baseApiUrl = "http://10.0.2.2:xxxx";`
  - `const String recommendApiUrl = "http://10.0.2.2:8000";`
  - `const String imageSearchApiUrl = "http://10.0.2.2:8001";`

Bước 3: Cài đặt package: `flutter pub get`

Bước 4: Chạy ứng dụng: `flutter run`

#### **g. Cài đặt Web Admin**

Bước 1: Clone source code

- Git clone [https://github.com/KimThao26042004/Web\\_Stylo.git](https://github.com/KimThao26042004/Web_Stylo.git)
- `cd Web_Stylo`

Bước 2: Cấu hình kết nối cơ sở dữ liệu

- Mở file appsettings.json và chỉnh sửa chuỗi kết nối SQL Server:  
`"ConnectionStrings": { "DefaultConnection":  
"Server=.;Database=StyloDB;Trusted_Connection=True;" }`

### Bước 3: Cấu hình kết nối Backend API

- Mở file cấu hình (ví dụ: ApiSettings.cs, appsettings.json hoặc Constants)
- Cập nhật địa chỉ Backend API: `"ApiBaseUrl": "https://localhost:xxxx/"`

### Bước 4: Build và chạy ứng dụng

## Phụ lục 2: Hướng dẫn sử dụng

### a. Hướng dẫn sử dụng ứng dụng người dùng (Flutter)

- Đăng ký / Đăng nhập
  - Mở ứng dụng.
  - Chọn Đăng ký nếu chưa có tài khoản, nhập đầy đủ thông tin và xác nhận.
  - Chọn Đăng nhập, nhập email/số điện thoại (tùy hệ thống) và mật khẩu.
  - Sau khi đăng nhập thành công, hệ thống chuyển về Trang chủ.
- Trang chủ
  - Trang chủ hiển thị các nhóm nội dung như: sản phẩm nổi bật, sản phẩm mới, sản phẩm theo danh mục...
  - Người dùng có thể nhấn vào từng sản phẩm để xem chi tiết.
- Xem chi tiết sản phẩm
  - Nhấn vào một sản phẩm bất kỳ.
  - Tại màn hình chi tiết:
    - Xem hình ảnh, mô tả, giá bán.
    - Chọn màu sắc/size (nếu có).
    - Nhấn Thêm vào giỏ hoặc Mua ngay để thực hiện đặt hàng.
- Gợi ý sản phẩm



- Hệ thống hiển thị danh sách sản phẩm được đề xuất dựa trên độ tương đồng.
- Nhấn vào sản phẩm để xem chi tiết hoặc thêm vào giỏ hàng.
- Tìm kiếm bằng hình ảnh
  - Mở chức năng Tìm kiếm bằng hình ảnh.
  - Chọn ảnh từ thư viện.
  - Hệ thống gửi ảnh lên API và trả về danh sách sản phẩm tương tự (kết quả hiển thị dạng danh sách, có ảnh và thông tin cơ bản).
  - Người dùng nhấn vào kết quả để xem chi tiết sản phẩm.
- Giỏ hàng và đặt hàng
  - Vào Giỏ hàng để xem các sản phẩm đã thêm.
  - Điều chỉnh số lượng / xoá sản phẩm nếu cần.
  - Nhấn Thanh toán:
  - Nhập thông tin giao hàng (họ tên, số điện thoại, địa chỉ).
  - Xác nhận đơn hàng.
  - Sau khi đặt hàng thành công, đơn hàng được ghi nhận trong hệ thống và hiển thị trong lịch sử mua hàng (nếu có).

#### **b. Hướng dẫn sử dụng trang quản trị (Web Admin)**

- Đăng nhập Admin
  - Truy cập Web Admin.
  - Nhập tài khoản quản trị và đăng nhập.
  - Sau khi đăng nhập thành công, hệ thống hiển thị Dashboard quản lý chính.
- Quản lý danh mục
  - Thêm danh mục mới.
  - Chỉnh sửa tên danh mục.
  - Xóa danh mục (tùy phân quyền và ràng buộc dữ liệu).

- Quản lý sản phẩm
  - Thêm sản phẩm mới (tên, giá, mô tả, danh mục, thương hiệu, ảnh...).
  - Cập nhật thông tin sản phẩm.
  - Xoá/ẩn sản phẩm (tùy thiết kế hệ thống).
  - Ảnh sản phẩm được lưu theo cấu trúc thư mục (ví dụ Content/images/...) và hiển thị trên giao diện.
- Quản lý đơn hàng
  - Xem danh sách đơn hàng theo trạng thái.
  - Xem chi tiết đơn (sản phẩm, số lượng, tổng tiền, địa chỉ giao hàng).
  - Cập nhật trạng thái đơn hàng (xác nhận/đang giao/hoàn thành/hủy...).

## TÀI LIỆU THAM KHẢO

- [1]. United Nations Conference on Trade and Development (UNCTAD), *Digital Economy Report 2021*, United Nations, 2021.
- [2]. Bộ Công Thương, *Sách trắng Thương mại điện tử Việt Nam 2024*, Cục Thương mại điện tử và Kinh tế số, 2024.
- [3]. McKinsey & Company, *The Value of Getting Personalization Right*, McKinsey Insights, 2021.
- [4]. Li Deng & Dong Yu, *Deep Learning: Methods and Applications*, Foundations and Trends in Signal Processing, 2014.
- [5]. Alex Krizhevsky, Ilya Sutskever & Geoffrey E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems, 2012.
- [6]. Python Software Foundation, *Python Documentation*, Python.org, 2024.
- [7]. Wes McKinney, *Python for Data Analysis*, O'Reilly Media (Sample & Author Version), 2018.
- [8]. Sebastián Ramírez, *FastAPI Documentation*, FastAPI Official Website, 2024.
- [9]. Google Research, *Google Colaboratory Documentation*, Google, 2024.
- [10]. Google Developers (n.d.). External data: Local Files, Drive, Sheets, and Cloud Storage. Google Developers. Truy cập tại:  
<https://colab.research.google.com/notebooks/io.ipynb>
- [11]. Google LLC, *Flutter Documentation*, Flutter.dev, 2024.
- [12]. Google Flutter Team, *Flutter Architectural Overview*, Flutter.dev, 2023.
- [13]. Nguyen, T., Lauw, H., “Learning Content-based Representations for Recommendation”, 2014.
- [14]. Alex Krizhevsky, Ilya Sutskever & Geoffrey E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems, 2012.
- [15]. RetailRocket (2017). *eCommerce Dataset*. Available at  
<https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset>