

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN

**ĐỒ ÁN TỐT NGHIỆP**  
NGÀNH: CÔNG NGHỆ THÔNG TIN  
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:  
**ỨNG DỤNG SINH ẢNH NGHỆ THUẬT  
SỬ DỤNG MÔ HÌNH HỌC SÂU**

Người hướng dẫn: TS. NGUYỄN VĂN HIỆU

Sinh viên thực hiện: NGUYỄN VĂN MẠNH

Số thẻ sinh viên: 102200024

Lớp: 20T1

Đà Nẵng, 06/2025

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN

# ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN  
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:  
ỨNG DỤNG SINH ẢNH NGHỆ THUẬT  
SỬ DỤNG MÔ HÌNH HỌC SÂU

Người hướng dẫn: TS. NGUYỄN VĂN HIỆU

Sinh viên thực hiện: NGUYỄN VĂN MẠNH

Số thẻ sinh viên: 102200024

Lớp: 20T1

Đà Nẵng, 06/2025

## NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

### I. Thông tin chung:

- Họ và tên sinh viên: Nguyễn Văn Mạnh
- Lớp: 20T1                      Số thẻ SV: 102200024
- Tên đề tài: Ứng dụng sinh ảnh nghệ thuật sử dụng mô hình học sâu
- Người hướng dẫn: Nguyễn Văn Hiệu                      Học hàm/ học vị: Tiến sĩ

### II. Nhận xét, đánh giá đồ án tốt nghiệp:

- Về tính cấp thiết, tính mới, khả năng ứng dụng của đề tài: (điểm tối đa là 2đ)  
.....  
.....
- Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án: (điểm tối đa là 4đ)  
.....  
.....
- Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp: (điểm tối đa là 2đ)  
.....  
.....
- Đề tài có giá trị khoa học/ có bài báo/ giải quyết vấn đề đặt ra của doanh nghiệp hoặc nhà trường: (điểm tối đa là 1đ)  
.....  
.....
- Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:  
.....  
.....

### III. Tinh thần, thái độ làm việc của sinh viên: (điểm tối đa 1đ)

.....

### IV. Đánh giá:

- Điểm đánh giá: ...../10 (lấy đến 1 số lẻ thập phân)
- Đề nghị:  Được bảo vệ đồ án     Bổ sung để bảo vệ     Không được bảo vệ

Đà Nẵng, ngày            tháng            năm 2025

**Người hướng dẫn**

## TÓM TẮT

Tên đề tài: “Ứng dụng sinh ảnh nghệ thuật sử dụng mô hình học sâu”

Sinh viên thực hiện: Nguyễn Văn Mạnh

Số thẻ SV: 102200024

Lớp: 20T1

Tóm tắt nội dung đồ án:

ArtGenix là một ứng dụng sinh ảnh nghệ thuật dựa trên các phương pháp, mô hình học sâu như: Neural Style Transfer, CycleGAN và Diffusion LoRA Style Transfer cho phép chuyển đổi ảnh thông thường thành các tác phẩm nghệ thuật độc đáo. Bên cạnh chức năng chính là tạo ảnh nghệ thuật, ArtGenix cung cấp các tính năng tương tác xã hội như chia sẻ ảnh nghệ thuật, bình luận trong thời gian thực và thích trên mỗi tác phẩm nghệ thuật, chia sẻ bài viết, bình luận trong thời gian thực, thích trên mỗi bài viết. Trò chuyện nhắn tin trực tuyến. Có thể nói rằng ArtGenix là phiên bản tối giản có đầy đủ các chức năng chính và cơ bản của Midjourney, Facebook, Instagram và Messenger.

Các chức năng chính:

- Người dùng: Sinh ảnh nghệ thuật, chia sẻ, bình luận, tương tác thích trên mỗi tác phẩm nghệ thuật, bài viết trong thời gian thực. Trò chuyện nhắn tin trực tuyến. Quản lý các thông báo liên quan đến tác phẩm nghệ thuật và bài viết. Quản lý tài nguyên của mỗi người dùng bao gồm: tác phẩm nghệ thuật, ảnh nội dung, ảnh phong cách, bộ sưu tập.
- Admin: Quản lý tài nguyên của hệ thống bao gồm: quản lý người dùng, quản lý tác phẩm nghệ thuật, thẻ, ảnh nội dung, ảnh phong cách, bộ sưu tập. Thống kê số liệu của hệ thống.
- Manager, Super Admin: Quản lý Admin

Ứng dụng ArtGenix hướng đến việc kết hợp công nghệ học sâu với các tính năng mạng xã hội, tạo ra một nền tảng sáng tạo và tương tác hiệu quả cho người dùng.

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Văn Mạnh Số thẻ sinh viên: 102200024

Lớp: 20T1 Khoa: Công nghệ thông tin Ngành: Công nghệ thông tin

1. Tên đề tài đồ án: Ứng dụng sinh ảnh nghệ thuật sử dụng mô hình học sâu
2. Đề tài thuộc diện:  Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
3. Các số liệu và dữ liệu ban đầu:

.....  
.....  
.....

4. Nội dung các phần thuyết minh và tính toán:

- Giới thiệu
- Chương 1: Lý thuyết và công nghệ
- Chương 2: Mô hình đề xuất
- Chương 3: Phân tích và thiết kế hệ thống
- Chương 4: Kết quả triển khai
- Kết luận & hướng phát triển
- Tài liệu tham khảo

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

- Use case Diagrams
- Sequence Diagrams
- Class Diagrams
- Activity Diagrams
- Database Design

6. Họ tên người hướng dẫn: TS. Nguyễn Văn Hiệu, Khoa Công Nghệ Thông Tin,  
Trường Đại Học Bách Khoa, Đại Học Đà Nẵng

7. Ngày giao nhiệm vụ đồ án: 17/02/2025

8. Ngày hoàn thành đồ án: 07/06/2025

Đà Nẵng, ngày tháng năm 2025

Trưởng Bộ môn .....

Người hướng dẫn

## LỜI NÓI ĐẦU

Trước tiên, em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô giáo trong khoa Công nghệ Thông tin – Trường Đại học Bách Khoa, Đại học Đà Nẵng. Trong suốt quá trình học tập tại trường, em đã được tiếp cận và học hỏi những kiến thức nền tảng quý báu về chuyên môn cùng với các kỹ năng cần thiết cho thực tiễn. Đây chính là hành trang quan trọng giúp em có thể tiếp cận, triển khai và hoàn thành đề án tốt nghiệp của mình.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến TS. Nguyễn Văn Hiệu giảng viên khoa Công nghệ Thông tin, người đã tận tình hướng dẫn, chỉ bảo và đồng hành cùng em trong suốt quá trình thực hiện đề tài. Sự hỗ trợ của thầy đã giúp em tháo gỡ nhiều khó khăn và hoàn thiện đề án đúng tiến độ.

Bên cạnh đó, em cũng xin gửi lời cảm ơn chân thành đến gia đình và bạn bè những người luôn bên cạnh, động viên và tiếp thêm động lực cho em trong suốt thời gian thực hiện đề án.

Dù đã nỗ lực hoàn thành đề án trong khả năng và thời gian cho phép, nhưng em hiểu rằng vẫn còn tồn tại những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy Cô để có thể hoàn thiện đề tài hơn nữa, cũng như tích lũy thêm kinh nghiệm cho công việc sau này.

***Một lần nữa, em xin chân thành cảm ơn !***

Đà Nẵng, ngày      tháng      năm 2025

**Sinh viên thực hiện**

## CAM ĐOAN

Tôi xin cam đoan:

1. Những nội dung trong báo cáo tốt nghiệp này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của giảng viên Nguyễn Văn Hiệu.
2. Mọi tham khảo dùng trong đề án đều được trích dẫn rõ ràng và trung thực tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

*Đà Nẵng, ngày    tháng    năm 2025*

**Sinh viên thực hiện**

# MỤC LỤC

<b>LỜI NÓI ĐẦU</b> .....	<b>i</b>
<b>CAM ĐOAN</b> .....	<b>ii</b>
<b>DANH SÁCH CÁC BẢNG</b> .....	<b>iii</b>
<b>DANH SÁCH CÁC HÌNH VẼ</b> .....	<b>iv</b>
<b>DANH SÁCH CÁC CÔNG THỨC</b> .....	<b>v</b>
<b>DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT</b> .....	<b>vi</b>
<b>MỞ ĐẦU</b> .....	<b>1</b>
<b>1. Tổng quan</b> .....	<b>1</b>
<b>2. Mục đích</b> .....	<b>1</b>
<b>3. Phạm vi, đối tượng và phương pháp nghiên cứu</b> .....	<b>1</b>
<b>4. Công nghệ và môi trường phát triển</b> .....	<b>3</b>
<b>5. Cấu trúc của đề án tốt nghiệp</b> .....	<b>3</b>
<b>CHƯƠNG 1: LÝ THUYẾT VÀ CÔNG NGHỆ</b> .....	<b>5</b>
<b>1.1. Laravel backend framework</b> .....	<b>5</b>
<b>1.2. Django backend framework</b> .....	<b>5</b>
<b>1.3. VueJS frontend framework</b> .....	<b>6</b>
<b>1.4. RESTful API</b> .....	<b>7</b>
<b>1.5. Git &amp; Github</b> .....	<b>9</b>
<b>1.6. Vercel &amp; Railway</b> .....	<b>10</b>
<b>CHƯƠNG 2: MÔ HÌNH ĐỀ XUẤT</b> .....	<b>11</b>
<b>2.1. Phát biểu bài toán</b> .....	<b>11</b>
<b>2.2. Lịch sử phát triển</b> .....	<b>12</b>
<b>2.3. Neural Style Transfer</b> .....	<b>12</b>
2.3.1. Kiến trúc Neural Style Transfer .....	<b>12</b>
2.3.2. Đề xuất cải tiến .....	<b>15</b>
<b>2.4. CycleGAN</b> .....	<b>16</b>
<b>2.5. Diffusion LoRA</b> .....	<b>18</b>

2.5.1.	Kiến trúc Stable Diffusion .....	18
2.5.2.	Diffusion LoRA Style Transfer.....	20
<b>2.6.</b>	<b>Thử nghiệm .....</b>	<b>23</b>
2.6.1.	Thử nghiệm Neural Style Transfer .....	23
2.6.2.	Thử nghiệm CycleGAN .....	30
2.6.3.	Thử nghiệm Diffusion LoRA Style Transfer.....	31
<b>2.7.</b>	<b>Kết luận.....</b>	<b>37</b>
<b>CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>		<b>38</b>
<b>3.1.</b>	<b>Bối cảnh và động lực phát triển.....</b>	<b>38</b>
<b>3.2.</b>	<b>Phân tích yêu cầu hệ thống .....</b>	<b>38</b>
3.2.1.	Yêu cầu chức năng .....	38
3.2.2.	Yêu cầu phi chức năng.....	39
<b>3.3.</b>	<b>Thiết kế hệ thống.....</b>	<b>40</b>
<b>3.4.</b>	<b>Thiết kế cơ sở dữ liệu.....</b>	<b>41</b>
3.4.1.	Lược đồ cơ sở dữ liệu .....	41
3.4.2.	Danh sách các bảng.....	43
<b>3.5.</b>	<b>Use case Diagrams .....</b>	<b>49</b>
3.5.1.	Biểu đồ Use case của Guest .....	50
3.5.2.	Danh sách biểu đồ Use case của User.....	50
3.5.3.	Danh sách biểu đồ Use case của Admin .....	55
<b>3.6.</b>	<b>Sequence Diagrams.....</b>	<b>59</b>
3.6.1.	Danh sách biểu đồ tuần tự nhóm chức năng Xác thực.....	59
3.6.2.	Danh sách biểu đồ tuần tự nhóm chức năng Quản lý nghệ thuật ....	62
3.6.3.	Danh sách biểu đồ tuần tự nhóm chức năng Quản lý bài viết .....	66
<b>3.7.</b>	<b>Activity Diagrams .....</b>	<b>70</b>
3.7.1.	Danh sách biểu đồ hoạt động nhóm chức năng Xác thực .....	70
3.7.2.	Danh sách biểu đồ hoạt động nhóm chức năng của Admin.....	72
3.7.3.	Danh sách biểu đồ hoạt động nhóm chức năng của User .....	74
<b>3.8.</b>	<b>Class Diagrams.....</b>	<b>75</b>

3.8.1.	Biểu đồ lớp User .....	75
3.8.2.	Biểu đồ lớp Tag, Album.....	76
3.8.3.	Danh sách biểu đồ lớp liên quan đến Art.....	77
3.8.4.	Danh sách biểu đồ lớp liên quan đến Article.....	79
3.8.5.	Biểu đồ lớp Message.....	80
<b>CHƯƠNG 4: KẾT QUẢ TRIỂN KHAI .....</b>		<b>81</b>
<b>4.1.</b>	<b>Triển khai .....</b>	<b>81</b>
4.1.1.	Vercel Client VueJS.....	82
4.1.2.	Railway Backend Laravel API.....	83
4.1.3.	Railway MySQL .....	84
4.1.4.	Kaggle Flask Ngrok API Model .....	85
<b>4.2.</b>	<b>APIs Server.....</b>	<b>86</b>
<b>4.3.</b>	<b>Giao diện ứng dụng.....</b>	<b>87</b>
4.3.1.	Trang chủ .....	87
4.3.2.	Nhóm chức năng Xác thực và Cài đặt tài khoản.....	87
4.3.3.	Nhóm chức năng Quản lý người dùng .....	88
4.3.4.	Nhóm chức năng liên quan đến Tag .....	89
4.3.5.	Nhóm chức năng liên quan đến Img Content .....	91
4.3.6.	Nhóm chức năng liên quan đến Img Style.....	94
4.3.7.	Nhóm chức năng liên quan đến Art .....	97
4.3.8.	Nhóm chức năng liên quan đến Article .....	100
4.3.9.	Nhóm chức năng liên quan đến Album .....	103
4.3.10.	Nhóm chức năng Tạo sinh ảnh nghệ thuật.....	106
4.3.11.	Nhóm chức năng Album Art, Img Content, Img Style.....	107
4.3.12.	Nhóm chức năng Nhắn tin, quản lý thông báo .....	109
4.3.13.	Nhóm chức năng Trang chủ, bài viết, sinh ảnh và khám phá.....	110
4.3.14.	Chức năng Xem thống kê của hệ thống .....	116
<b>4.4.</b>	<b>Mô hình sinh ảnh .....</b>	<b>117</b>
<b>4.5.</b>	<b>Kiểm thử hệ thống .....</b>	<b>118</b>

4.5.1. Kiểm thử Frontend .....	118
4.5.2. Kiểm thử Backend API .....	122
<b>KẾT LUẬN &amp; HƯỚNG PHÁT TRIỂN .....</b>	<b>124</b>
1. Kết quả đạt được.....	124
2. Hạn chế của đề tài.....	124
3. Hướng phát triển .....	125
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>126</b>

## DANH SÁCH CÁC BẢNG

Bảng 1.1 Một số API mẫu của hệ thống.....	9
Bảng 2.1 So sánh trước và sau cải tiến.....	16
Bảng 2.2 Bộ tham số của Neural Style Transfer .....	24
Bảng 2.3 Tác động của các tham số đến thời gian sinh ảnh và chất lượng ảnh .....	27
Bảng 2.4 Lựa chọn số Epoch tốt nhất với mỗi kích thước Resize .....	27
Bảng 2.5 Thời gian sinh ảnh bằng phương pháp Neural Style Transfer .....	28
Bảng 2.6 Bộ tham số tốt nhất được chọn .....	29
Bảng 2.7 Bộ tham số ảnh hưởng đến thời gian train CycleGAN.....	30
Bảng 2.8 Bộ tham số ảnh hưởng đến thời gian suy luận.....	30
Bảng 2.9 Bộ tham số suy luận .....	31
Bảng 2.10 Loss Gram Matrix và LPIPS.....	32
Bảng 2.11 Thời gian sinh ảnh Diffusion LoRA Style Transfer .....	34
Bảng 2.12 Bộ tham số đề xuất kích thước 256, 512.....	35
Bảng 2.13 Bộ tham số đề xuất kích thước 1024.....	36
Bảng 3.1 Yêu cầu phi chức năng.....	40
Bảng 3.2 Mô tả chi tiết bảng User.....	43
Bảng 3.3 Mô tả chi tiết bảng Password Resets.....	43
Bảng 3.4 Mô tả chi tiết bảng Tags.....	44
Bảng 3.5 Mô tả chi tiết bảng Taggings.....	44
Bảng 3.6 Mô tả chi tiết bảng Albums.....	44
Bảng 3.7 Mô tả chi tiết bảng Albumings.....	44
Bảng 3.8 Mô tả chi tiết bảng Img Contents.....	45
Bảng 3.9 Mô tả chi tiết bảng Img Styles .....	45
Bảng 3.10 Mô tả chi tiết bảng Arts.....	46
Bảng 3.11 Mô tả chi tiết bảng Comments .....	46
Bảng 3.12 Mô tả chi tiết bảng Arts Likeds.....	46
Bảng 3.13 Mô tả chi tiết bảng Art Notifies .....	46

Bảng 3.14 Mô tả chi tiết bảng Articles.....	47
Bảng 3.15 Mô tả chi tiết bảng Article Comments.....	47
Bảng 3.16 Mô tả chi tiết bảng Article Likeds .....	47
Bảng 3.17 Mô tả chi tiết bảng Article Notifies .....	47
Bảng 3.18 Mô tả chi tiết bảng Conversations.....	48
Bảng 3.19 Mô tả chi tiết bảng Messages.....	48
Bảng 3.20 Mô tả chi tiết bảng System Enums.....	48
Bảng 4.1 Triển khai hệ thống .....	81
Bảng 4.2 So sánh 3 mô hình.....	117
Bảng 4.3 Kết quả của 3 mô hình .....	117

## DANH SÁCH CÁC HÌNH VẼ

Hình 1.1 Ưu điểm của framework Laravel.....	5
Hình 1.2 Django backend framework .....	6
Hình 1.3 Ưu điểm của framework Vuejs.....	6
Hình 1.4 RESTful API.....	7
Hình 1.5 HTTP Status Codes .....	8
Hình 1.6 Git & Github .....	9
Hình 1.7 Version Control Github .....	10
Hình 1.8 Vercel & Railway Deploy .....	10
Hình 2.1 Lịch sử phát triển mô hình sinh ảnh nghệ thuật .....	12
Hình 2.2 Các lớp conv .....	13
Hình 2.3 Pipeline hoạt động của Neural Style Transfer.....	13
Hình 2.4 Quá trình tối ưu hóa sử dụng Gradient Descent.....	15
Hình 2.5 Cặp dữ liệu không đối xứng trong CycleGAN.....	16
Hình 2.6 Tổng quan kiến trúc CycleGAN.....	16
Hình 2.7 Kiến trúc Generator và Discriminator .....	17
Hình 2.8 Pipeline Stable Diffusion Model .....	19
Hình 2.9 Variational Autoencoders (VAEs) .....	19
Hình 2.10 Forward Diffusion (Quá trình khuếch tán tiến).....	20
Hình 2.11 Reverse Diffusion (Quá trình khuếch tán ngược) .....	20
Hình 2.12 Diffusion LoRA Style Transfer Pipeline Training .....	21
Hình 2.13 Diffusion LoRA Style Transfer Pipeline Inference.....	22
Hình 2.14 Dữ liệu huấn luyện Diffusion LoRA Style Transfer .....	23
Hình 2.15 Ảnh đầu ra của 200 epoch và 2.000 epoch (resize 1024).....	24
Hình 2.16 Ảnh đầu ra của 200 epoch và 1.000 epoch (resize 1024).....	24
Hình 2.17 Total Loss với các kích thước Resize.....	24
Hình 2.18 Total Loss với các giá trị Learning rate (resize 1024).....	25
Hình 2.19 So sánh kết quả đầu ra với các style image .....	26

Hình 2.20 So sánh trước và sau khi áp dụng cải tiến .....	26
Hình 2.21 Biểu đồ Total Loss chưa cải tiến .....	27
Hình 2.22 Biểu đồ Total Loss sau cải tiến.....	27
Hình 2.23 Biểu đồ thời gian sinh ảnh phương pháp Neural Style Transfer .....	28
Hình 2.24 Ma trận ảnh được tạo bởi bộ tham số tốt nhất.....	30
Hình 2.25 Biểu đồ loss khi train CycleGAN.....	30
Hình 2.26 So sánh ảnh đầu ra giữa 1 epoch và 5 epoch.....	31
Hình 2.27 So sánh ảnh đầu ra giữa resize 256 và resize 512 .....	31
Hình 2.28 Biểu đồ Style Loss Gram Matrix.....	32
Hình 2.29 Biểu đồ Loss LPIPS.....	33
Hình 2.30 Biểu đồ thời gian sinh ảnh Diffusion LoRA Style Transfer.....	34
Hình 2.31 Mối tương quan strength, image_cond_scale và kích thước resize .....	35
Hình 2.32 Ảnh tạo sinh kích thước 256 và 512.....	35
Hình 2.33 Ảnh tạo sinh kích thước 1024.....	36
Hình 2.34 Ma trận ảnh tạo sinh .....	36
Hình 3.1 Các tính năng cốt lõi và nổi bật của ArtGenix .....	39
Hình 3.2 Kiến trúc của hệ thống.....	40
Hình 3.3 Tech Stack .....	41
Hình 3.4 Database Diagrams .....	41
Hình 3.5 Lược đồ cơ sở dữ liệu.....	42
Hình 3.6 Chi tiết các bảng trong lược đồ cơ sở dữ liệu.....	43
Hình 3.7 Biểu đồ Use case tổng quát hệ thống.....	49
Hình 3.8 Biểu đồ Use case Guest và các tính năng liên quan .....	50
Hình 3.9 – 3.18 Danh sách biểu đồ Use case của User .....	50
Hình 3.19 – 3.27 Danh sách biểu đồ Use case của Admin.....	55
Hình 3.28 – 3.33 Danh sách biểu đồ tuần tự nhóm chức năng Xác thực .....	59
Hình 3.34 – 3.42 Danh sách biểu đồ tuần tự nhóm chức năng Quản lý nghệ thuật .....	62
Hình 3.43 – 3.50 Danh sách biểu đồ tuần tự nhóm chức năng Quản lý bài viết.....	66
Hình 3.51 – 3.53 Danh sách biểu đồ hoạt động nhóm chức năng Xác thực .....	70

Hình 3.54 – 3.57	Danh sách biểu đồ hoạt động nhóm chức năng của Admin	71
Hình 3.58 – 3.60	Danh sách biểu đồ hoạt động nhóm chức năng của User	71
Hình 3.61	Biểu đồ lớp User	75
Hình 3.62 – 3.63	Biểu đồ lớp Tag, Album	76
Hình 3.64 – 3.68	Danh sách biểu đồ lớp liên quan đến Art	76
Hình 3.69 – 3.71	Danh sách biểu đồ lớp liên quan đến Article	77
Hình 3.72	Biểu đồ lớp Message	80
Hình 4.1	Triển khai Client VueJS lên Vercel	82
Hình 4.2	Kết quả triển khai Client VueJS	82
Hình 4.3	Triển khai backend API Laravel lên Railway	83
Hình 4.4	Kết quả triển khai backend API Laravel	84
Hình 4.5	Triển khai cơ sở lưu trữ dữ liệu MySQL lên Railway	85
Hình 4.6	Kết quả triển khai cơ sở lưu trữ dữ liệu MySQL	85
Hình 4.7	Công khai API Model AI thông qua Flask & Ngrok trên Kaggle	85
Hình 4.8	Danh sách API của hệ thống	86
Hình 4.9	Trang chủ	87
Hình 4.10 – 4.13	Nhóm chức năng Xác thực và Cài đặt tài khoản	87
Hình 4.14 – 4.16	Nhóm chức năng Quản lý người dùng	87
Hình 4.17 – 4.22	Nhóm chức năng liên quan đến Tag	88
Hình 4.23 – 4.31	Nhóm chức năng liên quan đến Img Content	88
Hình 4.32 – 4.40	Nhóm chức năng liên quan đến Img	88
Hình 4.41 – 4.49	Nhóm chức năng liên quan đến Art	89
Hình 4.50 – 4.58	Nhóm chức năng liên quan đến Article	89
Hình 4.59 – 4.66	Nhóm chức năng liên quan đến Album	89
Hình 4.67 – 4.70	Nhóm chức năng Tạo sinh ảnh nghệ thuật	90
Hình 4.71 – 4.76	Nhóm chức năng Album Art, Img Content, Img Style	90
Hình 4.77 – 4.79	Nhóm chức năng Nhấn tin, quản lý thông báo	90
Hình 4.80 – 4.91	Nhóm chức năng Trang chủ, bài viết, sinh ảnh và khám phá	91
Hình 4.92	Admin xem thống kê của hệ thống	116

Hình 4.93 Kết quả Neural Style Transfer .....	117
Hình 4.94 Kết quả CycleGAN.....	117
Hình 4.95 Kết quả Diffusion LoRA Style Transfer .....	118
Hình 4.96 Kiểm thử tự động E2E bằng Cypress .....	119
Hình 4.97 Kiểm thử tự động E2E chức năng Authentication .....	119
Hình 4.98 Kiểm thử tự động E2E chức năng Cài đặt tài khoản .....	120
Hình 4.99 Kiểm thử tự động E2E chức năng Quản lý tài khoản người dùng .....	120
Hình 4.100 Kiểm thử tự động E2E chức năng Quản lý tài khoản admin.....	121
Hình 4.101 Kiểm thử tự động E2E chức năng Kích hoạt tài khoản admin.....	121
Hình 4.102 Kiểm thử tự động E2E chức năng Quản lý bài viết .....	121
Hình 4.103 Kiểm thử tự động E2E chức năng Xem, thêm bài viết .....	122
Hình 4.104 Kiểm thử tự động E2E chức năng Xóa, sửa bài viết .....	122
Hình 4.105 Kiểm thử tự động cho API server bằng Unit Test và Feature Test .....	123

## DANH SÁCH CÁC CÔNG THỨC

Công thức 2.1 Total Loss .....	13
Công thức 2.2 Content Loss .....	14
Công thức 2.3 Style Loss trong Neural Style Transfer .....	14
Công thức 2.4 Anisotropic Total Variation Loss .....	15
Công thức 2.5 Total Loss sau tối ưu.....	15
Công thức 2.6 Total Loss CycleGAN .....	17

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

<b>Tên</b>	<b>Mô tả</b>
API	Application programming interface
NST	Neural Style Transfer
LPIPS	Learned Perceptual Image Patch Similarity
LoRA	Low-Rank Adaptation
GAN	Generative Adversarial Network
VGG	Visual Geometry Group
DDPM	Denoising Diffusion Probabilistic Models
MSE	Mean Squared Error
GN	Group Normalization
GLIDE	Guided Language-to-Image Diffusion
CLIP	Contrastive Language-Image Pre-training

## MỞ ĐẦU

### 1. Tổng quan

Dự án này nhằm mục đích giới thiệu ArtGenix, một nền tảng sáng tạo ảnh nghệ thuật sử dụng các mô hình học sâu tiên tiến như Neural Style Transfer, CycleGAN và Diffusion LoRA, cho phép người dùng chuyển đổi ảnh thông thường thành các tác phẩm nghệ thuật độc đáo. Dự án đáp ứng nhu cầu ngày càng tăng về các công cụ hỗ trợ sáng tạo nghệ thuật số, đồng thời tích hợp các tính năng mạng xã hội như chia sẻ, bình luận, thích và nhắn tin trực tuyến, mang lại trải nghiệm liền mạch và tương tác.

Vì vậy, quyết định theo đuổi dự án ArtGenix xuất phát từ nhận thức về những hạn chế của các nền tảng hiện có trong việc kết hợp công nghệ học sâu với khả năng xây dựng cộng đồng sáng tạo. Hơn nữa, nghiên cứu về tạo sinh nói riêng và tạo sinh ảnh nghệ thuật nói chung hiện là một trong những chủ đề vẫn còn khá mới, thu hút sự quan tâm lớn từ cộng đồng khoa học và công nghệ. Việc phát triển ArtGenix không chỉ giải quyết các nhu cầu thực tiễn mà còn đặt nền móng cho các nghiên cứu sâu hơn trong lĩnh vực tạo sinh, mở ra tiềm năng khám phá mới trong tương lai.

### 2. Mục đích

Mục đích chính của ArtGenix là trao quyền cho người dùng, từ những người yêu nghệ thuật đến các nhà sáng tạo chuyên nghiệp, tạo ra các tác phẩm nghệ thuật số chất lượng cao. Ứng dụng không chỉ hỗ trợ sáng tạo cá nhân mà còn mở rộng ứng dụng trong giáo dục mỹ thuật, hội họa, giúp học sinh và giáo viên khám phá các phong cách nghệ thuật, cũng như trong ngành công nghiệp in ấn, tạo ra các mẫu thiết kế độc đáo cho áo quần, đồ dùng và sản phẩm thương mại. Bằng cách tích hợp các tính năng xã hội, ArtGenix xây dựng một cộng đồng trực tuyến nơi người dùng có thể kết nối, chia sẻ ý tưởng và truyền cảm hứng lẫn nhau, từ đó thúc đẩy sự sáng tạo và tương tác.

### 3. Phạm vi, đối tượng và phương pháp nghiên cứu

#### a. Phạm vi

– **Hệ thống sinh ảnh nghệ thuật:**

- Phát triển ứng dụng ArtGenix sử dụng các mô hình học sâu tiên tiến như Neural Style Transfer, CycleGAN và Diffusion LoRA để chuyển đổi ảnh thông thường thành các tác phẩm nghệ thuật độc đáo.
- Cung cấp giao diện thân thiện cho người dùng, hỗ trợ tải lên ảnh nội dung và ảnh phong cách, tùy chỉnh các thông số sinh ảnh.
- Tích hợp các tính năng mạng xã hội, bao gồm chia sẻ, bình luận, thích các tác phẩm nghệ thuật, và nhắn tin trực tuyến giữa người dùng.

- Quản lý tài nguyên người dùng, bao gồm tác phẩm nghệ thuật, bộ sưu tập, và thông báo liên quan đến hoạt động trên nền tảng.
- **Ứng dụng thực tiễn:**
  - Hỗ trợ giáo dục mỹ thuật, hội họa thông qua việc cung cấp công cụ sáng tạo để học sinh và giáo viên khám phá các phong cách nghệ thuật.
  - Ứng dụng trong ngành công nghiệp in ấn, tạo ra các mẫu thiết kế độc đáo cho áo quần, đồ dùng và sản phẩm thương mại.
  - Ứng dụng trong sáng tác truyện tranh.
  - Xây dựng các api để sử dụng làm bộ lọc ảnh cho các ứng dụng chỉnh sửa ảnh trên web, mobile.
- **Quản lý hệ thống:**
  - Phát triển giao diện quản trị cho Admin để quản lý người dùng, tác phẩm nghệ thuật, thẻ, và thống kê hoạt động hệ thống.
  - Hỗ trợ vai trò Manager/Super Admin để giám sát và quản lý đội ngũ Admin.

## b. Đối tượng nghiên cứu

- Người dùng yêu nghệ thuật và nhà sáng tạo: Các cá nhân quan tâm đến sáng tạo nghệ thuật số, từ người mới bắt đầu đến các nhà sáng tạo chuyên nghiệp, mong muốn sử dụng công nghệ học sâu để tạo ra các tác phẩm độc đáo.
- Cơ sở giáo dục mỹ thuật: Các trường học, trung tâm đào tạo mỹ thuật cần công cụ hỗ trợ giảng dạy và khám phá phong cách nghệ thuật.
- Doanh nghiệp in ấn và thiết kế: Các công ty trong lĩnh vực thời trang, nội thất, và sản phẩm thương mại có nhu cầu tạo mẫu thiết kế sáng tạo.
- Cộng đồng trực tuyến: Những người dùng muốn tham gia vào một nền tảng kết hợp sáng tạo nghệ thuật và tương tác xã hội.

## c. Phương pháp nghiên cứu

- Khảo sát nhu cầu thực tế từ người dùng yêu nghệ thuật, giáo viên mỹ thuật, và doanh nghiệp in ấn để xác định các yêu cầu chức năng của ArtGenix.
- Nghiên cứu các mô hình học sâu (Neural Style Transfer, CycleGAN, Diffusion LoRA) và các kỹ thuật tối ưu hóa để đảm bảo chất lượng sinh ảnh.
- Tham khảo các nền tảng tương tự (như MidJourney, Artbreeder) để phân tích ưu điểm, hạn chế và xây dựng sản phẩm độc đáo.
- Thu thập và phân tích dữ liệu ảnh (ảnh nội dung, ảnh phong cách) để huấn luyện và đánh giá hiệu quả của các mô hình học sâu.

- Tìm hiểu các công nghệ phát triển ứng dụng, bao gồm Python, TensorFlow, PyTorch, Keras (cho học sâu), VueJS, Laravel (cho giao diện người dùng) và MySQL (cho cơ sở dữ liệu).
- Nghiên cứu công cụ hỗ trợ phát triển như GitHub (quản lý mã nguồn), và DevOps (triển khai hệ thống).
- Thực hiện phát triển và thử nghiệm hệ thống ArtGenix, bao gồm các module sinh ảnh, mạng xã hội, và quản trị.
- Đánh giá hiệu quả của hệ thống thông qua phản hồi từ người dùng thử nghiệm và phân tích chất lượng tác phẩm nghệ thuật sinh ra.

#### 4. Công nghệ và môi trường phát triển

##### a. Công nghệ

- Laravel
- Vuejs
- MySQL
- Python
- Flask
- Django

##### b. Công cụ và môi trường phát triển ứng dụng

- Docker
- Git & Github
- Kaggle & Ngrok
- Vercel
- Railway

#### 5. Cấu trúc của đề án tốt nghiệp

**GIỚI THIỆU** – Chương này cung cấp cái nhìn tổng quan về dự án ArtGenix, nêu bật bối cảnh, ý nghĩa và tầm quan trọng của việc phát triển một nền tảng sinh ảnh nghệ thuật sử dụng công nghệ học sâu. Phần giới thiệu xác định các mục tiêu chính, phạm vi và sứ mệnh của dự án, đồng thời đặt nền tảng cho các chương tiếp theo bằng cách trình bày các phương pháp để đạt được mục tiêu đề ra.

**CHƯƠNG 1: LÝ THUYẾT VÀ CÔNG NGHỆ** – Chương này đi sâu vào các lý thuyết và công nghệ tri thức đa dạng được áp dụng trong dự án. Nó phân tích các cơ sở lý thuyết làm nền tảng cho việc thiết kế và triển khai dự án, đồng thời giới thiệu các công cụ và khung công nghệ được sử dụng để hiện thực hóa tầm nhìn.

**CHƯƠNG 2: MÔ HÌNH ĐỀ XUẤT** – Chương này tập trung vào việc nghiên cứu về cơ sở lý thuyết, làm rõ nguyên lý hoạt động của từng mô hình, dựa trên các công trình nghiên cứu liên quan. Tiếp theo, chương phân tích các cải tiến và tối ưu hóa được thực hiện so với các phương pháp gốc, nhằm nâng cao chất lượng ảnh nghệ thuật. Ngoài ra, chương mô tả chi tiết về tập dữ liệu, quy trình huấn luyện, phương pháp đánh giá hiệu quả và kết quả đầu ra của các mô hình Neural Style Transfer, CycleGAN và Diffusion LoRA. Nhấn mạnh vào độ chân thực, tính thẩm mỹ và khả năng sáng tạo của các tác phẩm.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** – Chương này tập trung vào giai đoạn phân tích và thiết kế của dự án, bao gồm việc xác định các tính năng cốt lõi, thiết lập thông số kỹ thuật cho yêu cầu phần mềm và xây dựng thiết kế cấu trúc cơ sở dữ liệu làm nền tảng cho dự án.

**CHƯƠNG 4: KẾT QUẢ TRIỂN KHAI** – Chương này trình bày chi tiết quy trình từng bước triển khai dự án, nêu bật các mốc quan trọng đạt được và cung cấp ảnh chụp màn hình minh họa hệ thống trong trạng thái hoạt động. Với việc tích hợp phần mô hình sinh ảnh và nội dung kiểm thử, chương đưa ra đánh giá toàn diện về quá trình triển khai, làm rõ các kết quả hữu hình từ nỗ lực phát triển, bao gồm hiệu quả của API backend, giao diện người dùng, mô hình sinh ảnh và kết quả kiểm thử.

**KẾT LUẬN & HƯỚNG PHÁT TRIỂN** – Phần kết luận đóng vai trò là phần cuối cùng của dự án, tóm tắt các phát hiện và kết quả chính. Phần này nhấn mạnh các vấn đề đã được giải quyết thành công thông qua dự án, đồng thời thừa nhận bất kỳ vấn đề nào còn chưa được giải quyết. Ngoài ra, phần kết luận đưa ra các cơ hội để phát triển và cải thiện hơn nữa, chỉ ra các hướng tiềm năng cho các cải tiến trong tương lai.

**TÀI LIỆU THAM KHẢO** – Phần này liệt kê các tài liệu tham khảo được sử dụng trong đề án, bao gồm sách, bài báo khoa học về học sâu, tài liệu kỹ thuật về các công nghệ phát triển và các nguồn tham khảo từ các nền tảng tương tự.

## CHƯƠNG 1: LÝ THUYẾT VÀ CÔNG NGHỆ

Chương này cung cấp một cái nhìn tổng quan về các lý thuyết thực hiện liên quan đến việc phát triển dự án ArtGenix. Nó bao gồm các kiến thức kỹ thuật cần thiết cho cả phía máy chủ, phía máy khách và hệ quản trị cơ sở dữ liệu.

### 1.1. Laravel backend framework

Laravel là PHP Web Framework miễn phí, mã nguồn mở, được tạo bởi Taylor Otwell và dành cho việc phát triển các ứng dụng web theo mô hình kiến trúc mô hình MVC và dựa trên Symfony PHP Framework. Một số tính năng của Laravel như là sử dụng hệ thống đóng gói module, quản lý package (Composer), hỗ trợ nhiều hệ quản trị CSDL quan hệ (MySQL, MariaDB, SQLite, PostgreSQL,...), các tiện ích hỗ trợ triển khai và bảo trì ứng dụng.



Hình 1.1 Ưu điểm của framework Laravel

#### Điểm mạnh của Laravel:

- Hỗ trợ tính năng hiện đại của PHP, tận dụng Namespaces, Interfaces, Overloading, Anonymous functions và cú pháp mảng ngắn gọn, đi kèm tài liệu phong phú cho mỗi phiên bản.
- Tích hợp dịch vụ mail mạnh mẽ, sử dụng API SwiftMailer cho dịch vụ mail đám mây hoặc local, xử lý nhanh, phù hợp cho website và dự án lớn. Tuân theo mô hình MVC, dễ sử dụng, bảo mật cao, hỗ trợ cache backend như Memcached và Redis ngay từ đầu.
- Công cụ Artisan tích hợp dòng lệnh mạnh mẽ, hỗ trợ tạo liên kết đến các route được đặt tên. Rút ngắn chu kỳ phát triển tích hợp nhanh, được hỗ trợ bởi cộng đồng Laracasts.

### 1.2. Django backend framework

Django là framework backend mã nguồn mở viết bằng Python, hỗ trợ phát triển ứng dụng web nhanh, an toàn. Tuân theo mô hình MVC với triết lý "batteries included",

Django tích hợp sẵn nhiều tính năng hữu ích. Nó đặc biệt phù hợp cho việc xây dựng API, hỗ trợ mạnh mẽ các ứng dụng AI nhờ tích hợp với các thư viện Python.



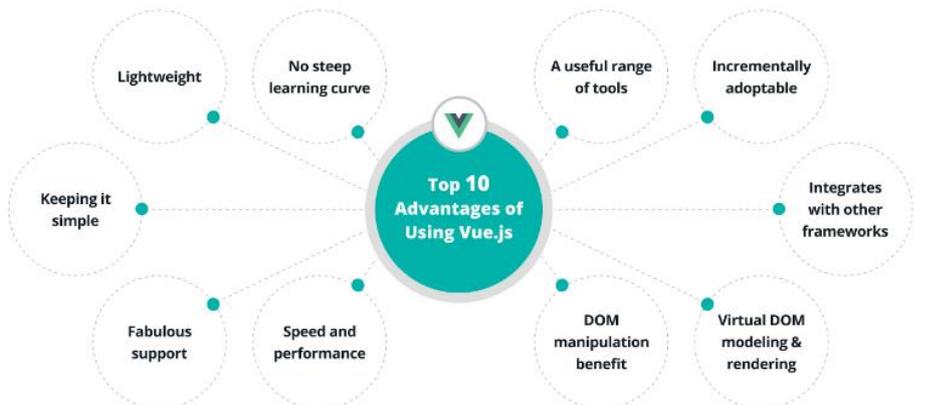
Hình 1.2 Django backend framework

### Điểm mạnh của Django:

- Hỗ trợ REST API mạnh mẽ, Django REST Framework (DRF) cho phép tạo API nhanh, tích hợp tốt với các thư viện AI như TensorFlow, PyTorch, Scikit-learn. Tài liệu chi tiết và cộng đồng lớn hỗ trợ phát triển, triển khai API cho ứng dụng AI.
- Hiệu suất tối ưu với cơ sở dữ liệu, tối ưu hóa truy vấn, phù hợp để quản lý dữ liệu huấn luyện và dự đoán. Tùy chỉnh linh hoạt dễ dàng tùy chỉnh endpoint API, đáp ứng yêu cầu phức tạp như xử lý dữ liệu thời gian thực.
- Bảo mật cao, cung cấp bảo vệ chống CSRF, XSS, SQL injection, đảm bảo an toàn khi xử lý dữ liệu.

### 1.3. VueJS frontend framework

Vue.js là một framework JavaScript mã nguồn mở, được phát triển bởi Evan You và cộng đồng, tập trung vào việc xây dựng giao diện người dùng và ứng dụng web tương tác một cách hiệu quả. Nó nổi bật với cách tiếp cận linh hoạt, sử dụng mô hình thành phần (component-based) và tích hợp dễ dàng với các dự án hiện có. Một số tính năng chính của Vue.js bao gồm hệ thống phản ứng (reactivity), Virtual DOM, và hỗ trợ các công cụ như Vue CLI và Vue Router.



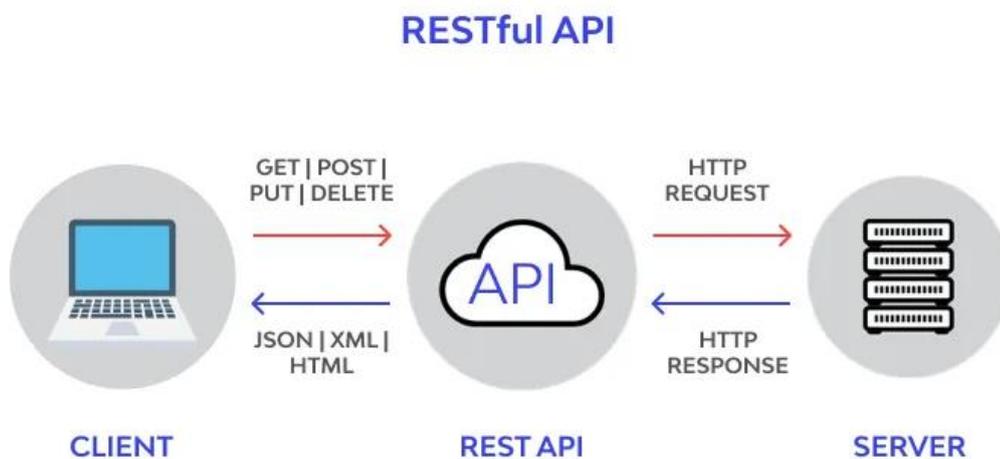
Hình 1.3 Ưu điểm của framework Vuejs

### Điểm mạnh của VueJS:

- Cú pháp đơn giản, Vue.js cung cấp cú pháp dễ học và dễ sử dụng, giúp các nhà phát triển nhanh chóng làm quen. Tài liệu tốt, cung cấp tài liệu chi tiết và rõ ràng, hỗ trợ người dùng tra cứu hiệu quả cho từng phiên bản.
- Ràng buộc dữ liệu, hỗ trợ dữ liệu hai chiều (two-way data binding), giúp đồng bộ hóa giao diện và logic dễ dàng. DOM ảo (Virtual DOM), tối ưu hóa hiệu suất bằng cách cập nhật chỉ những phần thay đổi của giao diện. Chuyển tiếp hấp dẫn, cung cấp các hiệu ứng chuyển tiếp mượt mà, nâng cao trải nghiệm người dùng. Hiệu suất cao, đảm bảo tốc độ xử lý nhanh, phù hợp cho cả ứng dụng nhỏ lẫn lớn.
- Nhẹ (Lightweight), kích thước nhỏ gọn, dễ tích hợp mà không làm tăng tải trọng ứng dụng. Linh hoạt, dễ dàng tích hợp với các dự án hiện có hoặc sử dụng cùng các framework khác.

### 1.4. RESTful API

RESTful API (Representational State Transfer) là một phong cách kiến trúc được sử dụng để thiết kế các ứng dụng mạng. Nó dựa trên một tập hợp các nguyên tắc và ràng buộc nhằm hỗ trợ giao tiếp giữa hệ thống client và server.



Hình 1.4 RESTful API

#### Các thành phần và khái niệm chính liên quan đến RESTful API:

- Resources (Tài nguyên): Trong REST, tài nguyên là các thực thể cơ bản được lộ ra thông qua API. Tài nguyên có thể là bất kỳ thông tin hoặc dữ liệu nào mà client muốn tương tác, chẳng hạn như người dùng, sản phẩm, đơn hàng, v.v. Mỗi tài nguyên được xác định bởi một URL (Uniform Resource Locator) duy nhất.
- HTTP Methods (Phương thức HTTP): RESTful API sử dụng các phương thức HTTP khác nhau để chỉ định loại hành động thực hiện trên tài nguyên. Các phương thức thường được sử dụng bao gồm:

- GET: Lấy về biểu diễn của một tài nguyên hoặc tập hợp tài nguyên từ server.
- POST: Gửi dữ liệu lên server để tạo mới một tài nguyên.
- PUT: Cập nhật một tài nguyên hiện có với dữ liệu mới do client cung cấp.
- DELETE: Xóa một tài nguyên khỏi server.
- PATCH: Cập nhật một phần của tài nguyên hiện có với dữ liệu mới do client cung cấp.
- Status Codes (Mã trạng thái): Mã trạng thái HTTP là các mã số ba chữ số được trả về trong phản hồi HTTP để chỉ ra kết quả của một yêu cầu. Một số mã trạng thái phổ biến trong RESTful API bao gồm:
  - 200 OK: Yêu cầu đã thành công.
  - 201 Created: Tài nguyên đã được tạo thành công.
  - 400 Bad Request: Yêu cầu bị sai cú pháp hoặc không hợp lệ.
  - 401 Unauthorized: Yêu cầu xác thực hoặc thông tin xác thực không hợp lệ.
  - 404 Not Found: Tài nguyên được yêu cầu không tồn tại.
  - 500 Internal Server Error: Một lỗi đã xảy ra trên server.

## HTTP Status Codes



Hình 1.5 HTTP Status Codes

### Một số API mẫu của hệ thống

Method	URL	Mô tả
POST	/api/auth/login	Người dùng đăng nhập
POST	/api/auth/register	Người dùng đăng ký
GET	/api/auth/logout	Người dùng đăng xuất
POST	/api/auth/login-google	Đăng nhập bằng tài khoản Google
POST	/api/auth/forgot-pw-sendcode	Gửi mã để đặt lại mật khẩu
POST	/api/auth/forgot-update	Đặt lại mật khẩu
GET	/api/auth/profile	Nhận thông tin tài khoản
POST	/api/auth/change-password	Thay đổi mật khẩu
POST	/api/auth/update-profile	Cập nhật tài khoản
POST	/api/auth/create-pw-google	Tạo mật khẩu cho tài khoản google
POST	/api/admin/user/add-admin	Thêm tài khoản admin

POST	/api/admin/user/change-role-admin	Thay đổi quyền của tài khoản admin
GET	/api/admin/user/manage-admin	Manager, Super admin quản lý admin
GET	/api/admin/user/change-active-admin	Thay đổi trạng thái hoặc động của tài khoản admin

Bảng 1.1 Một số API mẫu của hệ thống

## 1.5. Git & Github

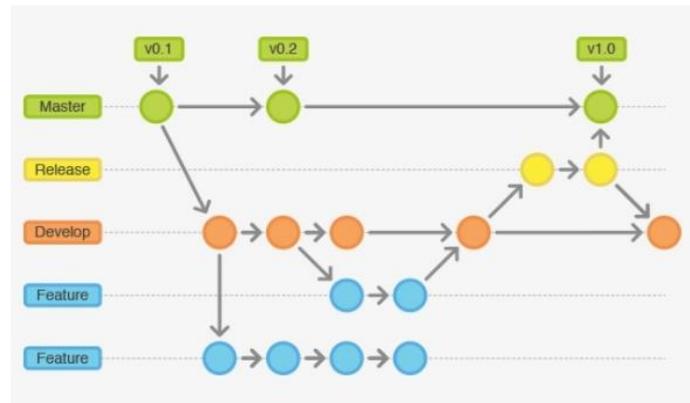
Git là một hệ thống kiểm soát phiên bản phân tán, được phát triển bởi Linus Torvalds, dùng để quản lý mã nguồn và theo dõi thay đổi trong dự án phát triển phần mềm. GitHub là một nền tảng trực tuyến dựa trên Git, cung cấp dịch vụ lưu trữ mã nguồn, quản lý dự án, và cộng tác nhóm. Cả hai công cụ này đóng vai trò quan trọng trong việc phát triển phần mềm, đặc biệt với các đội ngũ lập trình viên.



Hình 1.6 Git & Github

### Các thành phần và khái niệm chính liên quan đến Git & GitHub:

- **Repositories (Kho mã nguồn):** Trong Git, một repository là nơi lưu trữ toàn bộ lịch sử thay đổi của dự án. Trên GitHub, repository được lưu trữ trên đám mây và có thể được chia sẻ hoặc sao chép (fork) để cộng tác.
- **Commits (Cam kết):** Git sử dụng commits để ghi lại các thay đổi trong mã nguồn, mỗi commit được gắn với một thông điệp mô tả, giúp theo dõi lịch sử phát triển.
- **Branches (Nhánh):** Git hỗ trợ tạo các nhánh riêng biệt để phát triển tính năng hoặc sửa lỗi mà không ảnh hưởng đến mã nguồn chính, với GitHub cho phép quản lý nhánh qua giao diện trực quan.
- **Pull Requests (Yêu cầu kéo):** Trên GitHub, pull requests là cơ chế đề xuất và xem xét mã nguồn từ một nhánh, giúp đội ngũ hợp tác và kiểm tra chất lượng trước khi tích hợp.
- **Collaboration Features (Tính năng cộng tác):** GitHub cung cấp các công cụ như issues, wikis, và actions để quản lý nhiệm vụ, tài liệu hóa, và tự động hóa quy trình làm việc, hỗ trợ phát triển hiệu quả.



Hình 1.7 Version Control Github

## 1.6. Vercel & Railway

Vercel và Railway là hai nền tảng dịch vụ Platform-as-a-Service (PaaS) được thiết kế để đơn giản hóa quá trình triển khai và quản lý ứng dụng web, đặc biệt phù hợp với các nhà phát triển frontend và backend. Vercel tập trung vào trải nghiệm phát triển giao diện người dùng, trong khi Railway cung cấp một giải pháp toàn diện cho cả frontend và backend. Cả hai đều hỗ trợ tích hợp với Git và cung cấp quy trình triển khai liền mạch.



Hình 1.8 Vercel & Railway Deploy

### Điểm mạnh của Vercel & Railway:

- Triển khai dễ dàng, tự động triển khai từ Git repository, Vercel tối ưu cho static và serverless, Railway hỗ trợ microservices phức tạp.
- Hàm serverless Vercel cung cấp serverless functions cho backend logic, Railway hỗ trợ container serverless cho ứng dụng dài hạn. Hiệu suất toàn cầu, Vercel sử dụng Global CDN để tăng tốc tải trang, Railway dùng mạng Wireguard riêng tư để tối ưu hiệu suất.
- Tích hợp cơ sở dữ liệu, Railway hỗ trợ native PostgreSQL, MySQL, Redis, Vercel tích hợp dễ dàng với Railway cho cơ sở dữ liệu. Khả năng mở rộng, cả hai tự động scale tài nguyên, Vercel dựa trên lượt gọi hàm, Railway dựa trên CPU, bộ nhớ, lưu trữ. Trải nghiệm nhà phát triển, Vercel hỗ trợ tốt Vue.js với giao diện trực quan, Railway cung cấp quy trình phát triển liền mạch, ít cấu hình.

## CHƯƠNG 2: MÔ HÌNH ĐỀ XUẤT

Chương này trình bày chi tiết các phương pháp, mô hình học sâu được ứng dụng trong việc sinh ảnh nghệ thuật cho hệ thống ArtGenix, bao gồm Neural Style Transfer, CycleGAN và Diffusion LoRA. Chương tập trung vào việc phân tích lịch sử phát triển, cơ sở lý thuyết, kiến trúc cơ bản, quá trình huấn luyện, đánh giá, cũng như các cải tiến nhằm nâng cao hiệu quả của các mô hình này trong việc hỗ trợ người dùng tạo ra các tác phẩm nghệ thuật độc đáo. Đồng thời, các nội dung được triển khai nhằm đặt nền tảng cho việc tích hợp và tối ưu hóa trong các giai đoạn phát triển tiếp theo của dự án.

### 2.1. Phát biểu bài toán

Dự án ArtGenix đặt ra bài toán nghiên cứu các mô hình học sâu để sinh ảnh nghệ thuật, chuyển đổi ảnh thông thường thành các tác phẩm mang phong cách đặc trưng như tranh vẽ, tranh sơn dầu hoặc phong cách của các họa sĩ nổi tiếng (Monet, Van Gogh,...). Nghiên cứu hướng đến việc đề xuất một phương pháp sinh ảnh tối ưu cho hệ thống.

**Bài toán được định nghĩa như sau:**

- Đầu vào: Một ảnh thông thường làm ảnh nội dung.
- Đầu ra: Một ảnh nghệ thuật mang phong cách đặc trưng, chẳng hạn như tranh vẽ, tranh sơn dầu, hoặc phong cách của một nghệ sĩ cụ thể.
- Mục tiêu: Phát triển một hệ thống hiệu quả, tạo ra ảnh nghệ thuật chất lượng cao, thời gian xử lý nhanh và giao diện thân thiện, cho phép người dùng dễ dàng tùy chỉnh kết quả.

**Để giải quyết bài toán đặt ra, đề xuất nghiên cứu và đánh giá ba phương pháp:**

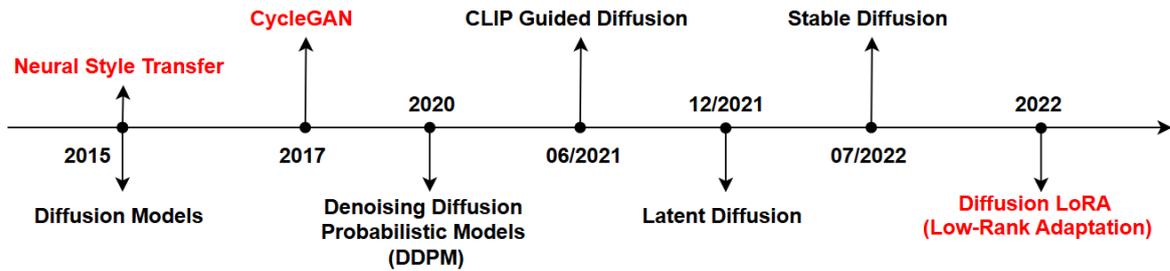
1. Neural Style Transfer: Chuyển đổi ảnh nội dung dựa trên một phong cách cụ thể, thường yêu cầu một ảnh phong cách tham chiếu.
2. CycleGAN: Học chuyển đổi phong cách từ tập dữ liệu mà không cần ảnh phong cách trực tiếp, phù hợp cho các ứng dụng hàng loạt.
3. Diffusion LoRA: Sinh ảnh nghệ thuật dựa trên ảnh nội dung và mô tả văn bản, mang lại tính linh hoạt trong việc điều chỉnh phong cách.

**Nghiên cứu tập trung vào việc so sánh các phương pháp dựa trên các tiêu chí:**

- Chất lượng ảnh: Đánh giá qua Gram Matrix để đo lường sự tương đồng phong cách và LPIPS để đo lường chất lượng tổng thể.
- Thời gian xử lý: Đảm bảo phù hợp với các ứng dụng thực tế.
- Tính linh hoạt: Khả năng đáp ứng các nhu cầu khác nhau của người dùng.

## 2.2. Lịch sử phát triển

Lĩnh vực sinh ảnh nghệ thuật bằng học sâu đã trải qua nhiều giai đoạn phát triển quan trọng, từ các phương pháp ban đầu dựa trên mạng nơ-ron tích chập (CNN) đến các mô hình phức tạp hơn như GAN và mô hình khuếch tán. Ba phương pháp chính được nghiên cứu bao gồm Neural Style Transfer (NST), CycleGAN và Diffusion LoRA đại diện cho các bước tiến quan trọng trong lĩnh vực này.



Hình 2.1 Lịch sử phát triển mô hình sinh ảnh nghệ thuật

- Neural Style Transfer (NST): Được giới thiệu bởi Gatys et al. vào năm 2015, là một trong những phương pháp đầu tiên sử dụng CNN để tách biệt và kết hợp nội dung và phong cách của hai ảnh khác nhau. Mở ra hướng nghiên cứu mới về khả năng sáng tạo của máy tính trong nghệ thuật.
- CycleGAN: Ra đời năm 2017 bởi Zhu et al., CycleGAN là một biến thể của Generative Adversarial Networks (GAN) cho phép chuyển đổi phong cách giữa hai tập ảnh mà không cần dữ liệu ghép cặp. Nâng cao phạm vi ứng dụng của GAN trong việc sinh ảnh nghệ thuật tự động.
- Diffusion LoRA: Phát triển gần đây vào khoảng năm 2022, Diffusion LoRA là một phương pháp tiên tiến dựa trên mô hình khuếch tán (Diffusion Models), cụ thể là Stable Diffusion, kết hợp với kỹ thuật Low-Rank Adaptation (LoRA) để tinh chỉnh mô hình. Phương pháp này đại diện cho xu hướng mới trong việc tạo ảnh nghệ thuật với khả năng tùy chỉnh cao thông qua văn bản.

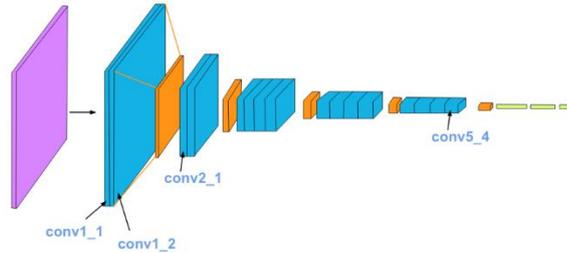
## 2.3. Neural Style Transfer

### 2.3.1. Kiến trúc Neural Style Transfer

Neural Style Transfer (NST) nhằm đến mục tiêu tái tạo ảnh mới bằng cách tích hợp đặc trưng nội dung từ ảnh nguồn và đặc trưng phong cách từ ảnh tham chiếu, phương pháp này sử dụng mạng nơ-ron tích chập (CNN), thường là VGG16 hoặc VGG19 đã huấn luyện trên ImageNet, với cấu trúc sâu gồm nhiều lớp tích chập, để trích xuất đặc trưng. Ảnh đầu ra giữ nguyên bố cục từ ảnh gốc, đồng thời mang màu sắc, kết cấu và nét họa tiết đặc trưng từ ảnh phong cách.

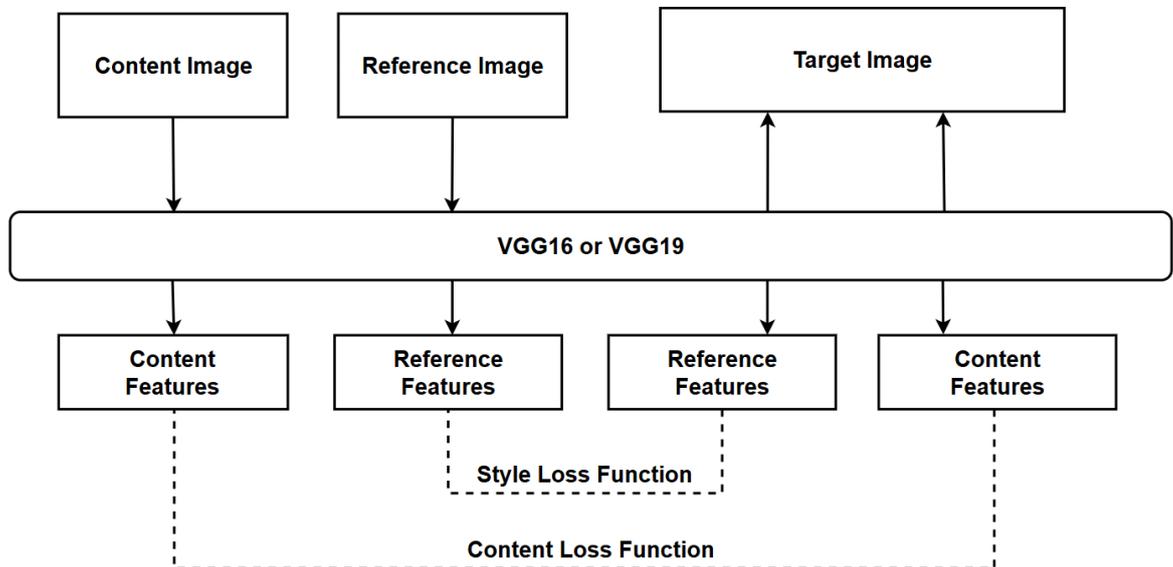
– **Pipeline hoạt động của NST bao gồm:**

- **Trích xuất đặc trưng:** Nội dung được lấy từ tầng sâu (như conv4\_2, conv5\_2) với feature maps cấp cao phản ánh bố cục và chi tiết tổng thể, trong khi phong cách được trích xuất từ tầng nông (như conv1\_1, conv2\_1, conv3\_1) thông qua ma trận Gram, đo lường tương quan giữa các kênh đặc trưng để biểu diễn kết cấu và bảng màu.



Hình 2.2 Các lớp conv

- **Tối ưu hóa ảnh đầu ra:** Khởi tạo ảnh sinh từ nhiễu ngẫu nhiên hoặc ảnh nội dung gốc, sau đó áp dụng thuật toán Gradient descent để cập nhật liên tục, cân bằng nội dung và phong cách dựa trên hàm mất mát tổng quát. Quá trình này lặp lại nhiều vòng để ảnh sinh hội tụ, phản ánh cả đặc trưng nội dung và phong cách.



Hình 2.3 Pipeline hoạt động của Neural Style Transfer

– **Hàm mất mát tổng quát được định nghĩa như sau:**

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

Công thức 2.1 Total Loss

- Content Loss: Đảm bảo ảnh đầu ra giữ chi tiết và bố cục chính của ảnh nội dung, được tính bằng sai số bình phương giữa đặc trưng tại tầng chọn.

$$\mathcal{L}_{content} = \frac{1}{2} \sum_{i,j} (F_{ij}^{gen} - F_{ij}^{content})^2$$

Công thức 2.2 Content Loss

Trong đó:

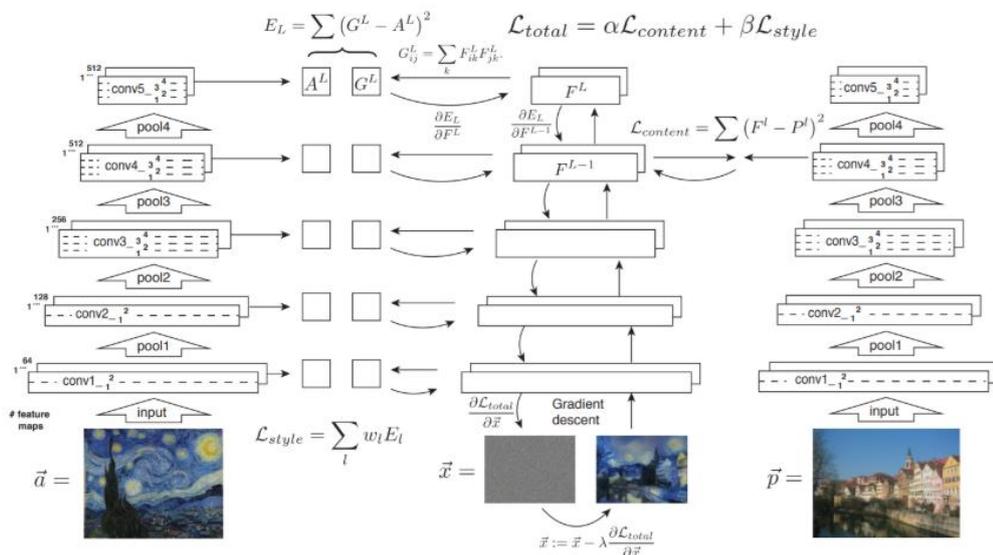
- $F_{ij}$  là đặc trưng tại tầng sâu, đo sự khác biệt giữa ảnh sinh và ảnh nội dung.
- Style Loss: Đảm bảo ảnh đầu ra mang đặc trưng phong cách (như nét cọ, họa tiết), được tính bằng sự khác biệt giữa ma trận Gram của ảnh đầu ra và ảnh phong cách.

$$\mathcal{L}_{style} = \sum_l w_l \times \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^{gen} - G_{ij}^{style})^2$$

Công thức 2.3 Style Loss trong Neural Style Transfer

Trong đó:

- $G_{ij}$  là ma trận Gram tại tầng  $l$ .
- $N_l$  và  $M_l$  lần lượt là số kênh và số phân tử,  $w_l$  là trọng số điều chỉnh tầm quan trọng của từng tầng.
- $\alpha, \beta$ : Trọng số điều chỉnh mức độ ưu tiên nội dung và phong cách.
- Quá trình tối ưu hóa sử dụng Gradient Descent để giảm thiểu  $\mathcal{L}_{total}$ , lặp lại qua nhiều vòng để tạo ảnh đầu ra vừa giữ nội dung gốc vừa mang phong cách nghệ thuật.



Hình 2.4 Quá trình tối ưu hóa sử dụng Gradient Descent

### 2.3.2. Đề xuất cải tiến

Cải tiến so với phương pháp gốc của Gatys et al. (2015) tập trung vào việc tích hợp Total Variation Loss (TV Loss) vào hàm mất mát tổng, nhằm giảm nhiễu và làm mịn chi tiết không tự nhiên trong ảnh đầu ra, từ đó nâng cao chất lượng thẩm mỹ và tính ổn định của quá trình tối ưu hóa.

#### a. Cơ sở lý thuyết

TV Loss là một hàm mất mát truyền thống trong xử lý ảnh, xuất phát từ công trình của Rudin et al. (1992), được thiết kế để giảm các biến thiên đột ngột giữa các pixel lân cận. Trong không gian rời rạc, TV Loss có thể được tính như sau:

$$L_{TV}(x) = \sum_{i,j} \left( (x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2 \right)$$

Công thức 2.4 Anisotropic Total Variation Loss

#### b. Vai trò trong quá trình tối ưu

Trong triển khai hiện tại, TV Loss được tính bằng bình phương sai khác giữa các pixel theo chiều ngang và dọc, sau đó tổng hợp lại để tạo thành giá trị mất mát. Mức độ ảnh hưởng của TV Loss được điều chỉnh bằng hệ số `smoothing_factor`.

$$L_{total} = L_{content} + L_{style} + \lambda_{TV} \times L_{TV}$$

Công thức 2.5 Total Loss sau tối ưu

Trong đó,  $\lambda_{TV} = \text{smoothing\_factor}$  điều chỉnh mức độ mịn mong muốn.

- Khi `smoothing_factor` nhỏ (ví dụ: 0.001), ảnh giữ được chi tiết phong cách nhưng đã loại bỏ được một phần nhiễu nhẹ.
- Khi `smoothing_factor` lớn hơn (ví dụ: 0.01 hoặc 0.1), ảnh mượt rõ rệt nhưng có thể làm mờ các nét cọ nghệ thuật quan trọng.

#### c. Lợi ích của cải tiến

- Chất lượng ảnh cải thiện rõ rệt: Giảm hiện tượng nhiễu lốm đốm hoặc đường nét gãy, đặc biệt trong các vùng đồng nhất của ảnh.
- Ổn định hơn trong quá trình tối ưu: Đường cong tổng mất mát ít dao động hơn khi sử dụng `smoothing_factor`, nhất là ở các giá trị hợp lý như 0.01 hoặc 0.05.
- Tính linh hoạt cao: Có thể điều chỉnh `smoothing_factor` để phù hợp với đặc trưng của từng phong cách (ví dụ: phong cách Van Gogh cần giữ cọ mạnh, còn Monet cần mượt hơn).

Mô tả	Phương pháp gốc (Gatys, 2015)	Phương pháp đề xuất
Hàm mất mát	Content Loss + Style Loss	Content + Style + TV Loss

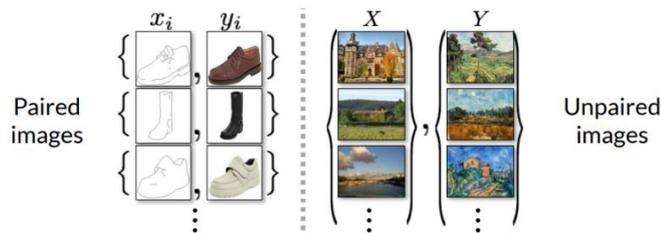
Ổn định tối ưu hóa	Dao động khi training dài	Mượt, ít dao động
Ảnh đầu ra	Có thể nhiều hoặc thô	Mịn hơn, chi tiết rõ hơn
Kiểm soát độ mượt	Không có	Có thông qua smoothing factor

Bảng 2.1 So sánh trước và sau cải tiến

Việc bổ sung Total Variation Loss là một cải tiến hợp lý, có cơ sở khoa học vững chắc và đã được áp dụng hiệu quả trong các công trình nghiên cứu hiện đại về NST. Trong đề tài này, cải tiến được triển khai bằng cách tích hợp `total_variation_loss()` vào hàm huấn luyện với tham số điều chỉnh `smoothing_factor`, giúp nâng cao đáng kể chất lượng ảnh đầu ra và đảm bảo tính ổn định của quá trình tối ưu hóa.

## 2.4. CycleGAN

CycleGAN (Cycle-Consistent Generative Adversarial Network) là một mô hình thuộc họ GAN, cho phép chuyển đổi ảnh giữa hai miền không đối xứng (unpaired domains) mà không cần cặp dữ liệu tương ứng. Trong bài toán chuyển đổi ảnh thành tranh nghệ thuật, CycleGAN có thể học đặc trưng phong cách từ tập tranh (ví dụ Monet, Van Gogh) và đặc trưng nội dung từ ảnh thực, tạo ra ảnh mới mang phong cách đích nhưng vẫn giữ nguyên bố cục gốc.

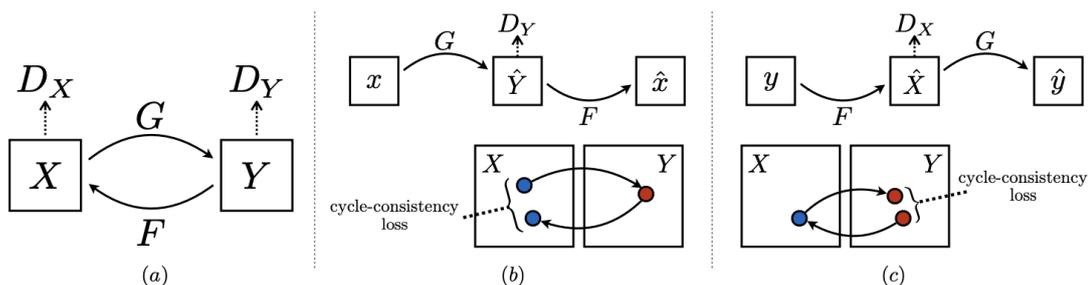


Hình 2.5 Cặp dữ liệu không đối xứng trong CycleGAN

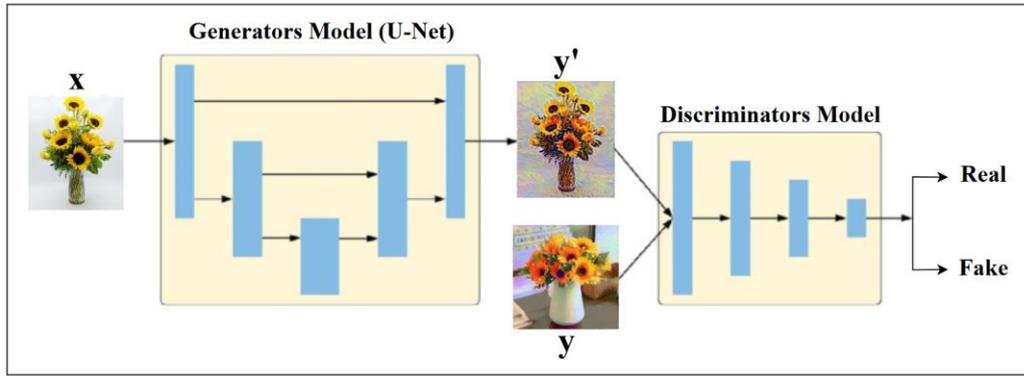
### a. Kiến trúc

– Kiến trúc CycleGAN gồm hai cặp mạng chính:

- Generator  $G: X \rightarrow Y$ , chuyển ảnh từ miền thực  $X$  sang miền phong cách  $Y$
- Generator  $F: Y \rightarrow X$ , chuyển ảnh từ miền phong cách  $Y$  về lại miền thực  $X$
- Discriminator  $D_Y$ : phân biệt giữa ảnh phong cách thật và ảnh sinh ra bởi  $G$
- Discriminator  $D_X$ : phân biệt giữa ảnh thật và ảnh sinh ra bởi  $F$



Hình 2.6 Tổng quan kiến trúc CycleGAN



Hình 2.7 Kiến trúc Generator và Discriminator

Toàn bộ pipeline tạo thành một chu trình khép kín nhằm giữ lại nội dung gốc trong khi thay đổi phong cách.

– **Quá trình Generator bao gồm hai hướng chuyển đổi:**

- $G: X \rightarrow Y$ : chuyển ảnh thực thành ảnh phong cách
- $F: Y \rightarrow X$ : chuyển ảnh phong cách về ảnh thực

Mỗi Generator sử dụng kiến trúc U-Net cải tiến, giúp giữ lại các đặc trưng không gian của ảnh gốc thông qua các kết nối skip-connection. Điều này rất quan trọng để đảm bảo nội dung ảnh không bị biến dạng khi áp dụng phong cách.

**Các Generator được huấn luyện với tổng hợp ba loại hàm mất mát:**

1. Adversarial Loss:  $\mathcal{L}_{GAN}(G, D_Y, X, Y)$  và  $\mathcal{L}_{GAN}(F, D_X, Y, X)$  đảm bảo ảnh sinh ra giống với ảnh trong miền đích.
2. Cycle-Consistency Loss:  $\mathcal{L}_{cyc}(G, F)$  đảm bảo tính nhất quán khi biến đổi hai chiều.
3. Identity Loss (tùy chọn):  $\mathcal{L}_{identity}(G, F)$  giúp hạn chế thay đổi khi ảnh đã thuộc miền đích.

**Tổng hàm mất mát của Generator:**

$$\mathcal{L}_{Gen} = \mathcal{L}_{GAN} + \lambda \cdot \mathcal{L}_{cyc} + \lambda_{id} \cdot \mathcal{L}_{identity}$$

Với  $\lambda$  và  $\lambda_{id}$  là các trọng số.

Công thức 2.6 Total Loss CycleGAN

– **Quá trình Discriminator**

Discriminator trong CycleGAN đóng vai trò đối trọng với Generator, có nhiệm vụ học cách phân biệt ảnh thật với ảnh được sinh ra.

- $D_Y$  phân biệt giữa ảnh  $y$  thật và ảnh  $G(x)$  sinh ra
- $D_X$  phân biệt giữa ảnh  $x$  thật và ảnh  $F(y)$  sinh ra

Mỗi Discriminator sử dụng kiến trúc PatchGAN, thay vì đánh giá toàn ảnh, nó phân tích ảnh thành các patch nhỏ (ví dụ 70x70) và đánh giá tính chân thực của từng vùng. Cách tiếp cận này giúp mô hình tập trung vào chi tiết cục bộ, phù hợp với phong cách nghệ thuật.

Hàm mất mát được sử dụng cho mỗi Discriminator là  $\mathcal{L}_{Disc}$

#### – Cách hai quá trình phối hợp

Trong quá trình huấn luyện, hai quá trình Generator và Discriminator được tối ưu luân phiên:

- Khi tối ưu Generator, các Discriminator được giữ nguyên (đóng băng), giúp Generator học cách đánh lừa chúng.
- Khi tối ưu Discriminator, các Generator được đóng băng, để Discriminator học cách phân biệt ảnh thật và ảnh giả chính xác hơn.

Cơ chế phối hợp này tạo ra thế cân bằng giữa hai mạng, giúp mô hình học cách chuyển đổi phong cách mượt mà mà không làm mất nội dung gốc.

#### b. Tiền xử lý và dữ liệu huấn luyện

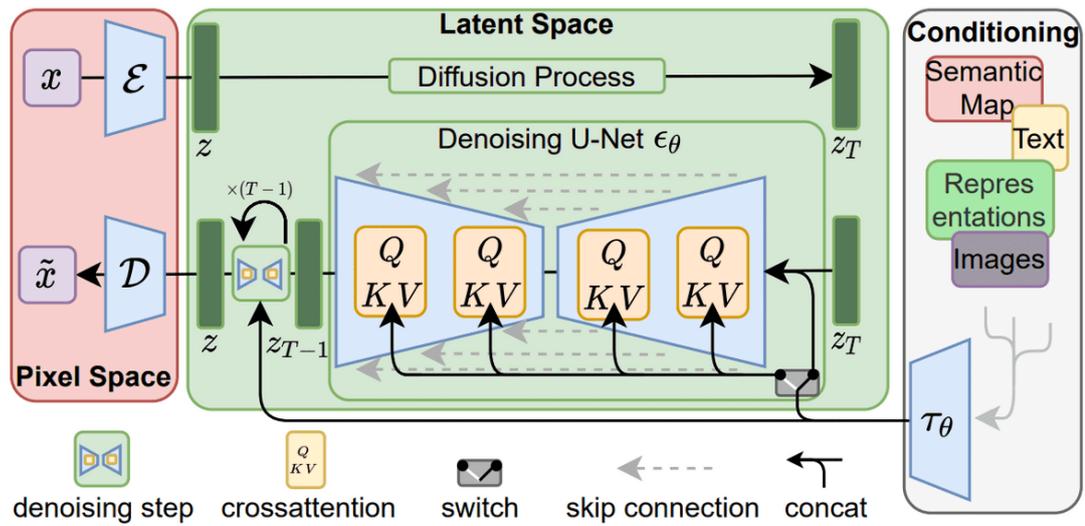
- Tập dữ liệu gồm hai miền ảnh không đối xứng: ảnh thật (photo) và ảnh phong cách (ví dụ: tranh Monet). Số lượng ảnh giữa hai miền có thể không cân bằng nhau.
- Tiền xử lý bao gồm:
  - Resize ảnh về kích thước 256x256 hoặc 1024x1024.
  - Chuẩn hóa giá trị pixel về khoảng  $[-1, 1]$ .
  - Tăng cường dữ liệu bằng random crop, horizontal flip, hoặc kỹ thuật như DiffAugment.

## 2.5. Diffusion LoRA

### 2.5.1. Kiến trúc Stable Diffusion

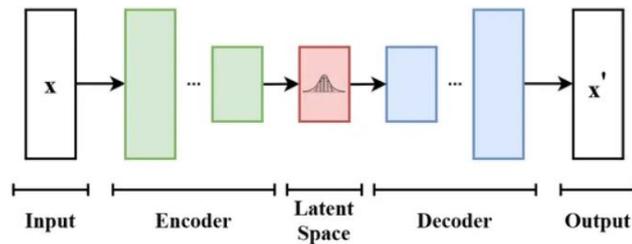
Stable Diffusion là một mô hình Text-to-Image Generation thuộc họ Diffusion Model, được giới thiệu năm 2022 bởi Stability AI và LAION. Khác với các phương pháp như Neural Style Transfer hay CycleGAN hoạt động trong không gian ảnh (pixel space), Stable Diffusion thực hiện quá trình khuếch tán và sinh ảnh trong không gian tiềm ẩn (latent space) nhằm tăng hiệu quả tính toán và chất lượng ảnh.

### Pipeline kiến trúc Stable Diffusion



Hình 2.8 Pipeline Stable Diffusion Model

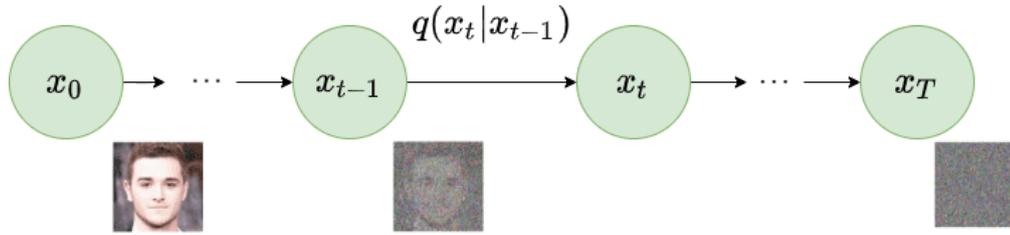
- **Perceptual Compression (VAE - Autoencoder):** Đây là bước nén ảnh ban đầu và giải nén ảnh đầu ra.



Hình 2.9 Variational Autoencoders (VAEs)

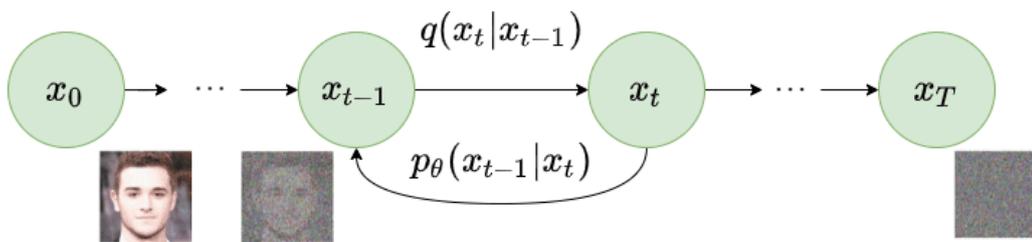
- Ảnh đầu vào  $x$  được đưa vào Encoder  $\varepsilon$  (image2vec) để chuyển thành vector tiềm ẩn  $z$ . Quá trình mã hóa giúp chuyển ảnh về không gian tiềm ẩn có chiều thấp hơn, từ đó giảm chi phí tính toán, tăng tốc độ xử lý và tiết kiệm tài nguyên.
 
$$z = \varepsilon(x)$$
- Sau khi hoàn tất sinh ảnh, vector  $\tilde{z}$  sẽ được Decoder  $D$  (vec2image) chuyển lại thành ảnh  $\tilde{x}$ 

$$\tilde{x} = D(\tilde{z})$$
- **Forward Diffusion (Thêm nhiễu):** Mục tiêu của mô hình là học cách loại bỏ nhiễu ngược lại, phục hồi ảnh gốc.
  - Vector tiềm ẩn  $z(0)$  sau khi mã hóa sẽ được thêm nhiễu dần qua  $T$  bước, kết quả là vector  $z(T)$ .
  - Mỗi bước thêm nhiễu tuân theo một phân phối xác suất chuẩn (Gaussian).
  - Quá trình này tạo ra noisy latent vector dùng làm đầu vào cho mạng sinh ảnh.



Hình 2.10 Forward Diffusion (Quá trình khuếch tán tiến)

- **Reverse Diffusion - Denoising U-Net:** Đây là mạng khử nhiễu chính, giúp khôi phục lại ảnh từ nhiễu
  - Từ vector  $z(T)$ , mô hình sẽ học khử nhiễu ngược dần về  $\tilde{z}(0)$ , gọi là quá trình khuếch tán ngược.
  - Quá trình này được thực hiện bằng một mạng U-Net  $\varepsilon_\theta$  được thiết kế với các skip connections và cross-attention.
  - Cross-attention giúp mô hình điều kiện hóa ảnh dựa trên văn bản đầu vào, đảm bảo rằng nội dung sinh ra khớp với mô tả trong prompt.



Hình 2.11 Reverse Diffusion (Quá trình khuếch tán ngược)

- **Conditioning – Text, Image Transformer:** Đây là phần điều kiện hóa giúp sinh ảnh khớp với mô tả văn bản đầu vào
  - Văn bản (prompt) được đưa vào một bộ mã hóa văn bản (text2vec), ví dụ như CLIP Text Encoder.
  - Từ đó sinh ra vector điều kiện  $\tau_0$ , Embedding được đưa vào U-Net qua các block cross-attention cùng với  $z(T)$  giúp mô hình sinh ảnh khớp với văn bản.
  - Ngoài ra, có thể điều kiện hóa bằng semantic map, representations hoặc ảnh.

### 2.5.2. Diffusion LoRA Style Transfer

Các phương pháp chuyển phong cách như CycleGAN dù hiệu quả nhưng tốn thời gian huấn luyện và bộ nhớ do phải tối ưu nhiều mạng từ đầu. Để khắc phục nhược điểm này, Stable Diffusion LoRA tận dụng sức mạnh của mô hình khuếch tán đã huấn luyện trước, giúp tăng tốc hội tụ và giảm chi phí tính toán.

Stable Diffusion hoạt động trong không gian tiềm ẩn thay vì không gian pixel, giúp nén thông tin hiệu quả hơn. Phương pháp này tinh chỉnh U-Net bằng kỹ thuật LoRA và dùng

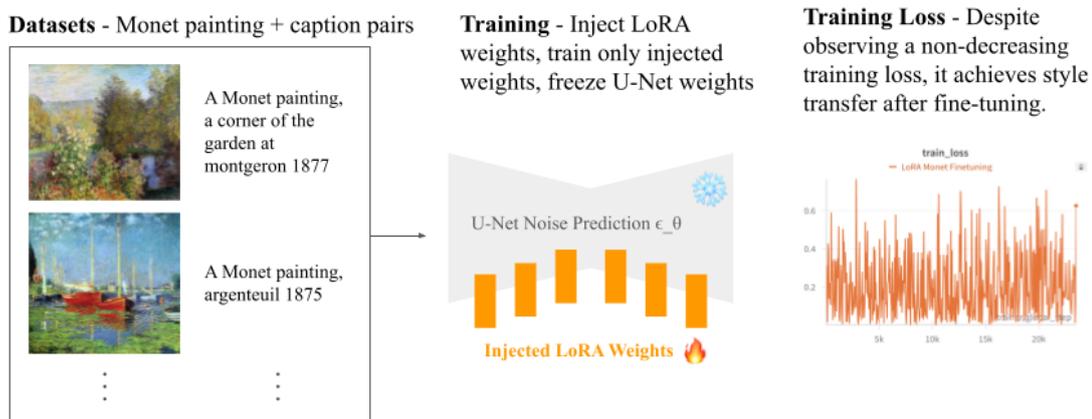
IP-Adapter để thêm điều kiện từ ảnh gốc, kết hợp với văn bản chỉ dẫn. Mô hình được huấn luyện trên các cặp ảnh – mô tả (caption) mang phong cách nghệ sĩ như Monet, giúp chuyển phong cách chính xác và linh hoạt.

**a. Thành phần chính của giải pháp**

- Stable Diffusion LoRA: Kỹ thuật LoRA tiêm các ma trận trọng số cấp thấp vào U-Net, chỉ huấn luyện một phần nhỏ tham số thay vì toàn bộ mạng, giữ nguyên kiến thức đã học từ mô hình gốc. Quá trình này được tối ưu hóa bằng cách sử dụng cặp (photo, caption) của các bức tranh Monet, trong đó caption bao gồm tên tranh tương ứng.
- IP-Adapter: Thành phần mã hóa thông tin từ ảnh gốc thành vector điều kiện, giúp giữ nguyên các chi tiết hình học trong khi áp dụng phong cách mới, đóng vai trò như một điều kiện bổ sung cho bộ khử nhiễu.
- Pipeline huấn luyện và suy luận bao gồm hai giai đoạn chính:
  - Giai đoạn huấn luyện tinh chỉnh mô hình bằng phương pháp LoRA hiệu quả, với tiền tố "A Monet painting, " thêm vào caption để định danh phong cách Monet, lấy cảm hứng từ DreamBooth.
  - Giai đoạn suy luận sử dụng mô hình đã học để khử nhiễu vector tiềm ẩn từ bước thời gian trung gian N, dựa trên hiểu biết từ SDEdit rằng quá trình giải phương trình vi phân ngẫu nhiên (SDE) có thể bắt đầu từ bất kỳ bước nào để chỉnh sửa ảnh gốc.

**b. Pipeline huấn luyện (Training Phase)**

**Methodology - LoRA Training**



Hình 2.12 Diffusion LoRA Style Transfer Pipeline Training

**– Dữ liệu và mục tiêu**

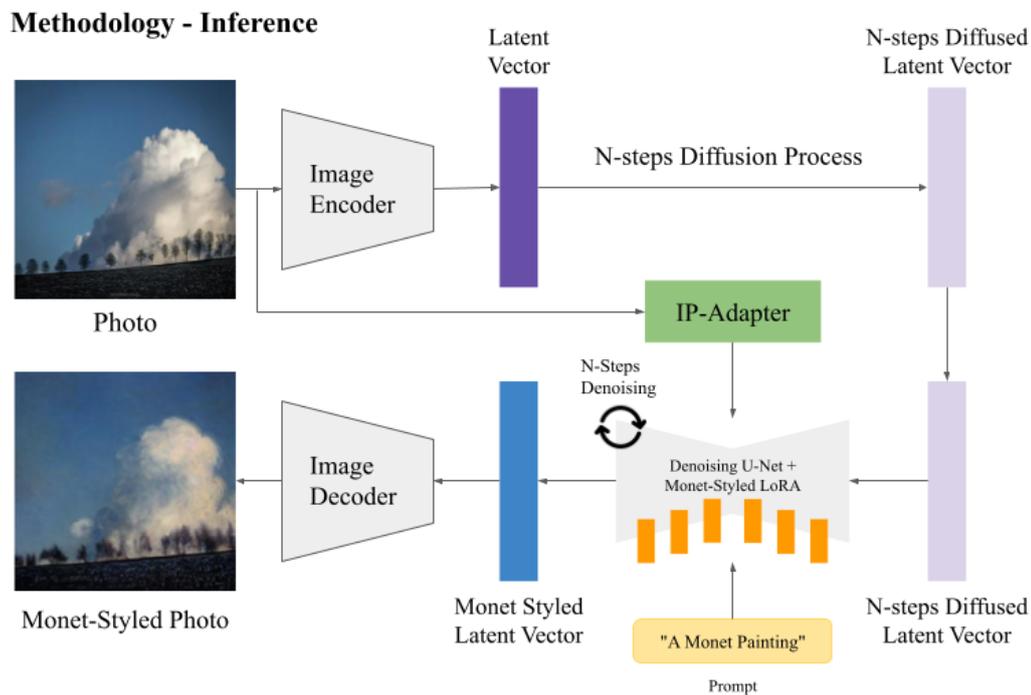
Giải pháp sử dụng tập dữ liệu gồm cặp ảnh (photo) và caption mô tả phong cách (ví dụ: "A Monet painting, a corner of the garden at Montgeron 1877"). Mục tiêu là huấn

luyện mô hình để chuyển đổi phong cách ảnh chụp thành phong cách tranh Monet thông qua tinh chỉnh LoRA, giảm nhu cầu huấn luyện lại toàn bộ mô hình.

– **Quy trình huấn luyện**

- Chuẩn bị dữ liệu: Ảnh chụp được mã hóa thành vector tiềm ẩn  $z$  bằng encoder của VAE, trong khi caption được mã hóa thành vector ngữ nghĩa bằng CLIP Text Encoder.
- Thêm nhiễu: Thực hiện forward diffusion để tạo  $z_T$  từ  $z_0$  theo công thức khuếch tán.
- Tinh chỉnh LoRA: Các trọng số U-Net được đóng băng (freeze), chỉ các ma trận LoRA được huấn luyện để điều chỉnh dự đoán nhiễu  $\epsilon_\theta$ , đảm bảo vector tiềm ẩn cuối cùng  $z_0$  mang phong cách Monet.
- Đánh giá: Kết quả được đánh giá dựa trên chất lượng phong cách chuyển đổi, bất chấp việc hàm mất mát huấn luyện có thể không giảm (non-decreasing training loss), do mục tiêu chính là style transfer thay vì tái tạo chính xác.

c. **Pipeline suy luận (Inference Phase)**



Hình 2.13 Diffusion LoRA Style Transfer Pipeline Inference

– **Dữ liệu và mục tiêu**

Đầu vào là một ảnh chụp và caption cố định "A Monet painting", với mục tiêu tạo ra ảnh đầu ra mang phong cách tranh Monet.

– **Quy trình suy diễn**

- Mã hóa ảnh: Ảnh chụp được nén thành vector tiềm ẩn  $z$  bằng encoder VAE.
- Mã hóa văn bản: Caption "A Monet painting" được mã hóa thành vector ngữ nghĩa bằng CLIP Text Encoder.
- Quá trình khuếch tán ngược: Sử dụng U-Net với trọng số LoRA đã huấn luyện để thực hiện  $N$  bước khử nhiễu (thường 20-50 bước, tùy scheduler) từ  $z_T$  về  $z_0$ . IP-Adapter (nếu áp dụng) cung cấp vector điều kiện bổ sung từ ảnh gốc, tích hợp qua cross-attention để giữ nội dung hình học.
- Giải mã: Vector tiềm ẩn  $z_0$  được giải mã bằng decoder VAE để tạo ảnh đầu ra (Monet-styled photo).

#### d. Xây dựng tập dữ liệu huấn luyện



Hình 2.14 Dữ liệu huấn luyện Diffusion LoRA Style Transfer

- Tập dữ liệu dùng để huấn luyện mô hình được xây dựng từ hai thành phần chính: tập ảnh (photo) và tập chú thích (caption). Cụ thể, tập ảnh được thu thập từ nguồn WikiArt, cung cấp các bức ảnh nghệ thuật chất lượng cao. Tập chú thích được tiền xử lý sao cho tên file của mỗi chú thích trùng với tên của ảnh tương ứng trong tập photo, đồng thời nội dung chú thích là mô tả của bức ảnh, cũng chính là tên của ảnh.
- Tập dữ liệu huấn luyện xuất phát từ 4 họa sĩ và phong cách bao gồm: Monet, Vangogh, Cezanne, Ukiyo-e.

## 2.6. Thử nghiệm

### 2.6.1. Thử nghiệm Neural Style Transfer

#### a. Bộ tham số thử nghiệm

Tham số	Giải thích
<b>resize_size</b>	Kích thước ảnh trước xử lý. Resize lớn giúp giữ nhiều chi tiết giúp ảnh đầu ra rõ và đẹp hơn.
<b>style_weight (<math>\beta</math>)</b>	Trọng số của hàm mất mát phong cách (style loss), quyết định mức độ ảnh hưởng của phong cách trong ảnh kết quả.

	$L_{total} = \alpha L_{content} + \beta L_{style}$
<b>content_weight (<math>\alpha</math>)</b>	Trọng số của hàm mất mát nội dung (content loss), quyết định mức độ bảo tồn nội dung của ảnh gốc.
<b>epochs</b>	Số bước cập nhật ảnh. Nhiều epochs giúp ảnh tinh chỉnh kỹ hơn nhưng tốn thời gian.
<b>learning_rate</b>	Tốc độ thay đổi ảnh sau mỗi bước học. Cao giúp tối ưu nhanh nhưng dễ mất ổn định, ngược lại thấp dẫn đến tối ưu chậm chậm nhưng mượt.
<b>smoothing_factor</b>	Cải tiến hệ số điều chỉnh TV loss để làm mịn ảnh. Giá trị nhỏ giúp ảnh ít mịn, giữ chi tiết, lớn giúp mịn hơn nhưng có thể mất nét.

Bảng 2.2 Bộ tham số của Neural Style Transfer

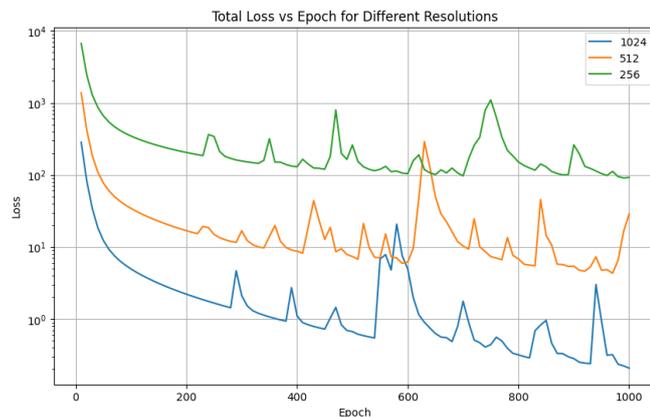
**b. Kết quả thử nghiệm**



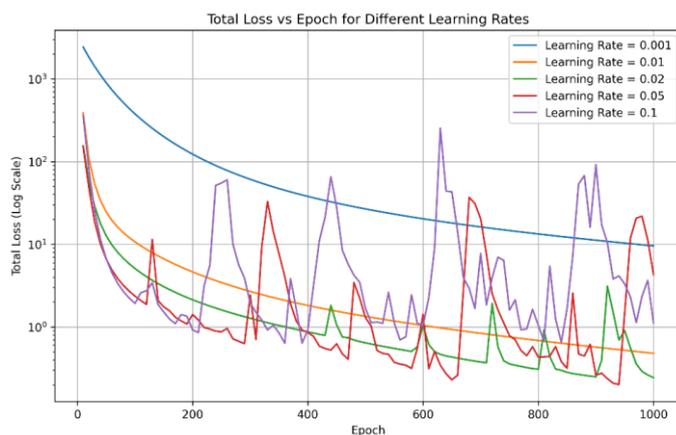
Hình 2.15 Ảnh đầu ra của 200 epoch và 2.000 epoch (resize 1024)



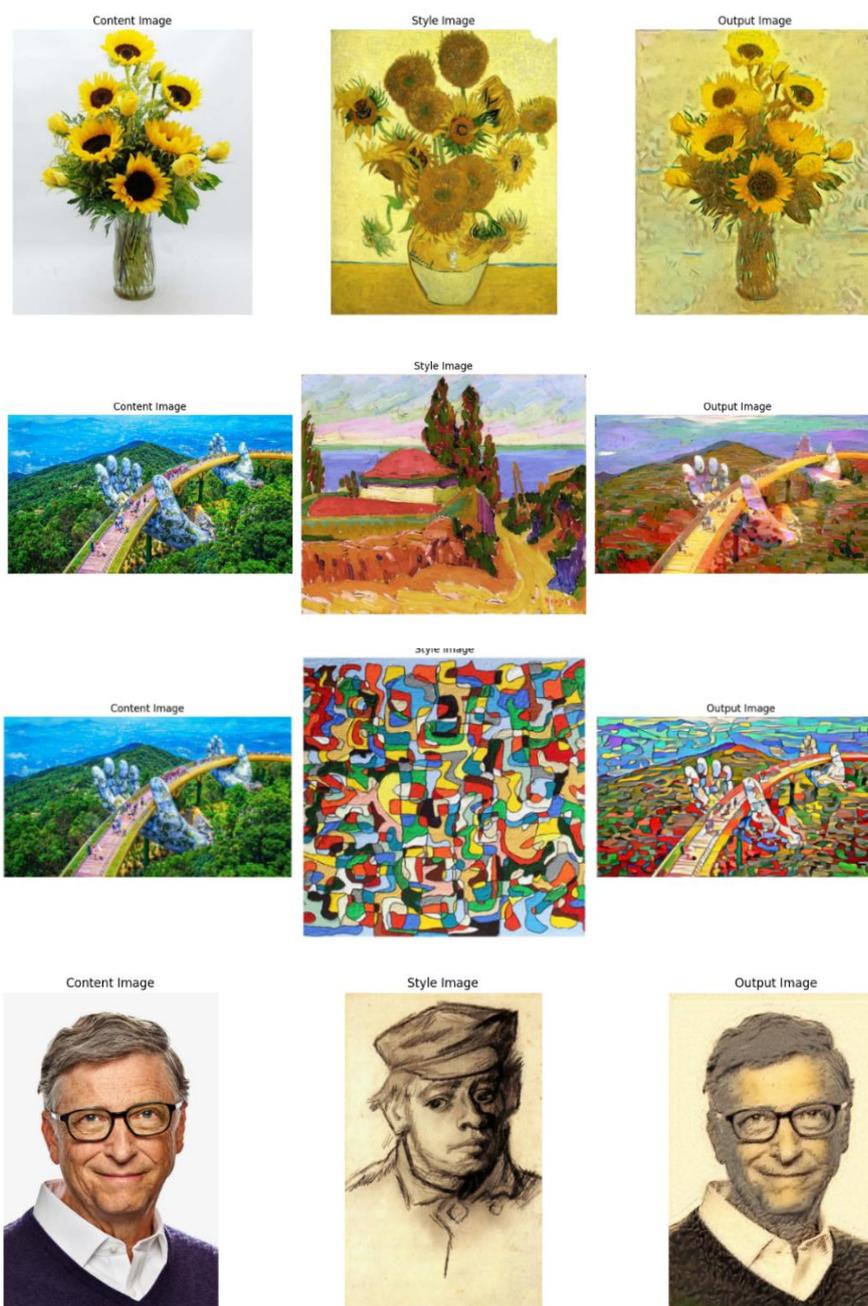
Hình 2.16 Ảnh đầu ra của 200 epoch và 1.000 epoch (resize 1024)



Hình 2.17 Total Loss với các kích thước Resize



Hình 2.18 Total Loss với các giá trị Learning rate (resize 1024)





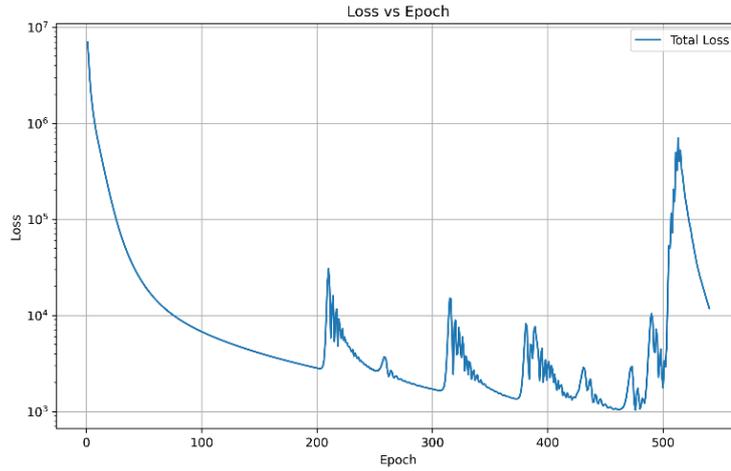
Hình 2.19 So sánh kết quả đầu ra với các style image

### Smoothing\_factor (cải tiến)

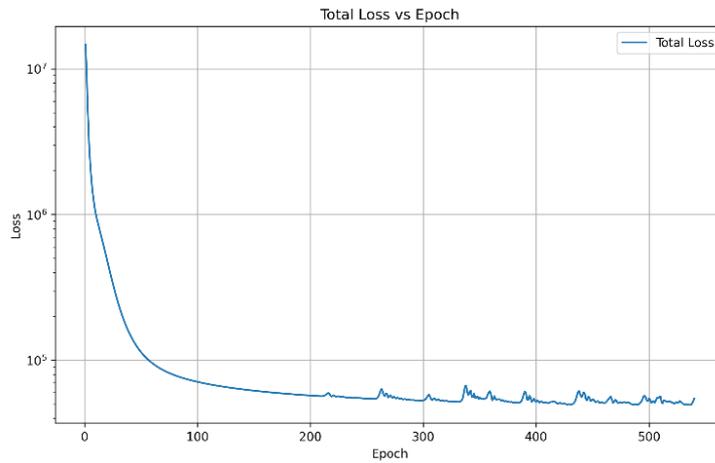
- Việc bổ sung smoothing factor thông qua thành phần Total Variation Loss (TV Loss) làm cho total loss tăng so với trường hợp không sử dụng, khi xét tại cùng một số epoch. Điều này là hợp lý vì TV Loss đóng vai trò như một thuật phạt, làm tăng giá trị tổng mất mát. Cơ chế này tương tự như việc giảm learning rate: học chậm hơn, ổn định hơn và ít dao động, nhưng đôi lại quá trình tối ưu hóa sẽ diễn ra chậm hơn.
- Mặc dù total loss lớn hơn, nhưng smoothing factor mang lại nhiều lợi ích đáng kể:
  - Ổn định quá trình học, giúp tránh hiện tượng nhiễu và dao động mạnh.
  - Cải thiện chất lượng ảnh đầu ra, đặc biệt là giữ được các đường nét rõ ràng và gần với ảnh nội dung ban đầu.
  - Khi tăng giá trị smoothing factor, tác động của TV Loss càng lớn, và ảnh đầu ra càng được “phạt” để giữ lại cấu trúc nội dung gốc.
- Quan sát hình ảnh thực nghiệm cho thấy: với cùng bộ tham số, ảnh có sử dụng smoothing factor thể hiện đường nét mượt và sắc nét hơn so với ảnh không sử dụng. Ngoài ra, đường cong Total Loss cũng ổn định hơn, thể hiện hiệu quả của việc cải tiến này.



Hình 2.20 So sánh trước và sau khi áp dụng cải tiến



Hình 2.21 Biểu đồ Total Loss chưa cải tiến



Hình 2.22 Biểu đồ Total Loss sau cải tiến

*c. Nhận xét*

Parameter	Impact Run time	Impact Quality
<b>resize_size</b>	✓	✓
<b>epochs</b>	✓	✓
<b>learning_rate</b>	—	✓
<b>smoothing_factor</b>	—	✓
<b>style_weight</b>	—	low
<b>content_weight</b>	—	low
<b>smoothing_factor</b>	—	✓

Bảng 2.3 Tác động của các tham số đến thời gian sinh ảnh và chất lượng ảnh

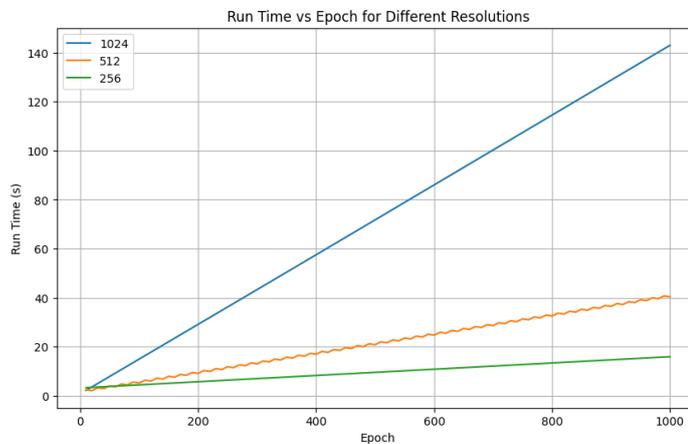
Resize	Epoch				
<b>1024</b>	<b>280</b>	<b>380</b>	<b>540</b>	<b>930</b>	<b>1000</b>
<b>512</b>	<b>410</b>	<b>590</b>	<b>830</b>	<b>920</b>	<b>970</b>
<b>256</b>	<b>600</b>	<b>650</b>	<b>700</b>	<b>880</b>	<b>990</b>

Bảng 2.4 Lựa chọn số Epoch tốt nhất với mỗi kích thước Resize

**Thời gian sinh ảnh**

STT	resize	Epoch	TIME	Note	<i>style<sub>weight</sub></i>	<i>content<sub>weight</sub></i>	<i>learning rate</i>
1	256	10	3.30s	<i>+0.13s for 10 epoch</i>	no impact	no impact	no impact
2	256	20	3.43s				
3	256	30	3.56s				
4	256	40	3.68s				
5	256	50	3.81s				
6	256	100	4.45s				
7	256	200	5.72s				
8	256	300	7.00s				
9	256	400	8.28s				
10	256	500	9.55s				
11	256	600	10.83s				
12	256	1000	15.93s				
12	512	10	2.48s	<i>+0.39s for 10 epoch</i>	no impact	no impact	
13	512	20	2.09s				
14	512	30	3.26s				
15	512	40	2.87s				
16	512	50	4.04s				
17	512	100	5.21s				
18	512	200	9.12s				
19	512	300	13.02s				
20	512	400	16.92s				
21	512	500	20.83s				
22	512	600	24.73s				
23	512	1000	40.35s				
24	1024	10	2.20s	<i>+1.42s for 10 epoch</i>	no impact	no impact	
25	1024	20	3.62s				
26	1024	30	5.03s				
27	1024	40	6.45s				
28	1024	50	7.87s				
29	1024	100	14.96s				
30	1024	200	29.13s				
31	1024	300	43.31s				
32	1024	400	57.51s				
33	1024	500	71.74s				
34	1024	600	85.99s				
35	1024	1000	142.96s				

Bảng 2.5 Thời gian sinh ảnh bằng phương pháp Neural Style Transfer



Hình 2.23 Biểu đồ thời gian sinh ảnh phương pháp Neural Style Transfer

## Chất lượng ảnh

- Kích thước ảnh đầu vào (resize) được chọn là 256, 512 và 1024. Resize càng lớn giữ được càng nhiều thông tin chi tiết, từ đó nâng cao chất lượng ảnh sinh, nên resize 1024 được ưu tiên trong các kịch bản có yêu cầu cao về độ sắc nét.
- Số lượng epoch có ảnh hưởng trực tiếp đến chất lượng và thời gian sinh ảnh. Các kết quả cho thấy từ 50 đến 200 epoch đã có thể tạo ra ảnh có phong cách rõ ràng, tuy nhiên để tăng độ mượt và giảm nhiễu, số epoch có thể tăng lên đến 1000, đánh đổi bằng thời gian huấn luyện lâu hơn.
- Tham số `style_weight` và `content_weight` không tác động nhiều đến kết quả đầu ra trong phạm vi thử nghiệm, chỉ có tác động nhỏ khi hai tham số quá lệch nhau, cho thấy độ nhạy của mô hình với hai tham số này là thấp.
- Learning rate có ảnh hưởng gián tiếp đến thời gian hội tụ và chất lượng ảnh thông qua số lượng epoch. Giá trị learning rate 0.05 được chọn là tối ưu, cân bằng giữa tốc độ hội tụ và độ ổn định trong huấn luyện.
- Hiệu quả tổng quát cao trên nhiều loại ảnh: Mô hình hoạt động tốt không chỉ với các phong cách nổi tiếng như Monet, Van Gogh, Ukiyo-e mà còn phù hợp với ảnh trừu tượng, ảnh phác họa, ảnh chì, ảnh màu vẽ tay và tranh sơn dầu. Điều này chứng minh tính linh hoạt và khả năng tổng quát hóa tốt của mô hình.
- Chất lượng ảnh đầu ra phụ thuộc nhiều vào style image. Các ảnh có nét vẽ đơn giản, rõ ràng, ít chi tiết như tranh màu, tranh sơn dầu cho kết quả rất giống tranh vẽ thật. Đặc biệt, style image có nét vẽ lớn và đa hướng sẽ giúp ảnh đầu ra mang đặc trưng phong cách mạnh mẽ mà vẫn giữ được nội dung gốc.
- Chọn style image có nội dung tương đồng với ảnh gốc (content image) cũng giúp tăng tính hài hòa và tự nhiên trong ảnh kết quả.
- Cải tiến với Total Variation Loss (`smoothing_factor`) cho thấy hiệu quả rõ rệt, đặc biệt với ảnh trừu tượng hoặc ảnh nhiều đường nét. Cải tiến này giúp giữ lại nét vẽ mà vẫn giảm nhiễu, cho ảnh sinh rõ phong cách và dễ chịu về mặt thị giác ngay cả khi thời gian huấn luyện không tăng.
- Bộ tham số tốt nhất được chọn

<code>resize_size</code>	<code>style_weight</code>	<code>content_weight</code>	<code>epochs</code>	<code>learning_rate</code>	<code>smoothing_factor</code>
1024	1	1000	540	0.05	0.01 or 10 or 100

Bảng 2.6 Bộ tham số tốt nhất được chọn



Hình 2.24 Ma trận ảnh được tạo bởi bộ tham số tốt nhất

### 2.6.2. Thử nghiệm CycleGAN

#### a. Kết quả thử nghiệm

- **Quá trình huấn luyện:** Dataset (~1000 tranh, 500 ảnh phong cảnh), thời gian train (6h – 12h).
- **Bộ tham số ảnh hưởng đến thời gian train**

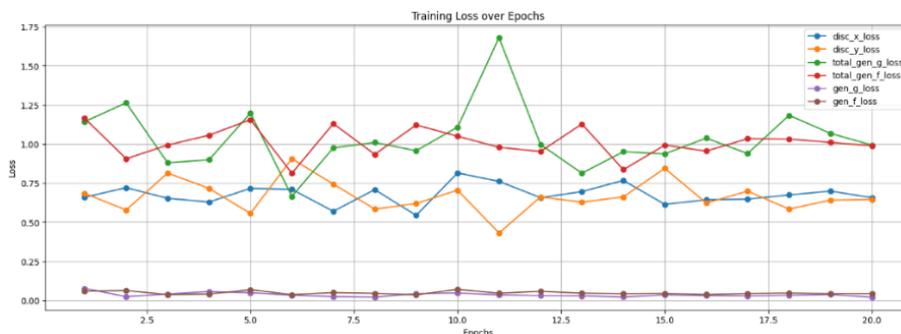
Tham số	Giải thích
Epoch	Số lần lặp qua toàn bộ tập dữ liệu để tối ưu.
Step per epoch	Số lần tối ưu trong mỗi epoch.
Resize	Kích thước resize ảnh đầu vào
LAMBDA_ID	Giá trị thấp làm giảm tác dụng của identity loss.

Bảng 2.7 Bộ tham số ảnh hưởng đến thời gian train CycleGAN

- **Thời gian sinh ảnh**

Size	Times
256, 512, 1024	3s

Bảng 2.8 Bộ tham số ảnh hưởng đến thời gian suy luận



Hình 2.25 Biểu đồ loss khi train CycleGAN

#### b. Nhận xét

- Thời gian sinh ảnh cực nhanh do mô hình đã được huấn luyện sẵn.
- Về cơ bản mô hình đã học được phong cách của tập dữ liệu huấn luyện, tuy nhiên ảnh được sinh ra có chất lượng thấp.

### 2.6.3. Thử nghiệm Diffusion LoRA Style Transfer

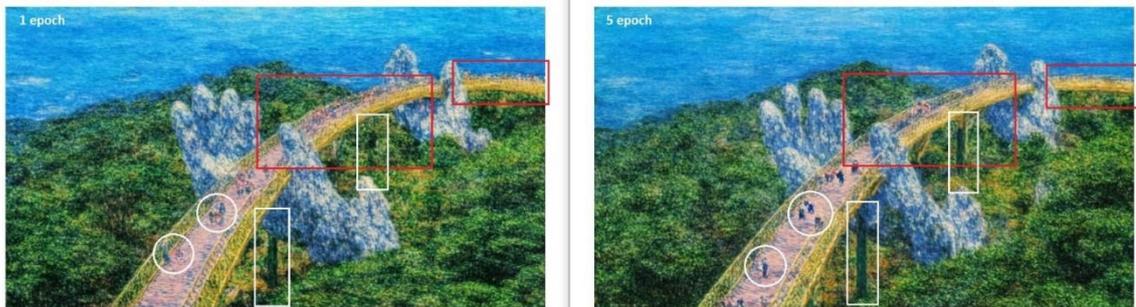
#### a. Bộ tham số thử nghiệm

Tên	Mô tả
<b>image_cond_scale</b>	Độ mạnh của IP-Adapter, điều chỉnh giữ nội dung ảnh gốc.
<b>strength</b>	Tỷ lệ nhiễu thêm vào ảnh gốc, quyết định mức độ thay đổi phong cách.
<b>prompt</b>	Văn bản chỉ dẫn phong cách, định hướng qua CLIP Text Encoder.
<b>infer_steps</b>	Số bước khuếch tán ngược, ảnh hưởng độ chi tiết và thời gian xử lý.
<b>resize</b>	Kích thước ảnh đầu ra sau resize, chỉ là bước hậu xử lý.

Bảng 2.9 Bộ tham số suy luận

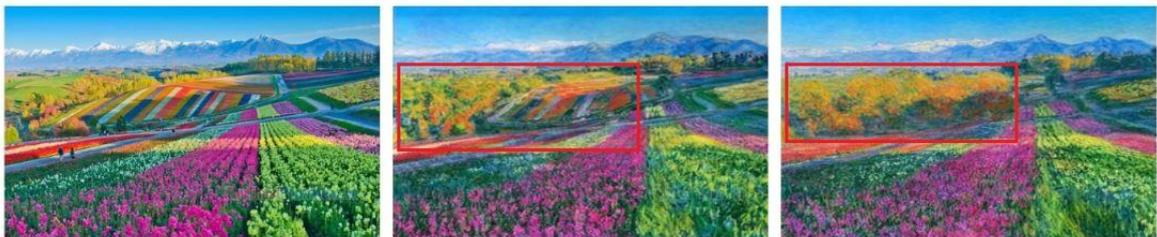
#### b. Kết quả thử nghiệm

- Mô hình sau 5 epoch cho chất lượng ảnh tốt hơn rõ rệt so với 1 epoch. Khi sử dụng ảnh đầu vào chất lượng cao, kết quả từ mô hình 5 epoch cho thấy chi tiết sắc nét hơn, không còn tình trạng mờ nhòe. Ví dụ, trong ảnh có người đi trên cầu, mô hình 1 epoch làm các nhân vật bị hòa lẫn vào nền, trong khi mô hình 5 epoch thể hiện rõ ràng dáng người, chi tiết cây cầu và cả ngón tay.



Hình 2.26 So sánh ảnh đầu ra giữa 1 epoch và 5 epoch

- Với ảnh resize 256 hoặc 512, do mô hình học phong cách quá kỹ, cùng một giá trị strength sẽ cho ảnh biến đổi mạnh hơn so với bộ tham số mặc định.



Hình 2.27 So sánh ảnh đầu ra giữa resize 256 và resize 512

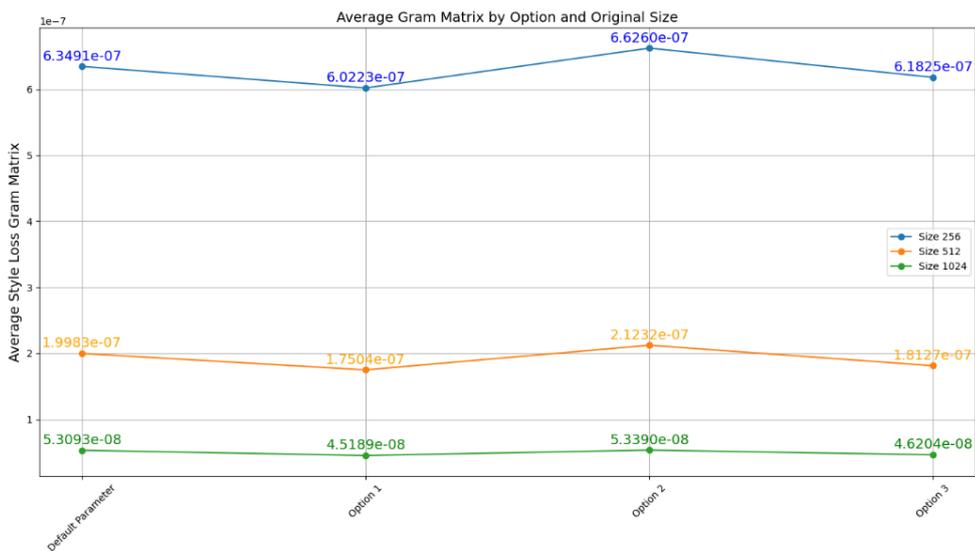
- Khi dùng ảnh đầu vào chất lượng cao, nếu giữ nguyên số bước suy diễn (inference steps), việc thay đổi số epoch không tạo ra khác biệt lớn về chất lượng ảnh đầu ra.
- Bên cạnh đánh giá định tính thông qua quan sát trực tiếp ảnh đầu ra, các phương pháp định lượng cũng được áp dụng nhằm lựa chọn bộ tham số tối ưu cho quá trình huấn luyện và suy luận mô hình LoRA.

Cụ thể, một số tổ hợp tham số đã được thiết lập, và để đánh giá khách quan hiệu quả giữa các tổ hợp này, hai chỉ số định lượng được sử dụng:

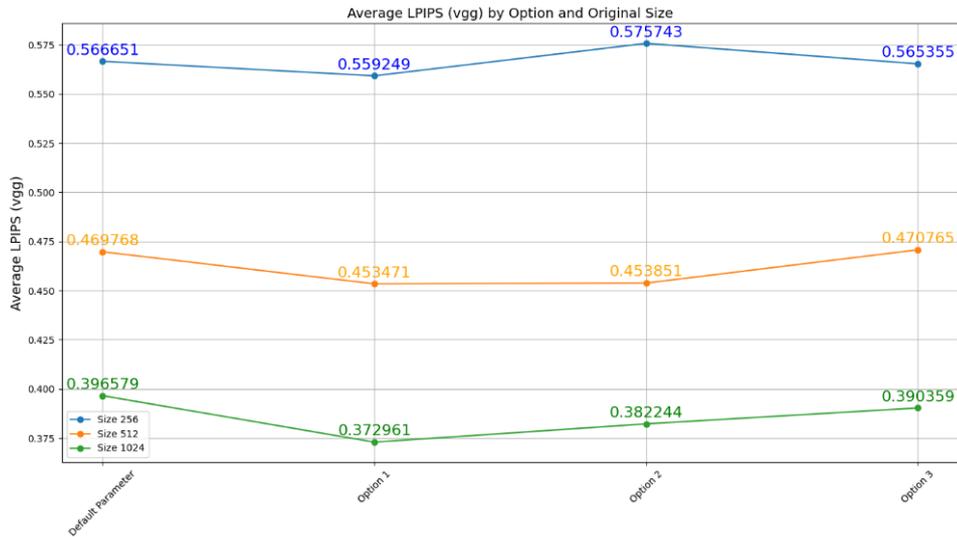
- Gram Matrix (style loss): đo mức độ mất phong cách, phản ánh khả năng duy trì phong cách từ ảnh tham chiếu.
- LPIPS (Learned Perceptual Image Patch Similarity): đánh giá độ tương đồng cảm quan giữa ảnh gốc và ảnh sinh.

Option	Original Size	Gram Marix Average	LPIPS Average
Bộ tham số mặc định	256	6.349056826593369e-07	0.5666508480906487
	512	1.9983462305361856e-07	0.4697678884863853
	1024	5.309317603119368e-08	0.39657871812582013
Bộ tham số 1	256	6.022328483368255e-07	0.5592490202188491
	512	1.7504219105290985e-07	0.45347144335508344
	1024	4.518870010628007e-08	0.37296128675341605
Bộ tham số 2	256	6.626017862032541e-07	0.5757432663440705
	512	2.1231955740219636e-07	0.45385064780712125
	1024	5.338989824021922e-08	0.38224403172731397
Bộ tham số 3	256	6.182490923833939e-07	0.5653547415137291
	512	1.8126740455315372e-07	0.47076498150825496
	1024	4.620442433229357e-08	0.39035936191678045

Bảng 2.10 Loss Gram Matrix và LPIPS



Hình 2.28 Biểu đồ Style Loss Gram Matrix



Hình 2.29 Biểu đồ Loss LPIPS

**c. Nhận xét**

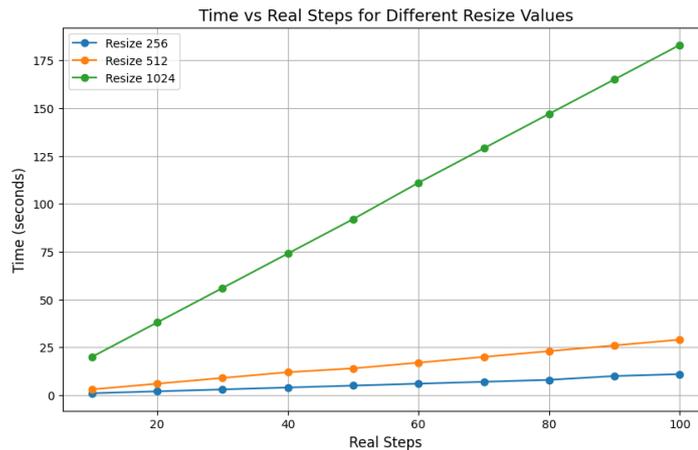
**Thời gian sinh ảnh**

- Strength càng cao thì lượng nhiễu đưa vào giai đoạn diffusion càng lớn, làm tăng cường độ phong cách ảnh, từ đó kéo dài thời gian sinh ảnh.
- Step (inference step) càng lớn thì ảnh càng chi tiết hơn về lý thuyết, nhưng sự gia tăng thời gian không hoàn toàn tương ứng do ảnh đầu vào vốn là ảnh thật, và thông tin gốc đã được giữ lại nhờ IP-Adapter xuyên suốt quá trình attention của U-Net.
- Resize càng lớn thì ảnh đưa vào U-Net càng chất lượng, giữ được nhiều chi tiết hơn, nhưng cũng làm tăng đáng kể thời gian xử lý.
- Image cond scale hầu như không ảnh hưởng đến thời gian sinh ảnh.

STT	resize	strength	$step_{inference}$	$step_{real}$ ( $strength \times step_{inference}$ )	Image cond scale	TIME	Note
1	256	0.1	100	10	no impact	01s	+1s
2	256	0.2	100	20	no impact	02s	
3	256	0.3	100	30	no impact	03s	
4	256	0.4	100	40	no impact	04s	
5	256	0.5	100	50	no impact	05s	
6	256	0.6	100	60	no impact	06s	
7	256	0.7	100	70	no impact	07s	
8	256	0.8	100	80	no impact	08s	
9	256	0.9	100	90	no impact	10s	
10	256	1.0	100	100	no impact	11s	
11	512	0.1	100	10	no impact	03s	+3s
12	512	0.2	100	20	no impact	06s	
13	512	0.3	100	30	no impact	09s	
14	512	0.4	100	40	no impact	12s	
15	512	0.5	100	50	no impact	14s	
16	512	0.6	100	60	no impact	17s	
17	512	0.7	100	70	no impact	20s	
18	512	0.8	100	80	no impact	23s	
19	512	0.9	100	90	no impact	26s	
20	512	1.0	100	100	no impact	29s	

21	1024	0.1	100	10	no impact	20s	+18s
22	1024	0.2	100	20	no impact	38s	
23	1024	0.3	100	30	no impact	56s	
24	1024	0.4	100	40	no impact	74s	
25	1024	0.5	100	50	no impact	92s	
26	1024	0.6	100	60	no impact	111s	
27	1024	0.7	100	70	no impact	129s	
28	1024	0.8	100	80	no impact	147s	
29	1024	0.9	100	90	no impact	165s	
30	1024	1.0	100	100	no impact	183s	

Bảng 2.11 Thời gian sinh ảnh Diffusion LoRA Style Transfer

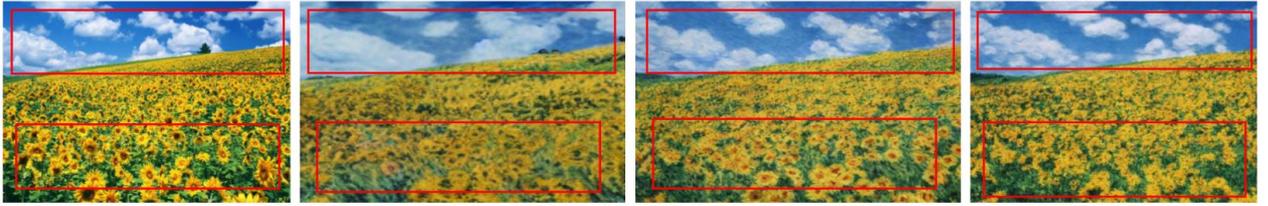


Hình 2.30 Biểu đồ thời gian sinh ảnh Diffusion LoRA Style Transfer

### Chất lượng ảnh

- Resize lớn hơn giúp giữ lại nhiều thông tin gốc, từ đó cải thiện rõ rệt chất lượng ảnh đầu ra. Tuy nhiên, cần đảm bảo ảnh gốc cũng có độ phân giải cao để phát huy hiệu quả này.
- Strength càng cao thì độ đậm nét của phong cách càng tăng. Tuy nhiên, với ảnh đầu vào chất lượng thấp, strength cao sẽ khiến ảnh bị biến dạng mạnh. Nếu ảnh gốc có chất lượng cao, resize lớn sẽ giúp ảnh giữ được nội dung tốt hơn dù tăng strength.
- Image cond scale không ảnh hưởng đến thời gian nhưng lại đóng vai trò quan trọng trong việc giữ nội dung ảnh gốc. Giá trị này càng cao thì ảnh đầu ra càng giống ảnh gốc hơn trong cùng bộ tham số.
- Chất lượng ảnh gốc (original quality) có ảnh hưởng rõ rệt:
  - Với ảnh chất lượng cao, strength từ 0.6 trở lên mới bắt đầu làm biến dạng ảnh.
  - Với ảnh chất lượng thấp, strength từ 0.4 đã khiến ảnh bị khác biệt lớn.





Hình 2.31 Môi trường quan strength, image\_cond\_scale và kích thước resize

- Step chỉ ảnh hưởng nhỏ đến chi tiết ảnh, nhưng vẫn hỗ trợ quá trình khử nhiễu hiệu quả hơn khi strength cao. Trong thực nghiệm, với ảnh chất lượng cao chỉ cần 30-50 step là đủ, trong khi ảnh chất lượng thấp cần đến 90-100 step để đạt độ chi tiết tối đa.
- Dựa trên đánh giá thông qua Gram Matrix và LPIPS, quyết định lựa chọn bộ tham số của Bộ tham số 1 là bộ tham số tốt nhất.

### Lựa chọn bộ tham số tốt nhất

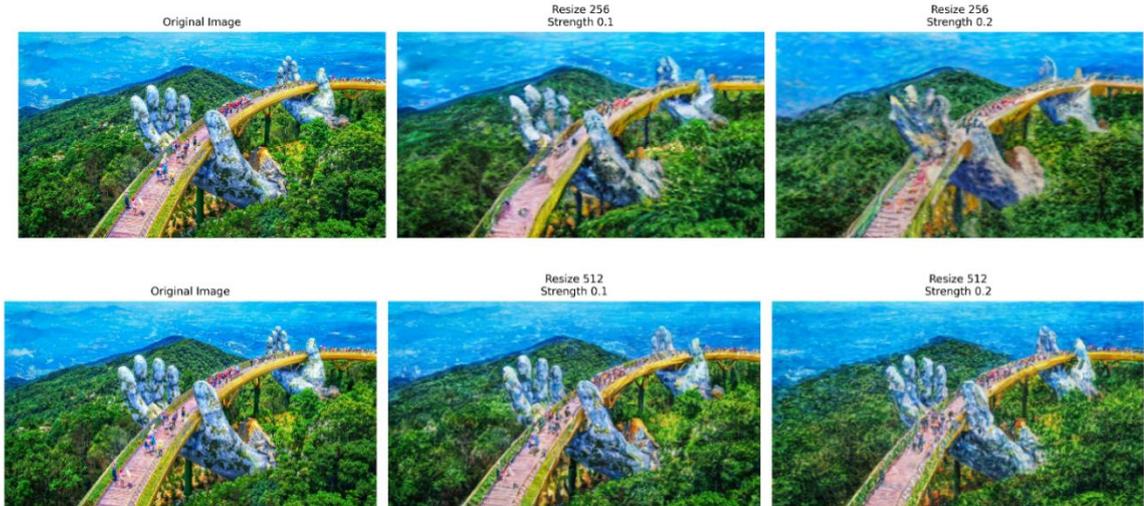
#### - Với kích thước 256 và 512

- Cận trên strength là khoảng 0.3, vượt qua 0.3 hình sẽ bắt đầu khác so với ảnh gốc. Dựa vào bảng time ở trên có thể dễ tính ra được thời gian sinh ảnh.

image_cond_scale	strength	infer_steps	resize	real step	Prompt	TIME
0.9 – 1.5	0.1 – 0.2	100	256	10 – 20	"Optional"	1s – 2s
0.9 – 1.5	0.1 – 0.2	100	512	10 – 20	"Optional"	3s – 6s

Bảng 2.12 Bộ tham số đề xuất kích thước 256, 512

- Đề xuất ít nhất là nên resize về 512 để giữ được nhiều thông tin ảnh, thời gian sinh ảnh cũng không quá lâu.



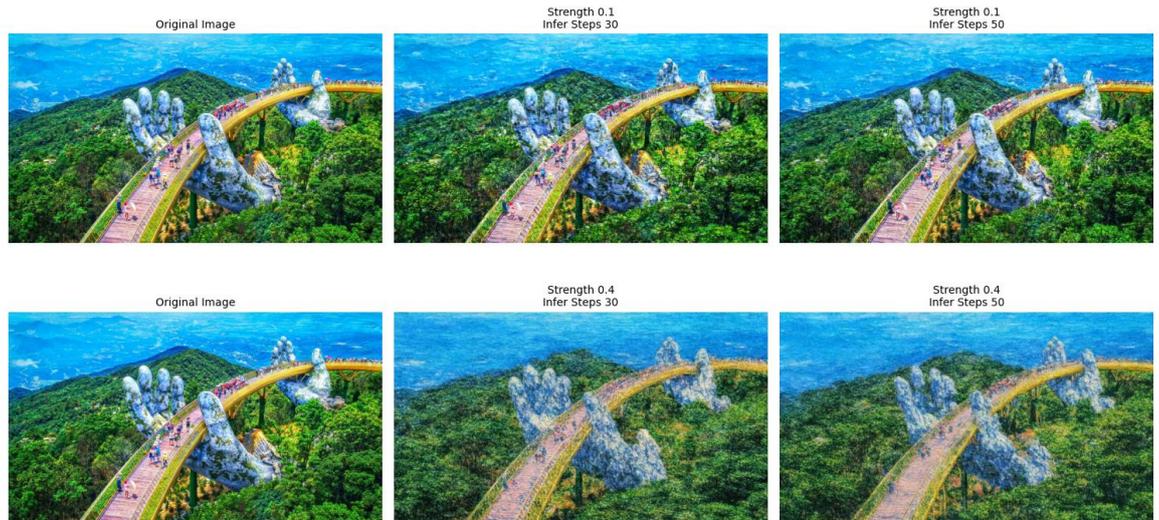
Hình 2.32 Ảnh tạo sinh kích thước 256 và 512

– Với kích thước 1024

- Cận trên strength là khoảng 0.5, vượt qua 0.5 hình sẽ bắt đầu khác so với ảnh gốc.

image	cond_scale	strength	infer_steps	resize	real_step	Prompt	TIME
	0.9 – 1.5	0.1 – 0.4	30	1024	03 – 12	"Optional"	7s – 24s
	0.9 – 1.5	0.1 – 0.4	50	1024	05 – 20	"Optional"	11s – 38s

Bảng 2.13 Bộ tham số đề xuất kích thước 1024



Hình 2.33 Ảnh tạo sinh kích thước 1024



Hình 2.34 Ma trận ảnh tạo sinh

## 2.7. Kết luận

Kết quả thử nghiệm cho thấy ba phương pháp Neural Style Transfer (NST), CycleGAN và Diffusion LoRA mang lại hiệu quả khác nhau trong bài toán sinh ảnh nghệ thuật:

- Neural Style Transfer (NST): Là phương pháp không yêu cầu dữ liệu huấn luyện ban đầu nhưng lại cho ra ảnh có chất lượng cao nhất và mang đậm nét tranh vẽ nhất. Thời gian sinh ảnh dao động từ 10 giây đến hơn 2 phút, tùy theo yêu cầu về chất lượng ảnh của người dùng.
- Diffusion LoRA: Yêu cầu thời gian huấn luyện từ 30 đến 60 phút, tùy thuộc vào tập dữ liệu. Thời gian sinh ảnh từ 3 giây đến hơn 3 phút. Ảnh đầu ra có chất lượng khá tốt, ưu điểm nổi bật là khả năng hiểu nội dung ảnh và biến đổi phù hợp theo phong cách, phù hợp với các tình huống cần tinh chỉnh nội dung mà vẫn giữ nguyên bố cục chính.
- CycleGAN: Là phương pháp cho kết quả kém nhất trong ba mô hình. Yêu cầu huấn luyện kéo dài từ 6 đến 12 giờ và ảnh đầu ra có chất lượng thấp hơn, khó kiểm soát phong cách và thiếu tính ổn định. Do đó, CycleGAN không phù hợp để triển khai trong các ứng dụng thực tế của hệ thống ArtGenix.

Hệ thống lựa chọn Neural Style Transfer làm phương pháp sinh ảnh nghệ thuật chính, nhờ vào chất lượng ảnh đầu ra vượt trội và khả năng hoạt động không cần huấn luyện. Hai phương pháp còn lại được tích hợp như các tùy chọn bổ sung, phục vụ cho các nhu cầu đặc biệt của người dùng.

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Trong chương này, phân tích hệ thống toàn diện được thực hiện, tập trung vào các chức năng cốt lõi và kiến trúc kỹ thuật của ArtGenix. Chương đi sâu vào các thiết kế chi tiết, phân tích cách hệ thống được cấu trúc và tổ chức để đáp ứng các yêu cầu chức năng và phi chức năng, bao gồm sinh ảnh nghệ thuật, tích hợp mạng xã hội, và quản lý tài nguyên người dùng.

### 3.1. Bối cảnh và động lực phát triển

Trong bối cảnh số hóa và sự phát triển mạnh mẽ của công nghệ học sâu, nhu cầu sáng tạo nghệ thuật số ngày càng gia tăng, đặc biệt trong các lĩnh vực như mỹ thuật, thiết kế và truyền thông. ArtGenix ra đời nhằm đáp ứng nhu cầu này, tích hợp các mô hình học sâu tiên tiến như Neural Style Transfer, CycleGAN và Diffusion LoRA để chuyển đổi ảnh thông thường thành các tác phẩm nghệ thuật độc đáo.

Hệ thống không chỉ hỗ trợ sáng tạo cá nhân mà còn cung cấp nền tảng mạng xã hội, cho phép người dùng chia sẻ, bình luận và tương tác với các tác phẩm trong thời gian thực. Điều này đáp ứng xu hướng xây dựng cộng đồng sáng tạo trực tuyến, giúp người dùng kết nối, truyền cảm hứng và nâng cao trải nghiệm nghệ thuật. ArtGenix cũng mở ra tiềm năng ứng dụng trong giáo dục mỹ thuật, hội họa, sáng tác truyện tranh, xây dựng api cung cấp cho các ứng dụng chỉnh sửa ảnh và ngành công nghiệp in ấn, từ đó thúc đẩy đổi mới sáng tạo và cải thiện hiệu quả kinh doanh cho các doanh nghiệp liên quan.

### 3.2. Phân tích yêu cầu hệ thống

#### 3.2.1. Yêu cầu chức năng

– **User:**

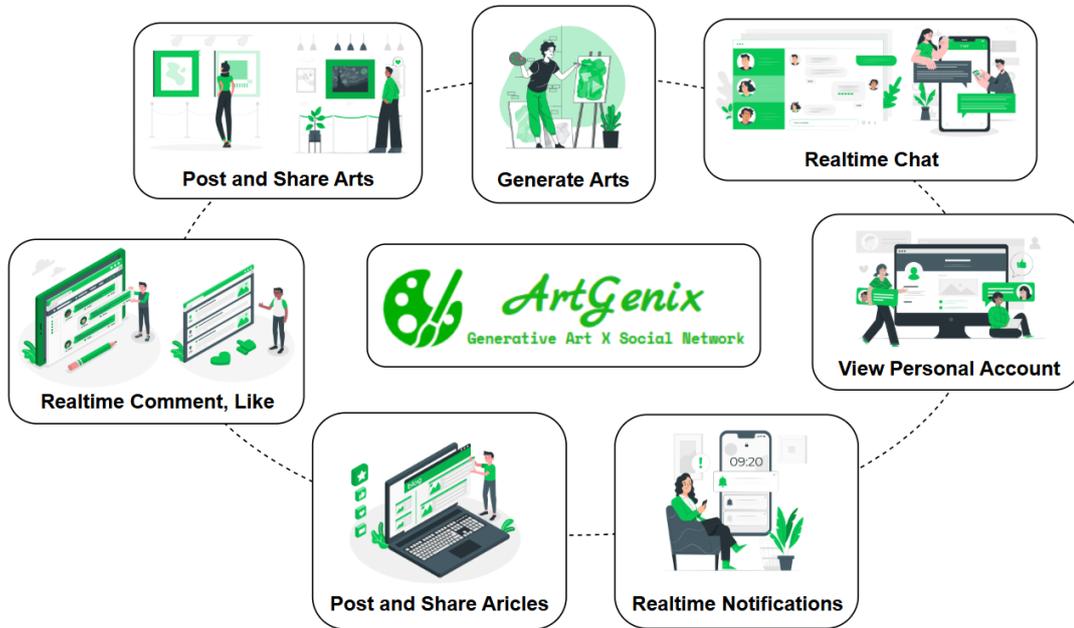
- Tải ảnh nội dung và phong cách để tạo tác phẩm nghệ thuật thông qua các mô hình học sâu.
- Tìm kiếm và khám phá các tác phẩm nghệ thuật từ cộng đồng.
- Chia sẻ, bình luận, thích và nhấn tin trực tuyến để tương tác với người dùng khác.
- Quản lý bộ sưu tập cá nhân, bao gồm lưu trữ và sắp xếp tác phẩm.
- Xem thông báo về hoạt động mạng xã hội và lịch sử tương tác.

– **Admin:**

- Quản lý tài khoản người dùng, bao gồm xác minh, khóa tài khoản.
- Kiểm duyệt nội dung (tác phẩm, bình luận) để đảm bảo tuân thủ quy định.
- Quản lý thẻ (tags) và danh mục để hỗ trợ tìm kiếm, phân loại tác phẩm, bài viết.
- Quản lý toàn bộ tài nguyên của hệ thống.

– **Quản lý (Manager/Super Admin):**

- Quản lý đội ngũ Admin, bao gồm phân quyền và giám sát hoạt động.
- Xem thống kê hệ thống, bao gồm số lượng tác phẩm, bài viết, người dùng hoạt động và tương tác mạng xã hội.
- Quản lý cấu hình các tham số của hệ thống như đường dẫn gọi đến API Model.



Hình 3.1 Các tính năng cốt lõi và nổi bật của ArtGenix

– **Các tính năng cốt lõi của hệ thống**

- Tạo tác phẩm nghệ thuật: Người dùng tải ảnh nội dung và phong cách, sử dụng các mô hình học sâu (Neural Style Transfer, CycleGAN, Diffusion LoRA) để tạo ra các tác phẩm nghệ thuật độc đáo, hỗ trợ sáng tạo cá nhân và ứng dụng trong giáo dục mỹ thuật, in ấn.
- Mạng xã hội: Tích hợp các tính năng chia sẻ, bình luận, thích và nhắn tin, cho phép người dùng xây dựng cộng đồng sáng tạo, trao đổi ý tưởng và truyền cảm hứng lẫn nhau.
- Quản lý bộ sưu tập: Người dùng lưu trữ, sắp xếp và quản lý các tác phẩm nghệ thuật trong bộ sưu tập cá nhân, tăng cường trải nghiệm cá nhân hóa.
- Thống kê và báo cáo: Quản lý truy cập các báo cáo về hoạt động hệ thống, như số lượng tác phẩm, tương tác người dùng, hỗ trợ tối ưu hóa vận hành và phát triển chiến lược.

3.2.2. **Yêu cầu phi chức năng**

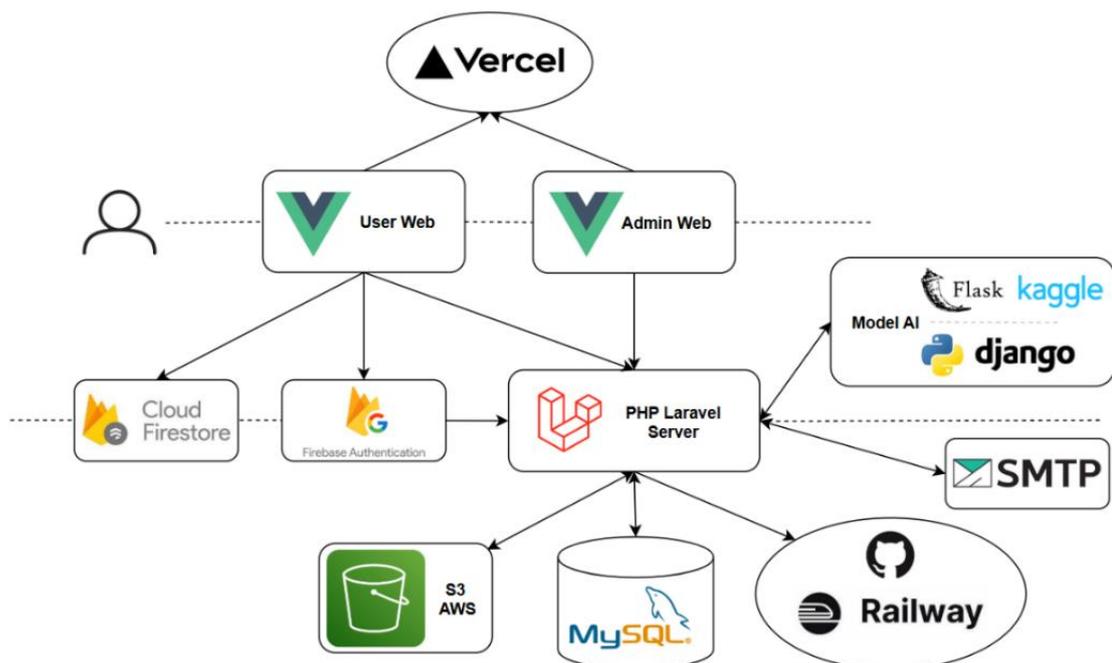
STT	Đặc tính	Yêu cầu
1	Bảo mật	1. Web Server triển khai Railway

		<ol style="list-style-type: none"> <li>Quản lý đăng nhập: Bảo vệ thông tin cá nhân của người dùng với các biện pháp bảo mật như mã hóa mật khẩu.</li> <li>Backup và phục hồi dữ liệu: Cung cấp khả năng sao lưu và khôi phục dữ liệu để ngăn chặn mất mát dữ liệu quan trọng.</li> <li>Dữ liệu hệ thống được sao lưu hằng ngày và được lưu an toàn bên ngoài trang web</li> </ol>
2	Hiệu Suất	<ol style="list-style-type: none"> <li>Tốc độ tải trang: Đảm bảo tốc độ tải trang nhanh chóng để duy trì trải nghiệm người dùng tốt.</li> <li>Hỗ trợ đa nền tảng: Đảm bảo khả năng hoạt động mượt mà trên nhiều nền tảng và thiết bị khác nhau.</li> <li>Thời gian nhập/xuất dữ liệu nhanh .Thời gian nhập/xuất dữ liệu nhanh</li> <li>Cho phép 1 lượng lớn truy cập bằng internet vào hệ thống</li> </ol>
3	Mở Rộng	<ol style="list-style-type: none"> <li>Khả năng lưu trữ dữ liệu lớn</li> <li>Cho phép nhiều người dùng thực hiện các thao tác giống nhau trong cùng khoảng thời gian</li> <li>Khả năng mở rộng: Xây dựng hệ thống có khả năng mở rộng để chứa lượng người dùng, tác phẩm nghệ thuật, bài viết tăng lên</li> </ol>
4	Trình duyệt	<ol style="list-style-type: none"> <li>IE6 trở lên (IE8 không được khuyến khích).</li> <li>Trình duyệt Chrome và Firefox</li> </ol>
5	Giả định	<ol style="list-style-type: none"> <li>Có thể tạm ngưng hệ thống nếu cần phải nâng cấp</li> </ol>

Bảng 3.1 Yêu cầu phi chức năng

### 3.3. Thiết kế hệ thống

Đây là phần kiến trúc của toàn bộ hệ thống (trong đó bao gồm cả phần triển trúc đã được triển khai)



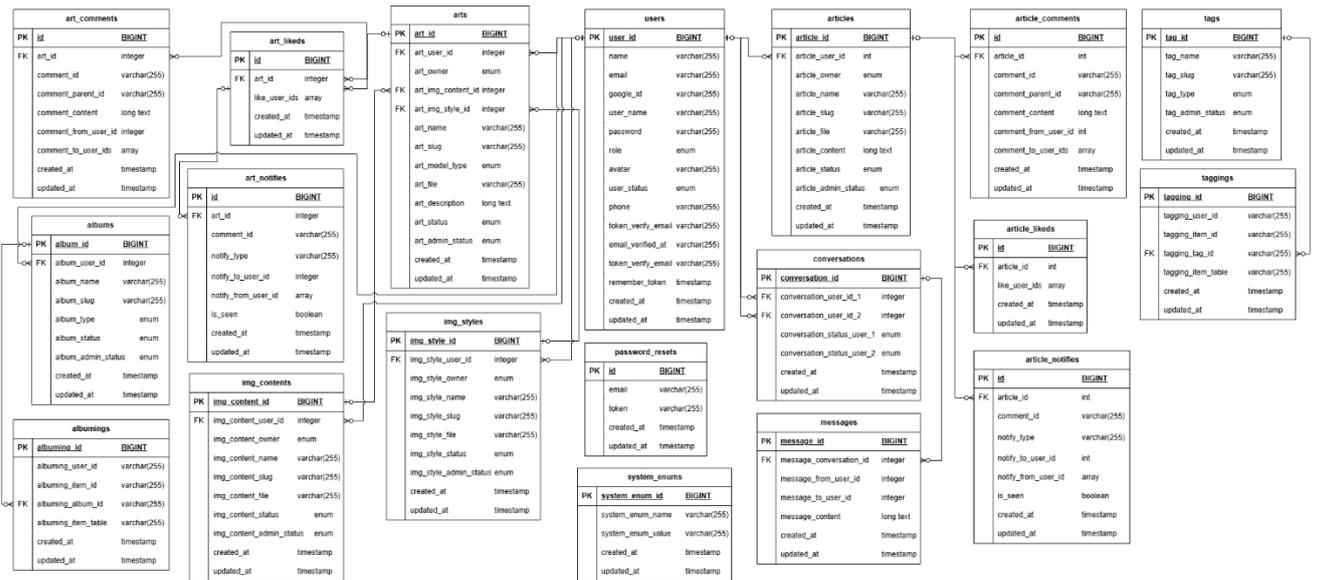
Hình 3.2 Kiến trúc của hệ thống



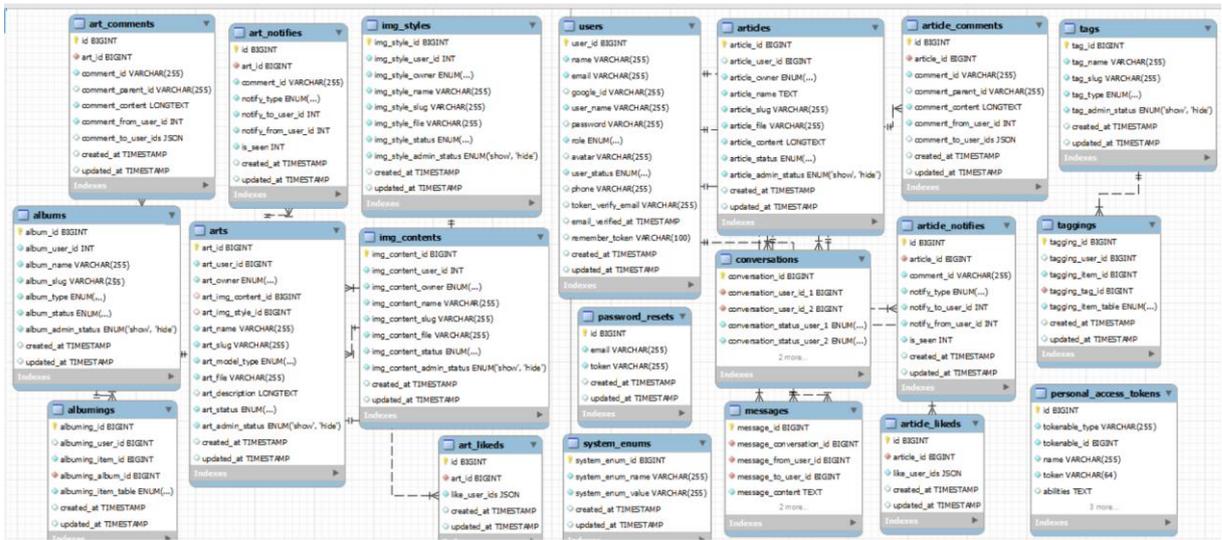
Hình 3.3 Tech Stack

### 3.4. Thiết kế cơ sở dữ liệu

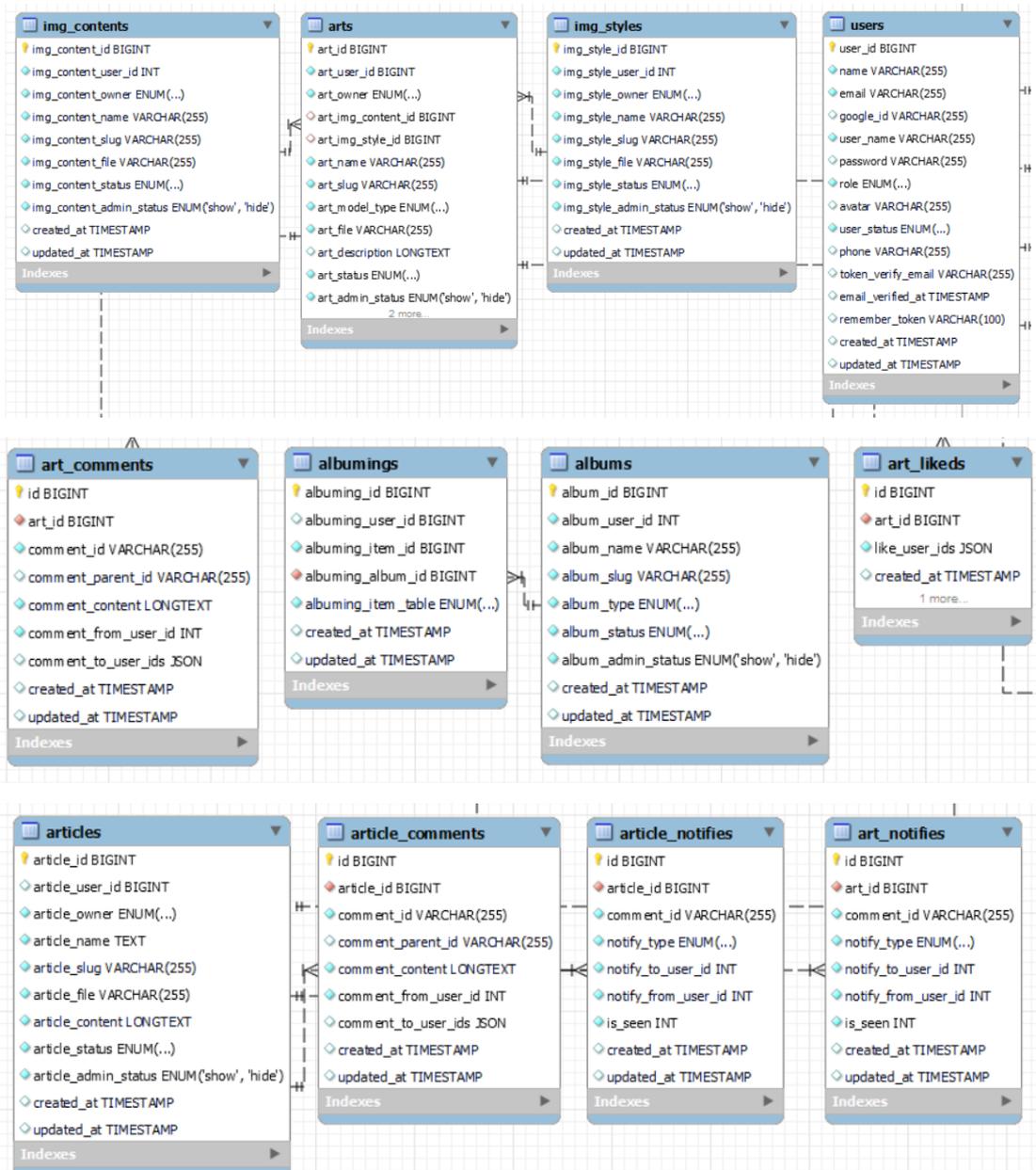
#### 3.4.1. Lựa chọn cơ sở dữ liệu

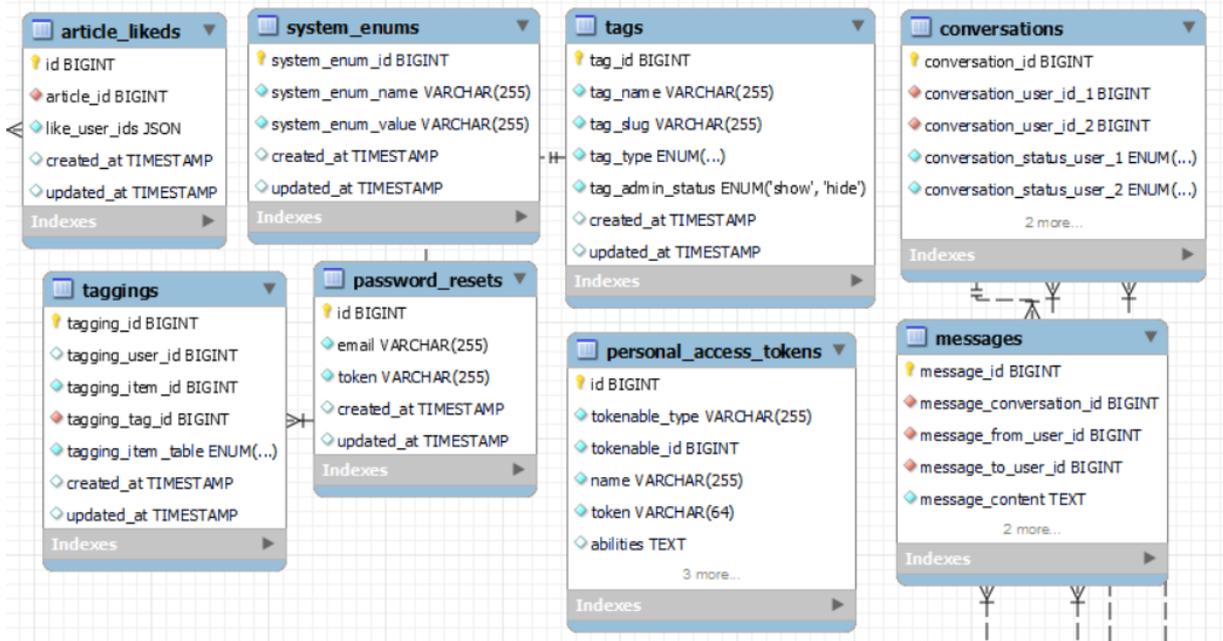


Hình 3.4 Database Diagrams



Hình 3.5 Lược đồ cơ sở dữ liệu





Hình 3.6 Chi tiết các bảng trong lược đồ cơ sở dữ liệu

### 3.4.2. Danh sách các bảng

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	user_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	name	VARCHAR(255)	Tên người dùng
3	email	VARCHAR(255)	Email người dùng, duy nhất
4	google_id	VARCHAR(255), NULLABLE	ID Google (nếu đăng nhập Google)
5	user_name	VARCHAR(255)	Tên hiển thị
6	password	VARCHAR(255), NULLABLE	Mật khẩu (mã hóa)
7	role	ENUM	Vai trò người dùng
8	avatar	VARCHAR(255), NULLABLE	Ảnh đại diện
9	user_status	ENUM	Trạng thái tài khoản
10	phone	VARCHAR(255), NULLABLE	Số điện thoại
11	token_verify_email	VARCHAR(255), NULLABLE	Mã xác thực email
12	email_verified_at	TIMESTAMP, NULLABLE	Thời gian xác thực email
13	remember_token	VARCHAR(100), NULLABLE	Token lưu đăng nhập
14	created_at	TIMESTAMP	Thời gian tạo
15	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.2 Mô tả chi tiết bảng User

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	email	VARCHAR(255)	Email người dùng, có chỉ số
3	token	VARCHAR(255)	Mã token để đặt lại mật khẩu
4	created_at	TIMESTAMP, NULLABLE	Thời gian tạo
5	updated_at	TIMESTAMP, NULLABLE	Thời gian cập nhật

Bảng 3.3 Mô tả chi tiết bảng Password Resets

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	tag_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	tag_name	VARCHAR(255)	Tên thẻ
3	tag_slug	VARCHAR(255)	Slug thẻ, duy nhất, định dạng: tag_type-tag_name
4	tag_type	ENUM	Loại thẻ ('arts', 'articles', 'img_styles', 'img_contents')
5	tag_admin_status	ENUM	Trạng thái quản trị ('show', 'hide'), mặc định 'show'
6	created_at	TIMESTAMP	Thời gian tạo
7	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.4 Mô tả chi tiết bảng Tags

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	tagging_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	tagging_user_id	BIGINT UNSIGNED, NULLABLE	ID người dùng gắn thẻ
3	tagging_item_id	BIGINT UNSIGNED	ID mục được gắn thẻ
4	tagging_tag_id	BIGINT UNSIGNED	ID thẻ, khóa ngoại liên kết với bảng tags
5	tagging_item_table	ENUM	Bảng chứa mục được gắn thẻ
6	created_at	TIMESTAMP	Thời gian tạo
7	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.5 Mô tả chi tiết bảng Taggings

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	album_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	album_user_id	INTEGER	ID người dùng tạo album
3	album_name	VARCHAR(255)	Tên album
	album_slug	VARCHAR(255)	Slug album, duy nhất
4	album_type	ENUM	Loại album ('arts', 'articles', 'img_styles', 'img_contents')
5	album_status	ENUM	Trạng thái album ('public', 'private')
6	album_admin_status	ENUM	Trạng thái quản trị ('show', 'hide')
7	created_at	TIMESTAMP	Thời gian tạo
8	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.6 Mô tả chi tiết bảng Albums

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	albuming_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	albuming_user_id	BIGINT UNSIGNED	ID người dùng thêm vào album
3	albuming_item_id	BIGINT UNSIGNED	ID mục được thêm vào album
4	albuming_album_id	BIGINT UNSIGNED	ID album, khóa ngoại liên kết với bảng albums
5	albuming_item_table	ENUM	Bảng chứa mục được thêm
6	created_at	TIMESTAMP	Thời gian tạo
7	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.7 Mô tả chi tiết bảng Albumings

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	img_content_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	img_content_user_id	INTEGER	ID người dùng tạo nội dung hình ảnh
3	img_content_owner	ENUM	Chủ sở hữu nội dung hình ảnh ('admin', 'user')
4	img_content_name	VARCHAR(255)	Tên nội dung hình ảnh
5	img_content_slug	VARCHAR(255)	Slug nội dung, duy nhất
6	img_content_file	VARCHAR(255)	Tệp nội dung hình ảnh
7	img_content_status	ENUM	Trạng thái nội dung
8	img_content_admin_status	ENUM	Trạng thái quản trị
9	created_at	TIMESTAMP	Thời gian tạo
10	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.8 Mô tả chi tiết bảng Img Contents

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	img_style_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	img_style_user_id	INTEGER	ID người dùng tạo phong cách hình ảnh
3	img_style_owner	ENUM	Chủ sở hữu phong cách hình ảnh ('admin', 'user')
4	img_style_name	VARCHAR(255)	Tên phong cách hình ảnh
5	img_style_slug	VARCHAR(255)	Slug phong cách, duy nhất
6	img_style_file	VARCHAR(255)	Tệp phong cách hình ảnh
7	img_style_status	ENUM	Trạng thái phong cách
8	img_style_admin_status	ENUM	Trạng thái quản trị
9	created_at	TIMESTAMP	Thời gian tạo
10	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.9 Mô tả chi tiết bảng Img Styles

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	art_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	art_user_id	BIGINT UNSIGNED	ID người dùng tạo tác phẩm nghệ thuật
3	art_owner	ENUM	Chủ sở hữu tác phẩm ('admin', 'user')
4	art_img_content_id	BIGINT UNSIGNED, NULLABLE	ID nội dung hình ảnh, khóa ngoại liên kết với bảng img_contents
5	art_img_style_id	BIGINT UNSIGNED, NULLABLE	ID phong cách hình ảnh, khóa ngoại liên kết với bảng img_styles
6	art_name	VARCHAR(255)	Tên tác phẩm nghệ thuật
7	art_slug	VARCHAR(255)	Slug tác phẩm, duy nhất, định dạng: art_id-art_name
8	art_model_type	ENUM	Loại mô hình AI
9	art_file	VARCHAR(255)	Tệp tác phẩm nghệ thuật
10	art_description	LONGTEXT, NULLABLE	Mô tả tác phẩm nghệ thuật
11	art_status	ENUM	Trạng thái tác phẩm
12	art_admin_status	ENUM	Trạng thái quản trị ('show', 'hide'), mặc định 'show'
13	created_at	TIMESTAMP	Thời gian tạo
14	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.10 Mô tả chi tiết bảng Arts

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	art_id	BIGINT UNSIGNED	ID tác phẩm, khóa ngoại với bảng arts
3	comment_id	VARCHAR(255)	ID bình luận
4	comment parent id	VARCHAR(255), NULLABLE	ID bình luận cha (nếu có)
5	comment content	LONGTEXT	Nội dung bình luận
6	comment from user id	INTEGER	ID người dùng gửi bình luận
7	comment_to_user_ids	JSON, NULLABLE	Danh sách ID người dùng được gắn thẻ
8	created at	TIMESTAMP	Thời gian tạo
9	updated at	TIMESTAMP	Thời gian cập nhật

Bảng 3.11 Mô tả chi tiết bảng Comments

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	art_id	BIGINT UNSIGNED	ID tác phẩm, khóa ngoại liên kết với bảng arts
3	like_user_ids	JSON	Danh sách ID người dùng đã thích tác phẩm
4	created at	TIMESTAMP	Thời gian tạo
5	updated at	TIMESTAMP	Thời gian cập nhật

Bảng 3.12 Mô tả chi tiết bảng Arts Likeds

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	art_id	BIGINT UNSIGNED	ID tác phẩm, khóa ngoại liên kết với bảng arts
3	comment_id	VARCHAR(255)	ID bình luận
4	notify_type	ENUM	Loại thông báo
5	notify to user id	INTEGER	ID người dùng nhận thông báo
6	notify from user id	INTEGER	ID người dùng tạo thông báo
7	is seen	INTEGER	Trạng thái xem
8	created_at	TIMESTAMP	Thời gian tạo
9	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.13 Mô tả chi tiết bảng Art Notifies

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	article_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	article_user_id	BIGINT UNSIGNED, NULLABLE	ID người dùng tạo bài viết
3	article_owner	ENUM	Chủ sở hữu bài viết ('admin', 'user')
4	article_name	TEXT	Tên bài viết
5	article_slug	VARCHAR(255)	Slug bài viết, duy nhất
6	article_file	VARCHAR(255)	Tệp liên quan đến bài viết
7	article_content	LONGTEXT	Nội dung bài viết
8	article_status	ENUM	Trạng thái bài viết

9	article_admin_status	ENUM	Trạng thái quản trị
10	created_at	TIMESTAMP	Thời gian tạo
11	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.14 Mô tả chi tiết bảng Articles

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	article_id	BIGINT UNSIGNED	ID bài viết, khóa ngoại liên kết với bảng articles
3	comment_id	VARCHAR(255)	ID bình luận
4	comment_parent_id	VARCHAR(255), NULLABLE	ID bình luận cha (nếu có)
5	comment_content	LONGTEXT	Nội dung bình luận
6	comment_from_user_id	INTEGER	ID người dùng gửi bình luận
7	comment_to_user_ids	JSON, NULLABLE	Danh sách ID người dùng được gắn thẻ
8	created_at	TIMESTAMP	Thời gian tạo
9	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.15 Mô tả chi tiết bảng Article Comments

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	article_id	BIGINT UNSIGNED	ID bài viết, khóa ngoại liên kết với bảng articles
3	like_user_ids	JSON	Danh sách ID người dùng đã thích bài viết
4	created_at	TIMESTAMP	Thời gian tạo
5	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.16 Mô tả chi tiết bảng Article Likeds

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	article_id	BIGINT UNSIGNED	ID bài viết, khóa ngoại liên kết với bảng articles
3	comment_id	VARCHAR(255)	ID bình luận
4	notify_type	ENUM	Loại thông báo
5	notify_to_user_id	INTEGER	ID người dùng nhận thông báo
6	notify_from_user_id	INTEGER	ID người dùng tạo thông báo
7	is_seen	INTEGER	Trạng thái xem
8	created_at	TIMESTAMP	Thời gian tạo
9	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.17 Mô tả chi tiết bảng Article Notifies

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	conversation_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	conversation_user_id_1	BIGINT UNSIGNED	ID người dùng 1, khóa ngoại với bảng users
3	conversation_user_id_2	BIGINT UNSIGNED	ID người dùng 2, khóa ngoại với bảng users

4	conversation_status_user_1	ENUM	Trạng thái xem của người dùng 1
5	conversation_status_user_2	ENUM	Trạng thái xem của người dùng 2
6	created_at	TIMESTAMP	Thời gian tạo
7	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.18 Mô tả chi tiết bảng Conversations

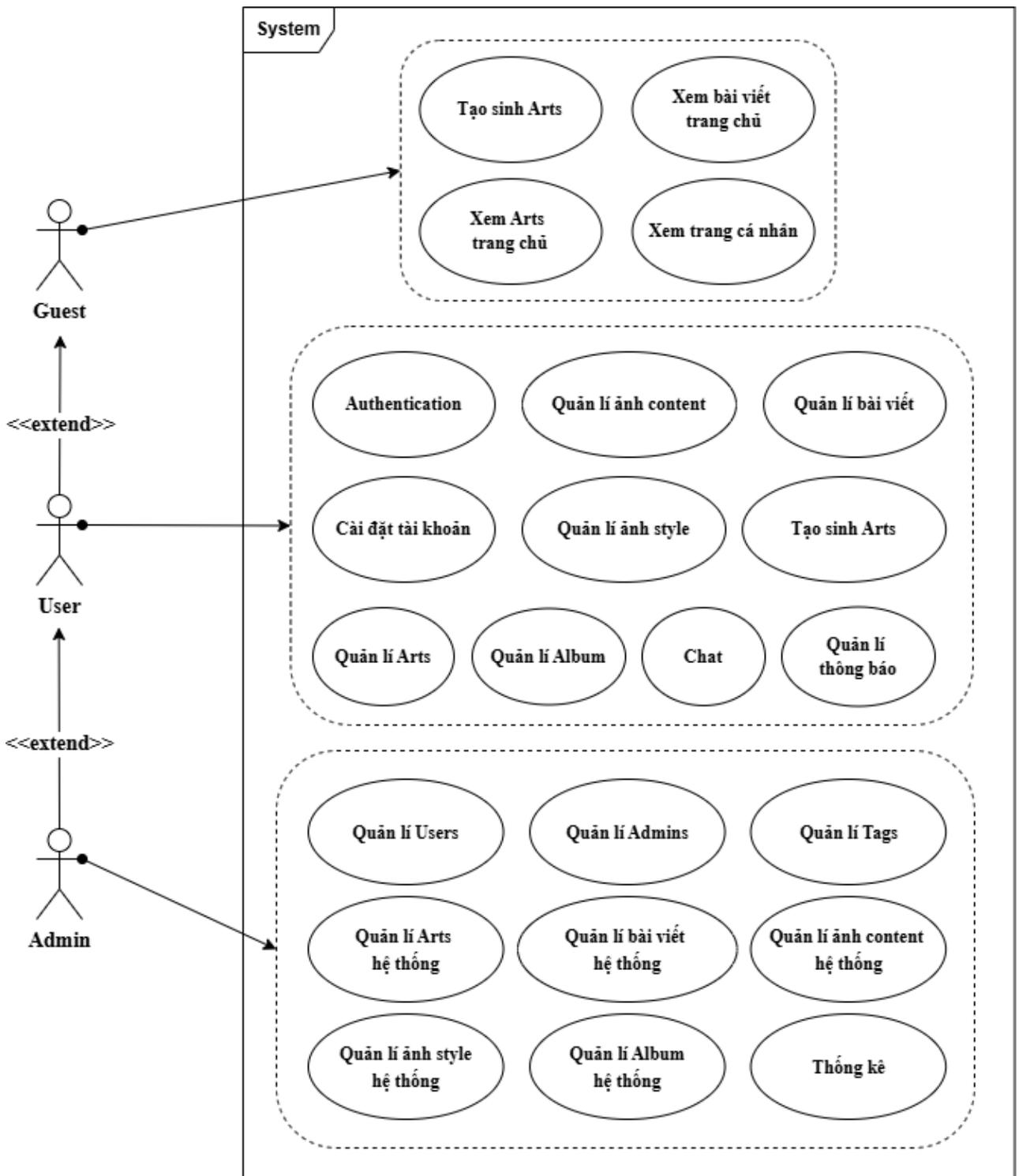
STT	Tên cột	Kiểu dữ liệu	Mô tả
1	message_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	message_conversation_id	BIGINT UNSIGNED	ID cuộc trò chuyện, liên kết bảng conversations
3	message_from_user_id	BIGINT UNSIGNED	ID người dùng gửi tin nhắn, liên kết bảng users
4	message_to_user_id	BIGINT UNSIGNED	ID người dùng nhận tin nhắn, liên kết bảng users
5	message_content	TEXT	Nội dung tin nhắn
6	created_at	TIMESTAMP	Thời gian tạo
7	updated_at	TIMESTAMP	Thời gian cập nhật

Bảng 3.19 Mô tả chi tiết bảng Messages

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	system_enum_id	BIGINT UNSIGNED	Khóa chính, tự động tăng
2	system_enum_name	VARCHAR(255)	Tên giá trị enum
3	system_enum_value	VARCHAR(255)	Giá trị enum
4	created_at	TIMESTAMP	Thời gian tạo
5	updated_at	TIMESTAMP	Thời gian cập nhật

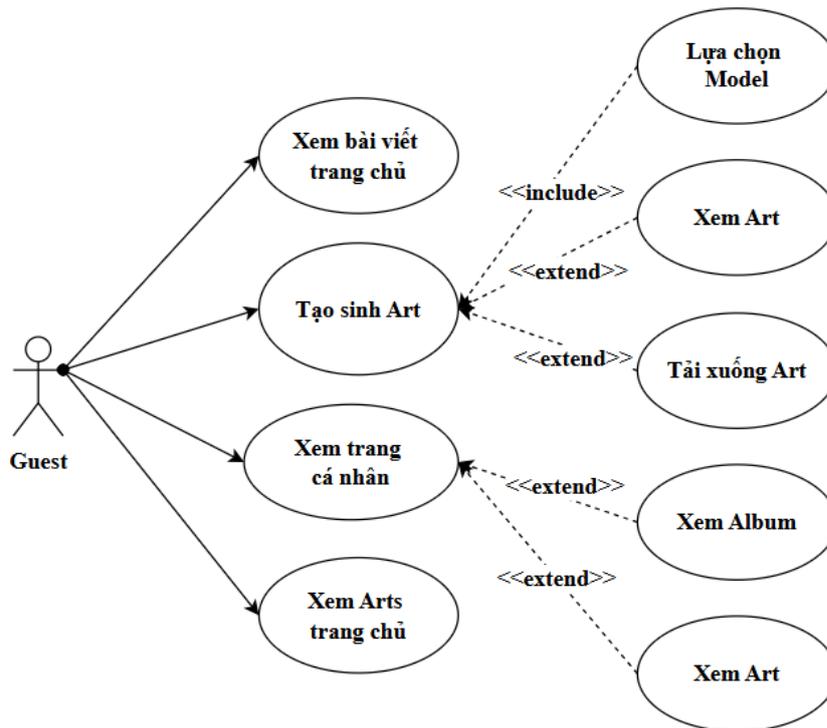
Bảng 3.20 Mô tả chi tiết bảng System Enums

### 3.5. Use case Diagrams



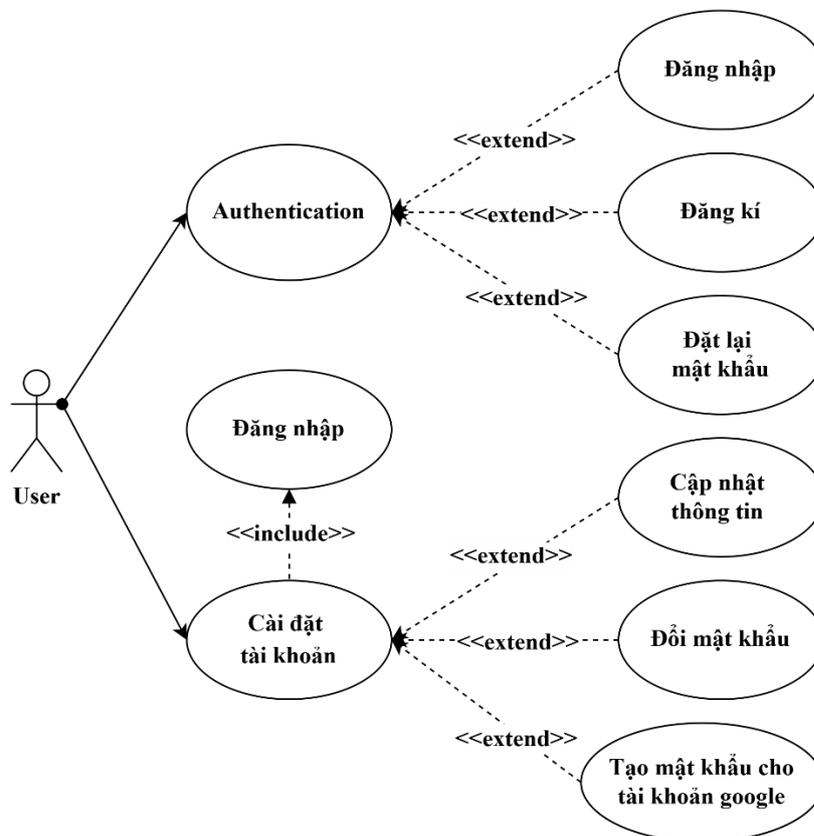
Hình 3.7 Biểu đồ Use case tổng quát hệ thống

### 3.5.1. Biểu đồ Use case của Guest

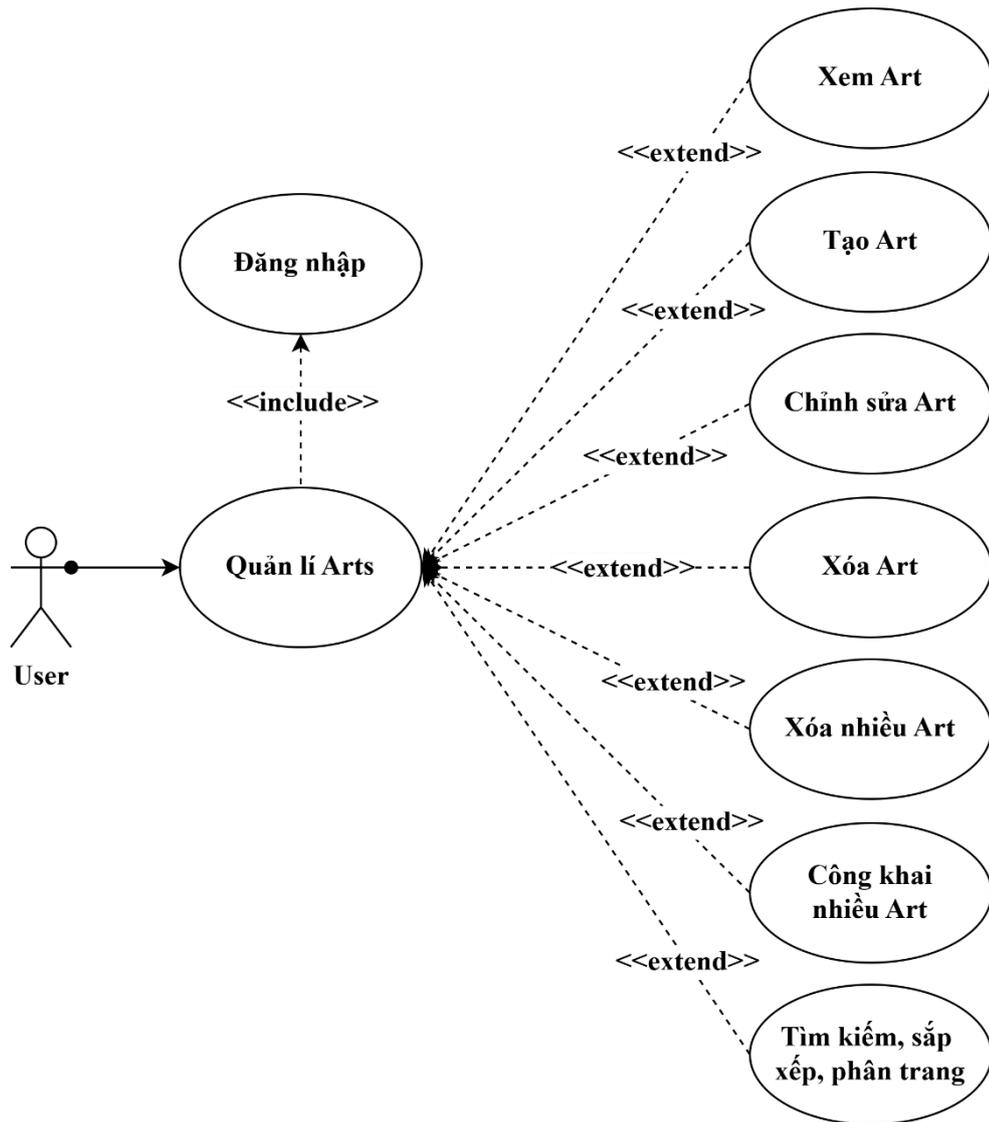


Hình 3.8 Biểu đồ Use case Guest và các tính năng liên quan

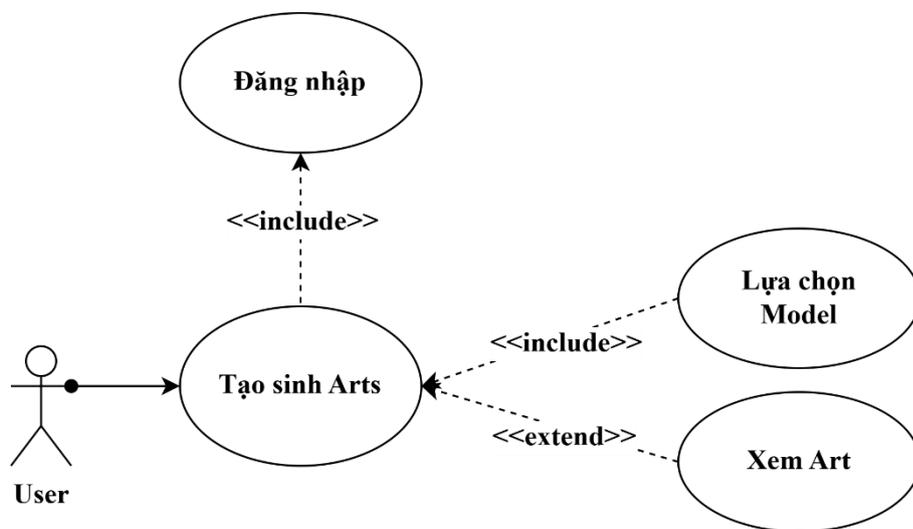
### 3.5.2. Danh sách biểu đồ Use case của User



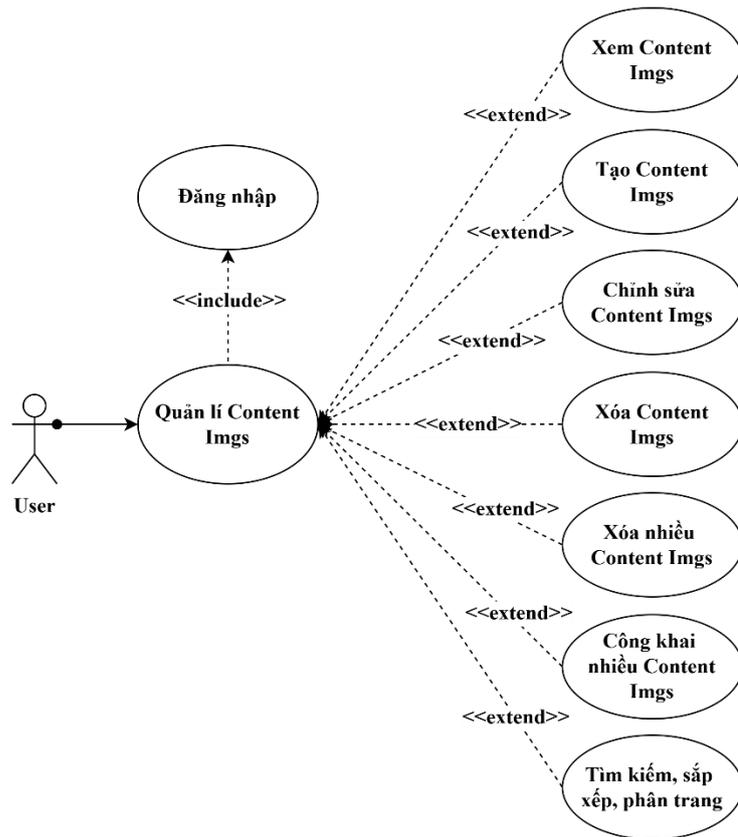
Hình 3.9 Biểu đồ Use case User xác thực và cài đặt tài khoản



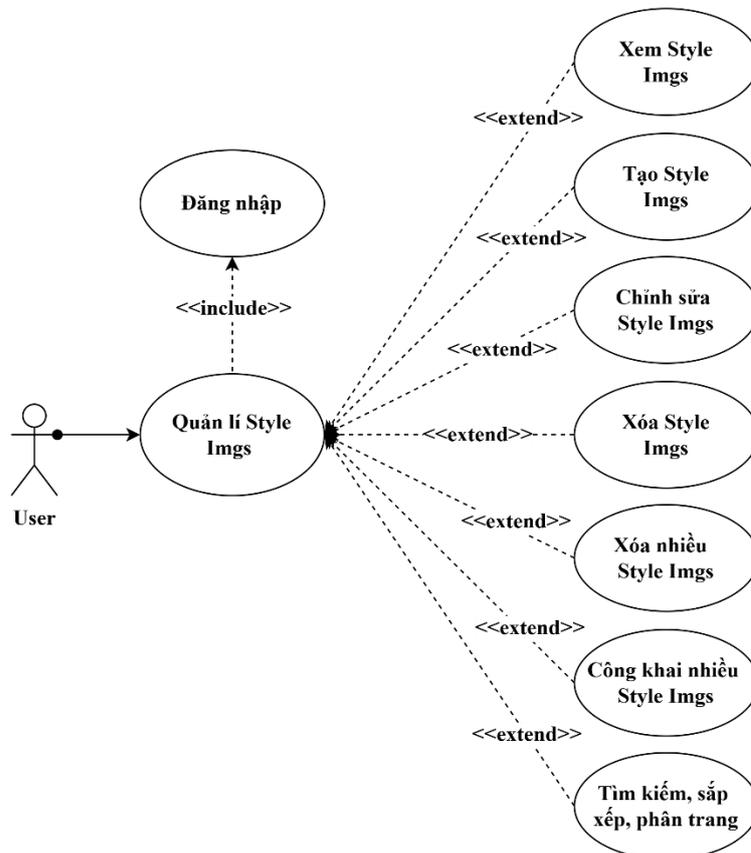
Hình 3.10 Biểu đồ Use case User quản lý tác phẩm nghệ thuật



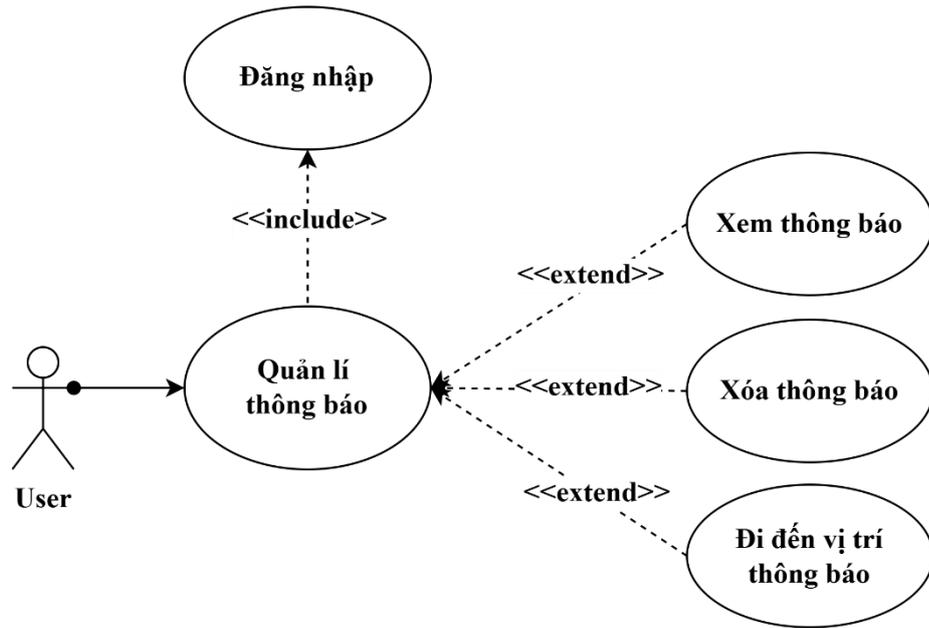
Hình 3.11 Biểu đồ Use case User tạo sinh tác phẩm nghệ thuật



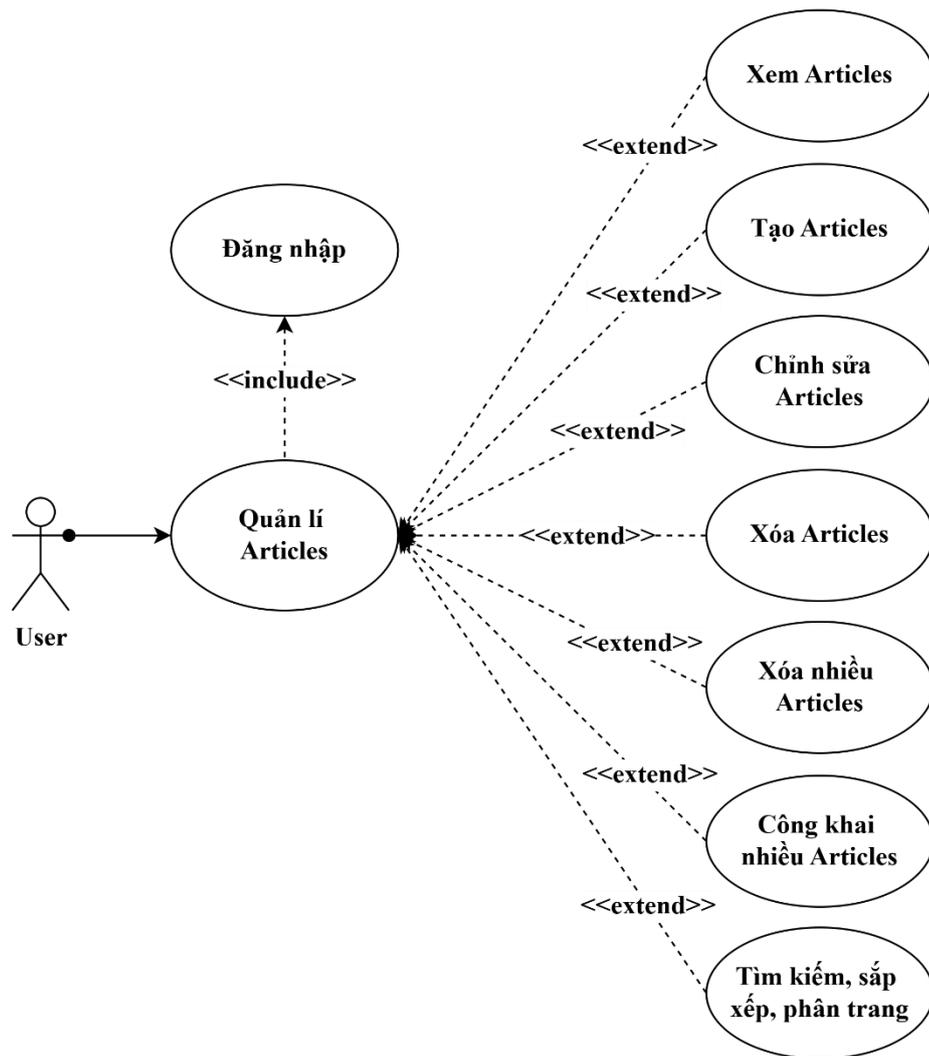
Hình 3.12 Biểu đồ Use case User quản lý ảnh nội dung



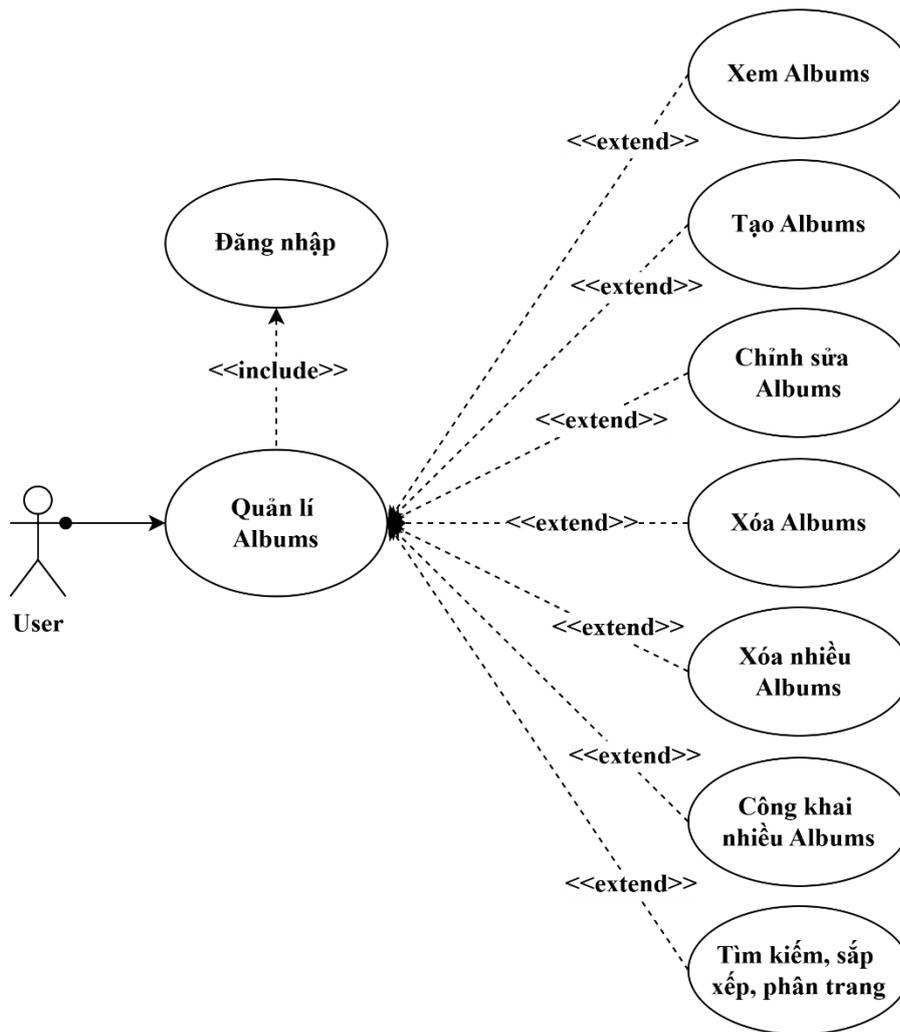
Hình 3.13 Biểu đồ Use case User quản lý ảnh phong cách



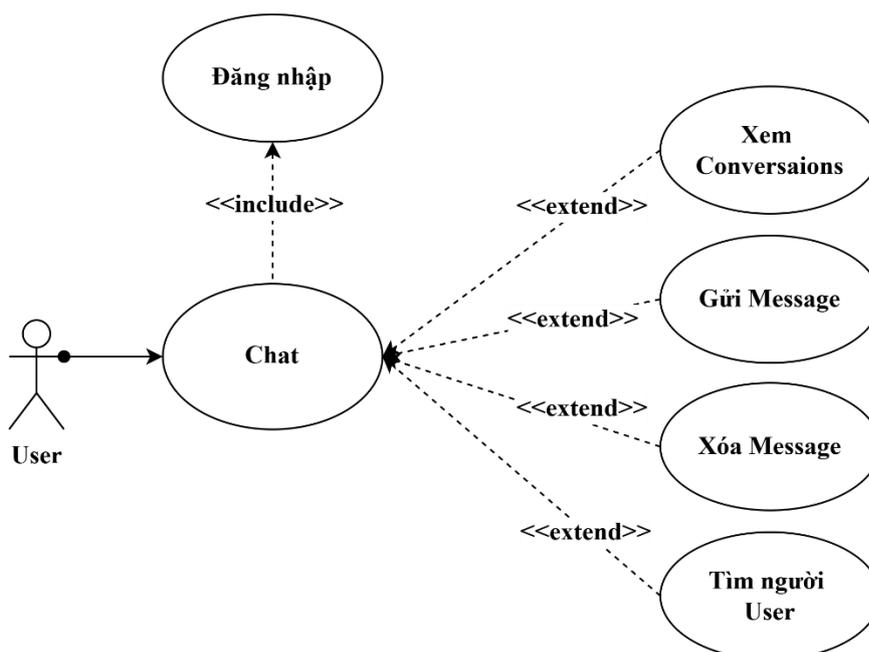
Hình 3.14 Biểu đồ Use case User quản lý thông báo



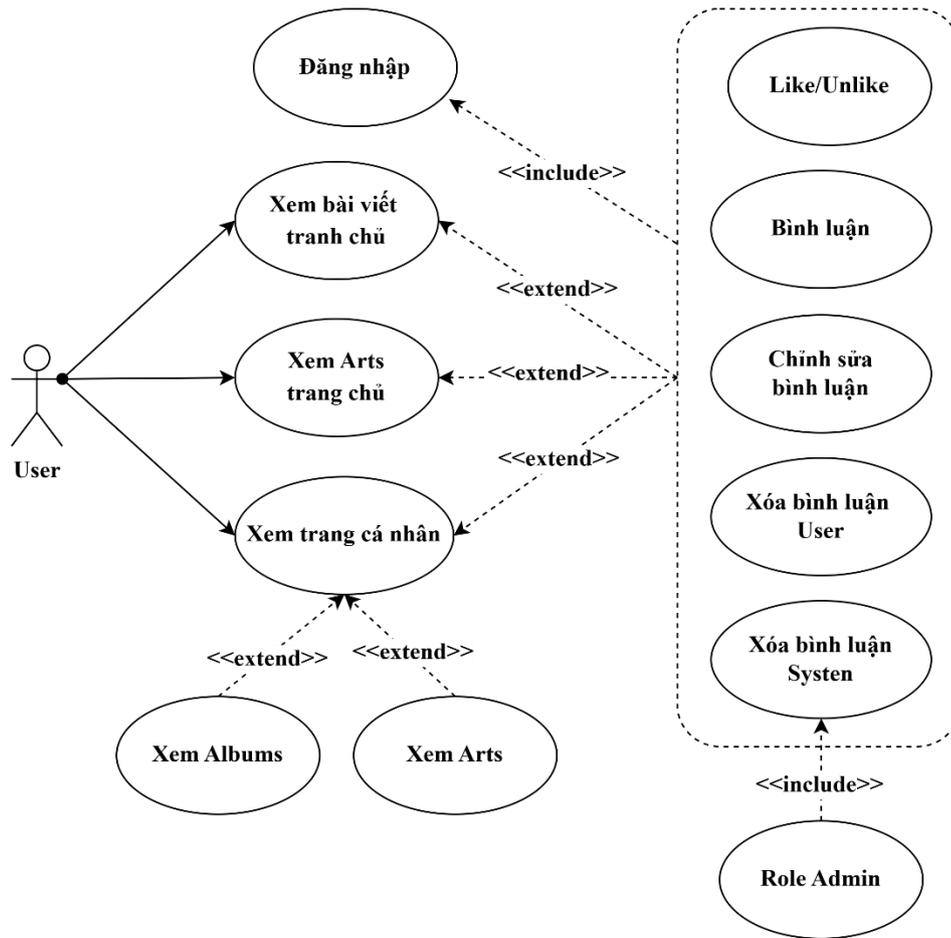
Hình 3.15 Biểu đồ Use case User quản lý bài viết



Hình 3.16 Biểu đồ Use case Use quản lý bộ sưu tập

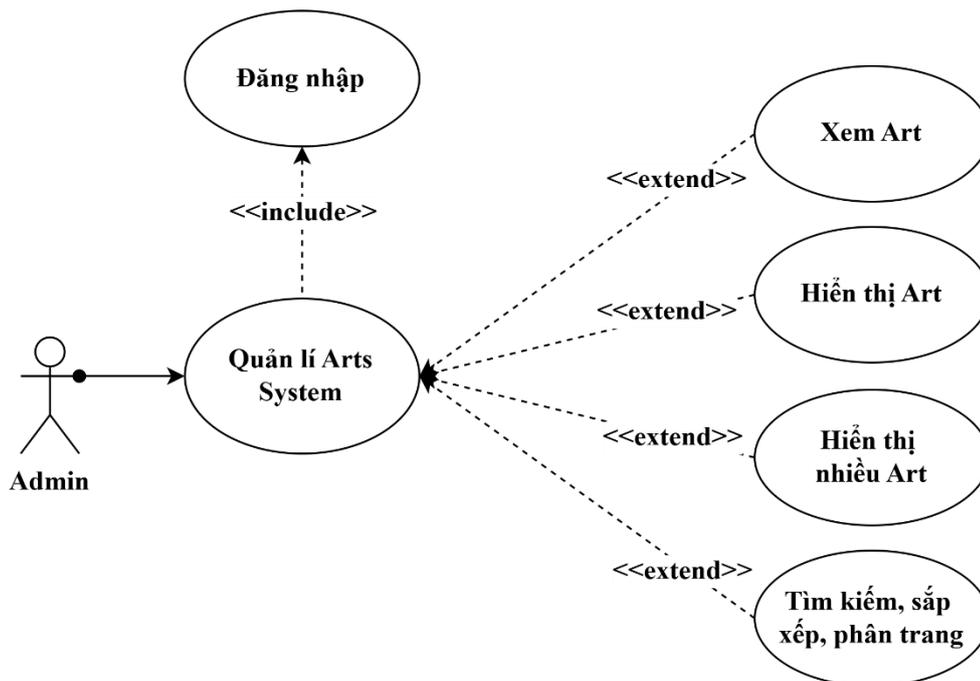


Hình 3.17 Biểu đồ Use case Use quản lý cuộc hội thoại, nhắn tin trực tuyến

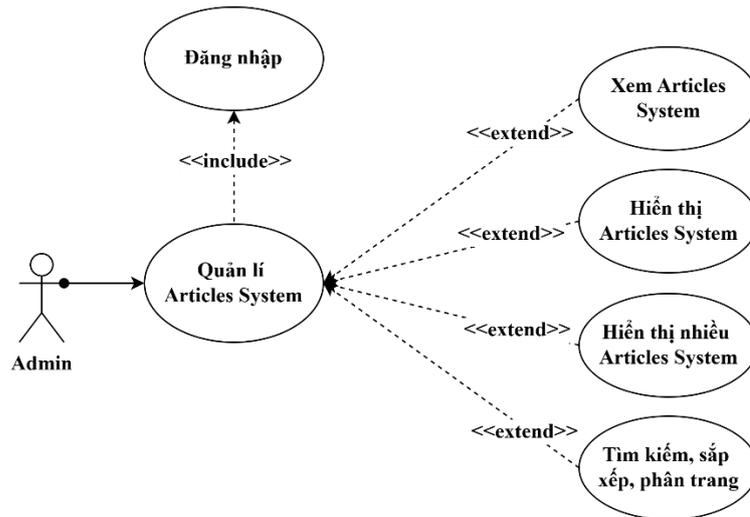


Hình 3.18 Biểu đồ Use case Use tương tác ở trang chủ

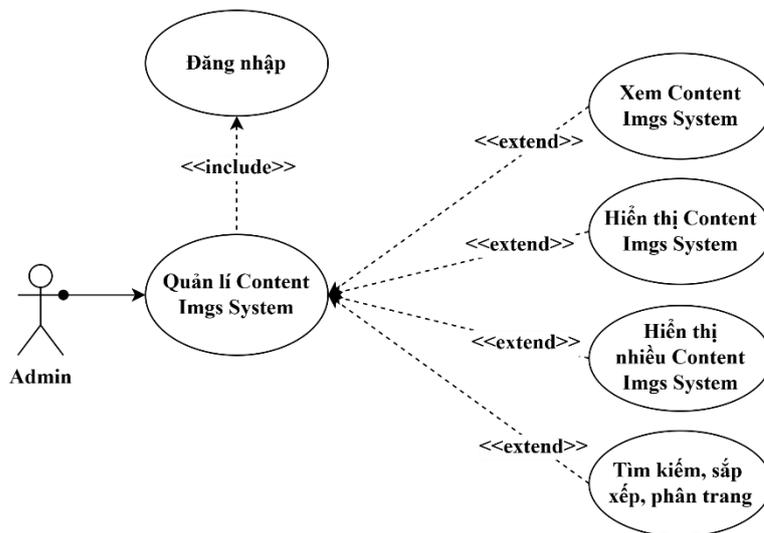
### 3.5.3. Danh sách biểu đồ Use case của Admin



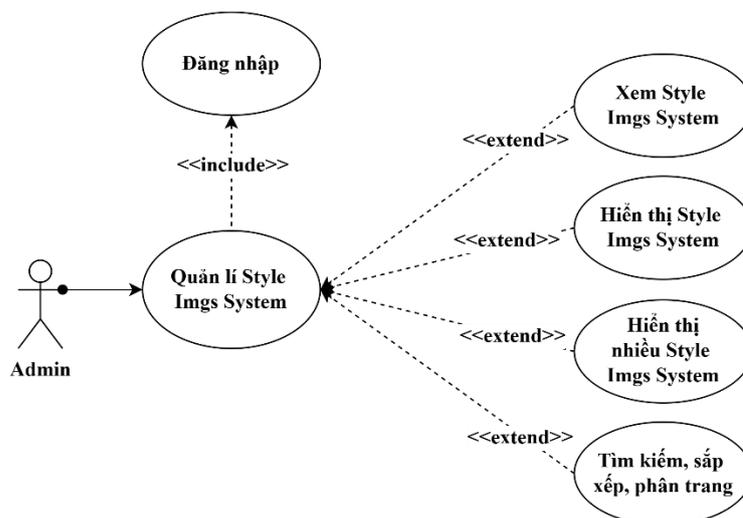
Hình 3.19 Biểu đồ Use case Admin quản lý tác phẩm nghệ thuật của hệ thống



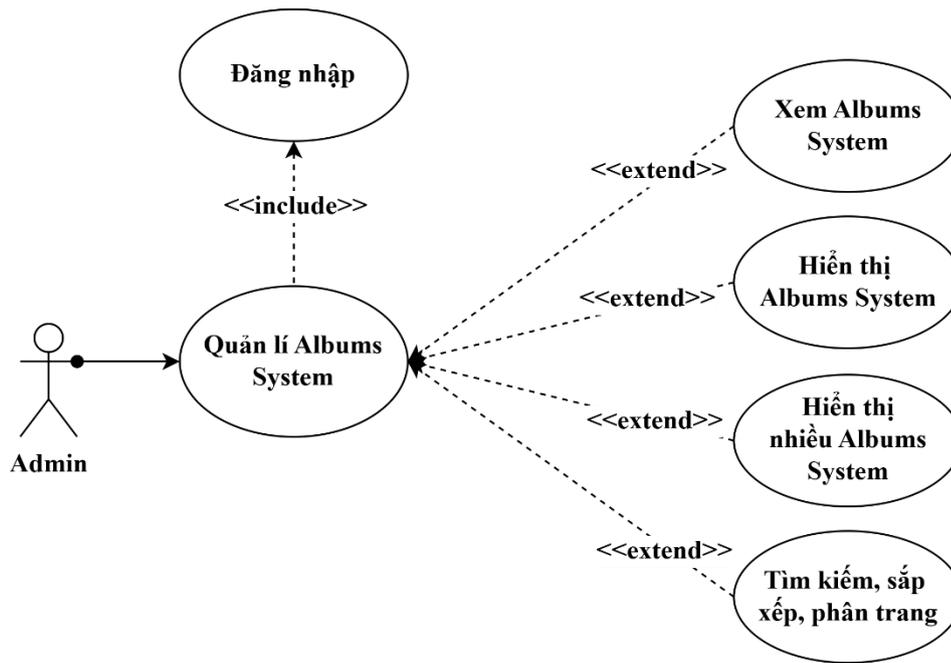
Hình 3.20 Biểu đồ Use case Admin quản lý bài viết của hệ thống



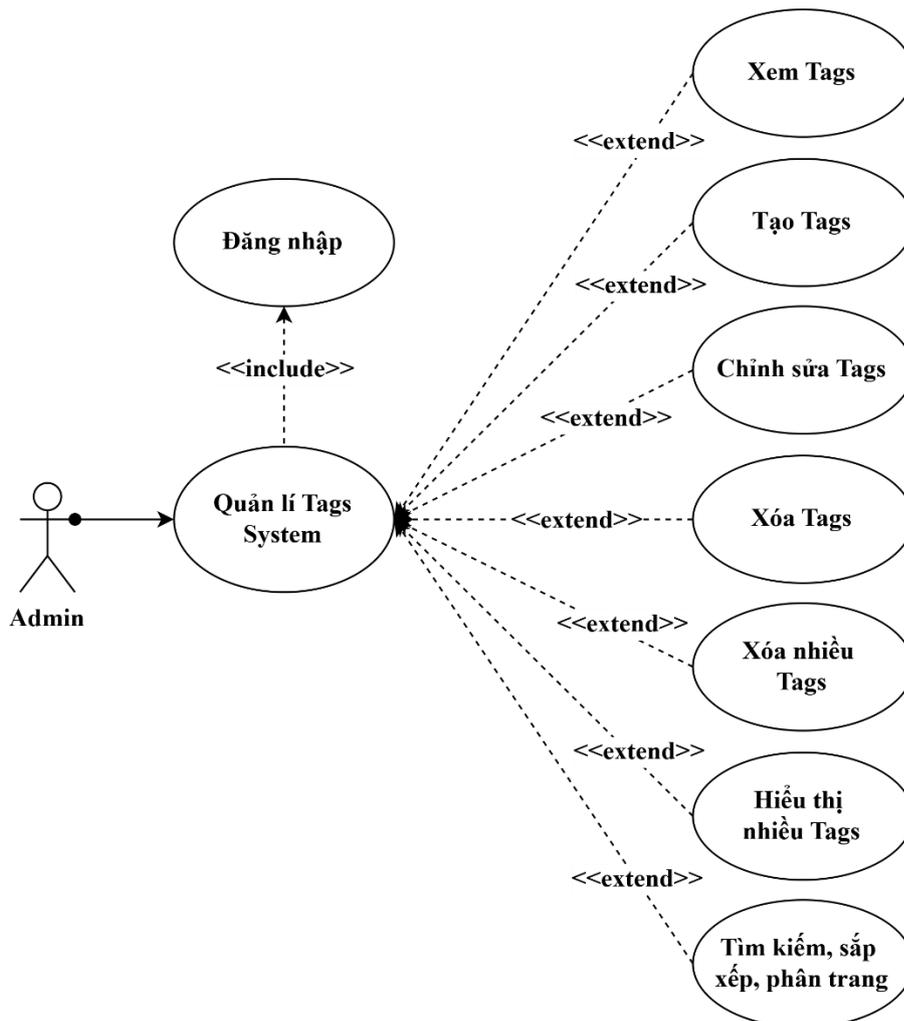
Hình 3.21 Biểu đồ Use case Admin quản lý ảnh nội dung của hệ thống



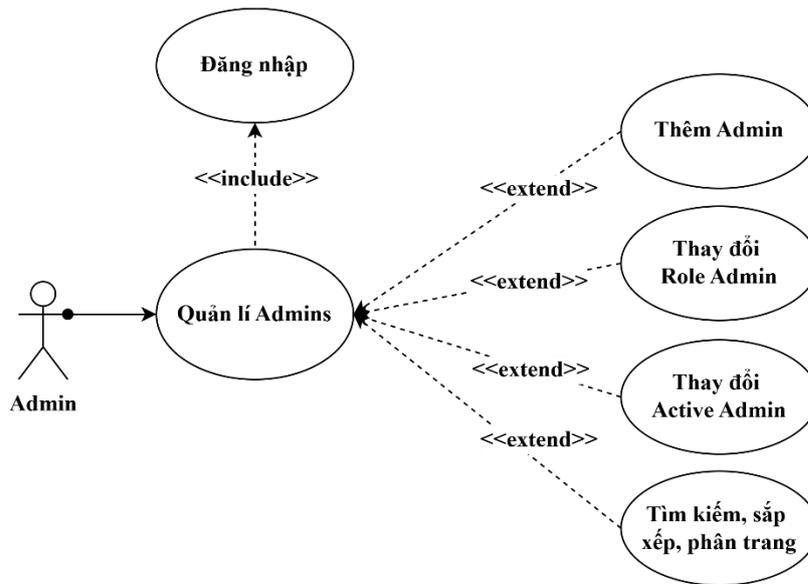
Hình 3.22 Biểu đồ Use case Admin quản lý ảnh phong cách của hệ thống



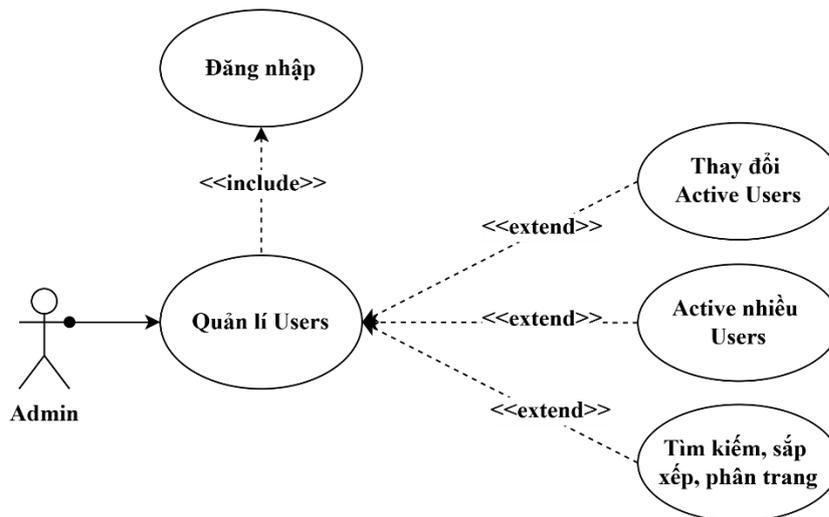
Hình 3.23 Biểu đồ Use case Admin quản lý bộ sưu tập của hệ thống



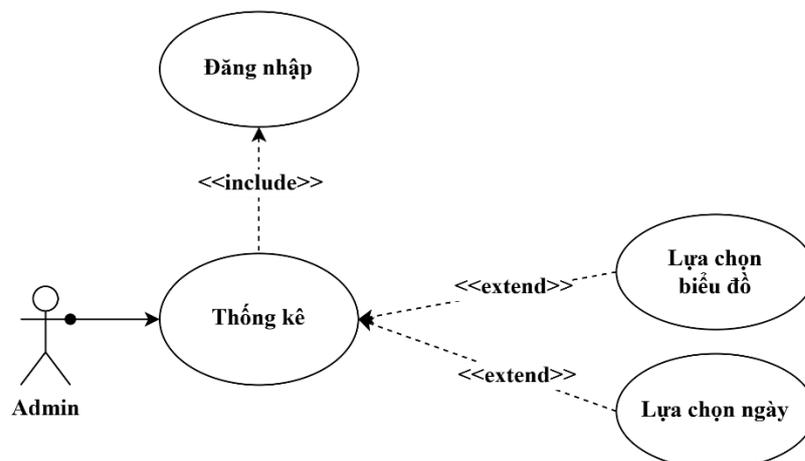
Hình 3.24 Biểu đồ Use case Admin quản lý thẻ của hệ thống



Hình 3.25 Biểu đồ Use case Manager, Super admin quản lý admin



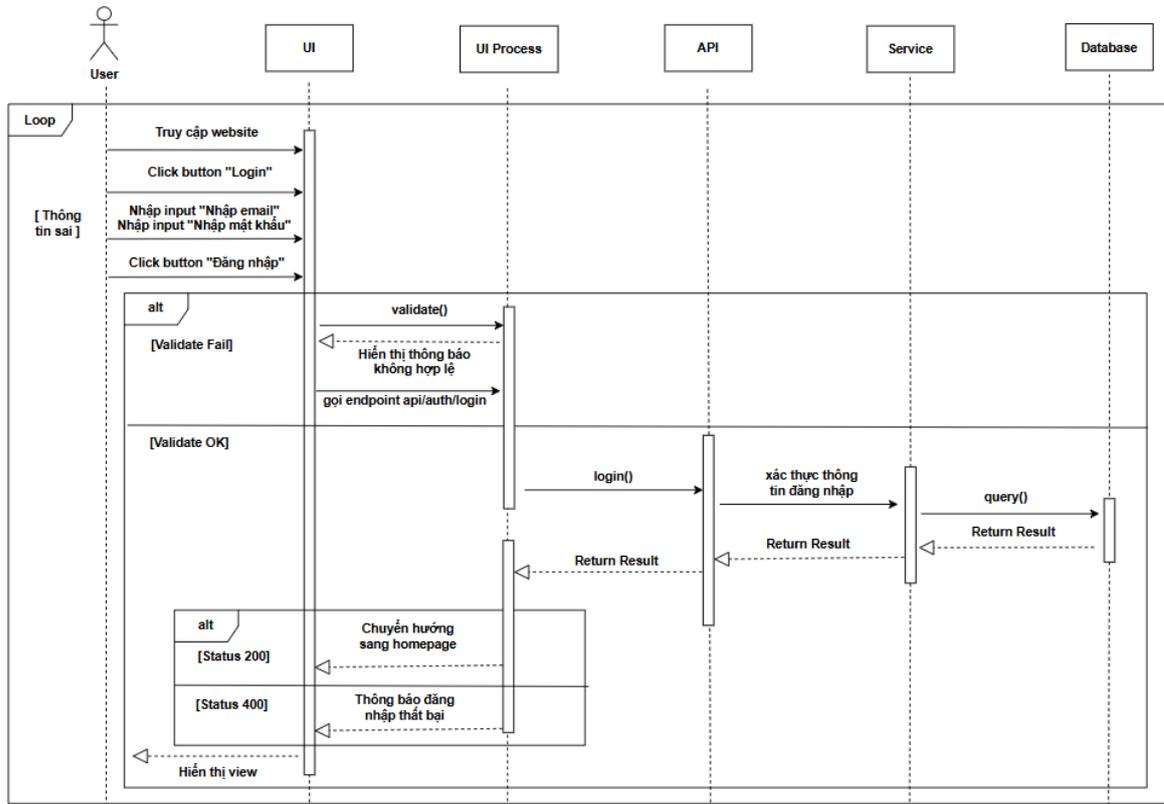
Hình 3.26 Biểu đồ Use case Admin quản lý User



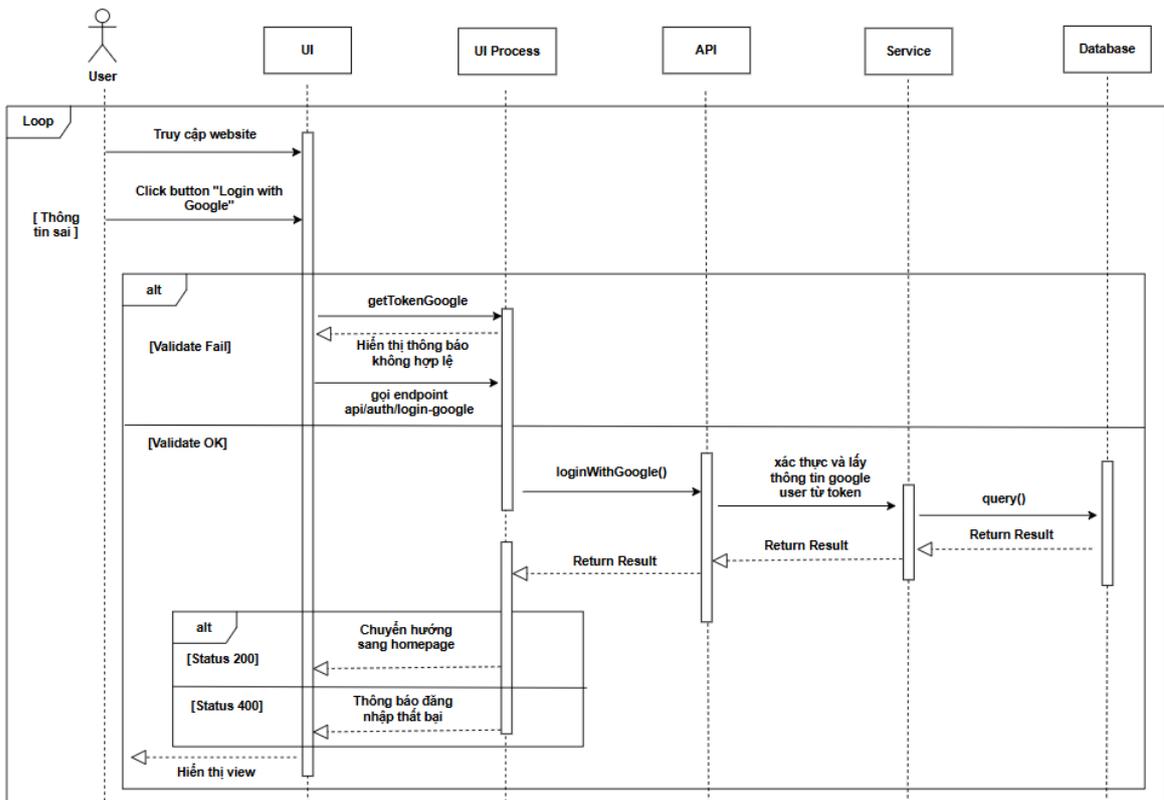
Hình 3.27 Biểu đồ Use case Admin xem thống kê

### 3.6. Sequence Diagrams

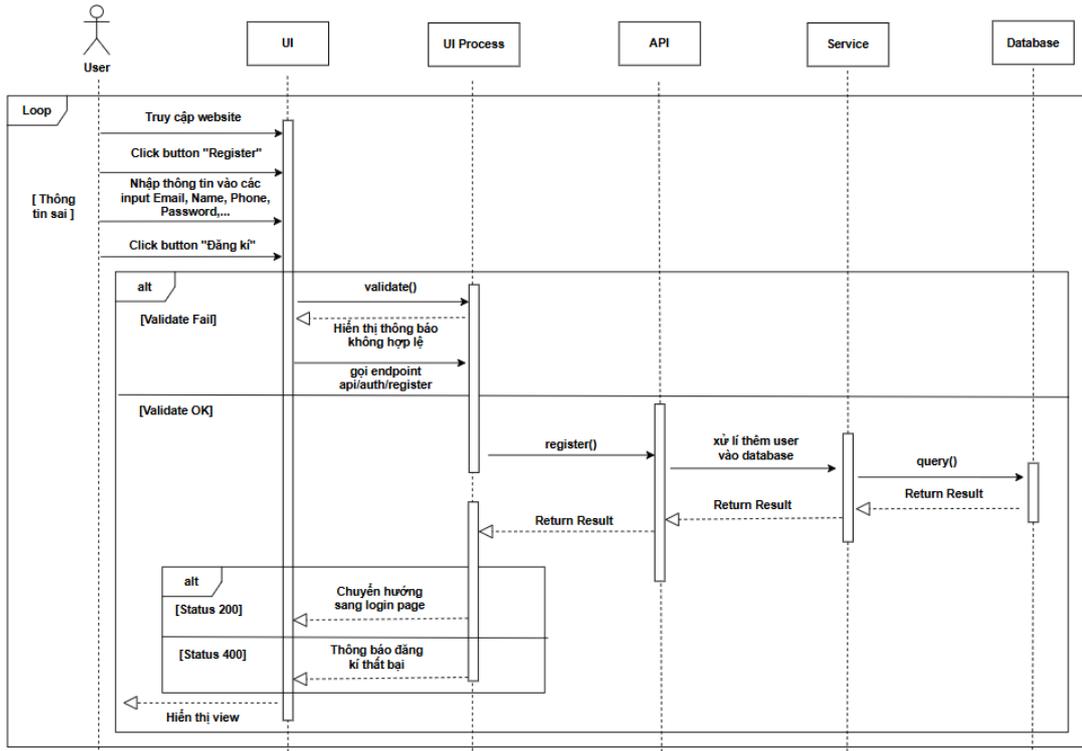
#### 3.6.1. Danh sách biểu đồ tuần tự nhóm chức năng Xác thực



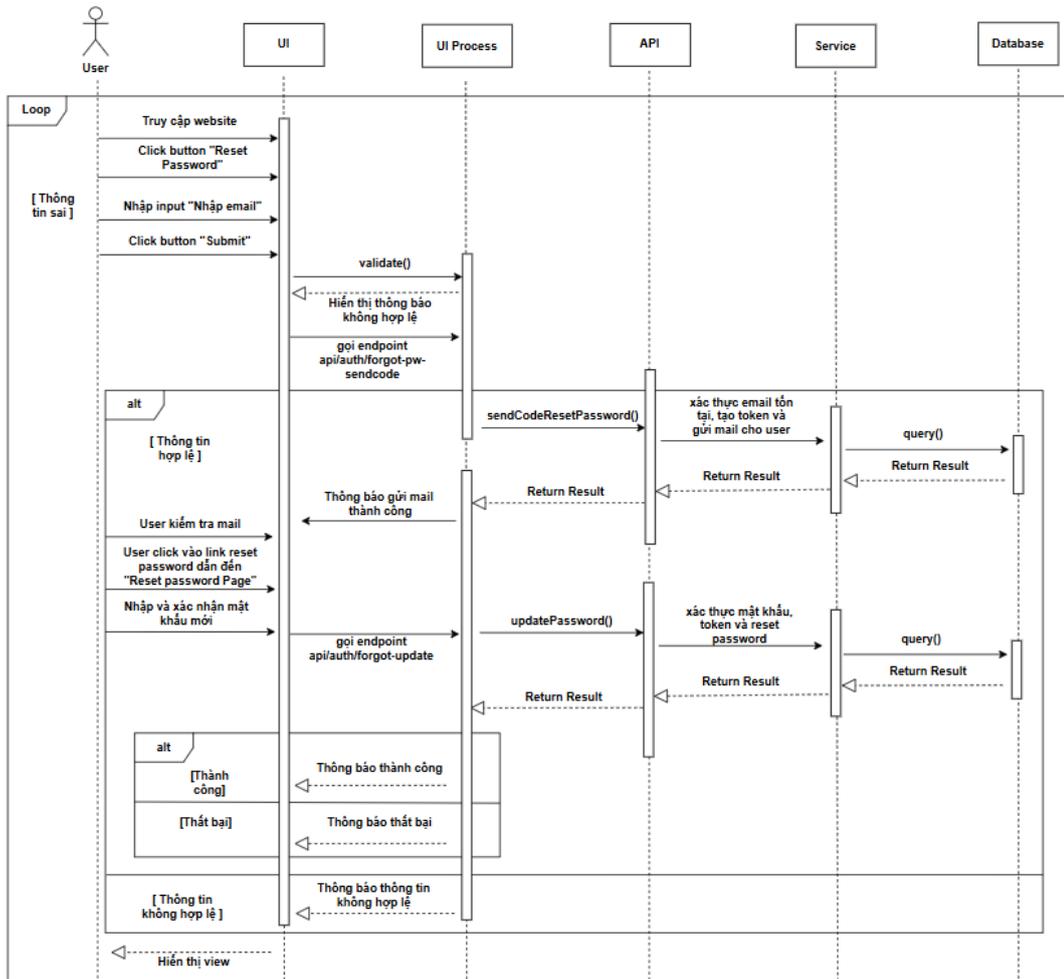
Hình 3.28 Biểu đồ tuần tự chức năng Đăng nhập



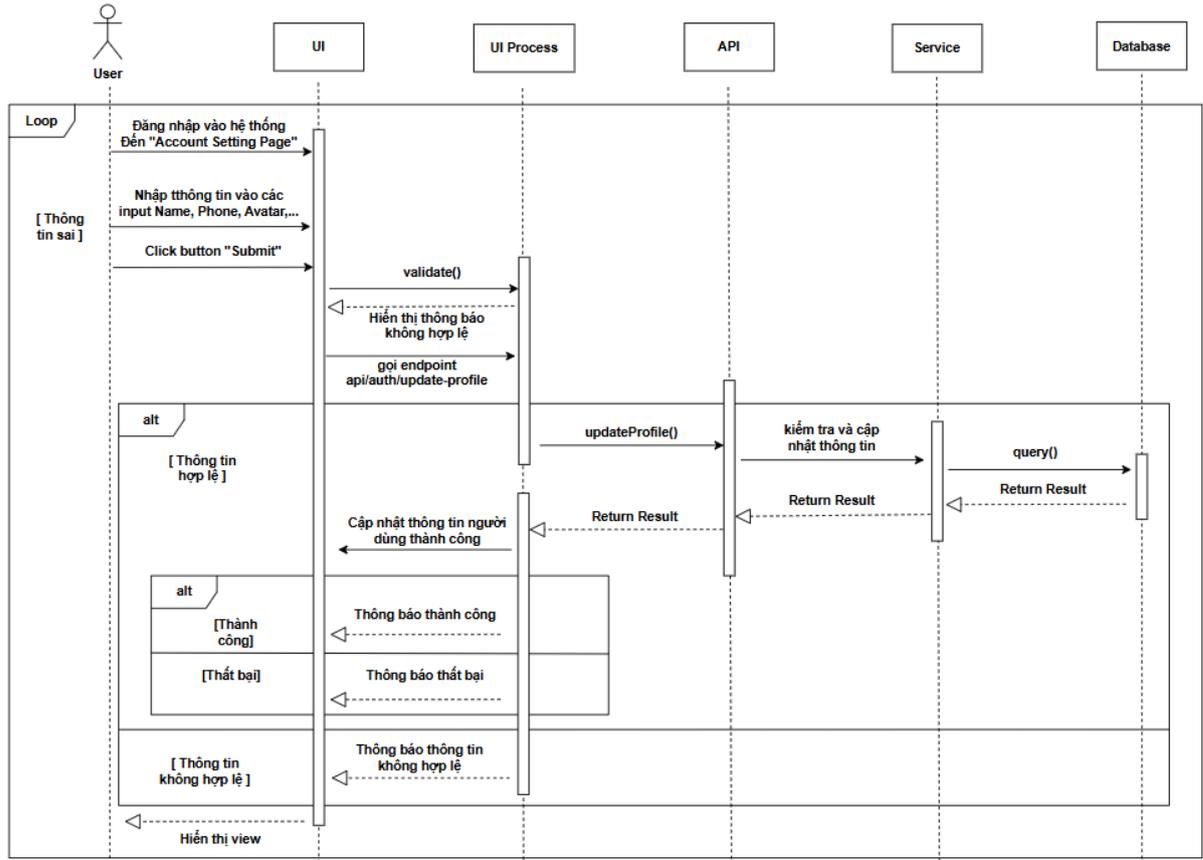
Hình 3.29 Biểu đồ tuần tự chức năng Đăng nhập bằng Google



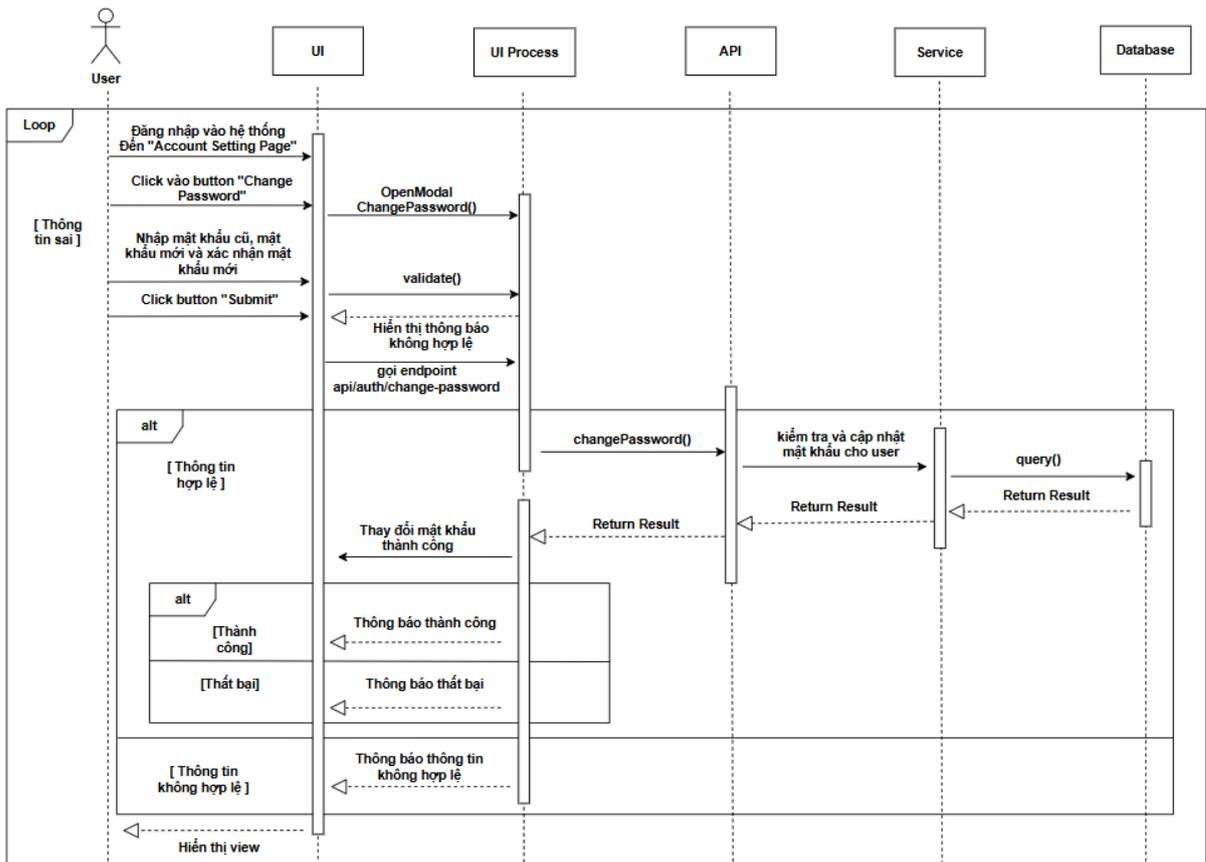
Hình 3.30 Biểu đồ tuần tự chức năng Đăng kí



Hình 3.31 Biểu đồ tuần tự chức năng Đặt lại mật khẩu

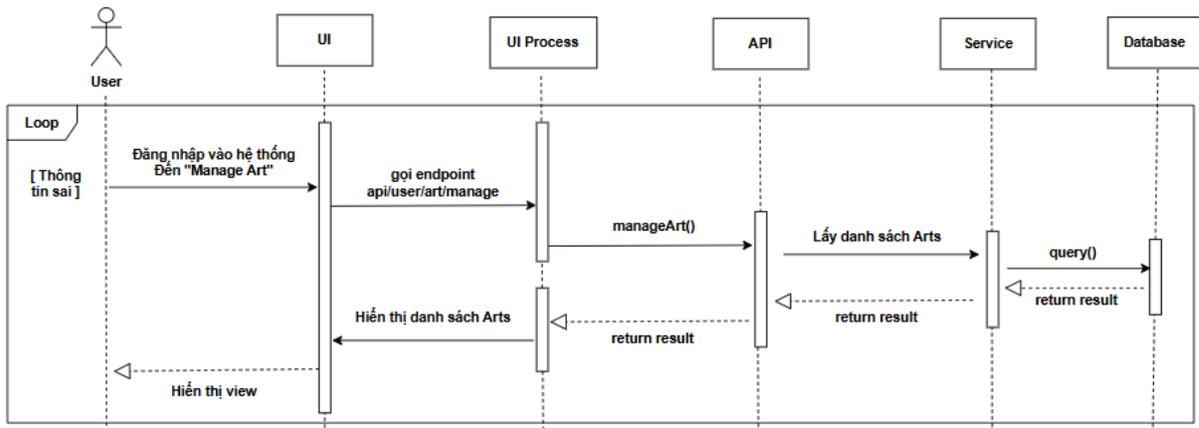


Hình 3.32 Biểu đồ tuần tự chức năng Cài đặt và cập nhật tài khoản

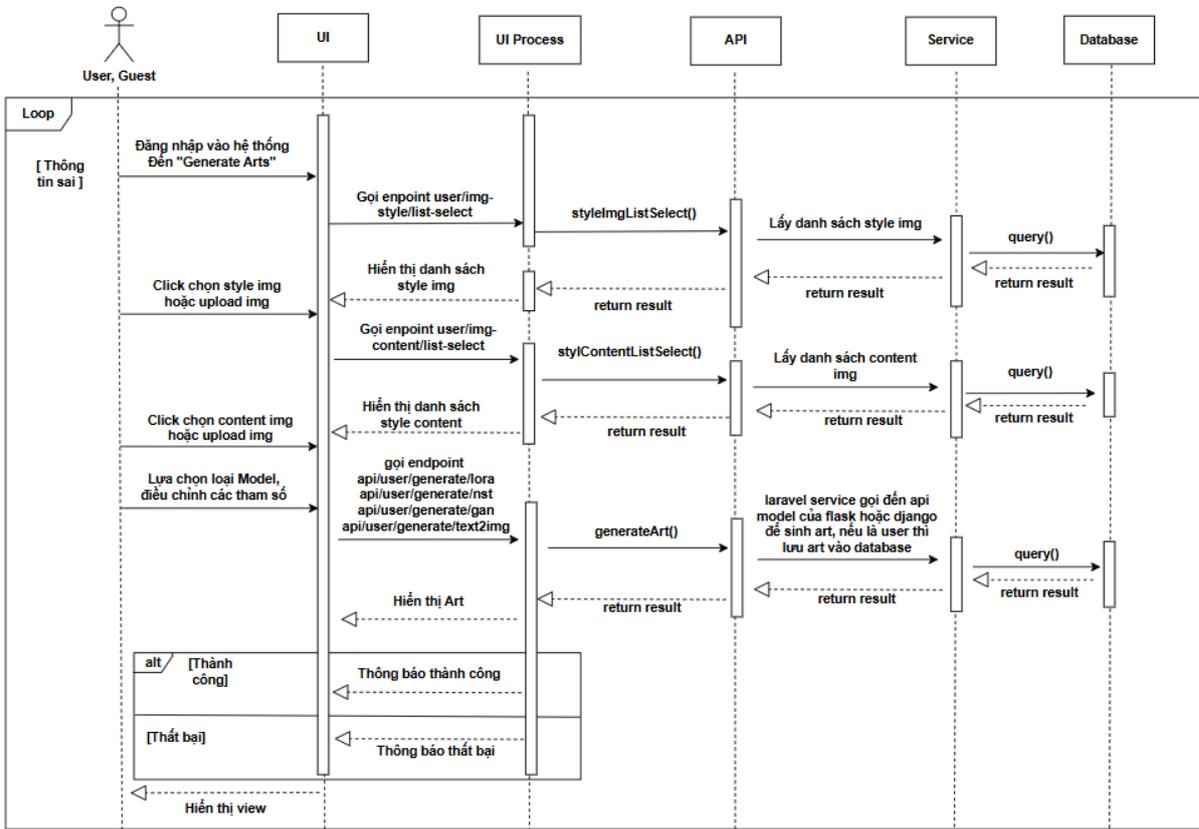


Hình 3.33 Biểu đồ tuần tự chức năng Thay đổi mật khẩu

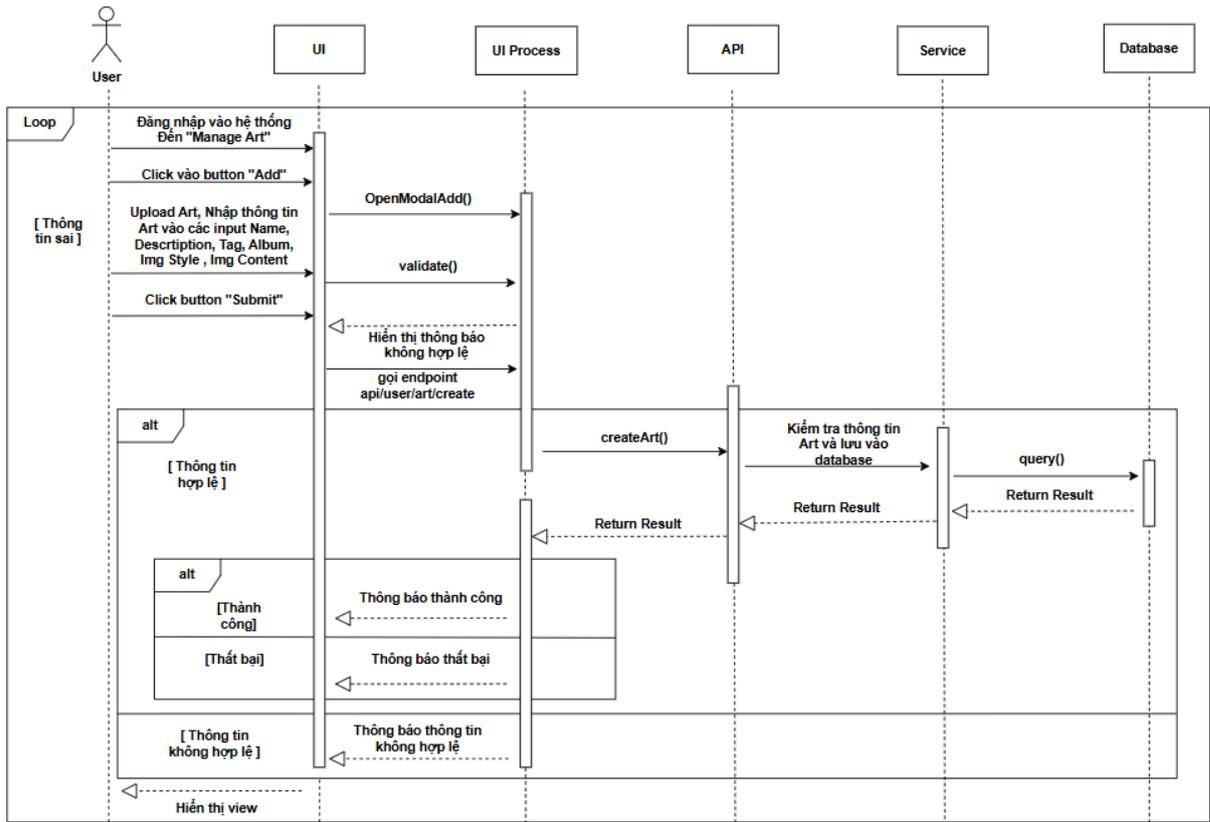
### 3.6.2. Danh sách biểu đồ tuần tự chức năng Quản lý nghệ thuật



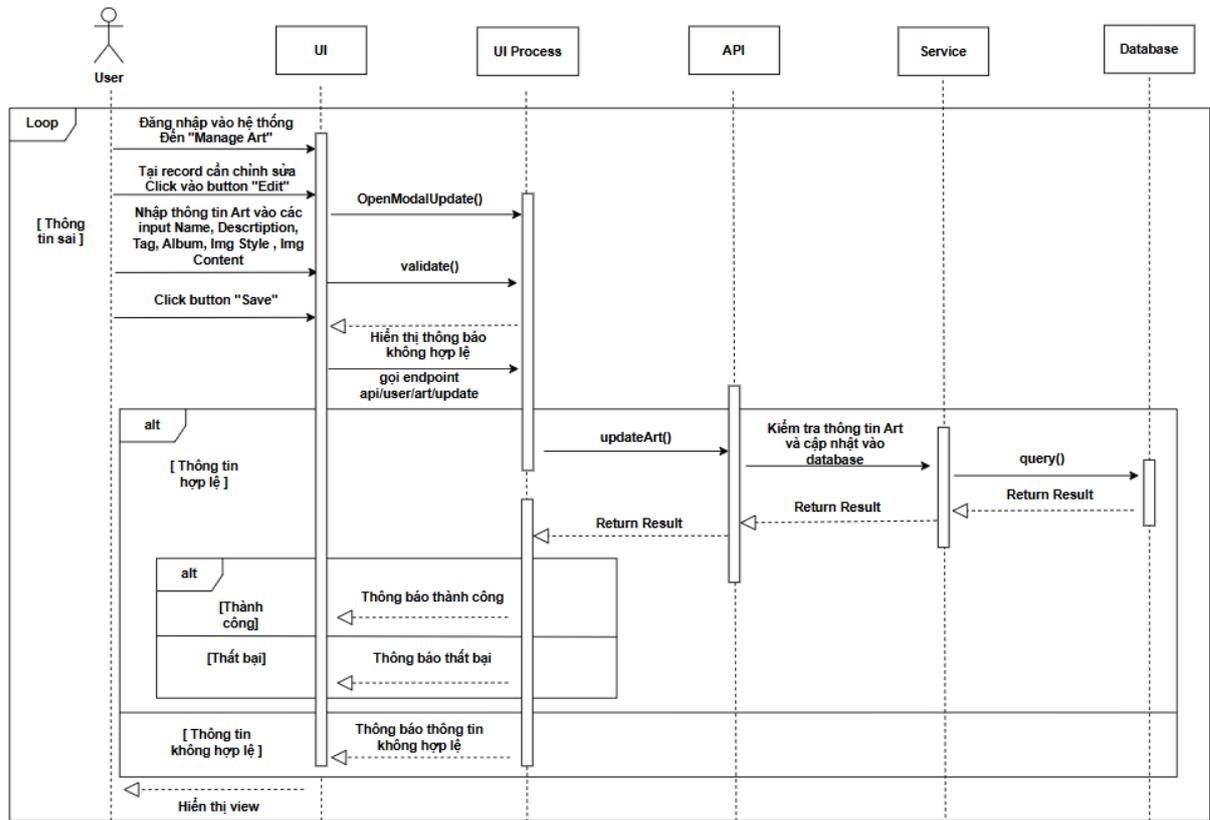
Hình 3.34 Biểu đồ tuần tự chức năng Quản lý tác phẩm nghệ thuật



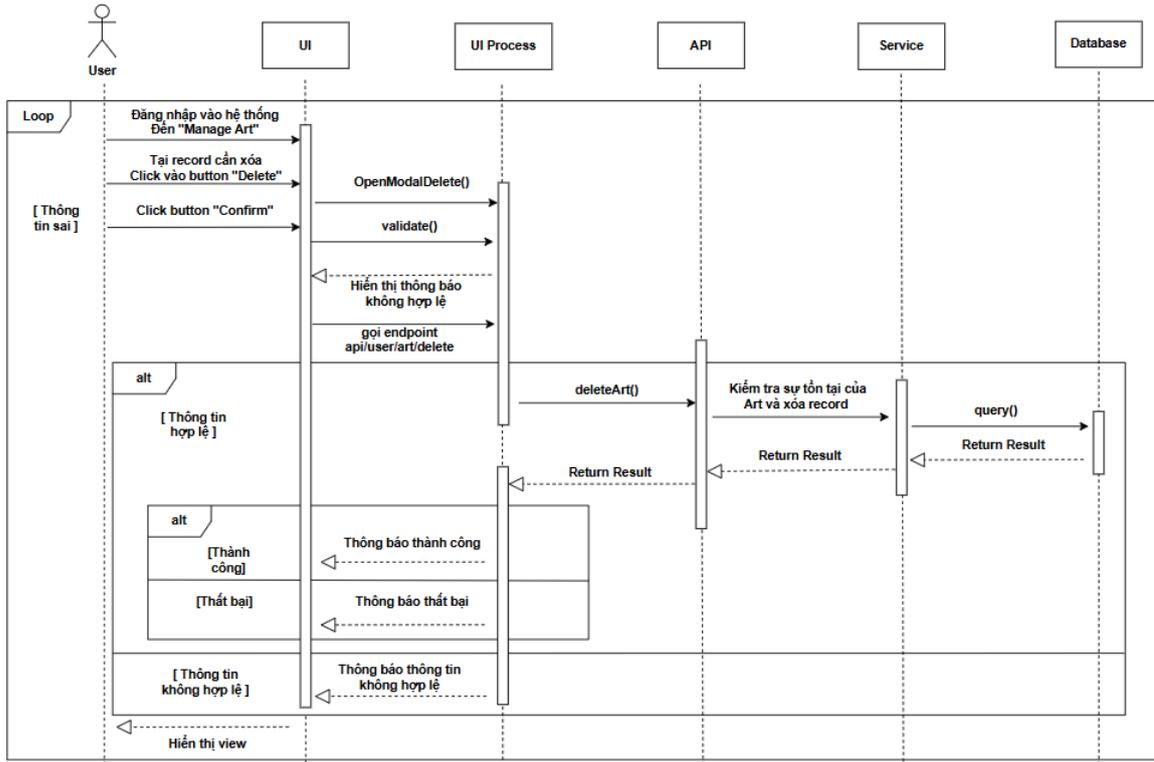
Hình 3.35 Biểu đồ tuần tự chức năng Tạo sinh ảnh nghệ thuật



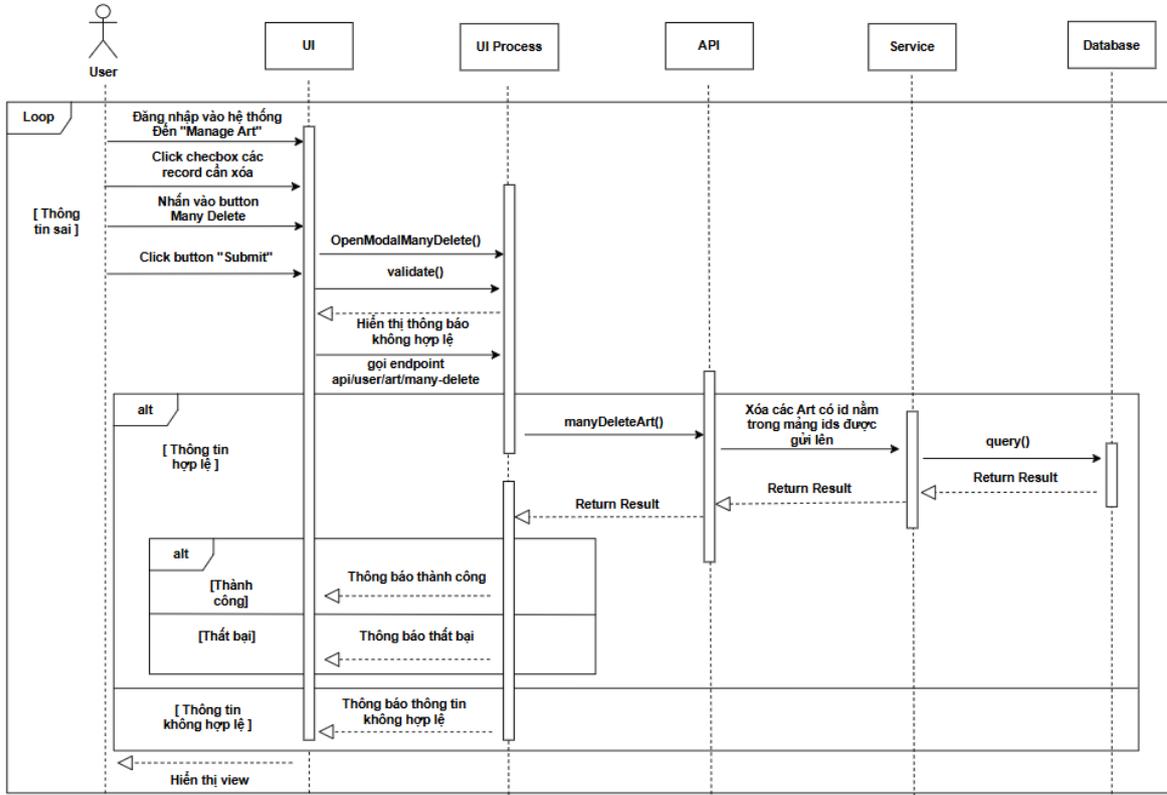
Hình 3.36 Biểu đồ tuần tự chức năng Thêm tác phẩm nghệ thuật



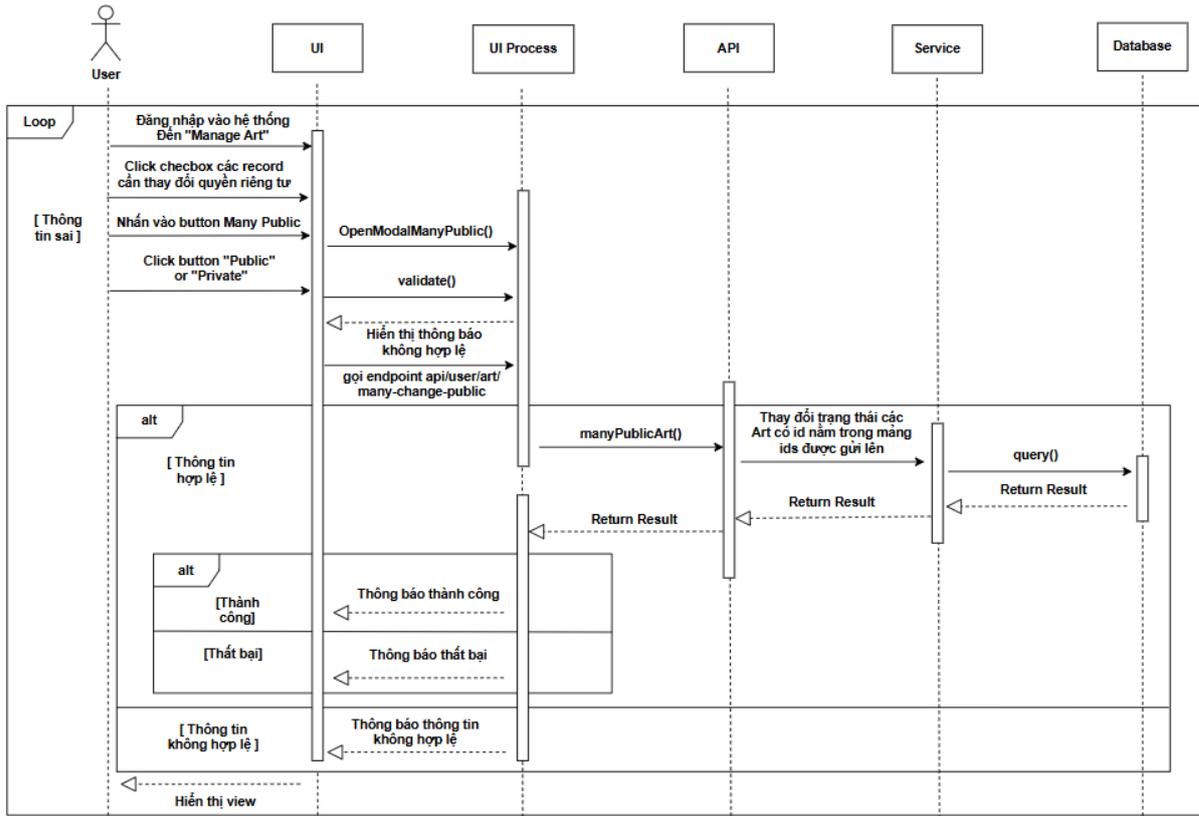
Hình 3.37 Biểu đồ tuần tự chức năng Chỉnh sửa tác phẩm nghệ thuật



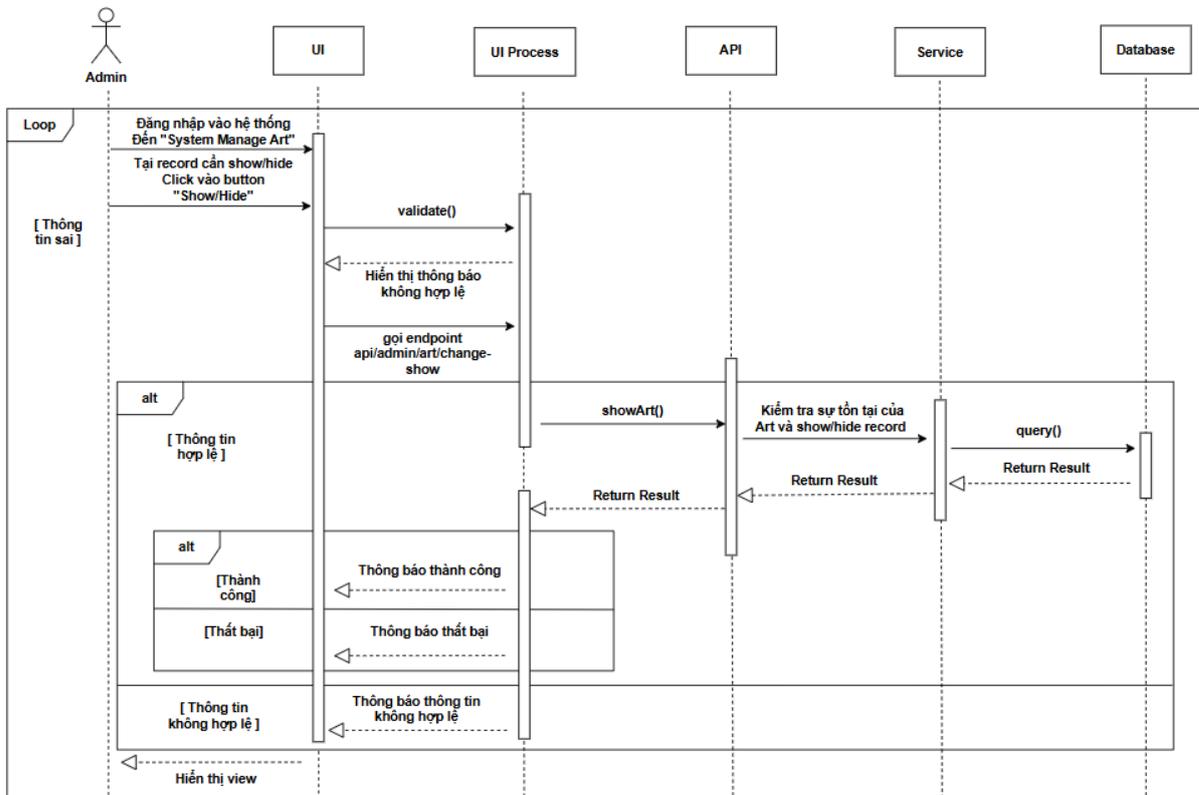
Hình 3.38 Biểu đồ tuần tự chức năng Xóa tác phẩm nghệ thuật



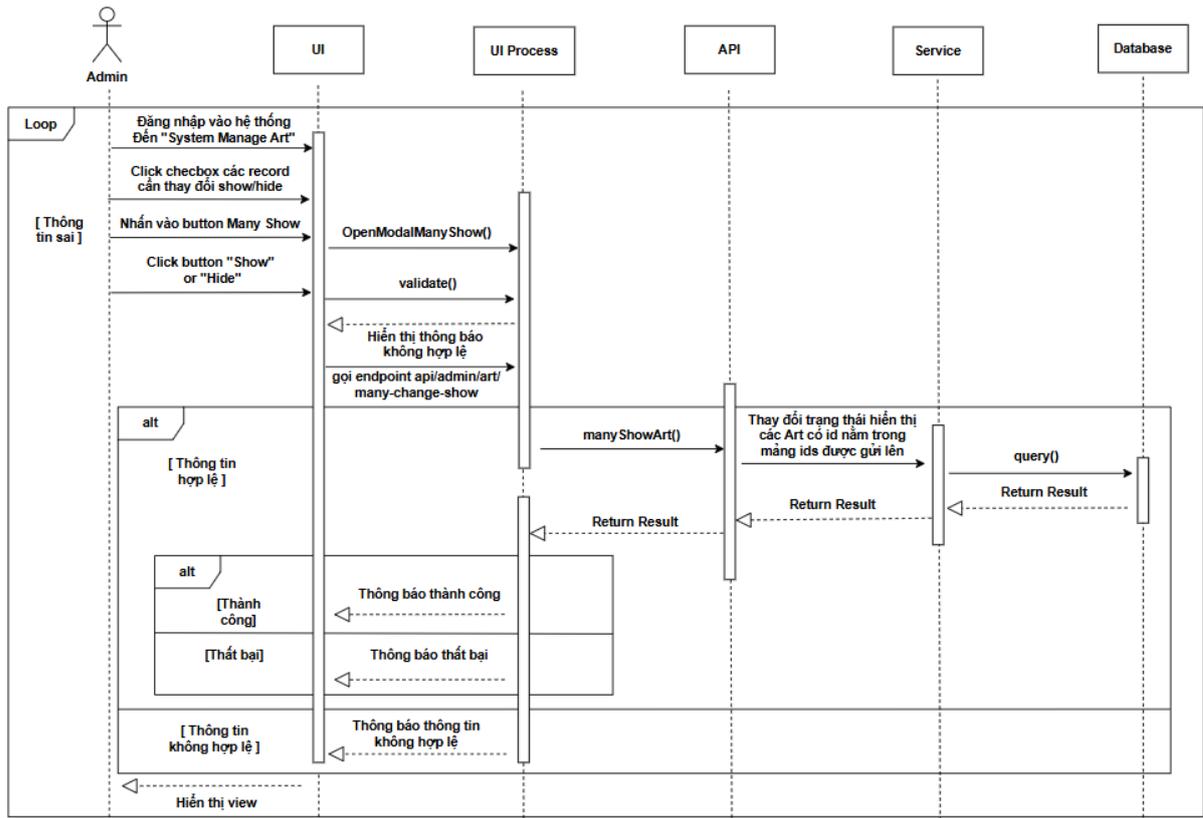
Hình 3.39 Biểu đồ tuần tự chức năng Xóa hàng loạt tác phẩm nghệ thuật



Hình 3.40 Biểu đồ tuần tự Thay đổi công khai nhiều ảnh nghệ thuật

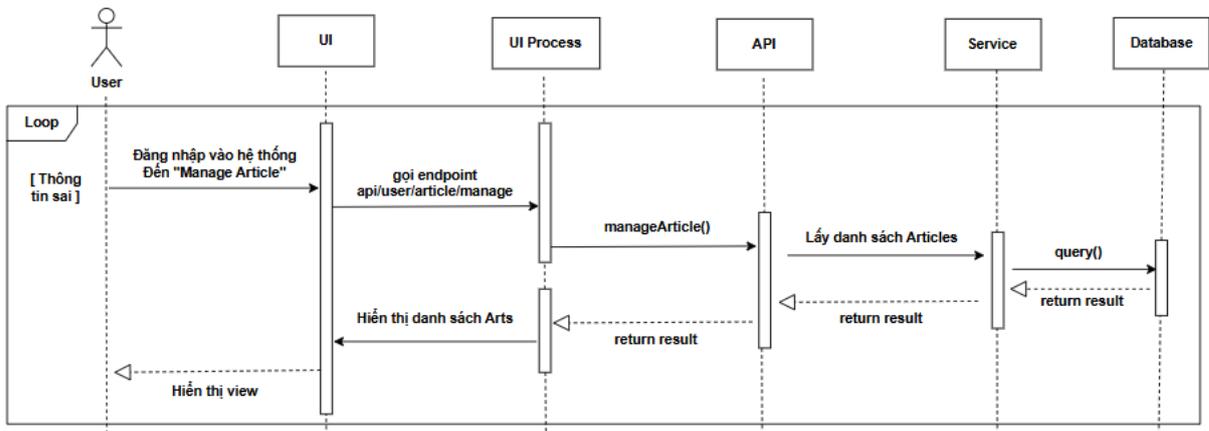


Hình 3.41 Biểu đồ tuần tự Thay đổi hiển thị nhiều ảnh nghệ thuật

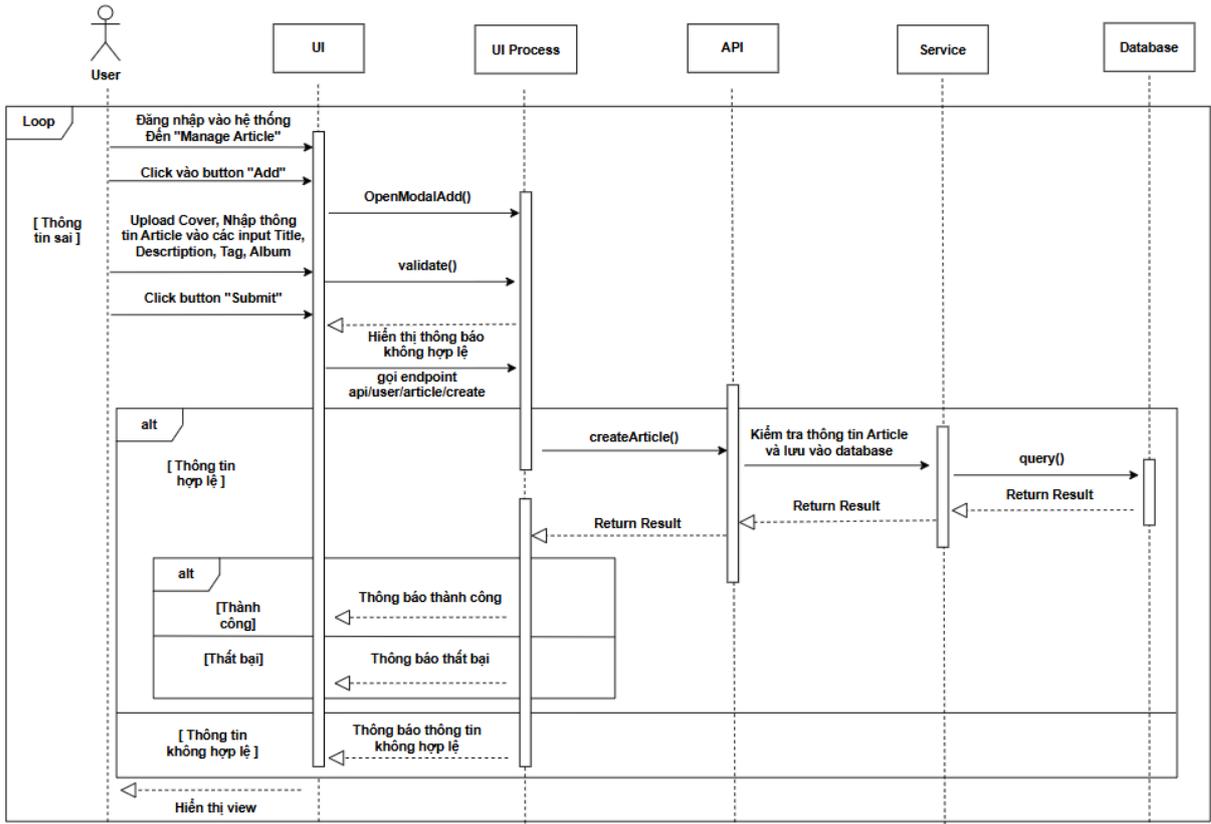


Hình 3.42 Biểu đồ tuần tự Thay đổi hiển thị nhiều ảnh nghệ thuật

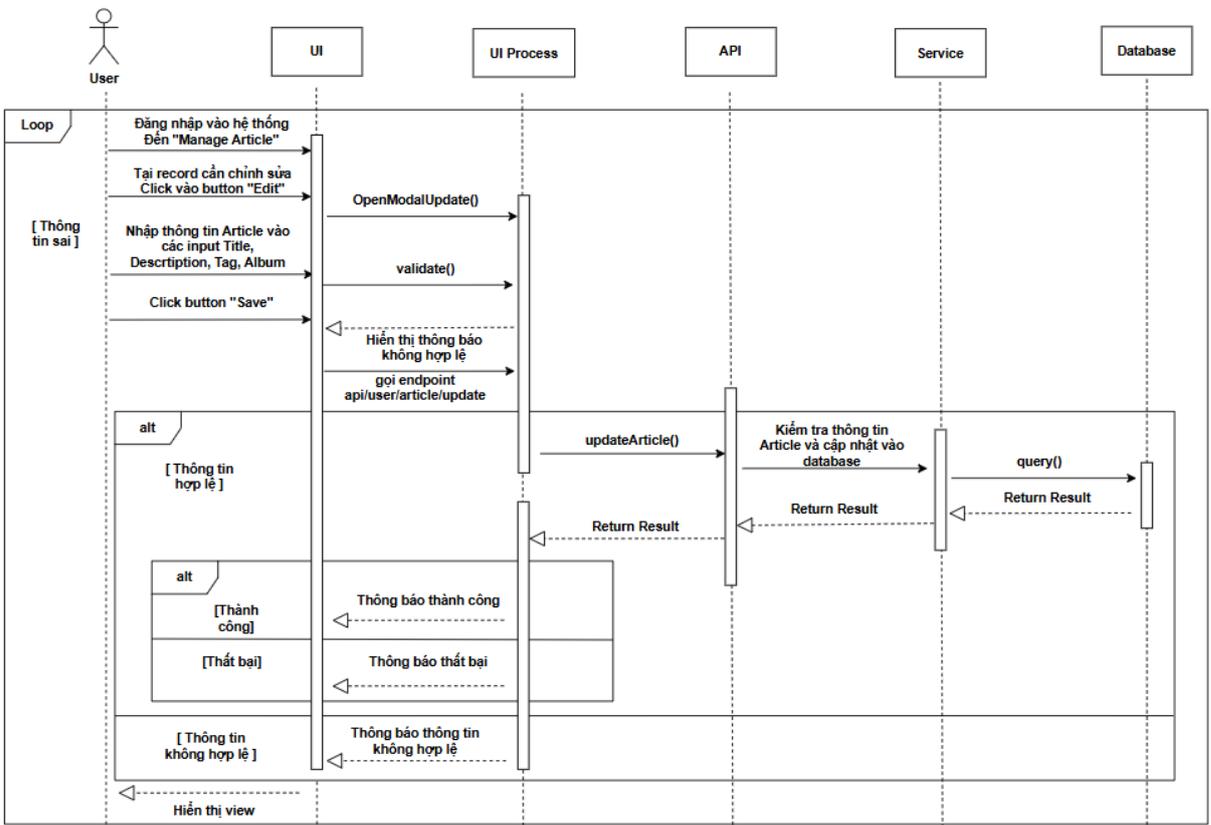
### 3.6.3. Danh sách biểu đồ tuần tự nhóm chức năng Quản lý bài viết



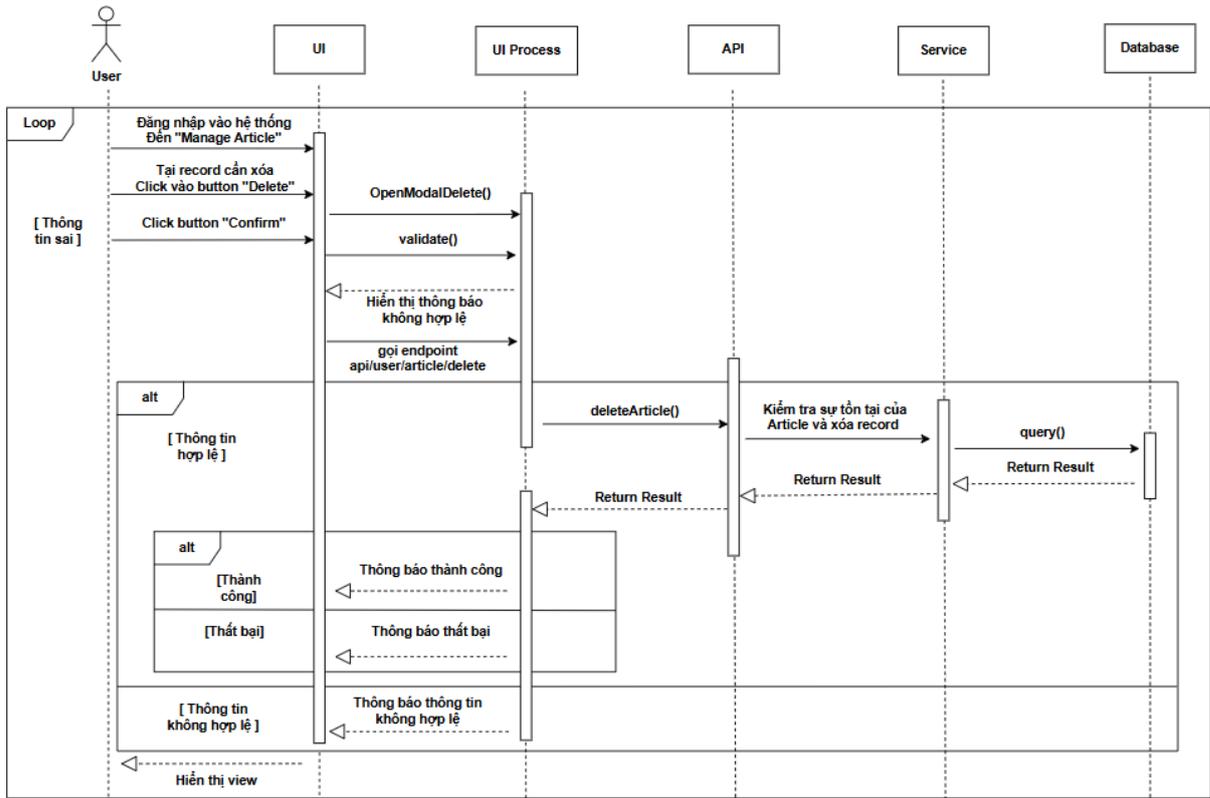
Hình 3.43 Biểu đồ tuần tự chức năng Quản lý bài viết



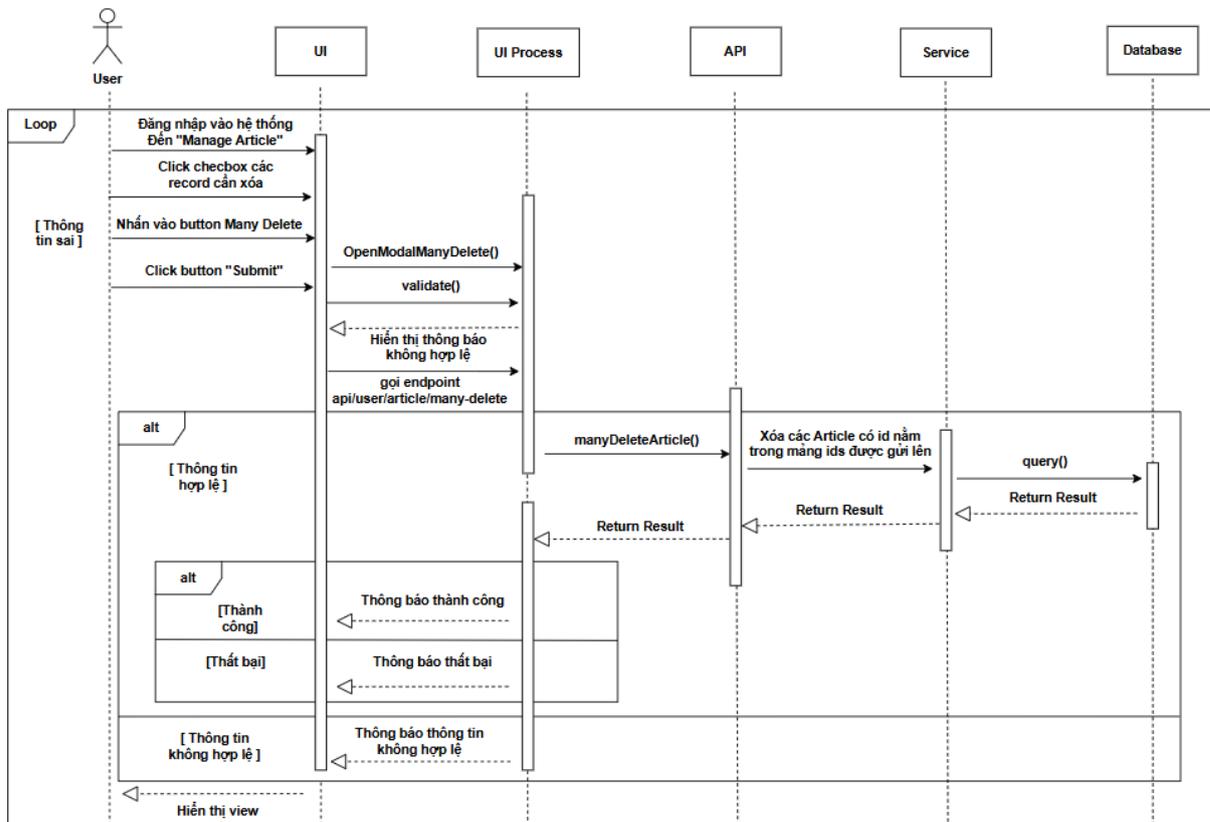
Hình 3.44 Biểu đồ tuần tự chức năng Thêm bài viết



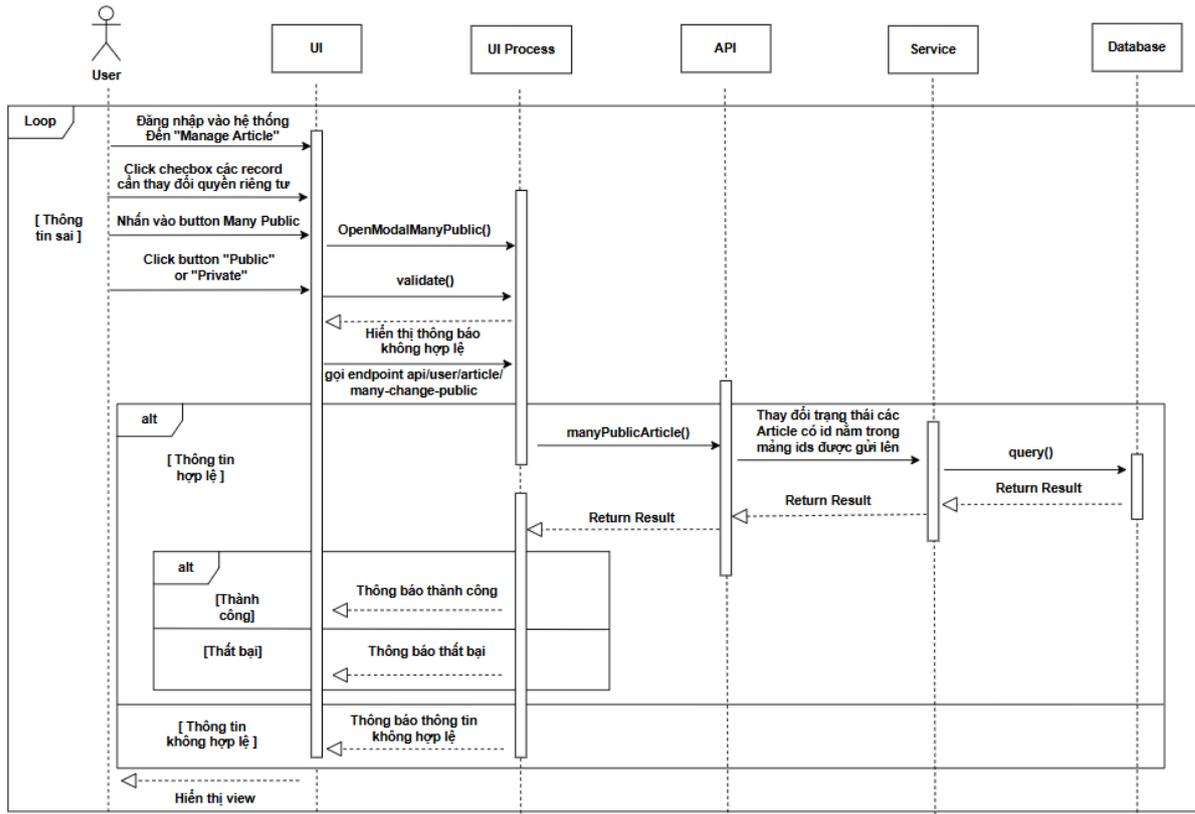
Hình 3.45 Biểu đồ tuần tự chức năng Chỉnh sửa bài viết



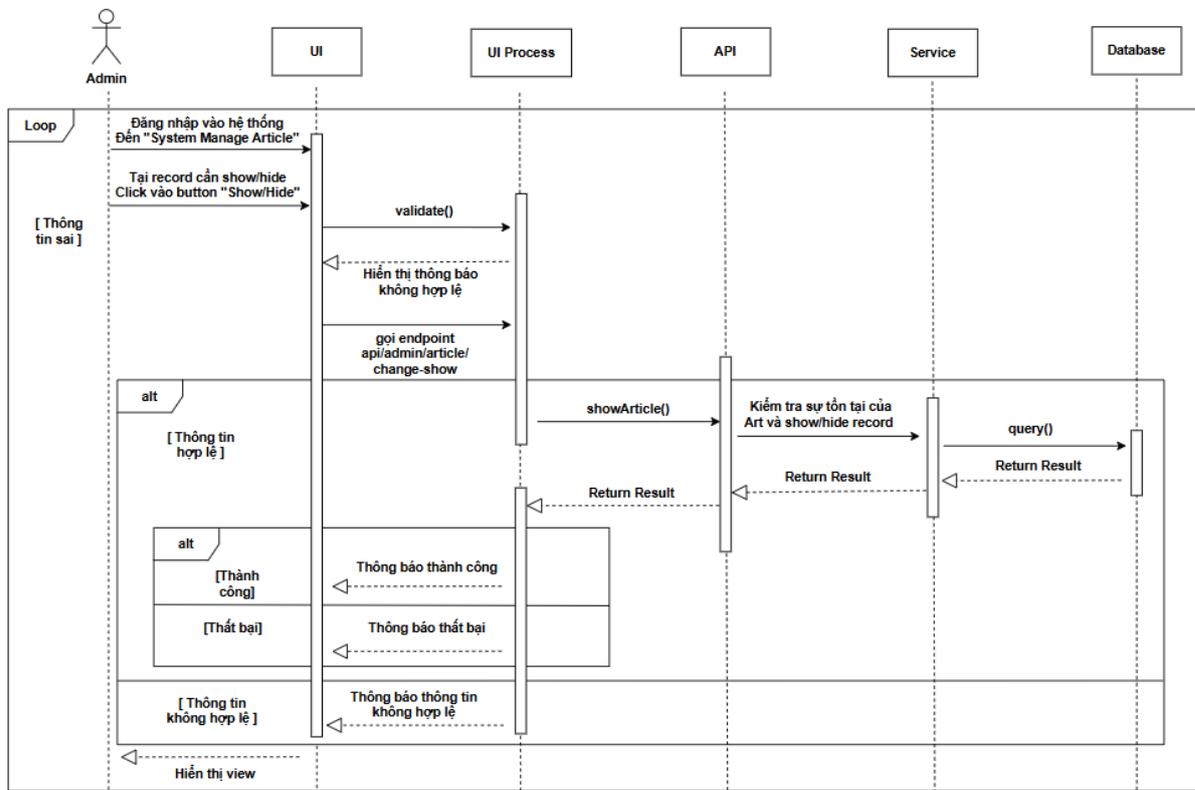
Hình 3.46 Biểu đồ tuần tự chức năng Xóa bài viết



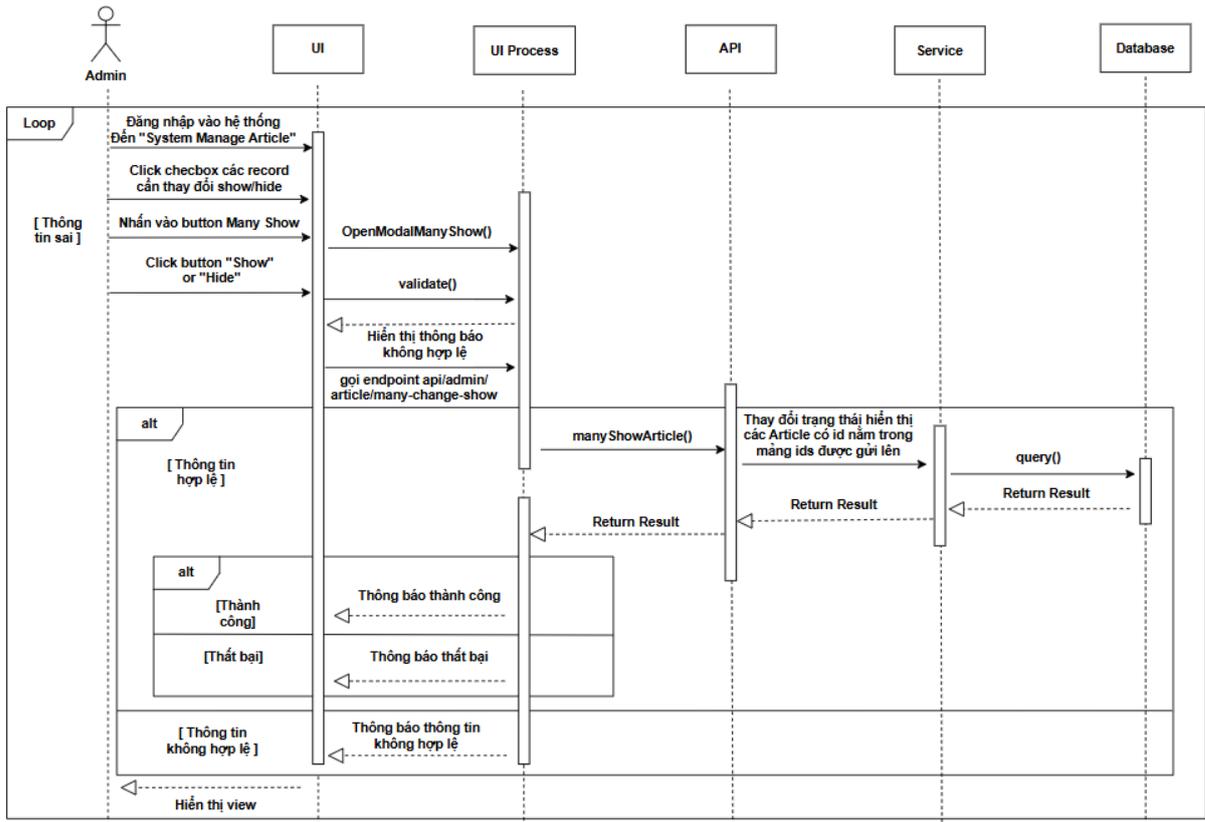
Hình 3.47 Biểu đồ tuần tự chức năng Xóa hàng loạt bài viết



Hình 3.48 Biểu đồ tuần tự chức năng Thay đổi công khai nhiều bài viết



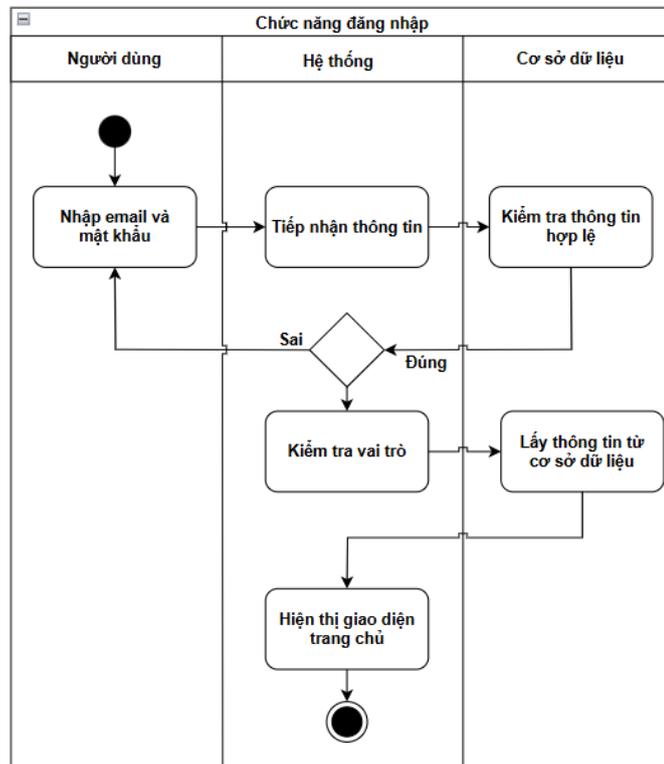
Hình 3.49 Biểu đồ tuần tự chức năng Thay đổi trạng thái hiển thị bài viết



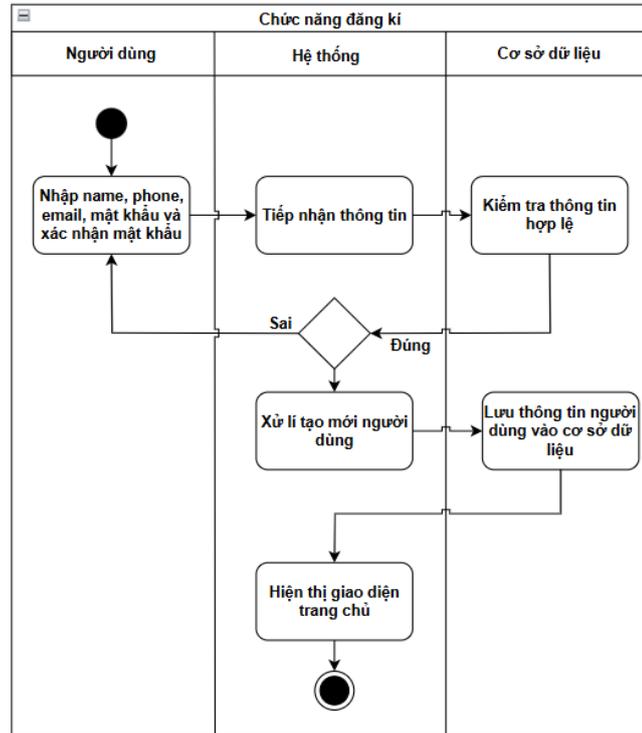
Hình 3.50 Biểu đồ tuần tự chức năng Thay đổi hiển thị nhiều bài viết

### 3.7. Activity Diagrams

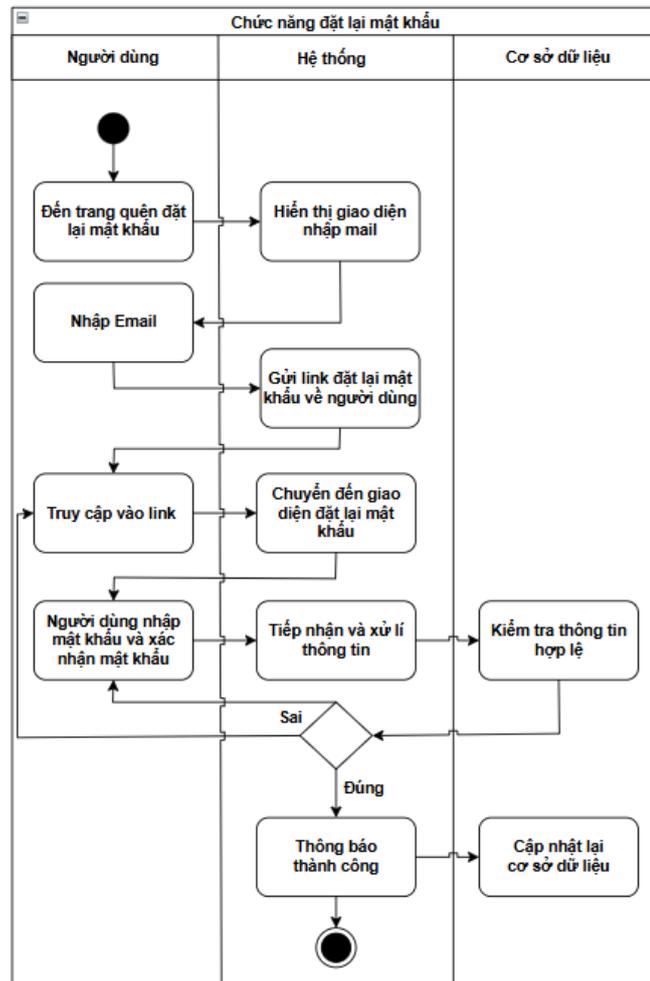
#### 3.7.1. Danh sách biểu đồ hoạt động nhóm chức năng Xác thực



Hình 3.51 Biểu đồ hoạt động chức năng Đăng nhập

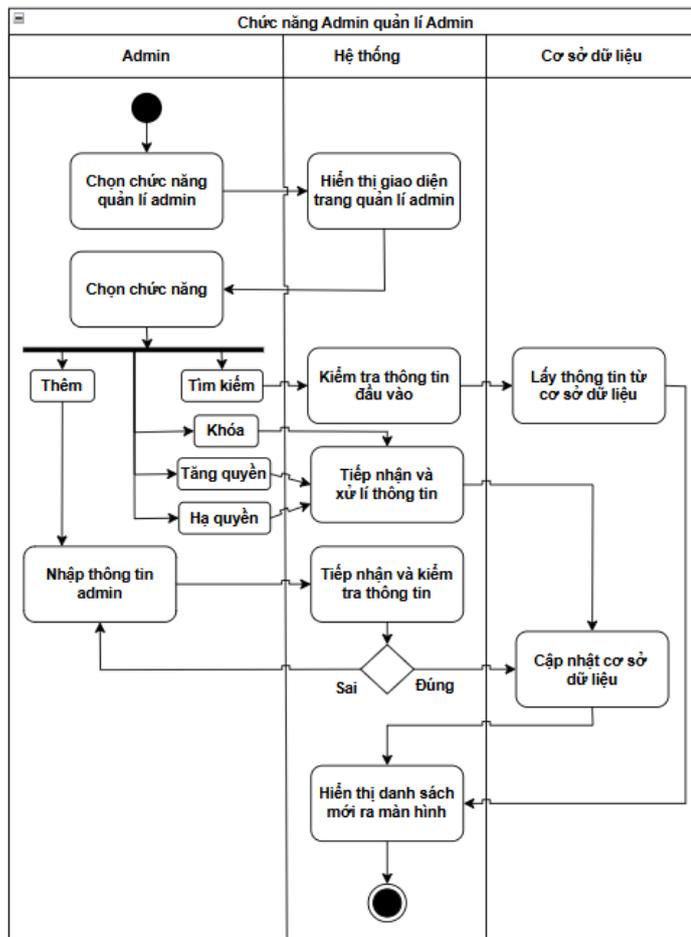


Hình 3.52 Biểu đồ hoạt động chức năng Đăng kí

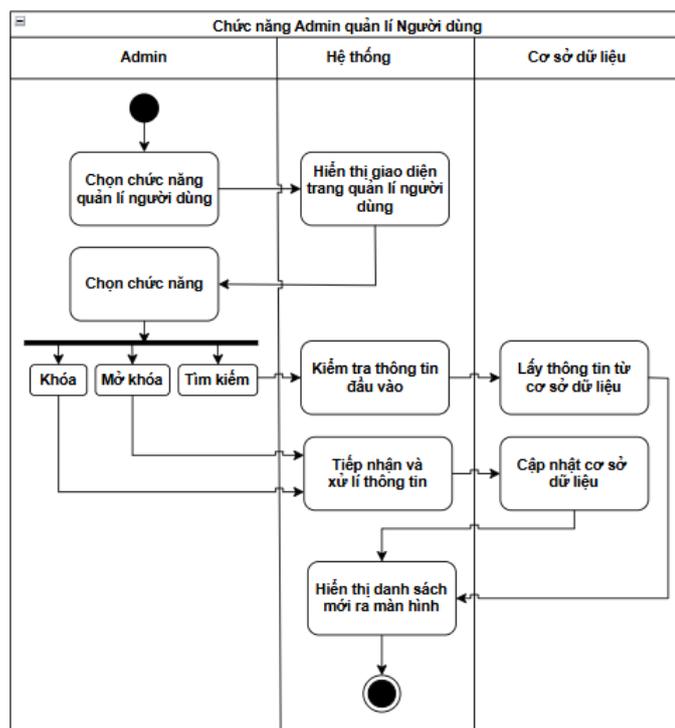


Hình 3.53 Biểu đồ hoạt động chức năng Đặt lại mật khẩu

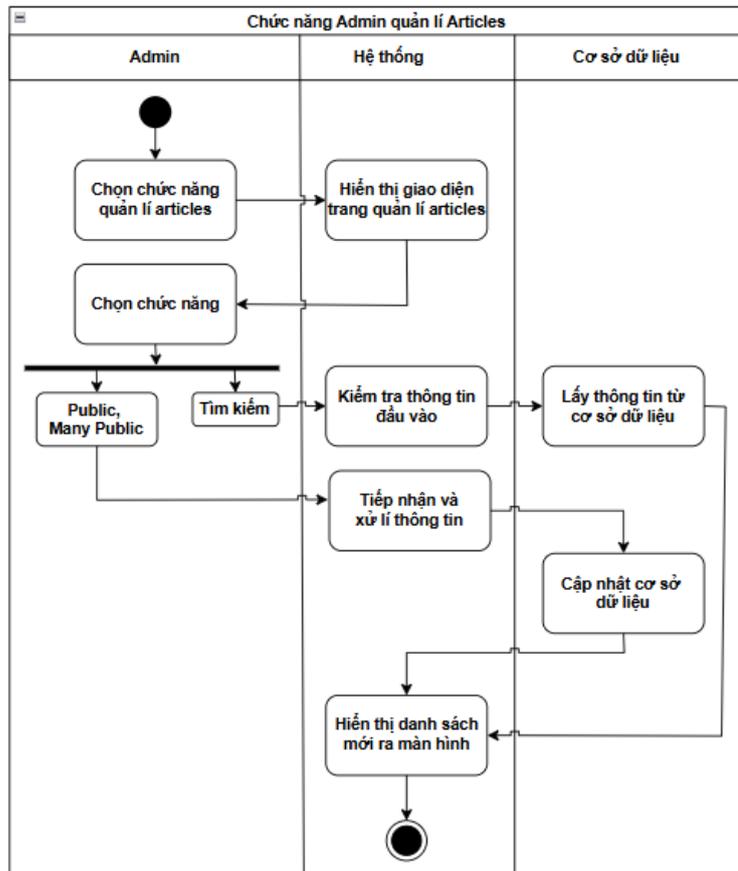
### 3.7.2. Danh sách biểu đồ hoạt động nhóm chức năng của Admin



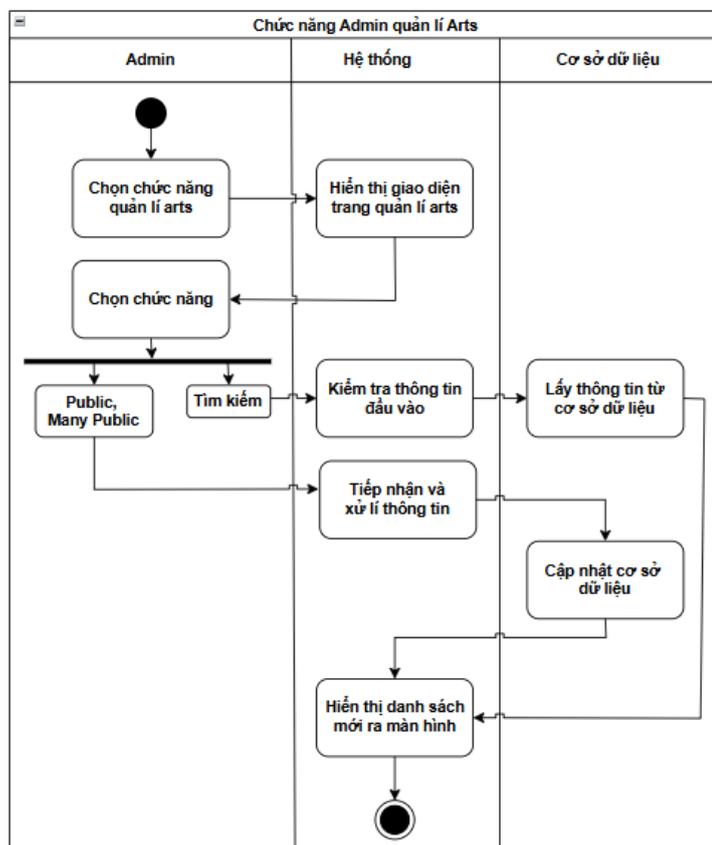
Hình 3.54 Biểu đồ hoạt động chức năng Manager quản lý Admin



Hình 3.55 Biểu đồ hoạt động chức năng Admin quản lý người dùng

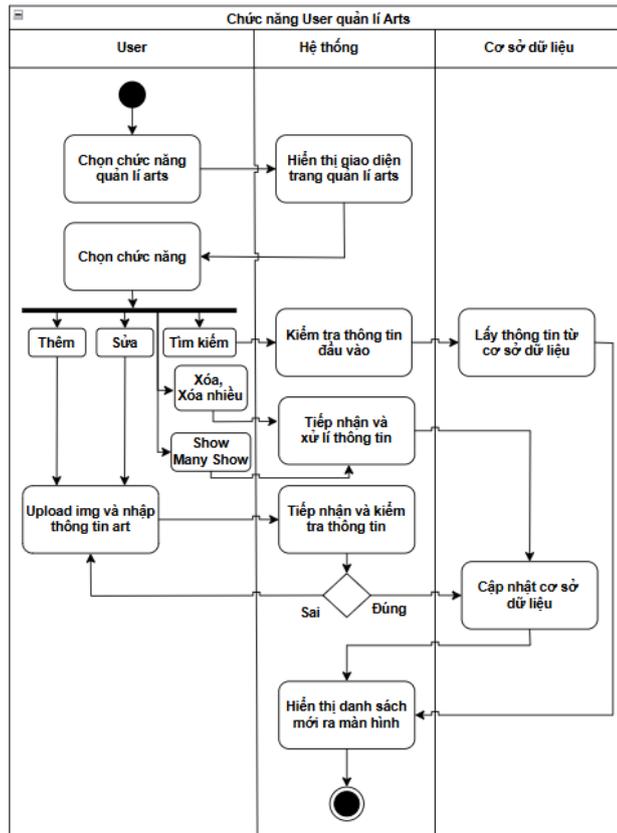


Hình 3.56 Biểu đồ hoạt động chức năng Admin quản lý bài viết hệ thống

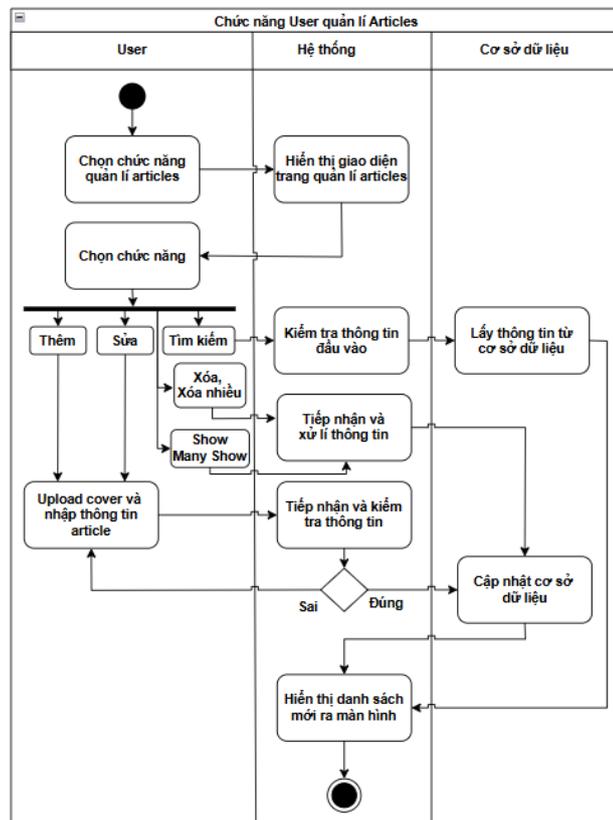


Hình 3.57 Biểu đồ hoạt động Admin quản lý ảnh nghệ thuật hệ thống

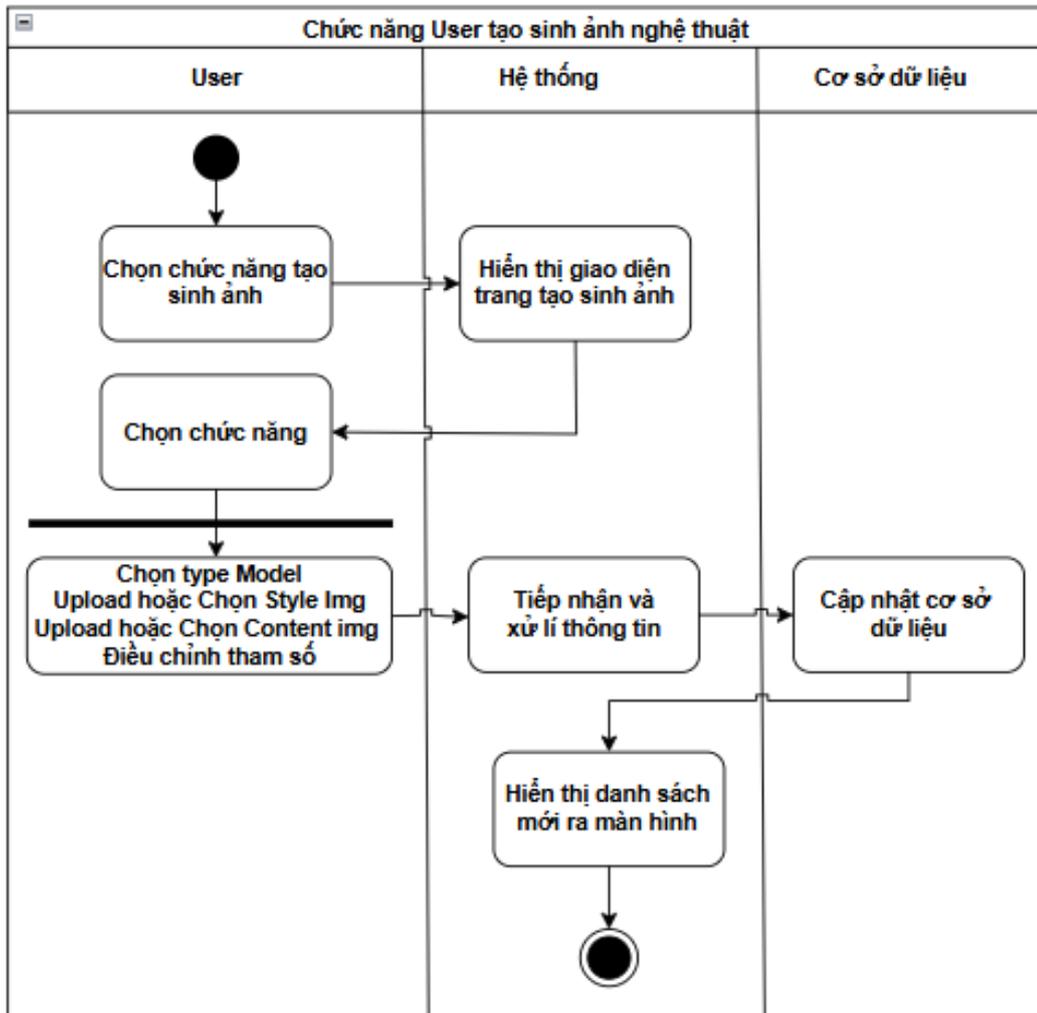
### 3.7.3. Danh sách biểu đồ hoạt động nhóm chức năng của User



Hình 3.58 Biểu đồ hoạt động chức năng User quản lý ảnh nghệ thuật



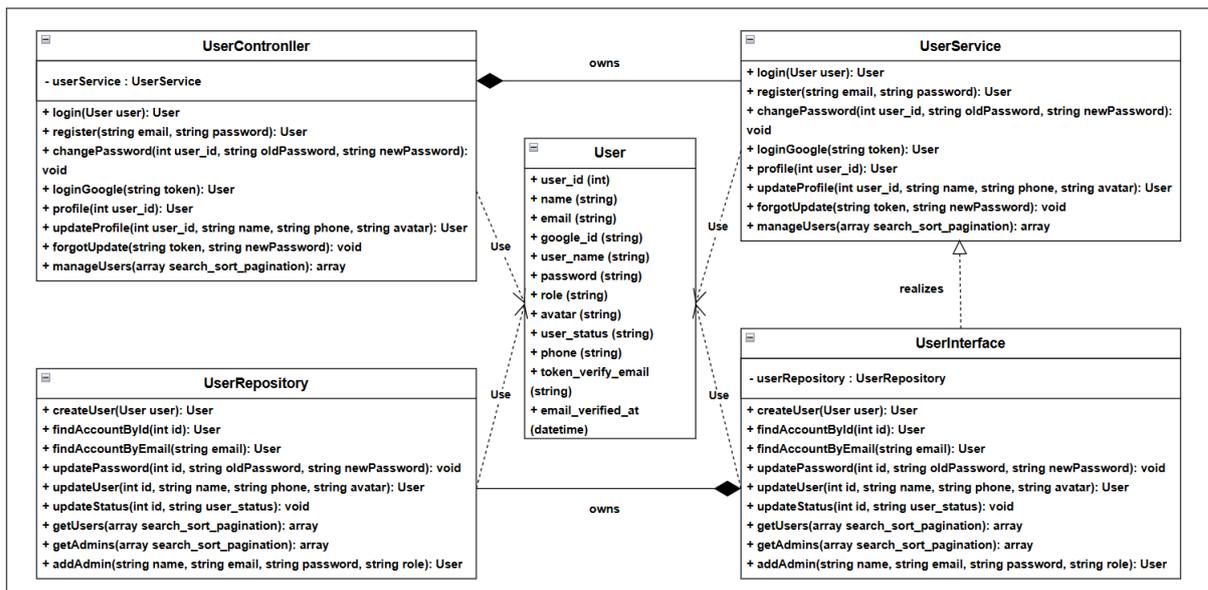
Hình 3.59 Biểu đồ hoạt động chức năng User quản lý bài viết



Hình 3.60 Biểu đồ hoạt động chức năng Sinh ảnh nghệ thuật

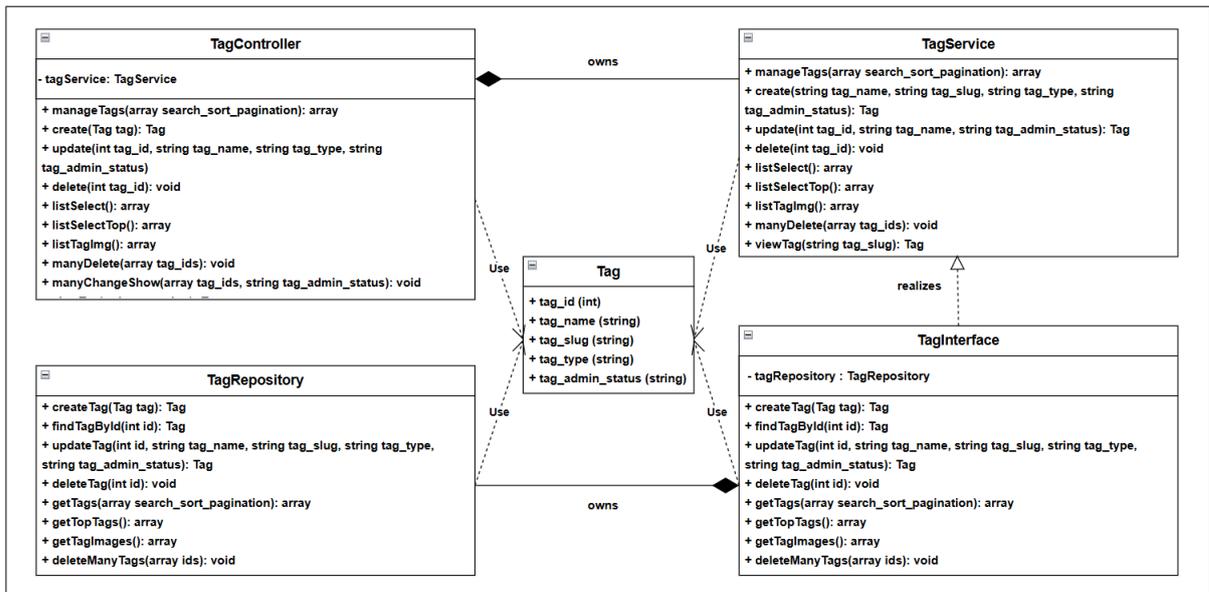
### 3.8. Class Diagrams

#### 3.8.1. Biểu đồ lớp User

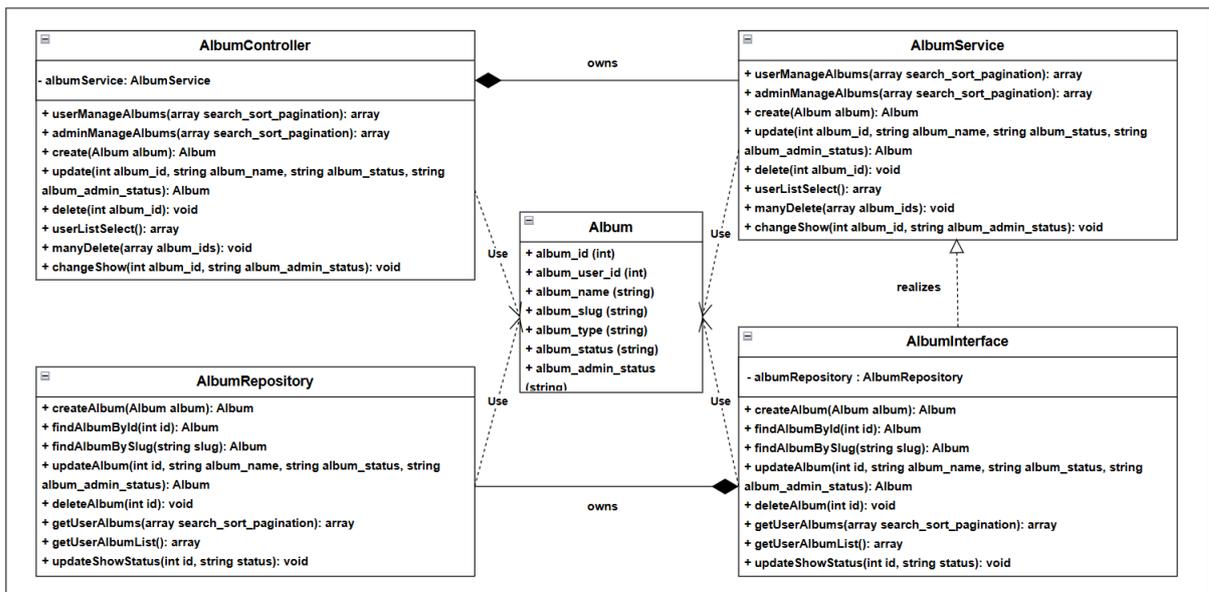


Hình 3.61 Biểu đồ lớp User

### 3.8.2. Biểu đồ lớp Tag, Album

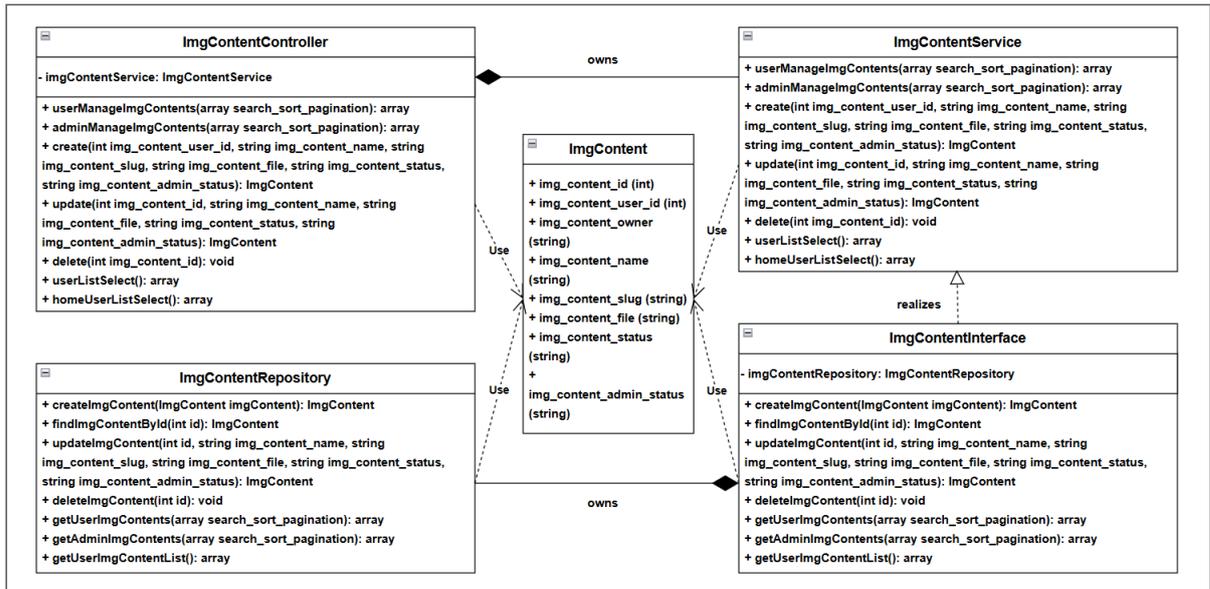


Hình 3.62 Biểu đồ lớp Tag

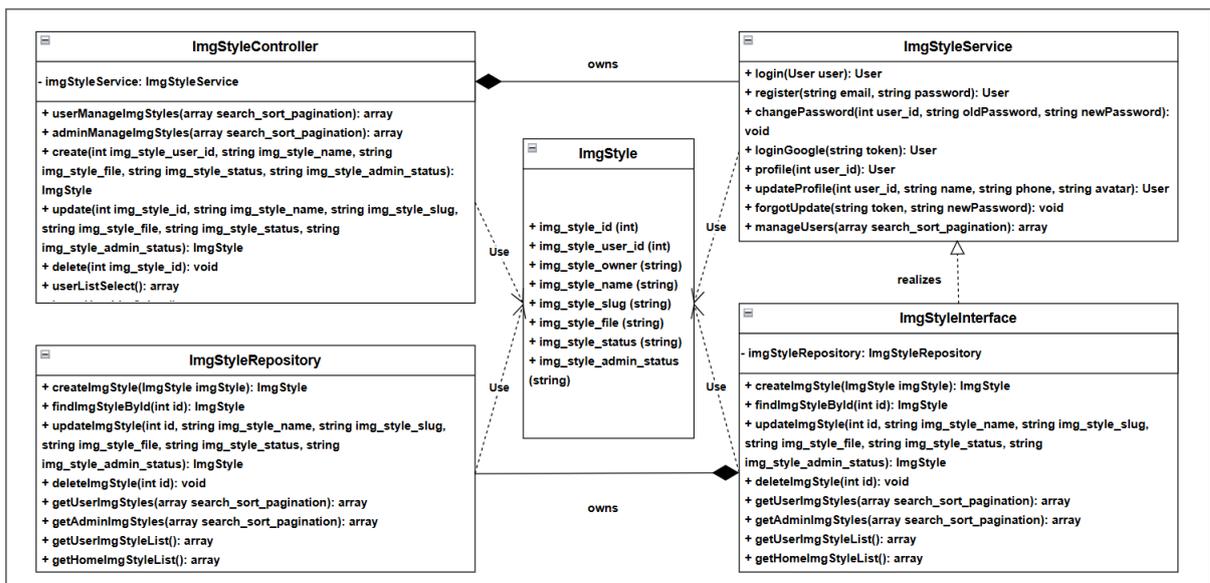


Hình 3.63 Biểu đồ lớp Album

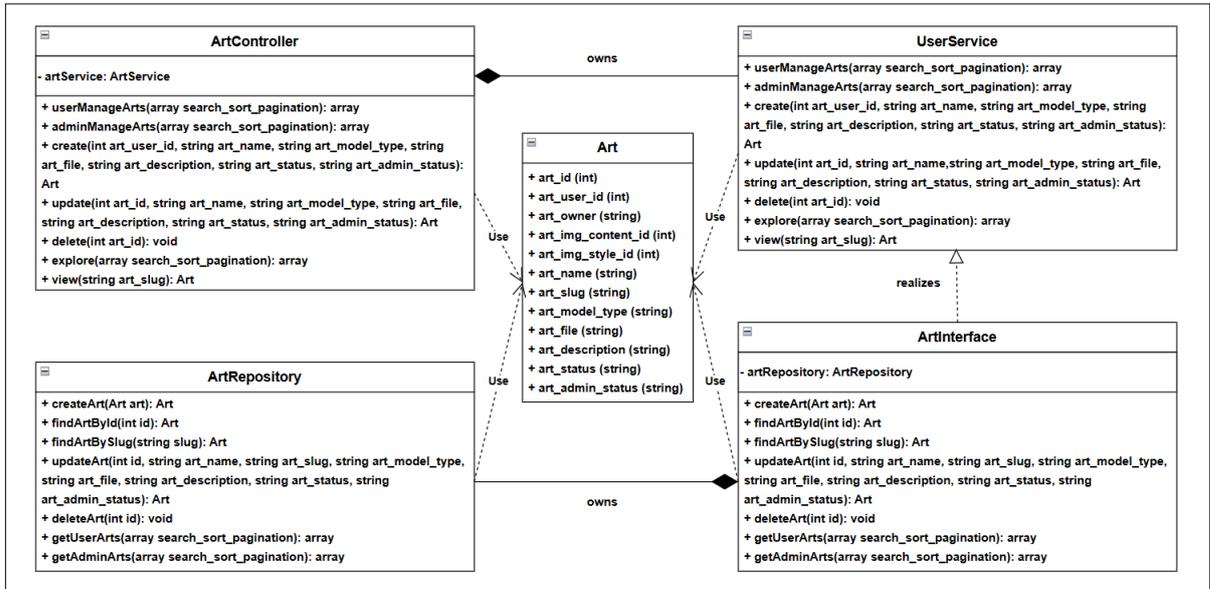
### 3.8.3. Danh sách biểu đồ lớp liên quan đến Art



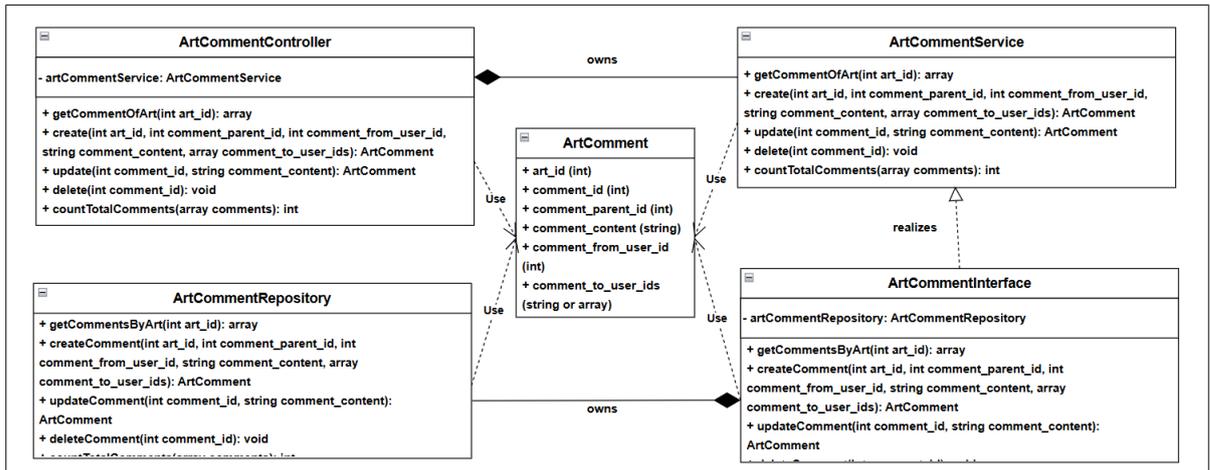
Hình 3.64 Biểu đồ lớp Img Content



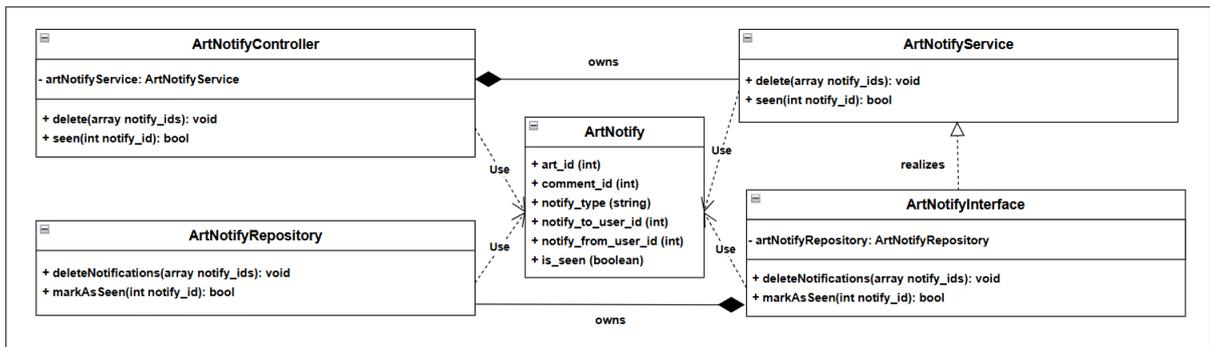
Hình 3.65 Biểu đồ lớp Img Style



Hình 3.66 Biểu đồ lớp Art

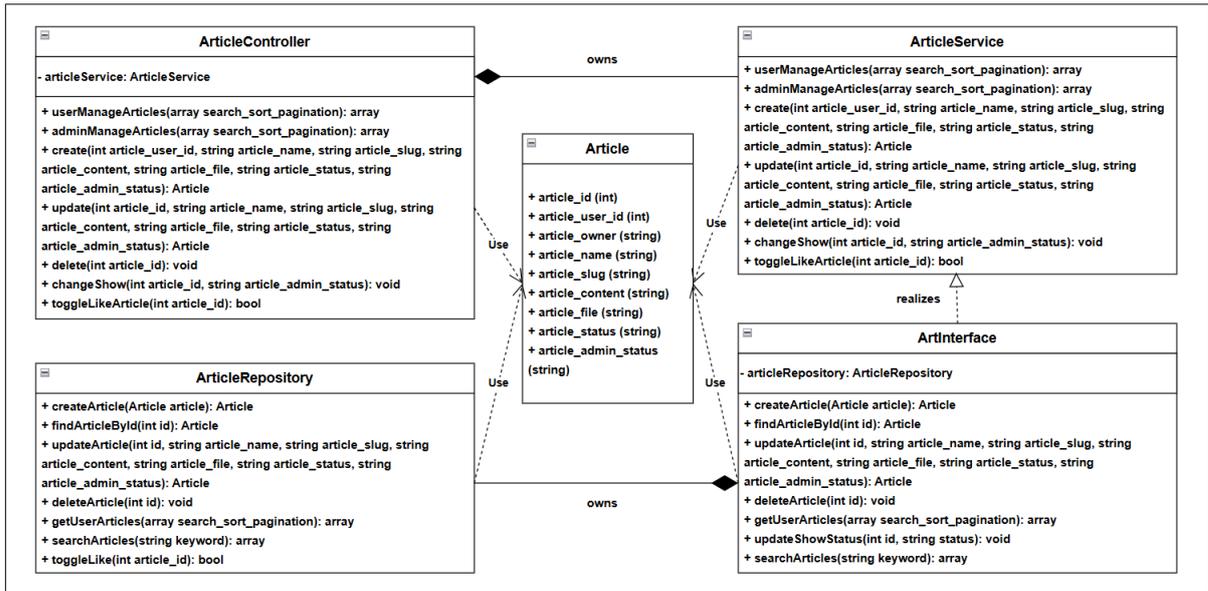


Hình 3.67 Biểu đồ lớp Art Comment

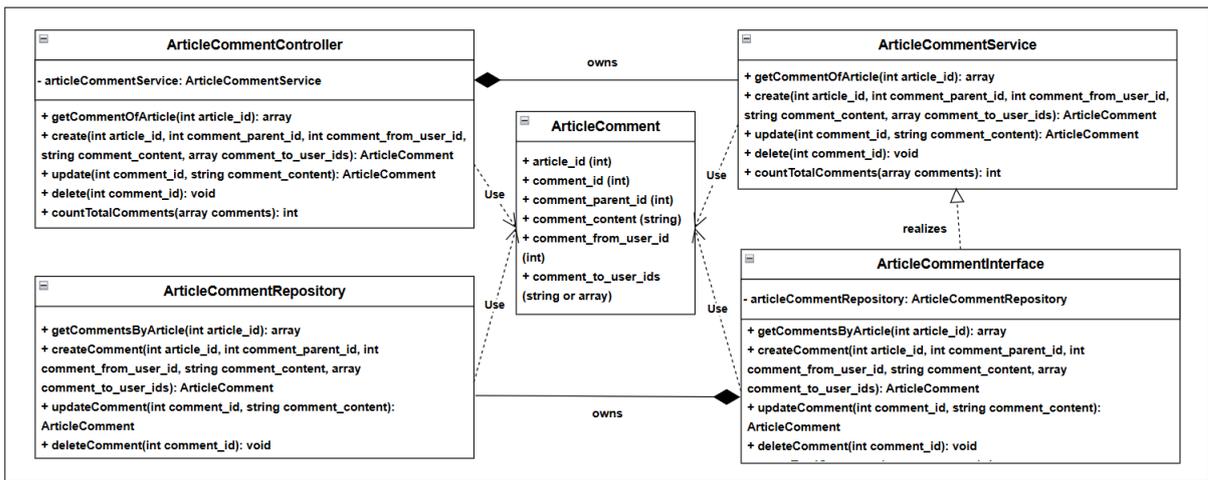


Hình 3.68 Biểu đồ lớp Art Notify

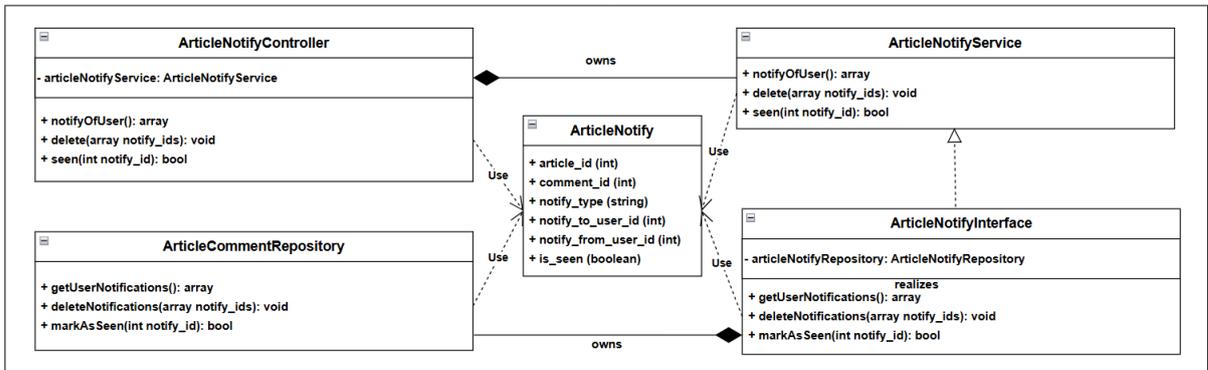
### 3.8.4. Danh sách biểu đồ lớp liên quan đến Article



Hình 3.69 Biểu đồ lớp Article

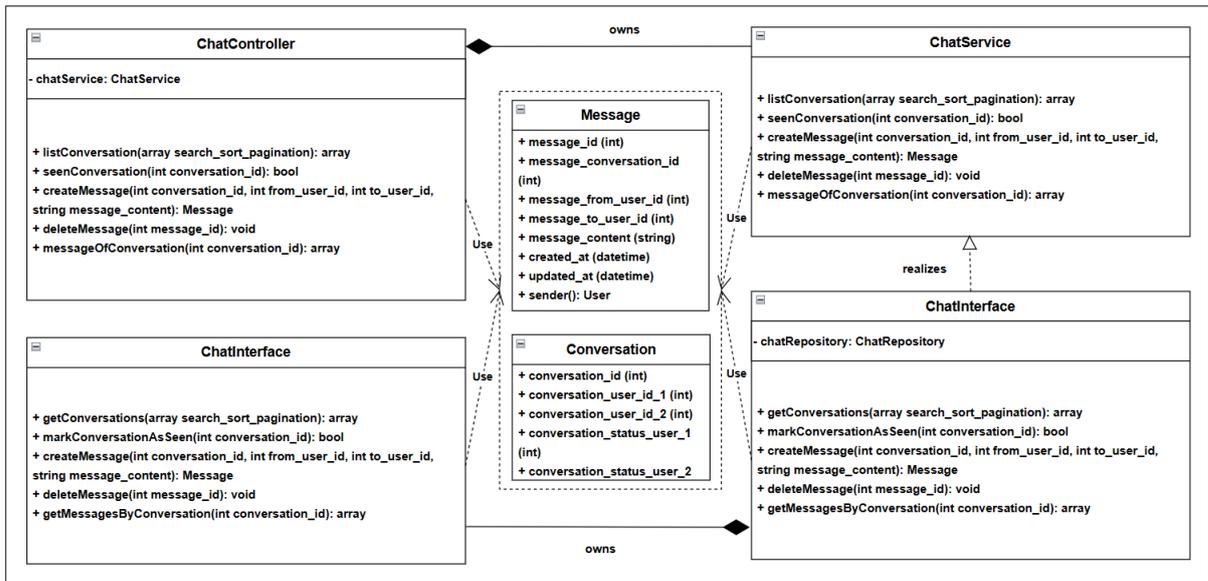


Hình 3.70 Biểu đồ lớp Article Comment



Hình 3.71 Biểu đồ lớp Article Notify

### 3.8.5. Biểu đồ lớp Message



Hình 3.72 Biểu đồ lớp Message

## CHƯƠNG 4: KẾT QUẢ TRIỂN KHAI

Chương này trình bày chi tiết quy trình từng bước triển khai hệ thống ArtGenix, tập trung vào việc hiện thực hóa API backend, giao diện người dùng, mô hình sinh ảnh và quá trình kiểm thử hệ thống. Chương nêu bật các mốc quan trọng đã đạt được trong quá trình phát triển, giới thiệu ảnh chụp màn hình hệ thống đang hoạt động, và cung cấp đánh giá toàn diện về các kết quả hữu hình từ nỗ lực triển khai, bao gồm hiệu quả của mô hình sinh ảnh và kết quả kiểm thử để đảm bảo tính ổn định và hiệu suất của hệ thống.

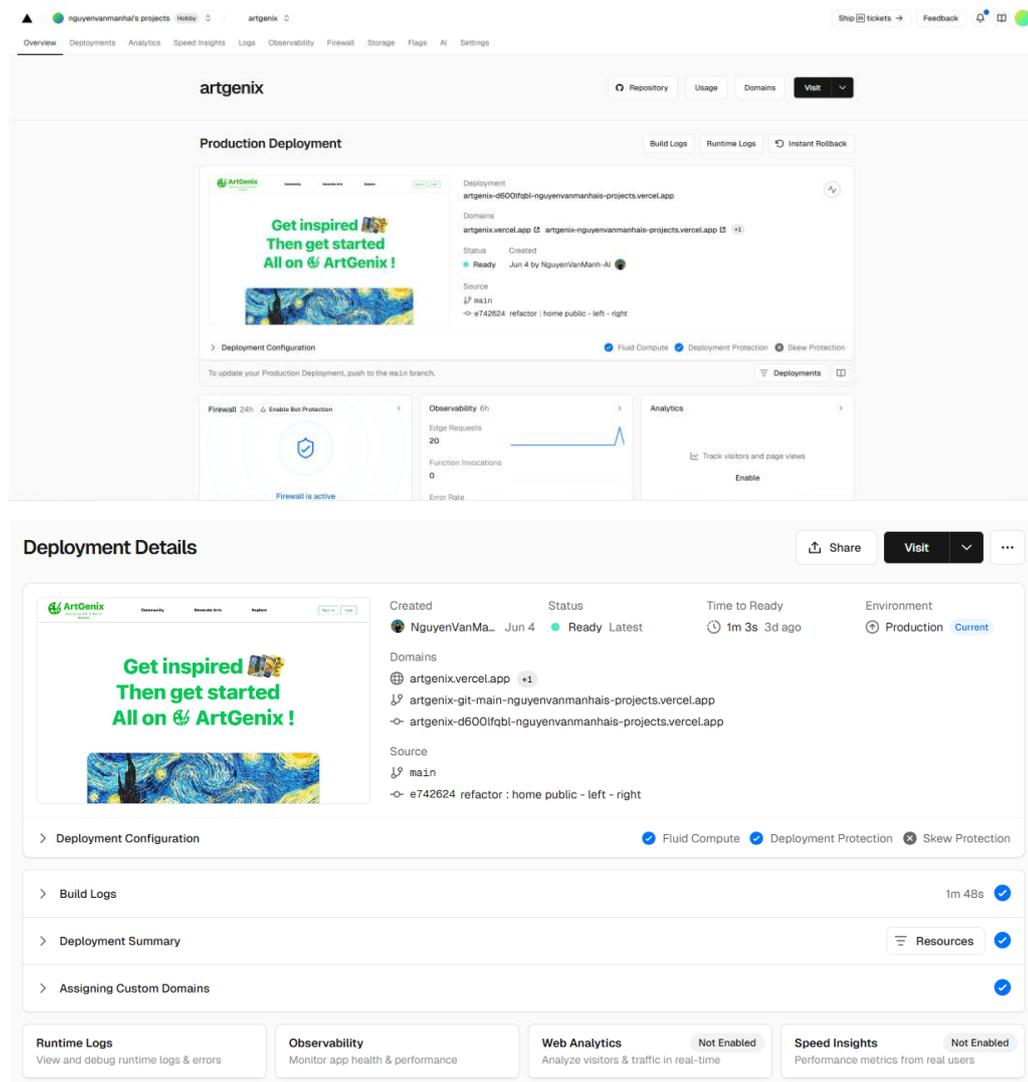
### 4.1. Triển khai

Hệ thống sử dụng Vercel để triển khai giao diện người dùng, Railway để triển khai dịch vụ API backend và cơ sở dữ liệu, cùng với Kaggle để triển khai các API Model AI, tất cả được tích hợp trong hệ thống ArtGenix.

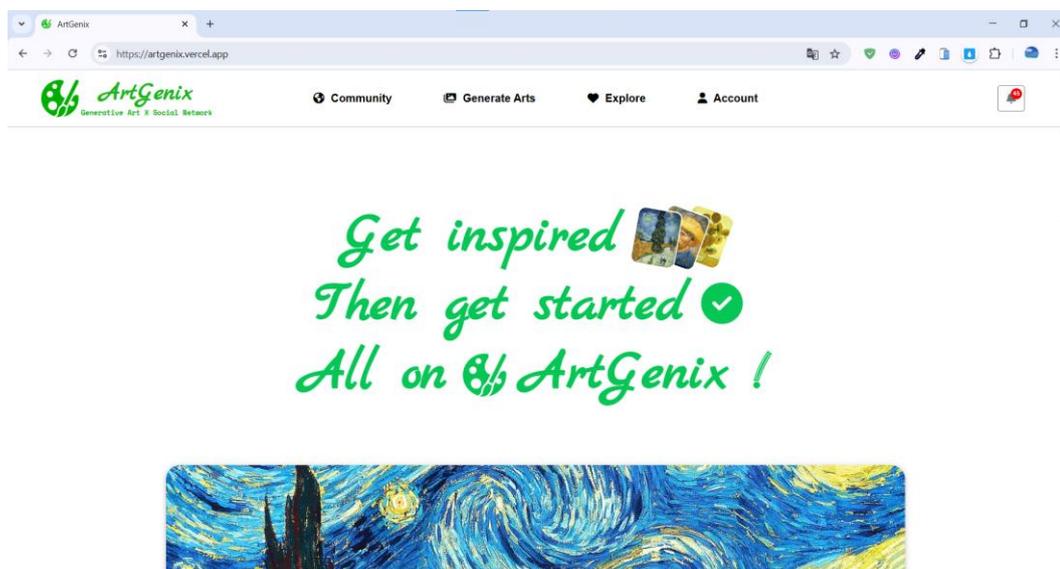
Dịch vụ	Địa chỉ dịch vụ	Nhà cung cấp	Ghi chú
Giao diện người dùng (VueJS)	<a href="https://artgenix.vercel.app/">https://artgenix.vercel.app/</a>	Vercel	100 deployments/day, 23GB size, 100MB static files, 1 concurrent build
API Backend (Laravel)	<a href="https://artgenix-production.up.railway.app/">https://artgenix-production.up.railway.app/</a>	Railway	Memory per container: 512 MB CPU per container: 2 vCPU Shared disk: 1 GB
MySQL DB	<a href="https://viaduct.proxy.rlwy.net">viaduct.proxy.rlwy.net</a>	Railway	Memory per container: 512 MB CPU per container: 2 vCPU Shared disk: 1 GB
API Model	<a href="https://uri.ngrok-free.app/api/nst">https://uri.ngrok-free.app/api/nst</a>	Kaggle	P100 – VRAM 16GB – CUDA cores – 4 CPU – RAM ~13GB – Disk working: 20GB
API Model	<a href="https://uri.ngrok-free.app/api/gan">https://uri.ngrok-free.app/api/gan</a>	Kaggle	P100 – VRAM 16GB – CUDA cores – 4 CPU – RAM ~13GB – Disk working: 20GB
API Model	<a href="https://uri.ngrok-free.app/api/lora">https://uri.ngrok-free.app/api/lora</a>	Kaggle	P100 – VRAM 16GB – CUDA cores – 4 CPU – RAM ~13GB – Disk working: 20GB
API Model	<a href="https://uri.ngrok-free.app/api/text2img">https://uri.ngrok-free.app/api/text2img</a>	Kaggle	P100 – VRAM 16GB – CUDA cores – 4 CPU – RAM ~13GB – Disk working: 20GB

Bảng 4.1 Triển khai hệ thống

### 4.1.1. Vercel Client VueJS

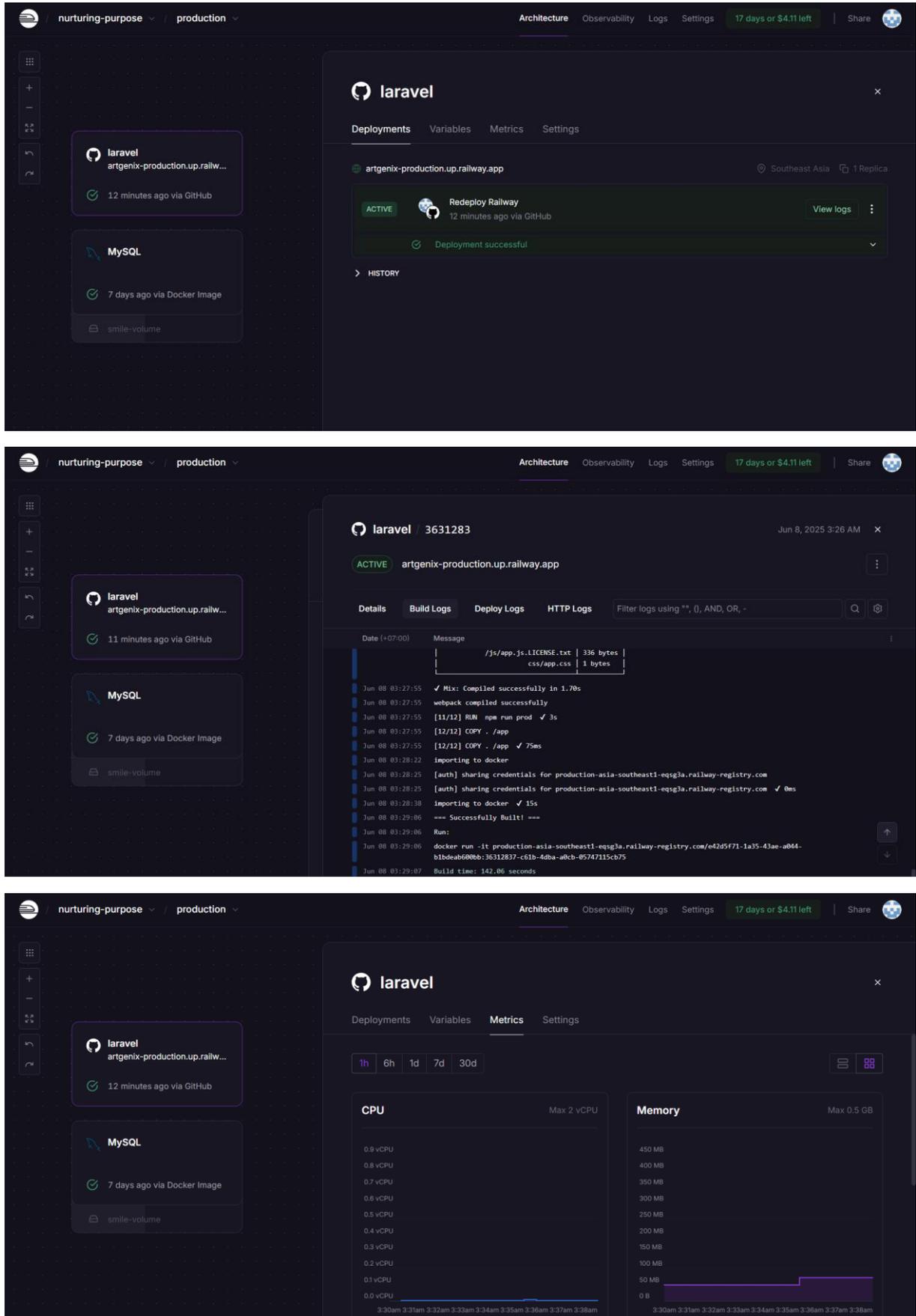


Hình 4.1 Triển khai Client VueJS lên Vercel

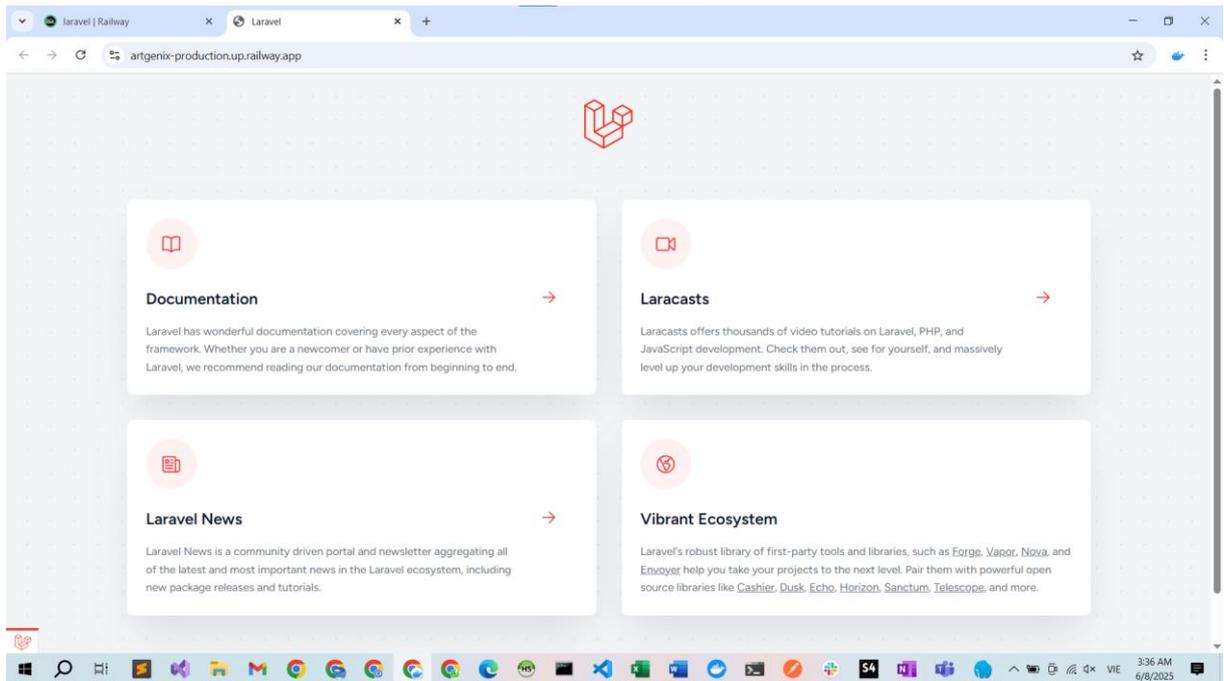


Hình 4.2 Kết quả triển khai Client VueJS

### 4.1.2. Railway Backend Laravel API

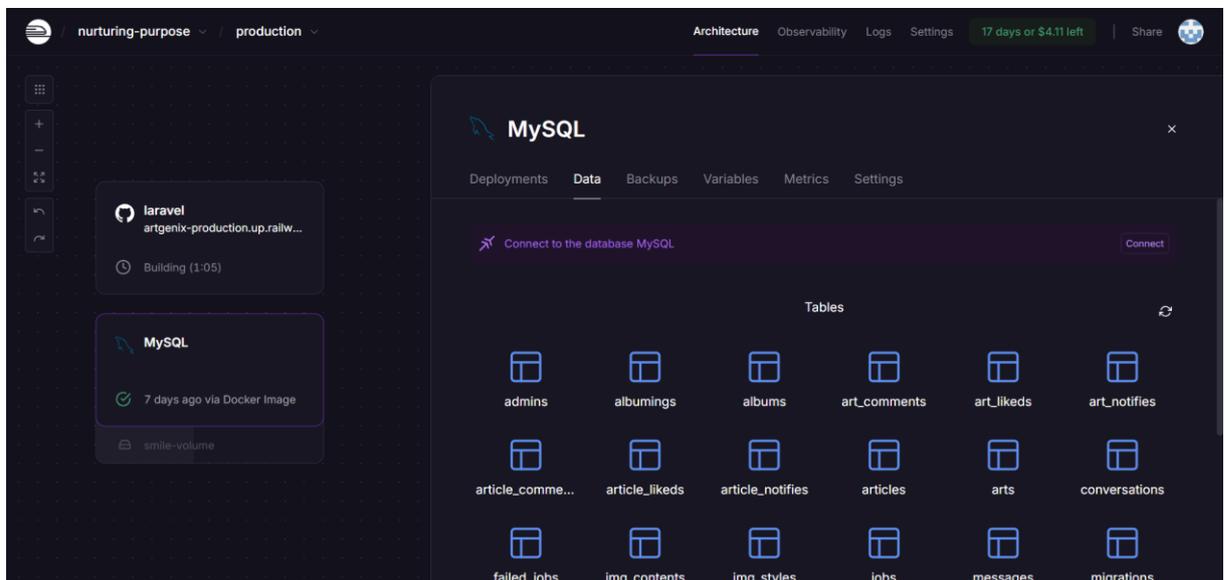
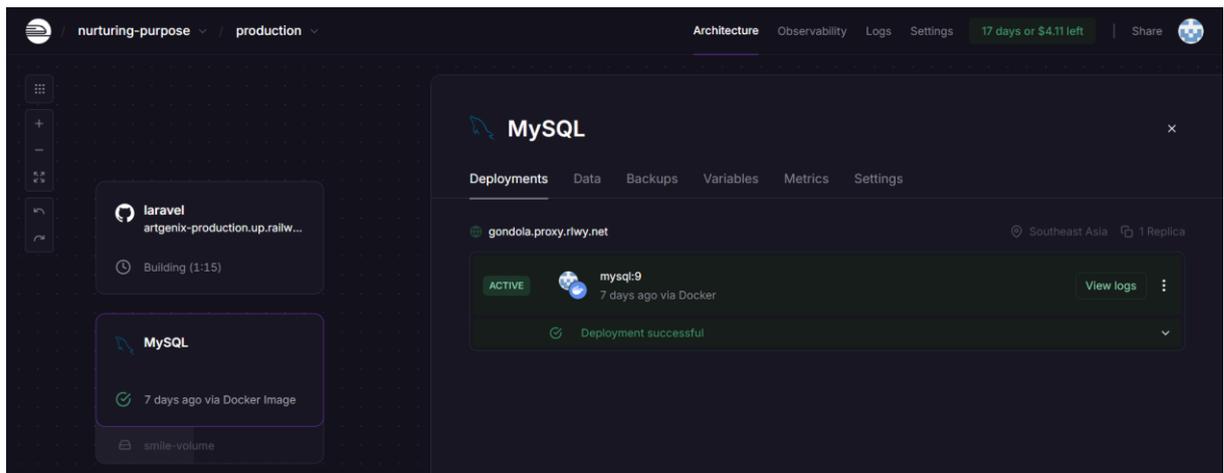


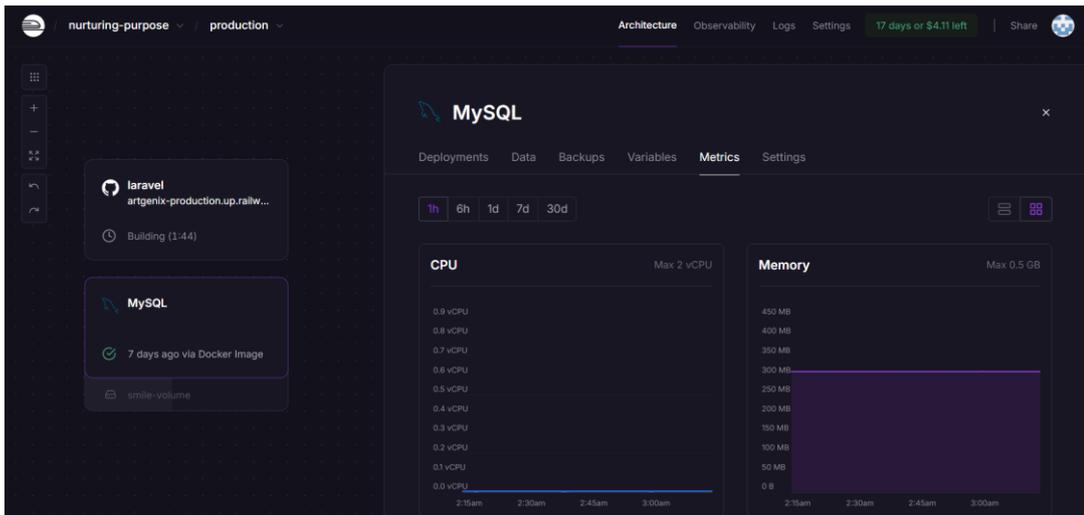
Hình 4.3 Triển khai backend API Laravel lên Railway



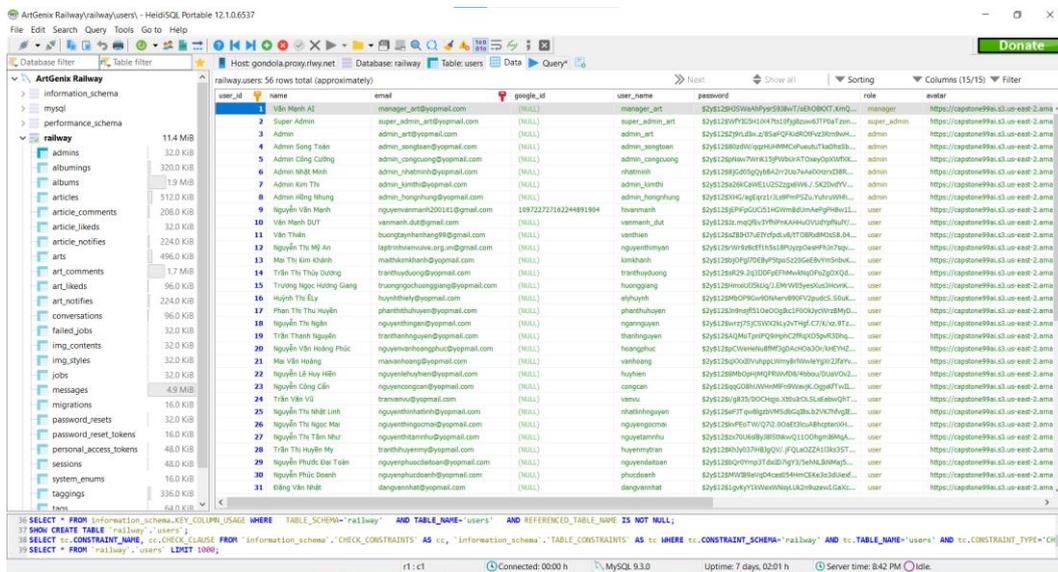
Hình 4.4 Kết quả triển khai backend API Laravel

### 4.1.3. Railway MySQL



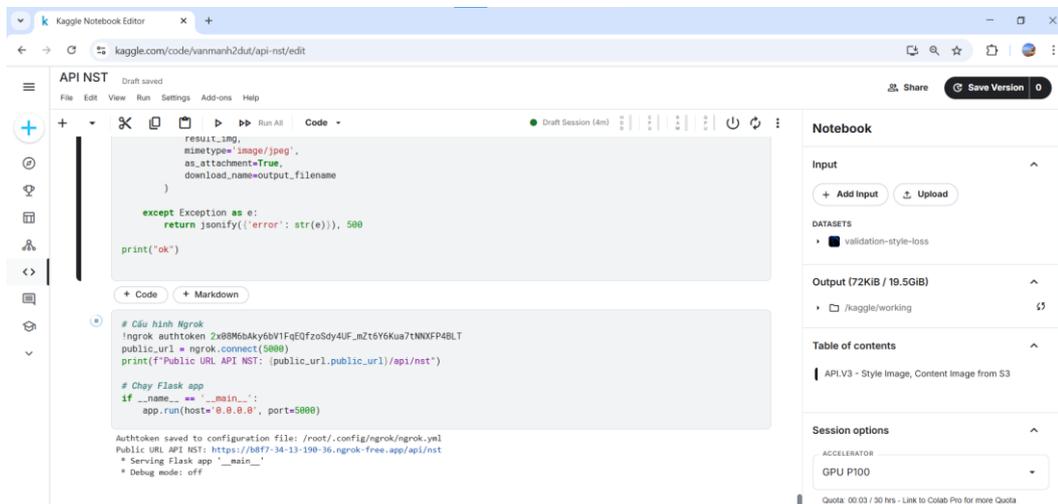


Hình 4.5 Triển khai cơ sở lưu trữ dữ liệu MySQL lên Railway



Hình 4.6 Kết quả triển khai cơ sở lưu trữ dữ liệu MySQL

#### 4.1.4. Kaggle Flask Ngrok API Model



Hình 4.7 Công khai API Model AI thông qua Flask & Ngrok trên Kaggle

## 4.2. APIs Server

Hệ thống đã triển khai hơn 136 API service cùng với 4 API Model API để hỗ trợ đầy đủ các tính năng cốt lõi của ArtGenix.

```

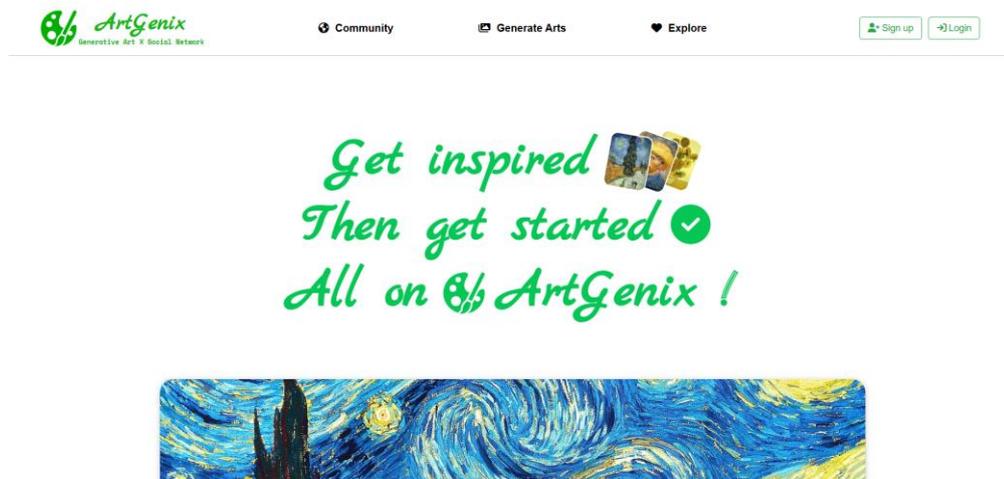
Cmder
POST /api/admin/album/change-show
GET /api/admin/album/change-show
POST /api/admin/album/many-change-show
POST /api/admin/art/change-show
POST /api/admin/art/many-change-show
POST /api/admin/article/change-show
POST /api/admin/article/many-change-show
POST /api/admin/img/content/change-show
POST /api/admin/img/content/manage
POST /api/admin/img/content/many-change-show
POST /api/admin/img-style/change-show
POST /api/admin/img-style/manage
POST /api/admin/img-style/many-change-show
POST /api/admin/statistical
POST /api/admin/system-enum/get
POST /api/admin/tag/delete
POST /api/admin/tag/manage
POST /api/admin/tag/many-change-show
POST /api/admin/tag/many-delete
POST /api/admin/tag/update
POST /api/admin/user/add-admin
POST /api/admin/user/change-active
POST /api/admin/user/change-active-admin
POST /api/admin/user/change-role-admin
POST /api/admin/user/manage
POST /api/admin/user/manage-admin
POST /api/admin/user/many-change-active
POST /api/auth/change-password
POST /api/auth/create-new-google
POST /api/auth/forget-pw-encode
POST /api/auth/forget-update
POST /api/auth/login
POST /api/auth/login-google
POST /api/auth/logout
POST /api/auth/logout
POST /api/auth/profile
POST /api/auth/register
POST /api/auth/update-profile
POST /api/document/add-art-to-album
POST /api/home/album/add-art-to-album
POST /api/home/art/comment/all
POST /api/home/art/explore
POST /api/home/art/related-content
POST /api/home/art/explore
POST /api/home/art/related-content
POST /api/home/art/related-style
POST /api/home/art/view
POST /api/home/article/comment/all
POST /api/home/article/detail
POST /api/home/article/list-user-community
POST /api/home/article/personal
POST /api/home/article/public
POST /api/home/article/recent-articles
POST /api/home/article/search-articles-or-article
POST /api/home/generate/gan
POST /api/home/generate/lena
POST /api/home/generate/nst
POST /api/home/generate/text2img
POST /api/home/img-content/list-select
POST /api/home/personal/album/art
POST /api/home/personal/album/list-album-img
POST /api/home/personal/album/view
POST /api/home/personal/list-account
POST /api/home/tag/list-select
POST /api/home/tag/list-select-top
POST /api/home/tag/list-tag-img
POST /api/home/tag/view
POST /api/home/user-mention
POST /api/outside/callback
POST /api/s/delete
POST /api/s/upload
POST /api/user/album/art
POST /api/user/album/create
POST /api/user/album/delete
POST /api/user/album/img-content
POST /api/user/album/img-style
POST /api/user/album/list-album-img
POST /api/user/album/list-select
POST /api/user/album/manage
POST /api/user/album/many-change-public
POST /api/user/album/many-delete
POST /api/user/album/update
POST /api/user/album/view
POST /api/user/art/comment/create
POST /api/user/art/comment/delete
POST /api/user/art/comment/update
POST /api/user/art/create
POST /api/user/art/delete
POST /api/user/art/list-select
POST /api/user/article/img-style
POST /api/user/album/list-album-img
POST /api/user/album/list-select
POST /api/user/album/manage
POST /api/user/album/many-change-public
POST /api/user/album/many-delete
POST /api/user/album/update
POST /api/user/album/view
POST /api/user/art/comment/create
POST /api/user/art/comment/delete
POST /api/user/art/comment/update
POST /api/user/art/create
POST /api/user/art/delete
POST /api/user/art/list-select
POST /api/user/article/manage
POST /api/user/article/many-change-public
POST /api/user/article/many-delete
POST /api/user/article/notify/all
POST /api/user/article/notify/delete
POST /api/user/article/notify/seen
POST /api/user/article/toggle-like
POST /api/user/article/update
POST /api/user/chat/conversation/list
POST /api/user/chat/conversation/seen
POST /api/user/chat/message/create
POST /api/user/chat/message/delete
POST /api/user/chat/message/list
POST /api/user/chat/search-user
POST /api/user/generate/gan
POST /api/user/generate/lena
POST /api/user/generate/nst
POST /api/user/generate/text2img
POST /api/user/img-content/create
POST /api/user/img-content/delete
POST /api/user/img-content/list-select

```

Hình 4.8 Danh sách API của hệ thống

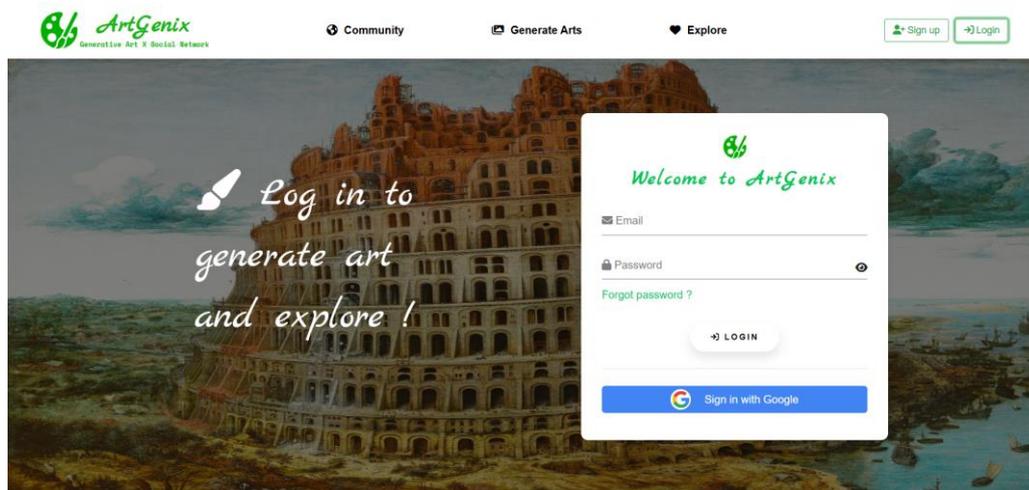
## 4.3. Giao diện ứng dụng

### 4.3.1. Trang chủ

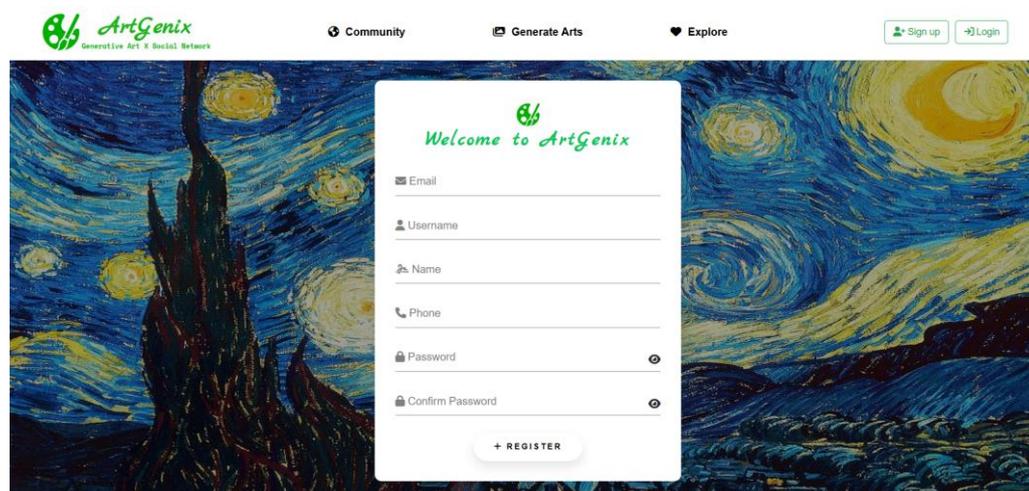


Hình 4.9 Trang chủ

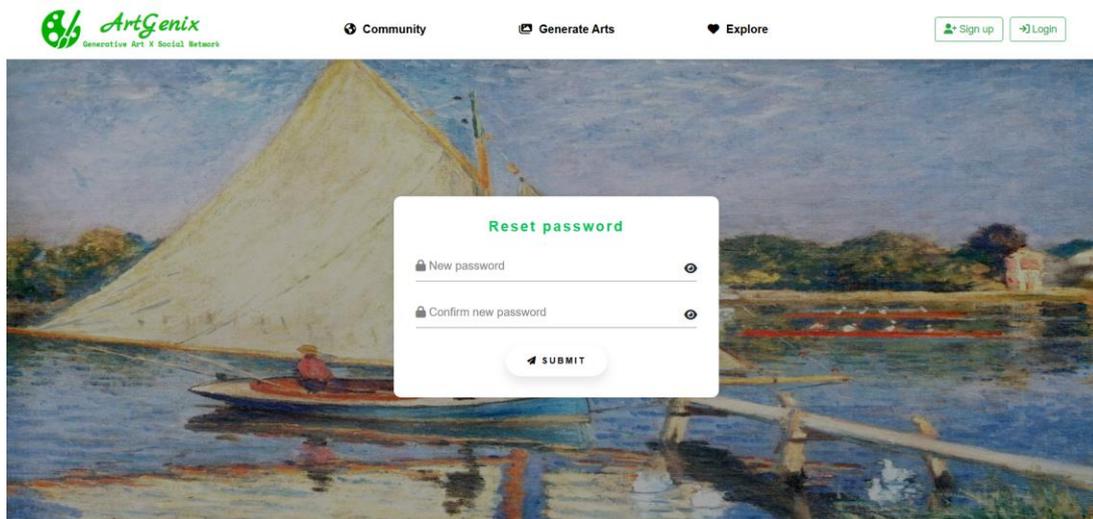
### 4.3.2. Nhóm chức năng Xác thực và Cài đặt tài khoản



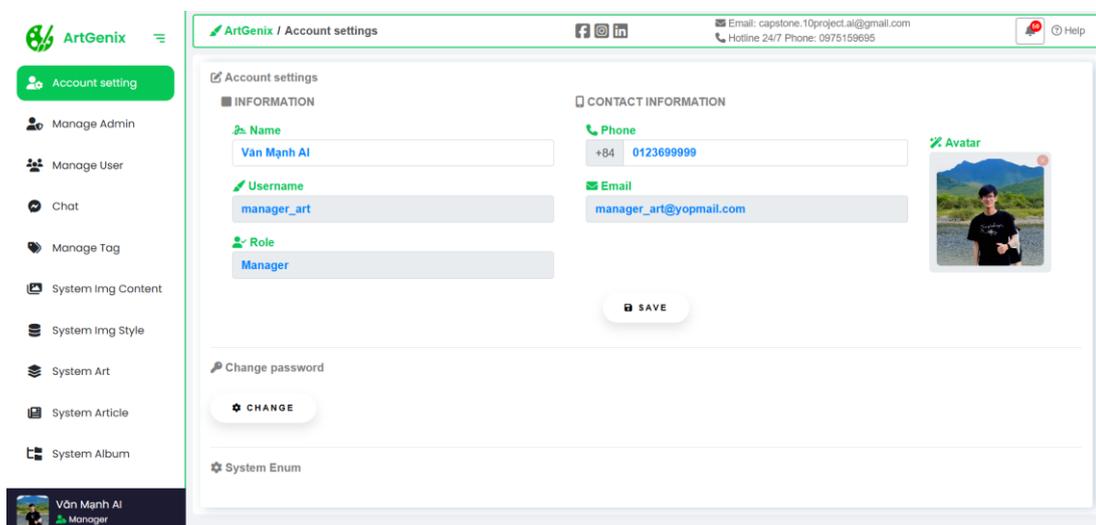
Hình 4.10 Đăng nhập



Hình 4.11 Đăng ký

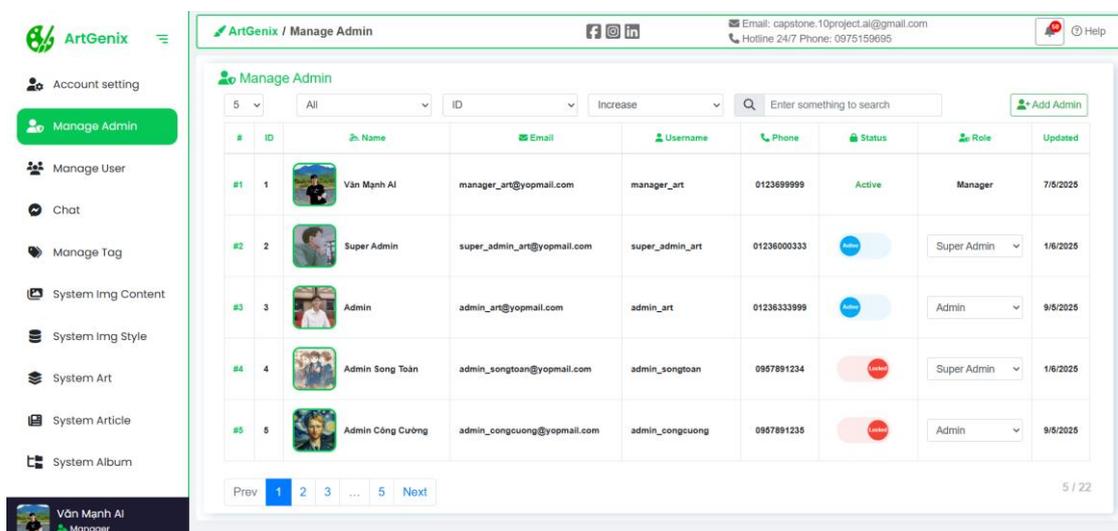


Hình 4.12 Đặt lại mật khẩu

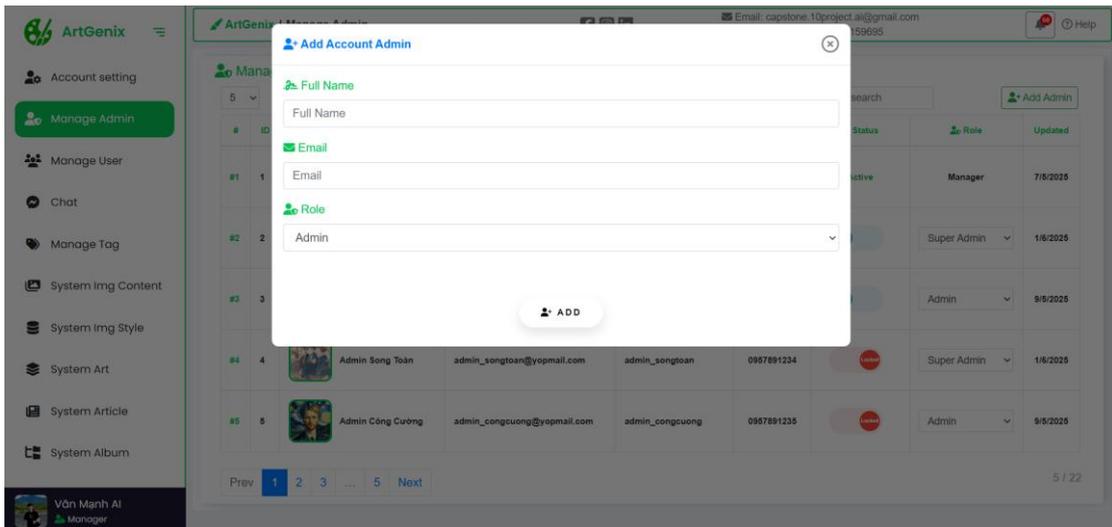


Hình 4.13 Cài đặt tài khoản

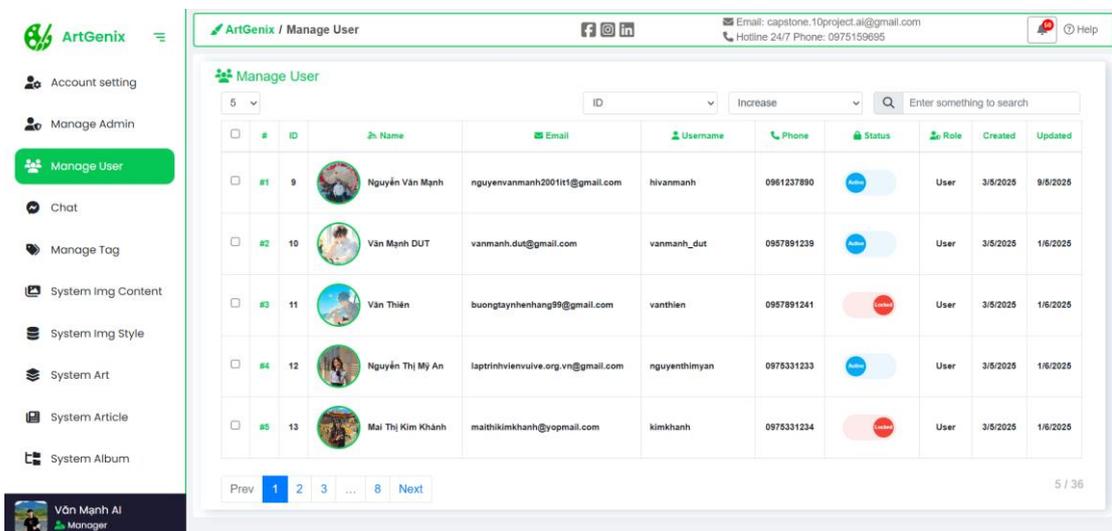
### 4.3.3. Nhóm chức năng Quản lý người dùng



Hình 4.14 Manager, Super admin quản lý admin

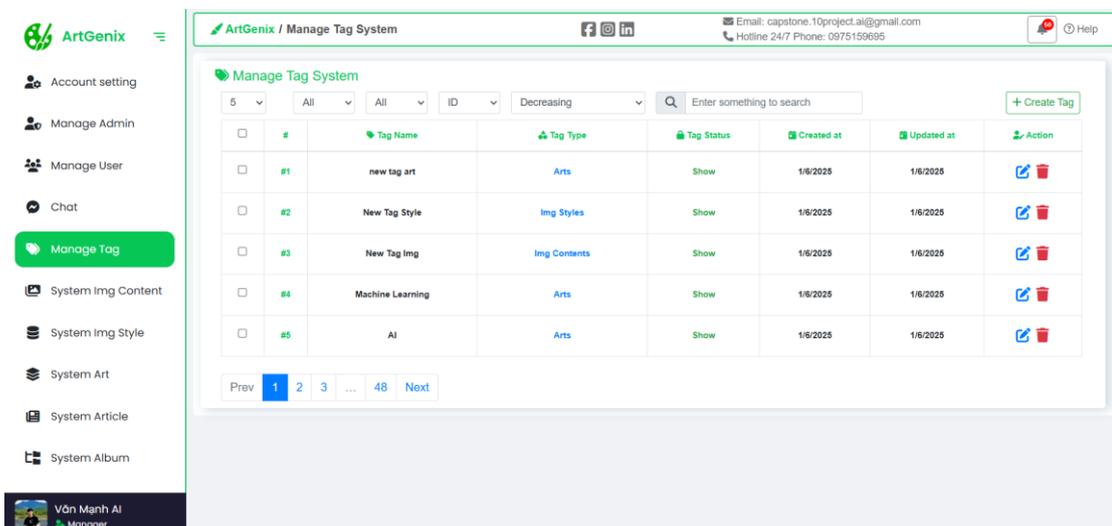


Hình 4.15 Thêm tài khoản admin

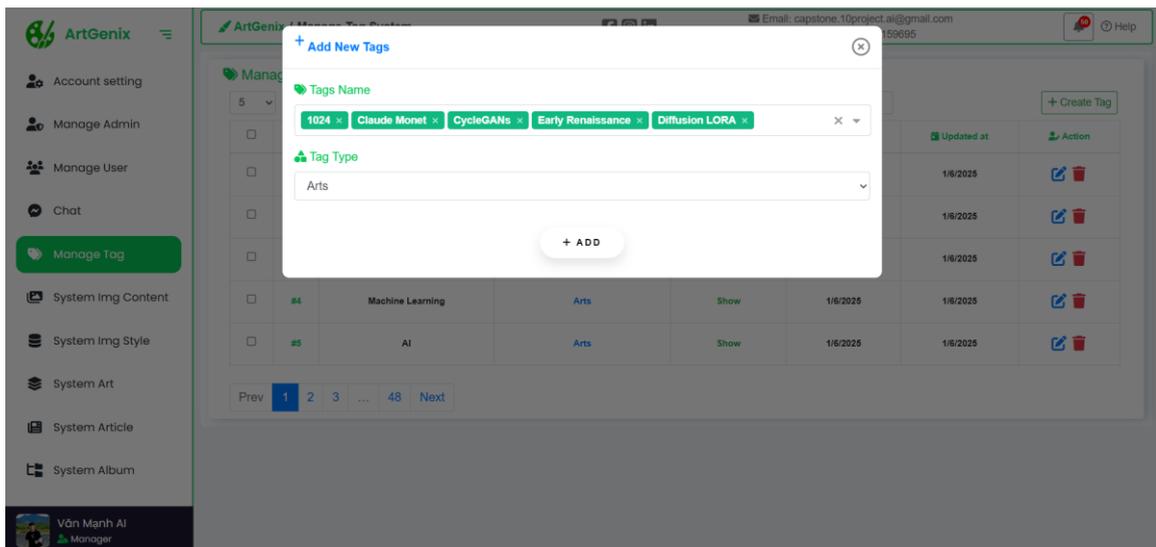


Hình 4.16 Admin quản lý User

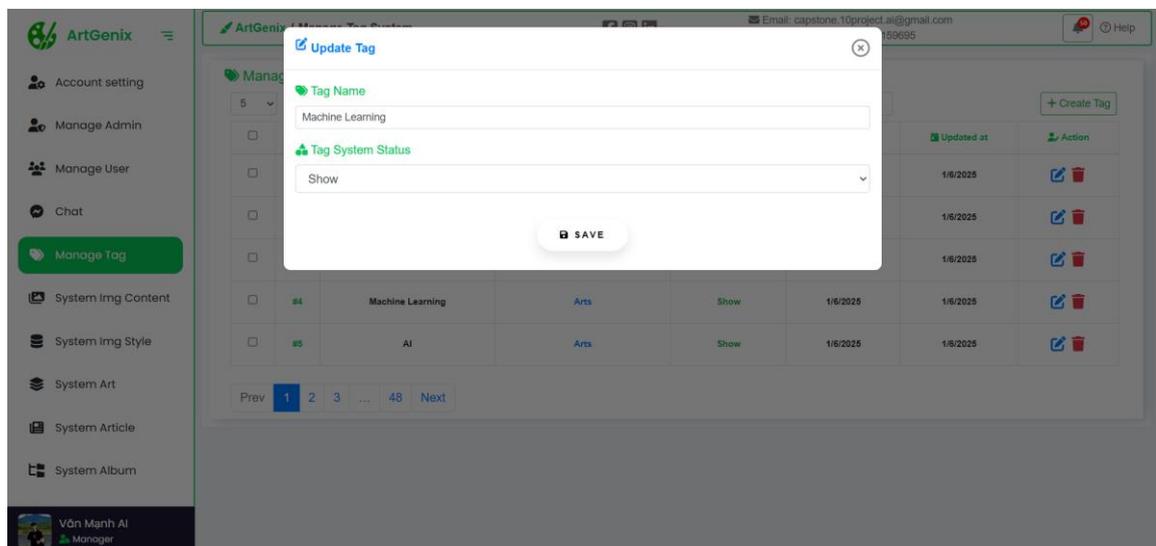
#### 4.3.4. Nhóm chức năng liên quan đến Tag



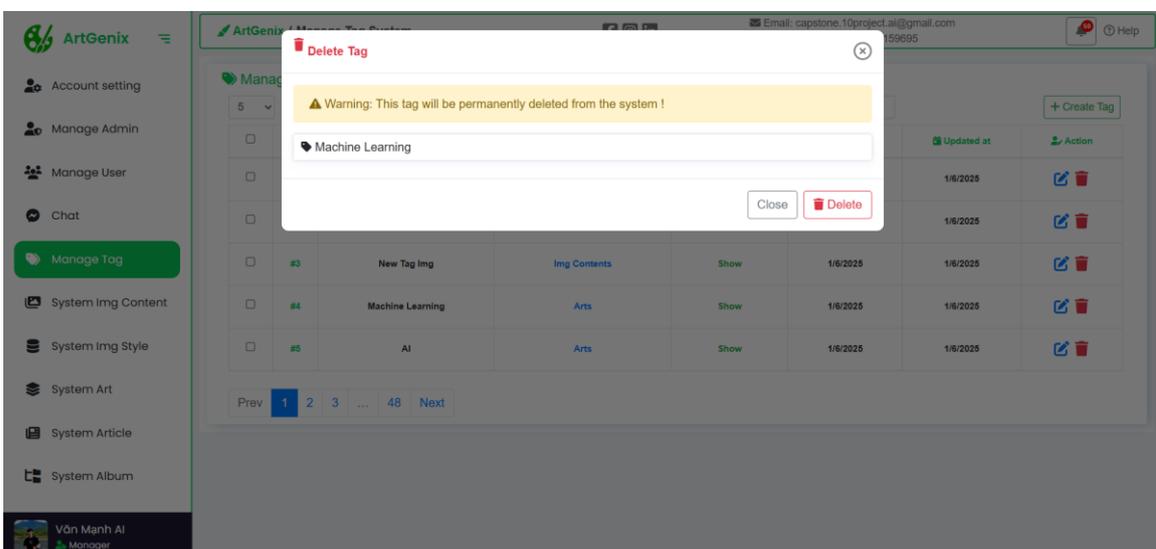
Hình 4.17 Admin quản lý Tag



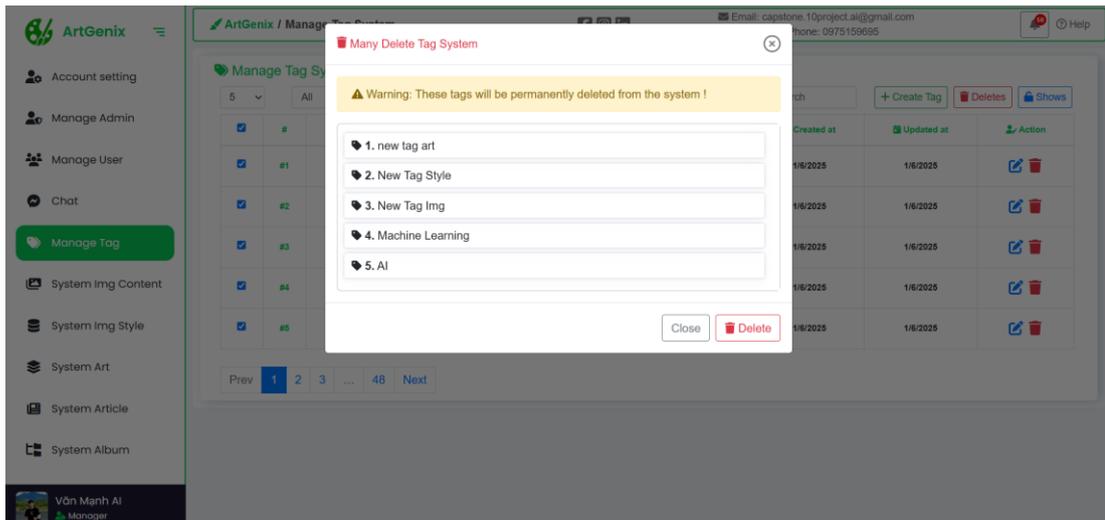
Hình 4.18 Admin thêm Tag



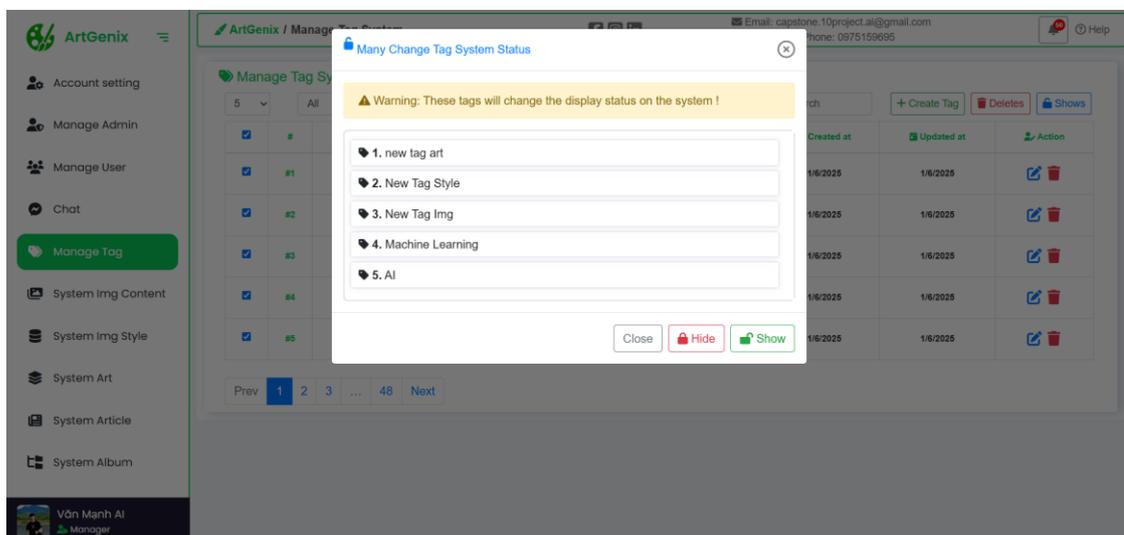
Hình 4.19 Admin chỉnh sửa Tag



Hình 4.20 Admin xóa Tag

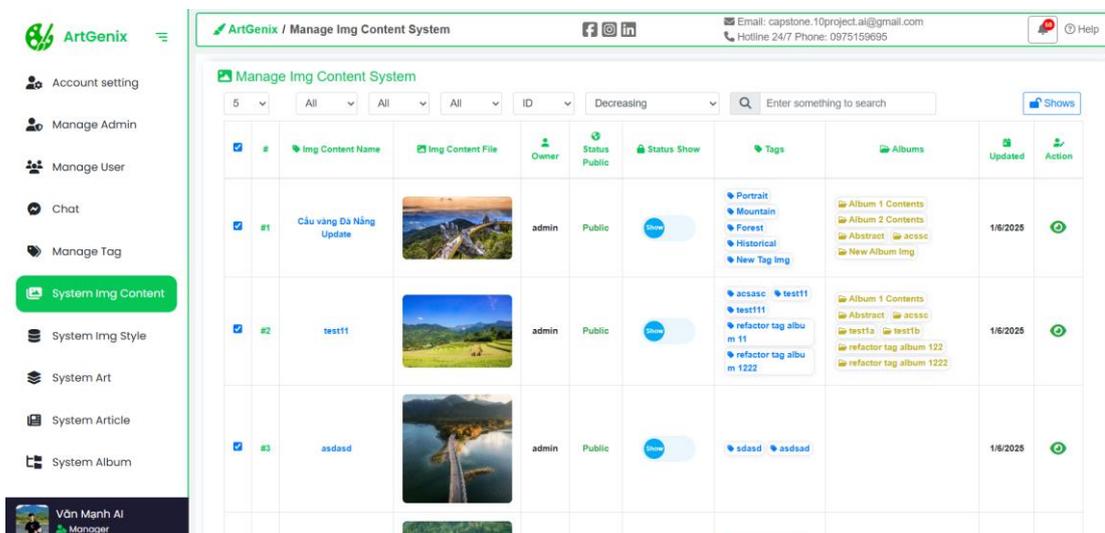


Hình 4.21 Admin xóa hàng loạt Tag

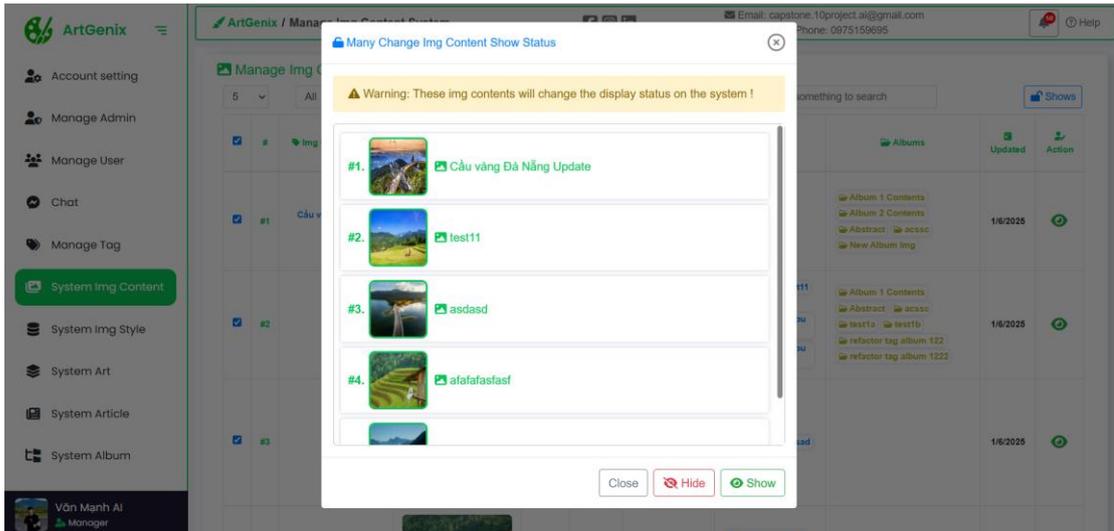


Hình 4.22 Admin thay đổi trạng thái hiển thị Tag

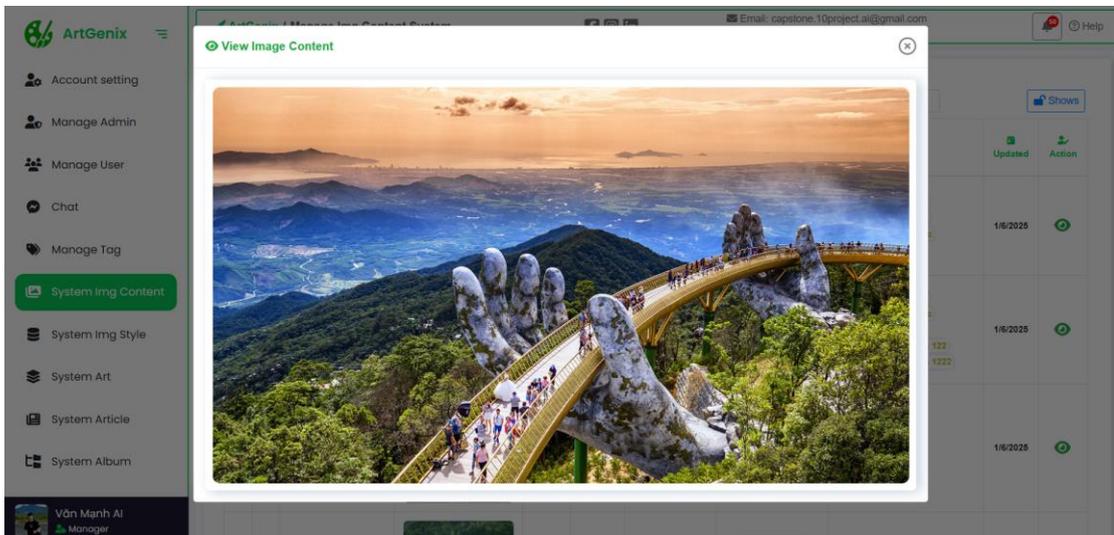
#### 4.3.5. Nhóm chức năng liên quan đến Img Content



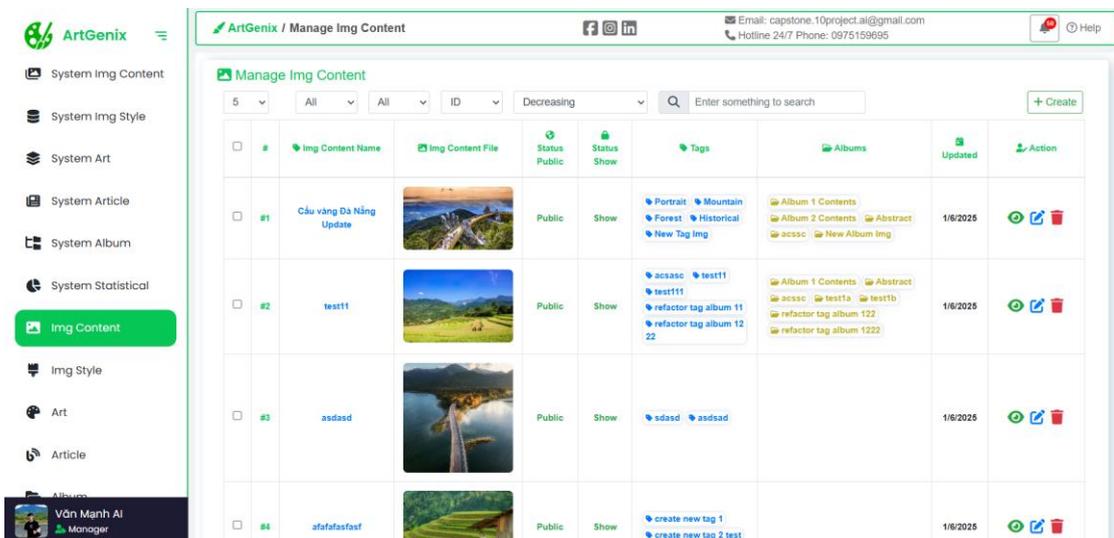
Hình 4.23 Admin quản lý Img Content



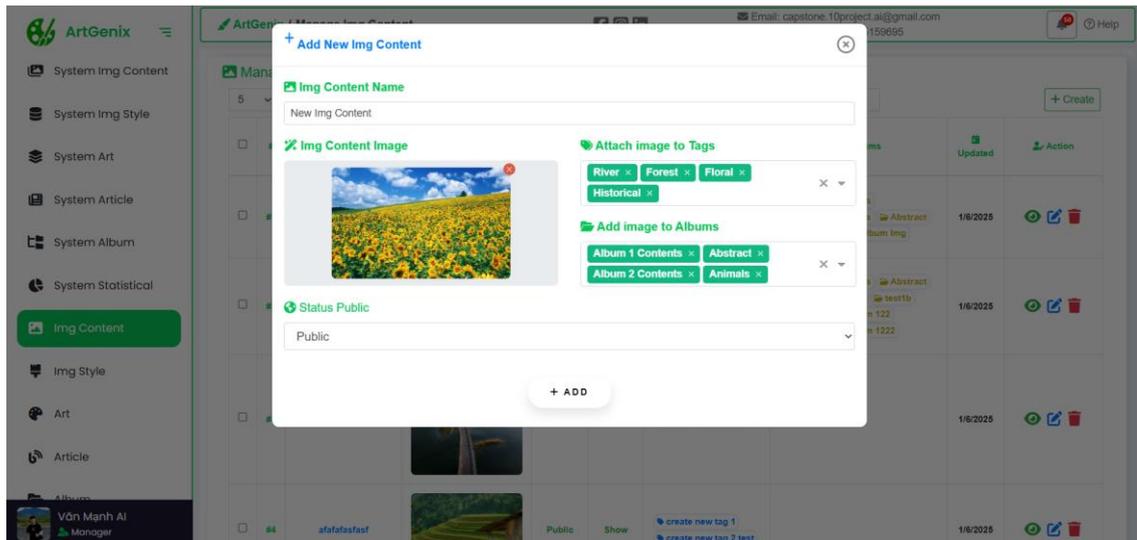
Hình 4.24 Admin thay đổi trạng thái hiển thị Img Content



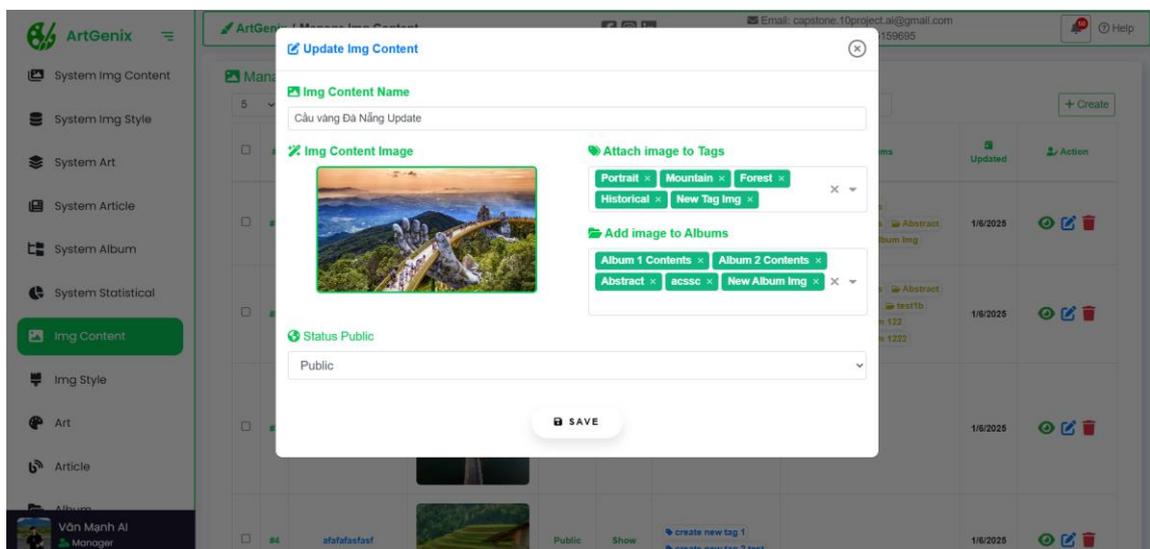
Hình 4.25 Admin xem chi tiết Img Content



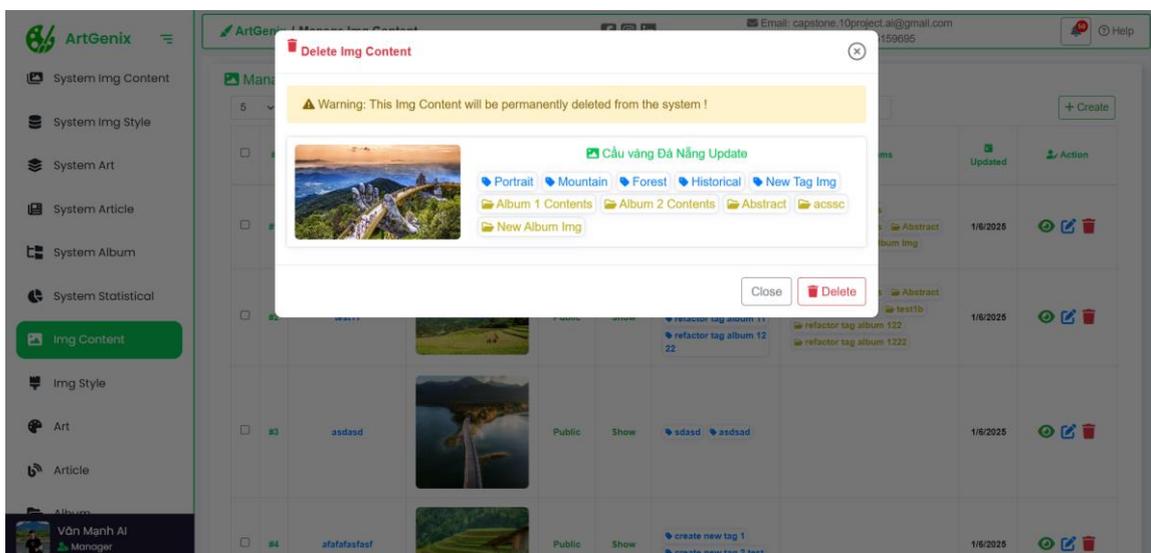
Hình 4.26 User quản lý Img Content



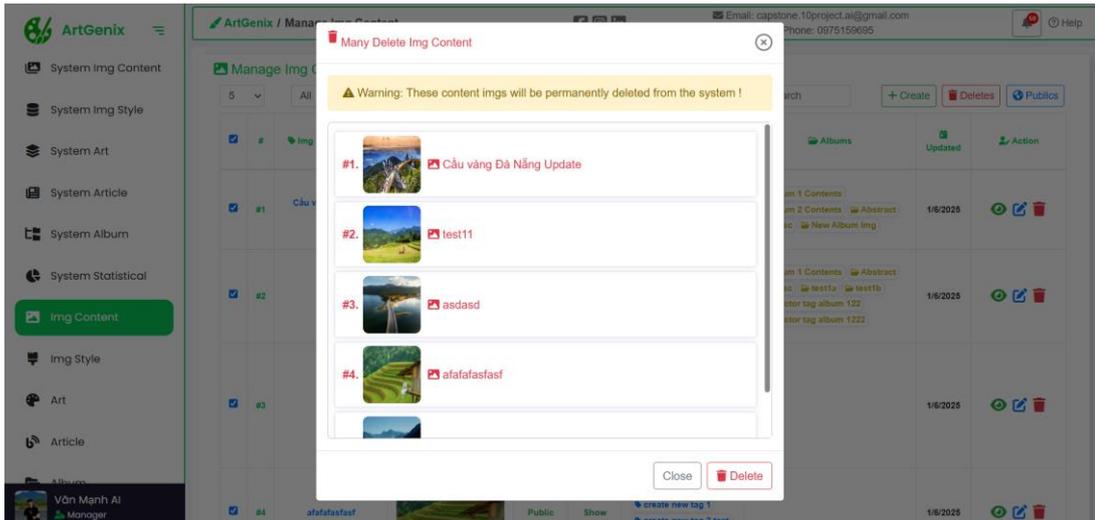
Hình 4.27 User thêm Img Content



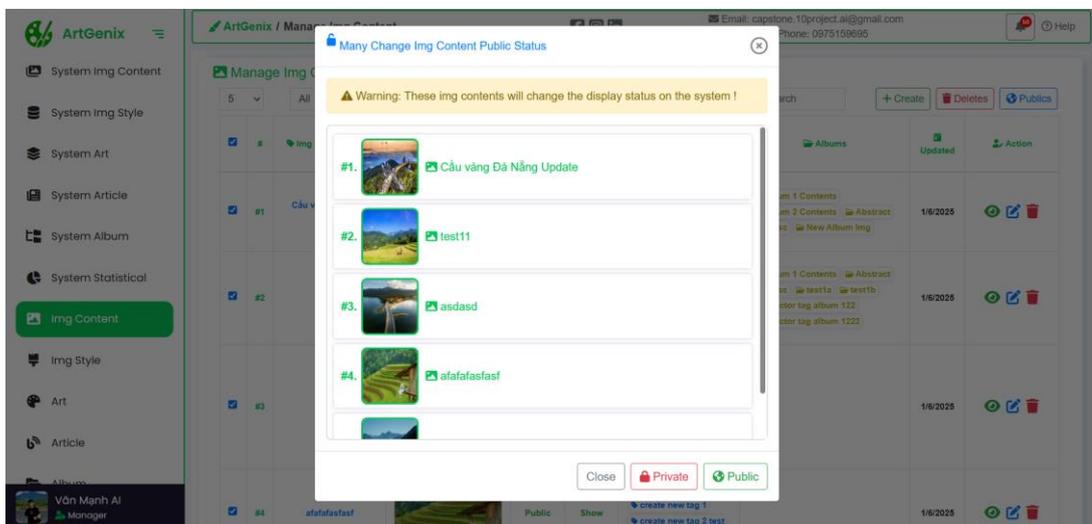
Hình 4.28 User chỉnh sửa Img Content



Hình 4.29 User xóa Img Content

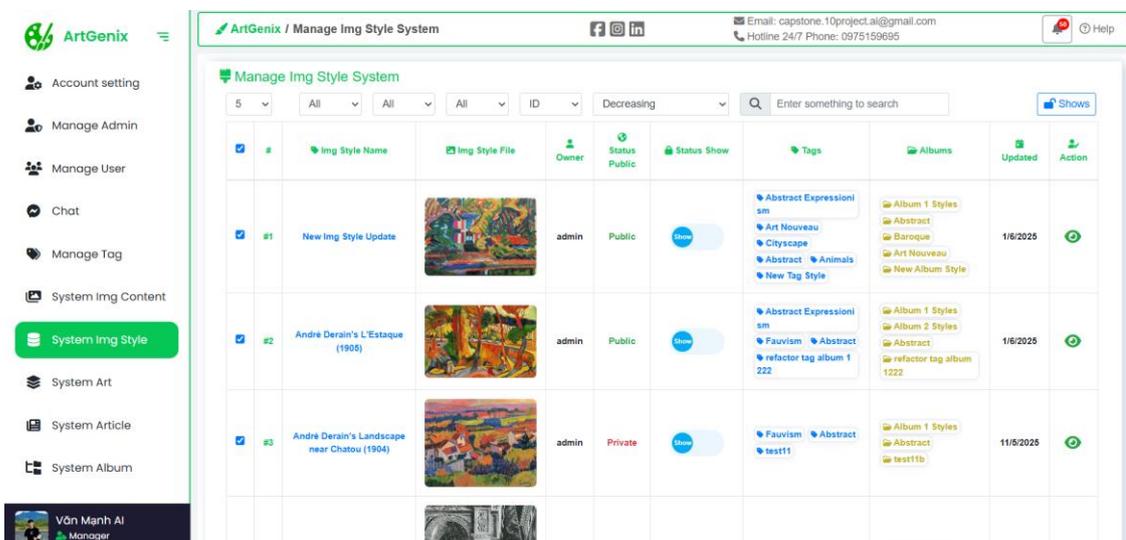


Hình 4.30 User xóa hàng loạt Img Content

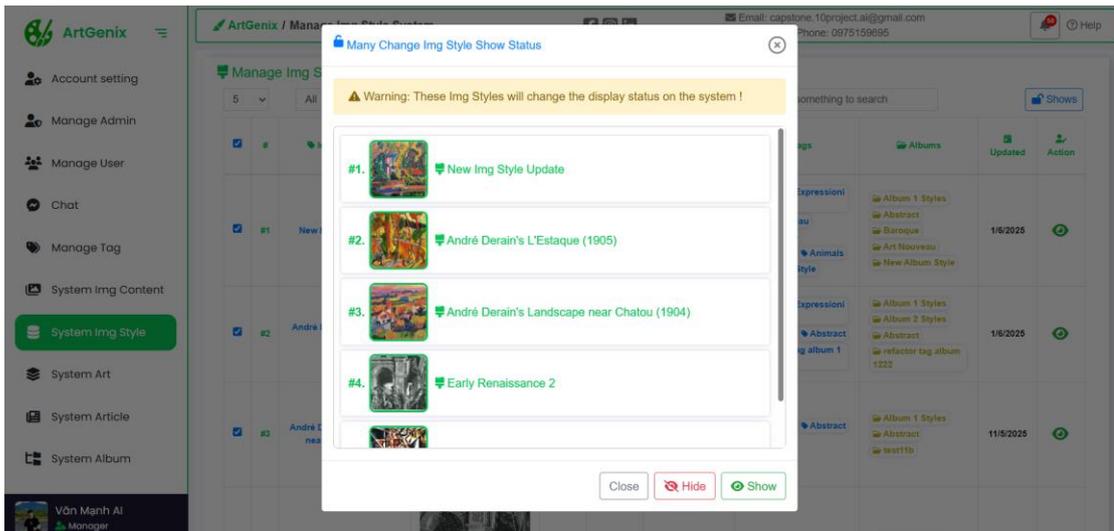


Hình 4.31 User thay đổi trạng hàng loạt trạng thái công khai Img Content

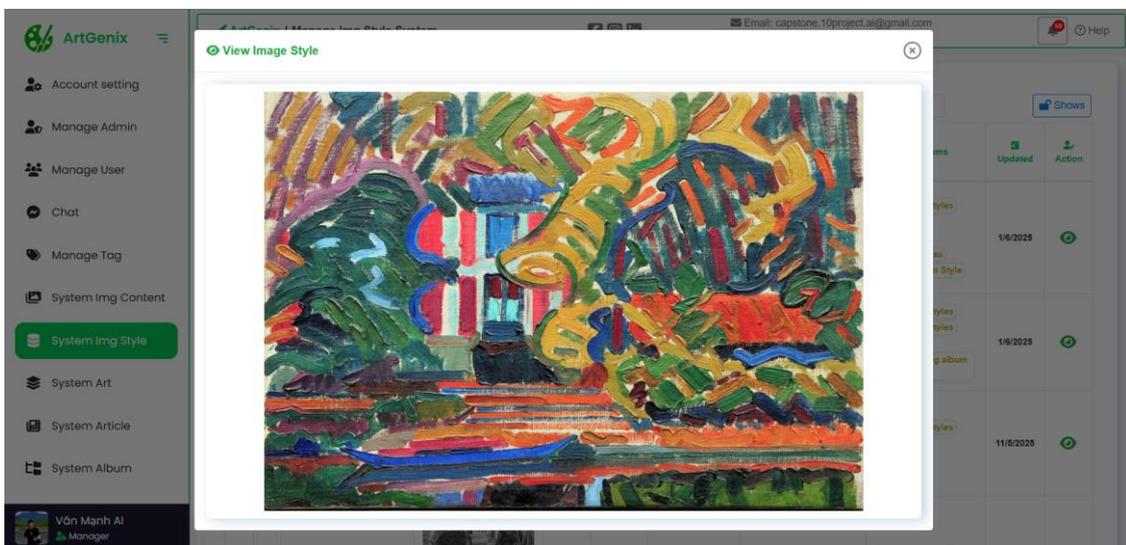
#### 4.3.6. Nhóm chức năng liên quan đến Img Style



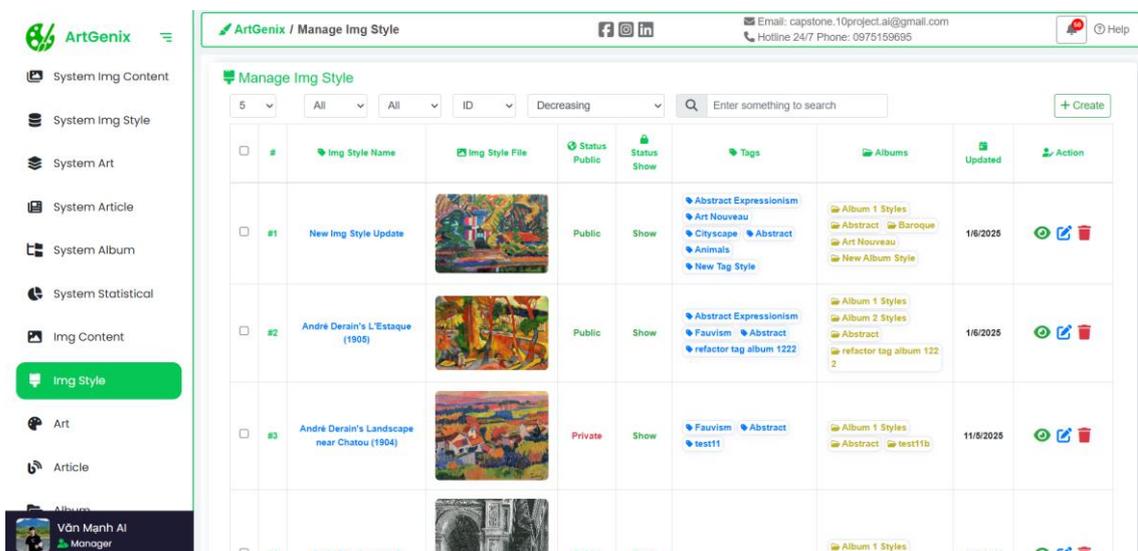
Hình 4.32 Admin quản lý Img Style



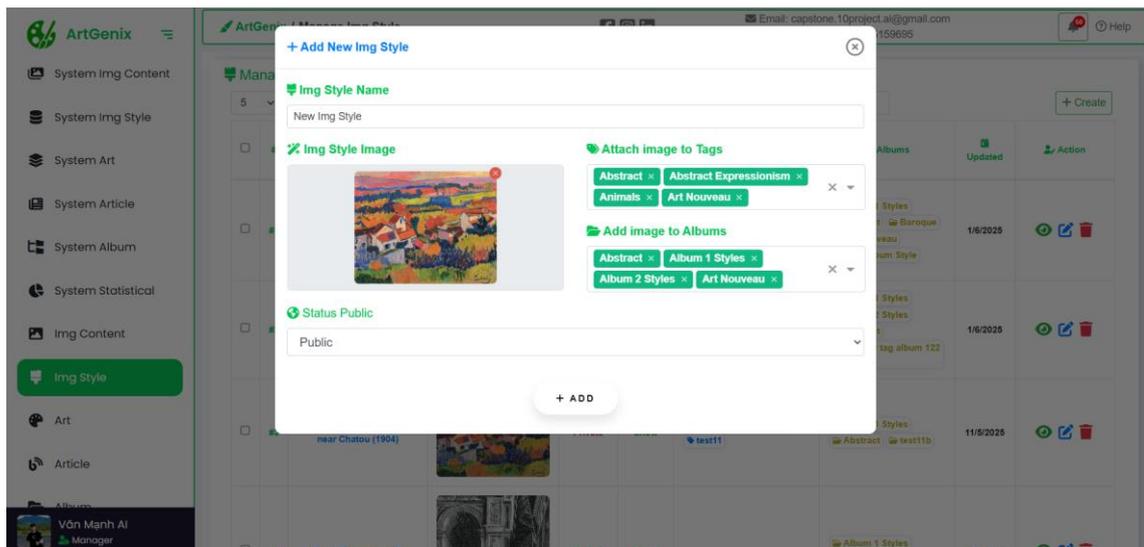
Hình 4.33 Admin thay đổi trạng thái hiển thị của Img Style



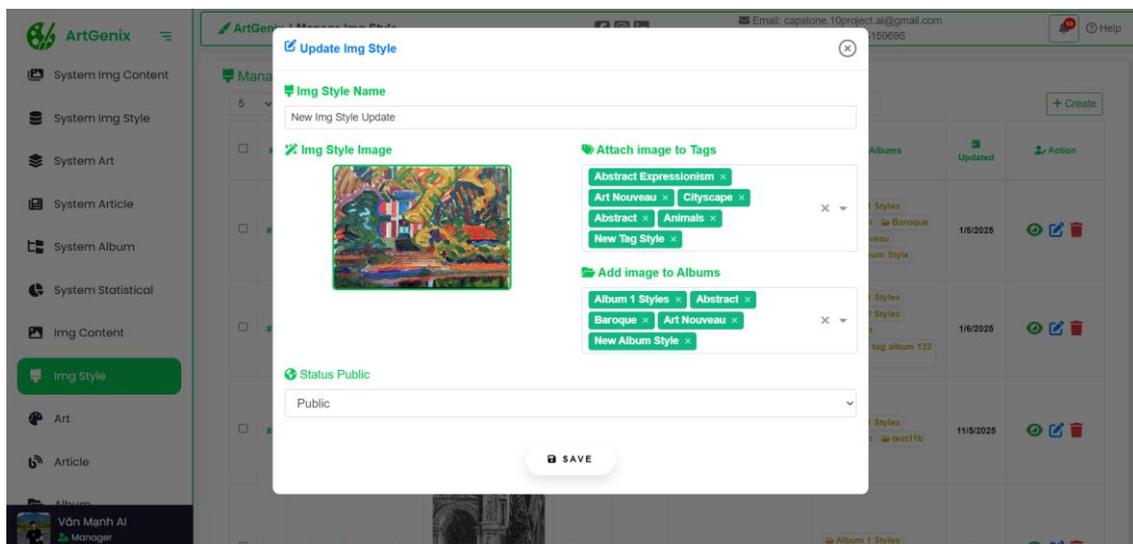
Hình 4.34 Admin xem chi tiết Img Style



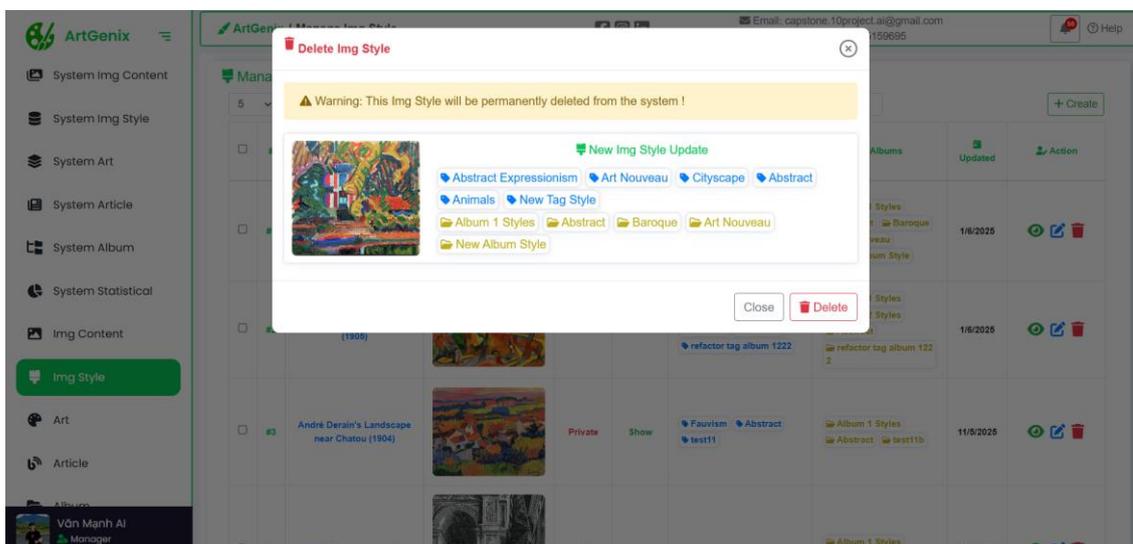
Hình 4.35 User quản lý Img Style



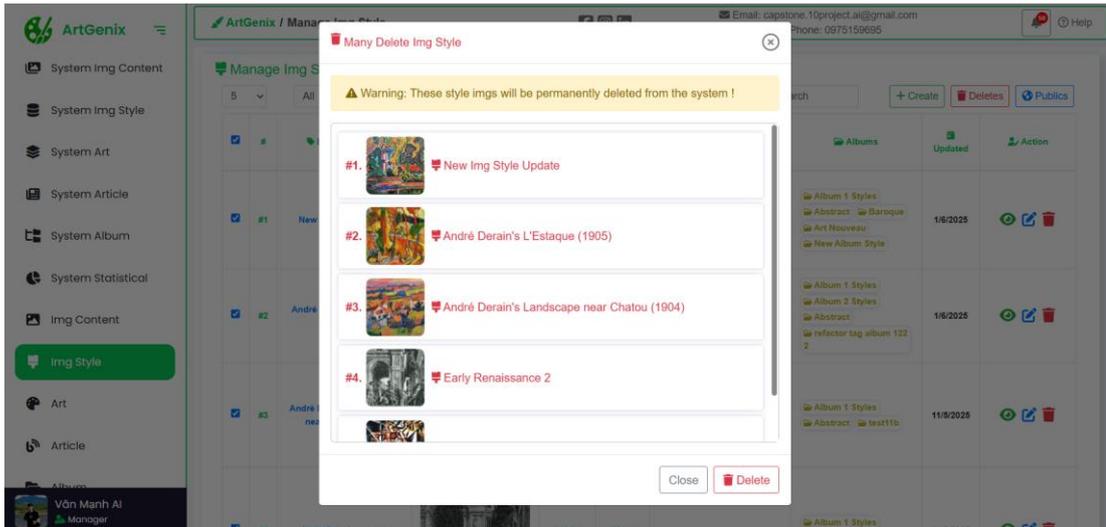
Hình 4.36 User thêm Img Style



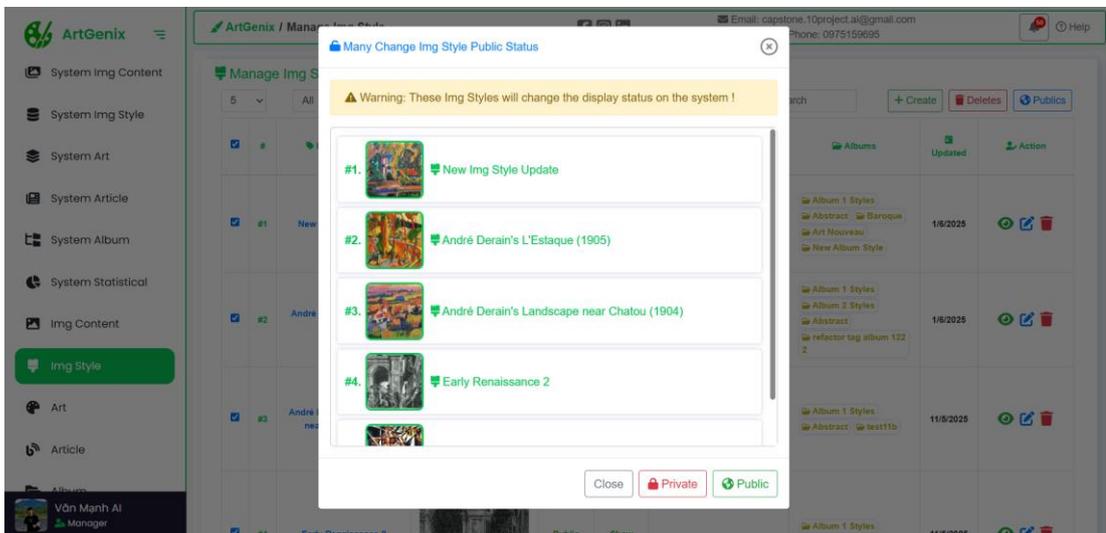
Hình 4.37 User chỉnh sửa Img Style



Hình 4.38 User xóa Img Style

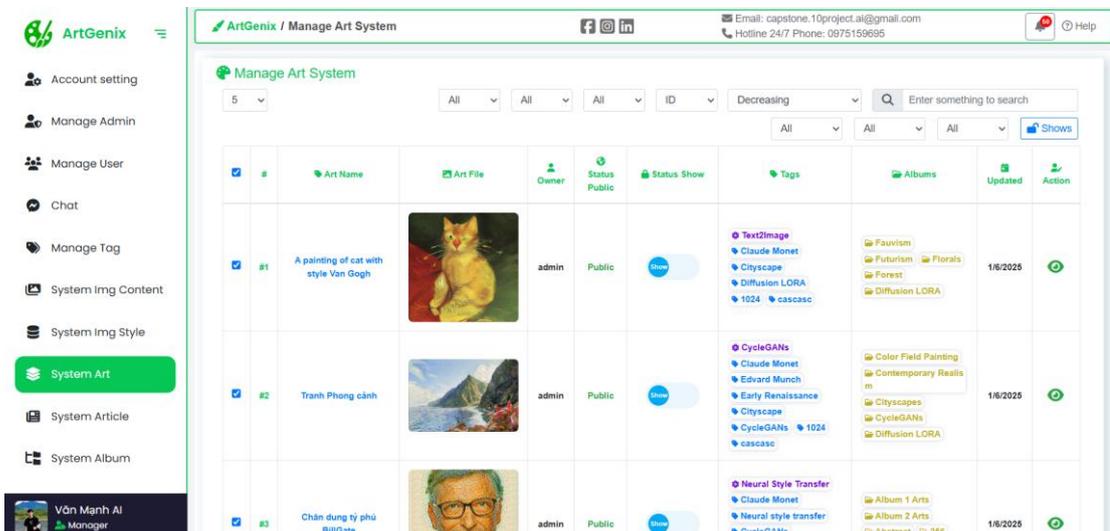


Hình 4.39 User xóa hàng loạt Img Style

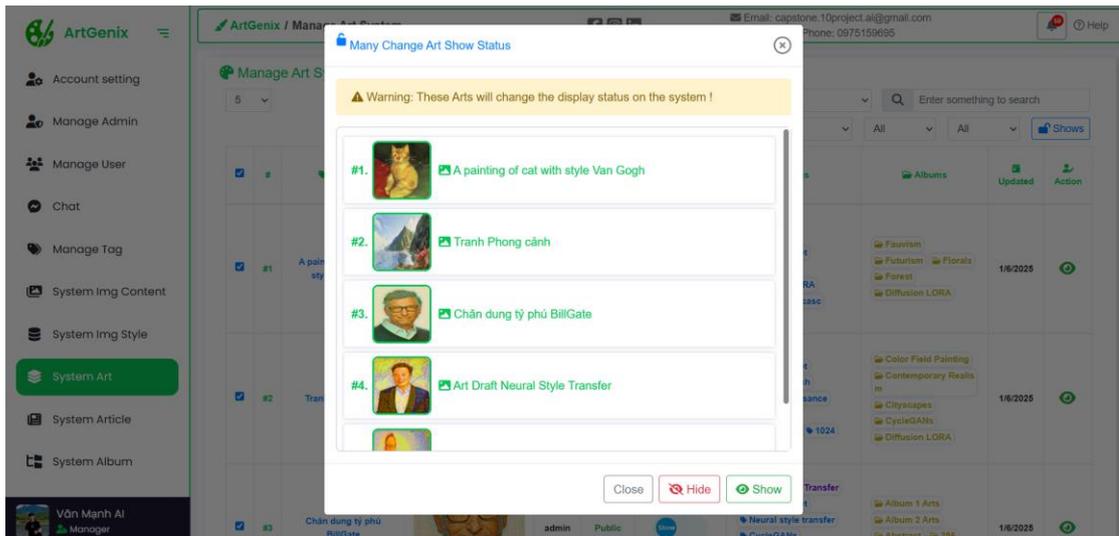


Hình 4.40 User thay đổi trạng hàng loạt trạng thái công khai Img Style

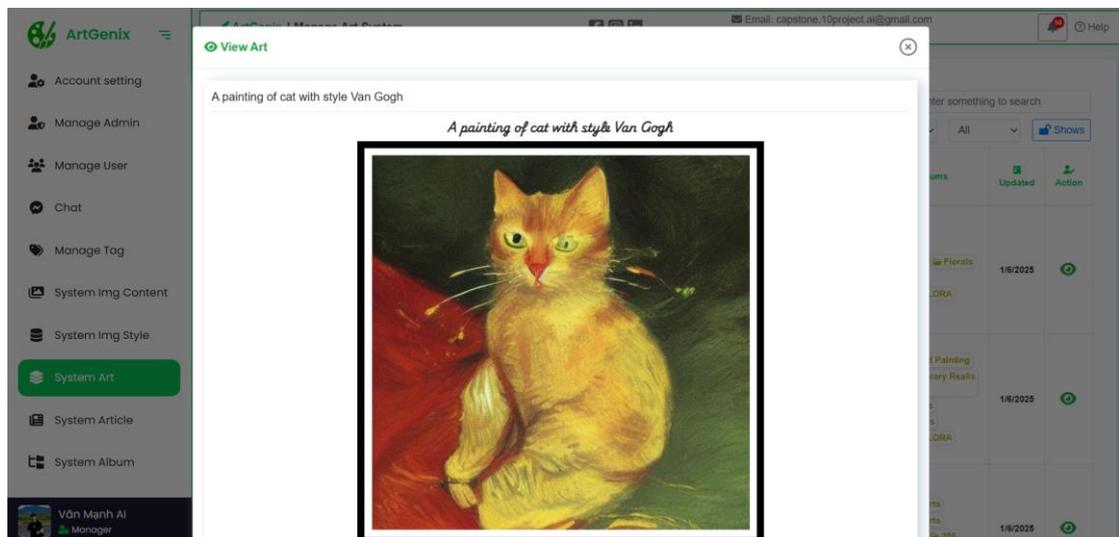
#### 4.3.7. Nhóm chức năng liên quan đến Art



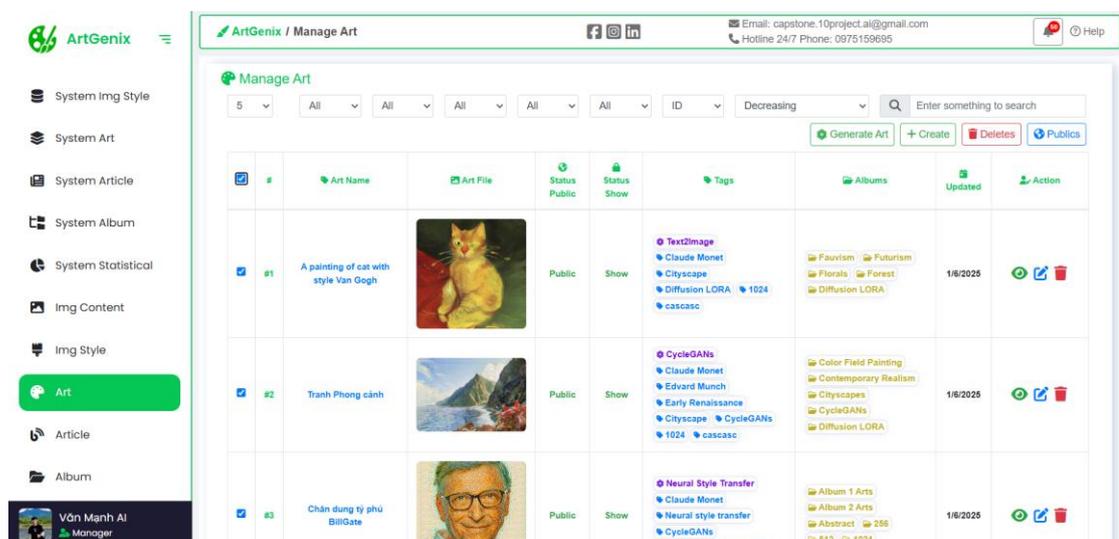
Hình 4.41 Admin quản lý Art



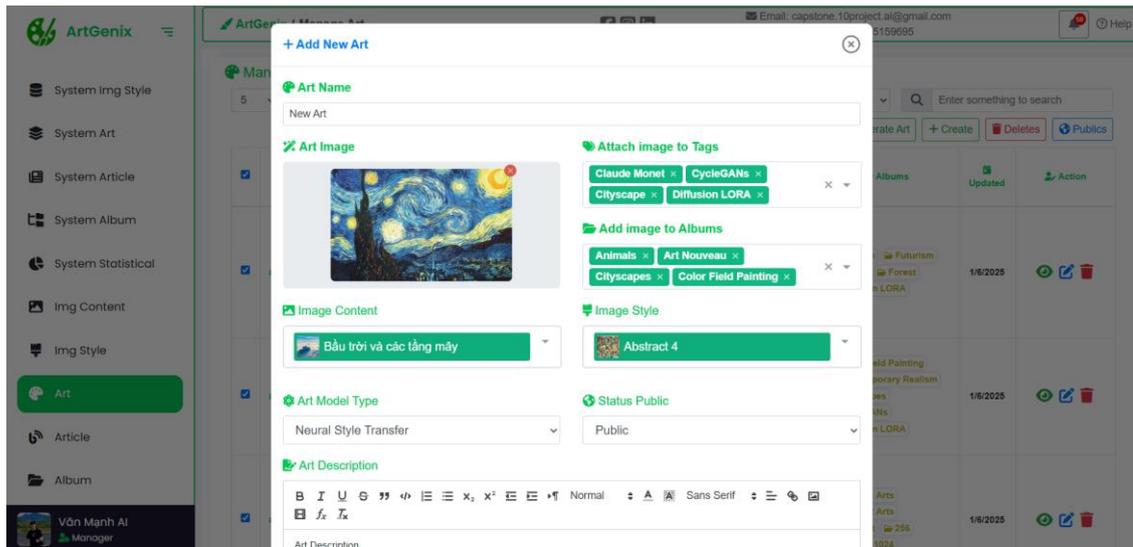
Hình 4.42 Admin thay đổi trạng thái hiển thị của Art



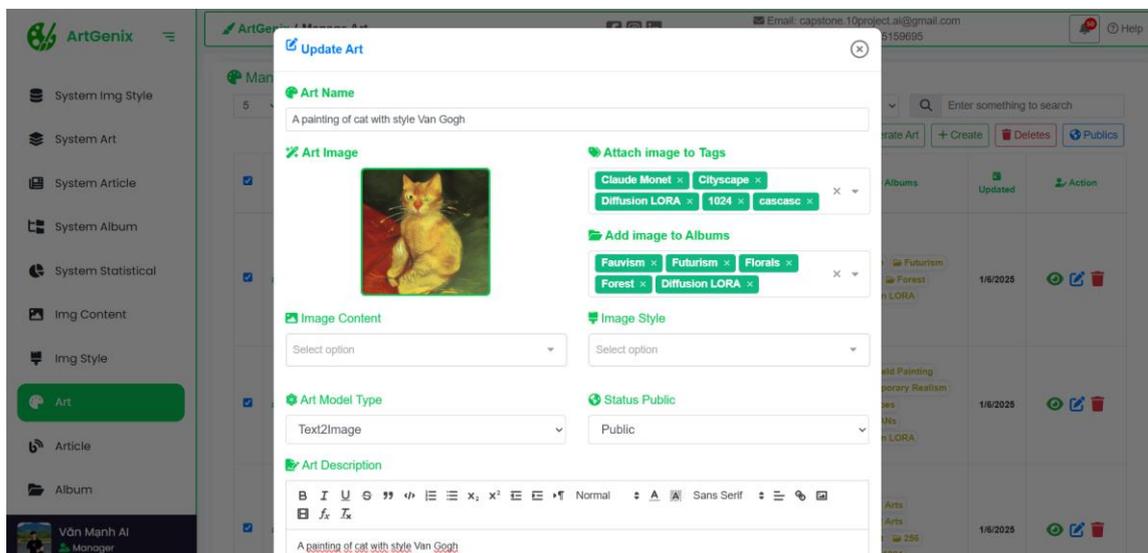
Hình 4.43 Admin xem chi tiết Art



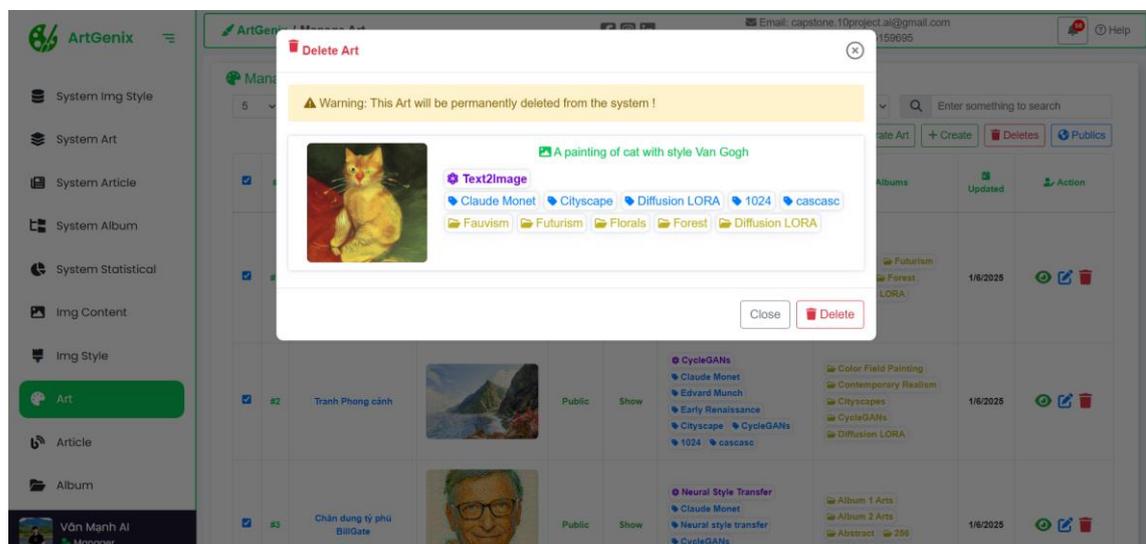
Hình 4.44 User quản lý Art



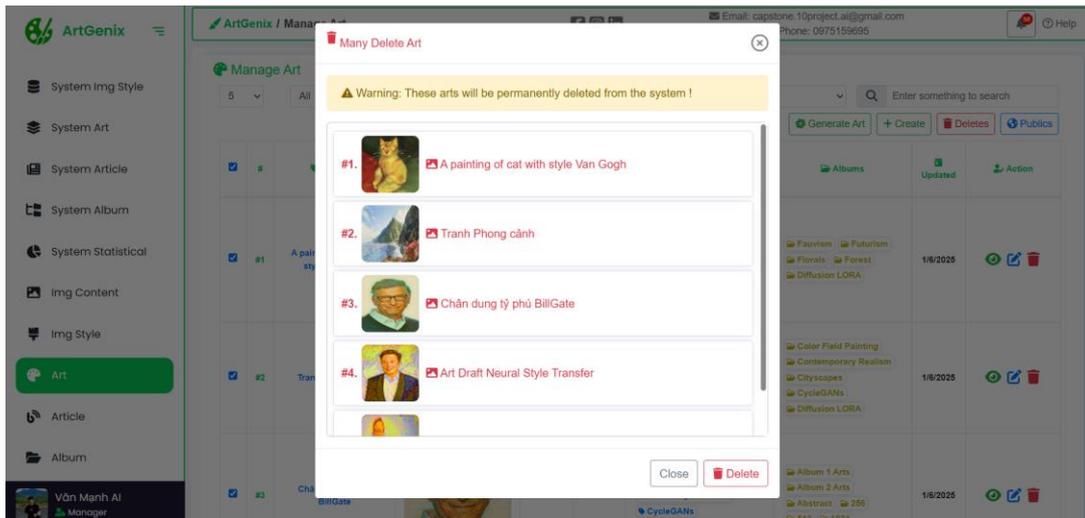
Hình 4.45 User thêm Art



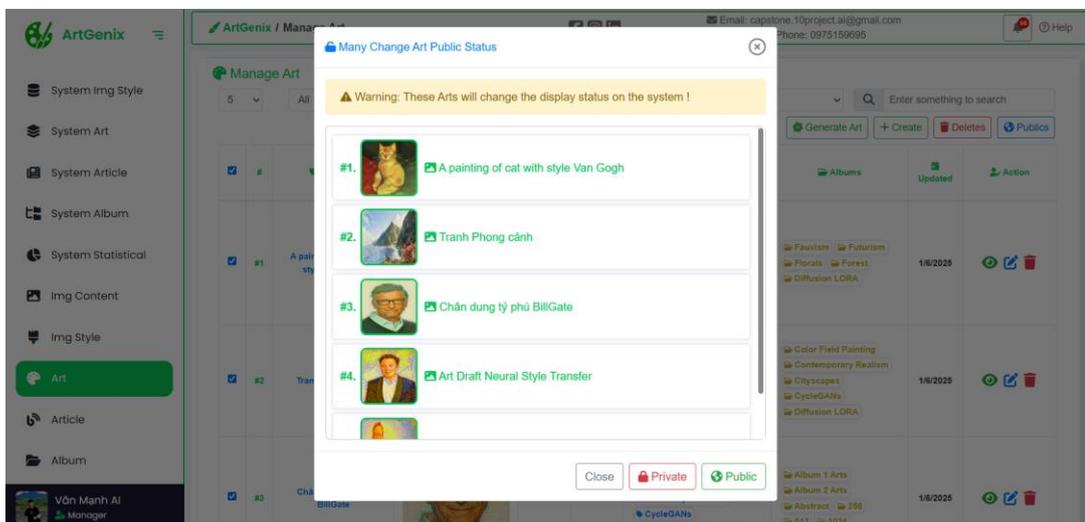
Hình 4.46 User chỉnh sửa Art



Hình 4.47 User xóa Art

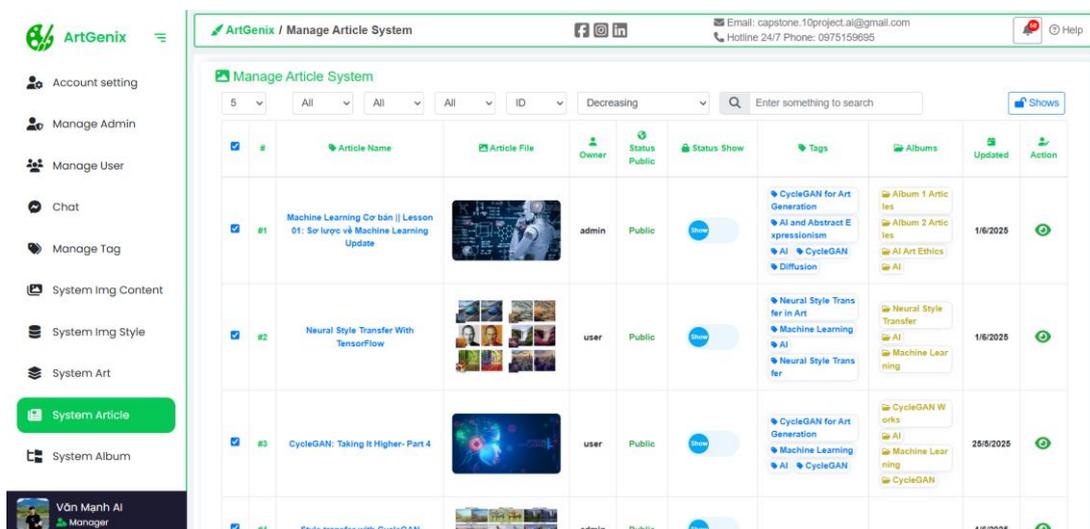


Hình 4.48 User xóa hàng loạt Art

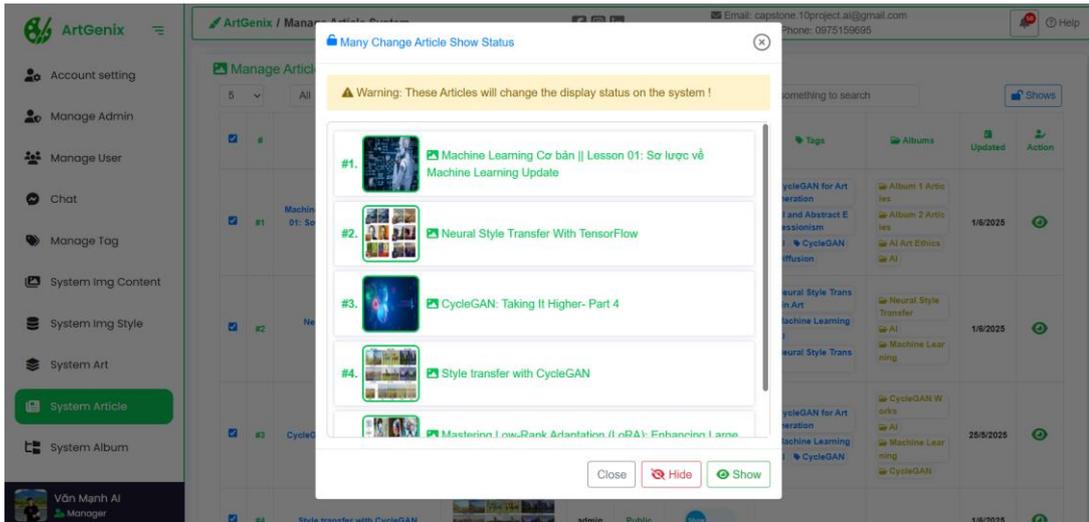


Hình 4.49 User thay đổi trạng hàng loạt trạng thái công khai Art

#### 4.3.8. Nhóm chức năng liên quan đến Article



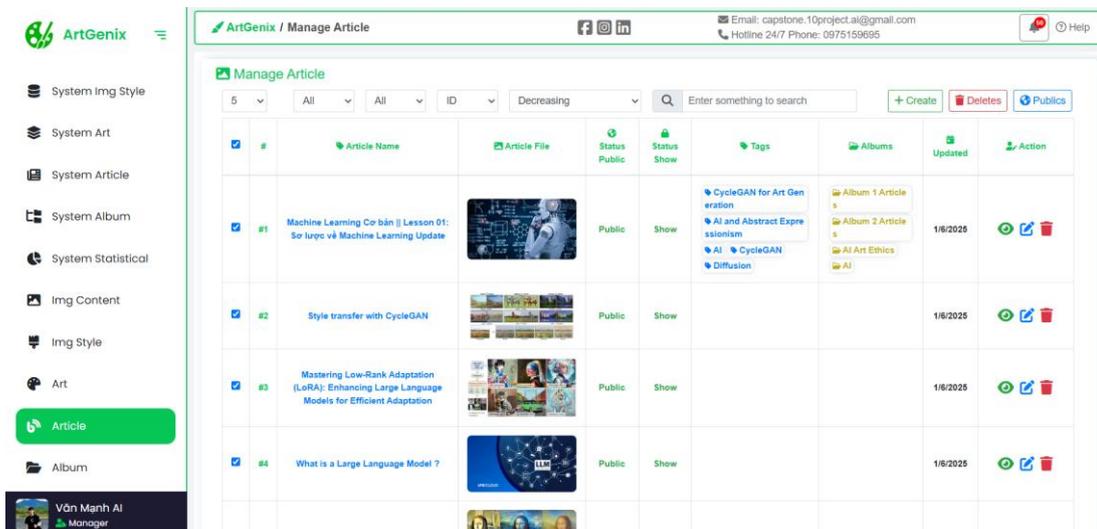
Hình 4.50 Admin quản lý Article



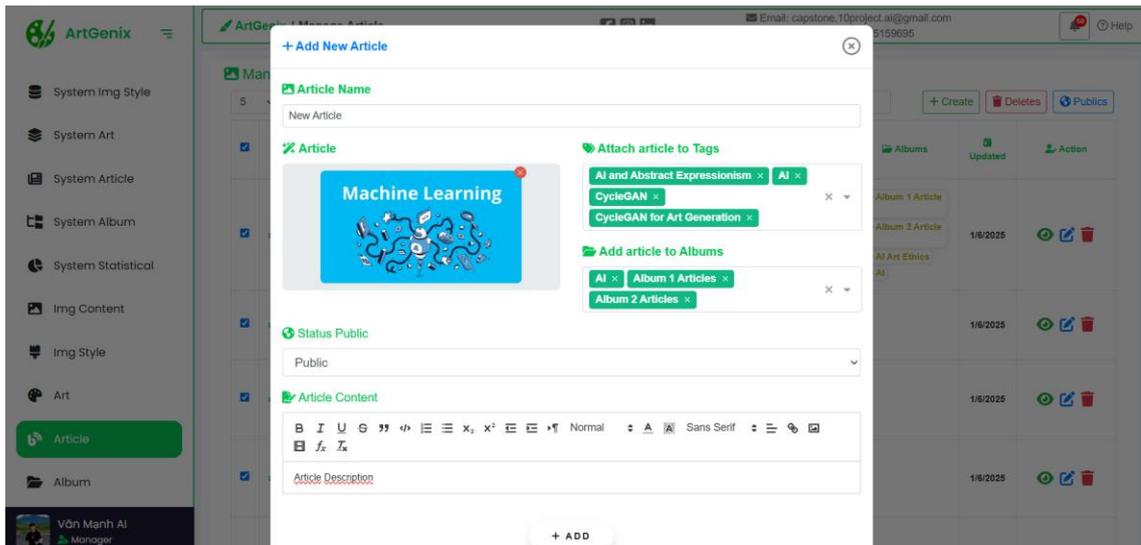
Hình 4.51 Admin thay đổi trạng thái hiển thị của Article



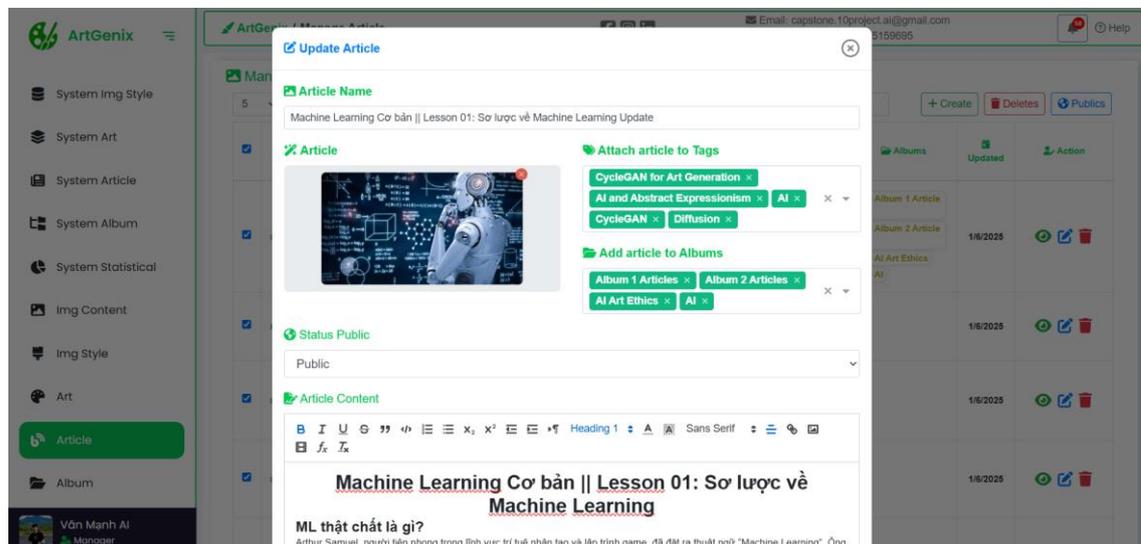
Hình 4.52 Admin xem chi tiết Article



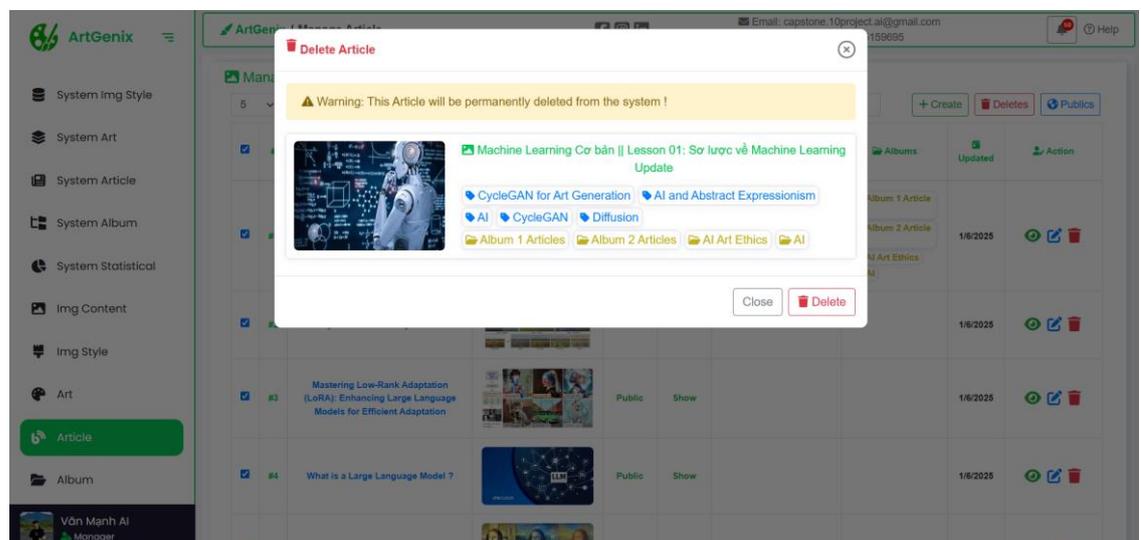
Hình 4.53 User quản lý Article



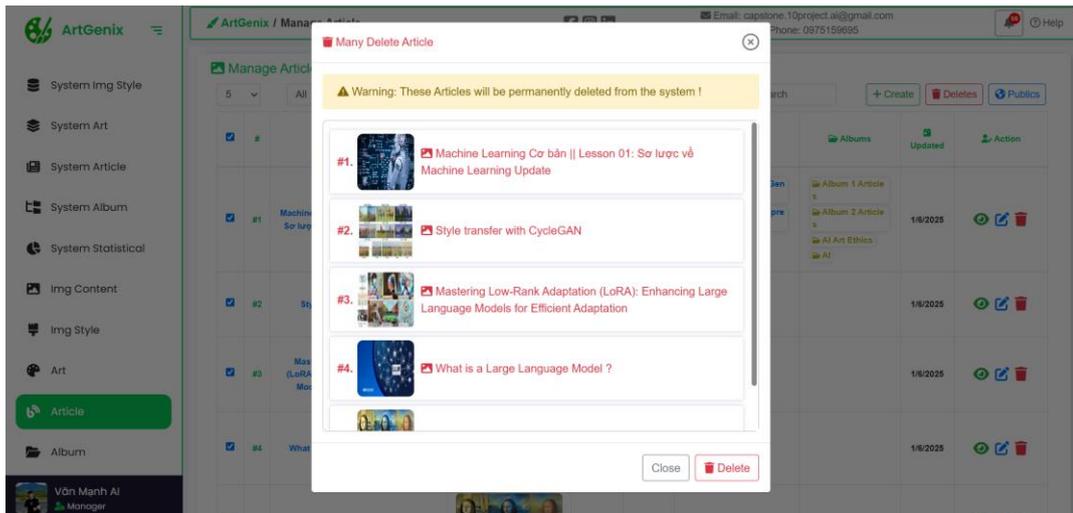
Hình 4.54 User thêm Article



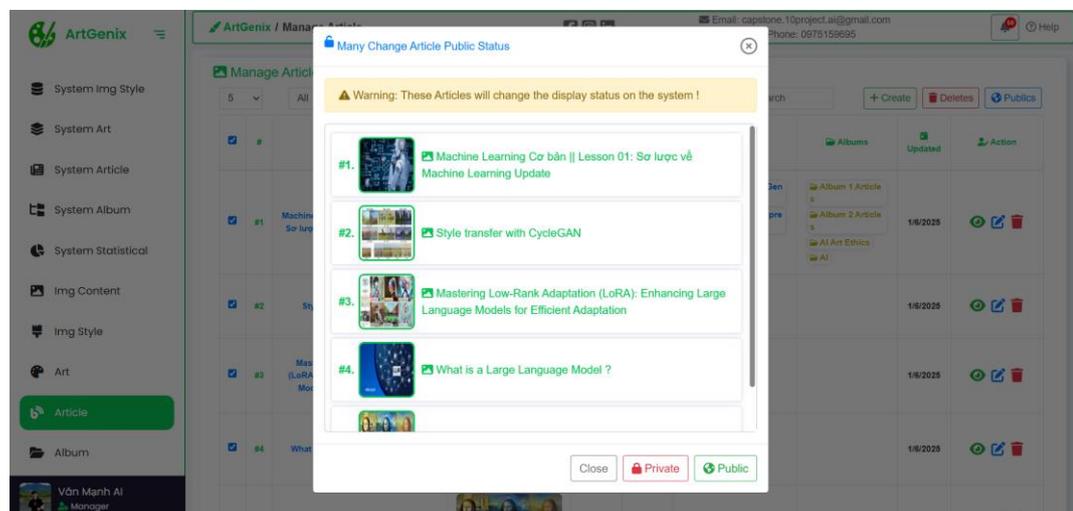
Hình 4.55 User chỉnh sửa Article



Hình 4.56 User xóa Article

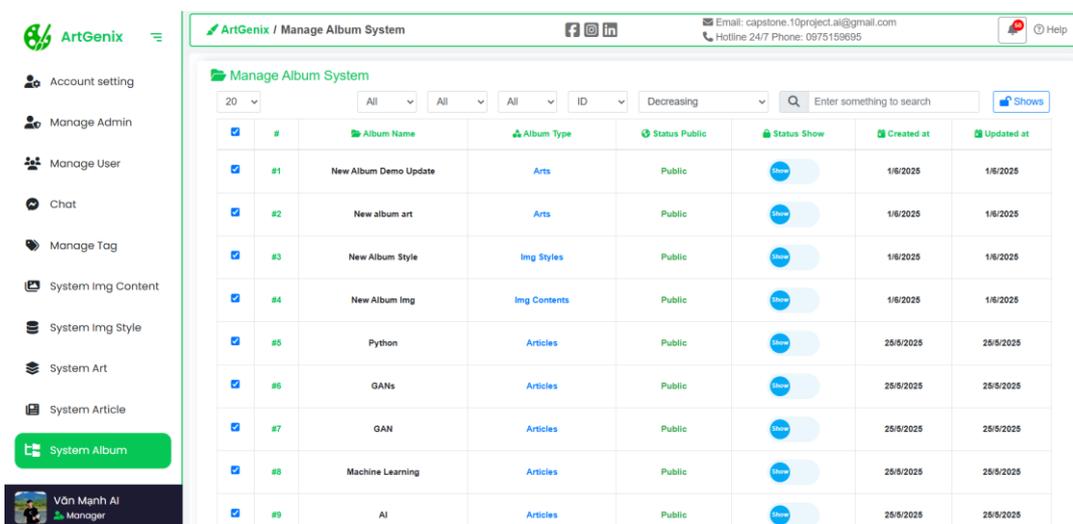


Hình 4.57 User xóa hàng loạt Article

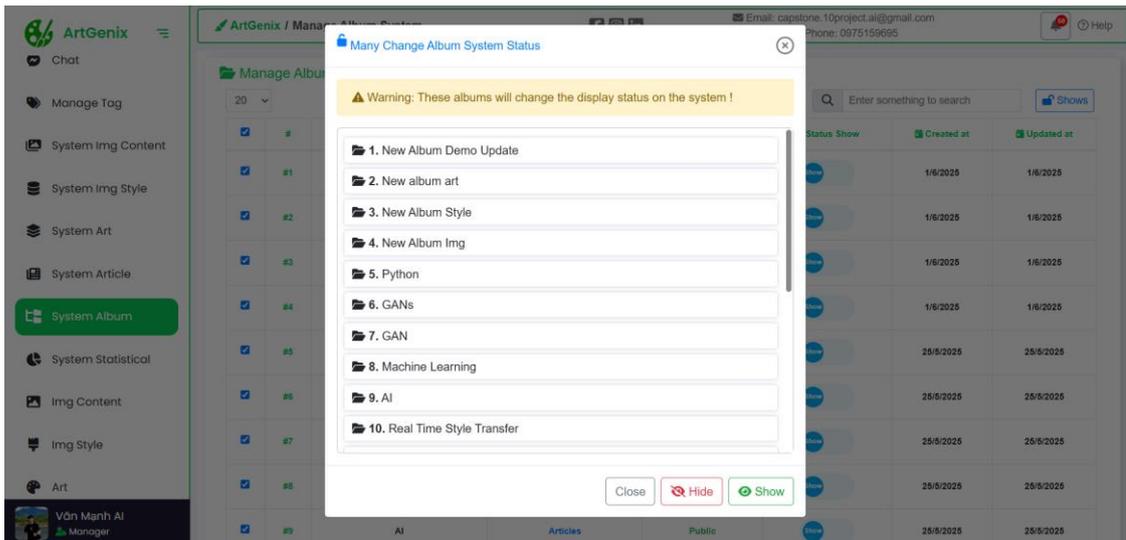


Hình 4.58 User thay đổi trạng hàng loạt trạng thái công khai Article

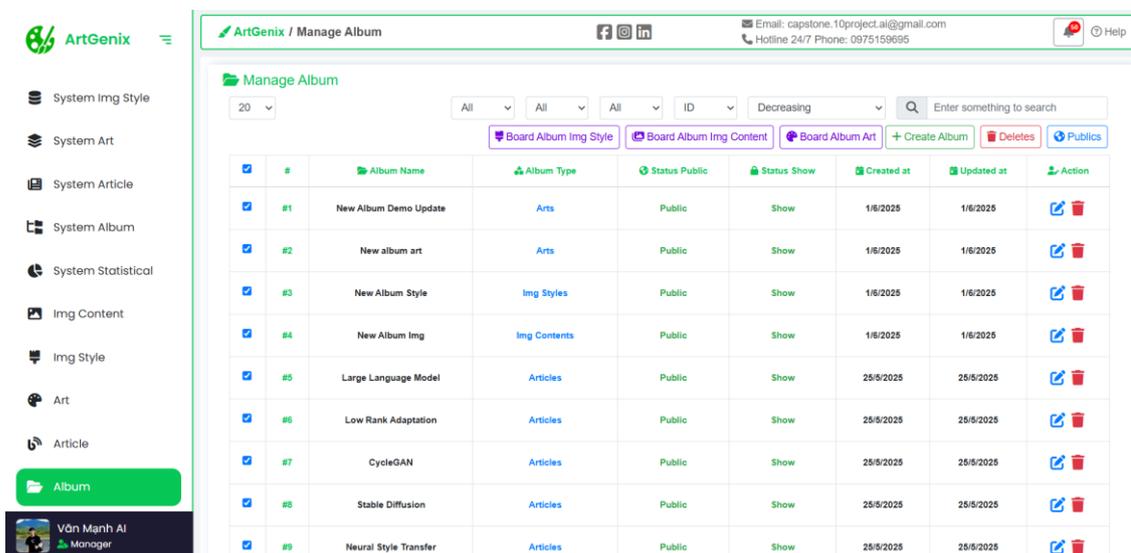
#### 4.3.9. Nhóm chức năng liên quan đến Album



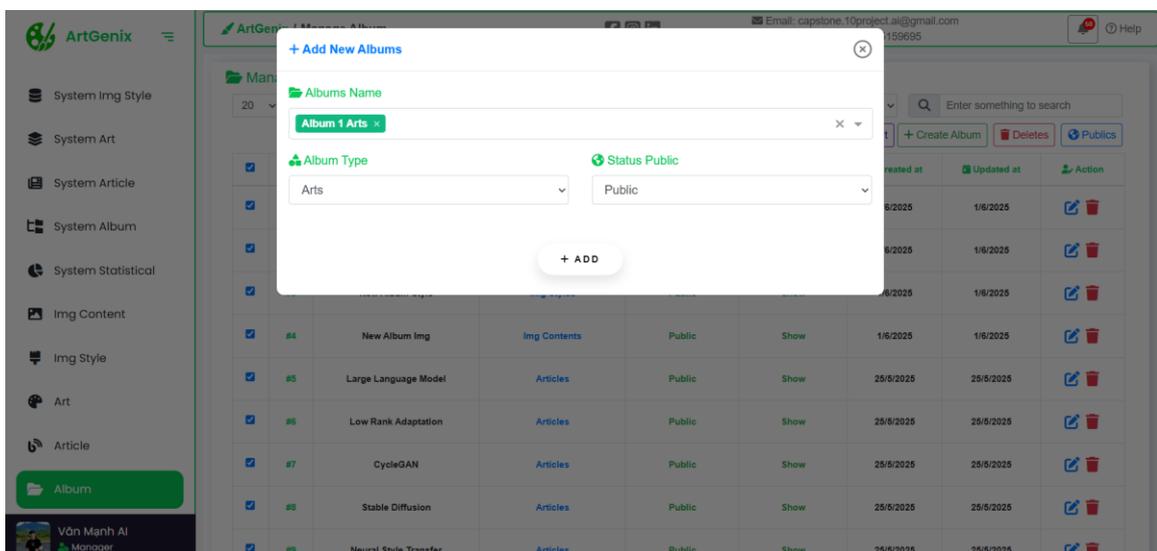
Hình 4.59 Admin quản lý Album



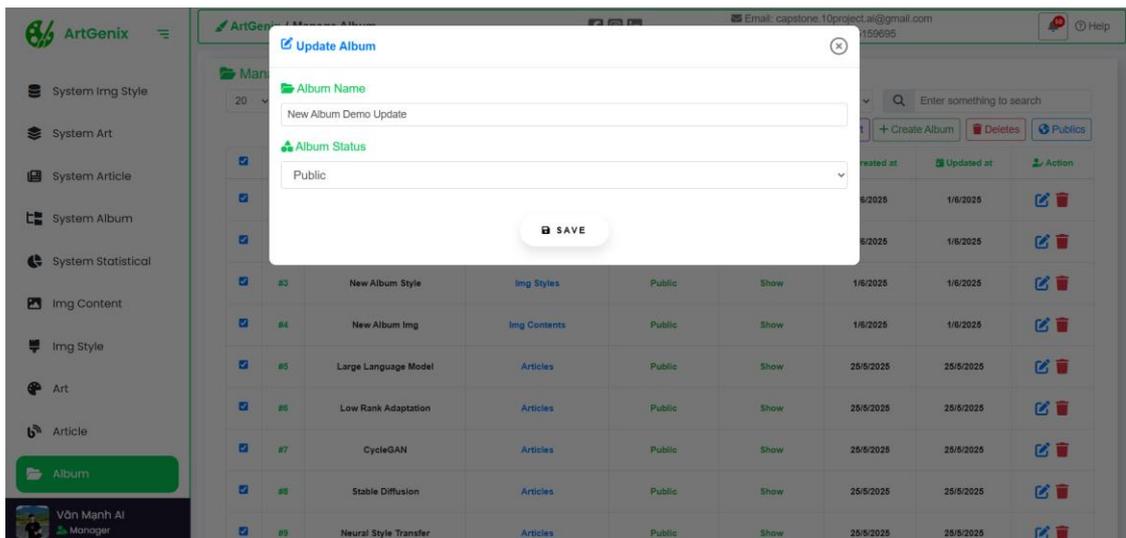
Hình 4.60 Admin thay đổi trạng thái hiển thị của Album



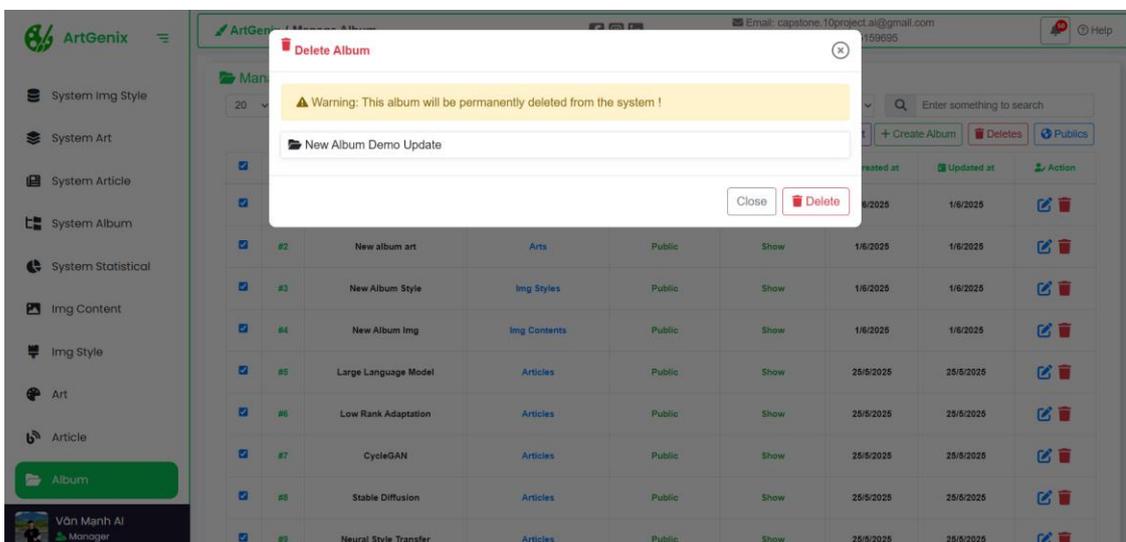
Hình 4.61 User quản lý Album



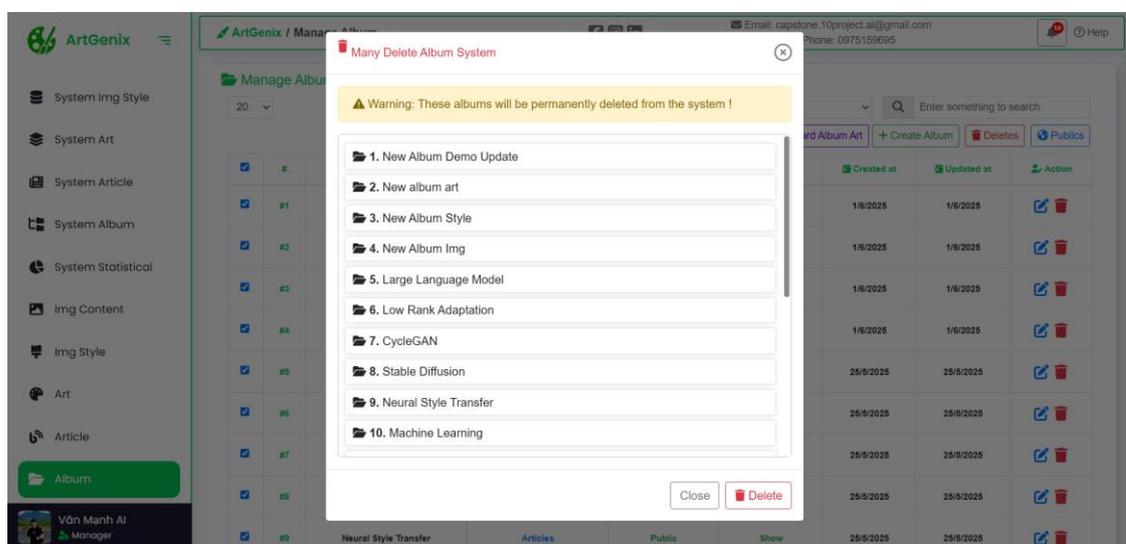
Hình 4.62 User thêm Album



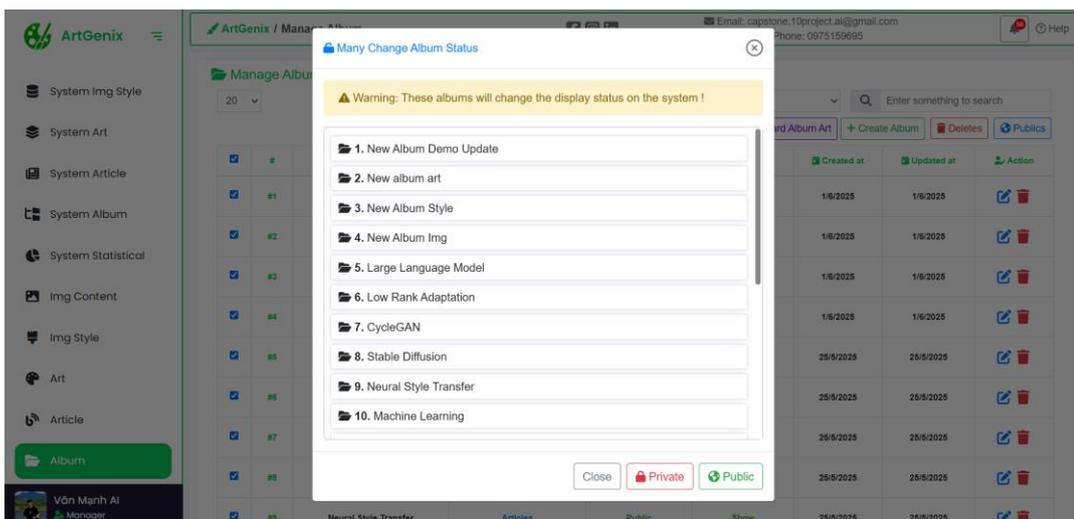
Hình 4.63 User chỉnh sửa Album



Hình 4.64 User xóa Album

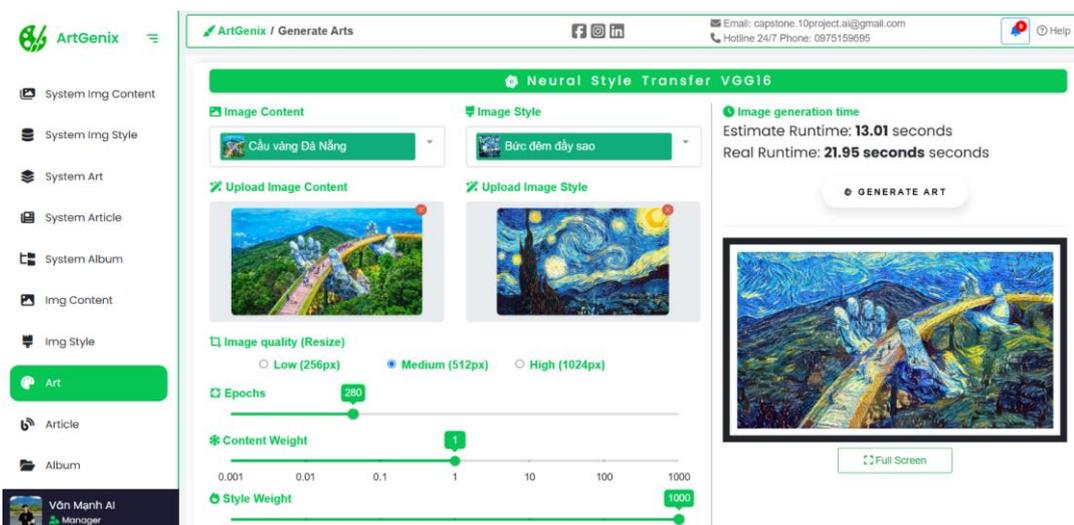


Hình 4.65 User xóa hàng loạt Album

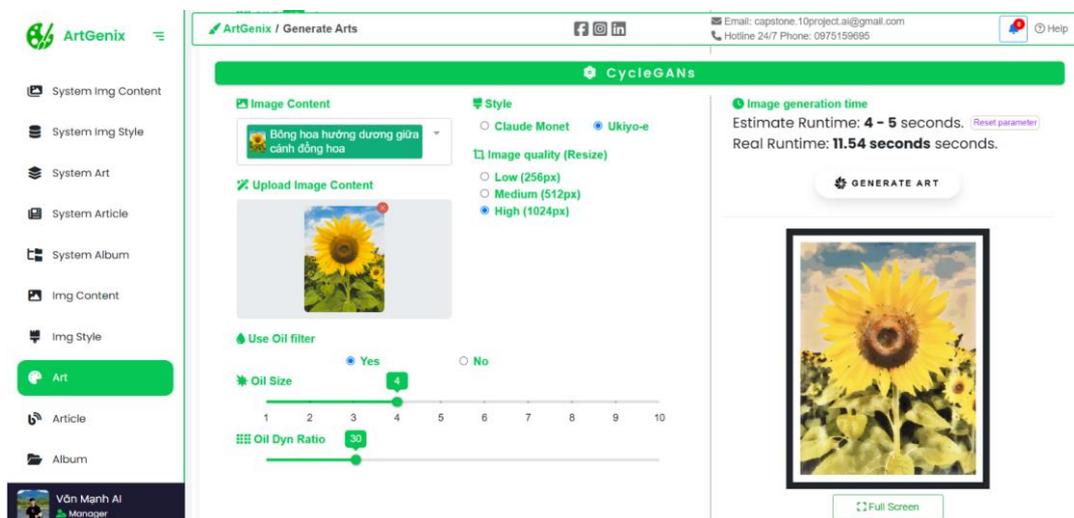


Hình 4.66 User thay đổi hàng loạt trạng thái công khai Album

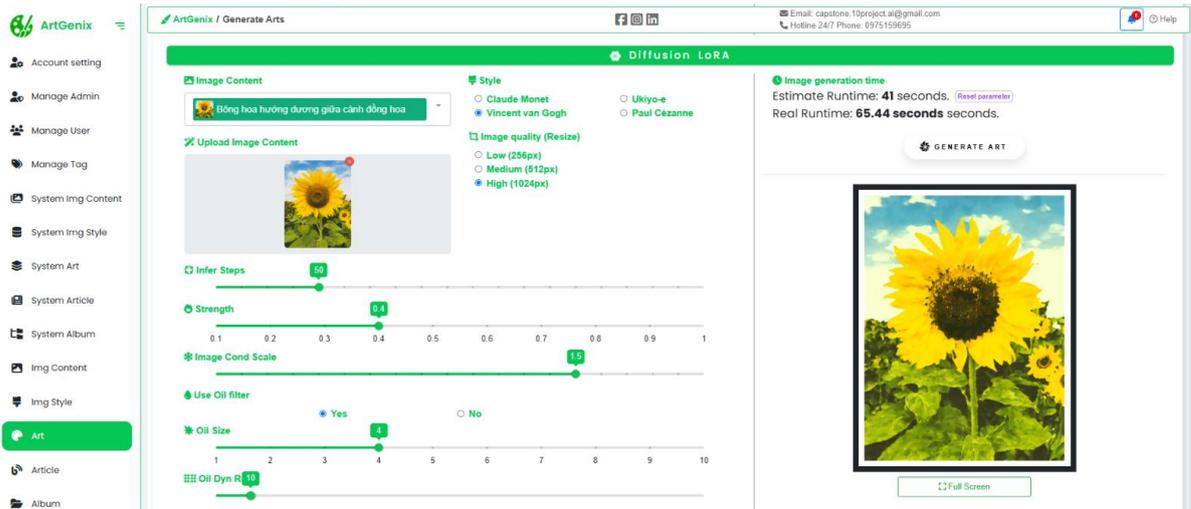
#### 4.3.10. Nhóm chức năng Tạo sinh ảnh nghệ thuật



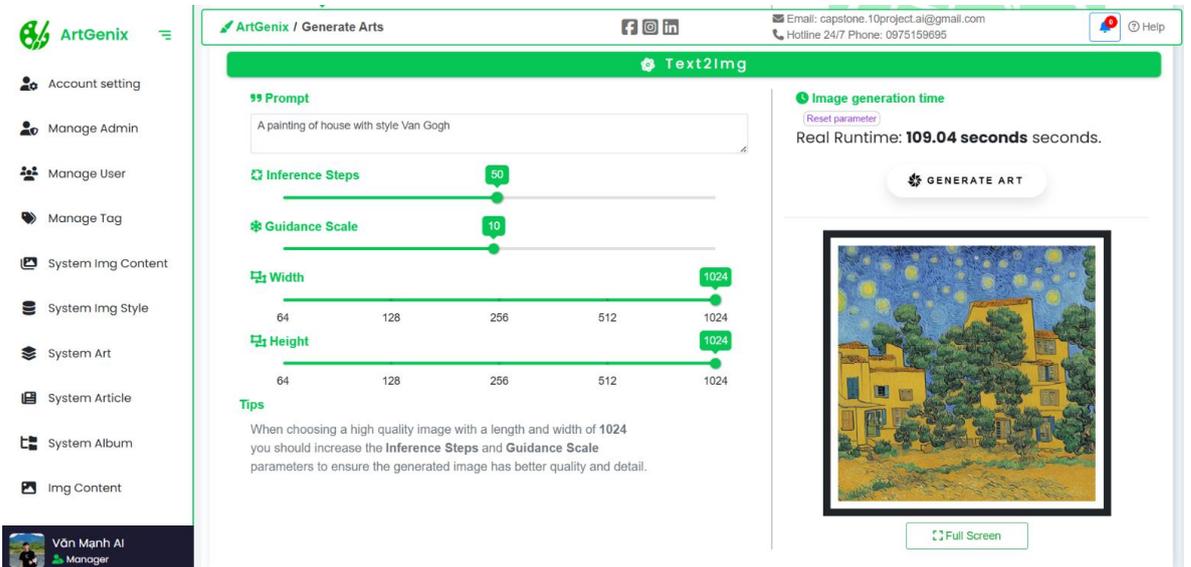
Hình 4.67 User tạo sinh ảnh nghệ thuật sử dụng Neural Style Transfer



Hình 4.68 User tạo sinh ảnh nghệ thuật sử dụng CycleGAN

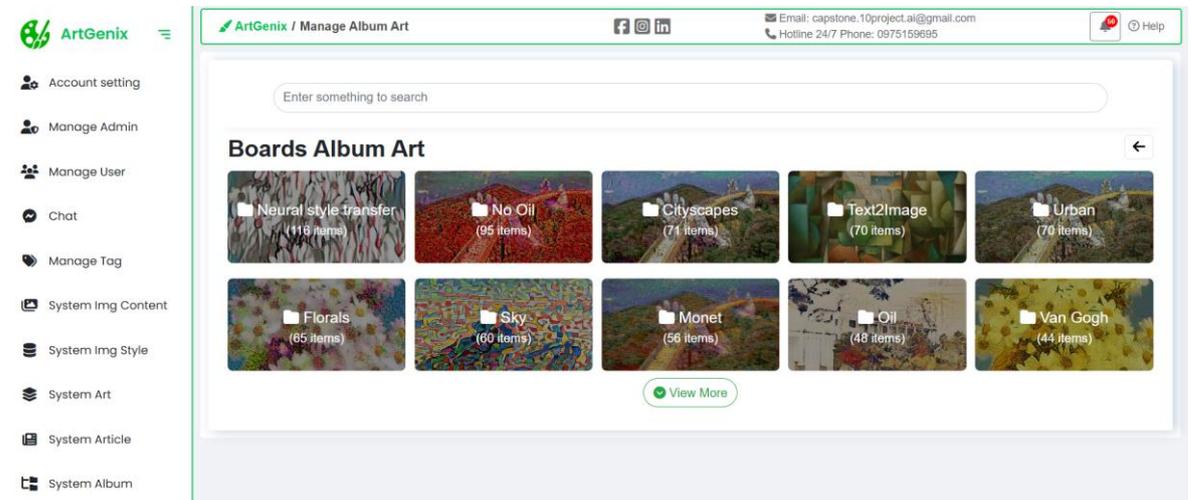


Hình 4.69 User tạo sinh ảnh nghệ thuật sử dụng Diffusion LoRA

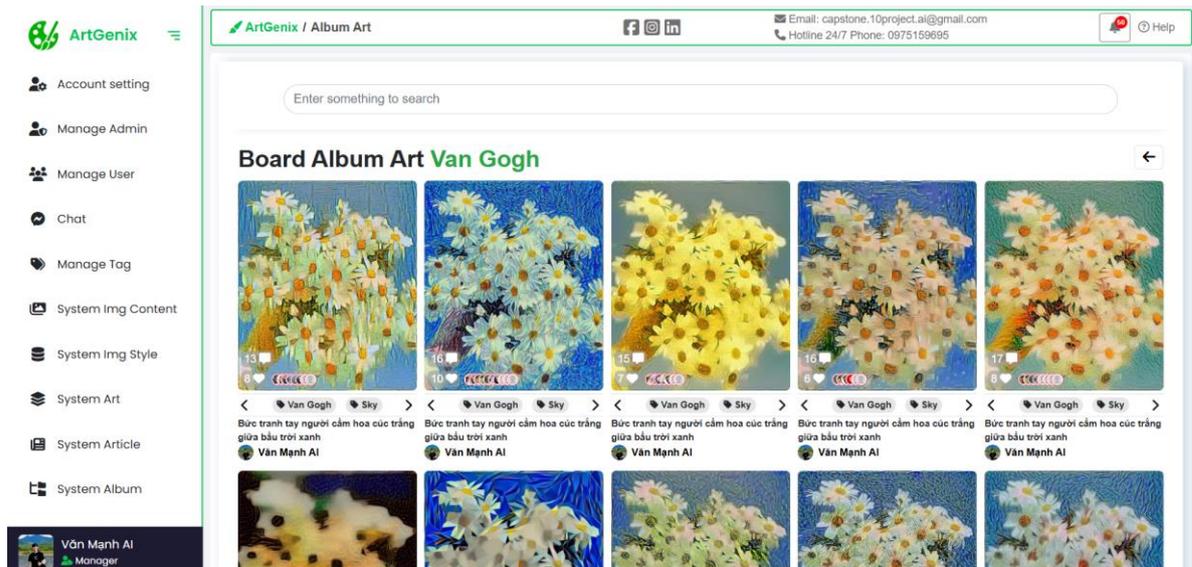


Hình 4.70 User tạo sinh ảnh nghệ thuật sử dụng Text2Img

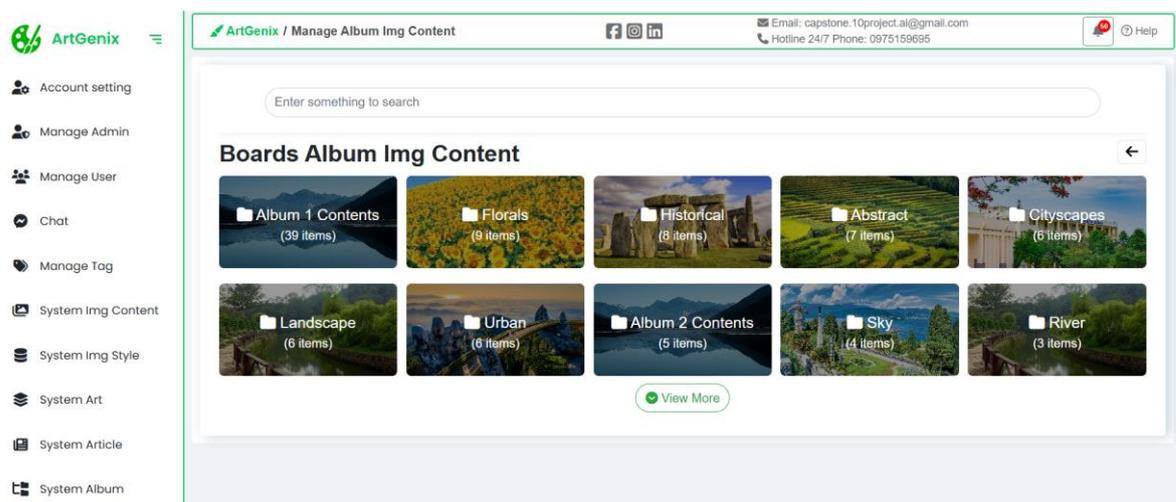
#### 4.3.11. Nhóm chức năng Album Art, Img Content, Img Style



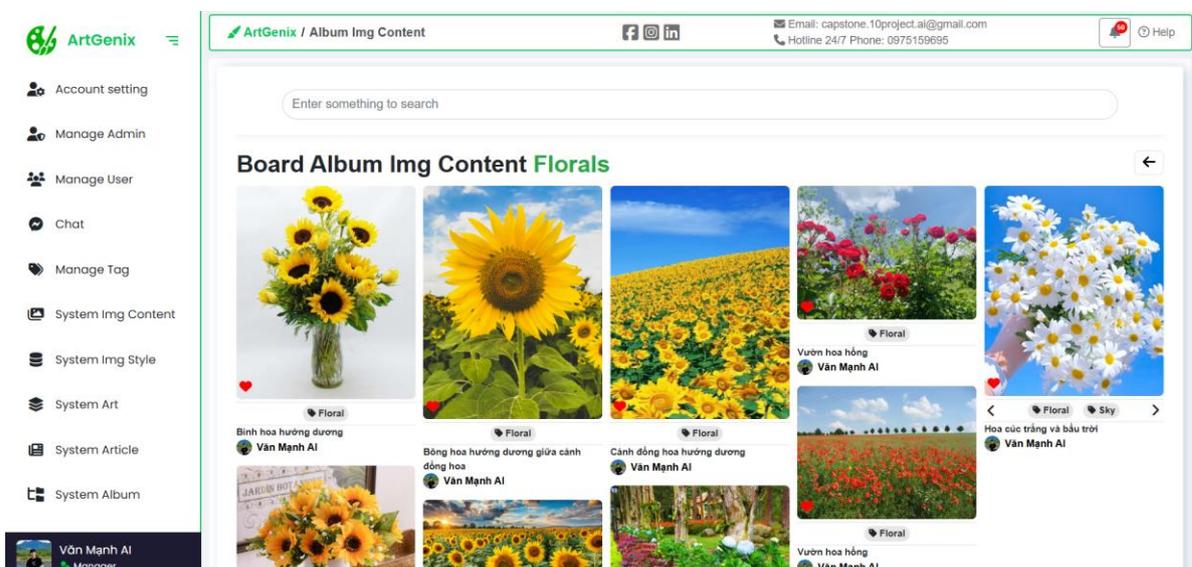
Hình 4.71 User quản lý Album Art



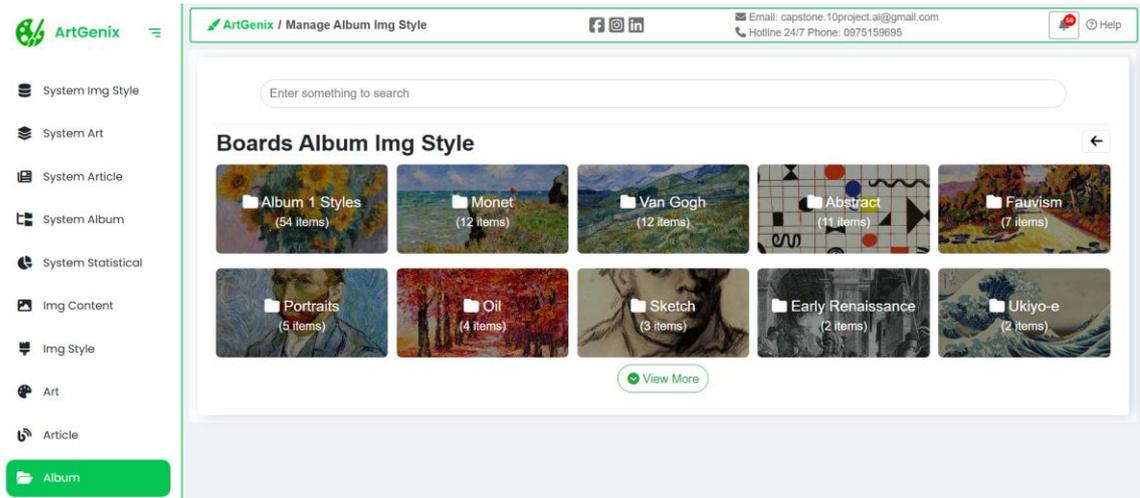
Hình 4.72 User xem Album Art



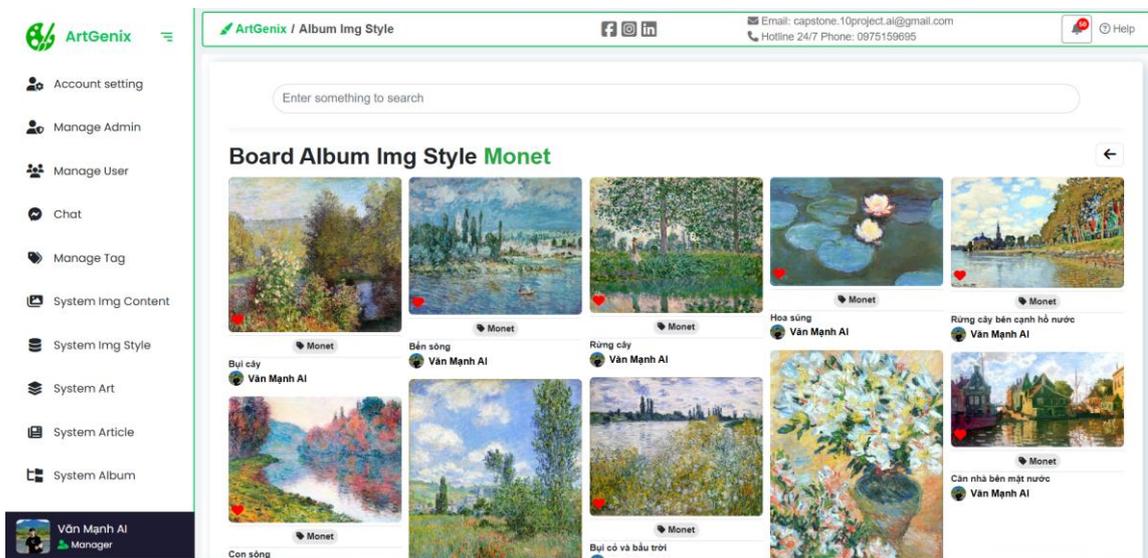
Hình 4.73 User quản lý Album Img Content



Hình 4.74 User xem Album Img Content

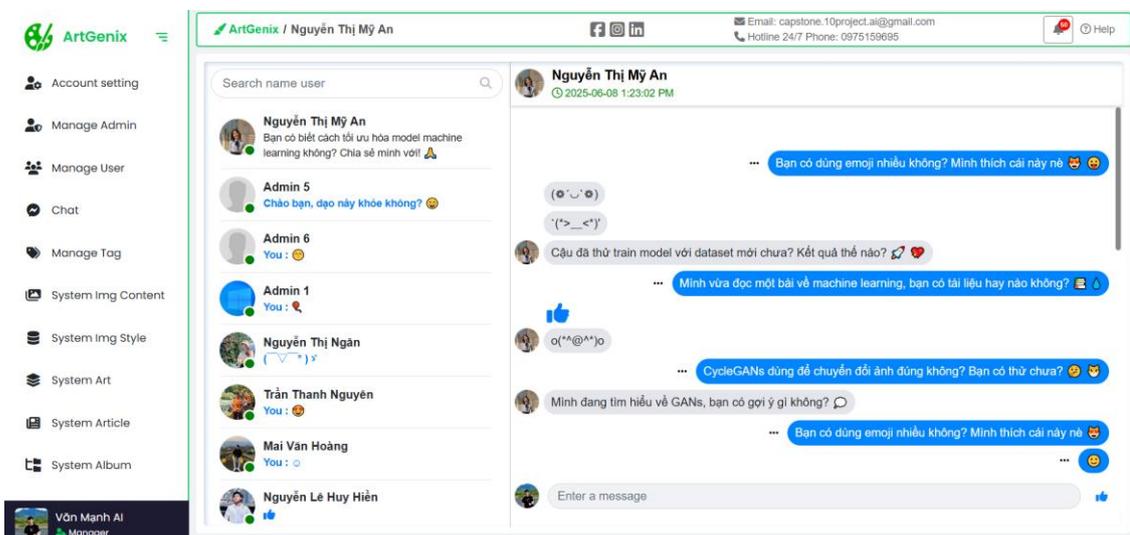


Hình 4.75 User quản lý Album Img Style

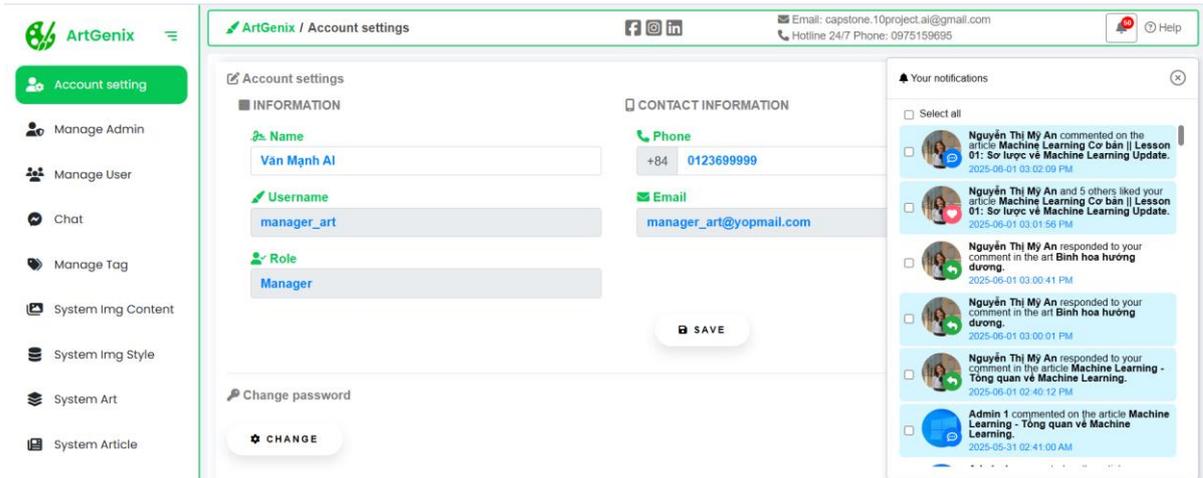


Hình 4.76 User xem Album Img Style

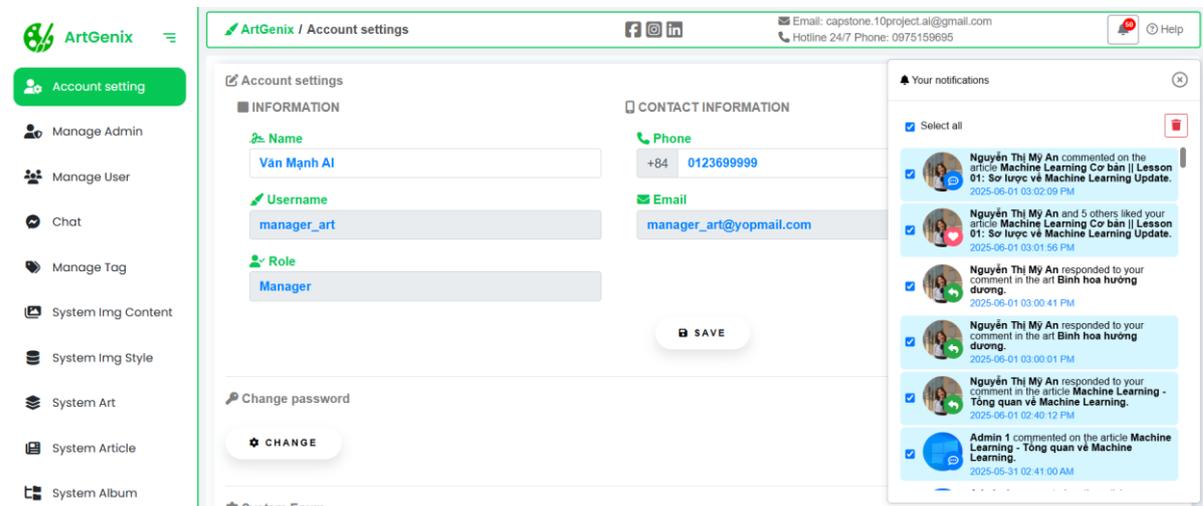
#### 4.3.12. Nhóm chức năng Nhắn tin, quản lý thông báo



Hình 4.77 User nhắn tin, trò chuyện trực tuyến

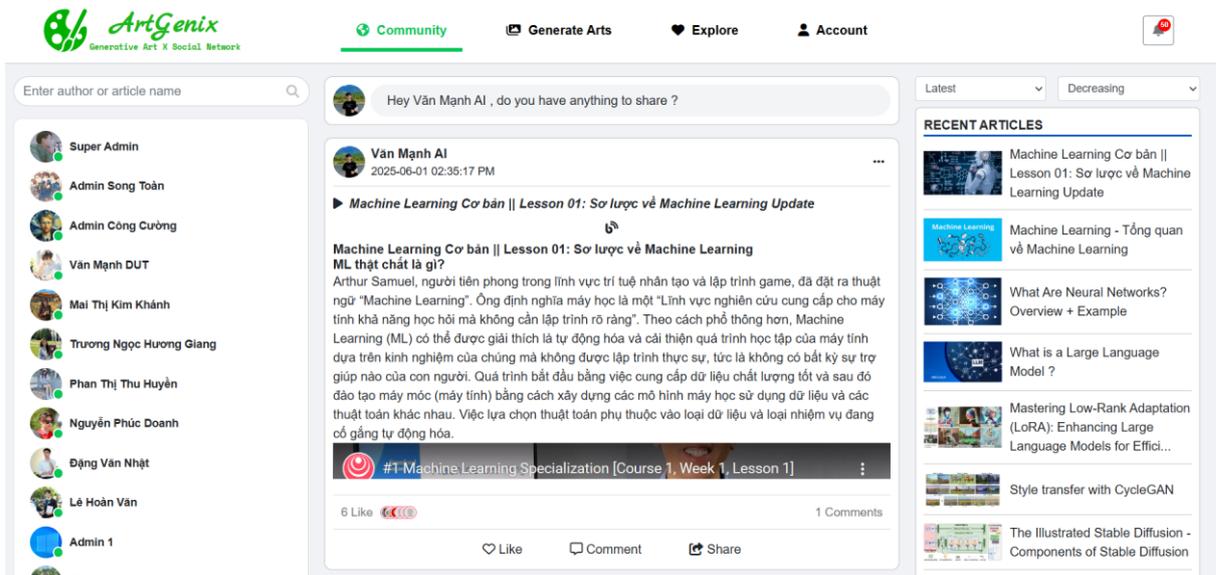


Hình 4.78 User quản lý thông báo

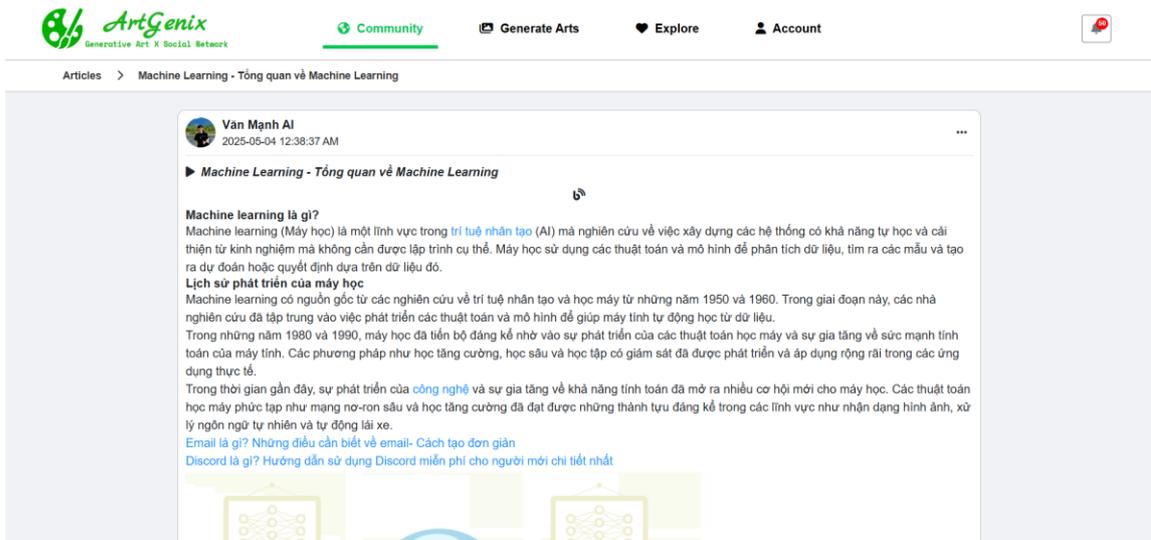


Hình 4.79 User xóa thông báo

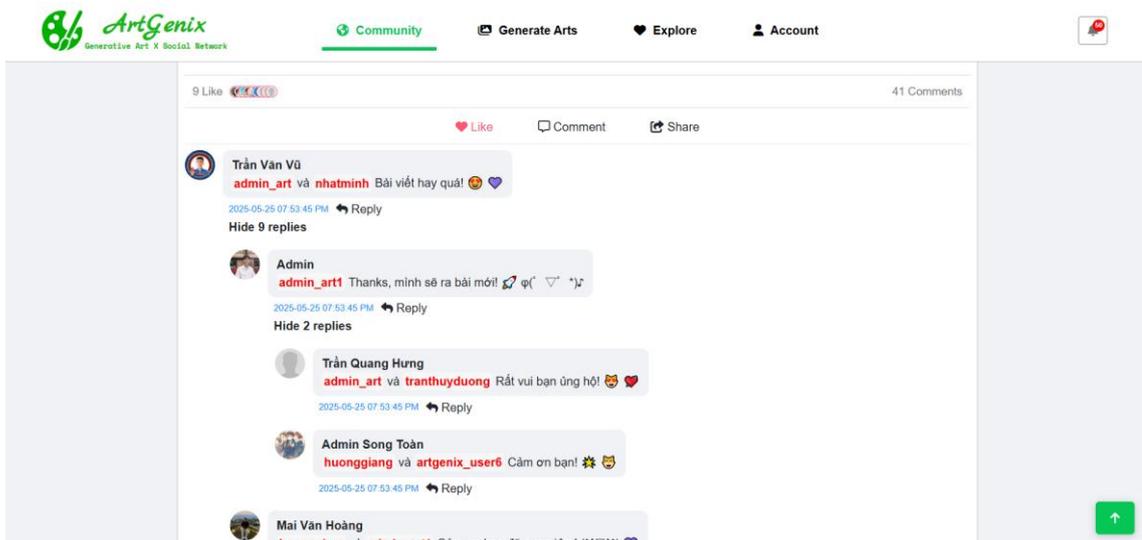
### 4.3.13. Nhóm chức năng Trang chủ, bài viết, sinh ảnh và khám phá



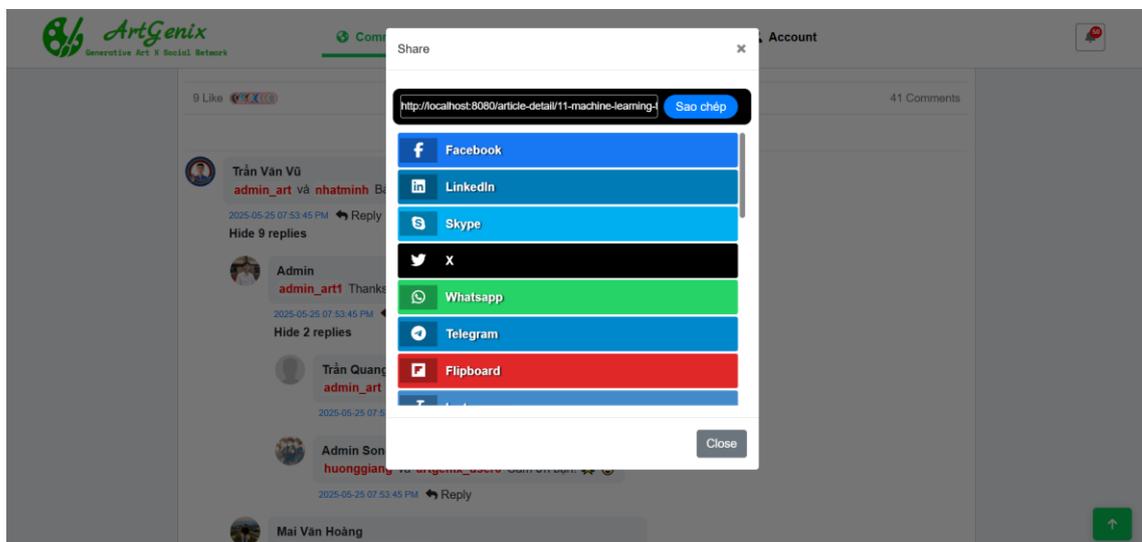
Hình 4.80 Trang chủ, cộng đồng chia sẻ bài viết



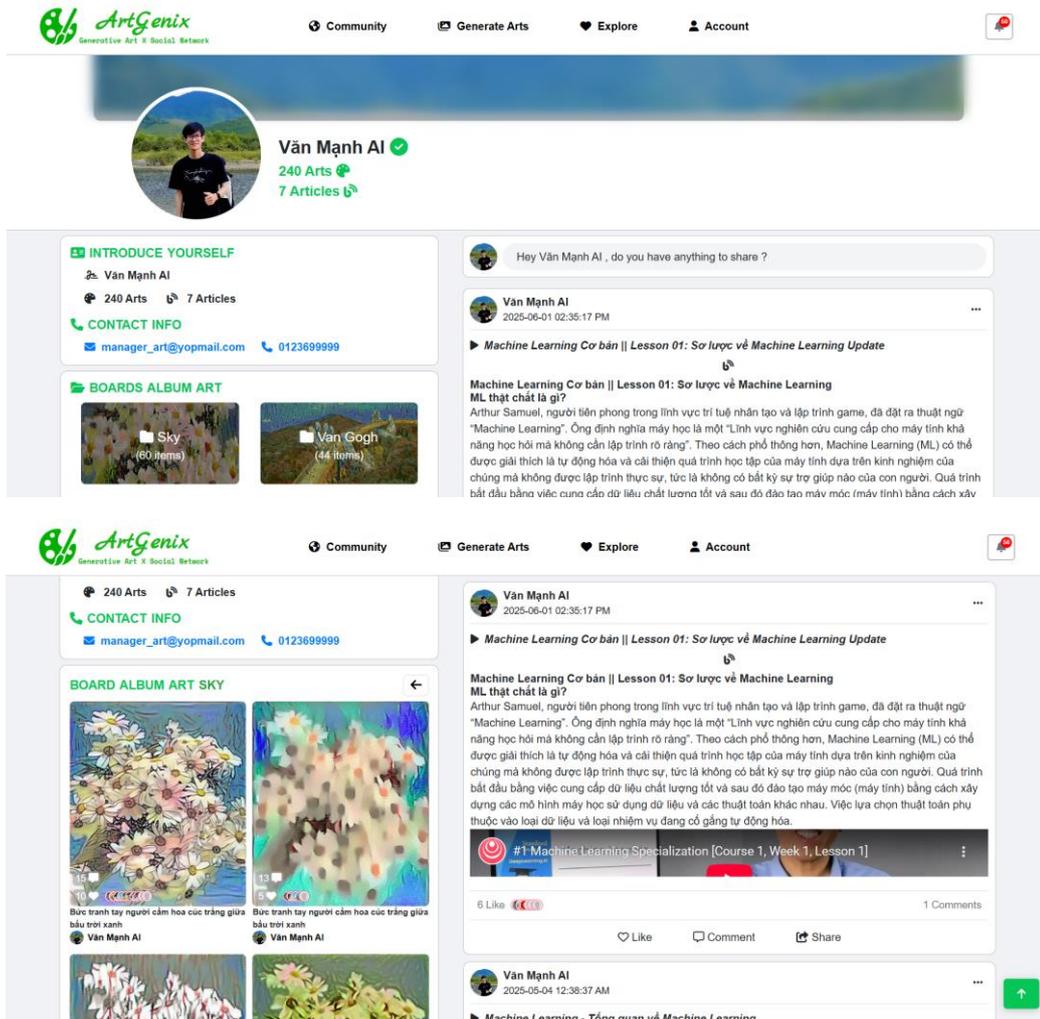
Hình 4.81 Trang chủ, xem bài viết chi tiết



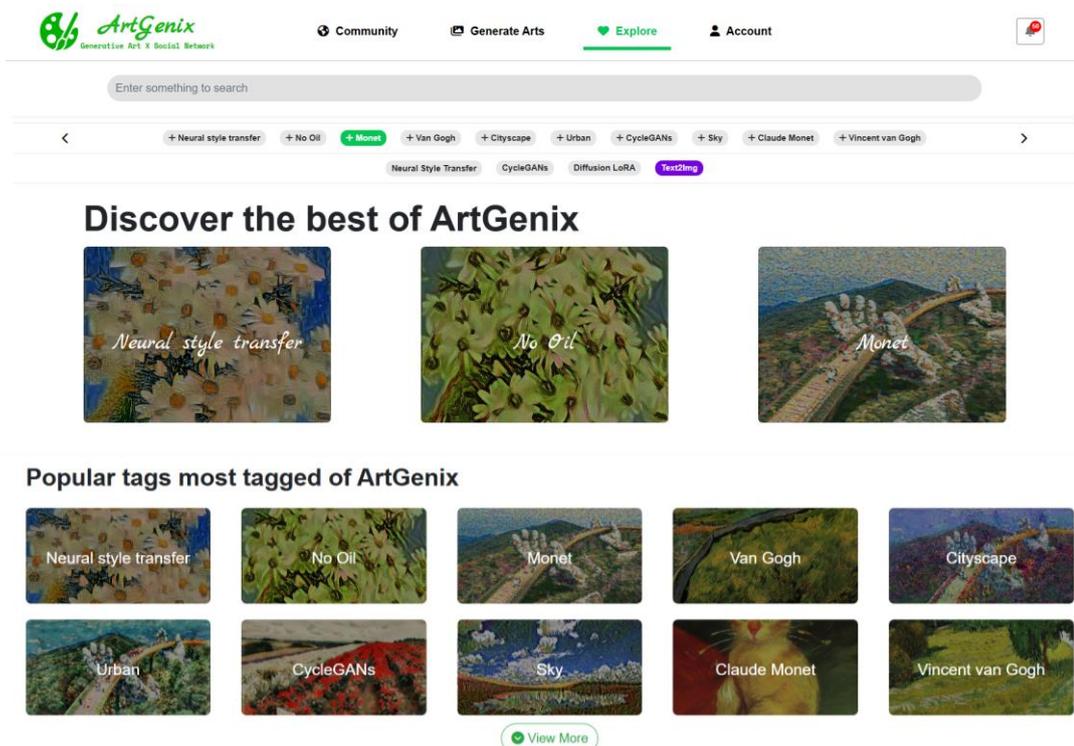
Hình 4.82 Trang chủ, tương tác bình luận và thích bài viết



Hình 4.83 Trang chủ, chia sẻ bài viết



Hình 4.84 Trang chủ, xem trang cá nhân



Explore See what's new on ArtGenix

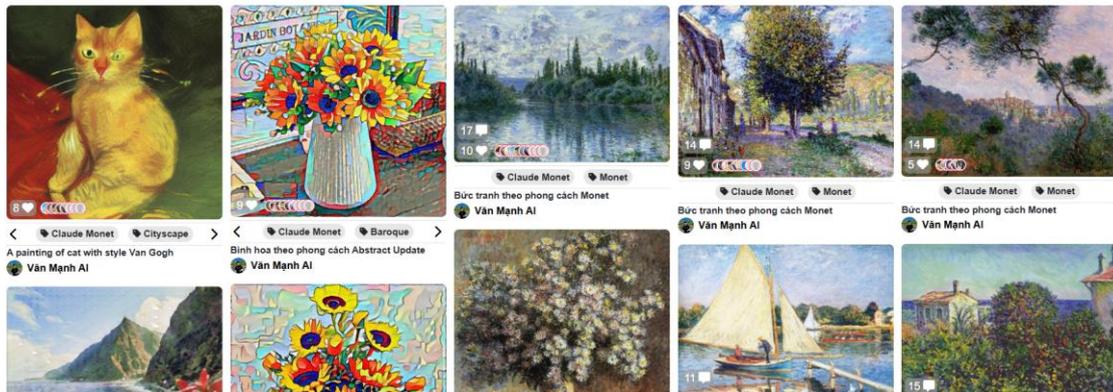


Hình 4.85 Trang chủ, khám phá các tác phẩm nghệ thuật



Enter something to search

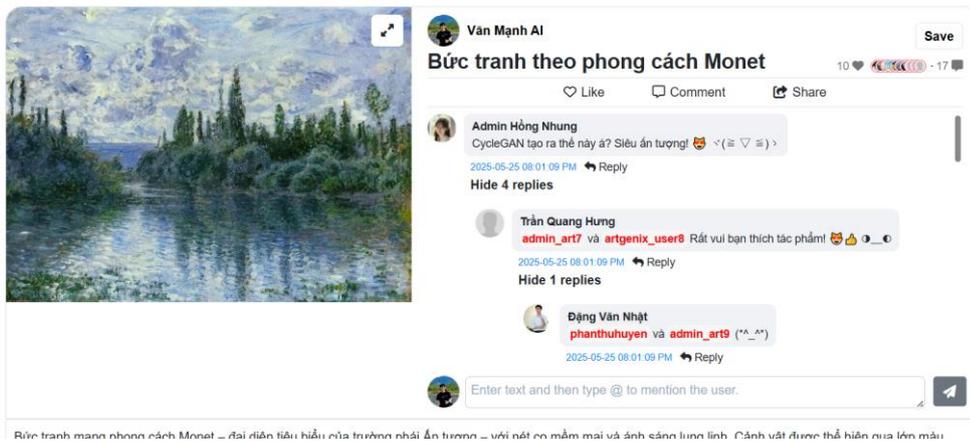
Explore what art related to this tag is on ArtGenix Claude Monet



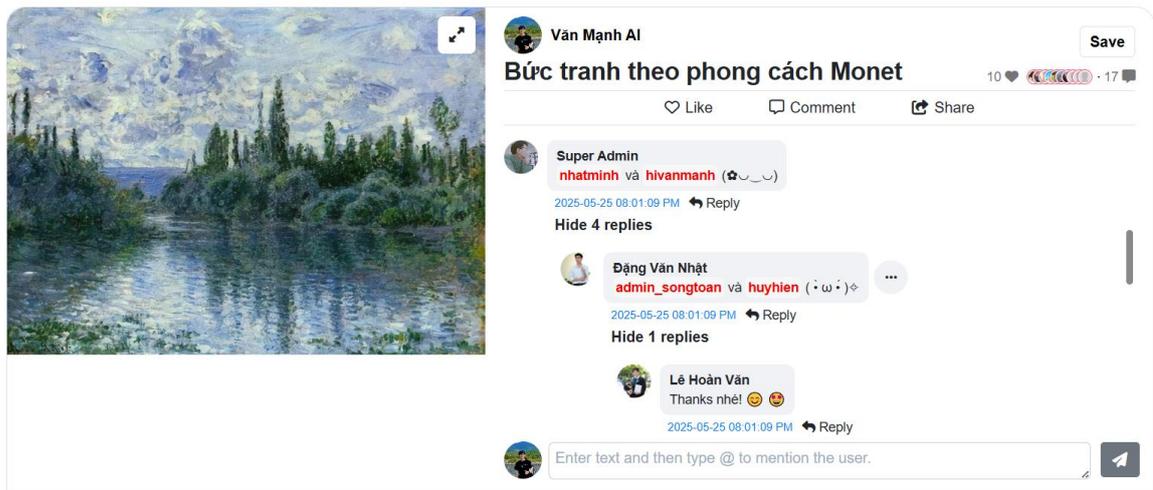
Hình 4.86 Trang chủ, xem tác phẩm nghệ thuật theo tag



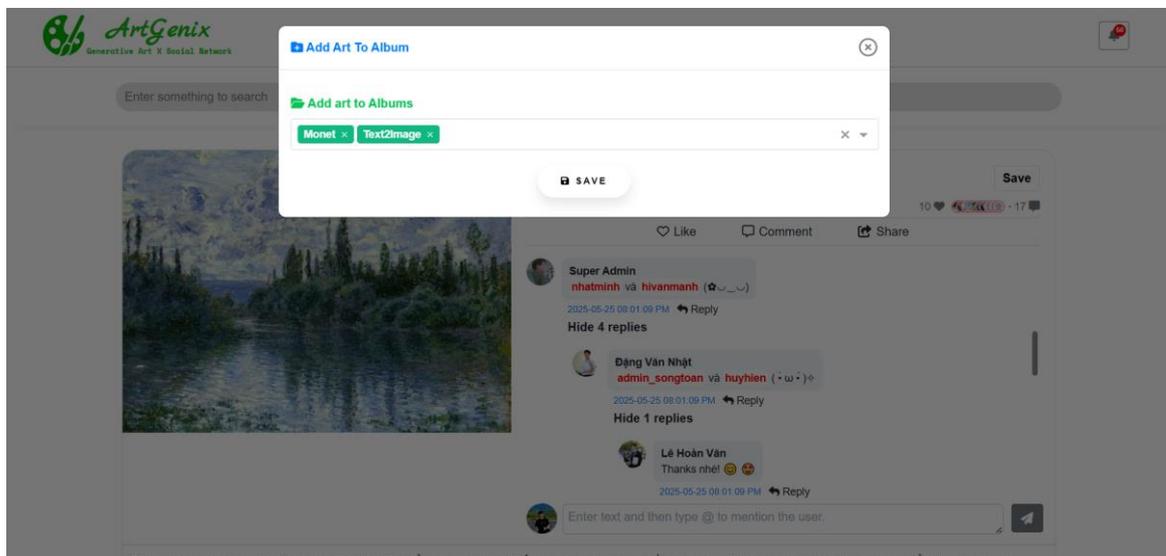
Enter something to search



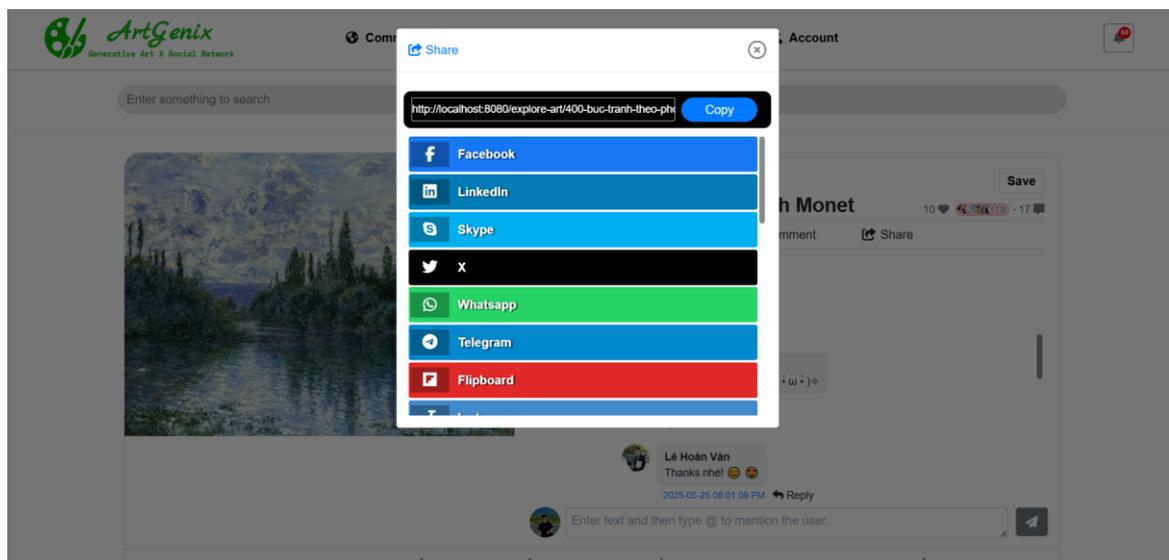
Hình 4.87 Trang chủ, xem chi tiết tác phẩm nghệ thuật



Hình 4.88 Trang chủ, tương tác bình luận và thích tác phẩm nghệ thuật



Hình 4.89 Trang chủ, lưu tác phẩm nghệ thuật vào bộ sưu tập cá nhân



Hình 4.90 Trang chủ, chia sẻ tác phẩm nghệ thuật

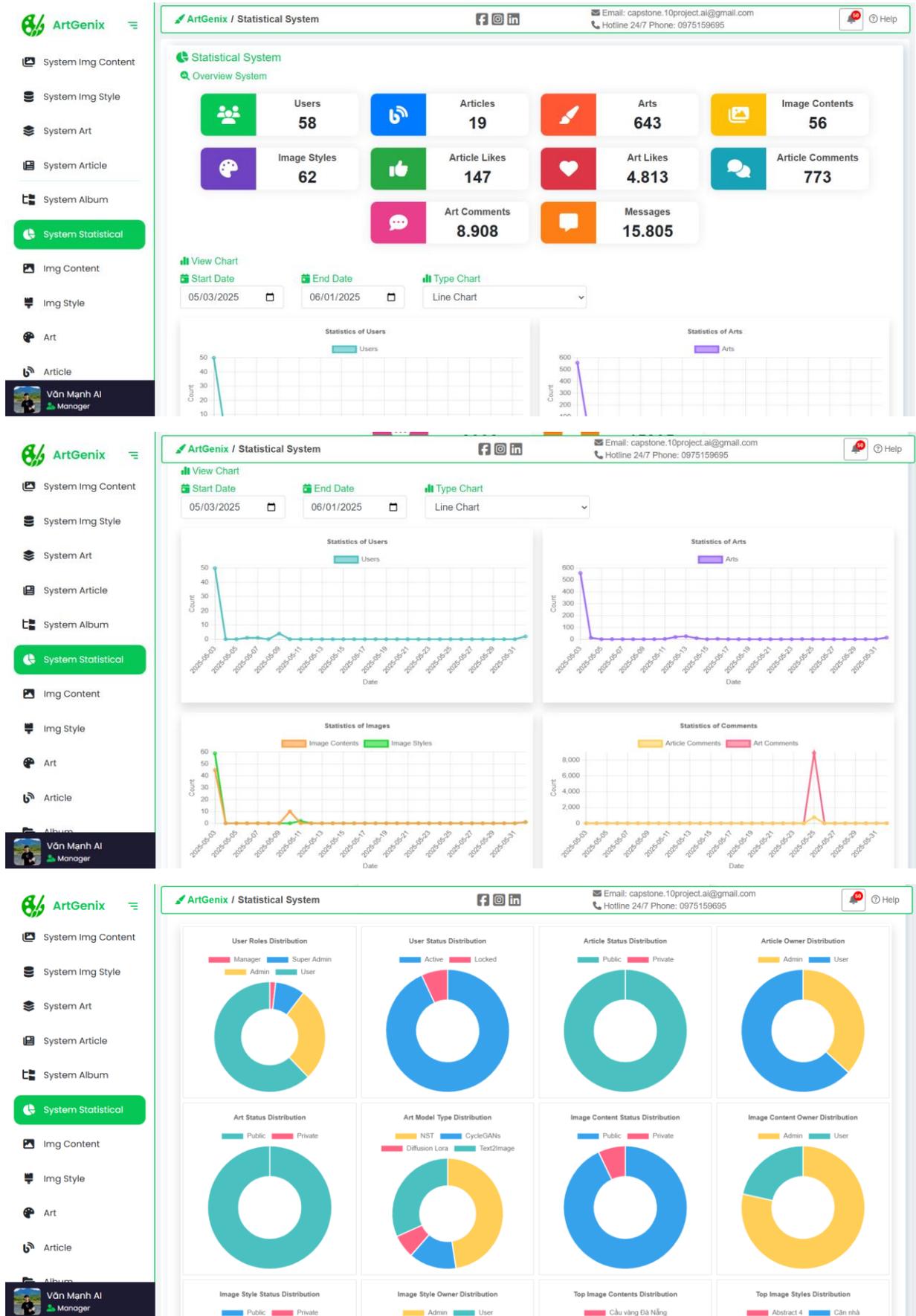
The image displays the ArtGenix web interface, which is divided into four main sections, each representing a different AI art generation method. Each section includes a control panel on the left and a preview window on the right.

- Neural Style Transfer VGG16:** Features sliders for Epochs (set to 280), Content Weight (set to 1), and Style Weight (set to 1000). It also has dropdowns for Image Content and Image Style.
- Diffusion LoRA:** Features sliders for Inference Steps (set to 30), Strength (set to 0.4), and Image Cond Scale (set to 1.5). It includes a 'Use Oil filter' checkbox (checked) and a 'Style' dropdown menu with options like Claude Monet, Vincent van Gogh, Ukiyo-e, and Paul Cézanne.
- CycleGANs:** Features sliders for Oil Size (set to 4) and Oil Dyn R (set to 10). It includes a 'Use Oil filter' checkbox (checked) and a 'Style' dropdown menu with options like Claude Monet and Ukiyo-e.
- Text2Img:** Features a text input field with the prompt 'A painting of house with style Van Gogh'. It includes sliders for Inference Steps (set to 10), Guidance S (set to 2.5), Height (set to 512), and Width (set to 512).

Each preview window shows a generated image of a mountain landscape with a crescent moon and stars. The 'Image generation time' is displayed for each method: 13.01 seconds for Neural Style Transfer VGG16, 44.6 seconds for Diffusion LoRA, and 4-5 seconds for CycleGANs. The Text2Img method does not show a specific time. A 'GENERATE ART' button is present in each preview window, and a vertical navigation bar with numbers 1, 2, 3, 4 is on the right side of each preview area.

Hình 4.91 Trang chủ, sinh ảnh nghệ thuật dành cho khách

### 4.3.14. Chức năng Xem thống kê của hệ thống



Hình 4.92 Admin xem thống kê của hệ thống

#### 4.4. Mô hình sinh ảnh

##### a. So sánh 3 mô hình

Trong quá trình thực nghiệm, hệ thống cho thấy nếu ảnh được lấy từ dịch vụ lưu trữ S3 thì thời gian xử lý thực tế tăng trung bình khoảng +3 giây so với thời gian ước tính. Ngược lại, nếu ảnh được tải trực tiếp từ người dùng (upload), thời gian xử lý thực tế có thể tăng +6 giây.

Tiêu chí	Neural Style Transfer	CycleGAN	Diffusion LoRA
Thời gian sinh ảnh	30 giây – hơn 2 phút	1 – 3 giây mỗi ảnh	3 giây – hơn 3 phút (tùy tham số và độ phân giải)
Chất lượng ảnh	Gần giống tranh gốc, chi tiết tốt	Thấp, dễ nhiễu, phong cách không ổn định	Chi tiết cao, linh hoạt theo phong cách và nội dung
Yêu cầu phần cứng	Thấp, chạy được trên CPU	Cao, cần GPU mạnh và bộ nhớ lớn	Trung bình, yêu cầu GPU nhưng không cần cấu hình cao

Bảng 4.2 So sánh 3 mô hình

##### b. Kết quả của 3 mô hình



Bảng 4.3 Kết quả của 3 mô hình

##### – Neural Style Transfer



Hình 4.93 Kết quả Neural Style Transfer

##### – CycleGAN



Hình 4.94 Kết quả CycleGAN

## – Diffusion LoRA Style Transfer



Hình 4.95 Kết quả Diffusion LoRA Style Transfer

### 4.5. Kiểm thử hệ thống

Phần này trình bày chi tiết quá trình kiểm thử hệ thống ArtGenix, tập trung vào việc xác minh hiệu quả, độ tin cậy, bảo mật và trải nghiệm người dùng của các thành phần cốt lõi. Đi sâu vào các phương pháp kiểm thử, bao gồm Unit Test và Feature Test cho API backend, cũng như Cypress E2E cho giao diện người dùng, đảm bảo hệ thống hoạt động đúng như mong đợi.

#### 4.5.1. Kiểm thử Frontend

##### a. Tổng quan

Kiểm thử giao diện dự án ArtGenix, dự án sử dụng cypress để kiểm thử auto test. Cypress thường được so sánh với Selenium, tuy nhiên Cypress khác nhau về căn bản cũng như kiến trúc. Cypress không bị ràng buộc bởi các cơ chế giống như Selenium. Nhờ đó, có thể viết kịch bản test nhanh hơn, dễ dàng hơn và đáng tin cậy hơn. Cypress cho phép thử ở các giai đoạn phát triển phần mềm:

- *End-to-end tests*
- *Integration tests*
- *Unit tests*

Cypress là 1 công cụ miễn phí, mã nguồn mở, được cài đặt cục bộ và có Dashboard Service để ghi lại các thử nghiệm.

##### Ưu điểm:

- Mã nguồn mở, không mất phí. Đơn giản, dễ cài đặt và viết kịch bản test.
- Document đầy đủ, chi tiết, có tích hợp sẵn các examples trong tool giúp cho người mới bắt đầu dễ tiếp cận hơn.
- Debug dễ dàng do các thao tác đã được ghi lại trong quá trình chạy kịch bản test. Không cần thêm thời gian wait hoặc sleeps vào scripts, Cypress tự động đợi phần tử xuất hiện trước khi tiếp tục.

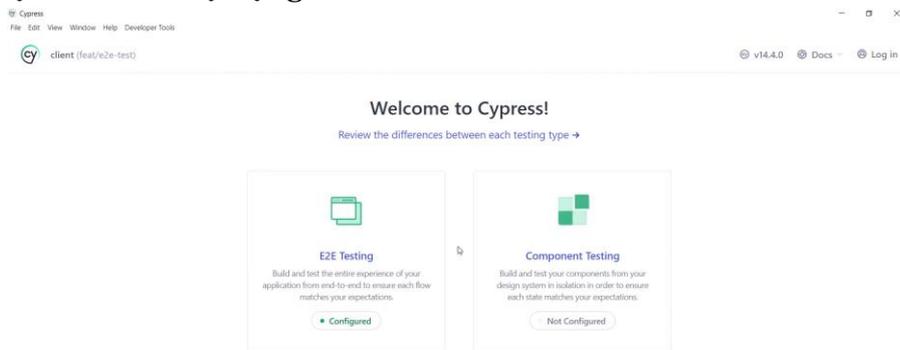
- Dễ dàng kiểm soát và kiểm tra các trường hợp liên quan đến băng thông mạng, có thể khai báo lưu lượng mạng theo bất kì cách nào.
- Hỗ trợ chụp ảnh lỗi và quay video toàn bộ quá trình chạy kịch bản test.
- Cung cấp trang Dashboard service có thể xem tổng quan, báo cáo kết quả sau mỗi lần chạy.

**Nhược điểm:**

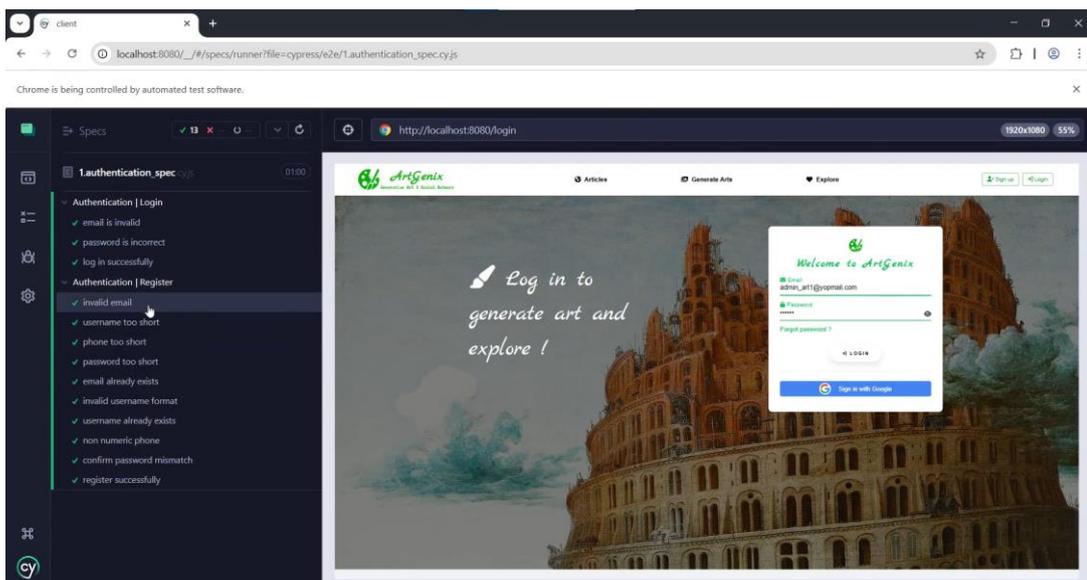
- Cộng đồng sử dụng chưa nhiều. Không hỗ trợ test trên native mobile app.
- Không thể hỗ trợ tương tác với nhiều tab do chạy bên trong trình duyệt.
- Nếu là bản free thì trang Dashboard service có 1 số hạn chế nhất định, nếu muốn sử dụng full dịch vụ thì phải mất phí.
- Chỉ hỗ trợ Javascript. Hỗ trợ 1 số ít framework: Mocha JS.

Trình duyệt được hỗ trợ còn hạn chế: chỉ hỗ trợ Firefox, Edge, Chrome, Electron.

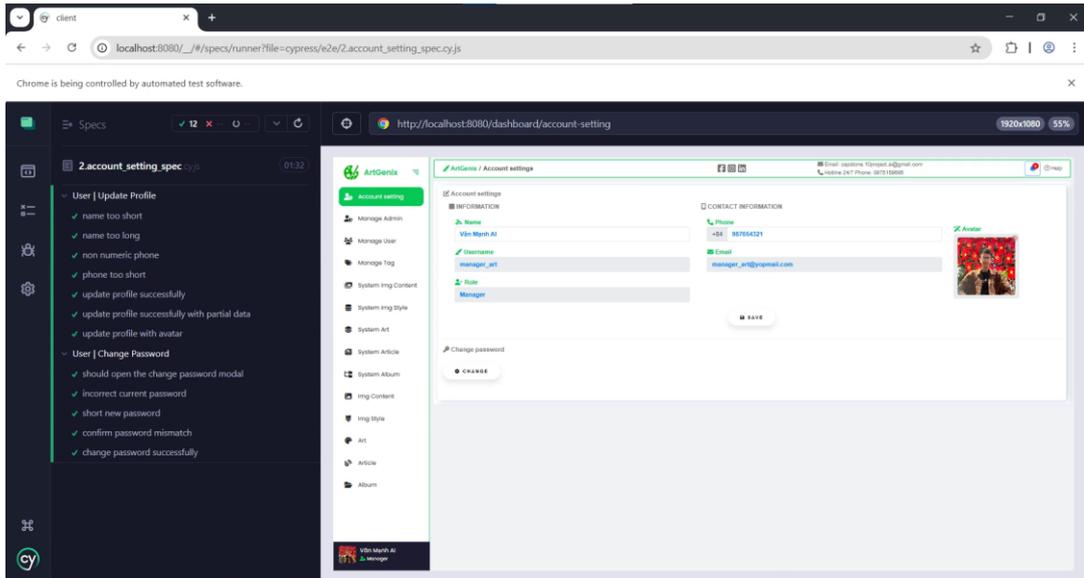
**b. Thực hiện kiểm thử tự động**



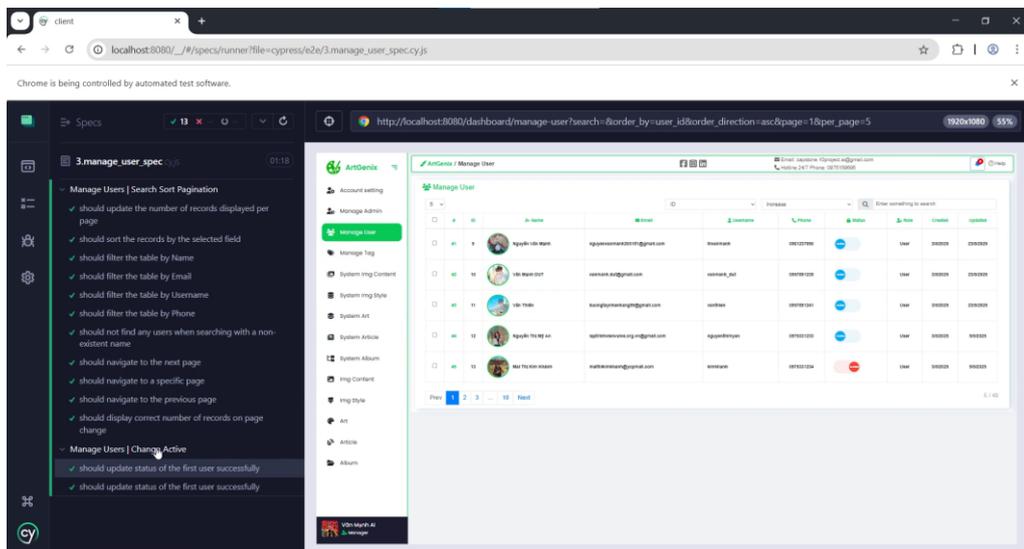
Hình 4.96 Kiểm thử tự động E2E bằng Cypress



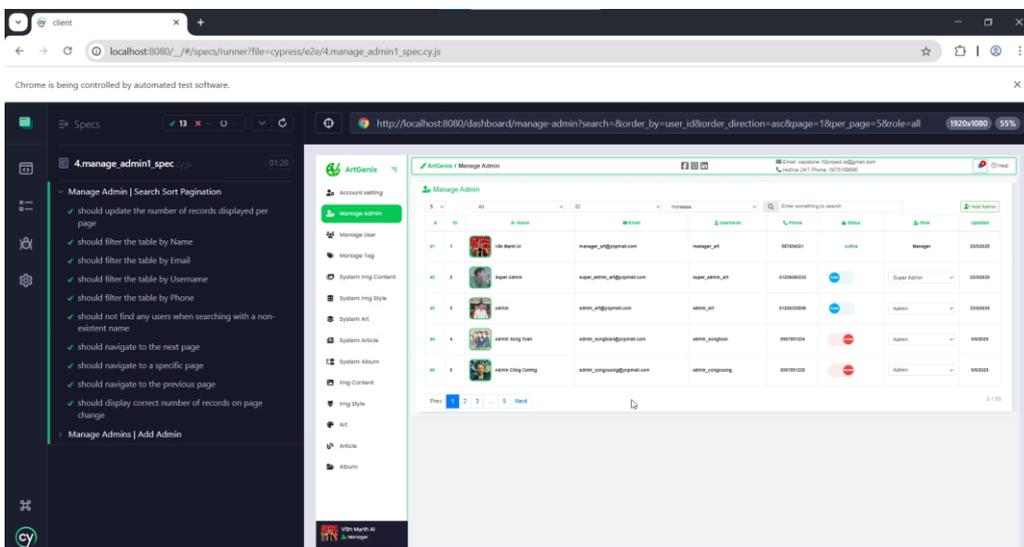
Hình 4.97 Kiểm thử tự động E2E chức năng Authentication



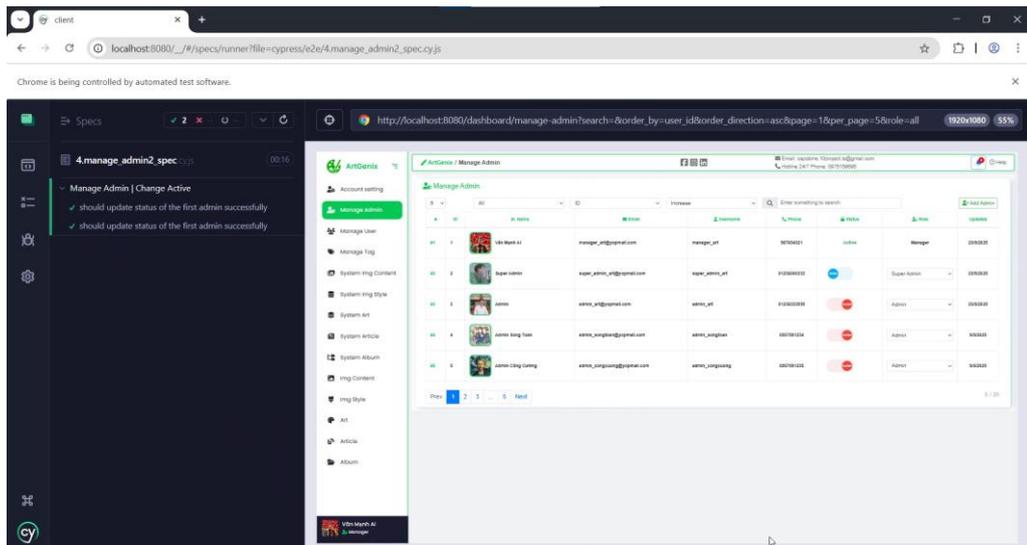
Hình 4.98 Kiểm thử tự động E2E chức năng Cài đặt tài khoản



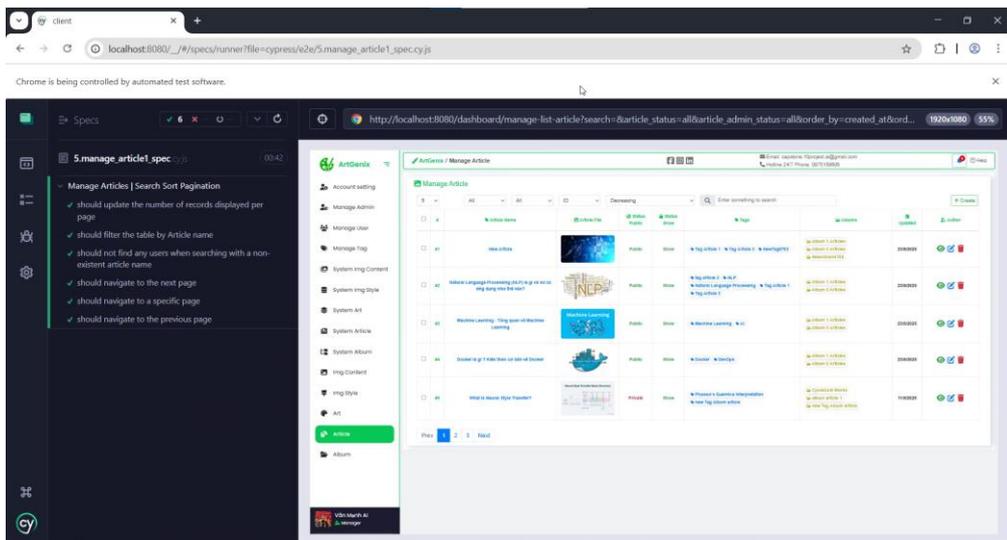
Hình 4.99 Kiểm thử tự động E2E chức năng Quản lý tài khoản người dùng



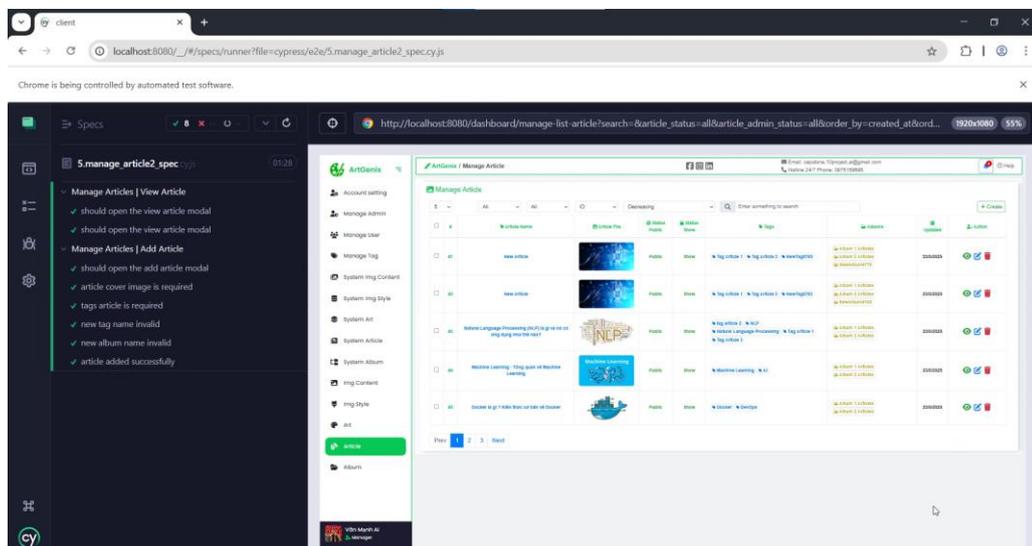
Hình 4.100 Kiểm thử tự động E2E chức năng Quản lý tài khoản admin



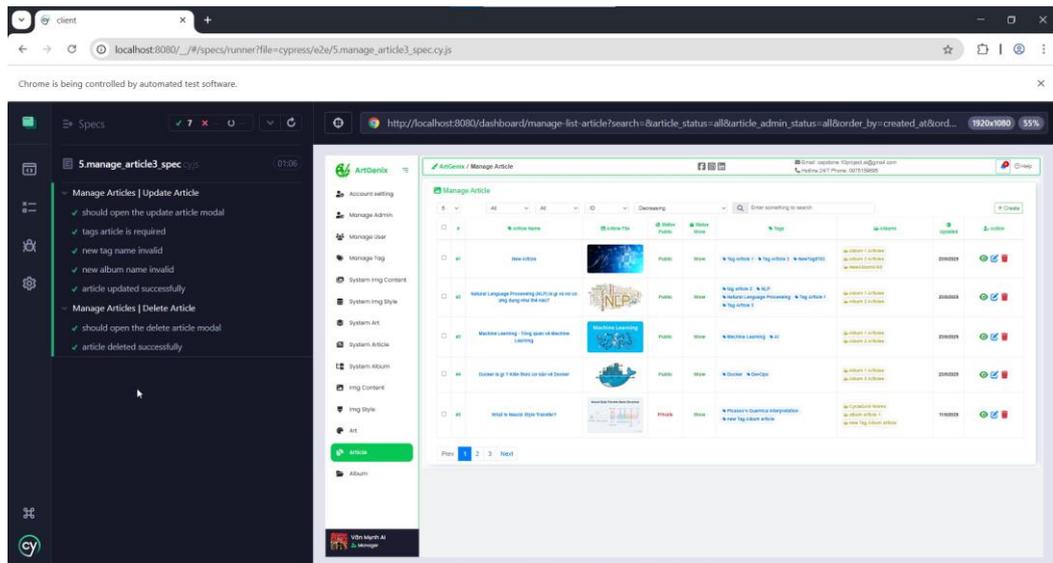
Hình 4.101 Kiểm thử tự động E2E chức năng Kích hoạt tài khoản admin



Hình 4.102 Kiểm thử tự động E2E chức năng Quản lý bài viết



Hình 4.103 Kiểm thử tự động E2E chức năng Xem, thêm bài viết



Hình 4.104 Kiểm thử tự động E2E chức năng Xóa, sửa bài viết

## 4.5.2. Kiểm thử Backend API

### a. Tổng quan

Kiểm thử Backend API đóng vai trò quan trọng trong việc đảm bảo tính đồng nhất và ổn định của hệ thống. Laravel cung cấp môi trường chuyên nghiệp để thực hiện kiểm thử tự động cho các API.

PHPUnit là framework phổ biến được sử dụng trong cộng đồng PHP để kiểm thử đơn vị và kiểm thử tích hợp. Laravel tích hợp sẵn PHPUnit, giúp lập trình viên viết các bài kiểm thử một cách hiệu quả.

Kiểm thử đơn vị trong Laravel giúp đảm bảo rằng từng phần của mã nguồn hoạt động chính xác. Điều này có thể bao gồm việc kiểm tra các hàm, lớp, và các thành phần nhỏ khác của ứng dụng.

Kiểm thử API endpoint trong Laravel giúp đảm bảo rằng các dịch vụ API trả về dữ liệu đúng cách và tuân thủ các chuẩn đã định.

#### Ưu điểm:

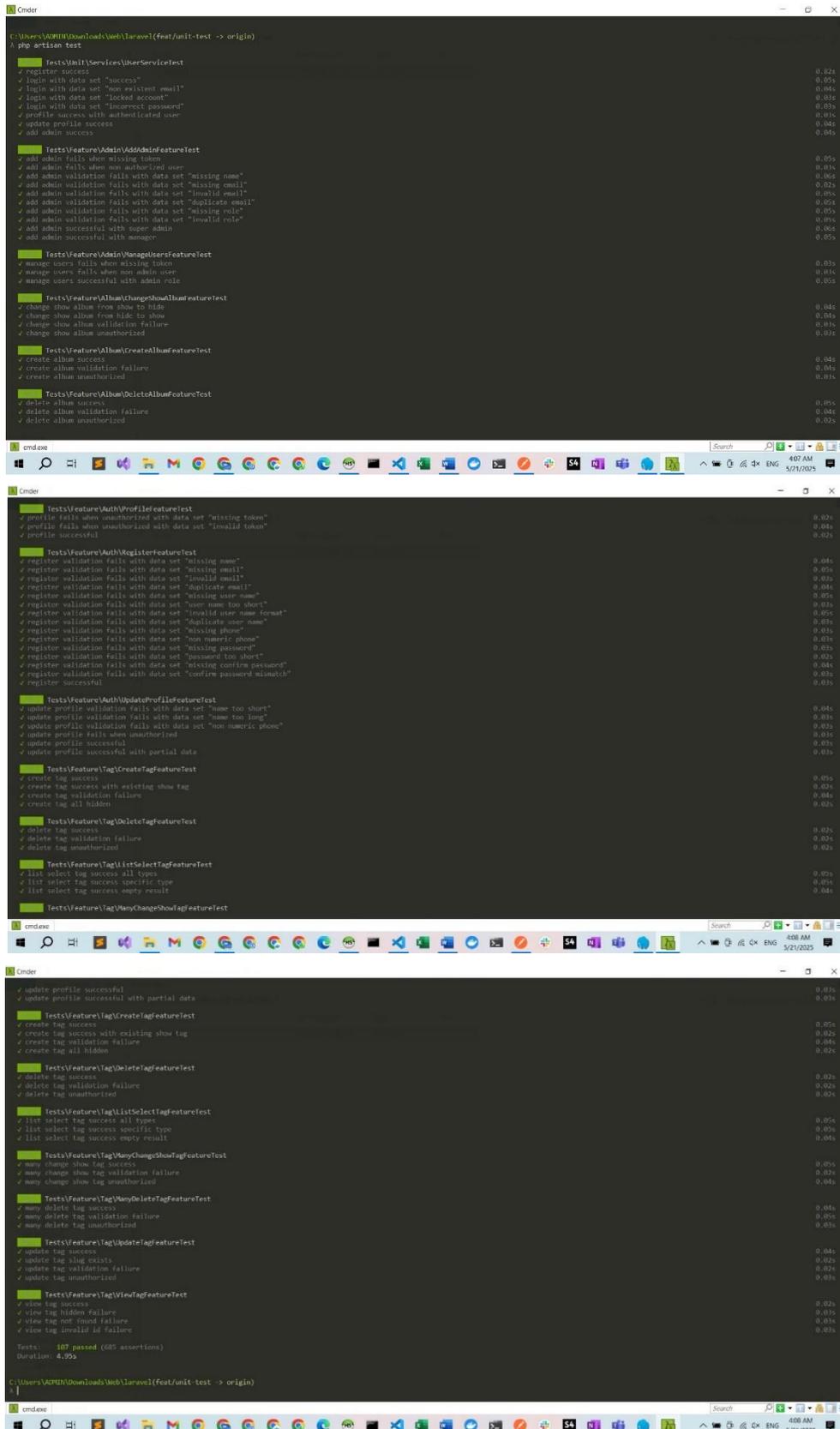
- Đồng bộ với Laravel, giúp tích hợp dễ dàng.
- Kiểm thử đơn vị và tích hợp chặt chẽ.
- Dễ dàng xác định lỗi và sửa chúng.

#### Nhược điểm:

- Cộng đồng sử dụng chưa nhiều so với các framework khác.
- Không hỗ trợ kiểm thử trên ứng dụng di động một cách tự nhiên.
- Yêu cầu hiểu biết vững về PHP và Laravel.

- Việc sử dụng PHPUnit và Laravel giúp đảm bảo rằng Backend API của dự án được kiểm thử một cách toàn diện và đáng tin cậy.

**b. Thực hiện kiểm thử tự động**



Hình 4.105 Kiểm thử tự động cho API server bằng Unit Test và Feature Test

## KẾT LUẬN & HƯỚNG PHÁT TRIỂN

### 1. Kết quả đạt được

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đề án đã đạt được những kết quả sau:

*Về mặt lý thuyết:* Quá trình phát triển ArtGenix đã nâng cao kỹ năng kỹ thuật trong việc xây dựng hệ thống tích hợp mô hình học sâu, phát triển giao diện người dùng với VueJS và triển khai API backend bằng Laravel. Xây dựng API server Laravel sử dụng Design Pattern: Repository Service Pattern, một lựa chọn mang lại lợi ích đáng kể nhờ kiến trúc này hỗ trợ dễ dàng bảo trì, mở rộng và phát triển hệ thống. Về mặt lý thuyết, đề án đã nghiên cứu sâu các mô hình như Neural Style Transfer, CycleGAN và Diffusion LoRA, đặt nền móng cho việc nghiên cứu chuyên sâu hơn vào lĩnh vực tạo sinh ảnh nghệ thuật nói riêng và tạo sinh nói chung.

*Về mặt thực tiễn ứng dụng:* Dự án đã thành công trong việc hiện thực hóa một nền tảng sinh ảnh nghệ thuật, hỗ trợ người dùng tạo tác phẩm độc đáo và xây dựng cộng đồng sáng tạo thông qua các tính năng mạng xã hội. Kết quả thực tiễn bao gồm giao diện thân thiện, hơn 136 API service cùng 4 API Model API hoạt động ổn định, và khả năng tương tác trực tuyến được đánh giá cao, đồng thời áp dụng hiệu quả các công cụ triển khai như Vercel, Railway, và Kaggle, cung cấp nền tảng vững chắc cho các dự án sáng tạo trong tương lai.

Mã nguồn của ứng dụng:

- <https://github.com/capstone-engineer/laravel>
- <https://github.com/capstone-engineer/client>
- <https://github.com/capstone-engineer/django>

Ứng dụng:

- API Backend Laravel Deploy: <https://artgenix-production.up.railway.app>
- Web Client Deploy: <https://artgenix.vercel.app>

### 2. Hạn chế của đề tài

- Hiệu suất hệ thống bị giới hạn bởi các gói miễn phí (Vercel: 23 GB dung lượng, 100 MB tệp tĩnh; Railway: 512 MB bộ nhớ, 1 GB đĩa), dẫn đến thời gian phản hồi chậm khi xử lý lượng lớn người dùng.
- Khả năng mở rộng của mô hình học sâu trên Kaggle còn hạn chế do phụ thuộc vào 4 tài khoản với GPU P100.

- Tính năng phân tích chi tiết cảm xúc từ bình luận của người dùng chưa được tích hợp, làm giảm khả năng hiểu sâu hơn về phản hồi cộng đồng

### 3. Hướng phát triển

- Tối ưu hóa hệ thống bằng cách chuyển sang các gói trả phí của Vercel và Railway để tăng tài nguyên (dung lượng và CPU), đồng thời nghiên cứu triển khai mô hình học sâu trên nền tảng đám mây mạnh hơn như AWS hoặc Google Cloud.
- Tích hợp tính năng phân tích chi tiết cảm xúc từ bình luận để cải thiện trải nghiệm người dùng.
- Phát triển thêm các tính năng mới, chẳng hạn như mô phỏng việc gắn tranh được tạo sinh lên vật dụng, dụng cụ hoặc quần áo, giúp người dùng xem trước khi áp dụng trong thực tế.
- Nâng cấp hệ thống lên mô hình trả phí với các gói nâng cao, cho phép người dùng thanh toán để mở rộng giới hạn số lượng ảnh được sinh trong một ngày, hỗ trợ phát triển bền vững.
- Nghiên cứu các design pattern tối ưu API và giao diện để thúc đẩy việc mở rộng hệ thống, hướng tới ứng dụng trong giáo dục nghệ thuật và ngành công nghiệp in ấn quy mô lớn.

## TÀI LIỆU THAM KHẢO

- [1] Laravel Documentation, "Laravel 10.x Documentation," [online] Available at: <https://laravel.com/docs/10.x> (accessed June 08, 2025, 03:30 PM +07).
- [2] Kinsta, "17 Methods to Optimize Laravel Performance," [online] Available at: <https://kinsta.com/blog/laravel-performance/> (accessed June 08, 2025, 03:30 PM +07).
- [3] Viblo, "Design Pattern: Repository Pattern," [online] Available at: <https://viblo.asia/p/design-pattern-repository-pattern-Do754OzWIM6> (accessed June 08, 2025, 03:30 PM +07).
- [4] Railway Documentation, "Deploy a Laravel App," [online] Available at: <https://docs.railway.com/guides/laravel> (accessed June 08, 2025, 03:30 PM +07).
- [5] Vercel, "How to Deploy a Vue.js Site with Vercel," [online] Available at: <https://vercel.com/guides/deploying-vuejs-to-vercel> (accessed June 08, 2025, 03:30 PM +07).
- [6] R. Rombach et al., "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," arXiv:2307.01952 [cs], [online] Available at: <https://arxiv.org/pdf/2307.01952> (accessed June 08, 2025, 03:30 PM +07).
- [7] E. J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," arXiv:2106.09685 [cs], [online] Available at: <https://arxiv.org/abs/2106.09685> (accessed June 08, 2025, 03:30 PM +07).
- [8] J. Li et al., "LoRA-Composer: Leveraging Low-Rank Adaptation for Multi-Concept Customization in Training-Free Diffusion Models," arXiv:2403.11627 [cs], [online] Available at: <https://arxiv.org/abs/2403.11627> (accessed June 08, 2025, 03:30 PM +07).
- [9] Z. Wang et al., "LoRA-Enhanced Distillation on Guided Diffusion Models," arXiv:2312.06899 [cs], [online] Available at: <https://arxiv.org/abs/2312.06899> (accessed June 08, 2025, 03:30 PM +07).
- [10] Y. Zhang et al., "Efficient Image Restoration through Low-Rank Adaptation and Stable Diffusion XL," arXiv:2408.17060 [cs], [online] Available at: <https://arxiv.org/abs/2408.17060> (accessed June 08, 2025, 03:30 PM +07).
- [11] Y. Zhang et al., "Face Swap via Diffusion Model," arXiv:2403.01108v1 [cs], [online] Available at: <https://arxiv.org/html/2403.01108v1> (accessed June 08, 2025, 03:30 PM +07).
- [12] J.-Y. Zhu et al., "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," arXiv:1703.10593 [cs], [online] Available at:

- <https://arxiv.org/pdf/1703.10593> (accessed June 08, 2025, 03:30 PM +07).
- [13] J.-Y. Zhu, "CycleGAN and pix2pix in PyTorch," [online] Available at: <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix?tab=readme-ov-file> (accessed June 08, 2025, 03:30 PM +07).
- [14] J. Zhu et al., "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," [online] Available at: <https://junyanz.github.io/CycleGAN/> (accessed June 08, 2025, 03:30 PM +07).
- [15] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," arXiv:1508.06576 [cs], [online] Available at: <https://arxiv.org/pdf/1508.06576> (accessed June 08, 2025, 03:30 PM +07).
- [16] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2414-2423, [online] Available at: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Gatys\\_Image\\_Style\\_Transfer\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf) (accessed June 08, 2025, 03:30 PM +07).
- [17] L. Jing, Y. Shen, and J. Liu, "Neural Style Transfer: A Review," arXiv:1705.04058 [cs], [online] Available at: <https://arxiv.org/abs/1705.04058> (accessed June 08, 2025, 03:30 PM +07).
- [18] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 1501-1510, [online] Available at: [https://openaccess.thecvf.com/content\\_ICCV\\_2017/papers/Huang\\_Arbitrary\\_Style\\_Transfer\\_ICCV\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_ICCV_2017/papers/Huang_Arbitrary_Style_Transfer_ICCV_2017_paper.pdf) (accessed June 08, 2025, 03:30 PM +07).
- [19] [19] V7 Labs, "Neural Style Transfer: Everything You Need to Know [Guide]," [online] Available at: <https://www.v7labs.com/blog/neural-style-transfer> (accessed June 08, 2025, 03:30 PM +07).
- [20] HackerNoon, "How Do Neural Style Transfers Work?," [online] Available at: <https://hackernoon.com/how-do-neural-style-transfers-work-7bedae0559a> (accessed June 08, 2025, 03:30 PM +07).
- [21] L. A. Gatys, A. S. Ecker, and M. Bethge, "Demystifying Neural Style Transfer," arXiv:1701.01036 [cs], [online] Available at: <https://arxiv.org/pdf/1701.01036> (accessed June 08, 2025, 03:30 PM +07).
- [22] A. Vaswani et al., "Attention Is All You Need," arXiv:1706.03762 [cs], [online] Available at: <https://arxiv.org/abs/1706.03762> (accessed June 08, 2025, 03:30 PM +07).