

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи освітнього ступеня «бакалавр»  
за спеціальністю 121 «Інженерія програмного забезпечення»  
(освітня програма «Інженерія програмного забезпечення»)

на тему:

### **«Система управління контентом для художньої галереї»**

Виконав студент групи ІПЗ-20-4  
САВЧЕНКО Данило Павлович

Керівник роботи:  
Кубрак Юрій Олександрович

Консультант:  
МОРОЗОВ Андрій Васильович

Рецензент:  
ЄФРЕМОВ Юрій Миколайович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**«ЗАТВЕРДЖУЮ»**

Зав. кафедри інженерії  
програмного забезпечення  
Тетяна Вакалюк

**«26» лютого 2024 р.**

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**

Здобувач вищої освіти: **САВЧЕНКО Данило Павлович**

Керівник роботи: **Кубрак Юрій Олександрович**

Тема роботи: **«Система управління контентом для художньої галереї»**,  
затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74/с**

Вихідні дані для роботи: **предметом дослідження є розробка та впровадження системи управління контентом (CMS) для художньої галереї, яка дозволяє ефективно керувати інформацією про художні твори, експозиції. Об'єктом дослідження є інформаційні процеси та технології, що забезпечують функціонування художньої галереї, включаючи зберігання, обробку даних про художні твори та експозиції.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

| Розділ | Консультант                  | Завдання видав | Завдання прийняв |
|--------|------------------------------|----------------|------------------|
| 1      | Кубрак Ю.О.,<br>Морозов А.В. | 27.02.2024р.   | 27.02.2024р.     |
| 2      | Кубрак Ю.О.,<br>Морозов А.В. | 27.02.2024р.   | 27.02.2024р.     |
| 3      | Кубрак Ю.О.,<br>Морозов А.В. | 27.02.2024р.   | 27.02.2024р.     |

## РЕФЕРАТ

Випускна робота бакалавра присвячена розробці системи управління контентом для художньої галереї. Проведено обґрунтування актуальності теми, аналіз сучасних систем управління контентом, а також досліджено можливості розробки ефективної системи для цифрової галереї на основі веб-технологій.

Метою роботи є розробка система управління контентом для художньої галереї, яка забезпечить зручне керування колекцією художніх робіт, полегшить процес публікації нових експонатів та надасть інтуїтивно зрозумілий інтефейс для користувачів.

Об'єктом дослідження є процес управління контентом у художній галереї, включаючи управління колекцією експонатів та взаємодію з користувачами. Предметом дослідження є інформаційні технології та веб-технології, які використовуються для розробки систем управління контентом.

Пояснювальна записка до випускної бакалаврської роботи містить 96 сторінок, 29 рисунки, 4 додатки та 31 літературне джерело.

Ключові слова: система управління контентом, художня галерея, веб-технології, управління експонатами, цифрова галерея, Laravel, програмне забезпечення.

|           |      |               |        |      |                                                    |                                           |      |         |
|-----------|------|---------------|--------|------|----------------------------------------------------|-------------------------------------------|------|---------|
|           |      |               |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ                           |                                           |      |         |
| Змн.      | Арк. | № докум.      | Підпис | Дата |                                                    |                                           |      |         |
| Розроб.   |      | Савченко Д.П. |        |      | Система управління контентом для художньої галереї | Літ.                                      | Арк. | Аркушів |
| Керівник  |      | Кубрак Ю.О.   |        |      |                                                    |                                           | 3    | 96      |
|           |      |               |        |      |                                                    | Житомирська політехніка<br>Група ІПЗ-20-4 |      |         |
| н. контр. |      | Тичина О.П.   |        |      |                                                    |                                           |      |         |
| Зав. каф. |      | Вакалюк Т.А.  |        |      |                                                    |                                           |      |         |

## ABSTRACT

The bachelor's thesis is dedicated to the development of a content management system for an art gallery. The relevance of the topic has been substantiated, modern content management systems have been analyzed, and the possibilities of developing an effective system for a digital gallery based on web technologies have been explored.

The purpose of the work is to develop a content management system for an art gallery that will provide convenient management of the collection of artworks, facilitate the process of publishing new exhibits, and offer an intuitive interface for users.

The object of the research is the process of content management in an art gallery, including managing the collection of exhibits and interacting with users. The subject of the research is the information technologies and web technologies used for developing content management systems.

The explanatory note to bachelor's work contains 96 pages, 29 pictures, 4 addition and 31 literary sources.

**KEYWORDS:** content management system, art gallery, web technologies, exhibit management, digital gallery, Laravel, software.

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 4    |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

## ЗМІСТ

|                                                                                                                                          |           |
|------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>РЕФЕРАТ .....</b>                                                                                                                     | <b>3</b>  |
| <b>ABSTRACT .....</b>                                                                                                                    | <b>4</b>  |
| <b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....</b>                                                                                                   | <b>7</b>  |
| <b>ВСТУП .....</b>                                                                                                                       | <b>8</b>  |
| <b>РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ<br/>ВЕБТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ<br/>КОНТЕНТОМ ХУДОЖНЬОЇ ГАЛЕРЕЇ .....</b> | <b>10</b> |
| 1.1 Постановка задачі .....                                                                                                              | 10        |
| 1.2 Аналіз аналогів програмного продукту .....                                                                                           | 12        |
| 1.3 Інструментальні засоби реалізації програмного продукту .....                                                                         | 17        |
| Висновки до першого розділу .....                                                                                                        | 19        |
| <b>РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ<br/>УПРАВЛІННЯ КОНТЕНТОМ ДЛЯ ХУДОЖНЬОЇ ГАЛЕРЕЇ .....</b>                                   | <b>21</b> |
| 2.1 Визначення варіантів використання та об'єктно-орієнтованої<br>структури системи .....                                                | 21        |
| 2.2 Розробка бази даних системи .....                                                                                                    | 33        |
| 2.3 Проектування та реалізація алгоритмів роботи системи .....                                                                           | 43        |
| 2.4 Реалізація функціоналу системи управління контентом<br>художньої галереї .....                                                       | 46        |
| Висновки до другого розділу .....                                                                                                        | 54        |
| <b>РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ<br/>УПРАВЛІННЯ КОНТЕНТОМ ХУДОЖНЬОЇ ГАЛЕРЕЇ .....</b>                                 | <b>56</b> |
| 3.1 Порядок встановлення та налаштування параметрів платформи .....                                                                      | 56        |
| 3.2 Структура інтерфейсу краудфандингової платформи .....                                                                                | 58        |
| 3.3 Загальна статистика сайту .....                                                                                                      | 68        |
| 3.4 Тестування платформи .....                                                                                                           | 70        |
| Висновки до третього розділу .....                                                                                                       | 74        |
| <b>ВИСНОВКИ .....</b>                                                                                                                    | <b>76</b> |

|                                  |    |
|----------------------------------|----|
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ..... | 78 |
| ДОДАТКИ .....                    | 81 |

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 6    |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – прикладний програмний інтерфейс

CMS – система управління контентом

CRUD – основні операції, які використовуються при роботі з прикладним програмним інтерфейсом, базою даних

JSON – формат обміну даними, часто використовується в API

SEO – пошукова оптимізація, процес покращення видимості веб-сайту в пошукових системах

SQL – мова структурованих запитів, використовується для управління і маніпулювання даними в реляційних базах даних

UI/UX – користувацький інтерфейс/досвід користувача

БД – база даних

СУБД – система управління базами даних

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 7    |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## ВСТУП

**Актуальність теми.** З розвитком цифрових технологій, художні галереї отримують нові можливості для залучення аудиторії та ефективного управління своїми колекціями. Впровадження систем управління контентом (CMS) стає невід'ємною частиною сучасних галерей, що дозволяє їм досягати глобальної аудиторії та підвищувати свою конкурентоспроможність. Кількість користувачів Інтернету перевищує 5 мільярдів, і цей показник продовжує зростати, збільшуючись на 200 мільйонів за останні 12 місяців [1]. Це відкриває великі можливості для онлайн-присутності та цифрової трансформації галерей.

Соціальні мережі та інтернет-платформи стали основними каналами комунікації та обміну інформацією, що значно розширює межі традиційних галерей. Зокрема, соціальні мережі надають унікальні можливості для залучення нової аудиторії, що раніше була недоступною через географічні обмеження [2]. Це особливо важливо в контексті збереження та популяризації культурної спадщини, оскільки цифрові платформи дозволяють зберігати інформацію про мистецькі твори у високій якості та робити їх доступними для широкого загалу.

Підвищення активності соціальних мереж та зростання їх впливу на громадське життя сприяє тому, що навіть державні діячі активно використовують ці платформи для комунікації з громадянами. Наприклад, президент України Володимир Зеленський та колишній президент США Барак Обама є активними користувачами соціальних мереж, таких як Facebook [3]. Це підкреслює важливість інтеграції сучасних технологій у всі сфери життя, включаючи мистецтво.

Розробка ефективної системи управління контентом для художньої галереї забезпечить зручне керування колекцією художніх робіт, спростить процес публікації нових експонатів та надасть інтуїтивно зрозумілий інтерфейс для користувачів. Це сприятиме не тільки збереженню культурних цінностей, але й їх популяризації серед широкого загалу, забезпечуючи доступ до мистецтва з будь-якої точки світу.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 8    |

**Метою випускної роботи** є розробка системи управління контентом для художньої галереї використовуючи сучасні вебтехнології.

Досягнення поставленої мети передбачає розв’язання таких завдань:

- проведення аналізу функціональних та нефункціональних потреб для CMS художньої галереї;
- аналіз існуючих сервісів;
- визначення вимог до розробки веб-додатку;
- визначення технологій для розробки веб-додатку;
- проектування та реалізація алгоритму роботи проекту.
- визначення об’єктно орієнтовної моделі для веб-додатку.
- розробка архітектури веб-додатку для CMS художньої галереї.
- реалізація програмного продукту за допомогою зпроектованої системи.
- тестування роботи веб-додатку.

**Об’єктом дослідження** є процес управління контентом у художній галереї. Це включає в себе різноманітні аспекти роботи з цифровими колекціями, зокрема, збереження, організацію, публікацію та оновлення експонатів. Крім того, об’єкт дослідження охоплює взаємодію з користувачами, забезпечення їх доступу до інформації про експонати та події, а також забезпечення безперервного розвитку галереї в цифровому просторі.

**Предметом дослідження** є інформаційні технології та веб-технології, які використовуються для розробки систем управління контентом. Особлива увага приділяється інструментам і методам, що застосовуються для створення ефективної CMS для художньої галереї. Це включає вивчення сучасних фреймворків, таких як Laravel, аналіз алгоритмів управління даними, проектування інтуїтивно зрозумілого інтерфейсу користувача, а також інтеграцію різних функцій, що полегшують управління та взаємодію з цифровими колекціями.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 9    |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

# РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ВЕБТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ХУДОЖНЬОЇ ГАЛЕРЕЇ

## 1.1 Постановка задачі

CMS художньої галереї повинна включати в себе легке та просте додавання, редагування та видалення експонатів, жанрів, художників; зручну каталогізацію та пошук картин; підтримка блогів та новин; кабінет користувача; систему відгуків; інтуїтивно зрозумілий та сучасний інтерфейс для користувачів.

Відзначимо основні етапи для реалізації поставленого завдання:

### 1. Аналіз проекту.

1.1. Визначення функціональних та нефункціональних потреб CMS для художньої галереї.

1.2. Аналіз існуючих сайтів/додатків для художніх галерей.

1.3. Вибір технологій розробки вебдодатку.

### 2. Проектування системи управління контентом для художньої галереї

2.1. Розробка сценаріїв використання CMS для художньої галереї.

2.2. Створення алгоритмів роботи вебдодатку.

2.2.1. Створення алгоритмів для системи входу та реєстрації користувачів.

2.2.2. Розробка алгоритмів для основних функцій сайту (додавання, редагування, видалення експонатів).

2.2.3. Проектування алгоритмів для пошуку та фільтрації інформації на сайті.

2.2.4. Розробка алгоритмів для функцій адміністратора та модератора.

2.3. Спроектувати архітектуру CMS для художньої галереї.

2.3.1. Створення діаграми компонентів фронтенд-частини.

2.3.2. Створення діаграми компонентів бекенд-частини.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 10   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

2.3.3. Розробка загальної архітектури системи за допомогою діаграми розгортання.

2.4. Визначити об'єктно-орієнтовану модель вебпроєкту.

3. Реалізація вебсайту.

3.1. Розробка фронтенд-частини.

3.1.1. Ініціалізація та встановлення необхідних залежностей для фронтенд-частини вебдодатку.

3.1.2. Створення вхідних файлів та головного компонента вебпроєкту.

3.1.3. Створення стилів для сторінок на вебсайті.

3.2. Розробка бекенд-частини.

3.2.1. ініціалізація та налаштування необхідних залежностей для бекенд-частини вебдодатку.

3.2.2. Створення головного файлу вебпроєкту.

3.2.3. Розробка конфігураційних файлів для бази даних, системи аутентифікації та завантаження зображень.

3.2.4. Розробка моделей вебсистеми.

3.2.5. Розробка контролерів вебсистеми.

3.2.6. Розробка шляхів вебсистеми.

4. Тестування вебдодатку.

4.1. Тестування усіх форм аунтефікації.

4.2. Тестування фільтрів/сортування.

4.3. Тестування адмін форм для додавання/редагування/видалення.

4.4. Тестування доступів.

4.5. Тестування кошика та особистого кабінету користувача.

4.6. Тестування процесу покупки.

4.7. Тестування захищеності шляхів та маршрутів вебсистеми.

4.8. Тестування API.

5. Розгортання системи управління контентом для художньої галереї.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 11   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

Результат реалізації поставленого завдання – реалізована система управління контентом для художньої галереї, що надає можливість для додавання/редагування/видалення/покупки контенту пов'язаного з художньою художньою галереєю.

## 1.2 Аналіз аналогів програмного продукту

Методи Для розробки повноцінної CMS художньої галереї потрібно переглянути існуючі розробки. Було знайдено 2 вебпроекти на дану тематику. Проаналізуємо їх детально.

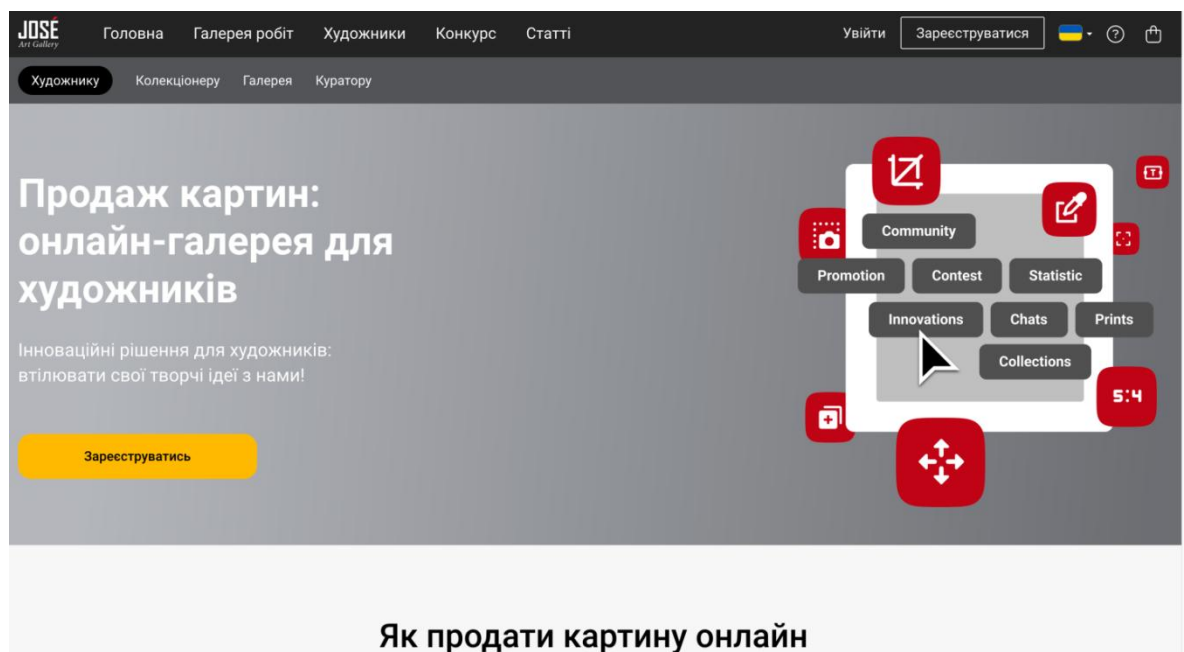


Рисунок 1.1 - Головна сторінка сайту «Jose Art Gallery»

Першим вебсайтом на розгляд є «Jose Art Gallery» [4]. Jose Art Gallery – це вебсайт, який надає платформу для художників та колекціонерів мистецтва, дозволяючи їм представити та продавати свої роботи, а також взаємодіяти з іншими користувачами (рис. 1.2).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 12   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

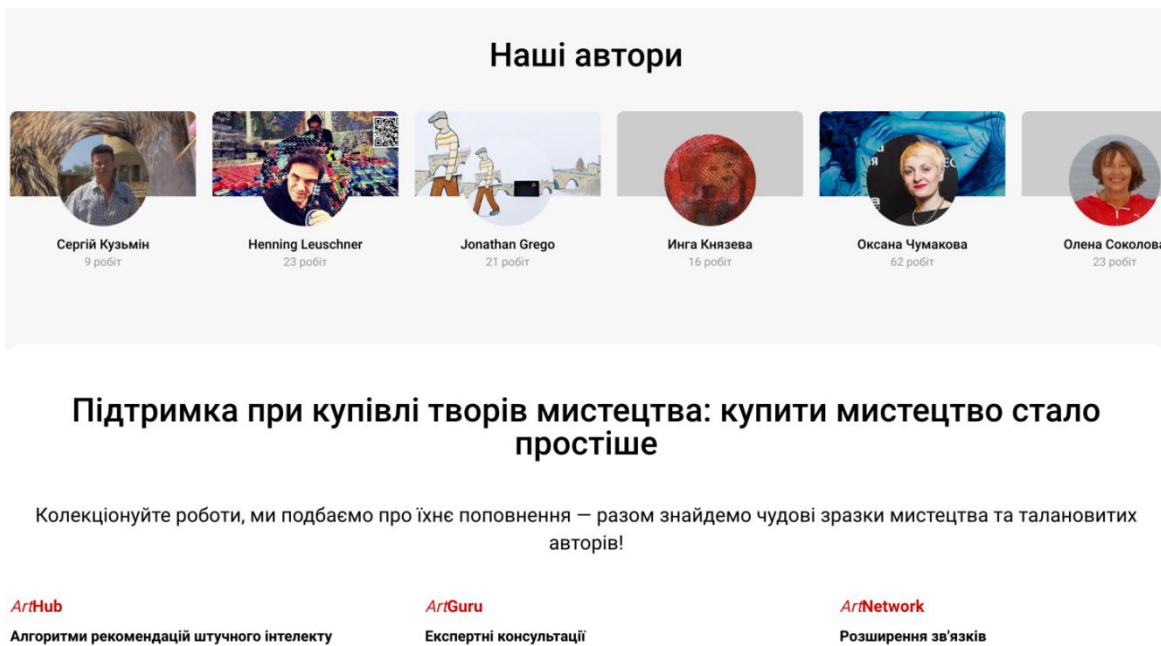


Рисунок. 1.2 - Взаємодія з іншими користувачами «Jose Art Gallery»

Для переходу на сторінку конкретного художника або роботи користувач може скористатися пошуком або навігацією через відповідні категорії з головної сторінки. Після цього він перейде на сторінку художника або конкретної роботи (рис. 1.3). На сторінці роботи представлено зображення художнього твору, його опис, інформацію про автора та ціни.

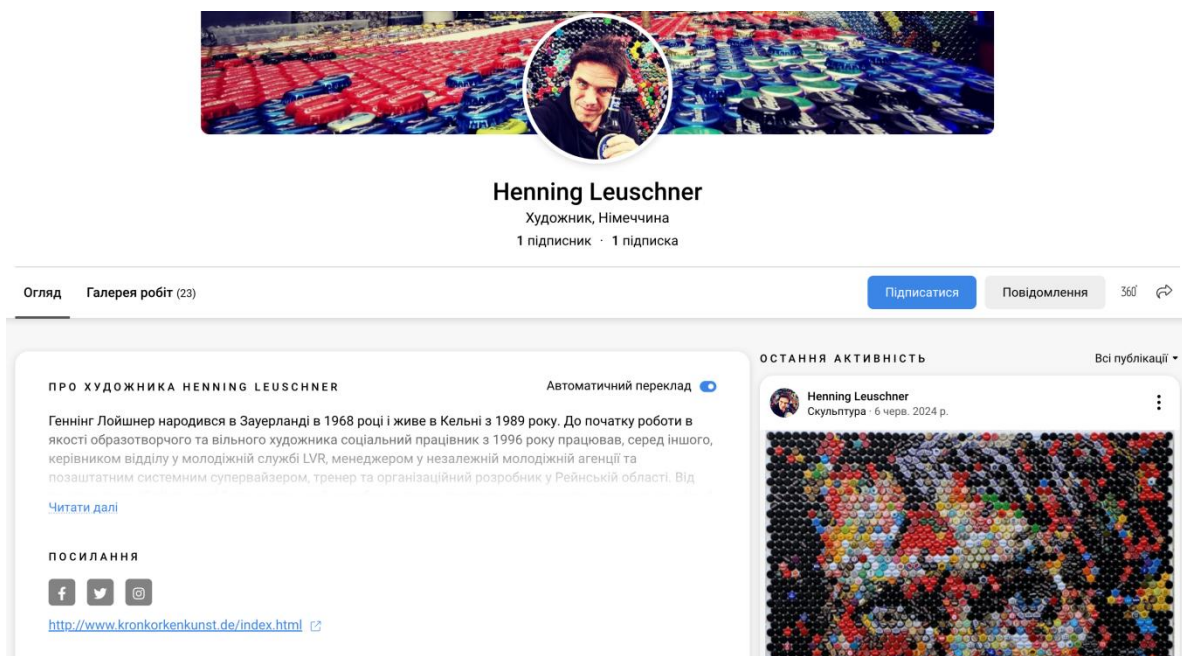


Рисунок 1.3 - Сторінка художника «Jose Art Gallery»

Jose Art Gallery також містить декілька секцій, включаючи блоги, новини, особисті кабінети користувачів, відгуки, можливості редагування картин, додавання художніх жанрів та авторів (рис. 1.4). Користувачі можуть оцінювати роботи, залишати відгуки та брати участь в обговореннях, що створює інтерактивне середовище для спільноти шанувальників мистецтва.



Рисунок 1.4 - Сторінка художнього твору «Jose Art Gallery»

Однією з переваг вебсайту є його добре структурований інтерфейс, що полегшує навігацію та пошук необхідної інформації. Окрім цього, можливість взаємодії між користувачами додає цінності платформі, дозволяючи художникам отримувати відгуки та підтримувати зв'язок з потенційними покупцями.

Проте є і деякі недоліки. Наприклад, сторінки можуть завантажуватися повільно через велику кількість зображень високої роздільної здатності. Крім того, деякі користувачі можуть вважати інтерфейс перевантаженим через велику кількість функціональних елементів на одній сторінці.

В цілому, Jose Art Gallery є потужним інструментом для управління художніми творами та взаємодії між митцями і колекціонерами. Вебсайт забезпечує зручний доступ до інформації про твори мистецтва та художників,

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 14   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

створюючи сприятливі умови для розвитку та популяризації сучасного мистецтва.

Останнім вебсайтом на розгляд є «Naturalist Gallery» [5]. Naturalist Gallery – це онлайн-платформа, присвячена експозиції та продажу художніх творів, переважно з тематики природи. Вебсайт дозволяє художникам представляти свої роботи, а колекціонерам – придбати їх або просто насолодитися мистецтвом (рис. 1.5).

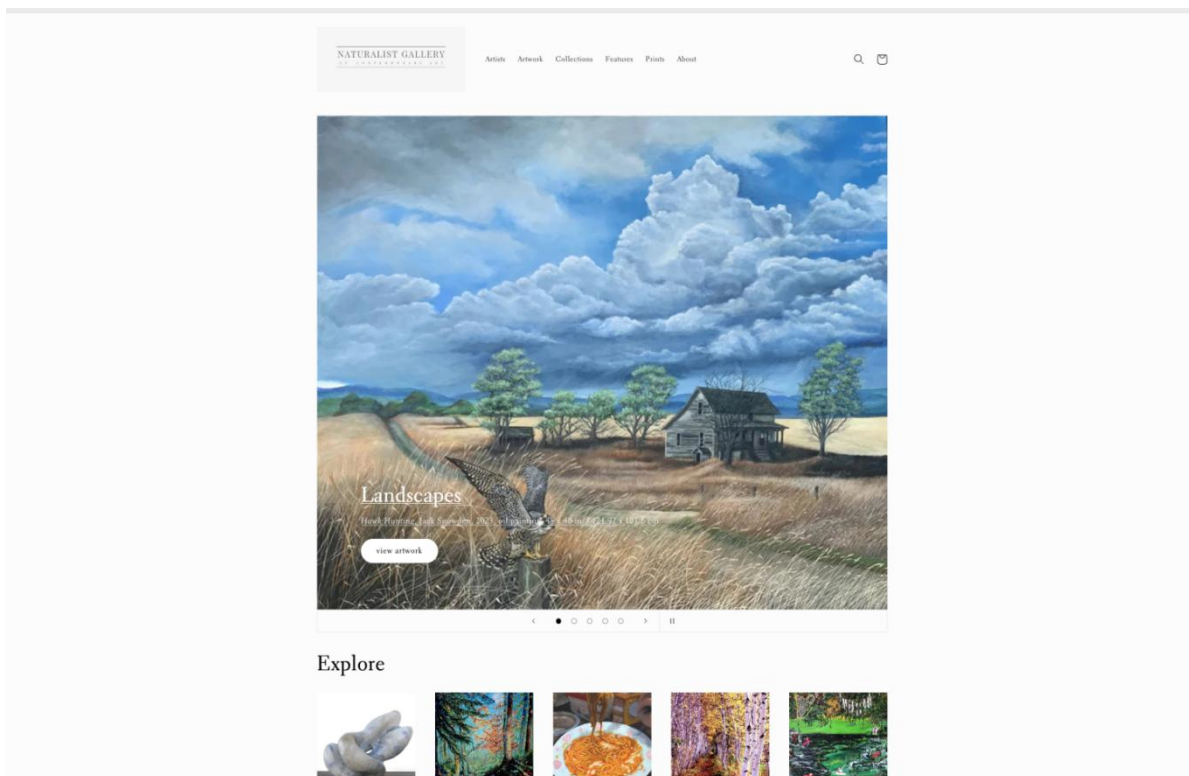


Рисунок 1.5 - Головна сторінка «Naturalist Gallery»

Головна сторінка вебсайту містить інформацію про останні додані роботи, рекомендованих художників, а також блоги та новини галереї. Для переходу на сторінку конкретного художника або твору користувач може скористатися пошуком або переглянути категорії з головної сторінки. На сторінці роботи представлено зображення художнього твору, його опис, інформацію про автора та ціни (рис. 1.6).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 15   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

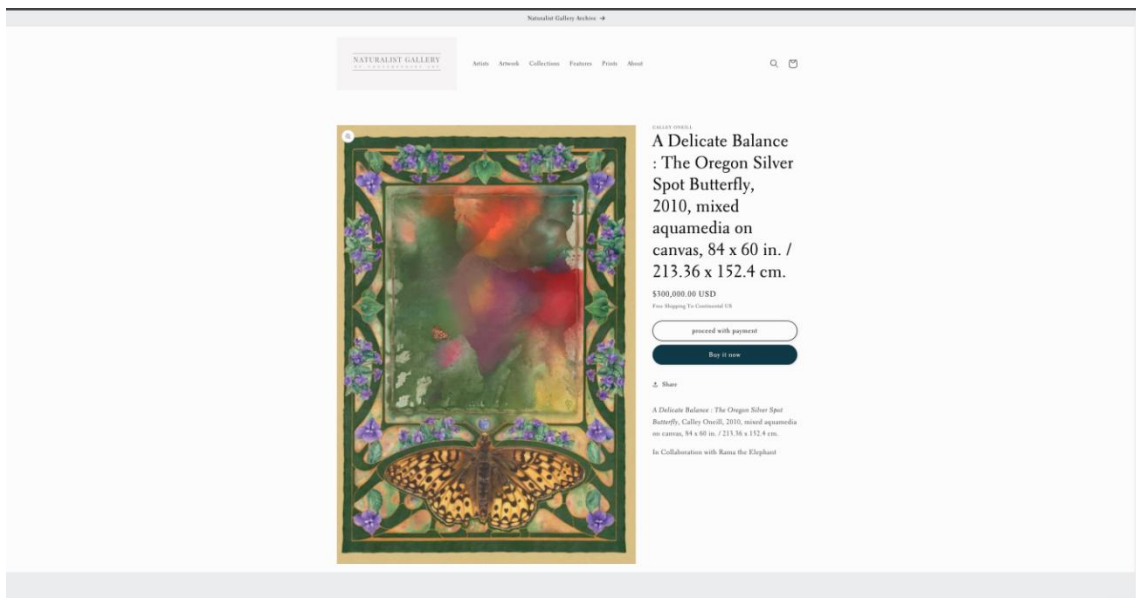


Рисунок 1.6 - Сторінка художнього твору «Naturalist Gallery»

Однією з особливостей вебсайту є секція блогів та новин, де публікуються статті про мистецтво та природу, інтерв'ю з художниками та огляди виставок. Це додає цінності платформі, залучаючи користувачів до активної взаємодії з контентом.

Користувачі також мають можливість залишати відгуки про твори мистецтва та брати участь в обговореннях. Інтерфейс вебсайту добре структурований, що забезпечує зручність навігації та пошуку необхідної інформації.

Проте, є деякі недоліки. Наприклад, швидкість завантаження сторінок може бути уповільненою через великий обсяг зображень високої якості. Крім того, іноді користувачі можуть відчувати перевантаження візуальним контентом, що ускладнює сприйняття інформації.

В цілому, Naturalist Gallery є потужним інструментом для презентації та продажу художніх творів, пов'язаних з природою. Вебсайт надає зручний доступ до інформації про твори мистецтва та художників, створюючи сприятливі умови для популяризації мистецтва природи.

Аналіз сайтів «Naturalist Gallery» та «Jose Art Gallery» виявив низку переваг і недоліків. Обидва сайти пропонують зручні та добре структуровані інтерфейси, багатий функціонал, що включає блоги, новини, відгуки та

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 16   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

обговорення, що сприяє активній взаємодії користувачів. Однак, вони мають спільні проблеми зі швидкістю завантаження через великий обсяг мультимедійних файлів та можливе перевантаження візуальним контентом. Загалом, ці платформи є ефективними інструментами для презентації та популяризації мистецтва, але потребують оптимізації для покращення користувацького досвіду.

### 1.3 Інструментальні засоби реалізації програмного продукту

Для того, щоб реалізувати систему управління контентом для художньої галереї, було обрано наступні інструментальні засоби:

- **PHP** є одною з найпопулярніших мов програмування для розробки вебдодатків. Версія 8.1 забезпечує покращену продуктивність, нові функціональні можливості та підвищену безпеку. Основні нововведення включають покращену обробку помилок, підтримку нових типів даних та функції асинхронного програмування. PHP використовується для серверної частини додатків, обробки форм, взаємодії з базами даних та управління сесіями [6];

- **Laravel** – це сучасний фреймворк PHP, який забезпечує швидкий і зручний процес розробки вебдодатків. Він надає широкий спектр інструментів та бібліотек для реалізації різноманітних функціональних вимог, таких як аутентифікація, маршрутизація, управління базами даних, черги завдань та багато іншого. Laravel підтримує архітектуру MVC (Model-View-Controller), що сприяє розділенню логіки програми та її представлення [7];

- **MySQL** – це система керування реляційними базами даних з відкритим вихідним кодом. Вона забезпечує швидке та надійне збереження та обробку даних. MySQL підтримує ACID транзакції, що гарантує цілісність даних, і має високий рівень масштабованості, що робить її ідеальним вибором для складних вебдодатків. MySQL також пропонує потужні інструменти для резервного копіювання та відновлення даних, реплікації та шардінгу [8];

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 17   |

– **Angular** – це платформа для розробки клієнтських додатків, яка використовує TypeScript. Вона забезпечує високопродуктивні та масштабовані вебзастосунки з компонентною архітектурою, що сприяє розділенню коду та його повторному використанню. Angular включає вбудовані рішення для маршрутизації, обробки форм, HTTP-запитів та іншого, що значно прискорює процес розробки [9];

– **TypeScript** – це надмножина JavaScript, яка додає статичну типізацію до мови. Використання TypeScript підвищує надійність і масштабованість коду, забезпечуючи кращу підтримку великих проектів. TypeScript компілюється у чистий JavaScript, що дозволяє використовувати його у будь-яких браузерах чи середовищах виконання [10];

– **JavaScript** – це мова програмування, що використовується для створення інтерактивних елементів на вебсторінках. Вона дозволяє додавати динамічні елементи, обробляти події користувача, змінювати вміст сторінки без її перезавантаження та створювати складні анімації. JavaScript є основою для багатьох сучасних фреймворків та бібліотек, таких як Angular, React та Vue.js [11];

– **SCSS** – це препроцесор CSS, який розширює можливості стандартного CSS, дозволяючи використовувати змінні, вкладені правила, міксини, функції та інші потужні механізми, що значно спрощують розробку стилів. SCSS компілюється у звичайний CSS, який підтримується всіма браузерами [12];

– **CSS (Cascading Style Sheets)** використовується для опису зовнішнього вигляду елементів на вебсторінці. Це мова стилів, яка визначає, як елементи HTML відображаються на екрані, у друці або інших медіа. CSS дозволяє встановлювати кольори, шрифти, відступи, розміри, розташування та багато інших аспектів зовнішнього вигляду вебсторінок [13];

– **Bootstrap** – це популярний фреймворк для розробки адаптивних та привабливих вебсайтів. Він містить готові компоненти та стилі, що значно спрощує процес розробки. Bootstrap забезпечує сумісність з різними

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 18   |

браузерами та пристроями, що дозволяє створювати сайти, які виглядають добре на будь-якому екрані [14].

### Висновки до першого розділу

Отже, в першому розділі було здійснено аналіз системи управління контентом (CMS) для художньої галереї та затверджено загальне її бачення.

У цьому розділі було поставлено загальне завдання та прописано основні етапи розробки вебпродукту:

- аналіз вебдодатку;
- проектування системи управління контентом для художньої галереї;
- реалізація вебпроєкту;
- тестування вебсайту;
- розгортання вебдодатку.

Наступним кроком було проаналізовано існуючі аналогічні розробки. Для цього було обрано два вебсайти: Jose Art Gallery та Naturalist Gallery. Проаналізувавши ці розробки, було зроблено висновки, що на вебсайті потрібно: забезпечити зручне керування колекцією художніх робіт; полегшити процес публікації нових експонатів; створити читабельний і привабливий дизайн; забезпечити інтерактивність через відгуки та обговорення; підтримати блог та новини; додати функціонал для управління художніми жанрами та авторами.

Для реалізації системи управління контентом для художньої галереї в даному розділі було вирішено використовувати наступне:

- мову програмування PHP 8.1;
- фреймворк Laravel 10;
- систему керування базами даних MySQL;
- платформу для розробки клієнтських додатків Angular;
- мову програмування TypeScript;

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 19   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

- мову програмування JavaScript;
- препроцесор SCSS (Sass);
- мову стилів CSS;
- фреймворк для створення адаптивного дизайну Bootstrap.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 20   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ДЛЯ ХУДОЖНЬОЇ ГАЛЕРЕЇ

### 2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

#### **Бізнес-вимоги:**

1. Основні цілі: веб-проект створюється з метою надати художній галереї сучасну та ефективну платформу для управління контентом, яка дозволить зручно керувати колекцією художніх робіт, покращити взаємодію з відвідувачами та збільшити охоплення аудиторії через онлайн-платформу.

2. Представлення проєкту: проєкт буде реалізовано у вигляді вебсайту, що містить структуровану інформацію про художні твори, авторів, жанри, а також новини та блоги, пов'язані з діяльністю галереї та мистецтвом.

Дотримання ознак системи управління контентом: вебсайт буде створений у вигляді системи управління контентом, яка включатиме наступні можливості: **управління колекцією**: можливість додавання, редагування та видалення експонатів, їх категоризація за жанрами, авторами та іншими критеріями; **персональні сторінки авторів**: створення сторінок художників з інформацією про їх творчість, виставки та біографію; **новини та блоги**: публікація новин, статей та блогів про мистецтво, виставки та події галереї; **фільтрація**: зручний пошук та фільтрація експонатів за різними критеріями. **кабінет**: персональні кабінети для користувачів; **API документація** для використання контенту вебдодатку на сторонніх сервісах.

#### **Вимоги користувачів:**

##### 1. Зовнішні користувачі:

- 1.1. Вхід на сайт.
- 1.2. Реєстрація на сайт.
- 1.3. Перегляд/додавання відгуків про галерею.
- 1.4. Додати/змінити про себе інформацію
- 1.5. Перегляд/купівля експонатів.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 21   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

- 1.6. Перегляд новин/блогів.
- 1.7. Перегляд/використання API документації.
- 1.8. Фільтрування/пошук ексконатів.
- 1.9. Перегляд художніх жанрів/художників.
- 1.10. Перегляд контент авторів.

2. *Внутрішній користувач – адміністратор:*

- 2.1. Додавання/редагування/видалення будь-якого контенту на вебсайті.
- 2.2. Перегляд аналітики.
- 2.3. Перегляд/редагування/видалення користувачів.
- 2.4. Перегляд логів.
- 2.5. Перегляд покупок.

3. *Характеристика об'єкта комп'ютеризації:*

Користувач на сайті буде мати можливість: переглядати та купувати картини; додавати і переглядати відгуки; переглядати новини і блоги; здійснювати фільтрацію або пошук по картинах; переглядати художників та художні жанри; використовувати API документацію для інтеграції на сторонні сервіси.

**Функціональні вимоги:**

1. Авторизація користувачів в системі: в системі повинна бути можливість реєстрації користувачів з присвоєнням їм відповідних ролей (звичайний користувач, адміністратор, супер-адмін).
2. Написання відгуків: на вебсайті має бути можливість для написання відгуку про сервіс.
3. Фільтрація: користувач повинен мати змогу знайти картину, художника, автора.
4. Наявність ролі адміністратора: на вебсайті має бути роль адміністратора, який може редагувати контент вебдодатку.

**Нефункціональні вимоги:**

1. Сприйняття:

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 22   |

1.1. Час відповіді системи для звичайних запитів не повинен перевищувати 3 секунд, а для більш складних запитів – 8 секунд.

1.2. Інтерфейс представлення ІС повинен бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки.

## 2. Надійність:

2.1. Середній час безперервної роботи – 30 робочих днів.

2.2. Максимальна норма помилок та дефектів в роботі системи – 1 помилка на 10000 запитів користувача.

## 3. Продуктивність:

Система повинна підтримувати мінімум 500 одночасно працюючих користувачів, пов'язаних зі спільною базою даних.

## 4. Можливість експлуатації

4.1. Масштабування – система повинна мати можливість збільшувати потужності (продуктивність) зі збільшенням користувачів таким чином, щоб це ніяк негативно не відобразилося на її роботі.

### **Системні вимоги:**

#### 1. Вимоги до середовища виконання:

Система повинна задовільняти вимогам на комп'ютері, що знаходиться в наступній мінімальній комплектації:

1.1. 4096 МБ ОЗП.

1.2. Процесор Intel Pentium 4 або новіший.

1.3. Для Windows: Операційна система Windows 10 або новіша.

#### 2. Вимоги до СУБД та доступу до даних:

2.1. У ядрі системи повинна бути представлення СУБД реляційного доступу.

### **Бізнес правила:**

Система повинна відповідати всім стандартам інтерфейса користувача Microsoft Windows.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 23   |

***Антивимоги:***

1. Не використовує інформацію, що може зашкодити здоров'ю користувачів.
2. Не використовує інформацію, заборонену нормами моралі та законодавством.

***Бізнес-логіка:***

1. Проєкт надає можливості для перегляду та покупки художніх робіт.
2. Вимоги до порядку впровадження: Впровадження проєкту має відбуватися поетапно, включаючи випуск альфа та бета версій.
  - 2.1. По закінченню відбудеться презентація проєкту.
  - 2.2. Попередньо випуск кінцевого варіанту проєкту призначено на 29.06.2024 року.
  - 2.3. Відкриття сайту.
3. Експлуатаційні вимоги:
  - 3.1. Інформація повинна завантажуватися без затримок та багів.

Аналіз вимог користувачів дав підстави для визначення варіантів використання CMS для художньої галереї. Для наглядної взаємодії між елементами додатку було створено діаграму варіантів використання (рис. 2.1).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 24   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

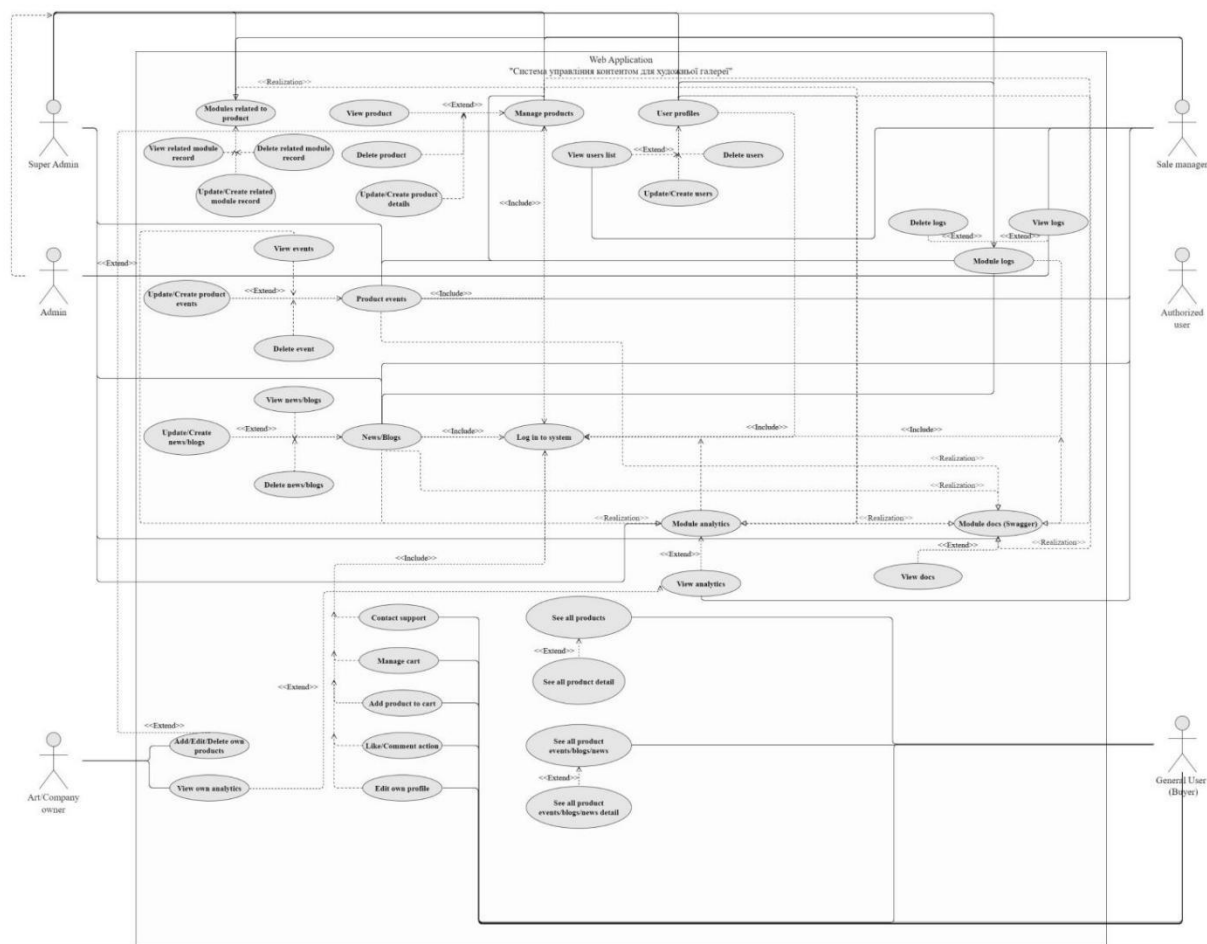


Рисунок 2.1 - Діаграма варіантів використання

Опис діаграми варіантів використання наведено в (табл. 2.1) – (табл. 2.7)

Таблиця 2.1

### Актори

| Актор           | Опис                                                                                                            |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| Super Admin     | Відповідає за загальне адміністрування системи та має всі права доступу.                                        |
| Admin           | Виконує завдання, пов'язані з управлінням контентом, такими як додавання/редагування продуктів, новин, подій т. |
| Sale Manager    | Відповідає за управління продажами, перегляд логів та аналітики.                                                |
| Authorized User | Зареєстрований користувач, який може взаємодіяти з контентом сайту.                                             |

## Продовження таблиці 2.1

|                   |                                                                                     |
|-------------------|-------------------------------------------------------------------------------------|
| Art Company Owner | Користувач, який має права на управління власними продуктами та перегляд аналітики. |
| General User      | Незареєстрований користувач, який може переглядати доступний контент сайту.         |

Таблиця 2.2

## Варіанти використання для Admin

| Варіант використання     | Опис                                                                           |
|--------------------------|--------------------------------------------------------------------------------|
| Module Artworks          | Керувати картинами (додавати, редагувати, видаляти).                           |
| Module Artist            | Керувати художниками (додавати, редагувати, видаляти).                         |
| Module Art Type          | Керувати художніми жанрами (додавати, редагувати, видаляти).                   |
| Module Blog/News/Authors | Керувати блогами, новинами, контент-авторами (додавати, редагувати, видаляти). |
| Module Users             | Керувати користувачами (редагувати, видаляти).                                 |
| Module Logs              | Переглядати загальні логи по сайту.                                            |
| Module Art Sales         | Переглядати статистику продажів по сайту.                                      |
| Module API Docs          | Переглядати та використовувати API для інтеграції в сторонні сервіси.          |
| Module Analytics         | Перегляд статистики по сайту.                                                  |
| Module Cart              | Управління корзиною, замовленнями, покупками. (Власними)                       |
| Module Comments          | Написання і перегляд усіх відгуків.                                            |

Таблиця 2.3

## Варіанти використання для Super Admin

| Варіант використання | Опис                                     |
|----------------------|------------------------------------------|
| All modules/actions  | Доступ до усіх дій та модулів в проєкті. |

Таблиця 2.4

## Варіанти використання для Sale Manager

| Варіант використання     | Опис                                                                  |
|--------------------------|-----------------------------------------------------------------------|
| Module Artworks          | Перегляд картин.                                                      |
| Module Artist            | Перегляд художників.                                                  |
| Module Art Type          | Перегляд художніх жанрів.                                             |
| Module Blog/News/Authors | Перегляд блогів, новин, контент-авторів.                              |
| Module Art Sales         | Переглядати статистику продажів по сайту.                             |
| Module API Docs          | Переглядати та використовувати API для інтеграції в сторонні сервіси. |
| Module Analytics         | Перегляд статистики по сайту.                                         |
| Module Cart              | Управління корзиною, замовленнями, покупками. (Власними)              |
| Module Comments          | Написання і перегляд усіх відгуків.                                   |
| Module Users             | Редагувати власний профіль.                                           |

Таблиця 2.5

## Варіанти використання для Authorized User

| Варіант використання     | Опис                                                                  |
|--------------------------|-----------------------------------------------------------------------|
| Module Artworks          | Перегляд картин.                                                      |
| Module Artist            | Перегляд художників                                                   |
| Module Art Type          | Перегляд художніх жанрів.                                             |
| Module Blog/News/Authors | Перегляд блогів, новин, контент-авторів                               |
| Module API Docs          | Переглядати та використовувати API для інтеграції в сторонні сервіси. |
| Module Cart              | Управління корзиною, замовленнями, покупками. (Власними)              |
| Module Comments          | Написання і перегляд відгуків.                                        |
| Module Users             | Редагувати власний профіль.                                           |

Таблиця 2.6

## Варіанти використання для Art Company Owner

| Варіант використання     | Опис                                                                               |
|--------------------------|------------------------------------------------------------------------------------|
| Module Artworks          | Перегляд картин. Додавання/редагування/видалення картин за компанією.              |
| Module Artist            | Перегляд художників.                                                               |
| Module Art Type          | Перегляд художніх жанрів.                                                          |
| Module Blog/News/Authors | Перегляд блогів, новин, контент-авторів                                            |
| Module API Docs          | Переглядати та використовувати API для інтеграції в сторонні сервіси.              |
| Module Cart              | Управління корзиною, замовленнями, покупками. (Власними та пов'язаних з компанією) |
| Module Comments          | Написання і перегляд відгуків. (Пов'язаних з компанією)                            |
| Module Users             | Редагувати власний профіль.                                                        |
| Module Art Sales         | Переглядати статистику продажів за компанією.                                      |
| Module Analytics         | Перегляд статистики за компанією.                                                  |

Таблиця 2.7

## Варіанти використання для Unauthorized User

| Варіант використання     | Опис                                    |
|--------------------------|-----------------------------------------|
| Module Artworks          | Перегляд картин.                        |
| Module Artist            | Перегляд художників.                    |
| Module Art Type          | Перегляд художніх жанрів.               |
| Module Blog/News/Authors | Перегляд блогів, новин, контент-авторів |

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать програмну реалізацію додатку. В додатку А наведено діаграму класів, яка використовується для його роботи.

Опишемо основні елементи діаграми:

### ***Контракти для відображення:***

- **ArtistShowContract**
  - **ShowArtistAction**
    - show(): відображає інформацію про художника.
  - **ShowUserAction**
    - show(): відображає інформацію про користувача.
- **ArtworkShowContract**
  - **ShowArtworkAction**
    - show(): відображає інформацію про твір мистецтва.
- **ArtTypeShowContract**
  - **ShowArtTypeAction**
    - show(): відображає інформацію про тип мистецтва.
- **AuthorShowContract**
  - **ShowSeoContentAuthorAction**
    - show(): відображає інформацію про автора.
- **BlogShowContract**
  - **ShowBlogAction**
    - show(): відображає інформацію про блог.
- **NewsShowContract**
  - **ShowNewsAction**
    - show(): відображає інформацію про новини.
- **UserLoginContract**
  - **UserLoginAction**
    - login(): виконує вхід користувача в систему.
- **UserLogoutContract**
  - **UserLogoutAction**
    - logout(): виконує вихід користувача з системи.
- **UserRegistrationContract**
  - **UserRegistrationAction**
    - register(): реєструє нового користувача.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 29   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

### *Сервіси:*

- **ArtistService**

- getAllArtists(): повертає список всіх художників;
- createArtist(): створює нового художника;
- updateArtist(): оновлює інформацію про художника;
- deleteArtist(): видаляє художника.

- **ArtworkService**

- getAllArtworks(): повертає список всіх творів мистецтва;
- createArtwork(): створює новий твір мистецтва;
- updateArtwork(): оновлює інформацію про твір мистецтва;
- deleteArtwork(): видаляє твір мистецтва.

- **ArtTypeService**

- getAllArtTypes(): повертає список всіх типів мистецтва;
- createArtType(): створює новий тип мистецтва;
- updateArtType(): оновлює інформацію про тип мистецтва;
- deleteArtType(): видаляє тип мистецтва.

- **UserCommentService**

- getAllUserComments(): повертає список всіх коментарів користувачів;
- createComment(): створює новий коментар;
- updateComment(): оновлює коментар;
- deleteComment(): видаляє коментар;
- getFirstSentence(): повертає перше речення коментаря.

- **SeoContentAuthorService**

- getAllAuthors(): повертає список всіх авторів SEO контенту;
- createAuthor(): створює нового автора;
- updateAuthor(): оновлює інформацію про автора;
- deleteAuthor(): видаляє автора.

- **BlogService**

- getAllBlogs(): повертає список всіх блогів;

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 30   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

- createBlog(): створює новий блог;
- updateBlog(): оновлює інформацію про блог;
- deleteBlog(): видаляє блог.
- **ImageService**
  - storeImage(): зберігає зображення;
  - deleteImage(): видаляє зображення;
  - updateImage(): оновлює зображення;
  - getAllImages(): повертає список всіх зображень.
- **NewsService**
  - getAllNews(): повертає список всіх новин;
  - createNews(): створює нову новину;
  - updateNews(): оновлює новину;
  - deleteNews(): видаляє новину.
- **ArtSaleService**
  - getAllArtSales(): повертає список всіх продажів мистецтва.
- **UserService**
  - getAllUsers(): повертає список всіх користувачів;
  - updateUser(): оновлює інформацію про користувача;
  - deleteUser(): видаляє користувача.

**Сервіси для логування:**

- **CreateGeneralLogService**
  - атрибути: dataBefore, dataAfter, fieldsToExclude, moduleName, moduleId, userId, action, changedFields;
  - методи:
    - \_\_construct(): конструктор класу;
    - setDataBefore(): встановлює дані до змін;
    - setDataAfter(): встановлює дані після змін;
    - setFieldsToExclude(): встановлює поля, які потрібно виключити;
    - setModuleName(): встановлює назву модуля;

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 31   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

- `setModuleId()`: встановлює ID модуля;
- `setUserId()`: встановлює ID користувача;
- `setAction()`: встановлює дію;
- `getChangedFields()`: повертає змінені поля.
- **AllGeneralLogService**
  - `__invoke()`: викликає виконання сервісу логування.
- **GetChangedFieldsAction**
  - атрибути: `dataBefore`, `dataAfter`, `fieldsToExclude`;
  - методи:
    - `invoke()`: викликає виконання дії;
    - `initiateGeneralLogChanges()`: ініціює зміни загального логування;
    - `setDataBefore()`: встановлює дані до змін;
    - `setDataAfter()`: встановлює дані після змін;
    - `setFieldsToExclude()`: встановлює поля, які потрібно виключити.
- **CreateGeneralLogContract**
- **AllGeneralLogContract**

**Трейти:**

- **ManagesMetaTagTrait**
  - використовується для управління мета-тегами.
- **HandlesImageUploadsTrait**
  - використовується для обробки завантаження зображень.
- **ArtworkFilterTrait**
  - атрибути: `page`, `sort`, `slug`, `status`, `only_main_img`, `art_type_id`, `subcategory_sort`, `max_price`, `bought`;
  - методи:
    - `filterFromRequest()`: фільтрує дані з запиту;
    - `getArtType()`: повертає тип мистецтва;
    - `getMainImg()`: повертає головне зображення;

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 32   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

- `getOnlyMainImg()`: повертає лише головне зображення;
- `getPrice()`: повертає ціну;
- `getStatus()`: повертає статус;
- `getSlug()`: повертає slug;
- `getSort()`: повертає сортування;
- `setPage()`: встановлює сторінку;
- `setStatus()`: встановлює статус;
- `toArray()`: перетворює об'єкт у масив.

## 2.2 Розробка бази даних системи

Для збереження даних та інформації у додатку використовується MySQL, яка є системою управління реляційними базами даних. Ця система забезпечує швидке та надійне зберігання і обробку даних, а також підтримує складні запити та транзакції.

База даних системи складається з наступних таблиць:

1. **art\_sales** – таблиця, яка зберігає дані про продажі мистецьких робіт.
2. **art\_types** – таблиця, яка зберігає інформацію про жанри мистецьких робіт.
3. **artists** – таблиця, яка зберігає інформацію про художників.
4. **artworks** – таблиця, яка зберігає інформацію про мистецькі роботи.
5. **blogs** – таблиця, яка зберігає інформацію про блоги.
6. **general\_log\_changes** – таблиця, яка зберігає зміни логів.
7. **general\_logs** – таблиця, яка зберігає загальні логи системи.
8. **images** – таблиця, яка зберігає інформацію про зображення.
9. **migrations** – таблиця, яка зберігає інформацію про міграції бази даних.
10. **news** – таблиця, яка зберігає інформацію про новини.
11. **oauth\_access\_tokens** – таблиця, яка зберігає токени доступу OAuth.
12. **oauth\_personal\_access\_clients** – таблиця, яка зберігає клієнтів особистого доступу OAuth.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 33   |

13. **page\_meta\_tags** – таблиця, яка зберігає мета теги сторінок.
14. **personal\_access\_tokens** – таблиця, яка зберігає особисті токени доступу.
15. **seo\_content\_authors** – таблиця, яка зберігає інформацію про авторів SEO контенту.
16. **subscription\_items** – таблиця, яка зберігає інформацію про елементи оплати.
17. **subscriptions** – таблиця, яка зберігає інформацію про оплати.
18. **user\_comments** – таблиця, яка зберігає коментарі користувачів.
19. **user\_roles** – таблиця, яка зберігає ролі користувачів.
20. **users** – таблиця, яка зберігає дані користувачів.

Опис кожної з таблиць бази даних наведений нижче у (табл. 2.8) - (табл. 2.22)

Таблиця 2.8

Опис таблиці art\_sales

| Назва поля        | Тип даних          | ПК | ЗК | Опис                             |
|-------------------|--------------------|----|----|----------------------------------|
| id                | BIGINT<br>UNSIGNED | +  | -  | Унікальний ідентифікатор продажу |
| artwork_id        | INT                | -  | +  | Ідентифікатор мистецької роботи  |
| buyer_id          | INT                | -  | +  | Ідентифікатор покупця            |
| sale_price        | DOUBLE(8,2)        | -  | -  | Ціна продажу                     |
| payment_method    | VARCHAR(255)       | -  | -  | Метод оплати                     |
| phone             | VARCHAR(255)       | -  | -  | Телефон                          |
| delivery_location | VARCHAR(255)       | -  | -  | Адреса доставки                  |
| status            | TINYINT(1)         | -  | -  | Статус продажу                   |
| created_at        | TIMESTAMP          | -  | -  | Дата створення запису            |

## Продовження таблиці 2.8

|            |           |   |   |                       |
|------------|-----------|---|---|-----------------------|
| updated_at | TIMESTAMP | - | - | Дата оновлення запису |
|------------|-----------|---|---|-----------------------|

Таблиця 2.9

## Опис таблиці art\_types

| Назва поля | Тип даних       | ПК | ЗК | Опис                           |
|------------|-----------------|----|----|--------------------------------|
| id         | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор жанру |
| name       | VARCHAR(255)    | -  | -  | Назва жанру                    |
| slug       | VARCHAR(255)    | -  | -  | Унікальний ключ для жанру      |
| status     | TINYINT(1)      | -  | -  | Статус жанру                   |
| is_deleted | TINYINT(1)      | -  | -  | Статус видалення               |
| created_at | TIMESTAMP       | -  | -  | Дата створення запису          |
| updated_at | TIMESTAMP       | -  | -  | Дата оновлення запису          |

Таблиця 2.10

## Опис таблиці artists

| Назва поля  | Тип даних       | ПК | ЗК | Опис                               |
|-------------|-----------------|----|----|------------------------------------|
| id          | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор художника |
| name        | VARCHAR(255)    | -  | -  | Ім'я художника                     |
| slug        | VARCHAR(255)    | -  | -  | Унікальний ключ для художника      |
| birth_date  | DATE            | -  | -  | Дата народження                    |
| death_date  | DATE            | -  | -  | Дата смерті                        |
| nationality | VARCHAR(255)    | -  | -  | Національність                     |

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 35   |

## Продовження таблиці 2.10

|            |            |   |   |                       |
|------------|------------|---|---|-----------------------|
| biography  | TEXT       | - | - | Біографія             |
| status     | TINYINT(1) | - | - | Статус художника      |
| is_deleted | TINYINT(1) | - | - | Статус видалення      |
| created_at | TIMESTAMP  | - | - | Дата створення запису |
| updated_at | TIMESTAMP  | - | - | Дата оновлення запису |

Таблиця 2.11

## Опис таблиці artworks

| Назва поля       | Тип даних          | ПК | ЗК | Опис                             |
|------------------|--------------------|----|----|----------------------------------|
| id               | BIGINT<br>UNSIGNED | +  | -  | Унікальний ідентифікатор картини |
| slug             | VARCHAR(255)       | -  | -  | Унікальний ключ для картини      |
| artist_id        | INT                | -  | +  | Ідентифікатор художника          |
| type_id          | INT                | -  | +  | Ідентифікатор жанру              |
| title            | VARCHAR(255)       | -  | -  | Назва картини                    |
| description      | TEXT               | -  | -  | Опис картини                     |
| biography        | TEXT               | -  | -  | Біографія                        |
| creation_date    | DATE               | -  | -  | Дата створення                   |
| acquisition_date | DATE               | -  | -  | Дата продажу                     |
| price            | DOUBLE(20,2)       | -  | -  | Ціна                             |
| status           | TINYINT(1)         | -  | -  | Статус картини                   |
| is_deleted       | TINYINT(1)         | -  | -  | Статус видалення                 |
| created_at       | TIMESTAMP          | -  | -  | Дата створення запису            |
| updated_at       | TIMESTAMP          | -  | -  | Дата оновлення запису            |

Таблиця 2.12

Опис таблиці blogs

| Назва поля      | Тип даних          | ПК | ЗК | Опис                           |
|-----------------|--------------------|----|----|--------------------------------|
| id              | BIGINT<br>UNSIGNED | +  | -  | Унікальний ідентифікатор блогу |
| slug            | VARCHAR(255)       | -  | -  | Унікальний ключ для блогу      |
| author_id       | INT                | -  | +  | Ідентифікатор автора           |
| name            | VARCHAR(255)       | -  | -  | Назва блогу                    |
| description     | TEXT               | -  | -  | Опис блогу                     |
| sub_description | TEXT               | -  | -  | Додатковий опис                |
| status          | TINYINT(1)         | -  | -  | Статус блогу                   |
| is_deleted      | TINYINT(1)         | -  | -  | Статус видалення               |
| created_at      | TIMESTAMP          | -  | -  | Дата створення запису          |
| updated_at      | TIMESTAMP          | -  | -  | Дата оновлення запису          |

Таблиця 2.13

Опис таблиці general\_log\_changes

| Назва поля | Тип даних          | ПК | ЗК | Опис                           |
|------------|--------------------|----|----|--------------------------------|
| change_id  | BIGINT<br>UNSIGNED | +  | -  | Унікальний ідентифікатор зміни |
| log_id     | INT                | -  | +  | Ідентифікатор лог-запису       |
| field_name | VARCHAR(255)       | -  | -  | Назва поля, що змінюється      |

Продовження таблиці 2.13

|            |              |   |   |                       |
|------------|--------------|---|---|-----------------------|
| old_value  | VARCHAR(255) | - | - | Старе значення        |
| new_value  | VARCHAR(255) | - | - | Нове значення         |
| created_at | TIMESTAMP    | - | - | Дата створення запису |
| updated_at | TIMESTAMP    | - | - | Дата оновлення запису |

Таблиця 2.14

Опис таблиці general\_logs

| Назва поля  | Тип даних       | ПК | ЗК | Опис                          |
|-------------|-----------------|----|----|-------------------------------|
| log_id      | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор логу |
| module_name | VARCHAR(255)    | -  | -  | Назва модулю                  |
| module_id   | INT             | -  | +  | Ідентифікатор модуля          |
| user_id     | INT             | -  | +  | Ідентифікатор користувача     |
| action      | VARCHAR(255)    | -  | -  | Опис дії                      |
| created_at  | TIMESTAMP       | -  | -  | Дата створення запису         |
| updated_at  | TIMESTAMP       | -  | -  | Дата оновлення запису         |

Таблиця 2.15

Опис таблиці images

| Назва поля | Тип даних       | ПК | ЗК | Опис                                |
|------------|-----------------|----|----|-------------------------------------|
| id         | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор зображення |
| module_id  | INT             | -  | +  | Ідентифікатор                       |

Продовження таблиці 2.15

|            |              |   |   |                                            |
|------------|--------------|---|---|--------------------------------------------|
| object_id  | INT          | - | + | Ідентифікатор<br>об'єкта                   |
| type       | VARCHAR(255) | - | - | Тип зображення                             |
| url        | VARCHAR(255) | - | - | URL зображення                             |
| main_img   | TINYINT(1)   | - | - | Основне<br>зображення (1 -<br>так, 0 - ні) |
| is_deleted | TINYINT(1)   | - | - | Статус видалення                           |
| created_at | TIMESTAMP    | - | - | Дата створення<br>запису                   |
| updated_at | TIMESTAMP    | - | - | Дата оновлення<br>запису                   |

Таблиця 2.16

Опис таблиці migrations

| Назва<br>поля | Тип даних          | ПК | ЗК | Опис                                      |
|---------------|--------------------|----|----|-------------------------------------------|
| id            | BIGINT<br>UNSIGNED | +  | -  | Унікальний<br>ідентифікатор<br>зображення |
| migration     | VARCHAR(255)       | -  | -  | Назва міграції                            |
| batch         | INT                | -  | -  | Номер партії<br>міграції                  |

Таблиця 2.17

Опис таблиці news

| Назва поля | Тип даних          | ПК | ЗК | Опис                                  |
|------------|--------------------|----|----|---------------------------------------|
| id         | BIGINT<br>UNSIGNED | +  | -  | Унікальний<br>ідентифікатор<br>новини |
| slug       | VARCHAR(255)       | -  | -  | Унікальний<br>ключ                    |

Продовження таблиці 2.17

|                 |              |   |   |                       |
|-----------------|--------------|---|---|-----------------------|
| author_id       | INT          | - | + | Ідентифікатор автора  |
| name            | VARCHAR(255) | - | - | Назва новини          |
| description     | TEXT         | - | - | Опис новини           |
| sub_description | TEXT         | - | - | Додатковий опис       |
| status          | TINYINT(1)   | - | - | Статус новини         |
| is_deleted      | TINYINT(1)   | - | - | Статус видалення      |
| created_at      | TIMESTAMP    | - | - | Дата створення запису |
| updated_at      | TIMESTAMP    | - | - | Дата оновлення запису |

Таблиця 2.18

Опис таблиці page\_meta\_tags

| Назва поля            | Тип даних       | ПК | ЗК | Опис                               |
|-----------------------|-----------------|----|----|------------------------------------|
| id                    | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор мета тегу |
| module_id             | INT             | -  | +  | Ідентифікатор модуля               |
| object_id             | INT             | -  | +  | Ідентифікатор об'єкта              |
| page_title            | VARCHAR(255)    | -  | -  | Заголовок сторінки                 |
| page_meta_description | TEXT            | -  | -  | Мета опис сторінки                 |
| h1_title              | VARCHAR(255)    | -  | -  | Заголовок H1                       |
| description           | TEXT            | -  | -  | Опис                               |
| is_deleted            | TINYINT(1)      | -  | -  | Статус видалення                   |

## Продовження таблиці 2.18

|            |           |   |   |                       |
|------------|-----------|---|---|-----------------------|
| created_at | TIMESTAMP | - | - | Дата створення запису |
| updated_at | TIMESTAMP | - | - | Дата оновлення запису |

Таблиця 2.19

## Опис таблиці seo\_content\_authors

| Назва поля  | Тип даних       | ПК | ЗК | Опис                                    |
|-------------|-----------------|----|----|-----------------------------------------|
| id          | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор контент-автора |
| name        | VARCHAR(255)    | -  | -  | Ім'я контент-автора                     |
| slug        | VARCHAR(255)    | -  | -  | Унікальний ключ для контент-автора      |
| birth_date  | DATE            | -  | -  | Дата народження                         |
| death_date  | DATE            | -  | -  | Дата смерті                             |
| nationality | VARCHAR(255)    | -  | -  | Національність                          |
| biography   | TEXT            | -  | -  | Біографія                               |
| status      | TINYINT(1)      | -  | -  | Статус контент-автора                   |
| is_deleted  | TINYINT(1)      | -  | -  | Статус видалення                        |
| created_at  | TIMESTAMP       | -  | -  | Дата створення запису                   |
| updated_at  | TIMESTAMP       | -  | -  | Дата оновлення запису                   |

Таблиця 2.20

Опис таблиці user\_comments

| Назва поля | Тип даних       | ПК | ЗК | Опис                             |
|------------|-----------------|----|----|----------------------------------|
| id         | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор відгуку |
| user_id    | INT             | -  | +  | Ідентифікатор користувача        |
| module_id  | INT             | -  | +  | Ідентифікатор модуля             |
| comment    | TEXT            | -  | -  | Текст відгуку                    |
| rating     | DOUBLE(8,2)     | -  | -  | Рейтинг                          |
| status     | TINYINT(1)      | -  | -  | Статус відгуку                   |
| is_deleted | TINYINT(1)      | -  | -  | Статус видалення                 |
| created_at | TIMESTAMP       | -  | -  | Дата створення запису            |
| updated_at | TIMESTAMP       | -  | -  | Дата оновлення запису            |

Таблиця 2.21

Опис таблиці user\_roles

| Назва поля  | Тип даних       | ПК | ЗК | Опис                     |
|-------------|-----------------|----|----|--------------------------|
| id          | BIGINT UNSIGNED | +  | -  | Унікальний ідентифікатор |
| name        | VARCHAR(255)    | -  | -  | Назва ролі               |
| description | VARCHAR(255)    | -  | -  | Опис ролі                |
| status      | TINYINT(1)      | -  | -  | Статус ролі              |

Таблиця 2.22

Опис таблиці user\_roles

| Назва поля   | Тип даних          | ПК | ЗК | Опис                                 |
|--------------|--------------------|----|----|--------------------------------------|
| id           | BIGINT<br>UNSIGNED | +  | -  | Унікальний ідентифікатор користувача |
| user_role_id | INT                | -  | +  | Ідентифікатор ролі користувача       |
| name         | VARCHAR(255)       | -  | -  | Ім'я користувача                     |
| email        | VARCHAR(255)       | -  | -  | Електронна пошта користувача         |
| password     | VARCHAR(255)       | -  | -  | Пароль                               |
| status       | TINYINT(1)         | -  | -  | Статус відгуку                       |
| is_deleted   | TINYINT(1)         | -  | -  | Статус видалення                     |
| created_at   | TIMESTAMP          | -  | -  | Дата створення запису                |
| stripe_id    | VARCHAR(255)       | -  | +  | Ідентифікатор Stripe                 |

### 2.3 Проектування та реалізація алгоритмів роботи системи

Додаток є зручним у використанні та забезпечує всі основні функції платформи, що дозволяє користувачам легко взаємодіяти з нею. Алгоритм роботи з платформою (рис. 2.2), а також детально описано його пункти нижче.

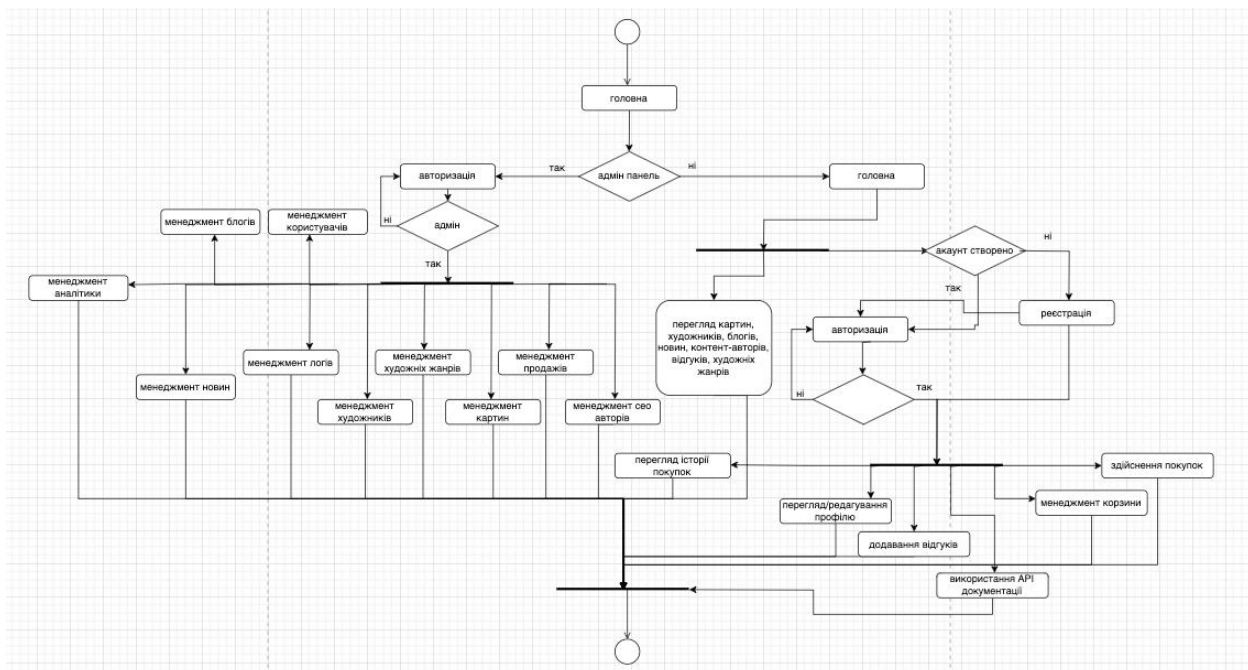


Рисунок 2.2 – Діаграма класів

Ця діаграма активностей відображає логіку роботи системи управління контентом для художньої галереї та допомагає краще зрозуміти процеси розробки та використання додатку.

У розробці системи були використані принципи SOLID, зокрема принцип єдиної відповідальності (SRP), який дозволяє уникнути змішування різних функцій у межах одного класу, що підвищує зрозумілість та підтримуваність коду [15]. Принцип DRY (Don't Repeat Yourself) був застосований для мінімізації дублювання коду, що спрощує подальшу підтримку та розширення системи [16]. Принцип KISS (Keep It Simple, Stupid) допоміг забезпечити простоту та зрозумілість реалізації, що полегшує сприйняття коду новими розробниками [17]. Використання шаблонів проектування, таких як Singleton та Factory, підвищило гнучкість та масштабованість системи [18].

#### Якщо авторизувався адміністратор:

- після авторизації адміністратор потрапляє до адміністративної панелі, яка розміщена на окремому сервері;
- адміністратор має можливість створювати новий контент, редагувати існуючі категорії та видаляти його за потреби;

|      |      |               |        |      |
|------|------|---------------|--------|------|
|      |      | Савченко Д.П. |        |      |
|      |      | Кубрак Ю.О.   |        |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |

- також адміністратор може переглядати аналітику, статистику та логи сайту;
- за потреби адміністратор може змінювати особисті дані користувачів, включно з паролем.

**Якщо авторизувався сейлз-менеджер:**

- після авторизації сейлз-менеджер також потрапляє до адміністративної панелі на окремому сервері;
- сейлз-менеджер має доступ до читання контенту сайту;
- сейлз-менеджер може переглядати аналітику, статистику та логи сайту.

**Якщо авторизувався користувач:**

- після авторизації користувач потрапляє до особистого кабінету на веб-додатку;
- користувач має можливість переглядати активні мистецькі роботи, доступні для придбання;
- користувач може додавати вподобані роботи до кошика, створювати замовлення та здійснювати оплату;
- користувач також може залишати відгуки на сайті і переглядати відгуки інших користувачів;
- у кабінеті користувача доступна можливість перегляду історії замовлень та стану поточних замовлень;
- також користувач може переглядати блоги, новини, сторінки художників, контент-авторів та будь-які інші контент-сторінки.

Таким чином, платформа забезпечує різні рівні доступу та функціональності для адміністраторів, сейлз-менеджерів та звичайних користувачів, що дозволяє ефективно керувати художніми роботами, продажами та взаємодією з клієнтами.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 45   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## 2.4 Реалізація функціоналу системи управління контентом художньої галереї

Контролер `ArtworksController` відповідає за управління функціоналом, пов'язаним з художніми роботами в системі. Він забезпечує CRUD операції (створення, читання, оновлення та видалення) для мистецьких робіт. Контролер використовує сервіс `ArtworkService` для обробки бізнес-логіки.

### **Основні методи:**

1. **Конструктор** класу `ArtworksController` ініціалізує об'єкт сервісу `ArtworkService`, який використовується для обробки бізнес-логіки, пов'язаної з мистецькими роботами. Це дозволяє відокремити логіку обробки даних від контролера, роблячи код більш організованим і підтримуваним (рис. 2.3).

```
public function __construct(ArtworkService $artworkService)
{
    $this->artworkService = $artworkService;
}
```

Рисунок 2.3 – Конструктор класу

2. Метод `index` відповідає за отримання списку всіх мистецьких робіт. Він приймає запит `IndexArtworkRequest`, який містить параметри фільтрації та пагінації. Метод викликає сервіс `ArtworkService` для отримання даних, обробляє їх та повертає у форматі JSON (рис. 2.4).

```
public function index(IndexArtworkRequest $request): JsonResponse
{
    return $this->defaultControllerHandler(function () use ($request) {
        $data = $this->artworkService->getAllArtworks($request->validated());
        return response()->json(['success' => true, 'data' => $data, 'message' => 'Усі картини.']);
    }, ModuleType::Artwork()->key);
}
```

Рисунок 2.4 – Отримання усіх мистецьких робіт

3. Метод `show` призначений для отримання детальної інформації про конкретну картину за її ідентифікатором. Він приймає параметр `id` з маршруту

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 46   |

запиту та викликає дію ShowArtworkAction для отримання відповідних даних. Результат повертається у форматі JSON (рис. 2.5).

```
public function show(Request $request, ShowArtworkAction $showArtworkAction):
JsonResponse
{
    return $this->defaultControllerHandler(function () use ($showArtworkAction, $request) {
        $data = $showArtworkAction->show(['id' => $request->route('id')]);
        return response()->json([
            'success' => true,
            'data' => [
                'data' => $data['data'],
                'additional_data' => $data['additional_data'],
            ],
            'message' => 'Детальна інформація про картину.'
        ]);
    }, ModuleType::Artwork()->key);
}
```

Рисунок 2.5 – Отримання інформації про конкретну картину

4. Метод store обробляє запити на створення нової мистецької роботи. Він приймає дані через запит StoreArtworkRequest, передає їх до сервісу ArtworkService для обробки та збереження, і повертає результат у форматі JSON. Успішне створення нової картини підтверджується відповідним повідомленням (рис. 2.6).

```
public function store(StoreArtworkRequest $request): JsonResponse
{
    return $this->defaultControllerHandler(function () use ($request) {
        $data = $this->artworkService->createArtwork($request->validated());
        return response()->json(['success' => $data['success'], 'data' => $data['data'], 'message'
=> 'Картину успішно створено.']);
    }, ModuleType::Artwork()->key);
}
```

Рисунок 2.6 – Обробка запиту на створення картини

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 47   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

5. Метод `update` відповідає за оновлення інформації про існуючу картину. Він приймає дані через запит `UpdateArtworkRequest`, доповнює їх ідентифікатором картини з маршруту, та передає їх до сервісу `ArtworkService` для обробки. Після успішного оновлення, результат повертається у форматі JSON (рис. 2.7).

```
public function update(UpdateArtworkRequest $request): JsonResponse
{
    return $this->defaultControllerHandler(function () use ($request) {
        $data = $this->artworkService->updateArtwork(array_merge($request->validated(),
['id' => $request->route('id')]));
        return response()->json(['success' => $data['success'], 'data' => $data['data'], 'message'
=> 'Картину успішно оновлено.']);
    }, ModuleType::Artwork()->key);
}
```

Рисунок 2.7 – Обробка запиту на оновлення картини

6. Метод `destroy` обробляє запити на видалення картини. Він приймає ідентифікатор картини через запит та передає його до сервісу `ArtworkService` для обробки видалення. Після успішного видалення, метод повертає результат у форматі JSON із відповідним повідомленням (рис. 2.8).

```
public function destroy(Request $request): JsonResponse
{
    return $this->defaultControllerHandler(function () use ($request) {
        $data = $this->artworkService->deleteArtwork($request);
        return response()->json(['success' => $data['success'], 'data' => $data['data'], 'message'
=> 'Картину успішно видалено.']);
    }, ModuleType::Artwork()->key);
}
```

Рисунок 2.8 – Обробка запиту на видалення картини

Сервіс `ArtworkService` обробляє бізнес-логіку для мистецьких робіт. Він включає методи для отримання, створення, оновлення та видалення картин.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 48   |

### Основні методи:

1. Метод `getAllArtworks` відповідає за отримання всіх картин з можливістю фільтрації. Він приймає параметри для фільтрації та пагінації, будує запит до бази даних з урахуванням цих параметрів і повертає результат. Якщо задано параметр `page`, дані повертаються з пагінацією, в іншому випадку - всі результати без пагінації (рис. 2.9).

```
public function getAllArtworks($params, $isFilter = false)
{
    $data = Artwork::query()
        ->with(['metaTag', 'images', 'type', 'artist', 'artist.image'])
        ->when(isset($params['status_list']), function ($query) use ($params) {
            $query->whereIn('status', $params['status_list']);
        })
        ->notDeleted()
        ->sortByPrice(!empty($params['sort']) ? $params['sort'] : null);
    if (isset($params['type_slug'])) {
        $data = $data->whereHas('type', function ($query) use ($params) {
            $query->where('slug', '=', $params['type_slug']);
        });
    }
    if (isset($params['only_main_img'])) {
        $data = $data->whereHas('images', function ($query) {
            $query->where('main_img', '=', Status::Active);
        });
    }
    if (isset($params['not_bought'])) {
        $data = $data->notBought();
    }
    if (isset($params['max_price']) && !$isFilter) {
        $data = $data->where('price', '<=', $params['max_price']);
    }
    if (isset($params['artist']) && !$isFilter) {
        $data = $data->whereHas('artist', function ($query) use ($params) {
            $query->where('slug', '=', $params['artist']);
        });
    }
}
```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 49   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

```

    });
}
if (isset($params['type']) && !$isFilter) {
    $data = $data->whereHas('type', function ($query) use ($params) {
        $query->where('slug', '=', $params['type_slug']);
    });
}
if (isset($params['page'])) {
    return $data->paginateScope();
} else {
    return $data->get();
}
}

```

Рисунок 2.9 – Отримання всіх картин з можливістю фільтрації

2. Метод `createArtwork` відповідає за створення нової картини. Він приймає масив параметрів, створює новий запис у базі даних і додає мета-теги для цієї картини. Після успішного створення нової картини, метод логгування додає запис про цю дію (рис. 2.10).

```

public function createArtwork($params): array
{
    $data = Artwork::create($params);
    $this->manageMetaTag($params, ModuleType::Artwork, $data->id);
    (new CreateGeneralLogService([
        'module_name' => ModuleType::Artwork()->key,
        'module_id' => $data->id,
        'user_id' => Auth::user()->id,
        'action' => Api::Create,
    ]))();
    return [
        'success' => true,
        'data' => $data->fresh(['logs', 'logs.changes', 'logs.user', 'logs.user.image', 'metaTag',
        'images', 'type', 'artist', 'artist.image']),
    ];
}

```

Рисунок 2.10 – Процес створення картини

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 50   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

3. Метод `updateArtwork` відповідає за оновлення інформації про існуючу картину. Він приймає масив параметрів, знаходить відповідну картину в базі даних, оновлює її дані, додає або оновлює мета-теги, а також записує зміни до журналу дій (рис. 2.11).

```
public function updateArtwork($params): array
{
    $data = Artwork::findOrFail($params['id']);
    $oldData = clone $data;
    $data->update($params);
    $this->manageMetaTag($params, ModuleType::Artwork, $data->id);
    $newData = $data->fresh(['logs', 'logs.changes', 'logs.user', 'logs.user.image', 'metaTag',
    'images', 'type', 'artist', 'artist.image']);
    (new CreateGeneralLogService([
        'data_before' => $oldData,
        'data_after' => $newData,
        'module_name' => ModuleType::Artwork()->key,
        'module_id' => $params['id'],
        'user_id' => Auth::user()->id,
        'action' => Api::Update,
    ]))();
    return [
        'success' => true,
        'data' => $newData,
    ];
}
```

Рисунок 2.11 – Процес оновлення картини

4. Метод `deleteArtwork` відповідає за видалення картини. Він знаходить відповідну картину в базі даних, виконує м'яке видалення (soft delete) картини та її мета-тегів, а також записує дію до журналу (рис. 2.12).

```
public function deleteArtwork($params): array
{
    $data = Artwork::with('metaTag')
        ->notDeleted()
        ->findOrFail($params['id']);
```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 51   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

```

$data->softDelete();
$data->metaTag()?->softDelete();
(new CreateGeneralLogService([
    'module_name' => ModuleType::Artwork()->key,
    'module_id' => $params['id'],
    'user_id' => Auth::user()->id,
    'action' => Api::Delete,
]))();
return [
    'data' => $data->fresh(['logs', 'logs.changes', 'logs.user', 'logs.user.image', 'metaTag',
'images', 'type', 'artist', 'artist.image']),
    'success' => true,
];

```

Рисунок 2.12 – Процес видалення картини

Цей детальний опис контролера ArtworksController та сервісу ArtworkService надає повне уявлення про реалізацію функціоналу системи управління контентом для художньої галереї, зображено в додатку Б. Використання стандартних методів CRUD дозволяє легко і ефективно управляти даними про мистецькі роботи, забезпечуючи зручність як для адміністратора, так і для кінцевих користувачів.

На діаграмі визначені такі елементи:

#### **Клієнтська частина (Frontend):**

- **web Application (Blade):** використовує шаблони Blade для відображення веб-інтерфейсу користувачів. Взаємодіє з бекендом через API для отримання та відправлення даних;
- **admin Application (Angular):** використовує Angular для створення динамічного інтерфейсу адміністрування. Взаємодіє з бекендом через API для виконання CRUD операцій та управління контентом.

#### **Серверна частина (Backend):**

- **laravel API:** обробляє всі запити від клієнтської частини та адміністративної панелі. Використовує контролери для обробки запитів,

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 52   |

моделі для взаємодії з базою даних, сервіси для бізнес-логіки та middleware для додаткової обробки запитів;

- **controllers:** приймають запити від клієнтів та адміністративної панелі, викликають відповідні сервіси та повертають відповіді;
- **models:** відповідають за взаємодію з базою даних;
- **services:** обробляють бізнес-логіку, виконують дії з моделями та іншими сервісами;
- **middleware:** виконує додаткову обробку запитів перед їх передачею в контролери;
- **routes:** визначають, який контролер буде обробляти конкретний запит.

#### База даних (MySQL):

- **users:** таблиця з інформацією про користувачів;
- **artists:** таблиця з інформацією про художників;
- **artworks:** таблиця з інформацією про мистецькі роботи;
- **artSales:** таблиця з інформацією про продажі мистецьких робіт;
- **artTypes:** таблиця з інформацією про типи мистецтв;
- **blogs:** таблиця з інформацією про блоги;
- **news:** таблиця з інформацією про новини;
- **images:** таблиця з інформацією про зображення;
- **generalLogs:** таблиця з логами;
- **userComments:** таблиця з коментарями користувачів.

#### Сервер автентифікації (Laravel Passport):

- **token Management:** управляє створенням та перевіркою токенів автентифікації;
- **authorization Middleware:** забезпечує доступ до захищених ресурсів тільки авторизованим користувачам.

#### Docker Containers:

- **laravel Backend Container:** контейнер, що містить серверну частину Laravel API;

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 53   |

- **angular Admin Container:** контейнер, що містить адміністративну панель на Angular;
- **mySQL Database Container:** контейнер, що містить базу даних MySQL.

### Висновки до другого розділу

В процесі проектування системи управління контентом для художньої галереї були детально проаналізовані всі варіанти використання програмного продукту відповідно до вимог користувачів та функціональних потреб. На основі отриманих результатів була спроектована загальна структура системи. Проектування включало моделювання структури проекту, визначення взаємодій між складовими частинами та їх функціональністю.

Було спроектовано базу даних, яка включає таблиці для зберігання інформації про користувачів, художників, мистецькі роботи, продажі, типи мистецтва, блоги, новини, зображення та коментарі користувачів. Це забезпечує цілісність та структурованість даних в системі.

Описано алгоритми роботи з додатком та послідовність дій користувачів та адміністраторів. Зокрема, було розглянуто процеси створення, редагування та видалення мистецьких робіт.

Основний функціонал системи включає можливість додавання та редагування мистецьких робіт, управління художниками та типами мистецтва, а також забезпечення безпеки та автентифікації користувачів за допомогою Laravel Passport. Також було розроблено адміністративну панель на Angular, яка забезпечує зручне управління контентом та взаємодію з API сервера.

Проектування системи також враховувало використання Docker для контейнеризації компонентів, що забезпечує ізоляцію, масштабованість та легкість розгортання системи.

Таким чином, другий розділ включав детальне проектування та моделювання системи управління контентом для художньої галереї, що

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 54   |

забезпечує ефективну та зручну взаємодію користувачів з системою, а також надійну та структуровану організацію даних.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 55   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ УПРАВЛІННЯ КОНТЕНТОМ ХУДОЖНЬОЇ ГАЛЕРЕЇ

### 3.1 Порядок встановлення та налаштування параметрів платформи

Встановлення та налаштування платформи системи управління контентом для художньої галереї вимагає розгортання всіх її компонентів та налаштування кожного елемента окремо, а також загальної системи в цілому.

Для опису архітектури платформи було створено діаграму розгортання (рис. 3.1).

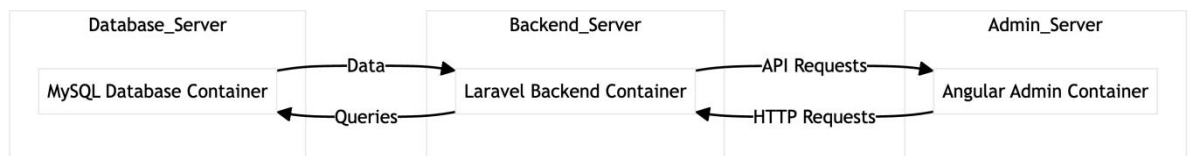


Рисунок 3.1 – Діаграма розгортання платформи системи управління контентом для художньої галереї

Опис процесу розгортання

#### API Server (Laravel Backend):

- **розгортання серверної частини додатку:** серверна частина додатку написана на Laravel, яка забезпечує основну логіку роботи додатку та взаємодію з базою даних MySQL. Для розгортання необхідно:

- **встановити всі необхідні залежності за допомогою Composer.** Composer є стандартним інструментом для управління залежностями в PHP [19];
- **налаштувати файл .env з параметрами підключення до бази даних та іншими конфігураціями.** Це включає налаштування бази даних, SMTP сервера для відправки email і інших параметрів конфігурації [20];
- **виконати міграції для створення таблиць у базі даних за допомогою команди php artisan migrate.** Міграції дозволяють створювати та змінювати структуру таблиць у базі даних програмним шляхом [21].

### **Admin Application (Angular):**

- **розгортання адміністративної панелі:** адмінка написана на Angular і знаходиться на окремому сервері. Для розгортання необхідно:

- **встановити всі необхідні залежності за допомогою команди** npm install. NPM є стандартним інструментом для управління залежностями у Node.js проектах [22];

- **налаштувати файл** environment.ts **з параметрами підключення до API.** Цей файл містить конфігурації для підключення до серверного API [23];

- **зібрати проект за допомогою команди** ng build. Ця команда компілює проект Angular в статичні файли для розгортання [24].

### **Database Server (MySQL):**

- **розгортання бази даних:** Для зберігання даних використовується база даних MySQL. Для розгортання необхідно:

- **встановити MySQL сервер.** MySQL є популярною системою керування реляційними базами даних [25];

- **створити базу даних та користувача з необхідними правами.** Це можна зробити за допомогою команд CREATE DATABASE і CREATE USER [26];

- **налаштувати підключення до бази даних у файлі** .env **серверної частини додатку.** Це забезпечує доступ додатку до бази даних [27];

- **виконати міграції для створення таблиць за допомогою команди** php artisan migrate. Це створює необхідні таблиці в базі даних [28].

### **Docker Containers:**

- **контейнеризація компонентів:** Використання Docker для контейнеризації забезпечує ізоляцію та легкість розгортання системи. Для цього необхідно:

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 57   |

- **створити Dockerfile для кожного компоненту (Laravel Backend, Angular Admin, MySQL).** Dockerfile містить інструкції для створення Docker образу [29];
- **створити docker-compose.yml для визначення сервісів та їх взаємодії.** Docker Compose дозволяє запускати багатоконтейнерні Docker додатки [30];
- **запустити команду docker-compose up -d для підняття всіх контейнерів.** Ця команда створює та запускає контейнери у фоновому режимі [31].

### 3.2 Структура інтерфейсу краудфандингової платформи

Система управління художньою галереєю складається з вебдодатку, який надає користувачам зручний доступ до перегляду контенту та замовлення картин. Інтерфейс вебдодатку містить кілька основних сторінок, кожна з яких виконує свої функції та має специфічний дизайн. Далі наводиться детальний опис кожної сторінки вебдодатку.

Головна сторінка є першою точкою взаємодії користувача з сайтом. Вона надає загальний огляд вмісту сайту та швидкий доступ до різних розділів. На головній сторінці розміщені блоки з посиланнями на останні блоги та новини, секція з найновішими відгуками користувачів про картини та художників, а також блок рекомендованих картин для покупки, який відображає картини на основі вподобань користувача або популярності (рис. 3.2) - (рис. 3.3).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 58   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

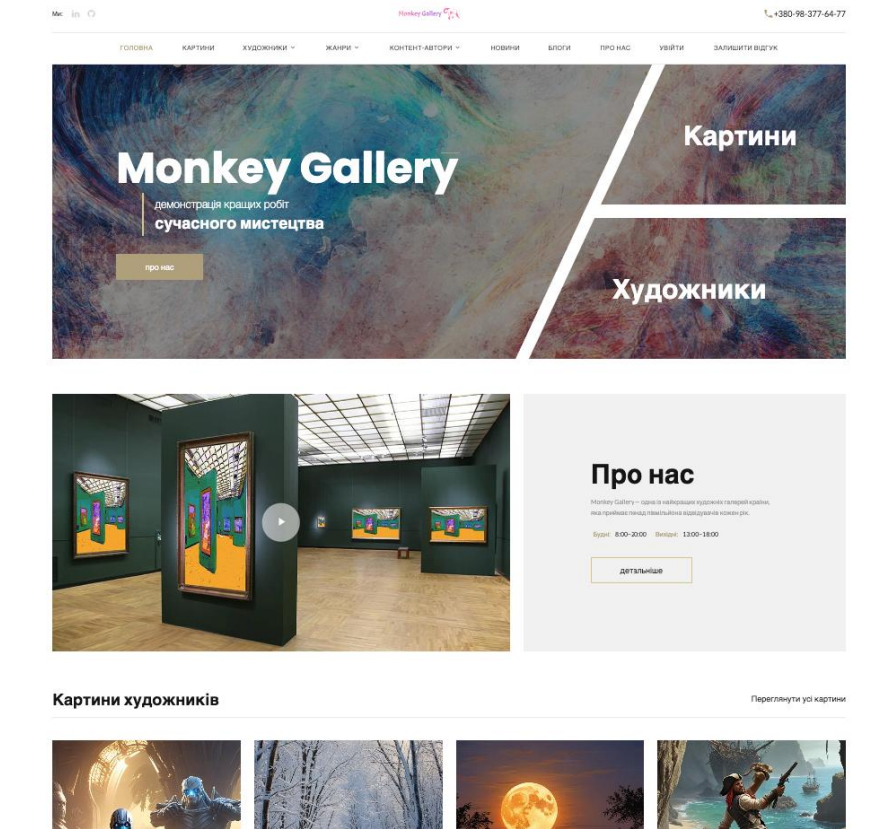


Рисунок 3.2 – Головна сторінка

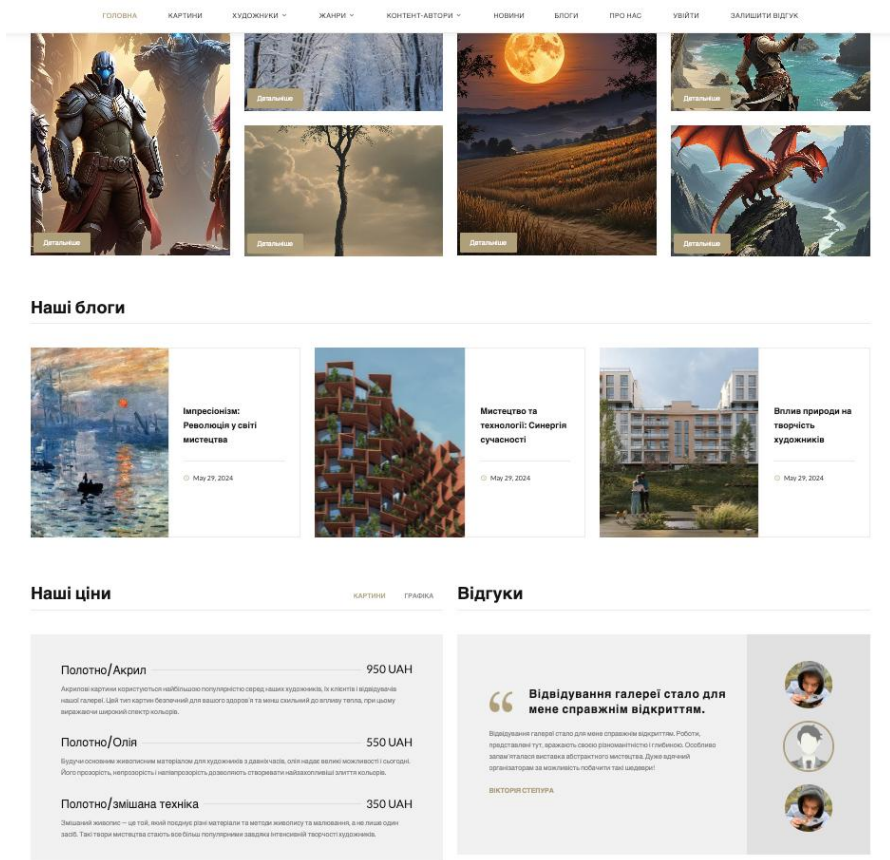


Рисунок 3.3 – Головна сторінка

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 59   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

Сторінка всіх художників надає користувачам можливість переглянути список усіх художників, представлених на сайті (рис. 3.4). Сторінка відображає імена, коротку інформацію та зображення художників. Користувач може знайти конкретного художника за допомогою кліку.

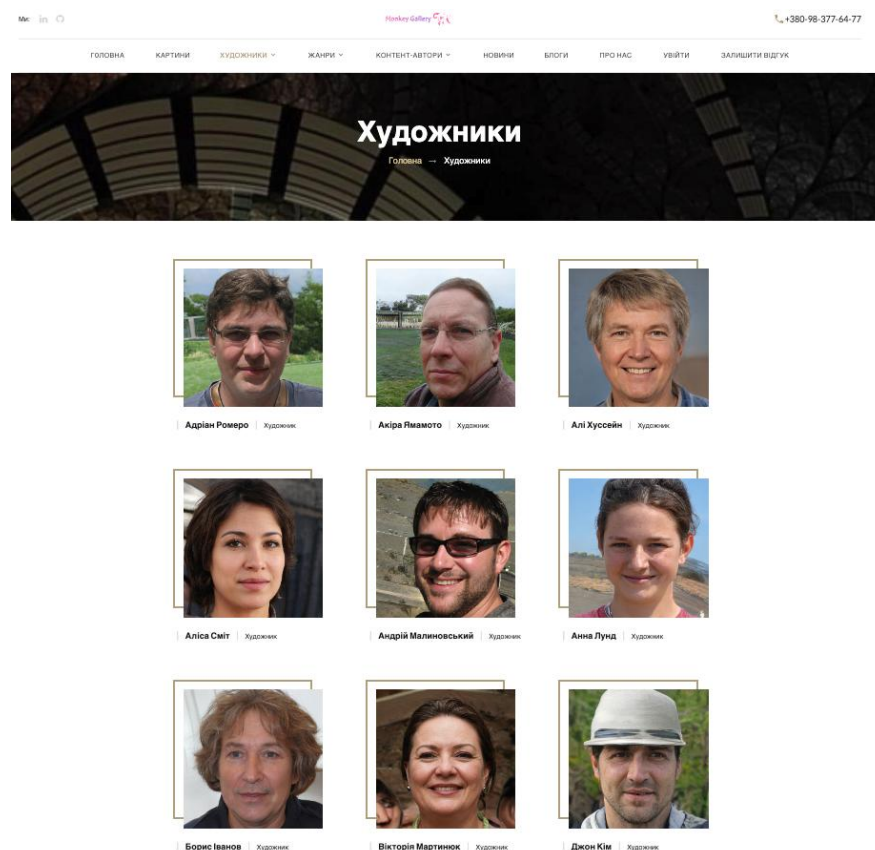


Рисунок 3.4 – Сторінка усіх художників

Детальна сторінка художника надає розширену інформацію про обраного художника. Вона містить біографію художника, включаючи дати народження та смерті, національність, коротку біографію та основні досягнення. Перелік картин, створених цим художником, з можливістю перегляду деталей кожної роботи. Це дозволяє користувачам краще зрозуміти стиль та творчість художника (рис. 3.5).

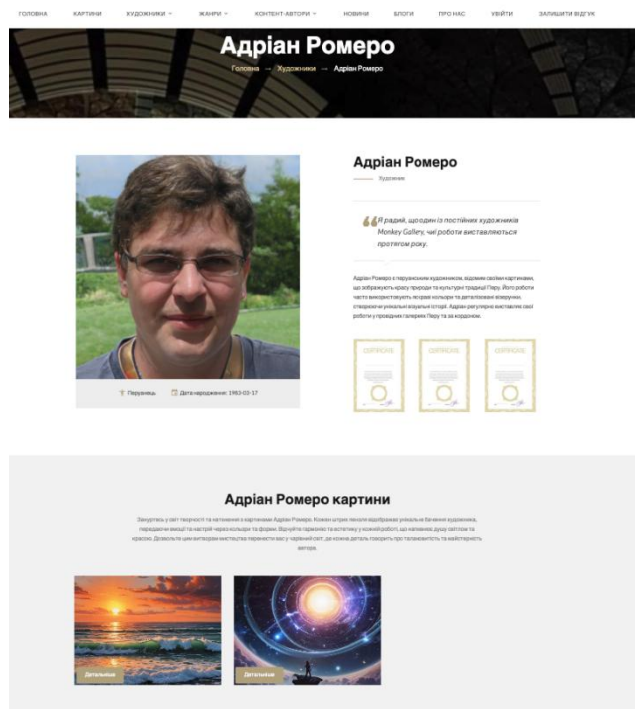


Рисунок 3.5 – Сторінка детальної інформації про художника

Сторінка всіх блогів відображає список усіх блогів, опублікованих на сайті. Сторінка містить заголовки, авторів та дати публікацій блогів (рис. 3.6).

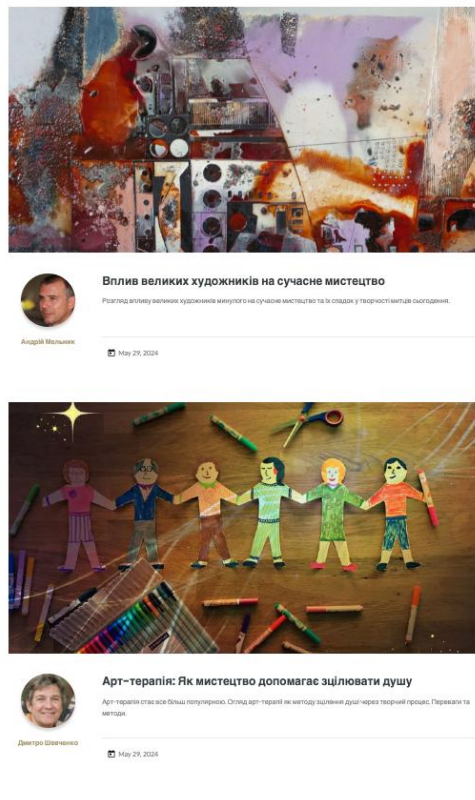


Рисунок 3.6 – Сторінка усіх блогів

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 61   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

Детальна сторінка блогу надає повний текст обраного блогу та додаткову інформацію (рис. 3.7). Вона містить основний контент блогу з зображеннями та вставками.

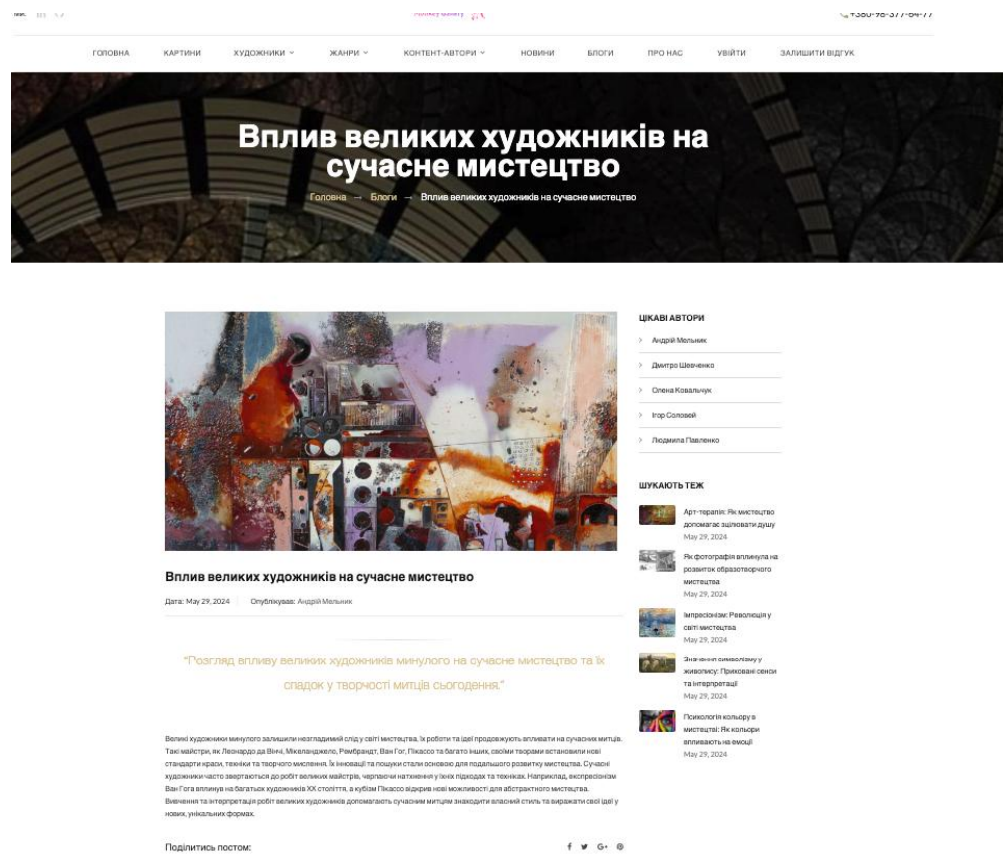


Рисунок 3.7 – Детальна сторінка блогу

Сторінка художніх жанрів відображає інформацію про доступні художні жанри та мінімальні ціни на картини в кожному жанрі (рис. 3.8). Сторінка містить перелік жанрів з інформацією про найнижчу ціну картин у кожному жанрі. Це дозволяє користувачам швидко оцінити ціновий діапазон картин у різних жанрах та вибрати відповідний жанр для подальшого перегляду.



Рисунок 3.8 – Сторінка художніх жанрів

Сторінка картин за жанрами відображає всі картини певного жанру (рис. 3.9). Користувач може фільтрувати та сортувати картини за різними критеріями, такими як ціна, популярність та дата додавання. На цій сторінці також можна переглянути деталі кожної картини, додати її до кошика. Функції фільтрації та сортування дозволяють користувачам швидко знайти картини, що відповідають їхнім вимогам та вподобанням.

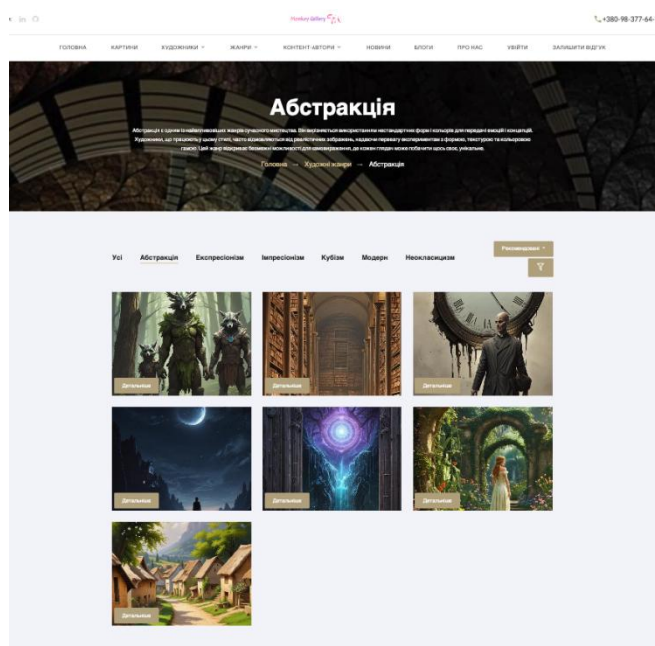


Рисунок 3.9 – Сторінка картин за жанрами

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 63   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

Для того, щоб користуватися сайтом з повним функціоналом, користувач повинен зареєструватися. Спочатку він побачить форму, за допомогою якої можна увійти на сайт або зареєструватися. Після реєстрації користувач отримує доступ до особистого кабінету, де може переглядати історію успішних покупок, корзину та змінювати особисті дані. Авторизовані користувачі також мають доступ до API документації (рис. 3.10). Це дозволяє користувачам максимально використовувати можливості платформи та отримувати персоналізований досвід.

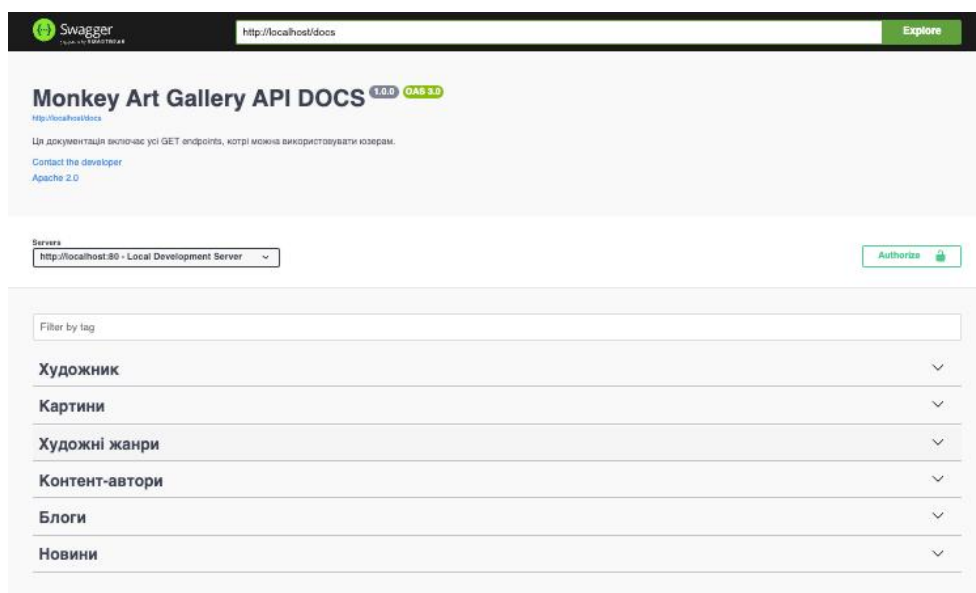


Рисунок 3.10 – API документація

Після реєстрації користувач може увійти на свою сторінку, де відображаються його фото профіля, інформація про нього та його відгуки. Користувач може переглядати інформацію про себе, натискаючи на стрілочки вправо або вліво. Після входу на панелі навігації з'являється блок, за допомогою якого можна перейти на свою сторінку, налаштування або вийти зі сторінки. Також на сторінці користувача є блок для фільтрації або сортування відгуків. Це дозволяє користувачам легко управляти своєю інформацією та відгуками, забезпечуючи зручний доступ до всіх функцій особистого кабінету (рис. 3.11) - (рис. 3.12).



ПОКУПКИ КОРЗИНА ОСОБИСТІ ДАНІ

| НАЗВА         | ЖАНР       | АВТОР           | ТИП ОПЛАТИ | ДАТА ПОКУПКИ | ЦІНА             | СТАТУС              |
|---------------|------------|-----------------|------------|--------------|------------------|---------------------|
| Тися Озеро    | Сюрреалізм | Стефанія Ковач  | card       | 2024-06-05   | 25.662.00<br>UAH | Успішно<br>оплачено |
| Мисли<br>Вони | Абстракція | Жан-Поль Дюваль | card       | 2024-06-05   | 35.024.00<br>UAH | Успішно<br>оплачено |
| Шлях<br>Война | Поп-арт    | Петро Ковальчук | card       | 2024-06-05   | 2.177.00<br>UAH  | Успішно<br>оплачено |

Рисунок 3.11 – Особистий кабінет

ПОКУПКИ КОРЗИНА **ОСОБИСТІ ДАНІ**

**BEARERAUTHN\_TOKEN**  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIzIiwianRpIjoZTUxOWWQ4MThjNDADNzUwMjhhNmIwNjI4YzZkYzZjQ0DFJMWMOZDVmMzQxh  
XP7FtJdAc5fgHQ16xjG6OHw7v2Upteoqlr9w8mhMKnpHZZyEOAKNzoLRULOTNRLashTvd41Ygnu3HIZWgzOP9OIxizeuLQmVltk4yN9mkwVJ2AGo  
t2CMACkPwOI4dq6AqGI\_Dx89KSeRMh0A5q0mDumvAcvIkxOgus1F5ktlDlaxBvesZYXs0jM6PMIIBC5CLkOEGn8w42RkDm5XzckuHxWgZ67ve2JqbnY  
P9KpJCyKdBljlcU6Vr2h\_PmvEAtt3cUDh4GiQJK9\_RDNqgOdKAK3Zuwt1\_niqkvAQkaDABlJGw6Wd-VGHRc4mpsYtpo9YsFuAagzVopMg53dbvg2EUuxIzB

**IMPR**  
Danylo Savhenko

**ПОШТА**  
danylo.savchenko96@gmail.com

**НОВИЙ ПАРОЛЬ**

**Оновити**

Рисунок 3.12 – Особистий кабінет

Ця структура інтерфейсу забезпечує зручність використання та навігації по сайту, дозволяючи користувачам легко знаходити та взаємодіяти з потрібним контентом.

Адміністративна панель системи управління контентом для художньої галереї надає адміністраторам і сейлз-менеджерам можливість керувати контентом, переглядати та редагувати записи, а також відстежувати зміни. Опис нижче стосується модуля управління картинами, який є типовим для всіх модулів у системі.

На головній сторінці модуля картин адміністраторам і сейлз-менеджерам відображається таблиця з усіма записами про картини. У таблиці містяться колонки, що включають статус картини, назву, жанр, художника, ціну, дату продажу, дату додавання та доступні дії (рис. 3.13). Адміністратори можуть використовувати фільтри для сортування записів або пошуку за ключовими словами.

Місцева • CEO • Користувачі • Налаштування

Картини [+ ДОДАТИ КАРТИНУ](#)

| Статус   | Назва               | Жанр         | Художник        | Ціна   | Дата продажу     | Дата додавання    | Дії                                 |
|----------|---------------------|--------------|-----------------|--------|------------------|-------------------|-------------------------------------|
| Активний | Повстання Фенікса   | Футуризм     | Петро Ковальчук | 19,151 | В наявності      | 6/24/00, 2:02 PM  | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Вежа Дракона        | Абстракція   | Адріан Ромеро   | 30,313 | 6/6/24, 12:00 AM | 6/23/00, 7:56 PM  | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Мисьі Волі          | Абстракція   | Жан-Поль Доваль | 35,024 | 6/5/24, 12:00 AM | 6/23/00, 5:28 PM  | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Містичні Острови    | Імпресіонізм | Борис Іванов    | 3,532  | В наявності      | 6/17/00, 11:00 AM | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Опівінене Положення | Поп-арт      | Маріє Гонзалес  | 41,655 | В наявності      | 6/16/00, 2:42 PM  | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Подорож Починається | Неокласицизм | Мануель Гарсія  | 26,115 | В наявності      | 6/16/00, 2:03 PM  | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Привований Сазис    | Романтизм    | Лаз Маршал      | 15,713 | В наявності      | 6/14/00, 7:02 PM  | <a href="#">✎</a> <a href="#">✖</a> |
| Активний | Астральний План     | Романтизм    | Адріан Ромеро   | 9,046  | В наявності      | 6/12/00, 12:57 PM | <a href="#">✎</a> <a href="#">✖</a> |

Рисунок 3.13 – Список картин

Для додавання нової картини адміністратор натискає кнопку "Додати картину". Відкривається форма, в якій потрібно заповнити поля: статус, назва, художній жанр, художник, дата створення, ціна, опис, зображення (рис. 3.14). Після заповнення форми новий запис додається до бази даних.

Базова інформація

Статус: **Активний** НЕАКТИВНИЙ

Назва:  Слоган:

Художній жанр:  Художник:

Дата створення:  Дата продажу:

Ціна (грн):

Опис:

Фото:

SEO:

[Оновити](#) [Видалити](#) [Відмінити](#)

Історія змін

- Danylo Savhenko Дія: завантажив файл 6/4/24, 8:48 PM
- Danylo Savhenko Дія: оновив файл 5/30/24, 8:22 PM
- Danylo Savhenko Дія: завантажив файл 5/30/24, 8:22 PM
- Danylo Savhenko Дія: створив 5/30/24, 7:31 PM

Рисунок 3.14 – Форма менеджменту картини

Для редагування існуючої картини адміністратор натискає кнопку редагування поруч із відповідним записом у таблиці. Для видалення картини адміністратор натискає кнопку видалення поруч із відповідним записом у таблиці. Після підтвердження видалення запис про картину видаляється з бази даних (рис. 3.15).

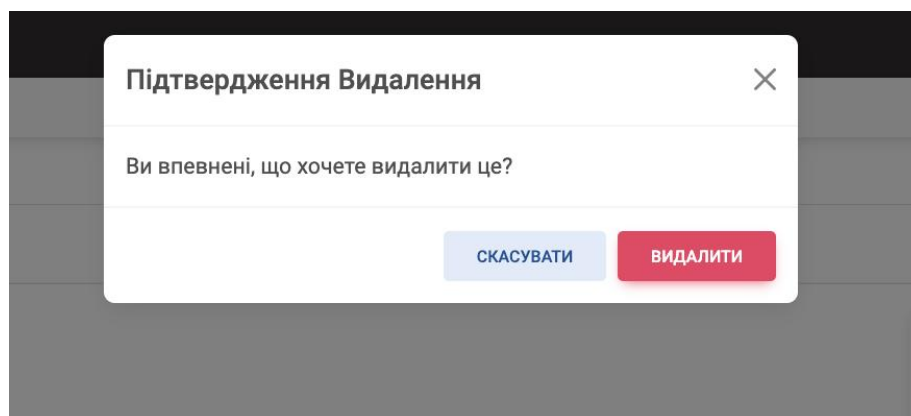


Рисунок 3.15 – Форма підтвердження видалення

Адміністратори також можуть завантажувати зображення для картин, встановлювати головне зображення та видаляти зайві зображення у розділі "Фото" (рис. 3.16). Це забезпечує якісне візуальне представлення картин на сайті.

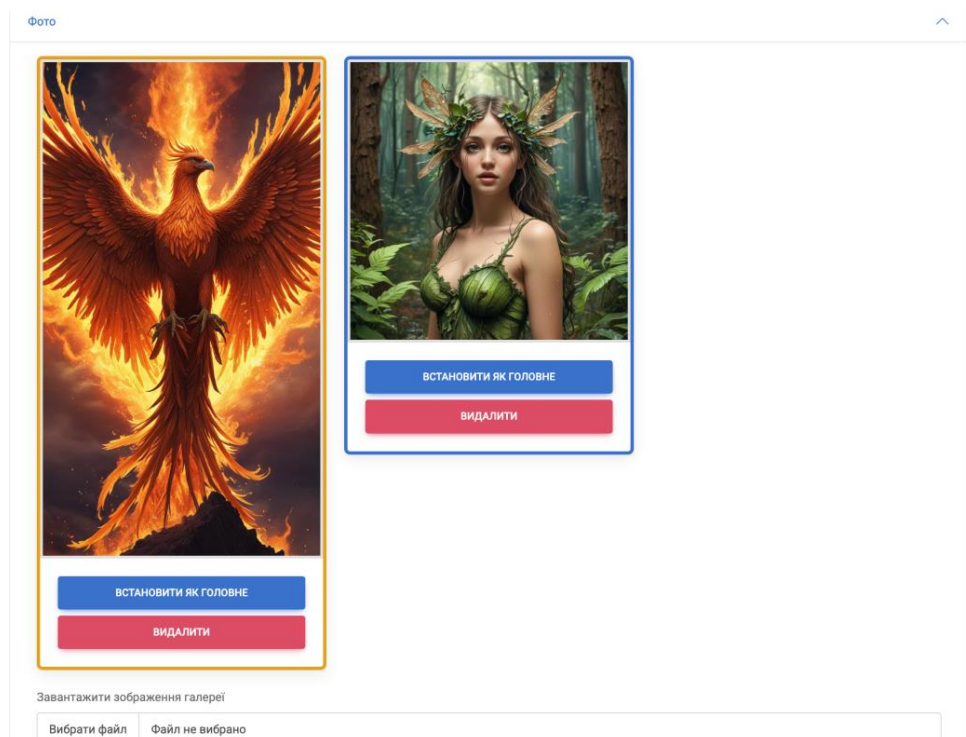


Рисунок 3.16 – Розділ зображень

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 67   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

У розділі "SEO" адміністратор може додавати мета-дані для покращення пошукової оптимізації сторінок картин. Поля для заповнення включають назву сторінки (META), опис сторінки під H1, опис сторінки (META) (рис. 3.17).

SEO

Рисунок 3.17 – Розділ SEO

### 3.3 Загальна статистика сайту

Сторінка аналітики надає адміністраторам можливість переглядати статистику щодо картин, художників та жанрів. На цій сторінці відображаються графіки, які показують кількість картин за художніми жанрами та художниками. Це дозволяє адміністраторам відстежувати популярність певних жанрів і художників, а також приймати обґрунтовані рішення щодо управління контентом (рис. 3.18) - (рис. 3.19).

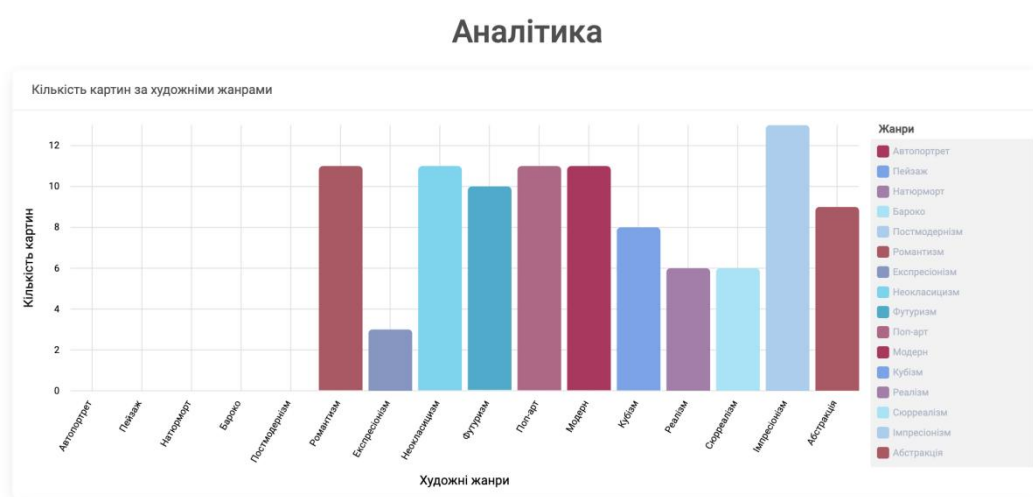


Рисунок 3.18 – Аналітика сайту

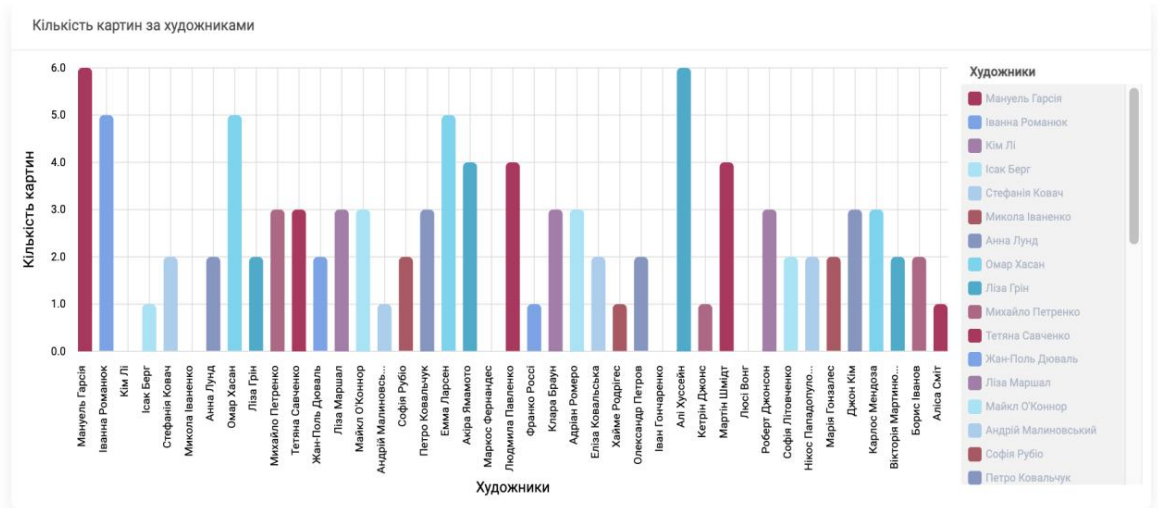


Рисунок 3.19 – Аналітика сайту

Сторінка продажів надає адміністраторам інформацію про всі здійснені продажі картин. У таблиці відображаються такі колонки: статус, картина, жанр, покупець, телефон, адреса, ціна та дата продажу. Адміністратори можуть переглядати деталі кожного продажу та аналізувати дані для покращення процесів продажу (рис. 3.20).

#### Продажі

| Статус | Картина      | Жанр       | Покупець        | Телефон         | Адреса                     | Ціна       | Дата продажу     |
|--------|--------------|------------|-----------------|-----------------|----------------------------|------------|------------------|
| Успіх  | Вежа Дракона | Абстракція | Тестік          | 321312321321321 | Вільський шлях 14, Житомир | 30,313 UAH | 6/6/24, 12:00 AM |
| Успіх  | Тихе Озеро   | Сюрреалізм | Danylo Savhenko | 0983776477      | Вільський шлях 14, Житомир | 25,662 UAH | 6/5/24, 12:00 AM |
| Успіх  | Міські Вогні | Абстракція | Danylo Savhenko | 0983776477      | Вільський шлях 14, Житомир | 35,024 UAH | 6/5/24, 12:00 AM |
| Успіх  | Шлях Воїна   | Поп-арт    | Danylo Savhenko | 0983776477      | Вільський шлях 14, Житомир | 2,177 UAH  | 6/5/24, 12:00 AM |

Попередня 1 Наступна

Рисунок 3.20 – Сторінка продажів

Сторінка логів відображає історію змін для кожного модуля системи. На цій сторінці адміністратор може переглянути, хто і коли вніс зміни, які поля були змінені, та які значення мали до і після змін. Це дозволяє відслідковувати всі дії, що були здійснені в системі, забезпечуючи прозорість та контроль над контентом (рис. 3.21).

Загальні логи

| Користувач                                                                                        | Назва модулю | ID об'єкта | Дія    | Старе значення   | Нове значення    | Дата створення   |
|---------------------------------------------------------------------------------------------------|--------------|------------|--------|------------------|------------------|------------------|
|  Danylo Savhenko | User         | 2          | оновив | Danylo Savhenko2 | Danylo Savhenko  | 6/5/24, 5:40 PM  |
|  Danylo Savhenko | User         | 2          | оновив | Danylo Savhenko  | Danylo Savhenko2 | 6/5/24, 5:40 PM  |
|  Danylo Savhenko | User         | 2          | оновив | Danylo Savhenko2 | Danylo Savhenko  | 6/4/24, 11:02 PM |
|  Danylo Savhenko | User         | 2          | оновив | Danylo Savhenko  | Danylo Savhenko2 | 6/4/24, 11:02 PM |
|  Богдан Карпенко | User         | 16         | оновив | Богдан Карпенко2 | Богдан Карпенко  | 6/4/24, 10:54 PM |
|  Богдан Карпенко | User         | 16         | оновив | Богдан Карпенко  | Богдан Карпенко2 | 6/4/24, 10:54 PM |
|  Богдан Карпенко | User         | 16         | оновив | Богдан Карпенко1 | Богдан Карпенко  | 6/4/24, 10:54 PM |
|  Богдан Карпенко | User         | 16         | оновив | Богдан Карпенко  | Богдан Карпенко1 | 6/4/24, 10:53 PM |

Рисунок 3.21 – Сторінка загальних логів

### 3.4 Тестування платформи

Проведемо тестування додатку. Спочатку перевіримо форму реєстрації (табл. 3.1).

Таблиця 3.1

Тестування форми реєстрації

| Номер тесту | Очікуваний результат                                            | Фактичний результат                                             | Пройдено/не пройдено (P/F) |
|-------------|-----------------------------------------------------------------|-----------------------------------------------------------------|----------------------------|
| 1           | Всі поля не зчитуються пустими                                  | Всі поля, не зчитуються пустими                                 | <b>P</b>                   |
| 2           | Поле логіну не зчитується, якщо такий логін вже існує           | Поле логіну не зчитується, якщо такий логін вже існує           | <b>P</b>                   |
| 3           | Поле для введення паролів не зчитуються, якщо вони не є рівними | Поле для введення паролів не зчитуються, якщо вони не є рівними | <b>P</b>                   |

Далі протестуємо функціонал додавання, редагування, видалення картини (табл. 3.2).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 70   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

Таблиця 3.2

## Тестування форми додавання картини

| <i>Номер тесту</i> | <i>Очікуваний результат</i>                                                  | <i>Фактичний результат</i>                                                   | <i>Пройдено/не пройдено (P/F)</i> |
|--------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------------|
| 1                  | Всі обов'язкові поля заповнені                                               | Всі обов'язкові поля заповнені                                               | <b>P</b>                          |
| 2                  | Поле назви не зчитується, якщо воно пусте                                    | Поле назви не зчитується, якщо воно пусте                                    | <b>P</b>                          |
| 3                  | Поле ціни не зчитується, якщо введено нечислове значення                     | Поле ціни не зчитується, якщо введено нечислове значення                     | <b>P</b>                          |
| 4                  | Поле дати створення не зчитується, якщо введено дату, що перевищує поточну   | Поле дати створення не зчитується, якщо введено дату, що перевищує поточну   | <b>P</b>                          |
| 5                  | Поле художника не зчитується, якщо вибрано художника, який не існує          | Поле художника не зчитується, якщо вибрано художника, який не існує          | <b>P</b>                          |
| 6                  | Після успішного заповнення та збереження картина з'являється у списку картин | Після успішного заповнення та збереження картина з'являється у списку картин | <b>P</b>                          |

Далі протестуємо форму редагування картини (табл. 3.3).

Таблиця 3.3

## Тестування форми редагування картини

| <i>Номер тесту</i> | <i>Очікуваний результат</i>                           | <i>Фактичний результат</i>                            | <i>Пройдено/не пройдено (P/F)</i> |
|--------------------|-------------------------------------------------------|-------------------------------------------------------|-----------------------------------|
| 1                  | Всі обов'язкові поля заповнені                        | Всі обов'язкові поля заповнені                        | <b>P</b>                          |
| 2                  | Зміни зберігаються після натискання кнопки "Зберегти" | Зміни зберігаються після натискання кнопки "Зберегти" | <b>P</b>                          |
| 3                  | Поле назви не зчитується, якщо воно пусте             | Поле назви не зчитується, якщо воно пусте             | <b>P</b>                          |

## Продовження таблиці 3.3

|   |                                                                            |                                                                            |          |
|---|----------------------------------------------------------------------------|----------------------------------------------------------------------------|----------|
| 4 | Поле ціни не зчитується, якщо введено нечислове значення                   | Поле ціни не зчитується, якщо введено нечислове значення                   | <b>P</b> |
| 5 | Поле дати створення не зчитується, якщо введено дату, що перевищує поточну | Поле дати створення не зчитується, якщо введено дату, що перевищує поточну | <b>P</b> |
| 6 | Поле художника не зчитується, якщо вибрано художника, який не існує        | Поле художника не зчитується, якщо вибрано художника, який не існує        | <b>P</b> |

Далі протестуємо форму видалення картини (табл. 3.4).

Таблиця 3.4

## Тестування форми видалення картини

| <i>Номер тесту</i> | <i>Очікуваний результат</i>                               | <i>Фактичний результат</i>                                | <i>Пройдено/не пройдено (P/F)</i> |
|--------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------|
| 1                  | Картина видаляється після підтвердження                   | Картина видаляється після підтвердження                   | <b>P</b>                          |
| 2                  | Картина не видаляється без підтвердження                  | Картина не видаляється без підтвердження                  | <b>P</b>                          |
| 3                  | Видалена картина більше не відображається у списку картин | Видалена картина більше не відображається у списку картин | <b>P</b>                          |

Тестуємо сторінку аналітики (табл. 3.5).

Таблиця 3.5

## Тестування сторінки аналітики

| <i>Номер тесту</i> | <i>Очікуваний результат</i>                        | <i>Фактичний результат</i>                         | <i>Пройдено/не пройдено (P/F)</i> |
|--------------------|----------------------------------------------------|----------------------------------------------------|-----------------------------------|
| 1                  | Відображаються графіки кількості картин за жанрами | Відображаються графіки кількості картин за жанрами | <b>P</b>                          |

## Продовження таблиці 3.5

|   |                                                        |                                                        |          |
|---|--------------------------------------------------------|--------------------------------------------------------|----------|
| 2 | Відображаються графіки кількості картин за художниками | Відображаються графіки кількості картин за художниками | <b>Р</b> |
| 3 | Можливість фільтрувати дані по жанрах та художниках    | Можливість фільтрувати дані по жанрах та художника     | <b>Р</b> |

Далі тестуємо сторінку додавання відгуку (табл. 3.6).

Таблиця 3.6

## Тестування сторінки відгуку

| <i>Номер тесту</i> | <i>Очікуваний результат</i>                                | <i>Фактичний результат</i>                                 | <i>Пройдено/не пройдено (P/F)</i> |
|--------------------|------------------------------------------------------------|------------------------------------------------------------|-----------------------------------|
| 1                  | Поле відгуку не зчитується, якщо воно порожнє              | Поле відгуку не зчитується, якщо воно порожнє              | <b>Р</b>                          |
| 2                  | Відгук зберігається після заповнення поля                  | Відгук зберігається після заповнення поля                  | <b>Р</b>                          |
| 3                  | Користувач бачить свій відгук на сторінці перегляду картин | Користувач бачить свій відгук на сторінці перегляду картин | <b>Р</b>                          |

Далі тестуємо процес оплати (табл. 3.7).

Таблиця 3.7

## Тестування процесу оплати

| <i>Номер тесту</i> | <i>Очікуваний результат</i>                                   | <i>Фактичний результат</i>                                    | <i>Пройдено/не пройдено (P/F)</i> |
|--------------------|---------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------|
| 1                  | Кнопка "Купити" додає картину до кошика                       | Кнопка "Купити" додає картину до кошика                       | <b>Р</b>                          |
| 2                  | Перехід до кошика після натискання кнопки "Переглянути кошик" | Перехід до кошика після натискання кнопки "Переглянути кошик" | <b>Р</b>                          |

|   |                                                                                  |                                                                                  |          |
|---|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------|----------|
| 3 | Натискання кнопки "Оплатити" ініціює процес оплати                               | Натискання кнопки "Оплатити" ініціює процес оплати                               | <b>Р</b> |
| 4 | Після успішної транзакції користувач перенаправляється на сторінку підтвердження | Після успішної транзакції користувач перенаправляється на сторінку підтвердження | <b>Р</b> |

Ці тестування охоплюють основні функціональні можливості платформи системи управління контентом для художньої галереї, забезпечуючи коректність та зручність використання системи як для користувачів, так і для адміністраторів.

### Висновки до третього розділу

У третьому розділі було детально розглянуто порядок встановлення та налаштування параметрів платформи системи управління контентом для художньої галереї, структуру інтерфейсу, загальну статистику сайту та проведено тестування платформи.

Були описані кроки, необхідні для розгортання платформи, включаючи встановлення та налаштування серверної частини на Laravel, адміністративної панелі на Angular та бази даних MySQL. Використання Docker для контейнеризації забезпечує ізоляцію та легкість розгортання системи, що спрощує управління різними компонентами платформи.

Був детально описаний інтерфейс користувача та адміністратора платформи. Було показано, як користувачі можуть переглядати контент, замовляти картини, залишати відгуки та управляти своїм обліковим записом. Адміністративний інтерфейс дозволяє адміністраторам і сейлз-менеджерам керувати картинами, переглядати продажі, аналізувати статистику та працювати з логами змін.

Були проаналізовані статистичні дані сайту, що включають кількість картин за жанрами та художниками, а також загальні показники продажів. Це дозволяє оцінити популярність різних жанрів і художників, а також ефективність роботи платформи.

Було проведено ретельне тестування різних форм та модулів платформи, включаючи форму реєстрації, логіну, додавання та редагування картин, форму відгуків та процес оплати. Кожен тест підтвердив, що система відповідає очікуванням користувачів, забезпечує зручність використання та надійність у роботі. Важливим аспектом було тестування процесу оплати, яке показало, що користувачі можуть легко здійснювати покупки та отримувати підтвердження успішних транзакцій.

Таким чином, розділ показав, що платформа є надійною, функціональною та готовою до використання. Здобуті результати свідчать про високу якість реалізації та готовність платформи до експлуатації у реальних умовах.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 75   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## ВИСНОВКИ

У процесі роботи над кваліфікаційною роботою було розроблено систему управління контентом для художньої галереї, яка відповідає сучасним вимогам та стандартам вебтехнологій. Проект включає в себе як теоретичні, так і практичні аспекти створення вебдодатків, забезпечуючи комплексне вирішення поставлених завдань.

У першому розділі було здійснено детальний аналіз напрямків використання вебтехнологій для реалізації системи управління контентом. Основною метою було визначення завдань проекту та аналіз існуючих аналогів. Було розглянуто декілька аналогічних систем, що дозволило виділити їх сильні та слабкі сторони. На основі проведеного аналізу було обрано найоптимальніші інструментальні засоби для реалізації проекту: PHP 8.1, Laravel 10, MySQL, Angular, TypeScript, JavaScript, SCSS, CSS, та Bootstrap. Цей вибір забезпечив високу продуктивність, масштабованість та зручність розробки.

Другий розділ був присвячений проектуванню та розробці системи управління контентом. На цьому етапі було визначено варіанти використання системи, створено об'єктно-орієнтовану структуру та розроблено базу даних. Було розроблено алгоритми роботи системи, що включають основні функції: управління художніми творами, користувачами, зберігання даних та забезпечення безпеки. Було реалізовано такі функціональні модулі, як додавання, редагування, видалення картин, управління категоріями, робота з користувачами та логування дій. Це забезпечило комплексний підхід до управління контентом та дозволило створити ефективну та надійну систему.

У третьому розділі було детально описано порядок встановлення та налаштування платформи, структуру інтерфейсу, загальну статистику сайту та проведено тестування системи. Розгортання компонентів системи здійснювалося за допомогою Docker, що забезпечило ізоляцію та легкість управління сервісами. Було описано процес розгортання серверної частини на Laravel, адміністративної панелі на Angular та бази даних на MySQL. Створена інтерфейсна частина забезпечує зручність та інтуїтивність використання для

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 76   |

кінцевих користувачів та адміністраторів. Було проведено тестування форм реєстрації, логіну, оплати, а також основних функцій адміністративної панелі, що підтвердило надійність та зручність використання системи.

Система управління контентом для художньої галереї досягнула своєї основної мети – забезпечення зручності та ефективності роботи з художніми творами для користувачів, адміністраторів та сейлз-менеджерів. Розроблений проект готовий до впровадження та експлуатації у реальних умовах, забезпечуючи високий рівень задоволення потреб користувачів та ефективність управління контентом. Результати роботи підтверджують, що система є надійною, масштабованою та готовою до подальшого розвитку та вдосконалення.

Цей проект є важливим внеском у галузь управління художніми галереями та може бути використаний як основа для подальших досліджень та розробок у цій сфері.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 77   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Statista. Number of internet users worldwide from 2013 to 2023.  
[Електронний ресурс] - Режим доступу до ресурсу:  
<https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>.
2. Boyd, D. M., & Ellison, N. B. (2007). Social Network Sites: Definition, History, and Scholarship. Journal of Computer-Mediated Communication, 13(1), 210-230 с.
3. Pew Research Center. Social Media Use by Government: Agencies and Elected Officials Embrace New Communication Tools. [Електронний ресурс] - Режим доступу до ресурсу:  
<https://www.pewresearch.org/internet/2011/05/04/social-media-use-by-government-agencies-and-elected-officials-embrace-new-communication-tools/>.
4. Jose Art Gallery. [Електронний ресурс] - Режим доступу до ресурсу:  
<https://joseartgallery.com/>.
5. Naturalist Gallery. [Електронний ресурс] - Режим доступу до ресурсу:  
<https://naturalist.gallery/>.
6. PHP 8.1 Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.php.net/releases/8.1/en.php>.
7. Laravel 10 Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://laravel.com/docs/10.x>.
8. MySQL Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://dev.mysql.com/doc/>.
9. Angular Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://angular.io/docs>.
10. TypeScript Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/>.
11. JavaScript Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
12. Sass Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://sass-lang.com/documentation>.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 78   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

13. CSS Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
14. Bootstrap Documentation. [Електронний ресурс] - Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>.
15. Martin, R. C. Agile Software Development: Principles, Patterns, and Practices. Prentice Hall, 2003. - 531 с.
16. Hunt, A., & Thomas, D. The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley, 1999. - 361 с.
17. Brooks, F. P. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, 1995. - 212 с.
18. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994. - 417 с.
19. Composer: Dependency Manager for PHP. [Електронний ресурс] - Режим доступу до ресурсу: <https://getcomposer.org/>.
20. Laravel Configuration. [Електронний ресурс] - Режим доступу до ресурсу: <https://laravel.com/docs/10.x/configuration>.
21. Laravel Migrations. [Електронний ресурс] - Режим доступу до ресурсу: <https://laravel.com/docs/10.x/migrations>.
22. Node Package Manager (NPM). [Електронний ресурс] - Режим доступу до ресурсу: <https://www.npmjs.com/>.
23. Angular Environment Configuration. [Електронний ресурс] - Режим доступу до ресурсу: <https://angular.io/guide/build>.
24. Angular CLI: ng build. [Електронний ресурс] - Режим доступу до ресурсу: <https://angular.io/cli/build>.
25. MySQL: The world's most popular open source database. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.mysql.com/>.
26. MySQL CREATE DATABASE Statement. [Електронний ресурс] - Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/create-database.html>.

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 79   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

27. Configuring Laravel Database Connections. [Електронний ресурс] - Режим доступу до ресурсу: <https://laravel.com/docs/10.x/database>.

28. Laravel Database Migrations. [Електронний ресурс] - Режим доступу до ресурсу: <https://laravel.com/docs/8.x/migrations>.

29. Dockerfile Reference. [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.docker.com/engine/reference/builder/>.

30. Docker Compose Overview. [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.docker.com/compose/>.

31. Docker Compose Command. [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.docker.com/compose/reference/up/>.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 80   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

# ДОДАТКИ

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ПЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 81   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

### Діаграма класів

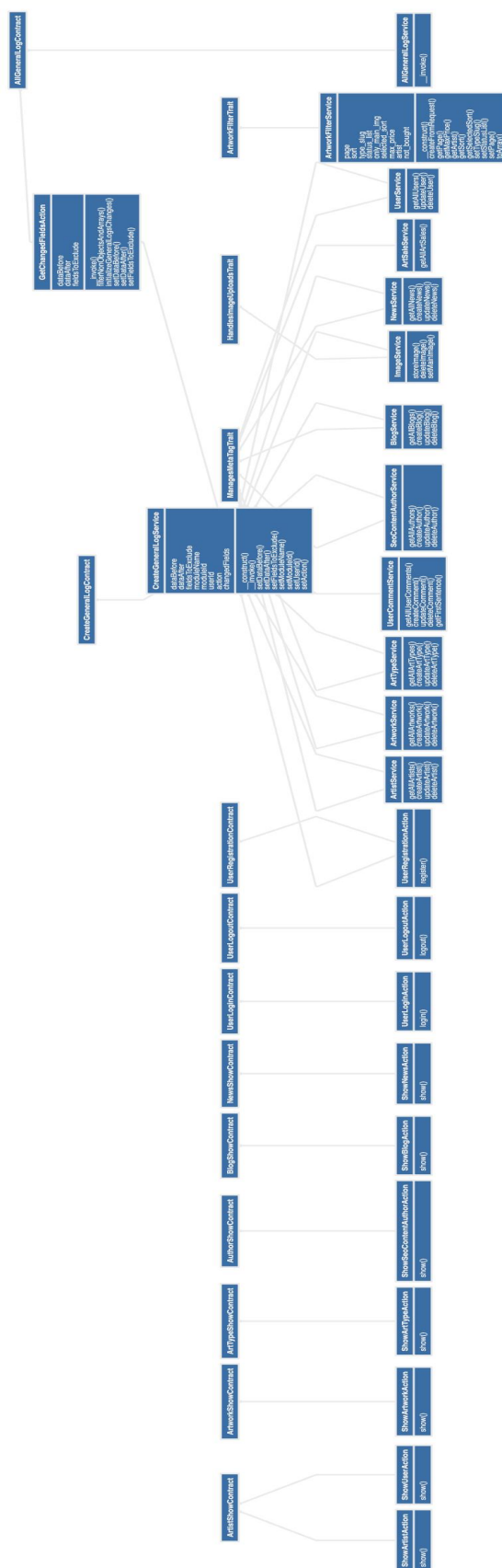


Рисунок А.1 – Діаграма класів

Діаграма компонентів

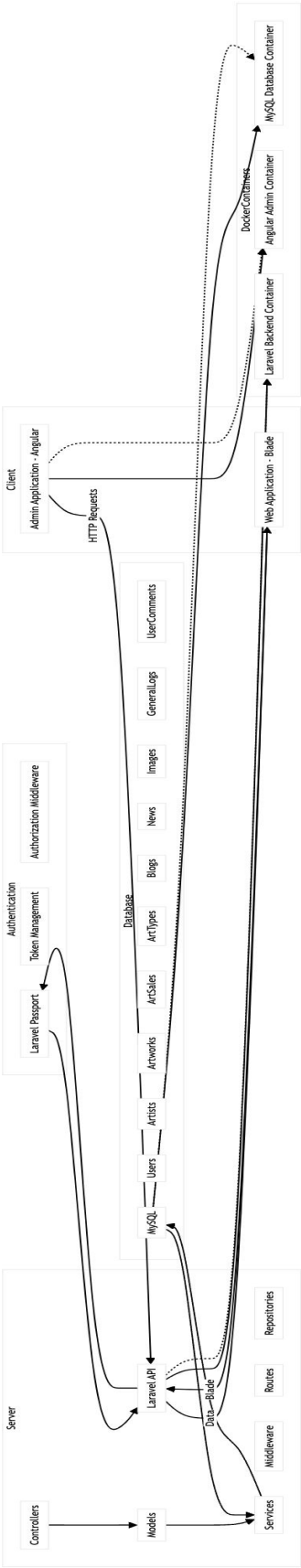


Рисунок Б.1 – Діаграма компонентів

## Скрипт сторінки редагування/додавання/видалення картини

artwork-detail.component.ts

```
import {Component, OnInit} from '@angular/core';
import {ActivatedRoute, Router} from '@angular/router';
import {ToastrService} from 'ngx-toastr';
import {ErrorHandlerService} from "../../services/error-handler.service";
import {DEFAULT_META_TAG, FILE_TYPE, NATIONALITIES_UA} from
'../../shared/constants';
import {ConfirmDeleteModalComponent} from "../../confirm-delete-modal/confirm-delete-
modal.component";
import {MdbModalRef, MdbModalService} from "mdb-angular-ui-kit/modal";
import {FormGroup, Validators, AbstractControl, FormBuilder} from '@angular/forms';
import {SlugUtilService} from "../../services/slug-util.service";
import {ArtTypeService} from "../../services/art-type.service";
import {ArtworkService} from "../../services/artwork.service";
import {ArtistService} from "../../services/artist.service";
```

```
@Component({
  selector: 'app-artwork-detail',
  templateUrl: './artwork-detail.component.html',
})
export class ArtworkDetailComponent implements OnInit {
  artworkForm: FormGroup;
  artwork: any = {
    title: "",
    slug: "",
    artist_id: "",
    type_id: "",
    description: "",
    creation_date: "",
    acquisition_date: "",
    price: "",
    status: true,
    ...DEFAULT_META_TAG
```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 84   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

```

};
isNewArtwork: boolean = true;
modalRef: MdbModalRef<ConfirmDeleteModalComponent> | null = null;
isLoading: boolean = true;
additionalData: any;
fileType: any = FILE_TYPE;
artistList: any;
artTypeList: any;

constructor(
  private route: ActivatedRoute,
  private router: Router,
  private artworkService: ArtworkService,
  private artistService: ArtistService,
  private artTypeService: ArtTypeService,
  private toastr: ToastrService,
  private errorHandler: ErrorHandlerService,
  private modalService: MdbModalService,
  private fb: FormBuilder,
  private slugUtilService: SlugUtilService
) {
  this.artworkForm = this.fb.group( {
    title: [this.artwork.title, [Validators.required, Validators.maxLength(255)]],
    slug: [this.artwork.slug, [Validators.required, Validators.maxLength(255)]],
    artist_id: [this.artwork.artist_id, [Validators.required]],
    type_id: [this.artwork.slug, [Validators.required]],
    description: [this.artwork.description, [Validators.required, Validators.minLength(100)]],
    creation_date: [this.artwork.creation_date],
    acquisition_date: [this.artwork.acquisition_date],
    price: [this.artwork.price, [Validators.required]],
    status: [this.artwork.status, Validators.required],
    meta_tag: this.fb.group( {
      page_title: [this.artwork.meta_tag.page_title],
      h1_title: [this.artwork.meta_tag.h1_title],
      description: [this.artwork.meta_tag.description],
    }
  )
}

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 85   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

```

    page_meta_description: [this.artwork.meta_tag.page_meta_description]
  }},
});

```

```

this.artworkForm.get('slug')?.valueChanges.subscribe(value => {
  const slug = this.slugUtilService.generateSlug(value);
  this.artworkForm.get('slug')!.setValue(slug, {emitEvent: false});
});
}

```

```

ngOnInit(): void {
  const idParam = this.route.snapshot.paramMap.get('id');
  const id = idParam ? Number(idParam) : 0;

```

```

  if (id) {
    this.isNewArtwork = false;
    this.getArtworkData(id);
  } else {
    this.isLoading = false;
  }

```

```

  this.getArtistList();
  this.getArtTypeList();
  this.artworkForm.markAllAsTouched();
}

```

```

getArtworkData(id: number) {
  this.artworkService.getArtworkById(id).subscribe(
    (response: any) => {
      if (response.success) {
        this.artwork = response.data.data;
        this.additionalData = response.data.additional_data;
        this.artworkForm.patchValue(this.artwork);

        if (this.artwork.meta_tag) {

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 86   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

```

        this.artworkForm.patchValue({meta_tag: this.artwork.meta_tag});
    }
} else {
    this.toastr.error(response.message, 'Помилка');
    this.router.navigate(['/admin/artworks/new']);
}
this.isLoading = false;
},
(error) => {
    this.errorHandler.handleError(error);
    this.isLoading = false;
    this.router.navigate(['/admin/artworks/new']);
}
);
}

```

```
onSubmit() {
```

```
    this.artworkForm.markAllAsTouched();
```

```
    if (this.artworkForm.valid) {
```

```
        this.isLoading = true;
```

```
        if (this.isNewArtwork) {
```

```
            this.createArtwork();
```

```
        } else {
```

```
            this.updateArtwork();
```

```
        }
```

```
    }
```

```
}
```

```
createArtwork() {
```

```
    this.artworkService.createArtwork(this.artworkForm.value).subscribe(
```

```
        (response: any) => {
```

```
            if (response.success) {
```

```
                this.isNewArtwork = false;
```

```
                const newArtworkId = response.data.id; // Assuming the response contains the new artist's ID
```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 87   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

```

this.router.navigate(['admin/artworks/${newArtworkId}']).then(() => {
  this.toastr.success(response.message, 'Успіх');
  this.getArtworkData(newArtworkId); // Fetch the new artist's data
});
} else {
  this.toastr.error(response.message, 'Помилка');
}
this.isLoading = false;
},
(error) => {
  this.errorHandler.handleError(error);
  this.isLoading = false;
}
);
}

updateArtwork() {
  this.artworkService.updateArtwork(this.artwork.id, this.artworkForm.value).subscribe(
    (response: any) => {
      if (response.success) {
        const newArtworkId = response.data.id; // Assuming the response contains the new artist's ID
        this.getArtworkData(newArtworkId); // Fetch the new artist's data
        this.toastr.success(response.message, 'Успіх')
      } else {
        this.toastr.error(response.message, 'Помилка');
      }
      this.isLoading = false;
    },
    (error) => {
      this.errorHandler.handleError(error);
      this.isLoading = false;
    }
  );
}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 88   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

```

openDeleteModal(artworkId: number): void {
  this.modalRef = this.modalService.open(ConfirmDeleteModalComponent, {
    data: {artworkId}
  });

  this.modalRef.onClose.subscribe((confirmed: boolean) => {
    if (confirmed) {
      this.deleteArtwork(artworkId);
    }
  });
}

deleteArtwork(artworkId: number): void {
  this.artworkService.deleteArtwork(artworkId).subscribe((response): any => {
    if (response.success) {
      this.router.navigate(['admin/artworks']).then(() =>
        this.toastr.success(response.message, 'Успіх')
      );
    } else {
      this.toastr.error(response.message, 'Помилка');
    }
  },
  error => {
    this.errorHandler.handleError(error);
  }
  );
}

getArtistList(){
  this.artistService.getAllArtists({
    status_list: [1], // Only Active
  }).subscribe((response): any => {
    if (response.success) {
      this.artistList = response.data;
    } else {

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 89   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

```

        this.toastr.error(response.message, 'Помилка');
    }
},
error => {
    this.errorHandler.handleError(error);
}
);
}

```

```

getArtTypeList(){
    this.artTypeService.getAllArtTypes({
        status_list: [1], // Only Active
    }).subscribe((response): any => {
        if (response.success) {
            this.artTypeList = response.data;
        } else {
            this.toastr.error(response.message, 'Помилка');
        }
    },
    error => {
        this.errorHandler.handleError(error);
    }
    );
}

```

```

get title(): AbstractControl {
    return this.artworkForm.get('title')!;
}

```

```

get slug(): AbstractControl {
    return this.artworkForm.get('slug')!;
}

```

```

get artist_id(): AbstractControl {
    return this.artworkForm.get('artist_id')!;
}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 90   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

}

```
get type_id(): AbstractControl {  
    return this.artworkForm.get('type_id')!;  
}
```

```
get description(): AbstractControl {  
    return this.artworkForm.get('description')!;  
}
```

```
get creation_date(): AbstractControl {  
    return this.artworkForm.get('creation_date')!;  
}
```

```
get acquisition_date(): AbstractControl {  
    return this.artworkForm.get('acquisition_date')!;  
}
```

```
get price(): AbstractControl {  
    return this.artworkForm.get('price')!;  
}
```

```
get status(): AbstractControl {  
    return this.artworkForm.get('status')!;  
}
```

```
get page_title(): AbstractControl {  
    return this.artworkForm.get('page_title')!;  
}
```

```
get h1_title(): AbstractControl {  
    return this.artworkForm.get('h1_title')!;  
}
```

protected readonly NATIONALITIES\_UA = NATIONALITIES\_UA;

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 91   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

artwork-detail.component.ts

```
import {Injectable} from '@angular/core';
import {HttpClient} from '@angular/common/http';
import {Observable} from 'rxjs';
import {environment} from "../../environments/environment";
```

```
@Injectable({
  providedIn: 'root'
})
```

```
export class ArtworkService {
  private apiUrl = `${environment.mainSiteUrl}/api/artworks`;
```

```
  constructor(private http: HttpClient) {
  }
```

```
  getAllArtworks(data: any): Observable<any> {
    let queryData = "";
```

```
    if (data.page) {
      queryData += `page=${data.page}`;
    }
```

```
    if (data.status_list && Array.isArray(data.status_list)) {
      data.status_list.forEach((item: any) => {
        if (queryData) {
          queryData += `&status_list[]={${item}}`;
        } else {
          queryData += `status_list[]={${item}}`;
        }
      });
    }
  }
```

```
  const url = `${this.apiUrl}?${queryData}`;
```

```
  return this.http.get(url);
}
```

```
  getArtworkById(id: number): Observable<any> {
    return this.http.get(`${this.apiUrl}/show/${id}`);
  }
```

```
  createArtwork(artworkData: any): Observable<any> {
    return this.http.post(`${this.apiUrl}/store`, artworkData);
  }
```

```
  updateArtwork(id: number, artworkData: any): Observable<any> {
    return this.http.put(`${this.apiUrl}/update/${id}`, artworkData);
  }
```

```
  deleteArtwork(id: number): Observable<any> {
    return this.http.delete(`${this.apiUrl}/delete/${id}`);
  }
```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 92   |

## Трейт для обробки завантажуваних зображень

HandlesImageUploadsTrait.php&lt;?php

```
namespace App\Traits;
```

```
use App\Enums\Api;
```

```
use App\Enums\ModuleType;
```

```
use App\Models\Images\Image;
```

```
use App\Services\Logs\CreateGeneralLogService;
```

```
use Illuminate\Http\UploadedFile;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use Illuminate\Support\Facades\Config;
```

```
use Illuminate\Support\Facades\Storage;
```

```
use Illuminate\Support\Str;
```

```
trait HandlesImageUploadsTrait
```

```
{
```

```
    public function uploadImage($image, $type = 'default'): bool|string|null
```

```
    {
```

```
        $directory = $this->getDirectoryByType($type);
```

```
        if ($image instanceof UploadedFile && $this->isValidImage($image)) {
```

```
            Storage::disk('public')->makeDirectory($directory);
```

```
            // Generate a unique filename
```

```
            $filename = $this->generateUniqueFilename($image);
```

```
            // Store the image with the unique filename
```

```
            return $image->storeAs($directory, $filename, 'public');
```

```
        }
```

```
        return null;
```

```
    }
```

```
    public function uploadImages(array $images, $type = 'default'): array
```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Савченко Д.П. |        |      | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                          | 93   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

```

{
    $paths = [];

    foreach ($images as $image) {
        $path = $this->uploadImage($image, $type);

        if (!empty($path)) {
            $paths[] = $path;
        }
    }

    // Remove domain from file paths
    return array_map(function ($path) {
        return $this->removeDomainFromPath($path);
    }, $paths);
}

private function removeDomainFromPath(string $path): array|string
{
    if (!empty($path)) {
        return str_replace(public_path(), "", $path);
    }

    return $path;
}

private function isValidImage(UploadedFile $file): bool
{
    return in_array($file->getMimeType(), ['image/jpeg', 'image/png', 'image/webp']);
}

private function generateUniqueFilename(UploadedFile $file): string
{
    // Generate a unique identifier
    $uuid = Str::uuid()->toString();

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 94   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |

```

// Get the file extension
$extension = $file->getClientOriginalExtension();

// Generate the unique file name
return $uuid . '-' . time() . '.' . $extension;
}

private function getDirectoryByType($type): string
{
    $directories = Config::get('filesystems.directories');

    return $directories[$type] ?? $directories['default'];
}

public function saveImage($image, $params): ?Image
{
    if (!empty($image) && !empty($params['module_id']) && !empty($params['object_id'])) {
        return Image::create([
            'module_id' => $params['module_id'],
            'object_id' => $params['object_id'],
            'url' => $image,
        ]);
    }

    return null;
}

public function saveImages($images, $params): array
{
    $storedImages = [];

    if (!empty($images) && !empty($params['module_id']) && !empty($params['object_id'])) {
        foreach ($images as $imageItem) {
            $storedItem = $this->saveImage($imageItem, $params);
        }
    }
}

```

|      |      |               |        |      |                            |      |
|------|------|---------------|--------|------|----------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІІ3.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                            | 95   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                            |      |

```

        if (!empty($storedItem)) {
            $storedImages[] = $storedItem;
        }
    }
}

return $storedImages;
}

public function removeImage($id): bool
{
    $data = Image::findOrFail($id);

    $path = $data->url;
    $fullPath = storage_path('app/public/'. $path); // Assuming the full path of the image file

    if (!file_exists($fullPath)) {
        return false; // File doesnt exist, return false
    }

    // Delete the file from the storage disk
    Storage::disk('public')->delete($path);
    /* General Logs*/
    (new CreateGeneralLogService([
        'module_name' => ModuleType::getKey($data->module_id),
        'module_id' => $data->object_id,
        'user_id' => Auth::user()->id,
        'action' => Api::DeleteFile
    ]))();
    /* General Logs*/
    // Delete the record from the database
    return $data->delete();
}
}

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Савченко Д.П. |        |      | ІІЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
|      |      | Кубрак Ю.О.   |        |      |                           | 96   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           |      |