

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток підтримки виставкової діяльності художника»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-71-9 Талаян Владислав Олександрович

Кваліфікаційну роботу бакалавра  
захищено на засіданні ЕК

з оцінкою \_\_\_\_\_ «\_» \_\_\_\_\_ 2021 р.

Науковий керівник \_\_\_\_\_

(підпис)

к.т.н., Антипенко В.П.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії \_\_\_\_\_

(підпис)

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Суми-2021

**Сумський державний університет**

**Факультет електроніки та інформаційних технологій**

**Кафедра комп'ютерних наук**

**Секція інформаційних технологій проектування**

**Спеціальність 122 «Комп'ютерні науки»**

**Освітньо-професійна програма «Інформаційні технології проектування»**

**ЗАТВЕРДЖУЮ**

Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик

«\_\_\_» \_\_\_\_\_ 2021 р.

## **ЗАВДАННЯ**

### **НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

Тальянін Владислав Олександрович

**1 Тема проекту** Web-додаток підтримки виставкової діяльності художника

**керівник роботи** Антипенко Вікторія Петрівна ,к.т.н. \_\_\_\_\_ ,

затверджені наказом по університету від «\_\_\_» \_\_\_\_\_ 2021 р. № \_\_\_\_\_

**2 Строк подання студентом роботи** «\_\_\_» \_\_\_\_\_ 2021 р.

**3 Вхідні дані до роботи** технічне завдання на розробку web-додатка підтримки виставкової діяльності художника

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** аналіз предметної області, проектування web-додатку, розробка web-додатку, висновки

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** Презентація \_\_\_\_\_ слайди

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

**7 Дата видачі завдання** \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Планування робіт	18.04.21- 20.04.21	
2	Постановка завдання проекту	20.04.21- 21.04.21	
3	Аналіз предметної області	22.04.21 –24.04.21	
4	Аналіз вимог технічної реалізації	25.04.21 –28.04.21	
5	Постановка задачі та вибір методів розробки	29.04.21-30.04.21	
6	Аналіз об'єкту розробки	01.05.21-03.05.21	
7	Вибір методів та засобів реалізації	04.05.21-06.05.21	
8	Планування проекту	07.05.21-13.05.21	
9	Розробка проекту	14.05.21-01.06.21	
10	Тестування проекту	02.06.21-07.06.21	
11	Здача проекту	08.06.21-09.06.21	

**Студент** \_\_\_\_\_  
(підпис)

Тялянін В.О.

**Керівник роботи** \_\_\_\_\_  
(підпис)

к.т.н., Антипенко В.П.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки виставкової діяльності художника».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел, найменувань та додатків. Загальний обсяг роботи – 65 сторінки, у тому числі 38 сторінок основного тексту, 2 сторінки списку використаних джерел та 25 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки виставкової діяльності художника. Здійснено розробку web-додатку з привабливим дизайном і інтуїтивно зрозумілим інтерфейсом і створено та структуровано базу даних. Додана можливість перегляду робіт художника та графіку виставок й оформлення замовлення на картину. Результатом даної роботи є web-додаток створений для підтримки виставкової діяльності художника. Практичне значення роботи полягає у розробці зручного та стильного web-портфоліо для творця в області образотворчого мистецтва. Використання даного web-додатку надасть можливість збільшити аудиторію художника наряду з кількістю замовлень за рахунок зручного формату демонстрації його робіт.

Ключові слова: WEB-ДОДАТОК, WEB-ПОРТФОЛІО, ХУДОЖНИК, КАРТИНА, ОБРАЗОТВОРЧЕ МИСТЕЦТВО.

## ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	8
1.1 Дослідження актуальності проблеми .....	8
1.2 Аналіз аналогів .....	9
1.3 Постановка задачі.....	13
2 Моделювання web-додатку підтримки виставкової діяльності художника .....	15
2.1 Моделювання процесу замовлення картини через web-додатку підтримки виставкової діяльності художника в IDEF0 .....	15
2.2 Моделювання варіантів використання web-додатку підтримки виставкової діяльності художника.....	16
2.3 Проектування бази даних .....	18
3 Розробка web-додатку підтримки виставкової діяльності художника .....	20
3.1 Архітектура web-додатку.....	20
3.2 Програмна реалізація web-додатку .....	21
3.3 Використання web-додатку .....	28
3.4 Тестування web-додатку .....	37
ВИСНОВКИ.....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	39
ДОДАТОК А.....	41
ДОДАТОК Б.....	50
ДОДАТОК В .....	58
ДОДАТОК Г.....	59

## ВСТУП

У наш час багато людей, які займаються творчою діяльністю, але навіть маючи неймовірний талант, можна не бути поміченим публікою або роботодавцями, якщо правильно не продемонстрували власні навички. Представлення своєї творчості новій та уже досвідченій аудиторії було і завжди буде важливим аспектом мистецтва. Зараз це можна зробити за допомогою Інтернету. Останній надав можливість творцям ділитися власними досягненнями з людьми із різних кутків світу. Як і поява доступних радіоприймачів створила першу по-справжньому глобальну платформу для музики, так і соціальні мережі надали платформу для творців мистецтва. За допомогою Всесвітньої мережі кожен може дізнатися, проте, які заходи присвячені мистецтву проводяться в даний момент, ознайомитися з роботами конкретного автора та замовити роботу. Більшість творців використовують соціальні мережі для організації власної діяльності, але вони використовують особливий макет, який може обмежити можливості для представлення мистецтва. Для людей, які професійно займаються творчою діяльністю найкращою альтернативою стане власний сайт-портфоліо, на якому вони можуть у зручному форматі продемонструвати свої роботи.

Сайт-портфоліо – це особливий тип персонального сайту, створений з метою презентації творчості його власника. За його допомогою можна набагато легше представити власні роботи у зручному вигляді.

В онлайн-портфоліо творець може представити свої роботи набагато ширшій аудиторії, ніж в галереях і музеях. Більш того, без захоплюючого портфоліо досить складно виділитися серед конкурентів.

Тому, метою даного проекту є розробка web-додатку підтримки виставкової діяльності художника. Основним призначенням web-додатку є демонстрація робіт художника, інформування аудиторії про заплановані виставки, а також надання можливості приймати замовлення на картини. Для досягнення мети потрібно вирішити наступні задачі:

- провести аналіз предметної області;
- виконати аналіз аналогів і визначити їх переваги та недоліки;
- розробити концептуальний дизайн web-додатку;
- виконати front-End розробку;
- виконати back-End розробку;
- розробити та підключити базу даних;
- протестувати web-додаток;
- завантажити розроблений web-додаток на хостинг.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Дослідження актуальності проблеми

Розвиток Інтернет-технологій вплинув на всі сфери діяльності людини й сфера образотворчого мистецтва не стала виключенням. Соціальні мережі, зокрема Instagram надали можливість художникам без проблем ділитися власними роботами з публікою й отримувати зворотній зв'язок від аудиторії. Але наявність одного тільки профілю в соціальній мережі вже не є достатнім, щоб виділитися серед інших художників. У наш час бажано мати власний web-сайт портфоліо, аби бути поміченим [1,2].

Web-портфоліо – це не просто зібрання картин, але також інструмент для бізнесу та маркетингу, який став невід'ємною частиною діяльності фотографів, дизайнерів, художників, архітекторів, стилістів і представників інших професій. Воно представляє набір найкращих робіт або найважливіших досягнень людини в певній сфері діяльності, представляє її уміння та основні якості [3]. За допомогою web-портфоліо автор може без проблем представити власні роботи зацікавленій аудиторії. Користуючись ним, роботодавець може легко зробити оцінку навичок автора, на основі якої прийняти рішення про запрошення спеціаліста на співбесіду. Це корисно як для людей, які шукають роботу, так і для тих, хто займається пошуком робітників. Найбільш поширеною формою web-портфоліо є невеликий web-сайт із контактною інформацією та прикладами робіт, де кожен може отримати доступ до галереї робіт дистанційно, що є зручним і актуальним в наші дні.

Тому, оскільки використання web-портфоліо в творчій діяльності є не просто бажаним, а необхідним, щоб стати поміченим аудиторією, було прийнято рішення розробити web-додаток підтримки виставкової діяльності художника, який дозволить йому виставляти власні роботи на огляд широкій публіці, повідомляти зацікавлених людей про графік актуальних виставок і спростити процес оформлення замовлення.



## 1.2 Аналіз аналогів

Для покращення якості розроблюваного продукту необхідно провести аналіз аналогів, виділити їх переваги та недоліки. Для цього було обрано три web-сайти такого ж типу.

Першим є сайт [caucezavaglia](#) [4] (рис 1.1), який представляє web-портфоліо художниці Кейсі Завалії. Він виконаний в досить мінімалістичному стилі. За допомогою використання тільки білих відтінків, а також наявність порожнього простору, даний сайт акцентує увагу саме на роботах художниці, а не на дизайні. Меню від контентної частини відділене тільки тонкою лінією. Шрифти є м'якими та приємними для очей. Галерея сайту (рис 1.2) представлена у вигляді звичайного слайдера, за допомогою якого можна обирати для перегляду тільки наступну або попередню картину, що не є досить зручним.

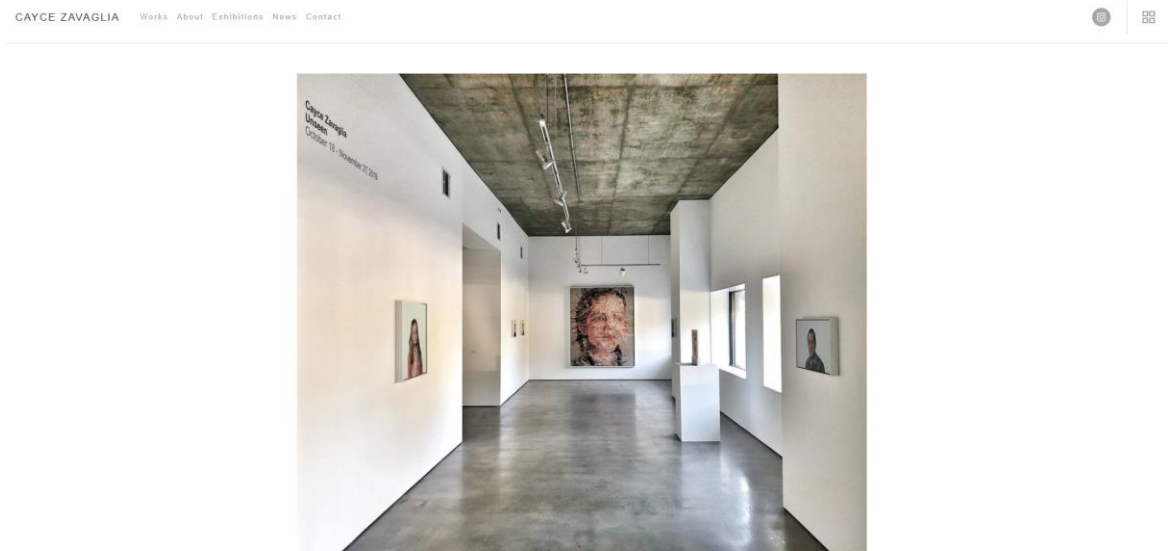


Рисунок 1.1 – Головна сторінка сайту «caucezavaglia»



Рисунок 1.2 – Галерея сайту «caucezavaglia»

На сайті midorikato [5] (рис 1.3), присвяченого роботам художниці Мідори Като, використані схожі з сайтом caucezavaglia дизайнерські прийоми, які привертають більшу увагу до робіт авторки, ніж до дизайну самого сайту. Він же створений у формі односторінкового лендінгу. У верхній частині сайту розташовані роботи художниці, а знизу – контактна інформація та форма зворотного зв'язку. Фіксоване меню розміщено з лівого боку сторінки. Галерея сайту midorikato (рис. 1.4) створена в більш класичному та зручному вигляді, ніж на сайті caucezavaglia.

TOP  
MIDORI KATO  
WORKS  
CALL ME, MAYBE  
ART STORE  
TWITTER  
INSTAGRAM

PHOTOGRAPHY



MIDORI KATO

CONTEMPORARY ARTIST IN TOKYO JAPAN

Рисунок 1.3 – Головна сторінка сайту «midorikato»



Рисунок 1.4 – Галерея сайту «midorikato»

Головна сторінка сайту shantellmartin [6] (рис 1.5), зосередженому на творчій діяльності художниці Шентелл Мартін, резонує з творчим почерком художниці. Елементи дизайну відповідають ілюстраціям митця. Використовуючи чорно-білу кольорову схему та напівжирний шрифт, її портфоліо ефективно передає її особистий стиль. Також фон навколо курсору стає анімованим, що надає сторінкам інтерактивності. Але інші сторінки сайту виглядають досить навантаженими. Галерея даного сайту (рис 1.6) на перший погляд може здатися доволі асиметричною та хаотичною, але до даного формату подачі швидко звикаєш. Картини зібрані в групи по темі, перед кожною групою розташована назва, а після – опис. Швидке переміщення між групами відбувається за рахунок меню, яке розміщене зліва.



Рисунок 1.5 – Головна сторінка сайту «shantellmartin»

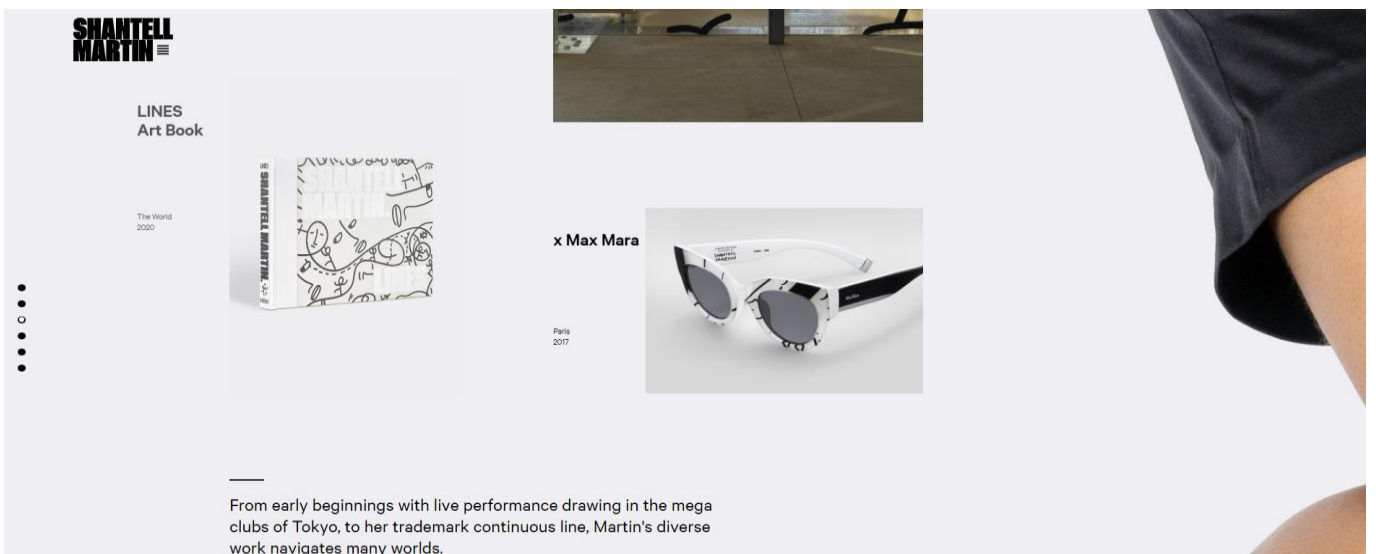


Рисунок 1.6 – Галерея сайту «shantellmartin»

У ході проведеного аналізу сайтів аналогів були виділені їх головні переваги та недоліки, результати якого наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика сайтів аналогів

Характеристика	<b>caucezavaglia</b>	<b>midorikato</b>	<b>shantellmartin</b>
Зрозумілий дизайн	+	+	+/-
Форма зворотного зв'язку	-	+	+
Можливість замовити роботу	-	+	+
Розклад виставок	+	-	+
Зручна галерея робіт	-	+	+/-

Виходячи з проведеного аналізу, стало зрозуміло, що галерея робіт сприймається краще, якщо вона виконана в класичному стилі таблиці, а не в формі слайдера. Тому було вирішено розробити галерею web-додатка саме у вигляді таблиці.

### 1.3 Постановка задачі

Розглянувши позитивні аспекти наявності портфоліо метою даної роботи є створення web-додатку підтримки виставкової діяльності художника. Головним завданням його використання є збільшення аудиторії митця наряду з кількістю замовлень за рахунок зручного формату демонстрації власних картин. Він повинен надавати можливість художнику виставляти картини для загального доступу публіки, приймати замовлення через форму зворотного зв'язку та повідомляти зацікавлених людей про заплановані виставки.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз предметної області;
- виконати аналіз продуктів-аналогів і визначити їх переваги та недоліки;

- розробити концептуальний дизайн web-додатку;
- виконати front-End розробку;
- виконати back-End розробку;
- розробити та підключити базу даних;
- протестувати web-додаток;
- завантажити розроблений web-додаток на хостинг.

Додаток буде складається з двох частин – клієнтської та адміністративної. Адміністративна панель надає змогу додавати та видаляти картини, створювати тематичні колекції робіт, змінювати графік проведення виставок і переглядати замовлення залишені користувачами. У свою черга клієнтська частина дозволить користувачам переглядати галереї робіт художника та графік виставок, а також залишити замовлення на картини.

Вимоги до дизайну розроблюваного web-додатку є такими:

- мінімалістичний стиль, який запам'ятовується, але при цьому залишається стриманим, щоб не відволікати увагу користувача від перегляду робіт художника;
- меню навігації web-додатку повинна розташовуватися вгорі;
- галерея web-додатку має бути виконана у табличному вигляді.

Для розробки даного web-додатку було обрано мови HTML [7] і CSS [8] для створення його верстки та JavaScript [9] для надання більшої візуальної привабливості. Як серверну мову програмування вибрано PHP [10] та MySQL [11] для роботи з базами даних.

Технічне завдання на створення web-додатку приведено у додатку А.

## 2 МОДЕЛЮВАННЯ WEB-ДОДАТКУ ПІДТРИМКИ ВИСТАВКОВОЇ ДІЯЛЬНОСТІ ХУДОЖНИКА

### 2.1 Моделювання процесу замовлення картини через web-додатку підтримки виставкової діяльності художника в IDEF0

IDEF0 – це практична мова опису бізнес-процесів. Завдяки цьому стандарту здійснюється передача інформації між консультантами, розробниками та користувачами [12].

На першому етапі зображуються контекстну діаграму – основний функціонал розроблюваного продукті без деталей.

Контекстна діаграма зображена на рисунку 2.1.

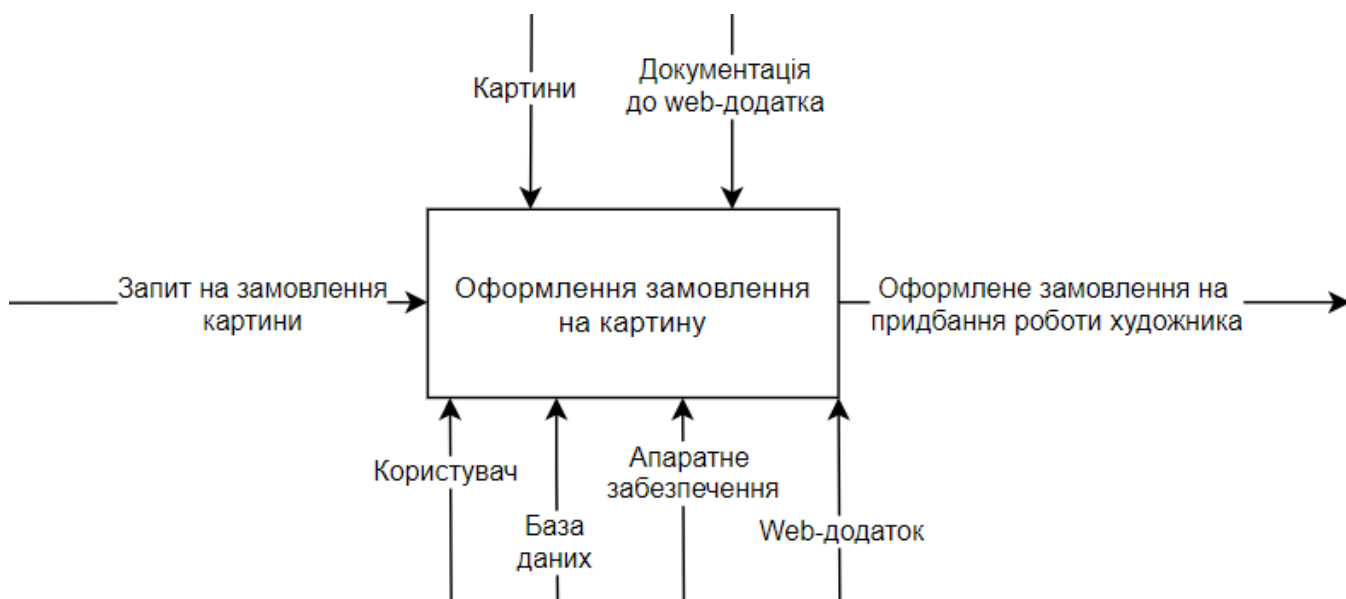


Рисунок 2.1 – Контекстна діаграма

На наступному етапі було проведено розбиття контекстної діаграми на діаграму декомпозицій, на якій зображена функції та детальна робота підсистем web-додатку.

Діаграма декомпозиції зображена на рисунку 2.2.

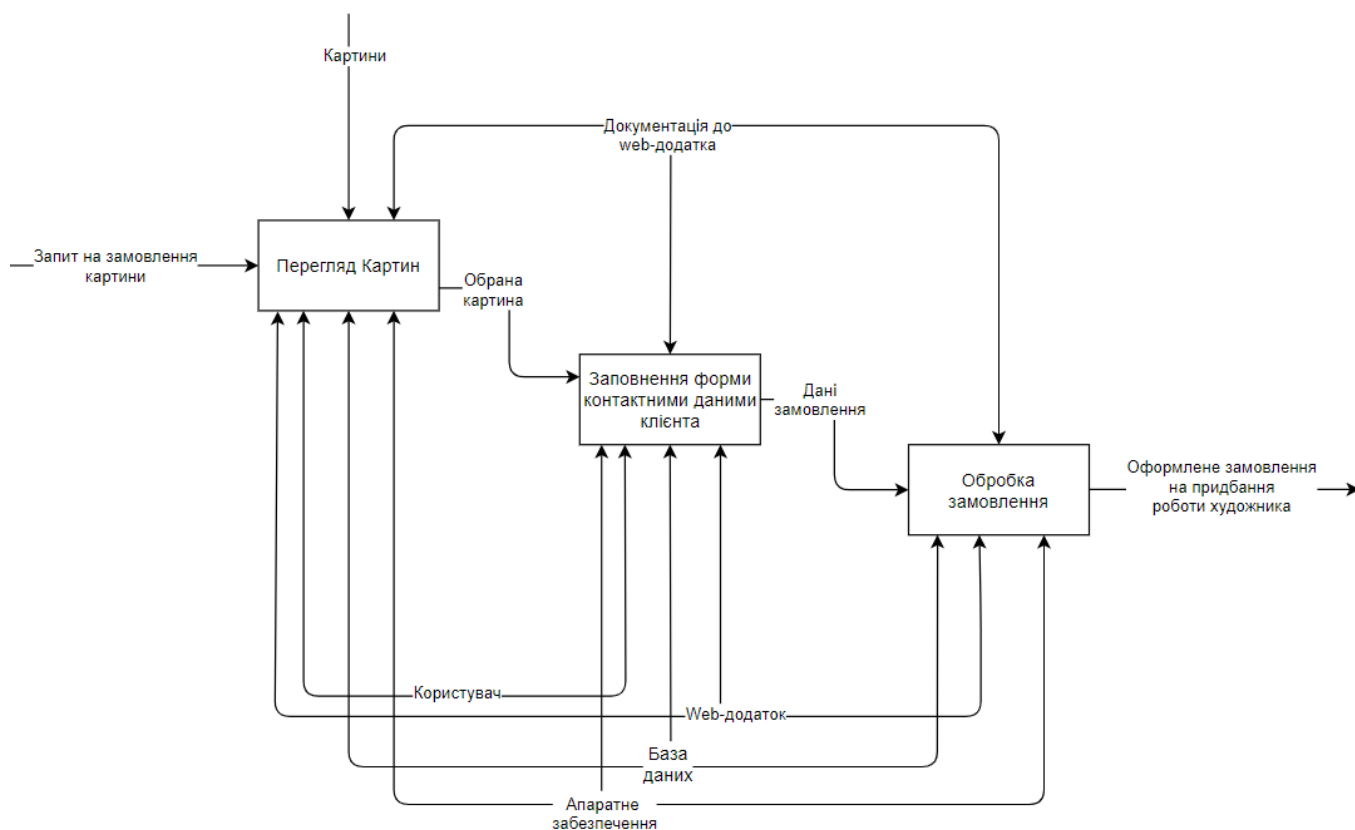


Рисунок 2.2 – Діаграма декомпозиції

## 2.2 Моделювання варіантів використання web-додатку підтримки виставкової діяльності художника

UML – мова графічного опису створення моделей. UML розроблений для використання в процесі розробки програмного забезпечення [13]. Основною ціллю застосування мови UML є встановлення взаєморозуміння між розробником і замовником щодо функціонування розробленого продукту [14].

Use Case – це діаграма варіантів використання, яка показує сценарії взаємодії між акторами та прецедентами й надає можливість зобразити розроблюваний продукт на концептуальному рівні [15].

Прецеденти – варіанти використання.

Актор – сутність, яка взаємодіє з продуктом.



Діаграма варіантів використання даного web-додатку представлена на рисунку 2.3.



Рисунок 2.3 – Діаграма варіантів використання

Варіанти використання web-додатку користувачем є такими:

- перегляд галереї картин;
- перегляд розкладу виставок;
- замовлення картин виставлених на продаж через форму зворотного зв'язку;
- замовлення картин за власними вимогами через форму зворотного зв'язку.

Варіанти використання web-додатку адміністратором є наступними:

- редагування розкладу;
- додавання і видалення картин;
- перевірка замовлень залишених користувачами.

- перегляд галереї картин;
- перегляд розкладу виставок.

### 2.3 Проектування бази даних

База даних розроблюваного web-додатку підтримки виставкової діяльності художника складається з наступних таблиць:

- Collection – зберігає назви колекцій, до яких може належати картина;
- Art – зберігає дані про картини: назва, шлях до зображення, ціна на картини, які виставлені на продаж, і булеве поле для відображення або приховування картини з продажу;
- Categories – зберігає назви категорій;
- Exhibitions – містить дані про виставки: назву, місце та дата проведення;
- Message – містить дані про замовлення;
- Clients – містить контактні дані клієнтів;
- Users – містить вхідні дані адміністратора.

Схема даних розроблюваної бази даних представлена на рисунку 2.4.

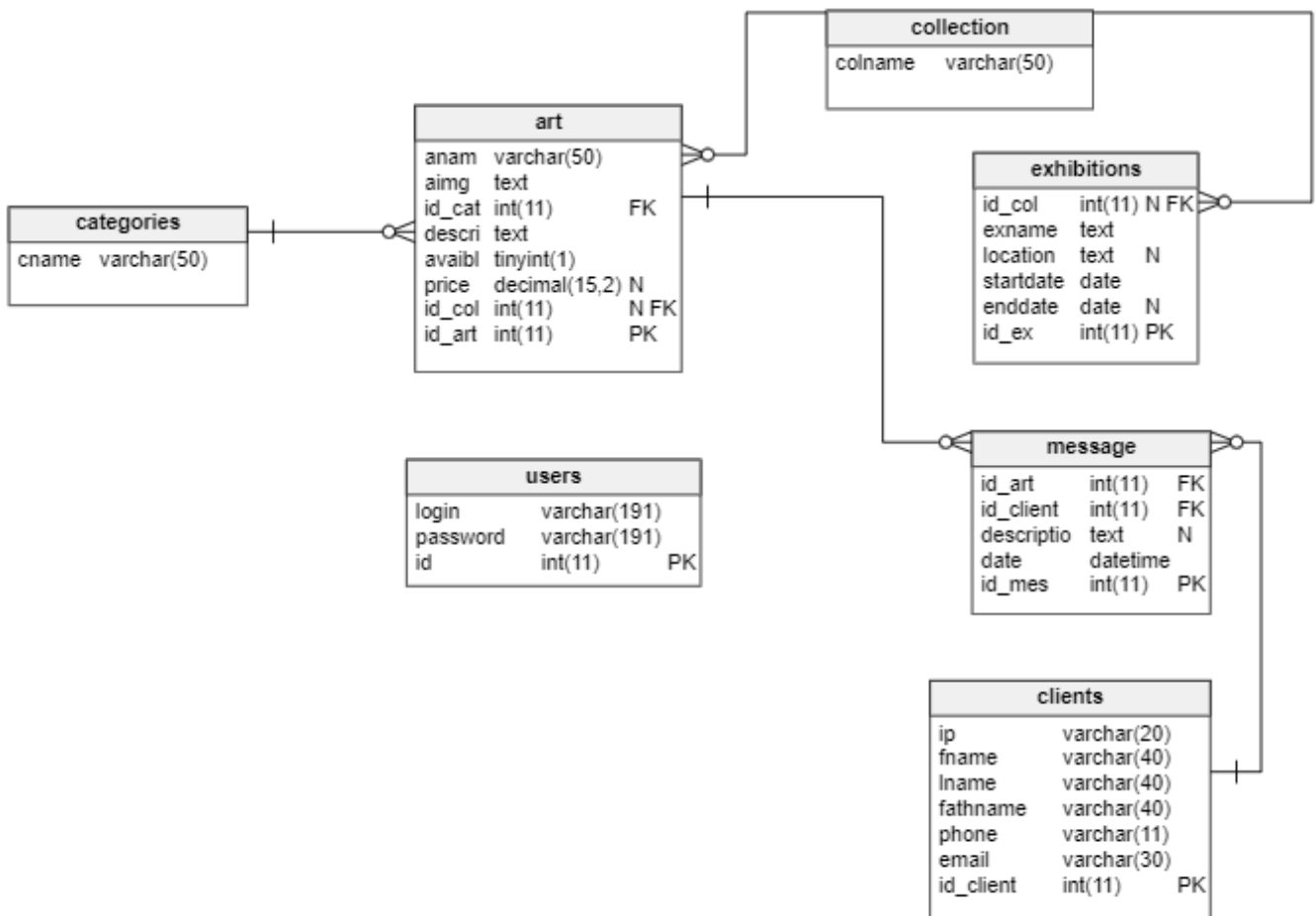


Рисунок 2.4 – Схема данных

### 3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ВИСТАВКОВОЇ ДІЯЛЬНОСТІ ХУДОЖНИКА

#### 3.1 Архітектура web-додатку

Web-додаток буде використовувати клієнт-серверну архітектуру – це мережева архітектура, в якій завдання або мережева навантаження розподілені між сервером та клієнтом. Користувач за допомогою браузера звертається до сервера на якому знаходиться web-додаток. Сторінки останнього слугують інтерфейсом, а php використаний для взаємодії з базою даних.

Архітектура web-додаток представлена на рисунку 3.1.

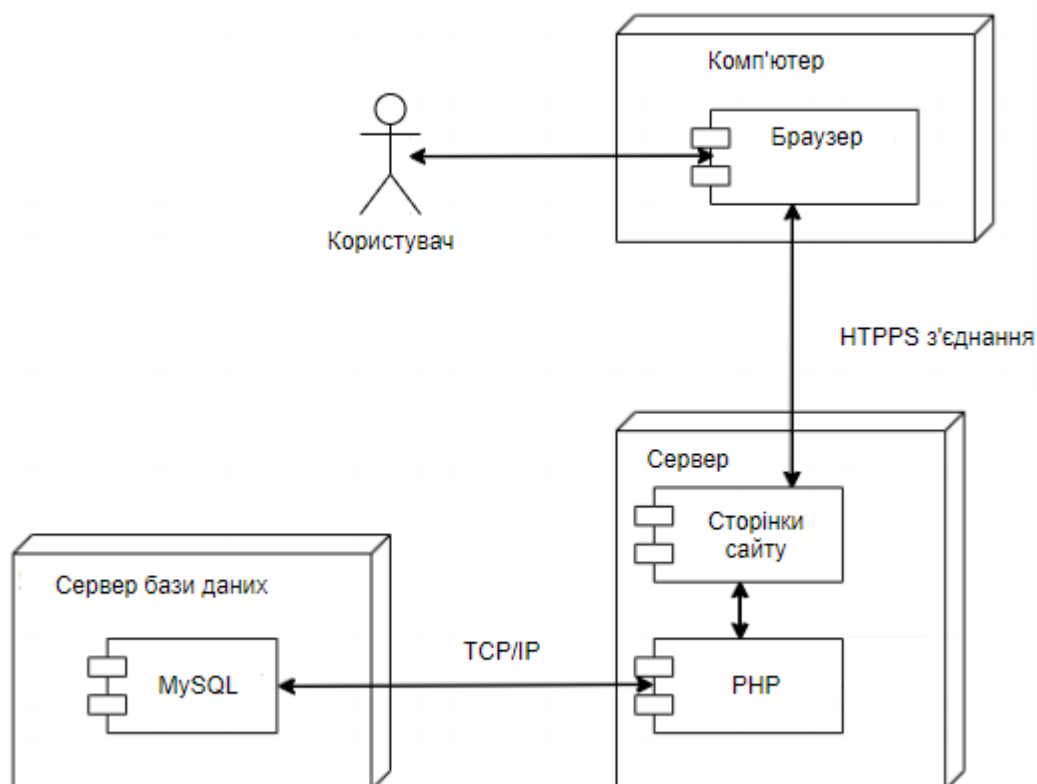


Рисунок 3.1 – Архітектура web-додатку

### 3.2 Програмна реалізація web-додатку

На першому етапі було розроблено дизайн web-додатку. Його виконано в мінімалістичному стилі. Кольорову гамму web-додатку у була розроблена в темних стриманих тонах для того, щоб зосередити увагу користувача саме на роботах художника. Для придання продукту візуальної динамічності для навігаційного меню було написано скрипт, який при завантаженні сторінки обирає один із декількох варіацій фону випадковим чином. Приклад зміни фону представлений на рисунку 3.2.



Рисунок 3.2 – Приклад зміни фону

На наступному етапі програмної реалізації проекту було створено базу даних (БД). Приклад запитів на створення таблиць бази даних зображено на рисунку 3.3.

```
CREATE TABLE `art` (
  `id_art` int(11) NOT NULL,
  `aname` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `aimg` text CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
  `id_cat` int(11) NOT NULL,
  `description` text NOT NULL,
  `available` tinyint(1) NOT NULL,
  `price` decimal(15,2) DEFAULT NULL,
  `id_col` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `categories` (
  `id_cat` int(11) NOT NULL,
  `cname` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `clients` (
  `id_client` int(11) NOT NULL,
  `ip` varchar(20) NOT NULL,
  `fname` varchar(40) NOT NULL,
  `lname` varchar(40) NOT NULL,
  `fathname` varchar(40) NOT NULL,
  `phone` varchar(11) NOT NULL,
  `email` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Рисунок 3.3 – SQL запит для створення таблиць бази даних

Після створення всіх таблиць була отримана фізична структура БД, яка відображена на рисунку 3.4.

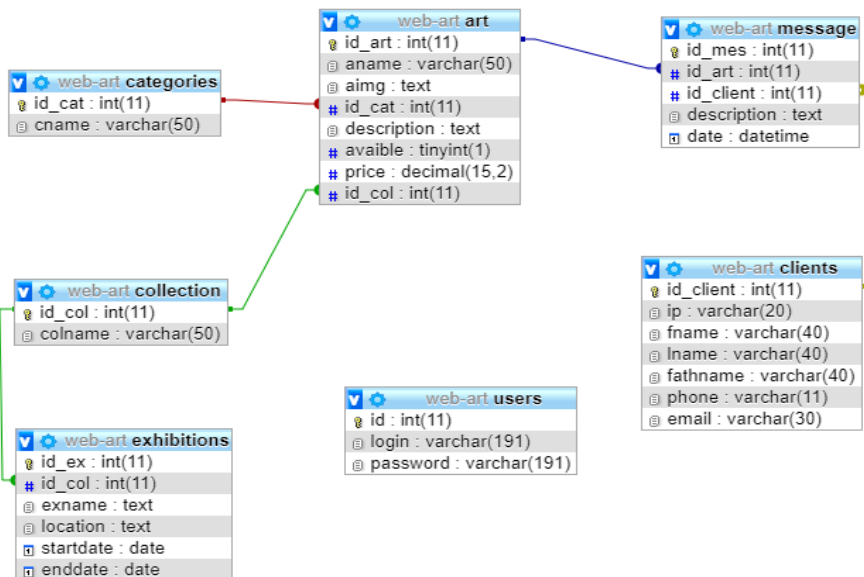


Рисунок 3.4 – Фізична структура бази даних

На наступному етапі за допомогою мови php було написано частина коду, яка відповідає за вивід картин із БД в галерею (рис. 3.5). Це здійснюємо за допомогою запиту до бази даних циклу do-while та printf.

```

$result = mysqli_query($db,"SELECT * FROM art where l=1 and id_art>0 $query_col $query_available order by $sorting");
$ava=0;
if (!$result)
{echo "<p><br> <strong></strong></p>";}
if (mysqli_num_rows($result) > 0){
$myrow = mysqli_fetch_array($result);
do
{
{$ava=$myrow["avaible"];
if($ava==0){$price=null;}
else{ $price=$myrow["price"]." грн";
}
printf("<a href='view_post.php?id_art=%s' class='column_center'><article class='product column_center'>
<div class='product_pic row_center'>
<img src='%s' alt='1'>
</div>
<h3>%s</h3>
<div class='price row_center'>%s</div>
</article></a>",$myrow["id_art"],$myrow["aimg"],$myrow["aname"], $price);}
while ($myrow = mysqli_fetch_array($result));}

```

Рисунок 3.5– Вивід картин до галереї

Також був написаний код для декількох пошукових фільтрів (рис. 3.6), які відповідають за порядок виводу картин, категорії до якої належить картина, а також чи доступна картина для продажу.

```

<div class="filter">
    <?
    $sor= $_GET["select1"];
    $sorting = $_GET["select1"];
    switch ($sor){
    case 'price-asc';
    $sorting = 'price ASC';
    break;
    case 'price-desc';
    $sorting = 'price DESC';
    break;
    case 'news';
    $sorting = 'id_art DESC';
    break;
    default:
    $sorting = 'id_art asc';
    break;
    }
    $id_cat=$_GET["select2"];
    if ((isset($_GET['select2'])&&($_GET['select2']!=0)) {$query_col = " AND id_cat=$id_cat";}else{$query_col ="";}
    $available=$_GET["av"];
    if (isset($_GET["av"])) {$query_available = " AND available=$available";}else{$query_available ="";}
    ?>
    <select name="select1" onchange='this.form.submit() '> <!--Supplement an id here instead of using 'name'-->
        <option value="def">За замовчуванням</option>
        <option value="news"<?php if ($sor== 'news')
    [echo "selected='selected'";]?>>Новинки</option>
        <option value="price-asc" <?php if ($sor== 'price-asc')
    [echo "selected='selected'";]?>>Спочатку дешеві</option>
        <option value="price-desc"<?php if ($sor== 'price-desc')
    [echo "selected='selected'";]?>>Спочатку дорогі</option>
    </select>
    <select name="select2" onchange='this.form.submit() '>
        <option value="0">Всі жанри</option>
    <?
    $rescol = mysqli_query($db,"SELECT * FROM categories");
    If (mysqli_num_rows($rescol) > 0){
    $mycol = mysqli_fetch_array($rescol);
    do
    {
    echo '<option value="'. $mycol['id_cat']. ' "' ;
    if ($mycol['id_cat']==$id_cat){ echo "selected='selected'";}
    echo '>';
    echo ' '. $mycol['cname']. ' ' ;
    echo '</option>';}
    while($mycol = mysqli_fetch_array($rescol));?></select><span>

```

Рисунок 3.6 – Пошукові фільтри

Потім був розроблений код для форми замовлення. Спочатку була написана та його частина, яка визначає IP користувача (рис. 3.7). Це зроблено для того, щоб можна було внести IP користувача в БД і при повторному замовленні поля форми автоматично заповнювались раніше введеними даними користувача. Обробка даних

с форми здійснюються за допомогою AJAX технології. За допомогою JS здійснюються збір даних із сторінки та перевірка на коректність введення (рис. 3.8), а за допомогою php вносяться дані в таблицю client та message (рис. 3.9).

```

]else{
    $aimg=$myrow["aimg"];
    $aname=$myrow["aname"];
    -$price=$myrow["price"];}
    $client = @$_SERVER['HTTP_CLIENT_IP'];
    $forward = @$_SERVER['HTTP_X_FORWARDED_FOR'];
    $remote = @$_SERVER['REMOTE_ADDR'];

    if(filter_var($client, FILTER_VALIDATE_IP)) $ip = $client;
    elseif(filter_var($forward, FILTER_VALIDATE_IP)) $ip = $forward;
    else $ip = $remote;
    $client = mysqli_query($db,"SELECT * FROM clients WHERE ip='$ip'");

]If (mysqli_num_rows($client) > 0){
    -$clientrow = mysqli_fetch_array($client);}
-?>

```

Рисунок 3.7 – Визначення IP користувача

```

collection.php | index.php | fom.php | fom.js | mail.php
1  $("#sendMail").on("click",function(){
2  var fname=$("#fname").val().trim();
3  var lname=$("#lname").val().trim();
4  var tname=$("#tname").val().trim();
5  var mail=$("#mail").val().trim();
6  var phone=$("#phone").val().trim();
7  var message=$("#message").val();
8  var ip=$("#ip").text();
9  var id_art=$("#id_art").text();
10
11  if(mail==""){
12      alert("Введіть email");
13      return false;
14  }
15  if(fname==""){
16      alert("Введіть ім'я");
17      return false;
18  }
19  if(lname==""){
20      alert("Введіть Прізвище");
21      return false;
22  }
23  if(tname==""){
24      alert("Введіть ваше ім'я по батькові");
25      return false;
26  }
27  if(phone==""){
28      alert("Введіть Телефон");
29      return false;
30  }
31  if(id_art==""){
32      alert("Неможливо визначити id картини");
33      return false;
34  }
35  if (isNaN(phone)) {
36      alert("Коректно введіть номер телефону");
37  }
38
39  $.ajax({
40      url: 'ajax/mail.php',
41      type: 'POST',
42      cache: false,
43      data: { 'fname':fname, 'lname':lname, 'tname':tname, 'phone':phone, 'mail':mail, 'id_art':id_art, 'ip':ip, 'message':message},
44      dataType: 'html',
45      beforeSend: function() {
46          $("#sendMail").prop("disabled", true);
47      },
48      success: function(data) {
49          if(!data) {
50              alert("Виникла помилка. Замовлення не залішено");
51          }
52          else{
53              $("#mailForm").trigger("reset");
54              alert(data);
55          }
56          $("#sendMail").prop("disabled", false)
57          document.location.href='index.php';
58      });
59  });

```

Рисунок 3.8 – JS файл для обробки форми



```

1 <?php
2 include ("../incl/db.php");
3 $mail=$_POST['mail'];
4 $fname=$_POST['fname'];
5 $lname=$_POST['lname'];
6 $tname=$_POST['tname'];
7 $phone=$_POST['phone'];
8 $message=$_POST['message'];
9 $id_art=$_POST['id_art'];
10 $id=1;
11 $ip=$_POST['ip'];
12 /* Здесь пишем что можно заносить информацию в базу */
13 $client = mysqli_query($db,"SELECT * FROM clients WHERE fname='$fname' and lname='$lname' and fathname='$tname' and phone='$phone' and email='$mail'");
14 if (mysqli_num_rows($client) > 0){
15     $cli = mysqli_fetch_array($client);
16     $id=$cli['id_client'];
17     if ($ip!=$cli['ip']){$clientupdate = mysqli_query ($db,"UPDATE clients SET ip='$ip' WHERE id_client='$id'");}
18 }
19 else{
20     $newclient = mysqli_query ($db,"INSERT INTO clients ( fname,lname,fathname,email,phone,ip) VALUES ('$fname', '$lname','$tname','$mail','$phone','$ip')");
21     $client2=mysqli_query($db,"SELECT * FROM clients WHERE fname='$fname' and lname='$lname' and fathname='$tname' and phone='$phone' and email='$mail'");
22     $cli = mysqli_fetch_array($client2);
23     $id=$cli['id_client'];
24 }
25 $result = mysqli_query ($db,"INSERT INTO message (id_art,id_client,description,date) VALUES ('$id_art', '$id','$message',NOW())");
26 if ($result == 'true') {echo "Заявка на заповнення успішно оформлена!";}
27 else {echo "Заявка на заповнення не була оформлена!";}

```

Рисунок 3.9 – Php файл для обробки форми

Потім був написаний код для сторінки графік виставок (рис. 3.10). Для виведення дат із БД у більш відповідному до стилю web-додатку форматі використовується масив с назвами місяців та функція date для форматування дати.

```

<h1>НАЙБЛИЖЧІ ВИСТАВКИ І ПОДІЇ</h1><hr>
<?
$date_m = array('січня', 'лютого', 'березня', 'квітня', 'травня', 'червня', 'липня', 'серпня', 'вересня', 'жовтня', 'листопада', 'грудня');
do
{ $timestart = strtotime($myrow['startdate']);

    $mysqlstartdate = $date_m[date('n', $timestart) - 1];
    $mysqlstartdate2 = date('j ', $timestart);

    echo' <article class="event">
    <time>' . $mysqlstartdate2 . ' ' . $mysqlstartdate . ' ';
    if (isset($myrow['enddate'])) {
        $timeend = strtotime($myrow['enddate']);
        $mysqlstartdate = $date_m[date('n', $timeend) - 1];
        $mysqlstartdate2 = date('j ', $timeend);
        echo' - ' . $mysqlstartdate2 . ' ' . $mysqlstartdate . ' ';
    }
    echo'</time><br>';
    echo'<span>' . $myrow['exname'] . '</span>';
    if (isset($myrow['id_col'])) { $id_col=$myrow['id_col'];
    $result2 = mysqli_query($db,"SELECT * FROM collection where id_col=$id_col");
    $myrow2 = mysqli_fetch_array($result2);
    echo'<br><span>Колекція: ' . $myrow2['colname'] . '<span>';
    if (isset($myrow['location'])) {echo'<br><span><span>Адреса: ' . $myrow2['location'] . '</span>';}
    echo'</article><hr>';}
while ($myrow = mysqli_fetch_array($result)); ?>

```

Рисунок 3.10 – Вивід графіку виставок

У тег <addre> вставляється адреса проведення виставки, яка записана в БД. За допомогою JS був написаний код, який перетворює всі дані, які знаходяться в тегу <addre> у посилання, що представляє пошуковий запит до Google Maps. Даний код зображено на рисунку 3.11.

```

$(document).ready(function(){
  $("adre").each(function(){
    var embed = "<a style='text-decoration: none; color:#bbb; font-size: 18px;'
href='https://maps.google.com/maps?&map;q="+ encodeURIComponent( $(this).text() ) +"&map;'>Місце проведення: "+$(this).text()+"<a>";
    $(this).html( embed );
  });
});

```

Рисунок 3.11 – Перетворення тегу в пошуковий запит

Для адміністративної панелі спершу був написаний код для авторизації (рис. 3.12), який звряє введений логін і пароль з тими, що знаходяться в БД.

```

1 <?php
2     require "incl/db.php";
3     $log=0;
4     if(isset($_SESSION['logged_user']) ){
5         echo "<script>document.location.href='admin.php';</script>";
6     }
7     else{
8         $data=$_POST;
9         if( isset($data['log']))
10        {
11            $errors=array();
12            $user=R::findOne('users', 'login = ?', array($data['login']));
13            if($user){
14                if(password_verify($data['password'], $user->password)){
15                    $_SESSION['logged_user']=$user;
16                    echo'Ви авторизовані';
17                    echo "<script>document.location.href='admin.php';</script>";
18                }
19            }
20        }
21        else{
22            echo 'неправильно введений логін або пароль'; }
23    }
24    else{
25        echo 'неправильно введений логін або пароль';
26        echo "<script>document.location.href='admin.php';</script>"; }}
27        else{echo "<script>document.location.href='admin.php';</script>"; }}

```

Рисунок 3.12 – Авторизація

Для панелі адміністратора був розроблений функціонал перевірки замовлень, а також додавання, редагування, видалення картин, виставки, категорій та колекцій. Для завантаження картини з попереднім переглядом використовується AJAX технології. Також даний код (рис. 3.13) виконує перевірку на формат а також за допомогою функції `getimagesize()` отримується роздільна здатність зображення картини по висоті й ширині, потім ці значення перевіряються на те чи більше вони, ніж 500 пікселів.

```

]if (!isset($_FILES[$input_name])) {
    exit;
}

$allow = array('jpg', 'jpeg', 'png', 'gif');

$url_path = '/art/tmp/';

$tmp_path = $_SERVER['DOCUMENT_ROOT'] . $url_path;

]if (!is_dir($tmp_path)) {
    mkdir($tmp_path, 0777, true);
}

$files = array();
$difff = count($_FILES[$input_name]) - count($_FILES[$input_name], COUNT_RECURSIVE);
]if ($difff == 0) {
    $files = array($_FILES[$input_name]);
} else {
]    foreach($_FILES[$input_name] as $k => $l) {
]        foreach($l as $i => $v) {
]            $files[$i][$k] = $v;
]        }
]    }
}

$response = array();
]foreach ($files as $file) {
    $error = $data = '';

    $ext = mb_strtolower(mb_substr(mb_strrchr(@$file['name'], '.'), 1));
]    if (!empty($file['error']) || empty($file['tmp_name']) || $file['tmp_name'] == 'none') {
]        $error = 'Не удалось загрузить файл.';
]    } elseif (empty($file['name']) || !is_uploaded_file($file['tmp_name'])) {
]        $error = 'Не удалось загрузить файл.';
]    } elseif (empty($ext) || !in_array($ext, $allow)) {
]        $error = 'Недопустимый тип файла.';
]    } else {
]        $info = @getimagesize($file['tmp_name']);
]        if (empty($info[0]) || empty($info[1]) || !in_array($info[2], array(1, 2, 3))) {
]            $error = 'Недопустимый тип файла.';
]        } else {

            $name = time() . '-' . mt_rand(1, 9999999999);
            $src = $tmp_path . $name . '.' . $ext;
            $thumb = $tmp_path . $name . '-thumb.' . $ext;

```

Рисунок 3.13 – Завантаження картин

### 3.3 Використання web-додатку

При переході на розроблений web-додаток користувач потрапляє на сторінку галереї (рис. 3.14), де відображені всі картини, які художник на нього завантажив.

Над картинами знаходяться пошукові фільтри, які відповідають за порядок відображення карти, категорії, до яких належить картини, та чи доступна картина для продажу.

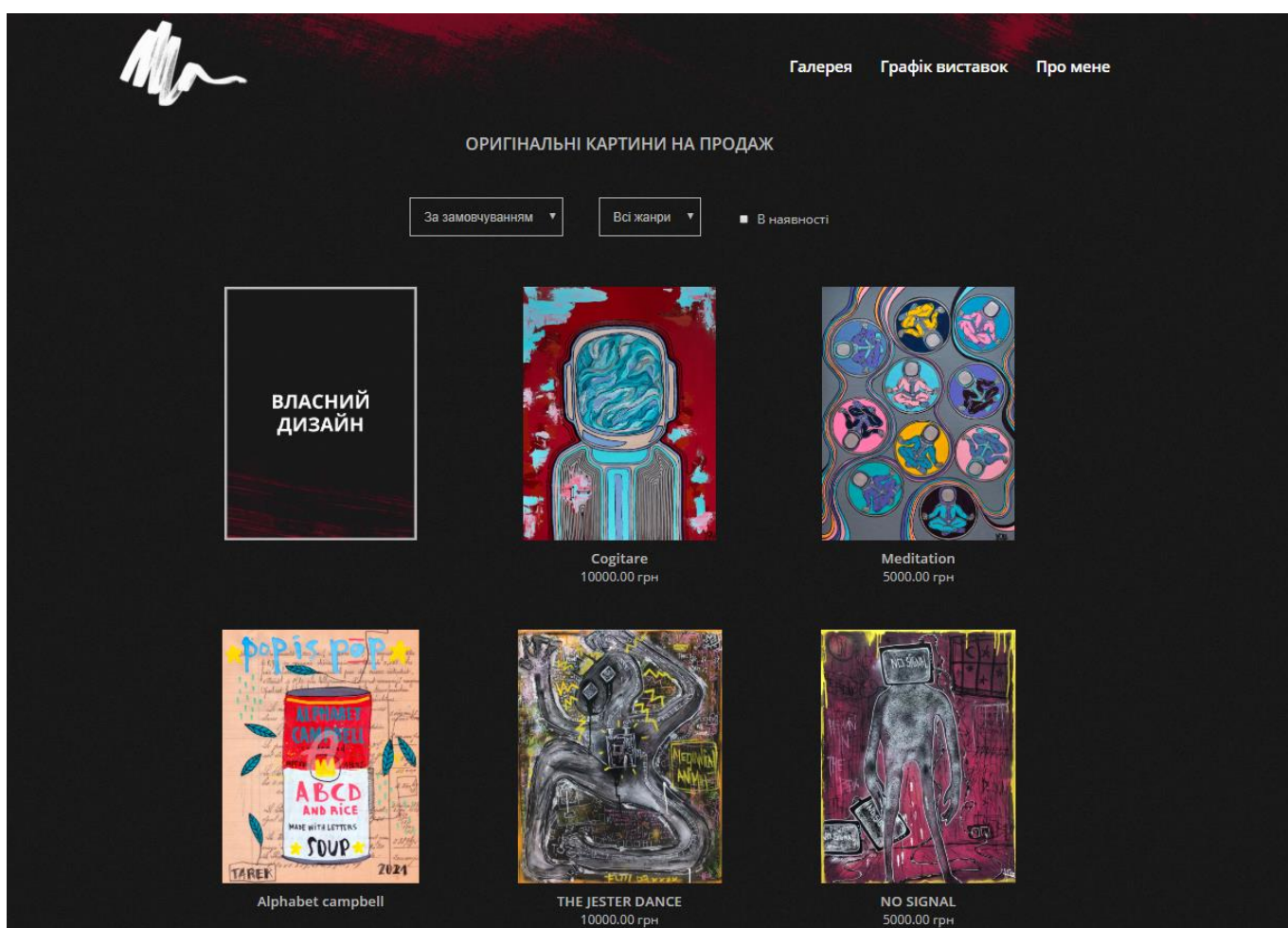


Рисунок 3.14 – Галерея

Обравши вподобану картину та натиснувши на неї, користувач потрапляє на сторінку самої картини (рис. 3.15), де справа відображені її характеристики, категорія й колекція, до якої вона належить, і ціна, а зліва зображена картина.

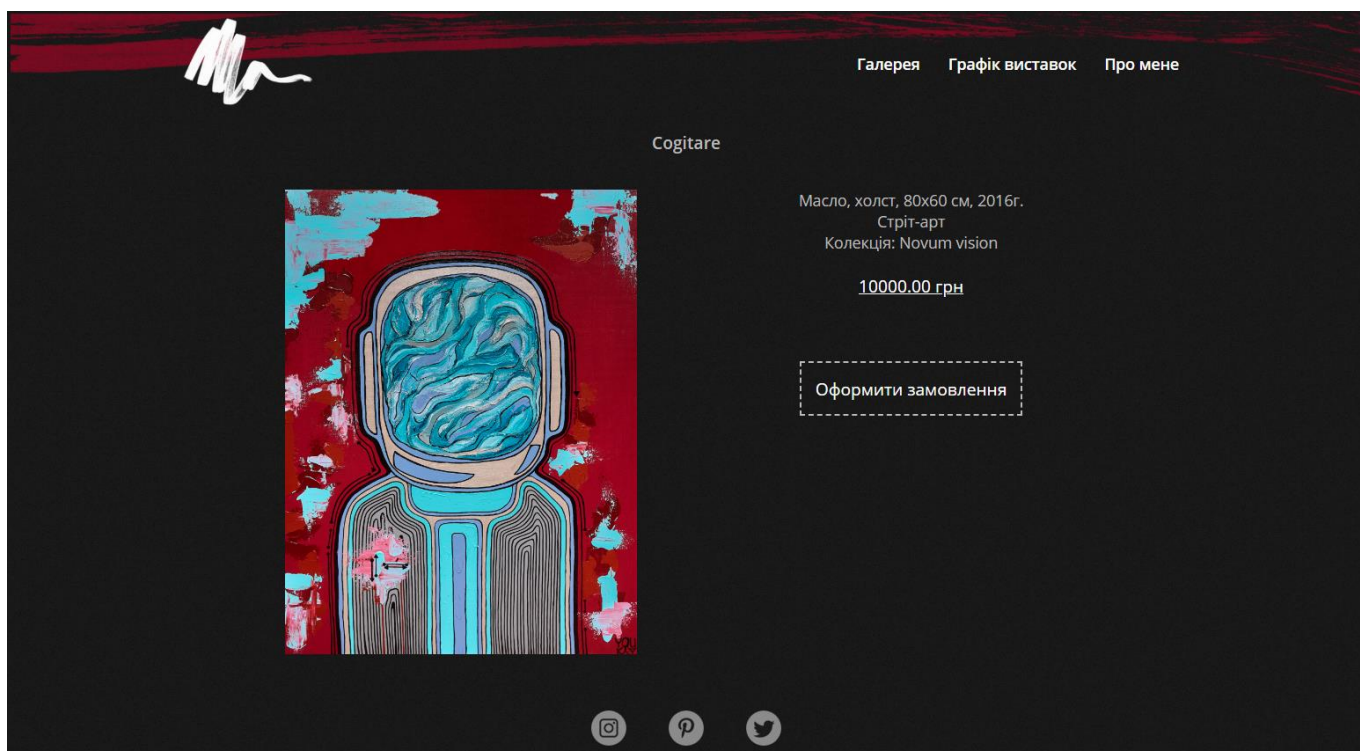


Рисунок 3.15 – Сторінка «Картини»

Для детального перегляду зображення можна натиснути на картину (рис. 3.16).

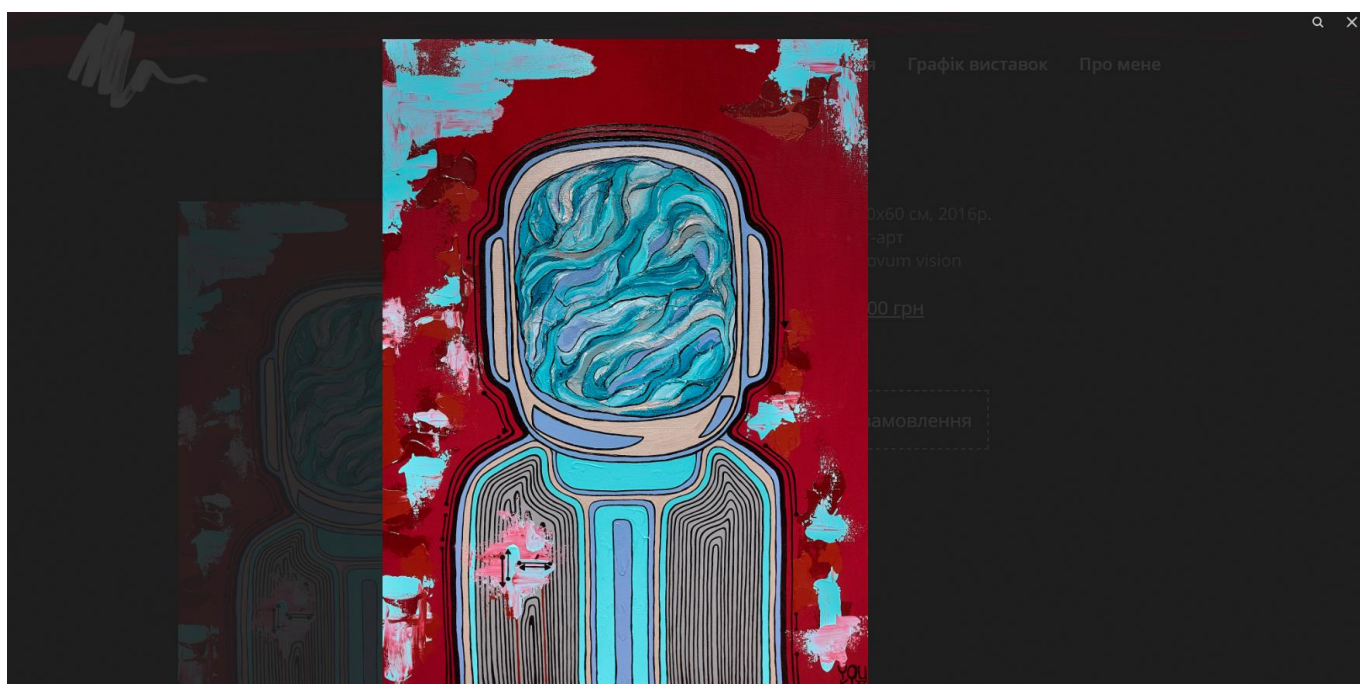


Рисунок 3.16 – Детальний перегляд картини

Натиснувши на кнопку «ОФОРМИТИ ЗАМОВЛЕННЯ», користувач потрапляє на сторінку замовлення (рис. 3.17), де зліва знаходиться форма, куди клієнт повинен ввести свої контактні дані, а справа – деталі замовлення. Якщо клієнт уже залишав замовлення з цього IP-адреса, форма заповнюється автоматично даними із БД.

ОФОРМЛЕННЯ ЗАМОВЛЕННЯ

Ваше ім'я:  
Владислав

Прізвище:  
Тальянін

По батькові:  
Олександрович

Email:  
Talyanin@mail.ru

Номер телефону:  
0950203989

Додаткові коментарі:

Відправити    Очистити

ДЕТАЛІ ЗАМОВЛЕННЯ:

Cogitare  
10000.00 грн.

Рисунок 3.17 – Форма замовлення

Якщо дані введені коректно, то після натискання на кнопку відправити користувачу висвітиться повідомлення, що замовлення успішно залишено (рис 3.18).

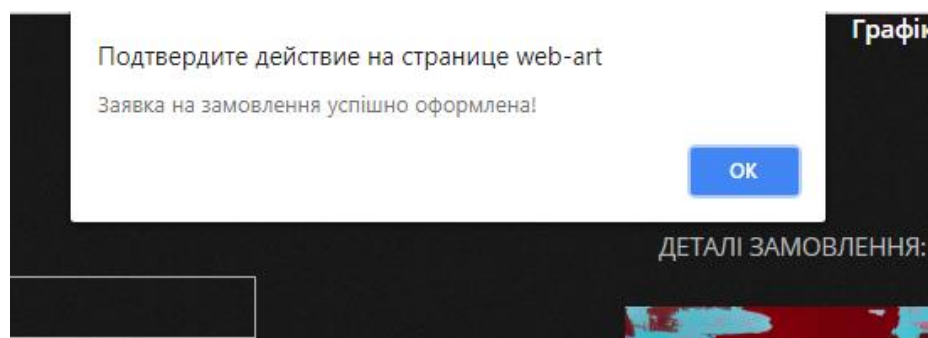


Рисунок 3.18 – Повідомлення про успішне замовлення

На web-додатку є сторінка, на якій відображений розклад найближчих виставок (рис. 3.19). На ній користувач може дізнатися де й коли проходить виставка даного художника. Натиснувши на місце проведення, користувачу відкриється Google Maps із уже введеним пошуковим запитом на пошук зазначеного місця.

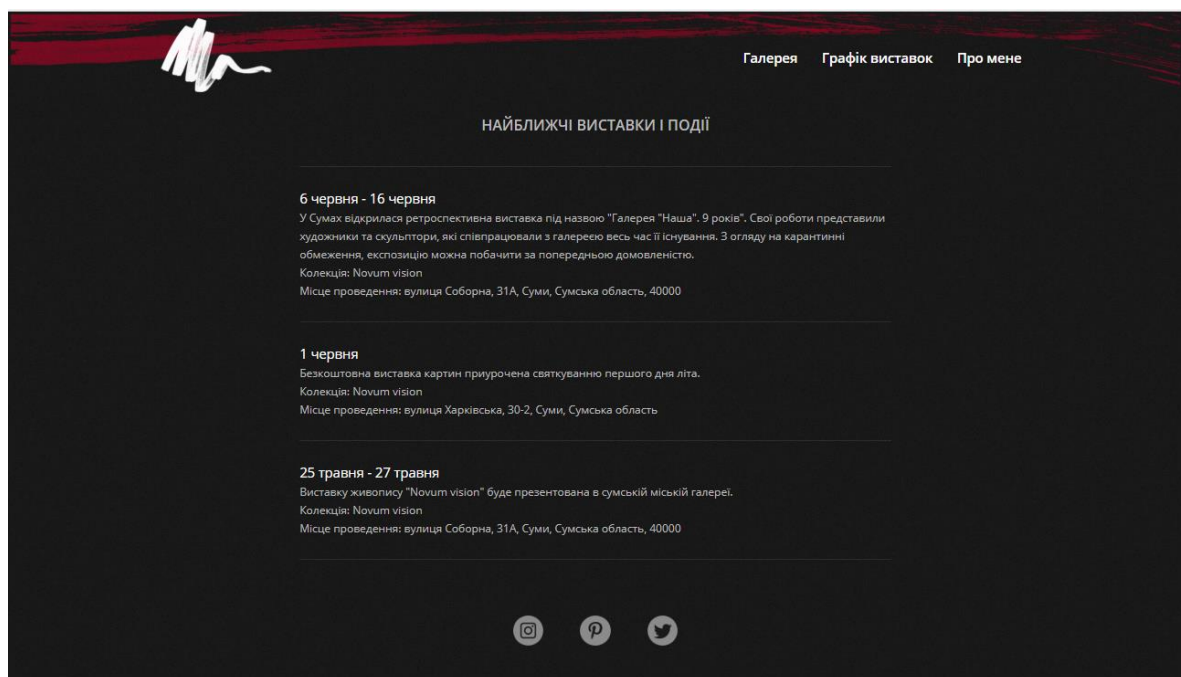


Рисунок 3.19 – Графік виставок

Також на web-додатку є сторінка «ПРО МЕНЕ» (рис. 3.20), де користувач може ознайомитися з короткою біографією художника та контактними даними для зв'язку з ним.

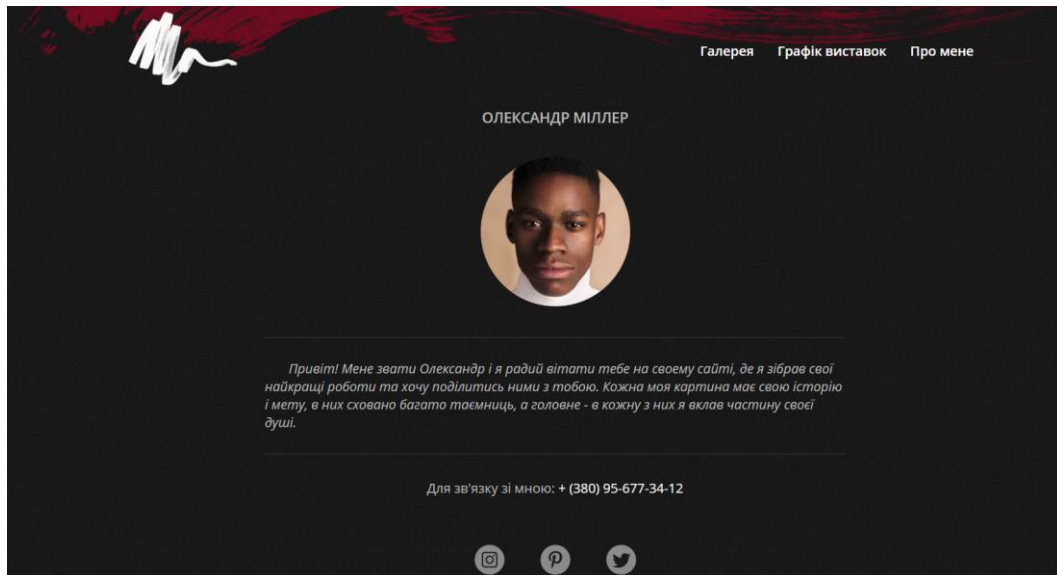


Рисунок 3.20– Про мене

Також web-додаток є повністю адаптивним. Приклад адаптивності сторінок зображено на рисунку 3.21.

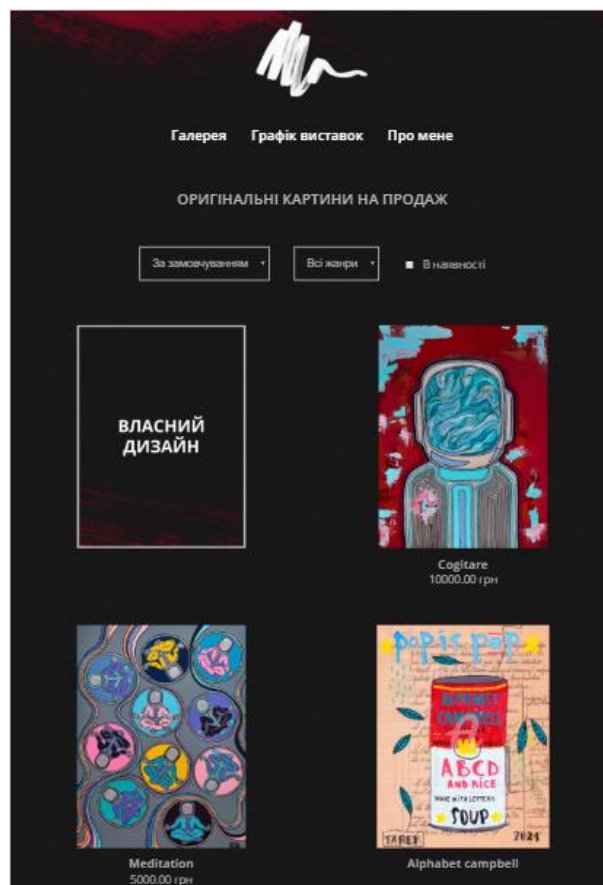


Рисунок 3.21 – Адаптивність сторінок



Для переходу в адміністративну панель адміністратору до адреси web-додатку необхідно додати /admin. Після чого йому відкриється сторінка авторизації (рис. 3.22), де треба ввести власний логін і пароль.

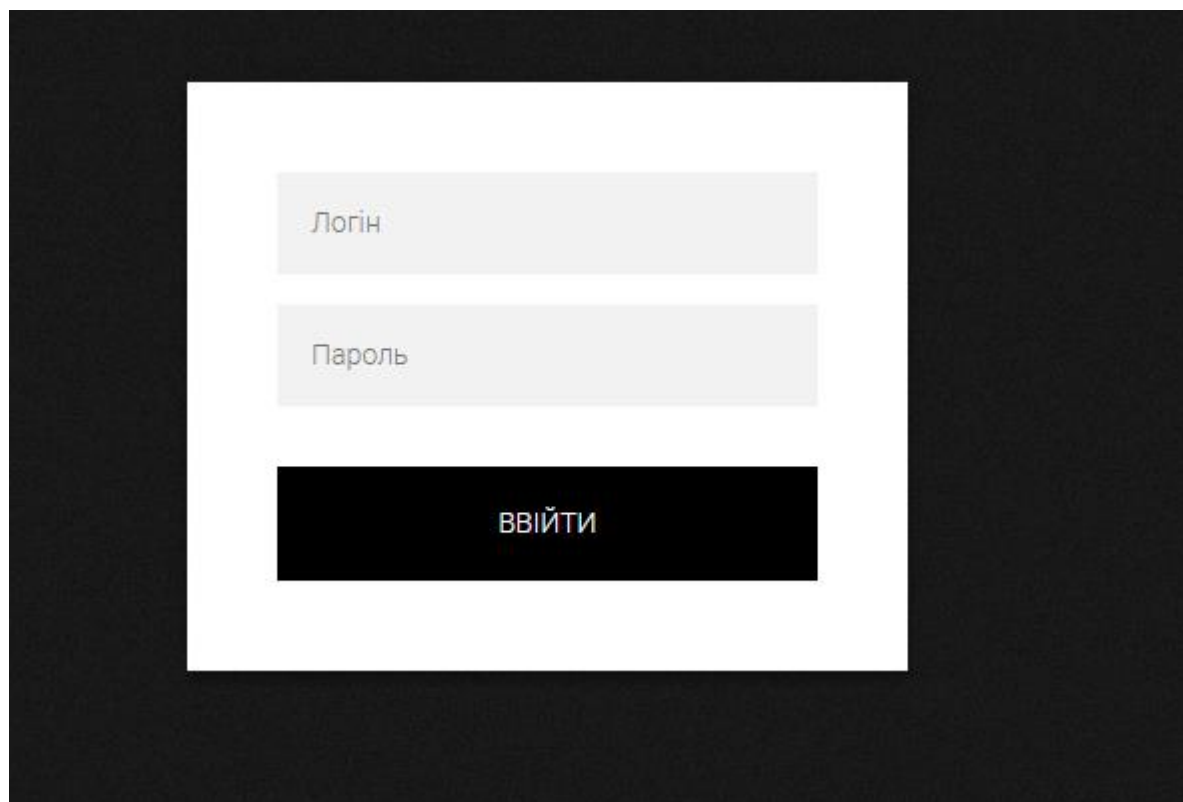


Рисунок 3.22 – Форма авторизації

Після вдалої авторизації користувач перейде на сторінку з переліком картин (рис. 3.23). За допомогою адміністративної панелі web-додаток надає можливість адміністратору переглядати замовлення та додавати, редагувати, видаляти картини, виставки, категорії, колекції. Для того, щоб додати картину треба натиснути на сторінці з переліком картин на картинку з надписом «Додати картину».

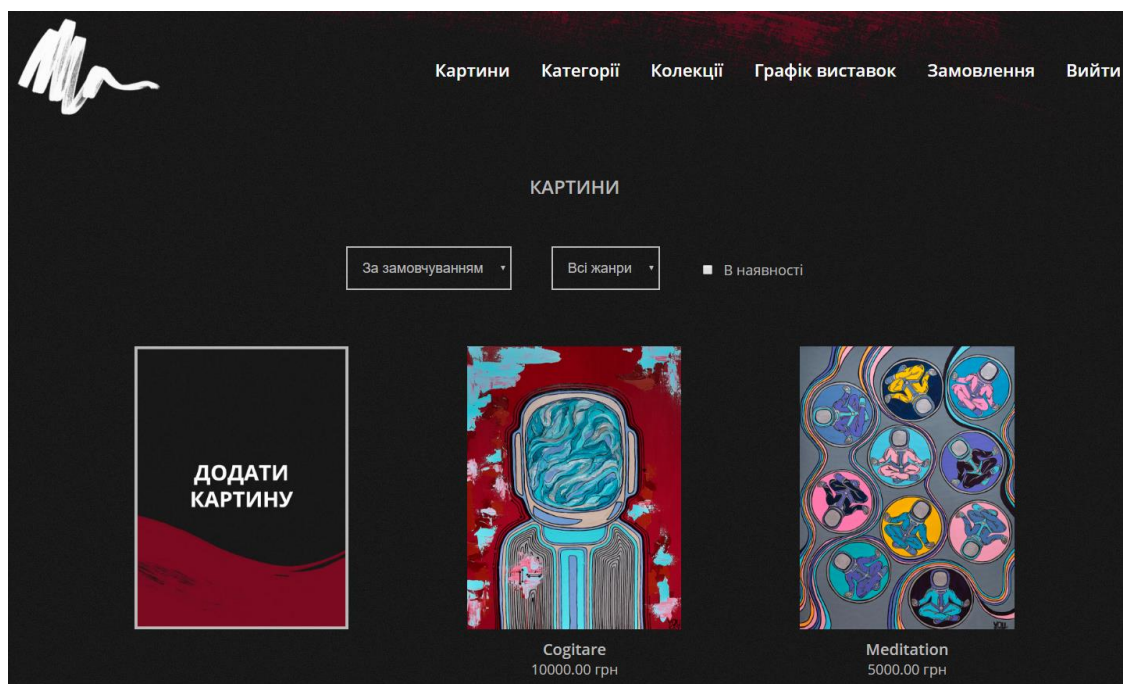



Рисунок 3.23 – Перелік картин

Після чого адміністратор перенаправляється на сторінку додавання картини (рис. 3.24), де він повинен заповнити форму з інформацією про картину та обрати її зображення. Якщо роздільна здатність зображення менша, ніж 500 пікселів по ширині або висоті, тоді адміністратору видається повідомлення, що у картинки занадто мала роздільну здатність (рис. 3.25). Якщо дані введені коректно й роздільна здатність картини відповідає умові, тоді видається повідомлення, що картина додана (рис. 3.26).

Додавання картини

Зображення:



Введіть назву картини:

Clo Sieurin

Введіть характеристики:

Акрил, Чорнила, 40x30 см, 2020р.

На продаж

Введіть ціну:

2000

Виберіть колекцію:

Novum vision

Виберіть категорію:

Поп-арт

Додати картину

Рисунок 3.24 – Додавання картини

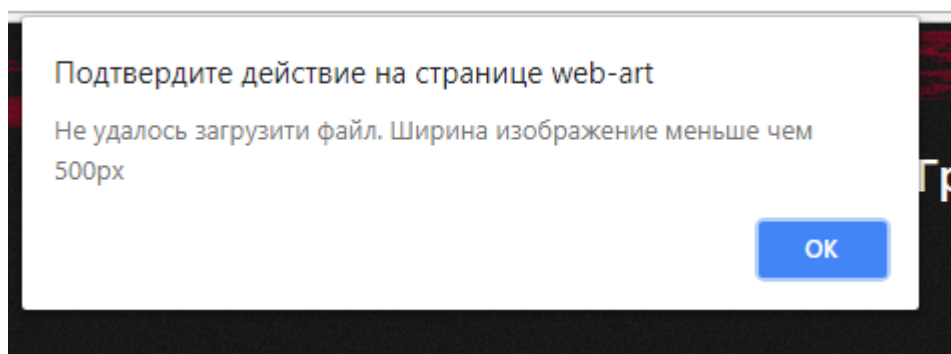


Рисунок 3.25 – Повідомлення, що роздільна здатність картини надто мала

Подтвердите действие на странице web-art

Картина Додана!

OK

Рисунок 3.26 – Повідомлення, що картина додана

Для перевірки замовлення треба натиснути на відповідний пункт в навігаційному меню. Перед адміністратором відкриється сторінка зі списком замовлень (рис. 3.27). та короткими відомостями про них.

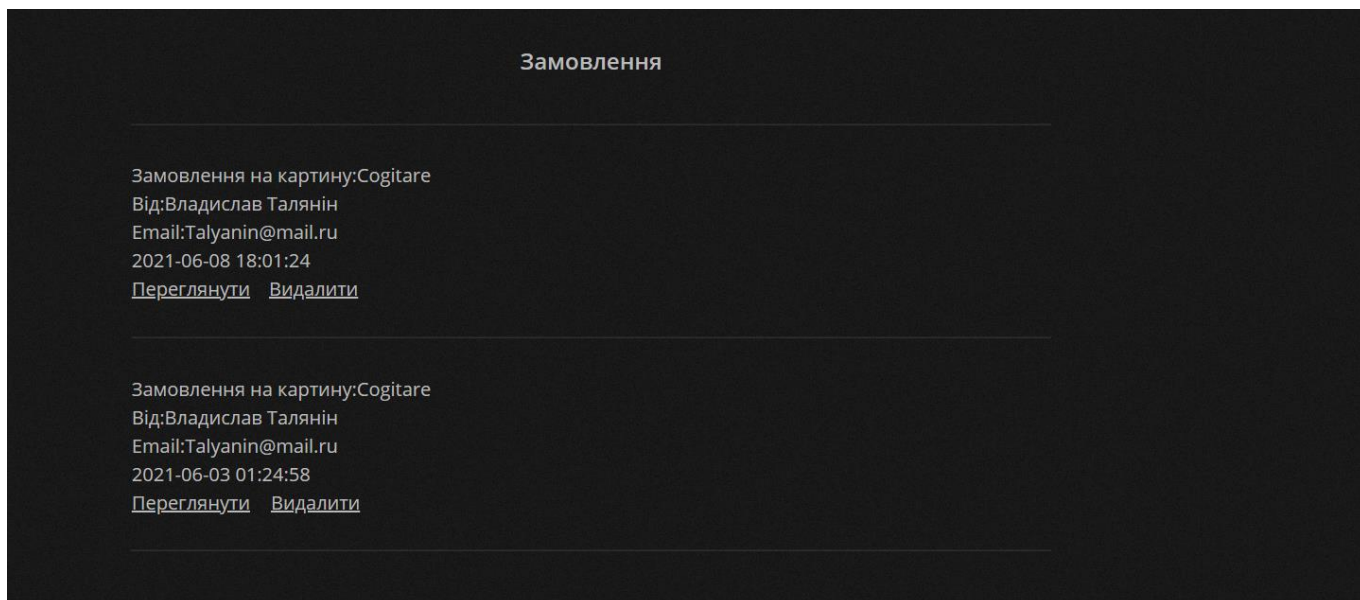


Рисунок 3.27 – Список замовлень

Для детального перегляду інформації про замовлення треба натиснути на пункт переглянути під замовленням. Сторінку з деталями замовлення зображено на рисунку 3.28.

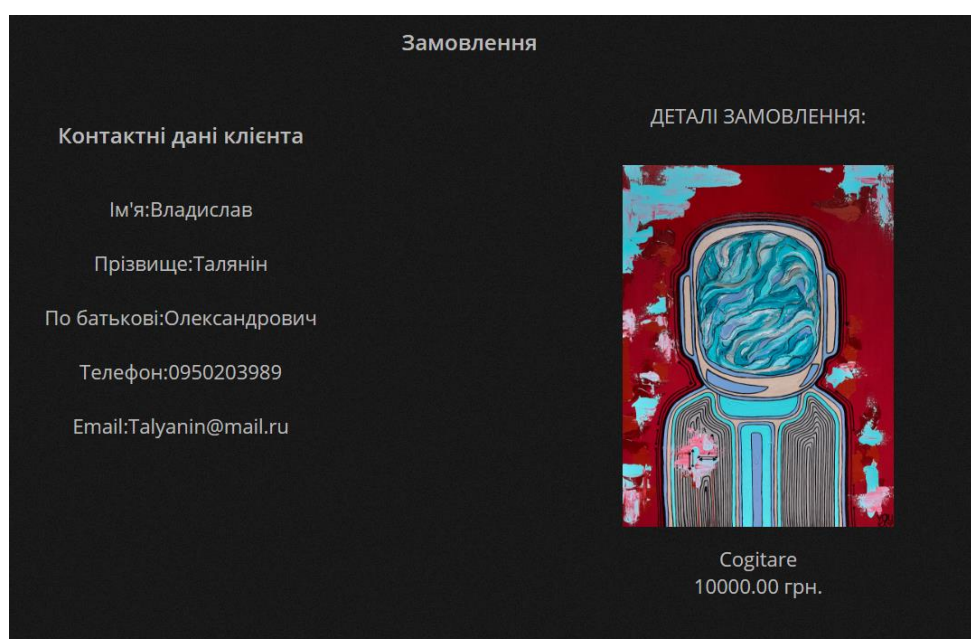


Рисунок 3.28 – Відображення замовлення

### 3.4 Тестування web-додатку

По завершенню розробки проекту було проведено тестування web-додатку. У таблиці 3.1 наведено його результати.

Таблиця 3.1 – Результати працездатності web-додатку

TC_ID	TEST SCENARIO	ACTION	ER	Статус
TC1.1	Навігаційне меню	Натиснення на посилання Галерея, Графік виставок, Про мене.	Перехід по відповідним посиланням	+
TC1.2	Замовлення картини	Заповнення форми замовлення та натиснення на кнопку відправити.	Замовлення залишено	+
TC1.3	Фільтри пошуку	Вибір різних параметрів фільтрів пошуку	Картини відсортовані за параметрами пошукових фільтрів	+
TC1.4	Авторизація	Заповнення форми авторизації та натиснення на кнопку ввійти	Перехід до адміністративної панелі	+
TC1.5	Додавання картин до галереї	Заповнення інформації про картину та завантаження файлу зображення	Картину додано до галереї	+

Після проведеного успішного тестування жодних невідповідностей у роботі web-додатка не виявлено.

## ВИСНОВКИ

У ході виконання дипломного проекту було виконано дослідження предметної області, аналіз аналогів для огляду основних недоліків та переваг програмних продуктів-аналогів, визначено мету та задачі проекту, сформовано список вимог до дизайну та функціоналу web-додатку підтримки діяльності виставкової діяльності художника й обрано засоби для його реалізації, виконано моделювання його роботи за допомогою IDEF та UML діаграм, а також спроектовано фізичну модель бази даних.

До того ж було виконано практичну реалізацію web-додатку підтримки виставкової діяльності художника. Його застосування надає змогу художнику виставляти картини для загального доступу публіки, приймати замовлення через форму зворотного зв'язку та повідомляти зацікавлених людей про заплановані виставки. Також використання даного програмного продукту дозволить збільшити аудиторію митця наряду з кількістю замовлень за рахунок зручного формату демонстрації власних картин.

Також було виконано планування робіт даного проекту (додаток Б) та здійснено апробацію результатів дослідження (додаток В). Лістинг основних модулів web-додатку надано в додатку Г.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Art explained: How the internet changed the art world [Електронний ресурс] – Режим доступу до ресурсу: <https://edition.cnn.com/style/article/how-the-internet-changed-art-steven-murphy/index.html>.
2. How much has the internet changed the art world? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bbc.com/culture/article/20140326-how-has-the-internet-changed-art>.
3. Web application to support the artist's exhibition activities. // Матеріали та програма МІЖНАРОДНОЇ НАУКОВО-ТЕХНІЧНОЇ КОНФЕНЦІЇ студентів та молодих учених / – Суми: МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ МІЖНАРОДНИЙ УНІВЕРСИТЕТ «АСТАНА» 2021. – С. 89.
4. CAUCE ZAVAGLIA [Електронний ресурс] – Режим доступу до ресурсу: <https://www.caucezavaglia.com/>.
5. MIDORI KATO contemporary artist in Tokyo Japan [Електронний ресурс] – Режим доступу до ресурсу: <https://www.midorikato.net/>
6. Shantell Martin [Електронний ресурс] – Режим доступу до ресурсу: <https://shantellmartin.art/>
7. Що таке HTML? Основи мови розмітки гіпертексту [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hostinger.ru/rukovodstva/shto-takoe-html/>
8. Що таке CSS: Пояснення простими словами [Електронний ресурс] – Режим доступу до ресурсу: <https://gb.ru/posts/chto-takoe-css-obyasnyаем-prostymi-slovami>.
9. Мова програмування JavaScript [Електронний ресурс] – Режим доступу до ресурсу: [https://flexberry.github.io/ru/gbt\\_javascript.html](https://flexberry.github.io/ru/gbt_javascript.html).
10. Що таке PHP - Web-Proger [Електронний ресурс] – Режим доступу до ресурсу: <http://web.spt42.ru/index.php/chto-takoe-php>.

11. Що таке MYSQL? [Електронний ресурс] – Режим доступу до ресурсу: <https://freehost.com.ua/faq/wiki/что-такое-mysql/>
12. Методология IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: [https://studme.org/87184/ekonomika/metodologiya\\_idef0](https://studme.org/87184/ekonomika/metodologiya_idef0).
13. Буч, Грейді Мова UML. Керівництво користувача / Грейді Буч, Джеймс Рамбо, Айвар Джекобсон. - М .: ДМК, 2015. - 432 с
14. Мюллер, Р.Дж. Бази даних та UML. Проектування / Р.Дж. Мюллер. - М .: ЛОРИ, 2017. - 420 с.
15. ГЛАВА 4 Диаграмма вариантов использования (use case diagram) [Електронний ресурс] – Режим доступу до ресурсу: <http://khpriip.mipk.kharkiv.edu/library/case/leon/gl4/gl4.html#1>.
16. Що таке WBS проекту, і навіщо вона потрібна [Електронний ресурс] - Режим доступу до ресурсу: <https://upravlenie-proektami.ru/что-такое-wbs-проекта-i-zachem-ona-nuzhna>
17. Организационная структура (Organizational Breakdown Structure, OBS) [Електронний ресурс] – Режим доступу до ресурсу: <https://pmpractice.ru/knowledgebase/gloss/detail/613.php>.
18. Діаграма Ганта - інструмент для тих, хто не любить зривати терміни [Електронний ресурс] - Режим доступу до ресурсу: <https://lifelhacker.ru/diagramma-ganta/>.



## **ДОДАТОК А**

### **ТЕХНІЧНЕ ЗАВДАННЯ**

**на створення програмного продукту  
«Web-додаток підтримки виставкової діяльності художника»**

## **1. Призначення й мета створення web-додатку**

### **1.1 Призначення web-додатку**

Web-додаток має надавати повну інформацію користувачам про діяльність художника, демонструвати роботи митця та спростити процедуру оформлення замовлення на купівлю картин.

### **1.2 Мета створення web-додатку**

Розробка web-додатку підтримки виставкової діяльності художника, застосування якого дозволить привернути увагу до діяльності митця та надасть можливість збільшення замовлення на продаж картин.

### **1.3 Цільова аудиторія**

До цільової аудиторії web-додатку можна віднести практично всіх людей, які зацікавлені в мистецтві, а саме поцінувачі образотворчого мистецтва.

## **2 Вимоги до web-додатку**

### **2.1 Вимоги до web-додатку в цілому**

#### **2.1.1 Вимоги до структури й функціонування web-додатку**

Web-додаток повинен завантажений на безкоштовний хостинг та бути доступний в мережі Інтернет.

#### **2.1.2 Вимоги до персоналу**

Від персоналу не має вимагатися особливих технічних навичок для підтримки й експлуатації web-додатку, окрім загальних навичок роботи з персональним

комп'ютером і стандартним веб-браузером, а також потрібно бути ознайомленим з інтерфейсом адміністративної панелі web-додатку.

### **2.1.3 Вимоги до збереження інформації**

Уся інформація надана у web-додатку буде зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

### **2.1.4 Вимоги до розмежування доступу**

Розроблюваний web-додаток має бути загальнодоступним.

Відповідно до прав доступу до інформації у web-додатку, усіх користувачів можна поділити на відвідувачів та адміністратора.

Відвідувачі можуть переглядати усі сторінки web-додатку, ознайомитись з роботами художника, графіком виставок та замовити картину.

Адміністратор може редагувати наповнення галереї та приймати замовлення залишені користувачами.

Доступ до адміністраторської панелі здійснюється за допомогою унікального логіна й пароля.

## **2.2 Структура web-додатку**

### **2.2.1 Загальна інформація про структуру web-додатку**

Структура web-додатку представлена набором сторінок, які також є пунктами головного меню.

Такими розділами є наступні:

- Галерея – на сторінці представленні роботи художника, деякі із них виставлені на продаж;
- Графік виставок – інформація про заплановані виставки;

– Про мене – на сторінці знаходиться основна інформація про діяльність художника та контакти для зв'язку.

### **2.2.2 Навігація**

Для навігації по web-додатку у шапці буде створена система контент меню. Меню необхідне для швидкого переміщення користувача по усім доступним сторінкам. Меню буде відображатися на всіх сторінках, щоб відвідувач мав змогу в будь-який момент часу перейти на будь-яку сторінку web-додатку.

### **2.2.3 Наповнення web-додатку (контент)**

Для управління контентом web-додатку буде використана панель адміністратора.

Заповнення та редагування контенту web-додатку має бути зроблено через панель адміністратора, використовуючи інформацію з бази даних.

Усю інформацію для наповнення web-додатку має надавати художник, включаючи картини та контактну інформацію.

### **2.2.4 Дизайн та структура додатку**

Стиль web-додатку має бути мінімалістичний, приємним для сприйняття, у якості основних кольорів було запропоновано використати чорні та білі відтінки.

Основою мають бути картини, виконані художником, інтерфейс web-додатку має бути інтуїтивно зрозумілим для використання.

Розташування елементів на головній сторінці web-додатку схематично показано на рисунку А.1.

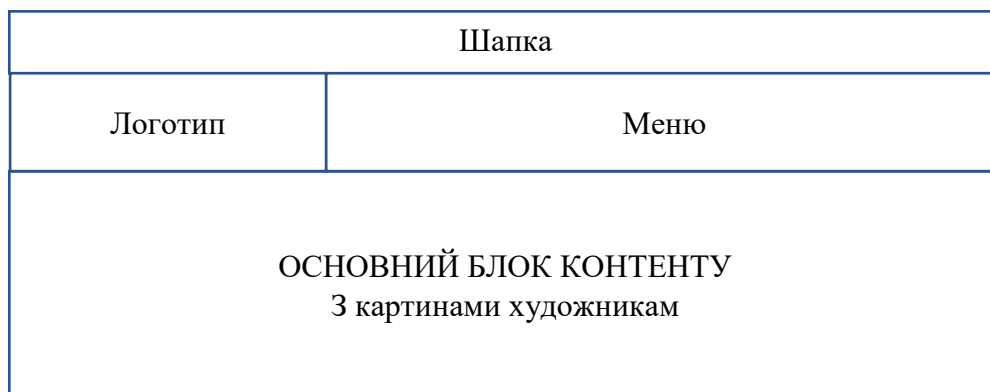


Рисунок А.1 – Схема галереї

### 2.2.5 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.

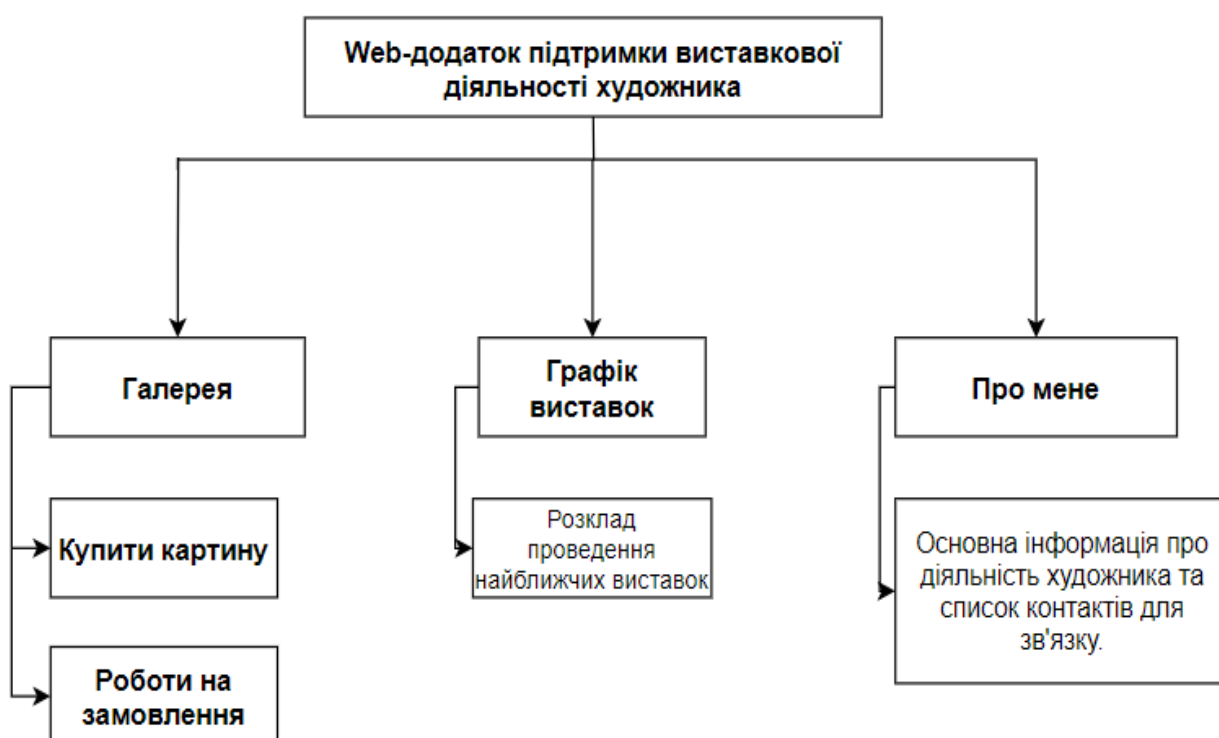


Рисунок А.2 – Карта web-додатку

## 2.3 Вимоги до функціонування системи

### 2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Перегляд робіт художника	Клієнт
UN-02	Перегляд картин виставлених на продаж	Клієнт
UN-03	Можливість замовити картину за власними вимогами	Клієнт
UN-04	Редагування даних	Адміністратор
UN-05	Перегляд графіку виставок	Клієнт

### 2.3.2 Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

- можливість перегляду галереї робіт;
- можливість перегляду графіку виставок;
- можливість перегляду картин виставлених на продаж;
- можливість замовити картину за допомогою онлайн-форми;
- адміністрування галереї картин, та графіку проведення виставок.

### 2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

<b>ID</b>	<b>Системні вимоги</b>	<b>Пріоритет</b>	<b>Опис</b>
SR-01	Каталог картин	М	Формує портфоліо художника
SR-02	База даних із картинами	М	Надає можливість відобразити картини в галереї
SR-03	База даних з графіком виставок	М	Надає можливість відобразити графік виставок
SR-04	Наявність модуля для замовлення картин	М	Надає можливість клієнту замовити картину
SR-05	Панель адміністратора	М	Відповідає за подальше наповнення та редагування контенту адміністратором

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є головною ціллю проекту.

## 2.4 Вимоги до видів забезпечення

### 2.4.1 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний українською мовою.

### 2.4.2 Вимоги до програмного забезпечення

Реалізація web-додатку відбувається з використанням таких технологій:

- PHP 7.4.4;
- MySQL 8.0.

Програмне забезпечення клієнтської частини повинне задовольняти наступній вимозі – веб-браузер Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

## 3 Склад і зміст робіт зі створення web-додатку

Докладний опис етапів роботи зі створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	1 день
2	Складання технічного завдання	2 дні
3	Підготовка прототипу	3 дні
4	Створення макету дизайну web-додатку	3 дні
5	Верстка	4 дні
6	Робота над модулями для web-додатку	3 дні
7	Робота з контентом	2 день
8	Тестування web-додатку	1 день



## Продовження таблиці А.3

9	Завершення роботи	1 день
	Загальна тривалість робіт	20 днів

**4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію**

Для того, щоб web-додатком могли користуватися клієнти та відвідувачі необхідно розмістити його у мережі Інтернет, тому необхідно придбати доменне ім'я та місце на хостингу. На хостинг переноситься web-додаток і наповнення бази даних з подальшою їх доробкою.

## ДОДАТОК Б

### Планування робіт

**Деталізація мети проекту методом SMART.** Продуктом дипломного проекту є web-додаток підтримки виставкової діяльності художника.

Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 – деталізація методом SMART

Specific (конкретна)	Ціллю проекту є розробка web-додатка підтримки виставкової діяльності художника.
Measurable (вимірювана)	Даний проект являється дипломним, тому показником виконання проекту є оцінка дипломної комісії.
Achievable (досяжність)	Ціль є досяжною оскільки розробник має необхідні уміння для виконання проекту.
Realistic (реалістичність)	Для досягнення цілей проекту наявні всі необхідні ресурси та знання.
Time-framed (обмеженість в часі)	Проект повинен бути виконаний до 27.05.2021.

**Планування змісту структури робіт ІТ-проекту.** Для оптимальної організації роботи при виконанні проекту була створена WBS діаграма. WBS проекту (Work Breakdown Structur) – це розкладання проекту на етапи, які повинні бути виконані для завершення проекту [16].

WBS діаграма представлена на рисунку Б.1.

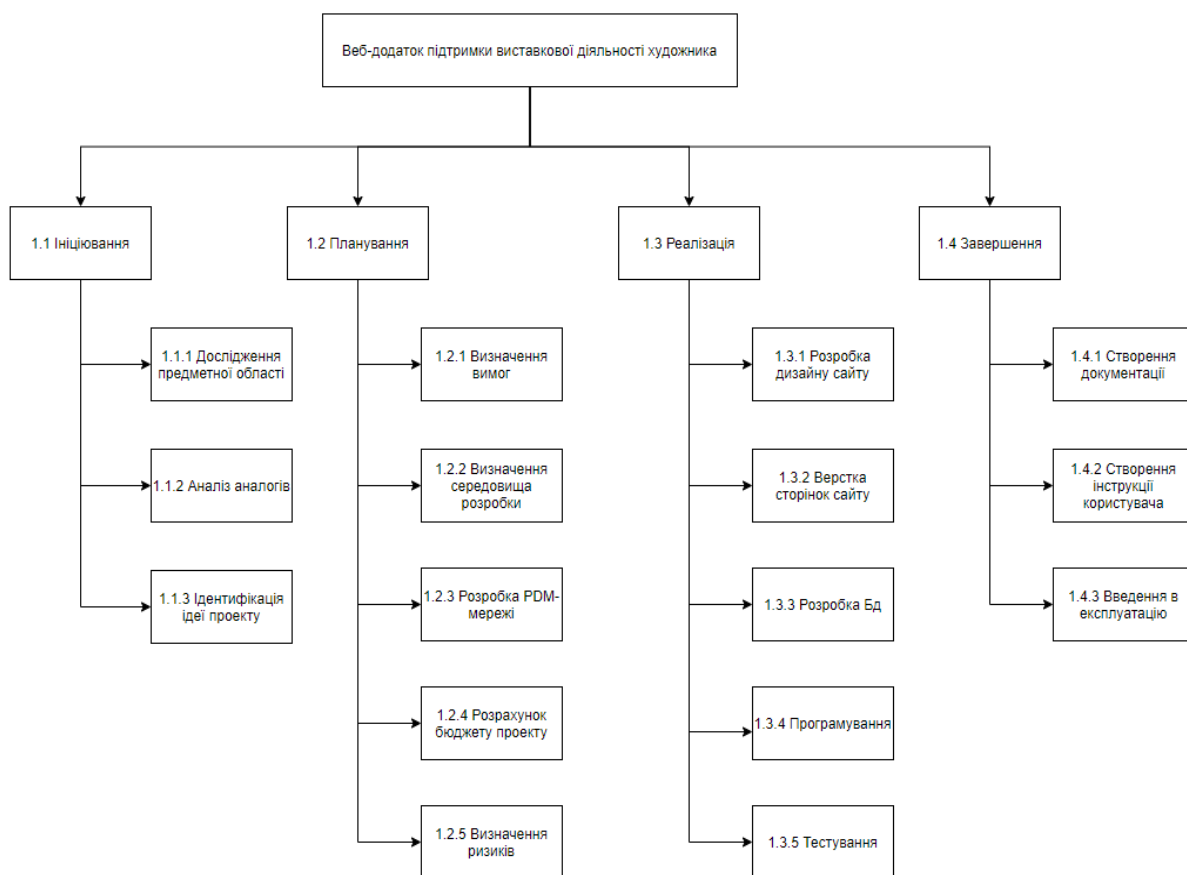


Рисунок Б.1 – WBS проекту

**Планування структури організації, для впровадження готового проекту.** На наступному етапі планування проекту була створена OBS Діаграма. OBS структура зображує ієрархічну структуру проекту (рис. Б.2), яка надає можливість зіставляти етапи проекту з виконавцями [17]. Учасниками розробки представлені у таблиці Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Талянін В.О.	Виконує розробку функціоналу та інтерфейсу проекту
Куратор	Антипенко В.П.	Формує завдання на розробку проекту.
Тестувальник	Рудик В.В.	Відповідає за тестування функціоналу та дизайну додатку.

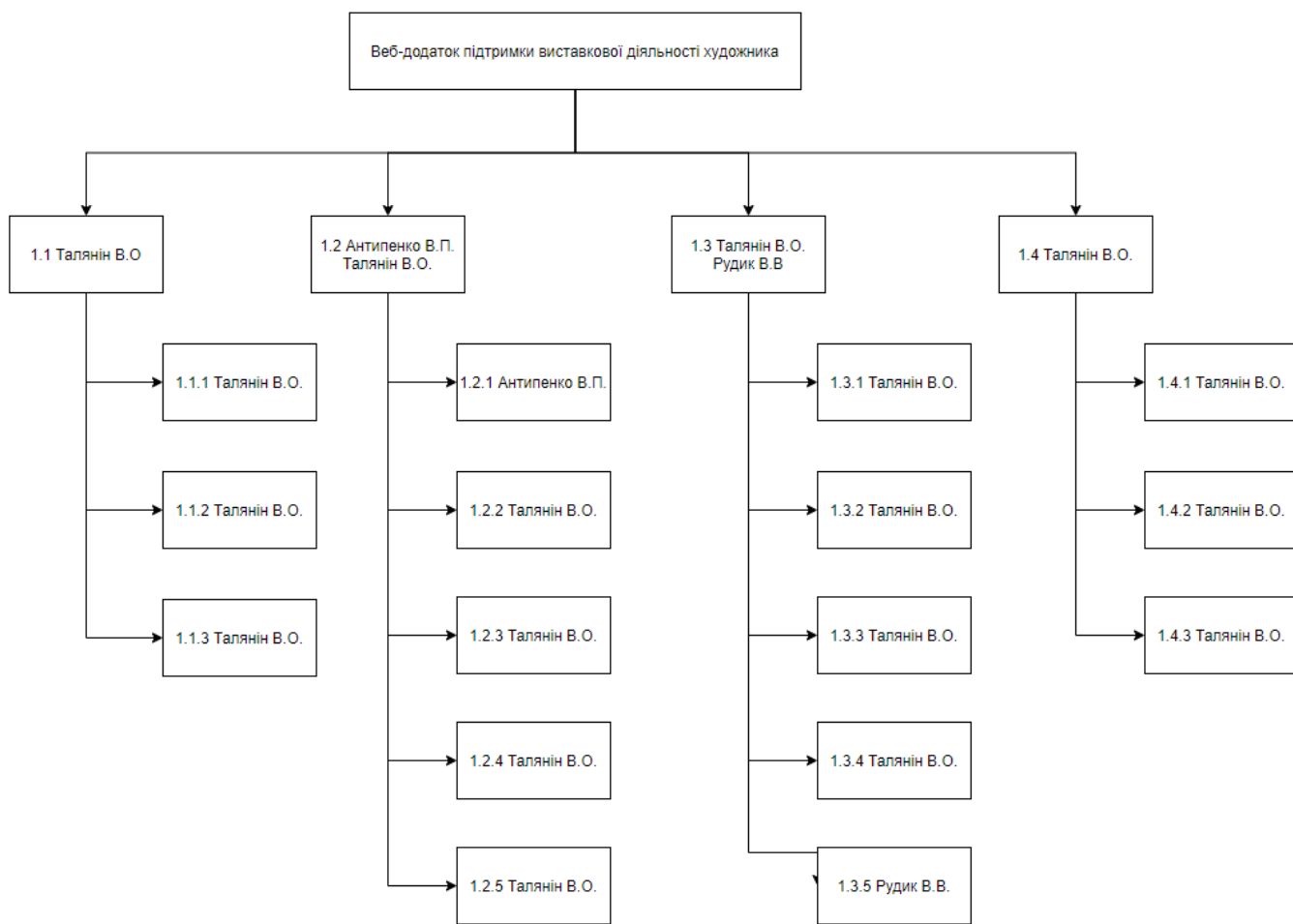


Рисунок Б. 2 – OBS проекту

**Побудова матриці відповідальності.** Використовуючи OBS та WBS структури, була побудована матриця відповідальності проекту. Матриця відображає вид роботи, за яку відповідає кожен учасник проекту. Дана матриця відображена в таблиці Б.3.

Таблиця Б.3 – Матриця відповідальності

WBS \ OBS		Антипенко В.П.	Телянін В.О.	Рудик В.В.
1.1	1.1.1		X	
	1.1.2		X	
	1.1.3		X	
1.2	1.2.1	X		
	1.2.2		X	
	1.2.3		X	
	1.2.4		X	
	1.2.5		X	

## Продовження таблиці Б.3

1.3	1.3.1		X	
	1.3.2		X	
	1.3.3		X	
	1.3.4		X	
	1.3.5			X
1.4	1.4.1		X	
	1.4.2		X	
	1.4.3		X	

**Розробка PDM-мережі.** Для наглядного представлення послідовності виконання етапи розробки проекту була створена PDM-мережа (рис. Б.4-Б.6).

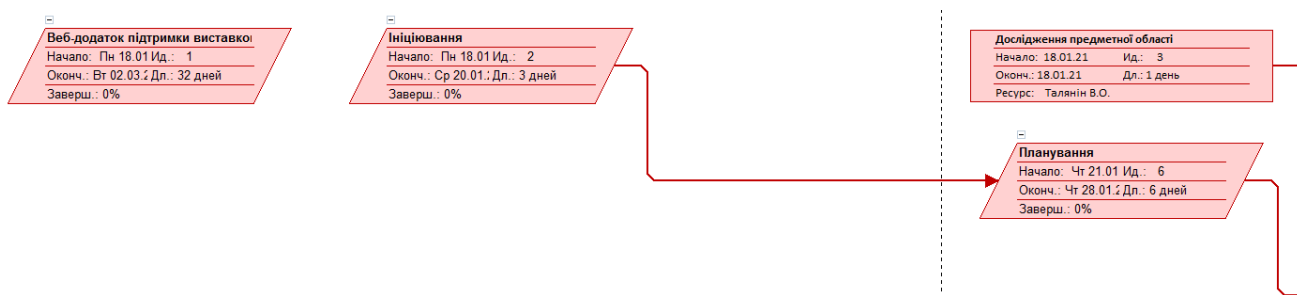


Рисунок Б.4 – Перша частина PDM-мережі

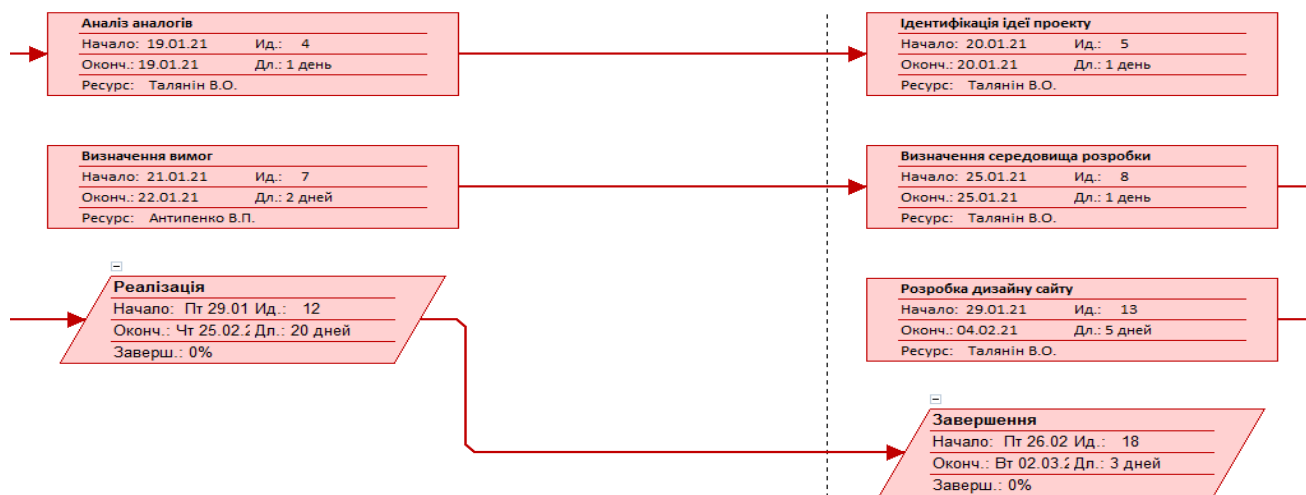


Рисунок Б.5 – Друга частина PDM-мережі

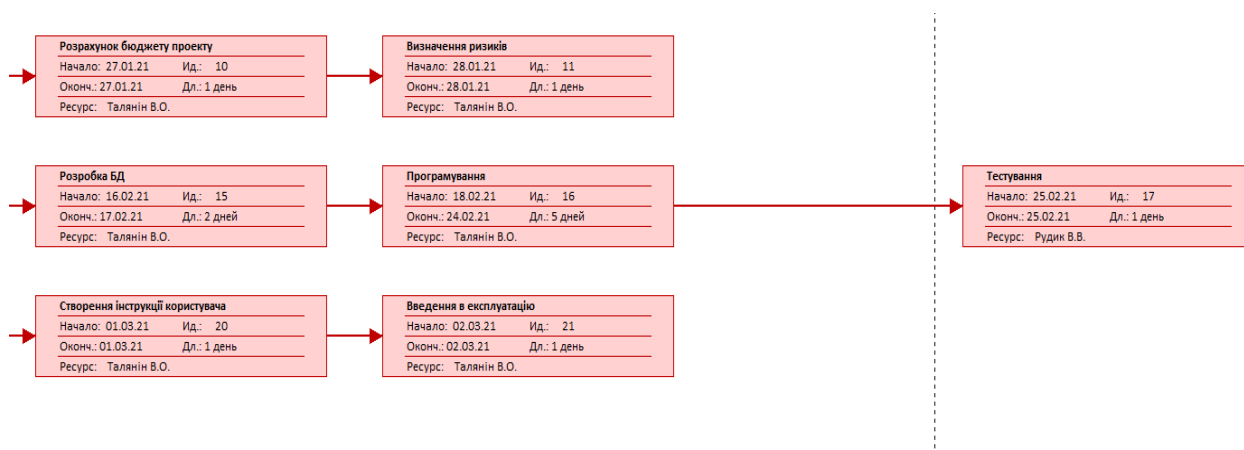


Рисунок Б.6 – Третя частина PDM-мережі

**Побудова календарного графіку створення програмного продукту проекту.** У ході виконання планування робіт за допомогою додатка MS Project було створено календарний графік виконання проекту (рис. Б.7). Також для наочної демонстрації строків виконання стадій розробки проекту створено діаграму Ганта (рис. Б.8) – це діаграма з шкалою часу, яка використовується для показу запланованих стадій проекту з часовою прив'язкою [18]. Це практичний спосіб зобразити, яка робота планується до здійснення в конкретний момент.

▲ <b>Веб-додаток підтримки виставкової діяльності художника</b>	<b>32 днів</b>		<b>Пн 18.01.21</b>	<b>Вт 02.03.21</b>	
▲ <b>Ініціювання</b>	<b>3 днів</b>		<b>Пн 18.01.21</b>	<b>Ср 20.01.21</b>	
Дослідження предметної області	1 день	Талянін В.О.	Пн 18.01.21	Пн 18.01.21	
Аналіз аналогів	1 день	Талянін В.О.	Вт 19.01.21	Вт 19.01.21	3
Ідентифікація ідей проекту	1 день	Талянін В.О.	Ср 20.01.21	Ср 20.01.21	4
▲ <b>Планування</b>	<b>6 днів</b>		<b>Чт 21.01.21</b>	<b>Чт 28.01.21</b>	<b>2</b>
Визначення вимог	2 днів	Антипенко В.П.	Чт 21.01.21	Пт 22.01.21	
Визначення середовища розробки	1 день	Талянін В.О.	Пн 25.01.21	Пн 25.01.21	7
Розробка PDM-мережі	1 день	Талянін В.О.	Вт 26.01.21	Вт 26.01.21	8
Розрахунок бюджету проекту	1 день	Талянін В.О.	Ср 27.01.21	Ср 27.01.21	9
Визначення ризиків	1 день	Талянін В.О.	Чт 28.01.21	Чт 28.01.21	10
▲ <b>Реалізація</b>	<b>20 днів</b>		<b>Пт 29.01.21</b>	<b>Чт 25.02.21</b>	<b>6</b>
Розробка дизайну сайту	5 днів	Талянін В.О.	Пт 29.01.21	Чт 04.02.21	
Верстка сторінок сайту	7 днів	Талянін В.О.	Пт 05.02.21	Пн 15.02.21	13
Розробка БД	2 днів	Талянін В.О.	Вт 16.02.21	Ср 17.02.21	14
Програмування	5 днів	Талянін В.О.	Чт 18.02.21	Ср 24.02.21	15
Тестування	1 день	Рудик В.В.	Чт 25.02.21	Чт 25.02.21	16
▲ <b>Завершення</b>	<b>3 днів</b>		<b>Пт 26.02.21</b>	<b>Вт 02.03.21</b>	<b>12</b>
Створення документації	1 день	Талянін В.О.	Пт 26.02.21	Пт 26.02.21	
Створення інструкції користувача	1 день	Талянін В.О.	Пн 01.03.21	Пн 01.03.21	19
Введення в експлуатацію	1 день	Талянін В.О.	Вт 02.03.21	Вт 02.03.21	20

Рисунок Б.7 – Календарний графік виконання проекту

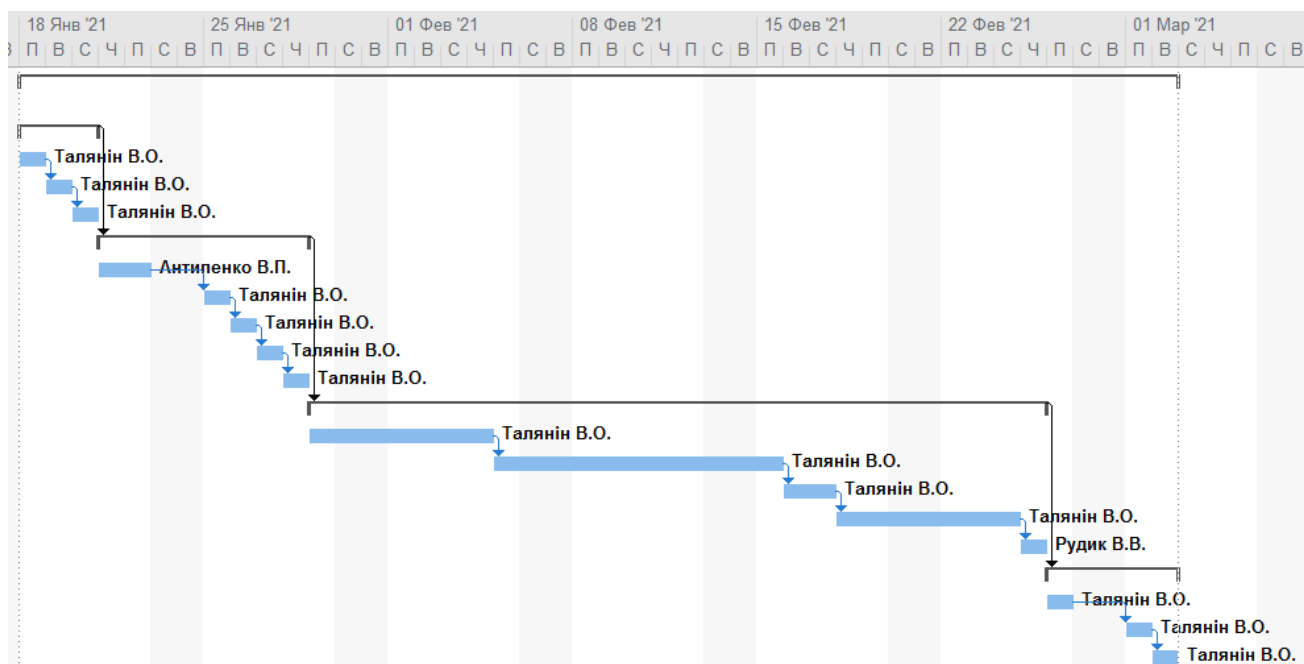


Рисунок Б.8 – Діаграма Ганта

**Аналіз ризиків.** Для адаптації робочого після форс-мажорних обставин або їх уникнення був проведений аналіз ризиків.

Таблиця Б.4 – Визначення ймовірності та впливу ризиків проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

На основі даної таблиці була побудована матриця ризиків (рис Б.9).

Ймовірність	3	4	5	6
	2	3	4	5
	1	2	3	4
		1	2	3
	Вплив			

Рисунок Б.9 – матриця ризиків даного проекту

Класифікація ризиків є такою:

- зелений колір – допустимі ризики;
- жовтий колір – виправданні ризики;
- червоний колір – критичні ризики.

Із допомогою даної інформації була створена таблиця класифікацій та реакцій на ризики, що відображена в таблиці Б.5.

Таблиця Б.5 – Класифікація та реакція на ризики

Опис ризику	Ймовірність	Вплив	Ранг	Реакція
Хвороба розробника	2	3	5	Виділити резервний час на випадок хвороби розробника
Пошкодження файлів	2	2	4	Робити бекуп файлів після кожного сеансу роботи
Часте внесення змін у ТЗ	1	2	3	Точно описати вимоги до проекту та виділити головні параметри проекту.
Некоректне тестування	2	2	4	Провести додаткове тестування залучивши до нього додаткових тестувальників
Вихід з ладу апаратного забезпечення	1	3	4	Провести ремонт основного апаратного забезпечення. На час ремонту знайти заміну апаратному забезпеченню
Недотримання графіку робіт	2	3	5	Зосередитися на створенні основного функціоналу додатку
Вибір неефективної технології розробки	2	3	5	Вибрати легку в освоєнні технологію розробки.



## Продовження таблиці Б.5

Збої в роботі програмного забезпечення	1	2	3	Переустановлення програми або інсталяція аналогічно за функціоналом програмного забезпечення
--	---	---	---	--

## ДОДАТОК В

### АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

#### **Web application to support the artist's exhibition activities**

V. Talyanin, *Student*; V. Antypenko, *Associate Professor*  
Sumy State University, Sumy, Ukraine

The development of Internet technologies has affected all spheres of human activity and the field of fine arts is no the exception. Social networks, in particular Instagram, provide an opportunity for artists to easily share their own works with the public and receive feedback from the audience. But having only a profile on a social network is no longer enough to stand out from other artists. Nowadays, it is advisable to have your own portfolio site in order to get noticed. A web portfolio is not just a collection of paintings, it is also a tool for business and marketing. A portfolio helps artists to find new clients, get noticed by agencies and gain followers to bring your views through the works to masses.

Thus, the goal of this project is to develop a web application to support the artist's exhibition activities. The application allows the artist to exhibit paintings at the public domain, take orders through the feedback form and inform interested parties about the planned exhibitions. The design of the web application has been made in a minimalist style which does not distract attention from the paintings, but at the same time emphasizes the artist's author style. The application consists of two parts – client and administrative. The administrative panel makes it possible to add and remove the images of paintings, create thematic collections of works, change the schedule of exhibitions and view orders added by users. In turn, the client part allows users to view the galleries of the artist's works along with the schedule of exhibitions. Moreover, clients can make a request to order available paintings.

To develop this web application HTML and CSS were chosen for site layout creation, JavaScript for providing more visual appeal, PHP was selected as the server-side programming language and MySQL for working with databases.

The result of this project is a developed web application to support the artist's exhibition activities in the form of a web portfolio. Its using will increase the artist's audience along with the number of orders due to a convenient online tool as well as a modern format of demonstrating and ordering his/her own paintings.

## ДОДАТОК Г

### ЛІСТИНГ ОСНОВНИХ МОДУЛІВ WEB-ДОДАТКУ

Файл «index.php»

```

<?php include ("incl/db.php");?>
<!DOCTYPE html>
<html lang="en">
<head><meta charset="UTF-8"><title>Художник и публицист - Александр Миллер</title>
    <link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,400;0,600;1,400;1,600&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="styles/main.css">
    <link rel="stylesheet" type="text/css" href="styles/header.css">
    <link rel="stylesheet" type="text/css" href="styles/content.css">
    <link rel="stylesheet" type="text/css" href="styles/media.css">
    <link rel="icon" href="favicon.ico" type="image/x-icon">
<script type="text/javascript" src="js/shtrih.js"></script></head><body>
<?php include ("block/header.php");?>
<main> <form style=" padding: 0;" method="GET" action="index.php">
    <div class="content column_center">
        <section class="gallery">
            <h1>ОРИГІНАЛЬНІ КАРТИНИ НА ПРОДАЖ</h1>

            <div class="filter">

                <?
                $sor= $_GET["select1"];
                $sorting = $_GET["select1"];

switch ($sor){
    case 'price-asc'; $sorting = 'price ASC'; break;
    case 'price-desc';$sorting = 'price DESC';
    break;case 'news';
    $sorting = 'id_art DESC';break;
    default:$sorting = 'id_art asc'; break;  }
$id_cat=$_GET["select2"];
if ((isset($_GET['select2']))&&($_GET['select2']!=0)) {$query_col = " AND id_cat=$id_cat";}else{$query_col ="";}
$avaible=$_GET["av"];if (isset($_GET["av"])) {$query_avaible = " AND avaible=$avaible";}else{$query_avaible ="";}
?><select name="select1" onchange='this.form.submit()'> <!--Supplement an id here instead of using 'name'-->
        <option value="def">За замовчуванням</option>
        <option value="news"<?php if ($sor== 'news')
{echo "selected='selected'";}?>>Новинки</option>
        <option value="price-asc" <?php if ($sor== 'price-asc')
{echo "selected='selected'";}?>>Спочатку дешеві</option>
        <option value="price-desc"<?php if ($sor== 'price-desc')
{echo "selected='selected'";}?>>Спочатку дорогі</option>
</select>
<select name="select2" onchange='this.form.submit()'>

<option value="0">Всі жанри</option>
<?
$rescol = mysqli_query($db,"SELECT * FROM categories");
If (mysqli_num_rows($rescol) > 0){
$mycol = mysqli_fetch_array($rescol);
do
{echo'<option value="'. $mycol['id_cat']. '";
if ($mycol['id_cat']==$id_cat){ echo "selected='selected'";}
echo'>;echo". $mycol['cname']. ";
echo'</option>';}while($mycol = mysqli_fetch_array($rescol));}?></select><span>

```

```



```

Файл «schedule.php»

```

<?php include ("incl/db.php");
$result = mysqli_query($db,"SELECT * FROM exhibitions order by startdate desc");

if (!$result)
{echo "<p>Виникла помилка <br> <strong>Код ошибки:</strong></p>";
exit(mysqli_error());}

if (mysqli_num_rows($result) > 0)
{
$myrow = mysqli_fetch_array($result);
}
else

```

```

{
echo "<p>Інформація по запиту не може бути вилучена в таблиці немає записів.</p>";
exit();
}
//if (!isset($id_art)) {}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Художник і публіцист - Александр Міллер</title>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,400;0,600;1,400;1,600&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="styles/main.css">
    <link rel="stylesheet" type="text/css" href="styles/header.css">
    <link rel="stylesheet" type="text/css" href="styles/content.css">
    <link rel="stylesheet" type="text/css" href="styles/media.css">

    <link rel="icon" href="favicon.ico" type="image/x-icon">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script type="text/javascript" src="/js/shtrih.js"></script>
<script src="js/custom.js"></script>
</head>
<body>
<?php include ("block/header.php");?>
    <main>
        <div class="content column_center">
            <section>
                <h1>НАЙБЛИЖЧІ ВИСТАВКИ І ПОДІЇ</h1><hr>

                <?

$date_m = array('січня', 'лютого', 'березня', 'квітня', 'травня', 'червня', 'липня', 'серпня', 'вересня', 'жовтня', 'листопада',
'грудня');

                do
                {
                    $timestart = strtotime($myrow['startdate']);

                    $mysqlstartdate = $date_m[date('n', $timestart) - 1];
                    $mysqlstartdate2 = date('j ', $timestart);

```

```

echo' <article class="event">
<time>' . $mysqlstartdate2 . ' ' . $mysqlstartdate . ";
if (isset($myrow['enddate'])) {
    $timeend = strtotime($myrow['enddate']);
    $mysqlstartdate = $date_m[date('n', $timeend) - 1];
    $mysqlstartdate2 = date('j', $timeend);
    echo' - ' . $mysqlstartdate2 . ' ' . $mysqlstartdate . ";}
echo'</time><br>';
echo'<span>' . $myrow['exname'] . '</span>';
if (isset($myrow['id_col'])) {$id_col=$myrow['id_col'];
$result2 = mysqli_query($db,"SELECT * FROM collection where
id_col=$id_col");

$myrow2 = mysqli_fetch_array($result2);
    echo'<br><span>Колекція: ' . $myrow2['colname'] . '<span>';}
    if (isset($myrow['location']))
{echo'<br><addre>' . $myrow['location'] . '</addre>';}
echo'</article><hr>';}
while ($myrow = mysqli_fetch_array($result)); ?>

    </div>
</main>
<footer>
    <div>
        <div class="socials row_center">
            <a href="https://www.instagram.com/"></a>
            <a href="https://www.pinterest.com/"></a>
            <a href="https://twitter.com/?lang=ru"></a>
        </div>
    </div>
</footer>
</body>
</html>

```

Файл «mail.php»

```

<?php
include("../incl/db.php");
$mail=$_POST['mail'];
$fname=$_POST['fname'];
$lname=$_POST['lname'];
$tname=$_POST['tname'];
$phone=$_POST['phone'];
$message=$_POST['message'];
$id_art=$_POST['id_art'];
$id=1;

```

```

$ip=$_POST['ip'];

$client = mysqli_query($db,"SELECT * FROM clients WHERE fname='$fname' and lname='$lname' and fathname='$fathname'
and phone='$phone' and email='$mail'");
If (mysqli_num_rows($client) > 0){
    $cli = mysqli_fetch_array($client);
    $sid=$cli['id_client'];
    if($ip!=$cli['ip']){$clientupdate = mysqli_query ($db,"UPDATE clients SET ip='$ip' WHERE id_client='$sid'");}
}
else{
    $newclient = mysqli_query ($db,"INSERT INTO clients (fname,lname,fathname,email,phone,ip) VALUES ('$fname',
'$lname','$fathname','$mail','$phone','$ip')");
    $client2=mysqli_query($db,"SELECT * FROM clients WHERE fname='$fname' and lname='$lname' and
fathname='$fathname' and phone='$phone' and email='$mail'");
    $cli = mysqli_fetch_array($client2);
    $sid=$cli['id_client'];
}
$result = mysqli_query ($db,"INSERT INTO message (id_art,id_client,description,date) VALUES ('$sid_art',
'$sid','$message',NOW())");
if ($result == 'true') {echo "Заявка на замовлення успішно оформлена!";}
else {echo "Заявка на замовлення не була оформлена!";}
?>

```

Файл «upload.php»

```

<?php
setlocale(LC_ALL, 'ru_RU.utf8');
date_default_timezone_set('Europe/Moscow');
mb_internal_encoding('UTF-8');
mb_regex_encoding('UTF-8');
mb_http_output('UTF-8');
mb_language('uni');

$input_name = 'file';

if (!isset($_FILES[$input_name])) {
    exit;
}

$allow = array('jpg', 'jpeg', 'png', 'gif');

$url_path = '/art/tmp/';

$tmp_path = $_SERVER['DOCUMENT_ROOT'] . $url_path;

if (!is_dir($tmp_path)) {
    mkdir($tmp_path, 0777, true);
}

$files = array();
$difff = count($_FILES[$input_name]) - count($_FILES[$input_name], COUNT_RECURSIVE);
if ($difff == 0) {
    $files = array($_FILES[$input_name]);
} else {
    foreach($_FILES[$input_name] as $k => $l) {
        foreach($l as $i => $v) {
            $files[$i][$k] = $v;
        }
    }
}

$response = array();
foreach ($files as $file) {

```

```

$error = $data = "";

$ext = mb_strtolower(mb_substr(mb_strchr(@$file['name'], '.'), 1));
if (!empty($file['error']) || empty($file['tmp_name']) || $file['tmp_name'] == 'none') {
    $error = 'Не удалось загрузить файл.';
} elseif (empty($file['name']) || !is_uploaded_file($file['tmp_name'])) {
    $error = 'Не удалось загрузить файл.';
} elseif (empty($ext) || !in_array($ext, $allow)) {
    $error = 'Недопустимый тип файла';
} else {
    $info = @getimagesize($file['tmp_name']);
    if (empty($info[0]) || empty($info[1]) || !in_array($info[2], array(1, 2, 3))) {
        $error = 'Недопустимый тип файла';
    } else {

        $name = time() . '-' . mt_rand(1, 999999999);
        $src = $tmp_path . $name . '.' . $ext;
        $thumb = $tmp_path . $name . '-thumb.' . $ext;

        if (move_uploaded_file($file['tmp_name'], $src)) {
            switch ($info[2]) {
                case 1:
                    $im = imageCreateFromGif($src);
                    imageSaveAlpha($im, true);
                    break;
                case 2:
                    $im = imageCreateFromJpeg($src);
                    break;
                case 3:
                    $im = imageCreateFromPng($src);
                    imageSaveAlpha($im, true);
                    break;
            }

            $width = $info[0];
            $height = $info[1];
            if($width<500){$error = 'Не удалось загрузити файл. Ширина изображение меньше чем 500px';}
            if($height<500){$error = 'Не удалось загрузити файл. Ширина изображение меньше чем 500px';}
            $h = $info[0];
            $w = ($h > $height) ? $width : ceil($h / ($height / $width));
            $tw = ceil($h / ($height / $width));
            $th = ceil($w / ($width / $height));

            $new_im = imageCreateTrueColor($w, $h);
            if ($info[2] == 1 || $info[2] == 3) {
                imagealphablending($new_im, true);
                imageSaveAlpha($new_im, true);
                $transparent = imagecolorallocatealpha($new_im, 0, 0, 0, 127);
                imagefill($new_im, 0, 0, $transparent);
                imagecolortransparent($new_im, $transparent);
            }

            if ($w >= $width && $h >= $height) {
                $xy = array(ceil(($w - $width) / 2), ceil(($h - $height) / 2), $width, $height);
            } elseif ($w >= $width) {
                $xy = array(ceil(($w - $tw) / 2), 0, ceil($h / ($height / $width)), $h);
            } elseif ($h >= $height) {
                $xy = array(0, ceil(($h - $th) / 2), $w, ceil($w / ($width / $height)));
            } elseif ($tw < $w) {
                $xy = array(ceil(($w - $tw) / 2), ceil(($h - $h) / 2), $tw, $h);
            } else {
                $xy = array(0, ceil(($h - $th) / 2), $w, $th);
            }
        }
    }
}

```



```

}
$w =$info[0];
$imageCopyResampled($new_im, $im, $xy[0], $xy[1], 0, 0, $xy[2], $xy[3], $width,
$height);

switch ($info[2]) {
    case 1: imageGif($new_im, $thumb); break;
    case 2: imageJpeg($new_im, $thumb, 100); break;
    case 3: imagePng($new_im, $thumb); break;
}

imagedestroy($im);
imagedestroy($new_im);

$data = '
<div class="img-item">
    <img id src="" . $url_path . $name . '.' . $ext . "'>
    <a href="#" onclick="remove_img(this); return false;"></a>
    <input type="hidden" id="aimge" name="aimge" value="" . $url_path . $name . '.'
. $ext . "'>

</div>;' } else {
    $error = 'Не удалось загрузить файл.';    }}
$response[] = array('error' => $error, 'data' => $data); } header('Content-Type: application/json');
echo json_encode($response, JSON_UNESCAPED_UNICODE);exit();

```