

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

**на тему: Розробка вебплатформи для експозиції та реалізації творів
мистецтва «GalleryHub»**

Виконав: студент IV курсу, групи КН-423
спеціальності: 122 Комп'ютерні науки

Андрій СТУКАЛО

Керівник Руслан СЛОБОДЯН

Рецензент _____
(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Стукалу Андрію Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебплатформи для експозиції та реалізації творів мистецтва «GalleryHub».

керівник роботи _____ Слободян Руслан Олесійович _____,
(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мова програмування TypeScript; фреймворк Express.js, бібліотеки: React, React Router, стандарти ДСТУ ISO/IEC/IEEE 29148:2025, ДСТУ ISO/IEC/IEEE 29119-1:2017, ДСТУ 3008:2015, ДСТУ 8302:2015.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу вебплатформи

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

2 Розробка технічного та робочого проєкту

2.1 Розробка структури вебплатформи

2.2 Створення та верстка сторінок вебплатформи

2.3 Розробка структури бази даних вебплатформи

2.4 Програмування вебплатформи

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

2.5 Тестування вебплатформи

3 Спеціальний розділ

3.1 Інструкція з розгортання вебплатформи

3.2 Інструкція з наповнення вебплатформи

3.3 Інструкція з популяризації та підтримки вебплатформи

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення розробки вебплатформи

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

4.3 Розрахунок витрат на електроенергію

4.4 Розрахунок суми амортизаційних відрахувань

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості вебплатформи

4.7 Розрахунок ціни вебплатформи

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Відповідальність за порушення законодавства про охорону праці

5.2 Комплекс заходів по зменшенню шуму в джерелі його виникнення

6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебплатформи «GalleryHub».

5. Перелік графічного матеріалу:

1. Схема структурна клієнтської частини
2. UML-діаграма варіантів використання
3. ER-діаграма бази даних
4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проєкту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	.06.2026	
12	Захист кваліфікаційної роботи	.06.2026	

7. Дата видачі завдання: 06 березня 2026 р.

Студент

(підпис)

Андрій СТУКАЛО

Керівник роботи

(підпис)

Руслан СЛОБОДЯН

ЗМІСТ

Анотація.....	7
Abstract.....	8
Вступ.....	9
1 Загальний розділ.....	10
1.1 Аналітичний огляд існуючих рішень.....	10
1.2 Технічне завдання	12
1.2.1 Найменування та область застосування	12
1.2.2 Призначення розробки.....	13
1.2.3 Вимоги до функціоналу вебплатформи.....	13
1.2.4 Вимоги до програмної документації.....	16
1.2.5 Техніко-економічні показники	17
1.2.6 Стадії та етапи розробки	17
1.2.7 Порядок контролю та прийому.....	19
2 Розробка технічного та робочого проєкту.....	20
2.1 Розробка структури вебплатформи	20
2.2 Створення та верстка сторінок вебплатформи	23
2.3 Розробка структури бази даних вебплатформи	38
2.4 Програмування вебплатформи	42
2.4.1 Написання клієнтської частини.....	42
2.4.2 Написання серверної частини.....	45
2.5 Тестування вебплатформи	49
3 Спеціальний розділ	57
3.1 Інструкція з розгортання вебплатформи	57
3.2 Інструкція з наповнення вебплатформи	62
3.3 Інструкція з популяризації та підтримки вебплатформи.....	65

					2026.КВР.122.423.21.00.00 ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Стукало А.В.			Розробка вебплатформи для експозиції та реалізації творів мистецтва «GalleryHub» Пояснювальна записка		
Перевір.		Слободян Р.О.					
Реценз.							
Н. Контр.		Приймак В.А.					
Затверд.							
					Літ.	Арк.	Аркушів
						5	100
					ВСП ТФК ТНТУ КН-423 м. Тернопіль		

4	Економічний розділ	67
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення НДР	67
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи	68
4.3	Розрахунок витрат на електроенергію	70
4.4	Розрахунок суми амортизаційних відрахувань.....	71
4.5	Обчислення накладних витрат.....	71
4.6	Складання кошторису витрат та визначення собівартості вебплатформи «GalleryHub»	72
4.7	Розрахунок ціни розробки вебплатформи «GalleryHub»	72
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	73
5	Охорона праці, техніка безпеки та екологічні вимоги	75
5.1	Відповідальність за порушення законодавства про охорону праці	75
5.2	Комплекс заходів по зменшенню шуму в джерелі його виникнення.....	76
	Висновки.....	78
	Перелік посилань	80
	Додатки	81
	Додаток А. Лістинг файлу «App.tsx».....	81
	Додаток Б. Лістинг файлу «ArtworkCard.tsx».....	84
	Додаток В. Лістинг файлу «HeroSlider.tsx»	88
	Додаток Г. Лістинг файлу «AddArtwork.tsx».....	91
	Додаток Д. Лістинг файлу «cartStore.ts»	99

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розробка вебплатформи для експозиції та реалізації творів мистецтва GalleryHub».

Метою цієї кваліфікаційної роботи є розробка клієнт-серверного вебзастосунку для публікації, перегляду, модерації та безпечного придбання творів мистецтва.

У загальному підсумку пояснювальна записка складається з 5 розділів. У загальній частині описуються аналітичний огляд існуючих рішень у сфері електронної комерції предметів мистецтва та детальний аналіз технічного завдання.

У другому розділі представлено процес створення програмного продукту, опис та обґрунтування вибору методів проєктування програми, а також тестування та налагодження програмних модулів. В спеціальній частині описані процес інсталяції вебплатформи в інтернет, детальна інструкція з користування вебплатформою для різних типів користувачів. Описано шляхи популяризації вебплатформи.

Для оцінки комерційної привабливості розрахунок вартості розробки та загальної економічної ефективності приведено в економічній частині пояснювальної записки. Основні питання охорони праці та техніки безпеки розглянуто в п'ятому розділі, зокрема проаналізовано відповідальність за порушення законодавства про охорону праці та розроблено комплекс заходів по зменшенню шуму в джерелі його виникнення.

Загальний обсяг пояснювальної записки становить 100 сторінок.

До складу кваліфікаційної роботи входить графічна частина, яка складається з структурної схеми вебплатформи, діаграми варіантів використання, діаграми бази даних, а також техніко-економічних показників, що виконані на окремих аркушах формату A1.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

ABSTRACT

Topic of the qualification thesis: Development of the "GalleryHub" web platform for the exhibition and sale of artworks.

The purpose of this qualification thesis is to develop a client-server web application for publishing, viewing, moderating, and securely purchasing artworks.

In summary, the explanatory note consists of 5 chapters. The general section provides an analytical review of existing solutions in the field of art e-commerce and a detailed analysis of the technical specifications.

The second chapter presents the software product development process, the description and justification of the chosen software design methods, as well as the testing and debugging of software modules. The dedicated section describes the deployment process of the web platform to the Internet and provides detailed user instructions for different types of users. Methods for promoting and popularizing the web platform are also outlined.

To evaluate commercial viability, the calculation of development costs and overall economic efficiency is presented in the economic section of the explanatory note. Key issues of occupational health and safety are addressed in the fifth chapter; in particular, liability for violations of occupational safety legislation is analyzed, and a set of measures to reduce noise at its source is developed.

The total volume of the explanatory note is 100 pages.

The qualification thesis includes a graphical part consisting of the web platform's structural diagram, a use case diagram, a database diagram, as well as technical and economic indicators, all executed on separate A1 sheets.

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

В умовах стрімкого розвитку цифрових технологій та глобальної трансформації мистецького ринку виникає нагальна потреба у створенні надійних інструментів для дистанційної взаємодії між сучасними художниками та колекціонерів. Зазначені зміни вимагають впровадження інноваційних рішень. Через необхідність оперативної адаптації арт-ринку до вимог цифрової епохи розробка власних програмних продуктів для демонстрації та продажу шедеврів стає вкрай важливою задачею сьогодні. Платформи стають головним каналом продажів.

Актуальність розробки спеціалізованої вебплатформи GalleryHub обумовлена необхідністю демократизації доступу до унікальних творів мистецтва та автоматизації процесів їх перевірки і безпечної реалізації. В основу побудови загальної архітектури ресурсу покладено сучасний компонентний принцип. Для досягнення високої ефективності процесів обміну інформацією було визначено основну мету створити функціональну систему з вбудованими механізмами модерації контенту. Користувачі отримують зручний інструментарій.

Під час безпосередньої реалізації даного проєкту розв'язано комплекс складних інженерних задач. Зокрема, було виконано детальне проєктування реляційної структури бази даних, реалізовано адаптивний інтерфейс на базі бібліотеки React та побудовано стабільну серверну частину. Вказаний підхід дозволяє постійно підтримувати високу швидкість мережевої взаємодії. Швидкість завантаження є критичною.

У даній роботі детально обґрунтовано вибір технологічного стеку Node.js та PostgreSQL як найбільш оптимального середовища для розробки масштабованих маркетплейсів із високим рівнем безпеки. Запропоновані у роботі програмні рішення мають вагоме практичне значення для розвитку локального мистецького середовища. Практичне використання отриманих результатів охоплює роботу приватних галерей та організацію онлайн-аукціонів. Створена платформа повністю готова до публічного мережевого доступу зараз.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Сьогодні багато художників намагаються знайти зручні способи для продажу своїх робіт через мережу. Творчі люди часто обирають між готовими торговими майданчиками та розробкою власного програмного продукту. Популярні сервіси на зразок Etsy пропонують швидкий старт для новачків. На рисунку 1.1 можна побачити приклад типового інтерфейсу такого магазину.

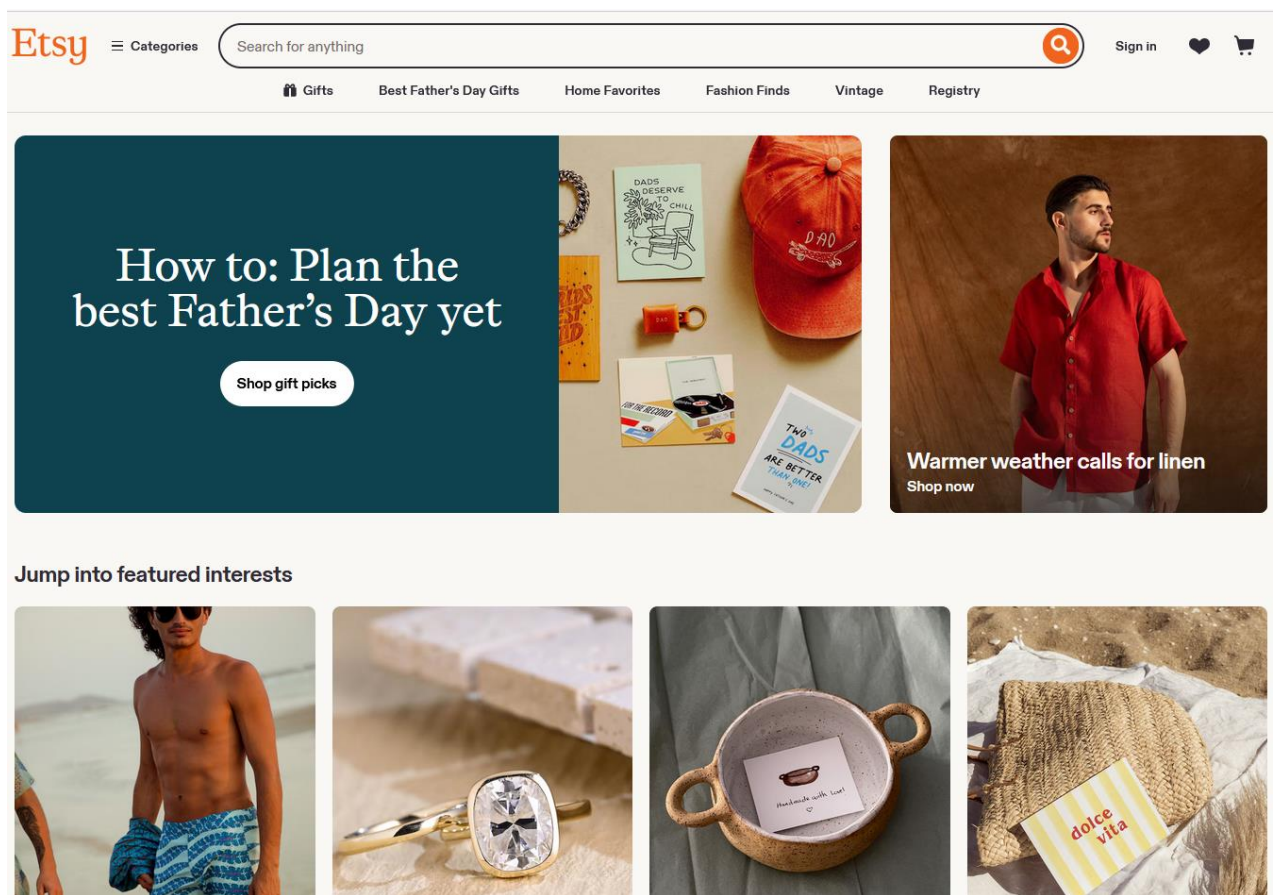


Рисунок 1.1 - Інтерфейс торгового майданчика Etsy

Готові платформи зазвичай мають багато прихованих недоліків для професійного митця. Вони стягують щомісячну плату за кожне виставлене на продаж полотно. Сервіси також забирають великий відсоток від кожної успішної угоди з покупцем. Власник такого магазину не може змінити структуру сторінок

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

або додати нові функції. Дизайн більшості профілів виглядає дуже однотипно та не передає духу галереї. Це заважає художнику створювати власний впізнаваний бренд у цифровому просторі.

Іншим варіантом є використання конструкторів сайтів на зразок Shopify або схожих інструментів. Ці програми дають більше свободи у виборі кольорів та шрифтів для сторінок. Однак вони все одно залишаються закритими для серйозних технічних змін коду. Також потрібно щомісяця платити за оренду майданчика навіть при відсутності продажів. Популярні хмарні сервіси часто не підтримують зручні для нас платіжні методи. Це створює зайві труднощі під час оформлення замовлень покупцями з нашого регіону. Сторінка магазину Shopify представлена на рисунку 1.2.

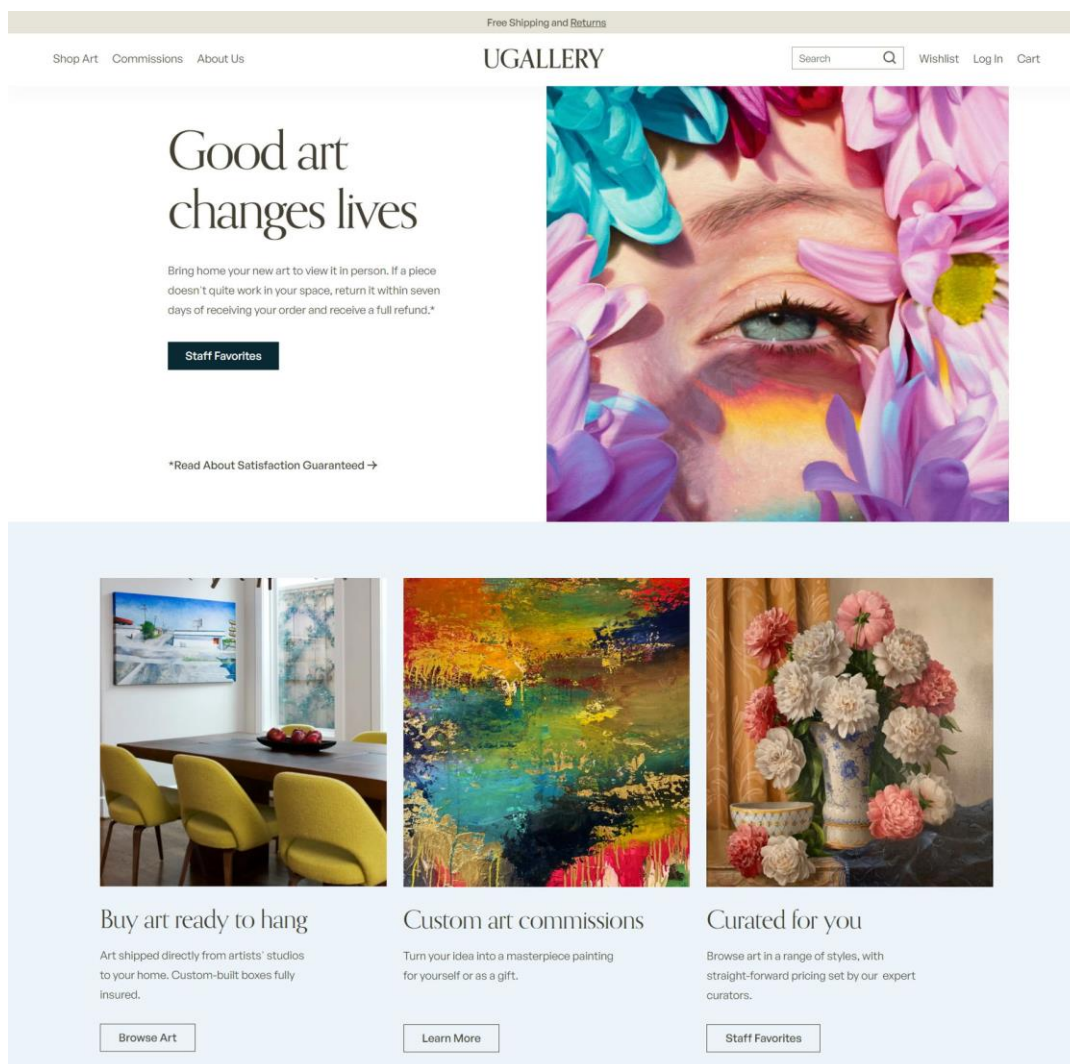


Рисунок 1.2 - Приклад магазину на Shopify

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

Розробка власного проекту Gallery Hub дає змогу повністю усунути ці проблеми. Митець отримує повний контроль над усіма даними та грошовими потоками. Програма працюватиме без жодних щомісячних відрахувань стороннім власникам платформ. Можна буде самостійно обирати способи завантаження фотографій та зберігання інформації. Це допомагає створити неповторну атмосферу справжньої мистецької виставки на екрані смартфона. Власна програма стане надійною базою для подальшого зростання приватного бізнесу.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

У межах даної кваліфікаційної роботи було спроектовано та реалізовано програмний продукт під назвою «GalleryHub - веб-платформа для експозиції та реалізації творів мистецтва». Скороченим позначенням системи є назва «GalleryHub». За своєю суттю розробка є спеціалізованим вебресурсом, який забезпечує зручну демонстрацію художніх робіт та автоматизує процеси їх комерційної реалізації в онлайн-просторі. Це дозволяє об'єднати митців та покупців.

Головною сферою практичного впровадження майбутнього рішення виступатиме сучасна галузь електронної комерції, орієнтована на дистрибуцію творів мистецтва та арт-об'єктів, включаючи віртуальні галереї та аукціони. Завдяки гнучкій структурі застосунок має передбачати можливість інтеграції у виставкову діяльність фізичних художніх салонів, приватних студій, а також некомерційних творчих об'єднань. Проектована платформа буде відкритою для незалежних авторів.

Для безперешкодного доступу до функціональних можливостей веб-платформи користувачі використовуватимуть стаціонарні персональні комп'ютери, сучасні ноутбуки, а також портативні планшети та мобільні телефони за наявності постійного стабільного інтернет-з'єднання. Взаємодія з

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

вебресурсом та запуск усіх функцій здійснюватиметься через сучасні веббраузери завдяки повній оптимізації під популярні оглядачі без необхідності інсталяції додаткового програмного забезпечення, що спростить взаємодію з платформою.

1.2.2 Призначення розробки

Головною метою проекту є побудова зручного цифрового простору для популяризації творчості. Планується створити інструмент для швидкої публікації нових робіт у мережі інтернет. Програма має гарантувати надійний зв'язок між автором та поціновувачем прекрасного. Використання цього додатка допоможе повністю відмовитися від паперових каталогів та довгих листувань. Покупець зможе самостійно обрати картину та оформити доставку за лічені хвилини. Це пришвидшить роботу митця та підвищить рівень продажів.

Конкретне призначення розробки включає реалізацію кількох важливих завдань для успіху бізнесу. Першим завданням є створення зрозумілого каталогу з якісними фотографіями всіх наявних полотен. Друге завдання полягає у забезпеченні безпечного збереження даних про користувачів та замовлення. Програма також має автоматично сповіщати адміністратора про нові запити від клієнтів. Використання хмарних сервісів дозволить швидко показувати зображення навіть при повільному зв'язку. Додаток стане основним місцем для знайомства глядачів із новими творами мистецтва.

1.2.3 Вимоги до функціоналу вебплатформи

Під час проєктування архітектури майбутнього вебресурсу за основу прийнято концепцію чіткого розподілу повноважень між різними категоріями користувачів системи. Для реалізації цієї стратегії передбачено розмежування прав доступу до інформаційних ресурсів та керуючих елементів інтерфейсу. Завдяки такому рішенню вдається мінімізувати ризики несанкціонованого доступу до баз даних. Це забезпечує надійну безпеку користувацьких профілів.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

З метою організації ефективної взаємодії в межах платформи виділено такі основні ролі користувачів:

- гості системи (неавторизовані відвідувачі сайту);
- зареєстровані користувачі (художніх творів або художники, що публікують власні полотна);
- адміністратори (керівники майданчика з повними правами доступу).

Для кожної із зазначених груп розроблено унікальний перелік функціональних вимог, що детально описують логіку роботи відповідних розділів вебресурсу. Діаграма варіантів використання зображена на плакаті 2026.КВР.122.423.21.00.00 ДВ графічної частини пояснювальної записки.

Вимоги до публічної частини інтерфейсу. Для забезпечення належного рівня взаємодії з відвідувачами сайту публічний інтерфейс має надавати широкі можливості для ознайомлення з художнім контентом. На головній сторінці необхідно реалізувати динамічний банер із демонстрацією останніх надходжень творів мистецтва. Дане рішення підвищить інтерес потенційних покупців. В межах основного каталогу передбачається впровадження гнучкого механізму пошуку та фільтрації за такими критеріями:

- тематичні категорії художніх робіт;
- матеріали виконання (наприклад, олія чи полотно);
- фізичні габарити (висота та ширина робіт);
- діапазон вартості та додаткові пошукові теги.

Для кожного окремого твору мистецтва має створюватися деталізована інформаційна картка з можливістю збільшення основного зображення при натисканні. Сюди також вноситься опис стилю та техніки. Важливою вимогою є забезпечення роботи локального кошика та списку обраних товарів, які зберігаються безпосередньо у веб-переглядачі користувача. Це спрощує вибір до моменту реєстрації.

Вимоги до особистого кабінету автора Для зареєстрованих митців необхідно передбачити персональний кабінет, який слугуватиме основним

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

інструментом на платформі. У межах цього розділу має бути спроектована зручна форма додавання нових художніх робіт із можливістю завантаження кількох зображень одночасно. Такий підхід прискорить наповнення каталогу. Через особистий кабінет художнику надається можливість здійснювати такі дії:

- редагувати текстові описи та технічні параметри власних творів;
- самостійно встановлювати ціни та змінювати статус наявності картин;
- відстежувати поточний стан модерації опублікованих робіт;
- переглядати історію замовлень від покупців для аналізу попиту.

Усі завантажені матеріали перед оприлюдненням мають отримувати статус очікування перевірки. Це допоможе уникнути публікації небажаного контенту.

З метою контролю за функціонуванням платформи передбачається розробка закритого адміністративного інтерфейсу з розширеними повноваженнями. Через цю панель адміністратор має здійснювати перевірку та затвердження заявок від художників на публікацію нових робіт. Система повинна блокувати некоректні оголошення. Додатково до цього висуваються вимоги щодо забезпечення управління метаданими:

- створення, редагування та видалення категорій мистецтва;
- динамічне оновлення бази даних використовуваних матеріалів;
- адміністрування списку тегів для кращої класифікації робіт;
- моніторинг оформлених замовлень та перегляд фінансової статистики.

Завдяки наявності аналітичних звітів керівництво отримує оперативну інформацію про динаміку продажів. Це дозволить вчасно коригувати стратегію розвитку.

Вимоги до адаптивності та технічних параметрів дизайну Для створення комфортних умов використання системи на різноманітних пристроях обов'язковою є реалізація адаптивної верстки інтерфейсу. Дизайн має автоматично та коректно масштабуватися під розміри екранів смартфонів, планшетів і настільних комп'ютерів. Це гарантує зручність використання сайту. Під час розробки шаблонів сторінок необхідно застосувати сучасні стандарти

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

стилізації та оптимізації для забезпечення високої швидкості завантаження. Швидкий відгук системи утримуватиме увагу користувачів.

1.2.4 Вимоги до програмної документації

Після завершення розробки заплановано підготовку пакета інструкцій та документації для подальшої технічної підтримки проєкту. Наявність супровідної документації дозволить власнику самостійно здійснювати адміністрування додатка без обов'язкового залучення сторонніх фахівців.

Першим документом виступатиме технічна інструкція з розгортання програми на вебсервері. У тексті буде детально описано порядок встановлення бази даних та налаштування середовища оточення для сервера. Передбачено наведення переліку команд для запуску фонових процесів додатка. Окремий розділ описуватиме процедуру підключення доменного імені та налаштування сертифікатів безпеки SSL, що дозволить швидко розгорнути проєкт у робочому середовищі після завершення етапу програмування.

Другий документ стосуватиметься регламенту обслуговування бази даних та оновлення коду. У ньому будуть представлені чіткі рекомендації щодо створення регулярних резервних копій усієї інформації на сайті. Текст описуватиме процес завантаження оновлених файлів програми на сервер без зупинки його роботи. Буде передбачено можливість самостійного проведення дрібних технічних робіт та очищення тимчасових файлів для забезпечення стабільного функціонування галереї.

Третім документом стане посібник користувача для художників та адміністраторів. Цей матеріал планується викласти простою мовою без складних технічних термінів. Буде надано детальні пояснення функцій інтерфейсу з додаванням знімків робочих екранів. Документ має допомогти новим авторам швидко опанувати процес завантаження перших картин, а адміністраторам — порядок модерації нових робіт та налаштування категорій. Таке керівництво дозволить мінімізувати час на навчання користувачів платформи.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

1.2.5 Техніко-економічні показники

Економічна частина проєкту базуватиметься на розрахунках витрат часу та фінансових ресурсів. Планується виконати всі роботи згідно із затвердженим графіком навчального процесу.

Загальний час на розробку всього програмного коду складе приблизно 150 годин. Цей термін включатиме проєктування бази даних, написання серверної логіки та верстку користувацького інтерфейсу. Загальна вартість створення платформи не повинна перевищувати 81 251 гривень. Вказана сума покриватиме оплату хостингу, доменного імені та роботу розробника. Такий бюджет є обґрунтованим для запуску стартової версії проєкту. Детальні розрахунки будуть представлені у четвертому розділі пояснювальної записки.

Прогнозований термін окупності вкладених коштів складе близько двох календарних років. Повернення інвестицій очікується завдяки залученню покупців через інтернет-простір. Впровадження платформи дозволить художникам реалізовувати свої роботи без посередників та високих додаткових комісій, що збільшить чистий прибуток від кожної успішної операції на торговому майданчику. Надалі це забезпечить можливість розширення функціоналу системи та масштабування бізнесу.

Таблиця з техніко-економічними показниками буде приведена у графічному матеріалі (плакаті) 2026.КВР.122.423.21.00.00 ТБ.

1.2.6 Стадії та етапи розробки

Створення вебплатформи «GalleryHub» планується реалізувати через кілька послідовних етапів, кожен з яких наблизитиме до отримання повністю готового до роботи продукту.

На першій стадії буде проведено аналіз вимог та планування архітектури. Передбачено вибір оптимальних мов програмування та інструментів для роботи з базою даних. Планується розробити точну схему всіх таблиць та зв'язків між

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

даними, визначити перелік головних сторінок та скласти графік виконання завдань для уникнення помилок на етапах написання коду. Також на цій стадії буде обрано хмарне сховище для збереження зображень творів мистецтва.

Друга стадія передбачатиме розробку візуального дизайну та макетів усіх екранів. Планується створити прототипи інтерфейсу для персональних комп'ютерів та мобільних телефонів у графічному редакторі. Буде підібрано кольорову палітру, яка не відволікатиме увагу від самих картин. Спроектують зручне розміщення кнопок фільтрації, пошуку та оформлення замовлення у кошику. Інтерфейс має бути максимально простим та зрозумілим для користувачів будь-якого віку.

Третя стадія включатиме написання серверної частини програми на базі Node.js. Заплановано налаштування з'єднання з базою даних та реалізацію логіки обробки запитів. Мають бути створені надійні маршрути для отримання списків товарів та авторизації. Передбачено безпечне шифрування паролів перед їхнім записом у базу даних. Також на цьому етапі буде реалізовано інтеграцію з обраним хмарним сервісом, що закладе основу для функціонування всього вебресурсу.

На четвертій стадії розроблятиметься клієнтська частина за допомогою бібліотеки React. Статичні макети дизайну будуть перетворені на живі інтерактивні сторінки галереї. Передбачено підключення користувацького інтерфейсу до сервера для відображення актуальних даних про картини. Планується реалізувати логіку роботи кошика, пошуку та особистого кабінету автора, а також здійснити верстку всіх елементів для мобільних екранів, що забезпечить зручну взаємодію з системою.

П'ята стадія буде присвячена ретельному тестуванню та усуненню виявлених помилок. Планується перевірка працездатності кожного посилання, коректності відображення зображень, а також надійності захисту конфіденційних даних адміністраторів та художників. Буде проведено випробування на різних пристроях для підтвердження адаптивності дизайну. Після успішного завершення тестування програма вважатиметься готовою до розгортання в мережі Інтернет.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

1.2.7 Порядок контролю та прийому

Процедура прийому готової роботи виступатиме фінальним етапом процесу реалізації проєкту, що дозволить підтвердити відповідність створеного програмного забезпечення всім вимогам початкового технічного завдання.

Перевірка розпочинатиметься з функціонального тестування можливостей вебсайту галереї. Передбачено демонстрацію процесів реєстрації нового автора та завантаження художньої роботи у базу даних. Буде продемонстровано роботу системи пошуку за категоріями, додавання товарів до кошика, а також процеси оформлення замовлення та надходження сповіщень про нову покупку.

Наступним кроком стане тестування стабільності системи у різних популярних веббраузерах. Програмний продукт має відображатися однаково коректно у Google Chrome, Mozilla Firefox та на мобільних пристроях. Планується перевірити швидкість завантаження сторінок за умови наявності великої кількості зображень високої роздільної здатності, а також переконатися у належній працездатності всіх інтерактивних елементів на сенсорних екранах смартфонів, що гарантуватиме доступність галереї для широкого кола користувачів.

Прийом роботи здійснюватиметься під час публічного захисту кваліфікаційної роботи перед комісією. Передбачено презентацію повністю робочого вебзастосунку та детальний опис обраних технічних рішень. За вимогою комісії виконуватимуться контрольні дії у реальному часі для підтвердження працездатності системи. Після демонстрації всіх задекларованих функцій робота вважатиметься виконаною та прийнятою до експлуатації.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури вебплатформи

Для забезпечення високої зручності користувачів під час взаємодії з платформою за основу проектування інтерфейсу обрано звичні та перевірені патерни побудови електронних маркетплейсів. Завдяки використанню класичних навігаційних рішень майбутні відвідувачі зможуть інтуїтивно орієнтуватися в структурі сайту з перших секунд перебування на сторінці. Це значно зменшить показник відмов. Для полегшення доступу навігаційні елементи та інтерактивні блоки необхідно розмістити у найбільш звичних зонах екрана.

Під час проектування логічної структури вебплатформи було сформовано детальну схему взаємозв'язків між усіма окремими функціональними сторінками. Завдяки побудові оптимальних маршрутів навігації користувачі зможуть переходити до цільової дії всього за кілька кліків. Це прискорить оформлення замовлень. На плакаті структурної схеми інтерфейсу платформи 2026.KBP.122.423.21.00.00 СС, продемонстровано повну структурну схему інтерфейсу вебплатформи GalleryHub.

Представлена на схемі логічна модель чітко розмежовує відкриту публічну частину та захищені внутрішні розділи для авторизованих користувачів та адміністраторів. Для створення цілісного уявлення про побудову інтерфейсу нижче наведено детальний опис структури ключових вебсторінок:

1) Головна сторінка. У верхній частині цієї сторінки необхідно розмістити динамічний промо-слайдер із плавними ефектами зміни кадрів для показу найкращих робіт. Під слайдером доцільно спроектувати сітку категорій у вигляді великих графічних плиток для швидкого переходу до потрібних художніх жанрів. У нижній частині сторінки мають бути розташовані картки нових творів мистецтва та окремий блок із профілями відомих авторів. Такий розподіл допоможе залучити користувачів до перегляду каталогу.

2) Галерея (Каталог робіт). Для зручності пошуку інтерфейс цієї сторінки

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

має складатися з двох основних колонок. У лівій частині потрібно розмістити бічну панель (Sidebar) із фільтрами за ціною, матеріалами, розмірами та тегами творів. У правій частині слід запроектувати адаптивну сітку карток арт-об'єктів із підтримкою сортування за датою додавання або ціною. Завдяки гнучкій фільтрації користувач легко знайде потрібне полотно.

3) Сторінка детальної інформації про твір. У лівій половині робочої області планується розмістити велике деталізоване зображення картини з можливістю відкриття його на весь екран. У правій колонці необхідно відобразити назву, ціну, технічні характеристики (матеріал, розміри) та картку профілю автора з кнопкою підписки. Трохи нижче слід передбачити помітні кнопки «Додати в кошик» та «В обране» для швидкої покупки.

4) Профіль художника. У верхній частині сторінки доцільно спроектувати персональний банер із фотографією митця та блоком для короткої біографії. Нижче має бути розміщена сітка всіх робіт цього автора з позначенням їхнього поточного статусу наявності чи продажу. Таке рішення створює ефект віртуальної персональної виставки.

5) Кошик та оформлення замовлення. У лівій частині екрана планується відобразити список вибраних товарів із мініатюрами, цінами та кнопками вилучення. У правій частині необхідно розмістити форму введення даних доставки (країна, місто, адреса, телефон отримувача) та підсумкову суму до оплати. Це дозволяє утримувати всю важливу інформацію перед очима.

6) Кабінет художника. У лівій частині кабінету передбачається вертикальне меню навігації для переходу між вкладками профілю, робіт та замовлень. У центральній зоні буде розміщено список завантажених автором картин із кольоровими індикаторами статусу модерації (схвалено, очікує, відхилено). Для додавання нових робіт потрібно реалізувати спливаючу кнопку з переходом на форму завантаження.

7) Панель адміністратора. Для зручності модерації інтерфейс буде розділений на бокове меню керування та основну робочу область. У центральній частині адміністратор бачитиме список заявок від художників, кожну з яких

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

можна схвалити або відхилити безпосередньо з таблиці. Таке компонування елементів підвищує швидкість обробки контенту.

Процес взаємодії з платформою супроводжується заповненням низки інтерактивних форм. Для збору та валідації даних було спроектовано структуру полів, кожне з яких відповідає вимогам цілісності інформації в базі даних.

Першою в циклі використання платформи є форма реєстрації та входу. Вона містить поля для введення електронної адреси, пароля, а також імені та прізвища користувача. При цьому в полі пошти реалізовано автоматичну перевірку на коректність формату адреси. Це дозволяє уникнути помилок при створенні облікового запису.

Для наповнення галереї новим вмістом розроблено форму додавання твору мистецтва. У межах цієї форми передбачено такі обов'язкові для заповнення поля:

- назва твору (текстовий ввід);
- детальний опис (розширене текстове поле для розкриття концепції роботи);
- вартість у національній валюті (числове поле);
- габарити: ширина та висота в сантиметрах (числові параметри);
- категорія та матеріал (вибір зі списку зумовлених значень);
- теги (динамічне поле з підтримкою інтелектуальних підказок);
- модуль завантаження зображень (підтримка мульти-завантаження до 10 файлів).

Завершальним етапом покупки є форма оформлення доставки. Щоб забезпечити надійну логістику, користувач повинен вказати країну, місто та конкретну адресу відділення або будинку. Додатково збираються контактні дані отримувача, такі як номер телефону та повне ім'я. Усі поля проходять сувору перевірку перед відправкою на сервер.

Адміністративна частина ресурсу також містить форми для керування метаданими. Зокрема, реалізовано інтерфейс для швидкого редагування або видалення тегів і категорій. Кожне поле в цих формах призначене для короткого

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

текстового іменування об'єкта, що безпосередньо впливає на відображення в навігаційних фільтрах галереї.

2.2 Створення та верстка сторінок вебплатформи

Для побудови клієнтського боку вебплатформи GalleryHub розробниками було обрано сучасну бібліотеку React версії 19, яка забезпечує швидку роботу інтерфейсу завдяки використанню технології Virtual DOM. Це прискорює реакцію на дії користувача. У ролі інструменту збірки та локального сервера розгортання застосовано утиліту Vite, що дозволяє миттєво відображати зміни в коді без перезапуску. Завдяки цьому швидкість верстки істотно зростає.

Для організації зручного робочого середовища та збереження чіткої структури файлів було сформовано початкову ієрархію каталогів клієнтської частини. Початкову ієрархію каталогів клієнтської частини разом із її базовим наповненням продемонстровано далі на рисунку 2.1.

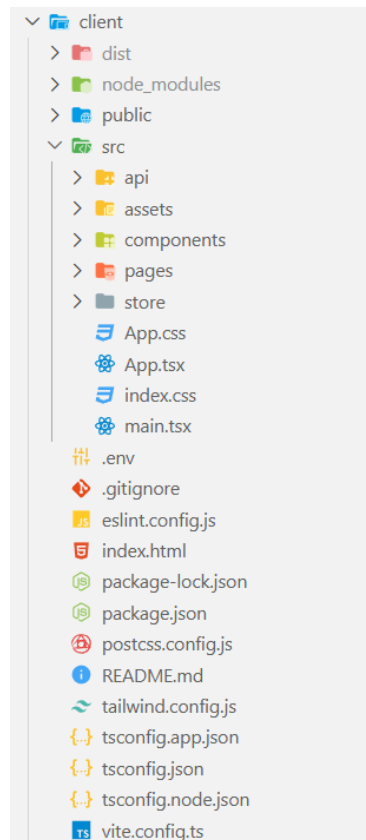


Рисунок 2.1 – Початкова структура проекту клієнтської частини GalleryHub

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

Показана на рисунку 2.1 структура каталогу містить директорію src, де було розміщено окремі папки для компонентів, сторінок та всіх глобальних стилів сайту. При цьому повний код компонента налаштування маршрутизації та відображення сторінок App.tsx представлений в додатку А пояснювальної записки для детального аналізу побудови клієнтської логіки. Під час написання програмних скриптів було впроваджено типізацію даних через мову TypeScript для підвищення рівня стабільності всієї системи. Це допомагає легко уникати критичних пошукових помилок. Кожен файл у цій структурі має своє чітке призначення для підтримки загальної модульності архітектури та зручного пошуку потрібних фрагментів коду.

Весь основний контент платформи відображається в межах головного контейнера index.html, який слугує єдиною точкою входу для всього створеного додатка в браузері. Його вихідний код подано в лістингу 2.1.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <title>client</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
</html>
```

Лістинг 2.1 – Вихідний код базового файлу index.html

Як демонструє лістинг 2.1, файл містить пряме підключення основного

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

скрипта `main.tsx`, що ініціалізує дерево компонентів та завантажує всі базові параметри стилів. Швидка реакція на дії користувача забезпечується через використання внутрішніх механізмів оновлення стану без повного перезавантаження сторінок у активному вікні браузера. Це суттєво економить апаратні ресурси пристрою.

Для швидкого оновлення візуального вигляду та адаптивності сторінок застосовано утилітарний CSS-фреймворк Tailwind CSS версії 4, стилі якого інтегруються безпосередньо у верстку компонентів.

Навігаційна панель спроектована як скрізний інтерактивний компонент із закріпленим позиціонуванням (sticky top-0) та високим індексом шару (z-50) для постійного відображення над контентом сторінки. У правій частині панелі реалізовано динамічне перемикання станів залежно від статусу авторизації за допомогою сховища стану `useAuthStore`. Для гостя (неавторизований стан) відображається кнопка «Увійти» з посиланням на форму авторизації. Для авторизованого покупця або художника з'являється блок із іконкою користувача, його ім'ям (яке веде до особистого кабінету) та кнопкою виходу (LogOut). Також за наявності ролі ADMIN надається лінк для переходу до панелі адміністрування. На панелі постійно присутні інтерактивні іконки кошика та списку обраного зі спеціальними індикаторами кількості доданих елементів (badges), що оновлюються в реальному часі. Зовнішній вигляд навігаційної панелі хедера сайту GalleryHub зображено на рисунку 2.2.

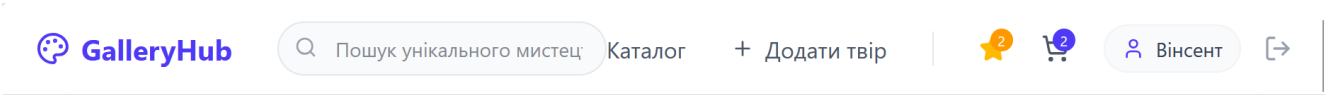


Рисунок 2.2 – Зовнішній вигляд навігаційної панелі хедера сайту GalleryHub

Для яскравого візуального залучення відвідувачів на головній сторінці сайту реалізовано велику промо-зону у вигляді слайдера HeroSlider. Зміна зображень творів мистецтва супроводжується плавними анімаціями загасання (fade), реалізованими через Framer Motion з контролем тривалості переходів. В

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

межах блоку розміщено текстовий опис та яскраву фіолетову кнопку заклик до дії (Call-To-Action), клік по якій плавно прокручує сторінку до каталогу. Повний код цього слайдера представлений в додатку В пояснювальній записки. Зовнішній вигляд промо-слайдера представлено далі на рисунку 2.3.

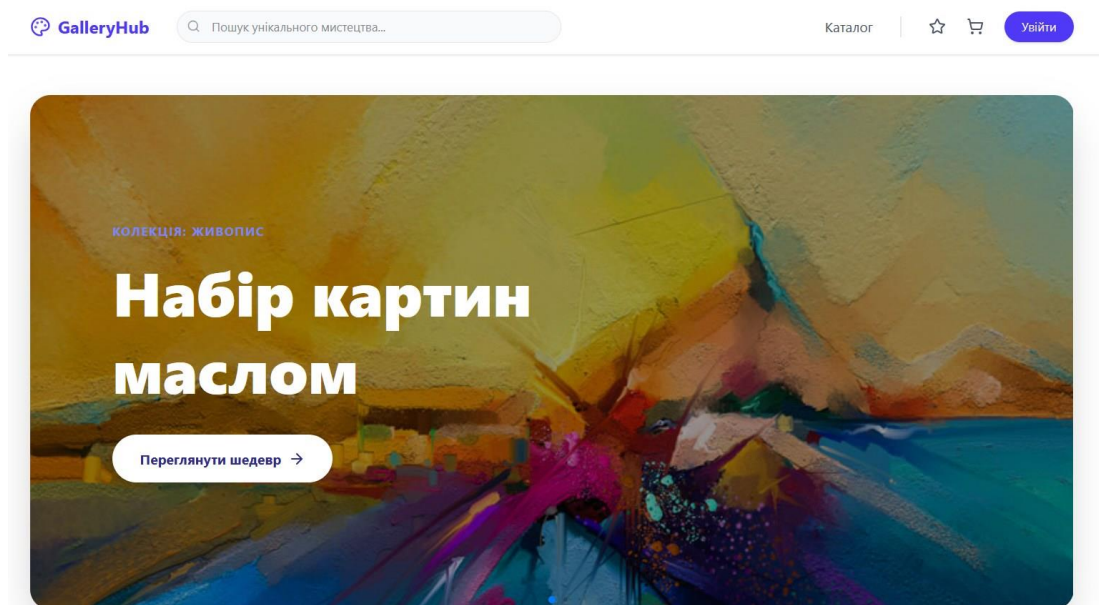


Рисунок 2.3 – Візуальний вигляд промо-слайдера HeroSlider на головній сторінці

Блок категорій CategoryGrid, який зображено на рисунку 2.4, слугує для швидкого переходу покупців до певних видів образотворчого мистецтва.

Купуйте за категоріями

Знайдіть ідеальний стиль для вашого простору

[Переглянути всі категорії](#)

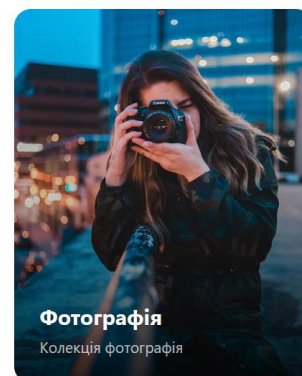
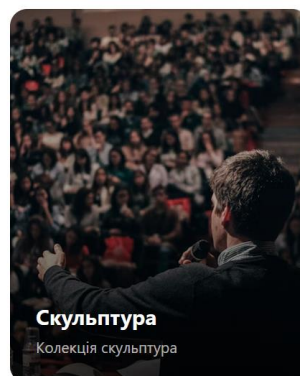


Рисунок 2.4 – Адаптивний блок категорій CategoryGrid

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Даний блок реалізований як адаптивна сітка карт із використанням CSS Grid (`grid grid-cols-1 sm:grid-cols-2 md:grid-cols-4 gap-6`). Кожна картка містить тематичне зображення з текстовим підписом. При наведенні курсору на картку спрацьовує мікроанімація підйому та збільшення масштабу зображення, що підвищує інтерактивність інтерфейсу.

Розділ популярних художників FeaturedArtists, що представлений на рисунку 2.5, відображає список рекомендованих авторів платформи з їхніми портретами та кількістю опублікованих картин.

Верстка виконана у вигляді горизонтального скрол-контейнера для зручності перегляду на мобільних пристроях. Для кожного художника створено окрему картку з кнопкою переходу на його персональну сторінку-портфоліо.

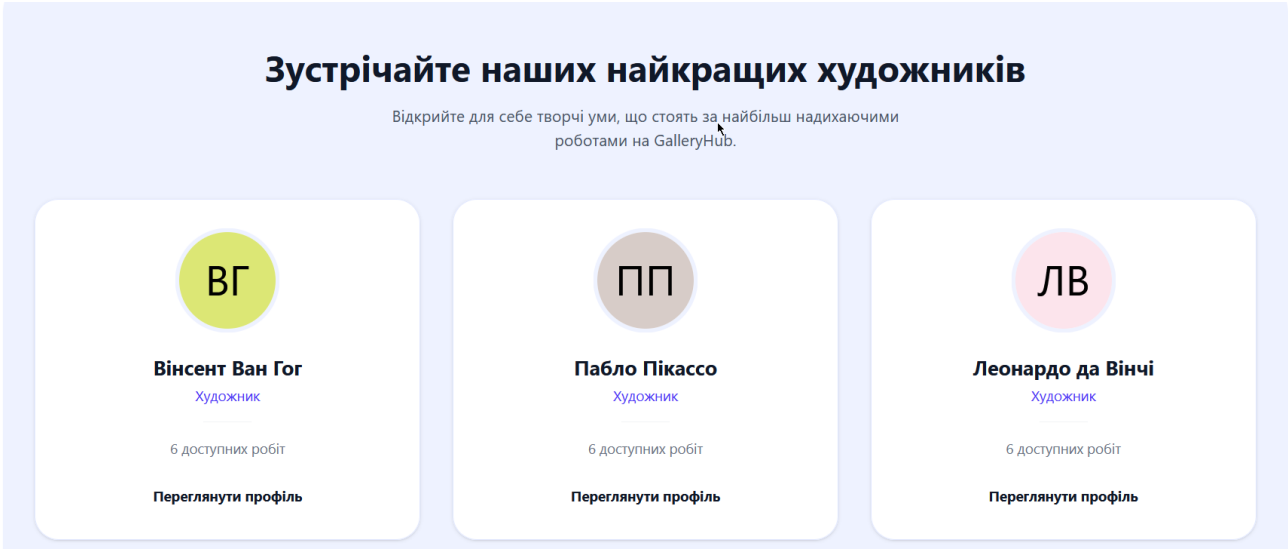


Рисунок 2.5 – Інтерфейсний блок популярних авторів FeaturedArtists

Нижній навігаційний блок Footer, зображений на рисунку 2.6, спроектований як триколонковий макет на базі Flexbox. Перша колонка містить логотип сайту та коротку інформаційну довідку про місію GalleryHub. Друга колонка містить структурований список посилань на інформаційні сторінки (контакти, правила платформи, допомога). Третя колонка містить кнопки переходу на соціальні мережі та форму швидкої підписки на розсилку новин.



Рисунок 2.6 – Інтерфейс підвалу сайту Footer

Сторінка каталогу є центральним інтерфейсом для пошуку творів мистецтва. Вона розбита на дві основні логічні зони: бічний сайдбар із фільтрами та головну сітку з картками картин. Макет автоматично перебудовується на мобільних пристроях, приховуючи фільтри під окрему кнопку, що відкриває модальне вікно знизу екрана. Загальний вигляд сторінки каталогу картин представлений на рисунку 2.7.

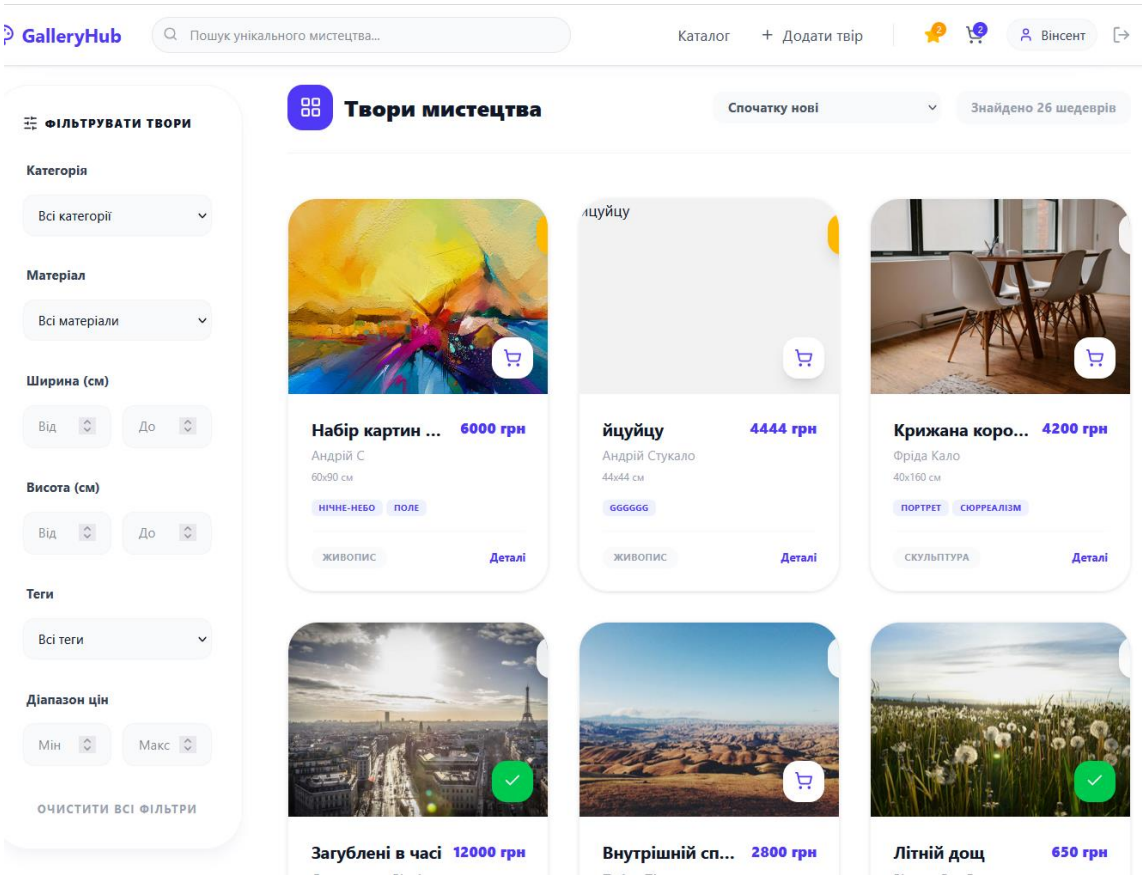


Рисунок 2.7 – Загальний вигляд сторінки каталогу картин

Компонент ArtworkCard є базовим елементом для відображення мініатюри твору мистецтва, його ціни, автора та додаткових тегів. Його програмну реалізацію наведено далі в лістингу 2.2.

```
import ...
export default function ArtworkCard({ artwork }: ArtworkProps) {
  const { user } = useAuthStore();
  const { addItem, items: cartItems } = useCartStore();
  const { toggleFavorite, items: favoriteItems } = useFavoriteStore();
  const isInCart = cartItems.some(item => item.id === artwork.id);
  const isFavorite = favoriteItems.some(item => item.id === artwork.id);
  const isAuthenticated = !!user;

  return (
    <div className="bg-white rounded-[2rem] shadow-sm border border-gray-
100 overflow-hidden hover:shadow-xl hover:shadow-indigo-100/50 transition-
all duration-300 group">
      {/Код рендерингу зображення та кнопок швидкої дії */}
    </div>
  );
}
```

Лістинг 2.2 – Програмний код компонента ArtworkCard

При цьому повний код компонента картки твору ArtworkCard представлений в додатку Б. Зовнішній вигляд компонента представлено далі на рисунку 2.8.

Бічна панель фільтрації, представлена на рисунку 2.9, дозволяє обмежувати вибір картин за категорією, матеріалом, розмірами (ширина/висота) та діапазоном цін за допомогою зручного повзунка-слайдера. Зміни параметрів фільтрації миттєво синхронізуються із параметрами URL-запиту через хук `useSearchParams` бібліотеки React Router, що дозволяє користувачеві ділитися посиланням на відфільтровані результати.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

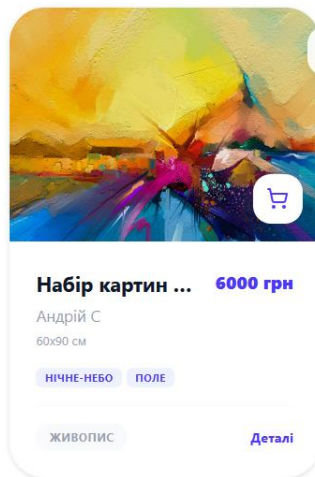


Рисунок 2.8 – Візуальний вигляд компонента картки твору ArtworkCard

Бічна панель фільтрації, представлена на рисунку 2.9, дозволяє обмежувати вибір картин за категорією, матеріалом, розмірами (ширина/висота) та діапазоном цін за допомогою зручного повзунка-слайдера. Зміни параметрів фільтрації миттєво синхронізуються із параметрами URL-запиту через хук `useSearchParams` бібліотеки React Router, що дозволяє користувачеві ділитися посиланням на відфільтровані результати.

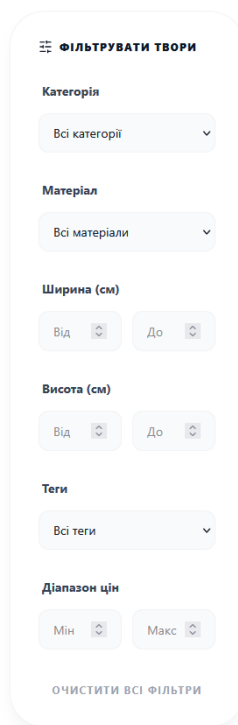


Рисунок 2.9 – Інтерфейс панелі фільтрів каталогу

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Розміщена у верхній частині каталогу панель пошуку та сортування, яку зображено на рисунку 2.10, містить текстове поле швидкого пошуку картин за назвою або описом, а також випадаючий список сортування (за зростанням чи спаданням ціни, датою додавання). Пошук реалізовано з оптимізацією обробки вводу для зменшення кількості запитів до сервера під час набору тексту.

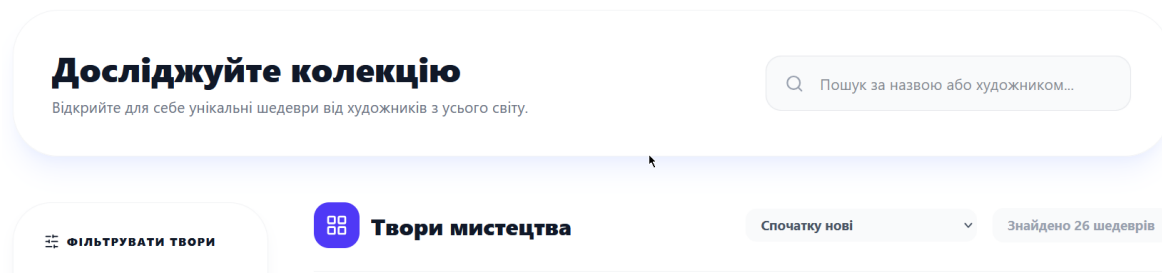


Рисунок 2.10 – Панель пошуку та сортування товарів

Ця сторінка детальної інформації про твір ArtworkDetails, зображена на рисунку 2.11, реалізує презентацію окремого художнього твору. Вона використовує двоколонковий макет. Ліва колонка відведена під галерею великих зображень із додатковими ескізами (thumbnails) під головним фото. Права колонка містить детальний опис концепції твору від автора, блок із технічними характеристиками, інформацію про художника та великі контрастні кнопки для додавання у кошик чи список обраного.

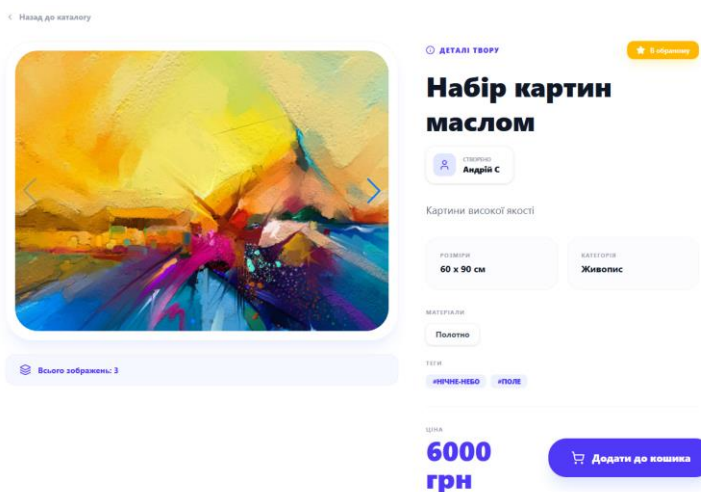


Рисунок 2.11 – Сторінка детальної інформації про твір ArtworkDetails

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

Сторінка обраного Favorites, яку представлено на рисунку 2.12, призначена для виведення списку робіт, які зацікавили користувача. Дані зберігаються в LocalStorage за допомогою Zustand і синхронізуються із сервером при вході в систему. Якщо список пустий, сторінка відображає графічну ілюстрацію з кнопкою переходу до каталогу для сприяння користувачу у перегляді робіт.

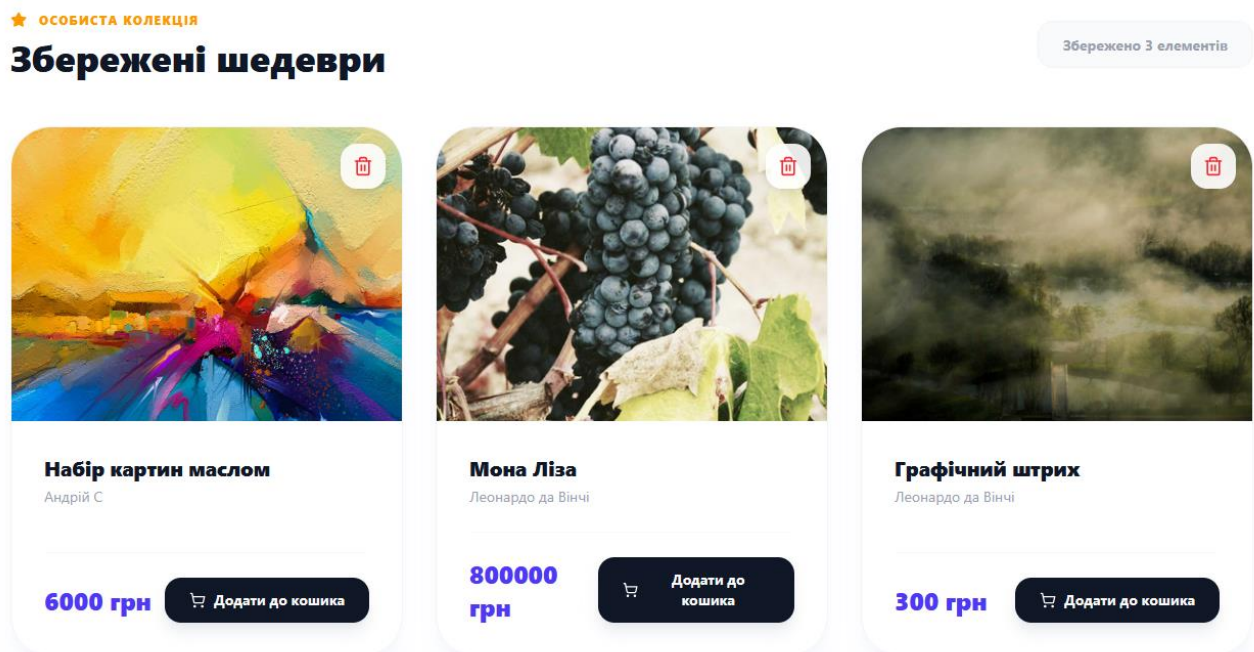


Рисунок 2.12 – Сторінка обраних творів мистецтва Favorites

Інтерфейс переліку товарів у кошику, зображений на рисунку 2.13, відображає перелік картин, які користувач готовий придбати. Кожна позиція містить мініатюру картини, назву, ім'я автора та ціну. Реалізовано можливість миттєвого видалення позицій зі списку з автоматичним перерахунком загальної суми покупки за допомогою Zustand сховища `cartStore`.

Форма оформлення замовлення та доставки, представлена на рисунку 2.14, суміщає перегляд кошика з інтерактивною формою для введення контактних даних та адреси доставки замовлення. Усі поля (телефон, ПІБ, адреса поштового відділення) проходять сувору валідацію за допомогою бібліотек Zod та React Hot Toast для виявлення помилок перед відправкою форми на сервер.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

1 Кошик

2 Доставка

3 Оплата

Перегляд кошика

Загублені в часі

12000 грн

Літній дощ

650 грн

Перейти до доставки →

Підсумок замовлення

Проміжний підсумок 12650 грн

Доставка **Безкоштовно**

Всього 12650 грн

ШВИДКА ДОСТАВКА НОВОЮ ПОШТОЮ

БЕЗПЕЧНА ОБРОБКА ПЛАТЕЖІВ

Рисунок 2.13 – Інтерфейс переліку товарів у кошику

✓ Кошик

2 Доставка

3 Оплата

Деталі доставки

Ім'я

Вінсент

Прізвище

Ван Гог

Номер телефону

Місто

Київ

Вулиця та номер будинку

вул. Хрещатик, 1

Поштовий індекс

01001

Назад

Наступний крок

Підсумок замовлення

Проміжний підсумок 12650 грн

Доставка **Безкоштовно**

Всього 12650 грн

ШВИДКА ДОСТАВКА НОВОЮ ПОШТОЮ

БЕЗПЕЧНА ОБРОБКА ПЛАТЕЖІВ

Рисунок 2.14 – Форма оформлення замовлення та доставки

Особистий кабінет користувача надає можливість керувати своїм профілем та переглядати історію замовлень. Для художників додатково виводиться вкладка «Мої твори», яка містить перелік усіх завантажених робіт із різними статусами модерації (APPROVED, PENDING, REJECTED). Код створення сховища кошика,

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

що взаємодіє з профілем, наведено в лістингу 2.3.

```
import { create } from 'zustand';
import { persist } from 'zustand/middleware';
import api from '../api/axios';

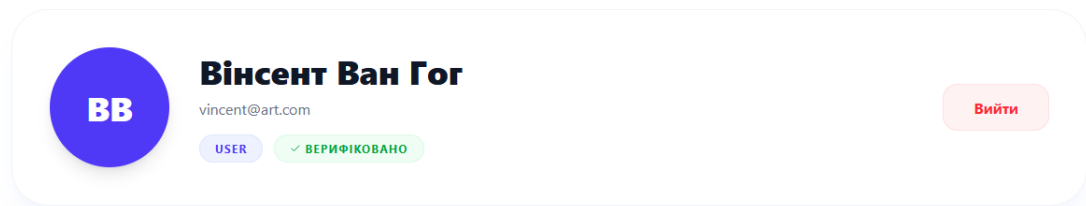
export const useCartStore = create<CartState>()(
  persist(
    (set, get) => ({
      items: [],
      addItem: async (artwork, isAuthenticated) => {
        const alreadyInCart = get().items.some(item => item.id ===
artwork.id);
        if (!alreadyInCart) {
          set({ items: [...get().items, artwork] });
          if (isAuthenticated) {
            await api.post('/users/cart', { artworkId:
artwork.id.toString() });
          }
        }
      },
      // Інші методи для видалення, очищення та синхронізації з сервером
    }),
    {
      name: 'galleryhub_cart',
    }
  )
);
```

Лістинг 2.3 – Програмний код сховища кошика cartStore на Zustand

При цьому повний код сховища стану кошика cartStore представлений в додатку Д пояснювальної записки. Зовнішній вигляд особистого профілю

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

користувача Profile зображено на рисунку 2.15.



Мої твори

7 робіт у списку

Рисунок 2.15 – Сторінка особистого профілю користувача Profile

Сторінка додавання картини AddArtwork, яку показано на рисунку 2.16, містить інтерактивну зону Drag-and-Drop для завантаження до 10 зображень картини, поля опису, вибору категорії та інноваційне поле введення тегів TagInput. Повний код цього компонента представлений в додатку Г пояснювальної записки.

Надішліть свій шедевр

Поділіться своїм мистецтвом зі світом. Заповніть деталі нижче.

Зображення твору (до 10 шт)

Натисніть, щоб завантажити декілька фото

PNG, JPG, GIF до 10MB кожне

Головне

Назва

Опис

Опишіть свій твір...

Ціна (ГРН)

\$ 100

Ширина (см)

50

Висота (см)

70

Категорія

Виберіть категорію

Матеріал

Виберіть матеріал

Теги твору

Введіть тег та натисніть E

Виберіть зі списку або введіть нові через Enter

Надіслати твір

Рисунок 2.16 – Інтерфейс додавання нового твору мистецтва AddArtwork

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

Вікно редагування інформації про картину, представлене на рисунку 2.17, реалізовано як спливаючу форму в кабінеті користувача, що дозволяє швидко оновлювати інформацію про вже завантажені твори мистецтва. Форма підтягує поточні дані з бази даних через API та дозволяє змінювати ціну, опис чи статус наявності картини з миттєвим збереженням результату.

The image shows a modal window titled "Редагувати твір" (Edit artwork) with a close button (X) in the top right corner. The form contains the following elements:

- Назва** (Name): A text input field containing "Мона Ліза".
- Опис** (Description): A text area containing "Відома картина".
- Ціна (грн)** (Price in UAH): A numeric input field with "9999999" and a dropdown arrow.
- Ширина** (Width): A numeric input field with "50" and a dropdown arrow.
- Висота** (Height): A numeric input field with "70" and a dropdown arrow.
- Теги твору** (Artwork tags): A text input field with a tag icon and the placeholder "Введіть тег та натисніть Enter...".
- Buttons**: "Скасувати" (Cancel) and "Зберегти зміни" (Save changes).

Рисунок 2.17 – Вікно редагування інформації про картину

Адміністративна панель AdminDashboard, зображена на рисунку 2.18, слугує для повного контролю за вмістом платформи. Вона розділена на кілька вкладок: модерація творів (перегляд нових заявок від художників, кожен з яких можна схвалити або відхилити), управління тегами (створення нових категорій, тегів та матеріалів) та список моїх творів чи замовлень для моніторингу загальних фінансових продажів.

Панель адміністратора



Рисунок 2.18 – Панель керування та модерації AdminDashboard

Окремим важливим розділом у панелі адміністратора є вкладка керування тегами, інтерфейс якої зображено на рисунку 2.19. У межах цього розділу реалізовано динамічну таблицю наявних у системі тегів. Це прискорює пошук потрібних значень. Для додавання нових тегів спроектовано компактну форму, яка містить текстове поле для назви та кнопку відправки на сервер. Редагування елементів реалізовано максимально зручно. При натисканні відповідної кнопки текстове поле в таблиці стає активним для введення нової назви безпосередньо в рядку. Після цього дані проходять перевірку. Для оновлення інформації система надсилає асинхронний запит до бази даних, що забезпечує миттєву зміну метаданих для всіх картин.

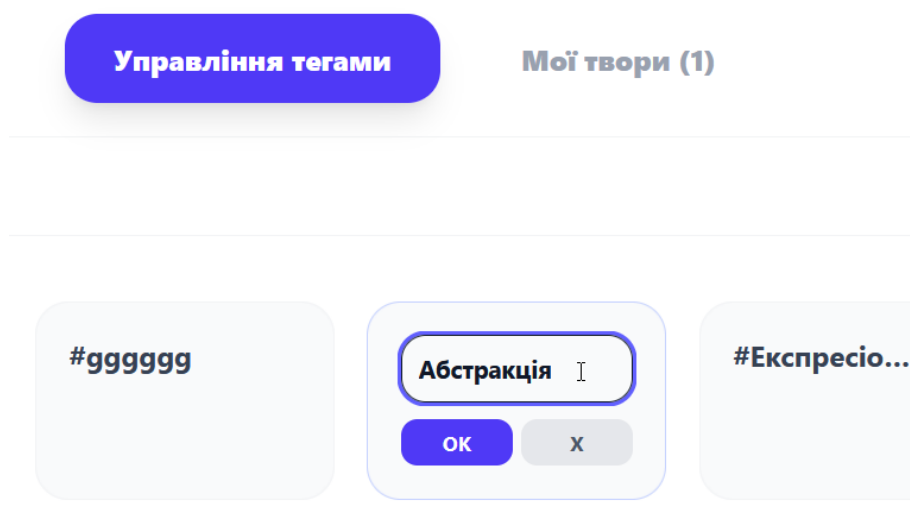


Рисунок 2.19 – інтерфейс розділу керування тегами в панелі адміністратора

Всі елементи інтерфейсу пройшли повну перевірку на адаптивність та сумісність із різними сучасними оглядачами, що забезпечує стабільний та зручний доступ користувачів до сервісу.

2.3 Розробка структури бази даних вебплатформи

Для забезпечення надійного збереження даних та підтримки високої швидкості обробки запитів у системі GalleryHub було обрано реляційну систему керування базами даних PostgreSQL. Завдяки використанню цієї технології забезпечується цілісність інформації та гнучкість при налаштуванні зв'язків між таблицями. Це гарантує стабільну роботу всього сервісу.

Детальна структурна схема бази даних із усіма зв'язками, типами полів та первинними й зовнішніми ключами подана в графічній частині кваліфікаційної роботи 2026.КВР.122.423.21.00.00 ДБ

Для логічного представлення даних у межах платформи було розроблено набір таблиць, опис структури яких наведено нижче:

Таблиця користувачів (users) – призначена для зберігання облікових даних та профільної інформації покупців і художників:

- id (BigInt, первинний ключ) – унікальний ідентифікатор користувача в базі даних;
- email (String, унікальне значення) – електронна пошта, що використовується для авторизації та сповіщень;
- password_hash (String) – зашифрований хеш пароля для безпечного входу;
- first_name (String) – ім'я користувача;
- last_name (String) – прізвище користувача;
- phone (String) – контактний номер телефону для зв'язку під час доставки;
- avatar_url (String) – посилання на зображення профілю;
- bio (String) – біографічні відомості про художника та його творчість;

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

- is_verified (Boolean) – статус верифікації облікового запису;
- role (Enum) – роль користувача в системі, яка визначає права доступу (USER, ADMIN);
- created_at, updated_at (DateTime) – мітки часу створення та оновлення запису.

Таблиця творів мистецтва (artworks) – містить інформацію про завантажені художні полотна та їхні параметри:

- id (BigInt, первинний ключ) – унікальний ідентифікатор картини;
- title (String) – назва художнього твору;
- description (String) – опис художньої концепції полотна від автора;
- width (Float) – ширина картини в сантиметрах;
- height (Float) – висота картини в сантиметрах;
- price (Decimal) – вартість роботи із високою фінансовою точністю;
- status (Enum) – поточний стан модерації або продажу (PENDING, APPROVED, REJECTED, SOLD);
- artist_id (BigInt, зовнішній ключ) – зв'язок із автором твору з таблиці users;
- category_id (BigInt, зовнішній ключ) – зв'язок із таблицею категорій;
- primary_image_id (BigInt, зовнішній ключ) – посилання на головне зображення полотна для прев'ю.

Таблиця категорій (categories) – призначена для класифікації творів за видами мистецтва:

- id (BigInt, первинний ключ) – унікальний ідентифікатор категорії;
- name (String) – текстова назва категорії (наприклад, Живопис);
- slug (String, унікальне) – людино-зрозуміле ім'я для відображення в URL браузера.

Таблиця матеріалів (materials) – містить перелік художніх матеріалів, що використовуються при створенні картин:

- id (BigInt, первинний ключ) – унікальний ідентифікатор матеріалу;

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

- name (String) – назва матеріалу (олія, акрил, полотно тощо);
- slug (String, унікальне) – ідентифікатор для швидкої фільтрації в каталозі.

Таблиця тегів (tags) – використовується для динамічного групування та пошуку картин за стилями чи темами:

- id (BigInt, первинний ключ) – унікальний ідентифікатор тегу;
- name (String) – текстова назва мітки;
- slug (String, унікальне) – унікальне ім'я для формування пошукових посилань.

–

Таблиця зображень (images) – зберігає файли робіт у хмарному сховищі Cloudinary:

- id (BigInt, первинний ключ) – унікальний ідентифікатор файлу;
- url (String) – пряме посилання на завантажене зображення;
- public_id (String) – унікальний ідентифікатор файлу в Cloudinary для видалення;
- artwork_id (BigInt, зовнішній ключ) – зв'язок із відповідною картиною в таблиці artworks.

Таблиця замовлень (orders) – містить інформацію про здійснені фінансові операції купівлі картин:

- id (BigInt, первинний ключ) – унікальний ідентифікатор замовлення;
- order_date (DateTime) – дата та час здійснення покупки;
- total_price (Decimal) – загальна сума чека замовлення;
- status (Enum) – поточний статус виконання (PENDING, PAID, SHIPPED, COMPLETED);
- payment_id (String) – ідентифікатор транзакції з платіжної системи;
- buyer_id (BigInt, зовнішній ключ) – посилання на покупця з таблиці користувачів.

Таблиця адрес доставки (addresses) – фіксує координати для відправки

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

картин покупцеві:

- id (BigInt, первинний ключ) – унікальний ідентифікатор адреси;
- country (String) – країна призначення доставки;
- city (String) – місто доставки;
- street (String) – вулиця та номер відділення або будинку;
- postal_code (String) – поштовий індекс;
- order_id (BigInt, зовнішній ключ, унікальне) – зв'язок один-до-одного із

замовленням з таблиці orders.

Таблиця елементів замовлення (order_items) – зв'язує замовлення із придбаними картинами та фіксує вартість на момент транзакції:

- id (BigInt, первинний ключ) – унікальний ідентифікатор рядка;
- fixed_price (Decimal) – фіксована ціна картини на момент покупки (для збереження звітності);
- order_id (BigInt, зовнішній ключ) – посилання на батьківське замовлення;
- artwork_id (BigInt, зовнішній ключ) – посилання на придбаний художній твір.

Таблиця кошика (cart_items) – зберігає тимчасовий вибір користувачів перед покупкою:

- id (BigInt, первинний ключ) – унікальний ідентифікатор;
- user_id (BigInt, зовнішній ключ) – зв'язок із профілем клієнта;
- artwork_id (BigInt, зовнішній ключ) – посилання на доданий до кошика твір.

У межах розробленої бази даних було реалізовано такі логічні зв'язки між сутностями:

- Зв'язок один-до-багатьох (One-to-Many) побудовано між користувачами (User) та картинами (Artwork), замовленнями (Order), оскільки один користувач може бути автором багатьох робіт чи здійснювати багато покупок. Аналогічний зв'язок діє між Category та Artwork, а також між Artwork та Image.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

– Зв'язок один-до-одного (One-to-One) реалізовано між замовленням (Order) та адресою доставки (Address), що гарантує наявність лише однієї точки доставки для конкретної покупки.

– Зв'язок багато-до-багатьох (Many-to-Many) спроектовано між Artwork та Material, а також Artwork та Tag. Це дозволяє гнучко додавати до картини кілька матеріалів та необмежену кількість тегів для розширеного пошуку.

Впроваджена структура реляційних зв'язків забезпечує високу цілісність даних та унеможливлює виникнення логічних колізій під час паралельної роботи багатьох користувачів.

2.4 Програмування вебплатформи

2.4.1 Написання клієнтської частини

Для написання скриптів та реалізації логіки клієнтської частини проєкту GalleryHub було використано середовище програмування Visual Studio Code. Завдяки розширеній підтримці мови TypeScript цей редактор дозволив забезпечити строгу типізацію та спростити пошук потенційних помилок у процесі розробки. Це суттєво підвищило якість коду. Усі додаткові пакети та сторонні бібліотеки інтегрувалися через менеджер пакетів npm.

На початковому етапі розробки клієнтського застосунку у файлі main.tsx було реалізовано підключення кореневого компонента до DOM-вузла для ініціалізації всього дерева інтерфейсу. Для забезпечення глобального доступу до стану авторизації та метаданих сторінок основний додаток обгорнуто в провайдери Helmet та BrowserRouter.

Програмний код ініціалізації наведено далі в лістингу 2.4.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

```

import React from 'react'
import ReactDOM from 'react-dom/client'
import { BrowserRouter } from 'react-router-dom'
import { HelmetProvider } from 'react-helmet-async'
import App from './App.tsx'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <HelmetProvider>
      <BrowserRouter>
        <App />
      </BrowserRouter>
    </HelmetProvider>
  </React.StrictMode>,
)

```

Лістинг 2.4 – Ініціалізація кореневого компонента у файлі main.tsx

Як демонструє лістинг 2.4, ініціалізація відбувається за допомогою методу `createRoot`, який прив'язує дерево React-компонентів до єдиного статичного елемента HTML. Початкову структуру папок клієнтського проєкту, яка містить логічно розділені директорії, наведено на рисунку 2.20.

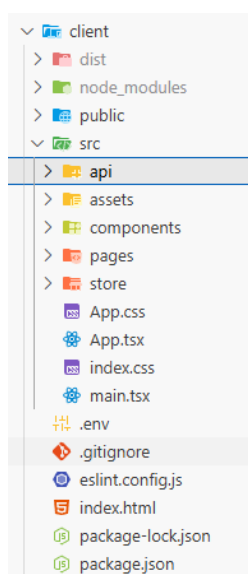


Рисунок 2.20 – Структура папок клієнтської частини проєкту GalleryHub

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення надійної взаємодії з асинхронним сервером було налаштовано централізований екземпляр клієнта Axios із автоматичним прикріпленням JWT-токенів авторизації. Це дозволяє уникнути потреби повторного написання заголовків під час кожного HTTP-запиту. Програмний код налаштування Axios наведено далі в лістингу 2.5.

```
import axios from 'axios';
const api = axios.create({
  baseURL: import.meta.env.VITE_API_URL || 'http://localhost:8080/api',
  headers: {
    'Content-Type': 'application/json',
  },
});
api.interceptors.request.use(
  (config) => {
    const token = localStorage.getItem('galleryhub_token');
    if (token) {
      config.headers.Authorization = `Bearer ${token}`;
    }
    return config;
  },
  (error) => {
    return Promise.reject(error);
  }
);
export default api;
```

Лістинг 2.5 – Конфігурація клієнта Axios у файлі axios.ts

Як демонструє лістинг 2.5, у системі налаштовано перехоплювач (interceptor), який перед відправкою кожного запиту зчитує токен користувача із локального сховища. Це гарантує автоматичну автентифікацію на сервері. Для захисту внутрішніх сторінок від неавторизованого доступу в App.tsx розроблено

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

механізм приватних маршрутів (PrivateRoute). При цьому повний код компонента налаштування маршрутизації та відображення сторінок App.tsx представлений в додатку А пояснювальної записки.

Логіка управління станом кошика базується на використанні Zustand-сховища, яке дозволяє об'єднувати товари при додаванні з картки та автоматично перераховувати загальну суму [9]. При цьому повний код сховища стану кошика cartStore представлений в додатку Д пояснювальної записки для детального аналізу логіки збереження даних.

2.4.2 Написання серверної частини

Для створення програмного ядра платформи GalleryHub було обрано середовище виконання Node.js разом із швидким фреймворком Express.js для побудови архітектури REST API. Це забезпечило масштабованість. Завдяки використанню моделі асинхронної обробки запитів вдалося досягти високої швидкості сервера під час одночасної роботи багатьох відвідувачів із зображеннями шедеврів. В основу архітектури покладено модульність.

Для розгортання робочого місця було проведено налаштування проєкту та встановлено повний набір програм через пакетний менеджер для керування сторонніми бібліотеками. Всі параметри вказані у package.json. Конфігурацію серверного модуля у файлі package.json із переліком підключених залежностей представлено на рисунку 2.21.

```
server > package.json > ...
20  "dependencies": {
21    "@prisma/adapters-pg": "^7.8.0",
22    "@prisma/client": "^7.8.0",
23    "bcryptjs": "^3.0.3",
24    "cloudinary": "^2.10.0",
25    "cors": "^2.8.6",
26    "dotenv": "^17.4.2",
27    "express": "^5.2.1",
28    "jsonwebtoken": "^9.0.3",
29    "multer": "^2.1.1",
30    "pg": "^8.21.0",
31    "prisma": "^7.8.0",
32    "tsx": "^4.22.3",
33    "zod": "^4.4.3"
34  },
```

Рисунок 2.21 – Конфігурація серверного модуля у файлі package.json

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

Головним вузлом ініціалізації сервера є файл `index.ts`, який імпортує конфігураційні змінні оточення, підключає проміжне програмне забезпечення CORS та налаштовує точки входу REST API. Програмний код ініціалізації Express сервера наведено далі в лістингу 2.6.

```
import express from 'express';
import cors from 'cors';
import dotenv from 'dotenv';
import prisma from './utils/prisma.js';
import './utils/serialize.js';
import authRoutes from './routes/auth.js';
import artworkRoutes from './routes/artwork.js';
import userRoutes from './routes/user.js';
import orderRoutes from './routes/order.js';
import metadataRoutes from './routes/metadata.js';
import artistRoutes from './routes/artist.js';
dotenv.config();
const app = express();
const PORT = process.env.PORT || 8081;
app.use(cors());
app.use(express.json());
app.use('/api/auth', authRoutes);
app.use('/api/artworks', artworkRoutes);
app.use('/api/users', userRoutes);
app.use('/api/orders', orderRoutes);
app.use('/api/metadata', metadataRoutes);
app.use('/api/artists', artistRoutes);
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

Лістинг 2.6 – Ініціалізація Express.js сервера у файлі `index.ts`

Як демонструє лістинг 2.6, усі запити перенаправляються на відповідні

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

спеціалізовані маршрутизатори для забезпечення логічної ізоляції окремих функцій. Внутрішню структуру папки маршрутів представлено на рисунку 2.22.

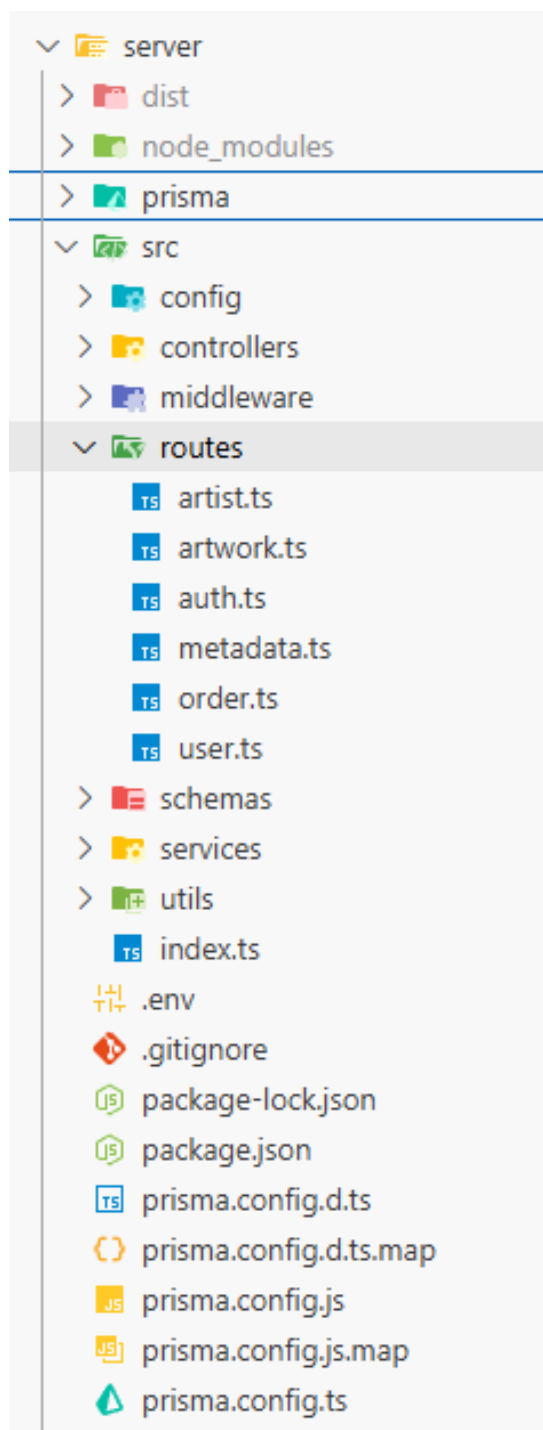


Рисунок 2.22 – Структура модулів маршрутизації серверної частини GalleryHub

Для забезпечення надійного захисту облікових записів використано пакет bcryptjs, який перетворює паролі у складні хеш-коди перед їх збереженням. Такий

підхід виключає витік. Особливу увагу приділено механізму перевірки осіб, який базується на веб-токенах JWT для створення захищених сесій. Для контролю доступу розроблено проміжні обробники `authenticate` та `authorize`, код яких наведено в лістингу 2.7.

```
import { Request, Response, NextFunction } from 'express';
import jwt from 'jsonwebtoken';

const JWT_SECRET = process.env.JWT_SECRET || 'secret';

export const authenticate = (req: AuthRequest, res: Response, next:
NextFunction) => {
  const authHeader = req.headers.authorization;
  if (!authHeader || !authHeader.startsWith('Bearer ')) {
    return res.status(401).json({ error: 'Unauthorized' });
  }
  const token = authHeader.split(' ')[1];
  try {
    const decoded = jwt.verify(token, JWT_SECRET) as { userId: string;
role: string };
    req.user = decoded;
    next();
  } catch (error) {
    return res.status(401).json({ error: 'Invalid token' });
  }
};
```

Лістинг 2.7 – Проміжний обробник перевірки авторизації `auth.ts`

Як показує лістинг 2.7, обробник зчитує токен `Bearer` із заголовків запиту та виконує його перевірку.

Якщо підпис невірний, система блокує виконання запиту. Це гарантує надійний захист внутрішніх ресурсів платформи від несанкціонованого доступу.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

2.5 Тестування вебплатформи

Задля підтвердження повної працездатності всіх розроблених модулів та перевірки їх відповідності технічному завданню було проведено комплексне тестування всієї платформи GalleryHub.

Основним завданням цього етапу стало виявлення та своєчасне усунення потенційних дефектів у логіці обробки даних. Система пройшла повну перевірку надійності. Додатково досліджено візуальне відображення всіх елементів інтерфейсу на різних екранах.

Під час верифікації функціоналу було застосовано декілька видів тестування, що дало змогу охопити всі аспекти роботи вебресурсу. Зокрема, було реалізовано такі підходи:

- Функціональне тестування. Перевірено коректність роботи всієї системи автентифікації, процесів завантаження всіх зображень художниками та механізмів формування замовлення.

- Тестування інтерфейсу (UI/UX). Досліджено точність відтворення розроблених макетів, плавність анімацій та зручність навігації між розділами галереї.

- Кросбраузерне та адаптивне тестування. Виконано перевірку відображення сторінок у популярних браузерях Google Chrome, Mozilla Firefox та Safari на пристроях із різною діагоналлю екрана.

- Тестування безпеки. Перевірено стійкість системи до неавторизованого доступу до адмін-панелі та коректність роботи всіх посередників для захисту приватних даних.

Задля систематизації процесу перевірки було розроблено детальний контрольний список (checklist), який містить усі ключові параметри для повної оцінки якості розробленого програмного продукту. Основні пункти перевірки наведено у таблиці 2.1.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

Таблиця 2.1 – Розширений контрольний список результатів тестування GalleryHub

Об'єкт перевірки	Очікуваний результат	Статус
Реєстрація нового користувача	Створення запису в базі даних	Успішно
Завантаження робіт художником	Поява фото у хмарному сховищі	Успішно
Фільтрація в каталозі	Миттєве оновлення списку товарів	Успішно
Оформлення замовлення	Фіксація адреси та суми в БД	Успішно
Адаптивність меню	Коректне згортання на смартфонах	Успішно
Авторизація адміністратора	Доступ до функціоналу модерації	Успішно
Оновлення даних профілю	Збереження змін у базі даних	Успішно
Перевірка завантаження картини	Коректне відображення прев'ю зображень	Успішно
Робота з тегами	Додавання та видалення тегів	Успішно
Створення seed-даних	Наповнення БД базовими категоріями та юзерами	Успішно

Процес тестування включав імітацію критичних навантажень та введення некоректних даних у форми для пошуку проблем. Зокрема, було перевірено реакцію системи на спроби завантаження файлів великого об'єму або вводу неправильного формату електронної пошти. Завдяки впровадженій валідації Zod усі помилки обробляються коректно. Користувач отримує зрозумілі підказки щодо вирішення проблеми.

Покроковий опис процесу тестування функціоналу GalleryHub:

1) Реєстрація нового користувача:

Для початку тестування необхідно зареєструвати нового користувача.

– Відкрито головну сторінку GalleryHub.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

- Натиснуто кнопку «Увійти» та обрано «Зареєструватися».
- Заповнено поля «Ім'я», «Прізвище», «Електронна адреса» та «Пароль» коректними даними.

- Натиснуто кнопку «Створити акаунт».

Очікується успішне створення нового облікового запису в системі. Користувач автоматично перенаправляється на головну сторінку, як це зображено на рисунку 2.23.

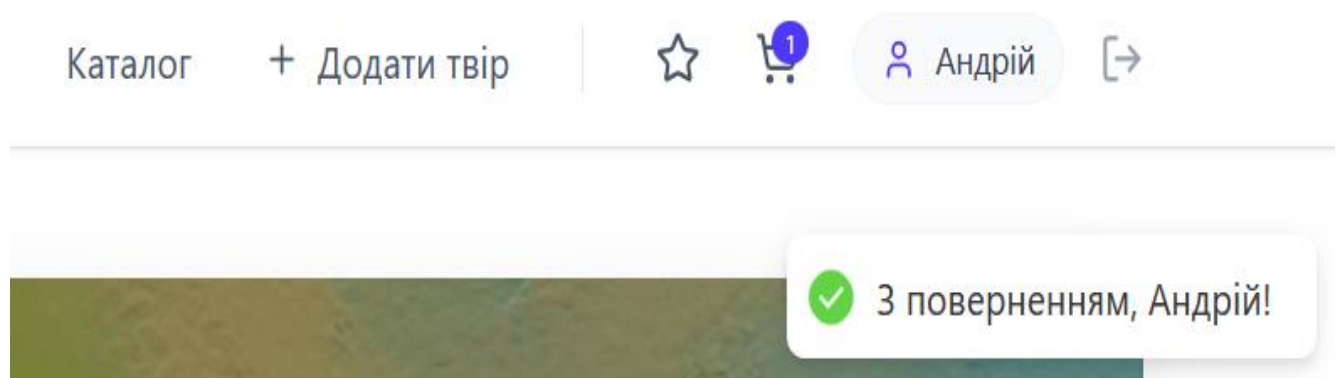


Рисунок 2.23 – Процес успішної реєстрації нового користувача

2) Публікація нового твору мистецтва:

Після реєстрації користувач повинен мати можливість додати свій перший твір.

- Здійснено вхід під новим обліковим записом художника.
- Натиснуто кнопку «Додати твір» у навігаційному меню.
- Заповнено форму: «Назва», «Опис», «Ціна», «Ширина», «Висота», обрано «Категорію» та «Матеріал».
- Введено декілька тегів, використовуючи функціонал підказок.
- Завантажено 3-5 зображень твору мистецтва.
- Натиснуто кнопку «Надіслати твір».

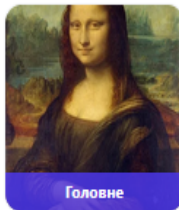
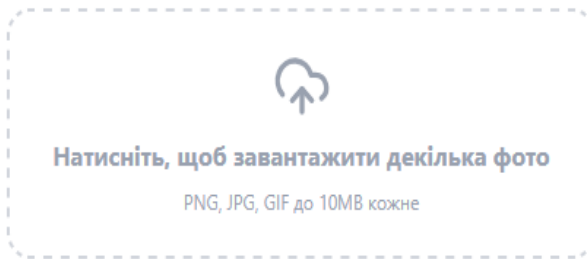
Очікується повідомлення про успішне надсилання твору на модерацію, форму заповнення якого наведено на рисунку 2.24.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Надішліть свій шедевр

Поділіться своїм мистецтвом зі світом. Заповніть деталі нижче.

Зображення твору (до 10 шт)



Назва

Т напр., Зоряна ніч

Опис

Опишіть свій твір...

Ціна (ГРН)

\$ 100

Ширина (см)

50

Висота (см)

70

Категорія

Виберіть категорію

Матеріал

Виберіть матеріал

Теги твору

Введіть тег та натисніть Enter

Виберіть зі списку або введіть нові через Enter

Надіслати твір

Рисунок 2.24 – Форма додавання нового твору художником

Після успішного відправлення роботи користувач переходить до особистого кабінету. Додана робота з'являється у профілі художника зі статусом «Очікує», що свідчить про її перебування на етапі перевірки перед подальшою публікацією. Таким чином, користувач має змогу контролювати стан поданої роботи безпосередньо у своєму профілі.

Приклад відображення доданої роботи наведено на рисунку 2.3.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Мої твори

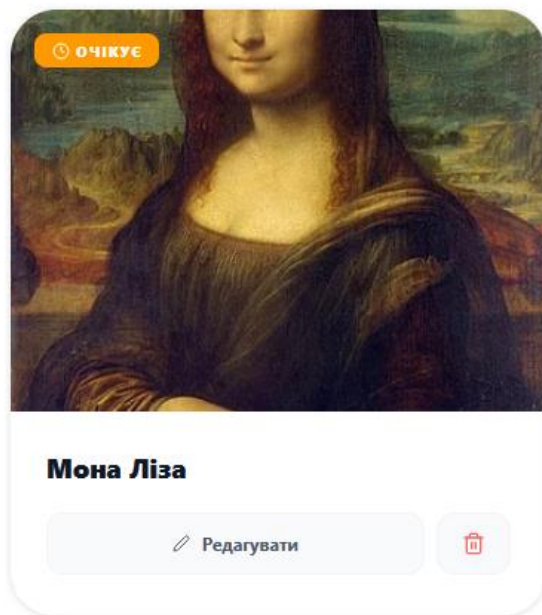


Рисунок 2.25 – Картка картини зі статусом «Очікує» у профілі автора

3) Перевірка твору адміністратором та його схвалення:

Процес перевірки заявки та інтерфейс прийняття рішення адміністратором показано на рисунку 2.26.

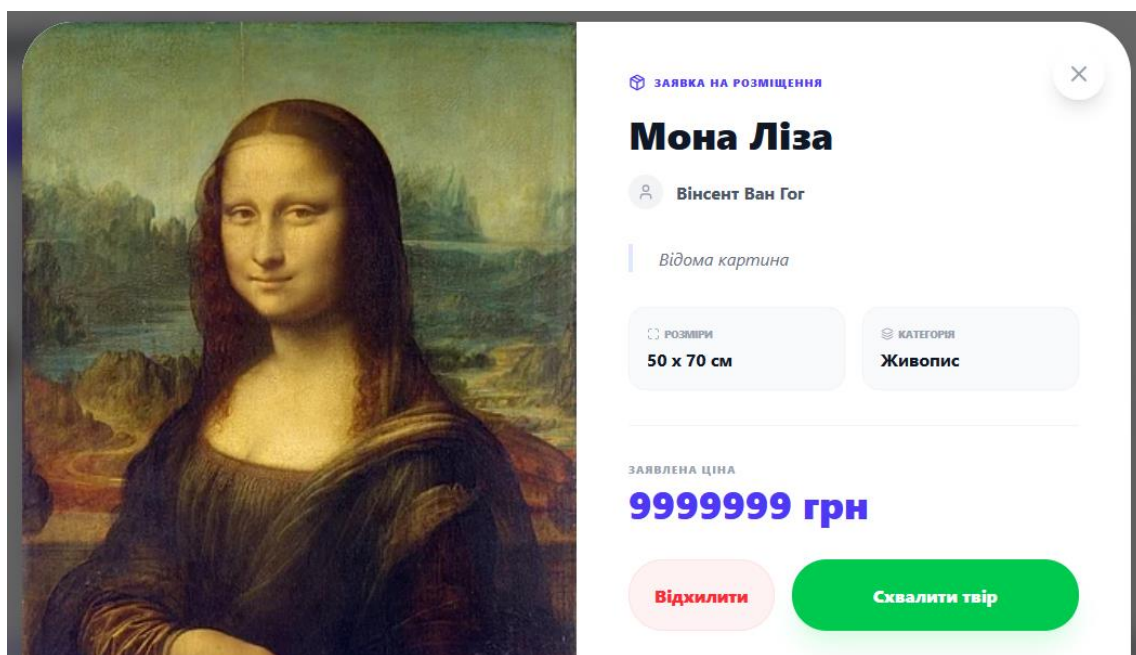


Рисунок 2.26 – Інтерфейс модерації та схвалення твору адміністратором

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

Очікується, що твір змінить статус на «Схвалено» та буде доступний у загальному каталозі. Зміна статусу картки твору на «Схвалено» представлена на рисунку 2.27.

Мої твори

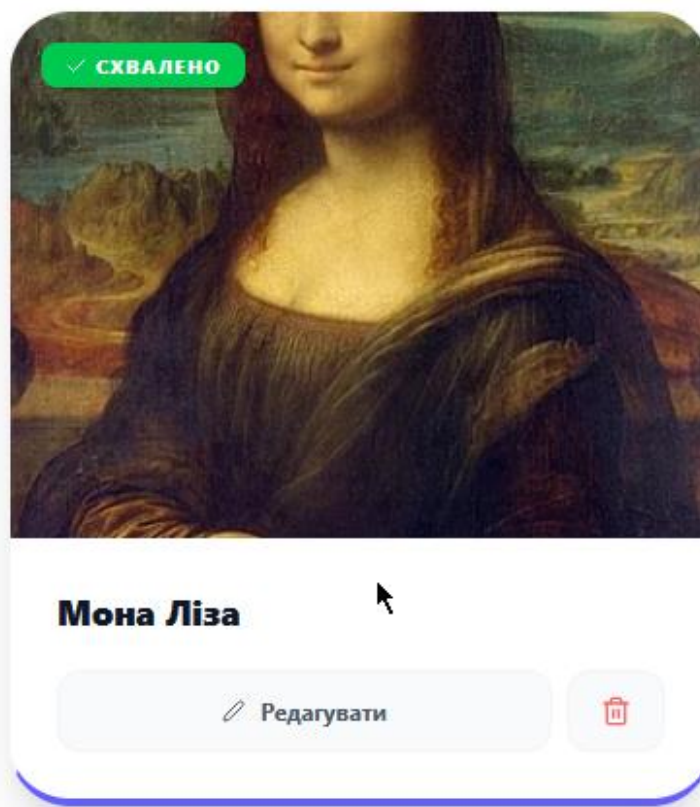


Рисунок 2.27 – Картка схваленого твору із зеленим маркером статусу

4) Перевірка наявності твору в каталозі:

Фінальна перевірка полягає у відображенні твору для всіх користувачів.

- Здійснено вихід з облікового запису адміністратора або перехід на головну сторінку як гість.
- Перейдено до розділу «Галерея» та використано фільтри або пошук для знаходження доданого твору.

Очікується, що доданий твір мистецтва відображається у загальному каталозі та доступний для перегляду всім відвідувачам платформи, що зображено на рисунку 2.28.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54



Набір картин ... 6000 грн

Андрій С
60х90 см

НІЧНЕ-НЕБО ПОЛЕ

живопис

Деталі



Мона Ліза 9999999 грн

Вінсент Ван Гог
50х70 см

живопис

Деталі

Рисунок 2.28 – Відображення схваленого твору в загальному каталозі

Додаткові методи тестування:

– Тестування продуктивності. Здійснено вимірювання швидкості завантаження сторінок та часу відповіді сервера при одночасному збільшенні кількості користувачів. Використано інструменти Google Lighthouse та PageSpeed Insights. Результати показали високу оптимізацію ресурсів.

– Тестування безпеки (Security Testing). Проведено аналіз на наявність поширених вразливостей (SQL Injection, XSS) за допомогою автоматизованих сканерів OWASP ZAP. Завдяки валідації Zod та механізмам JWT усі ризики вдалося мінімізувати.

– Тестування зручності використання (Usability Testing). Проведено оцінку інтуїтивності інтерфейсу та простоти виконання ключових завдань (пошук, додавання в кошик) з фокус-групою. Зафіксовано високий рівень задоволеності користувачів.

Зм.	Арк.	№ докум.	Підпис	Дата

– Тестування доступності (Accessibility Testing). Перевірено відповідність вебстандартам WCAG 2.1 для людей з обмеженими можливостями за допомогою інструментів Chrome DevTools Lighthouse. Система отримала високу оцінку.

Під час перевірки продуктивності було встановлено, що використання Vite та оптимізованих запитів Prisma забезпечує високу швидкість завантаження контенту навіть при низькій швидкості інтернет-з'єднання. Платформа продемонструвала стабільність. Результати проведених заходів дозволяють зробити висновок про повну готовність GalleryHub до експлуатації. Всі виявлені дрібні недоліки були усунені під час фінальної ітерації налагодження та перед публікацією.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розгортання вебплатформи

Для забезпечення загальнодоступності розроблюваної вебплатформи GalleryHub у глобальній мережі інтернет виконайте комплекс робіт з її розгортання на сучасних хмарних ресурсах, що надають надійні послуги хостингу. У процесі проєктування сучасних розподілених систем особливу увагу приділяють багаторівневій архітектурі, яка дозволяє масштабувати клієнтську та серверну частини незалежно [1] Для цього запуск всієї системи здійснить шляхом розділення її на окремі логічні компоненти, включаючи інтерфейс користувача та серверний код, та розміщення кожного вузла на відповідному хмарному сервісі. Завдяки цьому гарантується висока стабільність. Під час вибору сховища даних для вебплатформи створіть реляційну базу даних PostgreSQL. Застосування СКБД PostgreSQL гарантує надійність транзакцій, збереження цілісності реляційних зв'язків та гнучкість при роботі зі складними запитамі покупців [5]. яка успішно функціонує в межах безкоштовного хмарного тарифу платформи Supabase. Використання платформи Supabase як хостингу дозволяє отримати налаштований та захищений віддалений сервер PostgreSQL без необхідності тривалого ручного адміністрування операційної системи [6].

На першому етапі підготуйте сховище інформації для платформи GalleryHub в особистому кабінеті користувача на офіційному вебсайті сервісу Supabase, який дозволяє швидко створювати нові інстанси баз даних. Спочатку відкрийте вікно авторизації на сайті, здійснить вхід через особистий репозиторій GitHub та натисніть кнопку створення нової організації для керування проєктами. Зазвичай така процедура виконується досить швидко. Після цього введіть назву для створюваної бази даних, виберіть найближчий географічний регіон розташування сервера та обов'язково встановіть надійний пароль для подальшого доступу.

Для підключення серверного коду до бази даних відкрийте розділ

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

налаштувань проєкту, перейдіть на вкладку параметрів з'єднання та скопіюйте повний рядок підключення URI, що містить пароль. Збережіть отримані реквізити доступу у локальному файлі конфігурації для подальшого використання при налаштуванні змінних середовища серверного застосунку на хостингу Render. Завдяки цьому значно спрощуються всі наступні кроки. Зокрема, детальний вигляд головного екрана користувача з параметрами підключення до створеної бази даних на платформі Supabase наведено на відповідному рисунку 3.1.

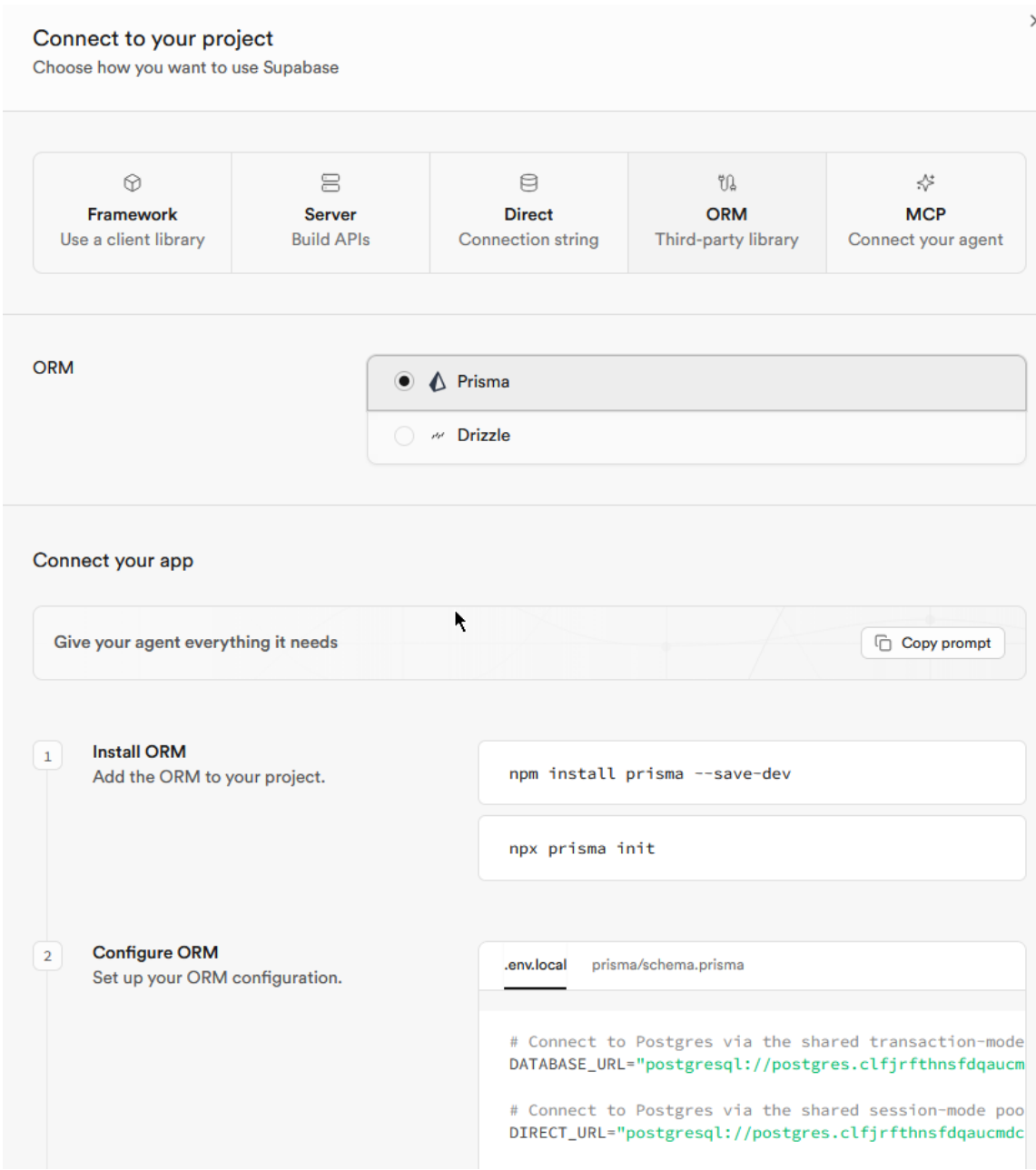


Рисунок 3.1 – Параметри підключення до бази даних у кабінеті Supabase

Для публікації створеного бекенд-сервісу відкрийте панель керування хмарного хостингу Render та створіть новий ресурс типу Web Service для забезпечення роботи фонового процесу Node.js. Асинхронна модель обробки подій у Node.js у поєднанні із фреймворком Express.js забезпечує високу пропускну здатність сервера при великій кількості одночасних підключень покупців [4]. Під час налаштування та створення сервісу обов'язково надайте Render доступ до вашого репозиторію на GitHub та виберіть правильну робочу гілку для збірки коду. Виконання такої операції є повністю обов'язковим для розгортання. У полі команди для встановлення залежностей введіть команду встановлення пакетів, а для запуску сервера запустіть відповідний скрипт компіляції та подальшої ініціалізації бекенду.

Далі перейдіть у спеціальний розділ «Environment Variables» на панелі керування створеним вебсервісом для додавання необхідних змінних середовища нашого серверного додатка, які керують підключенням. У цьому блоці обов'язково введіть отриманий раніше рядок підключення до Supabase та секретні ключі для авторизації користувачів за технологією JWT. Використання строгої типізації TypeScript при роботі із запитами авторизації та передачі токенів дозволяє мінімізувати виникнення помилок невідповідності форматів на етапі виконання [2]. Таким налаштуванням надійно забезпечується безпека збережених даних. Також додатково введіть конфігураційні дані для відображення медіафайлів та повної інтеграції з хмарним сховищем Cloudinary, що забезпечує збереження картинок художників. Інтеграція хмарного сховища Cloudinary дозволяє автоматично оптимізувати розміри та формати завантажених художниками зображень, суттєво прискорюючи завантаження сторінок [8]. Зокрема, конфігурація основних параметрів для успішного запуску та стабільної роботи серверної частини платформи представлена на відповідному рисунку 3.2.

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

Branch

The Git branch to build and deploy.

main

Edit

Root Directory Optional

If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a [monorepo](#).

server

Edit

Build Command

Render runs this command to build your app before each deploy.

server/ \$ npm install && npx prisma generate && npx prisma db push && n

Edit

Рисунок 3.2 – Налаштування Web Service для серверної частини на платформі Render

На завершальному етапі загального процесу перейдіть до розгортання клієнтської частини нашої платформи та створіть у хмарі Render новий Static Site для швидкої роздачі статичних сторінок. Під час початкового налаштування вкажіть шлях до репозиторію на GitHub, виберіть робочу гілку та пропишіть команду збірки продуктової версії React-додатка з використанням пакетного менеджера. Компонентно-орієнтована структура інтерфейсу дозволяє розділити складні сторінки галереї на окремі незалежні блоки із власним життєвим циклом [3]. Зі сторони сервісу цей процес є повністю автоматизованим. Для успішного створення файлів запусіть стандартну команду збірки Vite, а як цільову папку для публікації результату введіть каталог dist.

Для забезпечення правильної взаємодії клієнтської частини з бекендом відкрийте вкладку змінних середовища та введіть адресу нашого віддаленого сервера у глобальній мережі інтернет. Для успішного вирішення проблем із маршрутизацією на сторінці налаштувань Static Site додайте спеціальне правило перенаправлення всіх запитів на головний файл index.html для уникнення помилок 404. Така дія дозволяє повністю запобігти можливим помилкам.

Зокрема, фінальний стан інтерфейсу панелі керування після успішної побудови та публікації нашого статичного сайту зображено на відповідному рисунку 3.3.

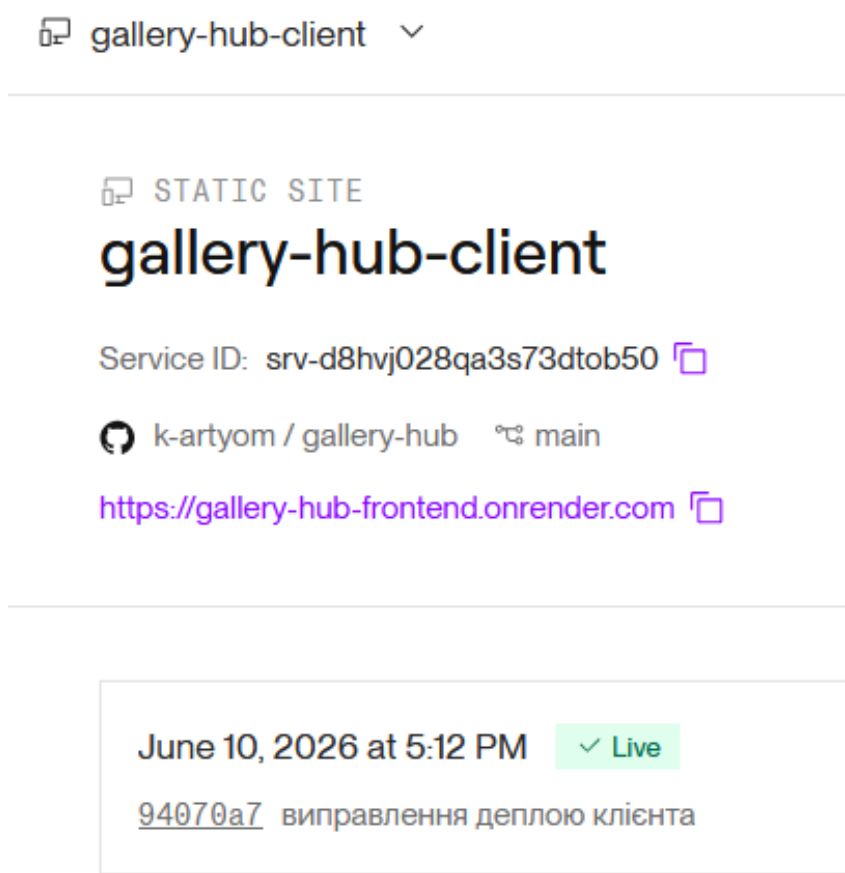


Рисунок 3.3 – Інтерфейс успішного розгортання статичного сайту

Після завершення всіх етапів налаштування та публікації здійсніть комплексну перевірку працездатності опублікованого в мережі інтернет ресурсу за отриманим посиланням, яке надає Render. Для перевірки коректності інтеграції виконайте тестову реєстрацію нового користувача, здійсніть завантаження зображення у галерею та запустіть симуляцію процесу купівлі картини в реальному часі. У результаті всі інтегровані функції працюють у штатному режимі. У результаті виконаних дій запустіть готову платформу GalleryHub, яка тепер повністю готова до тривалої роботи в реальних умовах її функціонування в мережі інтернет.

3.2 Інструкція з наповнення вебплатформи

Для забезпечення підтримання актуальності інформації на платформі GalleryHub та зручного процесу наповнення каталогу виконайте комплекс робіт з її регулярного обслуговування, що гарантує безперебійність. Спочатку запустіть автоматичне первинне заповнення бази даних необхідною технічною інформацією за допомогою вбудованого скрипта заповнення, що є дуже зручним для розробника. Завдяки цьому значно спрощується налаштування. Зокрема, під час першого запуску системи запустіть команду «`npm run prisma:seed`», яка автоматично додає перелік базових категорій живопису та типів матеріалів до бази. Інструмент Prisma ORM спрощує взаємодію з даними через автоматично згенерований Prisma Client, що дозволяє виконувати запити без ручного написання SQL-коду [7].

Наступним важливим кроком для наповнення системи є публікація художниками своїх нових картин у загальному каталозі вебплатформи через особистий кабінет користувача, що приваблює колекціонерів. Для цього кожен зареєстрований автор повинен пройти процедуру авторизації, щоб отримати повний доступ до спеціальної форми завантаження необхідних медіафайлів на сервер. Ця дія є повністю обов'язковою. Після успішного входу в систему обов'язково відкрийте вкладку особистого профілю та натисніть кнопку для додавання нового твору мистецтва у каталог.

Для додавання нових картин користувачем (художником) виконайте такі кроки:

- Спочатку відкрийте форму додавання нового арт-об'єкта, яка містить різноманітні обов'язкові для заповнення текстові та числові поля введення інформації.
- Введіть назву картини, детальний опис її художньої концепції, оберіть категорію мистецтва, вкажіть використані матеріали та фізичні габарити полотна у сантиметрах.
- Після цього завантажте зображення твору високої роздільної здатності,

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

обравши один основний головний файл та додаткові ракурси для демонстрації дрібних деталей.

– Зі списку запропонованих системою варіантів оберіть відповідні ключові теги для полегшення майбутнього пошуку та натисніть кнопку публікації заповненої форми.

Вигляд форми додавання картини зображено на рисунку 3.4.

Надішліть свій шедевр

Поділіться своїм мистецтвом зі світом. Заповніть деталі нижче.

Зображення твору (до 10 шт)

Натисніть, щоб завантажити декілька фото
PNG, JPG, GIF до 10MB кожне

Головне

Назва
Т напр., Зоряна ніч

Опис
Опишіть свій твір...

Ціна (ГРН)
\$ 100

Ширина (см)
50

Висота (см)
70

Категорія
Виберіть категорію

Матеріал
Виберіть матеріал

Теги твору
Введіть тег та натисніть E

Виберіть зі списку або введіть нові через Enter

Надіслати твір

Рисунок 3.4 – Процес додавання нового шедевра через інтерфейс художника

Зі сторони адміністратора платформи керування контентом здійснюється через спеціалізовану панель керування з розширеними правами доступу до реляційної бази даних нашого вебдодатка. Для виконання будь-яких адміністративних операцій спочатку пройдіть обов'язковий процес авторизації під обліковим записом, що має відповідну роль адміністратора системи у базі. Ця дія є повністю обов'язковою. Після успішного входу перейдіть до розділу

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

адміністрування та відкрийте вкладку керування каталогом наявних творів мистецтва для оновлення інформації.

Для додавання нової картини адміністратором виконайте такі кроки:

- Спочатку відкрийте вкладку каталогу в адмін-панелі та натисніть кнопку створення нового арт-об'єкта для заповнення картки.
- Введіть назву картини, оберіть автора зі списку зареєстрованих художників та обов'язково вкажіть категорію, матеріали й розміри роботи для бази.
- Після цього завантажте медіафайли через інтегрований сервіс Cloudinary та за потреби встановіть первинний статус перевірки роботи як схвалений.
- Введіть необхідні цінові параметри для продажу картини на платформі та натисніть кнопку збереження створеного запису в реляційній базі даних.

Для полегшення навігації та детальної фільтрації творів мистецтва на платформі GalleryHub використовується динамічна система ключових слів, що створює комфорт. Додавання нових тегів до системи здійснюється виключно адміністратором через спеціальну форму керування метаданими в загальній адмін-панелі нашого ресурсу. Ці процедури виконуються дуже просто. Після переходу до відповідного розділу відкрийте інтерфейс додавання міток та введіть назву нового унікального тегу в базу.

Для створення та додавання нового тегу в базу виконайте такі дії:

- Спочатку відкрийте вкладку «Управління тегами» в головному меню панелі адміністратора нашої вебплатформи.
- Введіть текстову назву тегу в полі введення з урахуванням регістру символів та обов'язково додайте короткий опис для нової мітки.
- Натисніть кнопку збереження для успішного запису нового тегу в базу даних та його відображення в загальному списку категорій.
- Запустіть перевірку відображення нового тегу у загальному списку фільтрації на сторінці галереї творів.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

Під час обслуговування платформи особлива увага приділяється модерації завантажених художниками творів для недопущення публікації неякісного контенту у каталозі. Усі нові роботи після заповнення форми користувачами автоматично отримують початковий статус очікування перевірки в нашій базі даних. Ця процедура є повністю обов'язковою. Після цього адміністратор повинен уважно переглянути надіслану роботу та прийняти остаточне рішення про її публікацію або відхилення з поясненням.

Для здійснення модерації нових картин виконайте такі кроки:

- Спочатку перейдіть до розділу «Панель адміністратора» та відкрийте вкладку «Перевірка творів» для перегляду списку нових заявок.
- Ознайомтеся з усіма деталями надісланої заявки, включаючи технічні характеристики та якість завантажених фотографій.
- Натисніть кнопку «Схвалити» для публікації роботи у каталозі або кнопку «Відхилити» у разі виявлення можливих порушень правил.
- Запустіть автоматичне сповіщення користувача про зміну статусу його роботи та перевірте її появу на нашому вебсайті.

3.3 Інструкція з популяризації та підтримки вебплатформи

Для забезпечення сталого розвитку платформи GalleryHub та залучення широкого кола митців було сформовано комплексну стратегію популяризації розробленого програмного продукту на ринку цифрового мистецтва. Основним завданням цього етапу є створення впізнаваного бренду, який викликає довіру у колекціонерів та авторів. Маркетингові заходи є необхідними.

Процес розповсюдження інформації про платформу базується на використанні сучасних каналів цифрової комунікації. Зокрема, було виділено такі пріоритетні напрямки для просування ресурсу:

- Інтеграція з візуальними соціальними мережами. Завдяки високій якості зображень у каталозі GalleryHub, роботи художників ідеально підходять для поширення в Instagram та Pinterest.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

– Партнерство з художніми навчальними закладами. Було запропоновано механізм залучення студентів творчих спеціальностей для публікації їхніх перших виставок на базі платформи. Це стимулює ріст.

– Використання вбудованої системи розсилок. Модуль підписки в нижній частині сайту дозволяє збирати базу потенційних покупців та повідомляти їх про нові надходження або інтерв'ю з майстрами.

Під час популяризації особливу увагу слід приділити роботі з професійними спільнотами в мережах LinkedIn та Facebook. Це дозволить залучити серйозних інвесторів та галереї для співпраці на корпоративному рівні. Ефект буде позитивним.

Важливою частиною життєвого циклу проєкту є його технічна підтримка та регулярне оновлення компонентів. Для стабільної роботи платформи в довгостроковій перспективі було розроблено такі рекомендації:

– Моніторинг ресурсів Cloudinary. Необхідно періодично перевіряти обсяги використаного трафіку та дискового простору для запобігання зупинки завантаження нових шедеврів.

– Регулярне оновлення залежностей. Використання TypeScript вимагає вчасної актуалізації типів та бібліотек безпеки для захисту від нових видів кіберзагроз.

– Резервне копіювання бази даних. Хостинг Render дозволяє налаштувати автоматичні бекапи PostgreSQL, що гарантує відновлення інформації у разі технічних збоїв.

Для підвищення лояльності користувачів передбачено впровадження системи технічної допомоги через електронну пошту. Всі звернення повинні оброблятися адміністраторами протягом робочого дня. Платформа GalleryHub має потенціал стати лідером у своєму сегменті завдяки поєднанню зручного інтерфейсу та продуманої стратегії підтримки. Постійний розвиток функціоналу забезпечить конкурентоспроможність проєкту.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини кваліфікаційної роботи є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки вебплатформи для експозиції та реалізації творів мистецтва «GalleryHub», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є клієнт-серверна вебплатформа «GalleryHub».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки вебплатформи «GalleryHub». Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Керівник проєкту	10
		Фронтенд-розробник	5
		Бекенд-розробник	5
2	Розробка технічного завдання	Керівник проєкту	6
		Фронтенд-розробник	4
		Бекенд-розробник	4
3	Дизайн інтерфейсу	Фронтенд-розробник	20
4	Розробка функціоналу	Фронтенд-розробник	25
		Бекенд-розробник	35
5	Тестування та відладка	Фронтенд-розробник	10
		Бекенд-розробник	10
6	Документування	Керівник проєкту	4
		Фронтенд-розробник	2
		Бекенд-розробник	2
7	Розгортання та підтримка	Фронтенд-розробник	2
		Бекенд-розробник	4
8	Управління проєктом	Керівник проєкту	2
Разом			150

Сумарний час виконання операцій технологічного процесу становить 150 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки вебплатформи «GalleryHub»

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою

$$Z_{\text{осн.}} = T_c \cdot K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника проєкту – 450 гривень за годину, фронтенд-розробника – 250 гривень за годину, бекенд-розробника – 300 гривень за годину); K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

- Керівника проєкту $Z_{\text{осн.}} = 22\,450 = 9\,900 \text{ грн.}$
- Фронтенд-розробника $Z_{\text{осн.}} = 68\,250 = 17\,000 \text{ грн.}$
- Бекенд-розробника $Z_{\text{осн.}} = 60\,300 = 18\,000 \text{ грн.}$

Сумарна основна заробітна плата становить:

$$Z_{\text{осн.}} = 9\,900 + 17\,000 + 18\,000 = 44\,900 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

- Керівника проєкту $Z_{\text{дод.}} = 9\,900 \cdot 0,1 = 990 \text{ грн.}$
- Фронтенд-розробника $Z_{\text{дод.}} = 17\,000 \cdot 0,1 = 1\,700 \text{ грн.}$
- Бекенд-розробника $Z_{\text{дод.}} = 18\,000 \cdot 0,1 = 1\,800 \text{ грн.}$

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод.}} = 990 + 1\,700 + 1\,800 = 4\,490 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{\text{о.п.}}$) визначаються за формулою:

$$B_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$B_{\text{о.п.}} = 44\,900 + 4\,490 = 49\,390 \text{ грн.}$$

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$B_{\text{ЄСВ}} = B_{\text{оп}} * 0,22 \quad (4.4)$$

$$B_{\text{ЄСВ}} = 49\,390 * 0,22 = 10\,866 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/ п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Керівник проєкту	450	22	9 900	990	2 396	1 3286
2	Фронтенд- розробник	250	68	17 000	1 700	4 114	22 814
3	Бекенд- розробник	300	60	18 000	1 800	4 356	24 156
Разом				44 900	4 490	10 866	60 256

Отже, загальні витрати на оплату праці становлять 60 256 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_{\text{в}} = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії (приймаємо 15, 94 грн).

В нашій системі системі використовується комп'ютерне обладнання.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

Витрати на електроенергію обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,1 кВт/год.

$$Z_{\text{ек}} = 0,1 \cdot 150 \cdot 15,94 = 239 \text{ грн.}$$

Витрати на електроенергію становлять 239 грн.

4.4 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B * H_A}{100\%} * T, \quad (4.6)$$

де: А – амортизаційні відрахування за звітний період, грн.; Б_В – балансова вартість групи основних фондів на початок звітного періоду, грн.; Н_А – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 25000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{25\,000,00 * 0,04}{150} * 150 = 1000 \text{ грн.}$$

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = B_{o.п.} \cdot 0,2..0,6 \quad (4.7)$$

де: H_B – накладні витрати.

$$H_B = 49\,390 \cdot 0,4 = 19\,756 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості вебплатформи «GalleryHub»

Для складення кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат вебплатформи «GalleryHub»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	60 256	74
2.	Витрати на електроенергію	239	1
3.	Амортизаційні відрахування	1 000	1
4.	Накладні витрати	19 756	24
5.	Собівартість	81 251	100

Собівартість (СВ) НДР розрахуємо за формулою:

$$C_B = B_{o.п.} + B_{c.з.} + 3e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює СВ= 81 251 грн.

4.7 Розрахунок ціни розробки вебплатформи «GalleryHub»

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_v (1 + P_{рен}) (1 + ПДВ), \quad (4.9)$$

де: C_v – собівартість; $P_{рен.}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 81\,251 (1 + 0,3) (1 + 0,2) = 126\,752 \text{ грн.}$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -C_v + \sum_{i=1}^t \frac{\Gamma_{\Pi}}{(1+i)^t}, \quad (4.10)$$

де: C_v – собівартість розробки; Γ_{Π} – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$ЧТВ = -81251 + \frac{45\,500}{(1+0,1)^1} + \frac{45\,500}{(1+0,1)^2} + \frac{45\,500}{(1+0,1)^3} = 31\,617 \text{ грн}$$

Якщо $ЧТВ \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{H_v}{\Gamma_{пр}} \quad (4.11)$$

де: $T_{пв}$ – період до повного відшкодування витрат, років; H_v – невідшкодовані витрати на початок року, грн.; $\Gamma_{пр}$ – грошовий потік на початок року, грн.

$$T_{ок} = 2 + \frac{2592}{45500} = 2,1 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						73
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.5 – Техніко-економічні показники вебплатформа «GalleryHub»

№ п/п	Показник	Значення
1.	Собівартість, грн.	81 251
2.	Плановий прибуток або грошовий потік, грн.	45 500
3.	Ціна, грн.	126752
4.	Чиста теперішня вартість, грн.	31 617
5.	Термін окупності, рік	2,1

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка вебплатформи «GalleryHub» є доцільною з економічної точки зору.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Відповідальність за порушення законодавства про охорону праці.

Законодавство України передбачає сувору відповідальність за порушення нормативних актів з охорони праці. Відповідальність поділяється на кілька видів залежно від тяжкості порушення та його наслідків. Забезпечення безпечних умов праці є головним обов'язком роботодавця, а працівники повинні дотримуватися встановлених інструкцій [15].

Види відповідальності:

- дисциплінарна відповідальність застосовується до працівників у вигляді догани або звільнення за порушення трудової дисципліни;
- адміністративна відповідальність передбачає накладення штрафів на посадових осіб та працівників за порушення санітарних норм і правил безпеки;
- матеріальна відповідальність настає у разі заподіяння шкоди майну підприємства внаслідок порушення вимог охорони праці;
- кримінальна відповідальність застосовується у випадках, коли порушення правил безпеки призвело до серйозних наслідків або загибелі людей.

Кожен з цих видів відповідальності має свою специфіку та порядок застосування. Роботодавець несе відповідальність за нестворення безпечних умов праці, тоді як працівник відповідає за недотримання інструкцій та правил безпеки під час виконання своїх обов'язків. Регулярне навчання та інструктажі з охорони праці допомагають уникнути порушень та зменшити ризики настання відповідальності [15].

В таблиці 5.1 подано класифікацію видів відповідальності та їх санкцій.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

Таблиця 5.1 - Види відповідальності та санкції

Вид відповідальності	Суб'єкт правопорушення	Можливі санкції
Дисциплінарна	Працівники підприємства	Догана або звільнення
Адміністративна	Посадові особи та працівники	Штрафні санкції
Кримінальна	Особи, які порушили правила з тяжкими наслідками	Позбавлення волі, виправні роботи
Матеріальна	Особи, з вини яких завдано шкоду	Відшкодування збитків

Застосування конкретного виду відповідальності залежить від результатів розслідування нещасного випадку або виявленого правопорушення під час перевірки. Державна служба України з питань праці контролює дотримання законодавства та має право застосовувати адміністративні санкції до порушників. Належне виконання вимог охорони праці є запорукою збереження здоров'я працівників та стабільної роботи підприємства.

5.2 Комплекс заходів по зменшенню шуму в джерелі його виникнення.

Шум є одним із найпоширеніших шкідливих виробничих факторів, що негативно впливає на здоров'я працівників. Зниження рівня шуму безпосередньо у джерелі його виникнення є найбільш ефективним способом захисту від шумового забруднення. Комплекс заходів передбачає впровадження технічних, організаційних та санітарно-гігієнічних рішень на етапі проектування та експлуатації обладнання [16]. Технічні та конструктивні заходи для зменшення шуму включають наступні дії:

- Конструктори заміняють ударні механізми на безударні для зниження рівня вібрації та звуку;
- Інженери використовують звукоізоляційні кожухи навколо шумного обладнання для локалізації звукових хвиль;

- Механіки встановлюють амортизатори та віброізолюючі прокладки між станками та фундаментами;
- Технологи замінюють металеві деталі на пластикові або текстолітові у вузлах тертя та зіткнення;
- Працівники регулярно змащують рухомі частини механізмів для зменшення тертя та супутнього шуму;

Окрім технічних рішень, важливу роль відіграють організаційні заходи. До них належить правильне планування виробничих приміщень, винесення шумного обладнання в окремі звукоізольовані зони та обмеження часу перебування працівників у зонах підвищеного шуму.

Регулярний моніторинг рівня шуму дозволяє вчасно виявляти перевищення допустимих норм та застосовувати додаткові засоби захисту [16]. Класифікація заходів по зменшенню шуму подана в таблиці 5.2.

Таблиця 5.2 - Класифікація заходів по зменшенню шуму

Група заходів	Приклади реалізації на виробництві
Технічні	Встановлення глушників, використання звукопоглинальних матеріалів
Організаційні	Раціональний режим праці та відпочинку, зонування приміщень
Архітектурно-планувальні	Акустична обробка стін та стелі, використання екранів

Ефективна боротьба з виробничим шумом вимагає комплексного підходу та регулярного оновлення парку обладнання. Використання сучасних малошумних технологій дозволяє не лише зберегти здоров'я працівників, але й підвищити загальну продуктивність праці. Впровадження цих заходів є обов'язковою складовою системи управління охороною праці на сучасному підприємстві.

ВИСНОВКИ

У результаті успішного виконання кваліфікаційної роботи було розроблено сучасну адаптивну вебплатформу для експозиції та реалізації творів мистецтва GalleryHub. У межах проєктування даний проєкт реалізовано відповідно до вимог технічного завдання та охоплює повний цикл розробки вебзастосунку від створення архітектури до реалізації клієнтської та серверної частин. Завдяки цьому розробка є повністю комплексною. Під час виконання роботи було проведено аналіз предметної області, визначено основні вимоги та спроектовано загальну архітектуру платформи.

Для забезпечення зберігання даних було створено логічну структуру бази даних PostgreSQL, яка включає основні сутності системи, зокрема користувачі, роботи, замовлення, матеріали та теги. Для забезпечення взаємодії між клієнтською та серверною частинами системи було спроектовано та реалізовано швидкий та функціональний REST API з використанням Express.js. Завдяки цьому надійно гарантується безпека збережених даних. Також додатково було створено систему модерації, яка дозволяє адміністратору оперативно перевіряти додані користувачами твори художників перед їх публікацією.

Зі сторони користувача клієнтська частина на основі React 19 забезпечує зручний графічний інтерфейс, перегляд художніх робіт, фільтрацію за параметрами, управління кошиком та обраним. На противагу цьому серверна частина повністю відповідає за бізнес-логіку вебзастосунку, обробку клієнтських запитів та взаємодію з базою даних через Prisma ORM. Завдяки успішній інтеграції всі модулі працюють стабільно. У процесі виконання роботи були повністю самостійно розроблені загальна структура вебплатформи, логіка взаємодії компонентів системи, реляційна база даних, REST API, а також реалізовано весь основний функціонал.

Для візуалізації розроблюваної системи було самостійно створено схему структурну, діаграму варіантів використання, діаграму бази даних, а також розраховано техніко-економічні показники проєкту. У результаті аналізу

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

встановлено, що отримані практичні результати повністю відповідають вимогам технічного завдання на виконання кваліфікаційної роботи. Завдяки цьому повністю підтверджується успішність виконаної розробки. У підсумку створена вебплатформа є функціонально завершеною та може бути успішно використана для реальної експозиції та продажу творів мистецтва в мережі інтернет.

У перспективі подальший розвиток проєкту може включати розширення функціоналу, зокрема інтеграцію платіжної системи для безпечної оплати, впровадження рекомендацій на основі штучного інтелекту та покращення аналітики. Поряд із цим, важливим практичним результатом виконаної роботи стало набуття навичок проєктування та розробки сучасних вебзастосунків із використанням клієнт-серверної архітектури. Завдяки цьому значно покращилися професійні навички автора роботи. Під час реалізації проєкту було закріплено теоретичні знання з побудови REST API, роботи з реляційними базами даних та організації взаємодії між фронтенд- і бекенд-частинами системи.

У процесі дослідження предметної області було підготовлено та опубліковано наукову доповідь на тему сучасних методів розробки вебплатформ для реалізації художніх творів мистецтва. Зокрема, матеріали дослідження пройшли успішну апробацію у формі доповіді на щорічній науково-технічній конференції молодих вчених та студентів нашого технічного університету. Ця підготовлена наукова доповідь отримала позитивні оцінки колег. Отриманий у процесі написання роботи практичний досвід аналізу предметної області та проєктування інформаційних систем дозволив сформулювати цілісне розуміння створення вебплатформ від етапу ідеї до готового програмного продукту.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Грицюк Ю.І. Проєктування інформаційних систем : навч. посібник. Львів : Вид-во Львівської політехніки, 2022. 320 с.
- 2) Vanderkam D. Effective TypeScript: 83 Specific Ways to Improve Your TypeScript. Sebastopol : O'Reilly Media, 2024. 384 p.
- 3) Schwarzmüller M. React Key Concepts. Birmingham : Packt Publishing, 2022. 402 p.
- 4) Herron D. Node.js Web Development: Server-side development with Express, SQL, and MongoDB. Birmingham : Packt Publishing, 2023. 756 p.
- 5) PostgreSQL 14 Administration Cookbook / Ciolli G., Frost S., Riggs R. and Simon H. Birmingham : Packt Publishing, 2022. 518 p.
- 6) Supabase Documentation : вебсайт. URL: <https://supabase.com/docs> (дата звернення: 15.05.2026).
- 7) Prisma Documentation : вебсайт. URL: <https://www.prisma.io/docs> (дата звернення: 20.05.2026).
- 8) Cloudinary Documentation : вебсайт. URL: <https://cloudinary.com/documentation> (дата звернення: 25.05.2026).
- 9) Zustand Documentation : вебсайт. URL: <https://docs.pmnd.rs/zustand> (дата звернення: 22.05.2026).
- 10) React 19 Documentation : вебсайт. URL: <https://react.dev> (дата звернення: 10.05.2026).
- 11) Березуцький В.В. Управління охороною праці. Харків : ФОП Панов А.М., 2021. 412 с.
- 12) Геревенко А.М. Організація охорони праці та безпеки життєдіяльності в закладі професійної освіти. Біла Церква : БІНПО ДЗВО УМО НАПН України, 2024. 66 с.

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						80
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А. Лістинг файлу «App.tsx»

```
import { Routes, Route, useLocation } from 'react-router-dom';
import { Toaster } from 'react-hot-toast';
import { AnimatePresence, motion } from 'framer-motion';
import Navbar from './components/Navbar';
import Footer from './components/Footer';
import Home from './pages/Home';
import Gallery from './pages/Gallery';
import ArtworkDetails from './pages/ArtworkDetails';
import Checkout from './pages/Checkout';
import AdminDashboard from './pages/AdminDashboard';
import Login from './pages/Login';
import ArtistProfile from './pages/ArtistProfile';
import Favorites from './pages/Favorites';
import Profile from './pages/Profile';
import AddArtwork from './pages/AddArtwork';
import { useAuthStore } from './store/authStore';
import { Navigate, Outlet } from 'react-router-dom';
const PrivateRoute = () => {
  const { user } = useAuthStore();
  return user ? <Outlet /> : <Navigate to="/login" />;
};
function App() {
  const location = useLocation();
  return (
    <div className="min-h-screen bg-white text-gray-900 flex flex-col font-sans selection:bg-indigo-100 selection:text-indigo-900">
      <Toaster
        position="top-right"
        toastOptions={{
          style: {
```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

```

        marginTop: '80px',
      }
    }}
  />
  <Navbar />
  <main className="flex-grow max-w-7xl w-full mx-auto px-4 sm:px-
6 lg:px-8 py-12">
    <AnimatePresence mode="wait">
      <motion.div
        key={location.pathname}
        initial={{ opacity: 0, y: 10 }}
        animate={{ opacity: 1, y: 0 }}
        exit={{ opacity: 0, y: -10 }}
        transition={{ duration: 0.3 }}
      >
        <Routes location={location}>
          <Route path="/" element={<Home />} />
          <Route path="/gallery" element={<Gallery />} />
          <Route path="/artwork/:id" element={<ArtworkDetails />} />
          <Route path="/artist/:id" element={<ArtistProfile />} />
          <Route path="/login" element={<Login />} />
          <Route path="/favorites" element={<Favorites />} />
          <Route path="/checkout" element={<Checkout />} />
          <Route element={<PrivateRoute />}>
            <Route path="/admin/*" element={<AdminDashboard />} />
            <Route path="/profile" element={<Profile />} />
            <Route path="/add-artwork" element={<AddArtwork />} />
          </Route>
        </Routes>
      </motion.div>
    </AnimatePresence>
  </main>
  <Footer />
</div>

```

```
);  
}  
export default App;
```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

Додаток Б. Лістинг файлу «ArtworkCard.tsx»

```
import { Link } from 'react-router-dom';
import { ShoppingCart, Check, Star } from 'lucide-react';
import { useCartStore } from '../store/cartStore';
import { useFavoriteStore } from '../store/favoriteStore';
import { useAuthStore } from '../store/authStore';
import toast from 'react-hot-toast';

interface ArtworkProps {
  artwork: any;
}

export default function ArtworkCard({ artwork }: ArtworkProps) {
  const { user } = useAuthStore();
  const { addItem, items: cartItems } = useCartStore();
  const { toggleFavorite, items: favoriteItems } = useFavoriteStore();
  const isInCart = cartItems.some(item => item.id === artwork.id);
  const isFavorite = favoriteItems.some(item => item.id === artwork.id);
  const isAuthenticated = !!user;

  const primaryImage = artwork.images?.find((img: any) => img.id ===
artwork.primaryImageId)?.url || artwork.images?.[0]?.url ||
'https://via.placeholder.com/400x300?text=No+Image';
  const handleAddToCart = (e: React.MouseEvent) => {
    e.preventDefault();
    e.stopPropagation();
    if (!isInCart) {
      addItem(artwork, isAuthenticated);
      toast.success(`${artwork.title} додано до кошика!`);
    } else {
      toast.error('Вже в кошику');
    }
  };

  const handleToggleFavorite = (e: React.MouseEvent) => {
    e.preventDefault();
  };
}
```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```

    e.stopPropagation();
    toggleFavorite(artwork, isAuthenticated);
    if (!isFavorite) {
        toast.success(`${artwork.title} додано до обраного!`);
    }
};
return (
    <div className="bg-white rounded-[2rem] shadow-sm border border-gray-
100 overflow-hidden hover:shadow-xl hover:shadow-indigo-100/50 transition-
all duration-300 group">
        <Link to={` /artwork/${artwork.id}`} className="block relative aspect-
[4/3] overflow-hidden">
            <img
                src={primaryImage}
                alt={artwork.title}
                className="w-full h-full object-cover group-hover:scale-110
transition-transform duration-500"
            />
            <div className="absolute inset-0 bg-black/5 group-hover:bg-black/0
transition-colors" />
            { /* Buttons Overlay */ }
            <div className="absolute top-4 right-4 flex flex-col gap-2
translate-x-12 group-hover:translate-x-0 transition-transform duration-
300">
                <button
                    onClick={handleToggleFavorite}
                    className={`p-3 rounded-2xl shadow-lg transition-all transform
hover:scale-110 ${
                        isFavorite ? 'bg-amber-400 text-white' : 'bg-white/90 text-
gray-400 hover:text-amber-400'
                    }`}
                >
                    <Star className={`h-5 w-5 ${isFavorite ? 'fill-current' : ''}`}
                />
            />

```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						85
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        </button>
    </div>
    <button
        onClick={handleAddToCart}
        className={`absolute bottom-4 right-4 p-3 rounded-2xl shadow-lg
transition-all transform hover:scale-110 ${
            isInCart ? 'bg-green-500 text-white' : 'bg-white text-indigo-
600 hover:bg-indigo-600 hover:text-white'
        }`}
    >
        {isInCart ? <Check className="h-5 w-5" /> : <ShoppingCart
className="h-5 w-5" />}
    </button>
</Link>
<div className="p-6">
    <div className="flex justify-between items-start mb-1">
        <Link to={` /artwork/${artwork.id}`} className="flex-1 min-w-0 mr-
2">
            <h3 className="font-bold text-lg text-gray-900 hover:text-
indigo-600 transition-colors truncate">{artwork.title}</h3>
        </Link>
        <span className="font-black text-indigo-600 shrink-
0">{artwork.price} грн</span>
    </div>
    <div className="flex flex-col gap-1 mb-4">
        <Link
            to={` /artist/${artwork.artistId}`}
            className="text-sm font-medium text-gray-400 hover:text-indigo-
500 transition-colors"
        >
            {artwork.artist ? `${artwork.artist.firstName}
${artwork.artist.lastName}` : 'Невідомий автор'}
        </Link>
        {artwork.width && artwork.height && (

```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						86
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        <span      className="text-[11px]      text-gray-400      font-
medium">{artwork.width}x{artwork.height} cm</span>
      )}
    </div>
    {artwork.tags && artwork.tags.length > 0 && (
      <div className="flex flex-wrap gap-1 mb-4">
        {artwork.tags.slice(0, 3).map((tag: any) => (
          <span key={tag.id} className="px-2 py-0.5 bg-indigo-50 text-
indigo-600 text-[9px] font-bold rounded-md uppercase tracking-wider">
            {tag.name}
          </span>
        ))}
      </div>
    )}
    <div className="mt-auto pt-4 border-t border-gray-50 flex justify-
between items-center">
      <span className="text-[10px] font-bold tracking-wider uppercase
px-3 py-1 bg-gray-50 text-gray-400 rounded-full truncate max-w-[120px]">
        {artwork.category?.name || 'Мистецтво'}
      </span>
      <Link
        to={` /artwork/${artwork.id}`}
        className="text-xs font-bold text-indigo-600 hover:underline
shrink-0"
      >
        Деталі
      </Link>
    </div>
  </div>
</div>
);
}

```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

Додаток В. Лістинг файлу «HeroSlider.tsx»

```
import { useState, useEffect } from 'react';
import { Swiper, SwiperSlide } from 'swiper/react';
import { Navigation, Pagination, Autoplay, EffectFade } from
'swiper/modules';
import { motion } from 'framer-motion';
import { Link } from 'react-router-dom';
import { ArrowRight } from 'lucide-react';
import api from '../api/axios';
import 'swiper/css';
import 'swiper/css/navigation';
import 'swiper/css/pagination';
import 'swiper/css/effect-fade';
export default function HeroSlider() {
  const [slides, setSlides] = useState<any[]>([]);
  useEffect(() => {
    const fetchLatestArtworks = async () => {
      try {
        const { data } = await
api.get('/artworks?size=3&sort=id,desc&status=APPROVED');
        setSlides(data.content || []);
      } catch (error) {
        console.error('Failed to fetch hero slides', error);
      }
    };
    fetchLatestArtworks();
  }, []);
  return (
    <div className="relative h-[600px] rounded-3xl overflow-hidden shadow-
2xl mb-20 group">
      <Swiper
        modules={[Navigation, Pagination, Autoplay, EffectFade]}
        effect="fade"
        spaceBetween={0}
        slidesPerView={1}

```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						88
Зм.	Арк.	№ докум.	Підпис	Дата		

```

navigation={{
  nextEl: '.swiper-button-next-custom',
  prevEl: '.swiper-button-prev-custom',
}} pagination={{ clickable: true, dynamicBullets: true }}
autoplay={{ delay: 5000, disableOnInteraction: false }}
loop={slides.length > 1}
className="h-full w-full"
>      {slides.map((artwork) => (
    <SwiperSlide key={artwork.id}>
      <div className="relative h-full w-full">
        <div className="absolute inset-0 bg-black/40 z-10" />
        <img
          src={artwork.images?.find((img: any) => img.id ===
artwork.primaryImageId)?.url || artwork.images?.[0]?.url ||
"https://images.unsplash.com/photo-1541963463532-
d68292c34b19?auto=format&fit=crop&q=80&w=2000"}
          alt={artwork.title}
          className="absolute inset-0 w-full h-full object-cover"
        /> <div className="relative z-20 h-full flex flex-col
justify-center items-start px-12 md:px-24 max-w-4xl">
          <motion.span
            initial={{ opacity: 0, y: 20 }}
            whileInView={{ opacity: 1, y: 0 }}
            transition={{ delay: 0.2 }}
            className="text-indigo-400 font-bold tracking-widest
uppercase text-sm mb-4"
          > Колекція: {artwork.category?.name || 'Мистецтво'}
          </motion.span>
          <motion.h2
            initial={{ opacity: 0, y: 30 }}
            whileInView={{ opacity: 1, y: 0 }}
            transition={{ delay: 0.4 }}
            className="text-5xl md:text-7xl font-extrabold text-white
mb-8 leading-tight"

```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89

```

    > {artwork.title}
  </motion.h2>
  <motion.div
    initial={{ opacity: 0, scale: 0.9 }}
    whileInView={{ opacity: 1, scale: 1 }}
    transition={{ delay: 0.6 }}
  >
    <Link
      to={` /artwork/${artwork.id}`}
      className="bg-white text-indigo-900 px-8 py-4 rounded-
full font-bold flex items-center gap-2 hover:bg-indigo-50 transition-colors
shadow-lg"
    >
      Переглянути шедевр
    <ArrowRight className="h-5 w-5" />
    </Link>
  </motion.div>
</div>
</div>
</SwiperSlide>      )}}
{/* Custom Navigation */}
<button className="swiper-button-prev-custom absolute left-6 top-
1/2 -translate-y-1/2 z-30 p-3 rounded-full bg-white/10 backdrop-blur-md
text-white border border-white/20 hover:bg-white/20 transition-all opacity-
0 group-hover:opacity-100 hidden md:block">
  <ArrowRight className="h-6 w-6 rotate-180" />
</button>
<button className="swiper-button-next-custom absolute right-6 top-
1/2 -translate-y-1/2 z-30 p-3 rounded-full bg-white/10 backdrop-blur-md
text-white border border-white/20 hover:bg-white/20 transition-all opacity-
0 group-hover:opacity-100 hidden md:block">
  <ArrowRight className="h-6 w-6" />
</button>
</Swiper>
</div>
);}

```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						90
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Г. Лістинг файлу «AddArtwork.tsx»

```
import { useState, useEffect } from 'react';
import { Helmet } from 'react-helmet-async';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';
import toast from 'react-hot-toast';
import { UploadCloud, DollarSign, Type, FileText, Layers, Palette, Trash2 }
from 'lucide-react';
import api from '../api/axios';
import TagInput from '../components/TagInput';
export default function AddArtwork() {
  const [formData, setFormData] = useState({
    title: '',
    description: '',
    price: '',
    width: '',
    height: '',
    categoryId: '',
    materialId: '',
    tags: '',
    images: [] as File[],
  });
  const [imagePreviews, setImagePreviews] = useState<string[]>([]);
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();
  const [categories, setCategories] = useState<any[]>([]);
  const [materials, setMaterials] = useState<any[]>([]);
  useEffect(() => {
    const fetchMetadata = async () => {
      try {
        const [catRes, matRes] = await Promise.all([
          api.get('/metadata/categories'),
          api.get('/metadata/materials')
        ]);
      }
    };
  });
```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

```

    });
    setCategories(catRes.data);
    setMaterials(matRes.data);
  } catch (error) {
    console.error('Failed to fetch metadata', error);
  }
};
fetchMetadata();
}, []));
const handleFileChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  const files = Array.from(e.target.files || []);
  if (files.length > 0) {
    setFormData(prev => ({ ...prev, images: [...prev.images, ...files]
})));
    const newPreviews = files.map(file => URL.createObjectURL(file));
    setImagePreviews(prev => [...prev, ...newPreviews]);
  }
};
const removeImage = (index: number) => {
  setFormData(prev => ({
    ...prev,
    images: prev.images.filter((_, i) => i !== index)
})));
  setImagePreviews(prev => prev.filter((_, i) => i !== index));
};
const handleSubmit = async (e: React.FormEvent) => {
  e.preventDefault();
  if (formData.images.length === 0) {
    toast.error('Будь ласка, завантажте хоча б одне зображення.');
```

```

    return;
  }
  setLoading(true);
  const submissionData = new FormData();
  submissionData.append('title', formData.title);
```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						92
Зм.	Арк.	№ докум.	Підпис	Дата		

```

submissionData.append('description', formData.description);
submissionData.append('price', formData.price);
submissionData.append('width', formData.width);
submissionData.append('height', formData.height);
submissionData.append('categoryId', formData.categoryId);
submissionData.append('materialId', formData.materialId);
submissionData.append('tags', formData.tags);

formData.images.forEach(image => {
  submissionData.append('images', image);
});
try {
  await api.post('/artworks', submissionData, {
    headers: {
      'Content-Type': 'multipart/form-data',
    },
  });
  toast.success('Твір відправлено на перевірку!');
  navigate('/profile');
} catch (err: any) {
  toast.error(err.response?.data?.message || 'Не вдалося надіслати твір.');
```

```

  } finally {
    setLoading(false);
  }
};
return (
  <div className="max-w-4xl mx-auto py-12">
    <Helmet>
      <title>Додати новий твір | GalleryHub</title>
    </Helmet>
    <motion.div
      initial={{ opacity: 0, y: 20 }}
      animate={{ opacity: 1, y: 0 }}

```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						93
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        className="bg-white p-8 rounded-[2.5rem] shadow-xl shadow-indigo-
100/50 border border-gray-100"
    >
        <div className="text-center mb-8">
            <h2 className="text-3xl font-bold text-gray-900 mb-2">Надішліть
свій шедевр</h2>
            <p className="text-gray-500">Поділіться своїм мистецтвом зі
світом. Заповніть деталі нижче.</p>
        </div>
        <form onSubmit={handleSubmit} className="space-y-6">
            <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
                {/* Left side: Image Upload */}
                <div className="space-y-4">
                    <label className="block text-sm font-bold text-gray-700 ml-
1">Зображення твору (до 10 шт)</label>
                    <div
                        className="relative border-2 border-dashed border-gray-300
rounded-2xl p-6 text-center cursor-pointer hover:border-indigo-500
transition-colors"
                        onClick={() => document.getElementById('image-
upload')?.click()}
                    >
                        <input type="file" id="image-upload" className="hidden"
onChange={handleFileChange} accept="image/*" multiple />
                        <div className="space-y-2 text-gray-400">
                            <UploadCloud className="h-10 w-10 mx-auto" />
                            <p className="font-bold">Натисніть, щоб завантажити
декілька фото</p>
                            <p className="text-xs">PNG, JPG, GIF до 10MB кожне</p>
                        </div>
                    </div>

                    {imagePreviews.length > 0 && (
                        <div className="grid grid-cols-3 gap-4">

```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						94
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        {imagePreviews.map((preview, index) => (
            <div key={index} className="relative group aspect-
square rounded-xl overflow-hidden border border-gray-100">
                <img src={preview} alt="Preview" className="w-full h-
full object-cover" />
                <button
                    type="button"
                    onClick={(e) => { e.stopPropagation();
removeImage(index); }}
                    className="absolute top-1 right-1 bg-red-500 text-
white p-1.5 rounded-lg opacity-0 group-hover:opacity-100 transition-
opacity"
                >
                    <Trash2 className="h-4 w-4" />
                </button>
                {index === 0 && (
                    <div className="absolute bottom-0 left-0 right-0
bg-indigo-600/80 text-white text-[10px] font-bold text-center py-1">
                        Головне
                    </div>
                )}
            </div>
        )})
    </div>
    { /* Right side: Form Fields */}
    <div className="space-y-4">
        <div className="space-y-2">
            <label className="block text-sm font-bold text-gray-700 ml-
1">Назва</label>
            <div className="relative">
                <Type className="absolute left-4 top-1/2 -translate-y-1/2
h-5 w-5 text-gray-400" />

```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						95
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        <input type="text" name="title" required onChange={(e) =>
setFormData({...formData, title: e.target.value})} placeholder="напр.,
Зоряна ніч" className="w-full bg-gray-50 border-gray-200 rounded-xl py-3
pl-12 pr-4 focus:ring-2 focus:ring-indigo-500" />
    </div>
</div>
<div className="space-y-2">
    <label className="block text-sm font-bold text-gray-700 ml-
1">Опис</label>
    <div className="relative">
        <FileText className="absolute left-4 top-4 h-5 w-5 text-
gray-400" />
        <textarea name="description" required onChange={(e) =>
setFormData({...formData, description: e.target.value})}
placeholder="Опишіть свій твір..." rows={4} className="w-full bg-gray-50
border-gray-200 rounded-xl py-3 pl-12 pr-4 focus:ring-2 focus:ring-indigo-
500" />    </div>
    </div>
<div className="space-y-2">
    <label className="block text-sm font-bold text-gray-700 ml-
1">Ціна (ГРН)</label>
    <div className="relative">
        <DollarSign className="absolute left-4 top-1/2 -
translate-y-1/2 h-5 w-5 text-gray-400" />
        <input type="number" name="price" required onChange={(e)
=> setFormData({...formData, price: e.target.value})} placeholder="100"
className="w-full bg-gray-50 border-gray-200 rounded-xl py-3 pl-12 pr-4
focus:ring-2 focus:ring-indigo-500" />
    </div>
</div>
<div className="grid grid-cols-2 gap-4">
    <div className="space-y-2">
        <label className="block text-sm font-bold text-gray-700
ml-1">Ширина (см)</label>

```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

```

        <input type="number" name="width" onChange={(e) =>
setFormData({...formData, width: e.target.value})} placeholder="50"
className="w-full bg-gray-50 border-gray-200 rounded-xl py-3 px-4
focus:ring-2 focus:ring-indigo-500" />
      </div>
      <div className="space-y-2">
        <label className="block text-sm font-bold text-gray-700
ml-1">Висота (см)</label>
        <input type="number" name="height" onChange={(e) =>
setFormData({...formData, height: e.target.value})} placeholder="70"
className="w-full bg-gray-50 border-gray-200 rounded-xl py-3 px-4
focus:ring-2 focus:ring-indigo-500" />
      </div>
    </div>
  </div>
</div>
<div className="grid grid-cols-1 md:grid-cols-3 gap-6">
  <div className="space-y-2">
    <label className="block text-sm font-bold text-gray-700 ml-
1">Категорія</label>
    <div className="relative">
      <Layers className="absolute left-4 top-1/2 -translate-y-1/2
h-5 w-5 text-gray-400" />
      <select name="categoryId" required onChange={(e) =>
setFormData({...formData, categoryId: e.target.value})} className="w-full
bg-gray-50 border-gray-200 rounded-xl py-3 pl-12 pr-4 focus:ring-2
focus:ring-indigo-500">
        <option value="">Виберіть категорію</option>
        {categories.map(cat => <option key={cat.id}
value={cat.id}>{cat.name}</option>)}
      </select>
    </div>
  </div>
  <div className="space-y-2">

```

```

        <label className="block text-sm font-bold text-gray-700 ml-1">Матеріал</label>
        <div className="relative">
            <Palette className="absolute left-4 top-1/2 -translate-y-1/2 h-5 w-5 text-gray-400" />
            <select name="materialId" onChange={(e) => setFormData({...formData, materialId: e.target.value})} className="w-full bg-gray-50 border-gray-200 rounded-xl py-3 pl-12 pr-4 focus:ring-2 focus:ring-indigo-500">
                <option value="">Виберіть матеріал</option>
                {materials.map(mat => <option key={mat.id} value={mat.id}>{mat.name}</option>)}
            </select>
        </div>
    </div>
    <div className="space-y-2">
        <label className="block text-sm font-bold text-gray-700 ml-1">Теги твору</label>
        <TagInput
            selectedTags={formData.tags}
            onChange={(tags) => setFormData({ ...formData, tags })}
        />
        <p className="text-[10px] text-gray-400 ml-1 italic">Виберіть зі списку або введіть нові через Enter</p>
    </div>
    <div>
        <button
            type="submit"
            disabled={loading}
            className="w-full bg-indigo-600 text-white py-4 rounded-2xl font-bold hover:bg-indigo-700 transition-all disabled:opacity-50"
        >
            {loading ? 'Надсилання...' : 'Надіслати твір'}
        </button> </form> </motion.div> </div> );};

```

					2026.КВР.122.423.21.00.00 ПЗ	Арк.
						98
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Д. Лістинг файлу «cartStore.ts»

```
import { create } from 'zustand';
import { persist } from 'zustand/middleware';
import api from '../api/axios';
export interface Artwork {
  id: number;
  title: string;
  price: number;
  artistFullName: string;
  images: Array<{ url: string; isPrimary: boolean }>;
}
interface CartState {
  items: Artwork[];
  addItem: (artwork: Artwork, isAuthenticated: boolean) => Promise<void>;
  removeItem: (id: number, isAuthenticated: boolean) => Promise<void>;
  clearCart: () => void;
  totalPrice: () => number;
  syncWithDB: () => Promise<void>;
  fetchFromDB: () => Promise<void>;
}
export const useCartStore = create<CartState>()(
  persist(
    (set, get) => ({
      items: [],
      addItem: async (artwork, isAuthenticated) => {
        const alreadyInCart = get().items.some(item => item.id ===
artwork.id);
        if (!alreadyInCart) {
          set({ items: [...get().items, artwork] });
          if (isAuthenticated) {
            try {
              await api.post('/users/cart', { artworkId:
artwork.id.toString() });
            } catch {
              // ignore
            }
          }
        }
      },
      removeItem: (id) => {
        set({ items: get().items.filter(item => item.id !== id) });
      },
      clearCart: () => set({ items: [] }),
      totalPrice: () => get().items.reduce((sum, item) => sum + item.price, 0),
      syncWithDB: () => {
        // sync with DB
      },
      fetchFromDB: () => {
        // fetch from DB
      }
    })
  )
);
```

					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						99
Зм.	Арк.	№ докум.	Підпис	Дата		

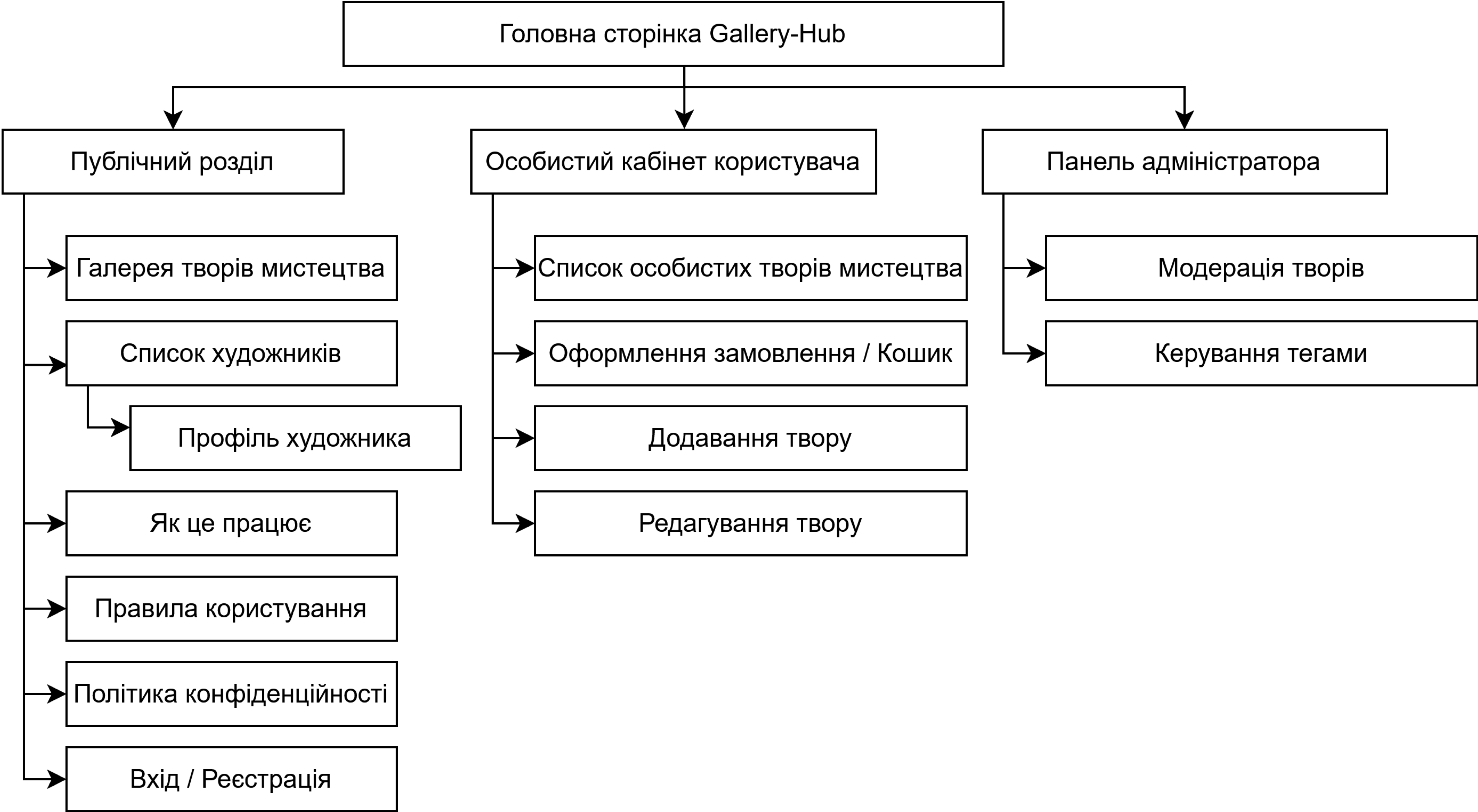
```

        } catch (error) {
            console.error('Failed to add to cart in DB', error);
        } }
    },
    removeItem: async (id, isAuthenticated) => {
        set({ items: get().items.filter(item => item.id !== id) });
        if (isAuthenticated) {
            try {
                await api.delete(`/users/cart/artwork/${id}`);
            } catch (error) {
                console.error('Failed to remove from cart in DB', error);
            } } },
    clearCart: () => set({ items: [] }),
    totalPrice: () => get().items.reduce((sum, item) => sum +
Number(item.price), 0),
    syncWithDB: async () => {
        try {
            const localIds = get().items.map(item => item.id.toString());
            if (localIds.length > 0) {
                await api.post('/users/cart/sync', localIds);
            }
        } catch (error) {
            console.error('Failed to sync cart with DB', error);
        } },
        fetchFromDB: async () => {
            try {
                const { data } = await api.get('/users/cart');
                const artworks = data.map((item: any) => item.artwork);
                set({ items: artworks });
            } catch (error) {
                console.error('Failed to fetch cart from DB', error);
            }
        },
    },
    { name: 'galleryhub_cart', } ));

```

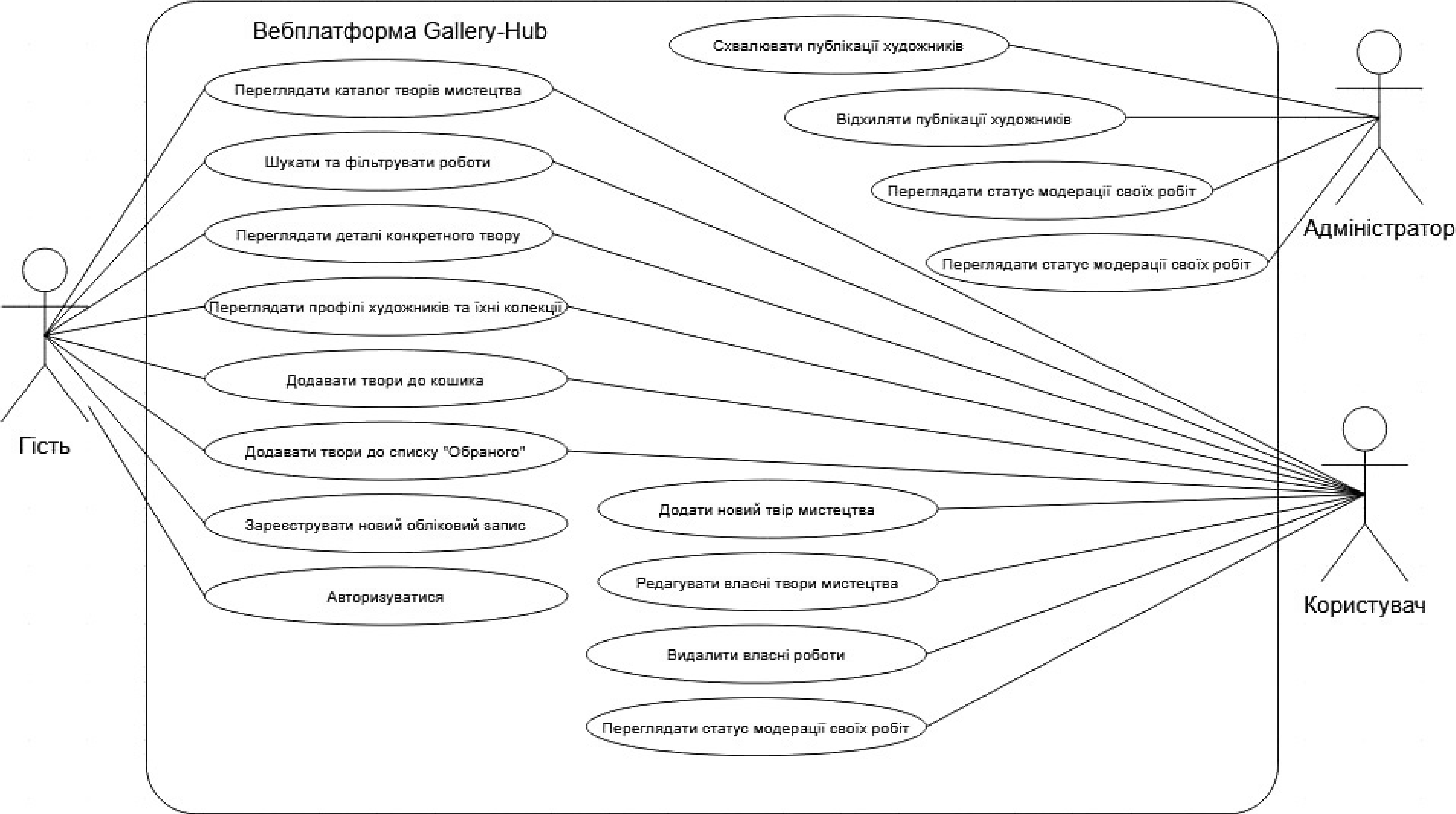
					2026.KBP.122.423.21.00.00 ПЗ	Арк.
						100
Зм.	Арк.	№ докум.	Підпис	Дата		

Схема структурна клієнтської частини



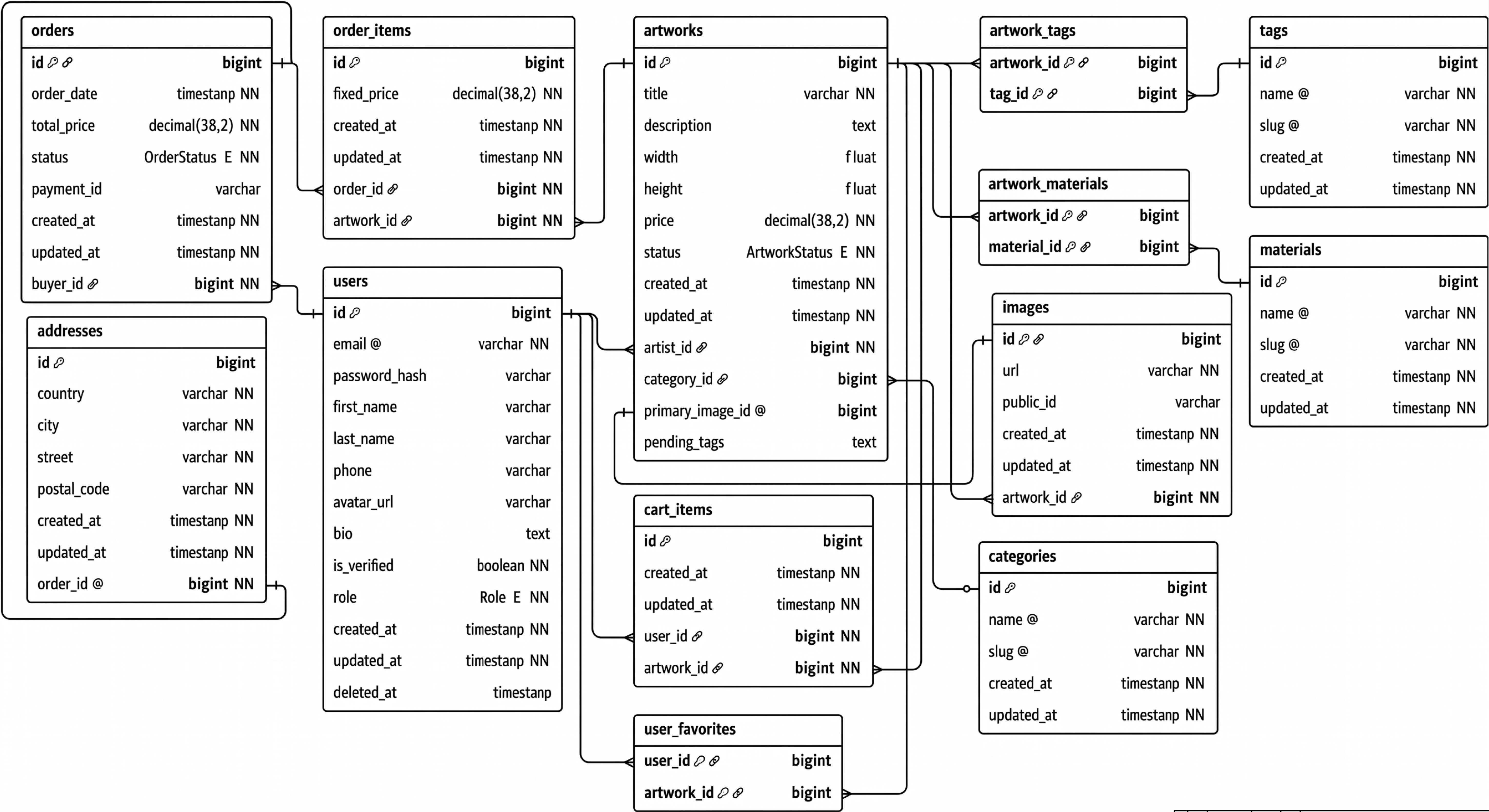
						2026.KBP.122.423.2100.00 CC		
Зм.	Арк.	№ доки.	Підпис	Дата	Розробка вебплатформи для експозиції та реалізації творів мистецтва «GalleryHub»	Лист	Маса	Масштаб
Розроб.	Стороженко А.В.				Схема структурної клієнтської частини			
Перевір.	Стороженко Р.О.					Арк.	Арк.	1
Текст.						ВСТ ТФК ТНТУ КН-423		
Рецензент						м. Тернопіль		
Начальник	Приймач В.А.					Формат А1		
Затв.								

UML-діаграма варіантів використання



2026.KBP.122.423.2100.00 ДВ						Розробка вебплатформи для експозиції та реалізації творів мистецтва «GalleryHub» UML-діаграма варіантів використання			Лист	Маса	Масштаб
Зм.	Арх.	№ докум.	Підпис	Дата		Архи	Архив	1			
Розроб.	Ступака А.В.					ВСП ТАК ТНТУ	КН-423				
Перевір.	Слободян Р.О.					м. Тернопіль					
Текст.						Формат А1					
Рецензент											
Начальник	Приймач В.А.										
Затв.											

ER-діаграма бази даних



					2026.KBP.122.423.2100.00 БД						
					Розробка вебплатформи для експозиції та реалізації творів мистецтва «GalleryHub» ER-діаграма бази даних	Лист	Маса	Масштаб			
Зна	Арх	№ докум	Підпис	Дата							
Розроб		Ступаків А.В.									
Перевір		Слободян Р.О.									
Т.контр											
Рецензент						Архиви		Архиви	1		
Н.контр		Примак В.А.				ВСТ ТФК ТНТУ КН-423					
Зам						м. Тернопіль					

Таблиця техніко-економічних показників

№	Показник	Одиниця вимірювання	Значення
1	Бібліотека інтерфейсу користувача	–	React v19
2	Фреймворк серверної частини	–	Express.js v5.2.1
3	Система керування базою даних	–	PostgreSQL
4	Основних таблиць у схемі бази даних	шт.	10
5	Допоміжних таблиць у схемі бази даних	шт.	3
6	Кількість ролей користувачів	шт.	3
7	Содіовартість	грн.	81 251
8	Плановий грошовий потік	грн.	45 500
9	Ціна	грн.	126 752
10	Чиста теперішня вартість	грн.	31 617
11	Термін окупності	рік.	2,1