

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” _____ 2023 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою “Комп’ютерні системи та мережі”
спеціальності 123 “Комп’ютерна інженерія”

на тему: Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain

Виконав: студент 4 курсу, групи ІВ-91
(шифр групи)

Мусійчук Ярослав Сергійович

(прізвище, ім’я, по батькові)

(підпис)

Керівник асистент, Нікольський С.С.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) ст. викладач, Виноградов Ю. М.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2023 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
Сергій СТИРЕНКО

_____ (підпис)

“ ___ ” _____ 2023 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Мусійчука Ярослава Сергійовича

1. Тема проєкту Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain

керівник проєкту _____ Нікольський Сергій Сергійович, асистент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 31.05 2023 року № 2101-с

2. Строк подання студентом проєкту 6 червня 2023 р.

3. Вихідні дані до проєкту технічна документація, теоретичні дані

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

Розділ 1. Огляд маркетплейс-платформ для торгівлі та створення NFT на базі Blockchain.

Розділ 2. Огляд інструментів та технологій для розробки системи.

Розділ 3. Деталі розробки системи.

Розділ 4. Дослідження та аналіз розробленої системи.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) діаграма потоків даних (структурна схема), діаграма класів (функціональна схема), алгоритм дій програмного забезпечення (принципова схема).

6. Консультанти розділів проєкту

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Виноградов Ю. М.		

7. Дата видачі завдання «22» листопада 2022 р.

Календарний план

№ п/п	Найменування етапів дипломного проєкту	Строк виконання етапів проєкту	Примітка
1.	<i>Затвердження теми проєкту</i>	<i>16.11.2022-22.11.2022</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>16.11.2022-29.12.2022</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>29.12.2023-07.01.2023</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>08.01.2023-15.01.2023</i>	
5.	<i>Програмна реалізація системи</i>	<i>15.01.2023-29.04.2023</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.05.2023-03.06.2023</i>	
7.	<i>Захист програмного продукту</i>	<i>05.06.2023</i>	
8.	<i>Передзахист</i>	<i>09.06.2023</i>	
9.	<i>Захист</i>	<i>19.06.2023</i>	

Студент

Ярослав МУСІЙЧУК

(підпис)

Керівник проєкту

Сергій НІКОЛЬСЬКИЙ

(підпис)

АНОТАЦІЯ

У даному бакалаврському дипломному проєкті була розроблена маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain. Платформа надає можливість користувачам створювати, редагувати та продавати унікальні NFT, що забезпечується захистом від копіювання та підробки за допомогою технології Blockchain. Крім того, на маркетплейсі можна придбати та продавати вже існуючі NFT, а також проводити торги на аукціонах. Платформа надає можливість здійснювати оплату за NFT у криптовалюті, що дозволяє зменшити витрати на операції та забезпечує швидкий та безпечний обмін цінностями між користувачами. Проєкт дозволяє забезпечити створення та обіг NFT в ефективний та безпечний спосіб.

ANNOTATION

In this project for a Bachelor's Degree, a marketplace platform for trading and creating NFTs based on Blockchain was developed. The platform allows users to create, edit, and sell unique NFTs, which is protected against copying and counterfeiting using Blockchain technology. In addition, the marketplace allows users to buy and sell existing NFTs, as well as to bid at auctions. The platform allows for payment for NFTs in cryptocurrency, which reduces transaction costs and ensures fast and secure exchange of values between users. The project allows for the creation and circulation of NFTs in an efficient and secure manner.

Довідки	Формат	Значення	Найменування	Кіл. листів	№ ек-земпляр	Додаток	
			Документація загальна				
			Знову розроблена				
	<i>A4</i>	<i>ІАЛЦ.467200.002 ТЗ</i>	Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain	4			
			Технічне завдання				
	<i>A4</i>	<i>ІАЛЦ.467200.003 ПЗ</i>	Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain	74			
			Пояснювальна записка				
	<i>A4</i>	<i>ІАЛЦ.467200.004 Д1</i>	Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain	1			
			Діаграма потоків даних (структурна схема)				
	<i>A4</i>	<i>ІАЛЦ.4672008.005 Д2</i>	Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain	1			
			Діаграма класів (функціональна схема)				
	<i>A4</i>	<i>ІАЛЦ.467200.006 Д3</i>	Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain	1			
			Алгоритм дій програмного забезпечення (принципова схема)				
	<i>A4</i>	<i>ІАЛЦ.467200.007 Д4</i>	Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain	4			
			Текст програмного коду				
<i>ІАЛЦ.467200.001 ОА</i>							
<i>Зм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп</i>	<i>Дата</i>			
<i>Розроб</i>	Мусійчук Я. С.				<i>Літ.</i>	<i>Аркуш</i>	
<i>Перев</i>	Нікольський С.С.					<i>Аркушів</i>	
						1	
						1	
				<i>Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain</i>		<i>КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-91</i>	
				<i>Опис альбому</i>			

ТЕХНІЧНЕ ЗАВДАННЯ
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему:

*«Маркетплейс-платформа для торгівлі та створення NFT на базі
Blockchain»*

Київ – 2023

ЗМІСТ

НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
ДЖЕРЕЛА РОЗРОБКИ.....	2
ТЕХНІЧНІ ВИМОГИ.....	3
Вимоги до розробленого продукту	3
Вимоги до програмного забезпечення.....	3
Вимоги до апаратної частини	3
ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.467200.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив	Мусійчук Я. С.				<i>Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain</i> Технічне завдання	Літ.	Аркуш	Аркушів
Перевірив	Нікольський С.С.						1	4
Н. Контр.	Виноградов Ю. М.					КПІ ім. Ігоря		
Затвердив	Стіренко С. Г.					Сікорського, ФІОТ, ІВ-91		

1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання поширюється на розробку маркетплейс-платформи для торгівлі та створення NFT на базі Blockchain.

Областю застосування даного проекту є торгівля та створення унікальних NFT на базі Blockchain. Платформа може бути використана художниками, митцями, дизайнерами, музикантами та іншими творчими професіями для створення та продажу своїх унікальних творів у форматі NFT. Крім того, платформа може бути використана колекціонерами для купівлі та продажу вже існуючих NFT, а також для проведення торгів на аукціонах. Окрім того, платформа може зацікавити інвесторів, які бажають здійснювати операції з цифровими активами, що забезпечує швидкий та безпечний обмін цінностями між користувачами.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проекту по освітньо-професійної програми “Комп’ютерні системи та мережі” спеціальності 123 “Комп’ютерна інженерія”, затверджене кафедрою Обчислювальної техніки Національного технічного Університету України “Київський Політехнічний інститут імені Ігоря Сікорського”.

3 МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного бакалаврського дипломного проекту є розробка маркетплейс-платформи для торгівлі та створення NFT на базі Blockchain, що забезпечує безпечний та ефективний обмін цінностями між користувачами, а також зменшення витрат на операції.

4 ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки є науково-технічна література та статті, що стосуються Blockchain та NFT технологій, документація різних блокчейн платформ та протоколів, а також приклади використання вже існуючих NFT маркетплейсів та інших проектів, що базуються на Blockchain.

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

Розроблена система має виконувати такі вимоги:

- Розробка простого та зручного для користувача інтерфейсу маркетплейсу NFT.
- Надання користувачам змоги вводити дані для створення, купівлі та продажу NFT.
- Можливість для користувачів підключати свій криптогаманець для здійснення транзакцій на платформі.

5.2. Вимоги до програмного забезпечення

- Операційна система: Windows, Mac або Linux.
- Веб-браузер останньої версії (наприклад, Google Chrome, Mozilla Firefox, Safari або Edge).
- Наявність встановленого і налаштованого криптогаманця (MetaMask).

5.3. Вимоги до апаратної частини

- Стабільне інтернет-з'єднання.
- Рекомендоване розширення екрану: 1280x720 або більше для оптимального перегляду вмісту маркетплейсу.
- ROM не менше ніж 64 ГБ.
- RAM не менше ніж 4 ГБ.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

6 ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	16.11.2022-22.11.2022
Вивчення та аналіз завдання	16.11.2022-29.12.2022
Розробка архітектури та загальної структури системи	29.12.2023-07.01.2023
Розробка структур окремих частин системи	07.01.2023-15.01.2023
Програмна реалізація системи	15.01.2023-29.04.2023
Виправлення помилок	01.05.2023-14.05.2023
Оформлення пояснювальної записки	15.05.2023- 03.06.2023

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему:

«Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain»

Київ – 2023

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	6
РОЗДІЛ 1. ОГЛЯД МАРКЕТПЛЕЙС-ПЛАТФОРМ ДЛЯ ТОРГІВЛІ ТА СТВОРЕННЯ NFT НА БАЗІ BLOCKCHAIN.....	8
1.1 NFT	8
1.2 Огляд існуючих NFT маркетплейсів.....	9
1.2.1 OpenSea	9
1.2.2 Nifty Gateway	10
1.2.3 SuperRare.....	12
1.2.4 Rarible.....	14
1.3 Порівняння існуючих NFT маркетплейсів	15
ВИСНОВОК ДО РОЗДІЛУ 1	16
РОЗДІЛ 2. ОГЛЯД ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ	17
2.1 Огляд технології блокчейн.....	17
2.1.1 Основи технології блокчейн	17
2.1.2 Функціонування та структура блокчейну	19
2.1.3 Типи блокчейнів.....	21
2.1.4 Блокчейн Ethereum.....	21
2.2 Огляд технології смарт-контрактів	23
2.2.1 Визначення та характеристики смарт-контрактів	23
2.2.2 Як працюють смарт-контракти.....	24
2.2.3 Застосування смарт-контрактів у NFT.....	25
2.3 Огляд Solidity як мови програмування для розробки смарт-контрактів	25

					ІАЛЦ.467200.003 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	<i>Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain Пояснювальна записка</i>	Літ.	Аркуш	Аркушів
Розробив		Мусійчук Я. С.					1	74
Перевірив		Нікольський С.С.				КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-91		
Реценз.								
Н. Контр.		Виноградов Ю.М.						
Затвердив		Стіренко С. Г.						

2.3.1 Основи Solidity	25
2.3.2 Особливості Solidity	26
2.4 Огляд стандарту ERC-721	27
2.4.1 Опис та характеристики стандарту ERC-721	27
2.4.2 Структура та принцип роботи ERC-721	27
2.4.3 Реалізація NFT з використанням ERC-721	28
2.5 Огляд фреймворку Hardhat	29
2.5.1 Визначення та характеристики Hardhat	29
2.5.2 Вплив Hardhat на розробку децентралізованих додатків	30
2.6 Огляд бібліотеки OpenZeppelin	31
2.6.1 Визначення та особливості OpenZeppelin	31
2.6.2 Наслідування шаблону ERC-721 в OpenZeppelin	32
2.7 IPFS (InterPlanetary File System)	33
2.7.1 Огляд IPFS та його роль у зберіганні NFT	33
2.7.2 Механізми розподіленого зберігання файлів у IPFS	34
2.7.3 Інтеграція IPFS з NFT маркетплейсом	34
ВИСНОВОК ДО РОЗДІЛУ 2	36
РОЗДІЛ 3. ДЕТАЛІ РОЗРОБКИ СИСТЕМИ	38
3.1 Розробка смарт-контракту	38
3.1.1 Вимоги до розробки смарт-контракту	38
3.1.2 Використання модулів OpenZeppelin	39
3.1.3 Структура даних NFTListing	40
3.1.4 Конструктор ERC721	42
3.1.5 Подія NFTTransfer	43
3.1.6 Функція для створення NFT	45
3.1.7 Функція для виставлення NFT на продаж	46
3.1.8 Функція для купівлі NFT	48

3.1.9 Функція для скасування продажу NFT.....	50
3.1.10 Функція для зняття коштів.....	51
3.2 Тестування смарт-контракту за допомогою Hardhat.....	52
3.2.1 Підготовка тестового середовища.....	52
3.2.2 Тестування функцій смарт-контракту	53
3.3 Реалізація механізму авторизації через криптогаманець.....	55
ВИСНОВОК ДО РОЗДІЛУ 3	57
РОЗДІЛ 4. ДОСЛІДЖЕННЯ ТА АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ.....	58
4.1 Огляд головної сторінки.....	58
4.2 Авторизація через криптогаманець MetaMask	60
4.3 Сторінка для створення NFT	63
4.4 Сторінка для перегляду та керування власними NFT	65
4.5 Процес купівлі NFT	67
4.6 Рекомендації щодо розвитку та вдосконалення NFT маркетплейсу .	69
ВИСНОВОК ДО РОЗДІЛУ 4	70
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72

ПЕРЕЛІК СКОРОЧЕНЬ

NFT	(Non-Fungible Token) Невзаємозамінний токен
ERC	(Ethereum Request for Comment) Протокол, що використовується в Ethereum для визначення різноманітних стандартів, особливо для токенів.
DeFi	(Decentralized Finance) Децентралізовані Фінанси, термін, що описує фінансові додатки та продукти, що використовують технологію блокчейн для створення прозорих та відкритих фінансових сервісів.
ETH	(Ethereum) Криптовалюта та платформа для децентралізованих додатків, що працюють на технології блокчейн.
USDC	(United States Dollar Coin) Стейблкоін, що прив'язаний до американського долара, який забезпечує стабільність вартості в світі криптовалют.
PoW	(Proof of Work) Доказ роботи, це консенсусний алгоритм, що використовується в багатьох криптовалютах для вирішення задачі підтвердження транзакцій та додавання нових блоків до блокчейну.
PoS	(Proof of Stake) Доказ застави, це тип алгоритму консенсусу в блокчейні, в якому валідатори блоків вибираються на основі кількості токенів, які вони тримають та ставлять у заставу.
TPS	(Transactions Per Second) Транзакції за секунду, метрика, що вимірює кількість транзакцій, які система може обробляти за одну секунду.
EVM	(Ethereum Virtual Machine) Віртуальна Машина Ethereum, це середовище виконання в Ethereum, що обробляє і виконує смарт-контракти.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

- JSON** (JavaScript Object Notation) Формат обміну даними, що використовується для збереження та передачі даних у форматі, який легко читається для людей та легко обробляється машинами.
- URL** (Uniform Resource Locator) Уніфікований локатор ресурсів, це веб-адреса, що вказує на місцезнаходження ресурсу в інтернеті.
- URI** (Uniform Resource Identifier) Уніфікований ідентифікатор ресурсів, це стрічка символів, що ідентифікує ім'я або ресурс в Інтернеті. URL є підмножиною URI.
- IPFS** (InterPlanetary File System) Протокол та мережева система, призначена для створення постійного та децентралізованого способу зберігання та обміну даними в Інтернеті.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

ВСТУП

За останні роки, Blockchain технології набули популярності у світі, пропонуючи ряд нових можливостей для різних галузей. Серед найбільш цікавих інновацій у цій сфері є NFT (Non-Fungible Token) - цифровий актив, який має унікальну ідентифікацію та незамінність. Він відрізняється від традиційних криптовалют тим, що кожен токен є унікальним та непідмінним, тому не може бути замінений іншим токеном з такою ж цінністю. Ринок NFT стрімко зростає, відкриваючи нові можливості для художників, музикантів, геймерів та інших творчих осіб для створення, продажу та обміну своїми цифровими активами.

У зв'язку з розвитком ринку NFT, виникає потреба у створенні NFT маркетплейсу - платформи, де користувачі зможуть створювати, купувати та продавати свої токени. Основна мета такого маркетплейсу полягає в тому, щоб забезпечити надійність та безпеку транзакцій, забезпечити доступ до широкого асортименту цифрових активів. Розробка такого маркетплейсу вимагає застосування блокчейн технологій, що забезпечать надійність, швидкість та безпеку всіх операцій.

Область використання NFT маркетплейсу є величезною та охоплює багато галузей, включаючи області мистецтва, музики, відеоігор та інших креативних дисциплін. Наприклад, художники можуть створювати цифрові версії своїх творів та продавати їх у вигляді NFT. Музиканти можуть випускати лімітовані серії своїх альбомів або індивідуальних треків у форматі NFT, що забезпечить їм додатковий прибуток та особливу цінність для шанувальників. Геймери та розробники ігор можуть створювати унікальні віртуальні предмети, персонажів або ігрові активи у вигляді NFT, що можуть бути продані або обміняні на інші цінності всередині гри або на спеціалізованому маркетплейсі. Інші творчі особистості, такі як письменники, кінорежисери, дизайнери тощо,

					ІАЛЦ.467200.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

також можуть використовувати токени для монетизації своїх творінь та розширення своєї аудиторії.

У даному дипломному проекті планується розробити NFT маркетплейс на основі Ethereum блокчейну, використовуючи технології Web3 та розробляючи смарт-контракти на мові програмування Solidity. Проект передбачає розробку алгоритмів для створення, купівлі та продажу NFT, а також оптимізацію швидкості та безпеки транзакцій.

Під час розробки проекту будуть враховані існуючі рішення на ринку, аналіз їх переваг та недоліків, а також розробка власного підходу до створення унікального маркетплейсу. Основні цілі проекту полягають у створенні зручного інтерфейсу, надійності транзакцій та доступності сервісу для різних категорій користувачів.

					ІАЛЦ.467200.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

ОГЛЯД МАРКЕТПЛЕЙС-ПЛАТФОРМ ДЛЯ ТОРГІВЛІ ТА СТВОРЕННЯ NFT НА БАЗІ BLOCKCHAIN

1.1 NFT

NFT - це криптовалютний токен, що використовується для позначення унікальних цифрових активів, таких як мистецькі твори, відео, аудіо або інші цифрові об'єкти. Відмінністю NFT від інших криптовалют, таких як Bitcoin або Ethereum, є те, що кожен NFT має свій унікальний ідентифікатор, що робить його необмінним і неповторним.

NFT побудований на базі блокчейн технології, яка забезпечує надійну та безпечну систему для зберігання та обміну активів між користувачами. Блокчейн забезпечує створення децентралізованої бази даних, що дозволяє зберігати інформацію про транзакції з токенами без необхідності довіряти будь-якому централізованому органу.

Криптографічна безпека, властива цим технологіям, забезпечує безпеку та конфіденційність транзакцій, що дає власникам можливість пересувати свої активи без ризику втрати.

Після створення NFT, він може бути проданий або обмінений між користувачами. Ціна токена визначається на основі ринкової вартості та унікальності об'єкта.

Однією з головних переваг таких токенів є те, що вони дозволяють власникам унікальних цифрових активів зберігати та продавати їх з гарантією їхньої автентичності та унікальності. Також, за допомогою NFT, створюється можливість віртуальної власності на цифрових активах, що дозволяє власникам встановлювати контроль доступ до них та отримувати прибуток від їх продажу.

					ІАЛЦ.467200.003 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

Ці характеристики підсилюють розповсюдження та впровадження NFT як нового виду цифрових активів, що може радикально змінити нашу взаємодію з цифровими ресурсами та власністю. [1, 2]

1.2 Огляд існуючих NFT маркетплейсів

1.2.1 OpenSea

OpenSea - це один з найбільших маркетплейсів NFT, який дозволяє користувачам купувати, продавати та взаємодіяти з цифровими активами, які представлені у форматі NFT.

Маркетплейс має ряд ключових функцій та можливостей для користувачів. Він надає можливість створювати та продавати унікальні NFT, тобто користувачі можуть вільно маніфестувати свою творчість та монетизувати свої власні токени. Щодо пошуку та покупки, платформа дозволяє користувачам відшукати та придбати токени з широкого спектра категорій, включаючи мистецтво, музику, відеоігри, домени та інше.

Що стосується торгівлі, користувачі можуть виставляти свої NFT на продаж на відкритому ринку, а також брати участь в аукціонах. Більш того, платформа надає можливість управління портфелем NFT, де користувачі можуть відслідковувати та керувати своїми NFT, вивчаючи їх вартість та історію транзакцій.

OpenSea підтримує різні типи NFT, включаючи малюнки, відео, музику та інші цифрові активи, створені на різних блокчейнах, таких як Ethereum, Polygon, Binance Smart Chain та Flow. Крім того, платформа надає користувачам можливість взаємодіяти з токенами, що підтримують ERC-20 токени, такі як криптовалюти та токени DeFi.

Типи доступних NFT на платформі також включають групові NFT, які можуть бути випущені колективами або індивідуальними авторами. Вони можуть включати музику, відео, інші форми мистецтва та забезпечувати

					ІАЛЦ.467200.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

інвесторів змогою інвестувати в культурні проекти. Це робить OpenSea одним з найбільш різноманітних маркетплейсів NFT на ринку.

Комісії становлять 2,5% від суми продажу плюс додаткові оплати за транзакції, що сплачуються у валюті Ethereum (ETH). Методи оплати включають оплату в Ethereum або стейблкоїнах, таких як USDC та DAI. Проте, платформа не значною мірою уступати по кількості функцій іншим конкурентам, таким як Nifty Gateway або SuperRare. Однак, OpenSea залишається однією з найбільш популярних платформ NFT завдяки своїй широкій базі користувачів та різноманітним типам токенів, які доступні для торгівлі.

Процес відправки та отримання NFT на маркетплейсі досить простий та зручний. Для початку продавець має створити акаунт на платформі та завантажити свій токен на маркетплейс. Після цього він може встановити ціну за свій NFT та обрати методи оплати.

Той, хто бажає придбати актив, у свою чергу, може легко знайти необхідний йому NFT на маркетплейсі за допомогою пошуку або вибрати з рекомендованих лотів. Після того, як він знайшов потрібний йому токен, він може купити його за встановленою продавцем ціною.

Платформа має високі стандарти захисту та безпеки. Кожен токен на платформі має унікальний ідентифікатор, який можна перевірити на блокчейні, щоб підтвердити його власність. OpenSea також застосовує механізм двофакторної автентифікації для захисту вхідних даних користувачів. Крім того, платформа використовує шифрування для захисту конфіденційної інформації користувачів, такої як паролі та приватні ключі. [3]

1.2.2 Nifty Gateway

Nifty Gateway - це маркетплейс, спеціалізований на продажу та купівлі цифрових мистецьких творів в форматі NFT. Заснована в 2018 році, платформа

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

дозволяє художникам, дизайнерам та іншим творчим особистостям продавати свої цифрові твори широкій аудиторії.

Типи доступних NFT на платформі різноманітні і охоплюють різні сфери творчості, такі як малюнки, графіку, відео, музику тощо. Найпопулярнішими на платформі є NFT відомих художників, дизайнерів та музикантів.

Однією з унікальних особливостей Nifty Gateway є миттєва видача купленого NFT. Це означає, що після успішної оплати NFT, покупець отримує доступ до свого цифрового твору миттєво, без очікування на підтвердження транзакції.

Процес відправки та отримання токенів на Nifty Gateway досить схожий на OpenSea. Користувачі можуть легко створювати свої NFT та розміщувати їх на маркетплейсі. Для покупців процес купівлі також досить простий - вони можуть шукати та купувати NFT, що їх цікавлять.

Комісійні відсотки на Nifty Gateway є дещо вищими, ніж на OpenSea, і становлять 5% від кожної транзакції. Цікавою особливістю є те, що при повторному продажу токенів, 10% буде відправлено їх першому власнику. Способи оплати включають криптовалюту та кредитні картки. Особливості захисту та безпеки також включають двофакторну аутентифікацію та заборону перенесення NFT з одного гаманця на інший без відповідного підтвердження власника. [4]

Nifty Gateway є більш спеціалізованим маркетплейсом порівняно з OpenSea, тому він має кілька додаткових функцій та особливостей, яких немає в OpenSea:

1. Платформа дозволяє художникам створювати збірки своїх токенів, щоб продавати їх разом. Це дозволяє художникам продавати свої твори як пакети, що додає вартість їхнім NFT.
2. Художники мають можливість вказати, які NFT є рідкісними та обмеженими за кількістю, що додає вартість їхнім творам.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

3. Платформа дозволяє художникам додавати оновлення до своїх NFT, що дає можливість створити пов'язані твори та колекції.
4. Можливість перепродажу дозволяє збільшити ліквідність NFT та залучити нових покупців.
5. Маркетплейс співпрацює з відомими художниками та брендами, що дозволяє створювати ексклюзивні та обмежені NFT колекції.

1.2.3 SuperRare

Маркетплейс SuperRare - це платформа для продажу унікальних цифрових мистецьких творів у форматі NFT. Ресурс ставить своєю метою забезпечення можливості взаємодії між творцями та колекціонерами, які шукають найбільш якісні й унікальні NFT.

Платформа пропонує ряд ключових функцій та можливостей для своїх користувачів. Вона надає можливість купувати та продавати унікальні NFT-шедеври від відомих митців по всьому світу. Щодо інтеграції, платформа сумісна з гаманцями, які підтримують Ethereum (ETH). Додатково, вона має систему рекомендацій, яка допомагає користувачам відкривати захоплюючі та унікальні твори від провідних художників світу. Більш того, є система аукціонів, що дозволяє відслідковувати і підвищувати ціну на обрані твори протягом визначеного часу.

Так само, як і попередньо розглянуті маркетплейси, платформа пропонує широкий вибір NFT, включаючи малюнки, відео, музику, 3D-моделі та інші цифрові формати мистецтва. Крім того, маркетплейс активно розвивається й постійно оновлюється з метою забезпечення максимального комфорту та задоволення користувачів.

Маркетплейс також відомий своєю співпрацею з відомими художниками та дизайнерами, що забезпечує високу якість та унікальність доступних токенів.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

Кожен NFT на платформі має власний унікальний номер та підпис автора. Крім того, присутня можливість створення власних колекцій NFT, додавання NFT до списку обраного та побудова власного профілю художника чи колекціонера.

Щодо оплати, вона здійснюється через криптовалюту Ethereum. Платформа забезпечує безпеку транзакцій та відстеження власності NFT після їх покупки. Комісійні відсотки складають 3% від суми продажу NFT. [5]

У порівнянні з OpenSea, SuperRare має функції, більш спеціалізовані на продаж високоякісних NFT. Основні відмінності включають наступне:

1. Платформа зосереджується на високоякісних NFT, тому всі роботи, що надходять на платформу, пройшли суворий відбір. Такий підхід дозволяє зберегти відвідувачам час на пошуки кращих NFT.
2. Присутній власний алгоритм контролю якості для кожного NFT, що дозволяє переконатися в його автентичності та унікальності.
3. У маркетплейсу є команда кураторів, які відповідають за відбір і курування NFT. Це дозволяє створити колекції з однієї тематики або жанру, що допомагає швидше знайти цікаві роботи для покупців.
4. Можливість голосувати за творчість, взаємодіяти з авторами та відвідувачами.

Процес відправки та отримання NFT:

- Після того, як NFT був відправлений на платформу, він проходить процес перевірки, щоб забезпечити відповідність стандартам якості платформи. Якщо NFT відповідає вимогам, він буде розміщений на ринку для продажу.
- Після успішної оплати, використовуючи криптовалюту Ethereum або кредитні / дебетові картки через платіжну систему Stripe, NFT переходить до гаманця покупця на платформі.

Щодо особливостей захисту та безпеки, то платформа використовує двофакторну автентифікацію для захисту від несанкціонованого доступу до

					ІАЛЦ.467200.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

облікових записів користувачів. Кожен NFT, що розміщений на SuperRare, має унікальний хеш-код, що забезпечує неможливість його підробки або копіювання. [5, 6]

1.2.4 Rarible

Rarible - це маркетплейс, що дозволяє користувачам створювати, продавати та купувати унікальні цифрові активи на блокчейні Ethereum. Маркетплейс був запущений в 2020 році і став дуже популярним серед художників та творців контенту.

Платформа надає користувачам ряд цікавих функцій та можливостей. На першому місці стоїть здатність створювати власні NFT, використовуючи медіафайли різних форматів, таких як зображення, аудіофайли або відео. По-друге, платформа слугує місцем для покупки та продажу NFT за допомогою різноманітних криптовалют, таких як Ethereum, DAI і USDT. Крім того, Rarible надає можливість проведення аукціонів для тих, хто бажає продати свої NFT через конкурентний процес. І нарешті, платформа має власні галереї, де користувачі можуть переглядати і придбати NFT, створені відомими художниками та творцями.

Із особливостей даного маркетплейсу можна виділити розширені функції для творців NFT, такі як можливість продажу лише одного екземпляру NFT або встановлення комісії за подальший продаж.

Комісійні відсотки в Rarible зазвичай становлять 2,5% від ціни продажу NFT. Однак, користувачі можуть встановлювати власні комісії. Крім того, на Rarible є спеціальна програма, яка дозволяє отримувати знижки на комісійні відсотки, якщо користувач має власний RARI токен.

Щодо способів оплати, Rarible приймає різні криптовалюти, такі як Ethereum, Binance Smart Chain, Polygon тощо. Також доступні традиційні способи оплати, такі як PayPal та кредитні картки. [7]

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

1.3 Порівняння існуючих NFT маркетплейсів

OpenSea виявився найбільшим і найбільш універсальним маркетплейсом, який підтримує широкий спектр NFT, включаючи мистецтво, доменні імена, віртуальну нерухомість та інше. Його основна перевага - це масштабність і гнучкість, але це може створювати додаткову складність для нових користувачів.

NiftyGateway вирізняється своїм фокусом на цифровому мистецтві. З одного боку, це забезпечує більш цілісний досвід для митців та колекціонерів мистецтва. З іншого боку, це обмежує його застосування для інших типів NFT.

SuperRare має більшу зосередженість на одноразових випусках цифрового мистецтва, що робить його привабливим для колекціонерів, які шукають ексклюзивність. Однак, це також може обмежити його привабливість для більш широкого спектра NFT.

Rarible акцентує свою увагу на децентралізації і сприянні рівних можливостей у процесі створення та обігу NFT. Це відкриває двері для широкого кола користувачів, однак, можуть виникнути проблеми з контролем якості та модерацією контенту.

Що стосується комісій, вони можуть значно варіюватися в залежності від платформи. OpenSea, наприклад, має нижчі комісії для продавців, але вони можуть бути вищими для покупців. NiftyGateway і SuperRare, з іншого боку, мають стабільні комісії, але вони можуть бути вищими за загальний рівень ринку. Rarible пропонує змінні комісії з урахуванням динаміки ринку.

У процесі створення та купівлі NFT важливо відзначити, що кожна платформа має свої власні процеси і інструменти. OpenSea і Rarible пропонують більш відкриті та гнучкі інструменти, тоді як NiftyGateway і SuperRare можуть пропонувати більш керований досвід.

					ІАЛЦ.467200.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВОК ДО РОЗДІЛУ 1

Після детального аналізу та оцінки найпопулярніших NFT маркетплейсів за рядом критеріїв, включаючи основні та додаткові функції, унікальні відмінності, вартість комісії, процес створення та купівлі NFT, системи захисту та інші особливості можна стверджувати, що розроблювана система NFT маркетплейсу буде конкурентоспроможною, якщо вона забезпечує наступні ключові аспекти:

1. Система повинна бути інтуїтивно зрозумілою та легкою у використанні, що дозволяє користувачам створювати, купувати та продавати NFT без складних процедур.
2. Вартість використання системи повинна бути конкурентною на ринку. Сюди враховуються комісії за створення, купівлю та продаж NFT.
3. Платформа має надавати користувачам можливість переглядати історію транзакцій, статистику та іншу важливу інформацію про свої NFT. Це може включати в себе історію власності, ціну продажу, середній час між продажами та інше.
4. Система повинна мати надійні механізми захисту, щоб гарантувати безпеку цифрових активів користувачів.
5. Система повинна легко інтегруватися з іншими популярними платформами та сервісами, такими як популярні криптовалютні гаманці.

З урахуванням цих факторів, розроблювана система має потенціал стати конкурентоспроможною на ринку NFT маркетплейсів. Успіх системи залежатиме від того, наскільки ефективно вона зможе втілити ці характеристики, а також від того, як добре вона відповідатиме на зміни в тенденціях ринку та потребах користувачів. Таким чином, розроблювана система повинна також зосереджуватися на тому, щоб створювати відкрите середовище, яке підтримує різноманітність митців та творчих виразів, а також стимулює взаємодію та обмін в рамках спільноти.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

РОЗДІЛ 2

ОГЛЯД ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ

2.1 Огляд технології блокчейн

2.1.1 Основи технології блокчейн

Основоположною концепцією технології блокчейн є децентралізована, розподілена система, яка реєструє транзакції на багатьох комп'ютерах таким чином, що зареєстровані транзакції не можуть бути змінені заднім числом. Ця особливість підвищує безпеку та прозорість усіх дій у системі. Блокчейн, по суті, є ланцюжком блоків, де кожен блок містить дані, що стосуються транзакції.

Кожен блок складається з криптографічного хешу попереднього блоку, часової мітки та даних про транзакцію, які зазвичай представлені у вигляді дерева Меркла - структури даних, що використовується в блокчейн технології для забезпечення цілісності та швидкої перевірки даних, зокрема транзакцій.

Дерево Меркла складається з вузлів, які представляють хеші (криптографічні підписи) даних. Верхній вузол дерева називається "коренем". Кожен батьківський вузол вираховується шляхом хешування хешів своїх дочірніх вузлів. Процес повторюється до тих пір, поки не буде досягнутий корінь дерева, який представляє загальний хеш всього набору даних.

Застосування дерева Меркла в блокчейн технології полягає в забезпеченні цілісності блоків та транзакцій. Кожен блок містить хеш свого заголовка, який включає в себе інформацію про попередній блок. При створенні нового блоку, його заголовок обчислюється шляхом хешування даних блока, включаючи хеш попереднього блоку. Таким чином, будь-яка зміна в будь-якому блоку вплине

на хеші всіх наступних блоків, що робить маніпулювання даними набагато складнішим.

Крім того, дерева Меркла дозволяють ефективно перевіряти належність даних до множини. Замість того, щоб перевіряти кожен елемент окремо, можна використовувати дерево для швидкої перевірки наявності або відсутності елемента у наборі даних.

Загалом, це дозволяє блокчейну забезпечити швидку перевірку та ефективну роботу з великими наборами даних, що робить їх важливою складовою блокчейн технології. [8]

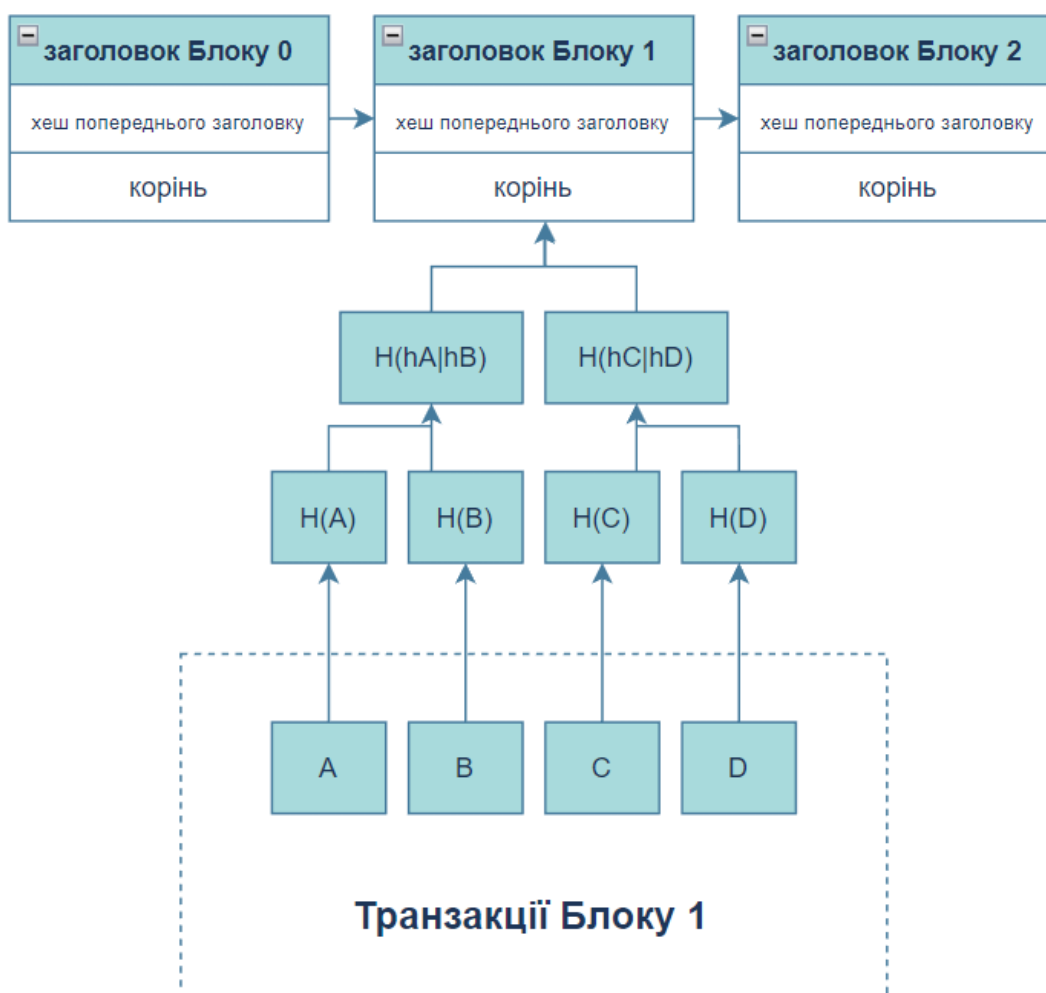


Рисунок 2.1 - Дерево Меркла

Дизайн ланцюжка за своєю суттю протистоїть модифікації даних. Система вимагає консенсусу більшості учасників мережі, що робить її надзвичайно безпечною технологією.

2.1.2 Функціонування та структура блокчейну

Як вже було згадано, структура блокчейну полягає в утворенні ланцюжка блоків, де кожен блок містить посилання на попередній блок шляхом хешування. Це, у свою чергу, створює послідовність блоків, яка зберігає всю історію транзакцій. Додавання нового блоку до ланцюжка відбувається після виконання перевірок валідності і підтвердження консенсусу.

Механізм консенсусу - це протокол або алгоритм, за допомогою якого учасники в мережі блокчейн досягають згоди щодо єдиного стану глобального реєстру транзакцій. Оскільки блокчейн є розподіленою системою, де кожен вузол може мати свою версію блокчейну, механізм консенсусу дозволяє учасникам досягти і підтримувати консистентну і правильну версію блокчейну.

Основна мета механізму консенсусу полягає в тому, щоб усі учасники мережі погодилися про те, який блок має бути доданий до ланцюжка блоків і які транзакції повинні бути підтверджені. Це досягається шляхом визначення правил і протоколів, які керують процесом вирішення конфліктів і прийняття рішень.

Різні протоколи консенсусу використовують різні методи для досягнення згоди. Наприклад, Proof of Work (PoW) вимагає великої обчислювальної потужності для розв'язання обчислювальних задач і отримання права створювати блоки. Proof of Stake (PoS) використовує володіння криптовалютою або її ставку як критерій для отримання права на створення блоків.

Механізми консенсусу також можуть включати інші елементи, такі як рейтингові системи, голосування, делеговані вузли та інші. Вибір конкретного механізму консенсусу залежить від конкретної блокчейн мережі і її потреб.

					ІАЛЦ.467200.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

Кожен механізм має свої переваги та недоліки, і вибір правильного механізму консенсусу є важливим завданням для стабільної та безпечної роботи блокчейну. [9]

Процес додавання нового блоку до ланцюжка включає кілька кроків:

1. Перш за все, новий блок проходить перевірку валідності. Це включає перевірку підписів транзакцій, правильності хешів і дотримання всіх правил протоколу.
2. Учасники мережі порівнюють новий блок зі своєю версією ланцюжка блоків. Вони перевіряють, чи відповідає новий блок певним правилам (наприклад, відповідний попередньому блоку за хешем) і чи має він найбільшу валідну роботу (у випадку PoW).
3. Якщо новий блок відповідає всім правилам, учасники мережі погоджуються, що це правильна версія ланцюжка. Вони виконують необхідні кроки, щоб підтвердити цей блок і прийняти його як частину ланцюжка.
4. Після підтвердження блоку він поширюється по мережі, щоб інші вузли також оновили свої версії ланцюжка блоків.
5. Після додавання нового блоку до ланцюжка він стає частиною постійного реєстру транзакцій, який є основою блокчейну.

Цей процес повторюється з кожним новим блоком, який формує ланцюжок блоків. Кожен блок містить посилання на попередній блок, що забезпечує послідовність та впорядкованість блоків у ланцюжку.

Після того, як блок додається до блокчейну, він стає загальнодоступним для перегляду будь-ким, включаючи такі деталі, як час створення блоку, дані про транзакцію та її учасника. Враховуючи прозорий характер блокчейну, фальсифікація даних практично неможлива. [10]

					ІАЛЦ.467200.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

2.1.3 Типи блокчейнів

Існує три основні типи блокчейнів: публічні блокчейни, приватні блокчейни та консорціумні блокчейни.

Публічні блокчейни: Це блокчейни з відкритим вихідним кодом, в яких будь-хто може брати участь як користувач, майнер, розробник або член спільноти. Транзакції, які відбуваються в цій мережі, повністю прозорі, і будь-хто може вільно переглядати їх. Bitcoin та Ethereum є яскравими прикладами публічних блокчейнів.

Приватні блокчейни: Вони не відкриті для всіх і вимагають запрошення, підтвердженого ініціатором мережі. Зазвичай вони використовуються бізнесом, який хоче використовувати технологію блокчейн, зберігаючи при цьому конфіденційність своїх даних. Прикладом приватної блокчейн-платформи є Hyperledger Fabric.

Консорціумні блокчейни: Також відомі як федеративні блокчейни, вони є напівдецентралізованими і працюють під керівництвом групи. На відміну від публічних блокчейнів, вони не дозволяють будь-якій особі з доступом до інтернету брати участь у процесі перевірки транзакцій. Консорціумні блокчейни здебільшого використовуються в банківському секторі. Прикладом є R3 Corda.

Кожен з цих типів блокчейнів має свої переваги та недоліки, і вибір між ними залежить від конкретного випадку використання та вимог. [11]

2.1.4 Блокчейн Ethereum

Для розробки системи було обрано блокчейн Ethereum, що є одним з найвідоміших і найпоширеніших блокчейнів в світі. Ефіріум (Ethereum) був створений у 2015 році і став першим блокчейном, що вніс концепцію смарт-контрактів, тобто програм, що виконуються безпосередньо на блокчейні без необхідності третіх сторін.

					ІАЛЦ.467200.003 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

Переваги Ethereum:

1. Даний блокчейн підтримує мову програмування Solidity, що дозволяє розробникам писати дуже гнучкі смарт-контракти з різноманітними функціями.
2. Він має одну з найбільших спільнот розробників у сфері блокчейну, що забезпечує швидке вирішення проблем та постійні оновлення.
3. Ethereum використовує алгоритм доказу ставки (Proof of Stake), що підвищує гарантії безпеки та ефективності. [12]

Таблиця 2.1 – Порівняння Ethereum з іншими популярними блокчейнами

	Ethereum	Bitcoin	Cardano	Polkadot
Рік створення	2015	2009	2017	2020
Смарт-контракти	Так	Ні	Так	Так
Мова програмування	Solidity	Script	Plutus, Solidity	Rust, JavaScript
Консенсус	PoS	PoW	PoS	PoS
Максимальна кількість токенів	Необмежено	21 мільйон	45 мільярдів	1 мільярд
Швидкість транзакцій	15-20 tps	4-7 tps	250 tps	1000 tps
Спільнота розробників	Велика	Велика	Середня	Велика

Ethereum має багато переваг в порівнянні з іншими блокчейнами, зокрема підтримку смарт-контрактів, велику спільноту розробників та гнучкість в розробці децентралізованих програм.

2.2 Огляд технології смарт-контрактів

2.2.1 Визначення та характеристики смарт-контрактів

Смарт-контракт - це контракт, який виконується самостійно, а умови угоди безпосередньо записані в кодї. Вони дозволяють здійснювати транзакції та укладати угоди між різними анонімними сторонами без потреби в центральному органі влади, правовій системі або зовнішньому механізмі примусового виконання.

Характеристики смарт-контрактів:

- Вони володіють автономністю, оскільки після активації можуть функціонувати незалежно без потреби в додаткових діях зі сторони ініціатора.
- Це децентралізовані інструменти, що працюють на базі блокчейну, забезпечуючи тим самим, що ніхто не має абсолютного контролю над ними.
- Безпека є ще однією їхньою ключовою властивістю, оскільки дані в смарт-контракті шифруються та зберігаються на блокчейні, що робить їх відмінно захищеними від злону.
- Вони мають високу точність і мінімізують ризик людської помилки, що може виникнути під час ручного введення даних.
- Додатково, смарт-контракти пропонують швидкість та ефективність, виконуючи та підтверджуючи умови контракту автоматично, що зменшує час обробки та усуває потребу в посередниках.
- Вони також надійні, забезпечуючи прозорість через реєстрацію кожної транзакції в публічному реєстрі, при цьому жодна із сторін не може змінити умови після того, як контракт стане дієвим.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

2.2.2 Як працюють смарт-контракти

Смарт-контракти працюють, дотримуючись попередньо встановлених правил. Як тільки ці умови виконуються, смарт-контракт автоматично виконує угоду, на яку він запрограмований. Ці правила записуються у вигляді коду і розгортаються на блокчейні. [13]

Наприклад, якщо особа А і особа Б домовилися, що А заплатить Б 5 ЕТН, якщо буде виконана певна умова, то для цієї угоди можна написати смарт-контракт. Як тільки зазначена умова настане, смарт-контракт автоматично переведе 5 ЕТН від А до Б.

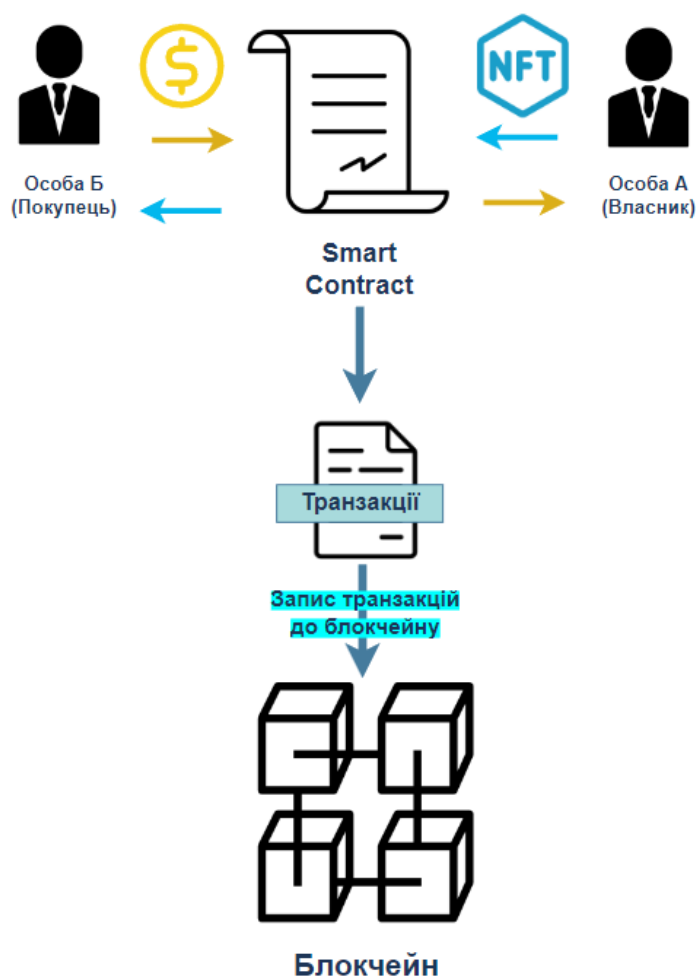


Рисунок 2.2 – Відображення етапів укладання угоди з використанням смарт-контрактів

Зм.	Арк.	№ докум.	Підпис	Дата

Дійсність транзакції та її виконання не потребують участі третіх осіб, оскільки смарт-контракт є одночасно і угодою, і механізмом виконання.

2.2.3 Застосування смарт-контрактів у NFT

Смарт-контракти відіграють вирішальну роль у створенні, володінні та транзакціях NFT.

Коли NFT створюється, пишеться смарт-контракт, який розміщується в блокчейні. Цей контракт включає правила та деталі, характерні для NFT, такі як ідентифікація власника та метадані активу.

Смарт-контракт, пов'язаний з NFT, встановлює право власності на актив. Він зберігає дані про поточного власника і минулі транзакції, забезпечуючи достовірне походження NFT. Більш того, коли NFT купується або продається, транзакція виконується за допомогою смарт-контракту. Контракт гарантує, що платіж буде здійснено і право власності на NFT перейде до нового власника.

Вони також можуть бути запрограмовані на обробку роялті. Це означає, що оригінальні творці можуть отримувати відсоток від продажів кожного разу, коли NFT перепродаються. [14]

Таким чином, смарт-контракти автоматизують і захищають створення, володіння і торгівлю NFT, роблячи їх невід'ємною частиною екосистеми NFT.

2.3 Огляд Solidity як мови програмування для розробки смарт-контрактів

2.3.1 Основи Solidity

Solidity - це об'єктно-орієнтована мова високого рівня для реалізації смарт-контрактів. Вона була спроектована і розроблена Ethereum, і в першу чергу використовується для смарт-контрактів Ethereum. Вона має статичну

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

типізацію, підтримує успадкування, бібліотеки та складні типи, визначені користувачем.

Синтаксис Solidity схожий на синтаксис JavaScript, що робить її більш доступною мовою для розробників, які працювали в контексті веб-розробки. Вона розроблена для роботи з віртуальною машиною Ethereum (EVM), яка є середовищем виконання смарт-контрактів в Ethereum. [15, 16]

2.3.2 Особливості Solidity

Смарт-контракт Solidity - це набір коду (його функцій) і даних (його стан), які знаходяться за певною адресою в блокчейні Ethereum. Ось деякі важливі особливості Solidity:

- Solidity включає в себе станові змінні, які представляють значення, що постійно зберігаються в контрактному сховищі та оголошуються вгорі контракту.
- Другою особливістю є функції, виконувані одиниці коду в контракті, які можуть бути як загальнодоступними, так і внутрішніми, подібно до захищених в інших мовах програмування, або приватними та зовнішніми.
- Також існують модифікатори, які змінюють поведінку функцій і розміщуються перед круглими дужками оголошення функції.
- Події є ще одною особливістю, що дозволяють легко використовувати механізми EVM для ведення журналів. Додатки можуть підписуватися на ці події та слухати їх через блокчейн Ethereum.
- Solidity також підтримує множинне успадкування, реалізоване через копіювання коду, і включає поліморфізм.
- Помилки обробляються за допомогою виключень, які повертають стан і скасовують всі зміни в стані під час поточного виклику та всіх його підвикликів, інформуючи викликача про помилку.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

- Мова пропонує ряд елементарних типів даних, які можна комбінувати для створення більш складних типів.. [15]

2.4 Огляд стандарту ERC-721

2.4.1 Опис та характеристики стандарту ERC-721

ERC-721 - це стандарт представлення NFT в блокчейні Ethereum. Це інтерфейс, який визначає набір стандартних функцій для управління NFT, що дозволяє легко обмінюватися та керувати ними децентралізовано. Стандарт був розроблений, щоб бути надійним і гнучким, щоб відповідати широкому спектру випадків використання токенами.

Характеристики стандарту включають:

- Кожен токен відзначається правом власності, при цьому стандарт забезпечує методи перевірки балансу адреси та переказу токенів.
- Методи перерахування токенів входять до стандарту, що може бути корисно для контрактів, що прагнуть впливати на всі NFT.
- Особливості стандарту охоплюють невзаємозамінність, що робить кожен токен унікальним з відмінними характеристиками та вартістю, відрізняючись від взаємозамінних токенів, де всі одиниці є ідентичними.
- Щодо метаданих, токени можуть мати пов'язані з ними деталі, такі як ім'я, символ та URL-адреса, де можна знайти файл JSON, який описує токен.

2.4.2 Структура та принцип роботи ERC-721

Токен ERC-721 має ряд основних операцій, які відіграють важливу роль у його функціонуванні. Ці операції забезпечують гнучкість та безпеку при взаємодії з цими унікальними токенами.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Одна з таких операцій - це запит балансу. Функція `balanceOf` дозволяє отримати кількість всіх NFT, призначених на конкретну адресу. Це дає можливість швидко оцінити кількість активів, що належать конкретному власнику.

Функція `ownerOf` використовується для отримання інформації про власника конкретного токена, що є важливим для перевірки прав власності.

Операція передачі, або `transferFrom`, дозволяє передати право власності на актив з однієї адреси на іншу, що забезпечує можливість обміну та продажу токенів.

Функція `approve` використовується для надання дозволу конкретній адресі на передачу права власності на певний токен.

Щоб дозволити або заборонити третій стороні ("оператору") керувати всіма своїми активами, використовується функція `setApprovalForAll`.

І, нарешті, для перевірки, чи дозволено оператору керувати всіма активами власника, використовується функція `isApprovedForAll`. Це дозволяє власнику контролювати, хто може управляти його активами. [17]

2.4.3 Реалізація NFT з використанням ERC-721

Для створення NFT з використанням стандарту ERC-721 необхідно написати і розгорнути смарт-контракт на блокчейні Ethereum.

Розробка смарт-контракту полягає в реалізації інтерфейсу ERC-721, включаючи у себе функції, описані в цьому стандарті. Функції, як передача токенів та перевірка права власності, є деякими з ключових елементів, що формують основу смарт-контракту.

Метадані теж можуть бути інтегровані з NFT, використовуючи розширення `ERC721Metadata`, що дозволяє додати унікальне ім'я, символ та URI для JSON-файлу, що описує токен. Це збільшує інформаційну цінність кожного створеного токена.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

Процес створення нових токенів, відіграє важливу роль в контракті. Кожен новий токен отримує унікальний ідентифікатор, що пов'язує його з власником. Це забезпечує їх незамінність та унікальність.

Розгортання контракту відбувається після його компіляції, через що він стає доступним в мережі Ethereum. Від цього моменту, контракт можна використовувати для створення, передачі та управління токенами. [18]

Стандарт гарантує, що NFT будуть сумісні з будь-якими додатками або сервісами, що підтримують цей стандарт, включаючи більшість гаманців Ethereum та багато NFT маркетплейсів.

2.5 Огляд фреймворку Hardhat

2.5.1 Визначення та характеристики Hardhat

Hardhat представляє собою комплексний інструментарій для розробки програмного забезпечення для Ethereum, який об'єднує в собі можливості системи виконання завдань, середовища для тестування та фреймворку для розробки. Його мета - спростити роботу розробників, надаючи інструменти для управління та автоматизації рутинних задач, що супроводжують процес створення смарт-контрактів та децентралізованих додатків. [19]

Тестування смарт-контрактів є критично важливим кроком у процесі розробки децентралізованих застосунків. Адже смарт-контракти не застраховані від помилок у програмуванні. Але, на відміну від помилок в звичайних програмах, помилка, допущена в смарт-контракті, виправляється набагато складніше і може дорого коштувати. Це пов'язано з рядом наступних факторів:

По-перше, через незворотність транзакцій на блокчейні, помилки можуть призвести до втрати великої кількості коштів.

По-друге, безпека смарт-контрактів є основою довіри до системи, а виправлення помилок після розгортання смарт-контракту може бути вкрай

					ІАЛЦ.467200.003 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

важким і дорогим процесом, оскільки смарт-контракти в Ethereum та більшості інших блокчейн платформ є незмінними після того, як вони були розгорнуті. Це означає, що їх код не можна змінити або оновити.

Hardhat має кілька характеристик, які відрізняють його від інших інструментів розробки Ethereum:

- Система має вбудовану консоль для ведення логів і набір інструментів для тестування. Він також надає детальні повідомлення про помилки і трасування стеку, які є більш інформативними, ніж ті, що надаються деякими іншими інструментами.
- Функціональність фреймворку можна розширити за допомогою плагінів. Це забезпечує високий ступінь гнучкості і дозволяє розробникам пристосовувати інструмент до своїх конкретних потреб.
- Він також дозволяє розробникам легко перемикатися між різними мережами Ethereum, будь то основна мережа, тестові мережі або локальна мережа розробки.
- Hardhat включає локальну EVM, що дозволяє розробникам запускати скрипти і тести без підключення до реальної мережі Ethereum. Ця локальна EVM також дозволяє використовувати функцію `console.log()` в Solidity, яка є унікальною для даного фреймворку.

2.5.2 Вплив Hardhat на розробку децентралізованих додатків

Hardhat суттєво вплинув на ландшафт розробки Ethereum. Надаючи ефективне, гнучке та дружнє до розробників середовище, він спростив процес створення та тестування смарт-контрактів та децентралізованих програм.

Розширені можливості налагодження та інформативні повідомлення про помилки Hardhat скоротили час, який розробники витрачають на виявлення та вирішення проблем. Його мережева діагностика полегшує плавний перехід між

					ІАЛЦ.467200.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

різними мережами Ethereum, роблячи розробку і тестування більш ефективними.

Крім того, локальна емуляція EVM від Hardhat дозволяє розробникам тестувати свій код, не покладаючись на реальну мережу Ethereum, заощаджуючи час та мережеві платежі.

2.6 Огляд бібліотеки OpenZeppelin

2.6.1 Визначення та особливості OpenZeppelin

OpenZeppelin - це бібліотека з відкритим вихідним кодом для створення та управління безпечними, багаторазовими та перевіреними смарт-контрактами на платформі Ethereum. Вона пропонує колекцію стандартизованих, перевірених в боях шаблонів контрактів та інструментів, які мають на меті зменшити складність та ризики, пов'язані з розробкою смарт-контрактів. Бібліотека широко використовується спільнотою розробників Ethereum і часто вважається основним ресурсом для безпечного кодування смарт-контрактів.

Деякі ключові особливості OpenZeppelin включають наступне:

- Контракти проходять перевірку експертами з безпеки, що забезпечує високий рівень безпеки і надійності наданого коду.
- Інша ключова риса - модульність. Бібліотека створена таким чином, що враховує модульність, надаючи розробникам можливість створювати та розширювати існуючі контракти, або використовувати її компоненти для створення власних контрактів.
- Ця система також підтримує оновлення, що дає можливість розробляти смарт-контракти, які можна оновлювати, дозволяючи розгортати нові версії контрактів, не втрачаючи при цьому їх поточний стан.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

- OpenZeppelin підтримується та оновлюється спільнотою розробників, що робить його ресурсом, який постійно розвивається. [20]

2.6.2 Наслідування шаблону ERC-721 в OpenZeppelin

OpenZeppelin надає готовий шаблон для контрактів ERC-721, який можна наслідувати, щоб створити власні унікальні NFT. Контракт ERC-721 в бібліотеці включає всі необхідні функції, що відповідають стандарту, а також додаткові функції, які полегшують роботу з токенами.

Створення власного NFT контракту, відбувається шляхом створення нового, який наслідується від базового ERC-721 контракту OpenZeppelin. Після цього можна додати власну логіку або налаштування, зберігаючи при цьому всі основні функції оригіналу. [21]

На рисунку 2.3 наведено базовий приклад того, як це може виглядати в Solidity:

```
pragma solidity ^0.8.0;
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
contract MyNFT is ERC721 {
    uint256 private _tokenIdCounter;
    constructor() ERC721("MyNFT", "MNFT") {
        _tokenIdCounter = 0;
    }
    function createNFT(address to) public returns (uint256) {
        _tokenIdCounter += 1;
        _mint(to, _tokenIdCounter);
        return _tokenIdCounter;
    }
}
```

Рисунок 2.3 – Вигляд смарт-контракту, що наслідується від стандарту ERC721 бібліотеки OpenZeppelin

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

У цьому прикладі, контракт MyNFT наслідується від ERC-721 з OpenZeppelin. Це дозволяє MyNFT використовувати всі стандартні функції ERC-721, такі як balanceOf, ownerOf, transferFrom і так далі. Додаткова функція createNFT була додана для створення нових NFT і автоматичного присвоєння їм унікального ідентифікатора.

Дана бібліотека надає змогу створювати значно більш складні контракти NFT, додаючи власну логіку і налаштування за потреби.

2.7 IPFS (InterPlanetary File System)

InterPlanetary File System (IPFS) - це протокол розподіленого зберігання файлів, який призначений для створення постійного і децентралізованого способу зберігання та обміну даними в інтернеті. IPFS є відкритим стандартом і використовує технологію блокчейн для забезпечення безпечного і постійного зберігання інформації. [22]

2.7.1 Огляд IPFS та його роль у зберіганні NFT

IPFS використовує унікальний підхід до зберігання файлів, що базується на хеш-адресах. Кожен файл отримує свій унікальний ідентифікатор, який залежить від його змісту. Це означає, що дві однакові копії файлу матимуть однаковий ідентифікатор, тому що хеш обчислюється на основі змісту файлу, а не на його назві або розташуванні. Такий підхід забезпечує унікальність та цілісність даних.

Це має велике значення для зберігання NFT, оскільки ці токени часто містять унікальний контент, який має бути безперечно пов'язаний з власником NFT. IPFS дозволяє зберігати цей контент безпосередньо в мережі, забезпечуючи легкий доступ до нього та його цілісність. [23]

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

2.7.2 Механізми розподіленого зберігання файлів у IPFS

IPFS використовує розподілений підхід до зберігання файлів. Коли користувач додає файл до IPFS, він розбивається на блоки, і кожен блок отримує свій унікальний хеш-ідентифікатор. Ці блоки розподіляються по вузлах (пірингу) мережі IPFS, які забезпечують зберігання та передачу даних.

Кожен вузол у мережі IPFS має своє власне локальне сховище, в якому зберігаються блоки, що отримані ним або запитані з інших вузлів. Коли користувач потребує доступу до файлу, він використовує хеш-ідентифікатор для пошуку та отримання відповідного блоку з будь-якого вузла, який має цей блок збереженим.

Розподілена природа IPFS дозволяє зменшити навантаження на окремі вузли та підвищити доступність файлів. Якщо який-небудь вузол виходить з мережі або видаляє блоки, інші вузли все ще зберігають копії цих блоків, що гарантує надійність і цілісність даних. [23]

2.7.3 Інтеграція IPFS з NFT маркетплейсом

IPFS може бути ефективно інтегровано з NFT маркетплейсами для зберігання вмісту NFT. Замість того, щоб безпосередньо зберігати вміст на централізованому сервері маркетплейсу, вміст NFT можна зберігати у мережі IPFS. Це забезпечує децентралізований доступ до контенту та надійне зберігання даних.

При інтеграції IPFS з NFT маркетплейсом, метадані NFT, такі як назва, опис, власник та інші атрибути, зберігаються на блокчейні, як це зазвичай робиться. Однак, замість безпосереднього зберігання вмісту NFT, на блокчейні зберігається посилання (хеш) на вміст, який знаходиться на IPFS.

Така інтеграція дозволяє знизити вартість зберігання великих обсягів даних на блокчейні, оскільки весь вміст може бути збережений в мережі IPFS, яка пропонує більш ефективні механізми зберігання файлів. Крім того, IPFS

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

забезпечує швидкий та надійний доступ до вмісту NFT, що поліпшує загальний досвід користувачів. [24]

Як видно на рисунку 2.4, NFT маркетплейс використовує блокчейн для зберігання метаданих NFT та посилань на IPFS. Користувачі можуть переглядати та торгувати NFT на маркетплейсі, а при необхідності отримати доступ до вмісту NFT, вони використовують хеш-посилання для отримання відповідного файлу з мережі IPFS.

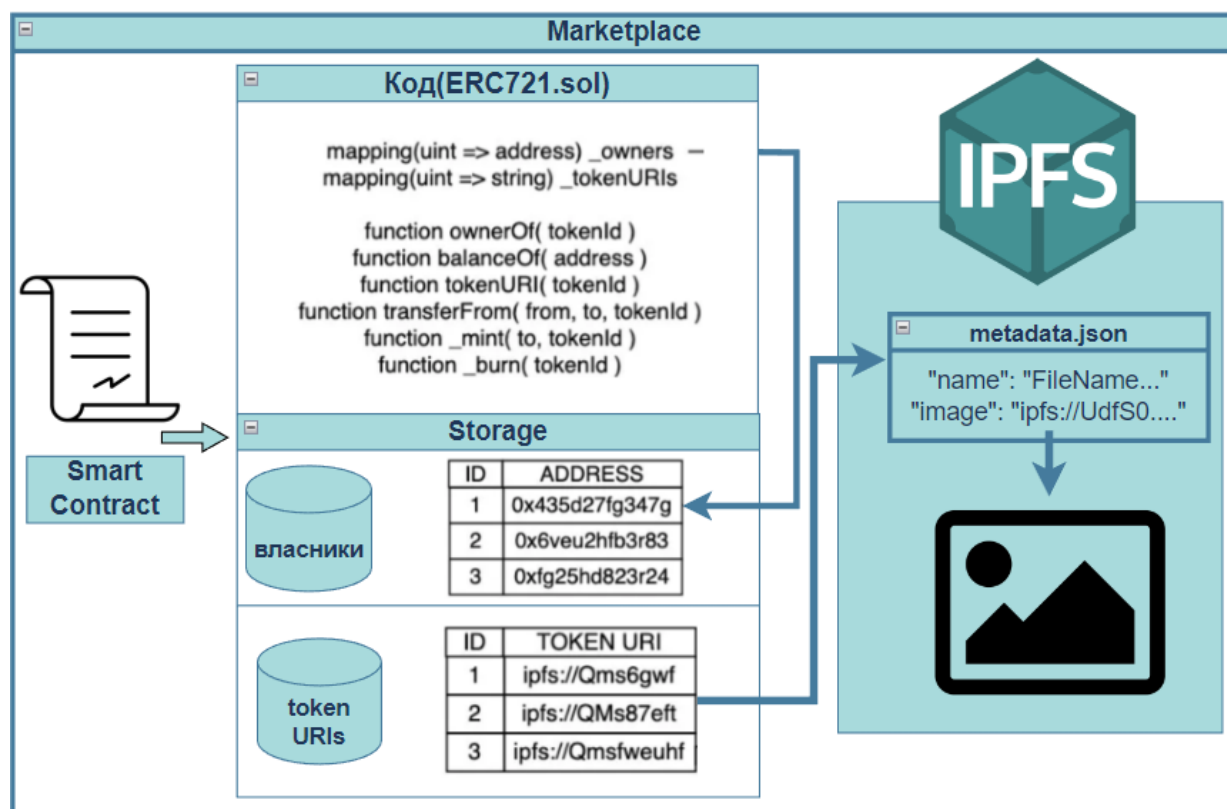


Рисунок 2.4 – Відображення взаємодії IPFS з NFT маркетплейсом

Така інтеграція дозволяє маркетплейсу забезпечити безпеку та надійність зберігання вмісту NFT, одночасно знижуючи вартість та покращуючи швидкість доступу до цього вмісту для користувачів.

ВИСНОВОК ДО РОЗДІЛУ 2

У даному розділі були описані ключові компоненти та технології, необхідні для створення NFT маркетплейсу.

Спочатку було розглянуто технологію блокчейн, яка лежить в основі будь-якого маркетплейсу NFT. Були досліджені основи, структура, принципи функціонування та різні види блокчейнів, зокрема Ethereum, який є основою для створення та обігу більшості невзаємозамінних токенів.

Технологія смарт-контрактів була ретельно вивчена, оскільки вона дозволяє створювати цифрові контракти, що автоматично виконують умови угоди без потреби в посередниках. Були розглянуті особливості, принцип роботи та застосування смарт-контрактів в NFT.

Для розробки смарт-контрактів було проведено дослідження мови програмування Solidity, що включає в себе основи, синтаксис та особливості цієї мови. Зокрема, було висвітлено використання Hardhat, як інструменту, що сприяє розробці, тестуванню та випуску Ethereum контрактів, його основних характеристик і можливостей.

Стандарт ERC-721 був визначений як ключовий елемент у створенні унікальних незамінних токенів на платформі Ethereum, оскільки визначає набір правил, які повинен виконувати кожен токен, та відкриває можливості для інтеракції між різними токенами на одній платформі. Це забезпечує, що кожен з токенів має унікальні властивості, а також впевненість у тому, що вони можуть бути безпечно передані або продані на будь-якому маркетплейсі, який підтримує цей стандарт.

Водночас, було визначено важливу роль OpenZeppelin як відкритої бібліотеки, що надає готові шаблони перевірені для смарт-контрактів, включаючи ті, що відповідають стандарту ERC-721. Наслідування шаблонів бібліотеки дозволяє зменшити кількість потенційних помилок і ризиків при розробці та зосередити увагу на унікальній логіці маркетплейсу та NFT.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Особливу увагу було приділено технології IPFS (InterPlanetary File System) через її роль у зберіганні NFT. Досліджено механізми розподіленого зберігання файлів у IPFS та можливості інтеграції IPFS з NFT маркетплейсом.

Вибір технологій був здійснений з акцентом на ефективність, безпеку, та можливість масштабування маркетплейсу. У цілому, розділ дає повну картину технологічного стеку, що використовується для розробки NFT маркетплейсу, та покладає основу для подальших досліджень та розробки.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

РОЗДІЛ 3

ДЕТАЛІ РОЗРОБКИ СИСТЕМИ

3.1 Розробка смарт-контракту

Враховуючи аналізовані матеріали та обрану стратегію реалізації, можна переходити до етапу розробки програмного продукту. Для створення ефективного NFT маркетплейсу, здатного конкурувати на сучасному ринку, необхідно детально описати основні компоненти системи, функціональні блоки програми та схему їх взаємодії.

Розробка смарт-контракту - це один із ключових елементів цієї системи, який вимагає особливої уваги. Смарт-контракти відіграють важливу роль в екосистемі блокчейна, оскільки вони дозволяють автоматизувати та безпечно виконувати транзакції без посередників. Вони стали невід'ємною частиною більшості криптовалютних платформ і маркетплейсів NFT.

Однак, розробка надійного та безпечного смарт-контракту є непростим завданням, особливо з урахуванням того, що всі транзакції в блокчейні є незворотними. Будь-яка помилка в коді може призвести до значних втрат або навіть до повного обмеження доступу до активів. Тому на цьому етапі ми зосередимося на деталях розробки смарт-контракту та його тестуванні.

3.1.1 Вимоги до розробки смарт-контракту

Основне завдання розробки смарт-контракту полягає в тому, щоб надати користувачам можливість безпечно та прозоро створювати, продавати та купувати NFT.

Вимоги до розробки смарт-контракту включають:

1. **Безпека:** Всі транзакції та дії в маркетплейсі повинні бути безпечними. Будь-яка помилка в коді може призвести до втрати активів або зловживання.

					ІАЛЦ.467200.003 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

2. Функціональність: Смарт-контракт повинен надавати повний набір функцій, необхідних для ефективного використання маркетплейсу, включаючи створення, виставлення на продаж, покупку та скасування NFT.
3. Прозорість: Усі дії, що відбуваються через смарт-контракт, повинні бути прозорими та публічно доступними для перевірки всіма користувачами системи.

Завдяки використанню Solidity та стандарту ERC721, смарт-контракт дозволяє впроваджувати вимоги до безпеки, прозорості та функціональності в рамках NFT маркетплейсу. Впровадження такого рішення допомагає впевнено керувати процесами створення, купівлі, продажу та обігу NFT в межах маркетплейсу.

До того ж, NFT, створені за допомогою розроблюваного смарт-контракту, зможуть використовуватися на будь-якому маркетплейсі, гаманці або іншому сервісі, що підтримує стандарт ERC721. Це надає користувачам гнучкість в обранні, де вони хочуть використовувати свої NFT, і збільшує вартість та привабливість NFT, створених на маркетплейсі.

3.1.2 Використання модулів OpenZeppelin

З огляду на потреби розроблюваного смарт-контракту, було обрано наступний перелік модулів, що надає бібліотека OpenZeppelin:

1. ERC721URIStorage: забезпечує базову реалізацію стандарту ERC721 для невзаємозамінних токенів (NFT). Він також містить додаткову логіку для зберігання URI метаданих для кожного токена, що дозволяє пов'язувати кожен NFT з унікальними метаданими.
2. Counters: надає простий числовий лічильник, який можна використовувати для нумерації ID токенів або для інших цілей, коли потрібно слідкувати за кількістю чогось.

					ІАЛЦ.467200.003 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

3. SafeMath: містить безпечні математичні операції, які автоматично перевіряють на переповнення та недоповнення перед виконанням операцій. Це важливо для уникнення помилок, які можуть призвести до несподіваної поведінки смарт-контрактів.

4. Ownable: додає механізм власності до вашого контракту, що дозволяє обмежити виконання певних функцій тільки власником контракту. Це може бути корисно для адміністративних завдань, таких як зміна параметрів контракту або управління резервними фондами.

Кожен з наведених модулів вносить важливі можливості та безпеку до розроблюваного проекту.

3.1.3 Структура даних NFTListing

Перед написанням безпосередньої логіки смарт-контракту, було визначено структуру даних NFTListing (рис. 3.1). Вона відіграє важливу роль у подальшому розробленні смарт-контракту.

```
struct NFTListing {  
    uint256 price;  
    address seller;  
}
```

Рисунок 3.1 – Структура даних NFTListing

Дана структура була створена для організації і зберігання інформації про продажі NFT в контексті смарт-контракту. До того ж, вона забезпечує зручність у керуванні даними і поліпшує зрозумілість коду.

Структура містить два поля: "price" та "seller". Поле "price" вказує на вартість продажу NFT, виміряну в wei (один ether дорівнює один квінтільйон

wei). У системі Ethereum, wei є найменшою монетною одиницею ефіра. Це можна порівняти з центом у доларовій валюті.

Ціна відображається у wei з двох основних причин:

1. Безпека типів даних: Solidity не підтримує дійсні числа. Це означає, що безпосередньо використовувати дробові значення в своєму коді не вийде. Для того, щоб представити дробове значення, потрібно зберігати його як ціле число, використовуючи найменшу можливу одиницю. У випадку Ethereum, це wei.
2. Уникнення помилок округлення: При використанні найменших одиниць уникнуто проблем з округленням, які можуть виникнути при роботі з дробовими значеннями. Це особливо важливо в контексті фінансових операцій, де точність є критичною.

Таким чином, представлення цін в wei дозволяє зберегти точність і безпеку при виконанні фінансових операцій в Ethereum. Ціна встановлюється продавцем при виставленні NFT на продаж.

Поле “seller” зберігає адресу Ethereum продавця, який виставив NFT на продаж. Оскільки після виставлення NFT на продаж, його фактичним власником стає сам смарт-контракт, записується адреса продавця, що дозволяє ідентифікувати його при подальших транзакціях. Саме ця адреса використовується при покупці NFT, щоб перерахувати кошти на баланс продавця. Дане поле дозволяє відслідковувати цю важливу інформацію. Це необхідно для підтримки відповідного функціонування механізму купівлі-продажу, оскільки без зберігання цієї інформації кошти з продажу не могли б бути коректно перераховані.

Крім того, ця інформація допомагає підтримувати прозорість ринку NFT, оскільки дозволяє будь-якому учаснику мережі визначити, хто виставив певний NFT на продаж. Це може бути корисно, наприклад, для встановлення автентичності NFT або для виявлення паттернів поведінки певних продавців.

					ІАЛЦ.467200.003 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

Використання структур даних, як NFTListing, є важливим аспектом розробки смарт-контрактів. Вони дозволяють організувати пов'язані значення в єдиний логічний блок, що спрощує читання, написання та управління кодом контракту.

В даному випадку, використання NFTListing допомагає групувати інформацію, яка стосується продажу NFT, в одному місці, замість того, щоб зберігати ці значення в окремих змінних. Це особливо корисно при виклику функцій, що потребують декількох пов'язаних параметрів, оскільки ви можете просто передати один NFTListing, замість передачі кожного параметра окремо.

3.1.4 Конструктор ERC721

Конструктор - це спеціальна функція, яка викликається при створенні екземпляра контракту. В даному контракті використовується конструктор ERC721, який був опублікований OpenZeppelin.

```
constructor(string memory name_, string memory symbol_) {  
    _name = name_;  
    _symbol = symbol_;  
}
```

Рисунок 3.2 – Конструктор ERC721 в бібліотеці OpenZeppelin

Даний конструктор приймає два параметри:

1. name - це повне ім'я проекту або платформи, яке використовується для визначення його в системах та ринках. Воно може бути вільним текстом, але повинно бути унікальним в мережі Ethereum. В даному випадку параметр name визначається як "MintMarket" при виклику конструктора, це послугує ім'ям для маркетплейсу.
2. symbol - символ токenu. Це короткий, унікальний ідентифікатор токenu в межах мережі Ethereum. Зазвичай, він складається з 3-5 літер і

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

представляє собою аббревіатуру імені токєну. В даному випадку параметр `symbol` визначається як "MINFT", що послує ім'ям для токєнів, створених на маркетплейсі.

Виклик конструктора базового контракту ERC721 з аргументами "MintMarket" та "MINFT" (рис. 3.3) - це ініціалізація базового контракту. Точніше, це називається "ініціалізацією базового контракту через конструктор підконтракту".

```
constructor() ERC721("MintMarket", "MINFT") {}
```

Рисунок 3.3 – Виклик конструктора ERC721 в кодї смарт-контракту

Таким чином, відбувається налаштування параметрів базового контракту ERC721 для розроблюваного підконтракту з відповідним ім'ям маркетплейсу і символом для NFT. Це важливо для розпізнавання та унікальності NFT на ринку. Коли люди будуть шукають NFT на вторинному ринку або в гаманці, вони зможуть впізнати його за цим ім'ям та символом.

3.1.5 Подія NFTTransfer

Наступним кроком, в тілі контракту було визначено подію NFTTransfer, яка використовується для публічного відстеження транзакцій, пов'язаних з NFT. У контексті цього контракту, ця подія може бути викликана під час створення нового NFT, виставлення токєну на продаж, його купівлі, або скасування продажу.

```
event NFTTransfer(  
    uint256 tokenID,  
    address from,  
    address to,  
    string tokenURI,  
    uint256 price  
);
```

Рисунок 3.4 – Оголошення події NFTTransfer

Дана подія приймає такі параметри:

- tokenID: це унікальний ідентифікатор NFT.
- from: це адреса, з якої токен був переданий. Якщо NFT щойно був створений, ця адреса буде адресою нулів, оскільки NFT ще не має попереднього власника.
- to: це адреса, на яку був переданий токен. Якщо NFT щойно був створений, це буде адреса, яка викликала функцію createNFT.
- tokenURI: це URI метаданих для NFT, що буде зберігатися на IPFS.
- price: це ціна NFT. Вона буде нульовою, якщо NFT ще не виставлено на продаж, або якщо він був перенесений без продажу (наприклад, при скасуванні продажу).

Виклик події NFTTransfer визначає запис в логах Ethereum про виконання конкретної дії. Завдяки цьому додатки та сервіси можуть слідкувати за цими подіями і відреагувати на них відповідно, наприклад, оновити стан своїх внутрішніх даних або повідомити користувачів. Інформація про цю подію включається в транзакцію, яка містить виклик функції, що спричинила подію. Ця інформація потім записується в блокчейн разом з деталями транзакції.

3.1.6 Функція для створення NFT

Для створення NFT було реалізовано відповідну функцію `createNFT`. Параметр `tokenURI`, який приймає ця функція на вхід, має важливе значення, оскільки він є посиланням на метадані токена. Дане посилання може бути будь-де в Інтернеті, але у даному випадку метадані будуть зберігатися в IPFS. Ця URI містить інформацію про токен, таку як назва, опис, зображення. Така інформація робить токен унікальним і цінним, особливо коли вона відображає цифрове мистецтво або інший колекційний предмет. Якщо посилання міняється, метадані токена, які вона відображає, також змінюються, тому посилання важливо зберігати без змін.

На рисунку 3.5 представлено вигляд функції, звідси можна помітити, що перед ім'ям параметру `tokenURI` вживається ключове слово `calldata`. У Solidity воно використовується для позначення незмінних параметрів функцій, які передаються зовнішньою транзакцією або через виклик делегату.

```
function createNFT(string calldata tokenURI) public {
    _tokenIDs.increment();
    uint256 currentID = _tokenIDs.current();
    _safeMint(msg.sender, currentID);
    _setTokenURI(currentID, tokenURI);
    emit NFTTransfer(currentID, address(0), msg.sender, tokenURI, 0);
}
```

Рисунок 3.5 – Функція `createNFT`

Важливо зауважити, що `calldata` є незмінною та зовнішньою, що означає, що вона може використовуватися тільки зовнішніми функціями, і її не можна змінити в межах функції. Використання `calldata` в контексті функції означає, що

ці дані будуть зберігатися поза областю пам'яті стека, що робить цю конструкцію більш економічною з точки зору витрат газу.

Далі розглянемо операції, що відбуваються в тілі функції для створення нового токена:

1. Спочатку генерується унікальний ідентифікатор для нового NFT.
2. Викликається функція `_safeMint`, яка використовується для безпечного створення нового токена. Функція `safeMint` перевіряє, чи дозволено викликаючому (тобто адресі `msg.sender`) створювати нові токени. Якщо так, вона викликає внутрішню функцію `mint`, яка додає новий токен до внутрішнього об'єкту `balances` контракту, збільшуючи баланс викликаючого на одиницю. Вона також викликає подію `Transfer`, що повідомляє про те, що новий токен було успішно створено і передано викликаючому.
3. Далі, за допомогою функції `_setTokenURI` зберігається URI метаданих для нового NFT.
4. З метою збереження всіх важливих деталей транзакції викликається подія `NFTTransfer`, що також слугує публічним повідомленням про створення нового NFT.

Таким чином, за допомогою розглянутих функцій та процесів, вдалося забезпечити надійне та безпечне створення NFT на розроблюваному маркетплейсі.

3.1.7 Функція для виставлення NFT на продаж

Наступним кроком було реалізовано функцію `listNFT`, яка є ключовою для роботи розроблюваного NFT маркетплейсу, оскільки вона дозволяє користувачам виставляти їх NFT на продаж.

Основною відповідальністю функції є забезпечення надійного та безпечного процесу виставлення NFT на продаж, що включає перевірку вхідних

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

даних, перенесення власності над NFT в контракт маркетплейсу та збереження інформації про продаж.

```
function listNFT(uint256 tokenID, uint256 price) public {  
    require(price > 0, "NFTMarket: price must be greater than 0");  
    transferFrom(msg.sender, address(this), tokenID);  
    _listings[tokenID] = NFTListing(price, msg.sender);  
    emit NFTTransfer(tokenID, msg.sender, address(this), "", price);  
}
```

Рисунок 3.6 – Функція listNFT

У тілі функції відбуваються наступні дії:

1. Перевіряється, що ціна, за якою виставляється NFT, більша за нуль. Це важливе обмеження, оскільки дозволяє уникнути непорозумінь та проблем, пов'язаних з безкоштовною передачею NFT. Якщо ціна менша або дорівнює нулю, виконання функції буде зупинено з відповідним повідомленням про помилку.
2. Викликається метод `transferFrom`, який передає власність над NFT від поточного власника (викликача функції) до контракту маркетплейсу. Це забезпечує, що NFT перебуває у власності маркетплейсу до того моменту, поки він буде проданий. Однак цей виклик вимагає, щоб викликач функції попередньо надав контракту маркетплейсу дозвіл на передачу NFT від його імені, використовуючи функцію `approve` або `setApprovalForAll` стандарту ERC721.
3. Після успішного перенесення власності над NFT, створюється новий об'єкт `NFTListing` з вказаною ціною і адресою продавця. Цей об'єкт

					ІАЛЦ.467200.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

додається до внутрішнього сховища `_listings` за ключем `tokenID`, що дозволяє швидко отримати інформацію про продаж NFT за його ID.

4. На завершення, генерується подія `NFTTransfer` з вказанням ID NFT, адреси продавця, адреси контракту маркетплейсу, порожнім URI метаданих і ціною. Ця подія використовується для відстеження переміщення NFT в контракт маркетплейсу та виставлення його на продаж.

3.1.8 Функція для купівлі NFT

З метою надання користувачам можливості купувати виставлені на продаж NFT, було реалізовано відповідну функцію `buyNFT` (рис. 3.7).

```
function buyNFT(uint256 tokenID) public payable {
    NFTListing memory listing = _listings[tokenID];
    require(listing.price > 0, "NFTMarket: nft not listed for sale");
    require(msg.value == listing.price, "NFTMarket: incorrect price");
    ERC721(address(this)).transferFrom(address(this), msg.sender, tokenID);
    clearListing(tokenID);
    payable(listing.seller).transfer(listing.price.mul(95).div(100));
    emit NFTTransfer(tokenID, address(this), msg.sender, "", 0);
}
```

Рисунок 3.7 – Функція `buyNFT`

Спочатку, в функції визначається змінна `listing` типу `NFTListing` за допомогою ідентифікатора токена `tokenID`, що передається як аргумент функції. Ця змінна представляє собою інформацію про виставлення NFT на продаж. Далі, перевіряється, чи ціна виставлення більша за нуль, що означає, що NFT

					ІАЛЦ.467200.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

дійсно виставлено на продаж. У випадку, якщо ціна рівна нулю, виконання функції припиняється із повідомленням про помилку.

Наступним кроком є перевірка відповідності значення, що надійшло разом з транзакцією (`msg.value`), ціні виставлення. Якщо ці значення не збігаються, виконання функції також припиняється із повідомленням про помилку.

Після цього, викликається метод `transferFrom` контракту ERC721. Цей метод передає власність над NFT від контракту маркетплейсу до викликача функції. А сума, яку оплатив покупець, надходить на адресу контракту маркетплейсу. Тоді виконується внутрішня функція `clearListing`, яка очищує відомості про виставлення NFT з внутрішнього сховища `_listings`.

Далі має відбутися операція переказу коштів з адреси контракту маркетплейсу на адресу продавця. На даному етапі важливо звернути увагу на можливість додання комісії, яку маркетплейс буде отримувати з продажу кожного токена. Ця комісія є способом монетизації маркетплейсу. За допомогою такої комісії можна отримувати прибуток від проведення транзакцій на платформі.

Важливо відзначити, що розмір комісії може бути встановлений в залежності від бізнес-моделі маркетплейсу і може змінюватися в залежності від різних факторів. Наприклад, вони можуть змінюватися в залежності від ціни продажу або типу NFT.

В даному випадку було прийнято рішення залишати 5% від суми з продажу кожного NFT. Після цього, частка від ціни продажу (95%) переказується продавцю NFT. При цьому важливо відзначити, що адреса продавця перетворюється в тип `payable`, що дозволяє надсилати на неї Ether.

В кінцевому етапі відбувається виклик події `NFTTransfer`, що повідомляє про перехід власності над NFT від контракту маркетплейсу до викликача функції.

					ІАЛЦ.467200.003 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Всі ці операції забезпечують надійний та безпечний процес купівлі NFT на маркетплейсі. Вони допомагають контролювати правильність виконання угоди, враховуючи інтереси як продавця, так і покупця, і надають важливу інформацію для відстеження та аналізу транзакцій на маркетплейсі.

3.1.9 Функція для скасування продажу NFT

В контексті потреб користувачів, важливо мати можливість скасувати виставлення NFT на продаж. Для цього була створена функція `cancelListing` (рис. 3.8).

```
function cancelListing(uint256 tokenID) public {
    NFTListing memory listing = _listings[tokenID];
    require(listing.price > 0, "NFTMarket: nft not listed for sale");
    require(listing.seller == msg.sender, "NFTMarket: you're not the seller");
    ERC721(address(this)).transferFrom(address(this), msg.sender, tokenID);
    clearListing(tokenID);
    emit NFTTransfer(tokenID, address(this), msg.sender, "", 0);
}
```

Рисунок 3.8 – Функція `cancelListing`

Спочатку, як і в попередній функції, визначається змінна, що містить інформацію про виставлення NFT на продаж. Далі, функція перевіряє, чи ціна виставлення більша за нуль. Це означає, що NFT дійсно було виставлено на продаж. Якщо ціна рівна нулю, виконання функції зупиняється із повідомленням про помилку.

Наступним кроком є перевірка, чи користувач, що викликав функцію, є дійсним продавцем даного NFT. Якщо викликач функції не є продавцем, виконання функції припиняється з повідомленням про помилку.

Після успішної перевірки, викликається метод `transferFrom` контракту ERC721. Цей метод передає власність над NFT назад від контракту маркетплейсу до викликача функції, який є продавцем даного NFT. Тоді, як і у випадку продажу токєну, виконується внутрішня функція `clearListing`, яка очищує відомості про виставлення NFT з внутрішнього сховища `_listings`.

В кінцевому етапі, викликається подія `NFTTransfer`, що повідомляє про перехід власності над NFT від контракту маркетплейсу назад до продавця.

Всі ці кроки разом гарантують, що продавці можуть безпечно та ефективно скасовувати свої виставлення на продаж, забезпечуючи при цьому прозорість та слідування правилам маркетплейсу.

3.1.10 Функція для зняття коштів

Важливим елементом управління маркетплейсом є можливість зняття накопичених на балансі контракту коштів. Для цього була створена функція `withdrawFunds` (рис. 3.9). Ця функція має велике значення для ефективного управління маркетплейсом. Вона дозволяє забезпечити виведення комісійних від продажу NFT, які накопичуються на балансі контракту. Без цієї функції кошти, зібрані з комісій, залишалися б на балансі контракту без можливості використання їх для покриття операційних витрат маркетплейсу або для подальшої монетизації.

```
function withdrawFunds() public onlyOwner {
    uint256 balance = address(this).balance;
    require(balance > 0, "NFTMarket: balance is zero");
    payable(msg.sender).transfer(balance);
}
```

Рисунок 3.9 – Функція `cancelListing`

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Спочатку визначається змінна `balance`, яка представляє собою поточний баланс контракту. Далі виконується перевірка, чи баланс більший за нуль. Це необхідно, оскільки немає сенсу виконувати зняття, якщо на балансі контракту немає коштів. Якщо баланс рівний нулю, виконання функції зупиняється з повідомленням про помилку.

У випадку, коли баланс більший за нуль, викликається метод `transfer`, який переказує всі кошти з балансу контракту на адресу викликача функції. Важливо відзначити, що ця функція доступна тільки для власника контракту (ознака `onlyOwner`).

3.2 Тестування смарт-контракту за допомогою Hardhat

Тестування смарт-контрактів було виконано як важлива частина розробки децентралізованих застосунків. Помилки у програмуванні смарт-контрактів є суттєвими, оскільки їх виправлення може бути вкрай складним і дорогим процесом.

Особливо це стає важливим через незворотність транзакцій на блокчейні, де помилки можуть призвести до втрати значної суми коштів. З огляду на важливість безпеки смарт-контрактів як основи довіри до системи, було прийнято рішення провести тестування всіх функцій з метою зниження кількості помилок перед його розгортанням.

3.2.1 Підготовка тестового середовища

Оскільки для тестування кожної із функцій потрібно мати екземпляр розгорнутого контракту, було прийнято рішення окремо винести код, що відповідальний за його розгортання (рис 3.10)

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

```

describe( title: "NFTMarket", fn: () => {
  let nftMarket: Contract;
  let signers: SignerWithAddress[];

  before( fn: async () => {
    // Deploy the NFTMarket contract
    const NFTMarket = await ethers.getContractFactory("NFTMarket");
    nftMarket = await NFTMarket.deploy();
    await nftMarket.deployed();
    signers = await ethers.getSigners();
  });
});

```

Рисунок 3.10 – Тестовий сценарій для розгортання смарт-контракту

В початковій частині коду відбувається підготовка для тестування: контракт розгортається, а змінна "signers" отримує список підписантів, які використовуються для симуляції різних користувачів в мережі Ethereum.

3.2.2 Тестування функцій смарт-контракту

Наступним кроком можна переходити до безпосереднього тестування функцій смарт-контракту.

Спочатку, у процесі тестування, викликається функція "createNFT" з аргументом tokenURI, який містить посилання на токен. Далі, результат виконання цієї функції очікується у формі транзакції.

Наступним чином, в квитанції про транзакцію вибирається ідентифікатор токена, що використовується для подальшого тестування. Звертається увага на те, що URI щойно створеного токена відповідає посиланню на токен, яке було передано при виклику функції "createNFT". На рисунку 3.11 наведено код тестування функції для створення NFT.

```

describe( title: "createNFT", fn: () => {
  it( title: "should create an NFT with the correct owner and tokenURI", fn: async () => {

    const tokenURI = "https://some-token.uri/";
    const transaction = await nftMarket.createNFT(tokenURI);
    const receipt = await transaction.wait();
    const tokenId = receipt.events[0].args.tokenId;

    const mintedTokenURI = await nftMarket.tokenURI(tokenId);
    expect(mintedTokenURI).to.equal(tokenURI);

    const ownerAddress = await nftMarket.ownerOf(tokenId);
    const currentAddress = await signers[0].getAddress();
    expect(ownerAddress).to.equal(currentAddress);

    const args = receipt.events[1].args;
    expect(args.tokenId).to.equal(tokenId);
    expect(args.from).to.equal(ethersConstants.AddressZero);
    expect(args.to).to.equal(ownerAddress);
    expect(args.tokenURI).to.equal(tokenURI);
    expect(args.price).to.equal( value: 0);

  });
});

```

Рисунок 3.11 – Тестовий сценарій для функції createNFT

Далі, проводиться перевірка власника створеного токена та перевіряється, чи адреса власника відповідає адресі, з якої було ініційовано транзакцію. Також відбувається аналіз події NFTTransfer, що міститься у квитанції про транзакцію. Відповідні аргументи події перевіряються з метою забезпечення правильності виконання функції.

За допомогою такого комплексного тестування можна гарантувати, що функція "createNFT" працює правильно і створює NFT з відповідним власником і URI токена.

У такий спосіб відбувається тестування всіх інших функцій в коді смарт-контракту. Процедура, що описана вище, є типовим підходом до тестування

					ІАЛЦ.467200.003 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

функцій на базі Ethereum і використовується для перевірки коректності їхньої роботи.

3.3 Реалізація механізму авторизації через криптогаманець

Для авторизації користувачів було реалізовано механізм, що базується на підключенні Ethereum гаманця користувача через MetaMask, що реалізовано за допомогою бібліотеки Web3Modal. Це рішення дозволяє користувачам використовувати їх власний гаманець для авторизації в додатку, забезпечуючи високий рівень безпеки та прозорості.

```
const connectWallet = async () => {
  setLoading( value: true);
  try {
    const web3modal = new Web3Modal( opts: { cacheProvider: true })
    const instance = await web3modal.connect();
    const provider = new Web3Provider(instance);
    const signer = provider.getSigner();
    const address = await signer.getAddress();
    setSigner(signer);
    setAddress(address);
  } catch (e) {
    console.log(e);
  }
  setLoading( value: false);
};
```

Рисунок 3.12 – Вигляд функції для підключення криптогаманця

Спочатку, створюється новий екземпляр Web3Modal, що є вікном для вибору гаманця користувача. Це вікно дозволяє користувачу вибрати свій Ethereum гаманець для взаємодії з додатком.

Далі, викликається функція connect() на створеному екземплярі Web3Modal. Ця функція повертає об'єкт провайдера, що представляє

підключений гаманець користувача. У випадку успішного виконання, користувач буде підключений до додатку через свій гаманець.

Після цього, створюється новий екземпляр Web3Provider з використанням отриманого провайдера. Web3Provider - це оболонка, яка дозволяє використовувати функції Ethereum через вибраний гаманець користувача.

Наступним чином, викликається функція `getSigner()` на створеному екземплярі Web3Provider. Ця функція повертає об'єкт `signer`, що представляє гаманець користувача і може використовуватися для підпису транзакцій.

Далі, викликається функція `getAddress()` на отриманому об'єкті `signer`. Ця функція повертає адресу гаманця користувача, що підтверджує його авторизацію.

Таким способом, користувач може підключити свій гаманець для взаємодії з Ethereum через додаток. Передбачуваний результат цього процесу - отримання адреси гаманця користувача і його об'єкту `signer`, які можуть бути використані для подальших транзакцій і взаємодій з blockchain.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

ВИСНОВОК ДО РОЗДІЛУ 3

У ході роботи, було успішно розроблено платформу NFT маркетплейсу, використовуючи сучасні технології і стратегії, описані в попередньому розділі цього проекту. Створення NFT, інтеграція з Ethereum блокчейном і реалізація системи авторизації за допомогою криптогаманця MetaMask стали ключовими пріоритетами в ході розробки.

На відміну від традиційних централізованих рішень, розроблена платформа витримує принципи децентралізації, що забезпечує збільшену прозорість, безпеку та надійність. Крім того, використання криптогаманця для авторизації не тільки підвищує зручність користувачів, але і гарантує безпеку їх криптовалютних активів.

Не менш важливим аспектом було створення безпечного механізму виведення коштів, який надає доступ до функції лише для власника контракту. Цей механізм відіграє ключову роль в захисті активів користувача від недобросовісних дій.

Основний фокус при виборі компонентів був на тому, щоб забезпечити високу безпеку, прозорість, а також можливість масштабування системи в майбутньому.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

РОЗДІЛ 4

ДОСЛІДЖЕННЯ ТА АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ

Розробка дизайну інтерфейсу NFT маркетплейсу була ключовою частиною проекту, враховуючи його важливість як основного способу взаємодії користувачів з сервісом. З огляду на те, що інтерфейс є першим, на що звертають увагу користувачі, було приділено значний час та зусилля його розробці. Основним критерієм при цьому було виконання початкового принципу: "Інтерфейс має бути максимально простим та інтуїтивно зрозумілим". Цей принцип було визначено з урахуванням різних категорій користувачів, що будуть використовувати маркетплейс, включаючи тих, хто може не мати великого досвіду роботи з подібними платформами.

Було створено чітку, зрозумілу навігацію, яка дозволяє користувачам легко перемикатися між вкладками маркетплейсу.

Також було важливо врахувати досвід користувача в контексті різноманітних процесів, що відбуваються на платформі. Це стосується, наприклад, створення, купівлі та продажу NFT. Важливо, щоб кожна з цих дій була максимально простою та зрозумілою.

4.1 Огляд головної сторінки

На головну або, як ще її називають, домашню сторінку користувачі потрапляють автоматично при першому вході. Вона служить вітальним екраном, на якому представлена основна інформація про сервіс. На цій сторінці користувачі можуть ознайомитися з головними особливостями та функціональністю платформи. Головна сторінка має інтуїтивний дизайн і легкодоступну навігацію, що дозволяє користувачам легко знайти необхідні розділи маркетплейсу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

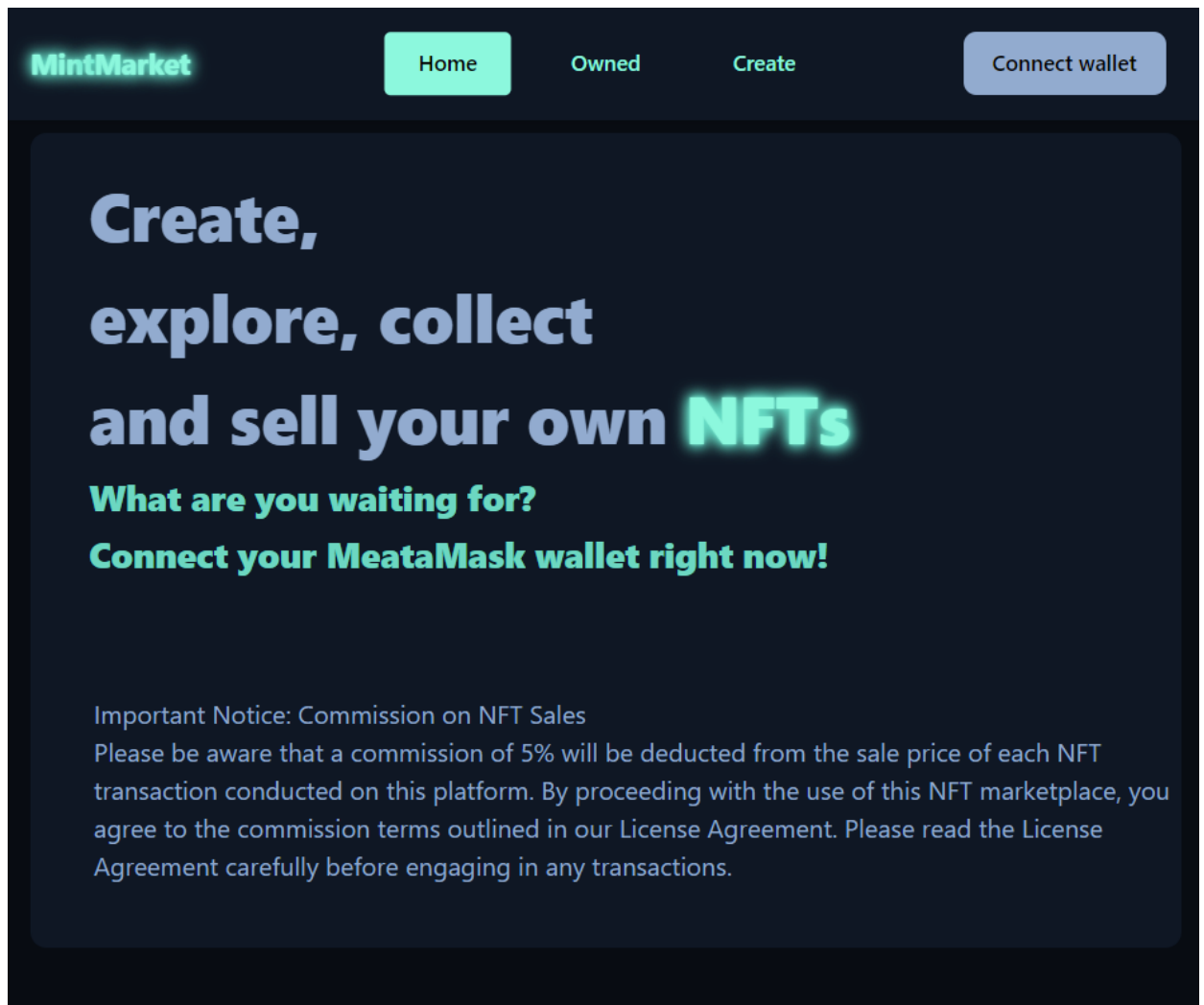


Рисунок 4.1 – Початкова сторінка платформи

В верхній частині екрана розташована навігаційна панель, яка служить основним інструментом для керування та переходу між різними секціями маркетплейсу. На панелі присутні три основні кнопки, що виконують роль засобів для ефективної та зручної навігації на сайті. Вони допомагають користувачам легко перемикатися між основними вкладками та функціями платформи.

Також, для зручності повернення до неї, було створено відповідну кнопку "Home" та розміщено логотип маркетплейсу в лівому кутку навігаційної панелі. Обидва елементи виконують однакову функцію - повернення на домашню сторінку.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

Саме так сторінка виглядає для неавторизованого користувача, який щойно потрапив на сайт. Центральну частину екрану займає текст, який надає короткий огляд можливостей сайту. Тут висвітлюється ряд основних дій, які можна виконати на платформі, включаючи створення, купівлю, продаж та ознайомлення з NFT.

Після цього користувачу пропонується авторизуватися за допомогою свого криптогаманця MetaMask. Цей процес забезпечує безпечний вхід на платформу та можливість взаємодії з усіма доступними функціями маркетплейсу.

У нижній частині сторінки міститься інформація про комісію, яку бере платформа. Для прозорості та зручності користувачів, вказано, що з продажу кожного NFT, платформа вираховує 5% комісії. Це важливий аспект, з яким користувачам варто ознайомитися перед тим, як починати використовувати сервіс.

4.2 Авторизація через криптогаманець MetaMask

На навігаційній панелі, окрім традиційних вкладок для переходу між різними секціями сайту, знаходиться ключова кнопка "Connect Wallet". Цей елемент виконує важливу роль у взаємодії користувача з платформою.

Дана кнопка є візуально виділеним елементом на панелі, що допомагає користувачам швидко визначити, як здійснити підключення до свого криптогаманця. При натисканні на неї відкривається вікно, яке направляє користувача через процес авторизації через криптогаманець MetaMask.

Цей процес є простим і зрозумілим, навіть для тих, хто вперше використовує NFT маркетплейс. Після успішного підключення гаманця, користувачу надається повний доступ до всіх функцій платформи, включаючи можливість переглядати, купувати, продавати та створювати NFT.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

Єдина вимога - у гаманці користувача повинно бути достатньо криптовалюти для здійснення транзакцій на платформі, а також для покриття витрат на газ, що є необхідним для проведення транзакцій на Ethereum блокчейні.

На рисунку 4.2 представлено спливаюче вікно MetaMask, яке з'являється після натискання на кнопку "Connect Wallet".

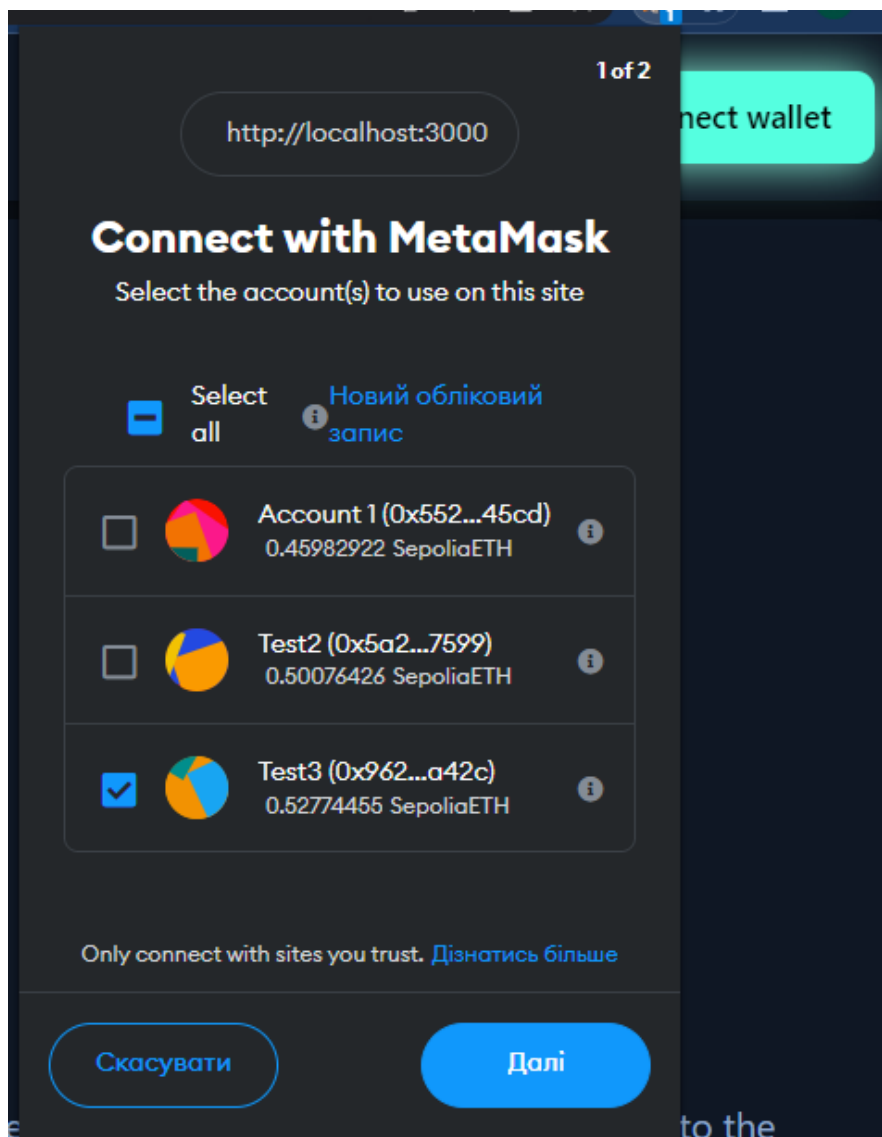


Рисунок 4.2 – Вікно підключення MetaMask

У вікні MetaMask користувачу пропонується вибрати потрібний гаманець для підключення та ввести свій пароль або інші необхідні дані для авторизації.

Процес здійснюється в безпечному режимі, гарантуючи конфіденційність інформації користувача.

Після успішної авторизації, сторінка маркетплейсу автоматично оновлюється. Вікно MetaMask, при цьому, може бути закрите, але з'єднання гаманця з платформою залишається активним до моменту виходу користувача з системи або якщо користувач вирішить вручну від'єднати свій гаманець.

Замість кнопки "Connect Wallet" на верхній навігаційній панелі тепер з'явився цифровий аватар користувача. Цей аватар - унікальне графічне зображення, що генерується на основі адреси гаманця користувача.

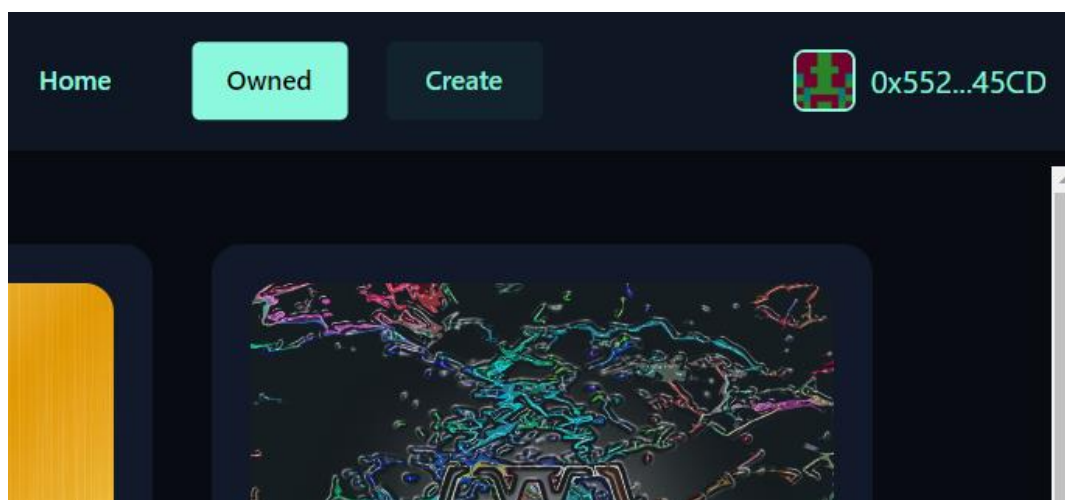


Рисунок 4.3 – Вигляд навігаційної панелі після авторизації

Цифровий аватар служить не лише елементом персоналізації, але й є практичним способом візуальної ідентифікації користувача. Він допомагає визначити, що користувач успішно з'єднався зі своїм гаманцем та є готовим до взаємодії з платформою. Поряд із цифровим аватаром відображається скорочена версія адреси гаманця користувача. Ця інформація допомагає користувачу переконатися, що він підключив вірний гаманець, та може бути використана для здійснення транзакцій на платформі. Для забезпечення безпеки, повна адреса гаманця не відображається, залишаючи достатню

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

кількість інформації для розпізнавання, але зменшуючи ризик небажаного використання адреси третіми особами.

4.3 Сторінка для створення NFT

За допомогою кнопки "Create", розташованої на верхній навігаційній панелі, користувачі мають можливість перейти на сторінку для створення власних NFT. Дана сторінка має інтуїтивно зрозумілу структуру. Зліва вона містить поле для вибору файлу, який користувач бажає перетворити на NFT. Підтримуються файли форматів JPEG, PNG та GIF. З правого боку розташовані поля для введення імені та опису токена.

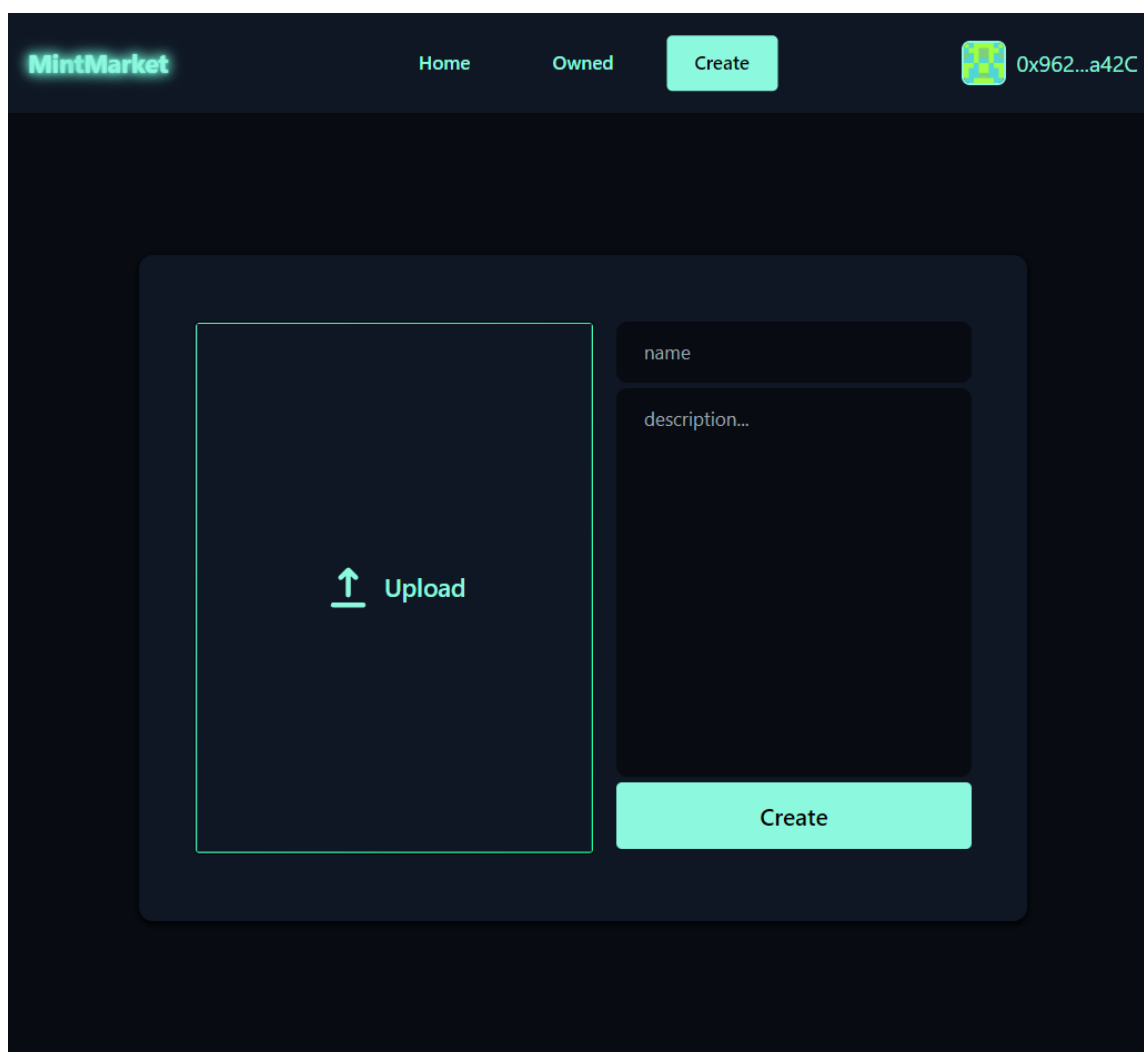


Рисунок 4.4 – Вигляд сторінки для створення NFT

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

Після вводу всіх даних користувач може запустити процес створення, натиснувши на відповідну кнопку у нижньому правому куті сторінки. Після цього, починається процес завантаження метаданих на IPFS.

Наступним кроком є створення NFT на блокчейні. Оскільки цей створення NFT, в дійсності, є транзакцією в мережі Ethereum, то для його підтвердження з'являється спливаюче вікно MetaMask (рисунок 4.5). Це вікно необхідне для того, щоб користувач міг відправити транзакцію в мережу, підтверджуючи свою дію та оплату витрат на газ для виконання цієї транзакції. До того ж комісія за транзакцію створення NFT може бути більшою ніж звичайні транзакції.

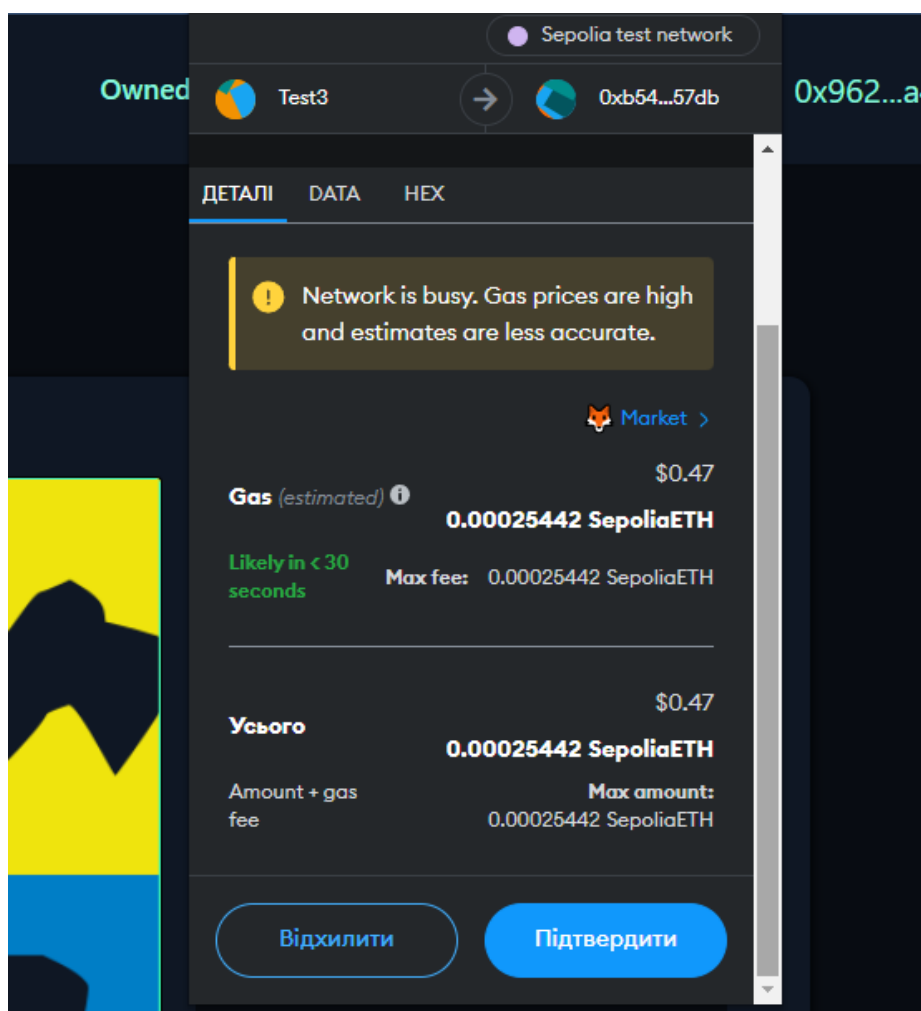


Рисунок 4.5 – Вікно MetaMask для підтвердження транзакції

При успішному завершенні всіх процедур, користувачу відображається спливаюче вікно з повідомленням про те, що він може очікувати появи свого нового NFT на сторінці "Owned". Оскільки процес створення може зайняти певний час, новий токен може з'явитися на цій сторінці не відразу. Користувачам рекомендується трохи зачекати перед тим, як він з'явиться в переліку.

4.4 Сторінка для перегляду та керування власними NFT

Щоб переглянути власні NFT на особистій сторінці, можна скористатися відповідною кнопкою "Owned" на верхній панелі. Тут токени відображаються у вигляді інтерактивних карток, кожна з яких вміщує зображення, назву та опис.

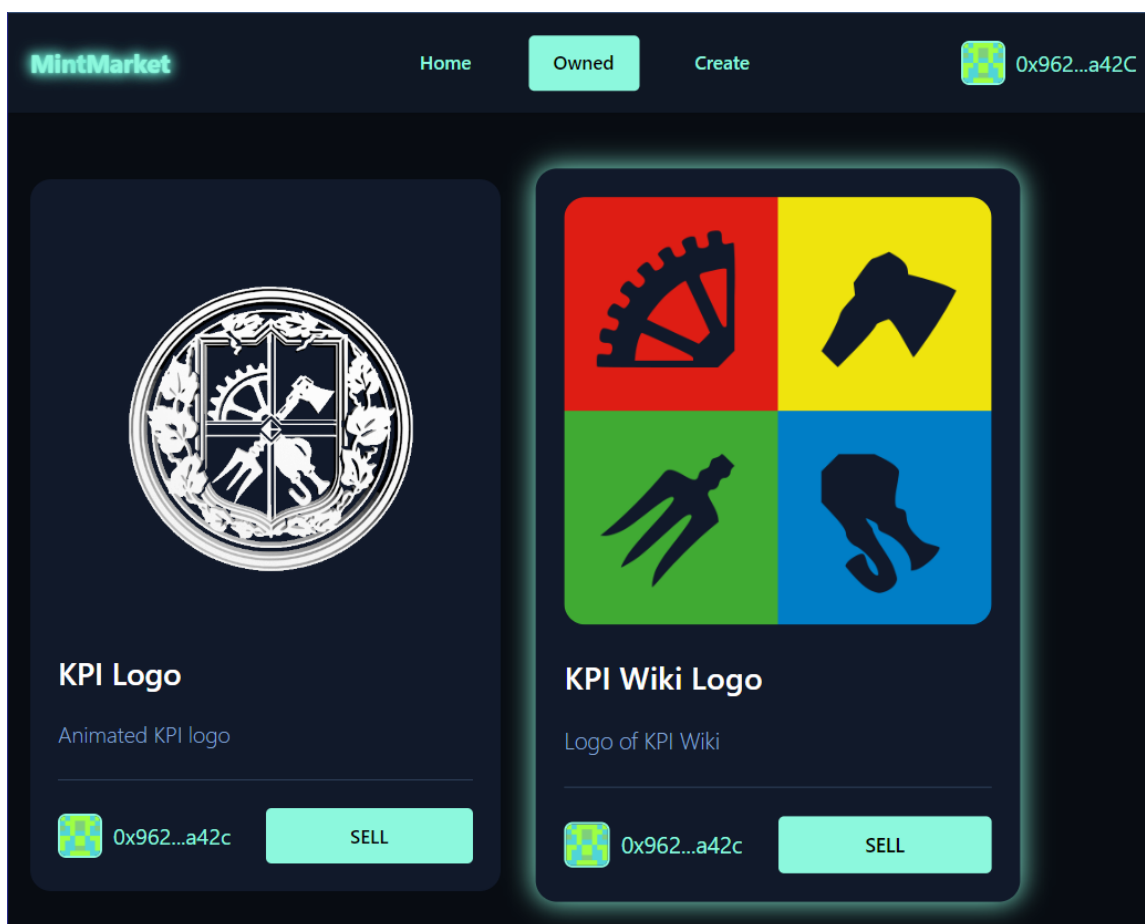


Рисунок 4.6 – Вигляд сторінки для перегляду та керуванням NFT

У випадку бажання виставити їх на продаж, користувачу необхідно лише обрати відповідний NFT і натиснути відповідну кнопку, розташовану в нижньому кутку картки. Саме ця дія активує спливаюче вікно, де можна встановити ціну в ефіріумі і підтвердити намір продажу. Зверху вікна розташований текст з нагадуванням про те, що користувач в будь-який час може скасувати продаж.

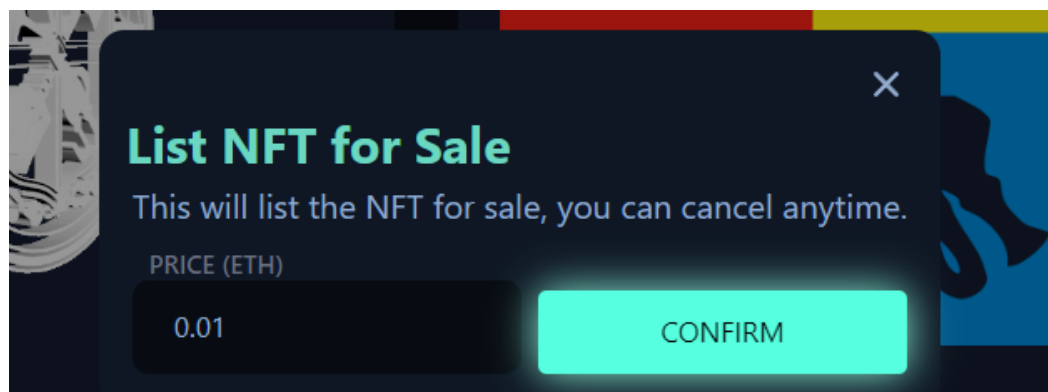


Рисунок 4.7 – Вигляд вікна для виставлення NFT на продаж

Але варто враховувати, що наявність достатніх коштів для оплати транзакції є важливою, оскільки право власності на NFT під час продажу переходить до контракту маркетплейсу, а при скасуванні навпаки. Тому важливо розуміти, що кожна така дія потребує витрат на газ.

Після підтвердження суми, за яку користувач бажає виставити свій NFT на продаж, з'являється спливаюче вікно MetaMask. В цьому вікні користувачу пропонується підтвердити транзакцію. У вікні, як зазвичай, вказані всі деталі транзакції, включаючи витрати на газ. Це дає користувачу повну інформацію про транзакцію та допомагає ухвалити обґрунтоване рішення перед тим, як її підтвердити.

По завершенню цього процесу, NFT буде відображено в нижній частині сторінки під заголовком "Listed". Втім, важливо зазначити, що це може не статися відразу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Реєстрація транзакції переходу власності на блокчейні може вимагати певного часу, залежно від швидкості мережі і рівня зайнятості. Основна відмінність виставлених на продаж NFT від інших полягає в кнопці, яка відображає ціну в Ethereum та дає можливість скасувати продаж.

4.5 Процес купівлі NFT

Розглянемо процес купівлі NFT. Для початку, виконаємо авторизацію за допомогою іншого облікового запису криптогаманця.

Для авторизованого користувача основний зміст головної сторінки змінюється. Замість початкового тексту, що описує основні можливості платформи, тут відображаються NFT, виставлені на продаж іншими користувачами платформи, а саме можемо побачити токен, який ми щойно виставили на продаж з іншого акаунту .

Однак, варто зауважити, що користувачу відображаються лише ті NFT, які він сам може придбати. Це означає, що всі NFT, виставлені на продаж самим користувачем, не відображаються в цьому переліку. Такий підхід забезпечує більшу зручність навігації та дозволяє користувачам зосередитися на виборі та купівлі NFT, замість перегляду власних лотів.

Щоб придбати NFT, користувачу потрібно натиснути на відповідну кнопку з вказаною вартістю, після чого відкривається спливаюче вікно MetaMask з детальною інформацією про транзакцію. Користувачу надається можливість підтвердити її, натиснувши відповідну кнопку. При цьому, обов'язковою є наявність достатнього балансу ефіріума на гаманці для покриття вартості NFT та комісії за транзакцію.

Наступним кроком відбувається процес передачі власності NFT від продавця до покупця на блокчейні Ethereum. Придбаний NFT тепер буде відображатися на сторінці "Owned" нового власника.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

Аби переконатися, що продавцю нараховано 95% від суми продажу, ми проведемо перевірку балансу його акаунту криптогаманця. На рисунку 4.8 відображено поточний баланс в ефіріумі відповідного акаунту до та після купівлі NFT з іншого акаунту.

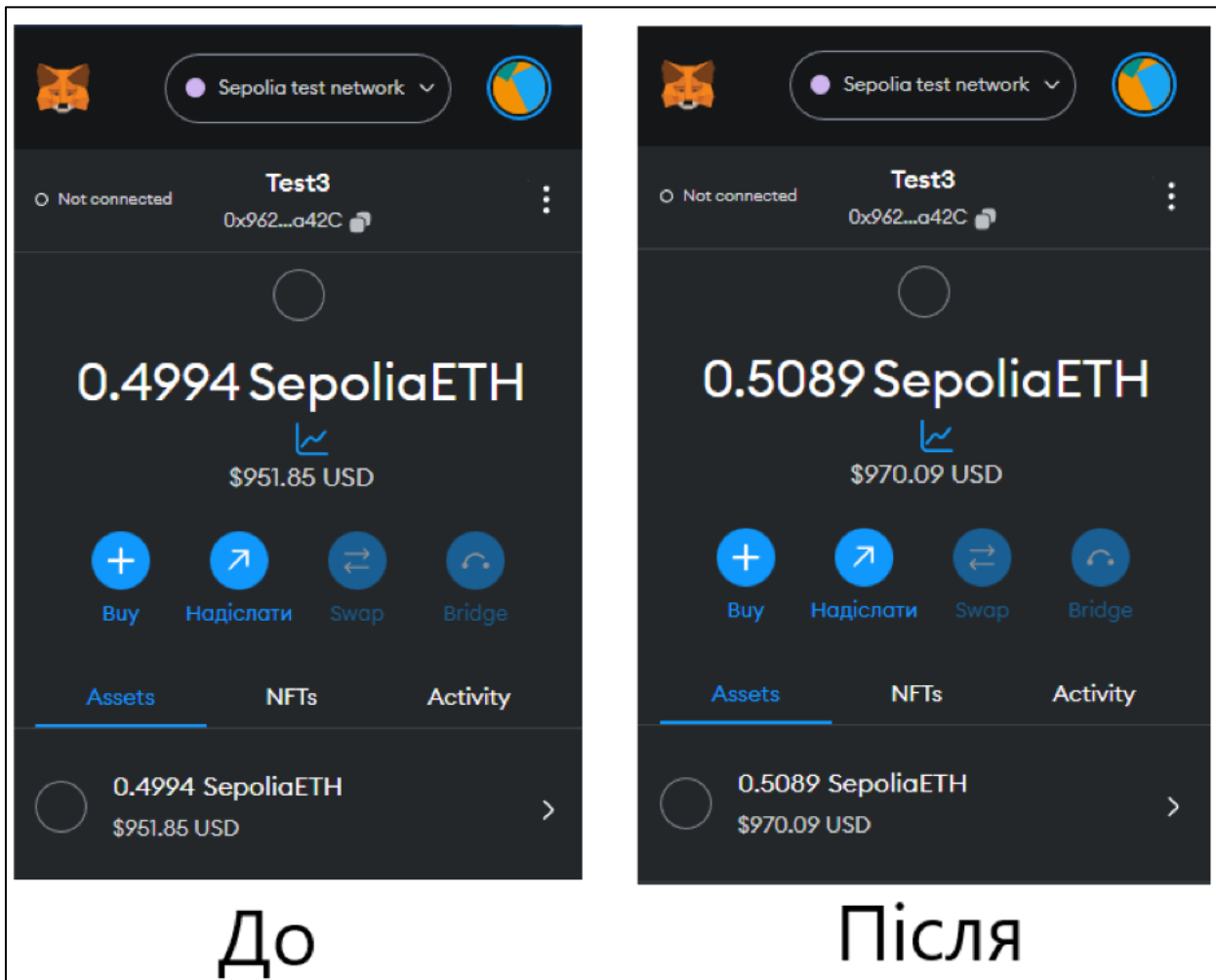


Рисунок 4.8 – Баланс акаунту продавця до та після продажу NFT

Різниця балансів становить 0.0095ETH, що дійсно відображає 95% від ціни продажу NFT (0.01 ETH). Це дає нам впевненість, що система коректно нараховує кошти відповідно до зазначених умов маркетплейсу

4.6 Рекомендації щодо розвитку та вдосконалення NFT маркетплейсу

Перш за все, можна додати функціонал пошуку за конкретними категоріями, тегами, або художниками. Це допоможе користувачам більш зручно та ефективно відшукати NFT, які їх найбільше цікавлять.

Важливим кроком у розвитку може стати розробка функціоналу для аукціонів. Це додатковий спосіб для продажу NFT, який може бути цікавим для багатьох користувачів.

І наостанок, додатковим напрямком для розвитку може стати розширення підтримки різних форматів файлів при створенні NFT. Наразі маркетплейс підтримує лише формати JPEG, PNG та GIF, але з урахуванням різноманітності контенту, який користувачі можуть бажати перетворити на NFT, цього може бути недостатньо.

Опція підтримки більшого спектра файлів, включаючи, наприклад, 3D моделі, різні формати аудіо та відео, анімацію, може значно розширити можливості для користувачів і збільшити привабливість маркетплейсу.

Впровадження цих змін вимагає додаткової розробки і тестування, але потенціальна вигода може бути значною.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

ВИСНОВОК ДО РОЗДІЛУ 4

У даному розділі було детально розглянуто інтерфейс та протестовано функціонал NFT маркетплейсу, при цьому основний акцент було зроблено на процеси авторизації, створення NFT, їх продажу та купівлю. В ході аналізу була проведена перевірка коректності роботи всіх основних функцій. Особливу увагу було приділено перевірці правильності нарахування коштів від транзакцій, що є критично важливим аспектом для будь-якого фінансового додатку.

При перевірці купівлі та продажу NFT було встановлено, що вартість ціни коректно відображається та нараховується на відповідні криптовалютні рахунки користувачів. Це свідчить про надійність та ефективність системи, а також про її здатність забезпечувати безпечні та прозорі транзакції для користувачів.

Наостанок було розглянуто потенційні шляхи покращення системи. Зокрема, було обговорено можливість покращення процесу створення, пошуку та купівлі NFT.

					ІАЛЦ.467200.003 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході роботи над даним проектом було створено і впроваджено NFT маркетплейс, що дозволяє користувачам безпечно створювати, купувати, продавати та переглядати NFT. Система була ретельно перевірена з метою забезпечення її надійності та ефективності, включаючи перевірку коректності нарахування коштів.

Незважаючи на високу якість реалізації, було визначено декілька напрямків для подальшого розвитку та вдосконалення системи. Це включає розширення підтримки типів файлів, а також додання функціоналу пошуку.

Результати цього проекту демонструють значну вдачу та ефективність розробленого NFT маркетплейсу з технічної і економічної точок зору. З технічного боку, маркетплейс успішно виконує свої основні функції, включаючи авторизацію, створення, продаж та перегляд NFT, забезпечуючи коректне нарахування коштів. Це свідчить про високу якість розробки, стабільність і надійність системи.

З економічного боку, маркетплейс має великий потенціал для заробітку, оскільки він вимагає від продавців невеликого відсотка з продажу як комісії, що створює цінність для власників платформи. Крім того, через зростаючу популярність NFT, попит на такі платформи продовжує зростати, що веде до збільшення економічного потенціалу проекту.

Нове технічне рішення, що було впроваджено в цьому проекті, полягає у використанні блокчейн технології для забезпечення безпечності та прозорості транзакцій. Перевага такого підходу полягає у високій ступені захисту та довіри, яку він надає користувачам.

Розроблений NFT маркетплейс - це не просто платформа для торгівлі цифровими активами, але й інструмент, який сприяє демократизації мистецтва та цифрового контенту, даючи можливість кожному створити, купити або продати NFT в простий і безпечний спосіб.

					ІАЛЦ.467200.003 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Non-Fungible Token (NFT): What It Means and How It Works [Електронний ресурс] – Режим доступу до ресурсу: <https://www.investopedia.com/non-fungible-tokens-nft-5115211>
2. The Advantages Of Non-Fungible Tokens (NFTs) [Електронний ресурс] – Режим доступу до ресурсу: <https://101blockchains.com/advantages-of-nfts/>
3. What Is OpenSea? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.investopedia.com/what-is-opensea-6362477>
4. What is Nifty Gateway and How does it Work? [Електронний ресурс] – Режим доступу до ресурсу: <https://phemex.com/academy/what-is-nifty-gateway-and-how-does-it-work>
5. What is SuperRare? [Електронний ресурс] – Режим доступу до ресурсу: <https://decrypt.co/resources/what-is-superrare-3-minute-guide-explained-art-collectible>
6. SuperRare: An Exclusive, Curated NFT Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gemini.com/cryptopedia/superrare-nft-marketplace-superrare-crypto-art-market>
7. Rarible (RARI): An NFT Platform for Digital Works [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gemini.com/cryptopedia/rarible-nft-art-marketplace-rari-coin>
8. Blockchain Merkle Trees [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/blockchain-merkle-trees/>
9. What Are Consensus Mechanisms in Blockchain and Cryptocurrency? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp>

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

10. How are blocks added to a blockchain? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.coindesk.com/learn/how-blocks-are-added-to-a-blockchain-explained-simply/>
11. What are the different types of blockchain technology? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology>
12. What is Ethereum and how does it work? [Электронный ресурс] – Режим доступа до ресурсу: <https://cointelegraph.com/learn/what-is-ethereum-a-beginners-guide-to-eth-cryptocurrency>
13. What are smart contracts on blockchain? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.ibm.com/topics/smart-contracts>
14. What Is an NFT Smart Contract? [Электронный ресурс] – Режим доступа до ресурсу: <https://cyberscrilla.com/nft-smart-contracts-explained/#:~:text=An%20NFT%20smart%20contract%20is,an%20intermediary%20or%20central%20authority.>
15. Solidity — Solidity 0.8.20 documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.soliditylang.org/en/v0.8.20/>
16. What is Solidity Programming: Data Types, Smart Contracts, and EVM? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-solidity-programming>
17. ERC-721 Non-Fungible Token Standard [Электронный ресурс] – Режим доступа до ресурсу: [https://ethereum.org/en/developers/docs/standards/tokens/erc-721/#:~:text=The%20ERC%2D721%20\(Ethereum%20Request,for%20tokens%20within%20Smart%20Contracts.](https://ethereum.org/en/developers/docs/standards/tokens/erc-721/#:~:text=The%20ERC%2D721%20(Ethereum%20Request,for%20tokens%20within%20Smart%20Contracts.)

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

18. How to Develop an NFT Smart Contract (ERC721) [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.alchemy.com/docs/how-to-develop-an-nft-smart-contract-erc721-with-alchemy>
19. Hardhat Explained – What is Hardhat? [Електронний ресурс] – Режим доступу до ресурсу: <https://moralis.io/hardhat-explained-what-is-hardhat/>
20. What is OpenZeppelin? The Ultimate Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://moralis.io/what-is-openzeppelin-the-ultimate-guide/>
21. OpenZeppelin Docs: Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.openzeppelin.com/learn/developing-smart-contracts>
22. What is IPFS? [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.ipfs.tech/concepts/what-is-ipfs/>
23. What Is IPFS? [Електронний ресурс] – Режим доступу до ресурсу: <https://beincrypto.com/learn/what-is-ipfs/>
24. Mint an NFT with IPFS [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.ipfs.tech/how-to/mint-nfts-with-ipfs/#a-short-introduction-to-nfts>

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

ДОДАТОК 1

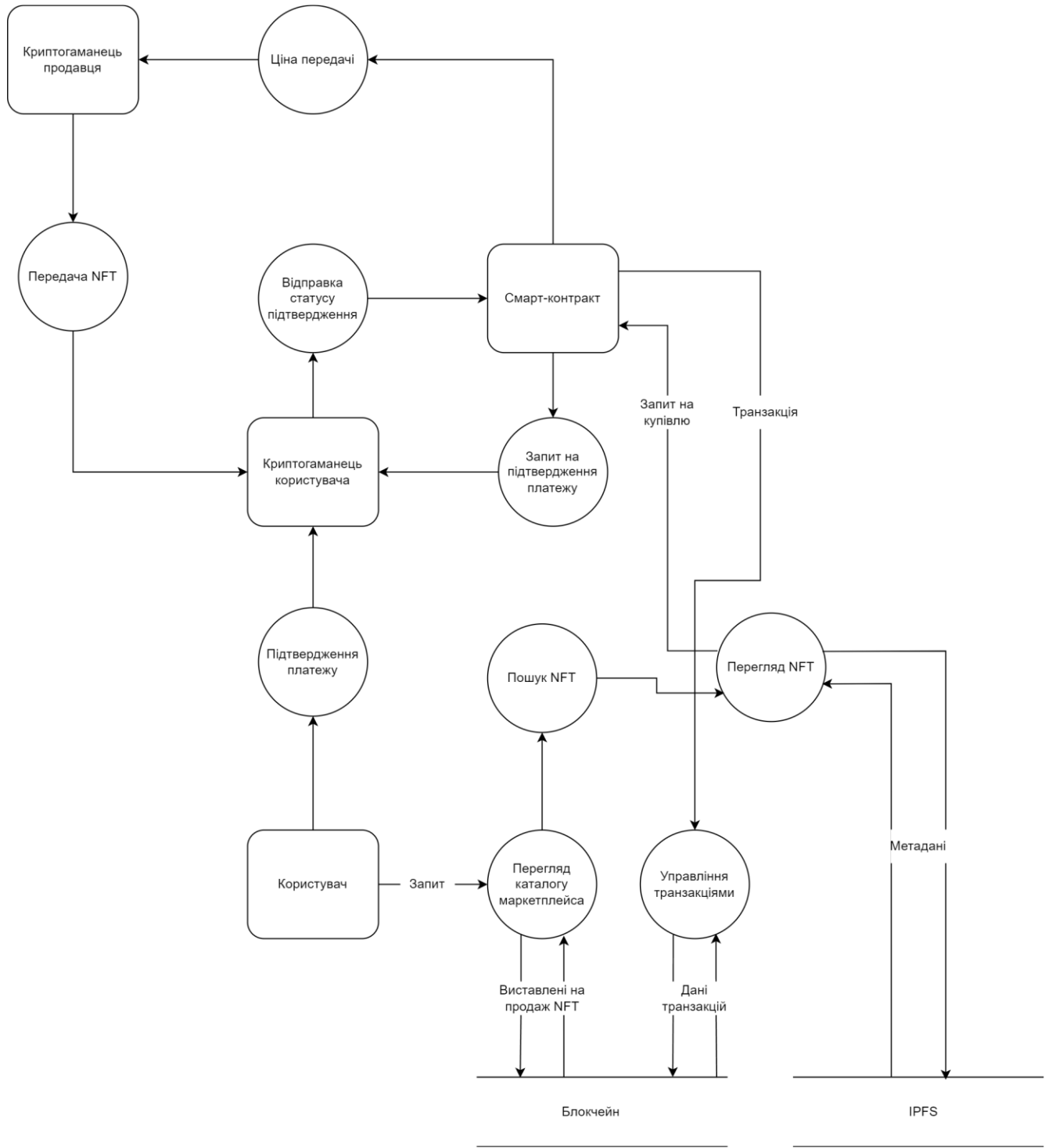
**Маркетплейс-платформа для торгівлі та створення NFT на базі
Blockchain**

Діаграма потоків даних (структурна схема)

ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2023 р



					ІАЛЦ.467200.004 Д1		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Мусійчук Я. С.			Літ.	Аркуш	Аркушів
Перевірів		Нікольський С. С.				1	1
Н. Контр.		Виноградов Ю. М.			КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-91		
Затвердив		Стіренко С. Г.					
Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain							
Діаграма потоків даних (структурна схема)							

ДОДАТОК 2

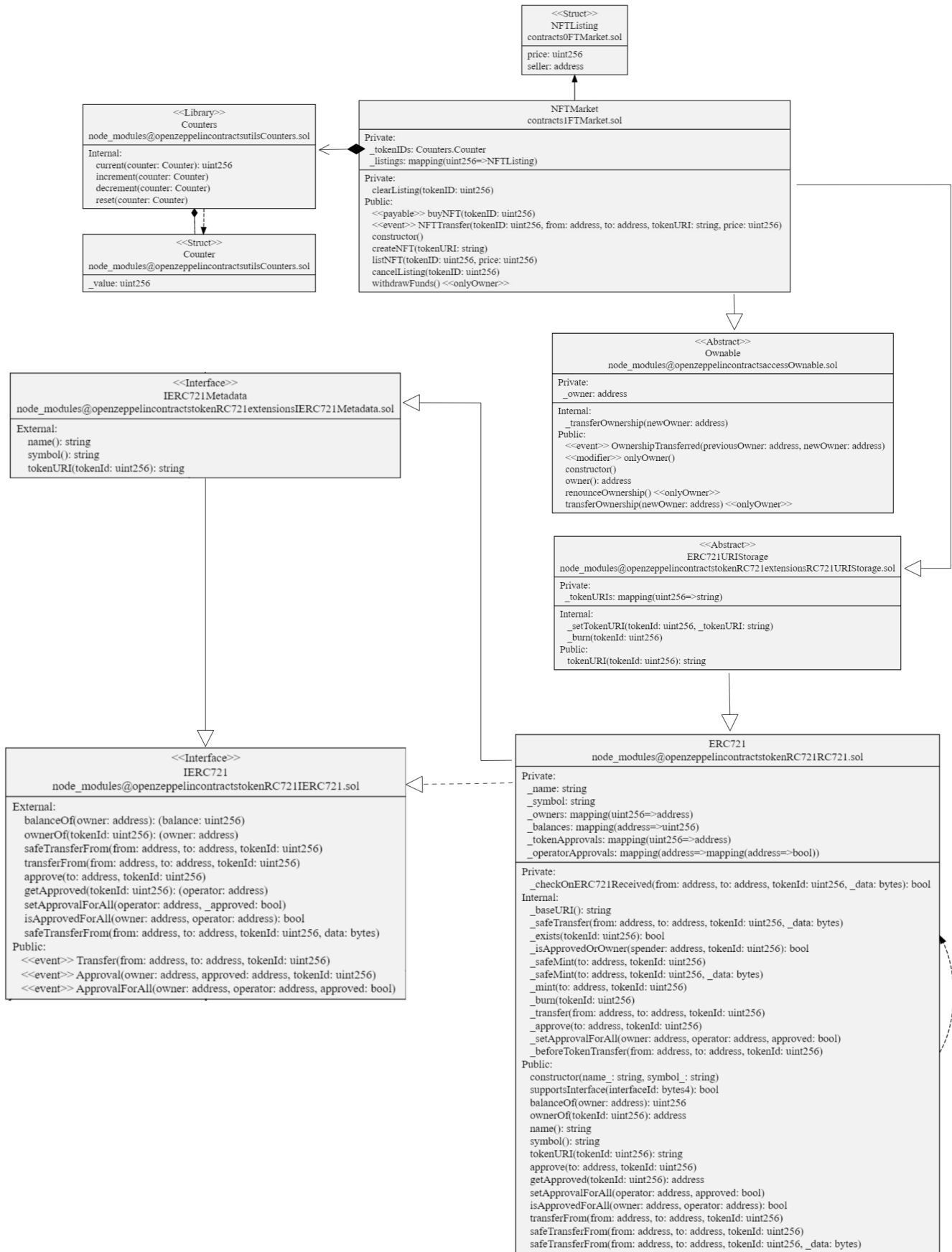
Маркетплейс-платформа для торгівлі та створення NFT на базі
Blockchain

Діаграма класів (функціональна схема)

ІАЛЦ.467200.005 Д2

Аркушів 1

Київ 2023 р



ІАЛЦ.467200.005 Д2

	№ докум.	Підпис	Дата
Розробив	Мусячук Я. С.		
Перевірив	Нікольський С. С.		
Н. Контр.	Виноградов Ю. М.		
Затвердив	Стіренко С. Г.		

Маркетплейс-платформа для
торгівлі та створення NFT на базі
Blockchain
Діаграма класів
(функціональна схема)

Літ.	Аркуш	Аркушів
	1	1
КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-91		

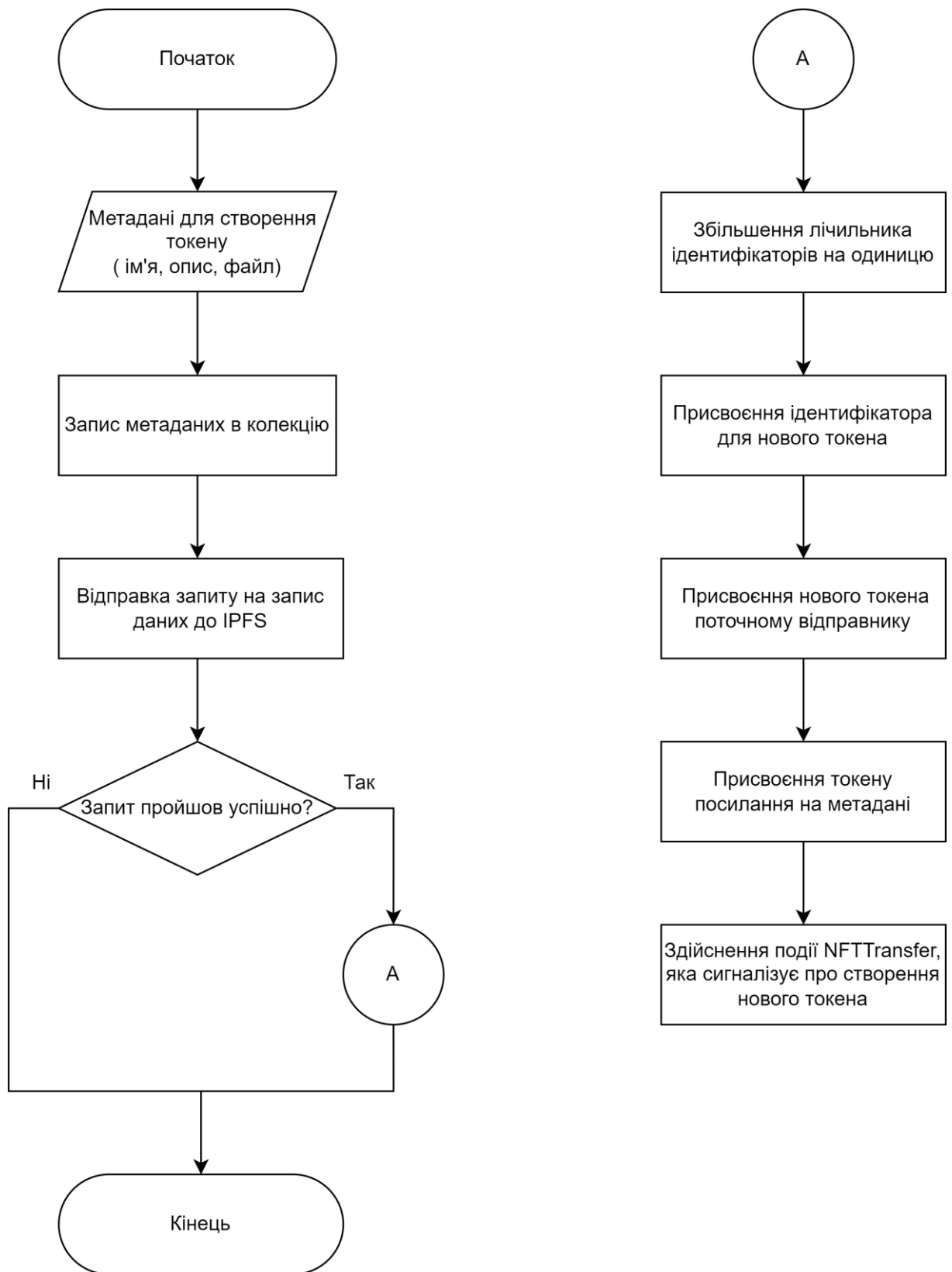
ДОДАТОК 3

**Маркетплейс-платформа для торгівлі та створення NFT на базі
Blockchain**

**Алгоритм дій програмного забезпечення (принципова схема)
ІАЛЦ.467200.006 ДЗ**

Аркушів 1

Київ 2023 р



				ІАЛЦ.467200.006 ДЗ				
		№ докум.	Підпис	Дата				
Розробив	Мусійчук Я. С.				<i>Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain</i> Алгоритм дій програмного забезпечення (принципова схема)	Літ.	Аркуш	Аркушів
Перевірив	Нікольський С. С.						1	1
Н. Контр.	Виноградов Ю. М.					КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-91		
Затвердив	Стіренко С. Г.							

ДОДАТОК 4

Маркетплейс-платформа для торгівлі та створення NFT на базі
Blockchain

Текст програмного коду
ІАЛЦ.467200.007 Д4

Аркушів 4

Київ 2023 р

Текст програмного коду смарт-контракту

```

pragma solidity ^0.8.11;
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/utils/Counters.sol";
import "@openzeppelin/contracts/utils/math/SafeMath.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

struct NFTListing {
    uint256 price;
    address seller;
}

contract NFTMarket is ERC721URIStorage, Ownable {
    using Counters for Counters.Counter;
    using SafeMath for uint256;
    Counters.Counter private _tokenIDs;
    mapping(uint256 => NFTListing) private _listings;

    event NFTTransfer(
        uint256 tokenID,
        address from,
        address to,
        string tokenURI,
        uint256 price
    );

    constructor() ERC721("MintMarket", "MINFT") {}

    function createNFT(string calldata tokenURI) public {
        _tokenIDs.increment();
        uint256 currentID = _tokenIDs.current();
        _safeMint(msg.sender, currentID);
        _setTokenURI(currentID, tokenURI);
        emit NFTTransfer(currentID, address(0), msg.sender, tokenURI, 0);
    }

    function listNFT(uint256 tokenID, uint256 price) public {
        require(price > 0, "NFTMarket: price must be greater than 0");
        transferFrom(msg.sender, address(this), tokenID);
        _listings[tokenID] = NFTListing(price, msg.sender);
        emit NFTTransfer(tokenID, msg.sender, address(this), "", price);
    }

    function buyNFT(uint256 tokenID) public payable {
        NFTListing memory listing = _listings[tokenID];
        require(listing.price > 0, "NFTMarket: nft not listed for sale");
        require(msg.value == listing.price, "NFTMarket: incorrect price");
        ERC721(address(this)).transferFrom(address(this), msg.sender, tokenID);
        clearListing(tokenID);
        payable(listing.seller).transfer(listing.price.mul(95).div(100));
        emit NFTTransfer(tokenID, address(this), msg.sender, "", 0);
    }

    function cancelListing(uint256 tokenID) public {
        NFTListing memory listing = _listings[tokenID];
    }
}

```

					ІАЛЦ.467200.007 Д4			
		№ докум.	Підпис	Дата				
Розробив	Мусійчук Я. С.				Маркетплейс-платформа для торгівлі та створення NFT на базі Blockchain Текс програмного коду	Літ.	Аркуш	Аркушів
Перевірив	Нікольський С. С.						1	4
Н. Контр.	Виноградов Ю. М.					КПІ ім. Ігоря		
Затвердив	Стіренко С. Г.					Сікорського, ФІОТ, ІВ-91		

```

    require(listing.price > 0, "NFTMarket: nft not listed for sale");
    require(listing.seller == msg.sender, "NFTMarket: you're not the seller");
    ERC721(address(this)).transferFrom(address(this), msg.sender, tokenID);
    clearListing(tokenID);
    emit NFTTransfer(tokenID, address(this), msg.sender, "", 0);
}

function withdrawFunds() public onlyOwner {
    uint256 balance = address(this).balance;
    require(balance > 0, "NFTMarket: balance is zero");
    payable(msg.sender).transfer(balance);
}

function clearListing(uint256 tokenID) private {
    _listings[tokenID].price = 0;
    _listings[tokenID].seller= address(0);
}
}

```

Текст програмного коду для відправки запиту на запис даних до IPFS

```

import { TransactionResponse } from "@ethersproject/abstract-provider";
import { BigNumber, Contract, ethers } from "ethers";
import { CreationValues } from "modules/CreationPage/CreationForm";
import useSigner from "state/signer";
import NFT_MARKET from "../..../artifacts/contracts/NFTMarket.sol/NFTMarket.json";
import { NFT_MARKET_ADDRESS } from "./config";
import { NFT } from "./interfaces";
import useListedNFTs from "./useListedNFTs";
import useOwnedListedNFTs from "./useOwnedListedNFTs";
import useOwnedNFTs from "./useOwnedNFTs";

const useNFTMarket = () => {
    const { signer } = useSigner();
    const nftMarket = new Contract(NFT_MARKET_ADDRESS, NFT_MARKET.abi, signer);

    const ownedNFTs = useOwnedNFTs();
    const ownedListedNFTs = useOwnedListedNFTs();
    const listedNFTs = useListedNFTs();

    const createNFT = async (values: CreationValues) => {
        try {
            const data = new FormData();
            data.append("name", values.name);
            data.append("description", values.description);
            data.append("image", values.image!);
            const response = await fetch("/api/nft-storage", {
                method: "POST",
                body: data,
            });
            if (response.status == 201) {
                const json = await response.json();
                const transaction: TransactionResponse = await nftMarket.createNFT(
                    json.uri
                );
                await transaction.wait();
            }
        } catch (e) {
            console.log(e);
        }
    };

    const listNFT = async (tokenID: string, price: BigNumber) => {

```

						ІАЛЦ.467200.007 Д4	Арк.
							2
Зм.	Арк.	№ докум.	Підпис	Дата			

```

const transaction: TransactionResponse = await nftMarket.listNFT(
  tokenID,
  price
);
await transaction.wait();
};

const cancellisting = async (tokenID: string) => {
  const transaction: TransactionResponse = await nftMarket.cancellisting(
    tokenID
  );
  await transaction.wait();
};

const buyNFT = async (nft: NFT) => {
  const transaction: TransactionResponse = await nftMarket.buyNFT(nft.id, {
    value: ethers.utils.parseEther(nft.price),
  });
  await transaction.wait();
};

return {
  createNFT,
  listNFT,
  cancellisting,
  buyNFT,
  ...ownedNFTs,
  ...ownedListedNFTs,
  ...listedNFTs,
};
};
};

export default useNFTMarket;

```

Текст програмного коду для авторизації через криптогаманець

```

import { JsonRpcSigner, Web3Provider } from "@ethersproject/providers";
import {
  createContext,
  ReactNode,
  useContext,
  useEffect,
  useState,
} from "react";
import Web3Modal from "web3modal";

type SignerContextType = {
  signer?: JsonRpcSigner;
  address?: string;
  loading: boolean;
  connectWallet: () => Promise<void>;
};

const SignerContext = createContext<SignerContextType>({} as any);

const useSigner = () => useContext(SignerContext);

export const SignerProvider = ({ children }: { children: ReactNode }) => {
  const [signer, setSigner] = useState<JsonRpcSigner>();
  const [address, setAddress] = useState<string>();
  const [loading, setLoading] = useState(false);

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

```

useEffect(() => {
  const web3modal = new Web3Modal();
  if (web3modal.cachedProvider) connectWallet();
  window.ethereum.on("accountsChanged", connectWallet);
}, []);

const connectWallet = async () => {
  setLoading(true);
  try {
    const web3modal = new Web3Modal({ cacheProvider: true });
    const instance = await web3modal.connect();
    const provider = new Web3Provider(instance);
    const signer = provider.getSigner();
    const address = await signer.getAddress();
    setSigner(signer);
    setAddress(address);
  } catch (e) {
    console.log(e);
  }
  setLoading(false);
};

const contextValue = { signer, address, loading, connectWallet };

return (
  <SignerContext.Provider value={contextValue}>
    {children}
  </SignerContext.Provider>
);
};

export default useSigner;

```

					ІАЛЦ.467200.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4