МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра дискретного аналізу та інтелектуальних систем
(повна назва кафедри)

Магістерська робота

Розробка NFT маркетплейсу. Децентралізований спосіб карбування та обміну токенами з використанням технології блокчейн.

Виконав: студент групи ПМіМ-23с
спеціальності
122 «Комп'ютерні науки»
(шифр і назва спеціальності)

Стрельніков В.Р.
(підпис) (прізвище та ініціали)

Керівник Коковська Я.В.
(підпис) (прізвище та ініціали)

Рецензент Бершацький Я.І.
(підпис) (прізвище та ініціали)

Львів – 2022

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет Прикладної математики та інформатики

Кафедра Дискретного аналізу та інтелектуальних систем

Спеціальність 122 «Комп'ютерні науки»

(шифр і назва)

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

Притула М. М.

"31" серпня 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Стрельнікову Віктору Романовичу

(прізвище, ім'я, по батькові)

1 . Тема роботи: Розробка NFT маркетплейсу. Децентралізований спосіб карбування та обміну токенами з використанням технології блокчейн.

керівник роботи Коковська Ярина Володимирівна, канд. фіз.-мат. наук, доц. ,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені Вченою радою факультету від **" 13 " вересня 2022 року №15**

2. Строк подання студентом роботи **12.02.2022р.**

3. Вихідні дані до роботи: Інтернет-ресурси за темою, цифрові зображення у сховищах даних із відкритим доступом. Децентралізований гаманець Metamask. Середовище розробки Visual Code.

4. Зміст магістерської роботи (перелік питань, які потрібно розробити)

1. Аналіз предметної галузі.
2. Опис використаних технологій.
3. Програма реалізація.
4. Тестування реалізації.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Архітектура та структура блокчейну.
2. Схема реєстрації токену у систему блокчейну.
3. Демонстрація роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання **31 серпня 2022 р.**

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми магістерської роботи	29.06.2022 - 03.07.2022	
2	Аналіз предметної галузі та опрацювання літератури	04.07.2022 - 08.08.2022	
3	Дослідження необхідних технологій	09.08.2022 - 28.08.2022	
4	Розробка системи реєстрації токену у блокчейн	29.08.2022 - 09.09.2022	
5	Розробка опції виставлення токену на продаж	10.09.2022 - 17.09.2022	
6	Розробка опції купівлі токену	18.09.2022 - 23.09.2022	
7	Тестування операцій та доповнення основної сторінки лістингів	24.09.2022 - 28.09.2022	
8	Розробка панелі перегляду власного цифрового балансу	29.09.2022 - 03.10.2022	
9	Дизайн платформи	04.10.2022 - 08.10.2022	
10	Створення звіту по реалізації роботи та теоретичним відомостям	09.10.2022 - 14.10.2022	
11	Написання вступу, висновків та оформлення магістерської роботи	15.10.2022 - 20.10.2022	
12	Подання магістерської роботи керівнику на рецензування	21.10.2022	

Студент Стрельніков В.Р.
Керівник Коковська Я.В.

РЕФЕРАТ

Магістерська робота складається із змісту, вступу, трьох розділів, висновку і списку використаних джерел. Загальний об’єм 32 сторінки, список джерел містить 5 найменувань, робота ілюстрована рисунками.

У даній роботі було реалізовано маркетплейс для децентралізованого способу купівлі, продажу, трансферу невзаємозамінних токенів з використанням системи блокчейну Ethereum.

Користувач здатен карбувати токен, який може містити дані формату JPEG, PNG, SVG. А також приймати певні метадані, наприклад текстову інформацію про зображення. Сформований токен можна розмістити на головній сторінці маркетплейсу, де інші користувачі здатні цей токен придбати. Наявний токен відображається у панелі власних NFT. Викарбований токен також можна відобразити та продати на NFT агрегаторі OpenSea.

Ключові слова: NFT, marketplace, token, blockchain, IPFS, Ethereum, digital property.

Зміст

Вступ…….....6
Мета роботи…….....6
Розділ 1. Теоретичні відомості..……....7
Блокчейн…….....7
Токен…….....9
Плюси та мінуси токенізації……10
Технологія NFT……11
Принцип реєстру у системі блокчейну……...13
Розділ 2. Розробка маркетплейсу.....16
Технології розробки…….....17
Реєстрація користувача…….....18
Карбування токену і лістинг……..….....20
Огляд власних NFT…….....….24
Основна панель лістингів та купівля токенів…….....25
Розділ 3. Додаткові дослідження.....29
Тестування правильності транзакцій у мережі Etherscan.....29
Взаємодія розробленого маркетплейсу з OpenSea……....30
Висновок…….....30
Перелік використаних матеріалів…….....31

Вступ

З початком всесвітньої епідемії багато культурних та технологічних подій призупинило свою діяльність. Проте в той же час з офф-лайну багато аспектів життя перейшло в онлайн. Не виключенням став розвиток генеративного мистецтва та культури NFT. Особливу нішу в NFT зайняли генеративні PFP колекції. Кожен окремий незмінний токен слугує квитком до приватних соціальних каналів, різноматіних подій та є представленням власної особистості у веб просторі. Наприклад, генеративна NFT PFP колекція CloneX, зроблена відомим митцем Такаші Муракамі, стала осередком людей, які зацікавлені в розвитку брендового одягу та його віртуальному представленні. Власники серії CloneX зможуть купувати віртуальні ассети своєму персонажу та отримати фізичний аналог в реальному житті. Багато таких проектів прагнуть до створення окремих метавсесвітів. Необхідним елементом створення всіх NFT проектів є вибір маркетплейсу, на якому токени будуть у публічному доступі.

У даній роботі я вирішив розробити один з таких маркетплейсів, який буде слугувати децентралізованим рішенням для операцій з NFT.

Метою магістерської роботи є створення програмної реалізації для закладення основи відкритого маркетплейсу. Відповідно до мети роботи розглянемо такі завдання:

- аналіз технічого завдання;
- створення смарт-контракту для токенізації результуючих зображень за допомогою Ethereum блокчейну;
- створення смарт-контракту для операцій купівлі-продажу NFT за допомогою Ethereum блокчейну;
- тестування функціональної частини;
- оформлення платформи;

РОЗДІЛ 1. Теоретичні відомості

1.1 Блокчейн.

Якщо намагатися зрозуміти суть Блокчейну, варто почати з його назви, «Blockchain» від англійського «block» – блок і «chain» – ланцюг, буквально «ланцюжок блоків», тобто. простою мовою блокчейн це інформація, що зберігається в ланцюжку, що складається з блоків.

Блокчейн — це система запису інформації таким чином, що ускладнює або робить неможливим зміну, злам або обман системи. Блокчейни, такі як Bitcoin та Ethereum, постійно зростають, оскільки блоки додаються до ланцюжка, що значно підвищує безпеку реєстру.

Ще недавно, передача вартості відбувалася за допомогою готівки, сьогодні основні дійові особи в цьому процесі - це банки, можливо, завтра - це сервіси, побудовані на технології блокчейн.

Почнемо з того, що блокчейн – це децентралізована база даних, яка призначена для зберігання послідовних блоків із набором характеристик (версія, дата створення, інформація про попередні дії в мережі). Аналоговим прикладом його структури є нескінченно довгий металевий ланцюг, в якому не можна розірвати або поміняти місцями ланки.

Основними принципами технології є:

- децентралізація;
- розподіленість;
- прозорість;
- безпека;
- незмінність;
- надійність;
- анонімність;
- швидкість;
- читабельність;

Весь ланцюжок блокчейну можна розглянути як книгу з можливістю додавати сторінки, але кожна нова сторінка пишеться після вже існуючої , інші ж не можна відредагувати, видалити чи міняти місцями. Нові «сторінки» в таку книгу додаються за допомогою транзакцій, але транзакцій не як грошовий переказ, а як виконання скрипту програми. Крім основних даних, кожен блок має унікальний набір параметрів: хеш попереднього блоку, хеш поточного блоку та список транзакцій.

В межах одного блоку може зберігатися кілька тисяч записів. Коли пам'ять у блоці закінчується, він закривається, підписується і переходить на новий блок у вигляді хеша або «відбитка».

Хеш — це якийсь набір символів, що має унікальний відбиток. Він формується виходячи з того, які транзакції і скільки зберігає в собі кожен блок.

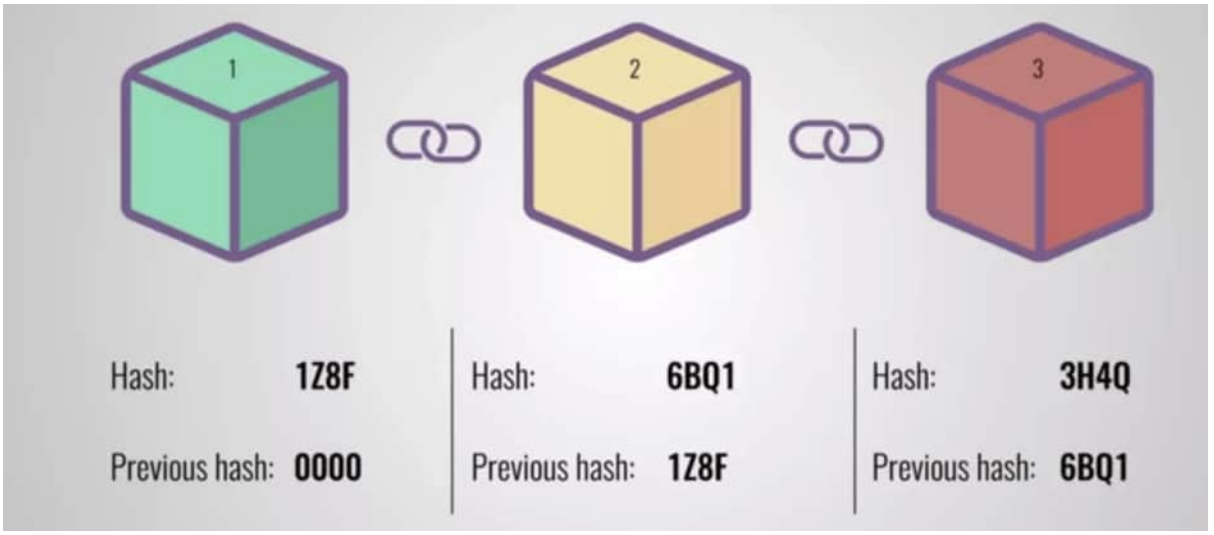

Рис 1.1 - схема блоків у системі блокчейну

При обробці транзакцій у блокчейні постійно перевіряються хеші, після чого система піднімається до останнього хешу, де підтверджується цілісність та вірність усіх попередніх кодів, щоб блок закрився.

Кожен блок містить у собі інформацію про попередній блок, таким чином не можливо підробити оригінальність блоку у цій системі.

Ця постійна перевірка виключає можливість того, щоб хтось додав собі «зайві» монети та надіслав недостовірну інформацію до мережі – такий блок не пройде перевірку та не буде доданий до мережі. Від сюди і назва - ланцюжок, який працює безперервно і додає ланки.

Проте, як відбувається транзакція в блокчейні? Для того щоб провести банківський переказ користувачеві необхідно відкрити рахунок використовуючи свої персональні дані, внести на нього кошти і тільки після цього можна проводити транзакції, які повинні відповідати вимогам фінансових організацій.

Для здійснення транзакції в блокчейні користувачеві необхідно лише два ключі: Публічний ключ та Приватний ключ.

«Публічний ключ» — це набір із цифр та символів, доступний для перегляду всім бажаючим у мережі блокчейну — це номер гаманця, його адреса, що використовується для переказу коштів.

"Приватний ключ" - це найцінніше. З його допомогою підписуються всі транзакції в гаманці, тому його потрібно дбайливо зберігати у надійному місці.

Всю інформацію, зашифровану приватним ключем користувача, будь-яка людина може розшифрувати за допомогою свого публічного ключа, але не може відкрити гаманець або переказати кошти. Таким чином можна дізнатися інформацію щодо кожної транзакції будь-якого рахунку, така система повністю прозора, але при цьому вона анонімна. Блокчейн не зберігає жодної особистої інформації та виявити власників приватних ключів важко.

1.2 Токен.

Якщо пояснювати простими словами, то токен – це цифровий сертифікат, який гарантує зобов'язання компанії перед його власником, аналог акцій на фондовій біржі у світі криптовалют. З появою комп’ютерів та Інтернету токени стали зручною технологією для обміну між цифровими, а деколи і фізичними активами. Токен, як технологія, виникла до появи блокчейну. До прикладу, система PayPal використовує власні токени.

У віртуальному світі токен є цифровою умовною одиницею, вартість якої виражається в будь-якому активі. Він синхронізований з базою даних, побудованою на технології блокчейн, де ведеться підрахунок усіх токенів. Отримати доступ до віртуальних токенів можна лише за наявності електронного підпису та через відповідну програму.

Токенізація – це трансформація активу на цифрову одиницю. Простіше кажучи, цей процес є перетворенням будь-якого активу реального світу на цифровий актив у вигляді однієї умовної одиниці, інформація про яку зберігається в блокчейні. Така трансформація дозволяє власнику токенів набагато безпечніше та швидше взаємодіяти з реальними активами.

Наведемо приклад - власник сирної лавки токенізував свій продукт. За належного попиту на його сир, власник може продавати свій продукт за допомогою токенів на зручному йому цифровому майданчику. Після цього люди, які купили його монети, можуть прийти до лавки і обміняти одиницю токену на один брусок сиру.

1.3 Плюси та мінуси токенізації

У токенізації є свої переваги:

- Удосконалення та прискорення торговельного обміну;
- Безпечне зберігання монет та перекази між власниками;
- Можливість здійснювати торгівлю без гарантів чи посередників, оскільки всі переклади регулюються через смарт-контракти;
- Розширення можливостей інфраструктури та торгового обміну в цілому;
- Значне спрощення роботи з продавцями товарів чи послуг за рахунок інтеграції системи у спеціальні мобільні програми.
- Підвищення безпеки бази даних, що можливе завдяки ретельній перевірці інформації, що надходить;
- Високошвидкісна обробка транзакцій за рахунок великої кількості незалежних серверів;
- Створення безпечного аудиту.
- У системі блокчейну вони надають децентралізований спосіб обміну.

Тим не менш, незважаючи на всі перераховані переваги токенізація може стати проблемою. На це є такі причини:

- Існує ризик втрати чи крадіжки особистих прав користувачів через кібератаки;
- Забезпечити анонімність даних у публічному блокчейні практично нездійсненне завдання;
- Складність розширення системи за децентралізованої роботи з монетами через ліміти за кількістю транзакцій.

1.4 Технологія NFT

NFT, або незмінний токен, - це одиниця обліку, за допомогою якої створюється цифровий зліпок для будь-якого унікального предмета. Серед них можуть бути: картини, фотографії, відео, музика, гіфки — словом, будь-який контент, який претендує на хоч якусь унікальність. Вони є великою цінністю серед колекціонерів, геймерів та любителів мистецтва, а купують та продають їх через аукціони.

Зберігаються ці самі токени в блокчейні - величезному ланцюжку блоків, кожен з яких містить інформацію. На відміну від, наприклад, серверів, де дані зберігаються в одному місці, ці блоки можуть знаходитися на безлічі пристроїв у різних куточках світу.

Такий спосіб шифрування робить блокчейн дуже важким для злому, тому що в кращому випадку вам вдасться зламати лише один блок з інформацією, а не весь ланцюжок цілком. Саме тому на блокчейні працює більшість криптовалют.

Токен - просто запис в одному з блоків, і, зазвичай, таких однотипних записів може бути дуже багато. Наприклад, кожен окремий біткоїн є точною копією іншого такого ж біткоїну, що дає можливість зіставляти їх із валютою.

Токен - просто запис в одному з блоків, і, зазвичай, таких однотипних записів може бути дуже багато. Наприклад, кожен окремий біткоїн є точною копією іншого такого ж біткоїну, що дає можливість зіставляти їх із валютою.

Але що робити у випадку, якщо потрібно створити унікальний токен, який не має аналогів? Відповідь - NFT.

NFT-токен - це спосіб перевести унікальні предмети з реального світу до блокчейну. Кожен з цих токенів неповторний, неподільний і існує в однині. Крім того, в блокчейні надійно зберігається вся необхідна інформація про нього.

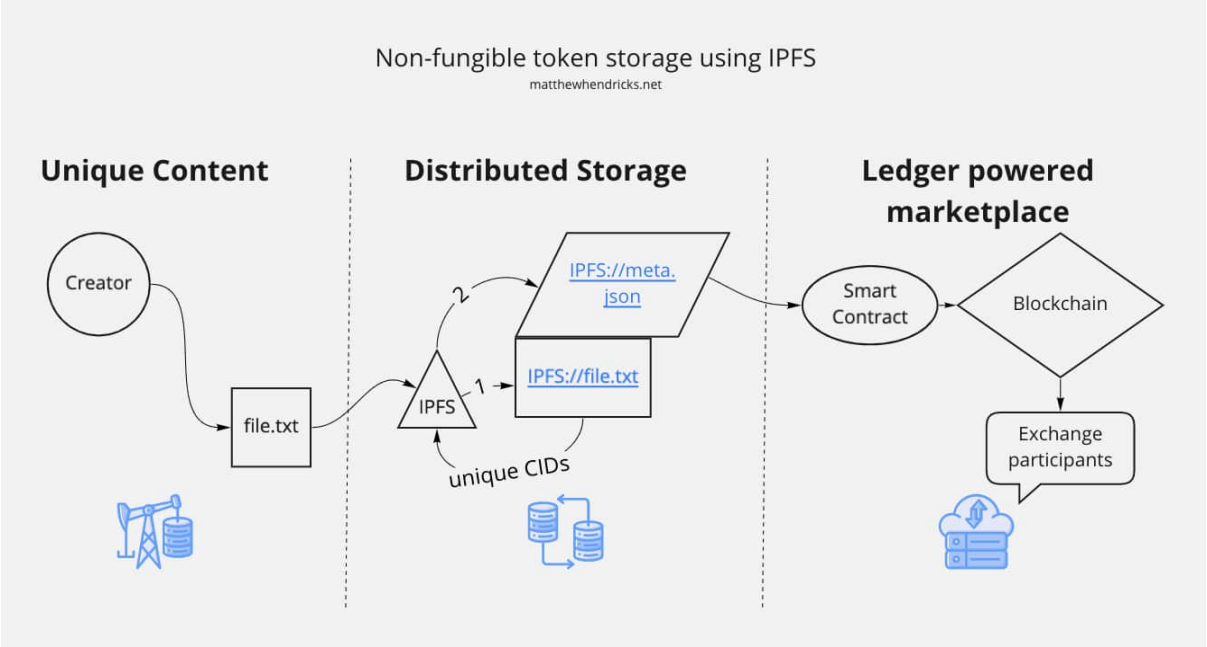

Рис 1.2 - схема запису файлу та метаданих у систему блокчейну

На Рис 1.2 бачимо схему запису текстового файлу та метаданих у систему блокчейну.

На першому кроці потрібно обрати необхідні дані для використання. Нехай це буде file.txt.

На другому кроці цей файл потрібно записати у протокол IPFS. Система збереже два окремі елементи: IPFS://file.txt ; IPFS://meta.json.

На останньому кроці ми зберігаємо посилання на посилання IPFS в комірку блокчейну за допомогою смарт-контракту.

1.5 Принцип реєстру у системі блокчейну

Отже, визначивши поняття блокчейну, токену та незмінного токену можемо перейти до реалізації нашого смарт-контракту, який проведе операцію токенізації та занесе наші матедані до системи блокчейну.

Для початку варто вибрати середовище розробки. Для цього я використовував Python, JavaScript та Moralis [1,2]. Сценарій роботи програми досить зрозумілий. Запустивши нашу веб-аплікацію, можемо побачити поля для заповнення необхідними даними. Перед цим, щоб мати змогу обробити дані в блокчейні треба підключати наш децентралізований носій. У моєму випадку таким носієм виступає криптогаманець Metamask. [3,5]

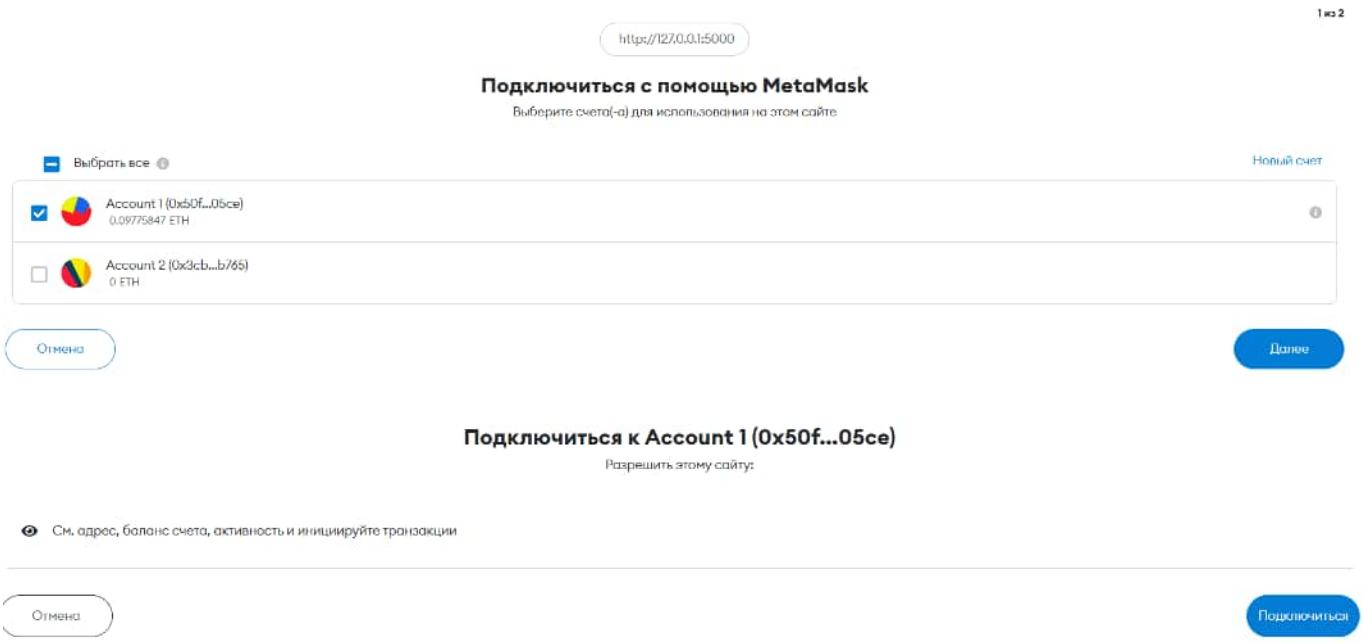

Рис 1.3 - підключення до гаманця

Нам потрібно вибрати необхідний аккаунт, на якому є достатня сума для проведення тестових транзакцій. У нашому випадку - це Acccount 1.

Далі підтверджуємо актуальність нашої операції за допомогою цифрового підпису.

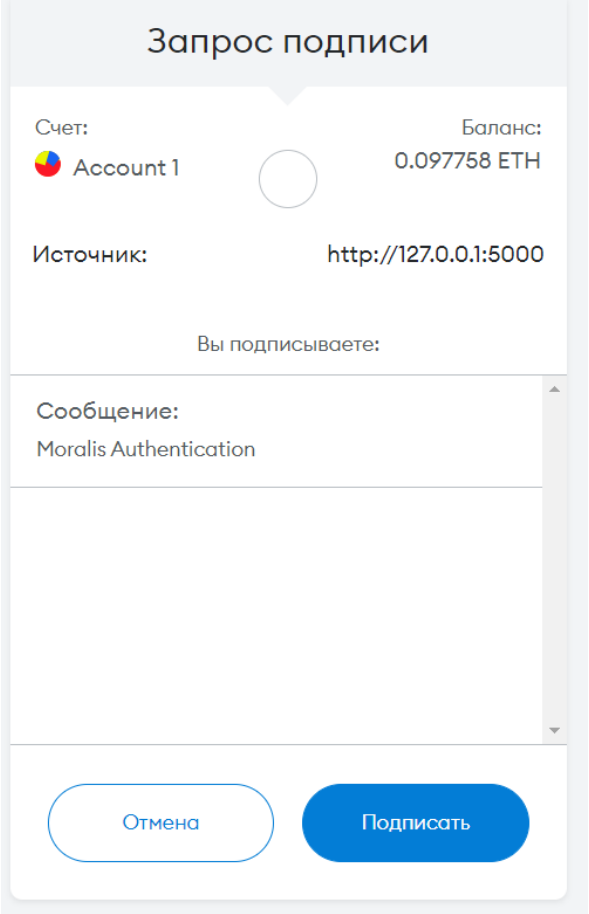

Рис 1.4 - створення цифрового підпису

Все це було реалізовано через метод Login. Він дає нам змогу передати певні атрибути користувача необхідні для логування через гаманець.

```
async function login(){
  document.getElementById('submit').setAttribute("disabled", null);
  document.getElementById('username').setAttribute("disabled", null);
  document.getElementById('useremail').setAttribute("disabled", null);
  Moralis.Web3.authenticate().then(function (user) {
      user.set("name",document.getElementById('username').value);
      user.set("email",document.getElementById('useremail').value);
      user.save();
      document.getElementById("upload").removeAttribute("disabled");
      document.getElementById("file").removeAttribute("disabled");
      document.getElementById("name").removeAttribute("disabled");
      document.getElementById("description").removeAttribute("disabled");
  })
}
```

Рис 1.5 - фрамент коду функції Login

У разі, якщо підключення гаманця пройшла успішно - можемо приступати до введення даних.

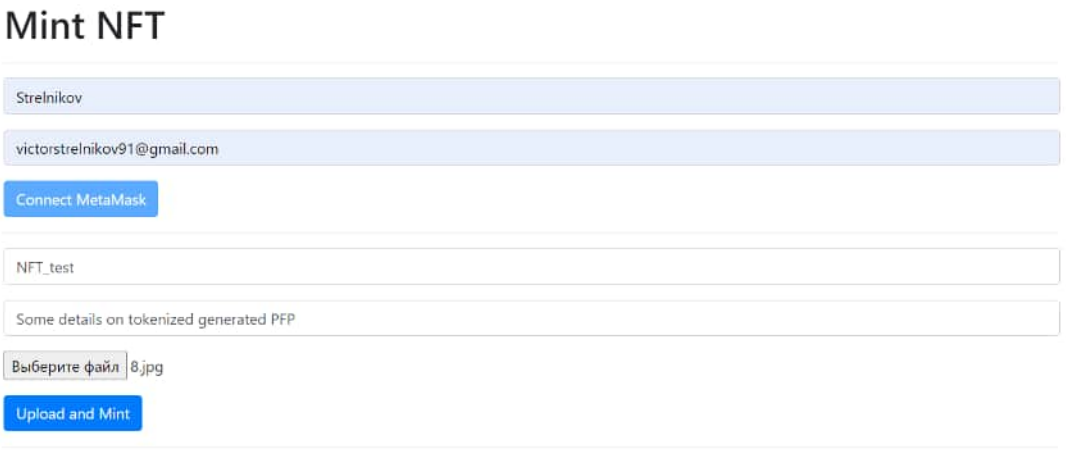

Рис 1.6 - фрагмент програми полів вводу інформації

```
async function upload(){
  const fileInput = document.getElementById("file");
  const data = fileInput.files[0];
  const imageFile = new Moralis.File(data.name, data);
  document.getElementById('upload').setAttribute("disabled", null);
  document.getElementById('file').setAttribute("disabled", null);
  document.getElementById('name').setAttribute("disabled", null);
  document.getElementById('description').setAttribute("disabled", null);
  await imageFile.saveIPFS();
  const imageURI = imageFile.hash();
  const metadata = {
    "name":document.getElementById("name").value,
    "description":document.getElementById("description").value,
    "image":"https://ipfs.io/ipfs/"+imageURI
  }
  const metadataFile = new Moralis.File("metadata.json", {base64 : btoa(JSON.stringify(metadata))});
  await metadataFile.saveIPFS();
  const metadataURI = "https://ipfs.io/ipfs/" + metadataFile.hash();
  const txt = await mintToken(metadataURI).then(notify)
}
```

Рис 1.7 - фрамент коду функції Upload

На Рис 1.7 можемо побачити фрагмент коду для запису нашого файлу у систему IPFS, як було раніше схематично зображено на ілюстрації Рис 1.2. Потрібно отримати кілька атрибутів для того, щоб здійснити необхідний запис: назва файлу, сам файл та опис, тобто наші метадані. Далі створюється спеціальне посилання по якому ми можемо надалі знайти наші матеріали та використати їх.

Після цього підтверджуємо ‘мінтинг’ нашого зображення.

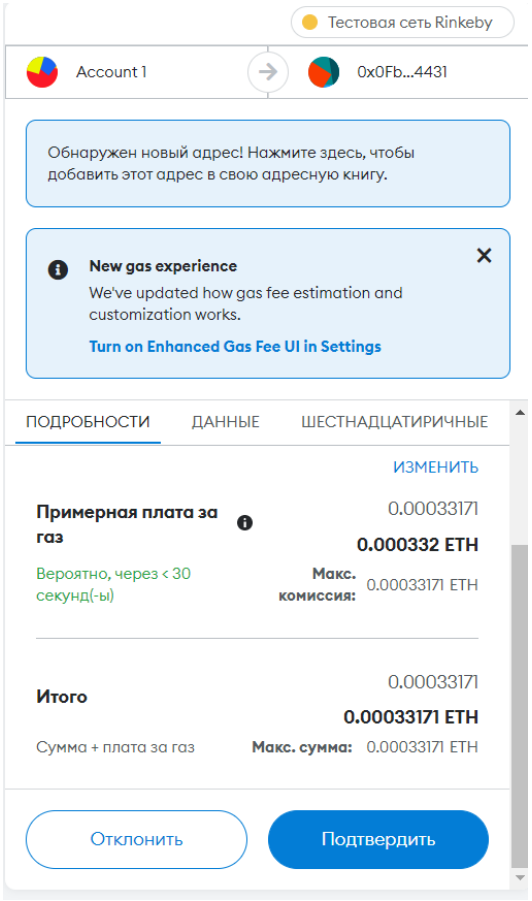

Рис 1.8 - підтвердження транзакції і оплата комісії за газ

РОЗДІЛ 2. Розробка маркетплейсу

Давайте перейдемо безпосередньо до реалізації нашого маркетплейсу. Спочатку нам потрібно виконати базову операцію реєстрації користувача. Далі для того, щоб оперувати нашим токеном, потрібно його створити. Після чого він буде відображений на маркетплейсі. Для зручності користувачів реалізуємо перегляд ‘цифрової власності’. Далі переглянемо всі наявні токени на нашому маркетплейсі по всіх користувачах, виберемо уподобані та придбаємо їх, використовуючи тестову версію токенів ERC-20.

2.1 Технології розробки

Для розробки моєї роботи я використовував наступні технології:

1. Solidity

Solidity — це мова програмування, створена для розробки смарт контрактів, які працюють на Ethereum Virtual Machine (EVM). Аплікації створені на Solidity перетворюються в байткод віртуальної машини Ethereum. Розробники, які програмують на мові програмування Solidity можуть створювати програми, що передбачають логіку записів транзакцій у блокчейн.

2. React

React — бібліотека розроблена мовою програмування JavaScript для створення веб-аплікацій. Вона допомагає зручно імплементувати можливість оновлення вебсторінки, що часто трапляється у розробці односторінкових застосунків. Ця технологія належить коропорації Meta.

3. Blockchain

Блокчейн — це, по суті, цифрова книга транзакцій, яка дублюється та розподіляється по всій мережі комп'ютерних систем у блокчейні. Кожен блок у ланцюжку містить кілька транзакцій, і щоразу, коли в блокчейні виникає нова транзакція, запис про цю транзакцію додається до книги кожного учасника. Децентралізована база даних, якою керують кілька учасників, відома як технологія розподіленої книги (DLT).

Це означає, що якщо один блок в одному ланцюжку було змінено, одразу було б очевидно, що його втрутилися. Якби хакери хотіли пошкодити систему блокчейну, їм довелося б змінити кожен блок у ланцюжку, у всіх розподілених версіях ланцюжка.

4. MetaMask

MetaMask — це програмний криптовалютний гаманець, який використовується для взаємодії з блокчейном Ethereum. Це дозволяє користувачам отримувати доступ до свого гаманця Ethereum через розширення браузера або мобільний додаток, який потім можна використовувати для взаємодії з децентралізованими програмами. MetaMask розроблено ConsenSys Software Inc., компанією, що розробляє блокчейн-програмне забезпечення, що зосереджується на інструментах та інфраструктурі на основі Ethereum.

2.2 Реєстрація користувача

Для того, щоб реалізувати логування користувачів у середовищі web3 використовують децентралізовані гаманці. Вони допомагають зберігати власні цифрові ресурси у досить простій та зручній формі. До таких носіїв відносять Metamask, Trust Wallet, Coinbase wallet, Phantom та інші. У даній роботі я використав гаманець Metamask через його популярність та зручний інтерфейс.

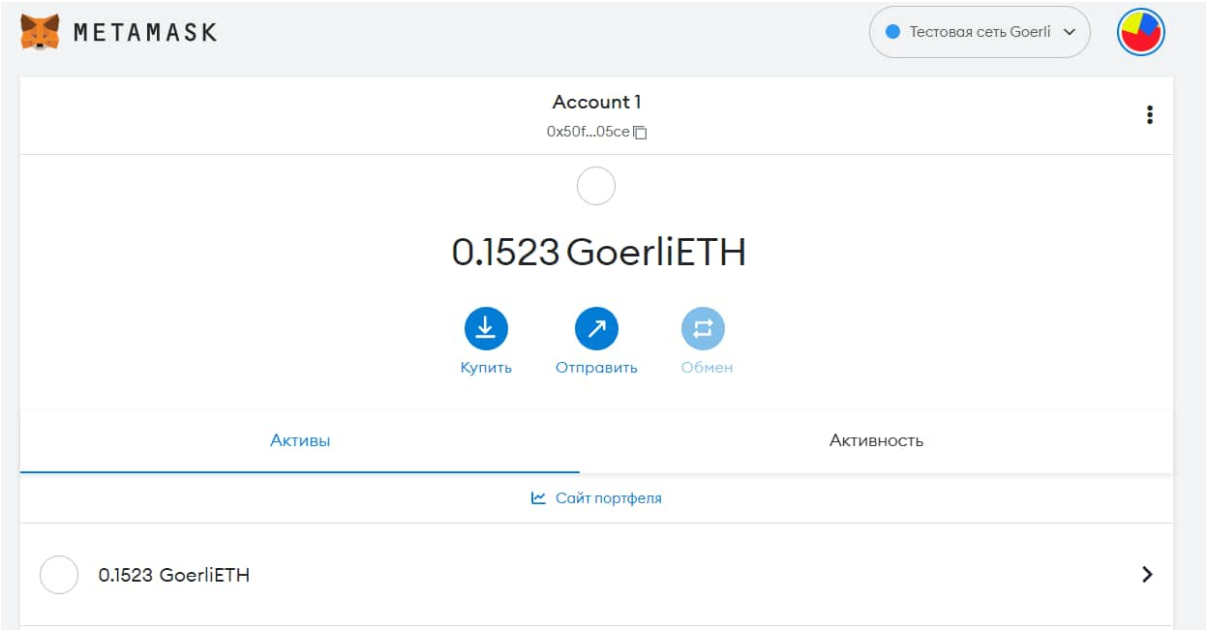

Рис 2.1 - вигляд балансу гаманця MetaMask

Для того, щоб ми могли працювати у тестовій варіанті блокчейну Ethereum я вибрав мережу Goerli. Далі на свій гаманець я перечислив тестові токени, які можна отримати безплатно на сайті Goerli Faucet.

Автентифікацію користувача проведемо за допомогою функції Web3Provider. Вона створена для отримання даних про гаманець. Зокрема, мережу блокчейну, адресу гаманця, баланс токенів та наявні NFT. [6,7]

```
export default function Profile () {
    const [data, updateData] = useState([]);
    const [dataFetched, updateFetched] = useState(false);
    const [address, updateAddress] = useState("0x");
    const [totalPrice, updateTotalPrice] = useState("0");

    async function getNFTData(tokenId) {
        const ethers = require("ethers");
        let sumPrice = 0;
        const provider = new ethers.providers.Web3Provider(window.ethereum);
        const signer = provider.getSigner();
        const addr = await signer.getAddress();

        let contract = new ethers.Contract(MarketplaceJSON.address, MarketplaceJSON.abi, signer)

        //create an NFT Token
        let transaction = await contract.getMyNFTs()


        const items = await Promise.all(transaction.map(async i => {
            const tokenURI = await contract.tokenURI(i.tokenId);
            let meta = await axios.get(tokenURI);
            meta = meta.data;

            let price = ethers.utils.formatUnits(i.price.toString(), 'ether');
            let item = {
                price,
                tokenId: i.tokenId.toNumber(),
                seller: i.seller,
                owner: i.owner,
                image: meta.image,
                name: meta.name,
                description: meta.description,
            }
            sumPrice += Number(price);
            return item;
        }))
```

Рис 2.2 - фрагмент коду функції Profile

Варто зазначити, що для логування за допомогою гаманця Metamask потрібно підписати смарт-контракт. Це своєрідна оперція підтвердження особистості. Після чого гаманець буде успішно з’єднано з необхідним додатком або сайтом.

2.3 Карбування токену і лістинг

Перейдемо до реалізації можливості карбування NFT. У попередніх матеріалах було згадано алгоритм створення невзаємозамінного токену. По такій же структурі буде реалізований і даний фрагмент аплікації. Перечислимо порядок операцій:

1) Підготовка необхідних матеріалів для запису у блокчейн. Це можуть бути файли формату JPEG, PNG, SVG, etc. Для цього я використав генеративні зображення, які були створені мною протягом минулої роботи.

Рис 2.3 - приклади зображень взяті для тестування маркетплейсу

2) Запис матеріалу у протокол IPFS.

```
async function OnChangeFile(e) {
    var file = e.target.files[0];
    try {
        //upload the file to IPFS
        const response = await uploadFileToIPFS(file);
        if(response.success === true) {
            console.log("Uploaded image to Pinata: ", response.pinataURL)
            setFileURL(response.pinataURL);
        }
    }
    catch(e) {
        console.log("Error during file upload", e);
    }
}
```

Рис 2.4 - фрагмент коду функції OnChangeFile

```
async function uploadMetadataToIPFS() {
    const {name, description, price} = formParams;
    if( !name || !description || !price || !fileURL)
        return;

    const nftJSON = {
        name, description, price, image: fileURL
    }

    try {
        //upload the metadata JSON to IPFS
        const response = await uploadJSONToIPFS(nftJSON);
        if(response.success === true){
            console.log("Uploaded JSON to Pinata: ", response)
            return response.pinataURL;
        }
    }
    catch(e) {
        console.log("error uploading JSON metadata:", e)
    }
}
```

Рис 2.5 - фрагмент коду функції uploadMetadataToIPFS

Запис метаданих.

3) Запис у систему блокечейну на основі посилання протоколу.

```
async function listNFT(e) {
    e.preventDefault();

    //Upload data to IPFS
    try {
        const metadataURL = await uploadMetadataToIPFS();
        const provider = new ethers.providers.Web3Provider(window.ethereum);
        const signer = provider.getSigner();
        updateMessage("Please wait.. uploading (upto 5 mins)")

        let contract = new ethers.Contract(Marketplace.address, Marketplace.abi, signer)

        const price = ethers.utils.parseUnits(formParams.price, 'ether')
        let listingPrice = await contract.getListPrice()
        listingPrice = listingPrice.toString()

        // create the NFT
        let transaction = await contract.createToken(metadataURL, price, { value: listingPrice })
        await transaction.wait()

        alert("Successfully listed your NFT!");
        updateMessage("");
        updateFormParams({ name: '', description: '', price: ''});
        window.location.replace("/")
    }
    catch(e) {
        alert( "Upload error"+e )
    }
}
```

Рис 2.6 - фрагмент коду функції listNFT

Крім цього у нашій програмі імплементуємо поле для введення ціни за токен. Це буде важливо для можливості продажі NFT. Ціна вказується у іншому токені стандарту ERC-20.

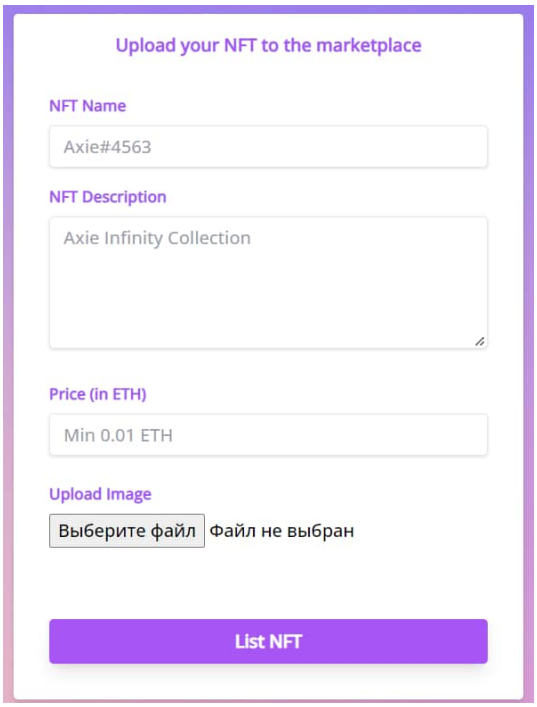

Рис 2.7 - фінальний вигляд панелі мінтингу NFT

Після створення NFT ми можемо пересвідчитися у правильності проведення реєстрації у мережі блокчейну, використовуючи сервіс для відстеження транзакцій Etherscan. У ньому за пошуком нашої підключеної адреси повинна появитися комірка даних про транзакцію. Цей запис містить інформацію про створений токен та інші деталі, такі як час, оплата за газ, адресати і навіть частини коду.

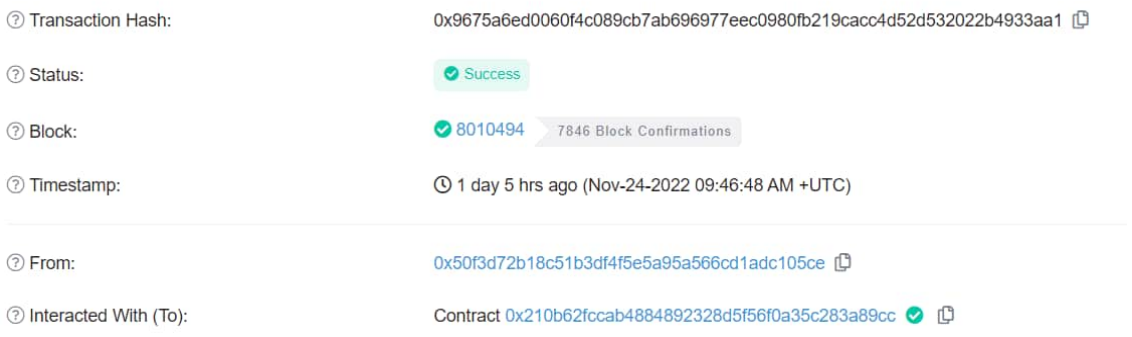

Рис 2.8 - транзакція у Etherscan

Крім цього, ми можемо відстежити створене NFT та трохи інформації у гаманці Metamask.

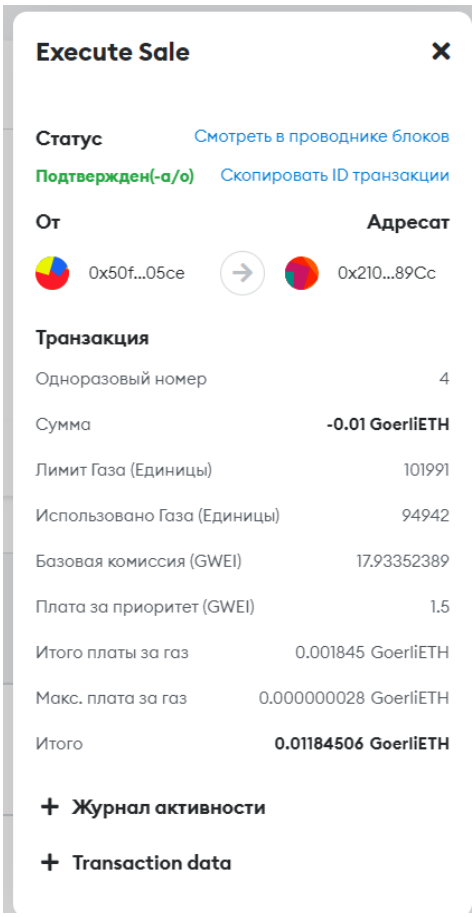

Рис 2.9 - транзакція у Metamask

Проте для чого нам використовувати сторонні сервіси, якщо було б зручніше реалізувати панель перегляду власної ‘цифрової власності’.

2.4 Огляд власних NFT

Отже, давайте перейдемо до створення окремої сторінки, де ми можемо переглянути власні невзаємозамінні токени.

Для того, щоб це зробити нам треба зчитати всі наявні токени у системі блокчейну Ethereum по власній адресі. За допомогою таких функцій як *tokenURI* та *contract* не виникає проблеми це реалізувати. Сформуємо отримані дані в окремі юніти. Далі відобразимо всі юніти через HTML розмітку. [1,2]

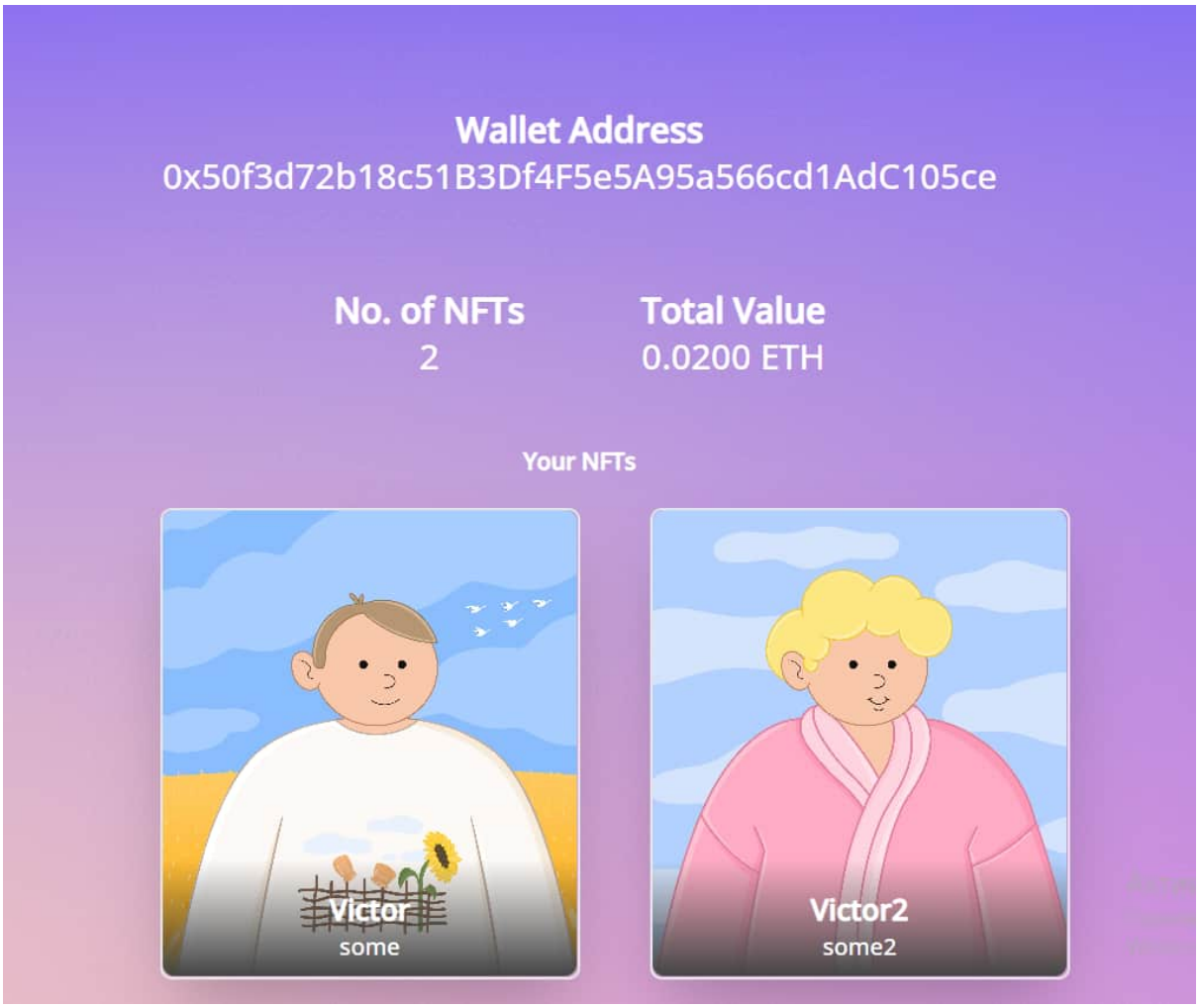

Рис 2.10 - вигляд панелі перегляду власних NFT

Бачимо, що NFT яке було створене на попередньому кроці відображається у комірці. Також вказується кількість наявних токенів та їхня загальна вартість.

Давайте перейдемо до можливості продажу нашого токену.

2.5 Основна панель лістингів та купівля токенів

Для того, щоб реалізувати повноцінний перелік всіх NFT на маркетплейсі потрібно створити смарт-контракт на основі окремої адреси, яка буде слугувати ‘’колекціонером’’ всіх адрес та операцій по відношенню до неї. Це означає, що через сторонню адресу, яка презентує наш маркетплейс, ми зможемо визначати всі дії всередині платформи.

Щоб мати таку можливість нам потрібно задеплоїти смарт контракт у тестову мережу Goerli. Після чого ми зможемо пересвідчитися у правильності виконання операції на Etherscan.

Рис 2.11 - операції на Etherscan

Після цього у нашій програмній аплікації стоворимо функцію getAllNFTs , яка парсуватиме всі можливі токени. [4,7,8]

```
async function getAllNFTs() {
    const ethers = require("ethers");
    const provider = new ethers.providers.Web3Provider(window.ethereum);
    const signer = provider.getSigner();
    let contract = new ethers.Contract(MarketplaceJSON.address, MarketplaceJSON.abi, signer)
    let transaction = await contract.getAllNFTs()

    const items = await Promise.all(transaction.map(async i => {
        const tokenURI = await contract.tokenURI(i.tokenId);
        let meta = await axios.get(tokenURI);
        meta = meta.data;

        let price = ethers.utils.formatUnits(i.price.toString(), 'ether');
        let item = {
            price,
            tokenId: i.tokenId.toNumber(),
            seller: i.seller,
            owner: i.owner,
            image: meta.image,
            name: meta.name,
            description: meta.description,
        }
        return item;
    }))

    updateFetched(true);
    updateData(items);
}
```

Рис 2.12 - фрагмент функції getAllNFTs

У переліку наявних NFT можемо вибрати уподобане зображення.

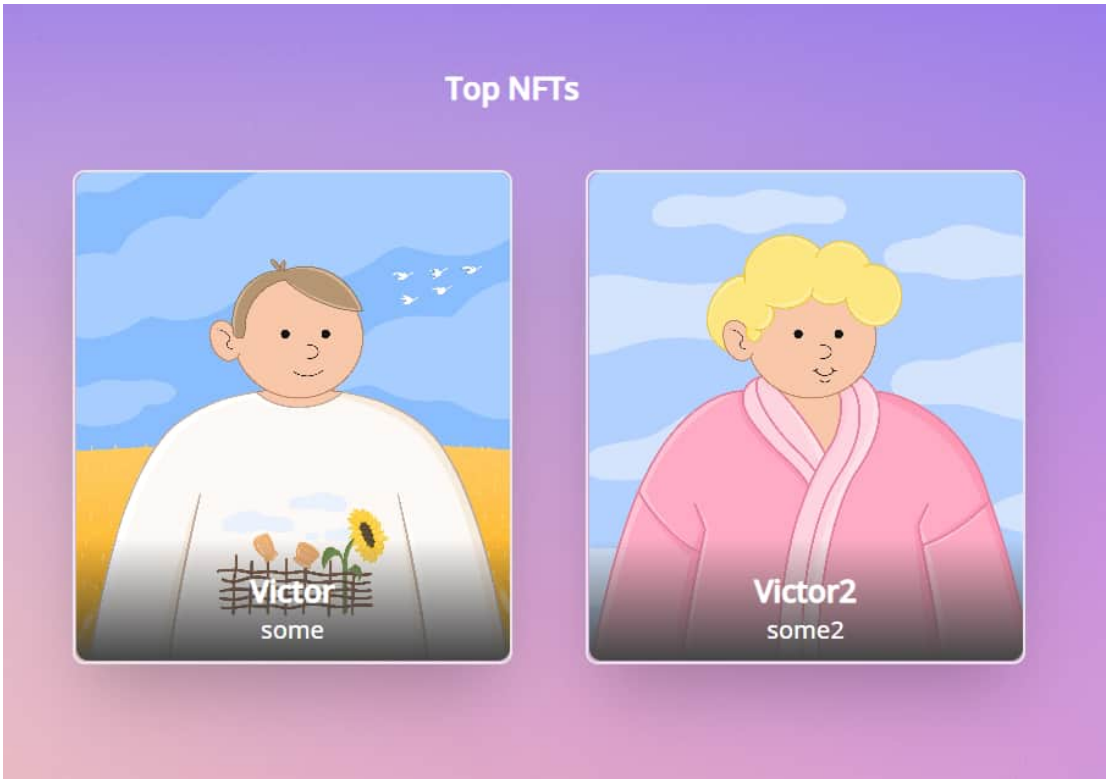

Рис 2.13 - основна панель маркетплейсу

Залишилося його придбати. Це нам вдасться зробити за допомогою функції executeSale , яка приймає два аргументи: номер токену та виставлену ціну у Ethereum. Фактично, проводячи взаємообмін токенами нам вдається продати NFT за бажану ціну.

```
async function buyNFT(tokenId) {
    try {
        const ethers = require("ethers");
        const provider = new ethers.providers.Web3Provider(window.ethereum);
        const signer = provider.getSigner();

        let contract = new ethers.Contract(MarketplaceJSON.address, MarketplaceJSON.abi, signer);
        const salePrice = ethers.utils.parseUnits(data.price, 'ether')
        updateMessage("Buying the NFT... Please Wait (Upto 5 mins)")
        let transaction = await contract.executeSale(tokenId, {value:salePrice});
        await transaction.wait();

        alert('You successfully bought the NFT!');
        updateMessage("");
    }
    catch(e) {
        alert("Upload Error"+e)
    }
}
```

Рис 2.14 - фрагмент функції buyNFT

Приклад проведення купівлі токену:

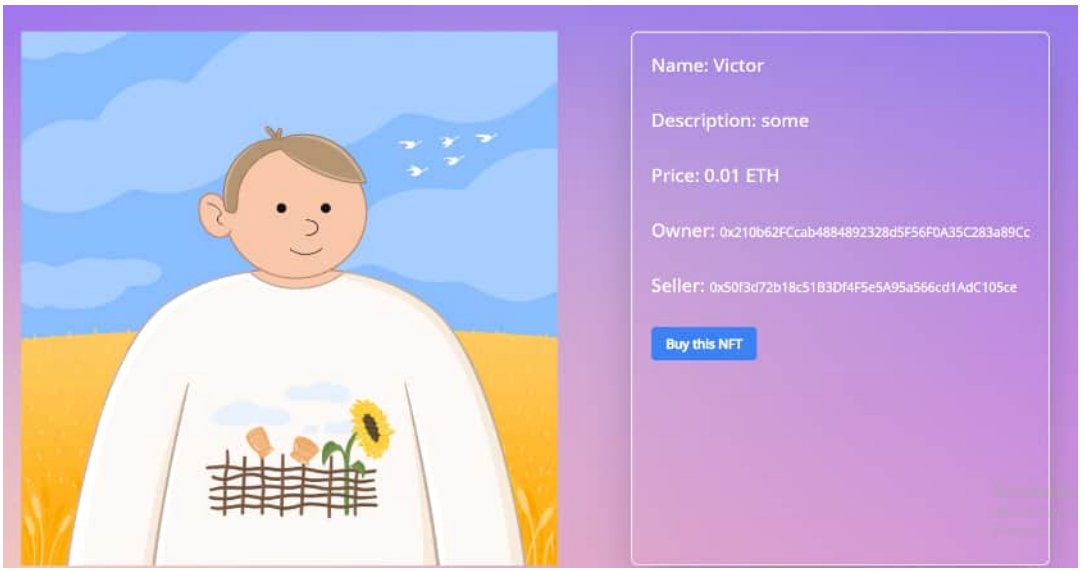

Рис 2.15 - приклад панелі купівлі NFT

Підтверджуємо транзакцію:

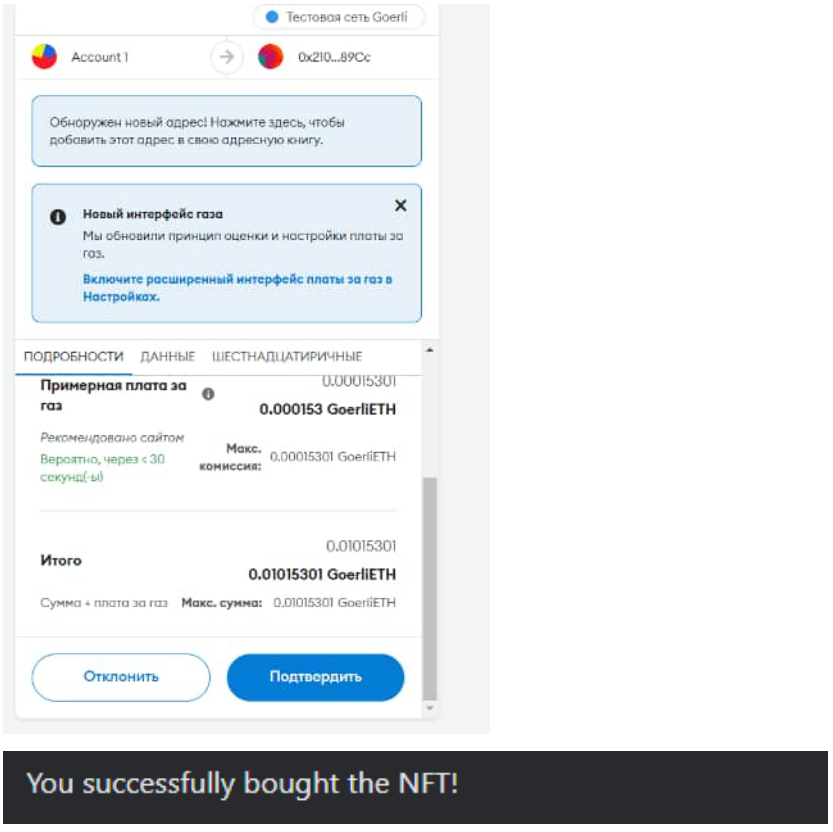

Рис 2.16 - підтвердження проведення транзакції

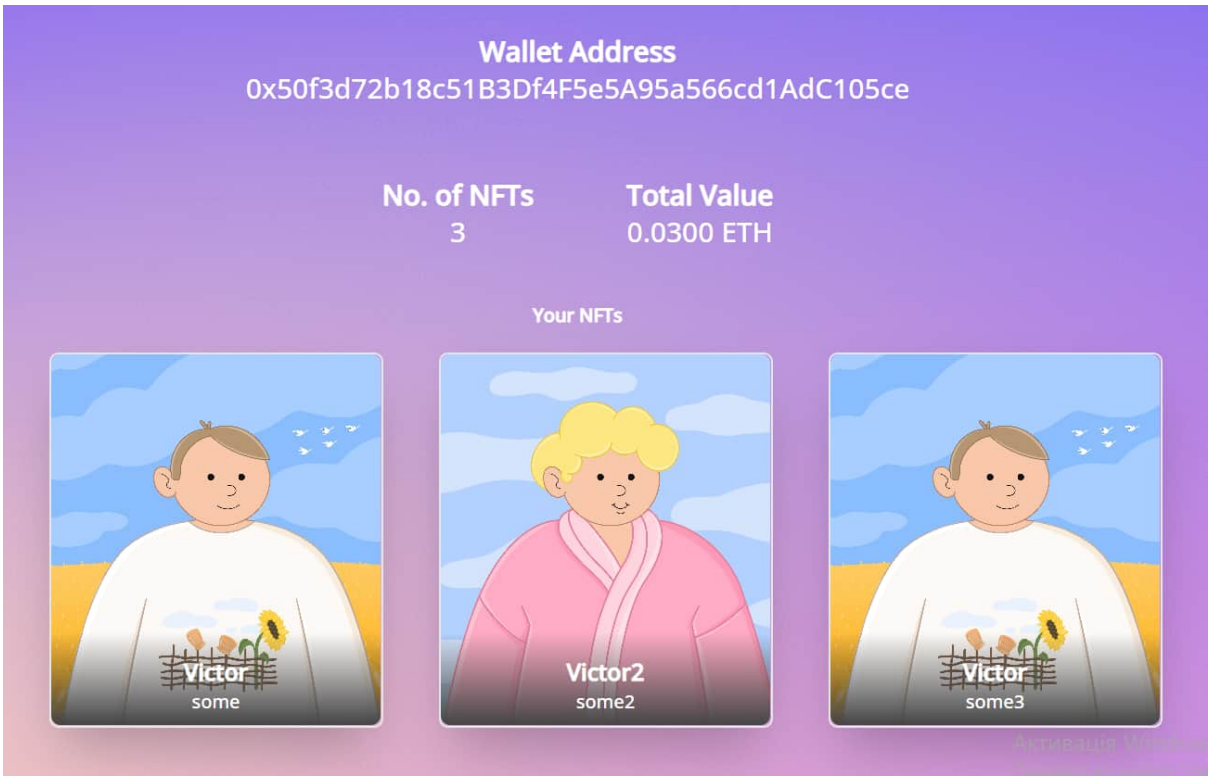

Рис 2.17 - перегляд балансу власних NFT

Перевіримо наявність токену з іншого гаманця.

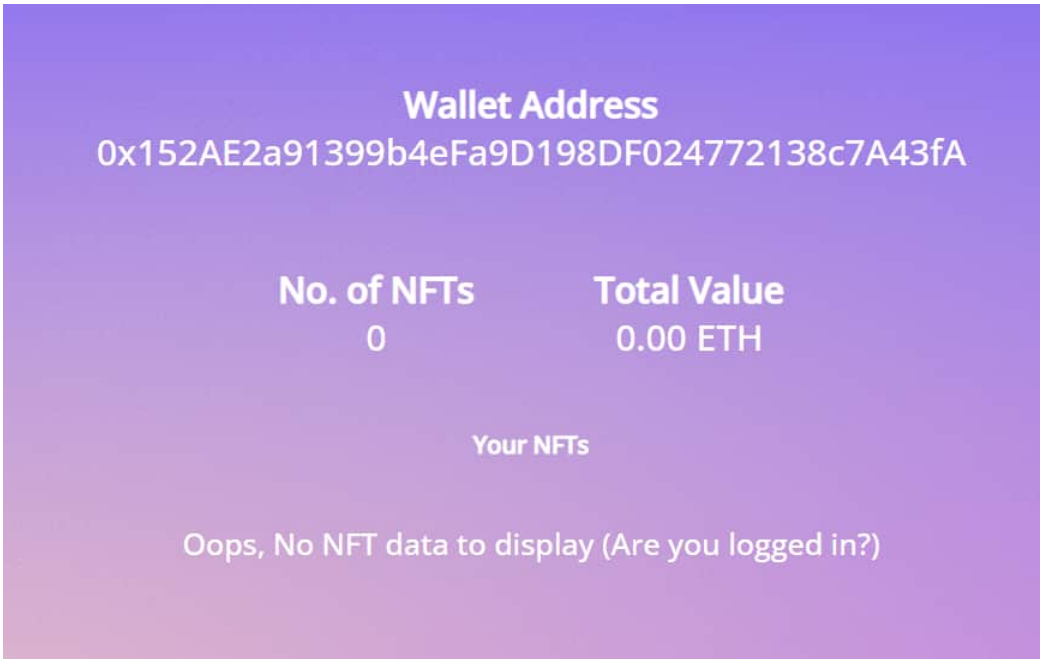

Рис 2.18 - перегляд балансу власних NFT з іншого гаманця.

Як бачимо купівля-продаж пройшли успішно.

РОЗДІЛ 3. Додаткові дослідження

3.1 Тестування правильності транзакцій у мережі Etherscan

Всі виконані операції досить зручно перевірити на платфомі Etherscan. Цей сервіс дає змогу не тільки побачити виконані дії у блокчейні, а й перевірити деталі про них.
Давайте розберемося з останньою операцією купівлі-продажу нашої NFT.

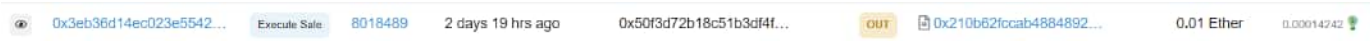

Рис 3.1 - стрічка останньої транзакції

У верхньому рядку можемо побачити деякі деталі про транзакцію. Наприклад, як номер операції, дата виконання, адресат, отримувач, сума переказу.

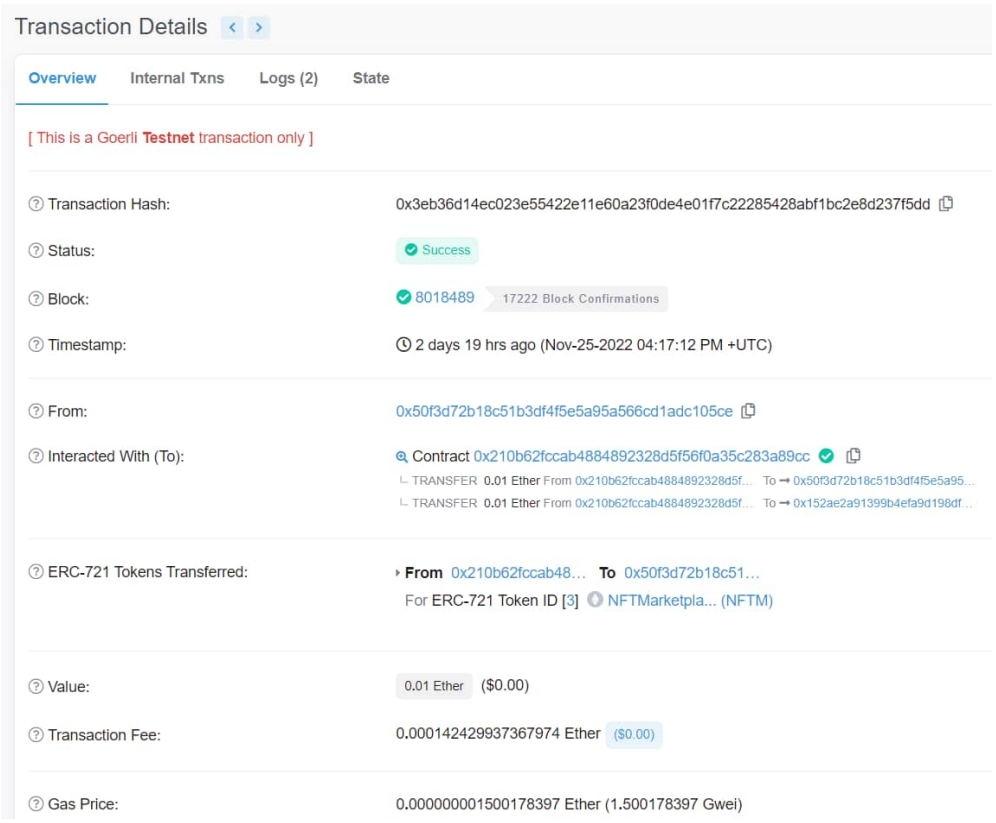

Рис 3.2 - повний огляд транзакції

Тут бачимо більше детальну інформацію. Наприклад, як вартість газу за проведення транзакції, Ефіріум адреси відправника - отримувача, номер блоку, статус транзакції. Крім цього, можемо також переглянути більше деталей про внутрішні транзакції, логи, стан гаманців учасників до та після цієї операції.

3.2 Взаємодія розробленого маркетплейсу з OpenSea

Крім використання нашого створеного маркетплейсу, ми можемо використовувати інтерфейс OpenSea, щоб переконатися у правильності відтворення наших сторінок, а також продажу NFT у середовищі з більшою аудиторією. OpenSea, як агрегатор всіх NFT на мережі блокчейну Ethereum та декількох інших, дозволяє без проблем використовувати токени і на власній платформі.

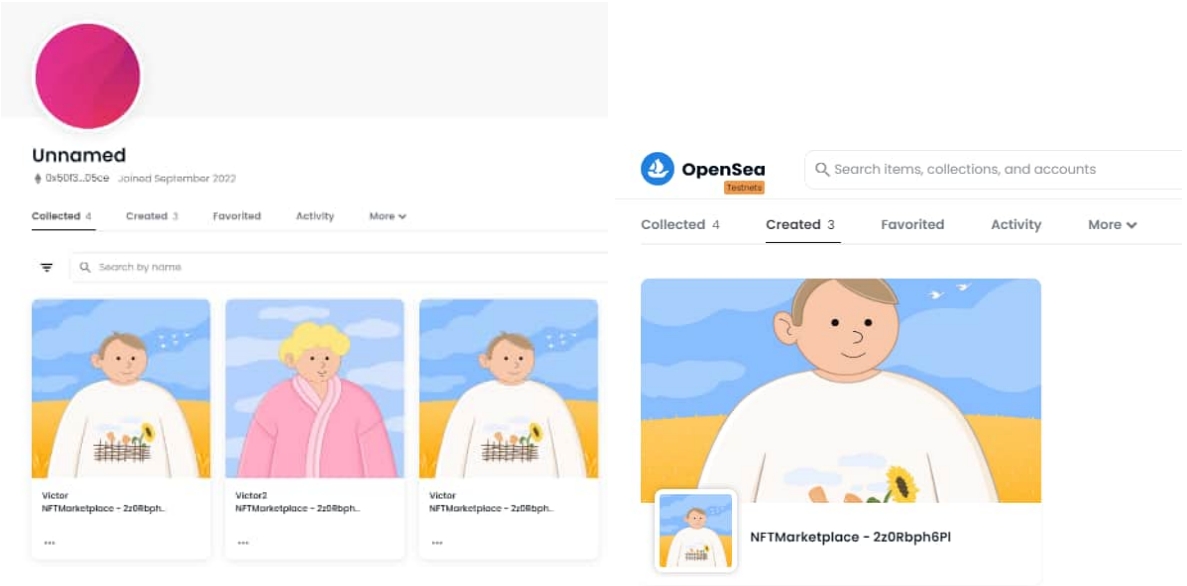

Рис 3.3 - перегляд балансу власних NFT з OpenSea.

Як бачимо, наші NFT є відтворені на маркетплейсі. Крім цього, OS показує навіть звідки ці токени були створені.

Висновок

Світ не стоїть на місці і багато подій провокують розвиток нової культури та технологій. У період епідемії всі виставки, галереї та мистецькі центри призупинили своє існування. Проте будучи в тимчасовій ізоляції, ентузіасти не припинили створення нових жанрів. Таким чином, поєднання популярної крипто-індустрії зі всіма прикладними технологіями задало новий ритм в якому виникло багато технологічно-мистецьких проектів. Серед них CryptoPunks, Bored Ape Yacht Club, CloneX та інші. Всі проекти об’єднює одна технологія розробки та платформа на якій публікуються токени цих проектів. Маркетплейс необхідна складова NFT середовища.

Тому в цій роботі був розроблений власний маркетплейс зі всіма потрібними операціями. Варто зауважити, що крім блокчейну Ethereum існує багато інших хороших альтернатив. Вибраний мною блокчейн часто критикують за велику комісію оплати транзакції, тому часто NFT проекти обирають такі мережі як Solana та Binance Smart Chain. Реалізований смарт-контракт можна розвинути та інтегрувати в більше середовище.

Зокрема, пророблену роботу можна розвинути до централізованого маркетплейсу з додатковими функціями списку транзакцій, панелі проектів та окремої секції промоушину NFT творців.

Використанні джерела

1. Douglas Crockford. "In JavaScript: The Good Parts" , pp. 286–304, 2008.
2. Jon Duckett. "JavaScript and jQuery: Interactive Front-End Development", pp. 286–304, 2014.
3. Andreas M. Antonopoulos and Gavin Wood . "Mastering Ethereum: Building Smart Contracts and DApps", pp. 105–183, 2018.
4. Rangel Stoilov, "Solidity Smart Contracts: Build DApps In Ethereum Blockchain", pp. 45–108, 2019.
5. Jitendra Chittoda. "Mastering Blockchain Programming with Solidity: Write Production-ready Smart Contracts for Ethereum Blockchain with Solidity", pp. 256–328, 2019.
6. Randall Kanna. "Hands-On Smart Contract Development with Solidity and Ethereum: From Fundamentals to Deployment", pp. 65–86, 2019.
7. Vitalik Buterin. "Proof of Stake: The Making of Ethereum and the Philosophy of Blockchains", pp. 124–148, 2022.
8. Michael G. Solomon. "Ethereum For Dummies", pp. 25–156, 2019.
9. Chris Collins, "NFT Art and Collectables for Beginners: The Must Have Guide for Understanding Non Fungible Tokens (NFTs) Paperback", pp. 38–59, 2021.