НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.023

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2022 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційні управляючі системи та технології»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система підтримки процесу аналізу і відбору резюме кандидатів за допомогою методів обробки текстів природною мовою»

Виконала:
студентка VI курсу, групи ІС-12мп
Коноплянка Дмитро Сергійович _____

Керівник:
доц., к.т.н., доц.,
Сперкач Майя Олегівна _____

Рецензент:
проф. каф. ОТ, д.т.н., проф.
Клименко Ірина Анатоліївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент_____

Київ – 2022 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___»_____2022 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Коноплянці Дмитру Сергійовичу

1. Тема дисертації «Інформаційна система з фракталізації NFT», науковий керівник дисертації Сперкач Майя Олегівна, доц., к.т.н., доц., затверджені наказом по університету від «09» 11 2022 р. № 4115-с

2. Термін подання студентом дисертації: 14.12.2022 р.

3. Об'єкт дослідження: фракталізація NFT.

4. Вихідні дані: розгорнуті смарт-контракти для фракталізації NFT.

5. Перелік завдань, які потрібно розробити: виконати огляд загальних положень web 3; виконанти порівняльний огляд популярних блокчейнів та вибрати на якому будуть розгорнуті смарт-контракти.розробити алгоритм аукціону з можливістю викупу токенів; описати та розробити смарт-контракти для фракталізації NFT; створити програмну реалізацію веб-застосунку для взаємодії зі смарт-контрактами.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: схема структурна взаємодії смарт-контрактів,, схема структурна послідовності виконаня, схема структурна децентралізованої бази даних, схема структурна створення смарт-контракту, структурна схема архітектури фреймворку nuxt, структурна схема виконання Ethereum virtual machine code у EVM, структурна схема виконання EVM, структурна схема архітектури EVM.

7. Орієнтовний перелік публікацій публікації відсутні

8. Консультанти розділів дисертації

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

9. Дата видачі завдання "10" вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Виконання огляду існуючих методів розв'язання даної задачі	20.09	Виконано
2	Виконати огляд загальних положень та концепцій web 3	03.10	Виконано
3	Розробка алгоритму аукціону з можливістю викупу токенів та продумування архітектури смарт-контрактів	13.10	Виконано
4	Розробка смарт-контрактів для фракталізації NFT	28.10	Виконано
5	Розробка веб-застосування для взаємодії зі смарт-контрактами	05.11	Виконано
6	Оформлення документації	20.11	Виконано
7	Подання роботи на попередній захист	22.11	Виконано
8	Подання роботи на основний захист	11.12	Виконано

Студент Дмитро КОНОПЛЯНКА

Науковий керівник Майя СПЕРКАЧ

РЕФЕРАТ

Магістерська дисертація: 100 с., 20 рис., 34 табл., 8 додатків, 40 джерел.

Актуальність. NFT (Non-fungible token) – це токен, який підтверджує право на власність унікальних предметів. За допомогою NFT можливо токенізувати мистецтво, колекційні речі й інші речі, які мають унікальні властивості. Протягом останніх років тема NFT стала популярною і як наслідок – унікальні речі, а особливо лімітовані колекції NFT значно виросли у ціні. Наприклад, ціна одного NFT з колекції “Bored Ape Yacht Club” [1] коштує 76 ETH, що дорівнює 123 706 американських доларів. Через зростання вартості, поріг входу в NFT збільшився, з чого випливає, що лише невелика кількість людей зможе придбати NFT. Також через зростання вартості в мережах блокчейну може слідувати брак ліквідності, що дуже погано для цілої мережі.

За таких високих вхідних бар'єрів фракціонування NFT є потенційним вирішенням виявлених проблем. Розбивка NFT на фракції демократизує цей новий ринок, дозволяючи зацікавленим сторонам з обмеженими коштами інвестувати за доступними цінами. Це приносить користь як інвесторам, так й NFT загалом, оскільки, приносить ліквідність ринку. Фракталізація NFT надають ринку багато доступних токенів, які пропонують відсоткове володіння популярними та унікальними NFT.

Мета дослідження – спрощення процесу придбання токенів за рахунок їх фракталізації згідно реалізованих смарт-контрактів.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- виконати огляд концепції Web 3 та ознайомитися з останніми розробками та дослідженнями на базі блокчейну;
- провести аналіз та обрати блокчейн з швидко працюючим консенсусом та не дорогими комісіями за його використання;
- розробити алгоритм проведення аукціону з можливістю викупу токенів;
- розробити смарт-контракти для фракталізації NFT;

- розробити програмну реалізацію системи для здійсненняв взаємодії зі смарт-контрактами фракталізації NFT;
- провести аналіз отриманих результатів.

Об’єкт дослідження – фракталізації NFT.

Предмет дослідження – Інформаційна система з фракталізації NFT.

БЛОКЧЕЙН, NFT, PROOF-OF-STAKE, NFT ФРАКТАЛІЗАЦІЯ, РОЗДРІБНЕНЕ ВОЛОДІННЯ NFT.

ABSTRACT

Master's dissertation: 100 p., 20 figures, 34 tables, 8 supplements, 40 sources.

Relevance**.** NFT (Non-fungible token) is a token that confirms ownership of unique items. NFTs can be used to tokenize art, collectibles, and other items that have unique properties. In recent years, the NFT theme has become popular and as a result, unique items, and especially limited edition NFT collections, have increased in value significantly. For example, the price of one NFT from the collection “Bored Ape Yacht Club” [1] costs 76 ETH, which is equal to 123,706 US dollars. Due to the increase in cost, the entry threshold for NFTs has increased, which means that only a small number of people will be able to purchase NFTs. Also, due to the increase in value in blockchain networks, a lack of liquidity can follow, which is very bad for the entire network.

With such high entry barriers, NFT fractionation is a potential solution to the identified problems. Fractionation of NFTs democratizes this new market, allowing stakeholders with limited funds to invest at affordable prices. This benefits both investors and NFTs in general, as it brings liquidity to the market. Fractalization of NFTs provides the market with many affordable tokens that offer percentage ownership of popular and unique NFTs.

Relationship of work with scientific programs, plans, topics. The work was performed at the Department of Information Systems and Technologies of the National Technical University of Ukraine "Kyiv Polytechnic Institute named after Igor Sikorskyi" within the framework of the topic "Information system for NFT fractalization".

The purpose of the research – simplification of the process of acquiring tokens due to their fractalization according to implemented smart contracts.

To achieve the goal, the following tasks must be completed:

- Review the concept of Web 3 and familiarize yourself with the latest blockchain-based developments and research;
- conduct an analysis and choose a blockchain with a fast-working consensus and inexpensive fees for its use;
- develop an algorithm for holding an auction with the possibility of redeeming tokens;

- develop smart contracts for NFT fractalization;
- develop a software implementation of the system for interaction with NFT fractalization smart contracts;
- analyze the obtained results.

The object of research – NFT fractalization.

The subject of the study – the Information System on Fractalization of NFT.

BLOCKCHAIN, NFT, PROOF-OF-STAKE, NFT FRACTALIZATION, FRACTIONAL OWNING NFT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

NFT	– Non-fungible token
ERC-20	– Ethereum Request for Comments, стандарт для смарт-контракту взаємозамінних токенів на блокчейні Ethereum
ERC-721	– Ethereum Request for Comments, стандарт для смарт-контракту невзаємозамінних токенів (NFT) на блокчейні Ethereum
ABI	– Application Binary Interface
ETH	– Криптовалюта Ефір
DAO	– Цифрова децентралізована автономна організація
EVM	– Ethereum Virtual Machine
EOA	– Externally-owned account
WEI	– найменша і неподільна частка ETH
Смарт-контракт	– програма в глобальному стані EVM
SFC	– Single file component
ABI	– Application Binary Interface
AMM	– Automated Market Maker
API	– Application Programming Interface

ВСТУП

Протягом останніх років тема NFT стала популярною і як наслідок – унікальні токени, а особливо лімітовані колекції NFT значно зросли у ціні та стали важкодоступними для широкого обсягу людей.

Зростання ціни NFT значно збільшило поріг в ходу з чого випливає, що менша кількість людей зможе його собі дозволити. В звичайній економіці декілька людей одночасно можуть володіти річчю та бути співвласниками один одному.

Фракталізація NFT надає можливість володіти унікальним асетом зразу декільком людям тим самим вирішуючи проблему ліквідності, проблему високих цін та надає можливістю бути співвласником унікального асету. Фракталізація NFT демократизує ринок дозволяючи більшому колу людей стати співласниками їх NFT та приєднатися до світу NFT в цілому.

Нещодавно в Австрії шедевр Густава Клімта початку 1900-х років, «Der Kuss», був розділений на 10 000 цифрових фрагментів і проданий у Віденському музеї [2]. Гарний кейс від музею, вони фракціонували картинку яка коштує мільйони доларів на 10 000 фракцій та дали змогу заволодіти частинкою більш широкій аудиторії шанувальників мистецтва. Ще одним прикладом може слугувати популярність фракційних NFT в метавсесвіті де нещодавно в метавсесвітній грі The Sandbox була придбана «ділянка землі» поруч з «особняком» репера Snoop Dog за 450 000 доларів [3].

Але фракціонування NFT має низку ризиків, один з яких це прирівнювання фракційних NFT комісією CED [4] (Securities and Exchange Commission) до цінних паперів та одна з найголовніших проблем фракційних NFT – реконструкція до вихідного NFT.

В даній дисертації розглядається NFT фракталізація на базі смарт смарт-контракту ERC-20 та використання функцій контракта ERC-721 з алгоритмом аукціону викупу за допомогою якого можна буде реконструювати фракціонне NFT в звичайне. Аукціон викупу схожий на звичайний аукціон за винятком того, що торги йдуть фракціонними NFT замість фізічних активів.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ КОНЦЕПЦІЇ WEB 3

1.1 Огляд концепції Web 3

Централізація допомогла мільярдам людей долучитися до мережі Інтернет та створила стабільну, легкодоступну та надійну інфраструктуру. Але найбільші гіганти які встигли укріпитися у великих просторах всесвітньої павутини в односторонньому порядку почали вирішувати, що можна робити, а що ні. Компанії також активно збирають особисті дані користувачів, що згодом може призвести до спаму чи навіть витоку даних компанії та попадання їх до зловмисників. Web 2 вимагає занадто багато довіри, користувачі мають вірити компаніям та сервісам, що вони безпечні та будуть керуватися в інтересах користувачів.

Така система вимагає занадто багато довіри, користувачі мають довіряти компаніям та сервісам, що вони безпечні та будуть керуватися в інтересах користувачів.

Основні проблеми класичної інтернет-інфраструктури (Web 2):

- необхідність довіряти, що компанії які зберігають дані користувачів, роблять це у відповідності до усіх правил і практик безпеки даних;
- необхідність довіряти, що ці компанії не використовують дані користувачів для власної вигоди чи передають третім особам;
- необхідність слідувати будь-яким правилам, встановленим компаніями-монополістами, не маючи можливості впливу на них;
- небезпека в будь-який момент втратити доступ до інтернет-сервісів, а отже і будь-яких даних чи активів, які там розміщенні, без попередження і відшкодування збитків, оскільки вся влада над даними користувачів знаходиться лише у руках корпорацій, що володіють цими інтернет-сервісами;
- повний контроль корпорацій-володарів сервісів над своїми ресурсами і відсутність у користувачів можливості оскаржити накладені на їх дані чи активи санкції чи звернутись до системи арбітражу для вирішення претензій і скарг.

Оскільки часто провайдерами інтернет-сервісів моделі Web 2 є надвеликі компанії-монополісти, що одноосібно володіють значною частиною популярних інтернет-сервісів, накладання санкцій від таких корпорацій, навіть помилкове, може

призвести до втрати користувачем значної частини можливостей онлайн, які часто приймаються як даність, таких як реєстрація доменних імен, захист сайтів від хакерських атак, доступ до соціальних медіа, що виступають публічним громадським майданчиком для поширення своїх ідей, чи навіть доступ до онлайн-банкінгу та можливості здійснювати покупки і продажі онлайн. Таким чином, у класичній моделі інтернет-сервісів користувач може повністю втратити власний бізнес та дохід всього лиш через помилку алгоритмів корпорацій, що відмітять його акаунт як підозрілий без можливості оскарження такого рішення.

Web 3 є відповіддю на ці проблеми. Замість Інтернету, монополізованого великими технологічними компаніями, Web3 використовує децентралізацію, будується, управляється та належить користувачам. Web3 передає владу в руки окремих осіб, а не корпорацій.

Web 3 став загальним терміном яким описує бачення нового, кращого, не вимагаючого довіри до Інтернету. У своїй основі Web 3 використовує блокчейни [5], криптовалюти [6] та NFT, для того, щоб остаточна влада належала користувачам.

Відмінність між Web 1, Web 2 та Web 3 полягає в наступному [7]:

Web 1 – лише для читання;

Web 2 – читання та запис;

Web 3 – власність, читання, запис.

Складно чітко охарактеризувати Web 3 але його створення керується кількома основними принципами.

Основні ідеї Web 3 полягають в наступному:

- Web 3 децентралізований;
- користувачі мають рівний доступ до Web 3;
- нативні платежі – криптовалюти [6];
- Web 3 не потребує довіри ні до кого, все що було зроблено в Web 3 можна перевірити власноруч та не покладатися на інших.

1.2 Основні принципи створення Web 3

1.2.1 Децентралізація Web 3

Децентралізація Web 3 досягається за рахунок блокчейну [5].

Блокчейн – це публічна база даних, яка оновлюється та використовується багатьма комп'ютерами в мережі. Головною відмінністю блокчейну від звичайної бази даних є структура даних. В той час як звичайна база даних даних складається з таблиць, блокчейн зберігає дані у блоках, пов'язаних між собою і поєдниних у єдиний ланцюжок.

«Блок» – це дані і стан, які зберігаються в послідовних групах. Даними слугують транзакції, які були здійснені в блокчейні. Наприклад, передати права свого NFT іншому власнику чи перевести ETH з одного гаманця на інший.

«Чейн» – ланцюжок, означає те, що всі блок пов'язані між собою криптографічно, а саме кожен наступний блок посилається на попередній блок. Таким чином, блоки з'єднуться разом та не можуть змінитися без зміни всіх наступних блоків.

«Вузол» – це комп'ютер користувача, який підключений до блокчейну. Саме між вузлами і розповсюджуються поточний стан публічної бази даних, які оновлюють його після додавання кожного нового блоку та верифікують, що всі вузли мають однаковий стан бази даних.

Така структура даних за своєю природою гарантує незворотню часову шкалу для даних, де після заповнення певного блока він стає невід'ємною і незмінною частиною ланцюжка.

Найбільш відома сфера застосування технології блокчейну – системи криптовалют, такі як Bitcoin і Ethereum, де вони забезпечують точність і безпеку запису даних, а також гарантують довіру між учасниками валютного ринку без необхідності третьої довіреної сторони.

На рисунку 1.1 наведено топ-15 найбільших криптовалют за капіталізацією у доларах США станом на початок 2022 року.

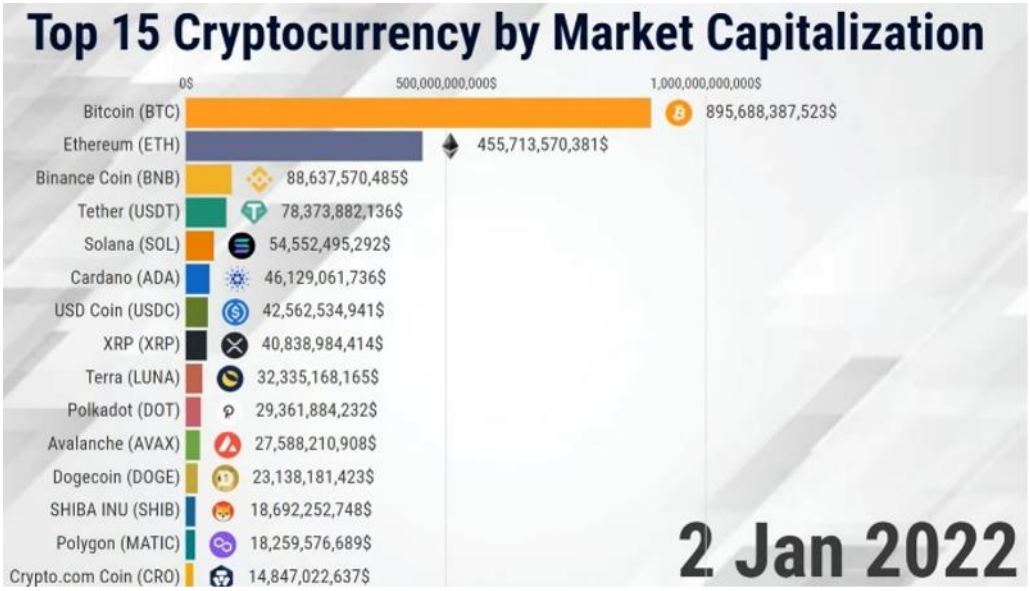

Рисунок 1.1 – Найбільші криптовалюти станом на початок 2022 року

Ринок криптовалют залишається достатньо волатильним, і значна кількість валют набирають значну популярність і капіталізацію незадовго після створення, потрапляють в топи на графіках, проте щвидко втрачають популярність і зникають з цих списків. Все ж певні криптовалюти стабільно потраплять у списки найбільших протягом довгого часу, а отже варто зосередити увагу саме на них і розглянути їх особливості.

Стабільно найпопулярнішими серед великих блокчейнів зараз можна назвати Bitcoin, Ethreum, Polygon та Solana.

Створений у 2008 році анонімним розробником під псевдонімом «Сатоші Накамото», Біткоїн є найпопулярнішою і найдорожчою серед усіх криптовалют. Оригінальною ідеєю Біткоїну було створення системи для переказу грошей між людьми без задіяння посередників за правилами web 3, проте згодом стало зрозуміло, що він добре підходить для використання у якості цифрової валюти для здійснення платежів та зберігання заощаджень.

Біткоїн працює на основі системи блокчейну з консенсусом proof-of-wok , яка базується на математичних алгоритмах, які гарантують, що сторона яка хоче додати блок до ланцюжку блокчейну, має «довести», що на додавання нового блоку було витрачено значні обчислювальні ресурси, при цьому у інших має бути можливість це перевірити, використавши незначну кількість обчислювальних ресурсів. Таким чином уникається небезпека маніпулювання даними блока з боку шахраїв.

У системі біткоїну присутній алгоритм, що автоматично змінює складність вирішення математичної задачі таким чином, щоб новий блок генерувався приблизно кожні 10 хвилин, незважаючи на те, яку загальну обчислювальну потужність виділено на вирішення цих задач.

Крім того, нагорода за додавання нового блоку зменшується вдвічі кожні 210_000 блоків. Отже, враховуючи, що єдиний спосіб створення нових біткоїнів - такі нагороди, а наразі створено більше 19млн біткоїнів, близько 2140 року буде досягнено абсолютний ліміт їх кількості – 21млн.

Біткоїн підтримує виконання смарт-контрактів, використовуючи для цього власну скриптову мову програмування Script. Проте, для запобіганню Denial-of-Service атакам, ця мова умисно була створена не Тьюринг-повною, без підтримки циклів. Отже, незважаючи на своє домінування у сфері криптовалюти, Біткоїн не є найкращим кандидатом для розробки смарт-контрактів через відносну обмеженість та недосконаліть інструментів для їх розробки та виконання.

Ethereum є одним із найкращих криптоактивів у криптоекосистемі. З ринковою капіталізацією в $186 101 941 070.28 Ethereum вважається одним із найрізноманітніших блокчейн-протоколів. На ринку цифрових предметів колекціонування Ethereum домінує, оскільки це була перша платформа, яка дала моджливість створювати NFT.

Головні особливості Ethreum:

- Ethereum зробив можливим розвиток смарт-контрактів. Ethreum є піонером концепції смарт-контрактів та створенню NFT;
- він використовує віртуальну машину Ethereum (EVM), яка «розуміє» контракти та дозволяє користувачам взаємодіяти з ними;
- блокчейн Ethereum може розміщувати різноманітні криптографічні токени за допомогою стандарту ERC-20. Іншими варіантами токену є ERC-721, ERC-1155 і ERC-1238;
- дозволяє користувачам створювати демократичні рішення з повною прозорістю, які також відомі як децентралізовані автономні організації (DAO), де немає єдиного лідера, який всім керує;

- дозволяє створювати Layer 2. Layer 2 – це окремий блокчейн, який розширює Ethereum і успадковує гарантії безпеки Ethreum;
- ethreum використовує proof-of-stake в якості механізму консенсусу.

Основним мінусом після переходу блокчейну на консенсус proof-of-stake все ще залишається комісія за транзакцію.

Solana була створена, щоб надати розробникам місце для розробки орієнтованих на користувача програм. Протокол Solana розроблений таким чином, що дозволяє створювати децентралізовані додатки, також відомий як DApp, шляхом розгортання смарт-контрактів в блокчейні. Solana прагне покращити масштабованість, представивши консенсус доказу історії proof-of-history (PoH) у поєднанні з базовим консенсусом proof-of-stake (PoS) блокчейну.

Головні особливості Solana:

- механізм консенсусу Solana proof-of-history, який дозволяє кожному вузлу мати власний годинник і приймати рішення, не консультуючись один з одним;
- використовує систему Gulf stream для очікування транзакцій у повній пам’яті до тих пор, поки не розпочнеться їхня черга на обробку. Він вміщує до 1 000 транзакцій одночасно;
- він може вмикати кілька смарт-контрактів одночасно;
- алгоритм Byzantine Fault Tolerance (BFT) Solana гарантує, що збій конкретного вузла не впливає на роботу всієї системи.

Головними мінусами для блокчейну Solana це апаратна підтримка, яка є дорожчою, ніж у більшості та має проблеми з децентралізацією.

Polygon це рішення (Layer 2) для масштабування Ethereum та розвитку інфраструктури. Layer 2 знімає транзакційне навантаження з Layer 1 і публікує завершені докази назад на Layer 1. Завдяки видаленню цього транзакційного навантаження з Layer 1 базовий рівень стає менш перевантаженим. Проект має на меті створити «Інтернет блокчейнів Ethereum», вирішуючи проблеми масштабованості існуючих блокчейнів, і пропонує розробникам набір інструментів для створення надмасштабованих і високопродуктивних блокчейнів і децентралізованих програм (DApps). Наразі

Головні особливості Polygon:

- Layer 2 для масштабування та розвитку інфраструктури Ethereum.
- повністю сумісний з Ethereum
- масштабування, яке підтримує віртуальну машину Ethereum (EVM).

Оскільки, Polygon є Layer 2 рішенням то коли Ethereum вирішить всі свої проблеми Polygon може втратити свою необхідність.

Далі буде наведена таблиця 1.1 з порівняннями описаних вище блокчейнів.

Таблиця 1.1 – Порівняння блокчейнів Ethereum, Solana та Polygon

Блокчейн	Ethereum	Solana	Polygon
Дата запуску	2013	2017	2017
Власний токен	ETH	SOL	MATIC
Тип токену	Нативний	Альткоїн	ERC-20
Мова програмування для смарт-контракту	Solidity, Vyper	Rust, C, C++	Solidity, Vyper
Метод консенсусу	Proof-of-stake	Proof-of-history та Proof-of-stake	Proof-of-stake та plasma-based sidechain
Архітектура	Stateful architectury	Stateless architecture	Multichain architecture
Швидкість транзакції в секунду	20 000 – 100 000	50 000 – 65 000	65 000
Збоїв в блокчейні	0	5+	0

Зваживши всі особливості, переваги і неділіки розглянутих блокчейнів, смарт-контракти будуть розгорнуті на блокчейні Ethereum який є стабліним, одиним із найпопулярніших блокчейнів та на мою думку має найкращі перспективи серед усіх інших блокчейнів.

Ethereum [8] – це блокчейн із вбудованим у нього комп’ютером. Це основа для створення застосунків і організацій (DAO) у децентралізований, без дозволів, стійкий до цензури спосіб.

У всесвіті Ethereum існує єдиний канонічний комп’ютер EVM стан якого узгоджений для всіх учасників мережі Ethereum. Кожен, хто бере участь у мережі Ethereum (кожен вузол Ethereum), зберігає копію стану цього комп’ютера. Крім того, будь-який учасник може транслювати запит для цього комп’ютера на виконання довільних обчислень. Кожного разу, коли такий запит транслюється, інші учасники мережі перевіряють, підтверджують і виконують обчислення. Це виконання викликає зміну стану в EVM, яка фіксується та поширюється між усіма вузлами мережі.

Запити на обчислення називаються транзакціями; записи про всі транзакції та поточний стан EVM зберігаються в блоці та потім додаються до блокчейну, який, у свою чергу, зберігається та погоджується всіма вузлами.

Криптографічні механізми гарантують, що коли транзакції перевіряються як дійсні та додаються до блокчейну, вони не можуть бути змінені пізніше. Якщо це навіть станеться, то всі наступні блоки перестануть бути вірними. Ці ж механізми також гарантують, що всі транзакції підписуються приватним ключем користувача, який має тільки він.

1.2.2 Рівний доступ до Web 3

У системі блокчейну відсутній будь-який централізований ресурс чи сервіс, що займався би зберіганням даних, обробкою транзакцій, арбітражем чи будь-чим іншим. Навпаки, всі, будь-то користувачі або вузли, мають рівний доступ до блокчейну і даних про всі виконані на ньому транзакції. Це означає, що будь-яка зацікавлена сторона може завантажити усю базу даних блокчейну на власні пристрої, переглядати історію усіх транзакцій, що у ньому записані.

Також слід зауважити, що навіть творці не мають змоги обмежити користувачів в доступі до блокчейну. А отже, система блокчейну є максимально відкритою і прозорою, що якомога унеможливлює махінації з боку недоброчесних гравців, що мають привілейований доступ до системи.

1.2.3 Нативні платежі

Нативними платежами є криптовалюта, за допомогою якої можна робити транзакції до інших користувачів чи смарт-конрактів [9].

Є різні види криптовалют. Найпопулярніші з них це ETH, Solana та Matic.

Оскільки, було вибрано блокчейн Ethereum буде використовуватися рідна криптовалюта ETH. Розглянемо більш детально криптовалюту ETH, як приклад криптовалюти в блокчейні.

Ефір (ETH) [8] – нативна криптовалюта Ethereum. Метою ETH є створення ринку для обчислень. Такий ринок забезпечує економічний стимул для учасників, перевіряти та виконувати запити на транзакції та надавати обчислювальні ресурси.

Також, слід розуміти, що будь-який учасник, який транслює запит на транзакцію, також повинен сплатити певну суму ETH мережі як комісію за обчислювальні витрати (перевірку транзакції). Мережа присудить частину комісії тому, хто врешті-решт виконає роботу з перевірки транзакції, та додасть її в блок, який потім вже буде закріплений в блокчейні.

Комісія сплаченого ETH відповідає ресурсам, необхідним для виконання обчислень. Комісія слугує захистом від зловмисників, які могли б «засмітити» мережу та справедливою винагородою за обчислювальні витрати від іншого вузла. Чим більш ресурсозатратною буде транзакція тим більше буде комісія.

ETH також використовується для забезпечення криптоекономічної безпеки мережі трьома основними способами:

1) ETH використовується як засіб для винагороди вузлів, які перевіряють транзакції, формують блок та слідкують за чесністю інших вузлів;
2) стає запорукою проти нечесної поведінки – якщо вузли спробують вести себе нечесно з них буде списаний відсоток [10] їх ETH, яку вони заклали для того, щоб бути валідатором, а згодом якщо такі випадки повторюватимуться знищить весь залог вузла.
3) використовується для зважування «голосів» валідаторів за щойно запропонованими блоками надаючи частину вибору алгоритму консенсусу.

1.2.4 Web 3 не потребує довіри

Однією з найважливіших особливостей системи блокчейну є створення «системи без довіри», при використанні якої відсутня необхідність покладатись на довіру до певного централізованого ресурсу, наданого третьою стороною. оскільки замість класичних посередників у транзакціях, таких як банки, валідація здійснюється усіма користувачами блокчейну. Така система має інструменти для досягнення консенсусу між всіма учасниками без необхідності довіряти чи навіть знати один одного. З її допомогою довіра розподіляється між всіма учасниками і потреба в ній мінімізується. Найпопулярнішими алгоритмами для досягнення довіри залишаються алгоритми proof-of-work і proof-of-stake.

Таким чином Web 3 працює, використовуючи стимули та економічні механізми замість того, щоб покладатися на вже існуючі сервіси чи компанії.

Будь-яка транзакція чи адреса на блокчейні Ethereum може бути перевірена будь-яким користувачем власноруч за допомогою сервісу від команди Ethereum Etherscan [10], який в повному обсязі надає інформацію. Ethersan надає обширну інформацію щодо транзакції.

Крім цього, Etherscan служить агрегатором даних про стан блокчейна і токенів. У секції Resources представлені різні метрики та статистика для первинної аналітики: денний графік ціни Ethereum, графік капіталізації, кількість токенів в обігу та інші дані які стосуються блокчейну Ethereum.

У меню Tokens є актуальна інформація про токені стандарту ERC-20, ERC-721 та ERC-1155: ціна, відсоткова зміна вартості, обсяг торгів, капіталізація та загальна кількість адрес, на яких зберігається актив.

Сервіс відображає деталі транзакції в мережі Ethereum, включаючи невдалі або підтвердження.

Далі буде наведений рисунок 2.1 з прикладом перевірки транзакції за допомогою Etherscan.

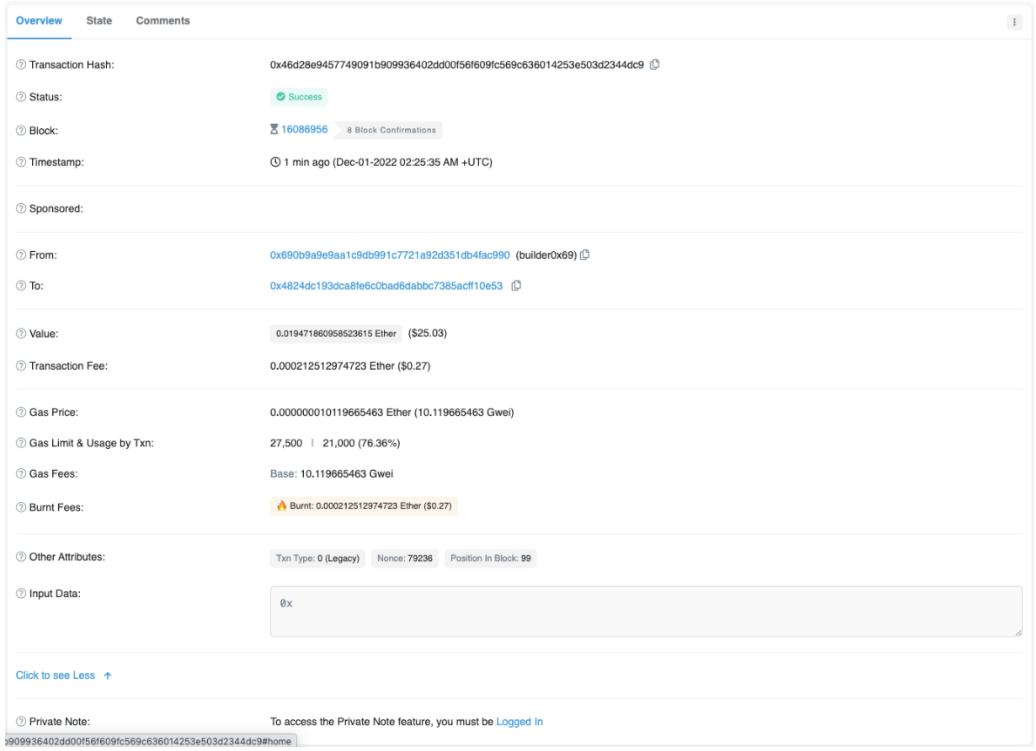

Рисунок 2.1 – Перевірка транзакції за допомогою Etherscan

Також можна перевірити адресу користувача та подивитись, які транзакції він робив та кому, загалом – повна прозорість. Додатково за адресою користувача можна подивитись до яких інших чейнів є ця адреса та перейти на них.

Далі буде наведений рисунок 2.2 з адресою користувача та всіми його зробленними транзакціями.

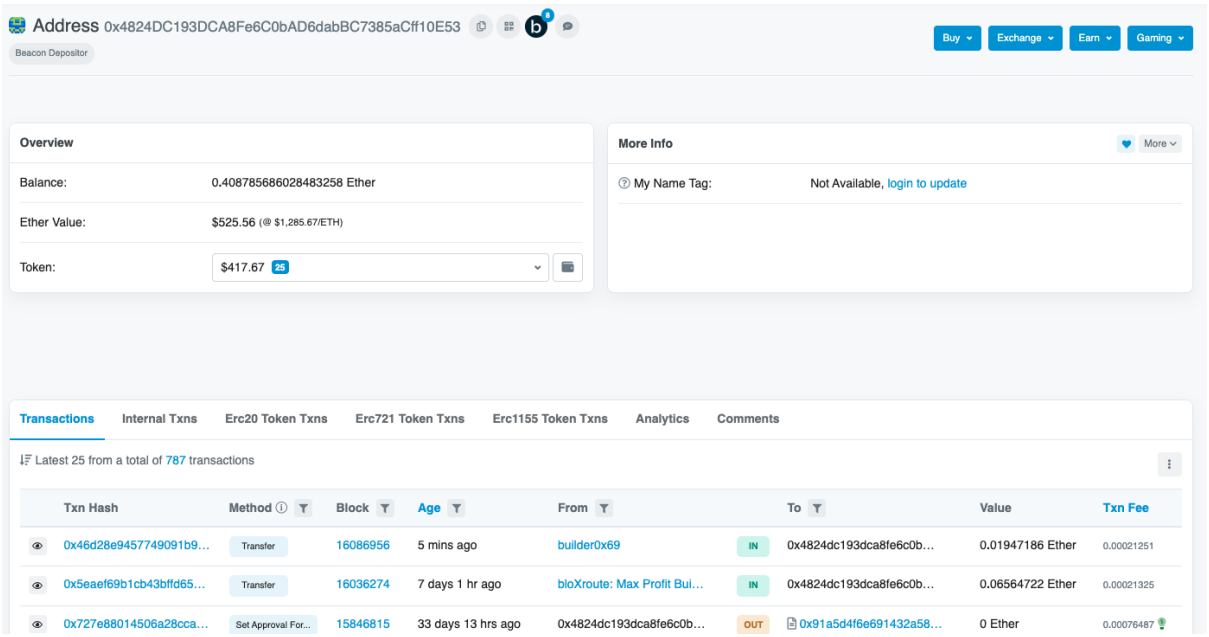

Рисунок 2.2 – Перевірка адреси користувача за допомогою EtherScan

1.3 Обліковий запис Ethereum

Обліковий запис Ethereum – це об’єкт у мережі Ethreum.

Обліковий запис – це зв’язок між адресою та станом облікового запису.

Облікові записи можуть контролюватися користувачами (EOA) або розгортатися як смарт-контракти. Далі буде наведений рисунок 2.3 на якому будуть наведені всі поля які містить аккаунт.

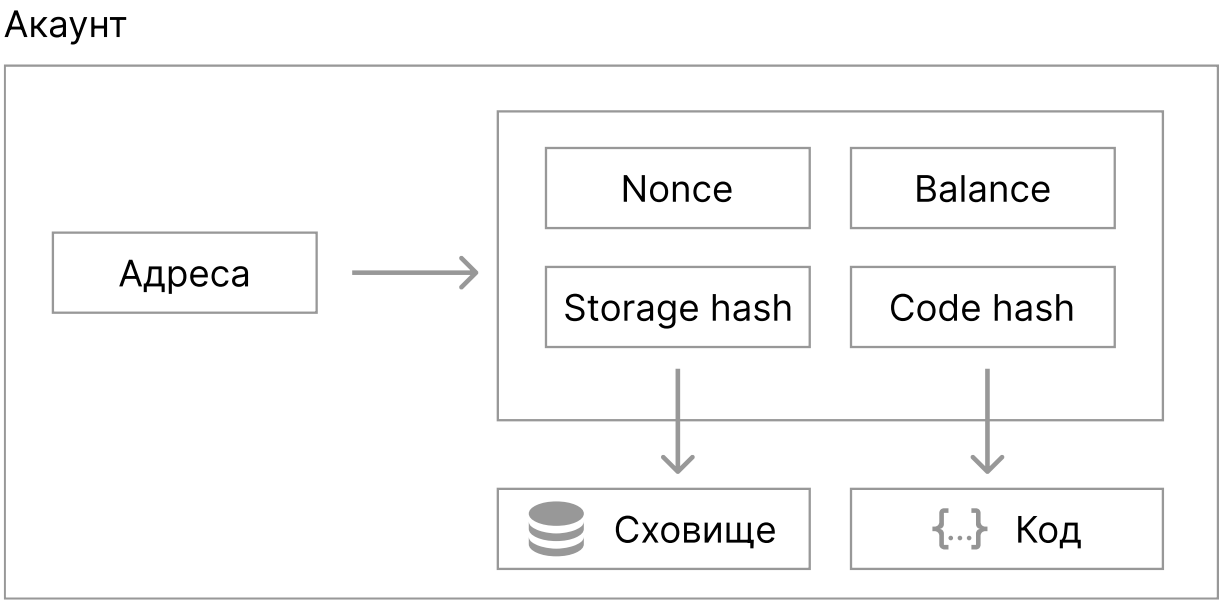

Рисунок 2.3 – Представлення об’єкту аккаунту в глобальному стані EVM

1.3.1 Аккаунт користувача

Аккаунт користувача [11] складається з криптографічної пари ключів: відкритого та закритого з яких потім генерується адреса, нонса та балансу. Далі буде наведений рисунок 2.4 на якому наведені всі поля які доступні зовнішньому користувачу.

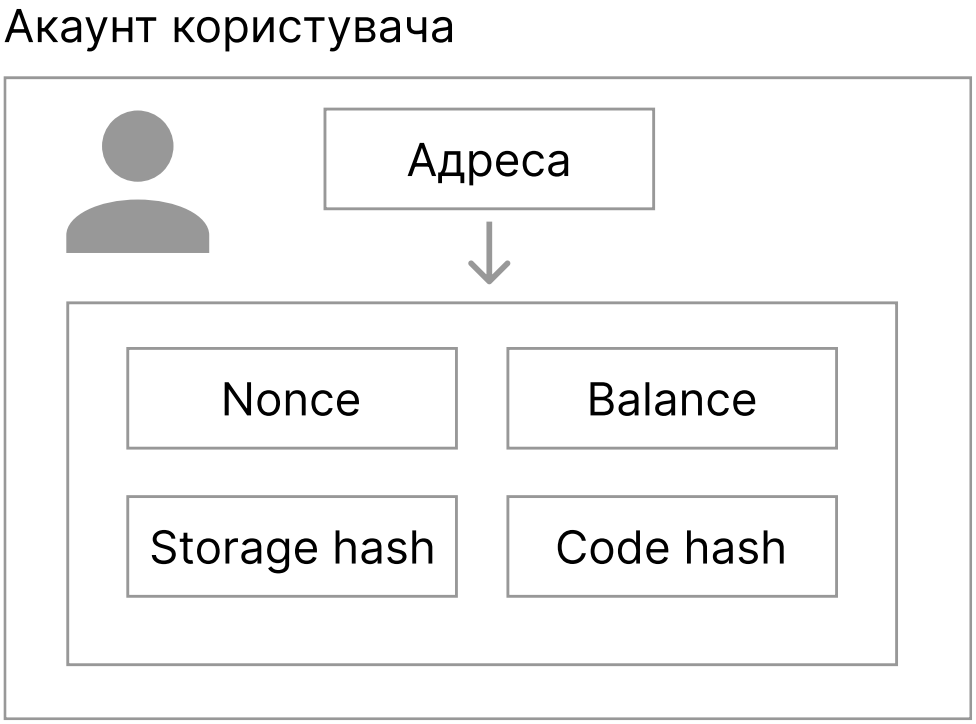

Рисунок 2.4 – Об’єкт акаунта користувача

Аккаунт користувача складається з наступних складових:

Nonce – лічильник який показує кількість транзакцій надісланих з аккаунта;

Balance – кількість wei [12], якими володіє ця адреса;

Adress – хешований приватний та публічні ключі;

Storage hash та code hash пусті для аккаунту користувача.

Приватний ключ складається з 64 шістнадцяткових символів, а публічний ключ генерується на основі приватного ключа за допомогою алгоритма цифрового підпису на базі еліптичних кривих. Ви отримуєте публічну ключ для свого аккаунту, взявши останні 20 байт хешу Keccak-256 відкритого ключа та додавши 0x на початок. Також слід зазначити, що приватний ключ можно згенерувати за допомогою Clef [13] або мнеомонічної фрази [14]. Мнемоніку легше зберігати, бо вона приймає людський вид та за допомогою мнемоніки можна легко згенерувати нові аккаунти, наприклад, додавши каунтер у кінець фрази [15]. Далі буде наведений рисунок 2.5 на якому наведені ключі за допомогою яких генерується адреса.

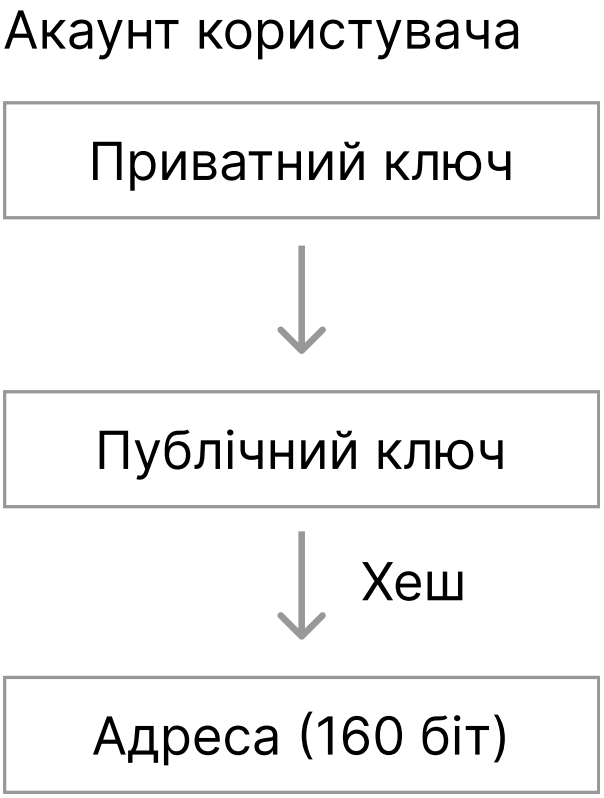

Рисунок 2.5 – Генерація адреси за допомогою приватного та публічного ключа

Ключі допомагають підтвердити, що транзакція дійсно була підписана відправником, і запобігають підробкам.

Приватний ключ безпосередньо використовується для підписання транзакцій. Потім інші користувачі зможуть взяти підпис, щоб отримати відкритий ключ, підтверджуючи автора транзакції. Також слід зазначити, що користувач не володіє ніякою криптовалютою напряму, він володіє лише приватним ключем.

Умови, які відносяться тільки для аккаунта користувача (EOA):

- створення аккаунта користувача безкоштовне;
- може ініціювати транзакції;
- транзакції між аккаунтами користувачів можуть бути лише в ETH або токенах;
- створений за допомогою криптографічної пари ключів.

1.3.2 Аккаунт смарт-контракту

Смарт-контракт – це невелика програма, яка опублікована в глобальний стан EVM, призначена для автономної роботи, що позбавляє необхідності людського

втручання для її виконання, арбітражу чи виконання результатів роботи. Виконати цю програму можуть усі відправивши запит на транзакцію на адресу контракта.

Смарт-контракт не має пари ключів та керується лише логікою, яка була прописана в контракті. Далі буде наведений рисунок 2.6 аккаунту смарт-контракту з усіма доступними йому полями.

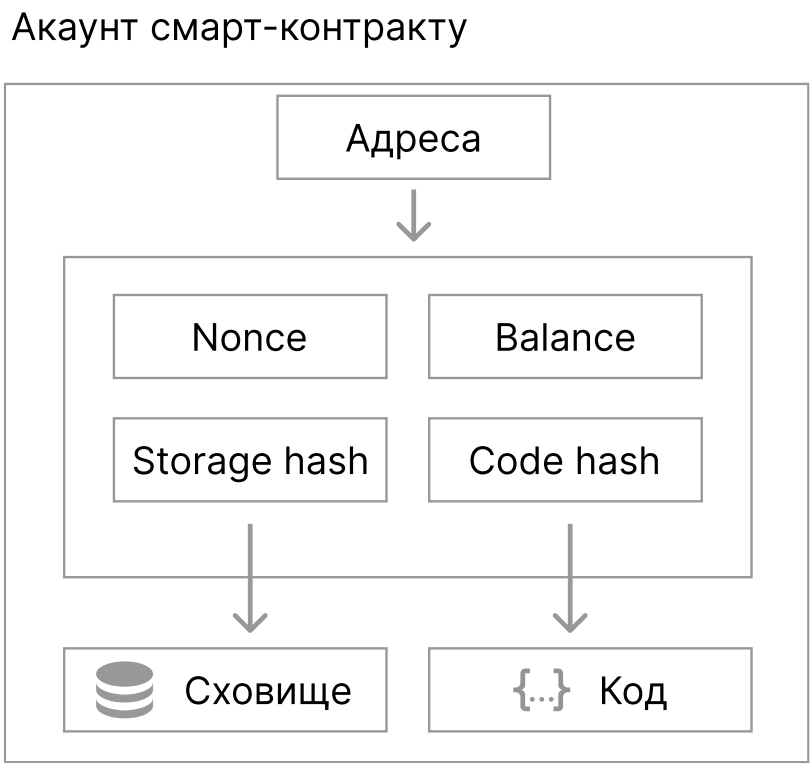

Рисунок 2.6 – Об’єкт аккаунту смарт-контракту

Адреса смарт-контракту зазвичай надається під час розгортання контракту в блокчейні Ethereum. Адреса складається з адреси аккаунту творця та кількості транзакцій, надісланих із цієї адреси («nonce»).

Аккаунт смарт-контракту складається з наступних складових:

Nonce – кількість смарт-контрактів створених аккаунтом;

Balance – такий само як і у аккаунтів користувача;

Code hash – хеш посилання на EVM до коду смарт-контракта;

Storage hash – 256-бітний хеш кореневого вузла trie Merkle Patricia, який кодує вміст сховища аккаунту.

Умови, які відносяться тільки для аккаунта смарт-контракту (EOA):

- створення контракту не безкоштовне і вимагає певних витрат, різниця між аккаунтом користувача в тому, що смарт-контракт використовує мережеве сховище;
- може надсилати транзакції тільки тоді, коли до самого смарт-контракту надійшла транзакція;
- при транзакції від іншого аккаунту смарт-контракт починає виконувати код контракту;
- смарт-контракт керується лише за рахунок коду, який в ньому написаний, цей код неможливо змінити.

1.4 Смарт-контракт

Смарт-контракт [16] – це програма, яка працює на блокчейні Ethereum. Це набір коду (його функції) і даних (його стан), який знаходиться за певною адресою в блокчейні Ethereum.

Смарт-контракти є різновидом аккаунту Ethereum. Вони мають баланс і можуть бути ціллю транзакцій та не контролюються користувачем, натомість вони розгорнуті в мережі та працюють за кодом, який в ньому прописаний.

Облікові записи користувачів можуть взаємодіяти зі смарт-контрактом, надсилаючи транзакції, які виконують функцію, визначену в смарт-контракті. Смарт-контракти не можна видалити за замовчуванням, а взаємодія з ними незворотня.

Смарт-контракти не можуть надсилати HTTP-запити. Це зроблено для того, щоб підвищити безпеку та не порушувати консенсус з децентралізацією. Ще одним обмеженням смарт-контрактів є максимальний розмір контракту, який не повинен перевищувати 24 Кб, інакше у контракта вичерпається весь газ [17]. В даному розуміння поняття газу – це одиниця, яка вимірює кількість обчислювальних зусиль, необхідних для виконання операцій, або простими словами являється комісією за виконання транзакцій.

Будь-які дані смарт-контракту мають бути призначені певному розташуванню: сховище або пам’ять. Змінювати сховище в смарт-контракті дорого, тому потрібно розглянути, де мають зберігатися дані.

Дані які постійно зберігаються в смарт-контракті називаються постійними даними і представлен у вигляді змінних. Ці значення постійно зберігаються в блокчейні. Вам потрібно оголосити тип, щоб контракт міг відстежувати, скільки пам’яті в блокчейні йому потрібно під час компіляції.

Значення, які зберігаються лише протягом часу виконання функції смарт-контракту, називаються змінними пам’яті. Оскільки вони не зберігаються постійно в блокчейні, використовувати їх набагато дешевше. Додатково до змінних, які визначаються у смарт-контракті, є деякі спеціальні глобальні змінні. Вони в основному використовуються для надання інформації про блокчейн або поточну транзакцію.

Функції смарт-контракту можуть отримувати інформацію або встановлювати інформацію у відповідь на вхідні транзакції.

Є два типи викликів функцій у смарт-контрактах:

- Внутрішні, вони не роблять виклик EVM. До внутрішних функцій можна отримати доступ лише зі смарт-контракту;
- зовнішні, роблять виклик EVM. Зовнішні функції є частиною інтерфейсу смарт-контракту, їх можна викликати як за допомогою інших смарт-контрактів або через транзакції користувачів.

У смарт-контракті також є функції геттери які не змінюють стан даних смарт-контракту. Типовим прикладом геттер функції є отримання результату приватної змінної, наприклад, щоб отримати баланс користувача.

Для написанна функцій смарт-контракту потрібно притримуватися наступного:

- Додати змінні параметрів та тип (якщо функція приймає параметри);
- декларація внутрішнього/зовнішнього використання функції;
- декларація pure/view/payable
- повертати значення із функції (якщо потрібно)

У смарт-контракта також є функція конструктора яка виконуються лише один раз під час першого розгортання контракту. Подібно до конструкторів у багатьох мовах програмування на основі класів, ці функції часто ініціалізують змінні стану

їхніми заданими значеннями. Також на відміну від інших мов програмування конструктори в смарт-контракті не мають перегрузок.

До змінних і функцій, які визначаються у своєму смарт-контракті, є деякі спеціальні вбудовані функції, наприклад, така як address.send () за допомогою якої смарт-контракти можуть відправляти ETH на інші акаунти будь то акаунти зовнішнього володіння (акаунт користувача) чи внутрішнього (акаунт смарт-контракту).

Смарт-контракти надзвичайно гнучкі та здатні контролювати великі обсяги даних, водночас керуючи незмінною логікою на основі коду, розгорнутого в блокчейні. Це створило динамічну екосистему децентралізованих програм яким не потрібно довіряти, які надають багато переваг перед застарілими системами. Вони також представляють можливості для зловмисників, які прагнуть отримати прибуток, використовуючи вразливості в смарт-контрактах.

Публічні блокчейни, такі як Ethereum, ще більше ускладнюють проблему захисту смарт-контрактів. Розгорнутий код контракту зазвичай не можна змінити, щоб виправити недоліки безпеки, тоді як активи, викрадені зі смарт-контрактів, надзвичайно важко відстежити, і їх здебільшого неможливо відновити через незмінність.

1.5 NFT

Оскільки, все стає більш цифровим, виникає потреба відтворювати такі властивості фізичних предметів, як дефіцит, унікальність і підтвердження права власності. Не кажучи вже про те, що цифрові елементи часто працюють лише в контексті продукту.

NFT (Non-fungible token) – це токен, який підтверджує право на власність унікальних предметів.

NFT [18] дають можливість призначати або вимагати права власності на будь-які унікальні цифрові дані, які можна відстежувати за допомогою блокчейну Ethereum як публічної книги. NFT карбується з цифрових об’єктів як представлення цифрових або нецифрових активів.

Право власності на NFT керується за допомогою унікального ідентифікатора та метаданих, які жоден інший маркер не може відтворити. NFT створюються за допомогою смарт-контрактів, які призначають право власності та керують передачею NFT.

Коли хтось створює або карбує NFT, він виконує код, що зберігається в смарт-контрактах, які відповідають різним стандартам, таким як ERC-721. Ця інформація додається до блокчейну, де керується NFT.

Процес карбування, починаючи з високого рівня, який складається з наступних етапів:

– створення нового блоку;
– перевірка інформації;
– запис інформації в блокчейн.

Висновок до розділу

Були розглянуті загальні положення Web 3, які поклали базу для багатьох блокчейнів в цілому. Web 3 розширює парадигму Web 2 (читання та запис) та додає ключовий третій пункт – власність. Також було розглянуті 3 найбільші блокчейни та вибраний блокчейн – Ethereum на якому будуть розгорнуті смарт-контракти NFT фракталізації. За домогою блокчейну Ethereum були наведені головні складники для NFT фракталізації, а саме акаунт користувача та акаунт смарт-контракту, смарт-контракт та NFT (non-fungible-token).

2 АЛГОРИТМ ПРОВЕДЕННЯ АУКЦІОНУ З МОЖЛИВІСТЮ ВИКУПУ ТОКЕНІВ

Контракт з фракталізації NFT буде розміщений на блокчейні Ethereum, який в свою чергу використовує механізм консенсусу proof-of-stake.

Що стосується блокчейну Ethereum, процес є формалізованим, і досягнення консенсусу означає, що принаймні 66% вузлів у мережі погоджуються щодо глобального стану мережі.

2.1 Механізм консенсусу proof-of-stake

Proof-of-stake [19] лежить в основі певних механізмів консенсусу, які використовуються блокчейном Ethereum для досягнення розподіленого консенсусу. Ethereum використовує proof-of-stake, коли валідатори явно вкладають капітал у формі ETH у смарт-контракт на Ethereum. Цей поставлений ETH діє як застава, яку можна знищити, якщо валідатор поводитиметься нечесно або ліниво.

Є дві основні поведінки, які можна вважати нечесними: пропонування кількох блоків в одному слоті (двозначність) і подання суперечливих підтверджень. Обсяг штрафу в ETH залежить від того, скільки валідаторів також скорочується приблизно в той самий час. Це відоме як «кореляційний штраф», і воно може бути незначним (приблизно 1% частки для одного валідатора) або може призвести до знищення 100% частки валідатора. Штраф накладається на півдорозі періоду примусового виходу, який починається з негайного штрафу (до 0,5 ETH) у День 1, кореляційного штрафу в День 18 і, нарешті, виключення з мережі в День 36.

Щоб взяти участь у якості валідатора, користувач повинен внести 32 ETH у депозитний контракт і запустити три окремі частини програмного забезпечення: клієнт виконання, клієнт консенсусу та валідатор. Після внесення свого ETH користувач приєднується до черги активації, яка обмежує кількість нових валідаторів, які приєднуються до мережі. Після активації валідатори отримують нові блоки від однорангових користувачів у мережі Ethereum. Транзакції, доставлені в блоці, виконуються повторно, а підпис блоку перевіряється, щоб переконатися, що блок дійсний. Потім валідатор надсилає голос (званий атестацією) на користь цього блоку через мережу.

У той час як у proof-of-work [20] час блоків визначається складністю майнінгу, у proof-of-stake темп є фіксованим. Час у proof-of-stake Ethereum поділяється на слоти (12 секунд) і епохи (32 слоти). Один валідатор вибирається випадковим чином, для того щоб пропонувати блок в кожному слоті. Цей валідатор відповідає за створення нового блоку та надсилання його на інші вузли в мережі. Також, у кожному слоті випадковим чином обирається комітет валідаторів, чиї голоси використовуються для визначення дійсності запропонованого блоку.

Коли блокчейн працює оптимально та чесно, у ланцюжку завжди є лише один новий блок, і всі валідатори це підтверджують. Однак валідатори можуть мати різні погляди на наступний блок для ланцюжка через затримку мережі або через те, що валідатор блоку помилився. Тому консенсусним клієнтам потрібен алгоритм, щоб вирішити, якому з них віддати перевагу. Алгоритм, який використовується в proof-of-stake Ethereum, називається LMD-GHOST [21], і він працює, ідентифікуючи форк, який має найбільшу вагу атестацій у своїй історії.

Метою атестації є голосування за погляд валідатора на наступний блок (ланцюжок), зокрема останній виправданий блок і перший блок у поточній епосі (відомі як контрольні точки джерела та цілі). Ця інформація поєднується для всіх валідаторів-учасників, що дозволяє блокчейну досягти консенсусу щодо його стану.

Життєвий цикл включення атестації:

1) генерація;
2) розмноження;
3) агрегація;
4) розмоноження;
5) включення блоку.

Атестація містить такі складові:

- aggregation_bits: бітовий список валідаторів, де позиція відповідає індексу валідатора в їхньому комітеті; значення (0/1) вказує, чи валідатор підписав дані (data);
- data: деталі, що стосуються атестації;
- signature: підпис BLS, який об’єднує підписи окремих валідаторів.

Перше завдання для перевіряючого валідатора – створити дані (data).

Дані містять таку інформацію:

- slot: номер слота, на який посилається атестація;
- index: число, яке визначає, до якого комітету належить валідатор у певному слоті;
- beacon_block_root: кореневий хеш блоку, який валідатор бачить на початку ланцюжка (результат застосування алгоритму форк-вибору);
- source: частина остаточного голосування, що вказує на те, що валідатори вважають останнім корректним блоком;
- target: частина остаточного голосування, що вказує на те, що валідатори вважають першим блоком у поточній епосі.

Вже після створення даних (data) валідатор може змінити aggregation bits на 1 тим самим підтвердивши свою участь.

Валідатори отримують винагороду за надання атестацій. Винагорода за атестацію залежить від двох змінних: базової винагороди та затримки включення.

$$\text{Базова винагорода} = \frac{\text{validator effective balance} * 2^6}{\sqrt{\text{Effective balance of all active validators}}}$$

$$\text{Винагорода за атестацію} = \frac{7}{8} * \text{базова винагорода} * \left(\frac{1}{\text{затримка}}\right)$$

Найкращий випадок для затримки включення дорівнює 1.

На момент, коли валідатори голосували за голову ланцюга (блок *n*), блок *n+1* ще не був запропонований.

Таким чином, атестації природно включаються на один блок пізніше, тому всі атестації, які проголосували за блок *n* як за голову ланцюга, були включені в блок *n+1*, а затримка включення дорівнює 1. Якщо затримка включення подвоюється до двох слотів, винагорода за атестацію зменшується вдвічі, оскільки для розрахунку винагороди за атестацію базова винагорода множиться на величину, зворотну затримці включення.

Штрафи за пропуск цільових і джерельних голосів дорівнюють винагороді, яку б отримав валідатор, якби вони їх подали. Це означає, що замість того, щоб

винагорода була додана на їхній баланс, вона буде вилучена з його балансу рівну вартістості винагороди.

Немає покарання за пропуск головного голосування (тобто головні голоси лише винагороджуються, а не штрафуються). Немає жодного штрафу, пов’язаного з inclusion_delay – винагорода просто не буде додана на баланс валідатора. Також немає покарання за те, що валідатор не запропонував блок.

2.2 Сховище для NFT

Для реалізації алгоритму фракталізації NFT потрібно створити сховище, яке буде зберігати в собі NFT. Тобто, для того щоб зробити фракталізацію з NFT потрібно створити сховище, яке буде зберігати поточне NFT.

При створенні сховища відбувається не тільки передача NFT, а й карбування фракцій з цього NFT. Одна фракція NFT це звичайний взаємозамінний токен стандарту ERC-20. ERC-20 токен повністю виконує всі потреби фракції NFT, а саме він є взаємозамінним та його можна знищити якщо NFT захочуть відтворити.

Сховище також реалізовує два типи відтворення NFT з фракцій, шляхом аукціону та викупом.

Викуп дозволяє відтворити NFT з фракцій, викупом можна скористатися якщо поточний власник NFT встановив резервну ціну.

2.3 Алгоритм викупу

Для того щоб викупити NFT зі сховища потрібно ввести поняття резервної ціни Reserve Price. Резервна ціна сховища – це ціна (в ETH), необхідна для старту аукціону та для повного придбання NFT, яке зберігається у сховищі. Резервна ціна встановлюється розрахованим середньозваженим значенням усіх голосів власників токенів.

Якщо власники токенів, які разом володіють менше ніж 50% від загальної суми фракцій, є єдиними, хто проголосував (або встановив резервну ціну 0), резервна ціна не буде встановлена, і викупити NFT неможливо. Крім того, власники токенів не можуть встановлювати резервну ціну, більшу або рівну 5x і меншу або дорівнює 1/5x поточному середньозваженому. Максимум та мінімум резервної ціни може

змінюватися за допомогою смарт-контракту Governance лише тим користувачем який розгорнув ций смарт-контракт.

Резервна ціна може змінюватися лише шляхом голосування, і на неї не впливає неявна оцінка сховища на основі пропозиції токенів у пулах ліквідності (Пул ліквідності це краудсорсинговий пул монет або токенів, які заблоковані в смарт-контракті та використовуються для полегшення торгів між цими активами на децентралізованій біржі (DEX)).

Для розуміння як працює резервна ціна будуть наведені два приклади:

1. Якщо ви володієте 100% фракцій та встановили резервну ціну на 100 ETH, резервна ціна становитиме 100 ETH. Однак якщо ви володієте 75% фракційних токенів, а хтось інший володіє 25% та голосує за резервну ціну викупу 50 ETH, то резервна ціна викупу NFT становитиме 87.5 ETH.
2. Якщо ви володієте лише 49% фракціями токенів і голосуєте за резервну ціну, але ніхто інший, хто володіє рештою 51% токенів, не проголосував за резервну ціну, резервна ціна для сховища не буде встановлена.

Також слід зазначити, що будь хто може спробувати проголосувати за встановлення резервної ціни, але якщо користувач не володіє жодним токеном то його голос буде дорівнювати нулю і резервна ціна ніяк не зміниться і такий користувач просто втратить свої ETH за нульовий голос.

Однак не все так просто з викупом NFT і може виникнути проблема коли користувач створив сховище, передав своє NFT та накарбував фракцій і продав частину таких токенів через автоматизований маркет AMM. Протокол не дозволяє безпосередньо взяти NFT зі сховища коли користувач є куратором, не володіючи 100% фракцій, але проблему можна вирішити за домогою однієї з двох наведених варіантів.

- Потрібно проголосувати за низьку резервну ціну, яку ви можете собі дозволити, встановіть тривалість аукціону на мінімальний термін, наприклад, 3 дні та ініціюйте викуп, сподіваючись, що ніхто інший не встигне зробити контр-ставку на вашу ставку. Якщо/коли ви виграєте аукціон, NFT буде

переведено у ваш гаманець, і ви зможете обміняти свої 99,999% фракційних токенів на ETH, який ви використали для перемоги на аукціоні;

- розпочати аукціон за резервною ціною, яку, на вашу думку, вартий NFT, і встановіть тривалість аукціону на мінімальний термін, наприклад на 3 дні. Подібним чином ініціюйте аукціон, сподіваючись, що ніхто інший не перевищить вашу ставку. Якщо/коли ви виграєте аукціон, NFT буде переведено на ваш гаманець. Якщо інший користувач перевищить вашу ставку, і ви втратите можливість отримати NFT, принаймні ви зможете продати свої 99,999% фракцій токенів часткового володіння та отримати ETH, який був використаний для перемоги на аукціоні іншим користувачем.

Другий варіант вирішення проблеми є більш надійним так як інші користувачі які володіють малою кількістю долей NFT фракталізації з меншою вирогідністю захочу поборотися за повну ціну NFT з нормально встановленою резервною ціною, і навіть якщо все ж таки один з інших користувачів переб'є ставку можна буде отримати велику частку від тієї суми за яку він це зробив.

2.4 Алгоритм аукціону

Алгоритм аукціону являє собою алгоритмом аукціону з можливістю викупу.

Слід зазначити, що в смарт-контракті для аукціону всі дані для функцій передаються за допомогою глобальної змінної – msg, частіше за все в полі value.

Аукціон має просту структуру та реалізовує функції які будуть наведені в таблиці 2.1:

Таблиця 2.1 – Опис функцій аукціону смарт-контракта

Сигнатура	Опис
function start() external payable	Функція, яка запускає аукціон. msg.value має бути більше або дорівнювати поточній резервній ціні. Також викликає подію Start та передає туди адресу користувача та запропоновану ціну для логування в глобальному стані EVM.
function bid() external payable	Функція для того щоб робити ставки на покупку всього сховища NFT. msg.value – це сума ставки. Також викликає подію Bid та передає туди адресу користувача та запропоновану ціну для логування в глобальному стані EVM.
function end() external	Функція викликається, коли таймер аукціону закінчився. Також викликає подію Won та передає туди адресу користувача який переміг в аукціоні та запропоновану ціну для логування в глобальному стані EVM.
function cash() external	Функція передбачала, щоб власники токенів зняли свою частку в ETH, використану для придбання базового NFT.

Для запобігання проблеми, коли користувач може зробити ставку за декілька секунд до закриття аукціону була додана перевірка, яка подовжує тривалість аукціону.

Окрім проблеми, коли користувач робить ставку під кінець аукціону існує проблема мінімальної ставки, яка буде затримувати завершення аукціону на невизначений термін.

По-крокове виконання алгоритму аукціону.

У наведеному по-кроковому прикладі в аукціоні будуть приймати участь 3 актори.

КРОК 1. Користувач 1 який має частку фракцій може розпочати аукціон за допомогою функцій start та передавши запропоноване значення ціни.

КРОК 2. Користувач 2 робить контр-ставку та перебиває Користувача 1.

КРОК 3. робить меншу ставку за поточну яку запропонував Користувач 2 та отримує помилку.

КРОК 4. Користувач 3 робить контр-ставку та перебиває користувача під самий кінець аукціону. В такому випадку аукціон буде подовжено на 15 хвилин для того щоб інші користувачі могли відреагувати та перебити ставку.

КРОК 5. Користувач 2 робить контр-ставку та перебиває Користувача 3.

КРОК 6. Користувач 2 виграв аукціон за закінченням часу проведення аукціону.

КРОК 7. Будь хто з користувачів має викликати функцію end та завершити аукціон в смарт-контракті, після визову функції аукціон стає завершеним та початкове NFT передається зі сховища до Користувача 2.

КРОК 8. Користувач 1, Користувач 2 та Користувач 3 викликають функцію cash для знищення їх долі ERC-20 токенів та переводу ETH на їх баланси.

На даному етапі слід вести поняття «governance» або «settings» для аукціону. Це правила аукціону, які встановлює автор смарт-контракту, тобто governance – це користувач який розмістив смарт-контракт у мережі.

Governance має можливість змінювати максимальний та мінімальний термін аукціону, мінімальну та максимальну проценту ставку (яка вирішує проблему з

нескінченними ставками) та з якою кількістю фракцій можливо буде розпочати аукціон по викупу всього NFT.

Функціонал Governance реалізується наступними функціями та неведений в таблиці 3.2:

Таблиця 2.2 – Опис функцій аукціону які контролює Governance

Сигнатура	Параметри	Опис
function setMaxAuctionLength(uint256 _length) external onlyOwner	Length – тривалисть аукціону	Функція за допомогою якої встановлюється максимальний термін тривалості аукціону. Також викликає подію UpdateMaxAuctionLength та передає туди старий та новий максимальний час тривалості аукціону для логування в глобальному стані EVM.
function setMinAuctionLength(uint256 _length) external onlyOwner	Length – тривалисть аукціону	Функція за допомогою якої встановлюється мінімальний термін тривалості аукціону. Також викликає подію UpdateMinAuctionLengthта передає туди старий та новий мінімальний час тривалості аукціону для логування в глобальному стані EVM.

Продовження таблиці 2.2

Сигнатура	Параметри	Опис
function setMinBidIncrease(uint256 _min) external onlyOwner	Min – мінімальна ставка в процентах	Функція за допомогою якої встановлюється мінімальна ставка. Також викликає подію UpdateMinBidIncrease передає туди стару та нову мінімальну ставку тривалості аукціону для логування в глобальному стані EVM.
function setMinVotePercentage(uint256 _min) external onlyOwner	Min – мінімальна ставка в процентах	Функція за допомогою якої встановлюється мінімальна процентна кількість фракцій NFT якими треба володіти для того щоб розпочати аукціон викупу. Також викликає подію UpdateMinVotePercentage передає туди стару та нову мінімальну кількість фракцій NFT для логування в глобальному стані EVM.

Для початку аукціону ціна має задоволняти наступні правила аукціону:

- Можна почати аукціон якщо статус сховища рівний inactive, тобто сховище в даний момент вже не виставлене на аукціон та не було викуплено;
- потрібно зробити ставку більшу за резервну ціну;
- потрібно мати % фракцій які встановив Governance для початку аукціону;
- кожна наступна ставка має бути більшою на % який встановив Governance від попередньої.

Закінчення аукціону відбувається коли час проведення аукціону буде вичерпано. Оскільки аукціон проводиться в смарт-контракті то потрібно власноруч викликати функцію end () та завершити його. Доки функція end не буде викликана NFT не буде переданий переможцю аукціону. Даний недолік обумовлений тим, що для смарт-контракту неможливо створити «задачу» яка виконається потім самостійно і також згідно с концепцією смарт-контракти не можуть робити виклики самостійно.

Висновок до розділу

Був розглянутий механізм консенсусу proof-of-stake на якому працює блокчейн Ethereum. Proof-of-stake є ефективних механізмом для включення нових блоків в ланцюг так як є більш енергоефективною на відміну від попередника proof-of-work, має нижчі бар’єри для того щоб бути валідатором та вводить цілу системи нагород та покарань для валідаторів за якими атаки на блокечейн стають економічно невигідними. Також було описане сховище для NFT за допомогою якого буде фракталізуватися NFT та алгоритм дефракталізації – алгоритм аукціону за допомогою якого можна відтворити NFT. Параметри алгоритму аукціону встановлюються користувачем Governance. Алгоритм аукціону захищений від ставки в останній момент часу та додає додатковий час щоб користувачі змогли відреагувати на зміну стану аукціону.

3 ОПИС РЕАЛІЗОВАНИХ СМАРТ-КОНТРАКТІВ NFT ФРАКТАЛІЗАЦІЇ

Смарт-контракти були написані на мові Solidity.

Solidity [22] – це об’єктно-орієнтована мова високого рівня для реалізації смарт-контрактів. Смарт-контракти – це програми, які керують поведінкою облікових записів у стані Ethereum. Solidity призначений для віртуальної машини Ethereum (EVM).

Solidity має статичну типізацію, підтримує наслідування, бібліотеки та складні типи, визначені користувачем, серед інших функцій.

Контракт у Solidity – це набір коду (його функцій) і даних (його стану), які знаходяться за певною адресою в блокчейні Ethereum. Поля оголушують змінну стану, як один слот у базі даних, який можна запитувати та змінювати, викликаючи функції коду, який керує базою даних. Для отримання доступу до змінної зазвичай не додається this, можна звертатися до полів просто через їх назву.

Контракти можуть викликати інші контракти або надсилати Ether на аккаунти користувачів за допомогою викликів повідомлень. Виклики повідомлень схожі на транзакції, оскільки вони мають джерело, ціль, корисні дані, ефір, газ і дані повернення. Насправді кожна транзакція складається з виклику повідомлення верхнього рівня, який, у свою чергу, може створювати подальші виклики повідомлення.

Контракти можна створити «ззовні» через транзакції Ethereum або всередині контрактів Solidity. Коли створюється контракт, його конструктор виконується тільки один раз та не має перегрузок, це означає, що конструктор в контракті тільки один. Конструктор необов’язковий.

Після виконання конструктора остаточний код контракту зберігається в блокчейні. Цей код містить усі загальнодоступні та зовнішні функції, а також усі функції, які доступні звідти через виклики функцій. Розгорнутий код не включає код конструктора або внутрішні функції, які викликаються лише з конструктора.

Смарт-контракт з фракталізації складається з декількох смарт-контрактів які забезпечуєть не тільки логіку фракталізації, а логіку сховища та контролювання, а саме:

- Governance;
- Vault Factory;
- Token Vault;
- Initialize Vault Proxy.

3.1 Смарт-контракт Governance

Смарт-контракт Governance відповідає за контроль над аукціоном, резервною ціною токена, куратора NFT та отримання гонорарів за надання послуг з фракталізації NFT.

Для безпеки та надійності смарт-контракт Governance наслідує абстрактний смарт-контракт Ownable який дає змогу виконувати операції контракту Governance лише користувачу який розгорнув цей контракт в блокчейні або якому були передані ці права. Найпоширенішою та основною формою контролю доступу є концепція власності: є обліковий запис, який є власником контракту та може виконувати на ньому адміністративні завдання. Цей підхід цілком доцільний для контрактів, які мають одного адміністратора. За замовчуванням власником контракту Ownable є користувач, який його розгорнув, що є якраз тим, що потрібно в нашому випадку з Governance. Ownable додається до інших функцій за допомогою модифікаторів в мові solidity які допомогають змінити поведінку функцій декларативним шляхом.

Далі буде наведений приклад сигнатури функції яка захищена модифікатором Ownable:

function setMaxCuratorFee(uint256 _fee) external onlyOwner

Ownable також дозволяє передати право власності з облікового запису власника на новий, і відмовитися від власності смарт-контракту, це загальна модель після завершення початкового етапу з централізованим адмініструванням. Слід бути обережним з Ownable бо повне видалення власника смарт-контракту означатиме, що захищені функції які міг викликати тільки власник будуть заблоковані.

Смарт-контракт Governance вже був частково описаний при описі реалізації алгоритму, але для NFT фракталізації буде недостатньо лише цього функціоналу.

Governance розширюється та вводить нові поняття для контролю над процесом фракталізації NFT – куратора. Куратор це першочерговий власник NFT, а в даному

середовищі – власник сховища. Всі можливості куратора будуть детально розглянуті у Token vault смарт-контракті.

Для того щоб забезпечити стабільність платформи Governance може примусово змінити куратора на іншого користувача та змінити зарезервовану ціну викупу всього NFT якщо аукціон по відтворенню NFT не розпочато.

При кожній виплаті гонорару куратору Governance як творець сервісу також буде отримувати гонорар за надання своїх послуг.

Governance має змогу встановлювати наскільки великим можуть бути гонорари куратора.

Для зручності написання смарт контракту Governance та використання його в інших смарт-контрактах потрібно зробити інтерфейс смарт-контракту. Інтерфейс в solidity схожий абстрактний смарт-контракт за винятком того, що в інтерфейсі не можуть бути реалізовані будь-які функції. Також слід зазначити, що інтерфейси не можуть оголошувати конструктори, змінні або модифікатори. Далі буде наведений інтерфейс IGovernance смарт-контракту Governance

Таблиця 3.1 – Опис функцій інтерфейсу смарт-контракту Governance

Сигнатура	Опис
function maxAuctionLength() external returns(uint256);	Сигнатура для зміни максимальної тривалості аукціону
function minAuctionLength() external returns(uint256);	Сигнатура для зміни мінімальної тривалості аукціону
function maxCuratorFee() external returns(uint256);	Сигнатура для змінення максимальної процентної кількості грошей яку куратор отримує за гонорари
function governanceFee() external returns(uint256);	Сигнатура для змінення процентної кількості гонорару яку Governance отримує за надання послуг
function minBidIncrease() external returns(uint256);	Сигнатура для змінення мінімальної межі на яку можно збілишити ставку на аукціоні

Продовження таблиці 3.1

function minVotePercentage() external returns(uint256);	Сигнатура для змінення мінімальної межі кількості голосів (потрібно для старту аукціону)
function maxReserveFactor() external returns(uint256);	Сигнатура для змінення максимальної межі резервної ціни NFT
function minReserveFactor() external returns(uint256);	Сигнатура для змінення мінімальної межі резервної ціни NFT
function feeReceiver() external returns(address payable);	Сигнатура для змінення користувача якому будуть надходити гонорари за надання послуг NFT фракталізації

Функціонал Governance доповнюється наступними функціями які описані в таблиці 3.1:

Таблиця 3.2 – Опис функцій смарт-контракту Governance

Сигнатура	Параметри	Опис
function setGovernanceFee(uint256 _fee) external onlyOwner	Fee - гонорар	Функція змінює поточний гонарар Governance. Викликає подію setGovernanceFee та передає туди старий та новий гонорар для логування в глобальному стані EVM.
function setMaxCuratorFee(uint256 _fee) external onlyOwner	Fee - гонорар	Функція змінює поточний максимальний гонорар куратора. Викликає подію UpdateCuratorFee та передає туди старий та новий гонорар для логування в глобальному стані EVM.

Продовження таблиці 3.2

function setFeeReceiver(address payable _receiver) external onlyOwner	Receiver – адреса governer	Функуція змінює на яку саме адресу буде поступати гонорар за надання послуг. Викликає подію UpdateFeeReceiver та передає туди попереднього та нового користувача для логування в глобальному стані EVM.
function setMaxReserveFactor(uint256 _factor) external onlyOwner	Factor – ціна в процентах	Функція змінює максимальний поріг зарезервованої ціни NFT. Викликає подію UpdateMax-ReserveFactor та передає туди старе та нове значення ціни для логування в глобальному стані EVM.
function setMinReserveFactor(uint256 _factor) external onlyOwner	Factor – ціна в процентах	Функція змінює мінімальний поріг зарезервованої ціни NFT. Викликає подію UpdateMinReserveFactor та передає туди старе та нове значення ціни для логування в глобальному стані EVM.
function setMinVotePercentage(uint256 _min) external onlyOwner	Min – мінімальна ціна 1000 – буде дорівнювати 100%	Функція змінює мінімальний поріг для голосів за допомогою яких можна розпочати аукціон щодо викупу NFT зі сховища Викликає подію UpdateMinVotePercentage та передає туди старе та нове значення голосів для логування в глобальному стані EVM.

3.2 Смарт-контракт Token Vault

Смарт-контракт Token Vault – це сердце всієї системи – сховище. Token Vault створює сховище в якому буде зберігатись фракталізоване NFT та фракції цього NFT. Для сховища фракталізації NFT потрібно реалізувати стандарт ERC-20 для взаємозамінних фракцій, та частковий функціонал ERC-721 для самого NFT.

ERC-20 [23] представляє стандарт для взаємозамінних токенів, іншими словами, вони мають властивість, завдяки якій кожен токен точно збігається (за типом і значенням) з іншим токеном. Це робить токени ERC20 корисними для таких речей, як засіб обміну, права голосу, ставки та добре підходить для фракталізації в якій одна фракція точно дорівнює будь якій іншій. Може виникати потреба розділити свої токени на довільні суми: скажімо, якщо всього є 5 FNFT, ви можете надіслати 1,5 FNFT другові, а 3,5 FNFT залишити собі. На жаль, Solidity та EVM не підтримують таку поведінку: можна використовувати лише цілі (цілі) числа, що створює проблему. Ви можете надіслати 1 або 2 токени, але не 1,5. Щоб обійти це, ERC20 надає поле десяткових знаків, яке використовується для визначення кількості знаків після коми. Щоб мати можливість передати 1,5 FNFT, десятковий знак має бути не менше 1, оскільки це число має один десятковий знак. Для досягнення даної поведінки смарт-контракт може використовувати більші цілі числа, так що баланс 50 буде відповідати 5 FNFT, переказ 15 відповідатиме 1,5 FNFT, що надсилаються, і так далі.

Десяткові дроби використовуються лише для відображення. Уся арифметика всередині смарт-контракту все ще буде виконуватися за правилами Ethereum та виконуватись з цілими числами, і саме різні користувацькі інтерфейси, наприклад, такі як гаманці повинні коригувати відображені значення відповідно до десяткових дробів. Загальна кількість токенів і баланс кожного облікового запису не вказуються в FNFT: вам потрібно розділити на x^десяткових знаків, щоб отримати фактичну суму FNFT. Ether і більшість смарт-контрактів використовують дясеткове значення 18 для смарт-контрактів токенів ERC-20. Під час карбування токенів або їх передачі фактично надсилатиється число num FNFT * x^десяткових знаків.

Також смарт-контракт Token Vault наслідує два допоміжні смарт-контракти ERC20Upgradeable та ERC721HolderUpgradeable. ERC20Upgradeable це

модифікований смарт-контракт стандарту ERC20 який допомагає вирішити одну з проблем смарт-контрактів – їх неможливо змінювати. За допомогою проксі та функції delegatecall яка є функцією низького рівня, схожа на call.

Можна привести приклад роботи функції настпним чином. Коли контракт А виконує виклик delegate до контракту В, код В виконується з пам’яттю контракту А.

Користувач здійснює виклик функції до проксі. Потім проксі делегує виклик делегату, де знаходиться код функції. Результат повертається до проксі, який пересилає його користувачу.

Оскільки delegatecall використовується для делегування виклику, викликана функція виконується в контексті проксі. Це означає, що сховище проксі-сервера використовується для виконання функції, що призводить до обмеження, що сховище договору делегування має бути лише доданим. Код операції delegatecall було введено в EIP-7 [24].

ERC721HolderUpgradeable доповнює ERC-721 стандарт функцією за допомогою якої можна отримати селектор solidity.

Коли маркер ERC-721 tokenId передається в цей контракт через функцію IERC721-safeTransferFrom користувачем from from, викликається onERC721Received. Функція має повернути свій селектор Solidity, щоб підтвердити передачу токену. Якщо повертається будь-яке інше значення або інтерфейс не реалізований одержувачем, передачу буде скасовано.

Смарт-контракт можна вважати стандартом ERC-20 тільки якщо він імплементує настпуні методи які описані в таблиці 3.3:

Таблиця 3.3 – Опис функцій смарт-контракту ERC-20

Сигнатура	Параметри	Опис
function name() public view returns (string)	-	Повертає ім'я токена, наприклад Fractional NFT
function symbol() public view returns (string)	-	Повертає символ токена, напиклад, FNFT

Продовження таблиці 3.3

Сигнатура	Параметри	Опис
function decimals() public view returns (uint8)		Повертає кількість десяткових знаків, які використовує токен
function totalSupply() public view returns (uint256)	-	Повертає загальну кількість токенів.
function balanceOf(address _owner) public view returns (uint256 balance)	address _owner – обліковий запис користувача	Повертає баланс іншого аккаунту
function transfer(address _to, uint256 _value) public returns (bool success)	uint256 _value – кількість токенів. address _to – публічний ключ аккаунту.	Передає кількість токенів на адресу address _to та запускає подію Transfer.
function transferFrom(address _from, address _to, uint256 _value) public returns (bool success)	uint256 _value – кількість токенів. address _to – публічний ключ аккаунту address _from – публічний ключ аккаунту	Використовується для робочого процесу виведення, дозволяючи контрактам передавати токени від імені власника
function approve(address _spender, uint256 _value) public returns (bool success)	_spender– публічний ключ аккаунту uint256 _value – кількість токенів.	Дозволяє spender`u знімати з вашого рахунку кілька разів, до суми _value. Якщо цю функцію викликати знову, вона перезаписує allowance на _value. Викликає подію Approval

Продовження таблиці 3.3

Сигнатура	Параметри	Опис
function allowance(address _owner, address _spender) public view returns (uint256 remaining)	_ spender – публічний ключ аккаунту _owner – публічний ключ аккаунту	Повертає суму, яку _spender ще може зняти з _owner.

Для фракцій NFT окрім стандарту ERC-20 ще потрібно реалізувати два методи, які дають змогу карбувати нові ERC-20 та знищувати їх. Вони наведені в таблиці 3.4.

Таблиця 3.4 – Опис функцій для контролювання кількості токенів ERC-20

Сигнатура	Параметри	Опис
function _mint(address account, uint256 amount) internal virtual	Account – аккаунт користувача на який будуть нараховані токени. Amount – кількість токенів.	Функція за допомогою якої карбуються нові токени.
function _burn(address account, uint256 amount) internal virtual	Account – аккаунт користувача токени якого будуть знищені. Amount – кількість токенів.	Функція за допомогою якої знищуються токени.

ERC-721 [25] представляє стандарт для NFT, іншими словами, цей тип токена є унікальним і може мати іншу цінність, ніж інший токен із того самого смарт-контракту. ERC721 є складнішим стандартом, ніж ERC20, із кількома необов’язковими розширеннями та розділений на декілька контрактів. ERC-721надає такі функції, як передача токенів з одного облікового запису на інший, отримання

поточного балансу токенів облікового запису, визначення власника певного токена, а також загальна кількість токенів, доступних у мережі.

Для NFT фракталізації непотрібно реалізовувати ERC-721 в повному обсязі так як сам NFT буде зберігатись в сховищі, в випадку фракталізації потрібно реалізувати IERC721Receiver інтерфейс для перевірки чи можна використовувати функцію safeTransferFrom в Vault factory.

Таблиця 3.5 – Частковий опис функцій ERC-721 які потрібні для передачі та утримування NFT в сховищі.

Сигнатура	Опис
Function onERC721Received(address operator, address from, uint256 tokenId, bytes data) public virtual overrides return (bytes4)	Щоразу, коли маркер IERC721 tokenId передається в цей контракт через IERC721.safeTransferFrom оператором from from, ця функція викликається. Смарт-контракт повинен повернути свій селектор Solidity, щоб підтвердити передачу. Якщо повертається будь-яке інше значення або інтерфейс не реалізований одержувачем, передачу NFT буде скасовано.
function safeTransferFrom(address from, address to, uint256 tokenId) public virtual override	Безпечно передає токен від from до to, спочатку перевіряючи, що одержувачі контракту знають про протокол ERC721 (IERC721Receiver), щоб запобігти вічне блокування NFT.

Token Vault зберігає в собі всю необхідну інформацію щодо переданого NFT, стан аукціону, куратора, поточного гонорара та Governance. Алгоритм аукціону також знаходиться в. сховищі, він був описаний раніше в мат забезпеченні. Сховище створюється за допомогою Vault factory. Функції Token Vault будуть розбиті на різні таблиці так як користувачі мають мати конкретні права для їх визову.

Token Vault імплементує наступні головні методи які наведені в таблиці 3.6 (також до цих методів входять методи аукціону):

Таблиця 3.6 – Головні функції смарт-контракту Token Vault

Сигнатура	Параметри	Опис
function initialize(address _curator, address _token, uint256 _id, uint256 _supply, uint256 _listPrice, uint256 _fee, string memory _name, string memory _symbol) external initializer	Curator – власник NFT Token – адреса NFT Id – айді токену NFT Supply – кількість ERC-20 токенів. listPrice - початкова ціна NFT. Fee – гонорар Name – назва ERC-20 токену Symbol - символ ERC-20 токену	Функція за допомогою якої ініціалізується сховище для NFT.
function updateUserPrice(uint256 _new) external	_new – бажана ціна в ETH	Функція для кінцевого користувача, щоб оновити бажану ціну продажу

Продовження таблиці 3.6

Сигнатура	Параметри	Опис
function _beforeTokenTransfer(address _from, address _to, uint256 _amount) internal virtual override	from – відправник токену ERC-20 to – приймач токену ERC-20 amount – кількість ERC-20 токенів	Функція, яка використовується для оновлення ціни відправника та одержувача при передачі токенів
function redeem() external		Функція яка знищує всі токени ERC-20 та передає ERC-721 до користувача який викликав функцію
function cash() external		Функція знищування токенів ERC-20 для отримання ETH
function _sendETHOrWETH(address to, uint256 value) internal	To – адреса користувача Value - кількість	Функція спробує передати ETH, якщо це вдасться передає WETH
function _attemptETHTransfer(address to, uint256 value) internal returns (bool)	To – адреса користувача Value - кількість	Функція спробує надіслати ETH яка не гарантується завершенням. Поверне false при невдачі.

Token Vault імплементує наступні методи для куратора які будуть наведені в таблиці 3.7:

Таблиця 3.7 – Функції смарт-контракту Token Vault які доступні тільки для поточного куратора сховищем

Сигнатура	Параметри	Опис
function updateCurator(address _curator) external	Curator – новий куратор сховища	Функція змінює куратора сховища
function updateAuctionLength(uint256 _length) external	Length – тривалість аукціону	Функція змінює тривалість аукціону на вказану
function updateFee(uint256 _fee) external	Fee – новий гонорар	Функція змінює кількість гонорару
function claimFees() external		Функція отримання гонорарів для куратора та Governance. Використовує функцію claimFees
function _claimFees() internal		функція розрахунку виплачування гонорарів

Token Vault імплементує настпуні методи для Governance які описані в таблиці 3.8:

Таблиця 3.8 – Функції смарт-контракту Token Vault які доступні тільки для Governance

Сигнатура	Параметри	Опис
function kickCurator(address _curator) external	Curator – Адреса нового куратора	Функція яка змінює куратора сховища
function removeReserve(address _user) external		Функція за допомогою якої можна прибрати резервну ціну NFT

Token Vault імплементує настпуні методи для перегляду які описані в таблиці 3.9:

Таблиця 3.9 – Опис функцій смарт-контракту Token Vault для перегляду

Сигнатура	Параметри	Опис
function reservePrice() public view returns(uint256)	-	Повертає резеврну ціну NFT

3.3 Смарт-контракт Vault Factory

Смарт-контракт Vault Factory реалізовує логіку карбування нових сховищ за допомогою Initialized Vault Proxy та зберігає в собі адреси відкарбованих сховищ, логіку Token Vault та Governance. Vault Factory додатково наслідує контракти Pausable та Ownable які потрібно для того щоб при нагальній необхідності заборонити карбування нових сховищ. Vault Factory імплементує настпуні методи які наведені в таблиці 3.10:

Таблиця 3.10 – Опис функцій смарт-контракту Vault Factory

Сигнатура	Параметри	Опис
function mint(string memory _name, string memory _symbol, address _token, uint256 _id, uint256 _supply, uint256 _listPrice, uint256 _fee) external whenNotPaused returns(uint256)	Curator – власник NFT Token – адреса NFT Id – айді токену NFT Supply – кількість ERC-20 токенів. listPrice - початкова ціна NFT. Fee – гонорар Name – назва токену Symbol - символ ERC-20 токену	Функція для створення нових сховищ. Викликає подію Mint

Продовження таблиці 3.10

function pause() external onlyOwner		Функція за допомогою якої можливо призупинити роботу контракту
function unpause() external onlyOwner		Функція за допомогою якої можливо відновити роботу контракту

3.4 Смарт-контракт Initialized Vault Proxy

Смарт-контракт Initialized Vault Proxy потрібен для того, щоб ініціалізувати кожне нове сховище. За допомогою проксі можливо розгорнути смарт-контракт дешевше.

За задумом смарт-контракти є незмінними. З іншого боку, якість програмного забезпечення значною мірою залежить від здатності оновлювати та виправляти код для створення ітераційних випусків. Незважаючи на те, що програмне забезпечення, засноване на блокчейні, значно виграє від незмінності технології, певний ступінь мінливості потрібен для виправлення помилок і потенційного вдосконалення продукту.

Найбільш актуальною проблемою, яку вирішує проксі, є те, як проксі відкриває весь інтерфейс логічного контракту, не вимагаючи однозначного відображення всього інтерфейсу логічного контракту.

Точка входу в контракт одна і в ній знаходиться роутер, що зчитує з вхідної транзакції хеш назви функції, і JUMPI, що робить на відповідну позицію в скомпільованому коді. Якщо переданий хеш не відповідає жодній відомій роутеру функції, виконується метод fallback. Метод delegatecall дозволяє викликати точку входу іншого контракту, використовуючи слоти сховища від поточного контракту.

Проксі контракт зберігає в одному із слотів (з високим номером) адресу контракту, код якого треба виконати (в нашому випадку Token Vault). При виклику проксі, викликається fallback, а звідти викликається delegatecall.

При оновленні контракту розгортається нова логіка, окремо, новий незмінний контракт, а потім адреса нового контракту зберігається в покажчик всередині Proxy-контракту.

Висновок до розділу

Для фракталізації NFT були розроблені смарт-контракти сховища, фабрика сховищ, governance та Initial Vault Proxy на мові Solidity. Смарт-контракт сховища є головним контрактом в якому зберігається NFT та створюються його фракції, також сховище відповідальне за аукціон та можливість викупу NFT. Смарт-контракти Vault Factory та Initial Vault Proxy працюють разом для розгортання для дешевого розгортання нових сховищ в блокчейні. Vault Factory додатково зберігає адресу користувача Governance який буде переданий кожному сховищу. Governance смарт-контракт відповідає за параметри для аукціону та адресою на яку будуть виплачуватись гонорари для розробників. Користувачі які підковані у питанні можуть створювати сховища з фракталізації NFT за допомогою etherscan.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Взаємодія зі смарт-контрактами

4.1.1 Засоби розробки

Взаємодія зі смарт-контрактами може бути виконана двома способами, перший з них це взаємодія з контрактами напряму за допомогою etherscan [10] або через власну реалізацію взаємодії зі смарт-контрактом. Взаємодія через etherscan непогана бо за допомогою його можна подивиться розгорнуту інформацію о смарт-контракті та його транзакціях але частіше розробники пишуть власні реалізації взаємодії для того щоб зменшити поріг входу для користувачів прибравши детальну інформацію яка звичайному користувачу буде зайва.

Для взаємодії зі смарт-контрактом буде використовуватися бібліотека ethers.js [26] та опен сорс веб-фреймворк nuxt [27] який написаний на базі одного з найпопулярніших JavaScript [28] фреймворків vue.js [29]. Слід зазначити, що хоч для взаємодії з контрактом буде написаний власний веб-застосунок всеодно слід давати посилання на etherscan якщо користувач захоче подивитися детально інформацію щодо транзакцій або перевірити код смарт-контракту.

Vue – це фреймворк JavaScript для створення інтерфейсів користувача. Він створений на основі стандартних HTML, CSS і JavaScript і забезпечує декларативну та компонентну модель програмування, яка допомагає ефективно розробляти інтерфейси користувача.

Vue має декілька головних функцій за допомогою яких значно спрощується написання веб-застосунків, а саме компонентну структуру, декларативне відображення та реактивність.

Компонент у Vue це SFC з трьома блоками script (логіка), template (html розмітка) та style(css або препроцесори).

Декларативне відображення у Vue розширює стандартний html синтаксисом шаблону, який дозволяє декларативно описувати вивід html на основі стану JavaScript.

Реактивність у Vue автоматично відстежує зміни стану JavaScript і ефективно оновлює DOM [30], коли відбуваються зміни. Також слід коротенько зазначити, що

Vue має свою стратегію оновлення DOM під назвою VDOM. Віртуальний DOM (VDOM) – це концепція програмування, де «віртуальне» представлення інтерфейсу користувача зберігається в пам’яті та синхронізується з «реальним» DOM.

Nuxt в свою чергу це обгортка над Vue яка структурує проект та значно розширює його можливості. Одніє. з головних функцій Nuxt це те що фреймоврк під капотом використовує серверний двигун Nitro який в свою чергу дає можливість користуватися фул-стак можливостями.

Для зберігання даних буде використовувитися сховище Pinia [31] для Nuxt. Pinia дозволяє ділитися станом між компонентами/сторінками та використовує Composition API під капотом.

Бібліотека ethers.js є повною та компактною бібліотекою для взаємодії з блокчейном Ethereum та його екосистемою.

Стандартний спосіб взаємодії з контрактами в екосистемі Ethereum це ввійковий інтерфейс програми контракту (ABI). За допомогою ABI з контрактом можно взаємодіяти як поза блокчейном, так і для взаємодії між контрактами. Дані кодуються відповідно до їх типу, як описано в цій специфікації. Кодування не є самоописовим, тому для декодування потрібна схема. Для взаємодії з контрактами у бібліотеки є мета-класс Contracts за допомогою якого будуть створені методи контракту.

В якості провайдеру був використаний провайдер Alchemy який надає можливість не тільки робити запити до блокчейну, а і є в своєму роді Amazon`ом для web 3. Alchemy робить блокчейн легким. Розробники отримують доступ до платформи для розробників Alchemy web3, центрального центру для розробки всього, що стосується блокчейну. Завдяки вбудованим потужним інструментам Alchemy спрощує розробку dApp, заощаджуючи незліченну кількість часу.

NFT API від Alchemy дозволяє швидко отримувати всю необхідну інформацію про NFT з блокчейну, включаючи Ethereum і Polygon.

Перекази це відображення вартості, якою обмінюються два рахунки, і іноді користувачі бажають переглянути історію транзакцій, пов’язаних із певним рахунком або адресою. Отримання історії транзакцій наразі є надзвичайно складним і неефективним завданням, яке вимагає від користувачів сканувати весь блокчейн та

індексувати все для пошуку транзакцій, пов’язаних із потрібною адресою. За допомогою Transfers API від Alchemy користувачі можуть запитувати всі історичні транзакції для адреси одним запитом.

Отримання всього про транзакції для номера блоку або хешу блоку за один виклик API за допомогою методу alchemy_getTransactionReceipts – для основної мережі та тестових мереж на Ethereum, Polygon, Optimism і Arbitrum.

Надання інформації про певні токени, наприклад метадані або баланси через Token API Alchemy.

Методи Trace API від Alchemy надають розробникам доступ до найдетальнішої інформації про активність у ланцюжку та дозволяють користувачам отримати повне відстеження зовнішнього впливу на будь-яку транзакцію, виконану в Ethereum.

Для написання логіки веб-застосування та взаємодією з контрактом буде використовуватися мова Typescript.

TypeScript [32] – це строго типізована мова програмування, яка ґрунтується на JavaScript і надає вам кращі інструменти будь-якого масштабу. За допомогою TypeScript при написанні коду буде допущено значно менше помилок за рахунок строгої типізації та код буде більш декларований, що допоже легше зрозуміти, що написано.

Серед інструменів розробки використовувались два IDE (Integrated development environment) Visual Studio Code з плагінами для solidity та Remix яка створена спеціально для написання смарт-контрактів.

Visual Studio Code [33] – це легкий, але потужний редактор вихідного коду, який працює на компютері та доступний для Windows, macOS і Linux. Він поставляється з вбудованою підтримкою JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов і середовищ виконання (таких як C++, C#, Java, Python, PHP, Go, .NET).

Найголовнішим плюс Visual Studio Code є те що він надає можливість легко кастомізувати середу під себе за допомогою плагінів.

Remix IDE [34] – це інструмент який не потрібно налаштовувати з графічним інтерфейсом для розробки смарт-контрактів. Використовується як експертами, так і початківцями, Remix допоможе вам робити ваші задачі вдвічі швидше. Remix добре

поєднується з іншими інструментами та забезпечує простий процес розгортання в ланцюжку за вашим вибором. Remix відомий своїм візуальним налагоджувачем. Remix – це місце, де кожен приходить, щоб дізнатися про Ethereum.

4.1.2 Архітектура Веб-застосунку

Nuxt та Vue мають функціональний стиль написання тому архітектуру можна розбити на 3 складові – компоненти, composables, storages. Nuxt це ssr rendering фреймворк тому за допомогою його можна розділити функціонал на приватний (який буде виконуватися тількі на сервері) та клієнтський який буде надсилатися користувачу до браузеру.

Composables – це цункція яка використовується для інкапсуляції та повторного використання логіки з відстеженням стану. Composables використовують Composition API.

Storages – це глобальні сховища які використовують під капотом Composition API (Реактивну систему Vue та її хуки).

Компоненти можна також поділити на дві складові – компоненти, які буду повторно використані та сторінки на яких буде відображатися інформація.

Основні компоненти веб-застосунку описані в таблиці 4.1:

Таблиця 4.1 – Опис сновних компонентів веб-застосунку

Компонент	Composables and Storages use	Опис
UserWallet	useWallet, useWalletStore	Дає змогу під’єднати свій Metamask [33] гаманець. Після під’єднання трансформується в меню якому є інформація щодо адреси користувача, балансу, профілю та можливості вийти з аккаунту

Продовження таблиці 4.1

Компонент	Composables and Storages use	Опис
UploadVault	useWallet, useWalletStore, useProvider, useVaultFactoryContract	Сторінка за допомогою якої по-крокова можливо сховищо та фракталізувати NFT
FullVault	useProvider, useVaultFactoryContract	Сторінка на якому відображається вся інформація о сховищі
VaultFractionBuy	useProvider, useVaultFactoryContract	Компонент який надає можливість купувати фракції
VaultAuctionBid	useProvider, useVaultFactoryContract	Компонент який надає можливість робити ставки на аукціоні
VaultFractionOwners	useProvider, useVaultFactoryContract	Компонент який відображає користувачів які придбали NFT фракції

Nuxt надає гнучку структуру layouts яка базується на vue slots, допомогає швидко створювати та змінювати головний компонент проекта. При зміні сторінок змінюватись буде лише slot в якому буде рендеритись головний компонент.

Структуру nuxt та фреймворків які основані на компонентах можна порівняти з багатошаровою структурою. При компонентній структурі слід дотримуватися атомарного підходу в якому свій функціонал буде зроблений в своєму компоненті. Компоненти в свій же час є незалежними в виконанні від інших компонентів.

Далі буде наведений приклад layout структури nuxt на рисунку 4.1.

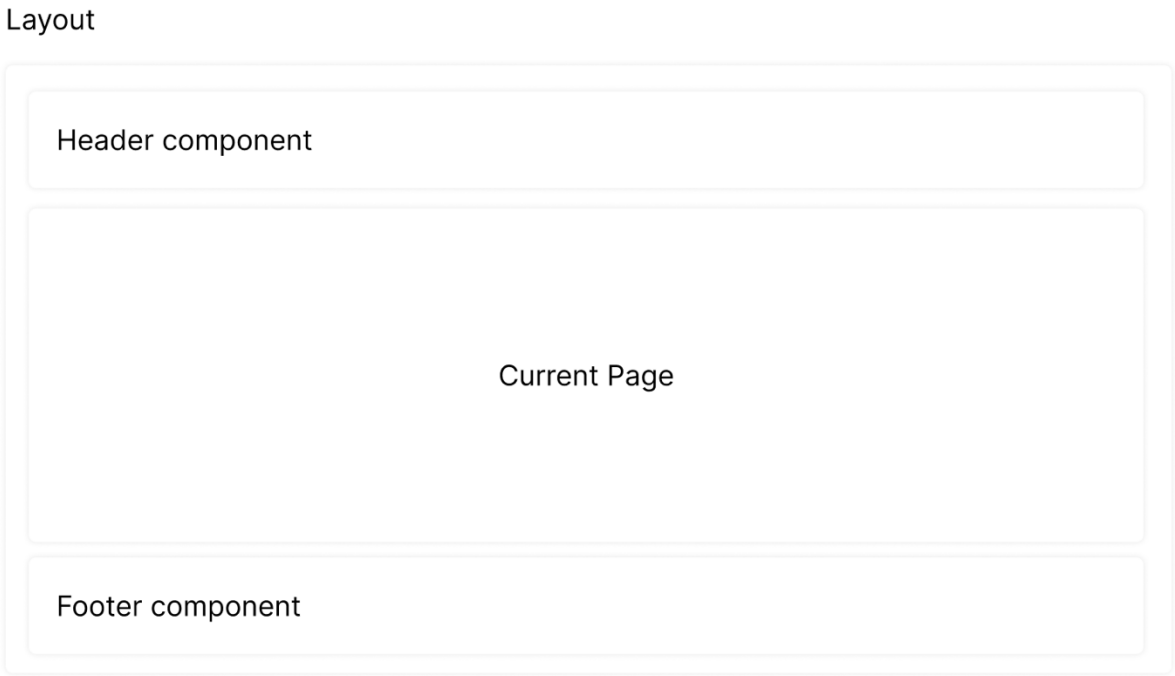

Рисунок 4.1 – Layout структура веб-застуснку

4.1.3 Інструкція користувача

Перед початком взаємодії з веб-застосунком потрібно встановити розширення для браузеру під назвою Metamask за допомогою якого буде під’єднуватися гаманець до веб-застосунку.

MetaMask [35] – це розширення для доступу до розподілених програм із підтримкою Ethereum або «Dapps» у браузері. MetaMask також дозволяє користувачеві створювати та керувати власними ідентифікаторами, тому, коли Dapp хоче виконати транзакцію та записати в блокчейн, користувач отримує безпечний інтерфейс для перегляду транзакції, перш ніж схвалити або відхилити її. Оскільки MetaMask додає функціональність у звичайний контекст браузера, MetaMask вимагає дозволу на читання та запис на будь-якій веб-сторінці. Після скачуванню потрібно пройти просто «реєстрацію», а саме генерування нової адреси. Нова адреса генерується зо допомогою мнемонічної фрази. Також для додаткового захисту MetaMask при збереженні мнемонічної фрази додатково шифрує її вказаним паролем. Ось як має виглядати готове розширення MetaMask. Приклад гаманця Metamask неведений на рисунку 4.2.

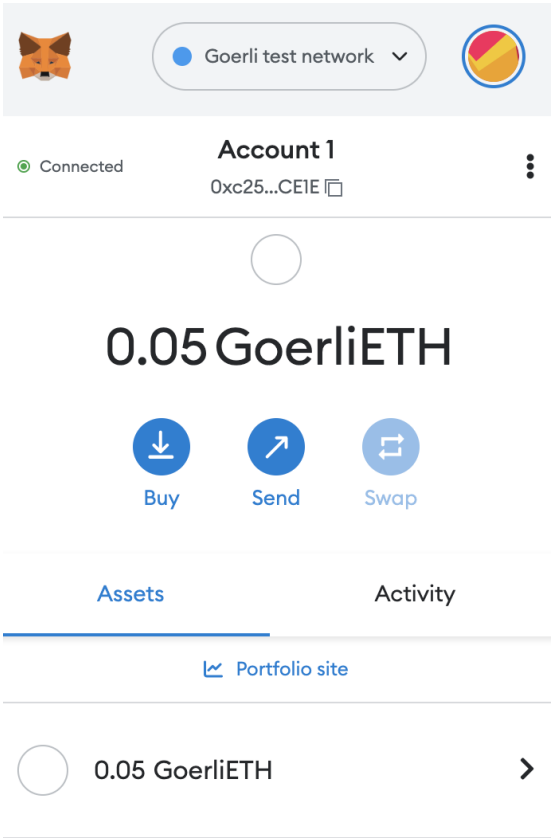

Рисунок 4.2 – MetaMask гаманець

Робота веб-застосунку була проведена на testnet блокчейні Goerli. Testnet – це тестова мережа в якій розробники тестують свої смарт-контракти.

Для початку роботи з веб-застосунком потрібно натиснути на кнопку wallet зверху в заголовку сайту.

Connected статус наголошує на тому, що MetaMask гаманець готовий для використання.

В веб застосунку також будуть зміни, кнопка wallet зміниться на адресу користувача. Далі на рисунку 4.3 будуть наведені два стани кнопки веб-застосування до та після підключення гаманця.

Рисунок 4.3 – Перехід стану після підключення ним гаманця

Наступним кроком спробуємо створити сховище та фракталізувати NFT. Для того щоб це було можливо потрібно мати на адресі хоча б один NFT який буде переданий у сховище. Якщо аккаунт новостворений то потрібно взяти трішки ETH та купити якесь NFT або створити його власноруч за допомогою будь якого смарт-контракту в мережі.

Для отримання своїх перших ETH в тестовій мережі можна скористатися один з faucet сайтів для одержання безкоштовних ETH. Goerli faucet від alchemy [36] дає 0.2 ETH один раз на день.

Після одержання ETH можна скористатися будь-якою платформою яка також працює на мережі Goerli та придбати там NFT, одна з найбільш платформ по продажу NFT це платформа OpenSea [37] яка також розгорнута в тестових мережах.

Повертаючись до фракталізації NFT після того як на аккаунті хоч одне NFT можна перейти до створення першого сховища. Для створення нового сховища потрібно натиснути на кнопку «New» біля аутенфікованої адреси в правому кутку.

Створення сховища для NFT складаєтьс з трьох кроків

1) вибір NFT, яке буде передану у сховище;
2) заповнення всіх деталей для створення сховища;
3) перевірка на можливість передачі NFT у сховище.

Далі на рисунку 4.4 наведена сторінка вибору NFT.

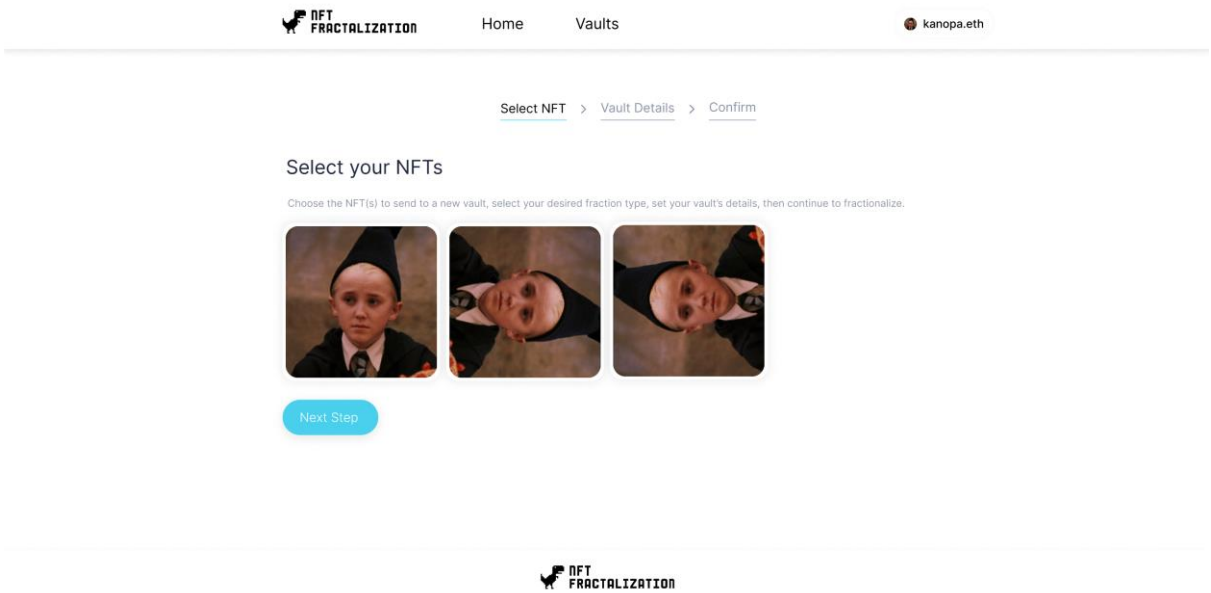

Рисунок 4.4 – Сторінка створення нового сховища NFT. Вибір NFT

Коли бажане NFT буде вибране можна переходити на наступний крок з деталями сховища.

Потрібно заповнити всі поля, а саме назву сховища, назву та символ взаємозамінного токену ERC-20 (фракцій), зарезервовану ціну та гонорар в процентах який буде виплачуватись щороку. Далі на рисунку 4.5 буде відображена веб-форма з усіма полями для створення сховища в якому буе зберігатися NFT та фракталізовані токени.

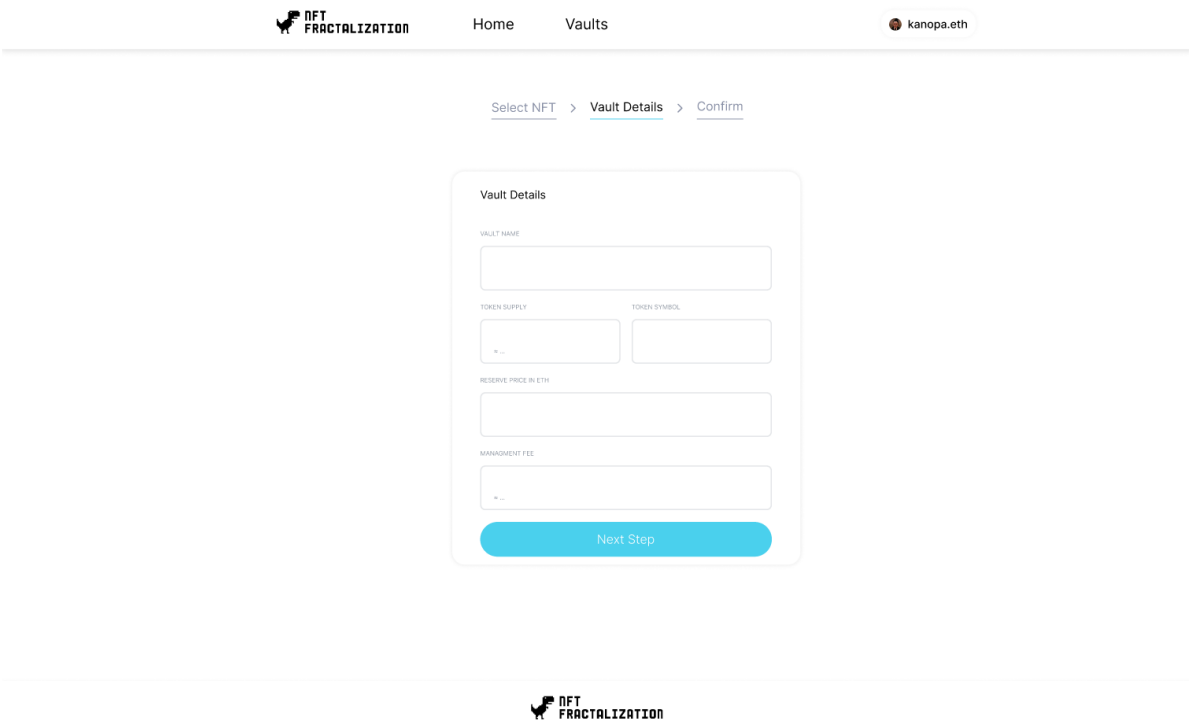

Рисунок 4.5 – Інформація о сховищі фракталізації NFT

Останнім кроком потрібно підтвердити передачу NFT для того щоб його можливо було передати в сховище. Для цього потрібно просто натиснути на кнопку Approve NFT та підтвердити транзакцію в Metamask. Після завершенні транзакції appove nft можна створювати сховиже, створення сховища також потрібно підтвердити в Metamask підписавши транзакцію. Далі на рисунку 4.6 буде наведений осанній крок створення сховища на якому потрібно підтвердити передачу NFT na зрештою створити сховище.

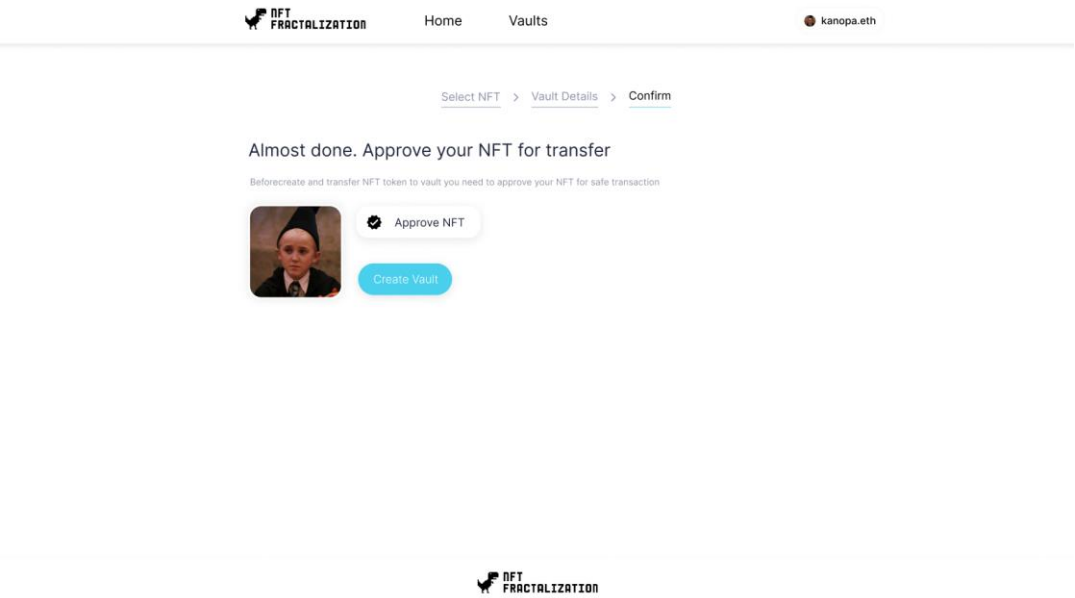

Рисунок 4.6 – Підтвердження передачі NFT у сховища та створення сховища

Сховище NFT успішно створено. NFT було розміщене у сховище та були викарбовано 10 000 фракцій ERC-20. Далі наведений рисунок 4.7 на якому наведена повна інформація щодо сховища фракталізованого NFT.

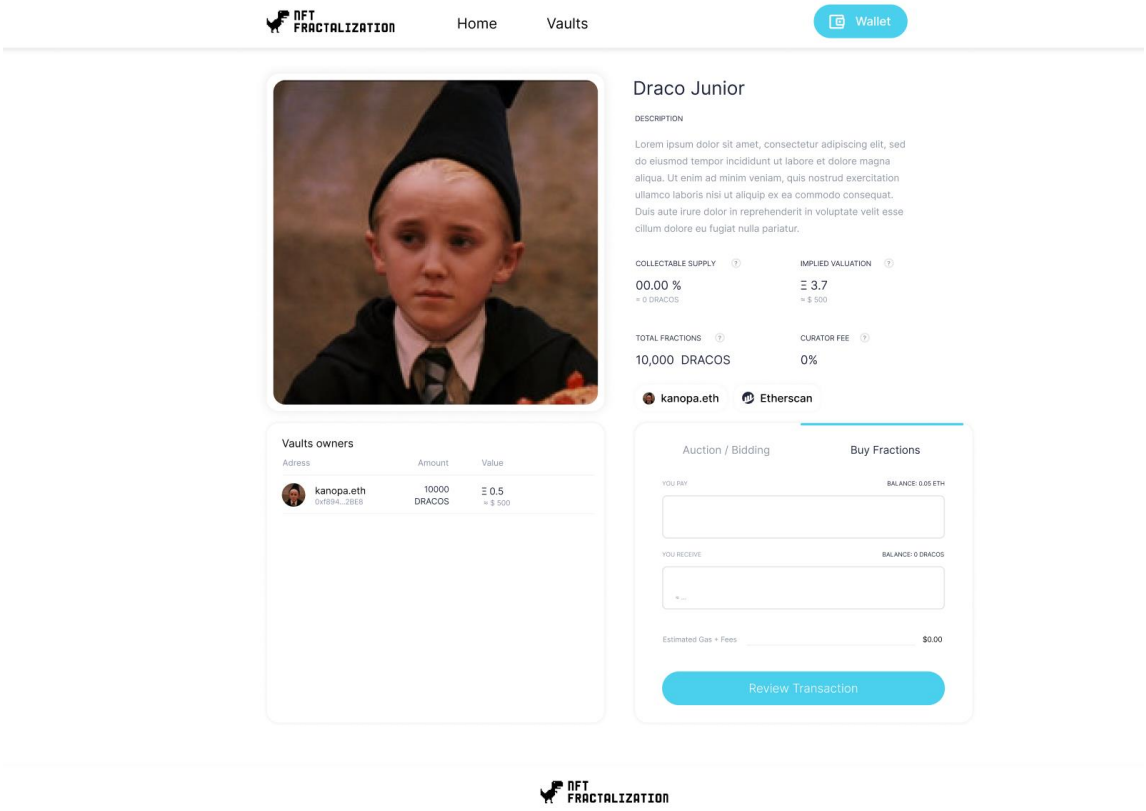

Рисунок 4.7 – Створене сховище фракталізованого NFT

Також можна переглянути профілі користувачів зі створеним сховищами NFT, на рисунку 4.8 зображений приклад профілю користувача.

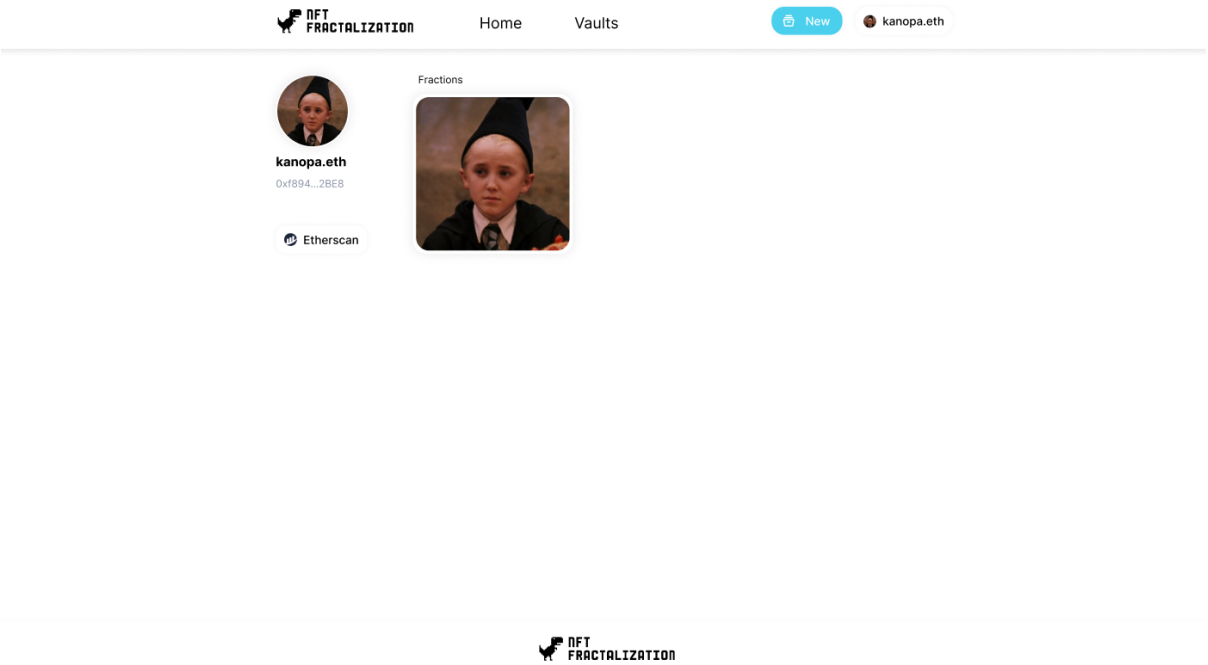

Рисунок 4.8 – Профіль користувача з фракталізованими NFT (сховищами)

Після успішної фракталізації NFT слід такод надати декілька варіантів подальшого розвитку новоствореного сховища фракталізації NFT.

Основними варінтами розвитку є:

- Надіслати фракції друзям для розділення та спільного володіння NFT;
- Передача їх членам спільноти або власникам певних токенів для популяризації власних фракцій в цих кругах;
- Надання ліквідності на децентралізованій біржі (DEX).

Для створення ліквідності через DEX потрібно розуміти, що таке пул луквідності. Пул ліквідності – це краудсорсинговий пул монет або токенів, які заблоковані в смарт-контракті та використовуються для полегшення торгів між цими активами на децентралізованій біржі (DEX).

Автоматизований маркет-мейкер [38](AMM) – це повністю автоматизована децентралізована біржа (DEX), де торгівля здійснюється проти пулу токенів, який називається пулом ліквідності. Алгоритм регулює значення та ціни токенів у пулах ліквідності. Оскільки AMM не покладаються на активний ринок покупців і продавців, угоди можуть відбуватися в будь-який час. Приклади популярних AMM включають Uniswap, Curve і Balancer.

Кожен, хто має підключення до Інтернету та володіє будь-яким типом токенів ERC-20, може стати постачальником ліквідності, надаючи токени до пулу ліквідності AMM. Постачальники ліквідності зазвичай отримують комісію за надання токенів пулу. Цю комісію сплачують трейдери, які взаємодіють із пулом ліквідності.

Uniswap [39] – це провідна децентралізована криптобіржа, яка працює на блокчейні Ethereum.

Платформу Uniswap було створено в 2018 році на базі блокчейну Ethereum, другого за величиною криптовалютного проекту за ринковою капіталізацією, що робить її сумісною з усіма токенами ERC-20 та інфраструктурою, наприклад сервісами гаманців, такими як MetaMask.

Uniswap також є повністю опен сорсним, що означає, що кожен може скопіювати код для створення власних децентралізованих бірж. Це навіть дозволяє користувачам безкоштовно розміщувати токени на біржі. Звичайні централізовані біржі орієнтовані на прибуток і стягують дуже високі комісії за розміщення нових монет, тому вже одне це є помітною різницею. Оскільки Uniswap є децентралізованою біржею (DEX), це також означає, що користувачі постійно контролюють свої кошти, на відміну від централізованої біржі, яка вимагає від трейдерів відмовитися від контролю над своїми закритими ключами, щоб замовлення можна було реєструвати у внутрішній базі даних, а не ніж виконуватися на блокчейні, що займає більше часу та дорожче. Зберігаючи контроль над закритими ключами, він усуває ризик втрати активів у разі злому біржі.

Для того щоб зробити новий пул потрібно скопіювати адресу смарт-контракту. Це можна зробити за допомогою Etherscan, далі буде навений рисунок 4.9 з адресою смарт-контракту.

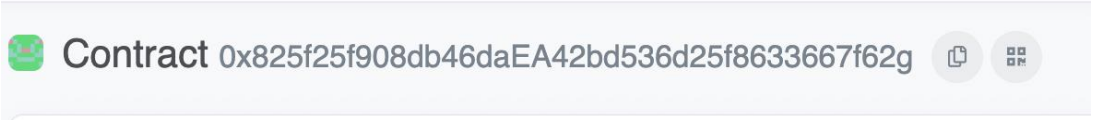

Рисунок 4.9 – Копіювання адреси смарт-контракту

Далі потрібно перейти на сайт Uniswap та створити нову позицію

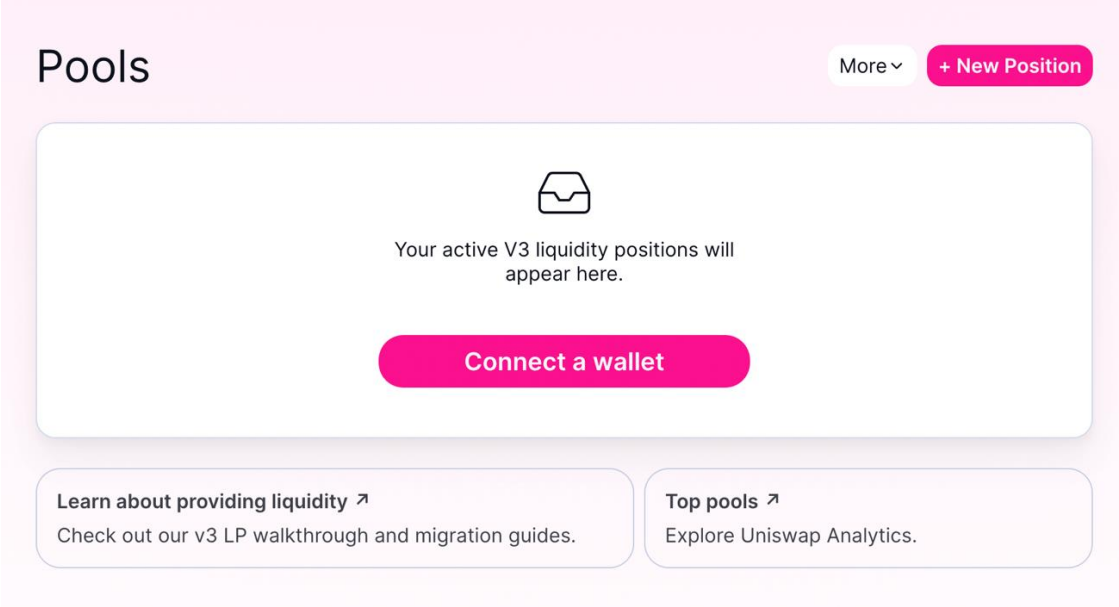

Рисунок 4.10 – Створення нової позиції на платформі Uniswap

Далі потрібно під’єднати свій гаманець (так само як і на веб-застосунку) NFT фракталізації і додати скопійовану адресу смарт-контракта.

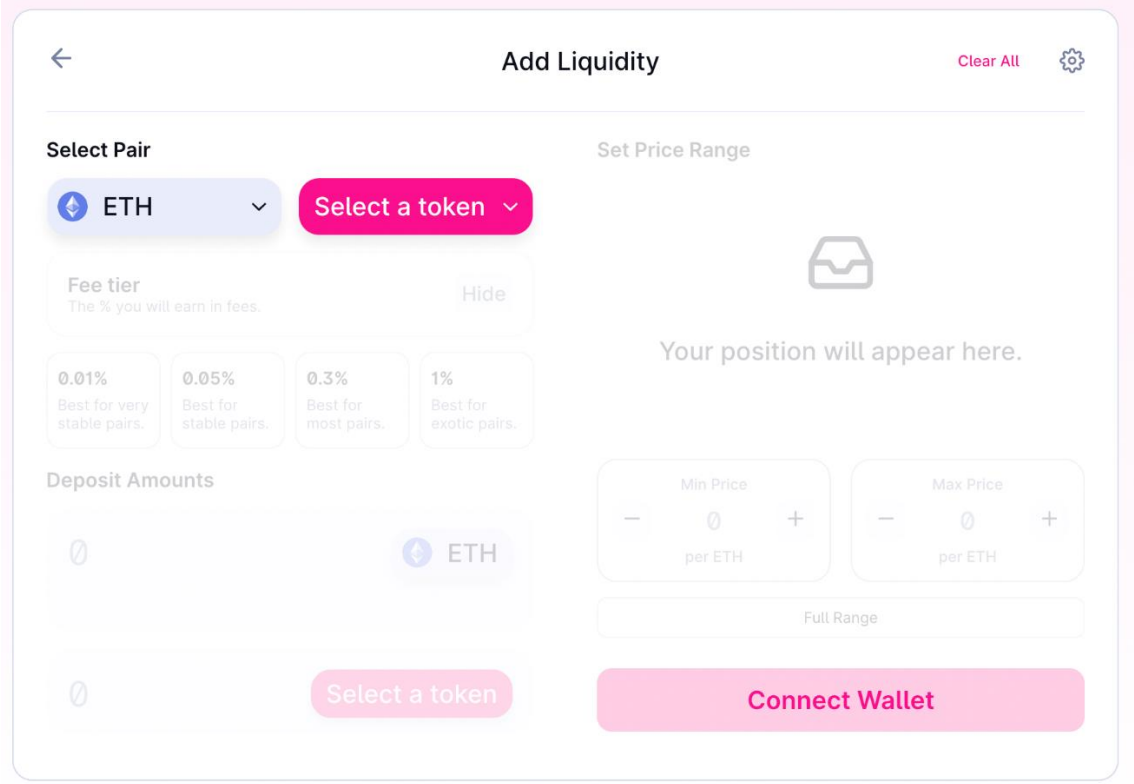

Рисунок 4.11 – Додавання ліквідності до токену

Налаштуйте свій пул. Далі потрібно буде вибрати рівень комісії (зазвичай рекомендується 0,3%), а потім визначитися з ціною вашого нового токена в ETH.

Кожен вказує ціну яку він бажає за свої токени. Вам також потрібно буде визначитися з діапазоном, у якому ви надаєте ліквідність. Виходячи з відповіді тут, вам може знадобитися надати один або обидва активи в пул.

Нижче наведено приклад налаштування пулу, де токени GOLEM коштують 6227.28 GLM за один ETH. Наданий діапазон цін становить від 3027.8 до 11266. Мінімальна ціна має становити 1% або менше поточного мінімуму під час налаштування початкового пулу ліквідності. Це дозволяє пулу залишатися в діапазоні та дозволяє іншим купувати вашу ліквідність.

Далі на рисунку продемонстроване додавання ліквідності зі вказаними діапазонами цін.

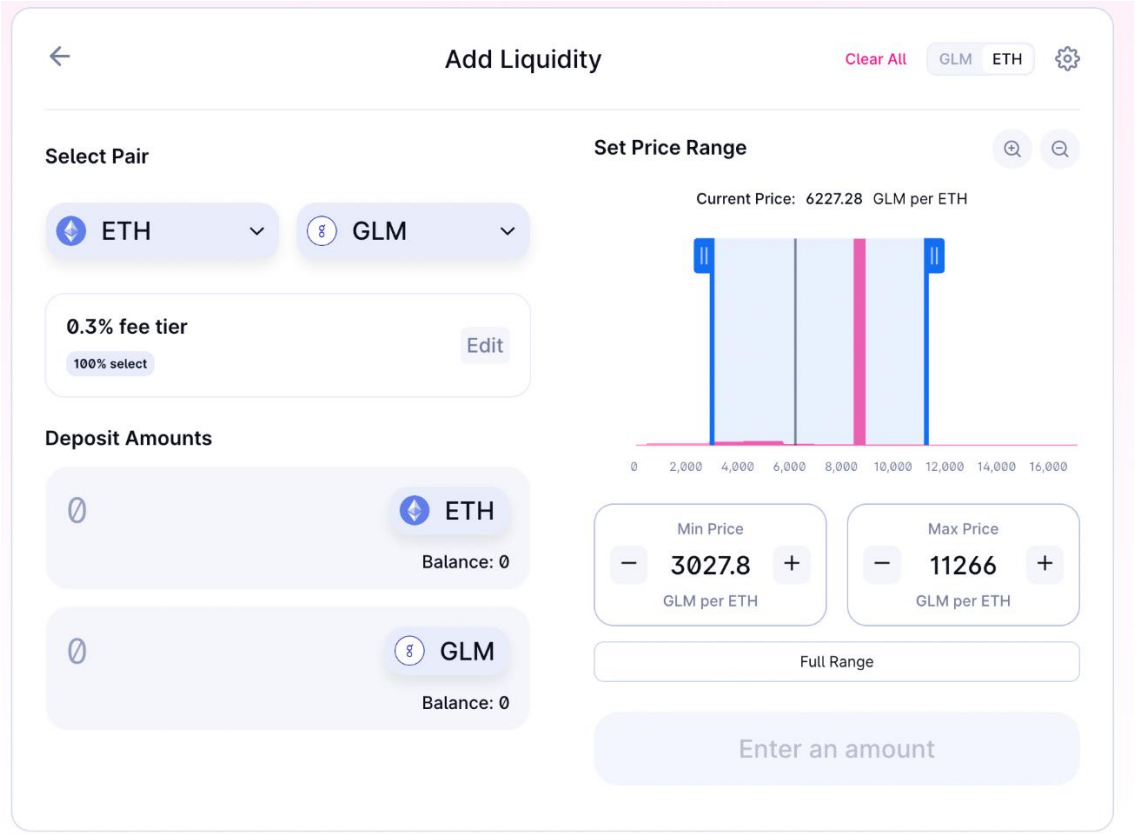

Рисунок 4.12 – Додавання ліквідності до зі вказаними діапазонами цін

Також можна змінити настройки та подивитися скільки ETH буде коштувати один токен GOLEM. Далі на рисунку 4.13 буде наведений приклад діапазону цін 1 токену GOLEM.

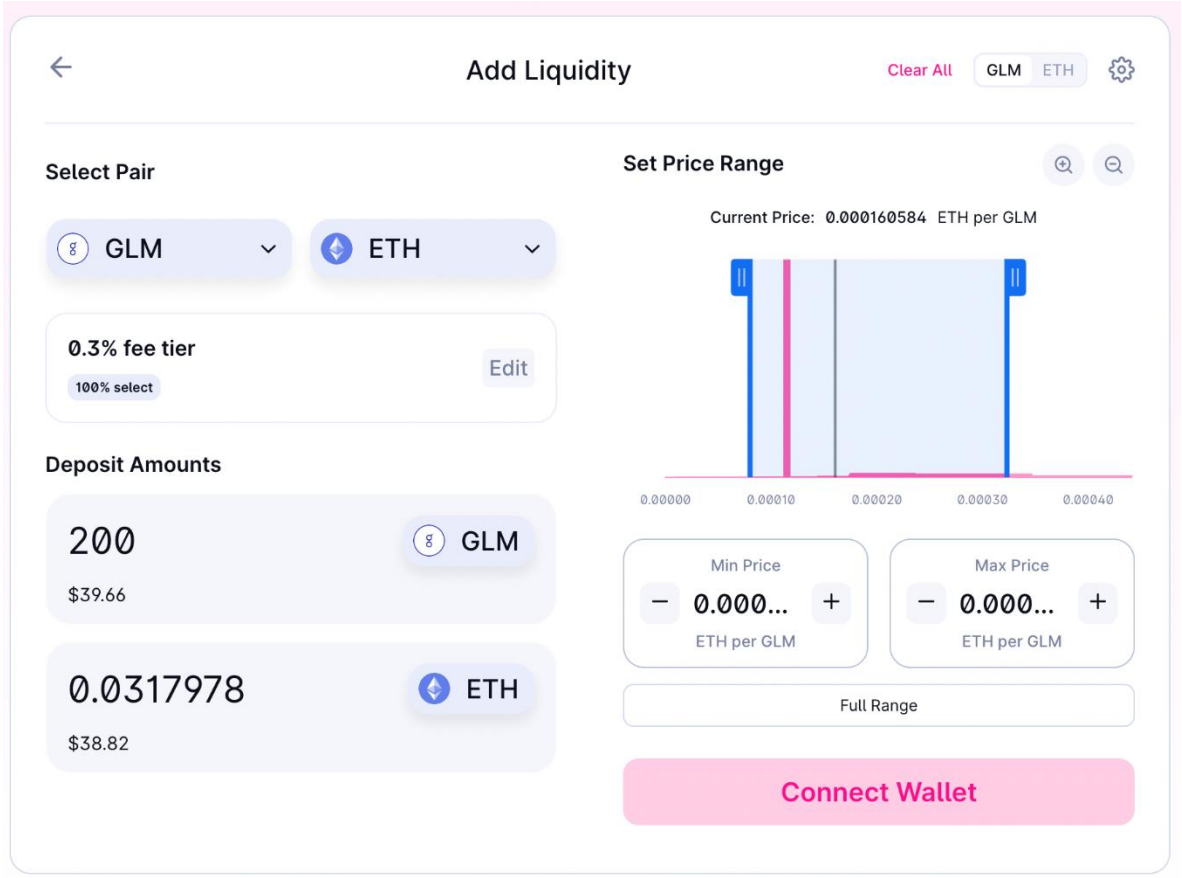

Рисунок 4.13 – Додавання ліквідності до зі вказаними діапазонами цін за 1 GOLEM

Якщо ви отримуєте повідомлення о помилці «недостатня ліквідність», то вам скоріш за все потрібно знизити суму ETH за одну фракцію в пулі.

Після цього потрібно підтвердити операцію ізгодом разгорнути пул. Тепер у вас є ліквідність, щоб люди могли купувати та збирати фракції.

Висновок до розділу

Для полегшення взаємодії зі смарт-контрактами було реалізоване веб-застосування з використанням фреймворку nuxt. Провайдером між веб-застосуванням та блокчейном виступає один з найпотужніших провайдерів – Alchemy.

Розроблене веб-застосування дозволяє зручно та в декілька кроків фракталізувати NFT та взаємодіяти з ним переглядаючи різноманітну інформацію. К доповненню веб застосування дозволяє переглядати всі сховища які були створені фабрикою сховищ та профілі користувачів з коллекцією їх фракталізованих NFT.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

Ідея стартап проекту полягає в наданні можливості користувачам фракталізувати їх унікальні ассети на фракції для можливості їх продажу та отримання гонорарів за їх утримання. Далі буде наведена таблиця 5.1 з описом ідей стартап проекту

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигода для користувача
Фракталізація NFT на дрібні фракції для можливості володіння частинкою унікального та дорогого продукту більшої кількості людей та можливість заробляти на користувачах які купили фракції шляхом виплати гонорарів	Фракталізація NFT, створення сховища NFT для фракталізації	Можливість володіти частинкою унікального асета та отримання додаткового заробітку від NFT шляхом сплачування гонорарів поточну куратору NFT
	Відтворення NFT з фракцій	Користувачі мають можливість відтворити NFT яке було розбите на фракції.

Заробіток від проекту здійснюється при отримані щорічних гонорарів за використання смарт-контрактів. При кожному зборі гонорарів куратором, частина гонорару йде користувачу який стоїть за розгортанням контракту Governance або користувач адреса якого вказана в Governance смарт-контракті.

Аналіз потенційних техніко-економічних переваг був здійснений за допомогою М. Портера.

Результати аналізу щодо визначення сильних, слабких та нейтральних характеристик наведені в таблиці 5.2:

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ з/п	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		NFT fractalization (FNFT)	Unicle	Otis			
1.	NFTфракталізація	NFT фракталізація є основним функціоналом.	NFT фракталізація є лише одним із напрямків на платформі тому питання фракталізації. NFT	NFT фракталізація не я ключовим функуціоналом платформи.		Фракталізація для Otis я не є основним функціоналом та функція погано описана з чого виникають труднощі з користуванням	Для FNFT та Unicle фракталізація є основним функціоналом та є його самою сильною стороною

5.2 Технологічний аудит ідеї проекту

Майже всі технології які були використані в ідеї проекту є open source, а це означає, що вони є безкоштовними та підтримуються комьюніті. Підтримка комьюніті являється як плюсом так і мінусом так як кожна людина може вирішити

проблему або додати новий функціонал так і люди які обслуговують (підтримують) проект можуть втратити інтерес до нього та перестати його підтримувати. Далі в таблиці 5.3 буде наведена технологічна здійсненність ідей проекту.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ з/п	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1.	Розробка смарт-контракту для здійсненя NFT фракталізації	Мова Solidity	+	+
		Мова Vyper	+	+
2.	Розробка веб-застосунку для взаємодії зі смарт-контрактами	Nuxt фреймворк	+	+
		Next фреймворк	+	+
3.	Розроблений веб-застосунок та смарт контракти будуть взаємодіяти через провайдера	Alchemy провайдер	+	± Безкоштовний якщо кількість транзакцій менше ніж 12 мільйонів
		Infura провайдер	+	± Безкоштовний якщо кількість транзакцій менше ніж 3 мільйони.

Обрана технологія реалізації проекту: Технології за допомогою яких буде розроблена ідея проекту є open source тому вони є загальнодоступними та наявними для всіх. Для реалізації смарт-контрактів була вибрана мова Solidity, а веб-застосування буде розроблене на Nuxt. В якості провайдера був вибраний Alchemy

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Перед початком аналізу ринку слід провести попередній аналіз потенційного ринку для розуміння показників стану ринку, який буде наведений в таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ з/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	8
2	Загальний обсяг продаж, грн/ум.од	Немає даних
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Для входу в ринок NFT потрібно мати чітке розуміння щодо блокчейну, гаманців та ринка NFT в цілому
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	20-30%

Ринок NFT хоча трохи і просів в порівнянні з 2021 роком але він все одно є дуже популярним, наприклад, за 2022 рік було розгорнуто близько 117 922 [40] смарт-контрактів, що є третю від усіх розгорнутих смарт-контрактів в цілому. Також швидкими темпами збільшується ріст скачування бібліотек, що явно показує те, що web 3 з кожним роком є все більш цікавий напрямок як для звичайних клієнтів так і для розробників.

Надалі слід визначити потенційних клієнтів які будуть користуватися продуктом та сформулювати вимоги які будуть у кожної з них до продукту, характеристика наведена в таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ з/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Можливість групового володіння унікальним асетом NFT	Артисти, творчі люди, колекціонери	Колекціонери на відміну від інших аудиторій мають тенденцію до зберігання власних активів.	Бути ознайомленими з концепцією блокчейну та NFT
2.	Можливість отримання гонорарів за часткове володіння NFT	Артисти, творчі люди, інвестори	При великій ліквідності інвестори можуть викупити та зібрати фракталізоване NFT у ціле.	Бути ознайомленими з концепцією блокчейну та NFT

Серед потенційних клієнтів можна виділити артистів, творчіх людей, колекціонерів та інвесторів.

Аналіз ринкового середовища. Для початку в таблиці 5.6 будуть наведені фактори загроз для стартапу.

Таблиця 5.6 – Фактори загроз

№ з/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Зростання конкуренції	За рахунок вдосконалення технологій з’являється все більше конкурентів	Впровадити маркетингову компанію та вдосконалити функціонал веб-застосунку
2.	Велика швидкістю розвитку web 3	Можливо з деяким часом буде запропонований стандарт ERC за яким будуть робити фракталізацію	Мати досвідчену команду розробників яка зможе швидко відреагувати на зміни
3.	Спад ринку NFT	NFT стає все менш популярним	Впровадити маркетингову компанію серед людей яким незнайома концепція web 3 задля залучення нових клієнтів

Таблиця 5.7 – Фактори можливостей

№ з/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Інвестори	Вливання інвестицій в проект	Розширення команди та функціоналу веб-застосунку
2.	Розширяти функціонал	Наразі функціонал обмежується лише NFT фракталізацією але у веб-застосунок можна додавати інтеграцію з новими смарт-контрактами.	Інтеграція веб застосунку з новими смарт-контрактами

Аналіз пропозиції за допомогою якого можна визначити загальні конкуренції на ринку буде наведений в таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції	Монополістична конкуренція. Можливість подивитись смарт-контракти та відтворити їх	Потрібно підтримувати маркетинг та покращувати веб-застосунок
За рівнем конкурентної боротьби	Національний	Немає особливого впливу так як всі проекти в рівних умовах
За галузевою ознакою	Внутрішньогалузева. Складно виділити гальзь в web 3, але боротьба йде лише у профільованих проектах	Потрібно шукати конкурентні переваги
Конкуренція за видами товарів	Товарно-видова	-
За характером конкурентних переваг	Нецінова	Головними перевагами є інтуїтивний ux дизайн та розкрученість проекту
За інтенсивністю	Марочна	Постійне вдосконалення та покращення досвіду користувача

Аналіз ринкових можливостей буде проведений за допомогою методики 5 сил Портеру. Аналіз Портера включає в себе три сили «горизонтальної» конкуренції: загроза появи продуктів-замінників, загроза появи нових гравців, рівень конкурентної

боротьби, та обидві сили «вертикальної» конкуренції: ринкова влада постачальників і ринкова влада споживачів.

Результати аналізу наведені в таблиц 5.9і:

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конку-ренти в галузі	Потенційні конкурен-ти	Постачаль-ники	Клієнти	Товари-замінники
	Otis, unicly	Всі, хто написав смарт-контракти та розгорнув їх в блокчейні	Відсутні	Анонімність, економічна вигода від фракталізації NFT на частини	Etherscan
Висновки:	Існує всього декілька сервісів, які написали свої смарт-контракти та розгорнули їх в мережі одна поняття фракталізації досить нове та не має еталонної або правильної реалізації	Поява конкурентів можлива але без додаткових dapps, веб-застосунків та довіри яка сформувалася з часом	Постачальників немає.	Якщо користувачі не будуть отримувати достойних гонорарів вони не будуть користуватися даними послугами та перейдуть до конкурентів	Продвинуті користувачі можуть самі написати смарт-контракти та розгорноти їх у мережі та користуватися ними за допомогою etherscan, але це дуже незручно та складно

Роблячи висновки з таблиці 5.9 можна зробити висновок, що проект має потенціал та може зайняти свою нішу на просторах NFT. Для підтримки проекту та

можливості бути конкурентними на ринку мають бути присутніми наступні фактори конкурентноспроможності:

– якісна реалізація смарт-контрактів;
– інтуїтивний та простий в користуванні веб-застосунок;
– досвідчена команда web 3 розробників;
– економічна вигідність для власників NFT фракталізувати NFT у нас за рахунок низьких гонорарів для розробників;
– маркетинг та інвестиція у рекламну компанію продукту.

Детальніше про кожну характеристику можна переглянути в таблиці 5.10.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ з/п	Фактор конкурентоспроможності	Обґрунтування чому характеристика значуща та надає перевагу над конкуретними проектами
1.	Якісна реалізація смарт-контрактів	Реалізація смарт-контракту – це перше в чому піде розбиратися користувач перед тим як скористатися NFT фракталізацією. Саме добре продуманий та якісно покритий тестами контракт може визвати довіру у користувачів у порівнянні з конкурентами
2.	Інтуїтивний та простий в користуванні веб-застосунок	Інтуїтивний та простий – це майже як антонім для web 3 тому це дуже гостре та критичне питання мати простий веб-застосунок на якому користувач зможе знайти відповіді на всі свої питання.
3.	Досвідчена команда web 3 розробників	Web 3 знаходиться на початку свого розвитку та постійно динамічно змінюється. Досвід

Продовження таблиці 5.11

1	2	3
		команди надасть можливість адаптуватися та швидко реагувати на зміни. Також контракти неможливо виправити якщо вони вже розгорнуті в мережі, саме тому досвід фахівці дуже важливий
4.	Економічна вигідність для власників NFT фракталізувати NFT у нас за рахунок виплати гонорарів кураторам та низьких гонорарів розробникам	Надання можливості фракталізувати NFT додає можливість пасивно заробляти шляхом виплати гонорарів куратору. Чим менше буде гонорар за надання послуг тим більше будуть отримувати куратори NFT.
5.	Маркетинг та інвестиція у рекламну компанію продукту	Як і в будь яких проектах будь-то web 2 або web 3 проекти потрібна маркетингова компанія щоб користувачі мали змогу дізнатися про сервіс

За характеристиками також можна зробити порівняння з вже існуючим конкурентом. Порівняння наведене в таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін

Характеристика	Бали (1 - 10)	Рейтинг товарів-конкурентів у порівнянні з unicly				
		-10	-5	0	5	10
Якісна реалізація смарт-контрактів	10			+		
Інтуїтивний та простий в користуванні веб-застосунок	8		+			
Досвідчена команда web 3 та web 2 розробників	5				+	
Економічна вигідність для власників NFT фракталізувати NFT у нас за рахунок низьких гонорарів для розробників	9		+			
Маркетинг та інвестиція у рекламну компанію продукту	2				+	

Результати аналізу стартап проекту будуть наведені у таблиці 4.4 у вигляді SWOT-аналізу.

Таблиця 5.12 – SWOT аналіз стартап проекту

Сильні сторони: Продумані та протестовані смарт-контракти. Веб-застосунок для взаємодії зі смарт-контрактами. Можливість заробляти на гонорарах від фракталізації NFT	Слабкі сторони: Мала кількість користувачів Недовіра до фракталізації NFT
Можливості: Розширення функціоналу веб застосування для більшого охвату користувачів Регуляція Governance гонорару за послуги.	Загрози: Поява конкурентів з можливістю фракталізувати NFT Динамічні зміни web3

При проведенні SWOT-аналізу були виявлені основні сильні та слабкі сторони стартап-проекту з NFT фракталізації. Основними сильними сторонами є перш за все сама ідея фракталізації за допомогою якої користувач який фракталізував своє NFT має змогу отримувати гонорари, смарт-контракти які були добре продумані та повністю покриті тестами та веб-застосування для взаємодії з ними. Також були виявлені і слабкі сторони проекту до яких можна віднести те, що у проекта немає своєї клієнтської бази та скепсис до самої ідеї фракталізації NFT.

Останнім етапом буде визначення альтернатив за якими буде впроваджений стартап-проект. Альтернативи ринкового впровадження будуть наведені в таблиці 5.13

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ з/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Реалізовувати проект і паралельно шукати інвесторів	80%	3 місяці
2.	Шукати інвесторів і тільки після реазіовувати проект	25%	6 місяців
3.	Реалізувати проект, а вже потім шукати інвесторів	50%	6 місяців

Реалізовувати проект і паралельно шукати інвестора є самою виграшною альтернативною серед усіх тому виберемо її. Інші альтернативи мають великі ризики за рахунок того, що web 3 постійно оновлюється і знання які були актуальні пів року тому но сьогоднішній день такими залишатися не будуть.

5.4 Розробка ринкової стратегії стартап-проекту

Для розробки ринкової стратегії потрібно розуміти на яку групу користувачів націлений проект, бачити наявний рівень конкуренції та попиту та наскільки просто буде війти в проект групам людей.

NFT є широким поняттям бо за допомогою нього можна зробити майже будь яку річ унікальною в цифровому просторі будь то реала річ або цифброва. В випадку фракталізації NFT будем виділяти наступні цільові групи користувачів: творчі люди, колекціонери, інвестори.

Цільові групи будуть детально розглянуті в таблиці 5.14.

Таблиця 5.14 – Аналіз цільових груп

Цільова група потенційних клієнтів	Готовність до використання продукту	Попит	Конкуренція в сегментах	Освоєння в сегменті (простота входу)
Творчі люди, артисти	Середня	Середній	Середня	Проект дозволить артистам дробити їх арти на фракції та отримувати гонарари за це. Авторам вже знайома базова концпеція NFT та вони легко зайти в проект
Колекціонери	Висока	Високий	Висока	Колекціонерам даний проект є особливо цікавим так як вони зможуть мати в своїх колекціях частки артів. Дана група також знайома з операціями NFT тому їм буде не складно зайти в проект
Інвестори	Середня	Середній	Мала	Інвестори відносятся до нових проектів більш консервативно, для входження інвесторів потрібна клієнтська база. Інвестори на ранніх етап не будуть заходити в проект
Обрані цільові групи: Творчі люди, артисти, колекціонери				

За описами цільових груп можна зробити висновки, що інвестори довіряють більш перевіриним часом проектам та на ранніх етапах цільова група не буде заохочена в участі проекту. Творчі люди, артисти та колекціонери навпаки будуть заохочені прийняти участь в проекті так як артисти будуть мати з цього більшу грошову вигоду, а колекціонери зможуть мати у себе в колекції унікальні арти.

Після розуміння цільових груп потрібно визначити базову стратегію для стартапу. Базова стратегія описана в таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ з/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Будемо працювати лише на ринку з фракталізації NFT але для декількох сегментів користувачів	Стратегія диференційованого маркетингу	Досвідчена команда, зручний та інтуїтивний дизайн, реалізація фракціонування NFT шляхом переміщенням його до сховища та можливістю викупу або проведенню аукціону для викупу	Стратегія диференціації

Далі слід вибрати стратегію конкурентної поведінки таблиця 5.16.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ з/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Так	Так, при успіху проекту компанію може розшириться та надавати послуги, щодо створення нових NFT або їх колекцій	Компанія буде копіювати можливістю перегляду профілю користувача та перегляду всіх його сховищ в цілому	Стратегія виклику лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 5.5), а також в залежності від обраної базової стратегії розвитку (табл 5.15) та стратегії конкурентної поведінки (табл. 5.16) розробляється стратегія позиціонування (табл. 5.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.17 – Визначення стратегії позиціонування

№ з/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Можливість спільно володіти унікальним асетом, проводити аукціони до збору NFT в єдине ціле, зручний та інтуїтивний інтерфейс, перегляд кабінету користувача, перегляд всіх сховищ в системі	Стратегія диференціації	Досвідчена команда, зручний та інтуїтивний дизайн, реалізація фракціонування NFT шляхом переміщенням його до сховища, можливість викупу або проведенню аукціону для викупу, новий функціонал фракціонування NFT асетів для спільного володіння.	Зручний та інтуїтивний дизайн, фракціонування NFT шляхом переміщенням його до сховища, можливість викупу або проведенню аукціону для викупу

Підводячи підсумки для проекта NFT фракталазції для маркетингу була вибрана стратегія диференційованого маркетингу бо проект націлений декілька соціальних груп та потребує розробки окремої стратегії.

Стратегія диференціації була вибрана як базова стратегія розвитку адже вона гарно підходить для програмного продукту з відмінним від інших функціоналом та відповідає очікування і вимогам користувачів. Стратегія конкуретної поведінки була вибрана стратегія виклику лідера адже на ринку вже є декілька сервісів які хоч і опосередковано але надають змогу спільного володіння NFT.

5.5 Розроблення маркетингової програми стартап-проекту

Перше, що слід зробити це зформувати маркетингову концепцію товару які отримають користувачі. Для цього у таблиці 5.18 буде підсумовані результати конкурентноспроможності проекту.

Таблиця 5.18 – Визначення стратегії позиціонування

№ з/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Можливість бути співвласником NFT шляхом фракталізації NFT на дрібні частини, економічна вигідність для власників NFT фракталізувати NFT у нас за рахунок виплати гонорарів кураторам та низьких гонорарів розробникам, зручний та інтуїтивний інтерфейс	Можливість бути співласником NFT та виплата гонрарів кураторам NFT	Можливість бути співласником NFT та отримувати пасивний заробіток від цього, низька гонорар для розробників, зручний та інтуїтивний веб-інтерфейс для взамєодії зі смарт-контрактами

Далі буде представлена трирівнева маркетингова модель продукту у таблиці 5.19

Таблиця 5.19 – Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Інформаційна система для фракталізація NFT «FNFT»</td></tr>
<tr><td rowspan="7">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Спільне володіння NFT</td><td></td><td>ТХ</td></tr>
<tr><td>2. Аукціон для зборки NFT з фракцій</td><td></td><td>ТХ</td></tr>
<tr><td>3. Кабінет користувача</td><td></td><td>ТХ</td></tr>
<tr><td>4. Перегляд всіх створених сховищ системою</td><td></td><td>ТХ</td></tr>
<tr><td colspan="3">Якість: Тестування смарт-контракту та веб-застосунку за документацією</td></tr>
<tr><td colspan="3">Пакування: Відсутнє</td></tr>
<tr><td colspan="3">Марка: Storm «FNFT»</td></tr>
<tr><td rowspan="2">III. Товар із підкріпленням</td><td colspan="3">До продажу: гарантія якості</td></tr>
<tr><td colspan="3">Після продажу: підртримка</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: захистом інтелектуальної власності</td></tr>
</table>

Отже, продукт буде захищений від копіювання зихистом інтелектуальної власності, а код веб-застосунку будуть комерційною таємницею.

Цінові межі ще одна складова яку потрібно визначити для розробки маркетингової програми. Цінові межі також передбачають аналіз ціни на товари аналоги або товари субститути та аналіз рівня доходів цільової групи. Визначення ціновиз мед буде представлене в таблиці 5.20

Таблиця 5.20 – Визначення меж встановлення ціни

№ з/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Безкоштовно	Безкоштовно	Користувачі які входять у цільові групи здебільшого являються середнім та багатим класами	Безкоштовно

Усі платформи, які надають послуги в NFT фракталізації є безкоштовними та лише беруть відсоток від виплачених гонорарів куратору NFT. Проект робити платним є недоцільним так як він буде суперечити одним з базисів web 3 та й інші проекти які надають дану послугу є безкоштовними.

Формування системи збуту відсутнє оскільки смарт-контракти та веб-застосування знаходяться у вільному доступі для користування.

Останньої складовою для створення маркетингової програми є розроблення концепції маркетингової комунікації яка буде спиратись на попередньо нещодавно обрану основу позиціонування та визначену специфіку поведінки користувачів. Концепцію маркетингових комунікацій буде представлено в таблиці 5.21

Таблиця 5.21 – Концепція маркетингових комунікацій

№ з/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Орієнтована в першу чергу на зручність	Контент маркетинг в соціальних мережах, реклама	Новизна підходу смарт-контрактів, зручний веб-застосунок	Ознайомити користувачів які вже знайомі з NFT новими можливостями для співвласного зберігання та заробітку	Показати зручність, простоту та новизну проекту

Оскільки, всі смарт-контракти, які були розгорнуті є безкоштовними та всі веб-застосунки для взаємодії між ними також то користування веб-застосуванням з фракталізації NFT буде безкоштовним. Проект буде отримувати дохід від отримання частки від сплачування гонорарів кураторам NFT які створили сховища для їх фракталізації. Для заохочення споживачів користуватися продуктом буде створена маркетингова компанія яка буде націлена на рекламу через соціальні мережі.

Завданням реклами буде ознайомити користувача з появою нової можливості щодо співвласного володіння NFT та заробітку на цьому.

Висновок до розділу

Проект є рентабельним за рахунок того, що в просторі NFT є проблема зі спільним володінням NFT яку повністю вирішує проект з фракталізації NFT, також він є комерційно вигідним за рахунок взяття частки зі сплачуваних гонорарів кураторів сховищ NFT. Є гарні перспективи щодо впровадження даного проекту оскільки він має функціонал якого поки не мають найпопулярніші проекти та сильні сторони серед схожих реалізацій фракталізації. Впроваджувати проект слід за альтернативою пошуку інвестора та пералельно реалізовувати проект адже web 3 швидко змінюється та можуть з’явитися або нові технології, або нові конкуренти.

Отже, проект є рентабельним, новим та конкурентноспроможним тому є доцільним його подальша імплементація.

ВИСНОВКИ

У результаті виконання магістерської дисертації були розроблені смарт-контракти за допомогою яких виконується фракталізація NFT та веб-застосунок для спрощеного користування смарт-контрактами.

Метою написання магістерської дисертації було спрощення процесу придбання токенів за рахунок їх фракталізації згідно реалізованих смарт-контрактів.

Для цього були розглянуті загальні положення Web 3. Web 3 розширює парадигму Web 2 (читання та запис) та додає ключовий третій пункт – власність. Були розглянуті 3 найбільші блокчейни та вибраний блокчейн – Ethereum на якому будуть розгорнуті смарт-контракти NFT фракталізації. За домогою блокчейну Ethereum були наведені головні складники для NFT фракталізації, а саме акаунт користувача та акаунт смарт-контракту, смарт-контракт та NFT (non-fungible-token).

Після огляду web 3 був розглянутий механізм консенсусу на якому працює блокчейн Ethereum proof-of-stake та концептуально описане сховище сховище, яке буде зберігати в собі NFT. Тобто, для того щоб зробити фракталізацію з NFT потрібно створити сховище, яке буде зберігати поточне NFT. При створенні сховища відбувається не тільки передеча NFT, а й карбування фракцій з цього NFT. Одна фракція NFT це звичайний взаємозамінний токен стандарту ERC-20.

Був розроблений алгоритм аукціону з можливістю викупу токенів та архітектуру сховища для фракталізації NFT та Governance який має змогу змінювати основні параметри аукціону, куратора та адресу користувача який буде получати частину від гонорару куратора за надавання послуг.

Алгоритм аукціону захищений від ставки в останній момент часу та додає додатковий час щоб користувачі змогли відреагувати на зміну стану аукціону.

Для фракталізації NFT були розроблені смарт-контракти сховища, фабрика сховищ, Governance та Initial Vault Proxy на мові Solidity.

Смарт-контракт сховища є головним контрактом в якому зберігається NFT та створюються його фракції, також сховище відповідальне за аукціон та можливість викупу NFT.

Смарт-контракти Vault Factory та Initial Vault Proxy працюють разом для розгортання для дешевого розгортання нових сховищ в блокчейні. Vault Factory додатково зберігає адресу користувача Governance який буде переданий кожному сховищу. Governance смарт-контракт відповідає за параметри для аукціону та адресою на яку будуть виплачуватись гонорари для розробників.

Для полегшення взаємодії зі смарт-контрактами було реалізоване веб-застосування з використанням фреймворку nuxt який є ssr friendly та під капотом використовує один з найпопулярніших фреймворків vue 3. Провайдером між веб-застосуванням та блокчейном виступає один з найпотужніших провайдерів – Alchemy.

Розроблене веб-застосування дозволяє зручно та в декілька кроків фракталізувати NFT та взаємодіяти з ним переглядаючи різноманітну інформацію. К доповненню веб-застосування дозволяє переглядати всі сховища які були створені фабрикою сховищ та профілі користувачів з коллекцією їх фракталізованих NFT.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. NFT “Bored Ape Yacht Club” [Електронний ресурс] – https://opensea.io/assets/ethereum/0xbc4ca0eda7647a8ab7c2061c2e118a18a936f13d/3539
2. Der Kuss розділена на 10 000 фракціонованих NFT – [Електронний ресурс] – https://www.euronews.com/culture/2022/02/14/gustav-klimt-s-the-kiss-selling-as-10-000-nfts-for-valentine-s-day
3. «ділянка землі» біля Snoop Dog за 450 000 доларів [Електронний ресурс] – https://www.rollingstone.com/culture/culture-news/sandbox-decentraland-virtual-land-sales-soar-metaverse-nfts-1267740/
4. Регуляція фракційних NFT як цінних паперів [Електронний ресурус] – https://finance.yahoo.com/news/fractionalized-nfts-were-regulated-securities-161701535.html
5. Blockchain [Електронний ресурс] –https://en.wikipedia.org/wiki/Blockchain
6. Cryptocurrency [Електронний ресурс] – https://en.wikipedia.org/wiki/Cryptocurrency
7. Короткий опис відмін Web 3 [Електронний ресурс] – https://twitter.com/himgajria/status/1266415636789334016
8. Ethereum blockchain [Електронний ресурс] – https://ethereum.org/en/developers/docs/intro-to-ethereum/
9. Smart-contract [Електронний ресурс] – https://ethereum.org/en/developers/docs/smart-contracts/
10. EtherScan [Електронний ресурс] – https://etherscan.io/
11. Аккаунти в блокчейні Ethereum [Електронний ресурс] – https://ethereum.org/en/developers/docs/accounts/
12. WEI [Електронний ресурс] – https://en.bitcoinwiki.org/wiki/WEI
13. Clef для створення аккаунту в блокчейні Ethereum [Електронний ресурс] - https://geth.ethereum.org/docs/clef/introduction
14. Мнемонічна фраза [Електронний ресурс] – https://en.bitcoinwiki.org/wiki/Mnemonic_phrase

15. Отримання ключів від мнемонічної фрази [Електронний ресурс] – https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/keys/#deriving-keys-from-seed
16. Смарт-контракт [Електронний ресурс] – https://ethereum.org/en/developers/docs/smart-contracts/
17. Gas and Fees [Електронний ресурс] – https://ethereum.org/en/developers/docs/gas/
18. NFT (non-fungible token) [Електронний ресурс] – https://ethereum.org/en/nft
19. Proof-of-stake consensus mechanism [Електронний ресурс] – https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/
20. Proof-of-work consensus mechanism [Електронний ресурс] – https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/
21. Fork choice algorithm LMD-GHOST [Електронний ресурс] – https://arxiv.org/pdf/2003.03052.pdf
22. Solidity Language [Електронний ресурс] – https://docs.soliditylang.org/en/v0.8.17/
23. ERC-20 token standard [Електронний ресурс] – https://ethereum.org/en/developers/docs/standards/tokens/erc-20/
24. EIP-7 delegatecall [Електронний ресурс] – https://github.com/ethereum/EIPs/blob/master/EIPS/eip-7.md
25. ERC-721: NFT [Електронний ресурс] – https://ethereum.org/en/developers/docs/standards/tokens/erc-721/
26. Ethers.js library [Електронний ресурс] – https://docs.ethers.io/v5/single-page/#/v5/getting-started/
27. Nuxt framework [Електронний ресурс] – https://nuxt.com/
28. Javascript language [Електронний ресурс] –https://developer.mozilla.org/en-US/docs/Web/JavaScript
29. Vue.js framework [Електронний ресурс] – https://vuejs.org/
30. Document Object Model (DOM) [Електронний ресурс] – https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

31. Pinia storage [Електронний ресурс] – https://pinia.vuejs.org/
32. Typescript language [Електронний ресурс] – https://www.typescriptlang.org/
33. Visual Studio Code IDE [Електронний ресурс] - https://code.visualstudio.com/docs
34. Remix IDE [Електронний ресурс] – https://remix-project.org/
35. Metamask wallet [Електронний ресурс] – https://metamask.io/https://metamask.io/
36. Goerli faucet для отримання перших ETH [Електронний ресурс] – https://goerlifaucet.com/
37. Testnet OpenSea [Електронний ресурс] – https://testnets.opensea.io/
38. Automated Market Maker (AMM) [Електронний ресурс] - https://www.gemini.com/cryptopedia/glossary
39. Uniswap [Електронний ресурс] – https://uniswap.org/
40. Web 3 developer report 2022 [Електронний ресурс] – https://www.alchemy.com/blog/web3-developer-report-q3-2022