

РЕФЕРАТ

Кваліфікаційна робота магістра на тему “Розробка децентралізованої веб-платформи для розміщення NFT та створення крипто-токенів на базі React, Motoko та ICP” Гушпіта Назарія Руслановича. Тернопільський національний технічний університет імені Івана Пулюя, кафедра програмної інженерії, спеціальність 121, група СПМ–61, Тернопіль, 2023. С. – 69, рис. – 57 , табл. – 1, додат. – 2, бібліогр. – 13.

Ця робота присвячена проектуванню та розробці інноваційної веб-платформи для створення та обміну NFT токенів.

У роботі розглядаються основні концепції та технології, що стоять за NFT та крипто-токенами, а також детально описуються використовувані технології: React для розробки клієнтської частини, Motoko для розробки серверної частини та ICP (Internet Computer Protocol) для забезпечення децентралізації.

Практична частина роботи включає проектування та розробку веб-платформи, що дозволяє користувачам створювати, купувати та продавати NFT та крипто-токени. Результати тестування підтверджують функціональність та ефективність розробленої системи.

Ця робота вносить важливий вклад у розвиток децентралізованих веб-платформ та може слугувати основою для подальших досліджень у цій галузі.

Ключові слова: DAPP, NFT, КРИПТО-ТОКЕНИ, REACT, МОТОКО, INTERNET COMPUTER, ICP, BLOCKCHAIN.

ANNOTATION

Master's qualification thesis on the topic "Development of a decentralized web platform for hosting NFTs and creating crypto-tokens based on React, Motoko and ICP" by Hushpit Nazarii Ruslanovych. Ternopil Ivan Puluj National Technical University, department of software engineering, specialty 121, group SPM-61, Ternopil, 2023. C. – 69, fig. – 57, tab. – 1, add. – 2, bibliography – 13.

This work is dedicated to the design and development of an innovative web-based platform for the creation and exchange of NFT (non-fungible tokens) and crypto tokens.

The paper examines the basic concepts and technologies behind NFTs and crypto tokens, and also describes in detail the technologies used: React for client-side development, Motoko for back-end development, and ICP (Internet Computer Protocol) for decentralization.

The practical part of the job involves designing and developing a web platform that allows users to create, buy and sell NFTs and crypto tokens. The test results confirm the functionality and efficiency of the developed system.

This work has an important contribution to the development of decentralized web platforms and can serve as a basis for further research in this field.

Keywords: DAPP, NFT, CRYPTO TOKENS, REACT, MOTOKO, INTERNET COMPUTER, ICP, BLOCKCHAIN.

ЗМІСТ

ВСТУП.....	4
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГИ.....	5
1.1. Актуальність теми.....	5
1.2. Мета та завдання розробки.....	6
1.3. Вимоги до платформи.....	7
1.4. Методи дослідження.....	9
1.4.1. Аналітичний метод.....	9
1.4.2. Експериментальний метод.....	10
2. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ДЕЦЕНТРАЛІЗОВАНОГО МАРКЕТПЛЕЙСУ.....	12
2.1. Огляд технологій.....	12
2.1.1. Огляд React.....	12
2.1.2. Огляд Moko та ICP.....	14
2.2. Принципи роботи NFT та крипто-токенів.....	18
2.3. Аналіз сучасних децентралізованих веб-платформ.....	22
3. ПРОЕКТУВАННЯ МАРКЕТПЛЕЙСУ.....	26
3.1. Визначення вимог до архітектури.....	26
3.2. Розробка архітектури платформи.....	26
3.3. Налаштування залежностей для розробки платформи.....	28
4. ВПРОВАДЖЕННЯ ТА ТЕСТУВАННЯ ВЕБ-ПЛАТФОРМИ.....	38
4.1. Проведення тестування веб-платформи.....	38
4.2. Робота користувача з платформою.....	42
5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	53
5.1. Охорона праці.....	53
5.2. Безпека в надзвичайних ситуаціях.....	55
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

ВСТУП

Поява технології блокчейн зробила революцію в різних секторах, включаючи фінанси, ланцюг поставок і цифрове мистецтво. Одним із найважливіших нововведень у цьому просторі є концепція невзаємозамінних токенів (NFT) і криптотокенів. NFT, унікальні цифрові активи, що представляють право власності на певний предмет або частину контенту, набули величезної популярності, особливо у світі цифрового мистецтва. Одночасно крипто-токени стали фундаментальною частиною децентралізованих фінансів, створюючи інноваційні фінансові продукти та послуги.

Ця робота присвячена розробці децентралізованої веб-платформи для розміщення NFT токенів та їх створення. Платформа використовує фреймворк React, популярної мови програмування JavaScript для створення користувальницьких інтерфейсів, і Motoko, мови програмування, яка використовує так звані каністри(вчислювальні блоки) в якості back-end, розміщення даних у блокчейні та взаємодії з Internet Computer. ICP надає блокчейн-платформу, яка працює на швидкості Інтернету з необмеженою пропускною здатністю. Крім того, платформа інтегрує Web3, набір бібліотек, які дозволяють розробникам взаємодіяти з локальним або віддаленим вузлом Ethereum за допомогою HTTP, IPC або WebSocket.

Метою цього проекту є дослідження потенціалу цих технологій у створенні зручної, ефективної та безпечної платформи для майнінгу для купівлі та продажу NFT токенів. Проект заглиблюватиметься в технічні аспекти цих технологій, їх інтеграцію та виклики, які виникають у процесі розробки.

Звіт структурований наступним чином: Розділ 1 описує актуальність теми та мету дослідження; Розділ 2 містить теоретичні основи використовуваних технологій; Розділ 3 обговорює проектування платформи; У розділі 4 представлено тестування та розгортання платформи; і, нарешті, розділ 5 завершує звіт охороною праці та безпекою життєдіяльності.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГИ

1.1. Актуальність теми

У сучасному світі технології швидко розвиваються, а інтернет стає все більш децентралізованим. Одним з найбільш перспективних напрямків цього розвитку є технологія блокчейн та пов'язані з нею Non-Fungible Tokens (NFTs). NFTs – це унікальні цифрові активи, які можуть представляти власність конкретного предмета або контенту. Вони знайшли широке застосування в цифровому мистецтві, іграх, колекціонуванні та інших сферах.

Розробка децентралізованої веб-платформи для розміщення, майнінгу та продажу NFT токенів на базі React, Motoko та ICP є актуальною темою, оскільки вона відповідає сучасним тенденціям розвитку технологій та потребам ринку. За допомогою бібліотеки React буде створено користувацький інтерфейс. Motoko – це мова програмування, розроблена для Internet Computer Protocol (ICP), який надає платформу блокчейну, що працює з високою швидкістю порівняно з конкурентами та необмеженою ємністю.

Створення додатків за допомогою Internet Computer (ICP) набуває все більшої актуальності в сучасному світі. ICP – це глобальний комп'ютер, який працює в інтернеті та дозволяє створювати веб-додатки та сервіси без потреби в серверах, базах даних, файрволах та інших компонентах традиційної архітектури.

Ось декілька причин, чому створення додатків за допомогою ICP є актуальним:

1. Висока продуктивність: ICP пропонує високу продуктивність та швидкість, порівняно з традиційними веб-додатками.

2. Безпека: ICP надає високий рівень безпеки, оскільки він використовує криптографічні протоколи для захисту даних.

3. Децентралізація: ICP дозволяє створювати децентралізовані додатки, що зменшує залежність від централізованих серверів та підвищує надійність.

4. Масштабованість: Завдяки своїй архітектурі, ІСР дозволяє легко масштабувати додатки, що робить його ідеальним для великих проєктів.

5. Відкритість: ІСР пропонує відкриту платформу, що сприяє інноваціям та розвитку.

Таким чином, створення додатків за допомогою ІСР є актуальним та перспективним напрямком в сфері веб-розробки. Воно відкриває нові можливості для розробників та користувачів, сприяючи розвитку більш безпечного, ефективного та децентралізованого інтернету.

Така платформа може забезпечити безпечне, прозоре та ефективне середовище для створення, розміщення, майнінгу та продажу NFTs. Вона може відкрити нові можливості для художників, геймерів, колекціонерів та інших користувачів, а також сприяти розвитку цифрової економіки.

Таким чином, розробка такої платформи є важливою та актуальною задачею, яка може мати значний вплив на сферу цифрових технологій та економіки в цілому. Дослідження цієї теми може допомогти зрозуміти основні принципи роботи NFTs, особливості їх створення та використання, а також потенційні переваги та виклики, пов'язані з використанням цих технологій.

1.2. Мета та завдання розробки

Мета: Головною метою цієї роботи є створення децентралізованої веб-платформи для розміщення, майнінгу та продажу NFT токенів на базі React, Motoko та ІСР. Ця платформа має на меті надати користувачам безпечне, прозоре та ефективне середовище для взаємодії з NFTs.

Завдання:

1. Дослідження: Провести дослідження щодо поточного стану NFT ринку, вивчити основні технології, що використовуються для створення NFT платформ.

2. Проектування: Розробити архітектуру платформи, визначити основні компоненти та їх взаємодію.
3. Розробка: Реалізувати платформу, використовуючи React для фронтенду, Motoko для бекенду, та ICP для блокчейну.
4. Тестування: Провести комплексне тестування платформи для забезпечення її надійності та ефективності.
5. Впровадження: Запустити платформу та забезпечити її постійну підтримку та оновлення.

Ці завдання спрямовані на досягнення поставленої мети та вирішення актуальних проблем, пов'язаних з розробкою децентралізованих веб-платформ для розміщення, майнінгу та продажу NFT токенів.

1.3. Вимоги до платформи

Функціональні вимоги визначають, що повинна робити система. Вони описують функції, які виявляє користувач [1]. Наприклад, функція може бути очевидною, як велика кнопка «Додати до кошика», або менш очевидною, як правильний розрахунок податку з продажу для онлайн-покупки користувача.

Нефункціональні вимоги визначають, як система повинна працювати всередині. Вони стосуються атрибутів якості системи, таких як продуктивність, безпека, масштабованість та інше. Наприклад, нефункціональна вимога може вказувати, що веб-сайт повинен завантажуватися за 2 секунди або менше, або що система повинна мати здатність обслуговувати до 10 000 користувачів одночасно.

Діаграма базових нефункціональних вимог зображена на рисунку 1.1.

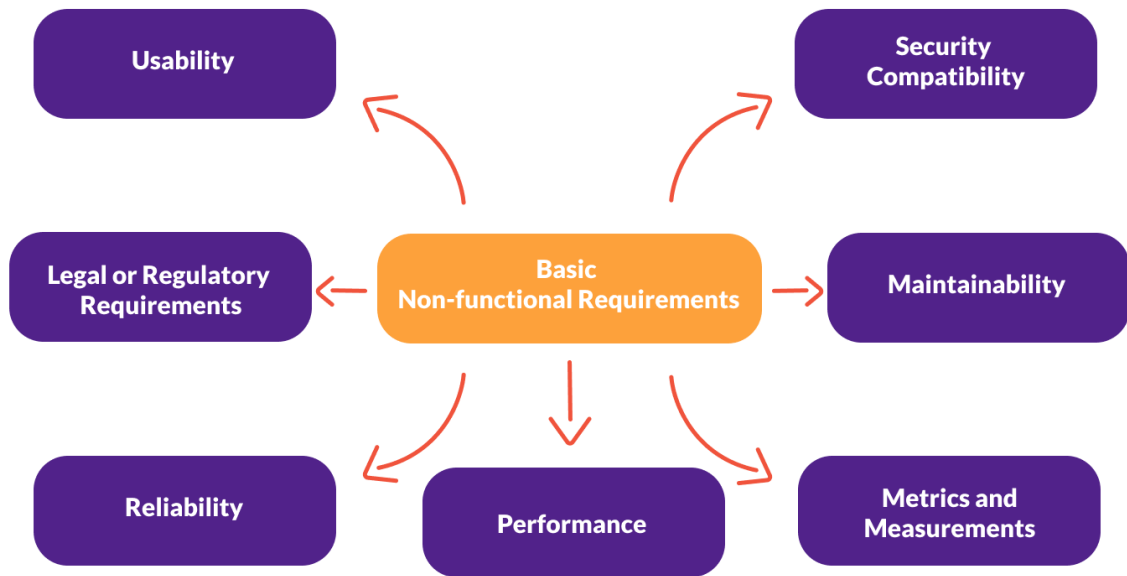


Рисунок 1.1 – мінімальні нефункціональні вимоги

Функціональні вимоги:

1. Створення NFT токенів: повинна дозволяти користувачам створювати NFT токени, включаючи завантаження цифрового контенту, встановлення атрибутів та властивостей токена.
2. Розміщення NFT токенів: має надавати можливість розміщення створених NFT токенів для перегляду іншими користувачами.
3. Майнінг NFT токенів: має функціонал для майнінгу NFT токенів.
4. Продаж NFT: платформа повинна дозволяти користувачам продавати свої NFT токени іншим користувачам через безпечну систему торгівлі.
5. Пошук та фільтрація: платформа повинна мати потужні інструменти пошуку та фільтрації для зручного пошуку NFT токенів.

Нефункціональні вимоги:

1. Безпека: платформа повинна забезпечувати безпеку транзакцій та даних користувачів.

2. Продуктивність: платформа повинна забезпечувати високу продуктивність та швидкість обробки запитів.

3. Надійність: платформа повинна бути надійною та стабільною, забезпечуючи безперебійну роботу сервісу.

4. Масштабованість: платформа повинна бути готовою до масштабування для обслуговування великої кількості користувачів.

5. Інтуїтивно зрозумілий інтерфейс: платформа повинна мати інтуїтивно зрозумілий інтерфейс, який зручно використовувати.

Ці вимоги є важливими для розробки ефективної, безпечної та користувацько-орієнтованої платформи для розміщення, майнінгу та продажу NFT токенів. Проте, вони можуть бути модифіковані або доповнені в залежності від конкретних потреб проекту.

1.4. Методи дослідження

1.4.1. Аналітичний метод

У цій роботі використовуються два методи дослідження для досягнення поставленої мети. Ось декілька з них:

Цей метод включає в себе аналіз наявної літератури, досліджень та ресурсів, щоб зрозуміти поточний стан технологій, які використовуються для створення NFT платформ. Це допомагає визначити найкращі практики та ідентифікувати можливі виклики. Аналітичний метод дослідження – це процес збору, аналізу та інтерпретації інформації для висновків та досягнення висновків[2]. Він включає в себе критичне мислення та оцінку фактів та інформації, що стосуються проведеного дослідження. Приклад життєвого циклу аналітичного методу (див. рис. 1.2).

Lifecycle Approach to Analytical Methods

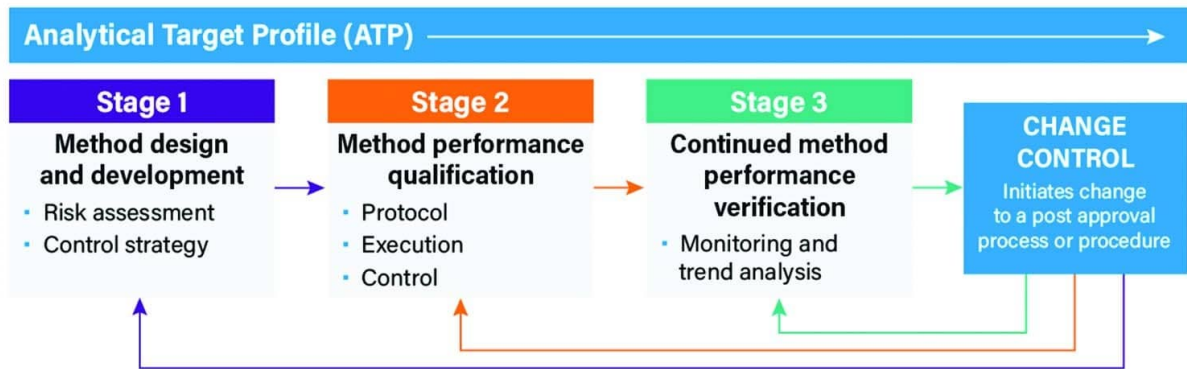


Рисунок 1.2 – життєвий цикл аналітичного методу дослідження

Пов'язуючи це з даною темою існує декілька кроків використання: визначення проблеми, потрібно визначити ключові проблеми, які намагаєтеся вирішити даною платформою. Це може включати питання зберігання даних, безпеки та швидкості транзакцій. Збір даних: зберегти всю необхідну інформацію та дані, які вам потрібні для розробки платформи.

Це може включати вивчення існуючих NFT платформ, технологій blockchain, та інших технологій, які плануєте використовувати. Аналіз даних: проаналізувати зібрані дані, використовуючи різні аналітичні методи. Це може включати вивчення тенденцій, шаблонів, кореляцій між різними змінними тощо.

На основі аналізу, можна зробити висновки про те, якнайкраще розробити платформу. Це може включати визначення найкращих технологій для використання, оптимальних стратегій зберігання даних, методів забезпечення безпеки тощо. Після розробки платформи, потрібно перевірити гіпотези, проводячи тести та аналізуючи. Цей процес допоможе систематично підходити до розробки та забезпечити ретельне вивчення всіх важливих аспектів.

1.4.2. Експериментальний метод

Цей метод включає в себе розробку прототипів та проведення експериментів для перевірки різних аспектів системи, таких як продуктивність, безпека та користувацький досвід. Він включає в себе два набори змінних: перший набір діє як константа, яку використовують для вимірювання різниці другого набору. Пов'язуючи це з темою “Розробка децентралізованої веб-платформи для розміщення, майнінгу та продажу NFT токенів на базі React, Motoko та ICP”, ось як ви можна застосувати експериментальний метод: формулювання гіпотези про те, як платформа може вирішити конкретні проблеми в області NFT.

Створення експерименту: розроблення експерименту, що включає в себе створення прототипу вашої платформи та визначення ключових метрик для вимірювання її ефективності. Проведення експерименту: провести експеримент, використовуючи прототип, та збирати дані про його ефективність.

Аналіз результатів: після проведення експерименту, можна аналізувати зібрані дані та визначити, чи платформа ефективно вирішує поставлені проблеми.

На основі аналізу, було зроблено висновки про ефективність платформи та внесення необхідних покращень.

2. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ДЕЦЕНТРАЛІЗОВАНОГО МАРКЕТПЛЕЙСУ

2.1. Огляд технологій

2.1.1. Огляд React

React: революційна бібліотека JavaScript для створення інтерфейсів користувача. React – це бібліотека JavaScript з відкритим вихідним кодом для створення інтерфейсів користувача, переважно для односторінкових програм. Він використовується для обробки шару перегляду для веб-програм і мобільних додатків. Логотип фреймворку (див. рис. 2.1).

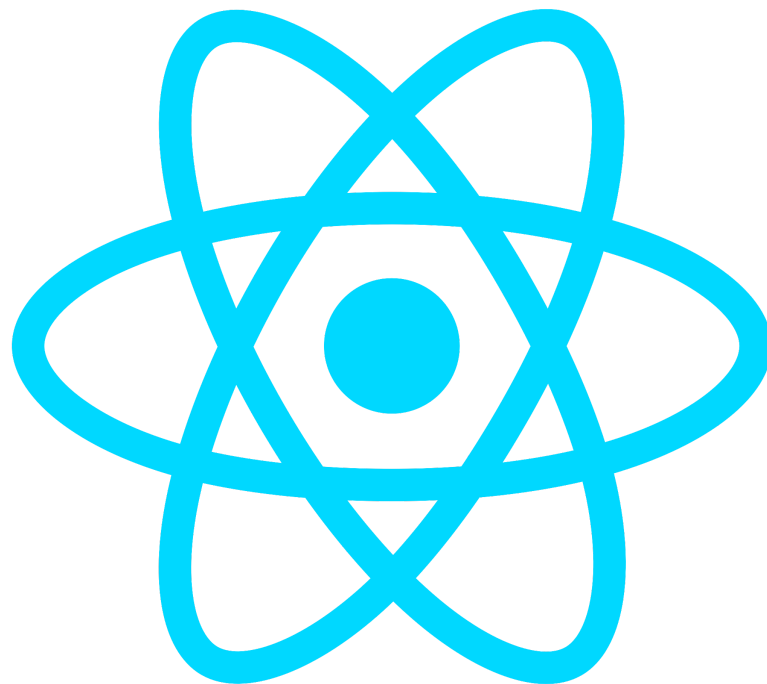


Рисунок 2.1 – Логотип фреймворку React

React був створений Джорданом Волке, інженером-програмістом у Facebook, і вперше був розгорнутий у стрічці новин Facebook у 2011 році та в Instagram у 2012 році[3].

Ключові особливості React. React використовує JSX, синтаксичне розширення для JavaScript, яке дозволяє вам писати HTML-подібний код у файлах

JavaScript. Хоча використовувати JSX у React не обов'язково, це рекомендовано, оскільки це спрощує код і покращує читабельність.

Компоненти. React побудовано навколо компонентів. Компонент – це самостійна частина коду, яка керує власним вмістом, представленням і поведінкою. Вони схожі на функції JavaScript, вони приймають довільні вхідні дані (так звані «реквізити») і повертають елементи React, які описують те, що має з'явитися на екрані Приклад компонентів додатку (див. рис. 2.2).

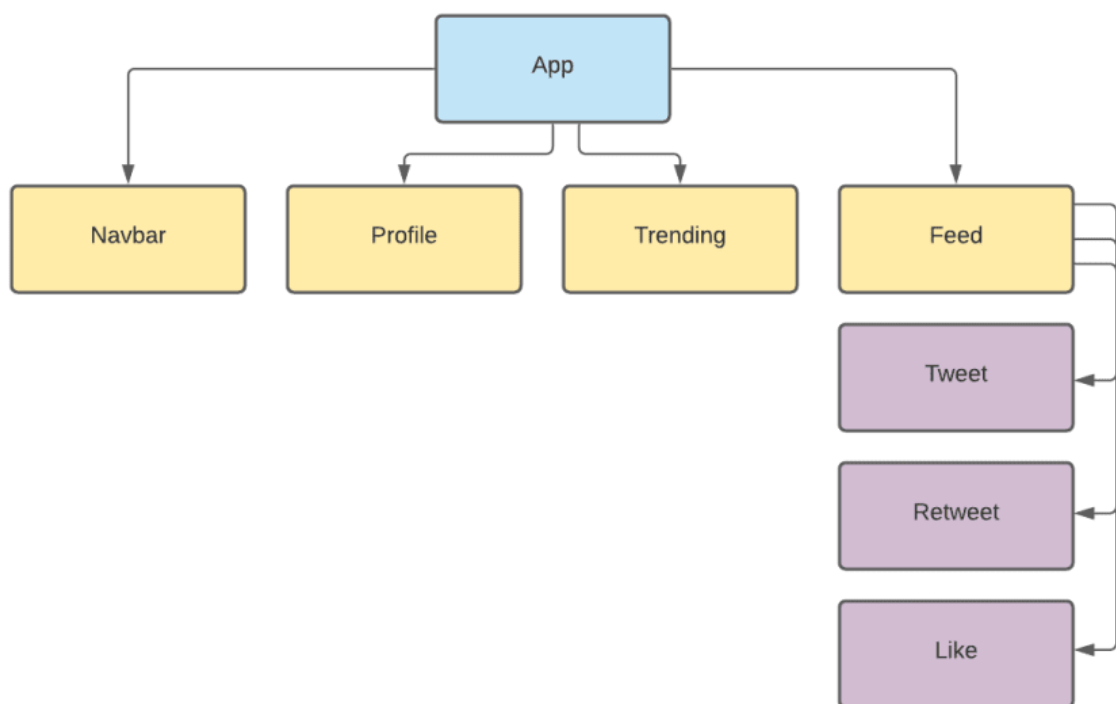


Рисунок 2.2 – діаграма React компонентів типового додатку

Одностороннє прив'язування даних і потік. React реалізує односторонній потік даних, що полегшує міркування про додаток. Flux – це шаблон, який допомагає зберігати дані односпрямований. Це означає, що під час розробки програми React часто вкладає дочірні компоненти в батьківські компоненти.

Віртуальний DOM. React використовує віртуальну DOM для підвищення продуктивності. Віртуальний DOM є представленням фактичного DOM у пам'яті. Коли стан компонента змінюється, React спочатку оновлює віртуальний DOM.

Потім він порівнює новий віртуальний DOM зі старим і вносить зміни в реальний DOM, щоб відповідати новому віртуальному DOM (див. рис. 2.3).

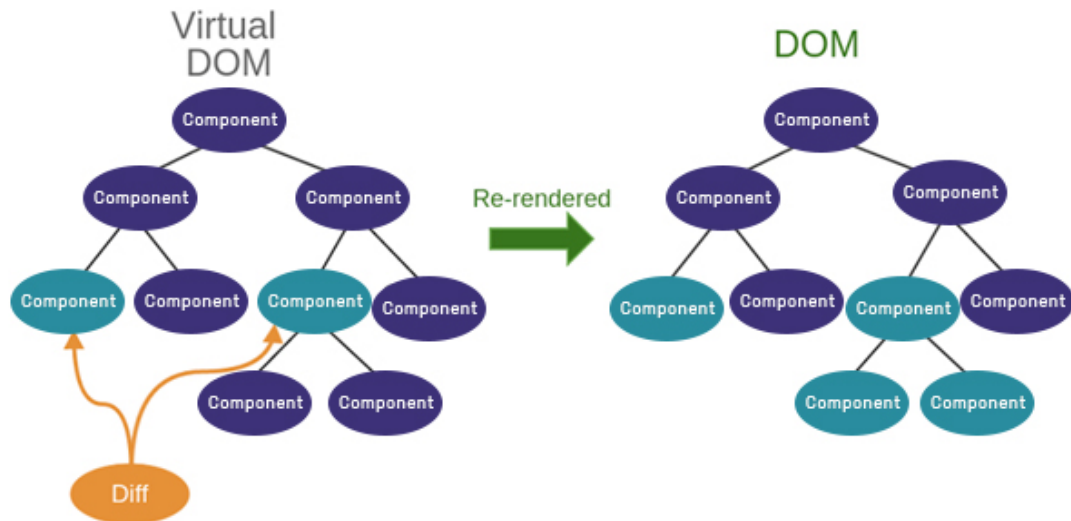


Рисунок 2.3 – віртуальний DOM у React

React революціонізував спосіб створення веб-додатків розробниками. Акцент на багаторазових компонентах, односторонньому потоці даних і віртуальному DOM не тільки покращив продуктивність, але й полегшив розуміння та підтримку програм. Оскільки React продовжує розвиватися та зростати в популярності, стає зрозуміло, що у нього світле майбутнє у світі веб-розробки.

Використання React з Motoko та Internet Computer.

Для поєднання та роботи з цими технологіями потрібно наступне:

- платформа Node.js і менеджер пакетів npm;
- DFINITY Canister SDK – це набір для розробки програмного забезпечення, який використовується для створення смарт-контрактів контейнера та керування ними в блокчейні ICP;
- провайдер мережі Internet Computer (наприклад, dfx).

2.1.2. Огляд Motoko та ICP

У сфері децентралізованої веб-розробки дві технології змінили правила гри: Motoko, мова програмування, розроблена для Інтернет-комп'ютера, і сам Інтернет-комп'ютерний протокол (ICP). Ці технології пропонують новий підхід до створення безпечних, масштабованих і ефективних децентралізованих програм.

Motoko – це статично типізована мова програмування, розроблена DFINITY. Вона призначена для безпосередньої підтримки моделі програмування Інтернет-комп'ютера, що полегшує ефективне створення програм і використання функцій Інтернет-комп'ютера[4].

Надійна статична типізація Motoko допомагає виявляти помилки під час компіляції, а не під час виконання, підвищуючи надійність програм. За допомогою Motoko можна створювати каністри. Каністра – це автономний об'єкт, який повністю інкапсулює свій стан і спілкується з іншими акторами тільки через асинхронні повідомлення.

Каністер включає в себе програму та стан. Він схожий на контейнер (див. рис. 2.4), оскільки обидва розгортаються як програмна одиниця, що містить скомпільований код та залежності для додатка або сервісу. Однак каністра також зберігає інформацію про поточний стан програми.

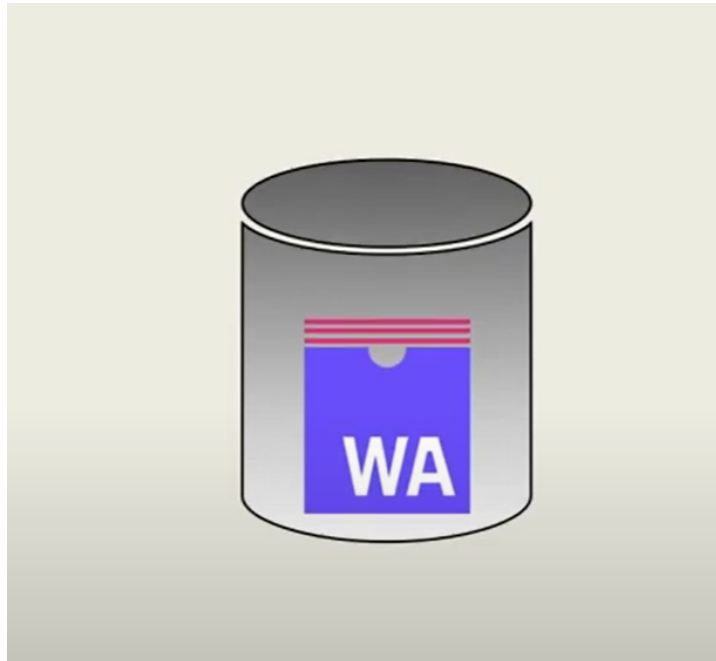


Рисунок 2.4 – візуальне представлення каністри в Internet Computer

Важливо зазначити, що каністра в Motoko виражається як актор. Наприклад, цей код визначає актора-Counter зі станом:

```
actor Counter {
  var value = 0;
  public func inc() : async Nat {
    value += 1;
    return value;
  };
}
```

Функція `inc()` що знаходиться в прикладі коду зверху може бути викликана цим та іншими акторами, щоб оновити та прочитати поточний стан його приватного поля `value`.

У контексті даної дипломної роботи, буде створено каністру NFT яка зберігатиме інформацію власника, назву, геттери та сеттери для роботи з NFT токеном, а також каністри використані для створення смарт-контрактів, які керують створенням, торгівлею та управлінням власністю NFT-токенів[5]. Кожна каністра матиме свою власну роль, наприклад, одна каністра буде відповідальна за майнінг NFT, а інша – за їх продаж.

Каністри в Motoko та Internet Computer Protocol (ICP) тісно пов'язані з WebAssembly (Wasm). WebAssembly – це компактний бінарний формат для

упаковки коду в портативну ціль компіляції. Він використовує свою сумісність з JavaScript, C++ та Rust, щоб допомогти розробникам розгортати клієнт-серверні веб-додатки.

У контексті ICP, каністра – це об’єкт, який містить код WebAssembly та стан, які разом формують програму (див. рис. 2.5). Коли розгортається каністра на Internet Computer, код WebAssembly каністер виконується на вузлах мережі.

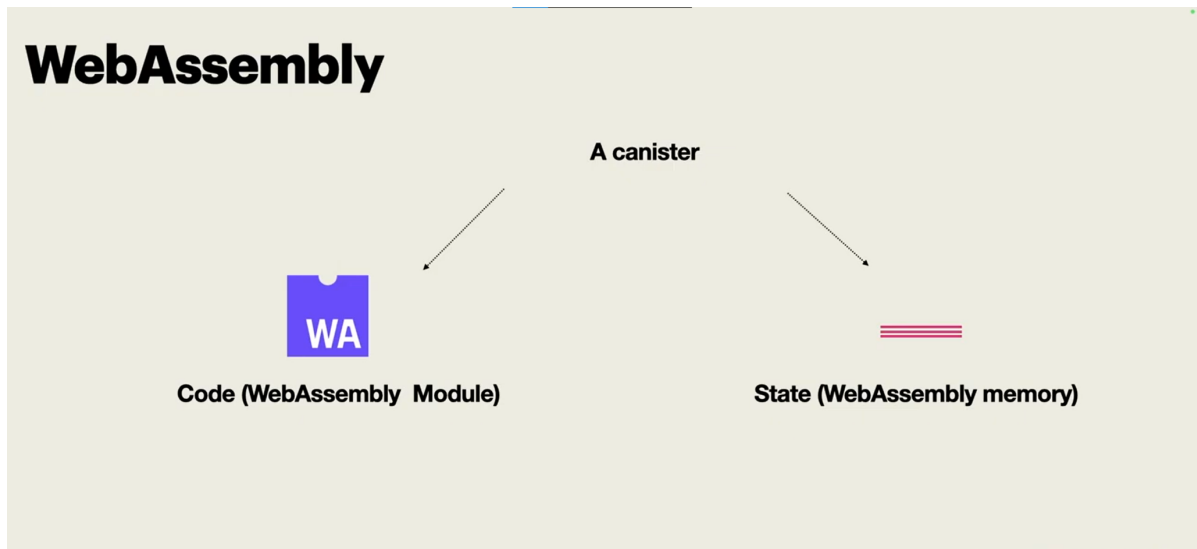


Рисунок 2.5 – бінарний формат WebAssembly в канстрі

Таким чином, каністри використовують WebAssembly як основу для своєї реалізації. Це дозволяє каністрам використовувати всі переваги WebAssembly, такі як швидкість, безпека та портативність, для створення ефективних децентралізованих додатків[6].

Це означає, що WebAssembly буде використано для написання ефективного коду для NFT маркетплейсу. Motoko саме й буде використано для написання коду, який потім компілюється в WebAssembly, і розгортати цей код як каністр на Internet Computer.

Internet Computer.

Інтернет-комп'ютерний протокол (ICP) — це революційний протокол, розроблений DFINITY, який дозволяє розробникам створювати веб-сайти, корпоративні ІТ-системи та інтернет-сервіси, встановлюючи свій код

безпосередньо в загальнодоступному Інтернеті[7]. ICP робить це за допомогою мережі незалежних центрів обробки даних, на яких працює протокол, створюючи децентралізований «Інтернет-комп'ютер».

ICP надає платформу на основі блокчейну, яка є безпечною, масштабованою та керованою децентралізованою мережею. Це робить його ідеальним рішенням для створення платформи для токенів NFT, оскільки він забезпечує необхідну інфраструктуру для децентралізованого управління цифровими активами.

Motoko та ICP стали міцною основою для NFT маркетплейсу. Функції Motoko дозволяють створювати надійні та ефективні смарт-контракти, тоді як ICP забезпечує децентралізовану інфраструктуру, необхідну для безпечного та масштабованого ринку NFT[8]. Разом вони являють собою потужний інструментарій для створення децентралізованих програм майбутнього.

2.2. Принципи роботи NFT та крипто-токенів

NFT – це невзаємозамінний токен, тобто це унікальний актив, який неможливо скопіювати та легко замінити. NFT діє як цифровий сертифікат власності – на матеріальний або нематеріальний актив – який записаний у блокчейні. NFT найчастіше пов'язані з цифровим контентом, таким як комп'ютерне мистецтво чи відеокліпи.

Унікальні властивості NFT – це те, що робить їх особливими та складними для обміну з подібними товарами чи активами. Фіатна валюта, як і монети та долари США, є взаємозамінною, тобто її можна обміняти або замінити іншим активом і зберігати ту саму вартість. NFT не можуть.

Про NFT можна думати як про колекціонування цифрового мистецтва. Традиційно колекціонування творів мистецтва має сенс, оскільки існує лише одна версія кожної картини. Окремі витвори мистецтва є особливими та цінними, тому

що їх у світі лише один екземпляр. Дефіцит додає цінності, і потрібно ставитися до нього обережно.

Принцип дефіциту також можна застосувати до світу цифрового мистецтва. Як пояснив у NFT Now Кевін Маккой, цифровий художник, який створив першу NFT у 2014 році, «у мене виникла ідея використовувати технологію блокчейн для створення незмінного походження цифрових зображень і права власності».

NFT купуються та продаються завдяки актам власності (див. рис. 2.6). Документ про право власності заблоковано за технологією блокчейн у реєстрі, який оновлюється в режимі реального часу, і його майже неможливо підробити, маніпулювати чи зламати.

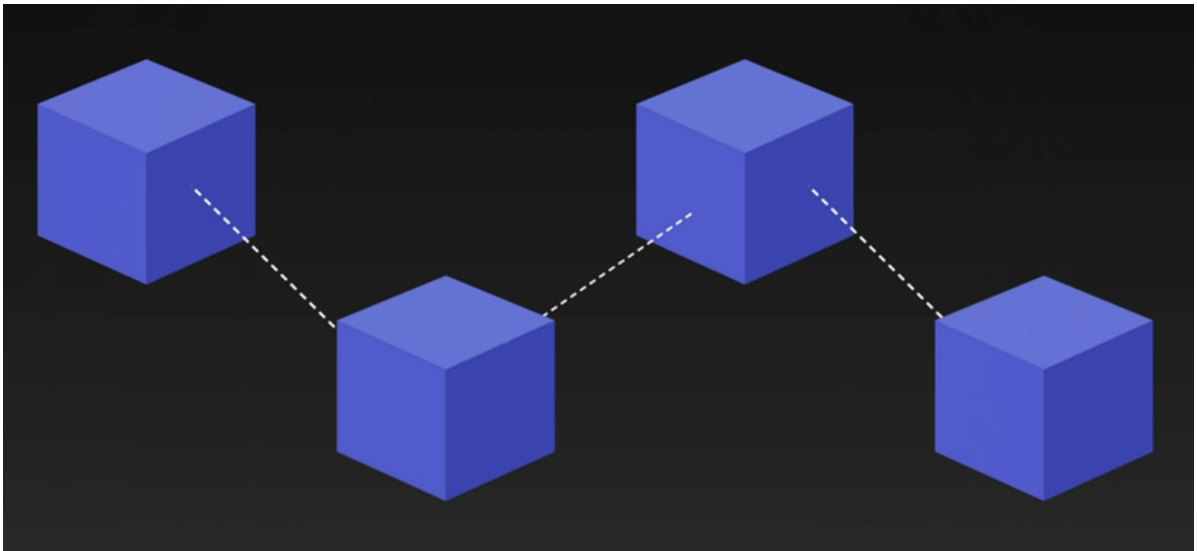


Рисунок 2.6 – документ про право власності NFT, закріплений на блокчейні.

Подібно до того, як біткойн має грошову цінність на основі чогось абстрактного (без фізичних об'єктів), цифрове зображення NFT має цінність, незважаючи на те, що воно нематеріальне. Деякі люди досі не переконані, але тепер мільярди доларів щороку переходять із рук у руки завдяки NFT, тобто вони мають суттєвий вплив на реальний світ – вірити в них чи ні.

NFT (незамінні токени) створюються шляхом «карбування», де NFT присвоюється унікальний ідентифікатор. Ідентифікатори NFT зберігаються в реєстрі, відомому як блокчейн. Ця книга є особливою, оскільки вона оновлюється в режимі реального часу, і її майже неможливо підробити чи зламати.

NFT належить комусь, тому що так зазначено в бухгалтерській книзі, що є остаточним доказом. Крім того, цифровий файл має спеціальний ключ, що

зберігається в його метаданих, який ідентифікує його, тому право власності є незаперечним. Мільйон людей можуть зробити копію онлайн-контенту, який захищає NFT, але лише одна особа може претендувати на право власності на NFT(див. рис. 2.7).

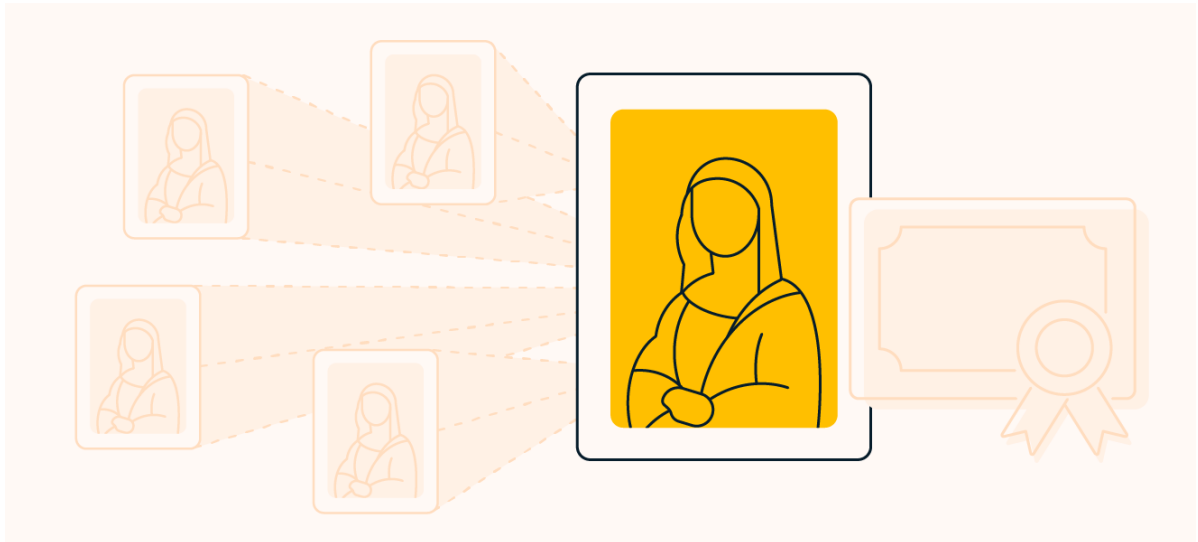


Рисунок 2.7 – може бути тільки один сертифікований власник на NFT токен

Блокчейн працює завдяки алгоритму – кожен новий запис визначається надскладною математичною задачею, яку важко було б вирішити суперкомп'ютерам. Це робить практично неможливим для хакерів маніпулювання блокчейном. Більше того, немає центрального органу, який би контролював блокчейн, тому люди довіряють його дифузній, децентралізованій природі.

Порівняно з багатьма технологіями, NFT все ще перебувають у зародковому стані. Очікується, що незабаром NFT будуть використовуватися для практичних цілей, таких як автентифікація продуктів (ліки, продукти харчування) або інтелектуальної власності (авторські права, патенти, торгові марки).

Наразі деякі з найвідоміших прикладів NFT:

- цифрове мистецтво;
- GIF-файли;
- предмети колекціонування;
- відео;
- нариси;
- ігри;

- віртуальні аватари (і їх одяг);
- музика;
- твіти;
- цифрова власність або нерухомість.

Торговий майданчик NFT NBA Top Shot дозволяє вам володіти грою відомих баскетболістів. Джессі Шварц купив відео данка Леброна Джеймса за 208 000 доларів у січні 2021 року. Хоча будь-хто ще може завантажити та переглянути це відео, він не є його власником і не може на ньому заробити. Оскільки відео було продано за таку ціну, воно придбало таку ж цінність, як і більш традиційне мистецтво.

Якщо говорити про мистецтво, то цифрові твори мистецтва є одним із основних напрямків використання NFT. Одним із найвідоміших прикладів є «Everydays: The First 5000 Days» Beeple, який у 2021 році було продано за 69 мільйонів доларів.

Практично все можна перетворити на NFT. Джек Дорсі, колишній генеральний директор Twitter, продав свій перший в історії твіт як NFT за понад 2,9 мільйона доларів.

NFT мають значущі реальні програми, які можуть змінити бізнес або професійне життя людини. NFT значно покращили можливості творців цифрового контенту та художників монетизувати свою роботу.

Ось погляд на різні галузі, де використовуються NFT, і яку вигоду від цього отримали художники та інші (див. рис. 2.8).

How are NFTs used?

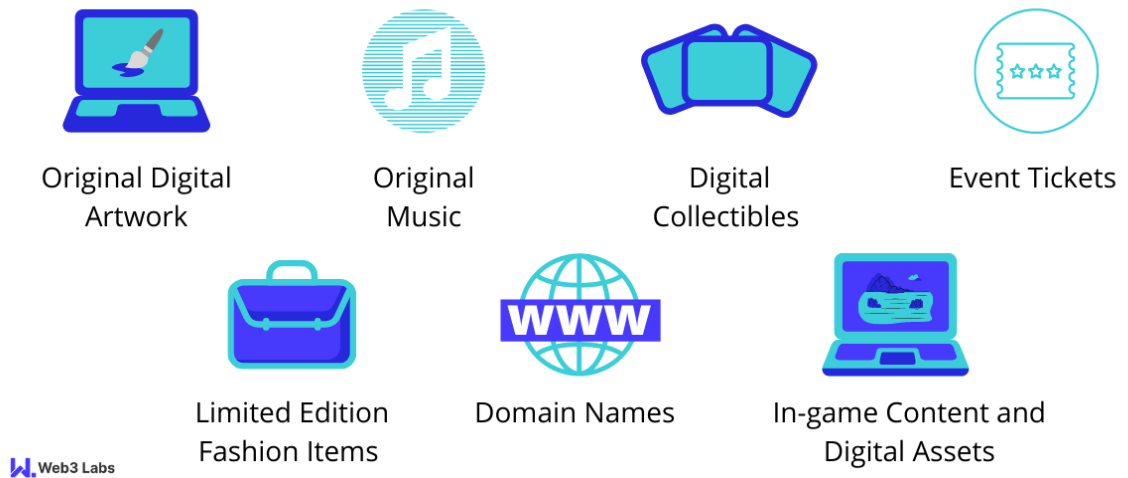


Рисунок 2.8 – сфери в яких використовуються NFT

2.3. Аналіз сучасних децентралізованих веб-платформ

NFT (Non-Fungible Tokens) – це токени, які демонструють право власності на цифрове творіння. Ці твори можуть включати ілюстрації, відео, музику тощо. Ці неймовірні творіння базуються на блокчейні, тій самій технології, яка підтримує такі популярні криптовалюти, як біткойн та ефіріум.

Але куди звернутися для отримання NFT? На це запитання буде надано таблицю представлення найкращих ринків NFT для купівлі та продажу NFT.

Таблиця 2.1 – Список найкращий маркетплейсів по продажах NFT токенів станом на 2023 рік.

Маркетплейс	Кількість NFT	Типи NFT	Криптовалюти к-сть.	Блокчейн
Opensea	80m+	Арт, Музика	150+	Ethereum, Polygon, Solana

Продовження таблиці 2.1

Rarible	300,000+	Арт	ETH, XTZ, FLOW, MATIC	Ethereum, Solana, Tezos, Flow, ImmutableX, Polygon
Binance	100,000+	Арт, Музика	BNB, BUSD, ETH	Ethereum
NiftyGateway	N/A	Арт, Музика	ETH	Ethereum
Gamestop	N/A	Колекційні персонажі	N/A	Ethereum
SuperRare	39,000	Арт, музика, рисунки	ETH	Ethereum
Coinbase NFT	100,000+	Арт	ETH	Ethereum
Larva Labs	10,000	Колекційні персонажі	Ethereum	Ethereum

Купівля та продаж NFT не повинні здійснюватися на платформі, про яку можна почути від сторонніх осіб. Потрібно використовувати законні, надійні та безпечні ринкові майданчики. Найкращі ринки NFT, якими можна скористатися:

2.3.1. OpenSea

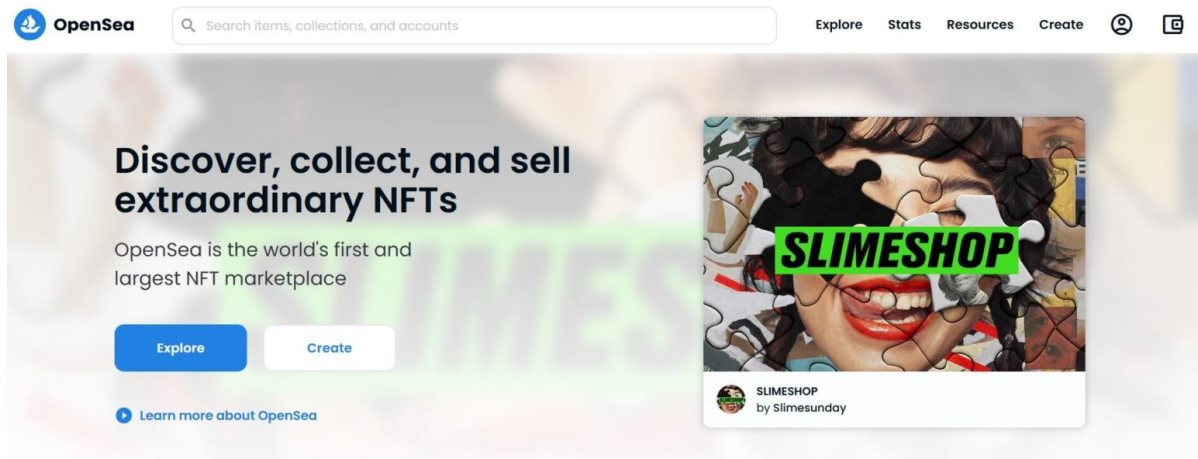


Рисунок 2.9 – головна сторінка маркетплейсу OpenSea

Можливо, найвідоміший ринок NFT, OpenSea — чудовий варіант, якщо ви тільки починаєте працювати з NFT. Це також чудовий вибір, якщо ви хочете досліджувати щось більше, ніж просто твори мистецтва NFT. OpenSea також пропонує доменні імена, віртуальні світи, колекційні картки та спортивні предмети колекціонування для тих, хто цікавиться не лише цифровими зображеннями.

OpenSea було запущено в 2017 році як ринок Ethereum, але з тих пір він розширився для підтримки USDC і Solana. Важливо пам'ятати, що є можливість використовувати фіатну валюту (євро, фунти, долари США тощо) на OpenSea. Однак почати швидко і легко. Потрібно лише кілька хвилин, щоб почати переглядати та купувати на ринку OpenSea.

Загалом, OpenSea — це простий у використанні ринок NFT, який чудово підходить для покупців і продавців усіх рівнів. У вас буде багато доступних варіантів незалежно від того, який тип NFT ви шукаєте.

2.3.2. Rarible

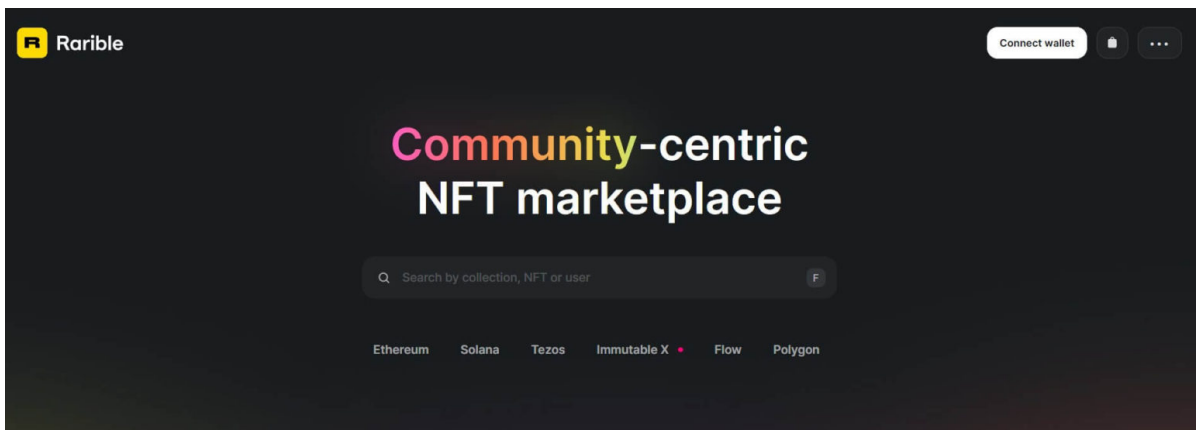


Рисунок 2.10 – головна сторінка маркетплейсу Rarible

Однією з найпростіших платформ для розуміння новими користувачами є ринок Rarible NFT. Щоб створити обліковий запис і почати купувати та продавати NFT, потрібно лише кілька хвилин. Або можна скористатися можливістю карбування Rarible і створити свій власний.

Кожен може створювати NFT на ринку Rarible. Вам не потрібні вже існуючі ілюстрації, щоб створювати NFT, і вам не потрібно вносити купу криптовалюти наперед. Це одна з найкращих платформ NFT для початківців.

Підключивши гаманець до Rarible, можна почати створювати, купувати та продавати NFT досхочу. Це все. З Rarible будь-хто може розробити портфоліо NFT або придбати твори мистецтва, які, на вашу думку, можуть бути цінними в майбутньому. Однак Rarible в основному зосереджується на мистецтві, тому, якщо ви хочете інші типи NFT, вам доведеться використовувати іншу платформу.

2.3.3. Nifty Gateway

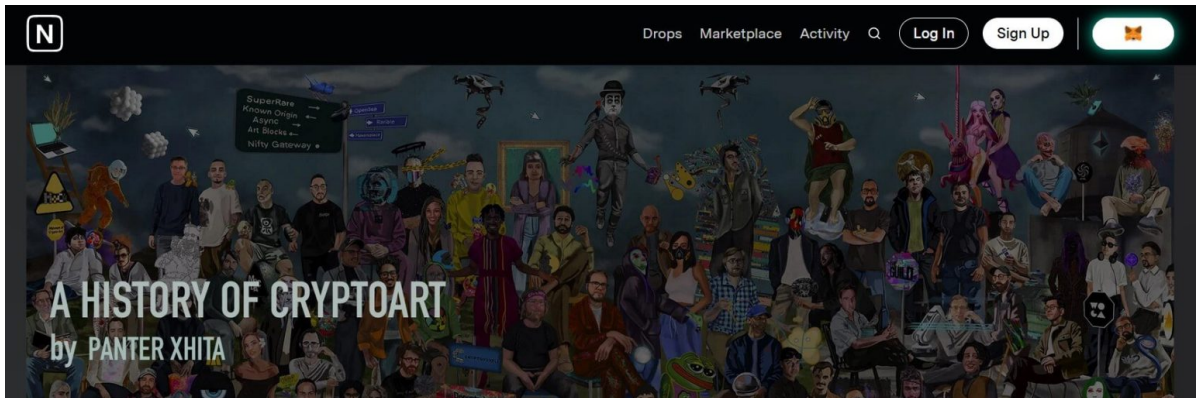


Рисунок 2.11 – головна сторінка маркетплейсу Nifty Gateway

Nifty Gateway входить до цього списку найкращих торгових майданчиків NFT, оскільки він керував створенням і запуском кількох найпопулярніших цифрових художників у просторі NFT. Якщо ви чули про Grimes і Beeple, то, мабуть, знаєте, що вони добре відомі тим, що почали працювати з Nifty Gateway.

Ця платформа є ринком кураторів мистецтва, який підтримує Gemini, що робить його ще одним сайтом NFT, пов'язаним із криптобіржею. NFT на ринку називаються Nifties і знаходяться в блокчейні Ethereum.

Окрім спеціального торгового майданчика, Nifty Gateway також зберігає на своїй платформі будь-які покупки користувачів NFT. Це означає, що вам не потрібно зберігати свої NFT у гаманці – Gemini зберігає предмет для вас. Звичайно, не кожен колекціонер хоче, щоб їхні NFT зберігалися на біржі, але це дозволяє легко перетворити NFT на фіатну валюту, якщо виникне потреба.

3.ПРОЕКТУВАННЯ МАРКЕТПЛЕЙСУ

3.1. Визначення вимог до архітектури

При проектуванні децентралізованої веб-платформи для розміщення, майнінгу та продажу NFT токенів на базі React, Moko та ICP, важливо визначити вимоги до платформи. Ось декілька ключових вимог:

Функціональність: Платформа повинна мати можливість створювати, продавати та купувати NFT токени. Вона також повинна мати можливість перегляду колекцій NFT та інформації про кожен NFT.

Безпека: Платформа повинна забезпечувати безпеку транзакцій та даних користувачів. Вона повинна використовувати надійні криптографічні методи для захисту від шахрайства та втрати даних.

Продуктивність: Платформа повинна бути в змозі обробляти велику кількість транзакцій та користувачів без значного впливу на продуктивність або швидкість.

Інтерфейс користувача: Платформа повинна мати інтуїтивно зрозумілий та простий у використанні інтерфейс користувача.

Сумісність: Платформа повинна бути сумісна з різними веб-браузерами та пристроями.

Масштабованість: Платформа повинна бути в змозі масштабуватися для підтримки зростання кількості користувачів та транзакцій.

Ці вимоги слугують основою для проектування та розробки платформи. Вони допомагають забезпечити, що платформа відповідає потребам користувачів та виконує свої основні функції ефективно.

3.2. Розробка архітектури платформи

Вся архітектура програми (або платформи) має два рівні, включаючи блокчейн шар і зовнішній рівень. Рівень блокчейну керує блокчейном і розумом контракти платформи, і будь-який користувач може працювати з блокчейном для запиту даних, оновлювати транзакції та публікувати смарт-контракти. Інтерфейс працюватиме як мережа додаток для користувачів. Інтерфейс можна розглядати як розподілену платформу, оскільки йому не потрібен сервер для зберігання даних, а зчитує дані безпосередньо з блокчейну.

Таким чином, порівняно з традиційною архітектурою додатків (див. рис. 13), D-App повністю децентралізовано, а інтерфейс розміщено в однорангових мережах замість централізованих сервери хостингу (див. рис. 3.1). Це гарантує, що D-App не контролюється жодною особою

і є стійким до цензури або припинення роботи.

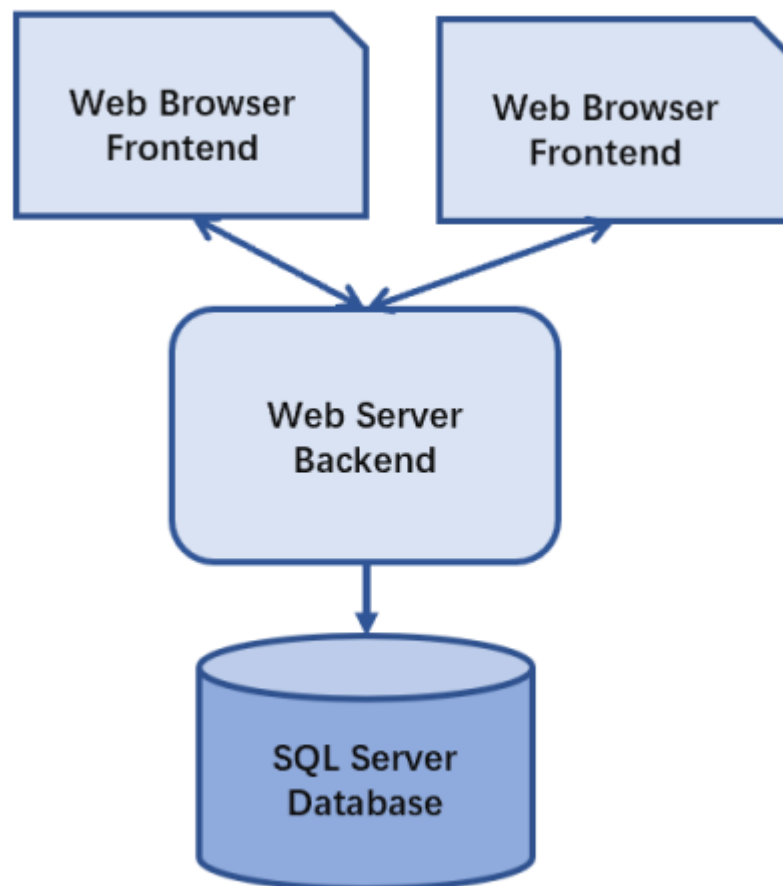


Рисунок 3.1 – традиційна архітектура програми

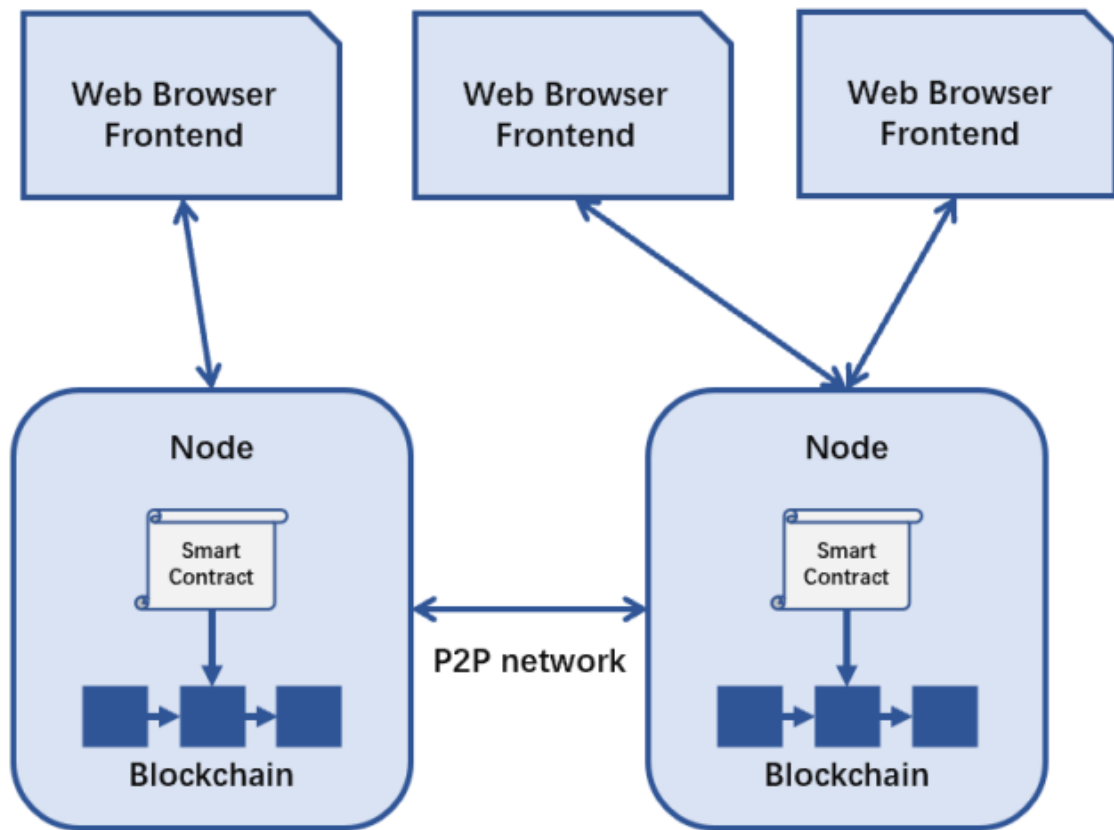
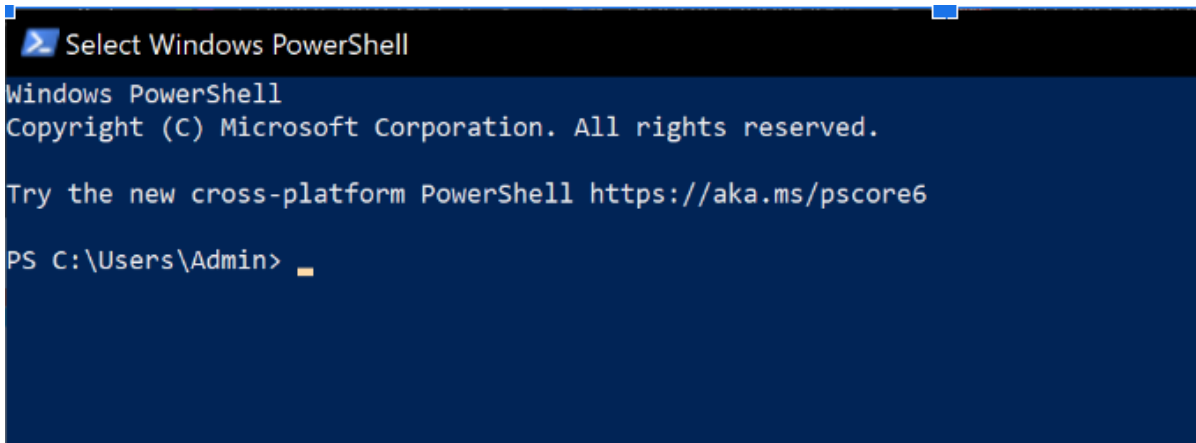


Рисунок 3.2 – архітектура розроблюваної платформи

3.3. Налаштування залежностей для розробки платформи

Для роботи з Internet Computer та Motoko в операційній системі Windows потрібно для початку встановити Windows Subsystem for Linux (WSL) – це прошарок сумісності для запуску виконуваних файлів операційної системи Windows. Щоб встановити WSL, потрібно виконати наступні кроки:

Відкрити PowerShell (див. рис. 3.3) у режимі адміністратора: клацнути правою кнопкою миші та вибрати “Запустити як адміністратор”, ввести команду `wsl --install`, а потім перезавантажити свій комп’ютер.



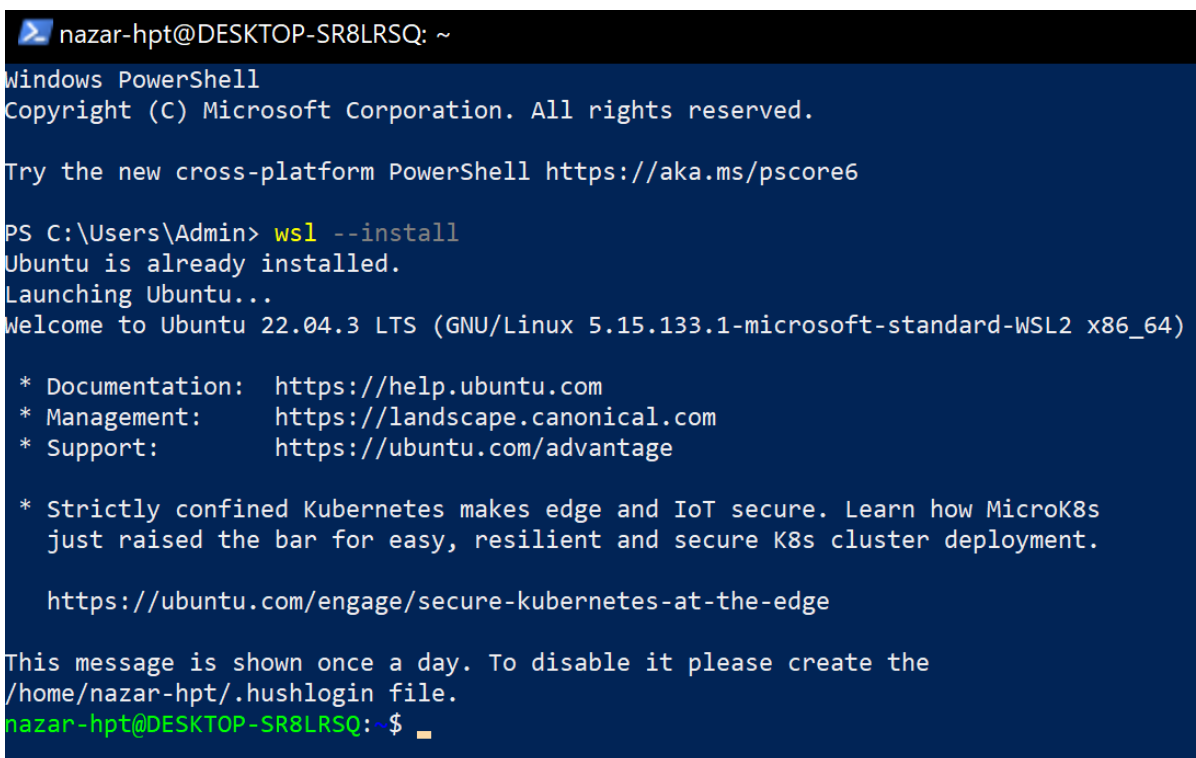
```
Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Admin>
```

Рисунок 3.3 – вікно оболонки PowerShell

Приклад виконання команди `wsl --install` зображено на рисунку 3.4, але оскільки підсистема Linux вже встановлена на даному комп'ютері тому користувачу буде виведена відповідь що система вже встановлена.



```
nazar-hpt@DESKTOP-SR8LRSQ: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Admin> wsl --install
Ubuntu is already installed.
Launching Ubuntu...
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.133.1-microsoft-standard-WSL2 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/nazar-hpt/.hushlogin file.
nazar-hpt@DESKTOP-SR8LRSQ:~$
```

Рисунок 3.4 – виконання команди `wsl --install` в оболонці PowerShell

Після перезавантаження вам буде запропоновано встановити ім'я користувача Ubuntu та пароль, і буде успішно встановлено WSL. Потрібно зберегти введену інформацію, бо вона знадобляться пізніше (див. рис. 3.5).

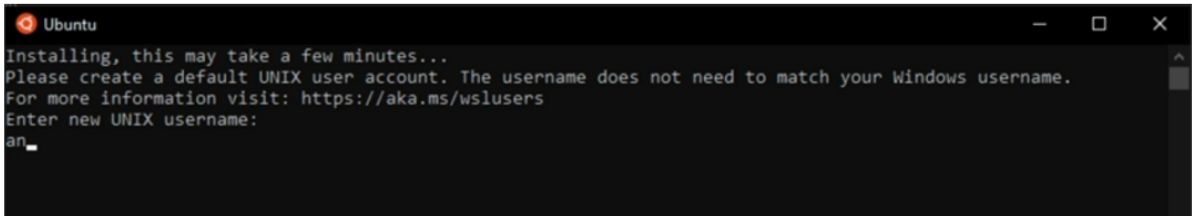


Рисунок 3.5– командний рядок Ubuntu для налаштування користувача

Щоб переконатися що все працює коректно, потрібно ввести команду яка зображена на рисунку 3.6.

```
wsl --list --verbose
```

Рисунок 3.6– команда для перевірки успішності встановлення WSL

Якщо все встановлено успішно, у командному рядку PowerShell буде виведена інформація яка зображена на рисунку 3.7.

```
PS C:\WINDOWS\system32> wsl --list --verbose
NAME      STATE      VERSION
* Ubuntu   Running    2
PS C:\WINDOWS\system32>
```

Рисунок 3.7– успішний результат перевірки встановлення WSL

Розробка платформи буде виконуватись за допомогою середовища розробки Visual Studio Code, але цього недостатньо, потрібно встановити наступні розширення:

- Motoko;

– Remote WSL.

Воно надає підтримку мови програмування Motoko, яка є високорівневою мовою смарт-контрактів для Internet Computer.

Ось деякі особливості цього розширення:

Перевірка типів: розширення забезпечує перевірку типів для розробки каністр Motoko. Форматування: розширення надає можливість форматування коду Motoko.

Фрагменти коду: розширення надає фрагменти коду для розробки каністр Motoko.

Інші функції: розширення також надає інші функції, такі як перехід до визначення, автозаповнення, перегляд всієї доступної інформації про автозаповнення та фрагменти коду.

Розширення Remote WSL для VS Code дозволяє використовувати Windows Subsystem for Linux (WSL) як основне середовище розробки прямо з VS Code. Ось декілька ключових особливостей цього розширення:

Розробка в Linux-середовищі: можна розробляти в Linux-середовищі, використовувати Linux-специфічні інструменти та утиліти, а також запускати та налагоджувати свої Linux-орієнтовані додатки, все це зручно з Windows.

Виконання команд та розширень безпосередньо в WSL: Розширення виконує команди та інші розширення безпосередньо в WSL, тому можна редагувати файли, розташовані в WSL або в монтированій файловій системі Windows (наприклад, /mnt/c), не турбуючись про проблеми з шляхами, сумісністю бінарних файлів або інші виклики між операційними системами (див. рис. 3.8).

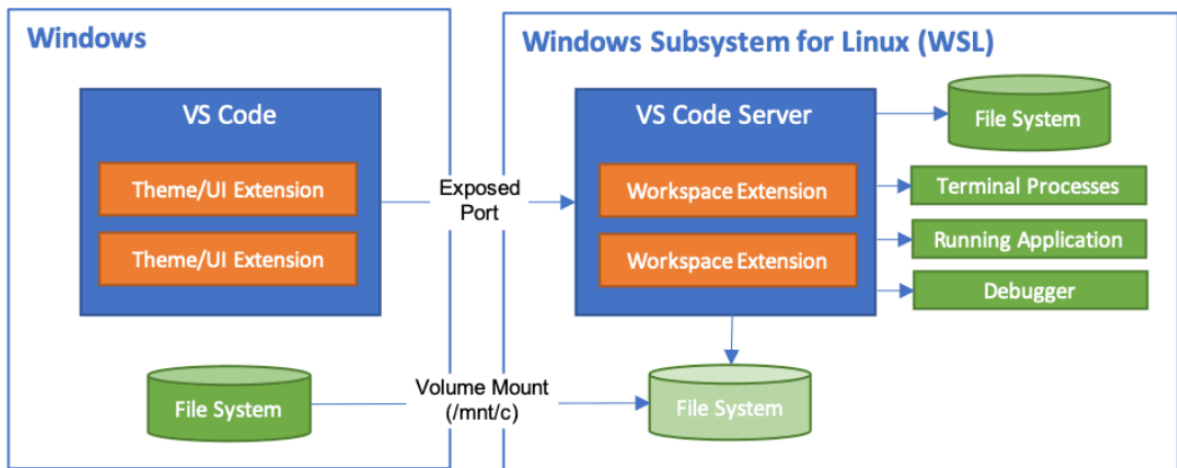


Рисунок 3.8 – архітектура взаємодії розширення WSL з встановленою оболонкою у системі

Локальний досвід розробки: Це дозволяє VS Code надавати досвід розробки на рівні локального робочого місця - включаючи повну IntelliSense (завершення), навігацію коду та налагодження - незалежно від того, де розміщений ваш код.

Також для роботи потрібно встановити в підсистему Linux платформу Node.js. Це можна зробити за допомогою менеджера пакетів Homebrew

Homebrew - це безкоштовна та відкрита система керування пакетами програмного забезпечення, яка спрощує встановлення програмного забезпечення на операційній системі Apple, macOS, а також на Linux. Homebrew дозволяє встановлювати, оновлювати або видаляти програмне забезпечення, виконуючи команди в терміналі. Можна використовувати Homebrew для легкого встановлення останніх версій різних інструментів розробника для macOS, таких як PostgreSQL, Python, PHP, Nginx тощо.

Щоб встановити Homebrew, потрібно виконати команду яка зображена на рисунку 3.9.

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/in
stall.sh)"
```

Рисунок 3.9 – команда встановлення Homebrew

Встановлення пакетів: Після встановлення Homebrew, можна встановлювати пакети за допомогою команди `brew install <package-name>`. Для встановлення Node.js потрібно ввести наступну команду: `brew install node`.

Основне без чого неможливо створювати проект на Internet Computer - це команда DFX.

DFX, відомий як DFINITY command-line execution environment, є основним інструментом для створення, розгортання та керування dapps, які ви розробляєте для Internet Computer. Команду `dfx` можна використовувати з різними прапорцями та підкомандами, щоб виконувати різні типи операцій.

Ось декілька ключових особливостей DFX:

Створення dapps: DFX дозволяє вам створювати dapps для Internet Computer. Розгортання dapps: За допомогою DFX можна розгортати свої dapps на Internet Computer. Керування dapps: DFX також дозволяє керувати своїми dapps після їх розгортання. DFX є важливим інструментом для розробників, які хочуть створювати та розгортати dapps на Internet Computer.

Для встановлення DFX потрібно відкрити командний рядок Ubuntu, та вставити код який зображений на рисунку 3.10.

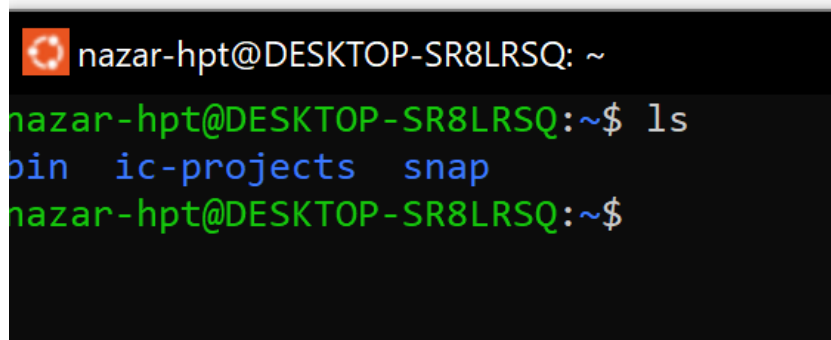
```
DFX_VERSION=0.9.3 sh -ci "$(curl -fsSL https://sdk.dfinity.org/install.sh)"
```

Рисунок 3.10– команда для встановлення DFX

Після виконання попередньої команди користувачу буде виведено інформацію про успішне встановлення та шлях з встановленим DFX.

3.4. Створення та запуск тестового проекту

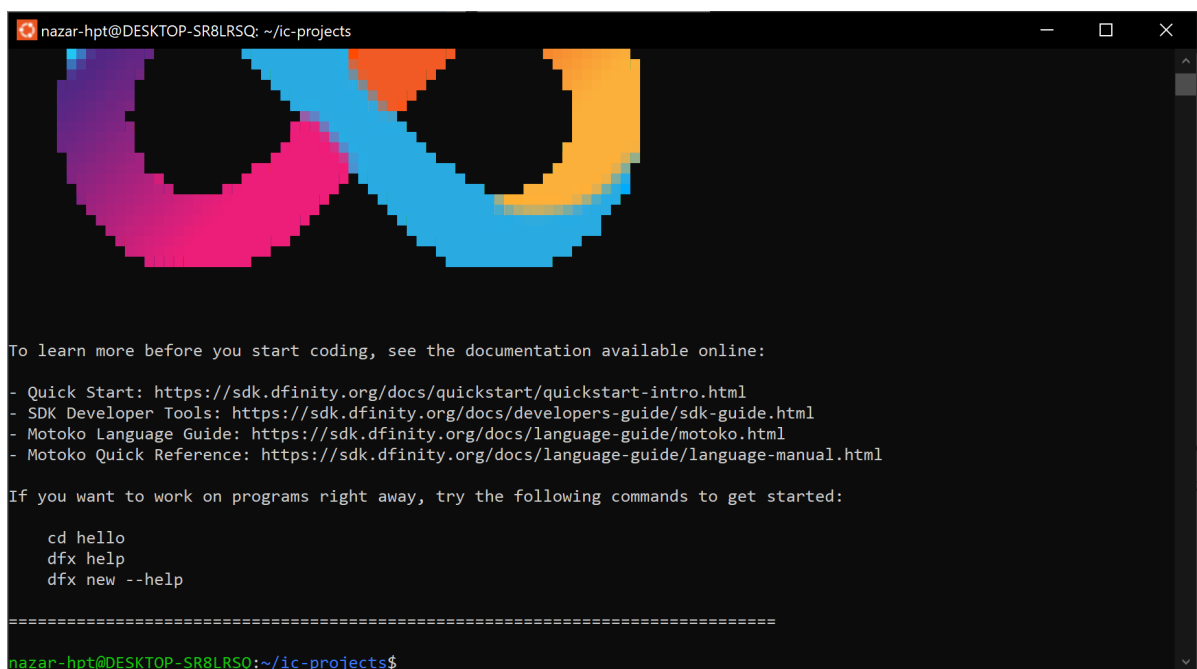
Для початку створимо тестовий DApp з назвою Hello. Потрібно запустити командний рядок Ubuntu та створити директорій де будуть розміщуватися всі проекти, це можна зробити за допомогою команди: `mkdir ic-projects`. Після виконання попередньої команди можна перевірити всі директорії та файли за допомогою команди `ls` (див. рис. 3.11).



```
nazar-hpt@DESKTOP-SR8LRSQ: ~  
nazar-hpt@DESKTOP-SR8LRSQ:~$ ls  
bin ic-projects snap  
nazar-hpt@DESKTOP-SR8LRSQ:~$
```

Рисунок 3.11 - використання команди `ls` для перевірки успішності створення директорії `ic-computer`

Після переміщення нашого командного рядка в директорію `ic-computer` створюємо наш перший Internet Computer DApp за допомогою команди: `dfx new hello` (див. рис. 3.12).



```
nazar-hpt@DESKTOP-SR8LRSQ: ~/ic-projects  
To learn more before you start coding, see the documentation available online:  
- Quick Start: https://sdk.dfinity.org/docs/quickstart/quickstart-intro.html  
- SDK Developer Tools: https://sdk.dfinity.org/docs/developers-guide/sdk-guide.html  
- Motoko Language Guide: https://sdk.dfinity.org/docs/language-guide/motoko.html  
- Motoko Quick Reference: https://sdk.dfinity.org/docs/language-guide/language-manual.html  
If you want to work on programs right away, try the following commands to get started:  
  cd hello  
  dfx help  
  dfx new --help  
=====
```

Рисунок 3.12 - успішне встановлення децентралізованого додатку

Для відкриття тестового проекту потрібно запустити VS Code та підключитись до WSL (див. рис. 3.13).

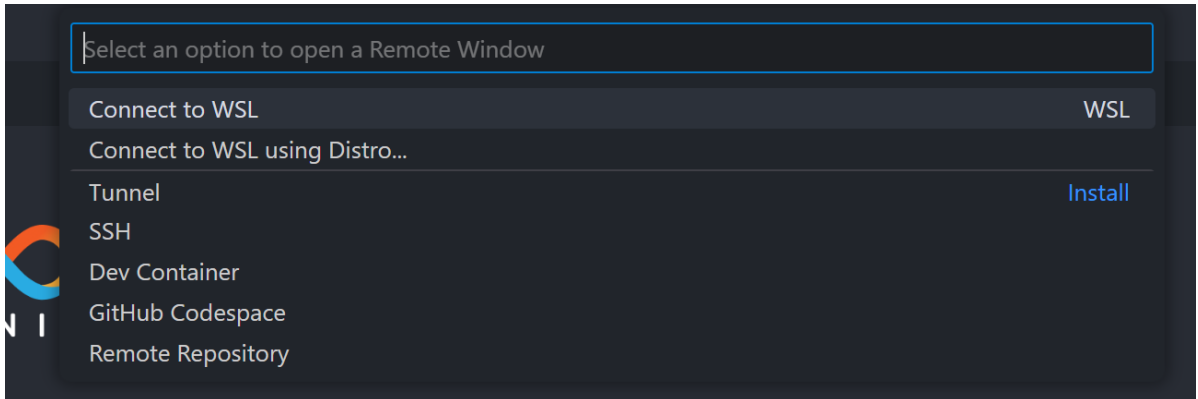


Рисунок 3.13 - підключення до WSL

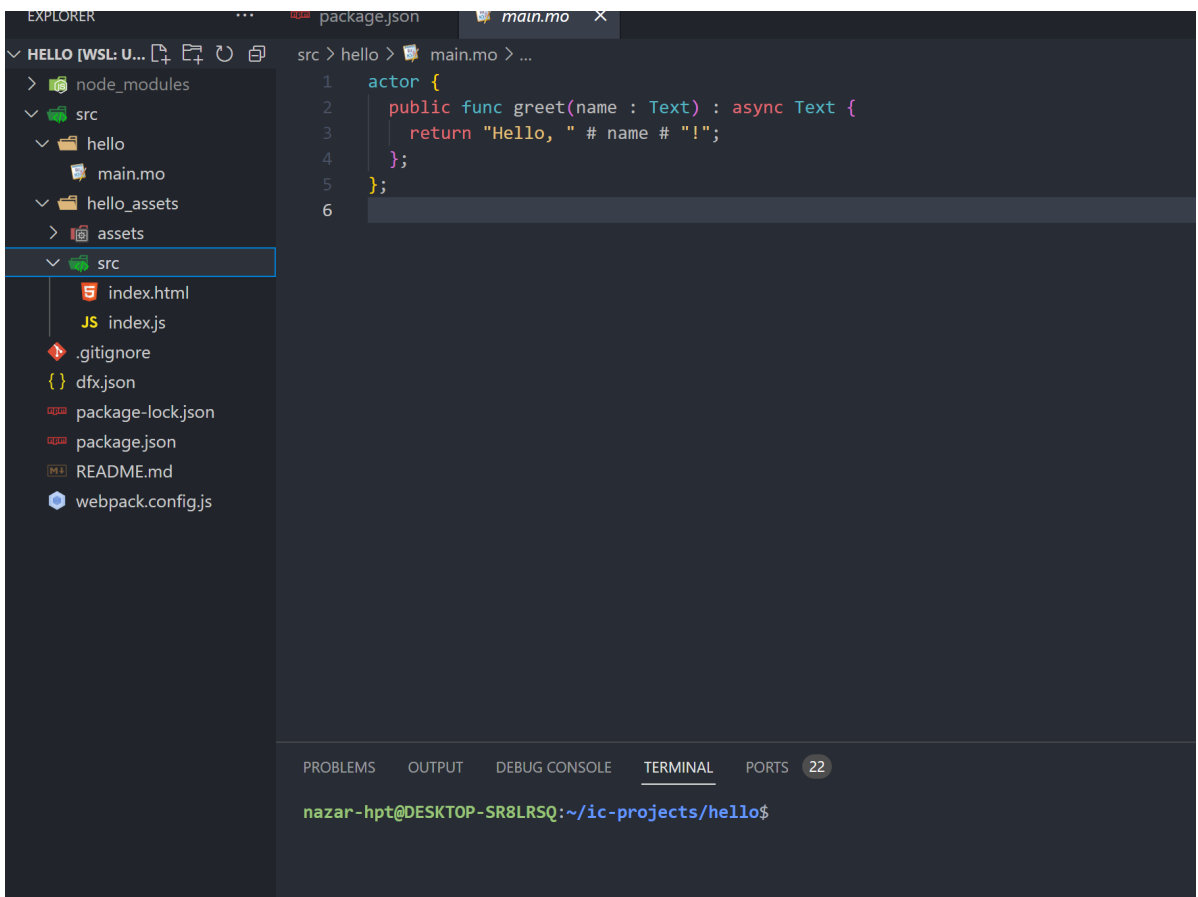


Рисунок 3.14 - структура проекту

Розгортання проекту. Потрібно натиснути на панелі інструментів Terminal (див. рис. 3.15).

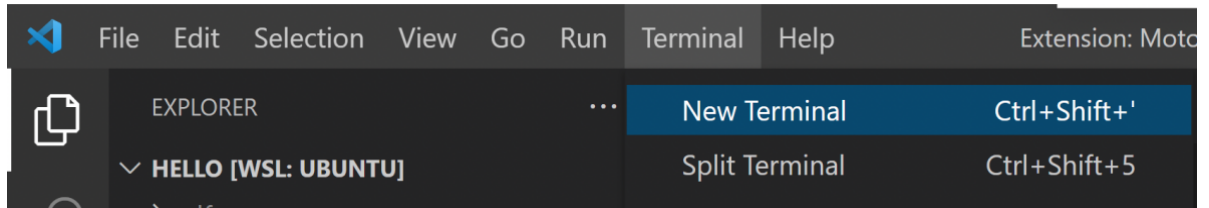
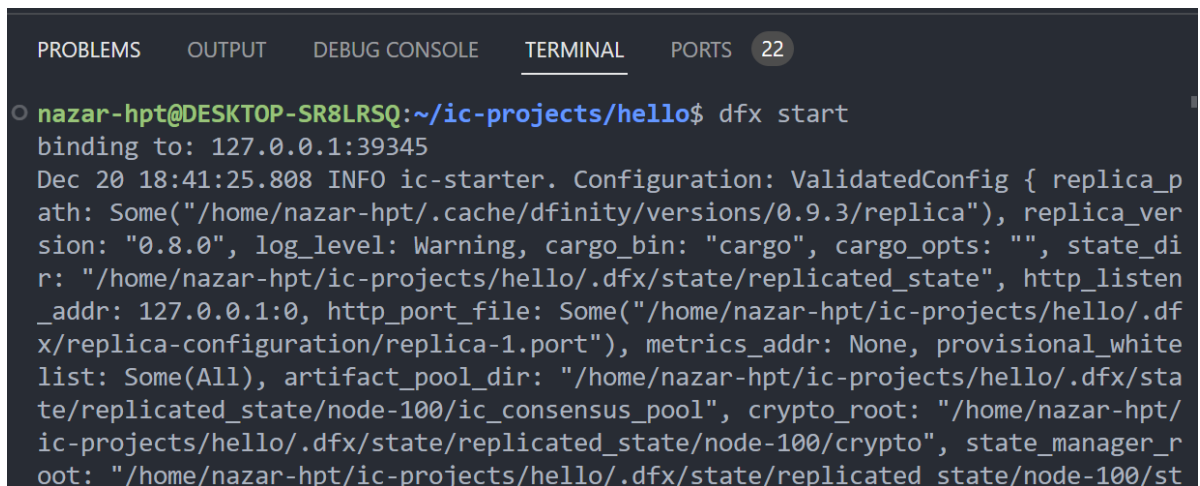


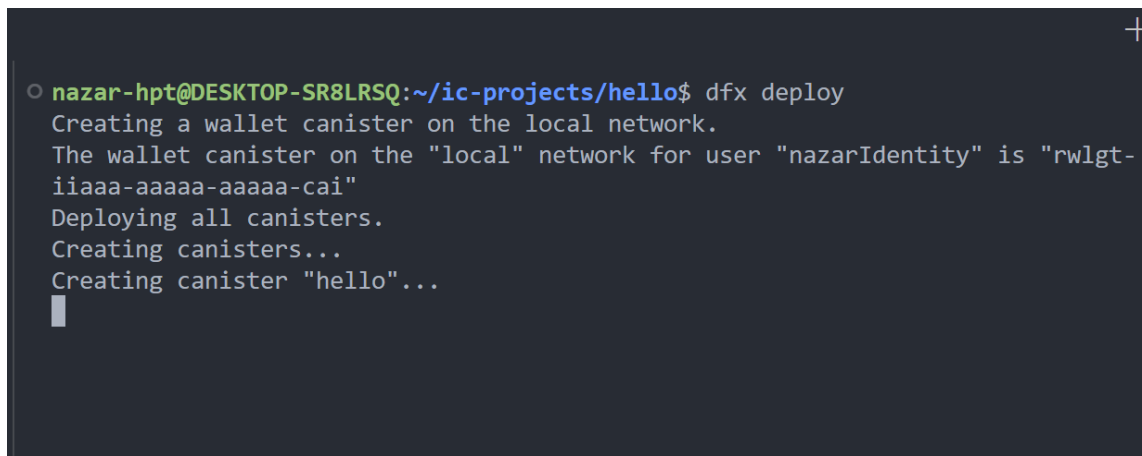
Рисунок 3.15 - створення нового терміналу

Запуск локального середовища: для його запуску це можна зробити за допомогою команди `dfx start --background` (див. рис. 3.16). Розгортання додатку: розгорнути свій додаток можна за допомогою команди `dfx deploy` (див. рис. 3.17).

A screenshot of a terminal window in Visual Studio Code. The terminal shows the command `dfx start` being executed. The output includes the binding address `127.0.0.1:39345` and a detailed log message from `ic-starter` with configuration details for a replicated state.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 22
nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/hello$ dfx start
binding to: 127.0.0.1:39345
Dec 20 18:41:25.808 INFO ic-starter. Configuration: ValidatedConfig { replica_path: Some("/home/nazar-hpt/.cache/dfinity/versions/0.9.3/replica"), replica_version: "0.8.0", log_level: Warning, cargo_bin: "cargo", cargo_opts: "", state_dir: "/home/nazar-hpt/ic-projects/hello/.dfx/state/replicated_state", http_listen_addr: 127.0.0.1:0, http_port_file: Some("/home/nazar-hpt/ic-projects/hello/.dfx/replica-configuration/replica-1.port"), metrics_addr: None, provisional_whitelist: Some(All), artifact_pool_dir: "/home/nazar-hpt/ic-projects/hello/.dfx/state/replicated_state/node-100/ic_consensus_pool", crypto_root: "/home/nazar-hpt/ic-projects/hello/.dfx/state/replicated_state/node-100/crypto", state_manager_root: "/home/nazar-hpt/ic-projects/hello/.dfx/state/replicated_state/node-100/st
```

Рисунок 3.16 - запуск локального серверу dfx

A screenshot of a terminal window in Visual Studio Code. The terminal shows the command `dfx deploy` being executed. The output indicates the creation of a wallet canister on the local network and the deployment of all canisters, including the 'hello' canister.

```
nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/hello$ dfx deploy
Creating a wallet canister on the local network.
The wallet canister on the "local" network for user "nazarIdentity" is "rwlgt-iaaaa-aaaaa-aaaaa-cai"
Deploying all canisters.
Creating canisters...
Creating canister "hello"...
```

Рисунок 3.17 - компіляція каністр та розміщення в локальному Internet Computer

Отже після усіх попередніх операцій, децентралізований веб додаток успішно розгорнутий на локальному сервері, щоб його побачити, потрібно у веб-браузері в рядку пошуку ввести: `http://localhost:8080/` (див. рис. 3.18).

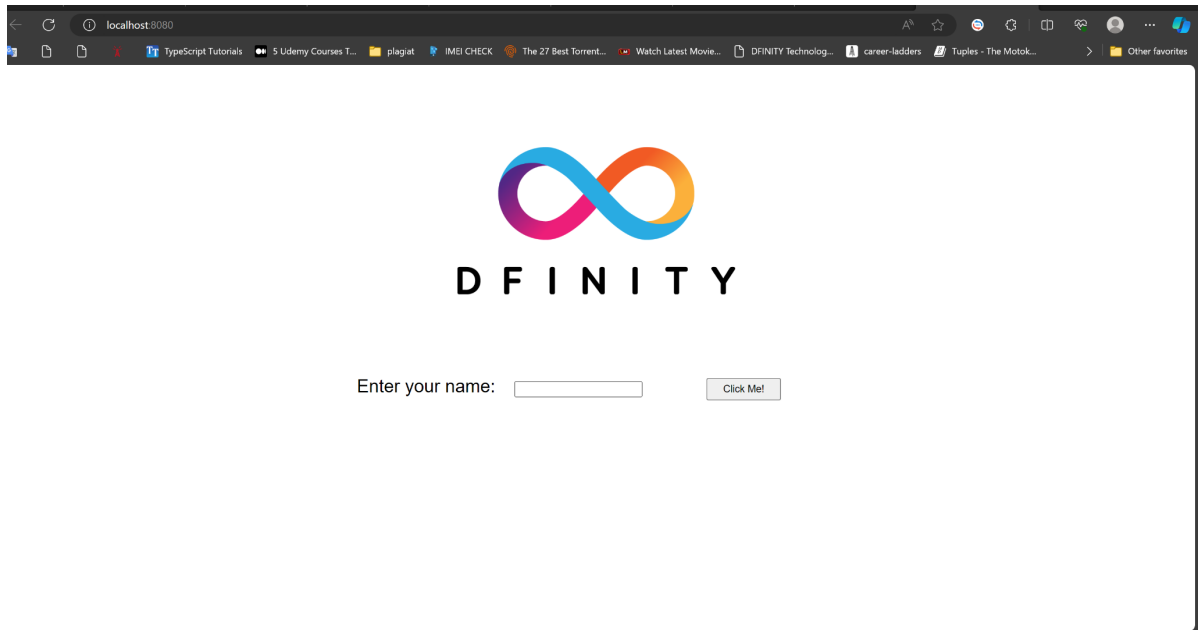


Рисунок 3.18 - тестовий шаблонний децентралізований додаток розгорнутий на локальному сервері

4. ВПРОВАДЖЕННЯ ТА ТЕСТУВАННЯ ВЕБ-ПЛАТФОРМИ

4.1. Проведення тестування веб-платформи

Тестування - це процес перевірки програмного продукту з метою виявлення помилок, недоліків та відхилень від вимог. Основна мета тестування полягає в тому, щоб переконатися, що програма працює коректно, відповідає вимогам та задачам, які вона повинна виконувати.

Дана платформа матиме обидва додатки які взаємодіятимуть між собою, перший - додаток для роботи з токени для купівлі NFT tokenів, де користувач матиме можливість перевірити свій баланс, отримати безкоштовні токени для купівлі NFT tokenів, та можливість переводу tokenів іншому користувачеві, другий - це маркетплейс NFT tokenів, де користувач матиме можливість, переглянути свої токени, розміщувати(майнити) NFT на маркетплейсі, виставляти їх на продаж та купити у інших користувачів. Спочатку розглянемо перший додаток. Архітектура веб-додатку зображена на рисунку 4.1.

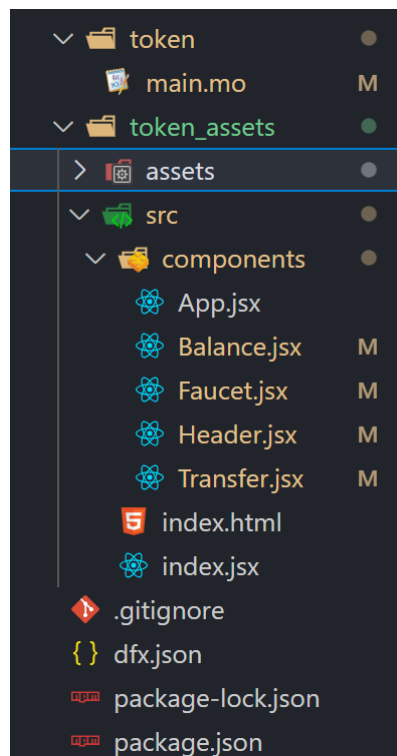


Рисунок 4.1 - файлова архітектура веб-додатку для менеджменту tokenів

Також у даному додатку присутній унікальний спосіб авторизації який було створено компанією Dfinity. Internet Identity — це система аутентифікації на основі пасків (passkey), яка дозволяє вам створювати та керувати кількома ключами для різних децентралізованих додатків на Internet Computer. Internet Computer — це децентралізована платформа для Web3-додатків, яка пропонує новий підхід до розробки та використання програмного забезпечення.

Основні особливості Internet Identity - це безпека та зручність, оскільки замість використання логінів та паролів потрібен тільки унікальний ключ. Відсутність відстеження:

При створенні облікового запису з даппом за допомогою Internet Identity створюється новий набір аутентифікаційних ключів.

Ваша ідентичність не може бути відстежена між різними даппами або веб-сайтами. Приклад коду для авторизації зображено на рисунку 4.2.

```
index.jsx ×
src > token_assets > src > index.jsx > ...
 1  import ReactDOM from "react-dom";
 2  import React from "react";
 3  import App from "../components/App";
 4  import { AuthClient } from "@dfinity/auth-client";
 5
 6  const init = async () => {
 7    const authClient = await AuthClient.create();
 8    if (await authClient.isAuthenticated()) {
 9      ReactDOM.render(<App />, document.getElementById("root"));
10    } else {
11      await authClient.login({
12        identityProvider: "https://identity.ic0.app/#authorize",
13        onSuccess: () => {
14          ReactDOM.render(<App />, document.getElementById("root"));
15        },
16      });
17    }
18  };
19
20  init();
21  |
```

Рисунок 4.2 - код для авторизації користувача

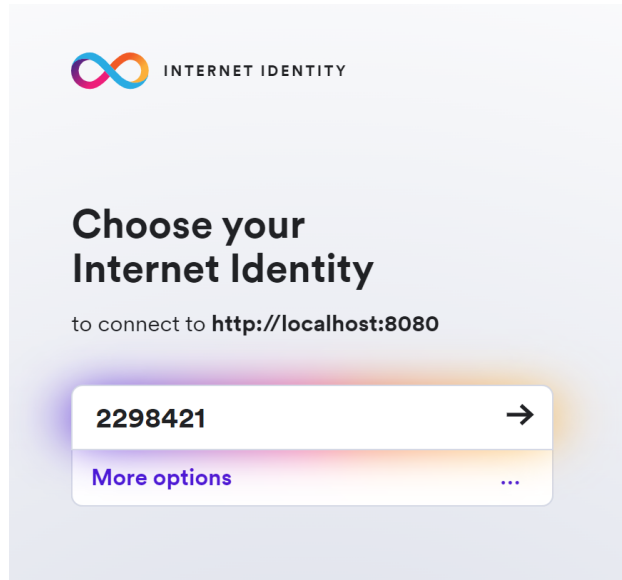


Рисунок 4.3 - вікно авторизації користувача

Після успішної авторизації, користувача буде перенесено на вікно додатку (див. рис. 4.4).

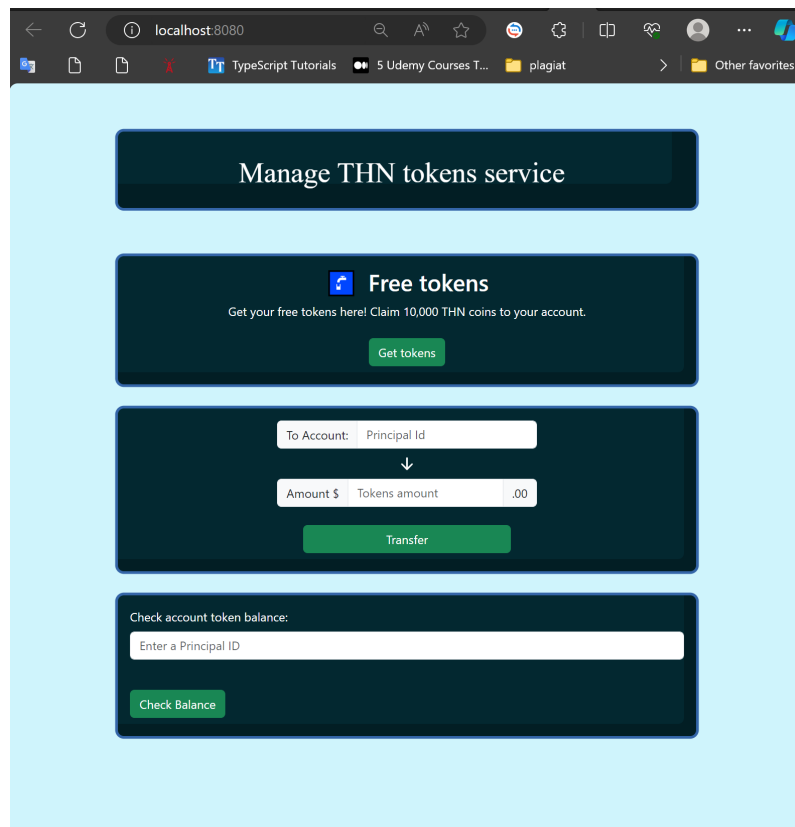


Рисунок 4.4 - основне вікно веб-додатку

В цьому веб-додатку користувач матиме можливість перевірити свій баланс, отримати безкоштовні токени для купівлі NFT токенів, та можливість переводу токенів іншому користувачеві.

Наступним веб-додатком до розгляду буде вже сама основна платформа для NFT токенів. Файлову архітектуру можна побачити на рисунку 4.5.

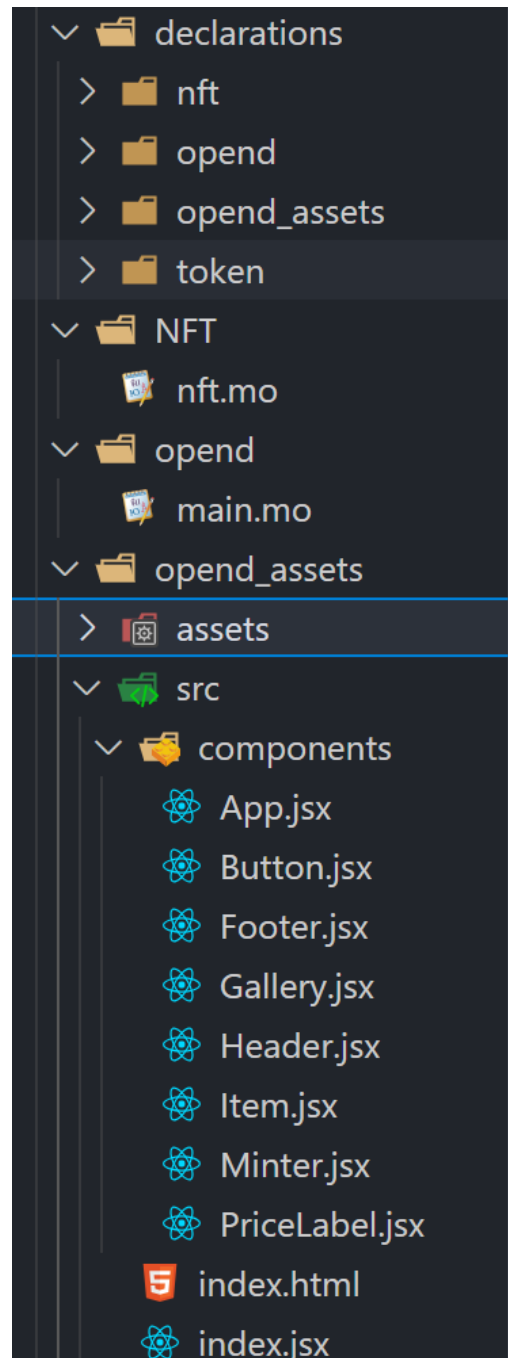


Рисунок 4.5 - файлова архітектура платформи

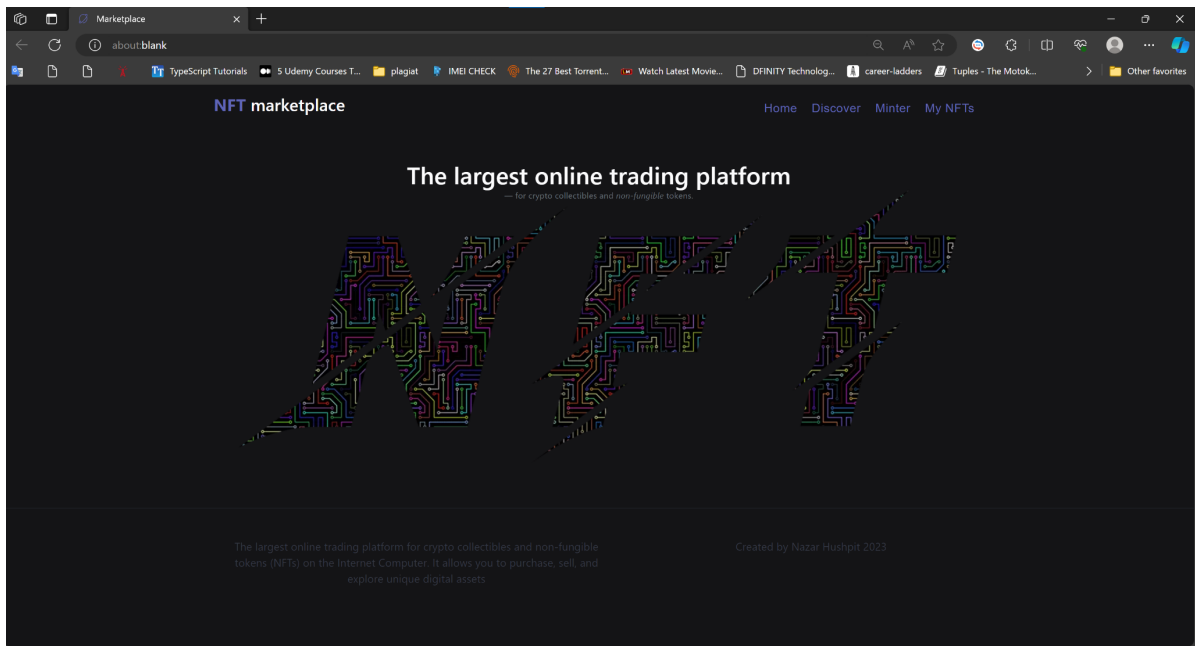


Рисунок 4.6 - головне вікно платформи для розміщення NFT токенів

4.2. Робота користувача з платформою

Для початку розглянемо додаток для роботи з токени для купівлі NFT токенів. Щоб зрозуміти як усе потрібно зрозуміти яким чином представлені користувачі в системі. Кожен користувач, NFT токен та кожна сутність в системі має свій власний ідентифіка - це так званий Principal ID.

Principal ID — це основний унікальний ідентифікатор на Internet Computer. Використовуючи Principal ID, можна створювати каністри (canisters) та автентифікуватися на додатках та сервісах, які працюють на Internet Computer. Principal ID є єдиним ідентифікатором, який додатки визнають для користувачів, і єдиним ID, який може володіти каністрами. Щоб дізнатись свій ідентифікатор потрібно ввести команду яка зображена на рисунку 4.7.

```

● nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/OPEND$ dfx identity get-principal
gdv4j-h4ggj-7dvce-pbbsi-unztr-z5fdf-4cqn-ks6jd-35epy-vupse-6ae
○ nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/OPEND$ █

```

Рисунок 4.7 - отримання Principal ID

Principal ID виступає як основний гаманець/ідентифікатор на Internet Computer. Всі інші під-ідентифікатори обробляються автоматично, створюючи безшовний користувацький досвід. Наприклад, для ICP це є Account ID. Це створює складний досвід для користувачів на Internet Computer. На відміну від Ethereum, де є одна адреса (ETH Address), яка є ідентифікатором та гаманцем для всіх ваших токенів в мережі, на Internet Computer вам потрібно керувати окремими ID/гаманцями для кожного токена. Це означає, що якщо потрібно відправити ICP та цикли комусь, вам потрібні різні адреси для кожного, які також відрізняються від основного ID цієї особи — Principal ID.

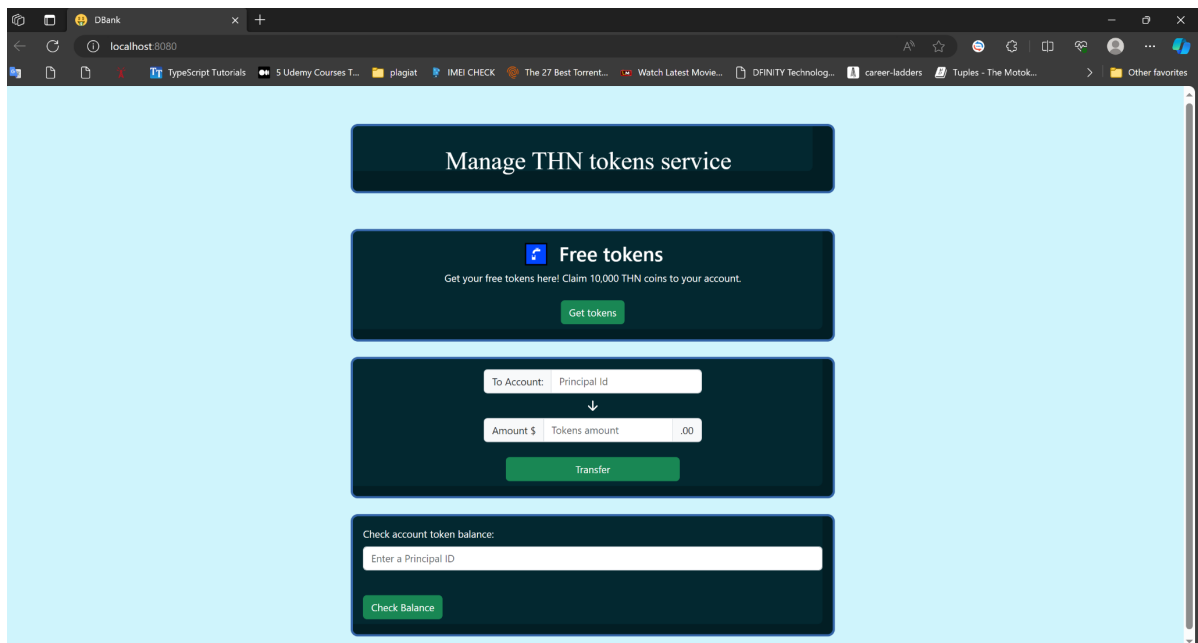
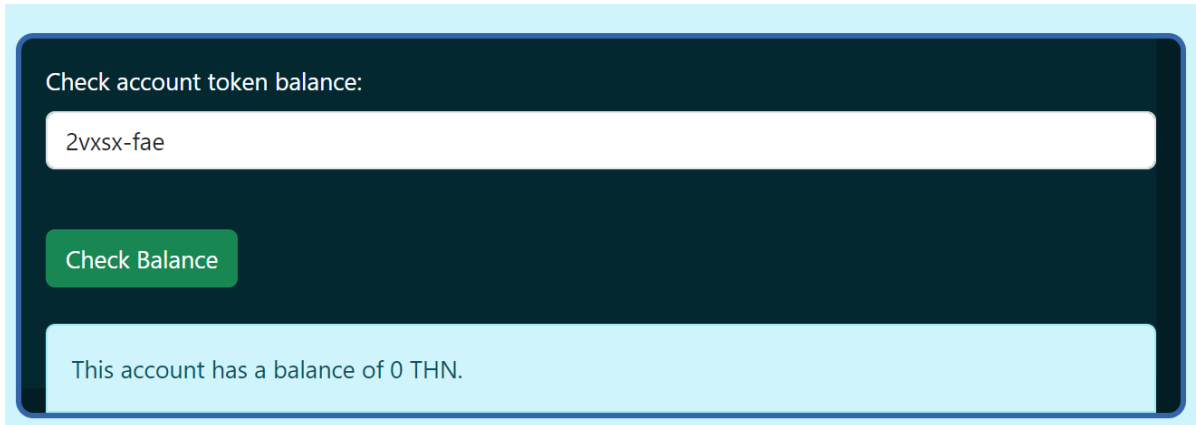


Рисунок 4.8 - головна сторінка додатку

Для перевірки балансу потрібно ввести Principal ID користувача в поле input після чого користувачеві буде виведено його баланс (див. рис. 4.9).



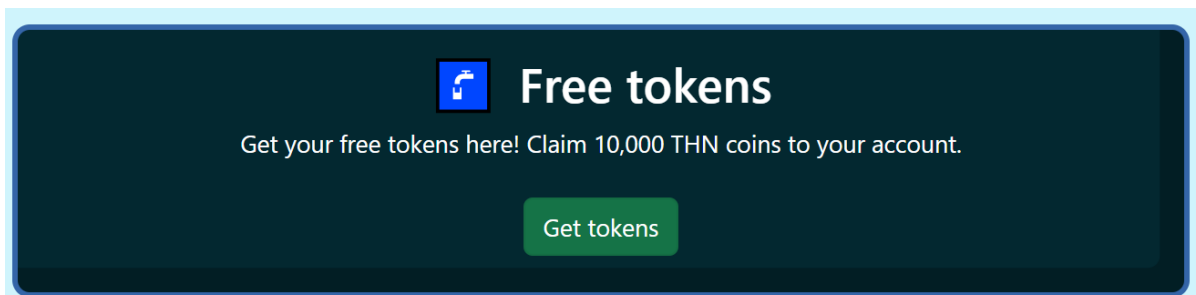
Check account token balance:


Check Balance

This account has a balance of 0 THN.

Рисунок 4.9 - перевірка балансу користувача

Як бачимо на рахунку користувача 0 токенів, проте у користувача є можливість отримати 10000 токенів безкоштовно за умови якщо він ще їх не отримував (див. рис. 4.10).



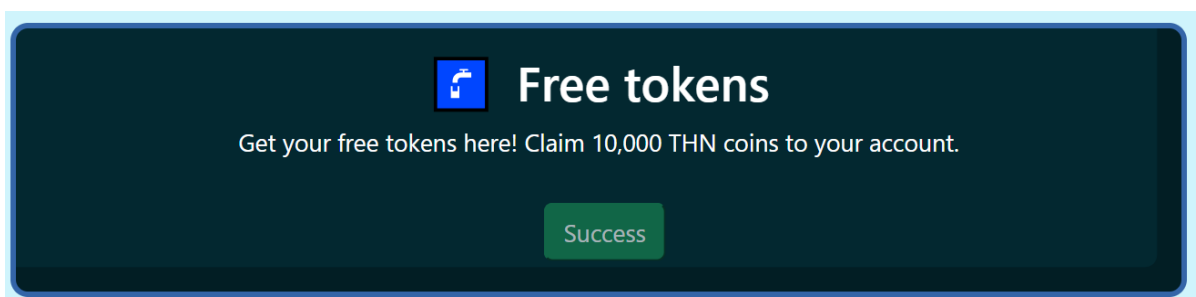
 **Free tokens**


Get your free tokens here! Claim 10,000 THN coins to your account.

Get tokens

Рисунок 4.10 - поле для отримання безкоштовних токенів

Після натиснення кнопки Get tokens на рахунок користувача буде нарахована відповідна сума, а назва кнопки зміниться на Success (див. Рис. 4.11).



 **Free tokens**

Get your free tokens here! Claim 10,000 THN coins to your account.

Success

Рисунок 4.11 - успішне виконання операції

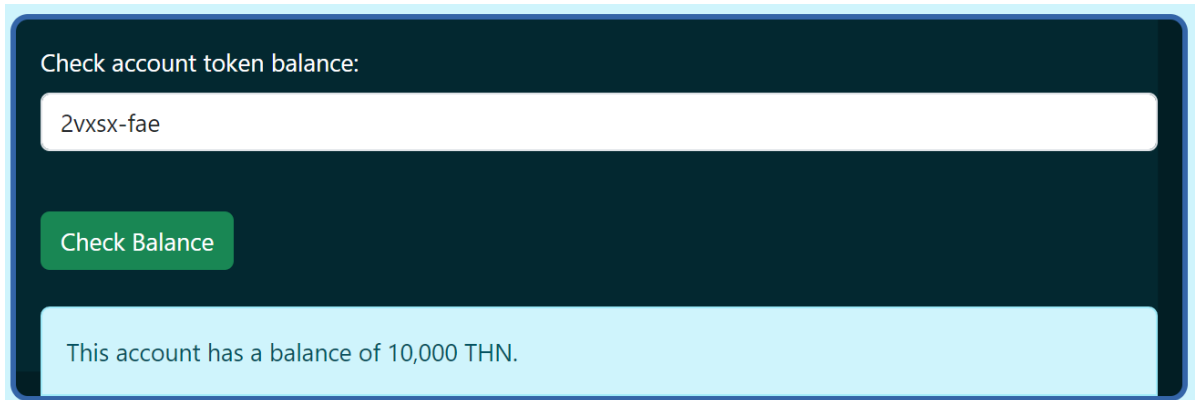


Рисунок 4.12 - баланс користувача після отримання безкоштовних токенів

Тепер перейдемо до наступного додатку та спробуємо завантажити наш перший NFT токен (див. рис. 4.13).

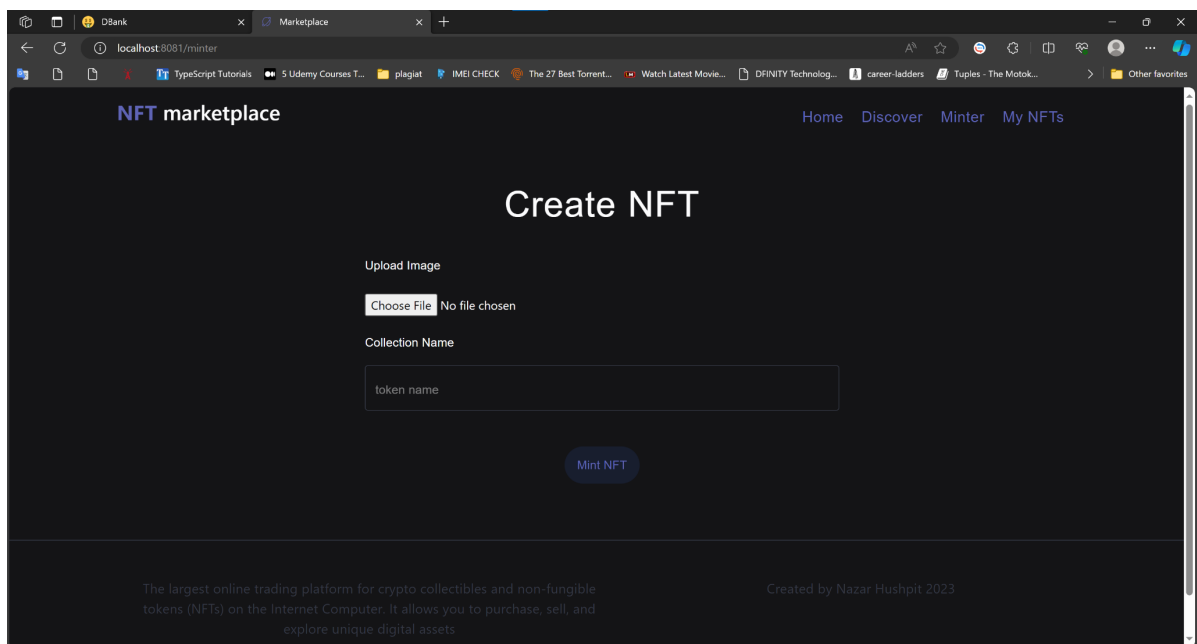


Рисунок 4.13 - вікно для завантаження NFT токена в маркетплейс

Після заповнення необхідних полів та натискання кнопки Mint NFT, з'явиться Loader повідомляючий користувача про виконання розрахунків для майнінгу NFT токена на маркетплейс (див. рис. 4.14).

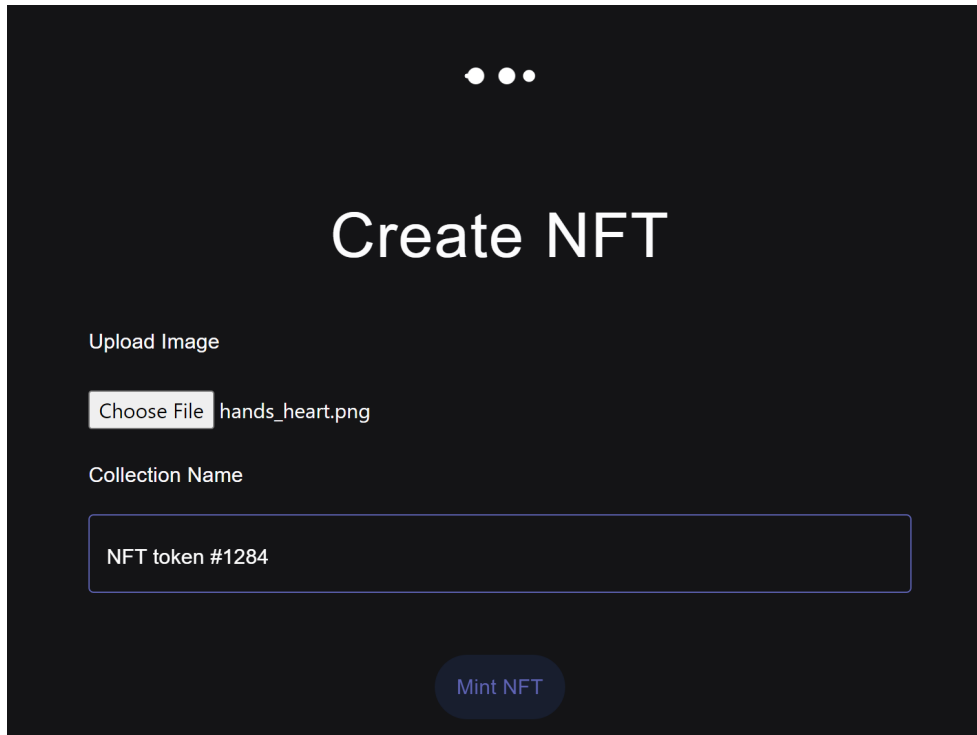


Рисунок 4.14 - процес завантаження токена на маркетплейс

Після успішного завантаження токена користувачу буде виведений токен, його назва та Principal ID власника токена (див. рис. 4.15).

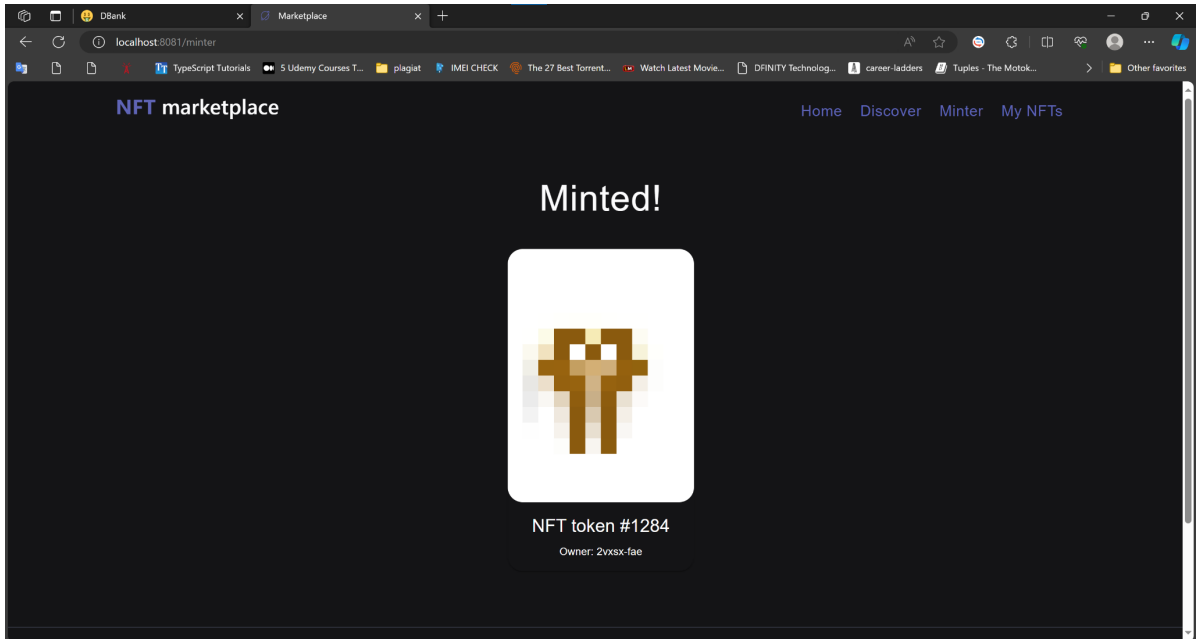


Рисунок 4.15 - успішне завантаження токена на платформу

У користувача є можливість переглянути власні токени, для цього потрібно в навігаційній панелі натиснути на My NFTs (див. рис. 4.16)

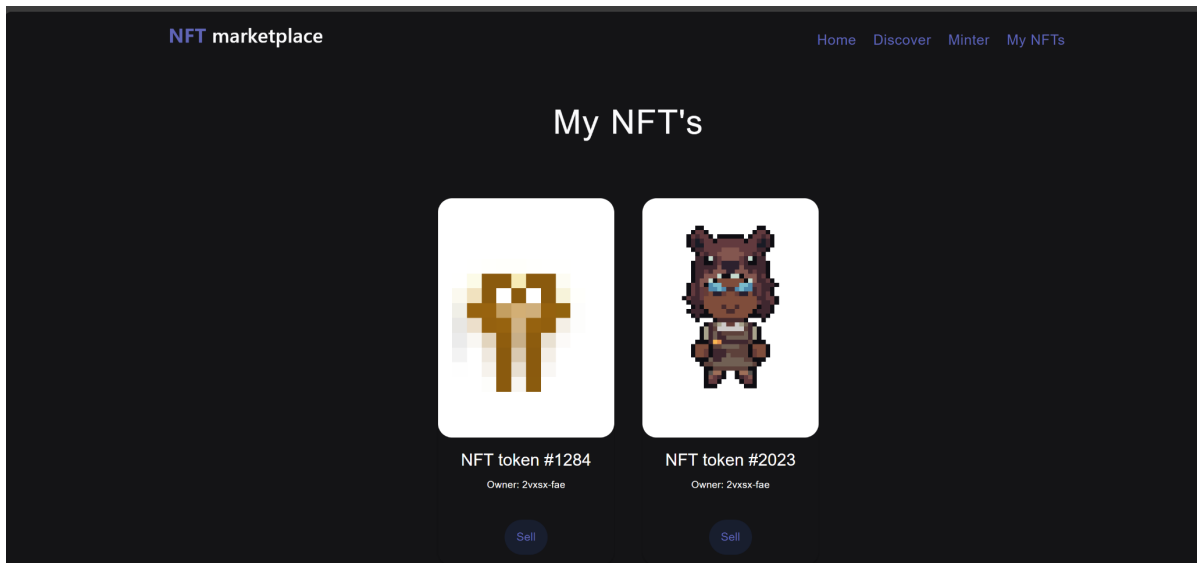


Рисунок 4.16 - список NFT токенів користувача

Маршрутизація маркетплейсом відбувається за допомогою бібліотеки react-router-dom, приклад коду для маршрутизації по додатку (див. рис. 4.17).

```

59     <Switch>
60       <Route exact path="/">
61         <main className="px-3">
62           <figure className="text-center mt-5">
63             <blockquote className="blockquote">
64               <h1> The largest online trading platform</h1>
65             </blockquote>
66             <figcaption className="blockquote-footer">
67               for crypto collectibles and{" "}
68               <cite title="Source Title">non-fungible</cite> tokens.
69             </figcaption>
70           </figure>
71           <h1></h1>
72           <p className="lead">
73             <img className="bottom-space" src={homeImage} />
74           </p>
75         </main>
76       </Route>
77       <Route path="/discover">{listingGallery}</Route>
78       <Route path="/minter">
79         <Minter />
80       </Route>
81       <Route path="/collection">{userOwnedGallery}</Route>
82     </Switch>
83   </BrowserRouter>
84   );
85 }
  
```

Рисунок 4.17 - код для маршрутизації маркетплейсом

Користувач має можливість продати власний токен, для цього потрібно перейти до списку своїх токенів, натиснути кнопку Sell та вказати вартість свого токена (див. рис. 4.18).

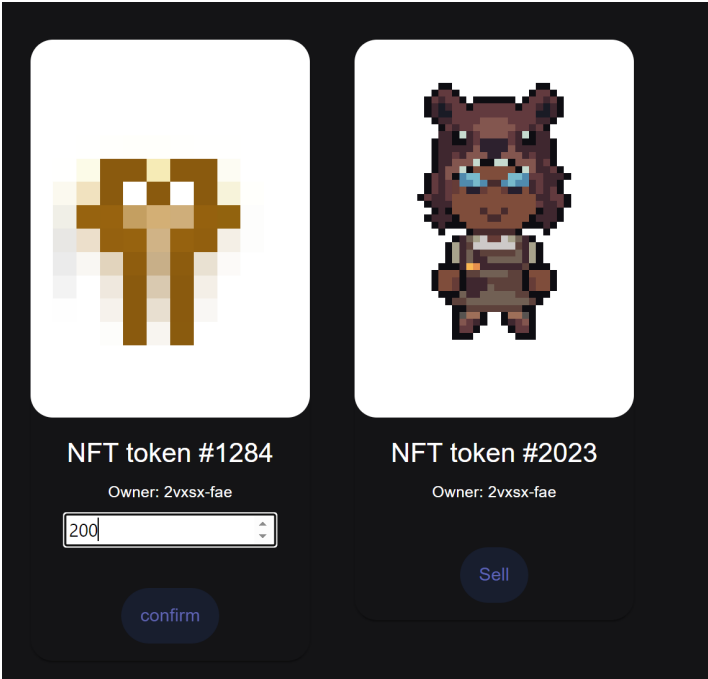


Рисунок 4.18- продаж та подальше розміщення токена на маркетплейсі

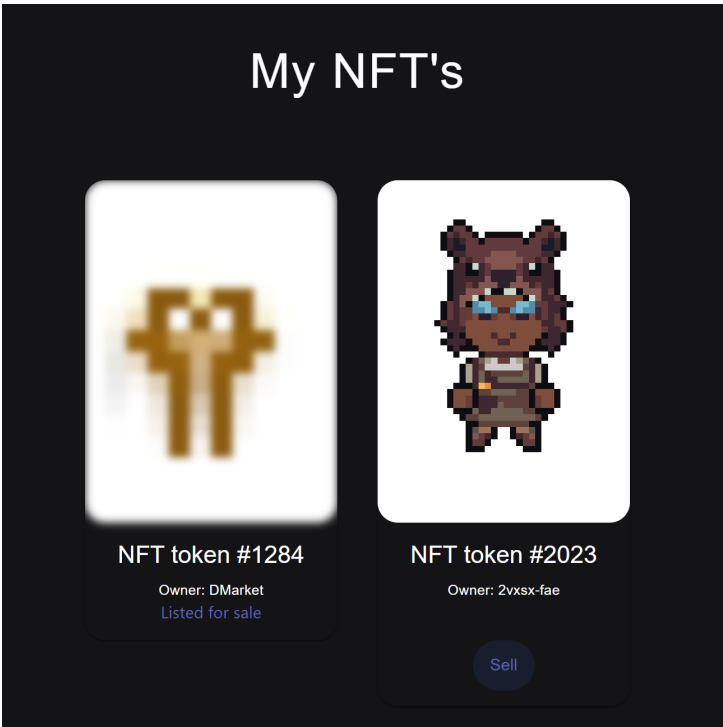


Рисунок 4.19 - вигляд токена після підтвердження продажу

Після розміщення токена для продажу інші користувачі матимуть можливість бачити NFT токен та його вартість. Також передбачена можливість купівлі токена користувачем який його збирається продати (див. рис. 4.20).

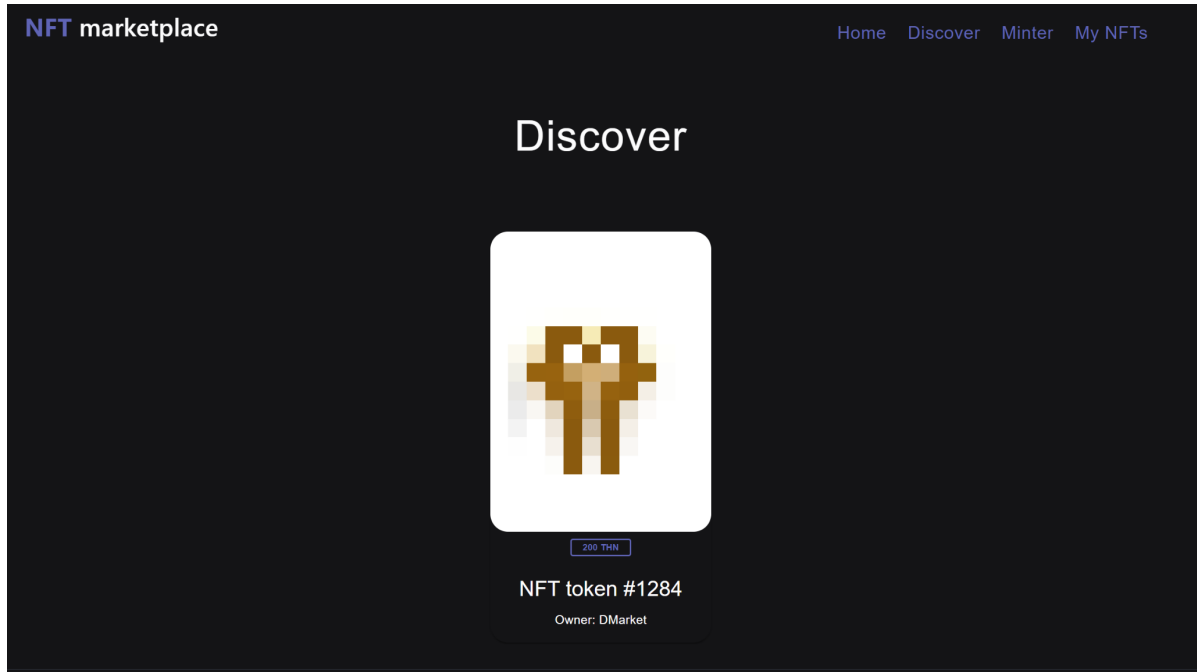


Рисунок 4.20 - токен який розміщено на маркетплейсі

Оскільки front-end має свій власний Principal ID, і нам потрібно перевірити можливість купівлі NFT токена, в такому разі можливий варіант завантаження токена через командний рядок, оскільки через нього є доступ до нашого справжнього Principal ID (див. рис. 4.21).

```
# Creating NFT for Testing
...
dfx canister call opend mint '(vec {137; 80; 78; 71; 13; 10; 26; 10; 0; 0; 0; 13; 73; 72; 68; 82; 0; 0; 0; 10; 0; 0; 0; 10; 8; 6; 0; 0; 0; 141; 50;
207; 189; 0; 0; 0; 1; 115; 82; 71; 66; 0; 174; 206; 28; 233; 0; 0; 0; 68; 101; 88; 73; 102; 77; 77; 0; 42; 0; 0; 0; 8; 0; 1; 135; 105; 0; 4; 0; 0;
0; 1; 0; 0; 0; 26; 0; 0; 0; 0; 0; 3; 160; 1; 0; 3; 0; 0; 0; 1; 0; 1; 0; 0; 160; 2; 0; 4; 0; 0; 0; 1; 0; 0; 0; 10; 160; 3; 0; 4; 0; 0; 1; 0; 0;
0; 10; 0; 0; 0; 59; 120; 184; 245; 0; 0; 0; 113; 73; 68; 65; 84; 24; 25; 133; 143; 203; 13; 128; 48; 12; 67; 147; 94; 97; 30; 24; 0; 198; 134;
1; 96; 30; 56; 151; 56; 212; 85; 68; 17; 88; 106; 243; 241; 235; 39; 42; 183; 114; 137; 12; 106; 73; 236; 105; 98; 227; 152; 6; 193; 42; 114; 40;
214; 126; 50; 52; 8; 74; 183; 108; 158; 159; 243; 40; 253; 186; 75; 122; 131; 64; 0; 160; 192; 168; 109; 241; 47; 244; 154; 152; 112; 237; 159;
252; 105; 64; 95; 48; 61; 12; 3; 61; 167; 244; 38; 33; 43; 148; 96; 3; 71; 8; 102; 4; 43; 140; 164; 168; 250; 23; 219; 242; 38; 84; 91; 18; 112;
63; 0; 0; 0; 73; 69; 78; 68; 174; 66; 96; 130;}, "NFT token #123")'
...
```

Рисунок 4.21 - код для завантаження NFT токена на маркетплейс

Після виконання попереднього коду буде виведений Principal ID створеного токена(див. рис. 4.22).

```
185, 108, 158, 159, 243, 40, 253, 180, 75, 122, 151,  
244; 38; 33; 43; 148; 96; 3; 71; 8; 102; 4; 43; 140;  
oken #123")'  
(principal "qoctq-giaaa-aaaaa-aaaea-cai")  
nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/OPEND$
```

Рисунок 4.22 - Principal ID новоствореного токена

Далі потрібно виконати команду для додавання NFT токена в список токенів які розміщені для продажу на маркетплейсі (див. рис. 4.23).

```
nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/OPEND$ dfx canister call opend listItem '(principal "qoctq-giaaa-aaaaa-aaaea-cai", 10)'  
("Success")  
nazar-hpt@DESKTOP-SR8LRSQ:~/ic-projects/OPEND$
```

Рисунок 4.23 - успішне додавання NFT токена на маркетплейс

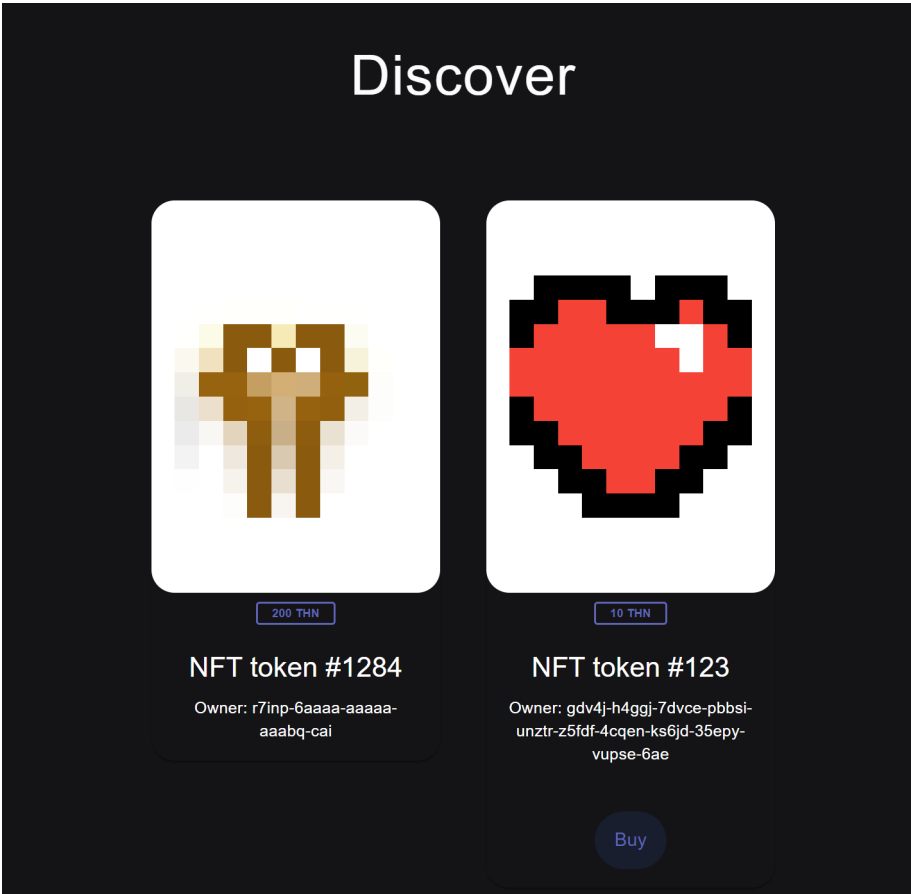


Рисунок 4.24 - NFT токен завантажений за допомогою командного рядка

Як видно з попереднього рисунку, тепер у користувача є можливість купити даний токен. Після натиснення кнопки Buy користувача буде сповіщено про успішну покупку NFT токена. Також буде стягнута відповідна сума з рахунку користувача (див. рис. 4.25).

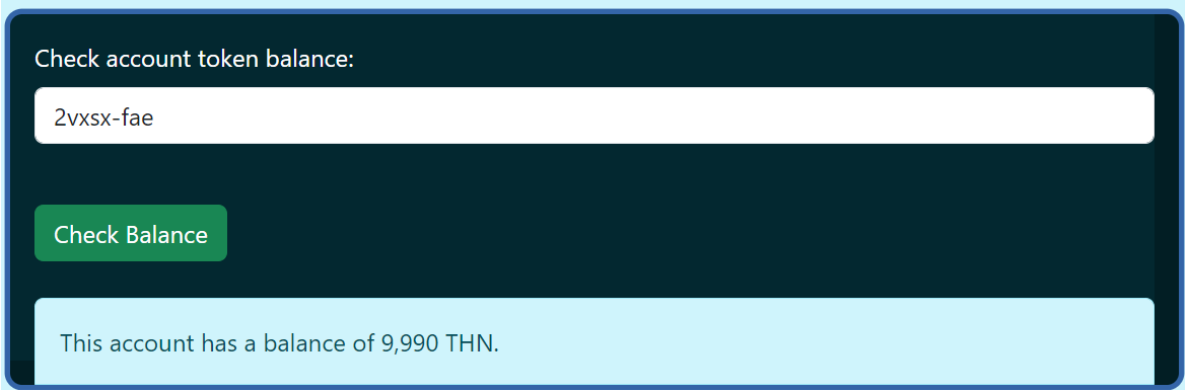


Рисунок 4.25 - рахунок користувача після купівлі NFT токена

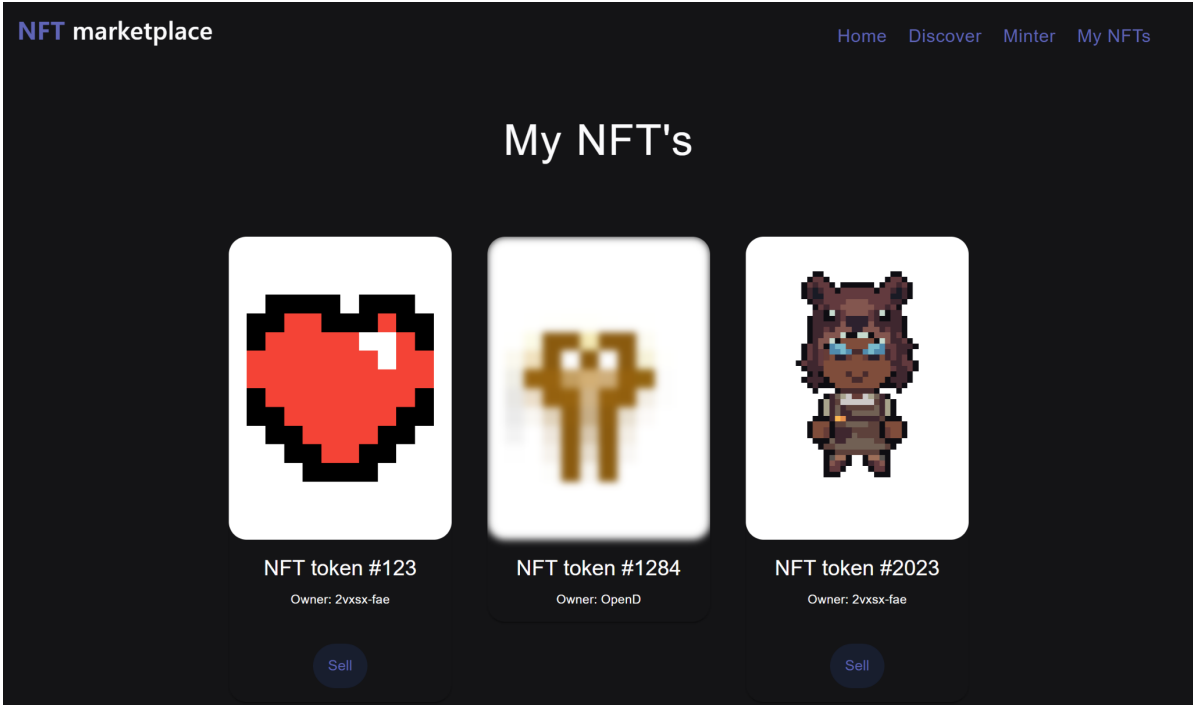


Рисунок 4.26 - список NFT токенів користувача після купівлі нового токена

5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

Під час роботи над розробкою децентралізованої платформи для розміщення NFT токенів, основна робота виконувалась за допомогою персонального комп'ютера, тому було важливо дотримуватись правил охорони праці, техніки безпеки та пожежної безпеки, які регламентують роботу з ПК.

Для забезпечення безпечної роботи з ПК та скорочення негативного впливу на здоров'я користувача, потрібно дотримуватись норм передбачених у наступних документах:

- Закон України «Про охорону праці»;
- НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», затверджені наказом Державного комітету України наказ Міністерства соціальної політики України від 14.02.2018 № 207;
- ДСанПІН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»;
- НПАОП 40.1-1.21-98. «Правила безпечної експлуатації електроустановок споживачів»;
- НАПБ А.01.001-2004 «Правила пожежної безпеки в Україні».

Перед початком розробки веб-платформи було проведено інструктажі з охорони праці, техніки безпеки та протипожежної безпеки.

При роботі над завданням з використанням ПК враховано, що площа на одне робоче місце повинна становити не менше ніж 6,0 м², а об'єм не менше ніж 20,0 м³; дотримано вимог стосовно освітлення, оптимальних умов мікроклімату, ергономічних характеристик основних елементів робочого місця, рівнів шуму, вібрації, електромагнітного, ультрафіолетового та інфрачервоного випромінювання та електростатичного поля викладених у ДСанПіН 3.3.2-007-98 [9].

Оскільки, основне навантаження під час роботи з ПК припадало на зорову систему, тому штучне освітлення в приміщеннях з робочими місцями, обладнаними ЕОМ та ПЕОМ, здійснювалось системою загального рівномірного освітлення, а значення освітленості на поверхні робочого столу в зоні розміщення документів становила 300-500 лк. За умов коли ці значення освітленості неможливо було забезпечити системою загального освітлення, використовувалось місцеве освітлення. Також, практикувались короткі перерви кожні 20 хв., що дозволило зменшити напруженість зорового нерву і відповідно знизити ймовірність його травмування. Згідно з рекомендаціями, час від часу виконувався спеціальний комплекс вправ для зниження втомлюваності очей та зменшення нервового напруження.

Щоб унеможливити ураження електричним струмом під час роботи було виконано наступні положення:

- лінія електромережі для живлення ЕОМ виконана як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів;

- ЕОМ підключаються до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення;

- заземлюючі конструкції, які знаходяться в робочому приміщенні було покрито ізоляцією, щоб унеможливити потрапляння працівника під напругу;

- у штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, є спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція така, що приєднання нульового захисного провідника відбувається раніше, ніж приєднання фазового та нульового робочого провідників[10].

Так, як проектування та розробка програмного забезпечення потребує високої концентрації уваги важливим аспектом є забезпечення шумоізоляції, для

цього обладнання, яке становило джерело шуму було розміщено в окремих приміщеннях.

Для захисту працівника ЕОМ та дотримання регламентованих норм світлового випромінення використовувалися екранні світлофільтри та локальні світлофільтри.

Все обладнання, яке використовувалося під час розробки програмного забезпечення було сертифіковане для використання на території України. Окрім усіх вище перерахованих вимог було виконано решту інструкцій описаних в законах та нормативно-правових актах, що стосуються охорони праці, техніки безпеки, протипожежної безпеки та електробезпеки при роботі з ЕОМ.

Враховуючи все вище описане, можна сказати що розробка децентралізованої платформи для розміщення NFT токенів була виконана згідно з вимогами охорони праці та техніки безпеки.

5.2. Безпека в надзвичайних ситуаціях

Трудова діяльність користувачів комп'ютерів відбувається в специфічному виробничому середовищі, яке має вплив на їх функціональний стан. Найбільш важливі фактори цього середовища - це фізичні чинники, такі як електромагнітні хвилі різних частот, електростатичні поля, шум, параметри мікроклімату та ряд світлотехнічних показників.

Ефективність роботи людини з комп'ютером в значній мірі залежить від її функціонального стану. Психофізіологічні та емоційні перенапруження, а також втома програміста можуть призвести до помилок в комп'ютеризованих системах керування, що, в свою чергу, може призвести до значних економічних втрат.

Трудовий процес користувачів комп'ютерів суттєво впливає на їх психофізіологічні можливості. Ця діяльність характеризується значними статичними фізичними навантаженнями, недостатньою руховою активністю та

напруженнями сенсорного апарату та вищих нервових центрів, які відповідають за функції уваги, мислення та регуляції рухів. Крім того, трудовий процес користувачів комп'ютерів супроводжується значними інформаційними навантаженнями.

Вплив професійних якостей та виробничого досвіду на роботу з комп'ютерами.

Професійні якості та виробничий досвід визначають внутрішні засоби діяльності користувачів комп'ютерів. Вони забезпечують надійну та безпомилкову роботу, а також дозволяють знаходити безпечні методи розв'язання виробничих завдань, навіть у нестандартних ситуаціях.

Зовнішні засоби діяльності, такі як організація робочого місця, форма та параметри його елементів, а також просторове розташування основного та допоміжного обладнання, можуть суттєво знизити фізичні та психофізіологічні навантаження, що діють на користувачів комп'ютерів[11].

Люди робота яких щільно пов'язана з комп'ютером стикаються з порушеннями органів зору, опорно-рухового апарату, центральної нервової системи, серцево-судинної системи, імунної системи та шкіри. Вони відзначають загальне недомагання, передчасне стомлення, головний біль та порушення функцій зору, які виникають через негативний психофізіологічний вплив на їх самопочуття та працездатність. Сучасна професія користувача ВДТ вимагає розумової праці, включаючи високу напруженість зорових функцій, одноманітну позу та багато стереотипних рухів кистями рук при обмеженій загальній руховій активності. Нервово-емоційний компонент також важливий, особливо при обмеженому часі. Крім того, робота з дисплеями на електронно-променевих трубках впливає на користувачів через фізичні фактори, такі як електростатичні поля, радіочастотне та рентгенівське випромінювання.

Оскільки робота користувачів комп'ютерів найчастіше відбувається в активній взаємодії з іншими людьми, виникає питання раціоналізації міжособистісних стосунків. Цей комплекс питань стосується як психологічних, так і соціально-психологічних аспектів трудових взаємовідносин, і є фактором

“ризик”, який впливає на функціональний стан користувачів комп’ютерів. Визначення та вивчення цих факторів дозволяє виділити основні причини виникнення станів напруженості, стомлення, стресу і вжити відповідні профілактичні заходи.

Трудова діяльність користувачів комп’ютерів відбувається у певному виробничому середовищі, яке впливає на їх функціональний стан. Виробниче середовище характеризується такими шкідливими факторами:

фізичні фактори виробничого середовища, до яких належать електромагнітні хвилі різних частотних діапазонів, електростатичні поля, шум, параметри мікроклімату та ціла низка світлотехнічних показників;

хімічні: пил, шкідливі хімічні речовини, що виділяються при роботі принтера і копіювальної техніки;

біологічні: підвищений вміст в повітрі патогенних мікроорганізмів (в приміщеннях з великою кількістю працівників, при недостатній вентиляції, та в період епідемії);

психофізіологічні: напруженнями сенсорного апарату (зору, уваги), вищих нервових центрів (інтелектуальні та емоційні навантаження), які забезпечують функції уваги, мислення, регуляції рухів, монотонність праці;

2) трудовий процес суттєво впливає на психофізіологічні можливості користувачів комп’ютерів, оскільки їх діяльність характеризується значними статичними фізичними навантаженнями; недостатньою руховою активністю. Окрім того, трудовий процес користувачів комп’ютерів відзначається значними інформаційними навантаженнями.

3) внутрішні засоби – це професійні риси та виробничий досвід, що обумовлюють надійну та без помилкову діяльність користувачів комп’ютерів, дозволяють знаходити безпечні методи розв’язання виробничих завдань навіть у нестандартних ситуаціях;

4) зовнішні засоби діяльності - визначаються ергономічними показниками щодо організації робочого місця, формою та параметрами його елементів, просторового розташування основного і допоміжного устаткування, можуть

суттєво знизити фізичні та психофізіологічні навантаження, що діють на користувачів комп'ютерів;

5) соціально-психологічні фактори трудових взаємовідносин.

В загальному, усі користувачі комп'ютерів поділяються на професіоналів та непрофесіоналів. До останніх можна віднести осіб, які використовують комп'ютер епізодично і він є для них не основним, а тільки допоміжним засобом (науково-технічні працівники, бібліотекарі, студенти, школярі, торговельні працівники та ін.).

Діяльність професіоналів можна поділити на три групи:

1 Діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження. Наприклад, робота операторів комп'ютерного набору, працівників довідкових служб[12].

2 Діяльність, яка пов'язана із здійсненням логічних операцій, що постійно повторюються. Це робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва.

3 Діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму. Наприклад, робота інженера-програміста, диспетчерів руху залізничного транспорту, аеропортів тощо.

Необхідно зазначити, що такий поділ досить умовний.

За даними ряду авторів у користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень, досить часто (40 – 70 %) виникають психологічні та поведінкові порушення (нервовість, роздратування, тривога, нерішучість, замкнутість, тощо).

Враховуючи несприятливий вплив цілого комплексу різноманітних виробничих факторів у користувачів можуть розвинути певні розлади здоров'я, що пов'язані з роботою за комп'ютером.

ВИСНОВКИ

У цій дипломній роботі я розглянув розробку децентралізованої веб-платформи для розміщення, майнінгу та продажу NFT токенів на базі React, Motoko та ICP. Використовуючи сучасні технології та унікальні можливості Internet Computer, створив потужний інструмент для взаємодії з NFT токенами.

Internet Computer — це революційна платформа, яка відкриває нові можливості для розробників, художників, фінансистів та всіх, хто цікавиться децентралізованими технологіями. Дана дипломна робота підтверджує, що децентралізовані веб-платформи — це не просто майбутнє, але й сучасність.

Internet Computer дозволяє нам забезпечити безпеку, швидкість та прозорість, що є ключовими аспектами в світі Web3.

Платформа дозволяє користувачам безпечно розміщувати свої NFT токени, майнити нові токени та здійснювати їх продаж. Вона побудована на React для зручного інтерфейсу користувача, Motoko для розробки розумних контрактів та ICP для забезпечення безпеки та швидкості.

Дана робота демонструє, як можна використовувати децентралізовані технології для створення інноваційних продуктів. Я вірю, що такі платформи в майбутньому стануть основою для розвитку цифрового мистецтва та фінансових інструментів.

Завдяки цій роботі, користувачі отримають доступ до нових можливостей та зможуть активно брати участь у світі NFT токенів. Ця платформа — це крок до майбутнього, де кожен може створювати, майнити та обмінювати NFT токени без посередників та забезпечувати безпеку своїх активів.

Internet Computer — це не просто блокчейн, це новий спосіб думати про розробку додатків, і моя робота відображає цей підхід. Я вірю, що децентралізовані платформи — це майбутнє, і я гордий, що можу бути частиною цього руху.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1.Що таке функціональні вимоги: приклади, визначення, повний посібник[Електронний ресурс] – Режим доступу: URL: <https://visuresolutions.com/uk/блог/функціональні-вимоги/>
- 2.[Електронний ресурс] – Режим доступу: URL: https://www.researchgate.net/publication/publication/320332845_Life_cycle_of_Analytical_Method
- 3.The library for web and native user interfaces [Електронний ресурс] – Режим доступу: URL: <https://legacy.reactjs.org/docs/getting-started.html>
- 4.Brown, C., & Miller, D. (2021). Programming Smart Contracts with Motoko. Springer.
- 5.[Електронний ресурс] – Режим доступу: URL: <https://developer.dfinity.org/>
- 6.[Електронний ресурс] – Режим доступу: URL: <https://webassembly.org/>
- 7.Williams, D., Zhang, L., & Thompson, J. (2020). Internet Computer: A Comprehensive Overview. Journal of Decentralized Computing, 15(2), 45-68.
- 8.Smith, A. (2019). Decentralized Applications with Internet Computer. O'Reilly Media.
9. ДСанПІН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин – 1998.
- 10.НПАОП 40.1-1.21-98 (ДНАОП 0.00-1.21-98) Правила безпечної експлуатації електроустановок споживачі. // Держнаглядохоронпраці (Державний комітет України з нагляду за охороною праці). – 1998.
- 11.Методичний посібник для здобувачі освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форми навчання «БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / В.С.Стручок -Тернопіль: ФОП Паляниця В. А., - 156 с. Отримано з <https://elartu.tntu.edu.ua/handle/lib/39196>.

12. Навчальний посібник «ТЕХНОЕКОЛОГІЯ ТА ЦИВІЛЬНА БЕЗПЕКА. ЧАСТИНА «ЦИВІЛЬНА БЕЗПЕКА»» / автор-укладач В. С. Стручок- Тернопіль: ФОП Паляниця В. А., - 156 с. Отримано з <https://elartu.tntu.edu.ua/handle/lib/39424>

13. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – Інженерія програмного забезпечення (Освітньо-професійна програма - «Програмне забезпечення систем», Освітньо-наукова програма - «Інженерія програмного забезпечення») для студентів усіх форм навчання / Упор.: М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк - Тернопіль: ТНТУ, 2020-51с.

ДОДАТКИ

Додаток А
Тези

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ
ХІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ
«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»



13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

УДК 004.7

Олег Пастух докт. техн. наук, професор, Назарій Гушпіт.

Тернопільський національний технічний університет імені Івана Пулюя

ГЛОБАЛЬНА СИСТЕМА ВЗАЄМОПОВ'ЯЗАНИХ КОМП'ЮТЕРНИХ МЕРЕЖ INTERNET COMPUTER

Oleh Pastukh Dr. Sc., Prof., Nazarii Hushpit

GLOBAL SYSTEM OF INTERCONNECTED COMPUTER NETWORKS INTERNET COMPUTER

Ключові слова: Internet Computer, блокчейн-технології, смарт-контракти, децентралізований-блокчейн, ICP.

Internet Computer - це децентралізована обчислювальна платформа на основі блокчейну, яка забезпечує безпечно та масштабоване середовище для смарт-контрактів і dApps. За допомогою Internet Computer можна:

Створювати децентралізовані веб-додатки: Розробники можуть створювати веб-додатки безпосередньо на Internet Computer, використовуючи “canister software”, що є еволюцією смарт-контрактів. Створення та обмін NFT та крипто-токенами: Internet Computer дозволяє користувачам створювати, купувати та продавати NFT та крипто-токени в децентралізованому середовищі. Безпосереднє розгортання додатків на Інтернеті: Розробники можуть розгортати свої додатки безпосередньо на Інтернеті без потреби в традиційних серверах або комерційних хмарних сервісах.

Це проект, спрямований на децентралізацію Інтернету, відхід від поточної моделі, де кілька великих корпорацій контролюють більшість онлайн-сервісів і даних. Це загальнодоступна мережа, яка об'єднує обчислювальну потужність численних незалежних центрів обробки даних по всьому світу. Ці центри обробки даних обслуговують вузлові машини, які працюють за протоколом Інтернет-комп'ютера, вдосконаленим криптографічним відмовостійким протоколом. Це гарантує безпеку смарт-контрактів, які працюють на блокчейні, їх неможливо підробити або зупинити.

Одним із ключових нововведень Internet Computer є концепція «каністерного програмного забезпечення», еволюція розумних контрактів. Програмне забезпечення Canister дозволяє розробникам створювати практично будь-яку систему або послугу безпосередньо в децентралізованій мережі. Це означає, що розробники можуть розгортати свої програми безпосередньо в Інтернеті, не потребуючи традиційних серверів або комерційних хмарних служб. Це не тільки знижує витрати, але й підвищує ефективність і надійність. Підсумовуючи, Internet Computer представляє революційний підхід до Інтернет-технологій. Завдяки децентралізації Інтернету та впровадженню інноваційних функцій, таких як каністерне програмне забезпечення та автономні безсерверні хмарні функції, він має потенціал змінити спосіб нашої взаємодії з цифровим світом. Як і будь-яка нова технологія, вона зіткнеться з проблемами та перешкодами, але потенційні переваги роблять її захоплюючою сферою вивчення та розвитку.

Інтернет-комп'ютер представляє сміливе бачення майбутнього Інтернету майбутнього, де контроль повертається користувачам, а інновації обмежені лише уявою.

Література

1. Internet Computer concepts. [Електронний ресурс]. – Режим доступу: <https://internetcomputer.org/docs/current/concepts>
2. How is the Internet Computer created by a protocol – [Електронний ресурс]: [<https://internetcomputer.org/what-is-the-ic/>]
3. Uses of the internet – [Електронний ресурс]: <https://thedigitalchain.com/uses-of-the-internet/>

Додаток Б
Диск з матеріалами магістерською