Міністерство освіти і науки України Львівський
національний університет імені Івана Франка Факультет
електроніки та комп’ютерних технологій

Кафедра радіоелектронних і комп’ютерних систем

КУРСОВА РОБОТА

« Створення NFT маркетплейсу на основі мережі Solana »

Виконав
Студент групи ФеІм-14
Яворів І.І.
Науковий керівник
проф. Монастирський Л.С.

Львів – 2023

Анотація

Ця курсова робота присвячена створенню маркетплейсу NFT (незмінних токенів) з використанням мережі Solana. NFT здобули значну популярність у сфері цифрового мистецтва та колекціонування. Мережа Solana пропонує швидкі транзакції та низькі комісії, що робить її ідеальним вибором для створення децентралізованого маркетплейсу.

Курсова робота має на меті дослідження процесу розробки функціонального маркетплейсу NFT на мережі Solana. Це включає завдання, такі як проектування користувацького інтерфейсу, інтеграція можливостей розумних контрактів Solana, реалізація безпечних платіжних систем та забезпечення ефективного зберігання та отримання метаданих NFT.

Курсова робота надасть всебічне розуміння мережі блокчейн Solana та її функцій. Вона також включатиме практичний досвід написання коду та розгортання розумних контрактів, взаємодію з API Solana та інтеграцію різних фреймворків для створення інтерактивного маркетплейсу.

Abstract

This coursework focuses on the creation of an NFT marketplace using the Solana blockchain network. NFTs, or Non-Fungible Tokens, have gained significant attention in the digital art and collectibles space. The Solana network offers fast transaction speeds and low fees, making it an ideal choice for building a decentralized marketplace.

The coursework aims to explore the process of developing a functional NFT marketplace on the Solana network. It involves tasks such as designing the user interface, integrating Solana's smart contract capabilities, implementing secure payment systems, and ensuring efficient storage and retrieval of NFT metadata.

The coursework will provide a comprehensive understanding of the Solana blockchain network and its features. It will also involve practical hands-on experience in coding and deploying smart contracts, interacting with the Solana API, and integrating various frontend frameworks to create an interactive marketplace.

Зміст

Анотація.....1

Abstract…….....…...1

Перелік використаних скорочень, умовних позначень, термінів….....4

Вступ.....5

1.Solana: ідеальна інфраструктура для ринку 7

1.1 Архітектура та механізм консенсусу мережі Solana 7

1.2 Масштабованість, висока пропускна здатність і низькі комісії 8

1.3 Функції безпеки та надійність мережі Solana..... 9

1.4 Переваги та недоліки мережі Solana, цілі та методологія дослідження... 10

2. **Розробка та впровадження маркетплейсу на базі Solana**..... 13

2.1 Ключові компоненти дизайну ринку, структура ринку, аутентифікація користувачів і розумні контракти..... 13

2.2 Токенізація активів та їх представлення в мережі Solana.....14

2.3 Інструменти розробки, розгортання смарт-контрактів та інтеграція гаманців Solana для взаємодії з користувачами 15

3. **Технології, що використовувались для створення сервісу**.....…17

3.1 LlamaIndex, ChatGPT, OpenAI.....17

3.2 Solana/web3.js, Shyft 19

3.3 React.js.....22

4. **Тестування та перевірка працездатності сервісу** 25

Висновки.....33

Використана література 35

Перелік використаних скорочень, умовних позначень, термінів

NFT (незамінний токен) - це унікальний цифровий актив, який підтверджує власність або автентичність певного об'єкта або контенту, такого як цифрове мистецтво, колекційні предмети чи віртуальна нерухомість, за допомогою технології блокчейну. Відмінною рисою NFT є його незамінність, оскільки кожен токен має унікальний ідентифікатор та особливі характеристики, які визначають його унікальну цінність.

PoH (proof-of-history) в мережі Solana - це механізм консенсусу, який забезпечує послідовність блоків шляхом використання часових міток на основі криптографічних підписів, що сприяє високій швидкості транзакцій та масштабованості.

pBFT (practical Byzantine Fault Tolerance) в мережі Solana - це алгоритм консенсусу, який забезпечує високу швидкість підтвердження транзакцій та стійкість до помилок шляхом досягнення консенсусу серед децентралізованої мережі в умовах можливих збоїв чи зловживань.

SDK (Software Development Kit) - це набір інструментів, бібліотек і документації, які надаються розробникам для створення програмного забезпечення для певної платформи, оперуючої системи або фреймворку.

Вступ

Децентралізовані ринкові майданчики привернули значну увагу в області технології блокчейн, пропонуючи нову парадигму для однорангової торгівлі та обміну активами. Ці ринки використовують потужність блокчейн-мереж для усунення посередників, підвищення безпеки та сприяння прозорості. У цьому контексті мережа Solana стала перспективною інфраструктурою для створення ефективних і масштабованих децентралізованих ринків.

Метою цієї курсової роботи є вивчення можливості створення ринку на основі мережі Solana, підкреслюючи її переваги та розглядаючи ключові питання дизайну та впровадження. Заглиблюючись в архітектуру, механізм консенсусу та унікальні особливості Solana, ми можемо зрозуміти, чому це ідеальний вибір для створення децентралізованих ринків. Також буде розглянуто ключові компоненти дизайну ринку, включаючи структуру ринку, аутентифікацію користувачів і смарт-контракти, а також токенізацію активів у мережі Solana.

Результати цього дослідження продемонструють важливість Solana для децентралізованих ринків, демонструючи її здатність забезпечувати безпечне, ефективне торгове середовище, орієнтоване на користувача. Крім того, ця курсова робота визначить майбутні напрямки досліджень для подальшого розширення можливостей ринків на основі Solana, таких як оптимізація дизайну смарт-контрактів, дослідження рішень для збереження конфіденційності та оцінка сумісності з іншими блокчейн-мережами.

Дослідницькі цілі цього проекту охоплюють різні аспекти створення ринку на основі мережі Solana та аналіз ефективності цього з точок зору розробника та користувача. По-перше, дослідження має на меті дослідити доцільність використання масштабованості Solana та низьких комісій за транзакції для створення ефективного ринку, який може обробляти великий обсяг транзакцій без надмірних витрат. Він передбачає аналіз технічних можливостей і обмежень Solana в контексті операцій на ринку.

По-друге, дослідження зосереджено на питаннях безпеки. Він перевіряє стійкість Solana до потенційних атак, оцінює ефективність механізму консенсусу щодо запобігання подвійним витратам та іншим шахрайським діям, а також досліджує найкращі методи захисту ринкових транзакцій і даних користувачів.

Крім того, дослідження заглиблюється в аспекти дизайну торгової площадки на базі Solana, включаючи структуру ринку та автентифікацію користувачів. У ньому досліджується, як можна використовувати функції Solana для створення зручного та інтуїтивно зрозумілого інтерфейсу ринку, забезпечуючи бездоганний досвід для покупців і продавців.

1. Solana: ідеальна інфраструктура для ринку

1.1 Архітектура та механізм консенсусу мережі Solana

Архітектура мережі Solana розроблена для вирішення проблем масштабованості, з якими стикаються багато інших блокчейн-платформ. Архітектура Solana складається з чотирьох основних рівнів: реплікованих кінцевих автоматів (RSM), proof-of-history (PoH), блоку обробки транзакцій (TPU) і сховища. Рівень RSM підтримує поточний стан мережі, а рівень PoH забезпечує запис усіх транзакцій. Рівень TPU обробляє перевірку та обробку транзакцій, а рівень зберігання власне зберігає дані блокчейну.

Унікальний консенсусний механізм, який застосовує Solana, — це proof-of-history(PoH). PoH вводить у мережу джерело часу, яке можна перевірити, генеруючи криптографічні докази, які встановлюють порядок і час для кожної транзакції. Така хронологія подій забезпечує ефективну перевірку транзакцій і консенсус між учасниками. Поєднуючи PoH із практичним консенсусом Byzantine Fault Tolerance (pBFT), Solana досягає швидкої остаточності та забезпечує згоду щодо створення транзакцій. Це дозволяє суттєво зменшувати час та комісію, що витрачається на проведення транзакції.

Багаторівнева архітектура та механізм консенсусу PoH дозволяють Solana обробляти транзакції паралельно на кількох вузлах, що забезпечує чудову масштабованість. На відміну від традиційних блокчейн-мереж, які часто стикаються з перевантаженнями та обмеженою пропускною здатністю, Solana продемонструвала здатність обробляти тисячі транзакцій за секунду з мінімальною затримкою. Розпаралелювання обробки транзакцій між вузлами

мережі дозволяє Solana масштабуватись горизонтально, враховуючи збільшені обсяги транзакцій без шкоди для швидкості чи ефективності.

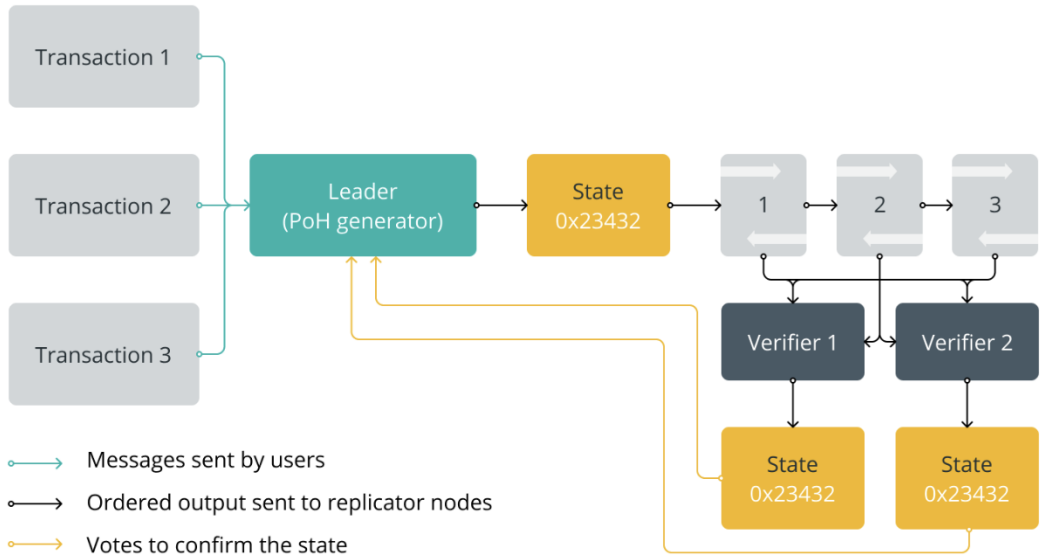

Рис 1.1. Схема механізму консенсусу в мережі Solana

1.2 Масштабованість, висока пропускна здатність і низькі комісії за транзакції

Масштабованість Solana є видатною особливістю, яка відрізняє її від багатьох інших блокчейн-мереж. Можливість паралельної обробки архітектури Solana забезпечує високу пропускну здатність, завдяки чому вона добре підходить для ринків із великим обсягом транзакцій. Ефективність Solana була перевірена і доведена, досягаючи тисяч транзакцій за секунду, навіть під час пікового використання.

Висока пропускна здатність мережі Solana досягається за рахунок різноманітних оптимізацій. Наприклад, консенсусний алгоритм Tower BFT визначає пріоритетність упорядкування та перевірки транзакцій, мінімізуючи

час, необхідний для досягнення консенсусу між учасниками транзакції. Крім того, Solana використовує прискорення на апаратному рівні, наприклад графічні процесори (GPU), для підвищення швидкості обробки транзакцій. Ці оптимізації сприяють здатності Solana обробляти значну кількість транзакцій у короткі терміни, гарантуючи, що учасники ринку можуть насолоджуватися швидкою та безперебійною торгівлею.

Окрім масштабованості та високої пропускної здатності, Solana пропонує низькі комісії за транзакції, що робить його привабливим варіантом для ринків. Ефективна конструкція мережі мінімізує споживання ресурсів, що призводить до економічно ефективної обробки транзакцій. Низькі комісії за транзакції зменшують бар'єри для входу та сприяють ширшій участі в екосистемі ринку. Користувачі можуть часто здійснювати транзакції без значних витрат, підвищуючи доступність і інклюзивність ринку.

1.3 Функції безпеки та надійність мережі Solana

Безпека є критично важливим аспектом будь-якого децентралізованого ринку, і Solana містить надійні функції безпеки для захисту від потенційних загроз і забезпечення цілісності транзакцій.

Solana використовує передові криптографічні методи, включаючи цифрові підписи та хеш-функції, для захисту транзакцій і даних користувачів. Ці криптографічні примітиви забезпечують автентичність, конфіденційність і захист записів транзакцій. Використовуючи надійну криптографію, Solana захищає від несанкціонованого доступу, маніпулювання даними та підробки, забезпечуючи безпечну основу для операцій на ринку.

Механізм консенсусу Solana, заснований на практичній Byzantine Fault Tolerance (pBFT), підвищує безпеку мережі. Через pBFT мережа валідаторів спільно перевіряє та підтверджує транзакції, забезпечуючи консенсус і запобігаючи маніпулюванню системою зловмисниками. Цей відмовостійкий підхід робить Solana стійким до атак і підвищує безпеку та надійність мережі.

Крім того, багаторівнева архітектура Solana та дизайн паралельної обробки сприяють її безпеці. Поділ обробки транзакцій і перевірки на окремі рівні зменшує ризик перевантаження мережі або маніпуляцій. У разі потенційної атаки розподілений характер мережі Solana гарантує відсутність єдиної точки збою, що робить її більш стійкою до зловмисних дій.

Загалом поєднання вдосконалених механізмів криптографії, консенсусу та принципів архітектурного дизайну робить мережу Solana надзвичайно безпечною та надійною. Його зосередженість на безпеці та стійкість забезпечує достовірність і надійність ринкових транзакцій, забезпечуючи міцну основу для створення децентралізованих ринкових майданчиків у мережі Solana.

1.4 Переваги та недоліки мережі Solana, цілі та методологія дослідження

Мережа Solana з її унікальними функціями та можливостями є ідеальною інфраструктурою для створення ефективних децентралізованих ринків. Однією з ключових переваг Solana є її масштабованість, яка дозволяє мережі обробляти значно більшу кількість транзакцій порівняно з багатьма іншими блокчейн-платформами. Завдяки своїй інноваційній архітектурі та

багаторівневому дизайну Solana досягає такої масштабованості шляхом розпаралелювання обробки транзакцій і використання високошвидкісного механізму консенсусу, відомого як Proof-of-History (PoH). Це дозволяє Solana запропонувати виняткову пропускну здатність і низьку затримку, що робить його дуже придатним для ринків, які потребують швидкої та ефективної обробки транзакцій.

Ще однією важливою перевагою Solana є низькі комісії за транзакції. У традиційних блокчейн-мережах високі комісії часто є перешкодою для входу, обмежуючи доступність децентралізованих ринків. Однак ефективний дизайн і економічна модель Solana гарантують, що користувачі можуть здійснювати транзакції з мінімальними комісіями, сприяючи ширшій участі та уможливлюючи мікротранзакції. Така економічна ефективність робить Solana привабливим вибором для ринків, де часто проводяться транзакції з невеликою вартістю.

Методологія дослідження, використана в цьому проекті, охоплює поєднання теоретичного аналізу та практичного впровадження. Це передбачає проведення обширного огляду інформації, щоб отримати дані з існуючих досліджень і найкращих галузевих практик, пов’язаних із децентралізованими ринками та мережею Solana. Дослідження також включає розробку прототипу сайту з використанням мов програмування, інструментів і фреймворків смарт-контрактів Solana. Ефективність ринку буде оцінюватися на основі таких критеріїв, як швидкість транзакцій, пропускна здатність, задоволеність користувачів і безпека.

Загалом це дослідження має на меті сприяти розумінню того, як мережу Solana можна ефективно використовувати для побудови децентралізованих ринків. Інформація, отримана в результаті цього дослідження, може допомогти компаніям, розробникам і окремим особам використовувати потенціал Solana для створення ефективних, безпечних і орієнтованих на користувача ринків, які переосмислюють спосіб торгівлі цифровими активами.

2. Розробка та впровадження Marketplace на базі Solana

2.1 Ключові компоненти дизайну ринку

Розробляючи ринок у мережі Solana, важливо враховувати різні ключові компоненти, які сприяють його функціональності та взаємодії з користувачем. Важливим аспектом є структура ринку, яка визначає, як покупці та продавці взаємодіють на ньому. Це включає визначення типу торгової моделі, такої як модель на основі книги замовлень або автоматизованої моделі маркет-мейкера (AMM), а також впровадження необхідних алгоритмів і протоколів для розміщення замовлень, зіставлення замовлень і розрахунків за угодами. Структура ринку має бути розроблена таким чином, щоб забезпечити чесну та ефективну торгівлю, забезпечуючи безперебійний досвід для користувачів.

Аутентифікація користувача є ще одним важливим компонентом дизайну ринку. Впровадження надійної та безпечної системи аутентифікації має важливе значення для перевірки ідентичності учасників і захисту від несанкціонованого доступу. Це може включати інтеграцію традиційних механізмів автентифікації, таких як комбінації імені користувача та пароля, або використання децентралізованих рішень ідентифікації, таких як протоколи автентифікації на основі блокчейну. Впроваджуючи надійні заходи автентифікації користувачів, ринок може забезпечити цілісність облікових записів користувачів і підвищити довіру між учасниками.

Розумні контракти відіграють ключову роль в автоматизації ринкових процесів і встановленні довіри між сторонами. Вони дозволяють виконувати попередньо визначені правила та умови, забезпечуючи прозорі та незмінні

взаємодії. Розробка та реалізація смарт-контрактів відповідно до вимог ринку передбачає визначення структури контракту, функцій і тригерів подій. Ці контракти можуть сприяти різноманітним операціям, зокрема виконанню замовлень, розрахункам за платежами, вирішенню суперечок і забезпеченню виконання умов умовного депонування. Добре розроблені та безпечні розумні контракти підвищують надійність і ефективність ринкових транзакцій.

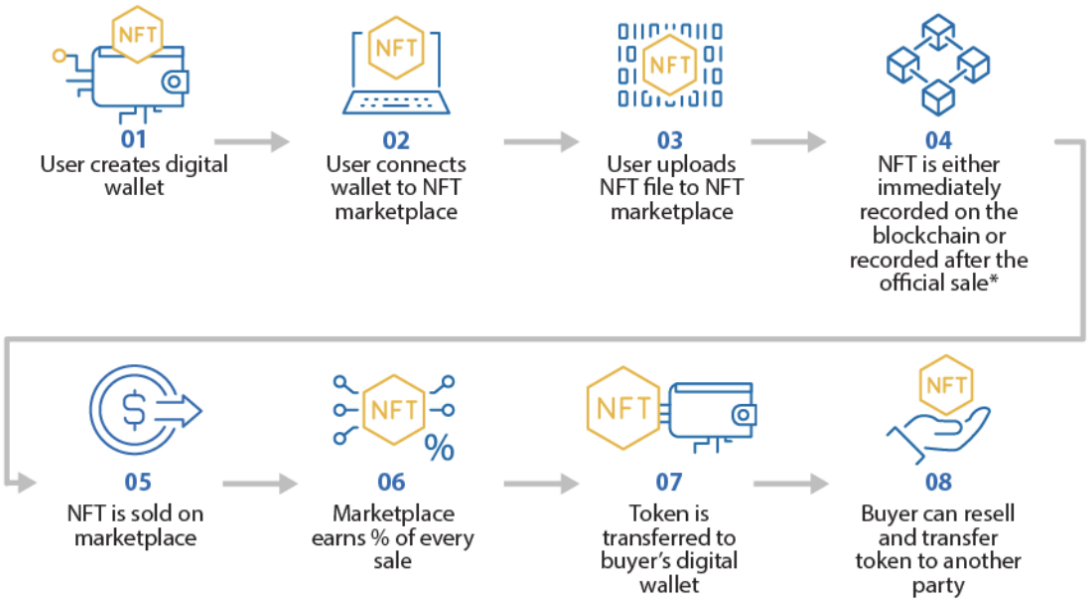

Рис 2.1. Схема роботи NFT маркетплейсу

2.2 Токенізація активів у мережі Solana

Токенізація є трансформаційною функцією мережі Solana, що дозволяє представляти як цифрові, так і реальні активи як токени в блокчейні. Завдяки токенізації активів ринок може забезпечити безперебійну передачу та торгівлю цими активами в екосистемі. У мережі Solana токенізація полегшується через бібліотеку програм Solana (SPL), яка забезпечує основу для створення взаємозамінних і незамінних токенів.

Токенізація на Solana передбачає створення токенів і керування ними за допомогою стандартів SPL. Взаємозамінні токени представляють подільні та взаємозамінні активи, такі як криптовалюти, тоді як невзаємозамінні токени (NFT) представляють собою унікальні та неподільні активи, такі як цифрові твори мистецтва або предмети колекціонування. Впровадження токенізації на ринку забезпечує часткове володіння, підвищує ліквідність і спрощує передачу активів. Це відкриває нові можливості для учасників брати участь у торгівлі активами, інвестиціях і створенні вартості на ринку Solana.

2.3 Інструменти розробки, розгортання смарт-контрактів та інтеграція гаманця Solana

Розробка торгової площадки в мережі Solana вимагає використання відповідних інструментів розробки та фреймворків. Solana надає ряд інструментів для полегшення розробки програм і взаємодії з мережею. Інтерфейс командного рядка Solana (CLI) дозволяє розробникам взаємодіяти з блокчейном Solana, розгортати та тестувати смарт-контракти та контролювати мережеву активність. Solana Software Development Kit (SDK) містить бібліотеки та утиліти для створення програм на Solana, спрощуючи процес розробки. Крім того, бібліотека Solana Web3.js забезпечує безперебійну інтеграцію функцій Solana у веб-додатки.

Розгортання смарт-контрактів у мережі Solana передбачає написання, компіляцію та розгортання смарт-контрактів, які визначають бізнес-логіку та правила ринку. Solana підтримує кілька мов програмування, включаючи Rust,

Solidity і C, що робить її доступною для розробників з різними мовними уподобаннями. Розробники можуть використовувати спеціальні інструменти розробки Solana, як-от Solana CLI, або веб-інтерфейси, як-от Solana Studio, для безпечного складання та розгортання смарт-контрактів. Ці інструменти спрощують процес розгортання та надають основні функції для керування контрактами та тестування.

Інтеграція гаманців Solana в користувальницький інтерфейс ринку є важливою для безперебійної взаємодії користувачів. Гаманці Solana дозволяють користувачам безпечно керувати своїми особистими ключами, підписувати транзакції та взаємодіяти з ринком. Використання популярних варіантів гаманців Solana, таких як Phantom Wallet, Sollet або Solflare, дозволяє користувачам отримувати доступ до своїх активів, переглядати баланси та безперешкодно виконувати транзакції. Завдяки інтеграції гаманців Solana ринок забезпечує зручність для користувачів, спрощуючи керування активами, виконання транзакцій і участь у ринковій площі Solana.

Враховуючи структуру ринку, автентифікацію користувачів, розробка та впровадження смарт-контрактів, а також токенізація активів та інтеграція інструментів розробки та гаманців Solana, ринок можна спроектувати та впровадити, щоб забезпечити безпечне, ефективне та орієнтоване на користувача торгове середовище в мережі Solana. Ці компоненти формують основу для успішного ринку, дозволяючи користувачам брати участь у децентралізованій торгівлі з легкістю та впевненістю.

3. Технології, що використовувались для виконання завдання

3.1 LlamaIndex, ChatGPT, OpenAI

Основна місія LlamaIndex — забезпечити інтерфейс між великими мовними моделями (дані, що використовує ChatGPT) і приватними зовнішніми даними. За останні кілька місяців він став одним із найпопулярніших фреймворків з відкритим вихідним кодом для розширення даних LLM (генерування з доповненим контекстом) для різноманітних випадків використання: відповіді на запитання, підсумовування, структуровані запити тощо.

В основі LlamaIndex дані надані користувачем є основними: Потім Retriever отримує найбільш релевантні вузли з індексу за запитом. Нарешті, QueryEngine синтезує відповідь на запит і отримані вузли.

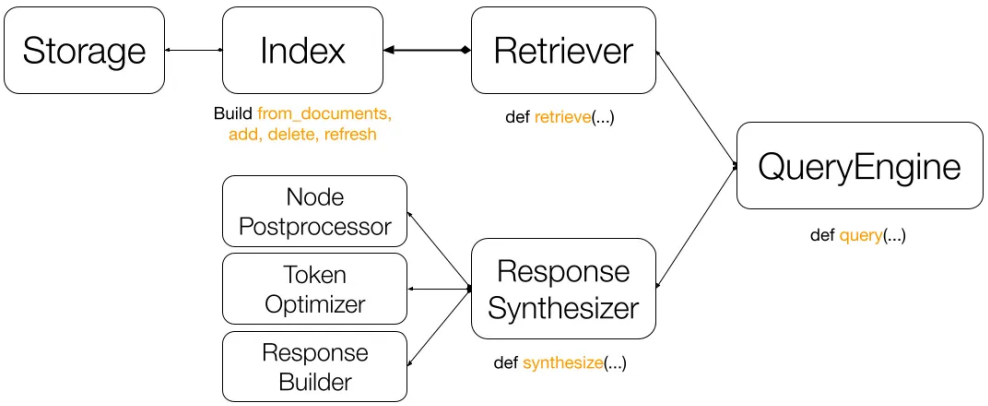

Рис.3.1 Схема роботи LlamaIndex

Загалом можна сказати, що LlamaIndex — це «фреймворк даних», який допоможе вам створювати програми LLM. Він надає такі інструменти:

- Пропонує з’єднування даних для прийому існуючих джерел даних і форматів даних (API, PDF-файли, документи, SQL тощо)
- Надає способи структурування даних (індекси, графіки), щоб ці дані можна було легко використовувати з LLM.
- Забезпечує розширений інтерфейс пошуку/запиту над вашими даними: можна надсилати будь-який запит на введення LLM, та отримати на виході контекст і вихідні дані, доповнені знаннями.
- Дозволяє легко інтегрувати із зовнішнім фреймворками програми (наприклад, із LangChain, Flask, Docker тощо).

LlamaIndex надає інструменти як для початківців, так і для досвідчених користувачів. Високорівневий API дозволяє користувачам-початківцям використовувати LlamaIndex для отримання та запиту своїх даних. API нижнього рівня дозволяють досвідченим користувачам налаштовувати та розширювати будь-які модулі (з’єднувачі даних, індекси, засоби пошуку, механізми запитів, модулі переранжування) відповідно до своїх потреб.

Розглянемо використання LlamaIndex на прикладі даного проекту:

LlamaIndex - це Python бібліотека, тому потрібно запустити локальний сервер, щоб мати можливість виконувати запити з нашого сайту. Для цього використаємо python фреймворк - Flask. Та виконаємо імпорти всіх необхідних бібліотек:

```
from flask import Flask, request, jsonify
from llama_index import LLMPredictor, GPTSimpleVectorIndex, SimpleDirectoryReader, PromptHelper, ServiceContext
```

```
from langchain.chat_models import ChatOpenAI
```

По стандарту дана бібліотека використовує модель з назвою “text-davinci-003” але в розробників є можливість вибрати будь-яку іншу із списку загальнодоступних. Для виконання цього завдання була вибрана модель - “gpt-3.5-turbo”, що найкраще підходить для аналізу великого масиву даних. Також було вказано декілька параметрів, що визначають довжину вхідного та вихідного текстів:

```
llm_predictor = LLMPredictor(llm=ChatOpenAI(temperature=0 ,model_name ="gpt-3.5-turbo"))
max_input_size = 8192
num_output = 4096
max_chunk_overlap = 40
```

Далі вже використовуючи функції PromptHelper, ServiceContext та GPTSimpleVectorIndex у нас є можливість передати у вибрану модель певний набір даних та запит, по якому AI буде наші дані аналізувати. На виході ми отримаємо рядок даних, що є відповіддю на наше запитання:

```
prompt_helper = PromptHelper(max_input_size, num_output, max_chunk_overlap)
service_context = ServiceContext.from_defaults (llm_predictor=llm_predictor,
prompt_helper=prompt_helper)
documents = SimpleDirectoryReader(path).load_data()
index = GPTSimpleVectorIndex.from_documents(documents, service_context=service_context)
response = index.query(quote)
```

3.2. Solana/web3.js, Shyft

Бібліотека `@solana/web3.js` — це потужна та широко використовувана бібліотека JavaScript для взаємодії з блокчейном Solana. Solana — це високопродуктивна блокчейн-платформа, розроблена для децентралізованих програм і масштабованих децентралізованих сервісів.

Бібліотека @solana/web3.js надає розробникам повний набір інструментів і функцій для взаємодії з блокчейном Solana. Він служить мостом між програмами чи сайтами написаними мовою програмування JavaScript і мережею Solana, дозволяючи розробникам створювати децентралізовані сервіси, взаємодіяти зі смарт-контрактами, надсилати транзакції та отримувати дані блокчейну.

Однією з ключових особливостей `@solana/web3.js` є його підтримка асинхронних операцій, завдяки чому він добре підходить для виконання складних і трудомістких завдань у блокчейні. Це дозволяє розробникам легко надсилати та підписувати транзакції, взаємодіяти зі смарт-контрактами та асинхронно отримувати дані з блокчейну, забезпечуючи ефективне та безблокове виконання.

Бібліотека також пропонує широкий спектр службових функцій і класів, які спрощують процес розробки. Він надає функції для керування ключами, операцій із маркерами, створення облікових записів, розгортання програм тощо. Ці утиліти абстрагуються від складнощів взаємодії з блокчейном Solana, дозволяючи розробникам зосередитися на створенні своїх програм, а не мати справу з деталями низького рівня.

Крім того, `@solana/web3.js` підтримує протоколи RPC (Remote Procedure Call) і WebSocket, забезпечуючи гнучкість і дозволяючи розробникам вибирати метод зв’язку, який найкраще відповідає їхнім потребам. Ця універсальність забезпечує оновлення в реальному часі та ефективний зв’язок із мережею Solana.

Іншою важливою особливістю `@solana/web3.js` є його велика документація та активна підтримка спільноти. Спільнота Solana надає вичерпну документацію, посібники та приклади, що полегшує розробникам розуміння та ефективне використання бібліотеки. Крім того, спільнота активно сприяє розвитку бібліотеки, забезпечуючи регулярні оновлення, виправлення помилок і нові функції.

Підсумовуючи, бібліотека `@solana/web3.js` є надійним і важливим інструментом для розробників, які хочуть створювати децентралізовані програми на блокчейні Solana. Завдяки широкому набору функцій, асинхронній підтримці, службовим функціям і активній підтримці спільноти бібліотека дає змогу розробникам створювати масштабовані та ефективні програми, які використовують потужність блокчейну Solana.

Shyft — це потужна платформа, яка полегшує взаємодію з різними блокчейн-мережами, включаючи Solana. Вона пропонує широкий спектр функцій, щоб дозволити розробникам взаємодіяти з блокчейном Solana та пов’язаними службами.

Endpoint create_detach дозволяє створювати NFT (незамінні токени) у блокчейні Solana. Це суттєво спрощує процес “мінтування” та зв’язування метаданих із NFT, полегшуючи розробникам керування та розгортання NFT на Solana.

Тіло запиту має бути у форматі multipart/form-data та містити такі поля:

мережа : назва мережі (це залежить від типу мережі, для якої нам потрібно створити токен).

адреса гаманця: відкритий ключ гаманця, пов’язаного з NFT.

ім'я : назва NFT.

Додаткові поля, що стосуються вашої реалізації NFT (наприклад, символ, опис, атрибути тощо).

Щоб взаємодіяти з цим ендпоїнтом, можна використовувати клієнтську бібліотеку HTTP, таку як axios у JavaScript, або будь-яку еквівалентну бібліотеку іншою мовою програмування.

3.3 React.js

React.js, часто називають React, — це потужна бібліотека JavaScript, розроблена Facebook для створення інтерфейсів користувача (UI). Він набув величезної популярності та став вибором для розробників завдяки своїй ефективності, можливості повторного використання та декларативному підходу. У цьому розділі буде розглянуто ключові концепції React.js, щоб проілюструвати його використання та переваги.

React.js базується на концепції компонентів, які є самодостатніми, багаторазово використовуваними будівельними блоками, які інкапсулюють частину інтерфейсу користувача та її поведінку. Компоненти можна складати та вкладати для створення складних структур інтерфейсу користувача. Цей модульний підхід дозволяє розробникам розбивати великі інтерфейси користувача на менші керовані частини, покращуючи організацію коду та можливість повторного використання.

React.js використовує віртуальну DOM (модель об’єктів документа) для ефективного оновлення та відтворення компонентів інтерфейсу користувача. Віртуальний DOM є спрощеним представленням фактичного DOM. Коли стан компонента або властивості змінюються, React порівнює віртуальний DOM з попередньою версією, щоб визначити мінімальний набір змін, необхідних для оновлення фактичного DOM. Цей підхід мінімізує непотрібні маніпуляції з DOM, що призводить до швидкого відтворення та покращення продуктивності.

React.js використовує декларативний синтаксис, що означає, що розробники описують, як вони хочуть вигляд інтерфейсу користувача, а React піклується про оновлення інтерфейсу користувача відповідно до бажаного стану. Цей підхід спрощує розробку інтерфейсу користувача шляхом абстрагування від низькорівневих маніпуляцій DOM.

Компонентна архітектура React сприяє повторному використанню коду, модульності та зручності обслуговування. Розробники можуть створювати інкапсульовані компоненти з чіткими межами, що полегшує розуміння, тестування та оновлення окремих частин програми, не впливаючи на весь інтерфейс користувача.

Використовуючи віртуальний DOM і його ефективний алгоритм розрізнення, React мінімізує кількість фактичних маніпуляцій DOM, необхідних під час оновлень. Цей підхід покращує продуктивність і покращує взаємодію з користувачем, особливо в складних інтерфейсах користувача, де часто відбуваються зміни

React.js має активну та підтримуючу спільноту, яка сприяє зростанню розгалуженої екосистеми бібліотек, інструментів і ресурсів. Розробники можуть скористатися цими внесками спільноти, щоб покращити свій досвід розробки React, оптимізувати робочі процеси та вирішувати різноманітні виклики.

React.js зробив революцію в тому, як розробники створюють інтерфейс користувача в додатках JavaScript. Його компонентна архітектура, віртуальний DOM, декларативний синтаксис і оптимізація продуктивності зробили його популярним вибором для створення масштабованих, ефективних і підтримуваних інтерфейсів користувача. Розбиваючи інтерфейси користувача на багаторазові компоненти, React дозволяє розробникам створювати складні програми, зберігаючи при цьому організацію коду та покращуючи можливість повторного використання. Завдяки широкій підтримці спільноти та екосистемі React продовжує розвиватися та дає змогу розробникам створювати чудові умови для користувачів у мережі.

4. Тестування та перевірка працездатності сервісу

При вході на сайт нас зустрічає форма, за допомогою якої користувач може створити власну NFT, а також проста панель навігації з кнопкою “Connect Wallet” при натисканні на яку, сайт отримує доступ до адреси крипто гаманця користувача, якщо він використовує будь-яке розширення для браузера, що підтримує мережу Solana.

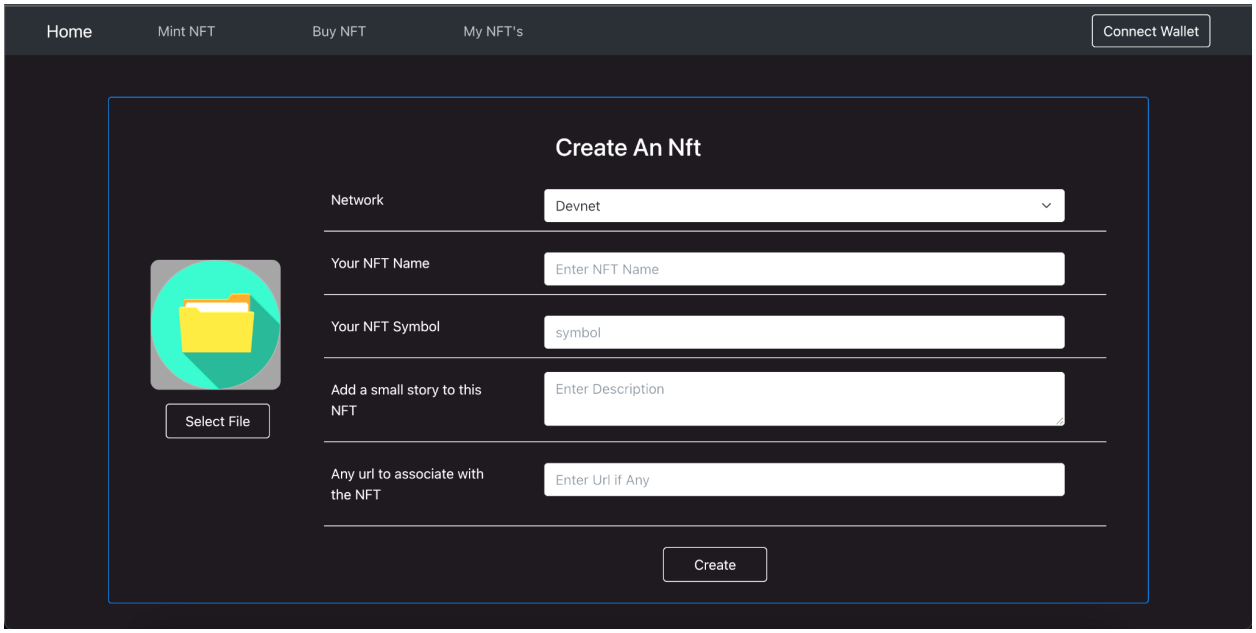

Рис 4.1. Загальний вигляд вікна Mint NFT

Після того, як користувач вибере потрібен гаманець, кнопка зникне, а на її місці відобразиться адреса крипто гаманця

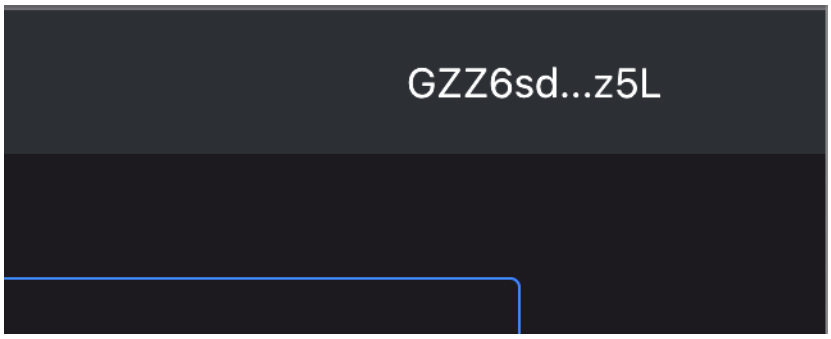

Рис 4.2. Адреса крипто гаманця користувача

Після цього користувач може створювати власні NFT просто заповнивиши наведену нище форму, яка має декілька стандартних полів:

1. Network - мережа, на якій буде створюватись токен:
    - Mainnet - основна мережа Solana
    - Devnet - мережа для розробників (комісія не стягується)
    - Testnet - мережа для тестувальників
2. NFT Name - назва NFT, яку створює користувач
3. NFT Symbol - символ NFT, яку створює користувач
4. Description - невеликий опис NFT, що дасть змогу потенційним покупцям познайомитись ближче з проектом який підтримує даний токен, або з ідеєю автора
5. Associate URL - посилання на сервіс, що підтримує дану колекцію або (як в даному випадку) на AI, що згенерував зображення

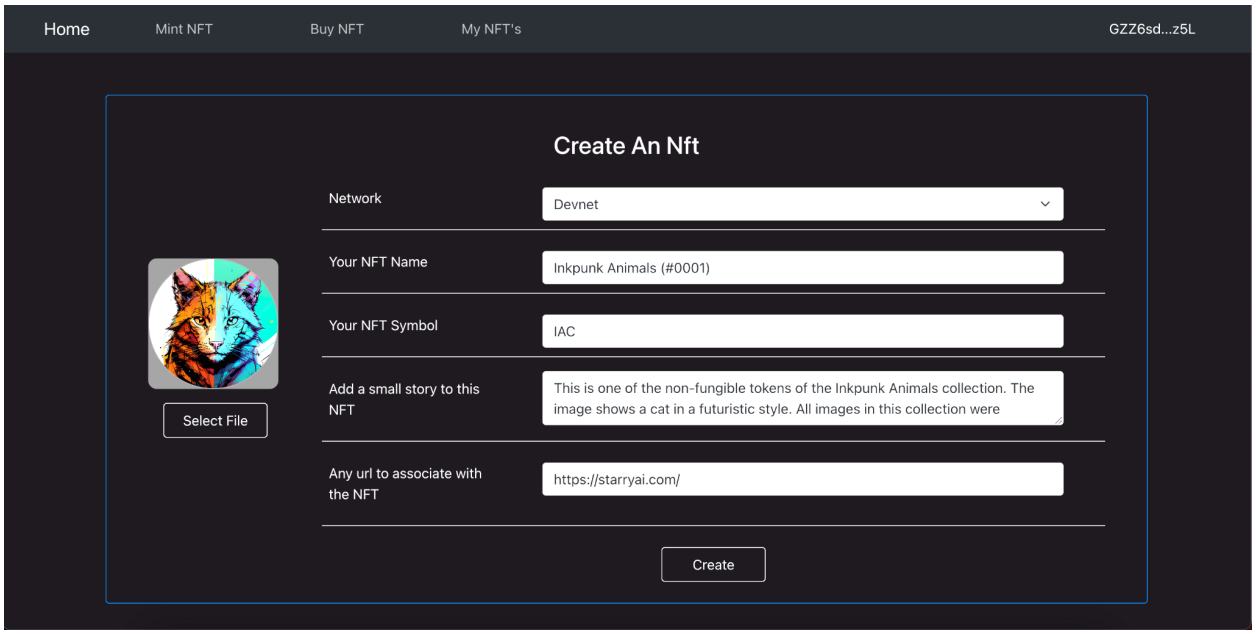

Рис 4.3. Заповнена форма створення нової NFT

Після натискання кнопки Create користувачу відкриється вікно з підтвердженням транзакції. Як бачимо комісія за транзакцію, що створить

NFT на мережі Solana дійсно невелика, вона складає 0.012 SOL, що на даний момент рівне приблизно 0.23$. Після підтвердження транзакції NFT з'явиться в крипто гаманці користувача і також на сайті на вкладці My NFT's.

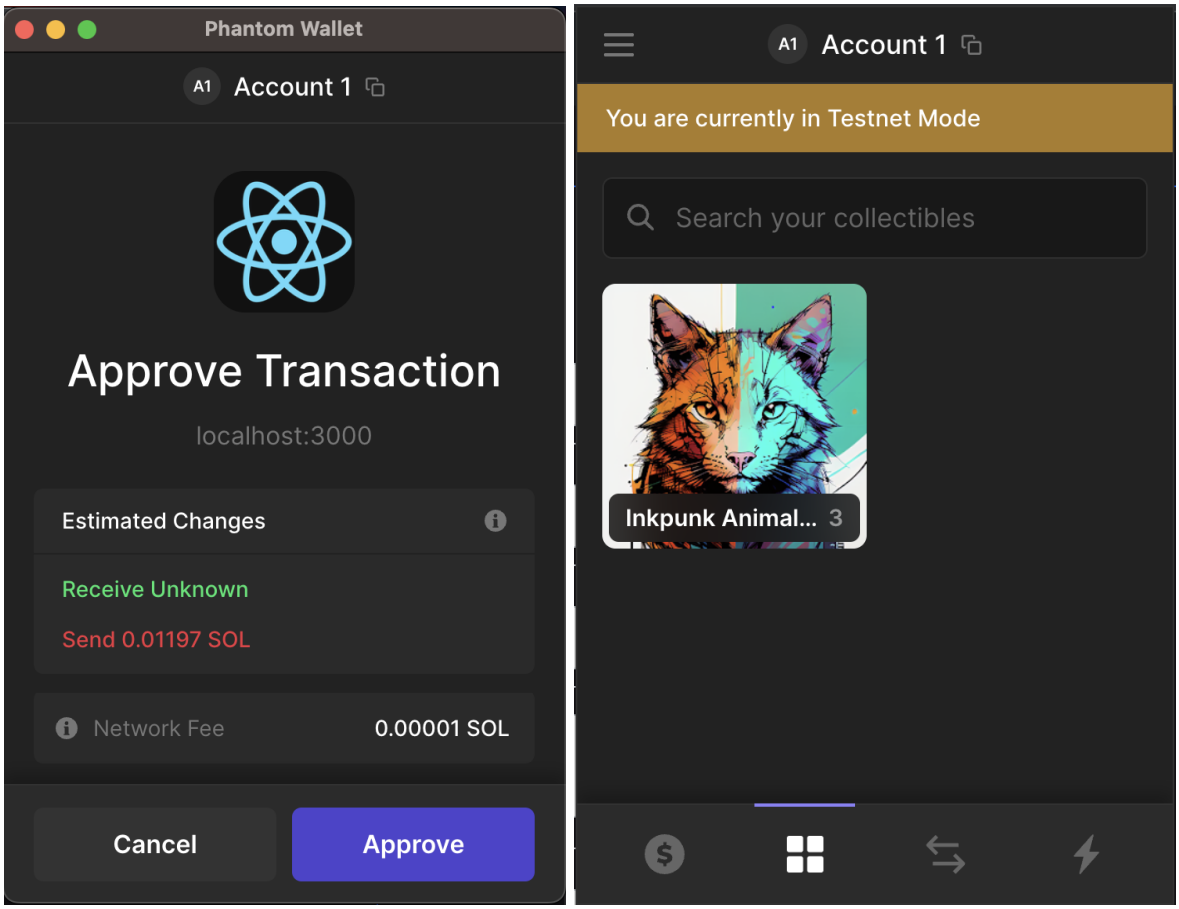

Рис 4.4. Підтвердження транзакції для створення NFT

Перейшовши на вкладку My NFT's користувач може побачити токени, що є у нього на акаунті. На прикладі показано 3 токени з однієї колекції, що були згенеровані за допомогою даного ресурсу. Також крім основної інформації в користувача ще є можливість продати чи надіслати цей токен.

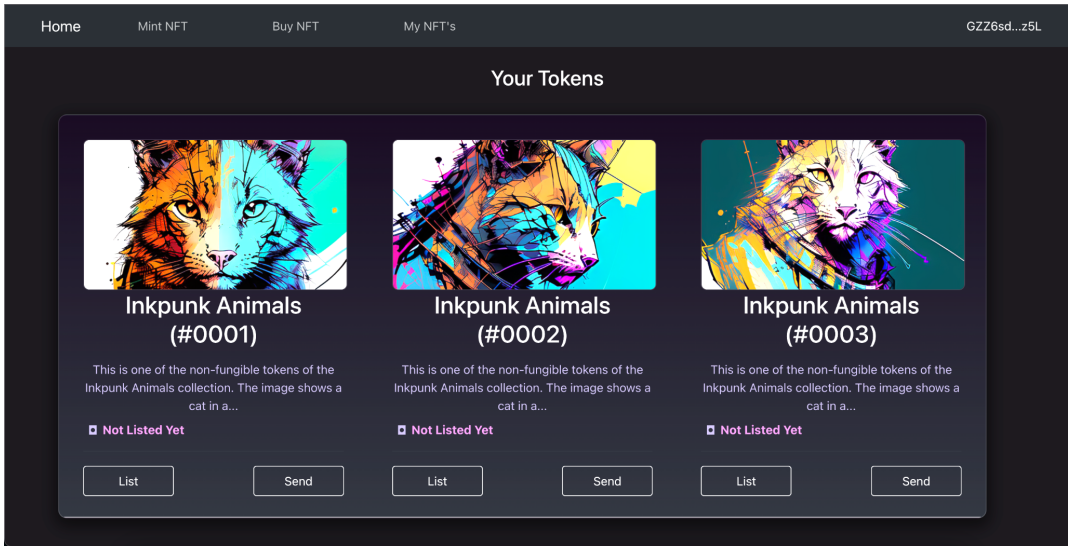

Рис 4.5. Загальний вигляд вікна Buy NFT

Після натискання кнопки Send у нас відкриється вікно де потрібно вказати адресу отримувача та натисну Next після цього пройде транзакція і користувач отримує про це сповіщення.

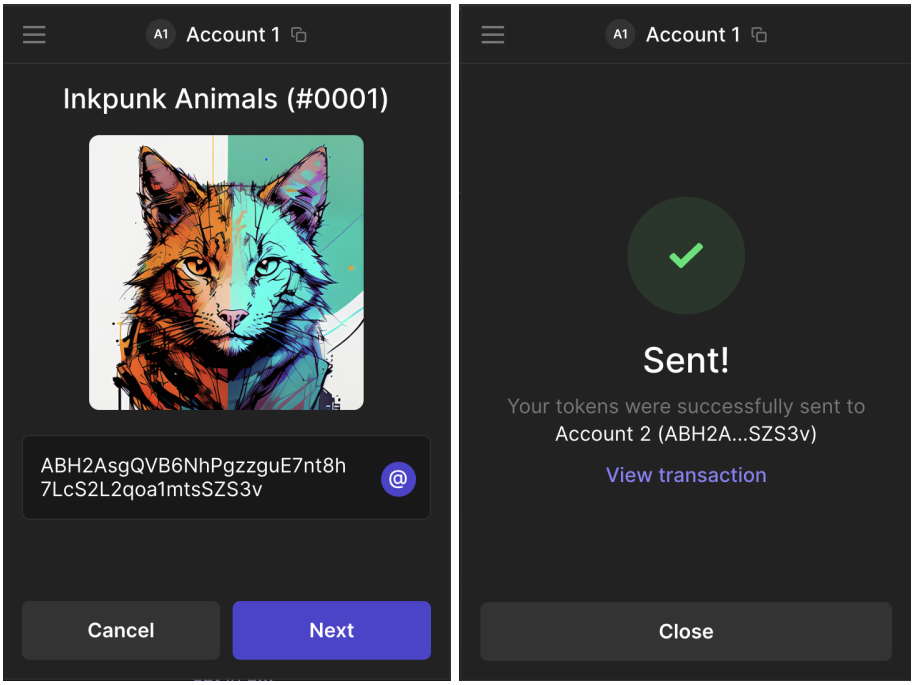

Рис 4.6. Вікно надсилання токена

Як бачимо на акаунті на який ми відправляли цей токен він також з’явився

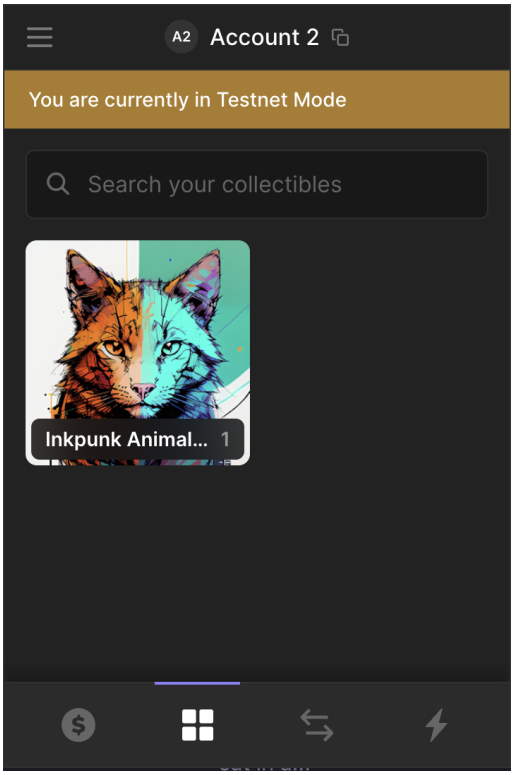

Рис 4.7. Список токенів акаунту #2

Аналогічним чином за допомогою кнопки List можна виставляти токени на біржі.

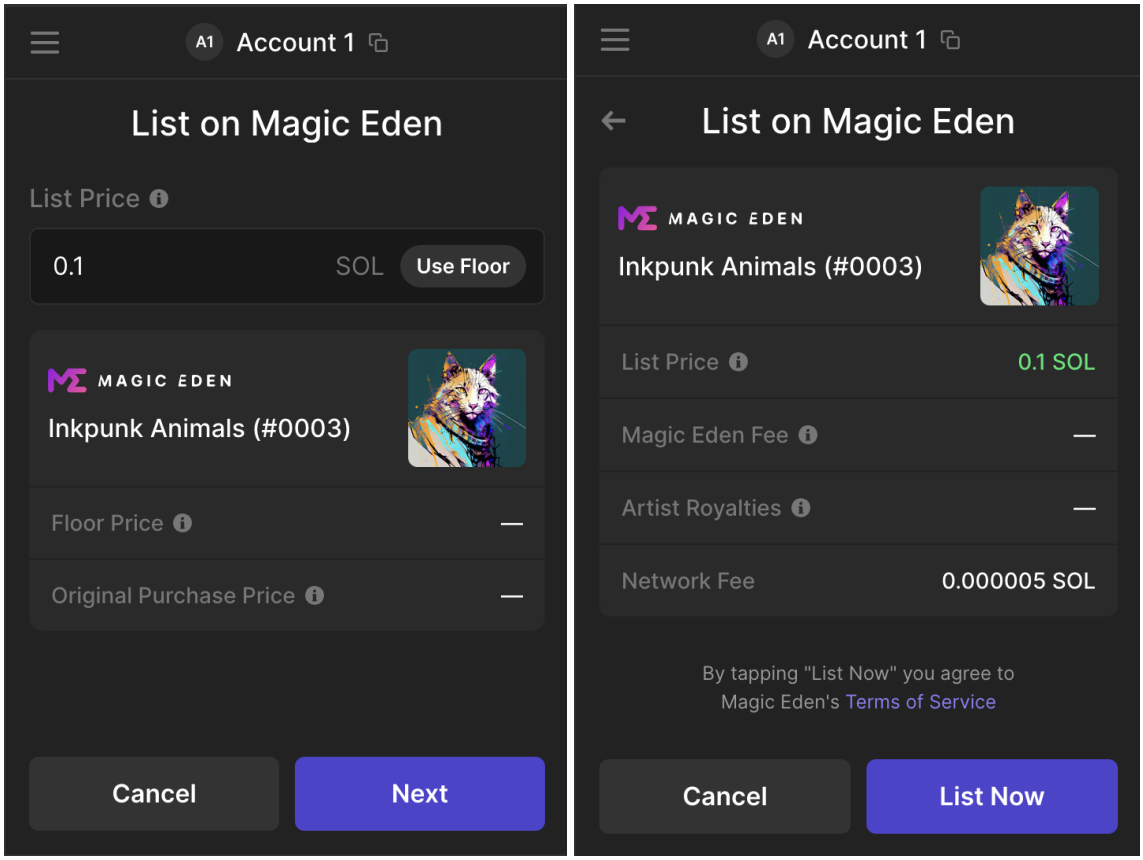

Рис 4.8. Процес лістингу токенів на біржі

Також на сайті є ще одне вікно, де користувач вже може купити чи поставити ставку на той чи інший токен. Загалом на сторінці відображається 100 токенів які є найпопулярнішими за останню годину на біржі Solana, але при потребі після скролу донизу користувач може довантажити ще 100 додаткових токенів. і також тут реалізований механізм “розумного пошуку”, що дозволяє користувачу ввести в текстовому полі будь-які дані та за допомогою ChatGPT буде підібрано токен, що найкраще відповідає опису.

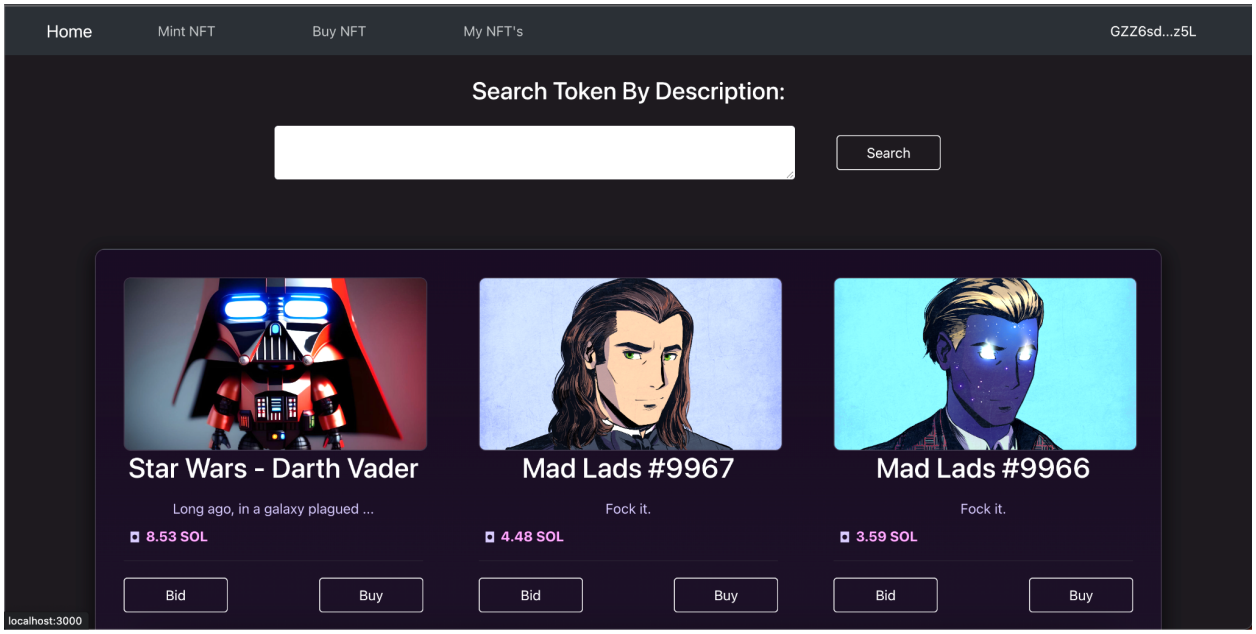

Рис 4.9. Загальний вигляд вікна Buy NFT

Давайте протестуємо пошук задавши наступний запит:

"*I'm looking for a token with an ape and a price of up to 10 SOL*"

Оскільки ми знаємо, що дані в базі даних ChatGPT є тільки до 2021 року, а нові токени створюються кожного дня, деякі втрачають свою популярність, а інші навпаки укріплюють свої позиції є необхідність задання якоїсь нової інформації для AI, це було реалізовано за допомогою LlamaIndex та було описано в пункті роботи #3.1. Загалом аналізується 10000 найпопулярніших токенів на даний момент.

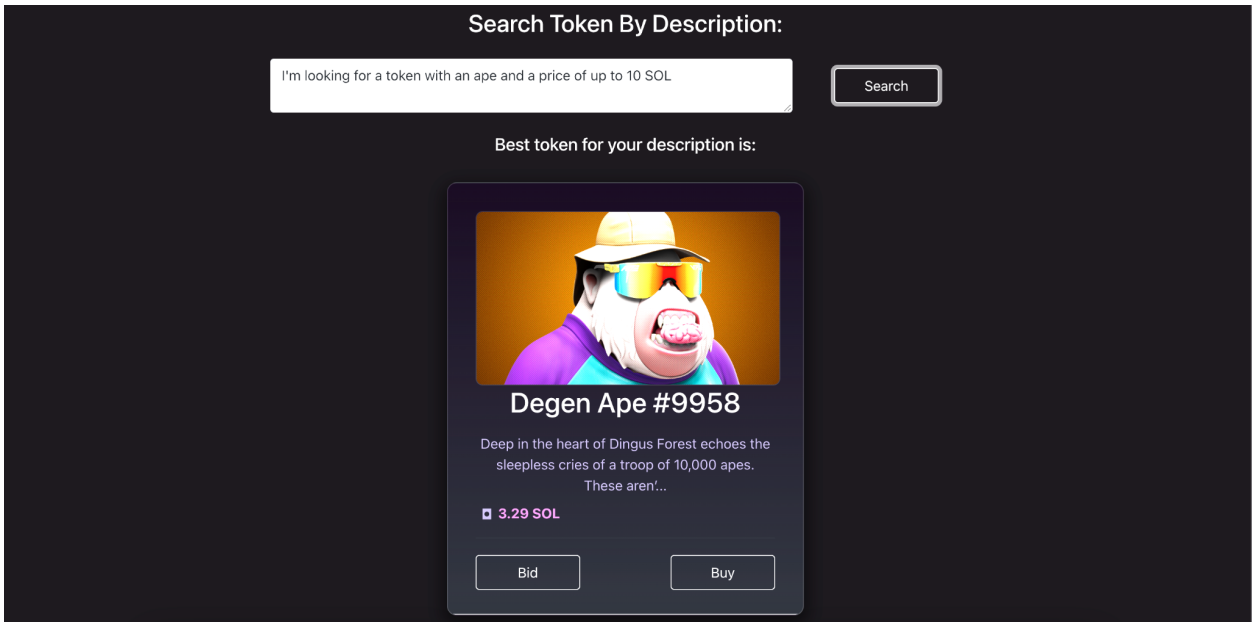

Рис 4.10. Результат “розумного” пошуку

Як бачимо, результат є вірним, оскільки користувач дійсно отримав токен, на якому зображена мавпа і його ціна є меншою 10 SOL.

Висновки

У цій курсовій роботі досліджувалось створення децентралізованого ринку на основі мережі Solana. Було обговорено архітектуру та механізм консенсусу мережі Solana, наголошуючи на її масштабованості, високій пропускній здатності та низькій комісії за транзакції. Мережа Solana забезпечує надійну інфраструктуру, яка може обробляти великий обсяг транзакцій, забезпечуючи швидку та економічно ефективну торгівлю.

Ми також підкреслили вибір інструментів розробки, розгортання смарт-контрактів та інтеграцію гаманців Solana для взаємодії з користувачами. Завдяки використанню інструментів розробки та інтеграції гаманців Solana ринок забезпечує зручний досвід, спрощуючи керування активами та виконання транзакцій.Оцінка ринку на базі Solana була зосереджена на ключових показниках, таких як пропускна здатність транзакцій, затримка, прийняття користувачами, безпека та ліквідність.

На завершення ця курсова робота підкреслила важливість мережі Solana для побудови децентралізованих ринків. Результати дослідження показали, що Solana пропонує надійну інфраструктуру, що забезпечує безпечну та ефективну однорангову торгівлю. Токенізація активів та інтеграція інструментів розробки ще більше покращують функціональність ринку.

Рухаючись вперед, майбутні дослідження на ринках на базі Solana можуть вивчати такі сфери, як оптимізація смарт-контрактів, методи підвищення конфіденційності, моделі управління, взаємодія з іншими

блокчейнами та інтеграція децентралізованих фінансових протоколів. Також можна підсумувати, що мережа Solana досить добре підходить для створення та роботи з NFT, оскільки є можливість додавання великої кількості інформації в токени та відносно низька комісія за транзакції.

Також за допомогою мов програмування JavaScript, Python і сервісів Shyft і ChatGPT було реалізовано сайт, за допомогою якого можна створювати, пересилати та продавати NFT токени. Також було реалізовано “розумний пошук” за допомогою штучного інтелекту. В подальшому цей продукт можна модернізувати, додавши можливість створення токенів на інших мережах та дослідити можливості інтеграції та взаємодії між ними.

Використана література

1. Solana in a nutshell: the definitive guide to enter the world of decentralized finance, lending, yield farming, dApps and master it completely., Sebastian Andres, 2021
2. The NFT Handbook: How to Create, Sell and Buy Non-Fungible Tokens., Matt Fortnow, Quharrison Terry, 2021

Інтернет - ресурси

1. https://gpt-index.readthedocs.io/en/latest/
2. https://docs.solana.com/developing/clients/javascript-api
3. https://medium.com/geekculture/solana-nft-marketplace-development-1e7d3c98c244
4. https://medium.com/javarevisited/solana-nft-marketplace-development-an-entrepreneur-guide-to-know-93f4dd17b8fd
5. https://medium.com/coinmonks/navigating-the-solana-nft-ecosystem-a-comprehensive-guide-for-traders-and-collectors-fd0c6b2aa457