

WhiteBIT

Технічне завдання: Реалізація Web3 авторизації

Мета

Реалізувати базову авторизацію користувача через Web3-гаманець (наприклад, MetaMask). Це має бути альтернативою традиційній авторизації (логін + пароль).

Компоненти системи

1. Frontend

- Проста сторінка з кнопкою "Увійти через Web3".
- При натисканні на кнопку:
- Запускається перевірка наявності Web3-гаманця в браузері (наприклад, MetaMask).
- Користувач підключає гаманець

Додатково

- Приклад для орієнтації — Web3 авторизація на WhiteBIT.
- Візуальний дизайн — мінімальний, без верстки.
- Документація роботи обов'язкова.
- Мова програмування: будь-яка;

Whitechain

Введення

Дане тестове завдання було підготовлено компанією **WhiteBIT** для студентів **університету НаУКМА**. Це завдання дає змогу компанії оцінити **аналітичні, технічні та архітектурні навички кандидатів**.

Після успішного виконання тестового завдання компанія розглядає його протягом **N робочих днів**, і **найкращим кандидатам може бути запропоновано пройти технічну співбесіду** на позицію **Junior / Trainee Blockchain Engineer** у компанії **WhiteBIT**.

Вимоги до коду

Код має бути виконаний на версії Solidity = 0.8.24, також він має бути задеплойований і верифікований у мережу Ethereum Sepolia, або/і отримати +100 до карми і задеплойовати та верифікувати у мережі Whitechain Testnet.

Має бути 100% покриття тестами свого контракту, і задеплойовано за допомогою Hardhat, скрипти мають бути написані на TypeScript. до коду мають відповідати формату natSpec. Має бути доданий README файл. Робоча версія

контракту має бути задеплоєна і посилання на нього надане, на ньому мають бути виконані голосування, а в разі гри — крафт речей, і всі інструкції для того, щоб задеплоїти проєкт. Код має бути завантажений у репозиторій і посилання передане відповідальній особі. Використання таких бібліотек є необов'язковим, та за бажанням студента:

UUPSUpgradeable
Initializable
AccessControl
Та й інші

Виконати завдання можна частково, якщо на весь функціонал не вистачає часу. Не описані сценарії завдання студент вирішує самостійно на власний розсуд, та необхідно їх вказати в README файлі, а також яке рішення було прийнято людиною. ChatGPT не забороняється.



Завдання 1: Голосування з різними типами

Кожен студент повинен обрати один тип голосування на свій розсуд та реалізувати його у вигляді смарт-контракту.

Нижче подано варіанти голосувань, серед яких можна вибрати один.

ВАРІАНТИ ГОЛОСУВАННЯ:

1. Голосування через NFT (1 голос = 1 аккаунт)

- Гравець може придбати NFT за внутрішню валюту (коїни) в контракті Marketplace.
 - Незалежно від кількості NFT на акаунті, гравець може проголосувати лише один раз у кожному голосуванні.
 - NFT не впливає на силу голосу.
 - Проголосувати можна тільки якщо у гравця є хоча б одне NFT.
 - Перемагає те рішення за яке проголосувало більше користувачів.
 - Голосування обмежене в часі.
-

2. Голосування через ERC20 (з перевіркою балансу, 1 голос = 1 аккаунт)

- Гравець може придбати ERC20 токени за коїни в контракті Marketplace.
 - Якщо на рахунку гравця більше X токенів, він має право проголосувати один раз.
 - Мінімальний баланс для участі: X токенів.
 - У кожному голосуванні гравець може голосувати лише один раз.
 - Перемагає те рішення за яке проголосувало більше користувачів.
 - Голосування обмежене в часі.
-

3. Голосування з винагородою у вигляді NFT (1 голос = 1 аккаунт)

- Кожен аккаунт має право проголосувати один раз у межах одного голосування.
 - Після голосування користувач отримує NFT як винагороду.
 - Отримане NFT можна продати в контракті Marketplace або перевести на інший аккаунт.
 - Перемагає те рішення за яке проголосувало більше користувачів.
 - Голосування обмежене в часі.
-

4. Голосування з винагородою у вигляді ERC20 (1 голос = 1 аккаунт)

- Кожен аккаунт може проголосувати один раз у кожному голосуванні.
 - Після голосування аккаунт отримує певну кількість ERC20 токенів.
 - Токени можна продати в контракті Marketplace або перевести на інший аккаунт.
 - Перемагає те рішення за яке проголосувало більше користувачів.
 - Голосування обмежене в часі.
-

5. Голосування з ваговим значенням (сила голосу = кількість NFT)

- Кожен аккаунт має право проголосувати один раз у межах одного голосування.
- Під час голосування враховується кількість NFT на рахунку гравця.
- Чим більше NFT — тим сильніше голос гравця (наприклад: 1 NFT = 1 голос).
- Гравець голосує лише один раз, але вага голосу = кількість NFT.
- Перемагає те рішення за яке проголосувало користувачі з більшою кількістю NFT.
- Голосування обмежене в часі.

Загальні умови для створення реєстру голосувань

- Усі студенти спільними зусиллями створюють контракт-реєстр голосувань (VotingRegistry).
- Усі суперечки вирішуються за допомогою “чу-ва-чі”.
- Усі учасники, які працювали над контрактом, мають бути зазначені з тегом @authors у хедері цього смарт-контракту.
- Контракт має бути задеплойований як мінімум у дві тестові мережі:
 - Ethereum Sepolia
 - Whitechain Testnet

- Адреси задеплогоного контракту потрібно надіслати однокрупникам для виконання продовження завдання.

Підключення власних контрактів голосування

- Після створення власного контракту голосування (VotingContract), студент зобов'язаний:
- Додати адресу свого контракту до **загального реєстру** (VotingRegistry).
- **Тільки після цього** його контракт отримає дозвіл на:
 - Створення нових голосувань.
 - Проведення процесу голосування.
- Якщо адреса контракту голосування **не зареєстрована в реєстрі**:
 - Створення голосувань заборонено.
 - Голосування у такому контракті неможливе.

Механіка VotingRegistry:

- Є можливість додати в реєстр контракт свого голосування
- Є можливість додати в реєстр новостворене голосування
- Є можливість видалити тільки свій контракт голосування
- Є можливість видалити голосування свого контракту
- Ведеться облік **усіх активних/завершених голосувань**
- Ведеться облік усіх **контрактів голосування, підключених до системи**
- Є можливість отримати список усіх голосувань певного контракту

Механіка VotingContract:

- Реалізована механіка голосування відповідно до вибраного типу голосування
- Можливість створення нового голосування (та автоматичне додавання його до реєстру VotingRegistry) якщо попереднє голосування завершилось.

Механіка NFTContract (ERC721):

- Є можливість створити NFT шляхом купівлі, або спалити його шляхом продажу його в Marketplace відповідно до вибраного типу голосування

Механіка ERC20VotingToken:

- Є можливість створити ERC20 шляхом купівлі, або спалити шляхом продажу його в Marketplace відповідно до вибраного типу голосування

Механіка Marketplace:

- Є можливість купити/продати NFT/ERC20 шляхом запиту метода на Marketplace відповідно до вибраного типу голосування

- NFT/ERC20 спалюється/створюється під час продажу/купівлі відповідно до вибраного типу голосування

Контракти:

- VotingRegistry (Загальний для всіх студентів)
- VotingContract (Контракт голосування від кожного студента)
- NFTContract (ERC721) (За необхідності від вибраного типу голосування)
- ERC20VotingToken (За необхідності від вибраного типу голосування)
- Marketplace (з функціоналом в залежності від вибраного типу голосування)

Завдання 2: Гра “Козацький бізнес”.

У грі існує 6 базових ресурсів NFT1155:

- Дерево (Wood)
- Залізо (Iron)
- Золото (Gold)
- Шкіра (Leather)
- Камінь (Stone)
- Алмаз (Diamond)

Гравці можуть об'єднувати ресурси та створювати унікальні предмети NFT721:

1. Шабля козака
 - 3× Залізо
 - 1× Дерево

- 1× Шкіра
- 2. Посох старійшини
 - 2× Дерево
 - 1× Золото
 - 1× Алмаз
- 3. Броня характерника (не обов'язково)
 - 4× Шкіра
 - 2× Залізо
 - 1× Золото
- 4. Бойовий браслет (не обов'язково)
 - 4× Залізо
 - 2× Золото
 - 2× Алмаз

Механіка NF–T1155 / NFT721:

- Створення NFT можливе лише через контракти Crafting або Search.
- Пряме створення або спалення NFT через базові контракти ResourceNFT1155 та ItemNFT721 — заборонене.
- Спалення NFT можливе тільки під час продажу предметів у контракті Marketplace.

Механіка MagicToken (ERC20):

- Токени MagicToken можна отримати лише через продаж предметів у контракті Marketplace.
- Пряме мінтинг токенів через контракт MagicToken заборонений. Мінт викликається виключно з Marketplace.
- Отримані MagicToken надходять на гаманець гравця після успішного продажу предмета.

Механіка Crafting / Search:

- Гравець може запускати пошук ресурсів раз на 60 секунд.
- Пошук генерує 3 випадкових ресурси (ResourceNFT1155), які надходять на гаманець гравця.
- Для створення предмета (ItemNFT721) через крафт, гравець повинен мати необхідну кількість ресурсів.
- Під час крафту:
 - Ресурси спалюються.
 - Створюється предмет (NFT721) з унікальним ID.
- Створені предмети можна:

- продавати на Marketplace,
- або передавати іншим гравцям.

Механіка Marketplace:

- Гравці можуть продавати предмети (NFT721) за MagicToken.
- Після купівлі предмета:
 - NFT спалюється.
 - Продавець отримує відповідну кількість MagicToken на свій гаманець.

Контракти:

- ResourceNFT1155
- ItemNFT721 (2-4шт)
- Crafting/Search
- Marketplace
- MagicToken (ERC20)