НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання комп’ютерного практикума

Дослідження безпечної реалізації та експлуатації децентралізованих додатків

Виконали студенти
групи ФЕ-31мп
Мятка І.І.
Кирилюк Д.В.
Столярчук Т.В.

Перевірила:
Байденко П. В.

Київ — 2024

Мета роботи: отримання навичок роботи із децентралізованими додатками та оцінка безпеки інформації при їх функціонуванні.

Для другого типу лабораторних робіт:

Розробка децентралізованого додатку на обраній системі децентралізованих додатків.

Теоретичні відомості:

Децентралізовані додатки (`Decentralized Applications` (`DApps`)) - це будь-які комп'ютерні програми, робота яких підтримується розподіленою мережею комп'ютерних вузлів, на відміну від підтримки одним сервером.

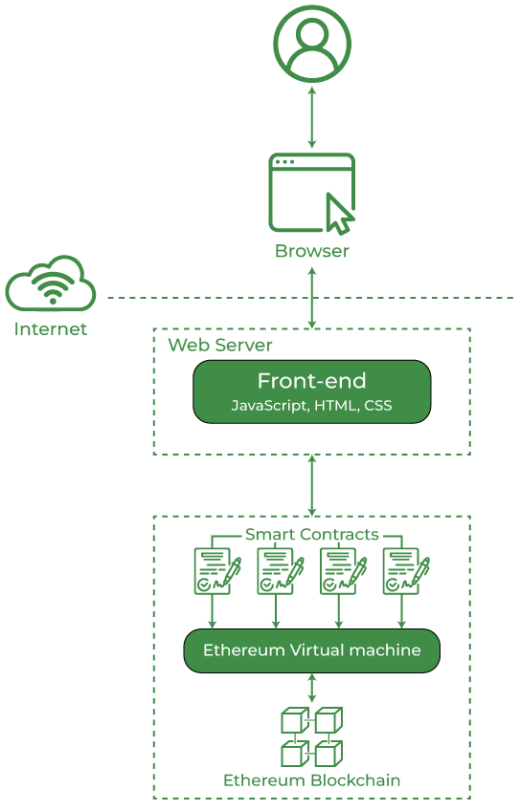
Концепція децентралізованих додатків була реалізована завдяки блокчейн платформам, які підтримують смарт-контракти, першим з яких був `Ethereum` (`ETH`). Крім того, що Ethereum є криптовалютою, він підтримує так звану `Ethereum Virtual Machine` (`EVM`), яку можна описати як розподілений комп'ютер, стан якого у будь-який момент точно визначається за допомогою алгоритму консенсусу.

Завдяки тому, як вони виконуються, `DApps` можуть забезпечувати таку ж якість обслуговування, як і звичайні додатки, водночас користуючись усіма перевагами децентралізації, такими як майже постійний час безвідмовної роботи та стійкість до цензури та корупції.

Є багато прикладів успішних `DApps` з мільйонами доларів ринкової капіталізації та сотнями активних користувачів, таких як ринкова платформа прогнозів `Augur (REP)`, ринок для простої комп'ютерної потужності `Golem (GNT)` та `Basic Attention Token (BAT)` - цифрова рекламна платформа на основі блокчейну.

Будь-який dapp має відповідати наступним критеріям :

- Відкритий вихідний код – програмний код програми доступний для всіх
- Децентралізація – застосування криптографічної технології, порівнянної з блокчейном
- Мотивація – додаток використовують крипто-токени / цифрові активи для стимуляції користувачів
- Алгоритм – створює токени і має вбудований механізм консенсусу



Для ефективної роботи dapps необхідний зв'язок між front-end та смарт-контрактами на Ethereum. Однак мережа Ethereum складається з безлічі вузлів , розподілених за децентралізованою системою. Кожен вузол зберігає копію всіх смарт-контрактів та пов'язаних із ними даних, а також стан віртуальної машини, в якій працює кожен смарт-контракт.

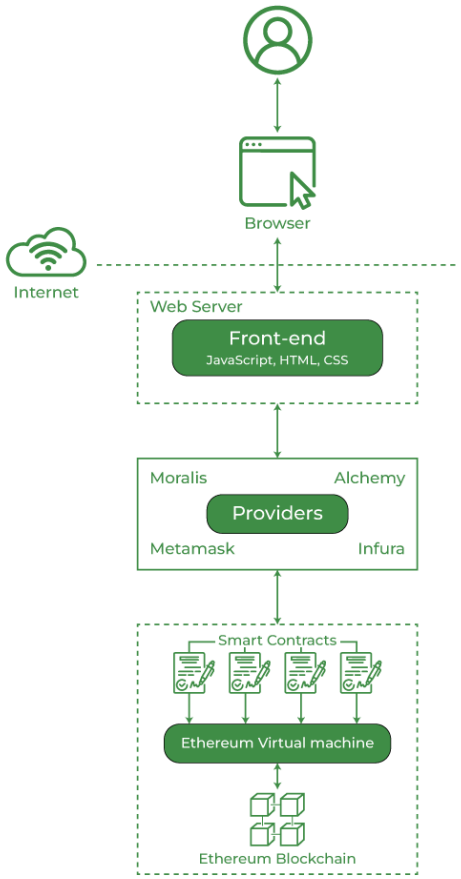
Щоб взаємодіяти з блокчейном, необхідно підключитись до вузла . Вузол відправляє запит на транзакцію, щоб продати її на віртуальній машині Ethereum (EVM). Майнери реєструють транзакцію та поширюють отриману зміну стану серед інших вузлів мережі. Передача транзакції може відбуватися двома способами :

- Самостійно шляхом налаштування програмного забезпечення, яке працює на вузлі блокчейна Ethereum;
- Сторонніми сервісами.

Сторонні служби можуть запропонувати допомогу у створенні, налаштуванні та забезпеченні вузлів.

Сторонні послуги можуть використовуватися для забезпечення різних функцій блокчейну без необхідності запуску повноцінного вузла. Встановлення нового вузла Ethereum на сервері може зайняти багато часу та грошей , а також у міру масштабування DApp кількість вузлів постійно збільшуватиметься .

Вузли , до яких підключається користувач для роботи в блокчейні, часто називають ” провайдерами “.

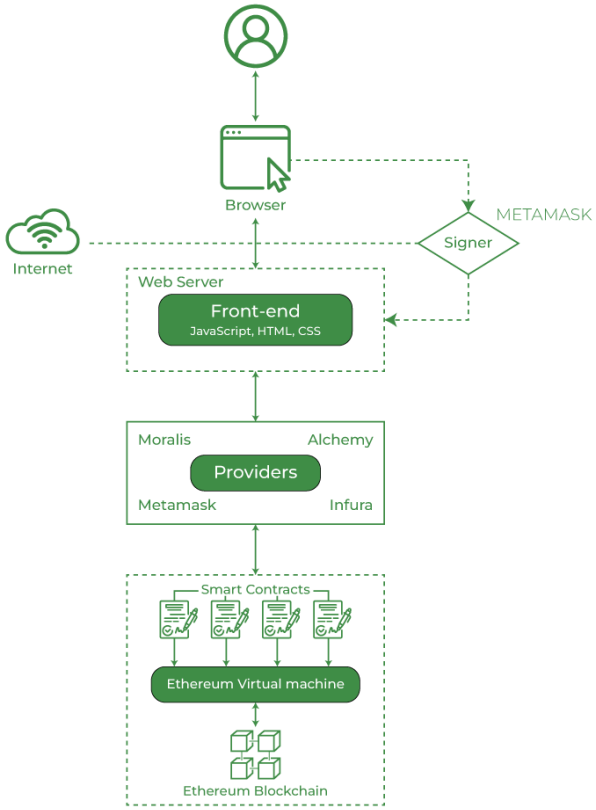


Провайдер `Ethereum` здійснює стандарт `JSON-RPC`. Це створює загальний набір інструментів, що дозволяє `front-end` взаємодіяти з блокчейном.

Після того, як ви підключилися до блокчейну за допомогою провайдера, ви можете обробити стан, що знаходиться в блокчейні.

Але якщо вам потрібно внести запис до протоколу стану перед тим як надіслати транзакцію, вам потрібно виконати ще одну умову – ” підписати ” транзакцію з використанням закритого ключа .

Щоразу, коли `front-end` **вимагає від користувача підписання , він зв’язується з** **`Metamask`** . Тут `Metamask` виконує функцію особи, що підписує.



Хід виконання:

Тестову версію сайту розміщено за посиланням

https://lab-3-blockchain.vercel.app

В якості розробки децентралізованого додатку було обрано створення веб-додатка, який підписує набране повідомлення особистим підписом гаманця. Обрана платформа – `Ethereum`, мова програмування – `JS` з використанням `React` та `Ethers.js` для використання в браузерному гаманці `MetaMask`.

1) **Використання Ethers**

Бібліотека `Ethers.js` має на меті бути повною і компактною бібліотекою для взаємодії з блокчейном Ethereum та його екосистемою.

Вона часто використовується для створення децентралізованих додатків (`dapps`), гаманців (таких як `MetaMask` і `Tally`) та інших інструментів і простих скриптів, які вимагають читання і запису в блокчейн.

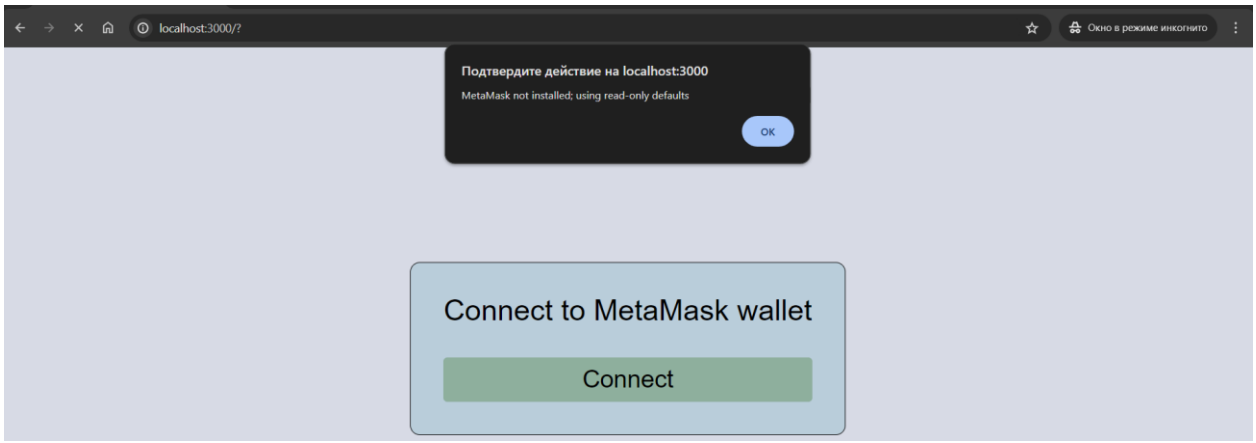
В контексті виконання роботи `Ethers` використовується для:

1) Визначення наявності у користувача гаманця `MetaMask` через наявність в об'єкті `windows` поля `ethereum`.
2) Отримання даних акаунту (адреса та баланс)
3) Підпис набраного повідомлення та одержання отриманої сигнатури

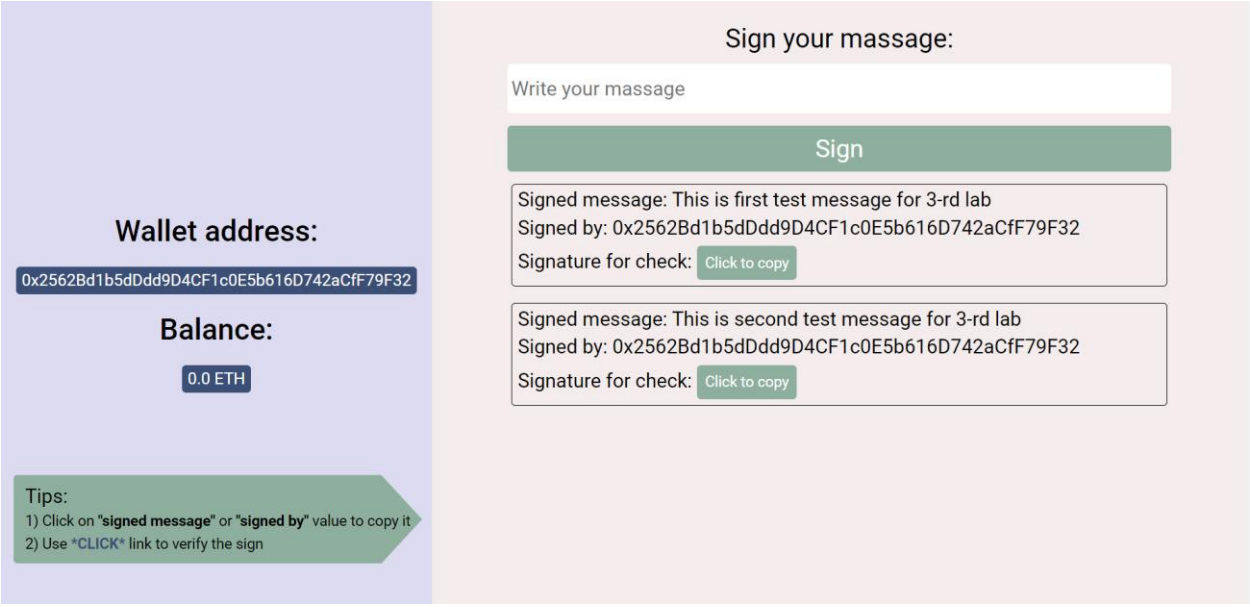
Використання бібліотеки не обмежується накладенням підпису, також можливе розгортання власних смарт-контрактів, або підключення та використання уже наявних. Окрім цього існують різні звичні функції, наприклад переказ коштів, одержання логів операцій, визначення розходу газу на операцію тощо.

2) **Тестування розробленого додатку**

Спробуємо під’єднатись до мережі без встановленого `MetaMask`:

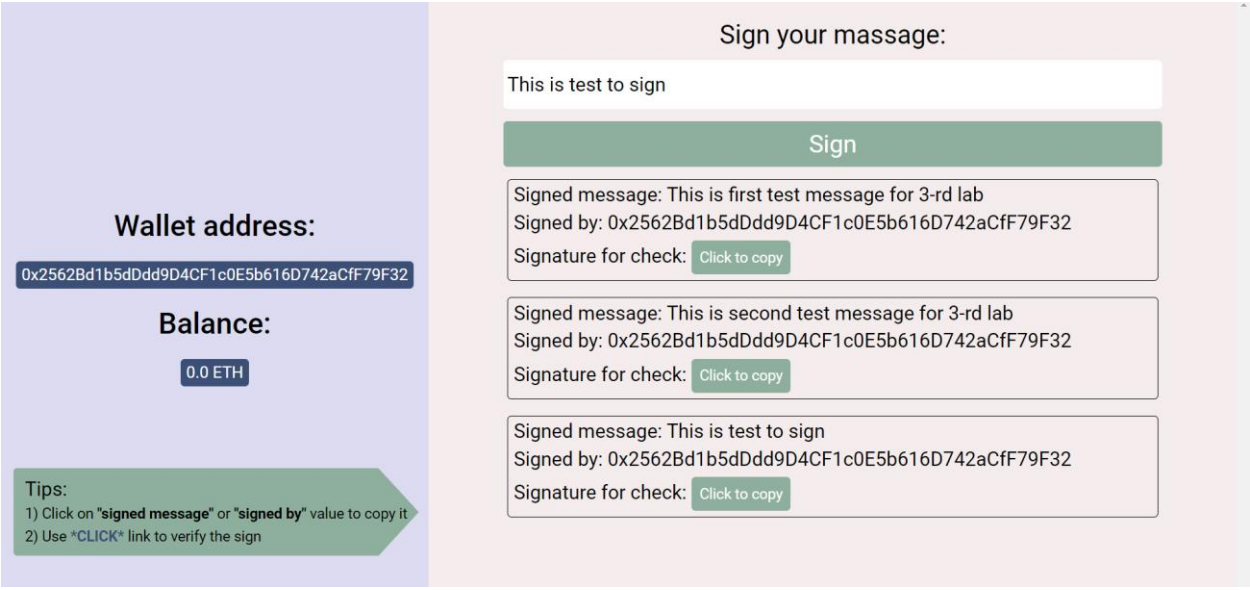


При використанні з `MetaMask` отримаємо, при першій валідації, `prompt` вікно з проханням введення паролю:

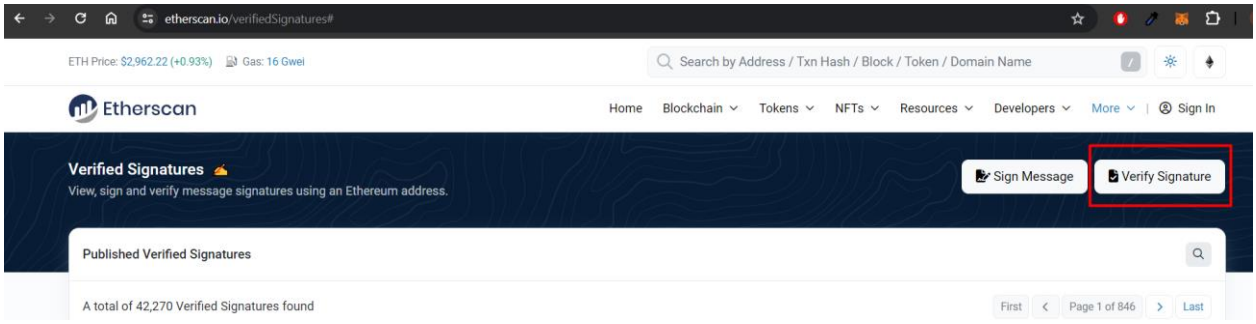


Початково наявні 2 підписаних повідомлення, адреса та баланс акаунту. Для підпису потрібно мати хоча б 1 символ в полі вводу.

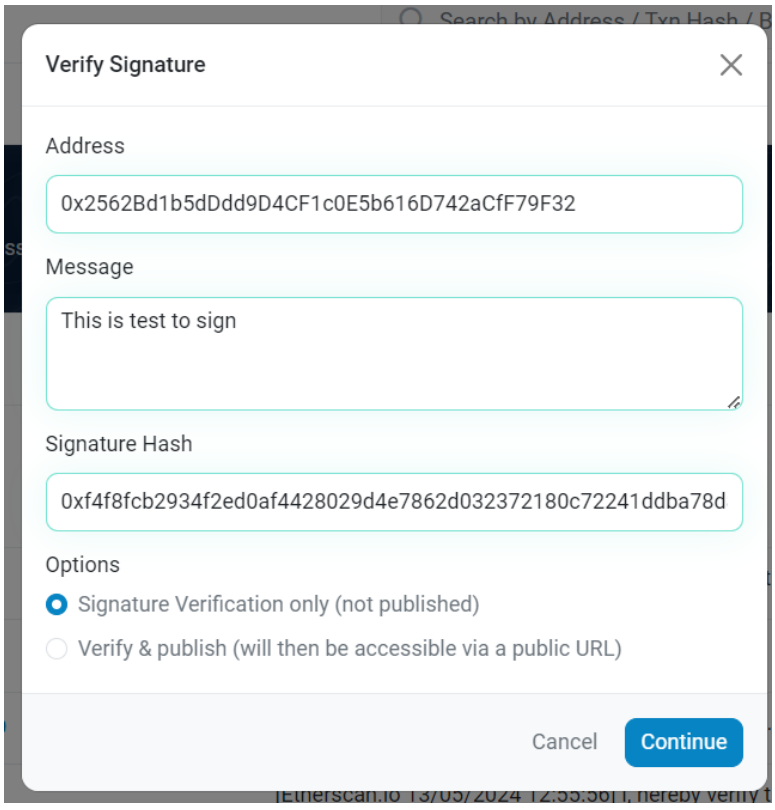
Отримали зміну стейту верифікованих записів:

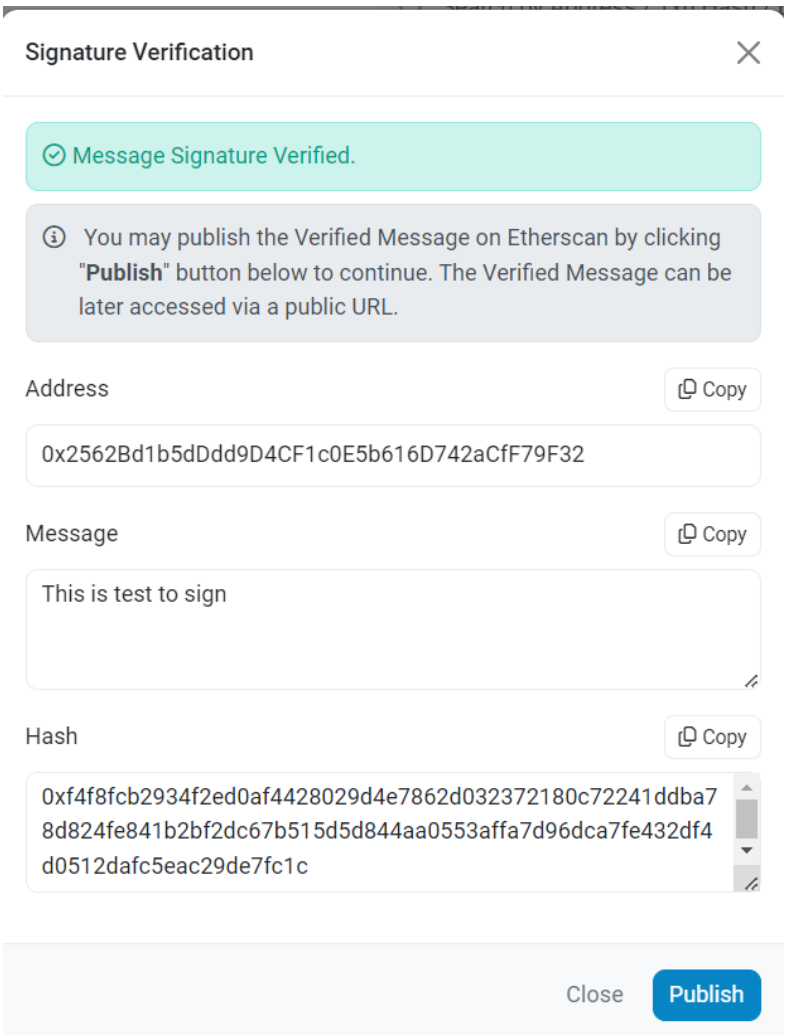


Перевірку можна здійнисти слідуючи вказівкам в `Tips`. За допомогою сайту https://etherscan.io/verifiedSignatures



У вікно введемо отримані дані:





Отримали позитивну відповідь, що говорить про підтвердження факту підпису гаманцем повідомлення. Отриманий сертифікат можливо розмістити публічно. Частою практикою є також підпис гаманцем не просто повідомлення, а його хеша, проте в цьому додатку все було реалізовано на string типах.

Висновок: у роботі реалізований один з найпростіших варіантів децентралізованого додатку, який може бути використаний, наприклад, для блогу, або форуму, де за кожним користувачем закріплено його адресу гаманця, а факт підпису можна підтвердити за допомогою блокчейну. Для подальшого розширення функціоналу можна впровадити систему автентифікації та авторизації, а також синхронізувати функціонал між всіма користувачами за допомгою використання БД, також опціонально впровадити можливість зміни адреси гаманця, або навіть самого гаманця. В якості покращення UI та UX потрібно зробити responsive дизайн під усі пристрої та зробити кращим і сам дизайн під десктоп.