УДК 004.932.4

ІНТЕГРАЦІЯ КРИПТОГАМАНЦЯ METAMASK ДЛЯ ПРОВЕДЕННЯ WEB3 ТРАНЗАКЦІЙ

Залізко В. Я.
e-mail: vladyslav.zalizko@nure.ua
Науковий керівник – д.т.н., проф. Машталір С.В.
Харківський національний університет радіоелектроніки, каф. ІНФ
м. Харків, Україна

This project focuses on integrating a cryptocurrency wallet for Web3 transactions in decentralized applications (dApps) [1]. Using JavaScript and React, we provide a secure and convenient way for users to connect to their crypto wallets, such as MetaMask. This allows them to make transactions, check their balances, and interact with smart contracts on the Ethereum blockchain [2]. This system not only improves security and user experience, but also opens up integration opportunities with DeFi platforms, NFT marketplaces, and other Web3 services.

Оскільки блокчейн-технології та Web3 змінюють спосіб взаємодії користувачів із цифровими активами, забезпечуючи децентралізацію та прозорість. На відміну від традиційних фінансових систем, Web3 дозволяє користувачам самостійно керувати своїми активами без посередників.

Ключову роль у Web3-екосистемі відіграють криптогаманці, які надають можливість:

- Зберігати криптовалюту (Ethereum, USDT, BNB тощо);
- Виконувати фінансові транзакції;
- Взаємодіяти зі смарт-контрактами;
- Отримувати доступ до децентралізованих застосунків.

MetaMask — один із найпопулярніших браузерних криптогаманців, який використовується для взаємодії з Ethereum та сумісними мережами (Polygon, Binance Smart Chain, Arbitrum тощо). Його інтеграція в вебзастосунок забезпечує користувачам безпечний і швидкий доступ до всіх можливостей Web3.

Щоб інтегрувати криптогаманець у React-додаток, можна використовувати бібліотеку ethers.js або web3.js. У цьому прикладі ми використовуємо ethers.js для роботи з Ethereum (лістинг 1.1).

Лістинг 1.1 Підключення гаманця

```
import { useState } from "react";
import { ethers } from "ethers";
import { toast } from "react-toastify";
const WalletConnect = () => {
  const [account, setAccount] = useState(null);
  const connectWallet = async () => {
    if (!window.ethereum) {
      toast.error("MetaMask не встановлено. Будь ласка, встановіть MetaMask.");
```

```
      return;
    }
    try {
      const provider =
  new ethers.providers.Web3Provider(window.ethereum);
      const accounts = await provider.send("eth_requestAccounts", []);
      setAccount(accounts[0]);
      toast.success(`Підключено: ${accounts[0]}`);
    } catch (error) {
      toast.error("Помилка підключення: " + error.message);
    }
  };
  return (
    <button onClick={connectWallet}>
      {account ? `Підключено: ${account.slice(0, 6)}...${account.slice(-4)}` : "Підключити гаманець"}
    </button>
  );
};
export default WalletConnect;
```

Цей код дозволяє користувачеві підключити MetaMask до вебзастосунку. Після успішного підключення на кнопці з’являється скорочений адрес гаманця.

Після підключення гаманця можна проводити криптовалютні операції. У цьому прикладі ми надішлемо 0.01 ETH на вказану адресу (лістинг 1.2).

Лістинг 1.2 Створення та підпис транзакції в мережі Ethereum

```
const sendTransaction = async () => {
   if (!window.ethereum) {
    toast.error("MetaMask не встановлено.");
    return;
    }
    try {
     const provider =
  new ethers.providers.Web3Provider(window.ethereum);
      const signer = provider.getSigner();
      const tx = await signer.sendTransaction({
       to: "0x1234567890abcdef1234567890abcdef12345678",
       value: ethers.utils.parseEther("0.01")
      });
      toast.success(`Транзакція надіслана: ${tx.hash}`);
    } catch (error) {
      toast.error("Помилка транзакції: " + error.message);
    }
  };
```

Цей код дозволяє користувачам надсилати Ethereum на вказану адресу, використовуючи свій підключений гаманець.

Крім простих транзакцій, Web3-застосунки взаємодіють зі смарт-контрактами. У лістинг 1.3 продемонстровано приклад виклику функції mint() у NFT-контракті, що надсилає транзакцію в блокчейн [3].

Лістинг 1.3 Взаємодія зі смарт-контрактом

```
const mintNFT = async () => {
  if (!window.ethereum) {
    toast.error("MetaMask не встановлено.");
    return;
  }
  const contractAddress = "0xYourSmartContractAddress";
  const abi = [
    "function mint() public"
  ];
  try {
    const provider =
new ethers.providers.Web3Provider(window.ethereum);
    const signer = provider.getSigner();
    const contract = new ethers.Contract(contractAddress, abi, signer);
    const tx = await contract.mint();
    toast.success(`NFT успішно створено! TX: ${tx.hash}`);
  } catch (error) {
    toast.error("Помилка: " + error.message);
  }
 };
```

Інтеграція криптогаманця у Web3-застосунок дозволяє користувачам безпечно здійснювати фінансові операції, брати участь у DeFi-проєктах, купувати NFT та взаємодіяти зі смарт-контрактами. Використання бібліотеки ethers.js у поєднанні з React робить цей процес максимально простим і ефективним.

Такі технології вже сьогодні трансформують цифрову економіку, відкриваючи можливості для фінансової незалежності, децентралізації та підвищення безпеки онлайн-транзакцій. У майбутньому інтеграція криптогаманців стане стандартом для більшості вебзастосунків, що працюють у сфері Web3.

Список використаних джерел:

1. Antonopoulos, A. M. (2017). "Mastering Ethereum: Building Smart Contracts and DApps." O’Reilly Media.
2. Dannen, C. (2017). "Introducing Ethereum and Solidity." Apress.
3. Buterin, V. (2013). "Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform.