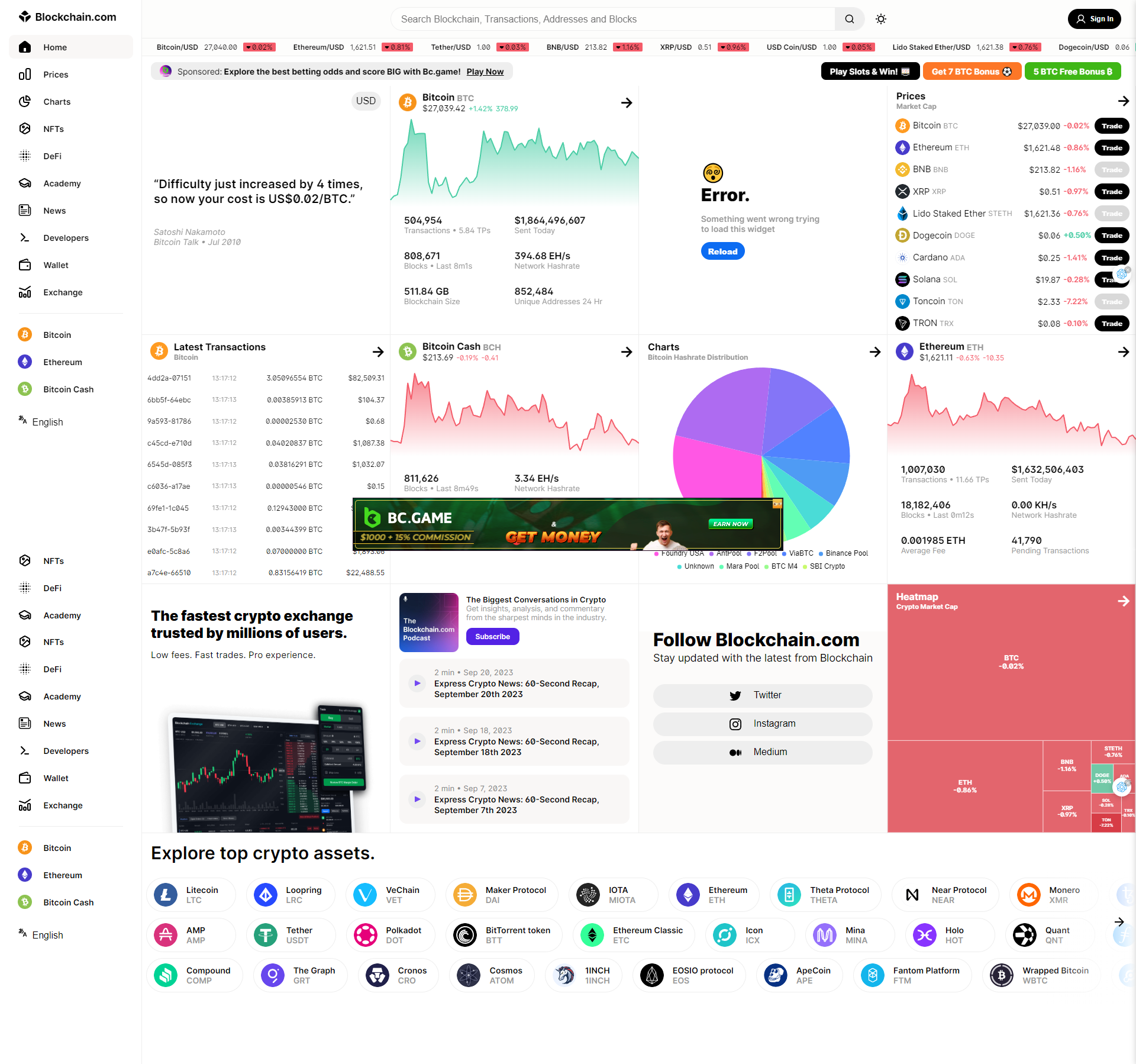
Họ và tên: **Lê Khắc Thanh Tùng**

MSSV: **52100943**

**Getting Your Hands on Blockchain**

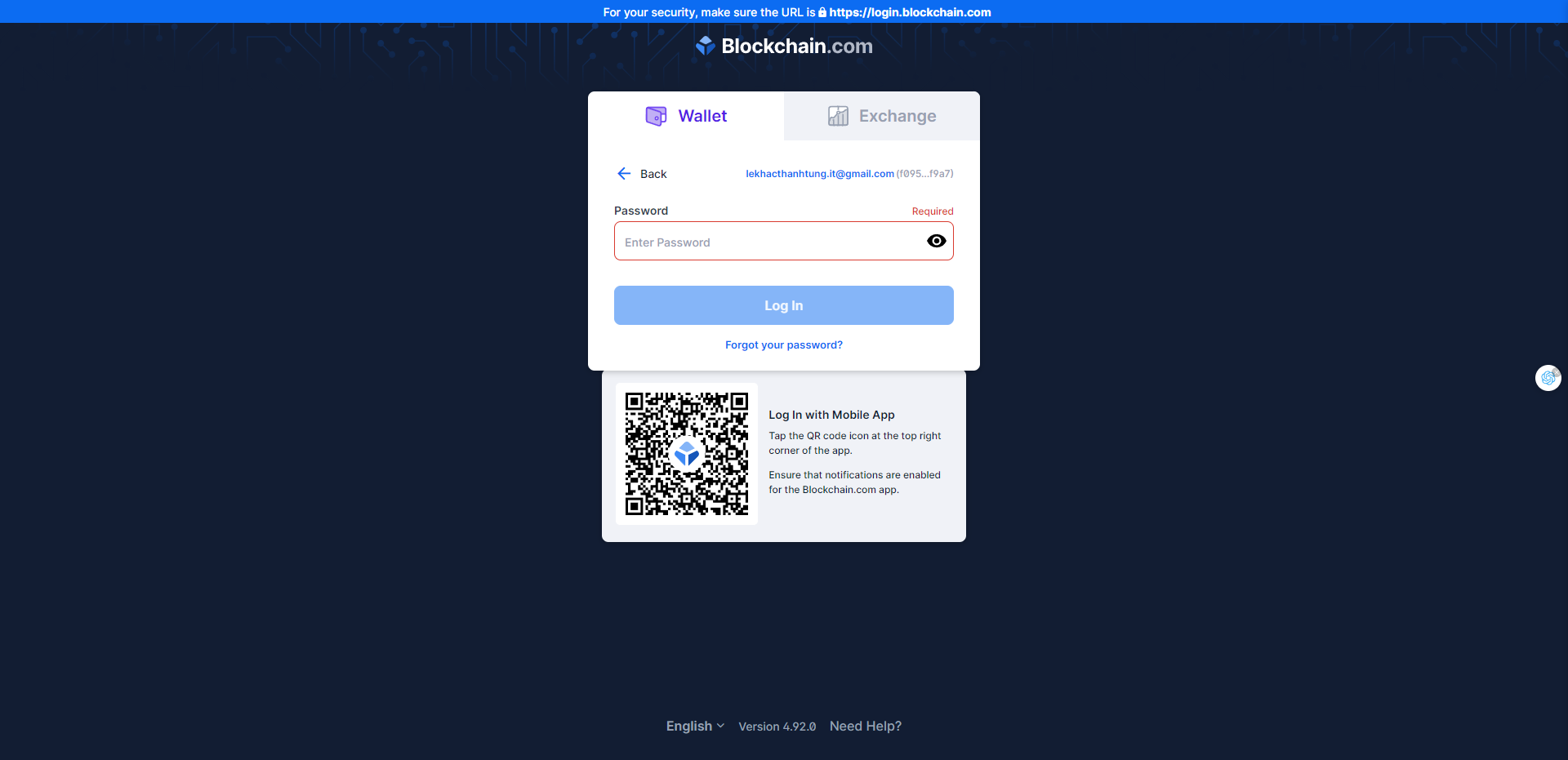
1. **Creating and using a Bitcoin wallet**

Bước 1: Truy cập vào đường dẫn https://www.blockchain.com/explorer

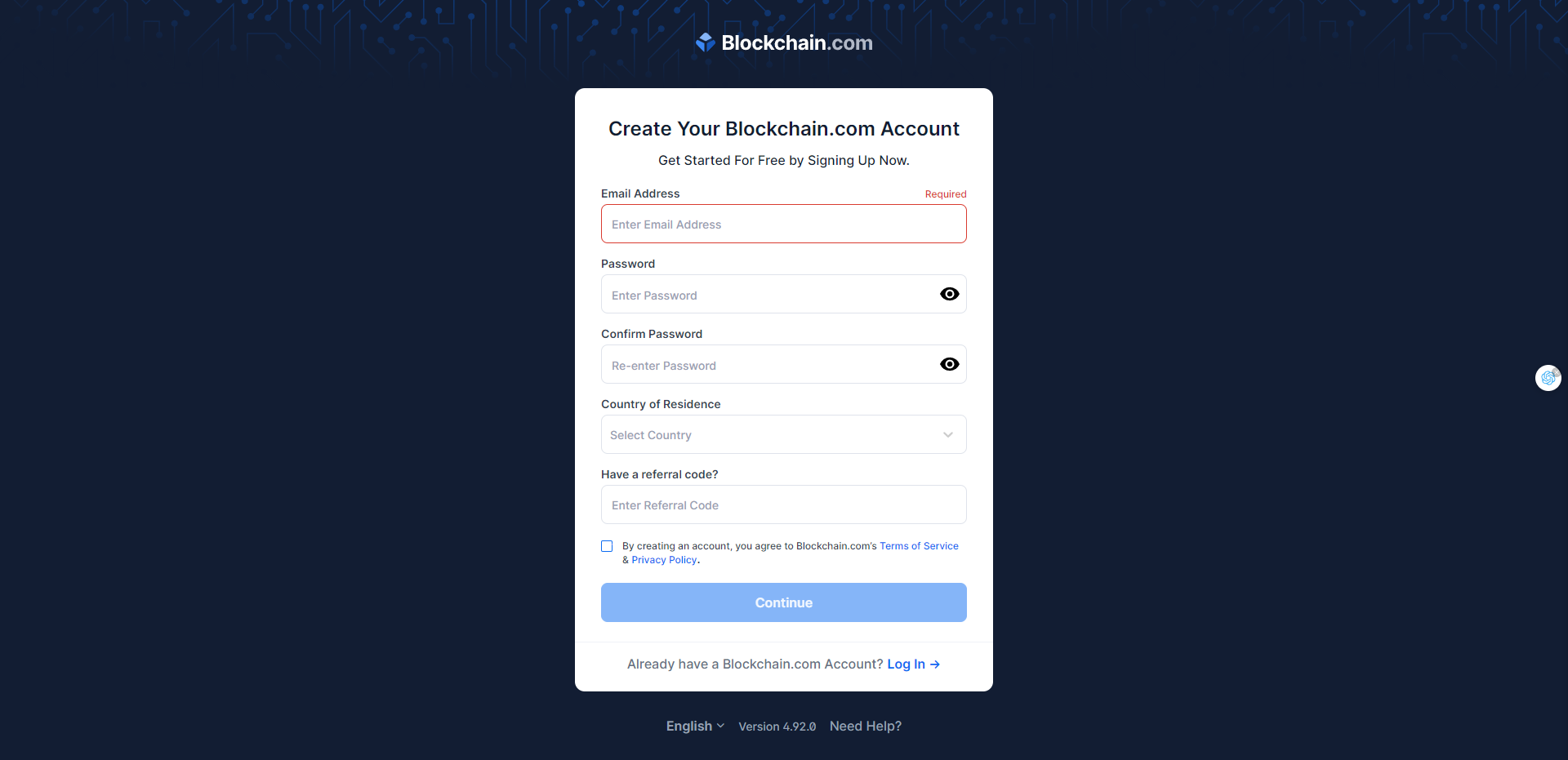


Bước 2: Nhấn vào nút Sign In

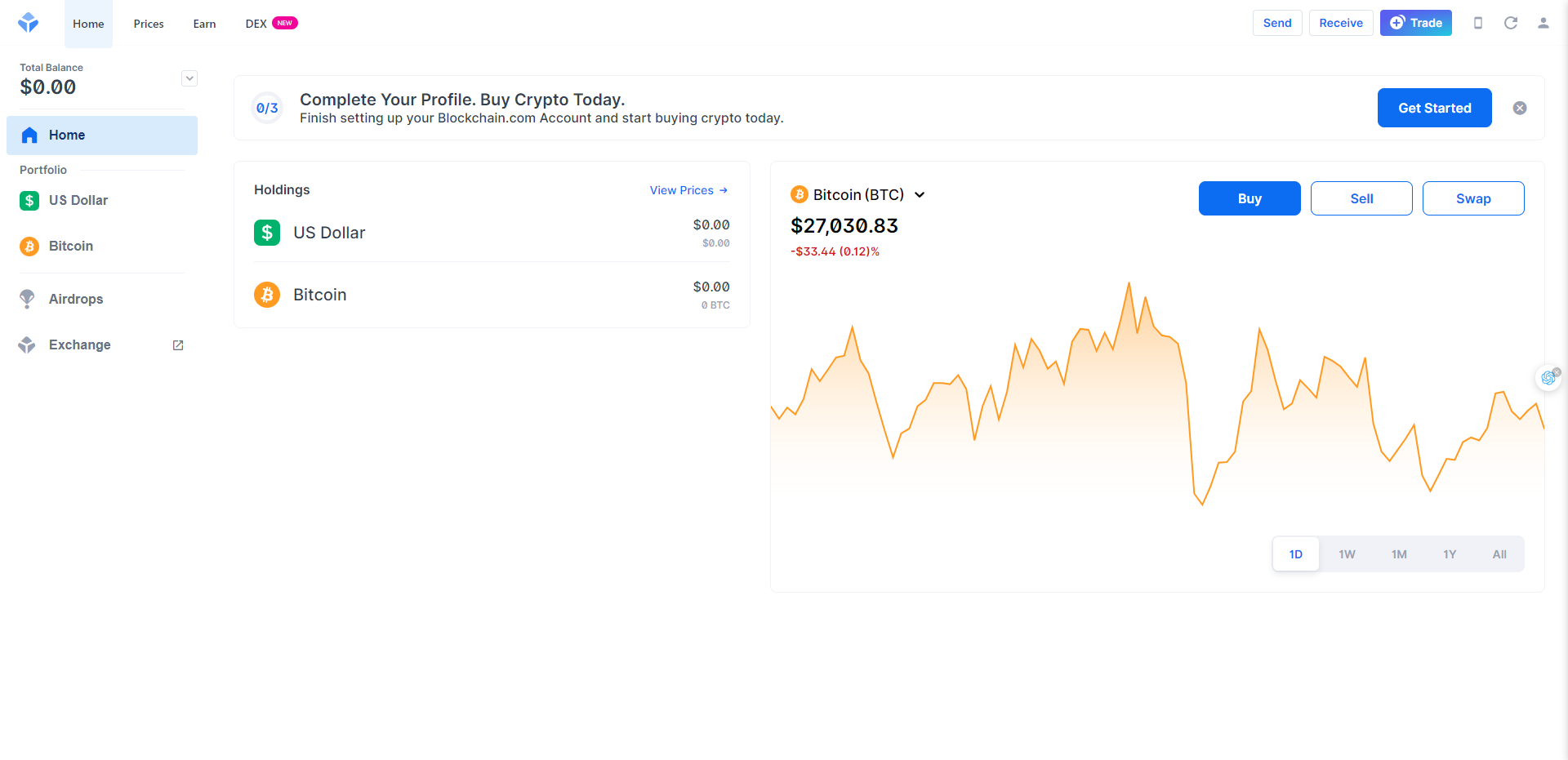
* Nếu đã có tài khoản rồi thì nhập mật khẩu rồi nhấn nút Log in để đăng nhập



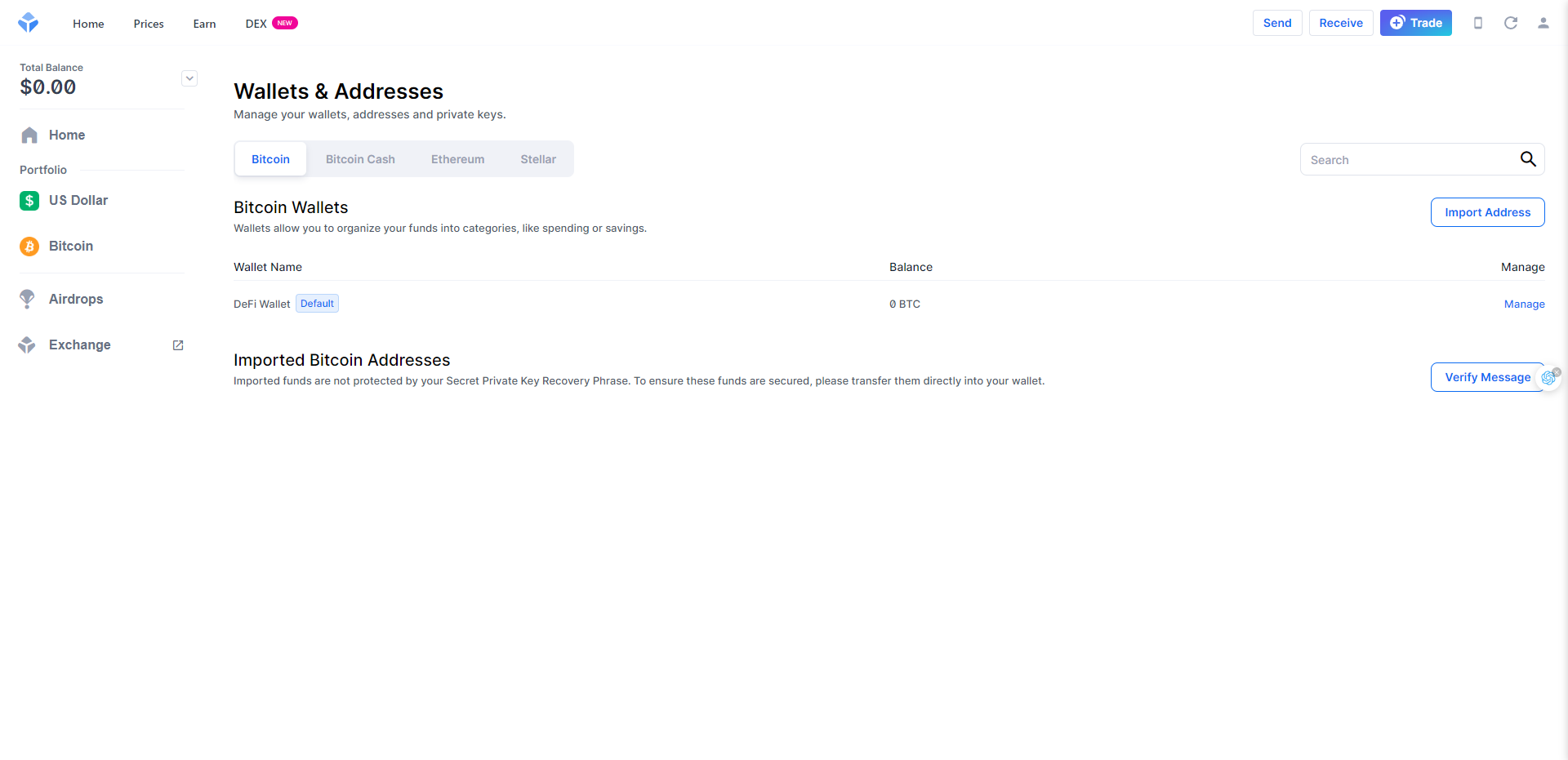
* Nếu chưa thì nhấn tạo mới wallet và nhập đầy đủ thông tin để tạo account



Sau khi đăng nhập thành công, hệ thống sẽ điều hướng đến trang <https://login.blockchain.com/en/#/home> và hệ thống sẽ yêu cầu ta xác thực thông tin để có thể mua, bán, trao đổi crypto

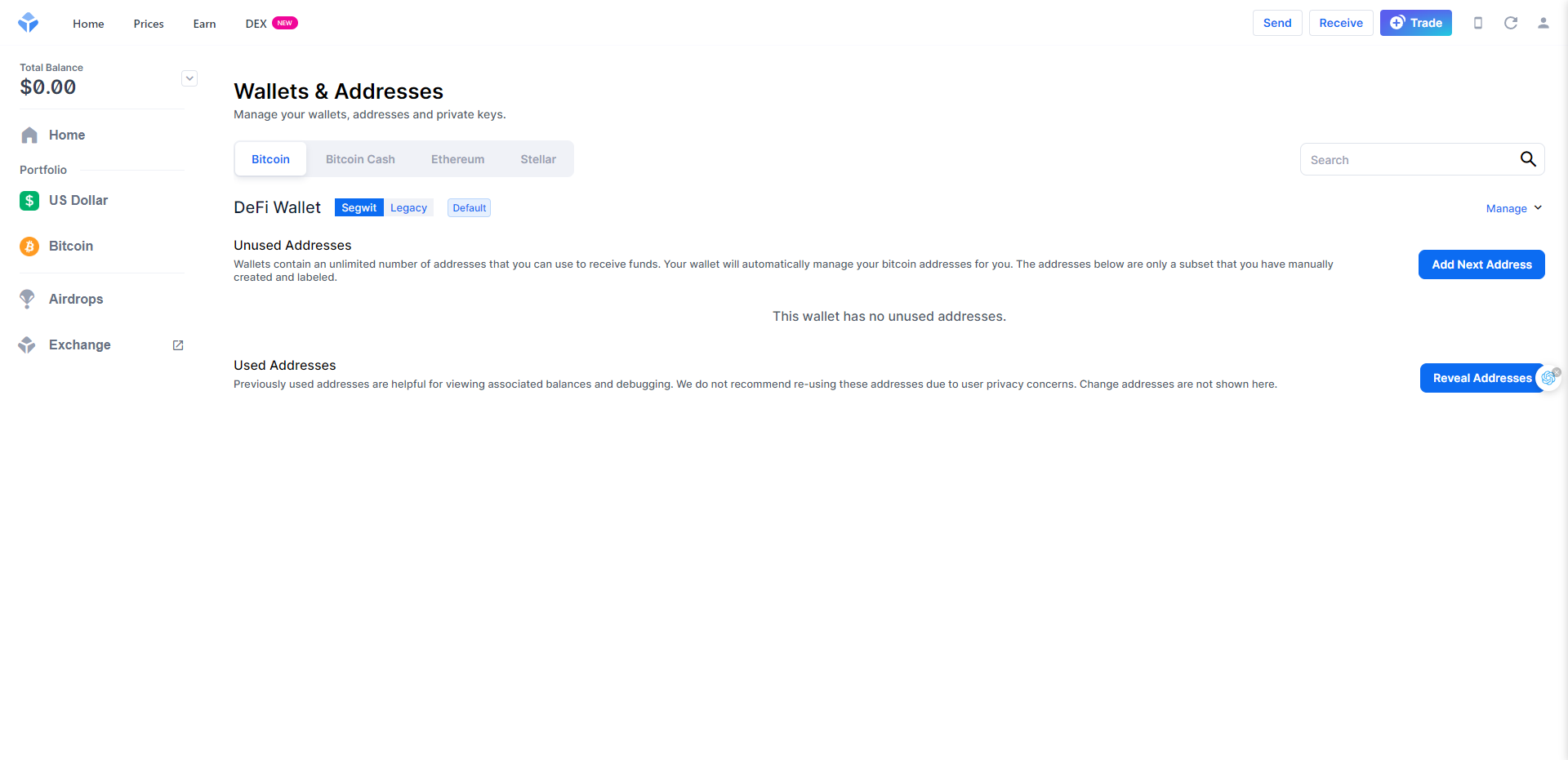


Truy cập vào trang <https://login.blockchain.com/en/#/settings/addresses/btc> để xem địa chỉ ví

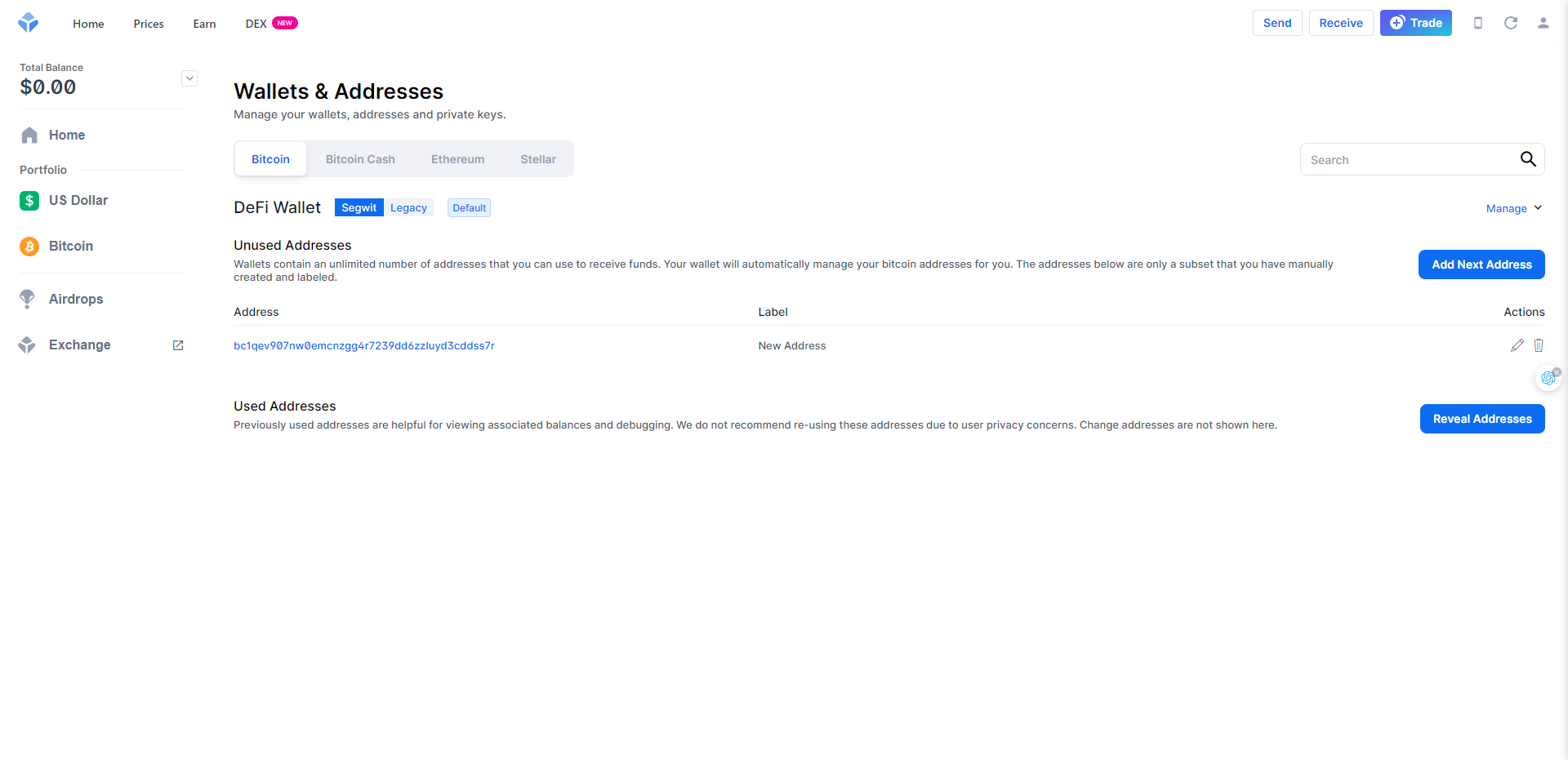


Như trong hình em đang có 1 Bitcoin Wallet là Defi Wallet

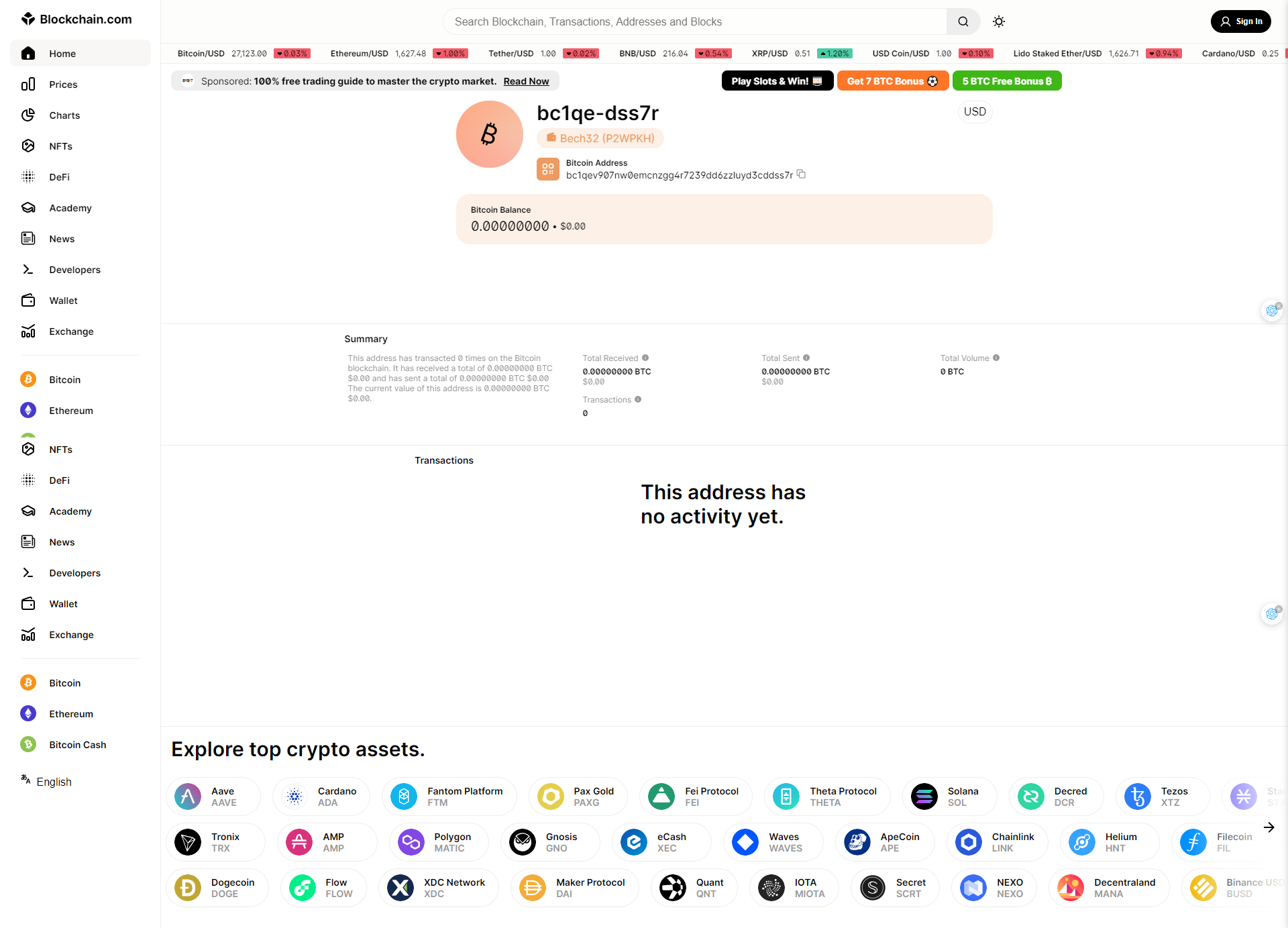
Nhấn vào manage để xem chi tiết ví



Ở đây ta có thể nhấn next address để generate ra address và ta có thể dùng address để nhận funds

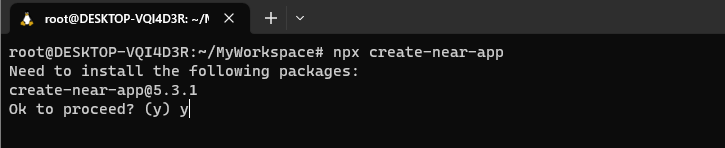


Và khi ta nhấn vào address đó nó sẽ điều hướng đến trang <https://www.blockchain.com/explorer/addresses/btc/bc1qev907nw0emcnzgg4r7239dd6zzluyd3cddss7r> để có thể xem các giao dịch trên address này



1. **Creating a simple smart contract**

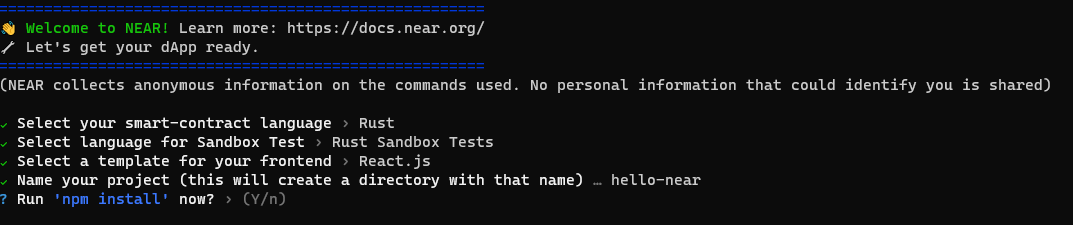
Do website [www.smartcontract.com](http://www.smartcontract.com) trong file không còn tồn tại nên em sẽ tạo smart contract trên NEAR Protocol



Gõ lệnh npx create-near-app để tạo NEAR project

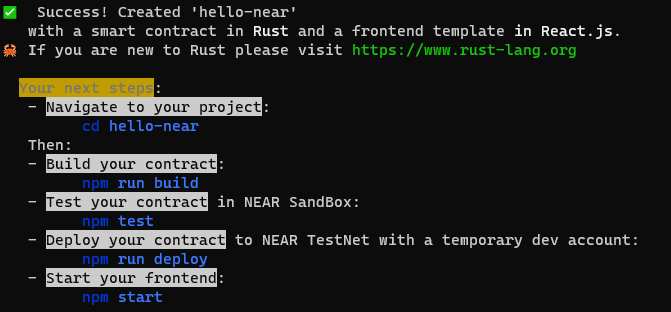
Nếu máy chưa có package thì sẽ cần phải tải package này

Ta có thể chọn ngôn ngữ cho smart contract, template cho front end, đặt tên cho dự án,...



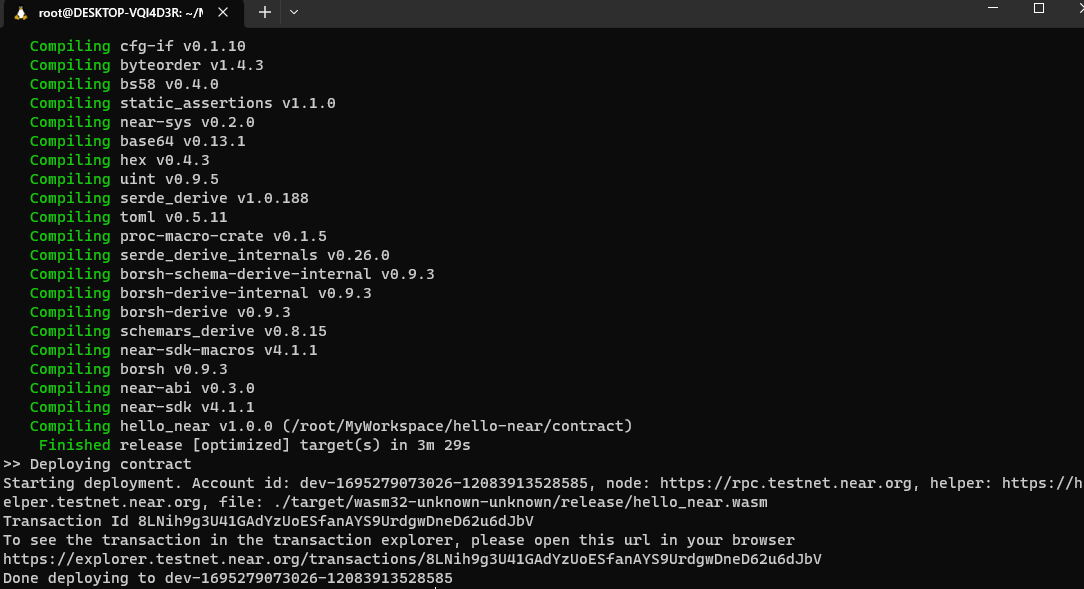
Nhấn y để tải về

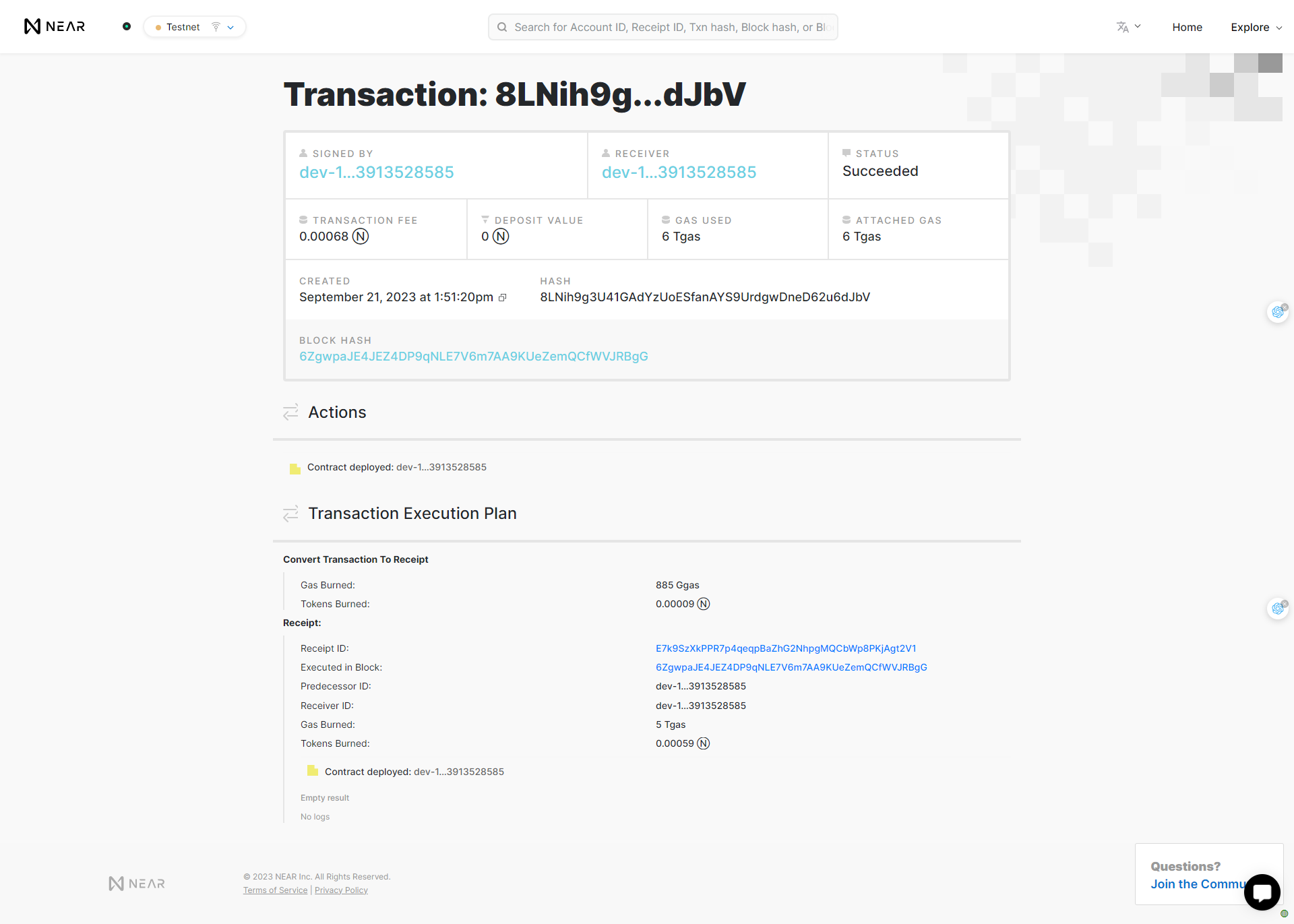
Thông báo sau khi tải xong



Sau đó ta gõ lệnh npm run deploy để deploy smart contract lên NEAR blockchain TestNet

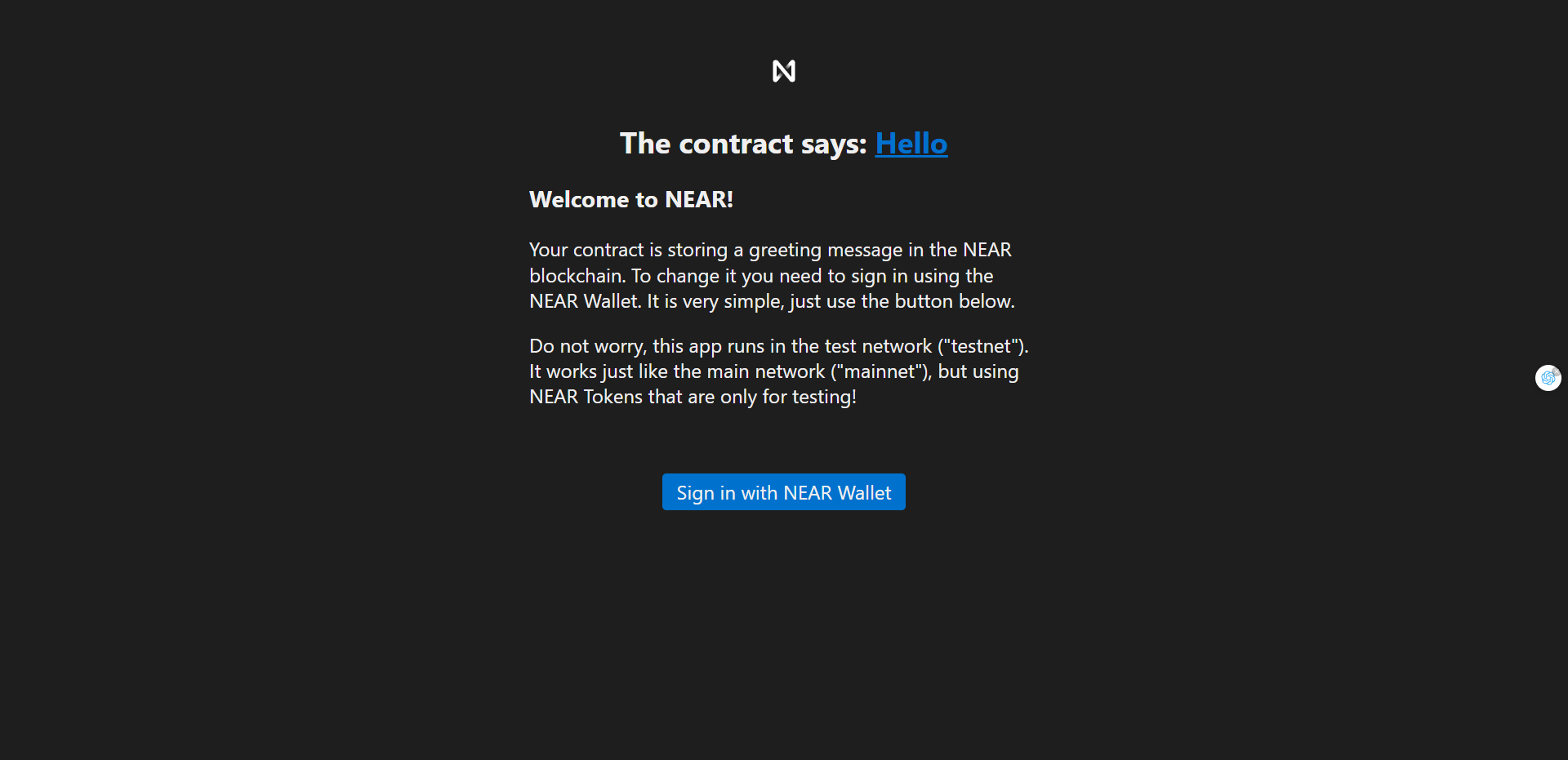
Dưới đây là thông tin khi deploy thành công smart contract, ta thấy được mã giao dịch của việc deploy contract và contract có public key là dev-1695279073026-12083913528585



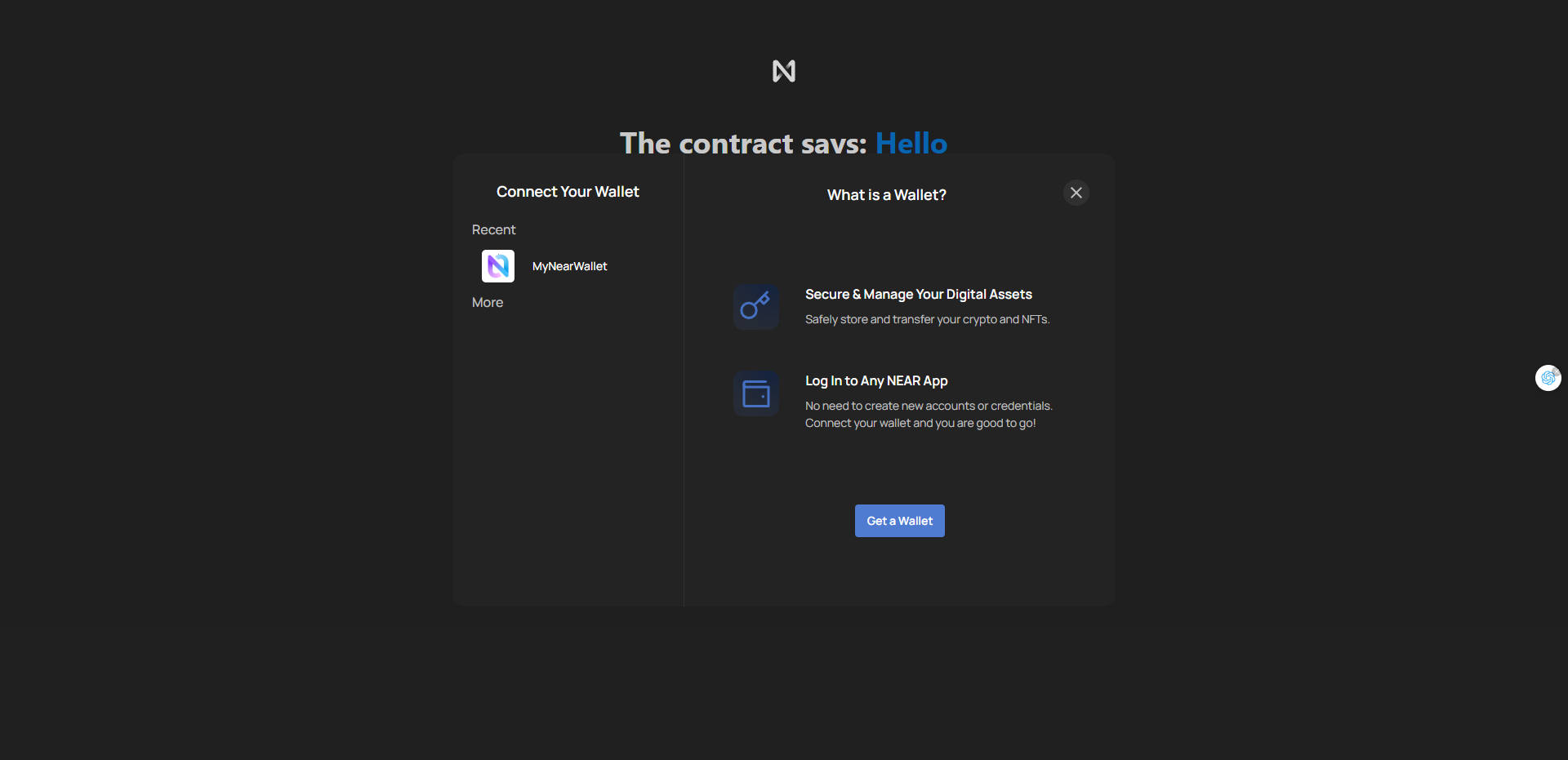


Tiếp theo ta gõ lệnh npm start để chạy phần front end của app và đây là giao diện web

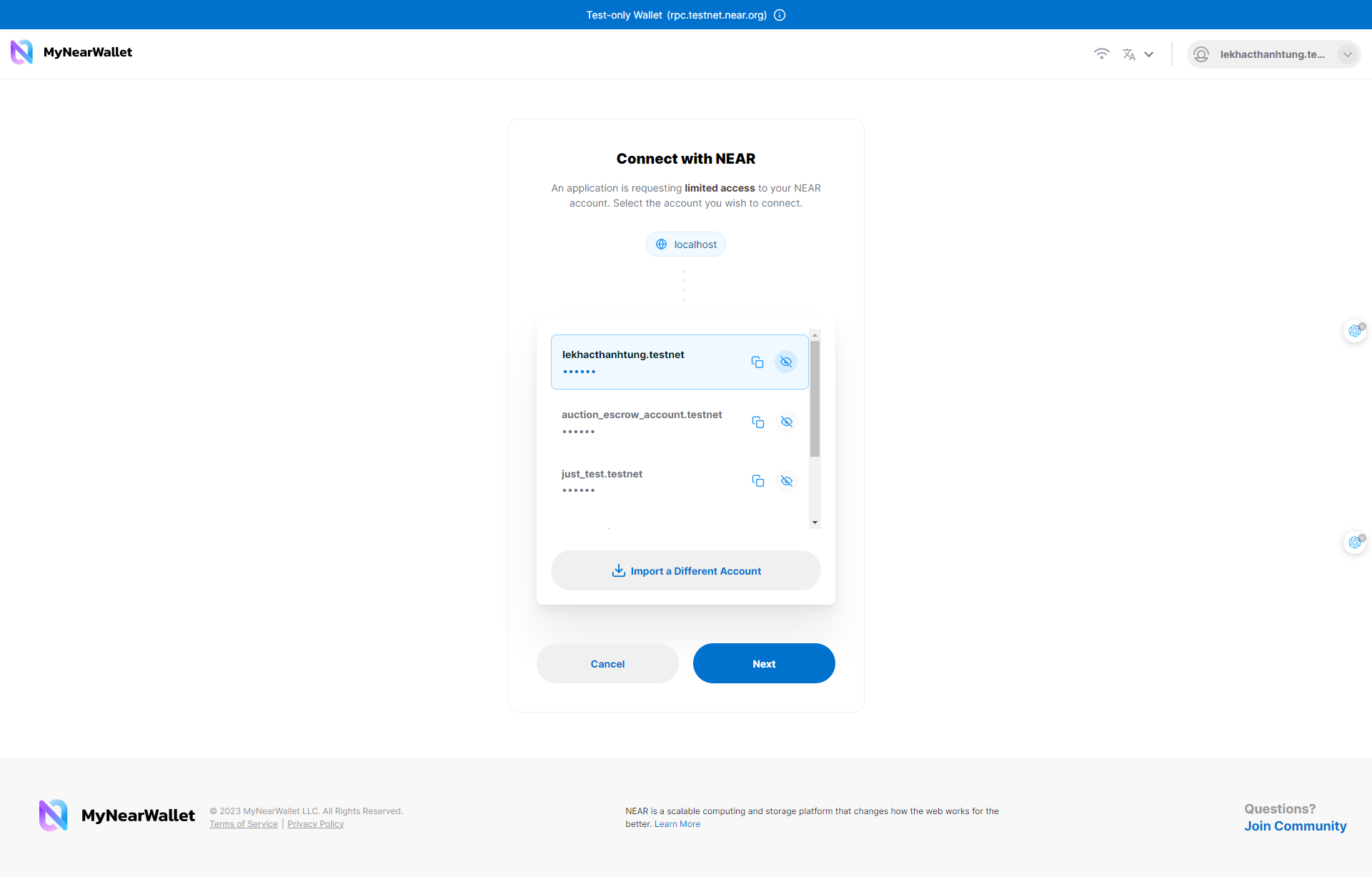
Ta có thể sign in với NEAR Wallet



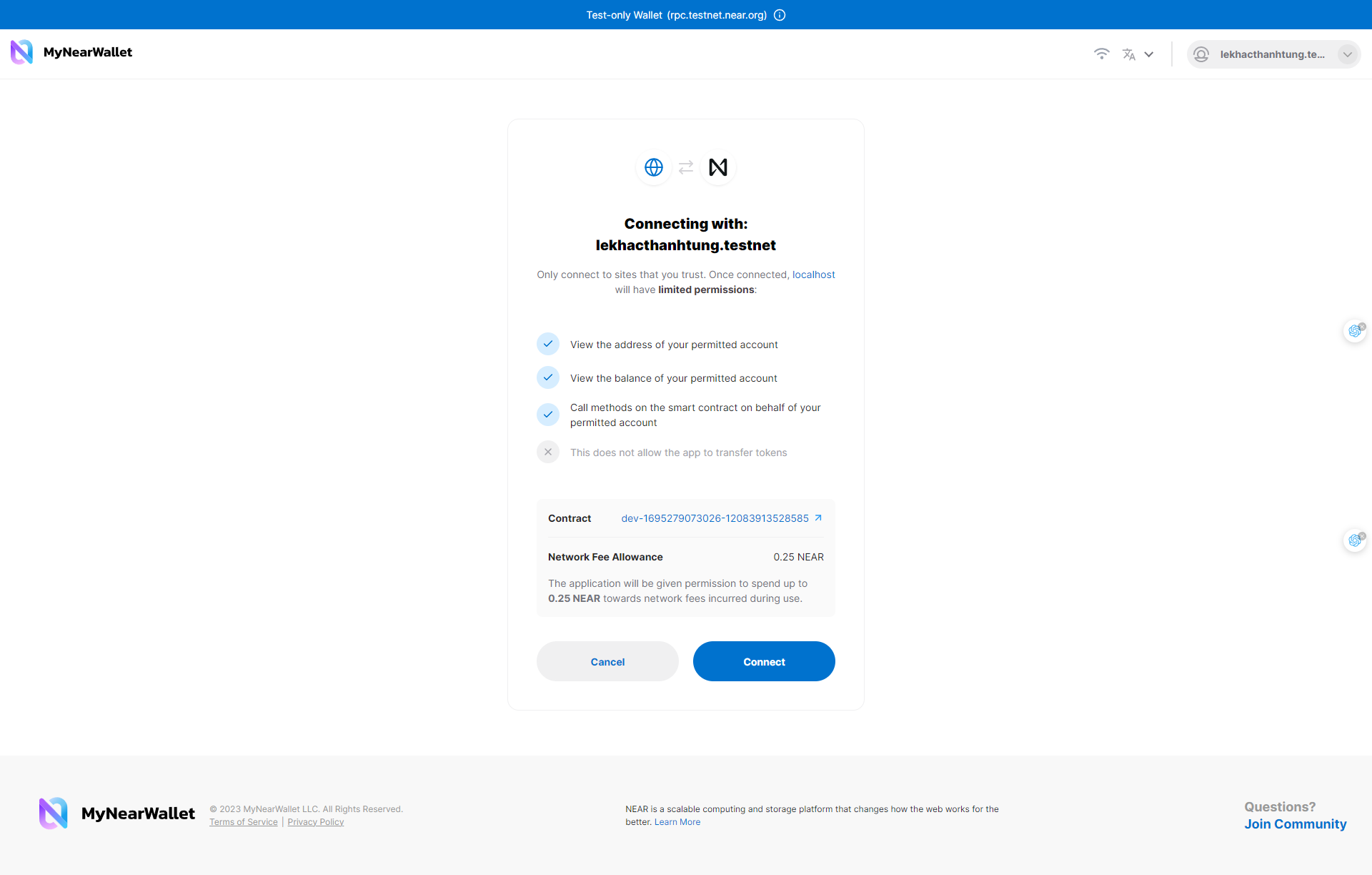
Chọn wallet (app có thể support nhiều wallet trong trường hợp này chỉ có MyNearWallet)



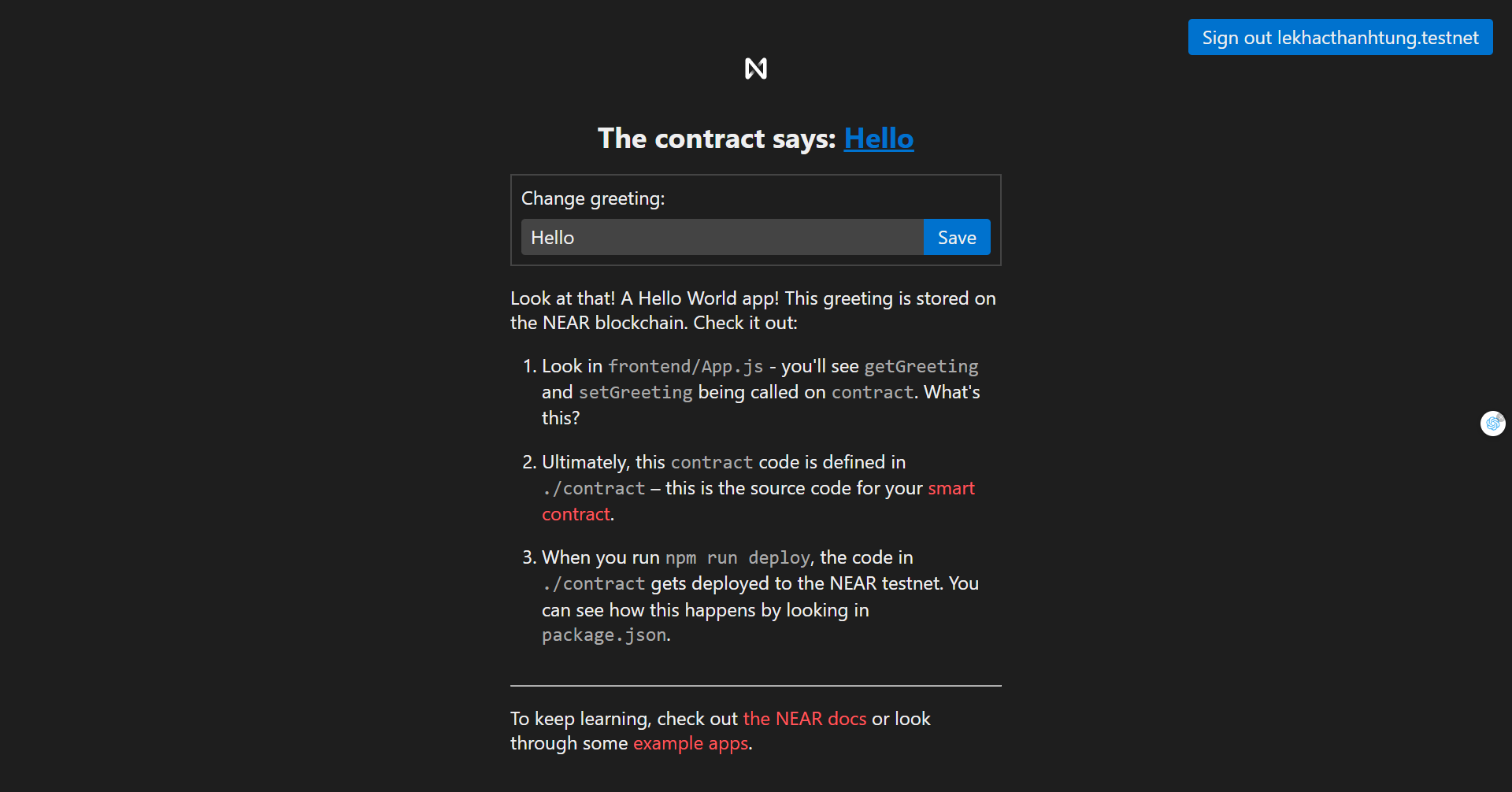
Nếu chưa có NEAR Wallet thì ta có thể tạo mới



Kết nối với wallet

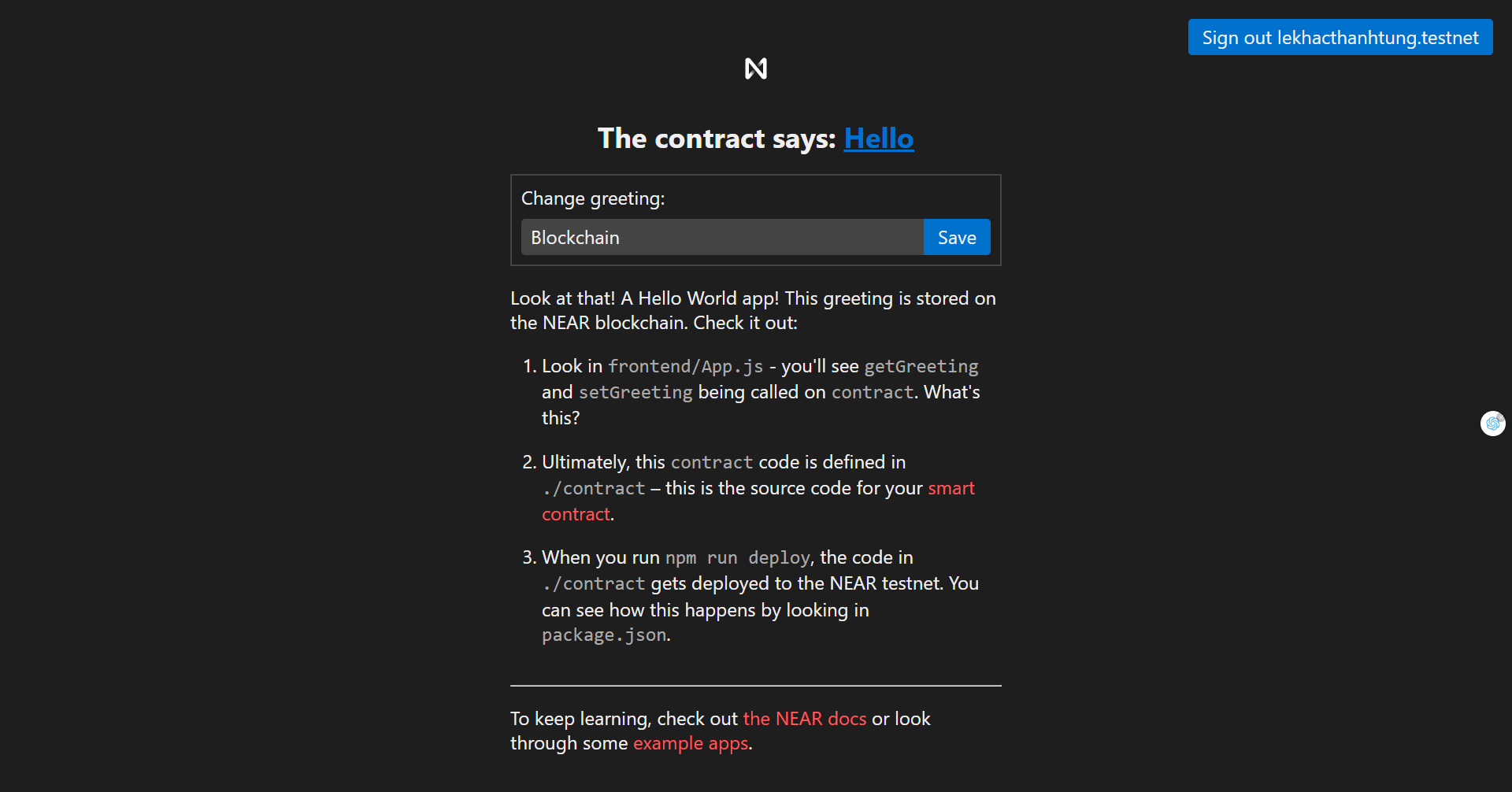


Sau khi đăng nhập với wallet thì chúng ta sẽ thấy được giao diện như thế này



Smart contract này chỉ có các chức năng như hiển thị, thay đổi greeting

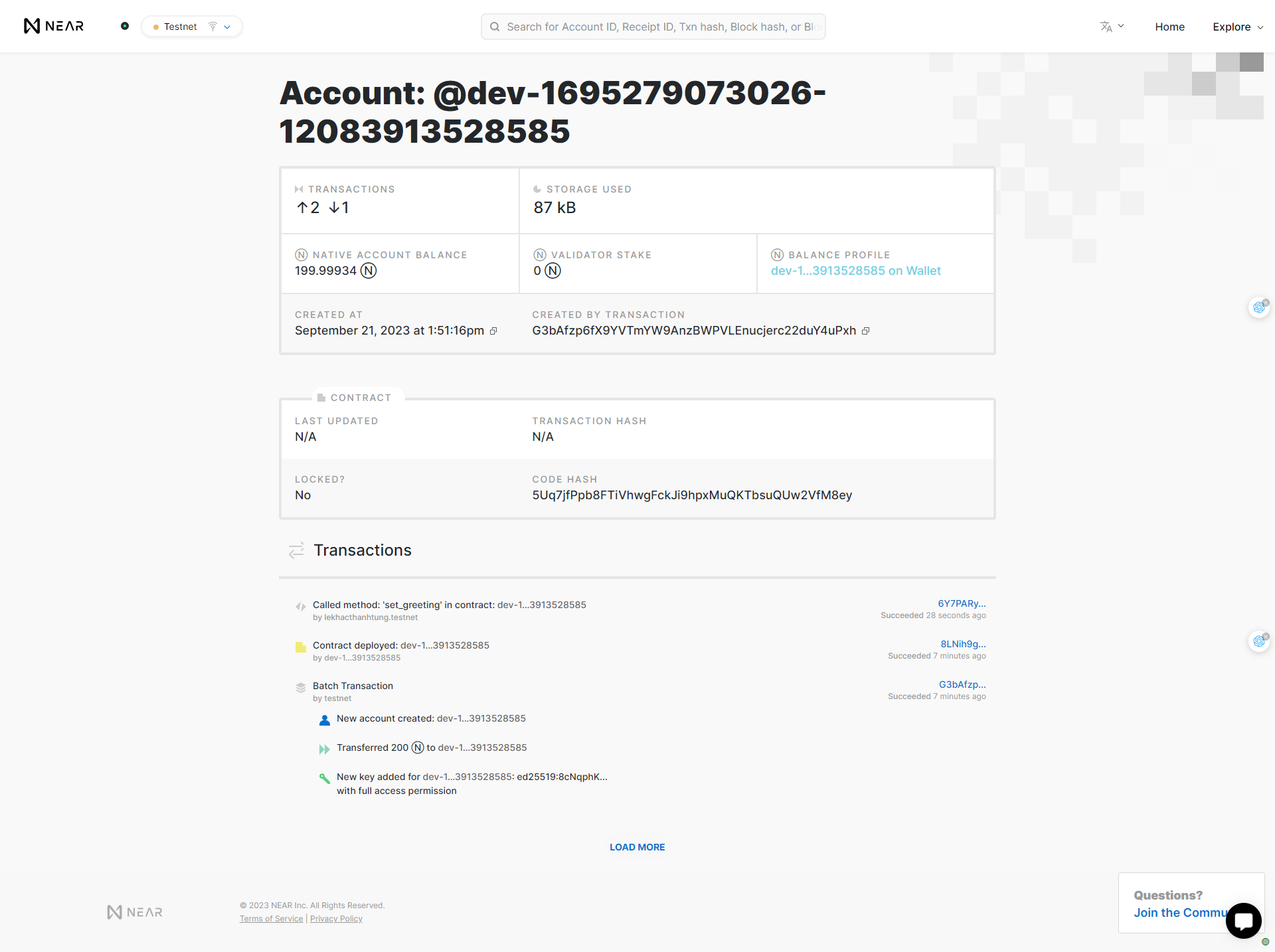
Giả sử em sẽ thay đổi Hello thành Blockchain

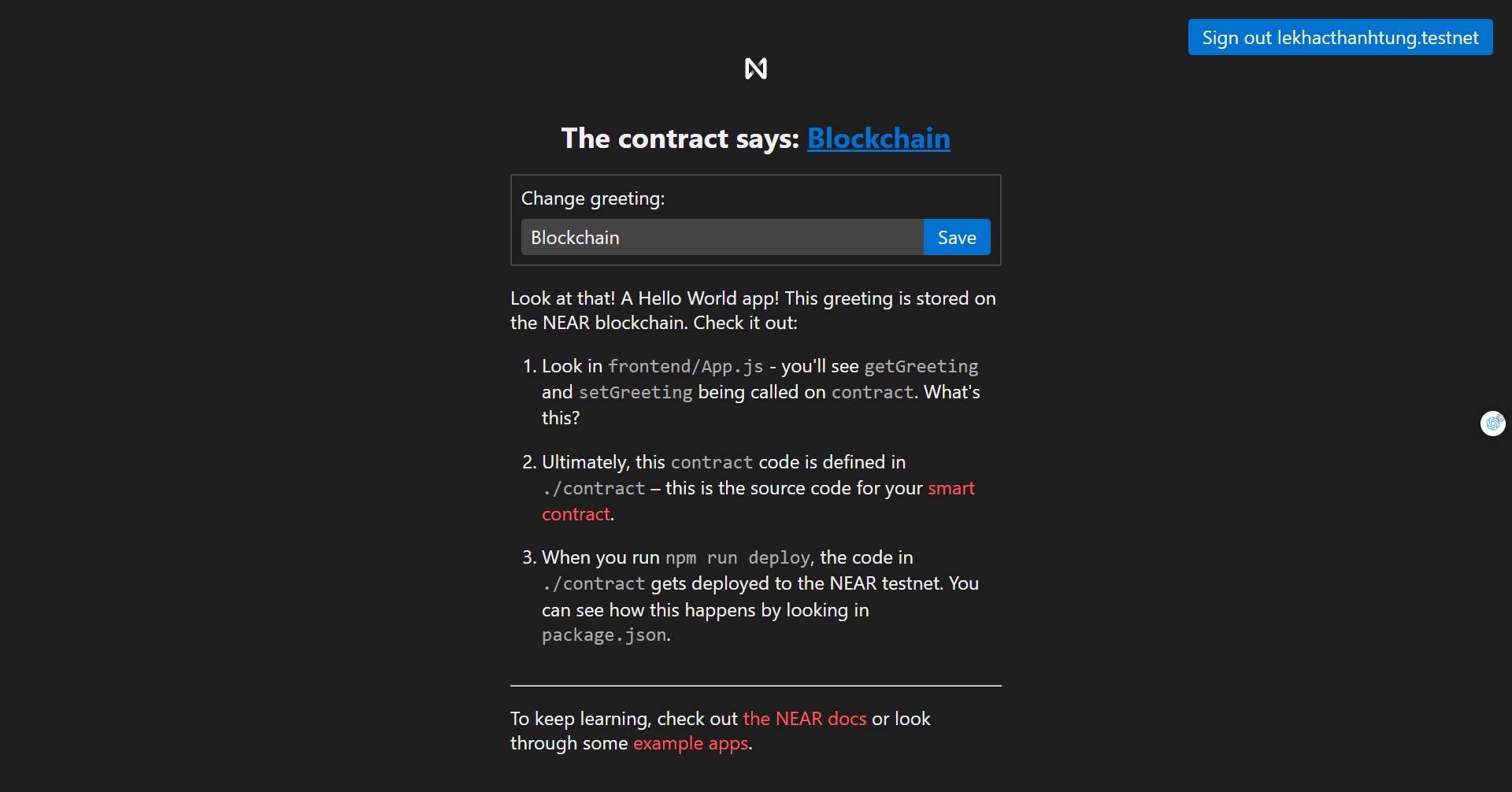


Nhấn save thì lúc này 1 transaction được thực hiện và thông tin transaction được lưu trữ trên blockchain

Ta có thể truy cập vào trang sau để xem các transaction được thực hiện trên smart contract của ta <https://explorer.testnet.near.org/accounts/dev-1695279073026-12083913528585>

Ta thấy method set\_greeting vừa được gọi và nó đã cập nhật lại giá trị từ Hello thành Blockchain

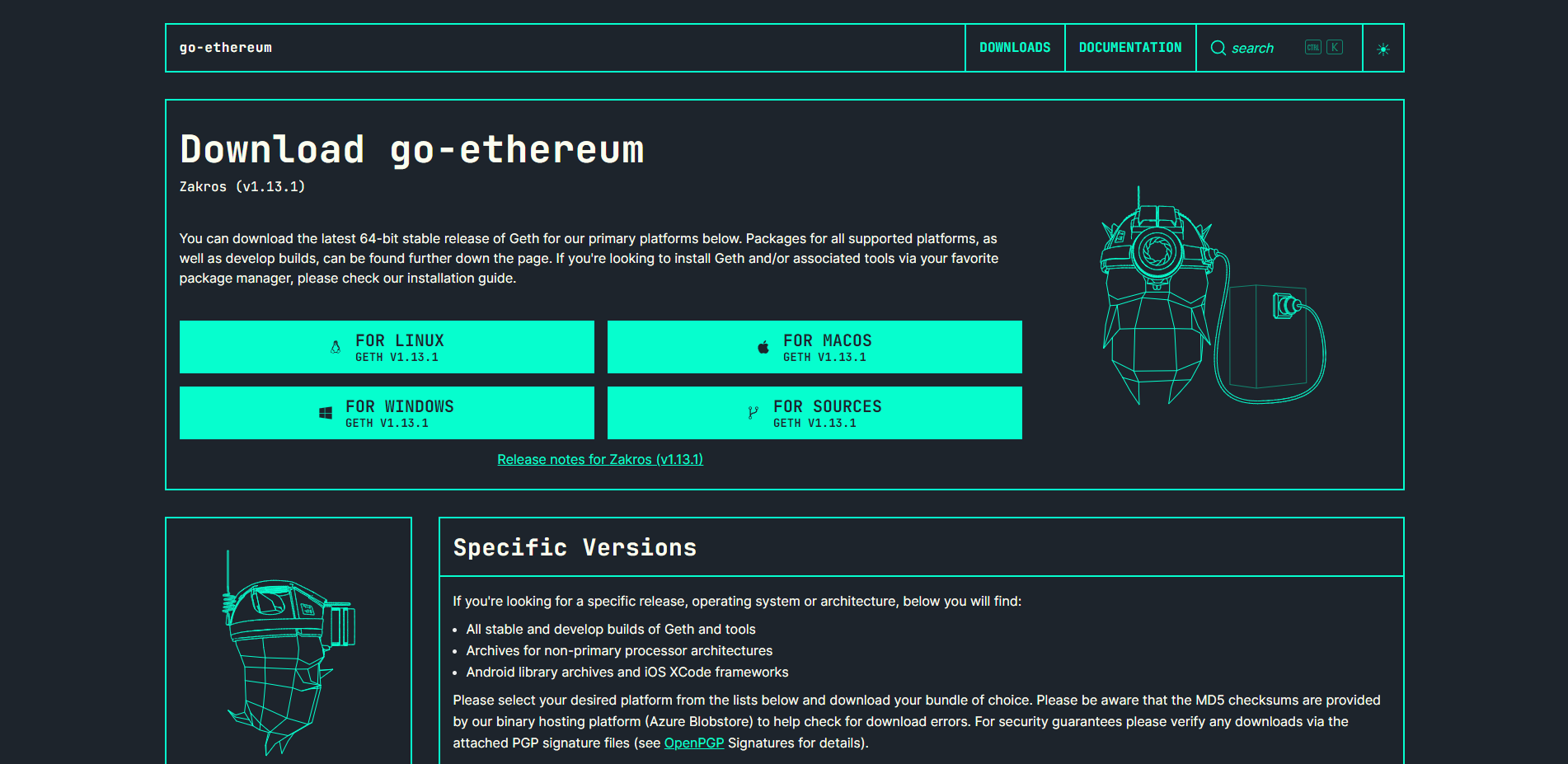




1. **Deploying a private blockchain (Etherum)**

Bước 1: Tải geth

Truy cập vào trang <https://geth.ethereum.org/downloads> để tải geth ứng với hệ điều hành đang sử dụng



Bước 2: Tạo folder my-private-blockchain



Bước 3: Tạo file genesis.json chứa thông tin về block đầu tiên sẽ lưu trên blockchain

Nội dung của file genesis.json:

| {  "config": {  "chainId": 12345,  "homesteadBlock": 0,  "eip150Block": 0,  "eip155Block": 0,  "eip158Block": 0,  "byzantiumBlock": 0,  "constantinopleBlock": 0,  "petersburgBlock": 0,  "istanbulBlock": 0,  "berlinBlock": 0,  "clique": {  "period": 5,  "epoch": 30000  }  },  "difficulty": "1",  "gasLimit": "8000000",  "extradata": "0x00000000000000000000000000000000000000000000000000000000000000007df9a875a174b3bc565e6424a0050ebc1b2d1d820000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",  "alloc": {  "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },  "f41c74c9ae680c1aa78f42e5647a62f353b7bdde": { "balance": "400000" }  } } |
| --- |

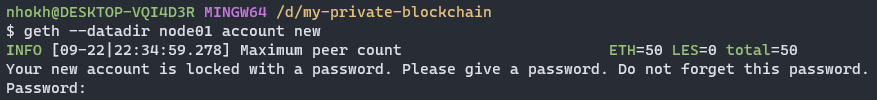
Mỗi node trong blockchain network của ta sẽ đều khởi tạo với cùng genesis block này.

Bước 4: tạo các local node

Đầu tiên tạo 3 thư mục sẽ ứng với 3 node trong blockchain của ta

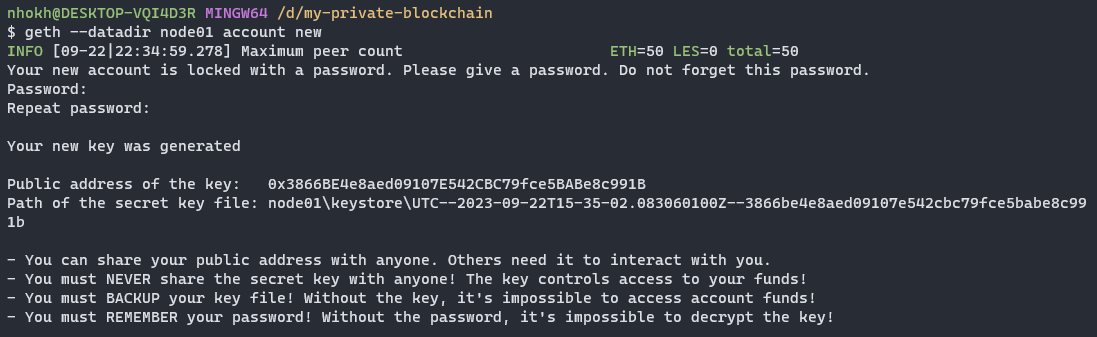


Tạo account trong node01



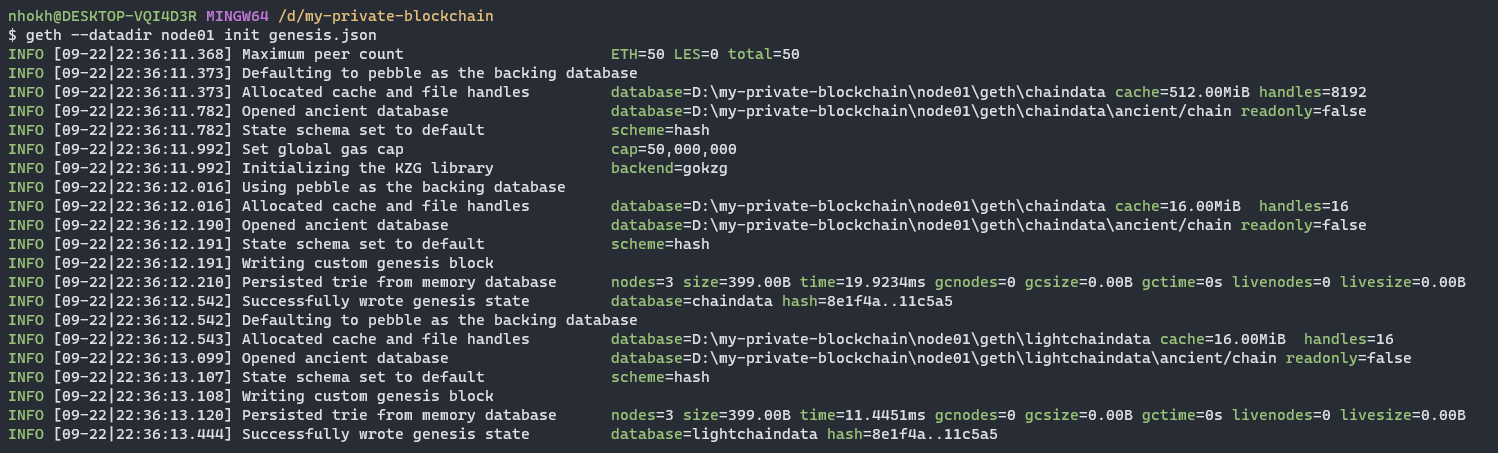
Ta sẽ nhập password cho account

Thông báo khi tạo account thành công gồm public key và địa chỉ lưu private key trên máy

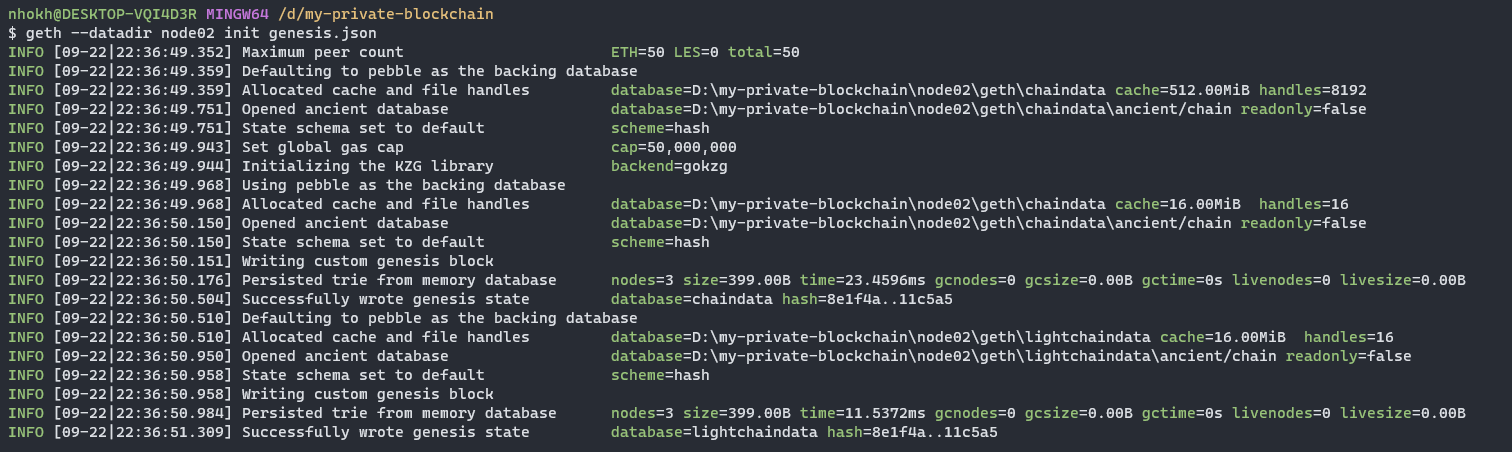


Bước 5: khởi tạo và run node

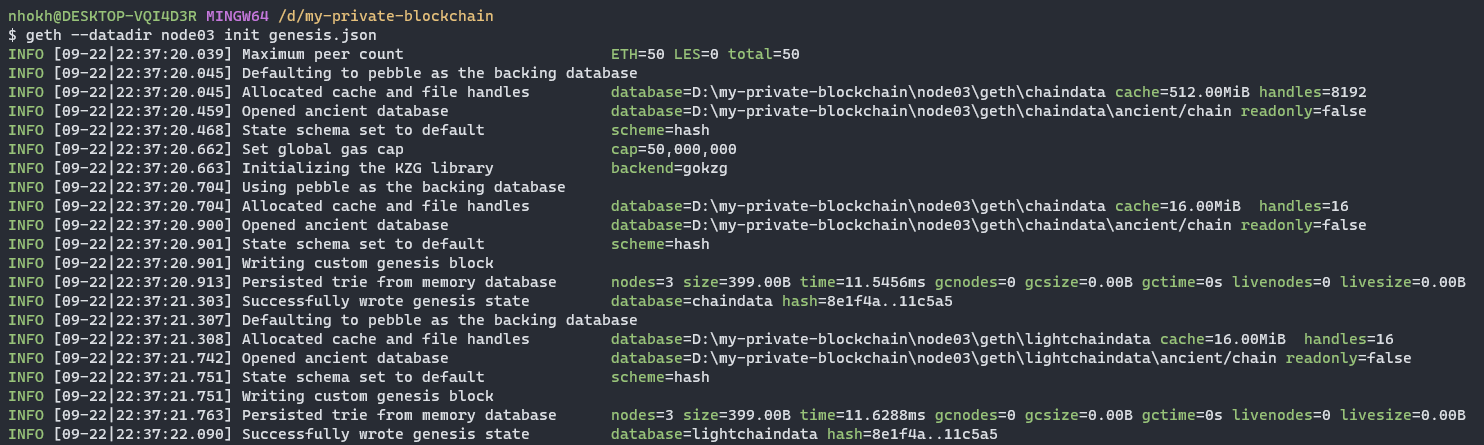
Khởi tạo node với genesis block



Khởi tạo node01



Khởi tạo node02

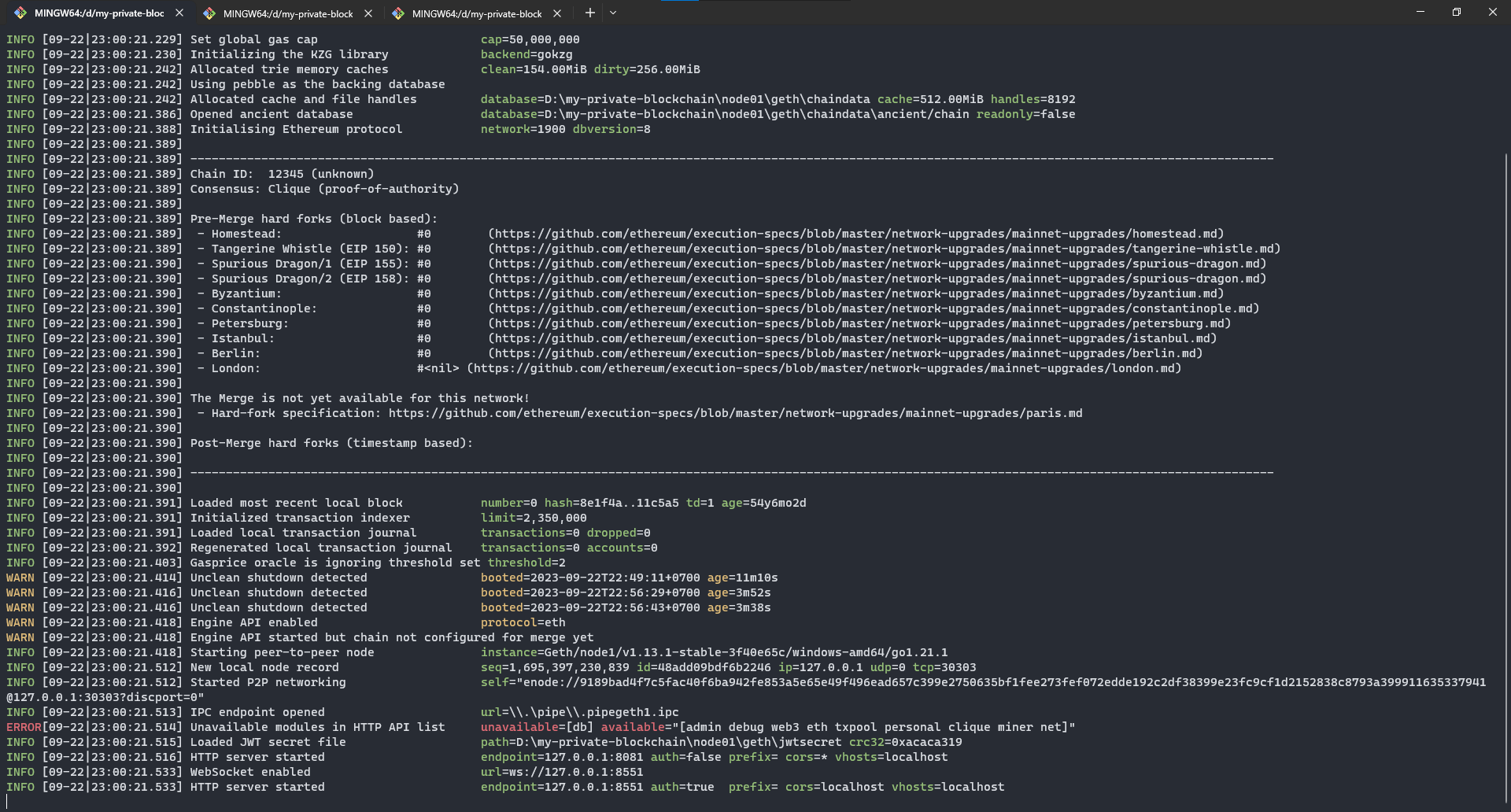


Khởi tạo node03

Chạy cùng lúc 3 node trên 3 terminal khác nhau

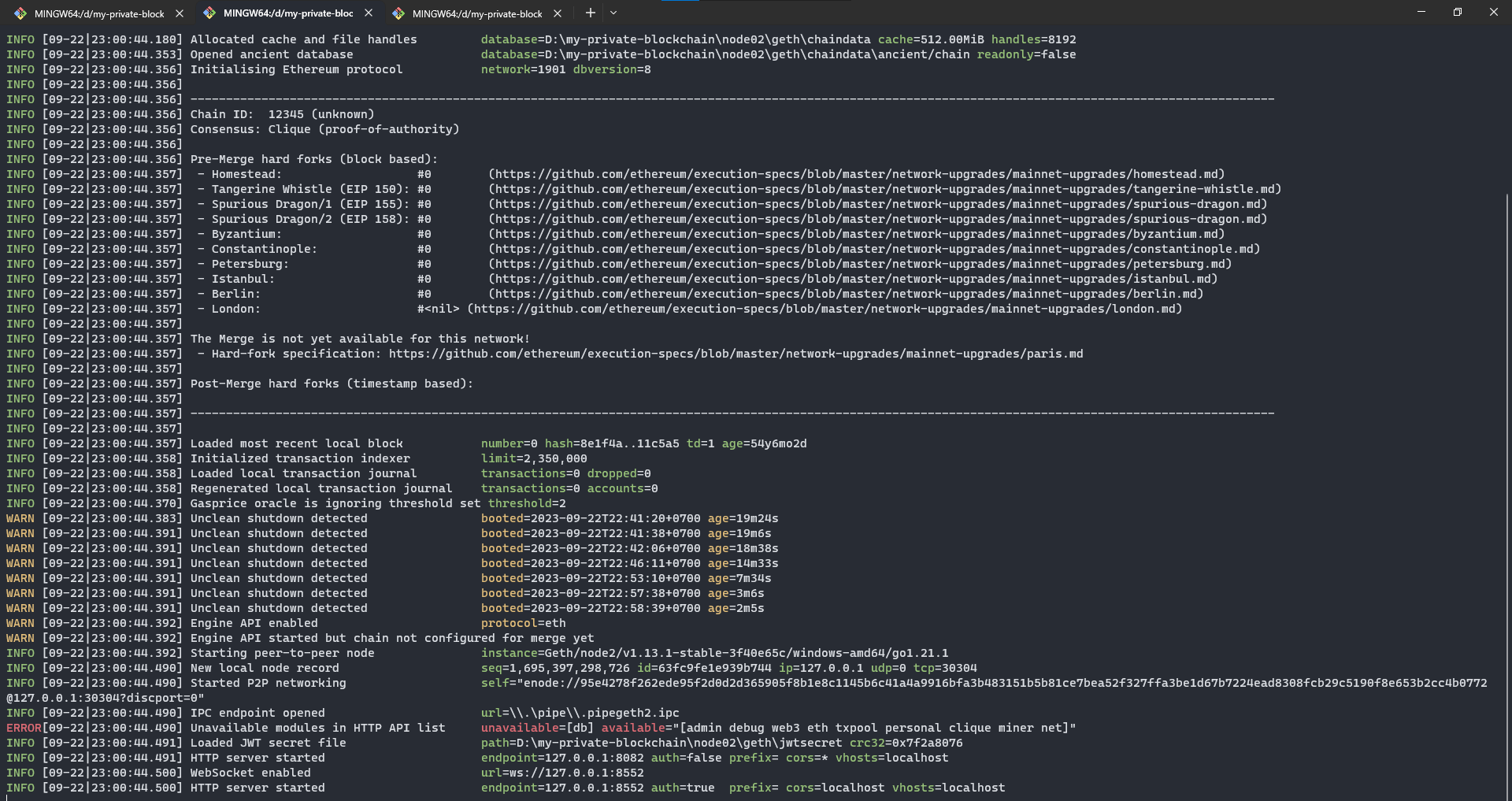
| geth --identity "node1" --http --http.port "8081" --http.corsdomain "\*" --datadir "node01" --port "30303" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 1900 --nat "any" --ipcpath \\.\pipe\geth1.ipc --authrpc.port 8551 |
| --- |

Chạy node01



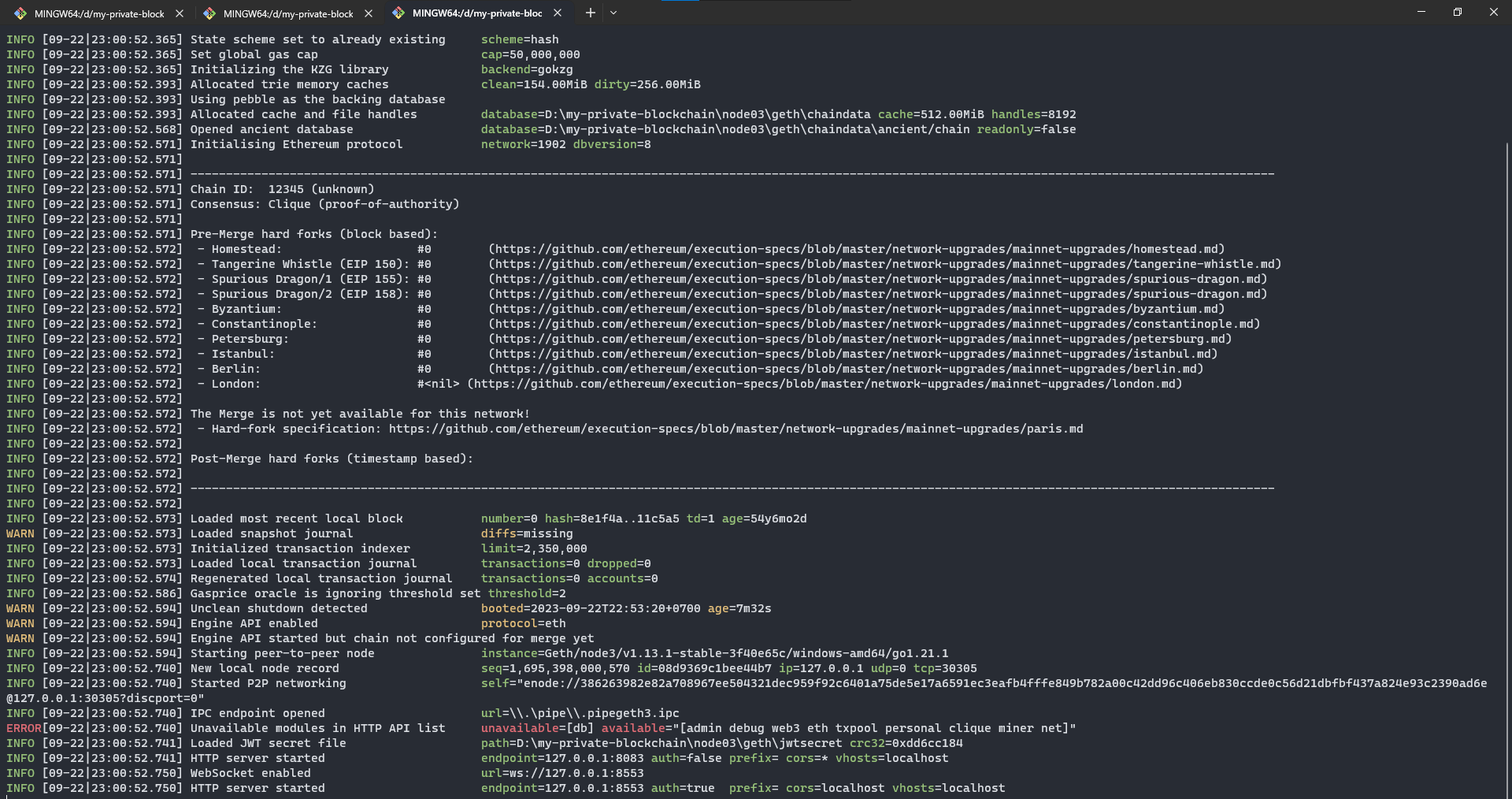
| geth --identity "node2" --http --http.port "8082" --http.corsdomain "\*" --datadir "node02" --port "30304" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 1901 --nat "any" --ipcpath \\.\pipe\geth2.ipc --authrpc.port 8552 |
| --- |

Chạy node02

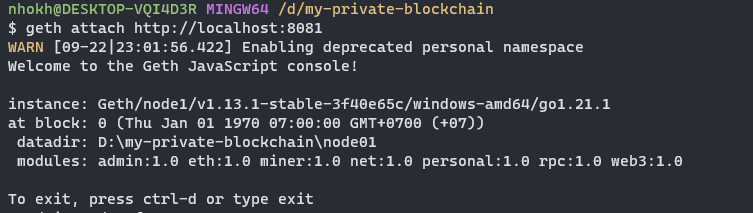


| geth --identity "node3" --http --http.port "8083" --http.corsdomain "\*" --datadir "node03" --port "30305" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 1902 --nat "any" --ipcpath \\.\pipe\geth3.ipc --authrpc.port 8553 |
| --- |

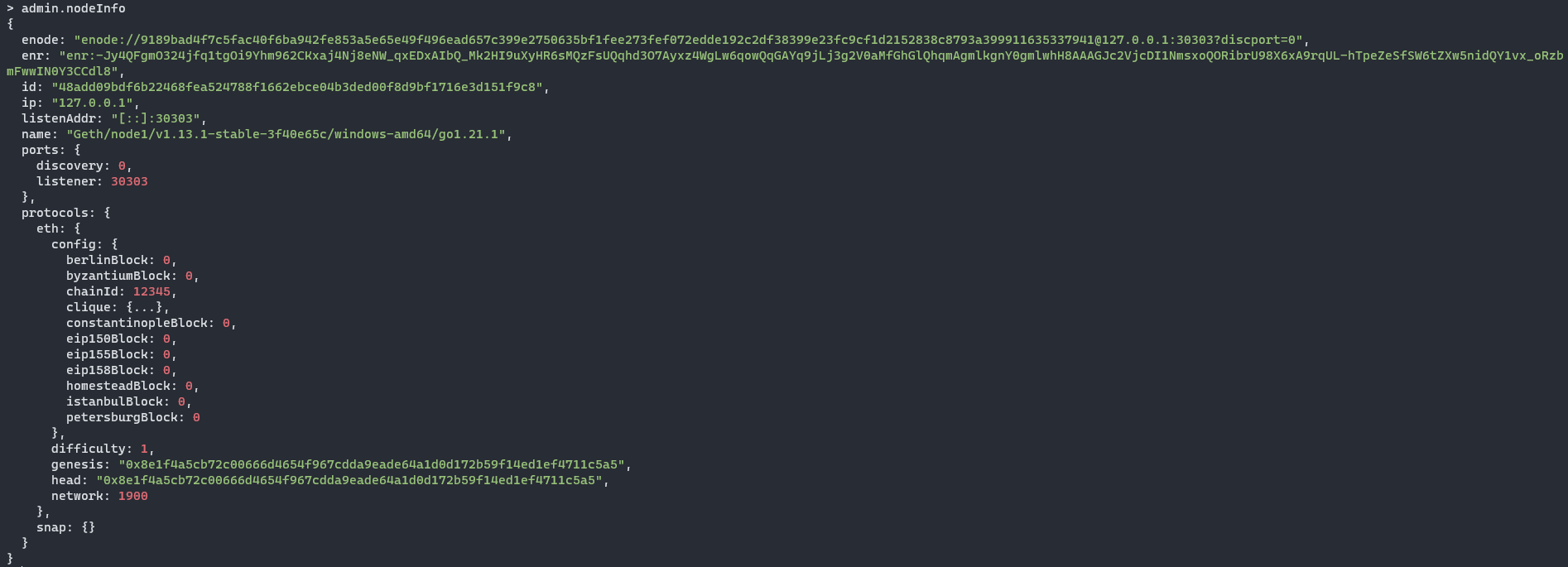
Chạy node03



Bước 6: kết nối với node thông qua CLI



Node đầu tiên là admin node

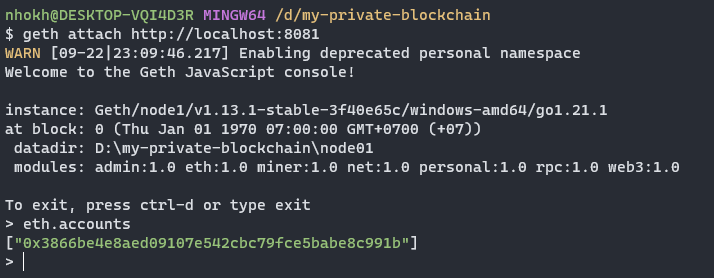


Copy enode address: "enode://9189bad4f7c5fac40f6ba942fe853a5e65e49f496ead657c399e2750635bf1fee273fef072edde192c2df38399e23fc9cf1d2152838c8793a399911635337941@127.0.0.1:30303?discport=0"

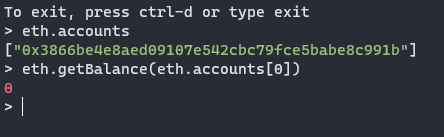
Bây giờ ta có thể kết nối tới những node khác bằng cách chạy lệnh sau ở mỗi terminal tương ứng của mỗi node

admin.addPeer("enode://9189bad4f7c5fac40f6ba942fe853a5e65e49f496ead657c399e2750635bf1fee273fef072edde192c2df38399e23fc9cf1d2152838c8793a399911635337941@127.0.0.1:30303?discport=0")

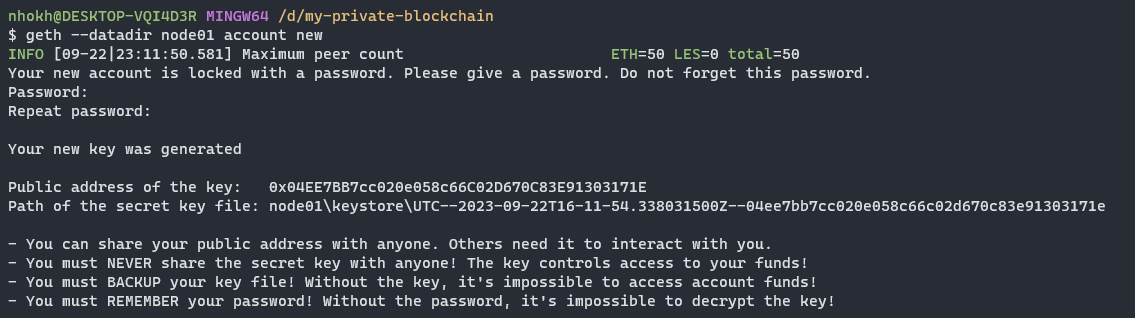
Xem account đã tạo ở node1



Kiểm tra balance của account



Tạo thêm account mới ở node1 để demo việc chuyển tiền





Ở đây em gặp các lỗi chưa fix được:

1. Khi gõ câu lệnh admin.addPeer() thì peerCount nó vẫn không thay đổi
2. Không chuyển được tiền (ở phiên bản geth mới nhất thì hàm chuyển tiền không được chuyển qua HTTP protocol nữa)
3. Miner





Mặc dù đã start nhưng bên terminal của node start miner vẫn không có thông tin gì về việc đào