Buổi 18. Tương tác hợp đồng Helloworld với giao diện frontend.

Xin chào các bạn , chào các bạn đã quay trở lại với khóa học lập trình aiken cơ bản trong bài học trước các bạn cũng đã biết cách viết một hợp đồng thông minh Helloworld và tương tác với chúng bằng commandline interface . trong bài học này mình giới thiệu cho các bạn một bài học thú vị hơn đó là tương tác với hợp đồng này bằng giao diện frontend.Oke chúng ta cùng bắt đầu bài học.

Trong bài học này gồm những nội dung chính sau. Thứ nhất,chúng ta sẽ Tạo một dự ám nextjs cơ bản trong bài học này mình sẽ dung nextjs làm framework frontend chính để thực hiện tướng tác với hợp đồng thông minh. Thư Hai, Chúng ta sẽ thực hiện đọc validator từ hợp đồng thông minh và hiển thì ra màn hình sau đó. Mình sẽ thực hiện viết ra các chức năng để kết nối Ví. Dự án này chúng ta sẽ sử dụng Nami làm ví chính để thực hiện tương tác với hệ sinh thái Cardano. Sau đó mình sẽ nhập 1000 ada để lock vào hợp đồng thông minh Oke bây giờ mình sẽ nhấn lock . Mình sẽ kí một giao dịch mình sẽ kí . Như vậy là mình đã kí thành công giao dịch khi mà transaction Hash được in ra màn hình chứng tỏ quá trình lock tài sản của mình thành công . trong bài học này sẽ dung lucid và api blockfrost làm công cụ chính để tương chính với hợp đồng thông minh Như vậy transactionHash đã được in ra và mình đã bị lock thành công 1000 ada vào hợp đồng thông minh Tiếp đến mình sẽ thực hiện unlock 1000 ada vào hợp đồng thông minh để lấy lại 1000 ada vừa lock vào mình sẽ thực hiện kí giao dịch để unlock mình sẽ nhập mật khẩu để kí Khi transaction hash được in ra chứng to quá trinh unlock tài sản của chúng ta thành công. Như vậy để biết đên bài học này các bạn cần chắc kiến thực frontend.

Bây giờ mình sẽ kiểm tra xem giao dịch của mình như thế nào bằng cách nhấn vào từng transaction Hash Như khóa tài sản mình sẽ xem và tada quá trình giao dịch tài sản đã được ghi lại như vậy quá trình khóa tài sản vào hợp đồng thực hiện thành công bây giờ chúng ta sẽ thực hiện kiểm tra hợp đồng unlock tài sản bằng việc nhấn vào transaction Hash

Install NodeJS

Oke như vậy mình đã thực hiện demo thành công dự án bây giớ chúng ta cùng bắt đầu bài học để bắt đầu bài học này các bạn cần có NodeJS trong bài học này mình sẽ thực hiện kiểm tra NodeJs bằng cách sử dụng từ khóa Node -v để kiểm tra Nếu các bạn chưa có NodeJs thì hãy lên gg và search NodeJS để tải NodeJS về đồng thời nên cài đặt Node 16 trở lên và phiên bản dành cho hầu hết người sử dụng thì sẽ ổn định hơn. Đi cùng với Nodejs và npm và npx các bạn có thể kiểm tra nó tương dự như kiểm tra NodeJs để kiểm tra sự tồn tại Oke Hiện tại dự án của mình đã sẵn sàng để phát triển bây giờ mình sẽ tạo dự án next js bằng cách npx create next app + tên dự án trong bài học này mình sẽ đặt tên là Helloworld fronend và Enter . Trong bài học này mình sẽ sử dụng typescript làm ngôn ngữ dev chính trong dự án Mình sẽ chọn Yes hêt để cài đặt một số sự phụ thuộc và các tính năng do nextjs cung cấp Như vậy mình đã sẽ cài đặt dự án Oke như vậy mình đã thực hiện thành công dự án bây giờ sẽ có thư mục dự án của mình vừa tạo là helloworld frontend mình sẽ thực hiên cd vào dự án để tải về

Install Lucid cardano and Cbor-x.

Hiện tại mình đã thực hiện cài đặt thành công dự án bây giờ mình có 2 phần quan trọng nữa cần được triển khai luôn đó là thứ nhất install lucidcardano. Mình sẽ vào chrome search Lucid cardano oke mình sẽ thực hiện install lucid cardano bằng cách copy npm install lucid cardano và thực hiện paste vào terminal để install nó về. Thứ 2 cùng với Lucid cardano là cbor-x mình sẽ thực hiện cbor-x giúp biểu diễn các đối tượng nhị phân ngắn gọn Giúp cho việc tiết kiệm được chi phí giao dịch trên cardano. Mình sẽ thực hiện cài đặt cbor-x bằng npm mình sẽ paste và enter Như vậy mình đã thực hiện cài đặt thành công dự án. Bây giờ chúng ta sẽ thực hiện mở phần dev lên và thực hiện phát triển nó . Mình sẽ chạy dự án này với từ khóa npm run dev dự án của mình sẽ được chạy ở PORT 3000 bây giờ mình cùng vào đó để kiểm tra. Như vậy dự án đang được biên dịch và khởi tạo . bây giờ chúng ta hay cùng xem kết quả nhé …. Oke Như vậy dự án của chúng ta đã được tạo thành công bây giờ hãy cùng mình vào trong dự án và sẽ một vài thay đổi nhé. Mình sẽ thực hiện xóa phần này đi và viết vào trong đây là aiken helloworld bây giờ mình sẽ vào trong giao diện của mình và xem một vài thay đổi Oke như vậy là giao diện của chúng ta đã thay đổi rồi.

ReadValidator

Bây giờ mình cùng bắt đầu viết chương trình đầu tiên của chúng tôi mình sẽ thực hiện copy file plutus.json được sinh ra trong quá trình biên dịch hợp đồng copy vào thư mục libs như vậy đã thành công tiếp đến mình sẽ tạo thư mục utils và thực hiện viết hàm đọc validator mình sẽ đặt tên là readvalidator.js

Mình sẽ thực hiện import file vừa biên dịch từ file pulus.json và viết 1 hàm read validator trong hàm này mình sẽ thực hiện các chuawcs năng chính sau thứ nhất là tìm các validator từ hợp đồng thông minh mình sẽ tim validator có tital là contract helloword nếu chưa có thì mình sẽ ném ra 1 lloxi là validator not found đồng thời

Sau khi có validator rồi mình sẽ thực hiện encode validator theo cborx mình sẽ biến dổi một chút và sau đó mình sẽ return trả về spendvalidator gồm type script mình sẽ lấy type là plutusverrsion 2 và

Usecontext dùng để chia sẻ state với các component khi mà người dùng thay đổi trạng thái thì context giúp làm việc này

Định nghĩa Datum và redeemer

Như vậy là mình đã thực hiện viết hàm đọc validator thành công. Bây giờ mình sẽ định nghĩa datum và redemer trong dự án này datum và redeemer khá quan trọng nó là điều kiện không thể thiếu trong hệ sinh thái cardano mình sẽ tạo một folder và định nghĩa datum và redeemer trong đó chúng ta cùng

cần được đính kèm với một UTXO tại một địa chỉ tập lệnh. Điều này được thực hiện để giảm yêu cầu bộ nhớ và cho phép truy cập nhanh chóng trong khi xác thực các giao dịch.

create file datum .ts để tạo datum trong đây trước tiên mình cần import data from lucid cardano sau đó mình sẽ định nghĩa cấu trúc của datum bằng data.object trong datum của hợp đồng thông minh có trường Ower mình sẽ định nghĩa chúng là kiểu byte array ứng với hợp đồng thông minh. Tiếp đó mình sẽ định nghĩa kiểu cho cho datum bằng data.static và trong minh sẽ ép kiểu là datum schema . Được rồi mình đã định nghĩa thanh công bây giờ mình sẽ export datum ra ngoài bằng cách ép kiểu cho datum là kiểu của datum mình vừa định nghĩa và unknown

Sau khi định nghĩa datum thành công mình sẽ định nghĩa redeemer . redmmer được gắn với đầu vào của giao dịch để mở khóa tiền từ một tập lệnh và được tập lệnh sử dụng để xác thực giao dịch Trong hợp đôg thông minh của chúng ta redeemer sẽ là msg mình cũng sẽ đinh nghĩa redeemer này giống như định nghĩa datum mình sẽ có một trường msg để thực hiện xác thực và nó sẽ có kiểu là byte array

Đồng thời mình sẽ định nghĩa kiểu của redeemer bằng data.static và ép kiểu cho redeemer

Sau đó minh sẽ thực hiện việc export HelloworlRedemer và ép kiểu cho nó là Kiểu của reddmer mà mình vừa định nghĩa