CMPE483

Homework 2

Kadir Gökhan Sezer

Demet Yayla

Zeynep Aydoğan

In the first project of this course, we made a smart contract written in solidity. The purpose of this assignment was to write an interface that communicates with this contract and performs its functions.

Icon

Description automatically generated**Frontend:**

Javascript already has two different libraries such as web3.js and ether.js that fulfill these connections. We've used both. We chose NextJS as the framework.

NextJS is actually a ReactJS framework. ReactJS is essentially a JS framework. In other words, we are using the framework of the framework.

**Backend:**

The backend is only used to save TX's. We chose Flask as the most convenient and simple way to do this.

Logo, company name

Description automatically generated**DB:**

To store the data, we chose postgresql, which is easy to communicate with python.



We deployed our interface to the internet. As a virtual machine, we chose EC2, a service of AWS. We get the domain from guzelhosting.com. We used the Route53 service where AWS registers the namespaces for the domain to IP connection.

url: bounance.online

Logo

Description automatically generated with medium confidenceIf you want to run this UI for local, hardhat may help you.

Firstly, you need to install all the dependencies

* npm i

Then,

1. **If you did not change the mygov.sol file:**

You need two separate terminals to execute these commands:

To run a node:

1-> npx hardhat node

To deploy the contract to the local node

2-> npx hardhat run --network localhost scripts/deploy.js

To run the UI

3->npm run dev

1. **If you did:**

npx hardhat clean

npx hardhat compile

rm -rf artifacts

cp -rf src/artifacts .

Then you can go the option a.

**SCREEENSHOTS**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, application

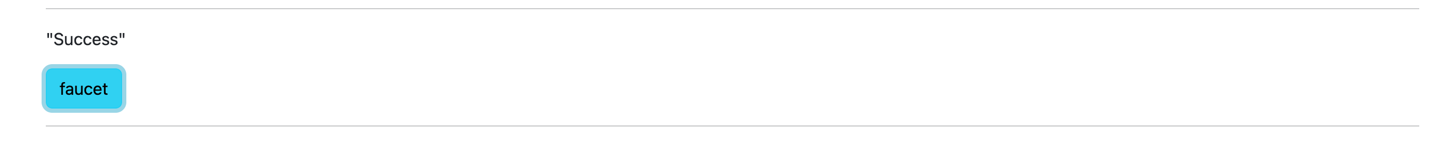
Description automatically generated

When we connect to an account:

Graphical user interface, text, application, email

Description automatically generated

Faucet:

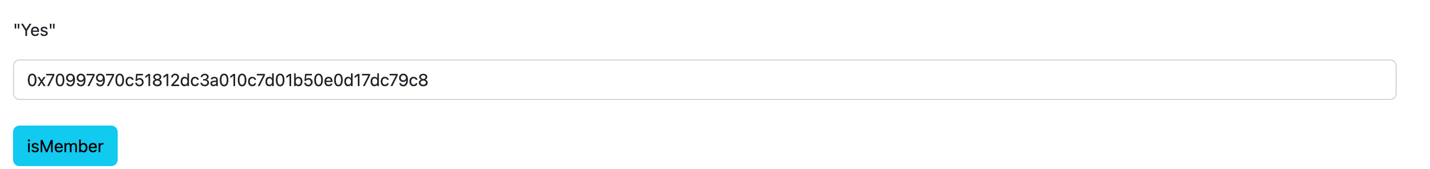


balanceOf:

Graphical user interface, application

Description automatically generated with medium confidence

isMember:



getUser:

