Egecan Serbester
Batuhan Çetin
Emre Türker

CMPE483 Homework 2 Report

Table of Contents

Project Overview

We developed an autonomous decentralised governance token contract called MyGov that can make some functionalities requested in Homework 1.
In this project, we developed a web-based user interface for the MyGov DAO project that we have developed in Homework 1. Thanks to this web application users can link their wallets, engage in surveys, contribute funds, propose projects, and vote on project financing. The smart contract was deployed on Sepolia Testnet and leveraged the ethers.js (v 5.6) library to interact with the Ethereum blockchain.

We used Remix for developing, testing and deploying the MyGovToken and USDStableCoin contracts in the Sepolia test network. We also used ethers.js Library to handle Ethereum-related tasks. Finally, we benefitted from Web3 Provider to connect the Ethereum blockchain and users to engage with the smart contract using their Ethereum wallets.

Egecan Serbester
Batuhan Çetin
Emre Türker

Implementation

At first, we corrected the previous mistakes that we made in the contracts, after we ensured all functionalities were implemented successfully by testing them on Remix, we developed the web-based user interface for our contracts.

Index.js

We used index.js for handling our contract's functionalities in the front end. The web interface can connect to our smart contract via Web3 and ethers.js libraries used in this class. Function responses and error handling were also handled in that section. Functions will be explained in the User Guide in more detail.

Constants.js

We created constants.js to simplify the interaction between MyGovToken and USDStableCoin contracts. ABI (Application Binary Interface) and contract addresses are stored as constants in this file. If the user wants to use another MyGovToken (different deployed) just changing the contract address is enough for using our web interface.

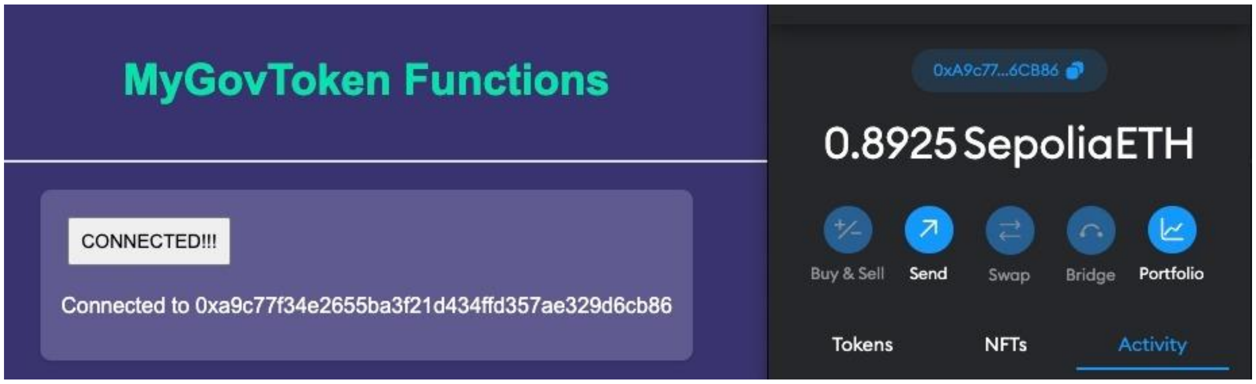
Index.html

We designed a basic HTML page that consists of input fields, buttons and response texts under that file.

Egecan Serbester
Batuhan Çetin
Emre Türker

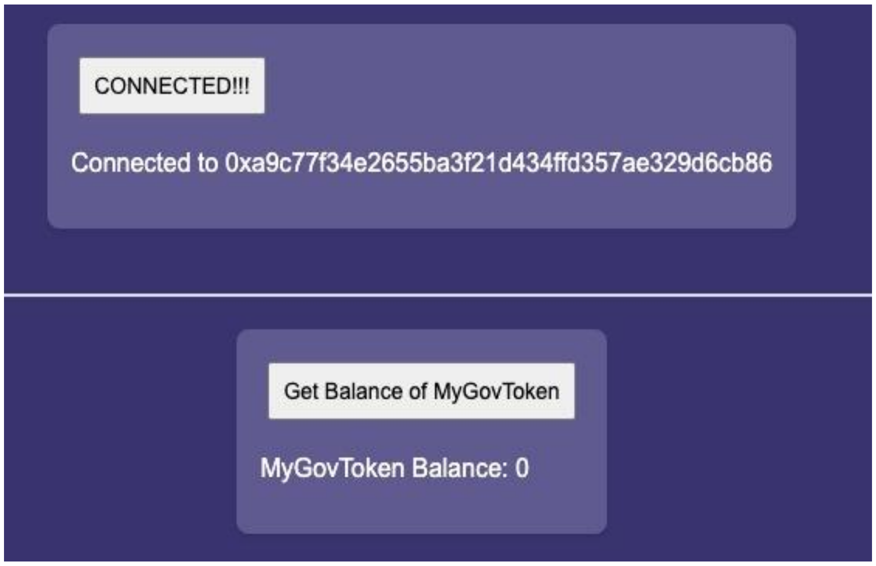
User Manual

Users must connect an ethereum wallet to use our Token's functionalities in our web interface.



Connected to 0xa9c77f34e2655ba3f21d434ffd357ae329d6cb86

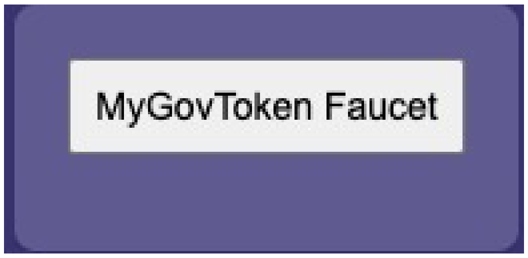
After users connect their Ethereum wallet, they can see their account addresses on the bottom of button.

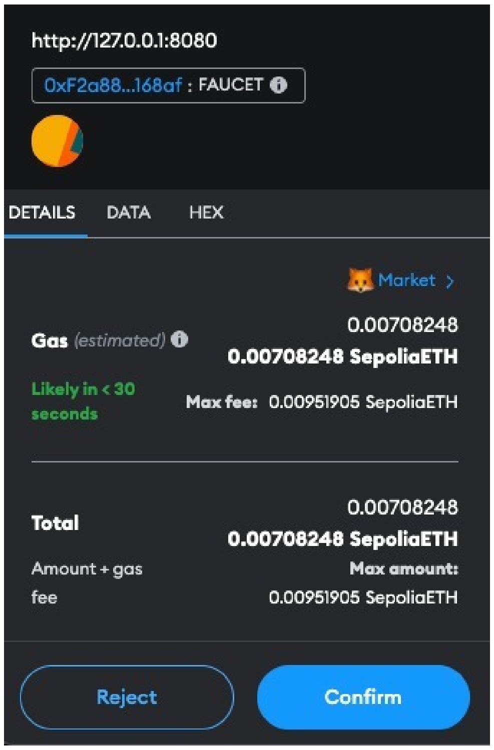


Connected to 0xa9c77f34e2655ba3f21d434ffd357ae329d6cb86

MyGovToken Balance: 0

Users can check their MyGovToken balances by clicking that button. At first it is 0 by default, they can increase their balances by Faucet it (or sending amount to their account by using transfer from functionality for the deployed MyGovToken contract address from Remix).



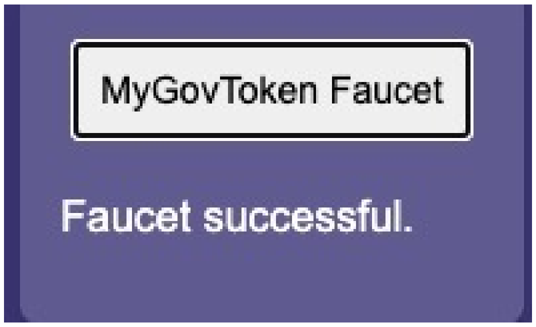


Gas (estimated) 0.00708248 **0.00708248 SepoliaETH**
Likely in < 30 seconds **Max fee:** 0.00951905 SepoliaETH

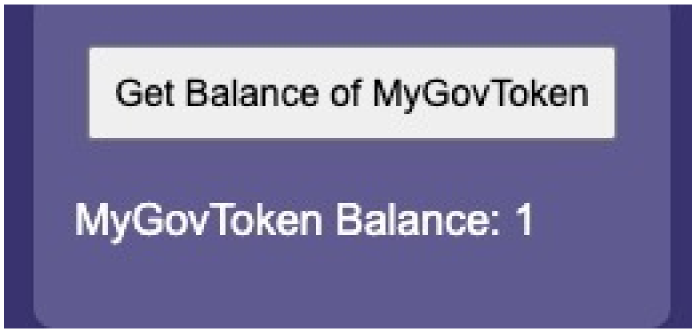
Total 0.00708248 **0.00708248 SepoliaETH**
Amount + gas fee **Max amount:** 0.00951905 SepoliaETH

Users can faucet a MyGovToken by clicking on that button. After clicking a screen from MetaMask will be popped, you should confirm that transaction to get a token.

After transaction is completed, a response message will show dynamically under the MyGovTokenFaucet button:

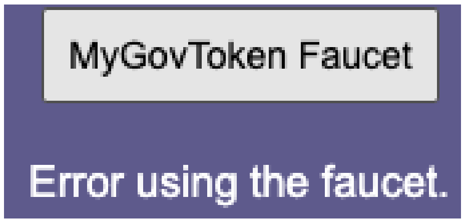


After that message, when user click on the Get Balance of My Token Button Again it will be updated as like this:



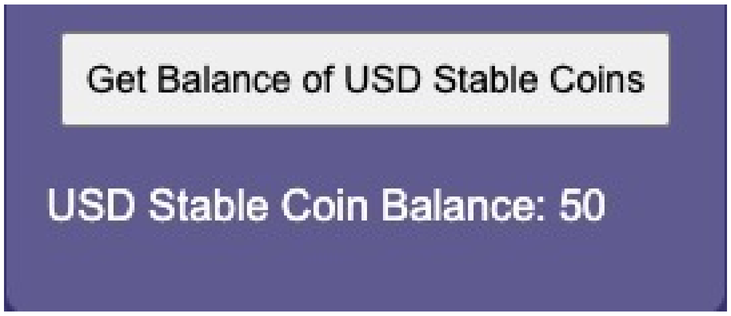
MyGovToken Balance: 1

However users can only use faucet MyGovToken once according to MyGovToken policies, they can not use that functionality again. When they try to do faucet again they will see that error message on the screen.



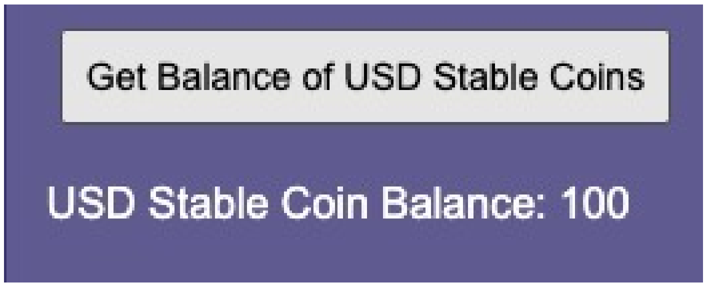
Error using the faucet.

Users can also see the balances of their USDStableCoins like they do in MyGovToken. It is 50 at first.



USD Stable Coin Balance: 50

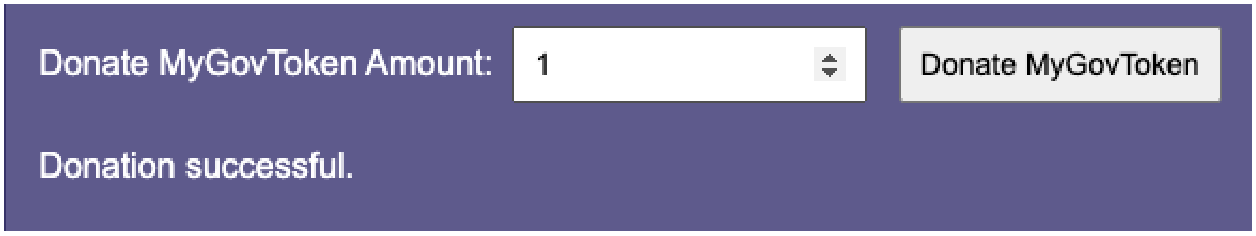
Users can faucet a USDStableToken by clicking the USDStableCoin Faucet button after they clicked that button the same story in Faucet MyGov Token will occur and at the end they see Faucet Successful message at the bottom of the button. After that when users check their balance, it will be increased as 50.



USD Stable Coin Balance: 100

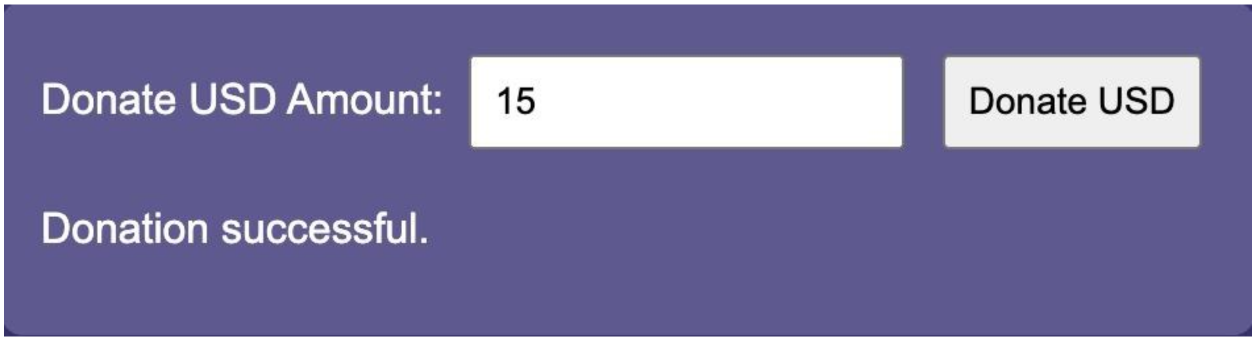
Unlike the MyGovToken Faucet function, users can faucet USDStableToken more than one.

Egecan Serbester
Batuhan Çetin
Emre Türker



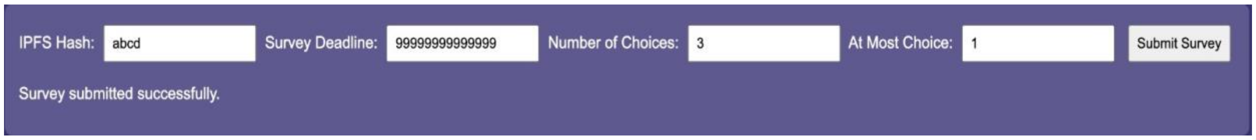
Donation successful.

Users can donate MyGovToken, after transaction result with success they can see the message under the button and the amount will be dropped their balances.



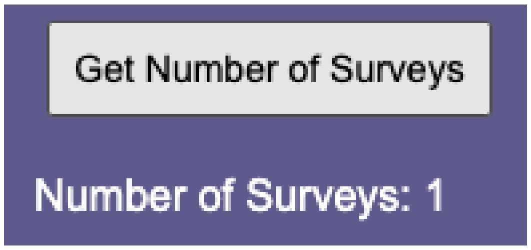
Donation successful.

Users can also donate USD Stable Token, after transaction result with success they can see the message under the button and the amount will be dropped their balances.


IPFS Hash: abcd
Survey Deadline: 99999999999999
Number of Choices: 3
At Most Choice: 1

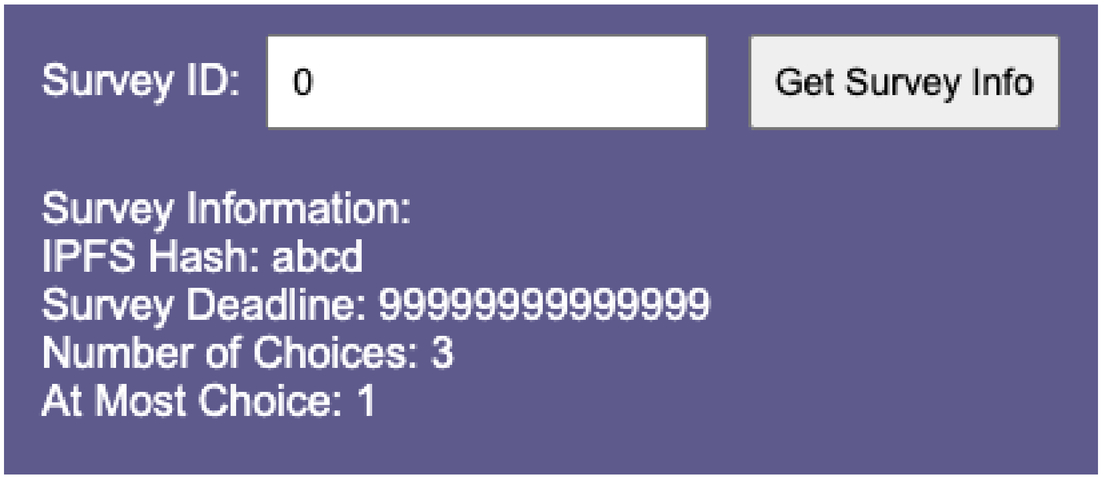
Survey submitted successfully.

Users can submit surveys with IPFS Hash, Survey Deadline (in seconds), number of choices and at most choice if they have required balance for USD and MyGovToken they can submit it successfully. After the transaction for submission is done, they will see the response message.



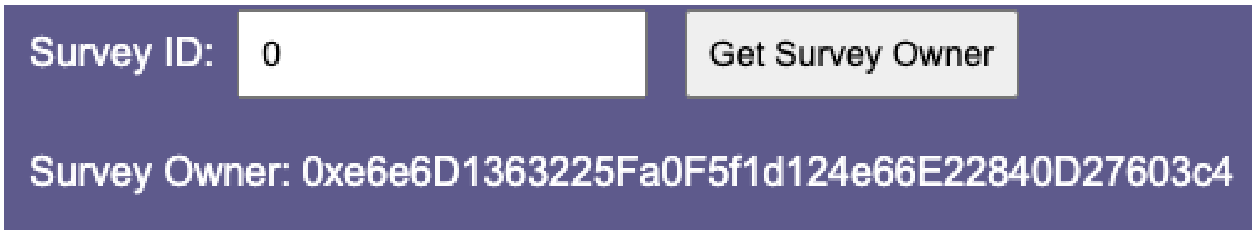
Number of Surveys: 1

Users can see how many surveys that the contract has by clicking get Number of Surveys.



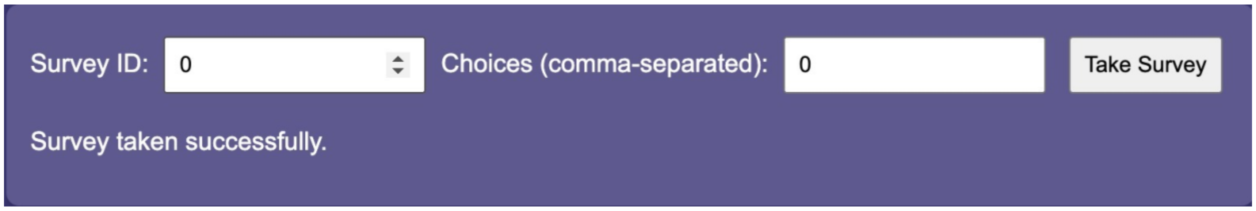
Survey Information:
IPFS Hash: abcd
Survey Deadline: 99999999999999
Number of Choices: 3
At Most Choice: 1

Users can see the survey info ID is starting from 0 and increment one by one as survey count increasing.



Survey Owner: 0xe6e6D1363225Fa0F5f1d124e66E22840D27603c4

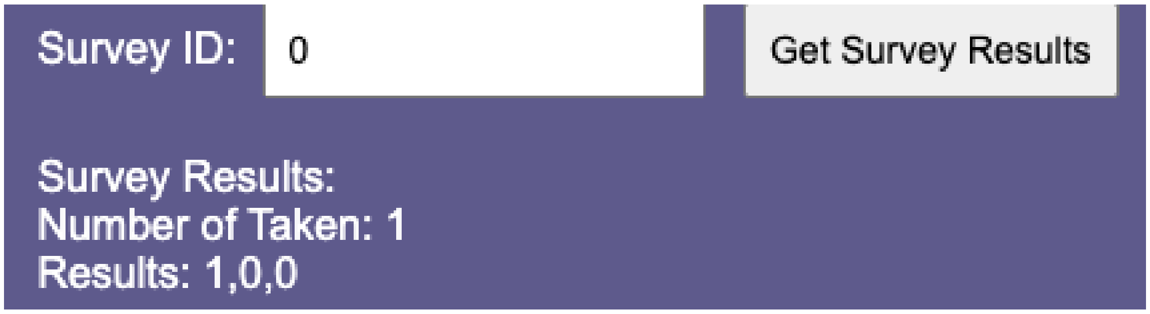
Users can see the survey owner (Because Batuhan was the owner the address is different from connecting address)



Survey taken successfully.

Users can make a choice for surveys (Choices are starting from 0, they will increase to the size of choices-1)

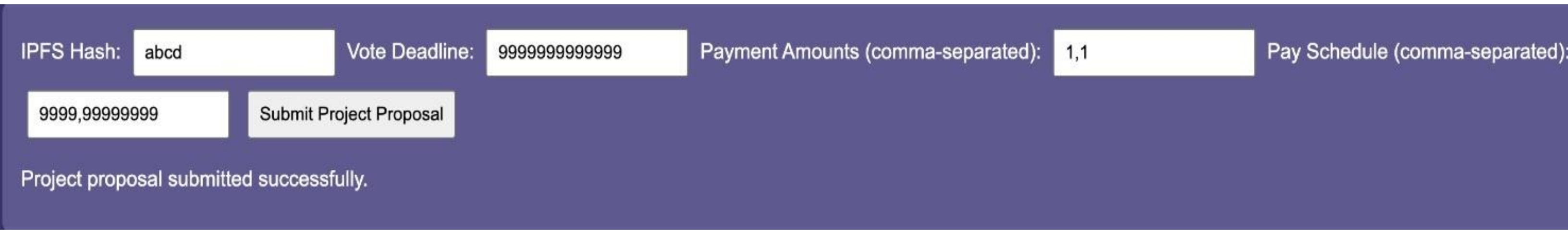
Egecan Serbester
Batuhan Çetin
Emre Türker


Survey ID: 0

Get Survey Results

Survey Results:
Number of Taken: 1
Results: 1,0,0

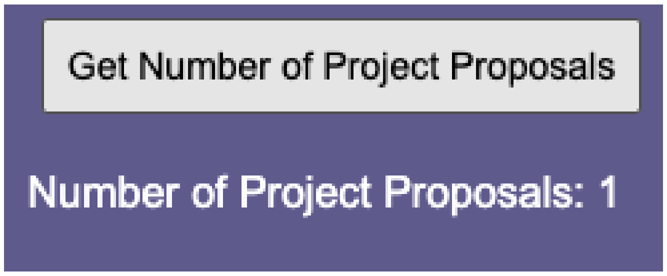
Users can see the survey results. 1,0,0 means: Choice 1 is chosen by one others are not chosen for now.


IPFS Hash: abcd
Vote Deadline: 9999999999999
Payment Amounts (comma-separated): 1,1
Pay Schedule (comma-separated): 9999,99999999

Submit Project Proposal

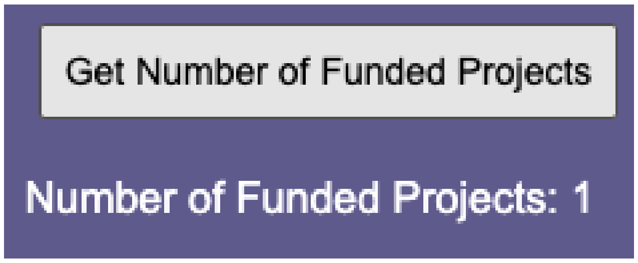
Project proposal submitted successfully.

Users can submit project proposal after it resulted with success they can see the response message.


Get Number of Project Proposals

Number of Project Proposals: 1

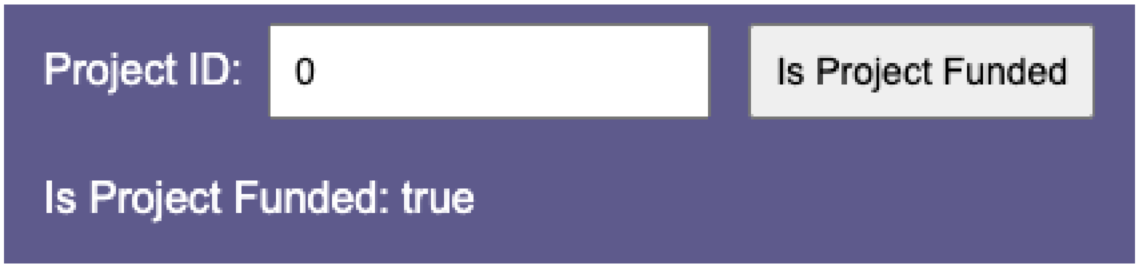
Users can see how many projects are proposed


Get Number of Funded Projects

Number of Funded Projects: 1

Users can also see how many projects are funded

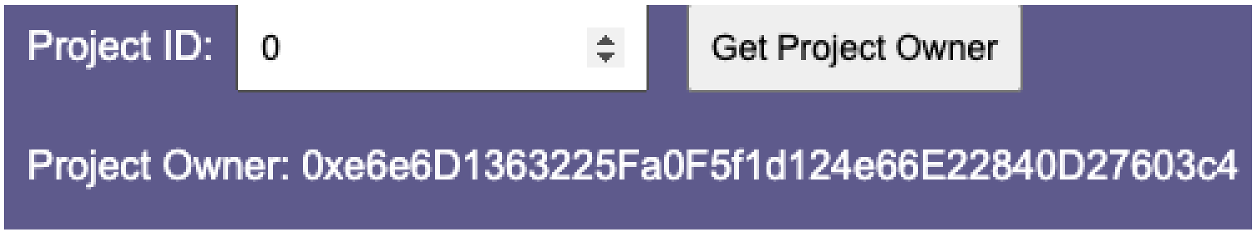
Egecan Serbester
Batuhan Çetin
Emre Türker


Project ID: 0

Is Project Funded

Is Project Funded: true

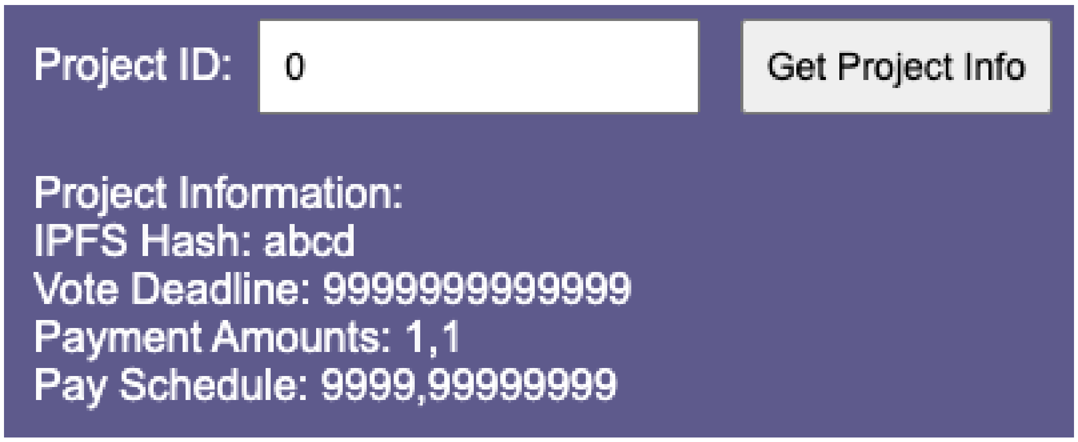
Users can see whether the project is funded


Project ID: 0

Get Project Owner

Project Owner: 0xe6e6D1363225Fa0F5f1d124e66E22840D27603c4

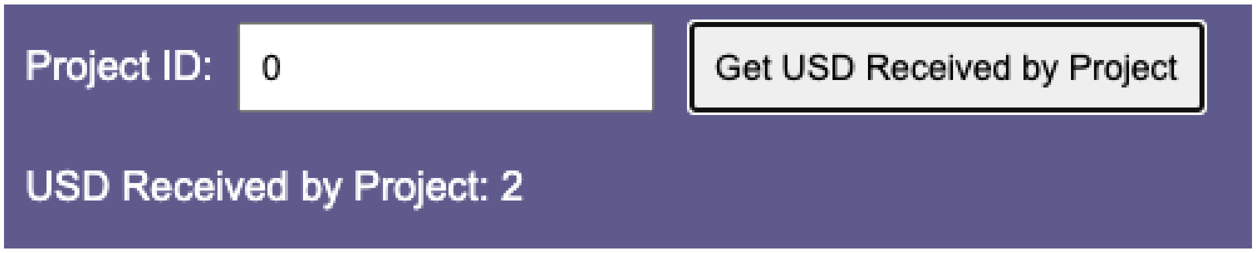
Users can see whether the owner of the project.


Project ID: 0

Get Project Info

Project Information:
IPFS Hash: abcd
Vote Deadline: 9999999999999
Payment Amounts: 1,1
Pay Schedule: 9999,99999999

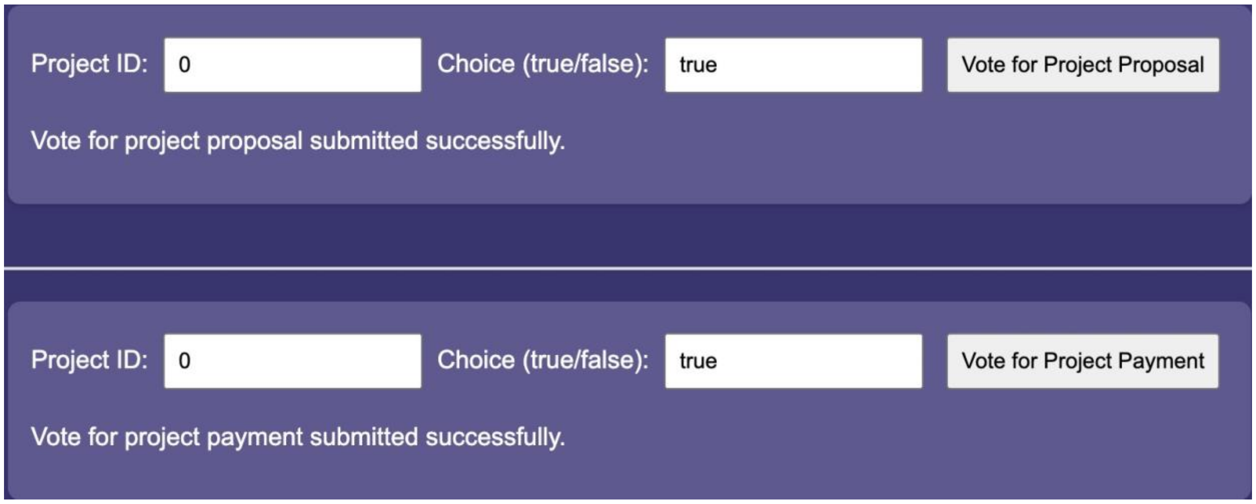
Users can see the information about the project by giving its ID.


Project ID: 0

Get USD Received by Project

USD Received by Project: 2

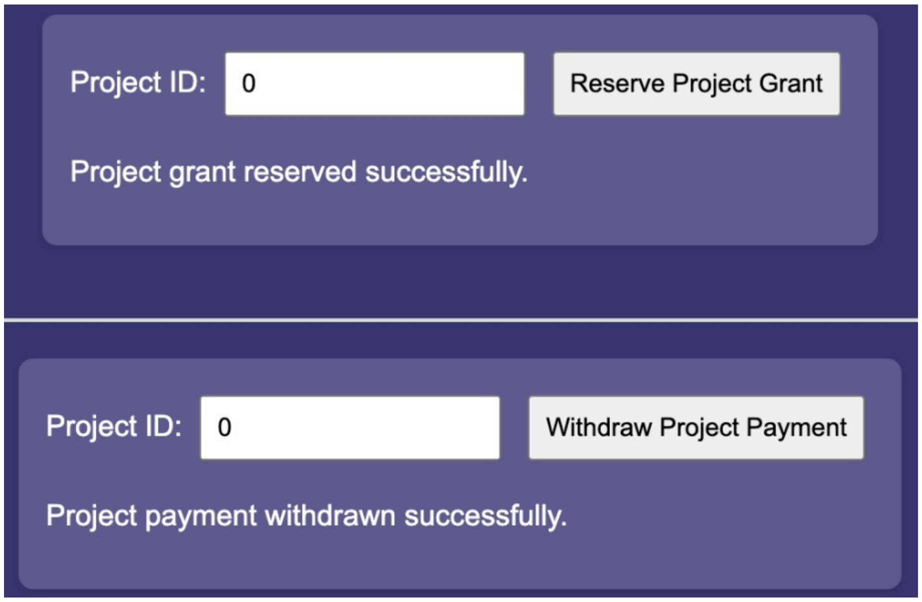
Users can see how much USD received by Project



Vote for project proposal submitted successfully.

Vote for project payment submitted successfully.

Users can vote fore project proposal and payment.



Project grant reserved successfully.

Project payment withdrawn successfully.

Users can reserve project grant and withdrawn the payment.