**Crowd Funding Smart Contract Release Two Deployment Document**

Soumil Vavikar

Purdue University Global

IN532 – Blockchain Application Development (dApps)

Dr. David Ostrowski

November 1, 2024

**Crowd Funding Smart Contract Release Two Deployment Document**

This document contains the development, testing, and deployment related documentation for the enhancements done to the “crowd funding” smart contract developed as part of the Unit 6 assignment.

**Major Enhancements**

A crowdfunding contract has been enhanced and the following functionality has been added:

|  |  |
| --- | --- |
| **Function Name** | **Business Functionality Description** |
| updateCrowdFundingEndTime | This function will update the crowdfunding’s end time. |
| getHighestContributor | The function will return the details of the highest contributor to the crowd funding. |
| closeCrowdFunding | This function will close the crowdfunding. |
| isOpen | This function will return true/false based on whether the crowdfunding is open or not. |

**Smart Contract and Supporting Files**

This smart contract is written in Solidity, and the "Hardhat" framework is used to develop, test, and deploy the smart contract.

A GitHub repository (<https://github.com/soumilvavikar/hardhat-crowd-funding>) has been created, and the smart contract, supporting files (test files, interaction files, and library files), test evidence, and commands required to start the local chain, deploy the smart contract, and test the smart contract have been pushed to the repository.

The entire workspace for the crowdfunding contract has also been submitted with this deployment document.

**Smart Contract Code**

To make crowdfunding smart contract readable and maintainable, two libraries have been created. The sol files for the libraries are attached with the smart contract code.

 

**Deployment and Execution Commands**

The deployment and execution commands remain the same as mentioned in the deployment document submitted as part of the Unit 6 assignment.

**Test Evidence(s)**

Extensive testing (which includes unit testing as well as end to end functional testing) of the developed smart contract has been done. The README.md for the project contains quick links to the “.md” files containing the test evidence (logs and screenshots) captured within the workspace under the “testevidences” folder. The test evidence can also be found below.

**Unit Test Evidence**

A total of nine tests have been added to assert the added features to the contract work as intended. The unit test file is attached below along with the unit testing evidence text file which contains the command, and the logs generated from the unit test run.

 

The unit test readme file “README\_UNIT\_TESTING.md” is present in the “testevidences” folder of the workspace.

A screenshot of a computer

Description automatically generated

**Deployment Test Evidence**

There has been no change done to the deployment process/steps from the process setup during the smart contract development work done during the Unit 6 assignment. The deployment of the crowdfunding smart contract has been done using “ignition module”. The file containing the module code required for deployment and the test evidence of successful local chain startup and deployment are attached below.



The deployment test evidence readme file “README\_E2E\_TEST\_EVIDENCES.md” is present in the “testevidences” folder of the workspace.

A screen shot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

**End to End Test Evidence**

The end-to-end interactions have been updated to test the newly added functions. The end-to-end test interactions file is attached below along with the end-to-end test evidence text file which contains the logs generated from the end-to-end test run.

 

The end-to-end test readme file “README\_E2E\_TEST\_EVIDENCES.md” is present in the “testevidences” folder of the workspace.

A screenshot of a computer

Description automatically generated