UDACITY

‹ Back to Blockchain Developer

Decentralized Star Notary

REVIEW

CODE REVIEW 3

HISTORY

▸ index.html 2

▾ README.md 1

```

# Decentralized Star Notary
```

SUGGESTION

This is an excellent video on how to write even better READMEs https://www.youtube.coı

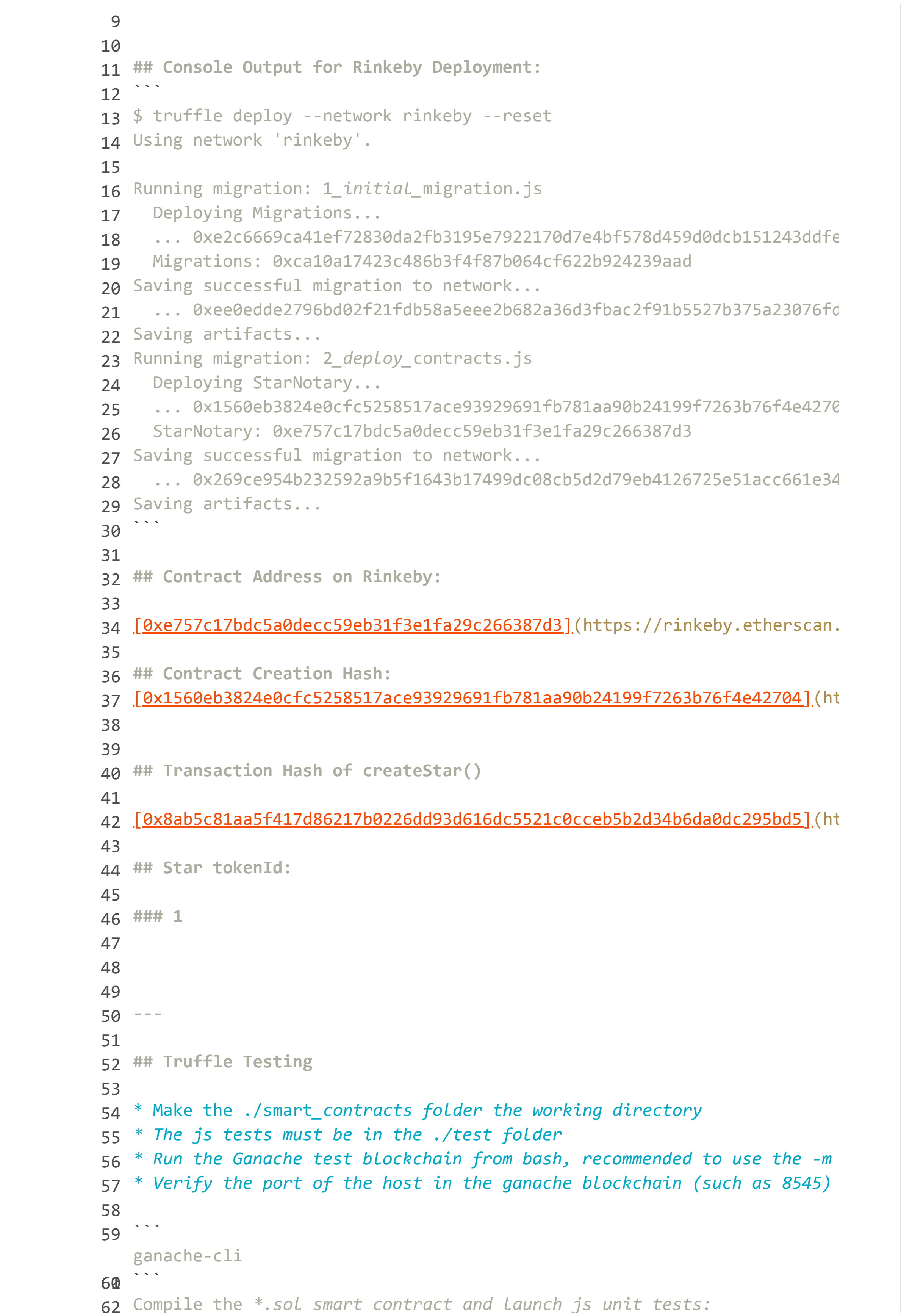
Also, for a professional looking project this is a good template https://gist.github.com/Pu

```

This is a class project to implement a blockchain-based notarization se

The project is implemented on the Ethereum blockchain under the Rinkeby

---
```



~~~

Console Output for Rinkeby Deployment:
```
$ truffle deploy --network rinkeby --reset
Using network 'rinkeby'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0xe2c6669ca41ef72830da2fb3195e7922170d7e4bf578d459d0dcb151243ddfe
  Migrations: 0xca10a17423c486b3f4f87b064cf622b924239aad
Saving successful migration to network...
  ... 0xee0edde2796bd02f21fdb58a5eee2b682a36d3fbac2f91b5527b375a23076fd
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying StarNotary...
  ... 0x1560eb3824e0cfc5258517ace93929691fb781aa90b24199f7263b76f4e4270
  StarNotary: 0xe757c17bdc5a0decc59eb31f3e1fa29c266387d3
Saving successful migration to network...
  ... 0x269ce954b232592a9b5f1643b17499dc08cb5d2d79eb4126725e51acc661e34
Saving artifacts...
```

Contract Address on Rinkeby:

[0xe757c17bdc5a0decc59eb31f3e1fa29c266387d3](https://rinkeby.etherscan.

Contract Creation Hash:
[0x1560eb3824e0cfc5258517ace93929691fb781aa90b24199f7263b76f4e42704](ht

Transaction Hash of createStar()

[0x8ab5c81aa5f417d86217b0226dd93d616dc5521c0cceb5b2d34b6da0dc295bd5](ht

Star tokenId:

1

Truffle Testing

* Make the ./smart_contracts folder the working directory
* The js tests must be in the ./test folder
* Run the Ganache test blockchain from bash, recommended to use the -m
* Verify the port of the host in the ganache blockchain (such as 8545)

```
ganache-cli
```
Compile the *.sol smart contract and launch js unit tests:
~~~

```
```
truffle test
```

Requirements: <br>
* node.js - javascript runtime
* IDE  - such as VS code with the Solidty extension by Juan Blanco
* Truffle - smart contract test environment
* Ganache - local ethereum blockchain for testing
* OpenZeppelin - standard set of tools for Solidity
* web3.js
* Infura - full node service

## Preparing Truffle

Install truffle:
```
npm install --global truffle
```
Navigate to smart_contracts folder and if empty, then:
```
truffle init
```
If on windows, delete the newly created truffle.js file in the smart_cc

## Installing OpenZeppelin
```
npm init -y
npm install --save-exact openzeppelin-solidity@2.0.0-rc.1
```

## Installing Ganache CLI
```
npm install -g ganache-cli
```

## Installing web3.js
```
npm install web3 --save
```


---

## Credits

[Hackernoon Article on Deployment](https://hackernoon.com/ethereum-deve
```

▸ style.css

▸ smart_contracts/truffle-config.js

▸ smart_contracts/test/StarNotaryTest.js

▸ smart_contracts/migrations/2_deploy_contracts.js

▸ smart_contracts/migrations/1_initial_migration.js

▸ notes.txt

RETURN TO PATH

Rate this review