DAPP : Reviewing using Blockchain.

Because   Now a days BigShots can buy their Reviews. They can bribe the Reviewing companies to put on great reviews, Jaw-Dropping ratings for their Movie or Product or anything. But what if we create a Decentralized Application to store reviews securely and no one can control them, alter them. Thus giving a fair change for people with real potential, real products to grow. This is a two way benefits road. Common people can access things much better using this and also Developers can get fair credits.

YOU CAN   Write Reviews, like others reviews, search for a review by typing a review if there are many and you are lazy to scroll through them, find the owner of that review and its unique ID(address, Id of Review for reference) , Remove review if you are the owner, count the total number of reviews you got overall.

1)Node Package Manager (NPM)

The first dependency we need is Node Package Manager, or NPM, which comes with Node.js. You can see if you have node already installed by going to your termial and typing:

```
$ node -v
```

2)Truffle Framework (Not Required, Optional)

The next dependency is the Truffle Framework, which allows us to build decentralized applications on the Ethereum blockchain. It provides a suite of tools that allow us to write smart contacts with the Solidity programming language. It also enables us to test our smart contracts and deploy them to the blockchain. It also gives us a place to develop our client-side application.
You can install Truffle with NPM by in your command line like this:

```
$ npm install -g truffle
```

3)Ganache

The next dependency is Ganache, a local in-memory blockchain. You can install Ganache by downloading it from the Truffle Framework website. It will give us 10 external accounts with addresses on our local Ethereum blockchain. Each account is preloaded with 100 fake ether. Once downloaded you can open start Ganache by clicking the icon. Ganache will be up and running and you can see that the initial block is 0 in the beginning. By default Ganache will be using the port : 7545

4)Access Remix IDE :

 Remix IDE, link will forward you to the IDE. Here you can write smartcontracts complie and deploy to ganache by changing the environment in the Remix IDE to Web3 Provider.

5)Steps:

- Create a Folder called, lets say RatingDAPP. Create a index.html and main.css files in that.

- Node -v   if this does not show the version install node package manager.
- Open remix ide and copy paste the "rating.sol" contract into it. Hit start to compile. Once it is complied without any errors you can go to run tab.

- Now before we can deploy the contract lets open Ganache by clicking the ganache Icon.

- Once the Ganache is up and running you can see 10 accounts with 100 fake ethers each. You can also see that the initial block number is 0.

- Then go to remix Ide and change the Environment in the Run tab to Web Provider.

- Hit ok for connecting to the ethereum node.

- Again hit Ok if the Web3 Provider, Ganache in our case is at localhost port : 7545(it by default shows the port 8545. So change it to 7545 and hit ok)

- Now Select the Contract you want to deploy which is reviewing in our case and hit Deploy.

- By clicking deploy it will create a transaction in Ganache which can be seen in the transactions tab. It contains the Transaction Hash, Contract Address, gas etc.

- We have successfully deployed our Contract to the Blockchain.

- Open index.html in that folder and then copy paste the entire code from the Review.html to this file.

- This page contains our entire front-end and the functionality for interacting with the contract from blockchain using Web3.

- For the Styling add the content of main.css file to your main.css file in RatingDAPP folder.

- Before opening the DAPP's index page first change the contract address present in index.html to the address you have recently deployed to.

- You can find this address in transaction in Ganache which was created when we deployed the contract from remix.

- Keep in mind that the ABI of the contract doesn't change as long as you don't alter the contract obviously.

- To use the DAPP in Chrome browser first install web server for chrome. Because you can directly open page from our local disk using the local file path as an address in chrome(Example - "C/ documents/RatingDapp" this wont work. Request needs to be coming from a server like - localhost:7545)
- Once you open the index.html page you can now interact using the frontend and all the transactions can be seen in the Ganache and are console logged also to debug easily and make life simpler and.

Cheers !