For description refer : <https://medium.com/@mvmurthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-1-40d2d0d807c2>

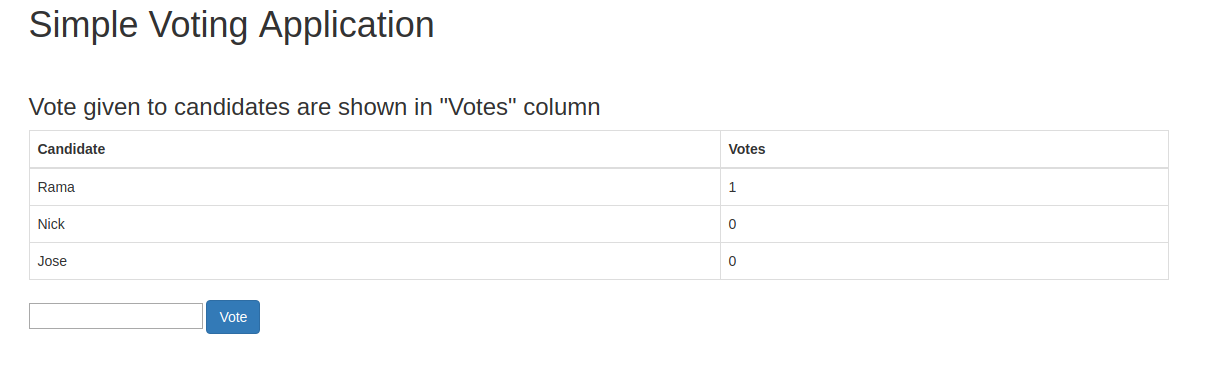
Files:

1)index.html contains the html page

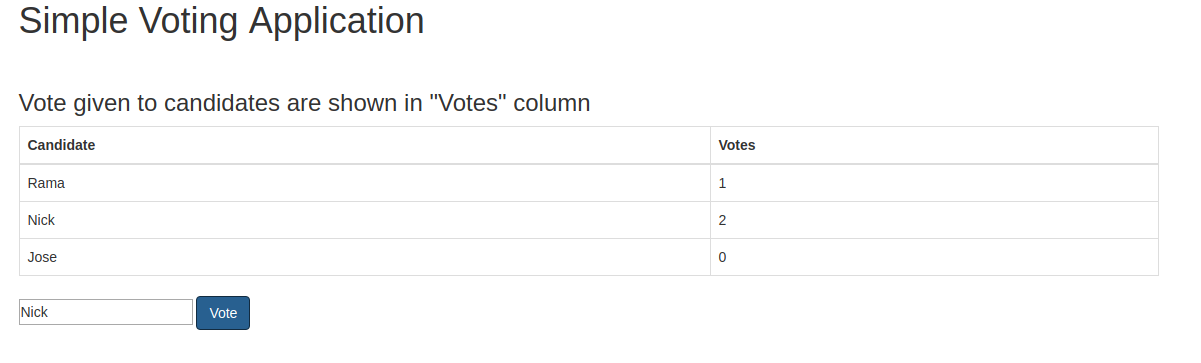
2)index.js contains the js code to connect to block chain

3)Voting.sol contains the smart contract

Screenshots



Type in the name and hit "Vote" to start voting



For error free commands follow below

1) install nvm and node :

<https://github.com/creationix/nvm>

>curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.2/install.sh | bash

Restart terminal

>command -v nvm

The above command should output nvm

>nvm install node

>npm install npm@latest -g

2) check node and npm version

>node –v

8.4+

>npm -v

5.3+

3)

>sudo apt-get install build-essential python

4) make folder

>mkdir hwv

>cd hwv

>npm install ethereumjs-testrpc [web3@0.20.1](mailto:web3@0.20.1)

>node\_modules/.bin/testrpc

You should see all test accounts

5) Open new terminal in same directory

Copy all three files > index.js,Voting.sol,index.html

6)~/hwv$sudo npm install solc

Output should be as below :

-----------------------------------------------------------------

[solc@0.4.16](mailto:solc@0.4.16) node\_modules/solc

├── [require-from-string@1.2.1](mailto:require-from-string@1.2.1)

├── [memorystream@0.3.1](mailto:memorystream@0.3.1)

├── [fs-extra@0.30.0](mailto:fs-extra@0.30.0) ([jsonfile@2.4.0](mailto:jsonfile@2.4.0), [klaw@1.3.1](mailto:klaw@1.3.1), [rimraf@2.6.1](mailto:rimraf@2.6.1))

└── [yargs@4.8.1](mailto:yargs@4.8.1) ([string-width@1.0.2](mailto:string-width@1.0.2), [os-locale@1.4.0](mailto:os-locale@1.4.0), [lodash.assign@4.2.0](mailto:lodash.assign@4.2.0), [which-module@1.0.0](mailto:which-module@1.0.0), [cliui@3.2.0](mailto:cliui@3.2.0), [window-size@0.2.0](mailto:window-size@0.2.0), [yargs-parser@2.4.1](mailto:yargs-parser@2.4.1), [read-pkg-up@1.0.1](mailto:read-pkg-up@1.0.1))

-----------------------------------------------------------------

7)start node

~/hwv$ node

> Web3=require('web3')

> web3=new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

>web3.eth.accounts

This should show all accounts

> code=fs.readFileSync('Voting.sol').toString()

> solc=require('solc')

> compiledCode=solc.compile(code)

> abiDefinition = JSON.parse(compiledCode.contracts[':Voting'].interface)  
> VotingContract = web3.eth.contract(abiDefinition)  
> byteCode = compiledCode.contracts[':Voting'].bytecode  
> deployedContract = VotingContract.new(['Rama','Nick','Jose'],{data: byteCode, from: web3.eth.accounts[0], gas: 4700000})  
**> deployedContract.address**  
> contractInstance = VotingContract.at(deployedContract.address)

8) Start voting

> contractInstance.totalVotesFor.call('Rama'){ [String: '0'] s: 1, e: 0, c: [ 0 ] }> contractInstance.voteForCandidate('Rama', {from: web3.eth.accounts[0]})

***Output*** '0xdedc7ae544c3dde74ab5a0b07422c5a51b5240603d31074f5b75c0ebc786bf53'

> contractInstance.voteForCandidate('Rama', {from: web3.eth.accounts[0]})

***Output***

'0x02c054d238038d68b65d55770fabfca592a5cf6590229ab91bbe7cd72da46de9'

> contractInstance.voteForCandidate('Rama', {from: web3.eth.accounts[0]})

***Output***

'0x3da069a09577514f2baaa11bc3015a16edf26aad28dffbcd126bde2e71f2b76f'

> contractInstance.totalVotesFor.call('Rama').toLocaleString()

***Output***

'3'

8)

Copy the address from output of **deployedContract.address** above and replace in index.js at :

contractInstance = VotingContract.at('0x2a9c1d265d06d47e8f7b00ffa987c9185aecf672');

9)

Open index.html in browser and you can see voting