We'll start by importing web3 into our main App.js component like this:

**import** Web3 **from** 'web3'

create a new function that will get called whenever our React component is loaded. It initiates web3.

**async** **componentWillMount**() { //lifecycle function of react

**await** **this**.loadWeb3() //custom function call

}

create the loadWeb3() function: //it detects the presence of etherium provider like metamask

**async** loadWeb3() {

**if** (window.ethereum) {

window.web3 = **new** Web3(window.ethereum)

**await** window.ethereum.enable()

}

**else** **if** (window.web3) {

window.web3 = **new** Web3(window.web3.currentProvider)

}

**else** {

window.alert('Non-Ethereum browser detected. You should consider trying MetaMask!')

}

}

create a function that loads data from the blockchain.

**import** Marketplace **from** '../abis/Marketplace.json'

**async** loadBlockchainData() {

**const** web3 = window.web3

// Load account

**const** accounts = **await** web3.eth.getAccounts()

**this**.setState({ account: accounts[0] })

**const** networkId = **await** web3.eth.net.getId() // know which network is connected

**const** networkData = Marketplace.networks[networkId]

**if**(networkData) {

**const** marketplace = web3.eth.Contract(Marketplace.abi, networkData.address)

// load contract from blockchain

**this**.setState({ marketplace })

**const** productCount = **await** marketplace.methods.productCount().call()

console.log(productCount.toString())

**this**.setState({ loading: false})

} **else** {

window.alert('Marketplace contract not deployed to detected network.')

}

}

set some default values for the state object: **constructor**(props) {

**super**(props)

**this**.state = {

account: '',

productCount: 0,

products: [],

loading: true

}

}

create a new component called Navbar.js:

**import** React, { Component } **from** 'react';

**class** **Navbar** **extends** **Component** {

render() {

**return** (

<**nav** className="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">

<**a**

className="navbar-brand col-sm-3 col-md-2 mr-0"

href="http://www.dappuniversity.com/bootcamp"

target="\_blank"

rel="noopener noreferrer"

>

Dapp University's Blockchain Marketplace

</**a**>

<**ul** className="navbar-nav px-3">

<**li** className="nav-item text-nowrap d-none d-sm-none d-sm-block">

<**small** className="text-white"><**span** id="account">{this.props.account}</**span**></**small**>

</**li**>

</**ul**>

</**nav**>

);

}

}

**export** **default** Navbar;

we can use React's props object, which is available to all React components to pass the account info like  {this.props.account}

import the Navbar component at the top of App.js:

**import** Navbar **from** './Navbar'

add navbar in render like:

<**Navbar** account={this.state.account} /> //it also passes the account value to the navbar component

 function that adds the product:

createProduct(name, price) {

**this**.setState({ loading: true })

**this**.state.marketplace.methods.createProduct(name, price).send({ from: **this**.state.account })

.once('receipt', (receipt) => {

**this**.setState({ loading: false })

})

In order to call this function with the form, we must bind it to the component inside the constructor like this:

**constructor**(props) { // to pass the function to the main component of html

// ...

**this**.createProduct = **this**.createProduct.bind(**this**)

}

The render function becomes:

render() {

**return** (

<div>

<Navbar account={**this**.state.account} />

<div role="main" className="col-lg-12 d-flex">

{ **this**.state.loading

? <**div** id="loader" className="text-center"><**p** className="text-center">Loading...</**p**></**div**>

: <**Main** createProduct={this.createProduct} /> //passing the funciton

}

</**div**>

</div>

);

Using function in form:

<form event.preventDefault()

const name = this.productName .value

const price = window.web3.utils.toWei(this.productPrice.value.toString(),’Ether’)

onsubmit={(event) => { this.props.createProduct(name,price)}}>

<input ref = {{input) => { this.productName = input }}>