CST4125: Blockchain Development
Week: 23
Title: Full Stack

Dr Ian Mitchell

smerf.net
Bedfordshire,
UK

January 2023

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Office Hours

- During term time only
- When: Autumn Term: Mondays 1100-1300hrs
- Please read notifications or emails
- There are occassions that these could be arranged online, e.g., due to industrial action or inclement weather

Deadlines

				Feedback	
Description	Submission	Weight	Deadline	Formative	Summative
1. Hyperledger	MyLearning	50%	18th December 2022	LW11-12	12/01/2023
2. Ethereum	MyLearning	50%	2nd April 2023	LW23-24	24/04/2023
Resits	MyLearning	50-100%	1st July 2023	None	None
Deferals	MyLearning	50-100%	1st July 2023	None	None

Lecture Objectives

- React
- Solidity
- Bulma/CSS
- Web3

Store Contract

- uses Safe Math
- Stores Ether for a duration
- Allow user to withdraw when the duration expires

Bank Contract
State variables

Solidity Code

```
1  //SPDX-License-Identifier: MIT
2  pragma solidity ^0.7.0;
3
4  import "./safeMath.sol";
5
6  contract Bank{
7      using safeArithmetic for uint;
8      mapping (address => uint) public balances;
9      mapping (address => uint) public lockTimes;
10     uint constant  public week = 604800; //7 * 24 * 60 * 60 seconds
```


Bank Contract
Balances

Solidity Code

```
12     function getBalance() public view returns(uint){
13         return address(this).balance;
14     }
15
16     function getBalance(address _addr) public view returns(uint){
17         return balances[_addr];
18     }
```


Bank Contract
Deposit & Increase Lock Times

Solidity Code

```
20     function deposit() public payable {
21         require(msg.sender.balance > msg.value, "Insufficient Funds");
22         balances[msg.sender] += msg.value;
23         lockTimes[msg.sender] = block.timestamp.add(week);
24     }
25
26     function increaseLockTime(uint _seconds) public {
27         lockTimes[msg.sender] = lockTimes[msg.sender].add( _seconds);
28     }
```


Bank Contract
Withdraw

Solidity Code

```
30     function withdraw() public payable{
31         address payable user = payable(msg.sender);
32         require(balances[msg.sender]>0, "No funds");
33         require(lockTimes[msg.sender]<=block.timestamp, "Account Frozen");
34         (bool sent, )=user.call{value: balances[user]}("");
35         require(sent, "TX not sent");
36         balances[user]=0;
37     }
38 }
```


Quick Build
Solidity

- Make a directory, `Bank` and `cd Bank`
- `truffle init`
- Move any Solidity files to contract directory
- Update Migrations folder with associated migrations files
- `truffle compile`
- Update `truffle-config` file
- Run Network on Ganache
- `truffle migrate`
- Normally, followed by `truffle console`

Quick Build
React

- `npx create-react-app bank`
- Download any CSS and Components
- Remove default code
- Rearrange files ensuring all are included in `App.js`
- `cd bank`
- `npm start`

Display Owner & Balance
Imports

React Code

```
0 import { useState} from 'react'
1 import './css/bulma2.css'
2 import Header from './Header.js'
3 import Row from './Row.js'
4 import Web3 from 'web3'
```


Display Owner & Balance
useState

React Code

```
 8 function App() {
 9   const [account, setAccount] = useState()
10   const [balance, setBalance] = useState()
11   const web3 = new Web3(Web3.givenProvider || 'http://localhost:7545')
12
13   const displayAccount = ( async () => {
14     let a = await web3.eth.getCoinbase()
15     setAccount(a)
16   })
17
18   const displayBalance = ( async () =>{
19     let b = await web3.eth.getBalance(account)
20     setBalance(b)
21   })
```


Display Owner & Balance
App

React Code

```
23   return (
24     <>
25     <div className="block">
26       <Header />
27       <Row account={account} balance={balance} />
28    </div>
29    <div className="block">
30       <button className="button is-warning" onClick={displayAccount}> Display
       owner</button>
31       <button className="button is-danger" onClick={displayBalance}> Display
       Balance</button>
32    </div>
33    </>
34   );
35 }
36
37 export default App;
```


Display Owner & Balance
Row

React Code

```
23         </>
24     )
25 }
26 export default Row;
```


Interaction with Blockchain

- Medium by Paul Bouchon
- Web3
- `call` - read
- `send` - write
  - from
  - to
  - gas
  - gasPrice
  - value
  - data
  - nonce
- Call Increment and Decrement on Blockchain
- Display results on front-end webpage

Setup
irumgier

- init
- react
- update
- migrate
- ganache
- import
- environment
- run

UI

React Code

```
46
47  return (
48    <>
49    <div className="block">
50      <Header />
51   </div>
52
53   <div className="card has-background-light is-medium">
54      <div className='card-content'>
55        <div className='title'> Value of X</div>
56        <div className='content is-large'>
57          <p>x= {x}</p>
58        </div>
59      </div>
60   </div>
61
62   <div className="block">
63    <div className='content'>
64      <button className="button is-primary" onClick={incrementHandler}> Increment
        +</button>
65      <button className="button is-danger" onClick={decrementHandler}> Decrement
       -</button>
66    </div>
```


State

React Code

```
8   const [x, setX] = useState()
9   const [account, setAccount] = useState()
10  const [deinc, setDeinc] = useState()
```


useEffect

React Code

```
12  useEffect (() => {
13    async function load() {
14    const web3 = new Web3(Web3.givenProvider || 'http://localhost:7545')
15    const network = await web3.eth.net.getNetworkType()
16    console.log("network", network)
17    const acc = await web3.eth.requestAccounts()
18    setAccount(acc[0])
19    console.log("sender:", account)
20    const SC = new web3.eth.Contract(CONTRACT_ABI, CONTRACT_ADDRESS)
21    setDeinc(SC)
22    let x = Web3.utils.BN()
23    x = await SC.methods.x().call()
24    setX(x.toString())
25    console.log('x=',x)
26    }
27    load()
28  }, [])
```


incrementHandler

React Code

```
30  const incrementHandler = (async()=>{
31    let x = Web3.utils.BN()
32    let y = await deinc.methods.increment().send({from: account})
33    x = await deinc.methods.x().call()
34    setX(x.toString())
35    console.log('x',x)
36  })
```


decrementHandler

React Code

```
38  const decrementHandler = ( async ()=>{
39    let x = Web3.utils.BN()
40    let y = await deinc.methods.decrement().send({from: account})
41    x = await deinc.methods.x().call()
42    setX(x.toString())
43    console.log('x', x)
44  })
```


Environment

React Code

```
1  import { useEffect, useState} from 'react'
2  import './css/bulma2.css'
3  import Header from './Header.js'
4  import Web3 from 'web3'
5  import {CONTRACT_ABI, CONTRACT_ADDRESS} from './environment.js'
```

Environment Code

```
1  export const CONTRACT_ADDRESS ='0xeFa348aEB2eF80855E2136Fe7bFFF16Cdb4953f9'
2  export const CONTRACT_ABI =  [
3      {
4        "inputs": [],
5        "name": "x",
```

Summary

- Difficult
- irumgier
- Connect
- effects
- state

Reading

- Ch. 5 in [1]
- Web3js
- reactjs
- Solidity

References I

[1] Santiago Palladino. *Ethereum for Web Developers*. Springer, 2020.