

Lab 1: Decentralized Market Place using Blockchain

Team:

- Saketh Varma Pericherla (sakethva@buffalo.edu)
- Prashanth Desu (pdesu@buffalo.edu)

Foreword:

- This submission also includes part3 implementation i.e. integrating web app and blockchain using web3.js. This should work with "ganache" as well as the "rinkeby" network.
- The live version of this app is at <http://blockchain-market.herokuapp.com/>
- The smart contract is present in the location "C:\Users\socket_var\Desktop\BlockchainProj1\truffle-build\contracts" named "Market.sol"
- The blockchain for the live website is hosted on a full node provided by infura on the Rinkeby network.
- Since transactions on the real networks needed to be signed by a private key we ask the user to input their account's private key and securely save the private key using encryption techniques.

The smart contract has functions grouped by the access modifiers as follows:

Chairperson:

- register() - Register a new user
- unregister() - Unregister a user

There are also additional functions like:

- addTokens() - Add more tokens to be distributed across the system.
- totalSupply() - Get total tokens in the system.
- getTokenBalance() - Get tokens remaining to be distributed across the users.

These functions might seem extraneous, but are needed to display the stats in the web application when the admin logs in.

Buyer and Seller:

- buy() - Called by a buyer to buy a product from the seller in tokens.

Note: All registered users can buy and sell.

Shared between chairperson and user:

- balanceOf() - Called by the chairperson or the owner's account to get the balance in the owner's wallet.
- addDeposit() - Called by the chairperson or the owner's account to add money to the owner's wallet.

Steps to run the project:

- The simplest way to run the project is to go to <http://blockchain-market.herokuapp.com/> and register. To run locally follow the below steps.
- Download the latest binaries for Node.js and MongoDB.
- Create a database called `bc_market_db`.
- Add a database owner account using the following snippet:

```
db.createUser({
  user: "bc_market_web",
  pwd: "Wkb6vsTHcNa",
  roles: [{ role: "dbOwner", db: "bc_market_db" }],
  passwordDigestor: "server"
});
```

- Install truffle using `npm install -g truffle` and install Ganache binary or CLI.
- Start the ganache process using CLI or open Ganache GUI, this will start the ganache private network process.
- Go to truffle-build directory and type `truffle console` to open up the truffle-ganache console.
- Type `migrate` in the ganache console to deploy smart contract to ganache.

Note 1: Make sure the address to which the contract is deployed is the same as the contract address specified in `.env` file.

Note 2: Make sure the admin address with which contract is deployed is the same as the admin address specified in `.env` file.

- Type the following code in the truffle console to get started by adding say, a 1000 tokens.

```
const contract = await Market.deployed();

contract.addTokens(1000);
```

Note: This is an optional one time operation, later when an admin is added in the database, more tokens can be added using UI.

- Go to the terminal and type `npm install`

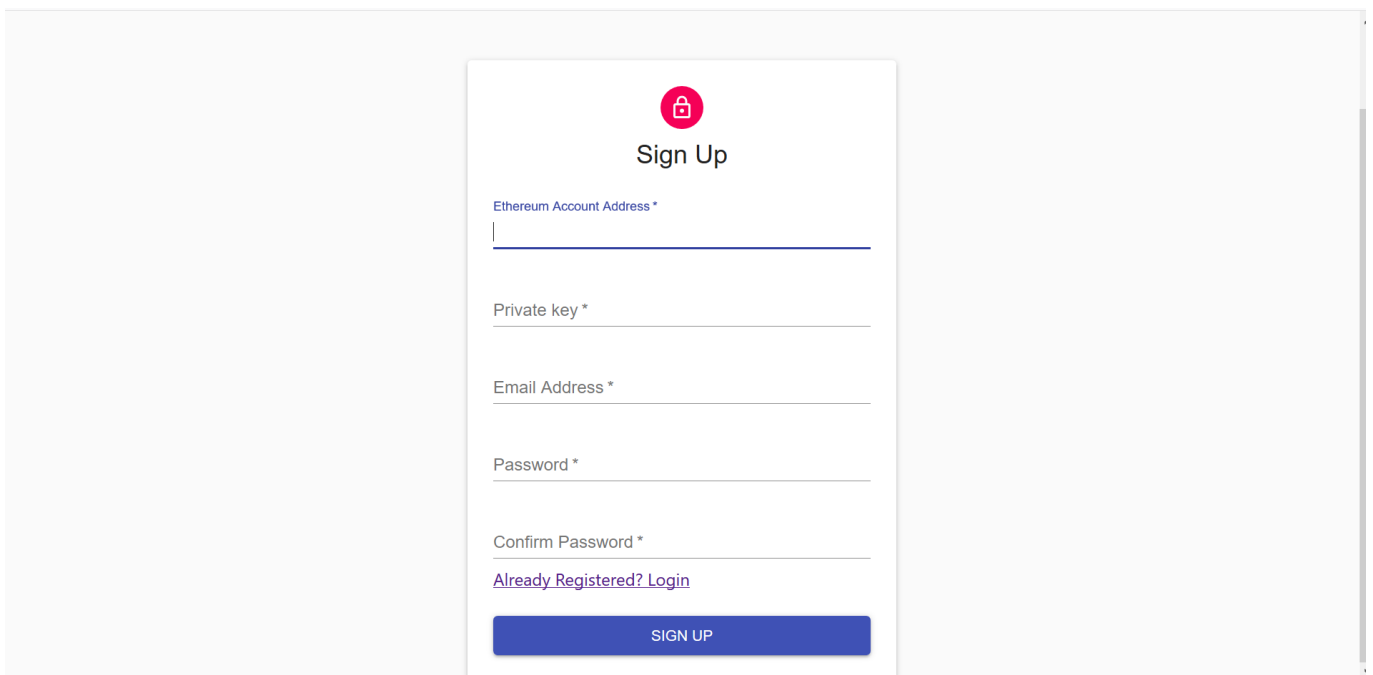
Note: `npm install web3` fails if there is no git binary. Make sure to install git before running this.

- Once all the dependencies are installed type `npm run dev-test-run`
- Open the browser and type <http://localhost:3003>
- Create an admin user entry in MongoDB. This is the account that acts as a chairperson for the website. **This is a one-time operation.** Password for the admin is "aaaaaa" for now.

```
users.insertOne({
  bcAddress: "0x629a11628711b02e350837Ca7F642140300fb1B3", // the same as
the address with which contract is deployed
  email: "sakethvarma@admin.com",
  password: "$2a$14$X04qELt13cJ/WBz.gU7GZeJF/9t/Mj27RRTfguBq17Qb10mbohCtq",
  isAdmin: "true"
});
```

Screens:

Signup Page:

A screenshot of a web application's signup page. The page has a light gray background. In the center, there is a white rectangular card with a thin gray border. At the top of the card is a red circular icon containing a white padlock. Below the icon, the text "Sign Up" is centered in a bold, black font. Underneath, there are five input fields, each with a label and an asterisk indicating it is required: "Ethereum Account Address *", "Private key *", "Email Address *", "Password *", and "Confirm Password *". Each input field has a thin blue border. Below the input fields, there is a link that says "Already Registered? Login" in a purple font. At the bottom of the card is a solid blue button with the text "SIGN UP" in white, uppercase letters.

Login Page:



Login

Email Address *

sakethbobby@gmail.com

Password *

•••••

☐ Remember me

[Create Account](#)

SIGN IN

Admin Page:

Total tokens in the network: 1400

Tokens remaining in the network: 700

[ADD TOKENS](#)

Registered Users:

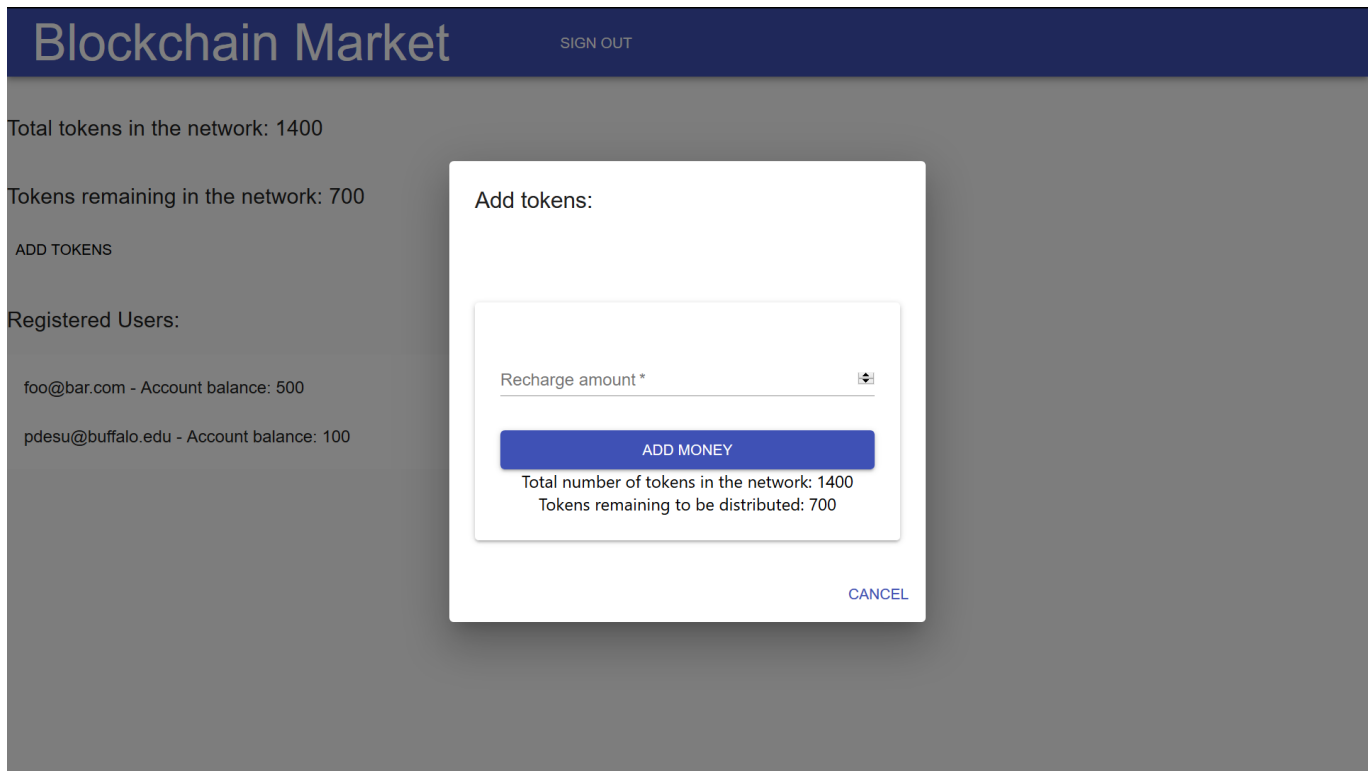
foo@bar.com - Account balance: 500

 [ADD DEPOSIT](#)

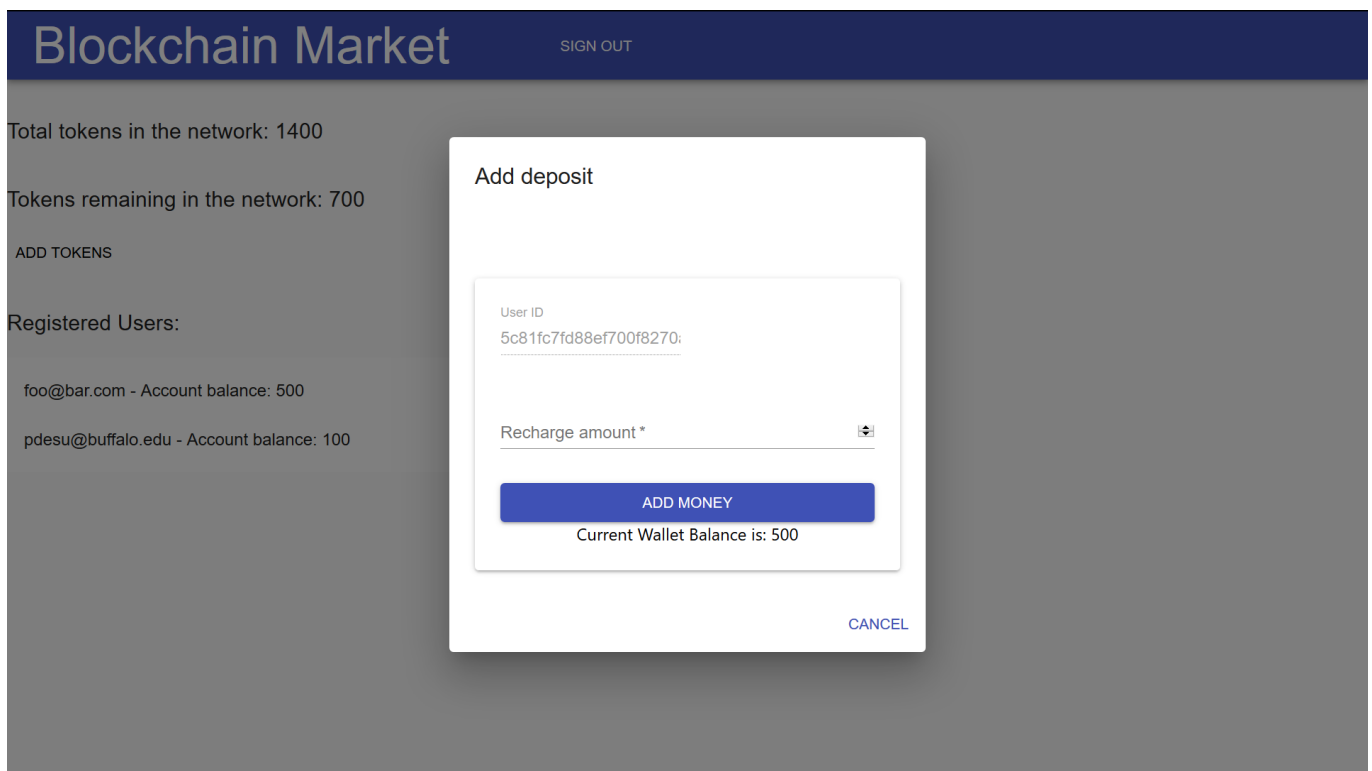
pdesu@buffalo.edu - Account balance: 100

 [ADD DEPOSIT](#)

Admin Add Tokens:



Admin Add Deposit:



User Buy Page:

Image Not Available

sneakers

Retail Price: 183

Available Units: 100

ADD TO CART

BUY

Image Not Available

guitar

Retail Price: 200

Available Units: 100

ADD TO CART

BUY

User Sell Page:

Product Name *

watch

Retail Price *

400

Number of units *

50

ADD PRODUCT

Image Not Available

sneakers

Image Not Available

guitar

User Cart Page:

Image Not Available

sneakers

Retail Price: 183

Available Units: 100

BUY

REMOVE FROM CART

User purchases page:



Purchase history



Wallet

Image Not Available

sneakers

Retail Price: 183

Available Units: 99

User wallet page:



User ID

5c81fc7fd88ef700f8270:

Confirm your password *

•••••

Recharge amount *

200



ADD MONEY

Current Wallet Balance is: 100