

README.md

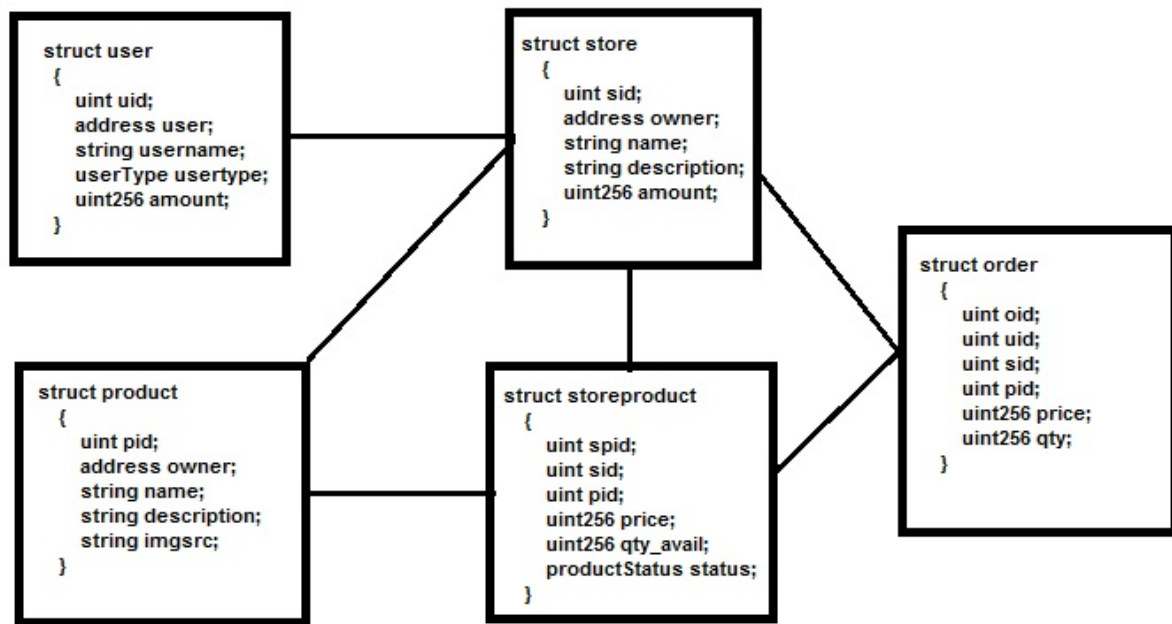
Marketplace Project was the chosen option for the FINAL PROJECT

1. Online Marketplace

Description: Create an online marketplace that operates on the blockchain

What does the Marketplace project do?

Marketplace is a distributed application initialized by an administrator which allows storeowners to create stores and add products to those stores for sale by users. When the contract is started an initial ADMIN is created. In the marketplace contract there is a bill of materials which is represented by Solidity structures which include users, stores, store products and orders. In a more real-world version the stores, the products and other data would be stored off the chain as it would create a scaling issue. The Ethereum contract would persist the Administrator, owner and shopper addresses and amount of tokens for each address in the application. That said we used Solidity to create and persist all of the contract entities. That said in the application store owners can create one or more stores. For each store, the owner can create and add products. Users (shoppers logged in to a MetaMask account) of the distributed application are initially shown a list of stores and their description. They can then choose a store and see a list of the products available in that store. Shoppers can then choose to purchase a product. When they purchase a product, an order is created the product quantity available is decreased and the shopper's account will be debited by the price of that item. The store owner's account will be credited for the price of that item.



MarketPlace Bill of Materials

How to set up Marketplace

MarketPlace runs on a local development server

MarketPlace is a truffle project.

Note: when using WINDOWS change truffle.js to truffle-config.js

The Marketplace.sol is in the /contracts directory

Truffle compile

Truffle compile will compile contracts

>truffle compile

Migration contract and migration scripts work as follows:

Use ganache-cli to start private blockchain

>ganache-cli

Ganache CLI v6.1.6 (ganache-core: 2.1.5)

Available Accounts

=====

- (0) 0xe629bec058d27a509b89f3508fe3f316dc27fde1 (~100 ETH)
- (1) 0xffb260a4b10daef122b99548d50659b0688a21da (~100 ETH)
- (2) 0xc646fd67453f2eeb5de9e6c03c6c01065be8bcab (~100 ETH)
- (3) 0x954a29005a77d19f43687b908eed5d93645baf24 (~100 ETH)
- (4) 0x7d3c85a5eba11b98e510fcc7b1adcff73c5018e5 (~100 ETH)
- (5) 0x456d84719ceabf23745e241e0ca66955183f88b8 (~100 ETH)
- (6) 0xa6ba1ba9fbb85aca1d44422f35b5f4813fd490e7 (~100 ETH)
- (7) 0xa031cac8271567992b370ed7657edc360dc0c6b4 (~100 ETH)
- (8) 0x705190cd0515c599b7e40b3837076d5bfab39229 (~100 ETH)
- (9) 0xcbec2a0415f509573822b3edac9e0c3fad905875 (~100 ETH)

Private Keys

- =====
- (0) 0xd187aca8a3e1a374345631a1efb39b7727814295eb530684c57c965a93b332e4
 - (1) 0x2f881156e1a168430280e1927cec1c318392f9d0f1c6d693c16983e43c0a444a
 - (2) 0x46eaf3c530c0e6cc9084ef017ca79e12922f1f377a9b83d9f66a6bc9ff0dd966
 - (3) 0xefc68cb70968a15edbafc6e62b44f14cf677ec2699eb2c8a811857fe45a55e08
 - (4) 0xdebded010ac20dfda57db80783bca933ca4ea9bc83cb78748d5640b63e3fb9f9
 - (5) 0xe22710e76c82ccdb2829b04a243e15c01efed5a57281532759c7001782286416
 - (6) 0xa8cb5f82fec36ea3013c03d021750961b30ce15109bf9297e8114ef5c913787a
 - (7) 0x5071f01d0768349cdd4cf9b8c4d71fdff70fac829453d51a376be84ae7f1372d
 - (8) 0x0ed0aab9f6ebb76bd819d58eb48427525493b3efddacfdccf8358931042f16da
 - (9) 0x3031933397c25d6f871827d3c6a67c392d5d761c6f63fb35ce0b58e26de92b1d

HD Wallet

=====

Mnemonic: sort today prevent fiction shove bitter lawn erupt adult type recall hold

Base HD Path: m/44'/60'/0'/0/{account_index}

Gas Price

=====

20000000000

Gas Limit

=====

6721975

Listening on 127.0.0.1:8545

Truffle migrate and deploy

will migrate contracts to a locally running ganache-cli test blockchain on port 8545

```
>truffle deploy
```

```
truffle deploy
```

```
Using network 'development'.
```

```
Running migration: 1_initial_migration.js
```

```
Deploying Migrations...
```

```
... 0x42614d56bd7067426bdf6577dcbd13772637409a0c46b7da899d4a4d35023e7
```

```
Migrations: 0x47333f4593bbf8c965d06062f492b8def897ecfd
```

```
Saving successful migration to network...
```

```
... 0x419fc13481ea1f139039003b75fc4288f812d97f8b1d801729d60222d43952ec
```

```
Saving artifacts...
```

```
Running migration: 2_deploy_contracts.js
```

```
Deploying Marketplace...
```

```
... 0x3a78f994335dfcb7734af75a18dc4b7181964f3937a86114ee6299df226efba8
```

```
MarketPlace: 0x3fc4f93d266668ed48e7289a9264aa0295e76e11
```

```
Saving successful migration to network...
```

```
... 0xd3ec3195540c371094f6f4638c2dc9885ec2f8e8bbf492e4d88d2f890d1ff29a
```

```
Saving artifacts...
```

Truffle test

All tests are in tests directory

Running truffle test will migrate contracts and run the tests

```
>truffle test
```

```
Using network 'development'.
```

```
Compiling .\contracts\MarketPlace.sol...
```

```
Compiling .\test\TestMarketPlace.sol...
```

```
Compiling openzeppelin-solidity/contracts/math/SafeMath.sol...
```

```
Compiling openzeppelin-solidity/contracts/ownership/Ownable.sol...
```

```
Compiling truffle/Assert.sol...
```

```
Compiling truffle/DeployedAddresses.sol...
```

```
TestMarketPlace
```

```
✓ testInitialUserCountUsingDeployedContract (197ms)
```

```
✓ testCreateStore (249ms)
```

```
✓ testCreateProduct (139ms)
```

✓ testCreateStoreProduct (131ms)

1) testSellProduct

> No events were emitted

2) testWithdraw

> No events were emitted

4 passing (3s)

2 failing

1) TestMarketPlace

testSellProduct:

Error: VM Exception while processing transaction: revert
at Object.InvalidResponse

2) TestMarketPlace

testWithdraw:

Error: VM Exception while processing transaction: revert
at Object.InvalidResponse

Run npm dev

>npm run dev

> marketplace@1.0.0 dev C:\Users\jbambara\Documents\GitHub\marketplace

> lite-server

**** browser-sync config ****

```
{ injectChanges: false,  
  files: [ './**/*.html,html,css,js' ],  
  watchOptions: { ignored: 'node_modules' },  
  server:  
    { baseDir: [ './src', './build/contracts' ],  
      middleware: [ [Function], [Function] ] } }
```

[Browsersync] Access URLs:

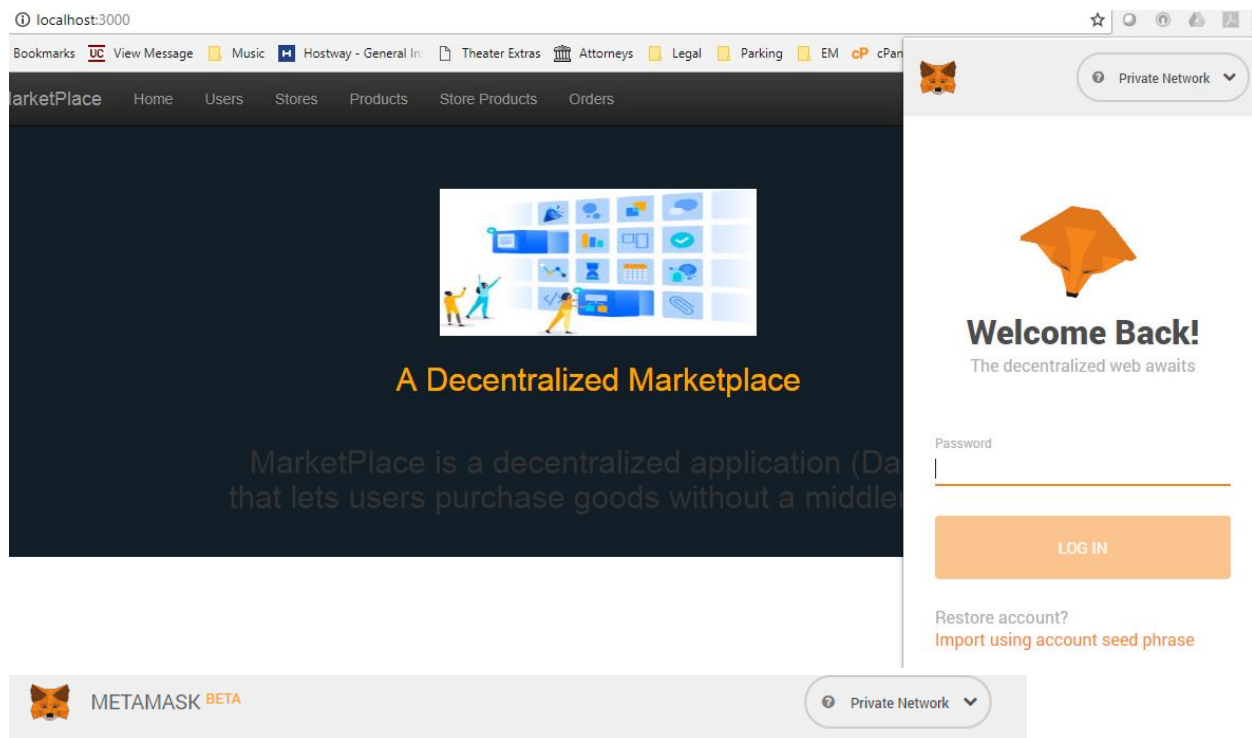
Local: http://localhost:3000

External: http://169.254.30.99:3000

UI: http://localhost:3001

UI External: http://169.254.30.99:3001

You should see this screen localhost:3000



< Back

Restore your Account with Seed Phrase

Enter your secret twelve word phrase here to restore your vault.

Wallet Seed

myself gym monitor rally suit badge simple shrug
trick asset treat draw

New Password (min 8 chars)

Confirm Password

Click on Import using account seed phrase and use the Mnemonic from ganache-cli:
sort today prevent fiction shove bitter lawn erupt adult type recall hold

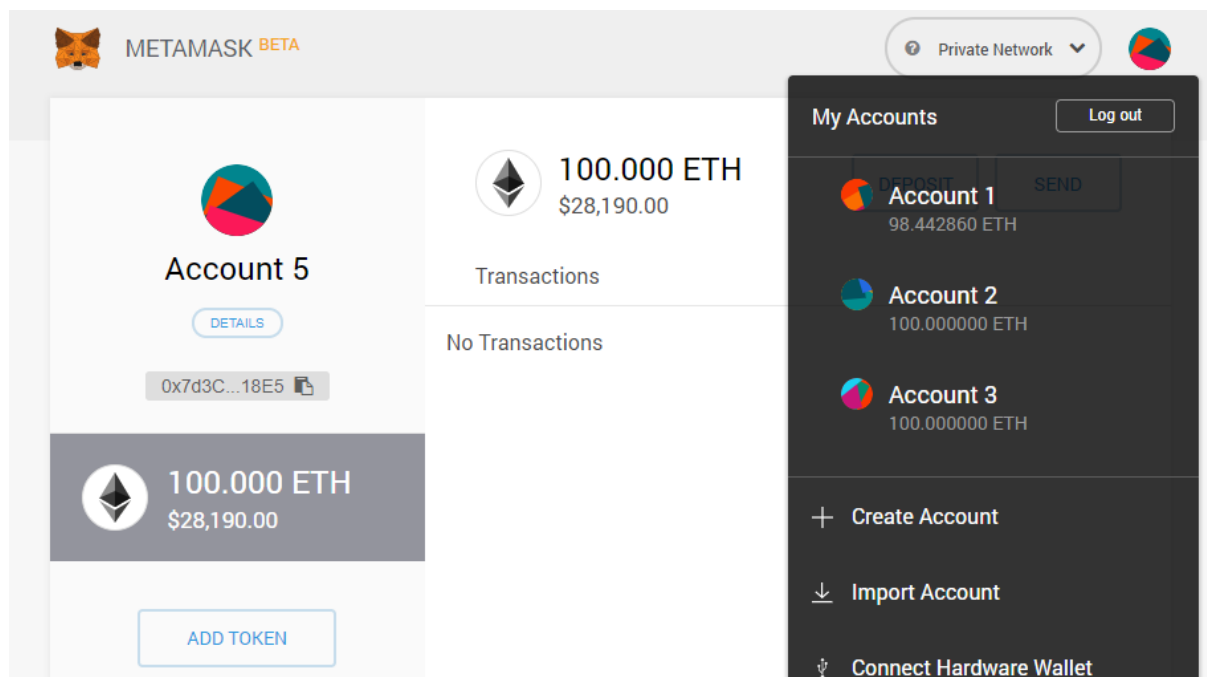
Available Accounts

=====

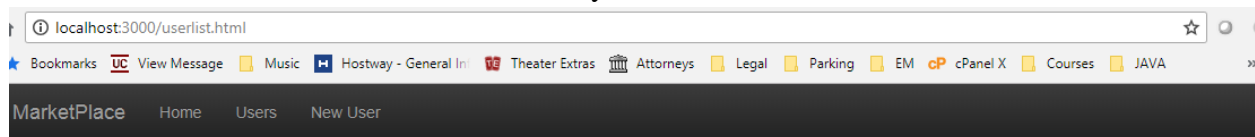
- (0) 0xe629bec058d27a509b89f3508fe3f316dc27fde1 (~100 ETH)
- (1) 0xffb260a4b10daef122b99548d50659b0688a21da (~100 ETH)
- (2) 0xc646fd67453f2eeb5de9e6c03c6c01065be8bcab (~100 ETH)

(3) 0x954a29005a77d19f43687b908eed5d93645baf24 (~100 ETH)

(4) 0x7d3c85a5eba11b98e510fcc7b1adcf73c5018e5 (~100 ETH)



Click on Users to see ADMIN user created by MarketPlace contract constructor



Current Users

Userid	UserName	UserType	UserAddress	UserAmount
0	Admin	Admin	0xe629bec058d27a509b89f3508fe3f316dc27fde1	0

Status

What does this page do?

This page shows the current users.

Network Status

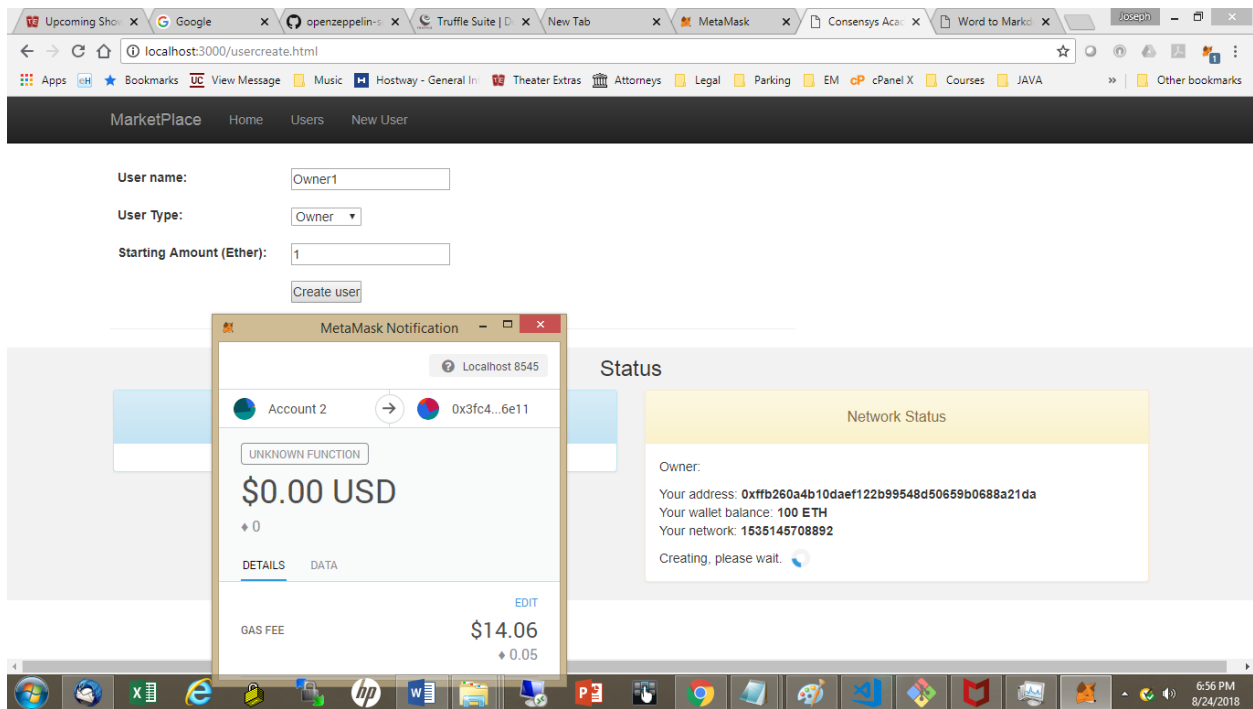
Owner:

Your address: 0x7d3c85a5eba11b98e510fcc7b1adcf73c5018e5

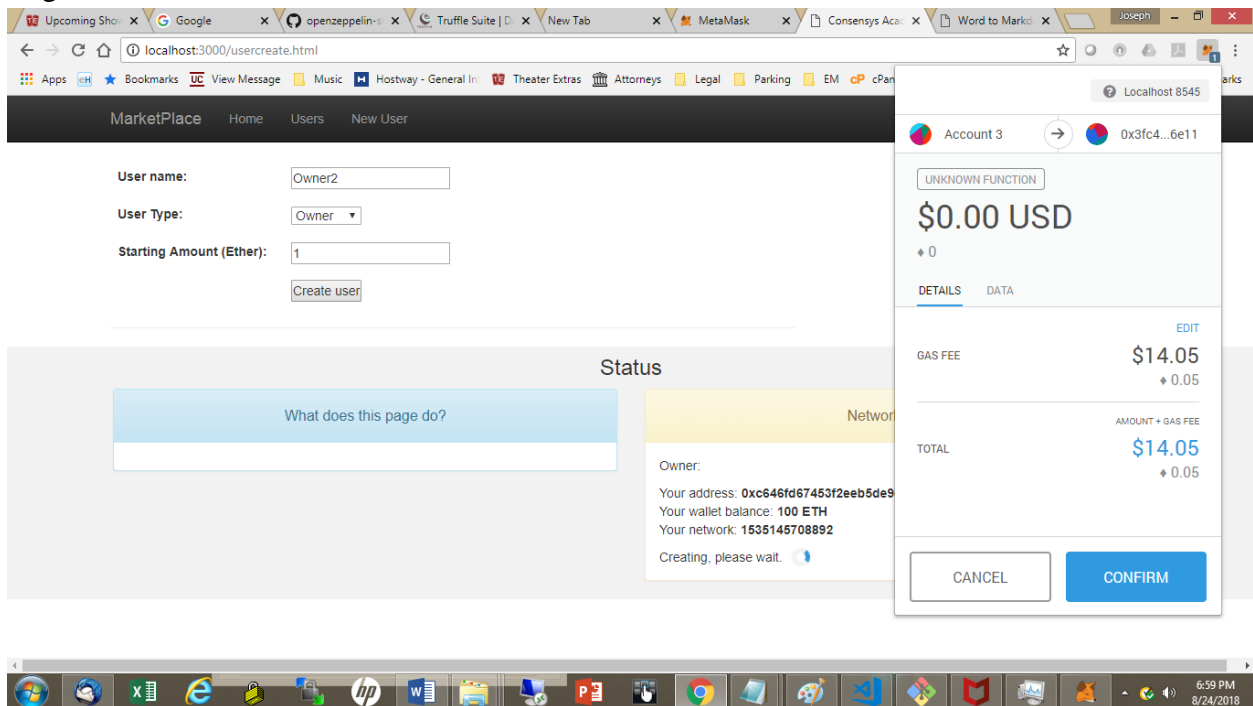
Your wallet balance: 100 ETH

Your network: 1535145708892

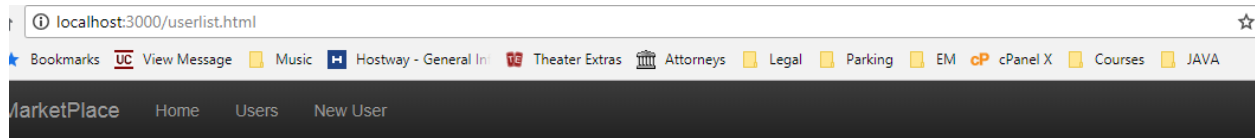
Logon as Account2



Logon as Account 3



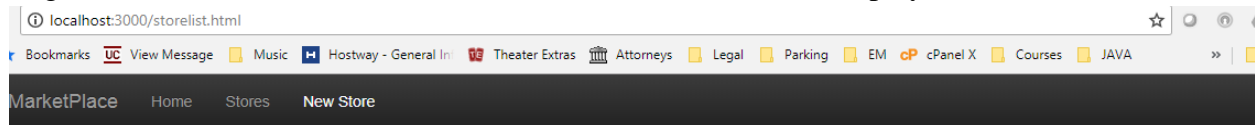
Display the Users



Current Users

Userid	UserName	UserType	UserAddress		UserAmount
0	Admin	Admin	0xe629bec058d27a509b89f3508fe3f316dc27fde1	Account1	0
1	Owner1	Owner	0xffb260a4b10daef122b99548d50659b0688a21da	Account2	10000000000000000000
3	Owner2	Owner	0xc646fd67453f2eeb5de9e6c03c6c01065be8bcab	Account3	10000000000000000000

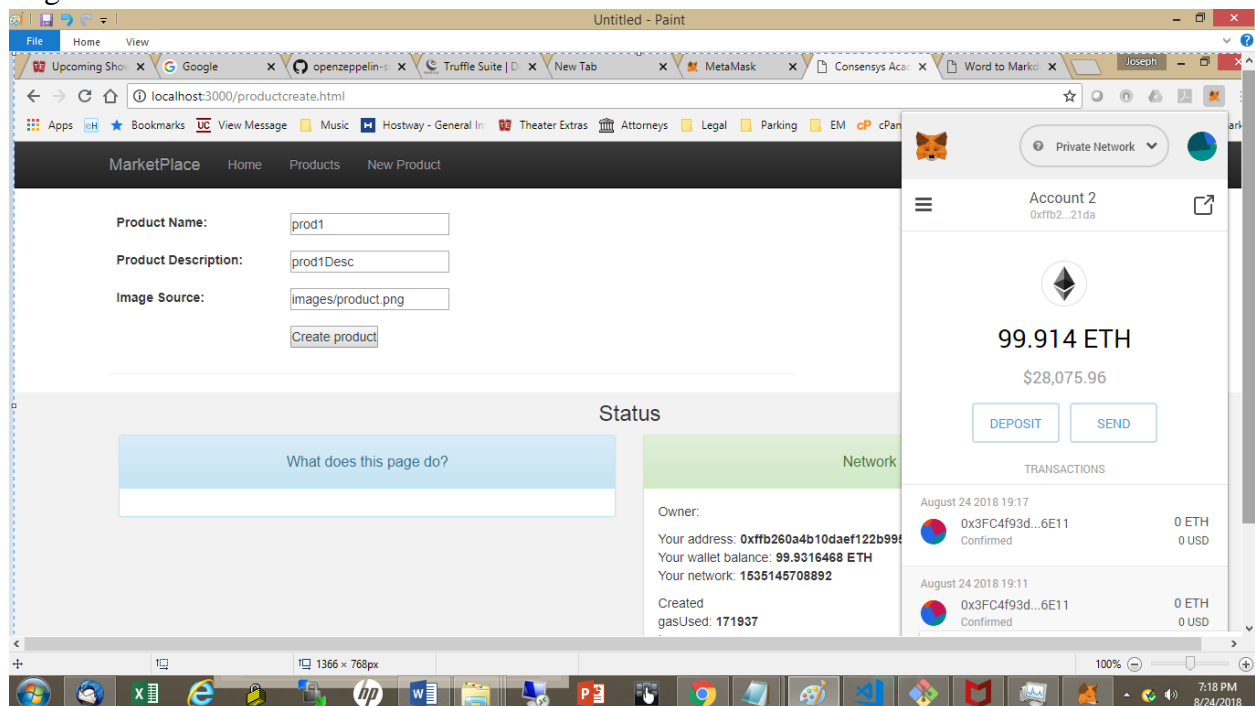
Logon as each Owner (switch MetaMask Account) and Create/Display Stores



Current Stores

StoreId	StoreName	StoreDescription	StoreAddress	StoreAmount
0	store1	store1Desc	0xc646fd67453f2eeb5de9e6c03c6c01065be8bcab	10000000000000000000
2	store2	store2Desc	0xffb260a4b10daef122b99548d50659b0688a21da	10000000000000000000

Logon as Owner and create a Product



List the Products

localhost:3000/productlist.html

☆ Bookmarks View Message Music Hostway - General Info Theater Extras Attorneys Legal Parking EM cPanel X Courses JAVA

MarketPlace Home Products New Product

Current Products

Id	Name	Description	ImgSrc	Address
0	prod1	prod1Desc	images/product.png	0xffb260a4b10daef122b99548d50659b0688a21da
1	prod2	prod2Desc	images/product.png	0xffb260a4b10daef122b99548d50659b0688a21da

What does this page do?

This page shows the current products.

Network Status

Owner:
Your address: 0xffb260a4b10daef122b99548d50659b0688a21da
Your wallet balance: 99.8972594 ETH
Your network: 1535145708892

Create StoreProducts

localhost:3000/storeproductlist.html

☆ JAVA »

MarketPlace Home

Id	Name	Description
1	store1	store1
2	store2	store2Desc

Choose a Store

Spid	Sid	Pid	Name	Description	ImgSrc	Price	Qty	Status
No products found in store								

C. Your Products

Id	Name	Description	ImgSrc
0	prod1	prod1Desc	images/product.png
1	prod2	prod2Desc	images/product.png

Choose a Product

D. Store Product

Add a Store Product

Product Id: 0

Name: prod1

Description: prod1Desc

Image Source: images/product.png

Price: 1

Quantity Available: 4

Status: Available

Create Store Product

Display Store Products

localhost:3000/storeproductlist.html
☆

MarketPlace
Home

A. Your Stores

Id	Name	Description
1	store1	store1
2	store2	store2Desc

Products in Store1

B. Products in Store

Spid	Sid	Pid	Name	Description	ImgSrc	Price	Qty	Status
0	1	0	prod1	prod1Desc	images/product.png	1	4	0
1	1	1	prod2	prod2Desc	images/product.png	2	7	0

C. Your Products

Id	Name	Description	ImgSrc
0	prod1	prod1Desc	images/product.png
1	prod2	prod2Desc	images/product.png

D. Store Product

Product Id:

Name:

Description:

Image Source:

Price:

Quantity Available:

Status:

Now lets Shop

localhost:3000/shopping.html

MarketPlace

Pick a Store

store1

shop

Desc: store1
Enter Store

store2

shop

Desc: store2Desc
Enter Store


Buy a Product



MarketPlace

Leave Store


prod1



Desc: prod1Desc
Price: 1
Qty: 4

Buy

prod2



Desc: prod2Desc
Price: 2
Qty: 7

Buy