A STUDY ON CRYPTO IDENTITY SYSTEM

Project Proposal for Creative Component

Abstract

This document gives brief idea on how we can implement Blockchain to secure any person identity system like SSN, so that it can be made more secure and free from identity theft. For the implementation, I have created a Ethereum test net on both Ganache and deployed a smart contract on it to verify that it works.

Dr. N. Park
(CHAIR)

Dr. Crick

Dr. Chan-tin

Shreyas Aserkar
shreyas.aserkar@okstate,edu

INDEX

1. Introduction 2

2. Why is SSN unsecured? 2

3. The Problem 3

4. Implementation 3

5. What is Smart Contract and how it is created? 3

6. Creating test-net on Ganache 6

7. Running Smart Contract on Ganache 17

8. Source Code 22

9. Conclusion 24

10. References 24

INTRODUCTION

What Is Blockchain?

A Blockchain is a decentralized database sometimes called ledger that contains continuously growing lists of records or data, called *BLOCKS*, which are linked and secured using cryptography. Each block typically contains a hash pointer as a link to a previous block, a timestamp and a transaction data.

In a typical Blockchain the entire data is spread out on a bunch of different computers that can be geographically separated, it is called peer to peer network. A typical peer to peer network is a distributed application architecture that partitions the task or workloads between peers. Peers are equally privileged, equipotent participants in the application, they are said to form a peer to peer network of nodes.

Blockchain is immutable, once the data is written to the blockchain, no one can change the contents of the block, even the administrator with highest privileges. This provides the benefits for audit, that this can be proof that the data is not tampered [6].

Why is SSN unsecured?

A Social Security number (SSN) is a nine-digit number issued to U.S. citizens, permanent residents, and temporary (working) residents [3].

The number is issued to an individual by the Social Security Administration (SSA). Its primary purpose is to track individuals for Social Security purposes. Then came the IRS, Banks and others who began using SSN for identification purpose [2].

The problem with the SSN is that it is prone to identity theft. A SSN is just s strings of digits on a paper card that did the job in 1930, which later got government approval for id purpose in 1972.

Why biometrics can’t be used to make it theft prone? A biometrics has its own security issues; also they are not fool proof as there have been many case of biometrics. In one of the referenced article it is said that “Bank of America recently started piloting a biometric system from Samsung that scans the iris to determine person identity. They’re not the only company using biometrics. Wells Fargo and British bank TSB are working on iris scanning for mobile banking as well."[1][4]

The Problem

The problem associated with SSN is that, it is vulnerable can easily be used for identity theft and since today SSN is linked with Credit cards, loans, bank account etc., so it needs better security features and management of identities so as to make them theft proof. And the best solution today to make it theft proof is to deploy a Cypto Identity system for SSN on a blockchain which will make it secure.

Implementation

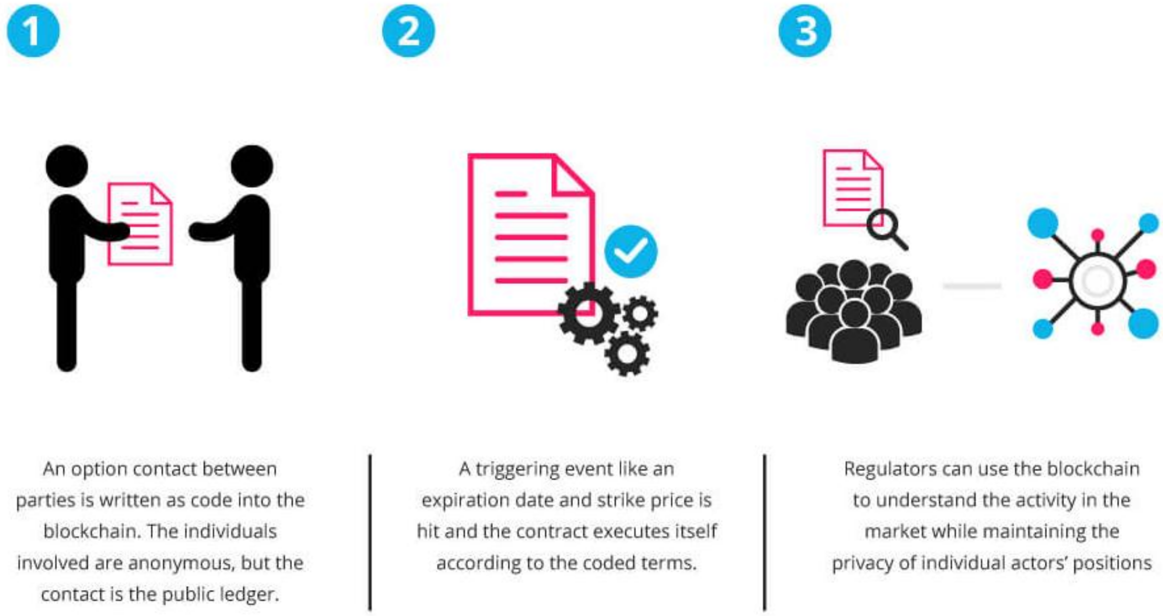
Whenever someone wants to apply for SSN, they can store the information required to apply for SSN into the functions created in the smart Contract in the form of Strings and Integer. In the implementation of the Crypto Identity System, I created a smart contract in Solidity, which basically had the functions which can store and retrieve the above mentioned Documents and this code is deployed on the test net I created on Truffle Ganache and check if an asset or currency is transferred into a program "and the program runs this code and at some point it automatically validates a condition and it automatically determines whether the asset should go to one person or back to the other person, or whether it should be immediately refunded to the person who sent it or some combination thereof.

What is smart contract and how it is created?

Smart Contract allows individual to exchange data in trusted conflict free manner without relying on the third party like bank, lawyer or a notary. It is a computer protocol intended to facilitate, verify or enforce the negotiation or performance of a contract. The most prominent smart contract implementation is the ETHEREUM BLOCKCHAIN platform.

Smart contracts help you exchange money, property, shares, or anything of value in a transparent, conflict-free way while avoiding the services of a middleman.

The best way to describe smart contracts is to compare the technology to a vending machine. Ordinarily, you would go to a lawyer or a notary, pay them, and wait while you get the document. With smart contracts, you simply drop an ethereum into the vending machine (i.e. ledger), and your escrow, driver's license, or whatever drops into your account. More so, smart contracts not only define the rules and penalties around an agreement in the same way that a traditional contract does, but also automatically enforce those obligations[6].



An option contact between parties is written as code into the blockchain. The individuals involved are anonymous, but the contact is the public ledger.

A triggering event like an expiration date and strike price is hit and the contract executes itself according to the coded terms.

Regulators can use the blockchain to understand the activity in the market while maintaining the privacy of individual actors' positions

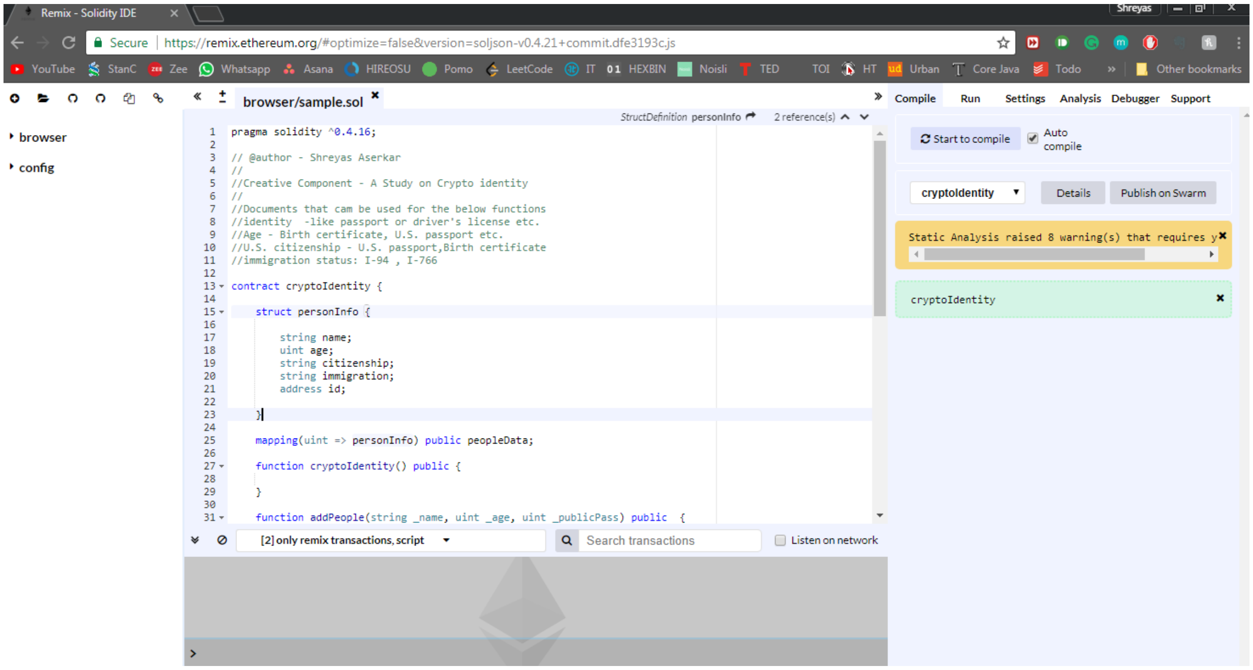
Figure 1

The picture above gives a brief idea on how a contract is working among two individuals in a trusted manner using a ethereum blockchain without the need of a third party[5].

How a Smart Contract is created?

To run on a vast decentralized network, we have to create a smart contract for Ethereum blockchain. Ethereum is special, because it's the first blockchain implementation to have a Turing Complete virtual machine built on top of it. This means that an Ethereum smart contract can (theoretically) be used to accomplish any computational task. In simpler terms, nearly any program can be run on of Ethereum.

The smart contracts are written in a language called SOLIDITY, which is a language that is similar to JavaScript. To write this language, the IDE called REMIX is provided, which is an online ide, that can be opened in any browser [5].



```
pragma solidity ^0.4.16;

// @author - Shreyas Aserkar
//
//Creative Component - A Study on Crypto identity
//
//Documents that cam be used for the below functions
//identity  -like passport or driver's license etc.
//Age - Birth certificate, U.S. passport etc.
//U.S. citizenship - U.S. passport,Birth certificate
//immigration status: I-94 , I-766

contract cryptoIdentity {

    struct personInfo {

        string name;
        uint age;
        string citizenship;
        string immigration;
        address id;

    }

    mapping(uint => personInfo) public peopleData;

    function cryptoIdentity() public {

    }

    function addPeople(string _name, uint _age, uint _publicPass) public  {
```

```
Static Analysis raised 8 warning(s) that requires y
```

```
cryptoIdentity
```

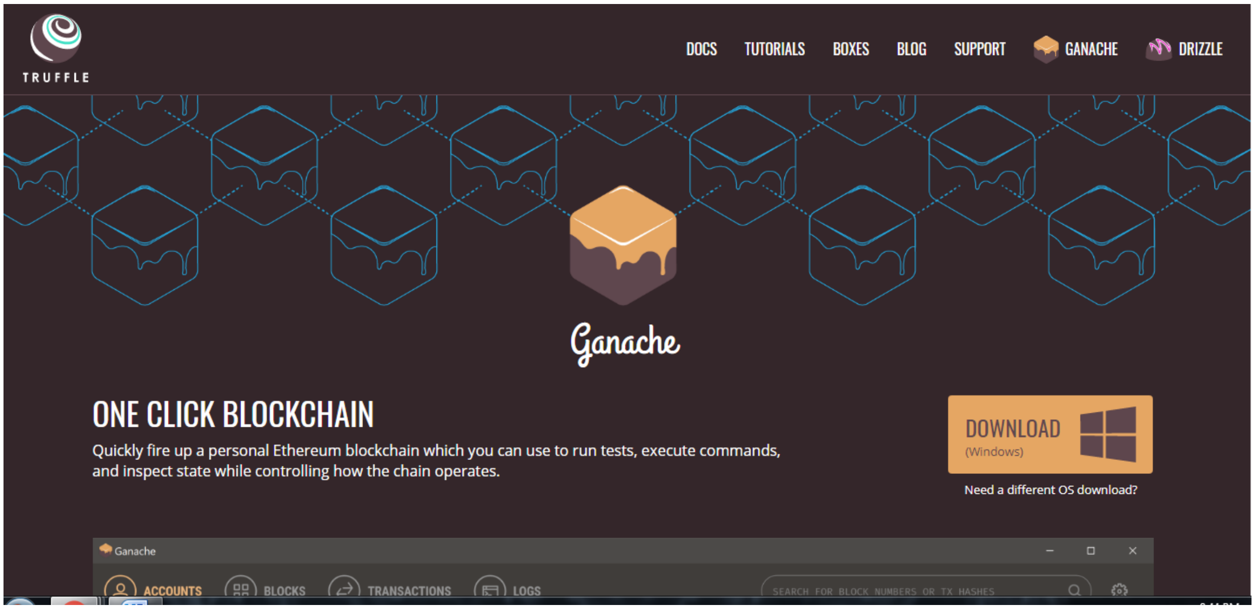
Figure 2, Remix – online complier for Solidity

The program written in Solidity has extension ".sol" as we can see it is given as "sample.sol" here in the above screen shot. The name "Sample.sol" was later change to "cryptoIdentity.sol" for the program purpose.

Creating a test net on Ganache

To proceed further, we have to follow the below steps :

1. Download "Ganache" and install it on your system. It can be downloaded from the link : http://truffleframework.com/ganache/ . It is basically a tool which we will be using to create a private blockchain that runs locally.

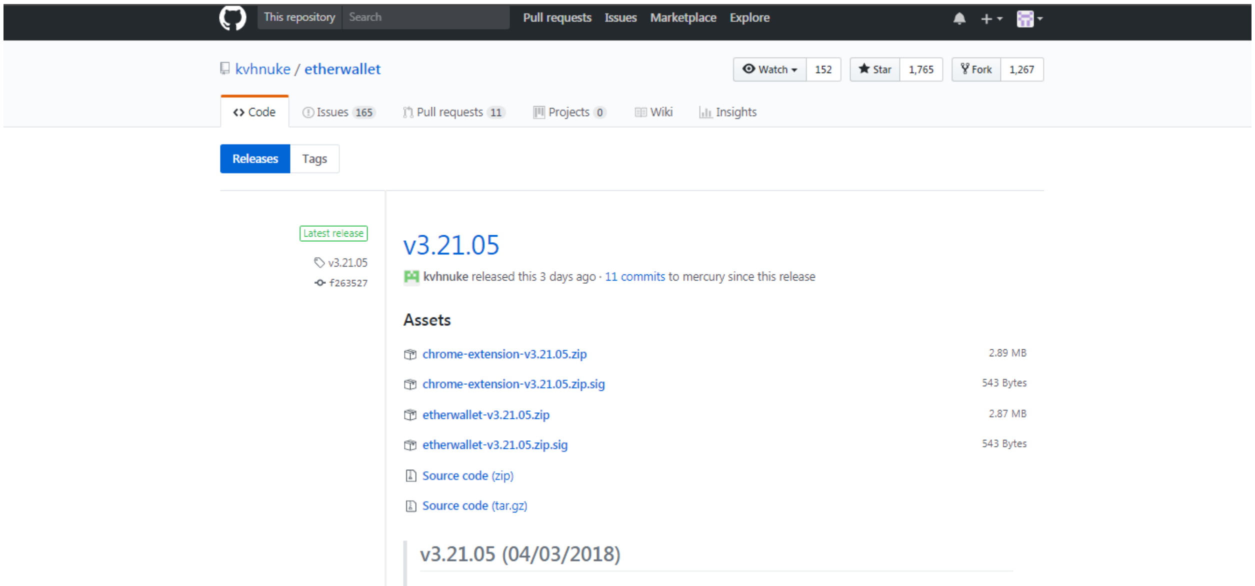


ONE CLICK BLOCKCHAIN

Quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

Figure 3

2. Now we have to download the copy of MyEthereum wallet on the system. It can be downloaded from the link : https://github.com/kvhnuke/etherwallet/releases .



v3.21.05

kvhnuke released this 3 days ago · 11 commits to mercury since this release

Assets

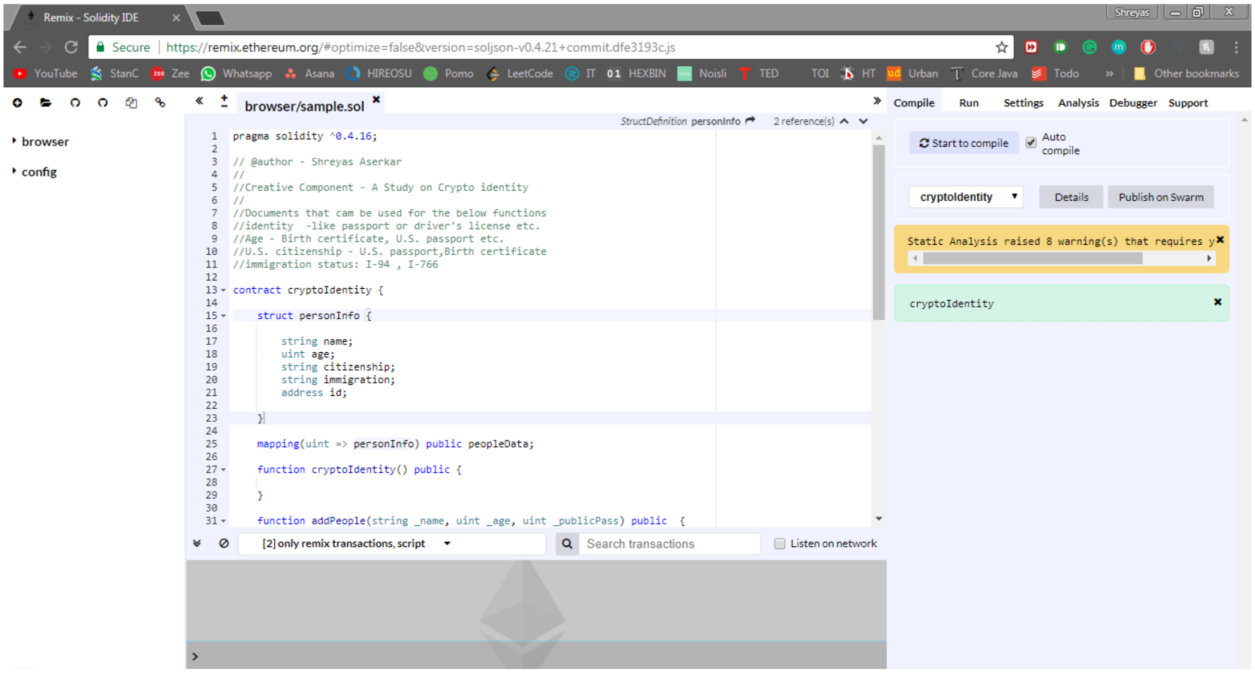
chrome-extension-v3.21.05.zip	2.89 MB
chrome-extension-v3.21.05.zip.sig	543 Bytes
etherwallet-v3.21.05.zip	2.87 MB
etherwallet-v3.21.05.zip.sig	543 Bytes
Source code (zip)	
Source code (tar.gz)	

v3.21.05 (04/03/2018)

Figure 4

3. Now that we have downloaded the Ganache and MyEthereum wallet , we will open the remix which I explained above by going to the URL: https://remix.ethereum.org

It will open the screen similar to below snapshot:



```
pragma solidity ^0.4.16;

// @author - Shreyas Aserkar
//
//Creative Component - A Study on Crypto identity
//
//Documents that cam be used for the below functions
//identity  -like passport or driver's license etc.
//Age - Birth certificate, U.S. passport etc.
//U.S. citizenship - U.S. passport,Birth certificate
//immigration status: I-94 , I-766

contract cryptoIdentity {

    struct personInfo {

        string name;
        uint age;
        string citizenship;
        string immigration;
        address id;

    }

    mapping(uint => personInfo) public peopleData;

    function cryptoIdentity() public {

    }

    function addPeople(string _name, uint _age, uint _publicPass) public  {
```

Static Analysis raised 8 warning(s) that requires y

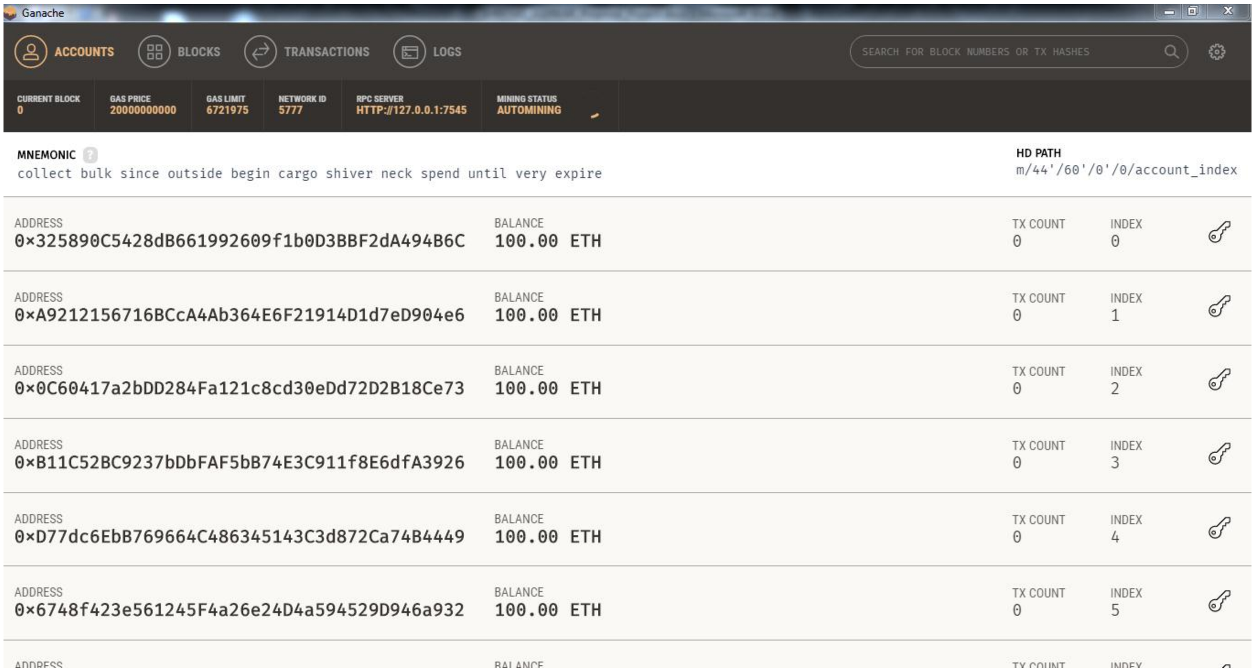
cryptoIdentity

Figure 5

Remix is an online compiler for Solidity language

4. Now we have to write out Solidity code into remix and then keep it aside till we use it further.

5. Now keeping aside the code we wrote in Remix, open the Ganache.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
0	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

MNEMONIC
collect bulk since outside begin cargo shiver neck spend until very expire

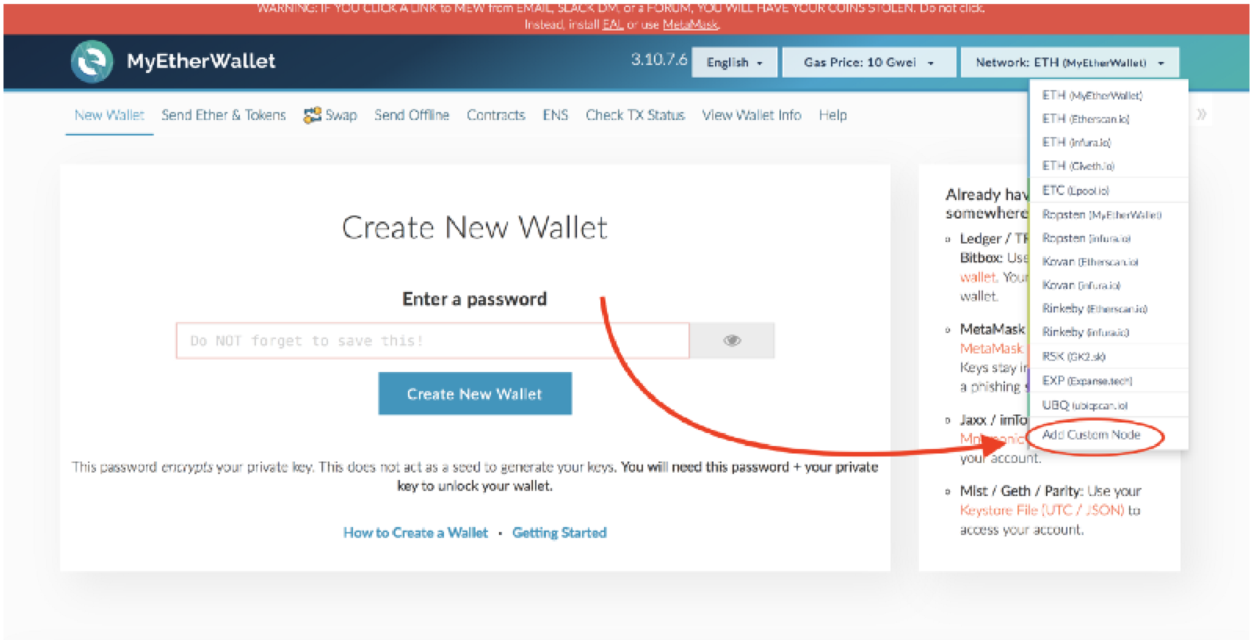
HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×325890C5428dB661992609f1b0D3BBF2dA494B6C	100.00 ETH	0	0
0×A9212156716BCcA4Ab364E6F21914D1d7eD904e6	100.00 ETH	0	1
0×0C60417a2bDD284Fa121c8cd30eDd72D2B18Ce73	100.00 ETH	0	2
0×B11C52BC9237bDbFAF5bB74E3C911f8E6dfA3926	100.00 ETH	0	3
0×D77dc6EbB769664C486345143C3d872Ca74B4449	100.00 ETH	0	4
0×6748f423e561245F4a26e24D4a594529D946a932	100.00 ETH	0	5

Figure 6

On the top of the screen we see the "RPC server" with the default ip of HTTP://127.0.0.1:7545. This will be needed later.

6. Now unzip the Myetheruem wallet and open the "index.html" file in your browser, you will see the following screen like the below snapshot.



WARNING: IF YOU CLICK A LINK to MEW from EMAIL, SLACK DM, or a FORUM, YOU WILL HAVE YOUR COINS STOLEN. Do not click. Instead, install EAL or use MetaMask.

Create New Wallet

Enter a password

Do NOT forget to save this!

Create New Wallet

This password *encrypts* your private key. This does not act as a seed to generate your keys. You will need this password + your private key to unlock your wallet.

How to Create a Wallet · Getting Started

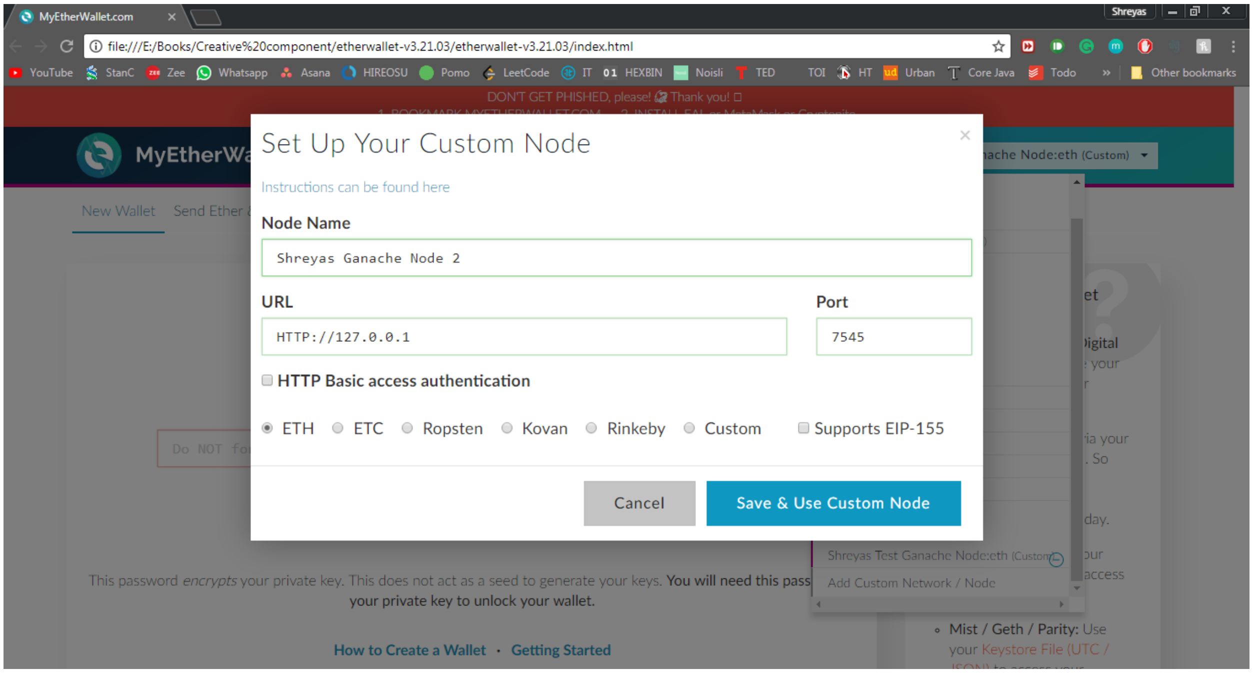
Already hav
somewhere

- Ledger / TR
  Bitbox: Use
  wallet. Your
  wallet.
- MetaMask
  MetaMask
  Keys stay in
  a phishing
- Jaxx / imTo
  Mnemonic
  your account.
- Mist / Geth / Parity: Use your Keystore File (UTC / JSON) to access your account.

Figure 7

In the rightside corner , we can see that network we need to connect, if we click on it, there will be a dropdown menu. By default it connects to the Main network, so we need to create our own custom network, so that it can connect to our private testnet.

7. Now put your RPC information from the Ganache into the required field and name your test net.



MyEtherWallet.com
Shreyas
file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html
YouTube
StanC
Zee
Whatsapp
Asana
HIREOSU
Pomo
LeetCode
IT
HEXBIN
Noisli
TED
TOI
HT
Urban
Core Java
Todo
Other bookmarks
DON'T GET PHISHED, please! Thank you!
MyEtherWa
ache Node:eth (Custom)
New Wallet
Send Ether
Shreyas Test Ganache Node:eth (Custom)
Add Custom Network / Node

Set Up Your Custom Node

Instructions can be found here

Node Name

Shreyas Ganache Node 2

URL

HTTP://127.0.0.1

Port

7545

□ HTTP Basic access authentication

● ETH ○ ETC ○ Ropsten ○ Kovan ○ Rinkeby ○ Custom □ Supports EIP-155

Cancel

Save & Use Custom Node

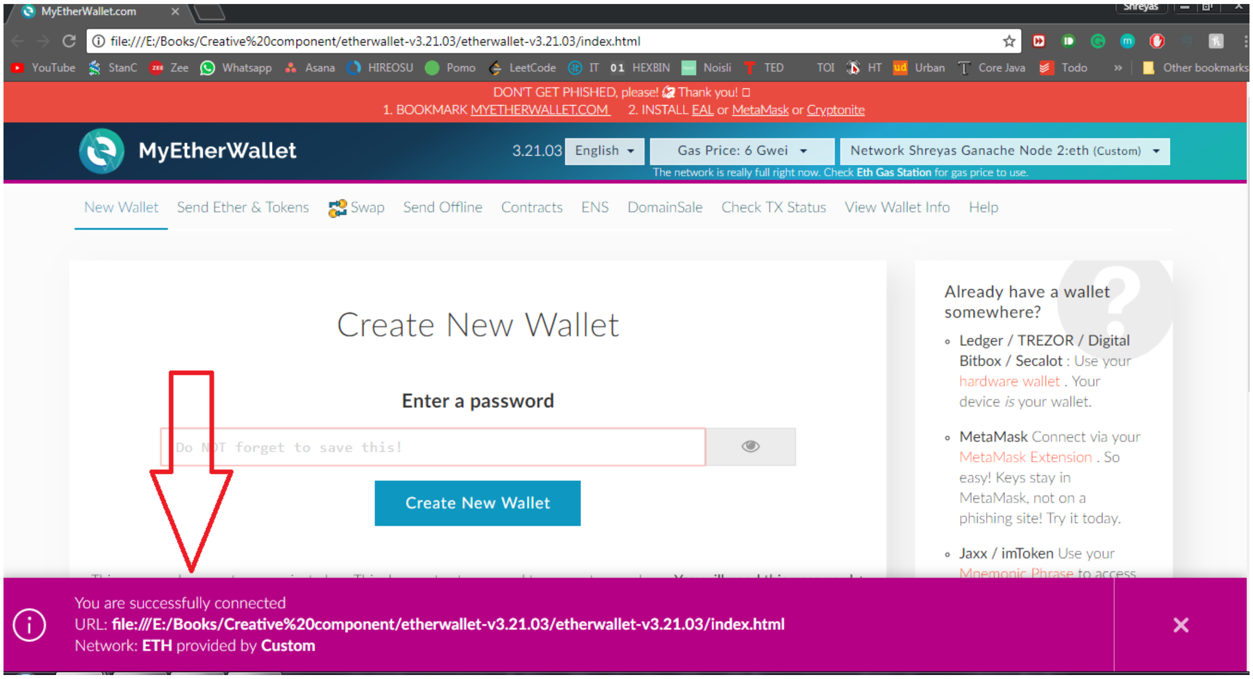
This password *encrypts* your private key. This does not act as a seed to generate your keys. You will need this pass
your private key to unlock your wallet.

How to Create a Wallet · Getting Started

- Mist / Geth / Parity: Use your Keystore File (UTC /

Figure 8

After you enter all the above information , it sucessfully connects to the test network, just like the below snapshot :


MyEtherWallet.com
file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html
YouTube StanC Zee Whatsapp Asana HIREOSU Pomo LeetCode IT HEXBIN Noisli TED TOI HT Urban Core Java Todo Other bookmarks
DON'T GET PHISHED, please! Thank you!
1. BOOKMARK MYETHERWALLET.COM 2. INSTALL EAL or MetaMask or Cryptonite
MyEtherWallet
3.21.03
English
Gas Price: 6 Gwei
Network Shreyas Ganache Node 2:eth (Custom)
The network is really full right now. Check Eth Gas Station for gas price to use.
New Wallet
Send Ether & Tokens
Swap
Send Offline
Contracts
ENS
DomainSale
Check TX Status
View Wallet Info
Help
Do NOT forget to save this!
Create New Wallet

Create New Wallet

Enter a password

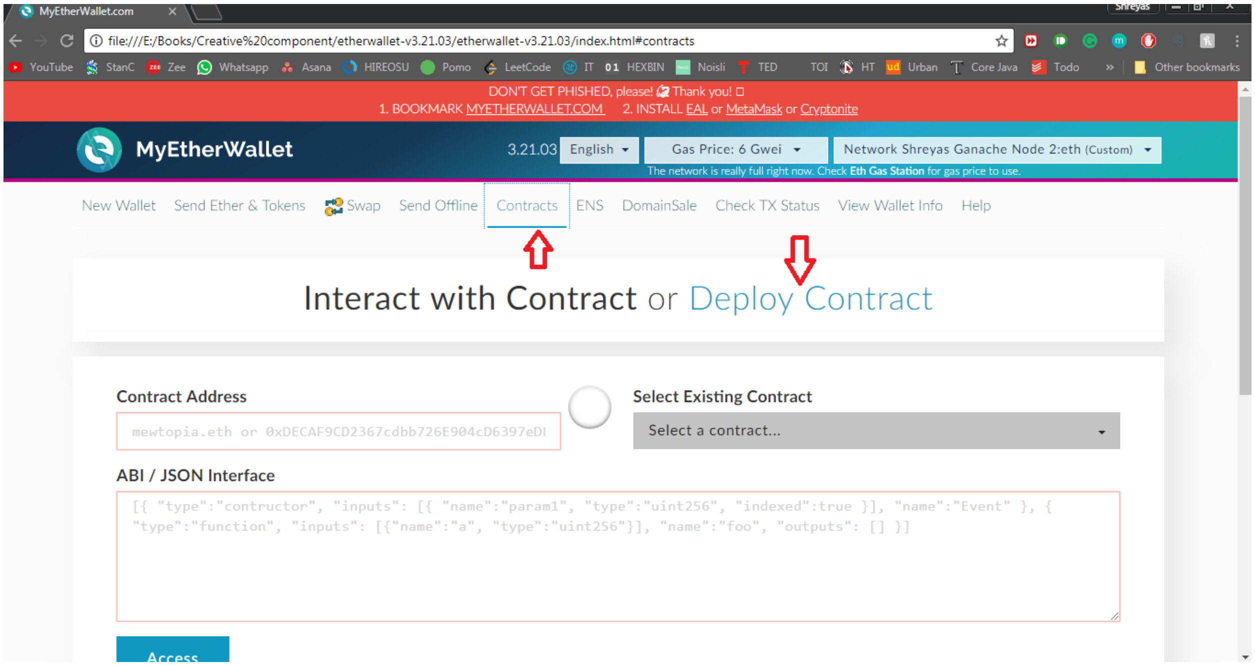
Already have a wallet somewhere?

- Ledger / TREZOR / Digital Bitbox / Secalot : Use your hardware wallet . Your device *is* your wallet.
- MetaMask Connect via your MetaMask Extension . So easy! Keys stay in MetaMask, not on a phishing site! Try it today.
- Jaxx / imToken Use your Mnemonic Phrase to access

You are successfully connected
URL: **file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html**
Network: **ETH** provided by **Custom**

Figure 9

8. Now we have to upload our smart contract to this test net For this we have to choose the option “Contracts” from the below snapshot and further select “Deploy contract”.


MyEtherWallet.com
file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html#contracts
DON'T GET PHISHED, please! Thank you!
1. BOOKMARK MYETHERWALLET.COM 2. INSTALL EAL or MetaMask or Cryptonite
MyEtherWallet
3.21.03
English
Gas Price: 6 Gwei
Network Shreyas Ganache Node 2:eth (Custom)
The network is really full right now. Check Eth Gas Station for gas price to use.
New Wallet
Send Ether & Tokens
Swap
Send Offline
Contracts
ENS
DomainSale
Check TX Status
View Wallet Info
Help
Access

Interact with Contract or Deploy Contract

Contract Address

mewtopia.eth or 0xDECAF9CD2367cdbb726E904cD6397eDI

Select Existing Contract

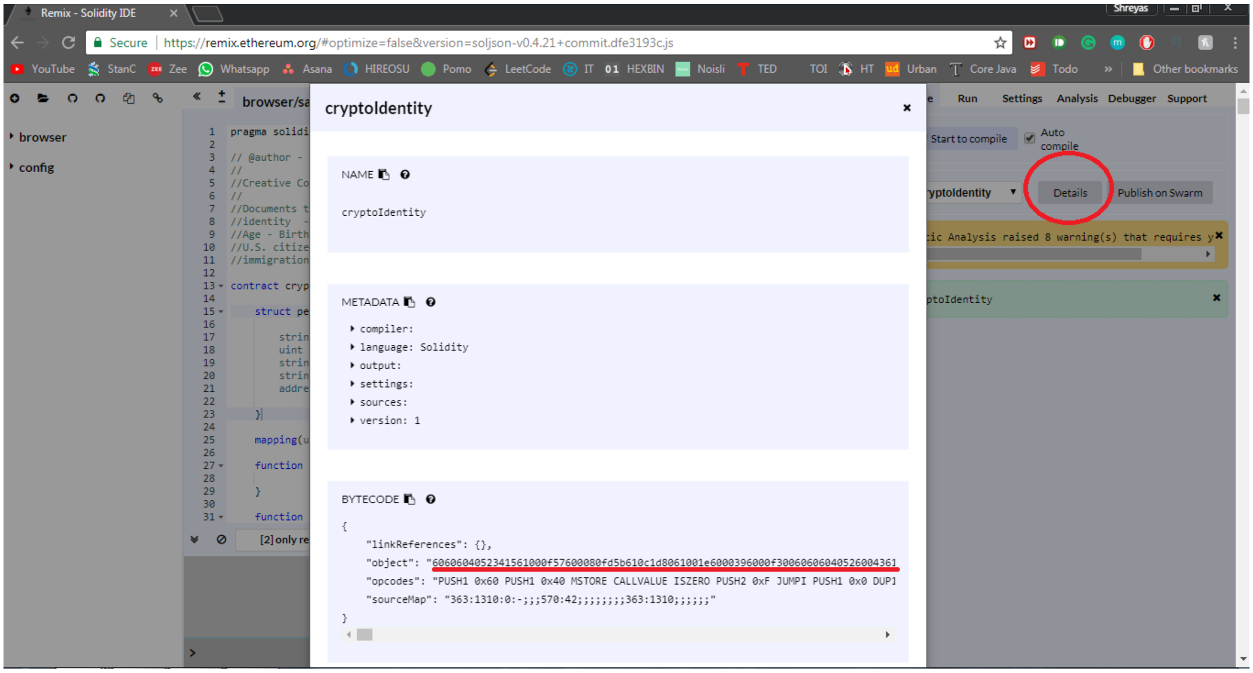
Select a contract...

ABI / JSON Interface

```
[{ "type":"contructor", "inputs": [{ "name":"param1", "type":"uint256", "indexed":true }], "name":"Event" }, { "type":"function", "inputs": [{"name":"a", "type":"uint256"}], "name":"foo", "outputs": [] }]
```

Figure 10

9. Now we have to copy the bytecode from the program we coded in Remix, for this we go to details tab in the right hand side corner of the remix and click on it. Here we will see a lot of background information of the program, here we copy the Bytecode as shown in the below snapshot.



cryptoIdentity

NAME

cryptoIdentity

METADATA

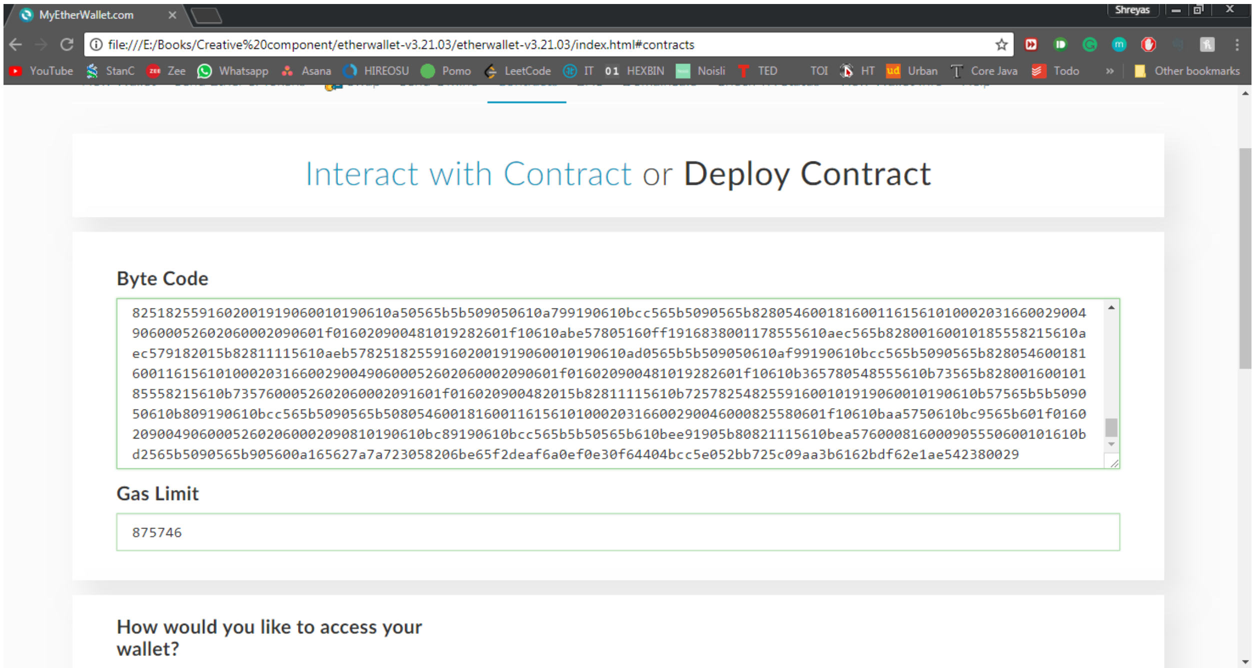
```
▸ compiler:
▸ language: Solidity
▸ output:
▸ settings:
▸ sources:
▸ version: 1
```

BYTECODE

```
{
    "linkReferences": {},
    "object": "6060604052341561000f57600080fd5b610c1d8061001e6000396000f3006060604052600436l
    "opcodes": "PUSH1 0x60 PUSH1 0x40 MSTORE CALLVALUE ISZERO PUSH2 0xF JUMPI PUSH1 0x0 DUP1
    "sourceMap": "363:1310:0:-;;;570:42;;;;;;;;363:1310;;;;;;"
}
```

Figure 11

We copy this Bytecode and paste it into the Byte Code section of the “Deploy Contract” in MyEthereum wallet.



Interact with Contract or Deploy Contract

Byte Code

```
8251825591602001919060010190610a50565b5b509050610a799190610bcc565b5090565b82805460018160011615610100020316600290 04
90600052602060002090601f016020900481019282601f10610abe57805160ff1916838001178555610aec565b82800160010185558215610a
ec579182015b82811115610aeb578251825591602001919060010190610ad0565b5b509050610af99190610bcc565b5090565b828054600181
600116156101000203166002900490600052602060002090601f016020900481019282601f10610b365780548555610b73565b828001600101
85558215610b73576000526020600020916 01f016020900482015b82811115610b72578254825591600101919060010190610b57565b5b5090
50610b809190610bcc565b5090565b50805460018160011615610100020316600290046000825580601f10610baa5750610bc9565b601f0160
20900490600052602060002090810190610bc89190610bcc565b5b50565b610bee91905b80821115610bea576000816000905550600101610b
d2565b5090565b905600a165627a7a723058206be65f2deaf6a0ef0e30f64404bcc5e052bb725c09aa3b6162bdf62e1ae542380029
```

Gas Limit

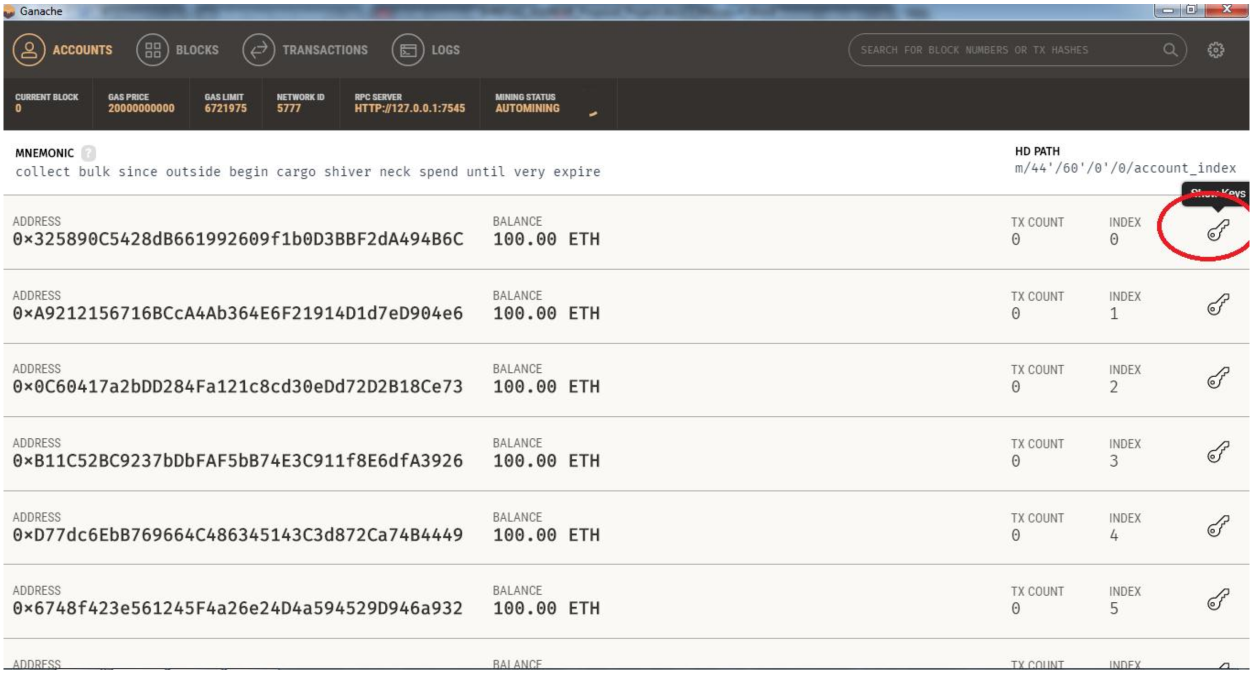
875746

How would you like to access your wallet?

Figure 12

Now if we go down further, we can see that there is an option to choose to access your wallet: MetaMask / Mist, Ledger Wallet,TREZOR, Digital Bitbox, Secalot, Keystore / JSON File , Mnemonic Phrase , Private Key , Parity Phrase. We choose the Private key . Now to provide the private key , We have to got back to Ganache to access the private key.

Ganache gave us 5 addresses that we can use to interact with our private blockchain. To use one of them to upload this contract, we can go back to Ganache and click the key icon for any of the addresses.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
0	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

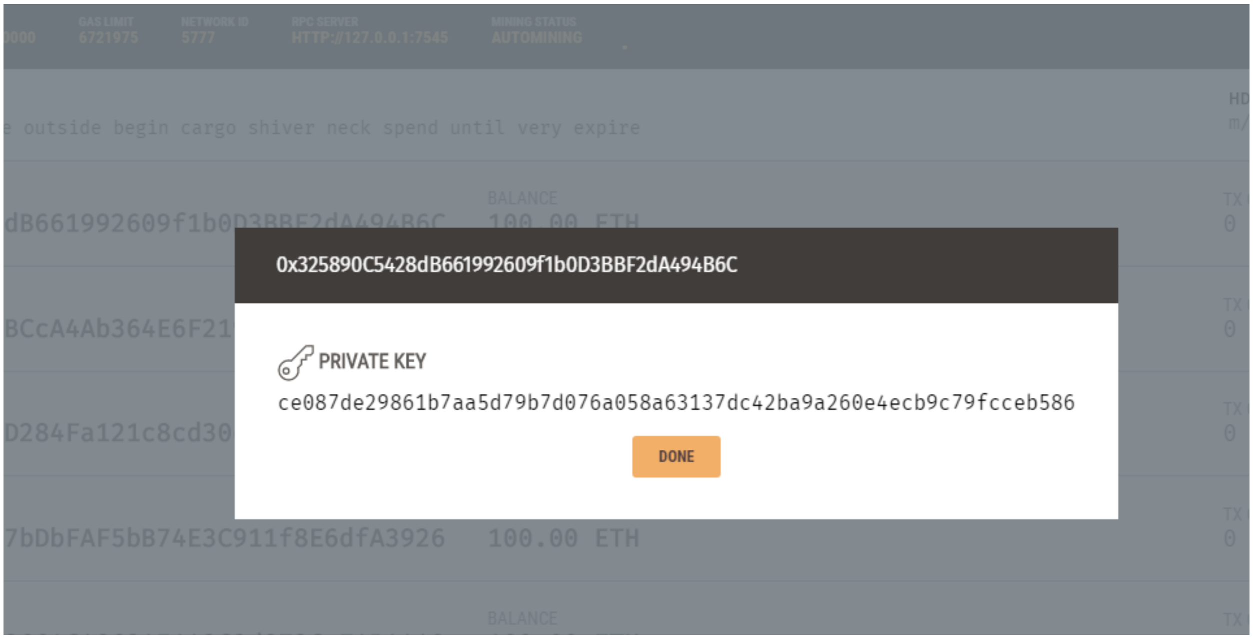
MNEMONIC
collect bulk since outside begin cargo shiver neck spend until very expire

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×325890C5428dB661992609f1b0D3BBF2dA494B6C	100.00 ETH	0	0
0×A9212156716BCcA4Ab364E6F21914D1d7eD904e6	100.00 ETH	0	1
0×0C60417a2bDD284Fa121c8cd30eDd72D2B18Ce73	100.00 ETH	0	2
0×B11C52BC9237bDbFAF5bB74E3C911f8E6dfA3926	100.00 ETH	0	3
0×D77dc6EbB769664C486345143C3d872Ca74B4449	100.00 ETH	0	4
0×6748f423e561245F4a26e24D4a594529D946a932	100.00 ETH	0	5

Figure 13

We click on this key to get the private key of the node,

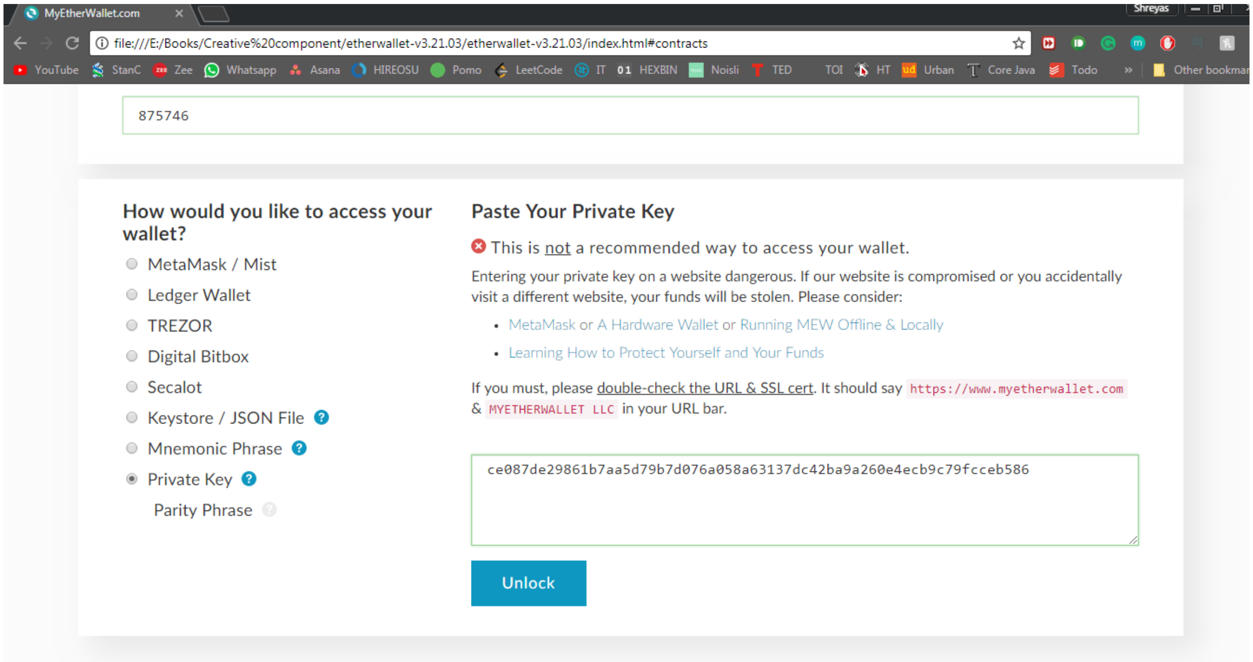

0x325890C5428dB661992609f1b0D3BBF2dA494B6C

PRIVATE KEY

ce087de29861b7aa5d79b7d076a058a63137dc42ba9a260e4ecb9c79fcceb586

Figure 14

We copy this Private key and paste it into the Private key section in the MyEthereum Wallet and click unlock.



875746

How would you like to access your wallet?

○ MetaMask / Mist

○ Ledger Wallet

○ TREZOR

○ Digital Bitbox

○ Secalot

○ Keystore / JSON File

○ Mnemonic Phrase

● Private Key

Parity Phrase

Paste Your Private Key

This is not a recommended way to access your wallet.

Entering your private key on a website dangerous. If our website is compromised or you accidentally visit a different website, your funds will be stolen. Please consider:

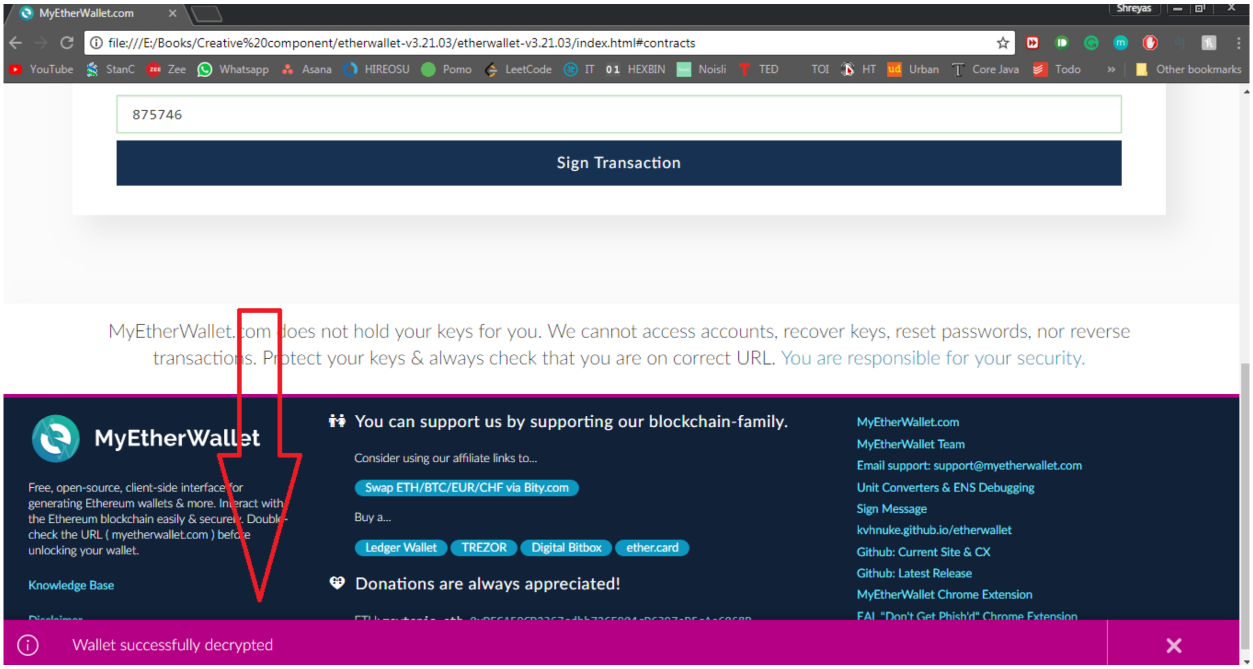
- MetaMask or A Hardware Wallet or Running MEW Offline & Locally
- Learning How to Protect Yourself and Your Funds

If you must, please double-check the URL & SSL cert. It should say `https://www.myetherwallet.com` & `MYETHERWALLET LLC` in your URL bar.

```
ce087de29861b7aa5d79b7d076a058a63137dc42ba9a260e4ecb9c79fcceb586
```

Figure 15

If unlocked successfully, will show the below.


MyEtherWallet.com
file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html#contracts
YouTube
StanC
Zee
Whatsapp
Asana
HIREOSU
Pomo
LeetCode
IT
HEXBIN
Noisli
TED
TOI
HT
Urban
Core Java
Todo
Other bookmarks
Sign Transaction
MyEtherWallet
Swap ETH/BTC/EUR/CHF via Bity.com
Ledger Wallet
TREZOR
Digital Bitbox
ether.card
Knowledge Base
MyEtherWallet.com
MyEtherWallet Team
Email support: support@myetherwallet.com
Unit Converters & ENS Debugging
Sign Message
kvhnuke.github.io/etherwallet
Github: Current Site & CX
Github: Latest Release
MyEtherWallet Chrome Extension
EAL "Don't Get Phish'd" Chrome Extension

875746

MyEtherWallet.com does not hold your keys for you. We cannot access accounts, recover keys, reset passwords, nor reverse transactions. Protect your keys & always check that you are on correct URL. You are responsible for your security.

Free, open-source, client-side interface for generating Ethereum wallets & more. Interact with the Ethereum blockchain easily & securely. Double-check the URL (myetherwallet.com) before unlocking your wallet.

You can support us by supporting our blockchain-family.

Consider using our affiliate links to...

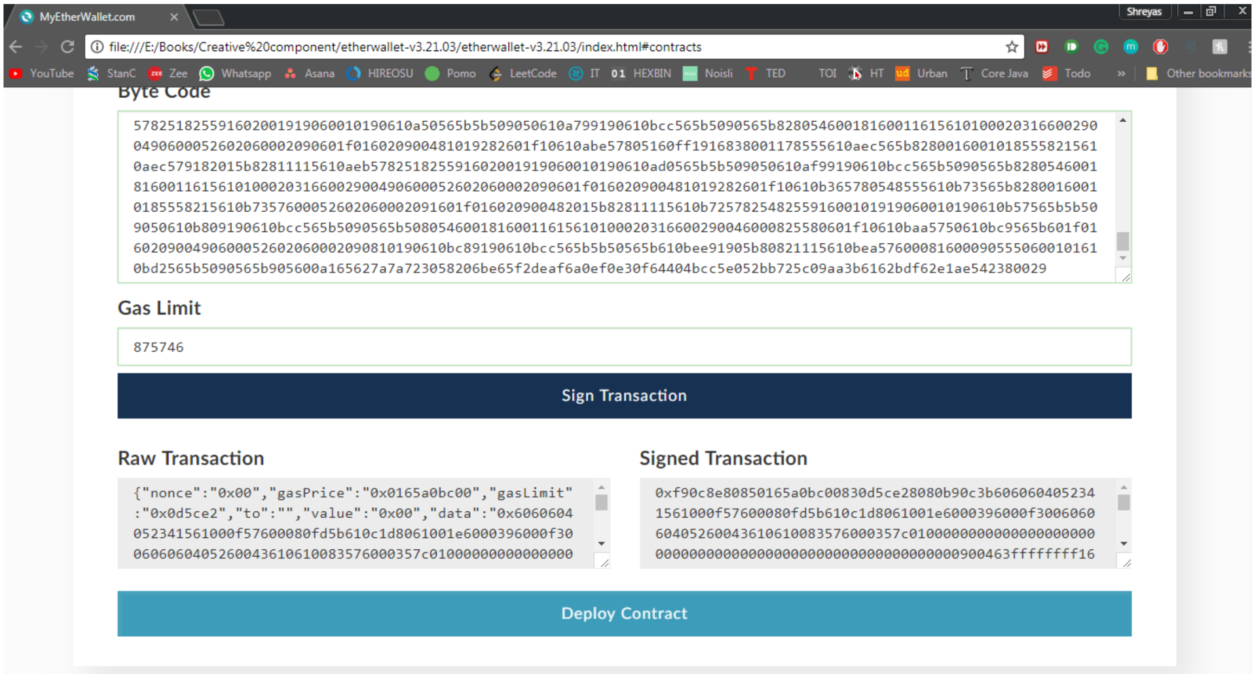
Buy a...

Donations are always appreciated!

Wallet successfully decrypted

Figure 16

10. Now we Sign the transaction, by clicking on clicking on the “Sign transaction“ button and Deploy contract.


MyEtherWallet.com
file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html#contracts
YouTube StanC Zee Whatsapp Asana HIREOSU Pomo LeetCode IT HEXBIN Noisli TED TOI HT Urban Core Java Todo Other bookmarks
Shreyas

Byte Code

```
57825182559160200191906001019061 0a50565b5b509050610a799190610bcc565b5090565b82805460018160011615610100020316600290
```

```
578251825591602001919060010190610a50565b5b509050610a799190610bcc565b5090565b82805460018160011615610100020316600290
0490600052602060002090601f016020900481019282601f10610abe57805160ff1916838001178555610aec565b828001600101855582156 1
0aec579182015b82811115610aeb5782518255916020019190600101906 10ad0565b5b509050610af99190610bcc565b5090565b8280546001
8160011615610100020316600290049060005260206000209060 1f016020900481019282601f10610b365780548555610b73565b8280016001
0185558215610b735760005260206000209160 1f016020900482015b82811115610b725782548255916001019190600101906 10b57565b5b50
9050610b809190610bcc565b5090565b508054600181600116156101000203166002900460008255806 01f10610baa5750610bc9565b601f01
60209004906000526020600020908101906 10bc89190610bcc565b5b50565b610bee91905b80821115610bea576000816000905550600101 61
0bd2565b5090565b905600a165627a7a723058206be65f2deaf6a0ef0e30f64404bcc5e052bb725c09aa3b6162bdf62e1ae542380029
```

Gas Limit

```
875746
```

Raw Transaction

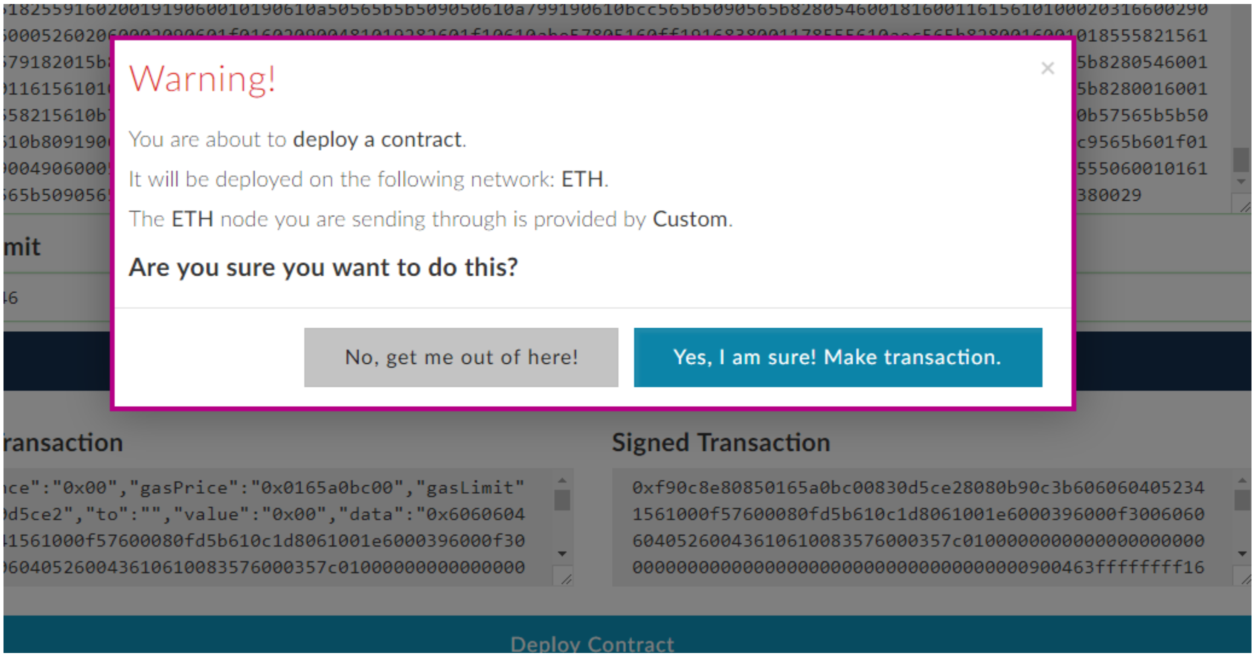
```
{"nonce":"0x00","gasPrice":"0x0165a0bc00","gasLimit"
:"0x0d5ce2","to":"","value":"0x00","data":"0x6060604
052341561000f57600080fd5b610c1d8061001e6000396000f30
060606040526004361061008357600035 7c01000000000000000
```

Signed Transaction

```
0xf90c8e80850165a0bc00830d5ce28080b90c3b606060405234
1561000f57600080fd5b610c1d8061001e6000396000f3006060
604052600436106100835760003 57c010000000000000000000000
00000000000000000000000000000000000900463ffffffff16
```

Figure 17

After you deploy the contract, confirm the transaction.



Warning!

You are about to **deploy a contract**.

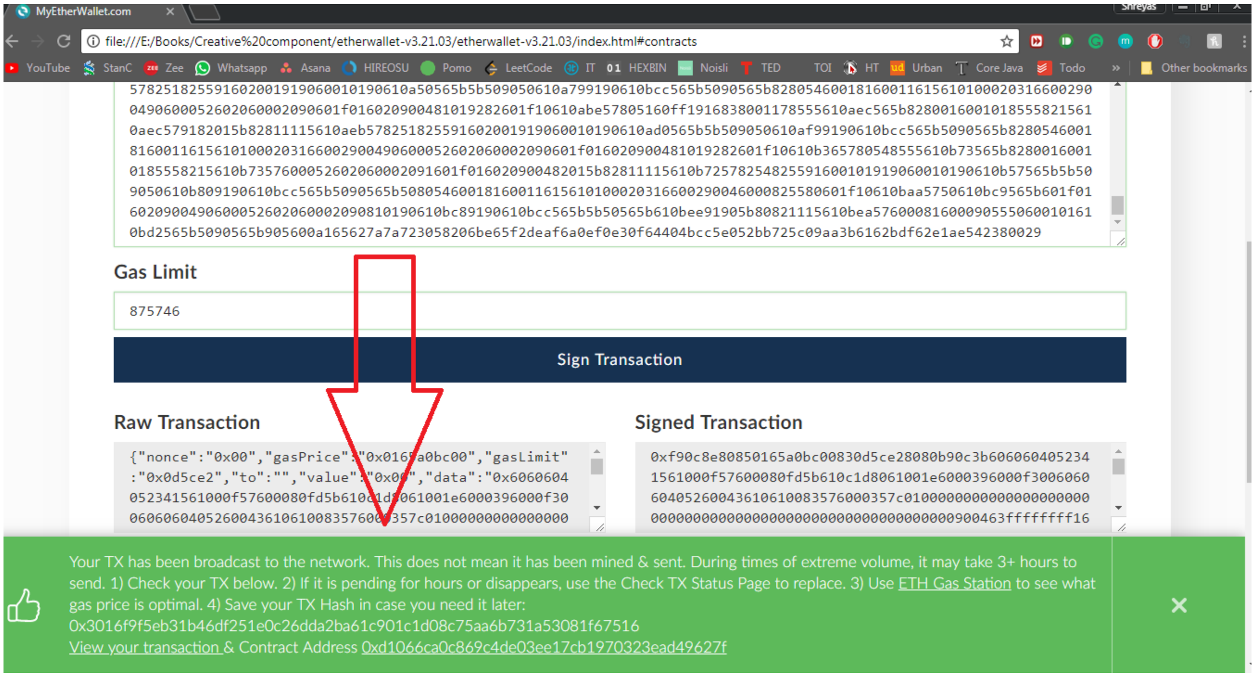
It will be deployed on the following network: ETH.

The ETH node you are sending through is provided by Custom.

Are you sure you want to do this?

Figure 18

After the contract is deployed successfully, it will show the message that it is transaction is done successfully.


MyEtherWallet.com
file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html#contracts
YouTube StanC Zee Whatsapp Asana HIREOSU Pomo LeetCode IT HEXBIN Noisli TED TOI HT Urban Core Java Todo Other bookmarks
Shreyas

```
578251825591602001919060010190610a50565b5b509050610a799190610bcc565b5090565b828054600181600116156101000203166002900
490600052602060002090601f016020900481019282601f10610abe57805160ff19168380011785555610aec565b82800160010185555821561
0aec579182015b82811115610aeb578251825591602001919060010190610ad0565b5b509050610af99190610bcc565b5090565b8280546001
81600116156101000203166002900490600052602060002090601f016020900481019282601f10610b365780548555610b73565b8280016001
0185558215610b7357600052602060002091601f016020900482015b82811115610b7257825482559160010191906001019061 0b57565b5b50
9050610b809190610bcc565b5090565b50805460018160011615610100020316600290046000825580601f10610baa5750610bc9565b601f01
602090049060005260206000209081019061 0bc89190610bcc565b5b50565b610bee91905b80821115610bea576000816000905550600101 61
0bd2565b5090565b905600a165627a7a723058206be65f2deaf6a0ef0e30f64404bcc5e052bb725c09aa3b6162bdf62e1ae542380029
```

Gas Limit

875746

Sign Transaction

Raw Transaction

```
{"nonce":"0x00","gasPrice":"0x0165a0bc00","gasLimit"
:"0x0d5ce2","to":"","value":"0x00","data":"0x6060604
052341561000f57600080fd5b610c1d8061001e6000396000f30
06060604052600436106100835760003357c0100000000000000
```

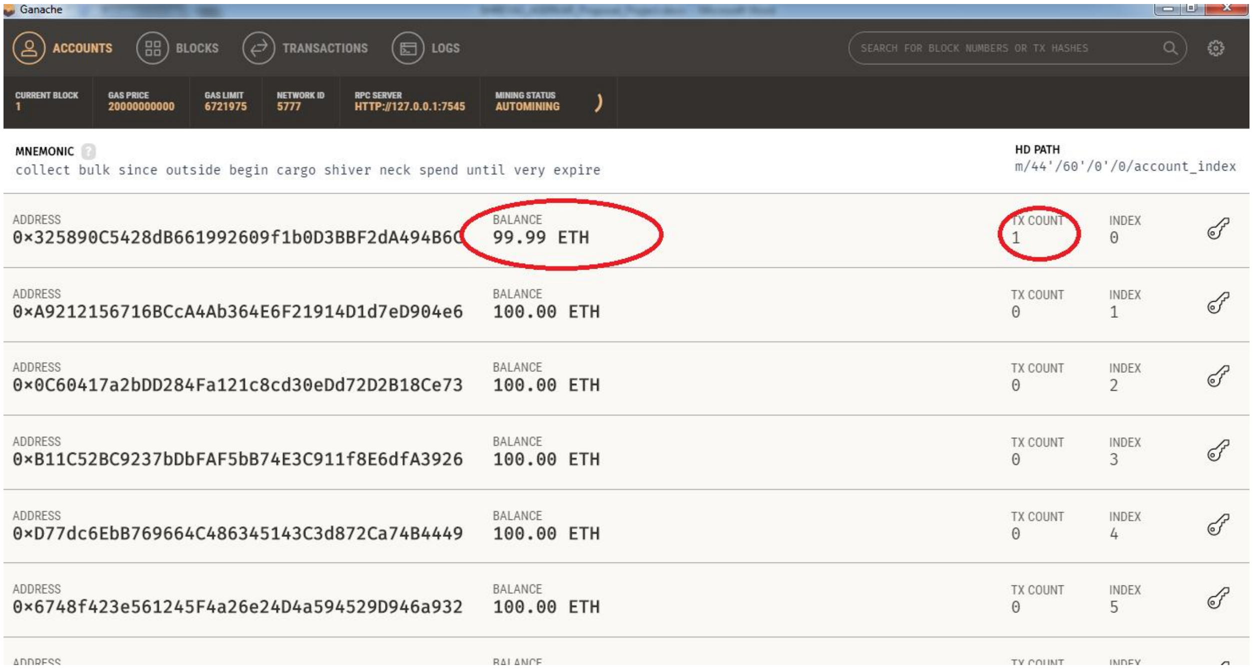
Signed Transaction

```
0xf90c8e80850165a0bc00830d5ce28080b90c3b606060405234
1561000f57600080fd5b610c1d8061001e6000396000f3006060
60405260043610610083576000357c0100000000000000000000
00000000000000000000000000000000000000900463ffffffff16
```

Your TX has been broadcast to the network. This does not mean it has been mined & sent. During times of extreme volume, it may take 3+ hours to send. 1) Check your TX below. 2) If it is pending for hours or disappears, use the Check TX Status Page to replace. 3) Use ETH Gas Station to see what gas price is optimal. 4) Save your TX Hash in case you need it later:
0x3016f9f5eb31b46df251e0c26dda2ba61c901c1d08c75aa6b731a53081f67516
View your transaction & Contract Address 0xd1066ca0c869c4de03ee17cb1970323ead49627f

Figure 19

11. Now we go back to Ganache to see if there is any transaction is done.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
1	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

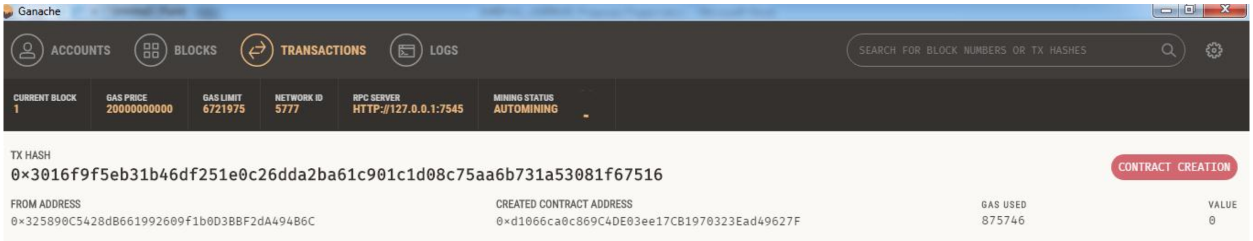
MNEMONIC
collect bulk since outside begin cargo shiver neck spend until very expire

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×325890C5428dB661992609f1b0D3BBF2dA494B6C	99.99 ETH	1	0
0×A9212156716BCcA4Ab364E6F21914D1d7eD904e6	100.00 ETH	0	1
0×0C60417a2bDD284Fa121c8cd30eDd72D2B18Ce73	100.00 ETH	0	2
0×B11C52BC9237bDbFAF5bB74E3C911f8E6dfA3926	100.00 ETH	0	3
0×D77dc6EbB769664C486345143C3d872Ca74B4449	100.00 ETH	0	4
0×6748f423e561245F4a26e24D4a594529D946a932	100.00 ETH	0	5

Figure 20

We go to the transaction and check if the transaction is reflecting.



CURRENT BLOCK 1
GAS PRICE 20000000000
GAS LIMIT 6721975
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING

TX HASH
0×3016f9f5eb31b46df251e0c26dda2ba61c901c1d08c75aa6b731a53081f67516

FROM ADDRESS
0×325890C5428dB661992609f1b0D3BBF2dA494B6C

CREATED CONTRACT ADDRESS
0×d1066ca0c869C4DE03ee17CB1970323Ead49627F

GAS USED
875746

VALUE
0

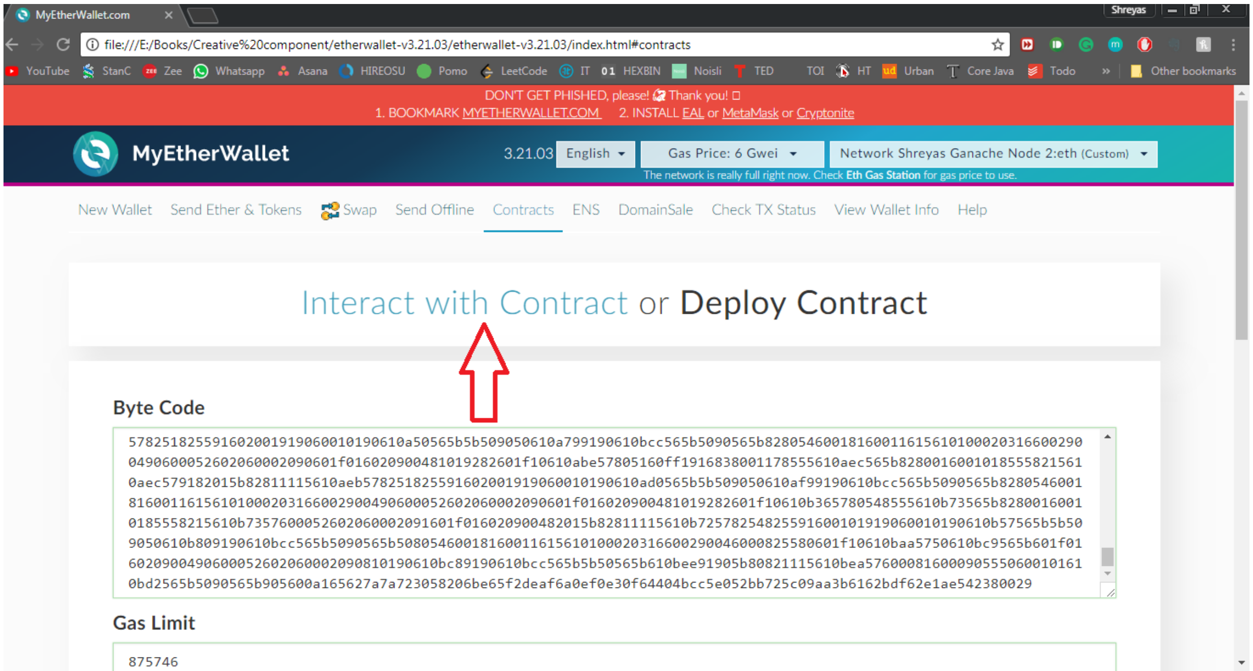
Figure 21

Now we can say that our contract is uploaded to the Private blockchain.

Running Smart Contract on Ganache

Till the previous steps we have uploaded the smart contract to the blockchain. Now we will interact with the contract, for this we will follow the below steps:

1. To interact with the smart contract we deployed in the previous step, We go to MyEthereum Wallet and click on the "Interact with Contract".



DON'T GET PHISHED, please! Thank you!
1. BOOKMARK MYETHERWALLET.COM 2. INSTALL EAL or MetaMask or Cryptonite

The network is really full right now. Check Eth Gas Station for gas price to use.

Interact with Contract or Deploy Contract

Byte Code

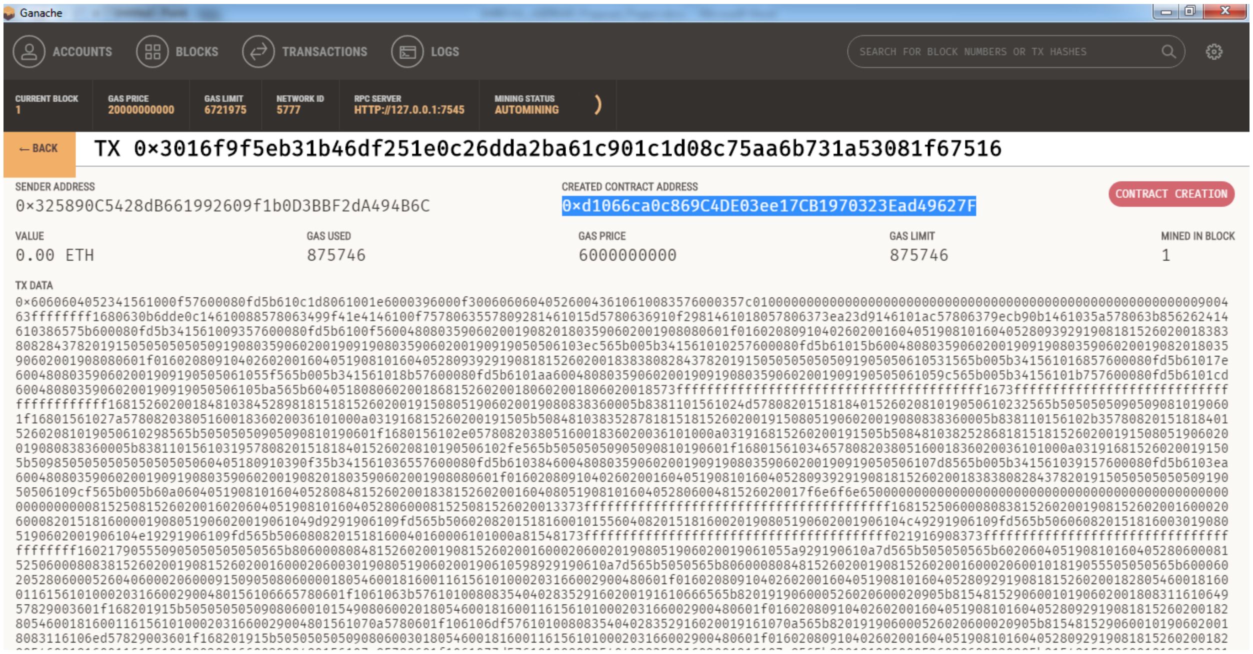
```
578251825591602001919060010190610a50565b5b509050610a799190610bcc565b5090565b828054600181600116156101000203166002900
```

Gas Limit

875746

Figure 22

2. Upon clicking the "Interact with contract", a new page will open up , which will ask for "Contact Address" and "ABI/JSON Interface". To get both the value , we have to open up Ganache >Transactions , here we will get the Contact Address of the latest block on which we have deployed the contract, below is the screenshot of the Ganache showing the contract address which we need to refer to.



CURRENT BLOCK 1 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING

TX 0×3016f9f5eb31b46df251e0c26dda2ba61c901c1d08c75aa6b731a53081f67516

SENDER ADDRESS
0×325890C5428dB661992609f1b0D3BBF2dA494B6C

CREATED CONTRACT ADDRESS
0×d1066ca0c869C4DE03ee17CB1970323Ead49627F

VALUE
0.00 ETH

GAS USED
875746

GAS PRICE
6000000000

GAS LIMIT
875746

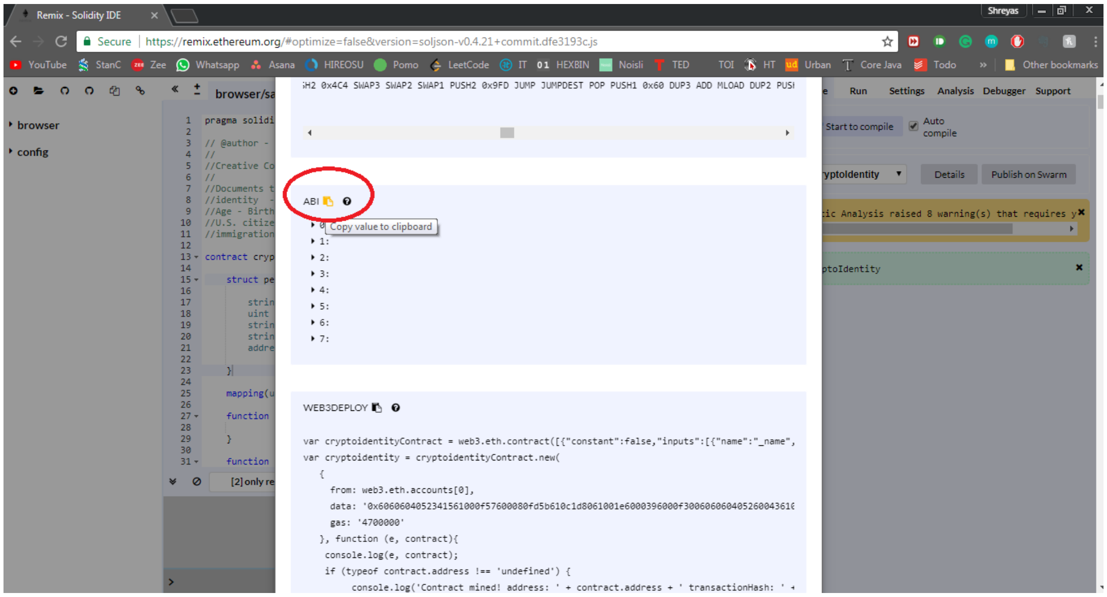
MINED IN BLOCK
1

TX DATA

```
0×6060604052341561000f57600080fd5b610c1d8061001e6000396000f300606060405260043610610083576000357c0100000000000000000000000000000000000000000000000000000000009004
63ffffffff1680630b6dde0c14610088578063499f41e4146100f7578063557809281461015d5780636910f2981461018057806373ea23d9146101ac57806379ecb90b1461035a578063b856262414
610386575b600080fd5b341561009357600080fd5b6100f5600480803590602001908201803590602001908080601f0160208091040260200160405190810160405280939291908181526020018383
80828437820191505050505050919080359060200190919080359060200190919050506103ec565b005b341561010257600080fd5b61015b6004808035906020019091908035906020019082018035
90602001908080601f01602080910402602001604051908101604052809392919081815260200183838082843782019150505050505091905050610531565b005b341561016857600080fd5b61017e
600480803590602001909190505061055f565b005b341561018b57600080fd5b6101aa600480803590602001909190803590602001909190505061059c565b005b34156101b757600080fd5b6101cd
60048080359060200190919050506105ba565b604051808060200186815260200180602001806020018573ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffff
ffffffffffff168152602001848103845289818151815260200191508051906020019080838360005b8381101561024d578082015181840152602081019050610232565b5050505090509081019060
1f16801561027a5780820380516001836020036101000a031916815260200191505b50848103835287818151815260200191508051906020019080838360005b838110156102b35780820151818401
52602081019050610298565b50505050905090810190601f1680156102e05780820380516001836020036101000a031916815260200191505b50848103825286818151815260200191508051906020
019080838360005b838110156103195780820151818401526020810190506102fe565b50505050905090810190601f1680156103465780820380516001836020036101000a03191681526020019150
5b509850505050505050505050604051809103 90f35b341561036557600080fd5b6103846004808035906020019091908035906020019091905050 6107d8565b005b341561039157600080fd5b6103ea
600480803590602001909190803590602001908201803590602001908080601f0160208091040260200160405190810160405280939291908181526020018383808284378201915050505050509190
50506109cf565b005b60a0604051908101604052808481526020018381526020016040805190810160405280600081526020017f6e6f6e650000000000000000000000000000000000000000000000
00000000008152508152602001602060405190810160405280600081525081526020013373ffffffffffffffffffffffffffffffffffffffff16815250600080838152602001908152602001600020
6000820151816000019080519060200190610 49d9291906109fd565b506020820151816001015560408201518160020190805190602001906104c49291906109fd565b506060820151816003019080
5190602001906104e19291906109fd565b5060808201518160040160006101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffff
ffffffff1602179055509050505050565b80600080848152602001908152602001600020600201908051906020019061055a929190610a7d565b505050565b602060405190810160405280600081
52506000808381526020019081526020016000206003019080519060200190610598929190610a7d565b5050565b80600080848152602001908152602001600020600101819055505050565b600060
20528060005260406000206000915090508060000180546001816001161561010002031660029004806 01f0160208091040260200160405190810160405280929190818152602001828054600181 60
0116156101000203166002900480156106665780601f1061063b57610100808354040283529160200191610666565b820191906000526020600020905b815481529060010190602001808311610649
57829003601f168201915b505050505090806001015490806002018054600181600116156101000203166002900480601f0160208091040260200160405190810160405280929190818152602001 82
8054600181600116156101000203166002900480156107 0a5780601f106106df5761010080835404028352916020019161070a565b820191906000526020600020905b815481529060010190602001
8083116106ed57829003601f168201915b505050505090806003018054600181600116156101000203166002900480601f0160208091040260200160405190810160405280929190818152602001 82
```

Figure 23

We copy this address and paste it into the "Contact address" section of MyEthereum Wallet. Now to get the "ABI/JSON Interface" we again go back to Remix , click on the Details on the right hand side of the page, and copt the ABI from the details by clicking on the copy button marked in the below snapshot and paste this information into the "ABI/JSON Interface" in MyEthereum Wallet.



```
pragma solidi
// @author -
//
//Creative Co
//
//Documents t
//identity  -
//Age - Birth
//U.S. citize
//immigration

contract cryp

    struct pe

        strin
        uint
        strin
        strin
        addre

    }

    mapping(u

    function

    }

    function
```

```
:H2 0x4C4 SWAP3 SWAP2 SWAP1 PUSH2 0x9FD JUMP JUMPDEST POP PUSH1 0x60 DUP3 ADD MLOAD DUP2 PUSI
```

tic Analysis raised 8 warning(s) that requires y

ptoIdentity

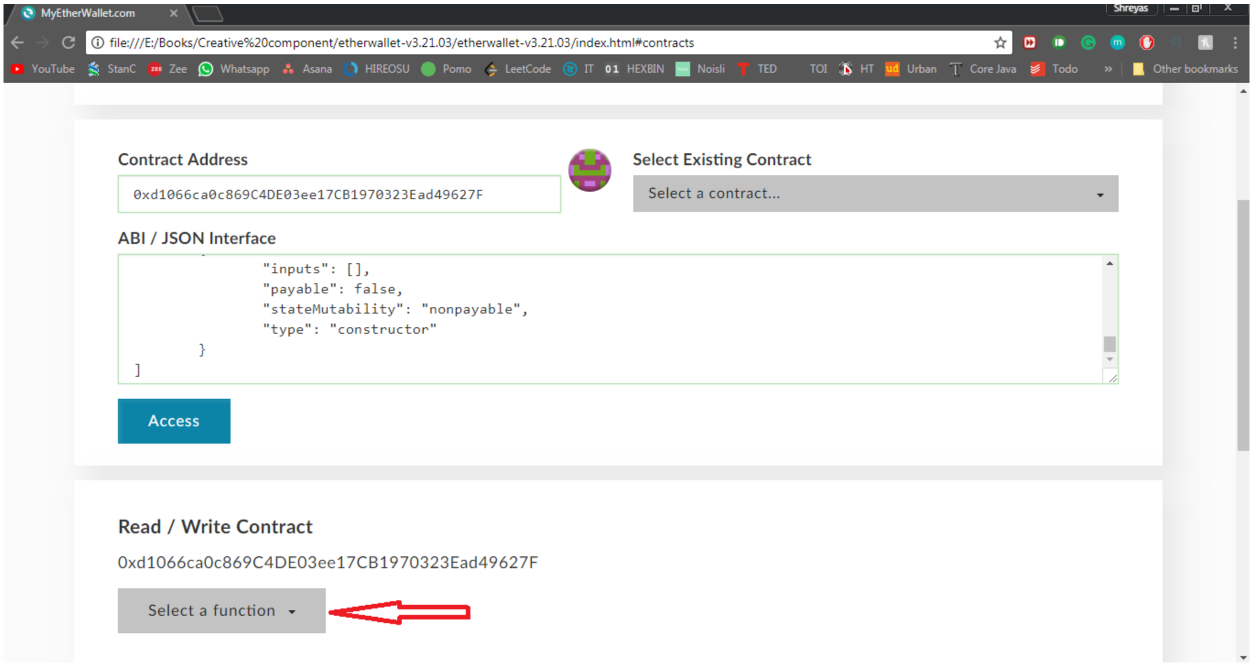
ABI
- 0
- 1:
- 2:
- 3:
- 4:
- 5:
- 6:
- 7:

WEB3DEPLOY

```
var cryptoidentityContract = web3.eth.contract([{"constant":false,"inputs":[{"name":"_name",
var cryptoidentity = cryptoidentityContract.new(
   {
     from: web3.eth.accounts[0],
     data: '0x6060604052341561000f57600080fd5b610c1d8061001e6000396000f30060606040526004361(
     gas: '4700000'
   }, function (e, contract){
    console.log(e, contract);
    if (typeof contract.address !== 'undefined') {
         console.log('Contract mined! address: ' + contract.address + ' transactionHash: ' +
```

Figure 24

Now we click on the Access button, and now we can see all the functions we coded in the Remix, just like the below snapshot.



file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html#contracts

Contract Address

0xd1066ca0c869C4DE03ee17CB1970323Ead49627F

Select Existing Contract

Select a contract...

ABI / JSON Interface

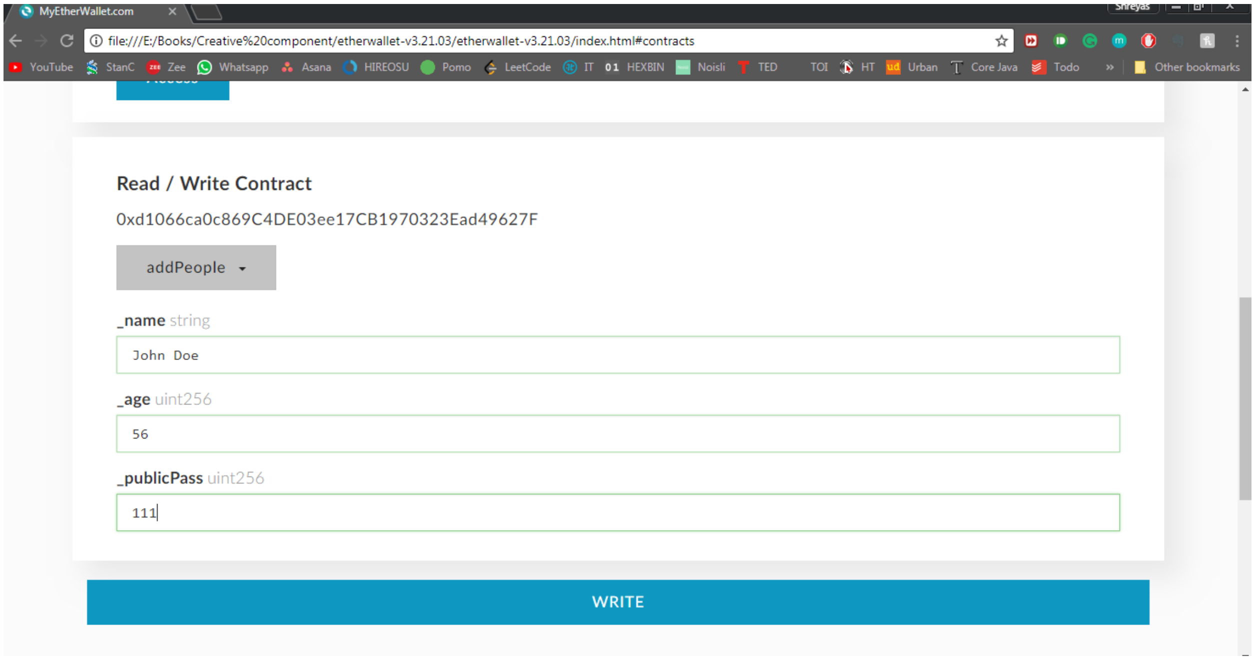
```
                "inputs": [],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "constructor"
        }
]
```

Read / Write Contract

0xd1066ca0c869C4DE03ee17CB1970323Ead49627F

Figure 25

3. Now that we have Interacted with the contract, we will use its functions Like adding Person information, and check that if is deployed to all the blocks and if the transaction is successful.



file:///E:/Books/Creative%20component/etherwallet-v3.21.03/etherwallet-v3.21.03/index.html#contracts

Read / Write Contract

0xd1066ca0c869C4DE03ee17CB1970323Ead49627F

_name string

John Doe

_age uint256

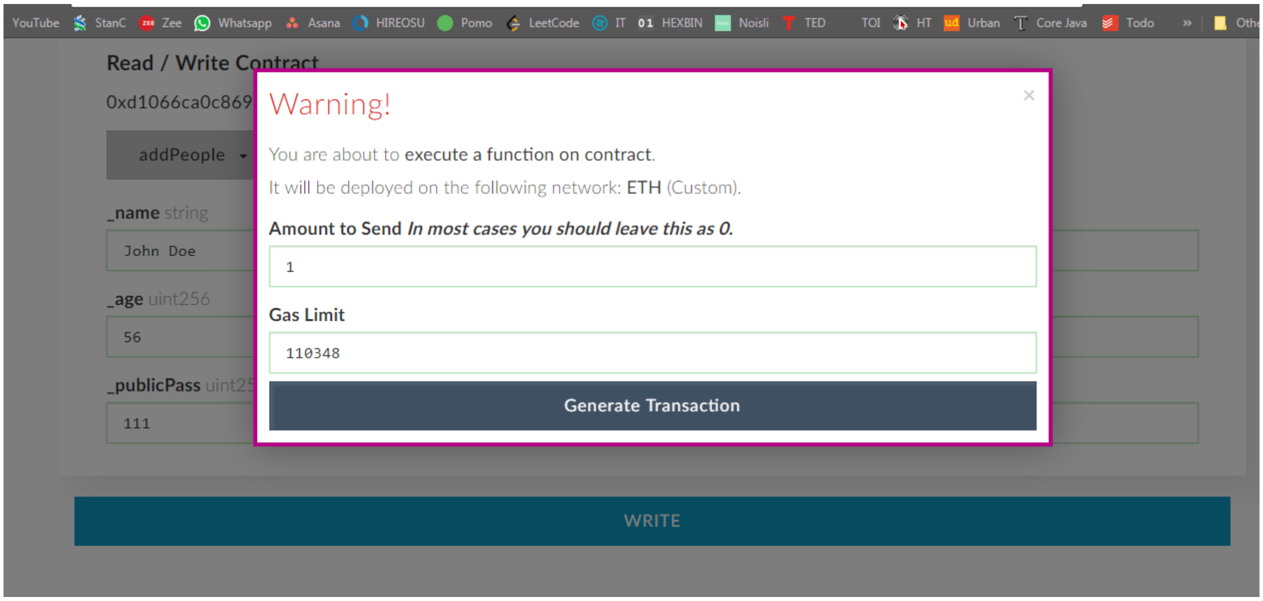
56

_publicPass uint256

111

Figure 26

Now if we click on the Write button, it will ask for some amount that you to send in order to generate the transaction, also it by default chooses the gas Limit, which is by default generated, when a contract is created.



Read / Write Contract

0xd1066ca0c869

_name string

John Doe

_age uint256

56

_publicPass uint25

111

Warning!

You are about to execute a function on contract.

It will be deployed on the following network: ETH (Custom).

Amount to Send *In most cases you should leave this as 0.*

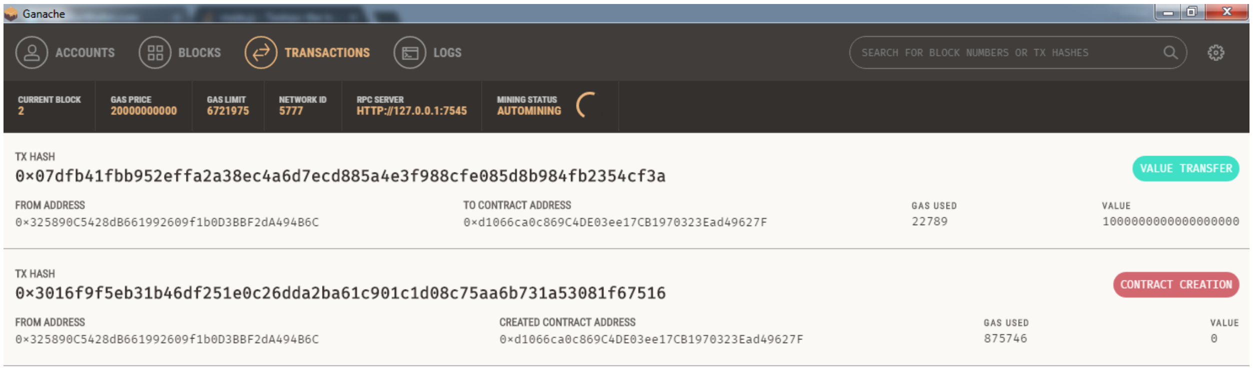
1

Gas Limit

110348

Figure 27

After you click on the “Generate Transaction” the transaction will be generated successfully.The below figure shows that the “addPeople” function transaction was completed successfully. As a new transaction block is added to the blockchain.



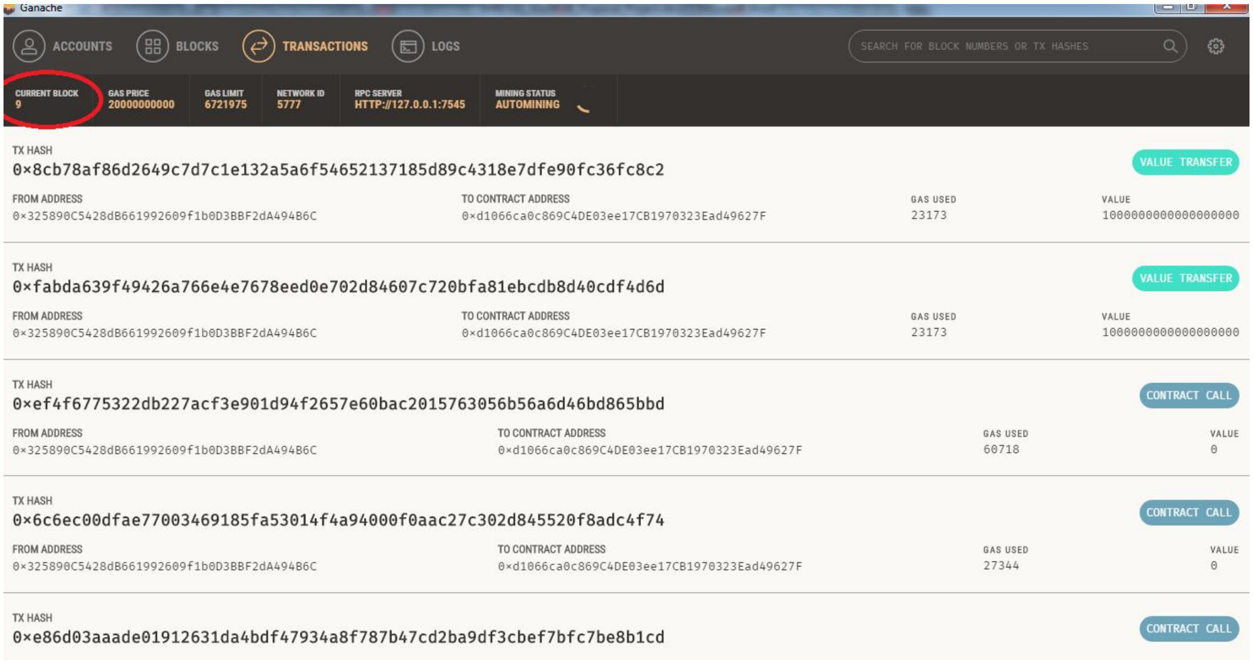
CURRENT BLOCK 2 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING

TX HASH
0×07dfb41fbb952effa2a38ec4a6d7ecd885a4e3f988cfe085d8b984fb2354cf3a
VALUE TRANSFER
FROM ADDRESS 0×325890C5428dB661992609f1b0D3BBF2dA494B6C
TO CONTRACT ADDRESS 0×d1066ca0c869C4DE03ee17CB1970323Ead49627F
GAS USED 22789
VALUE 1000000000000000000

TX HASH
0×3016f9f5eb31b46df251e0c26dda2ba61c901c1d08c75aa6b731a53081f67516
CONTRACT CREATION
FROM ADDRESS 0×325890C5428dB661992609f1b0D3BBF2dA494B6C
CREATED CONTRACT ADDRESS 0×d1066ca0c869C4DE03ee17CB1970323Ead49627F
GAS USED 875746
VALUE 0

Figure 28

After checking that the transactions are happening successfully, I invoked a few functions and added Identity information associated with each person individually and it worked fine. Below is the transaction blocks that were formed after I performs some functions from the contract.

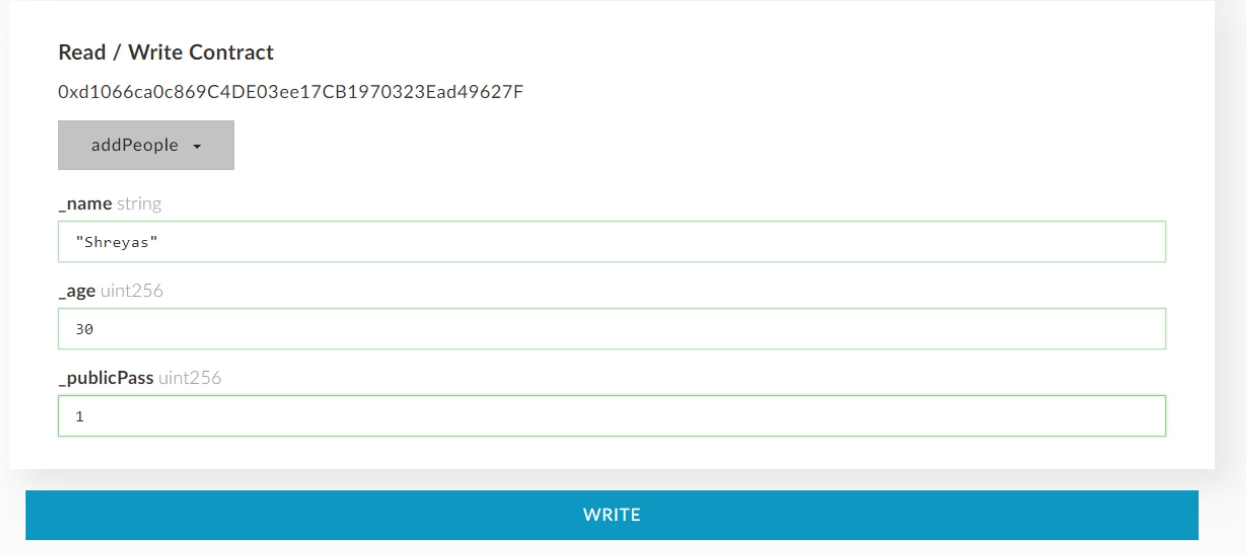


```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  NETWORK ID  RPC SERVER             MINING STATUS
9              20000000000  6721975    5777        HTTP://127.0.0.1:7545  AUTOMINING

TX HASH
0×8cb78af86d2649c7d7c1e132a5a6f54652137185d89c4318e7dfe90fc36fc8c2                VALUE TRANSFER
FROM ADDRESS                                TO CONTRACT ADDRESS                          GAS USED   VALUE
0×325890C5428dB661992609f1b0D3BBF2dA494B6C  0×d1066ca0c869C4DE03ee17CB1970323Ead49627F   23173      1000000000000000000

TX HASH
0×fabda639f49426a766e4e7678eed0e702d84607c720bfa81ebcdb8d40cdf4d6d                VALUE TRANSFER
FROM ADDRESS                                TO CONTRACT ADDRESS                          GAS USED   VALUE
0×325890C5428dB661992609f1b0D3BBF2dA494B6C  0×d1066ca0c869C4DE03ee17CB1970323Ead49627F   23173      1000000000000000000

TX HASH
0×ef4f6775322db227acf3e901d94f2657e60bac2015763056b56a6d46bd865bbd                CONTRACT CALL
FROM ADDRESS                                TO CONTRACT ADDRESS                          GAS USED   VALUE
0×325890C5428dB661992609f1b0D3BBF2dA494B6C  0×d1066ca0c869C4DE03ee17CB1970323Ead49627F   60718      0

TX HASH
0×6c6ec00dfae77003469185fa53014f4a94000f0aac27c302d845520f8adc4f74                CONTRACT CALL
FROM ADDRESS                                TO CONTRACT ADDRESS                          GAS USED   VALUE
0×325890C5428dB661992609f1b0D3BBF2dA494B6C  0×d1066ca0c869C4DE03ee17CB1970323Ead49627F   27344      0

TX HASH
0×e86d03aaade01912631da4bdf47934a8f787b47cd2ba9df3cbef7bfc7be8b1cd                CONTRACT CALL
```

Figure 29

Now we try to check if different user can access each other's data and further change it. For that I associated two different accounts having separate private keys and deployed contract on both of them.

PERSON1:



Read / Write Contract

0xd1066ca0c869C4DE03ee17CB1970323Ead49627F

_name string

"Shreyas"

_age uint256

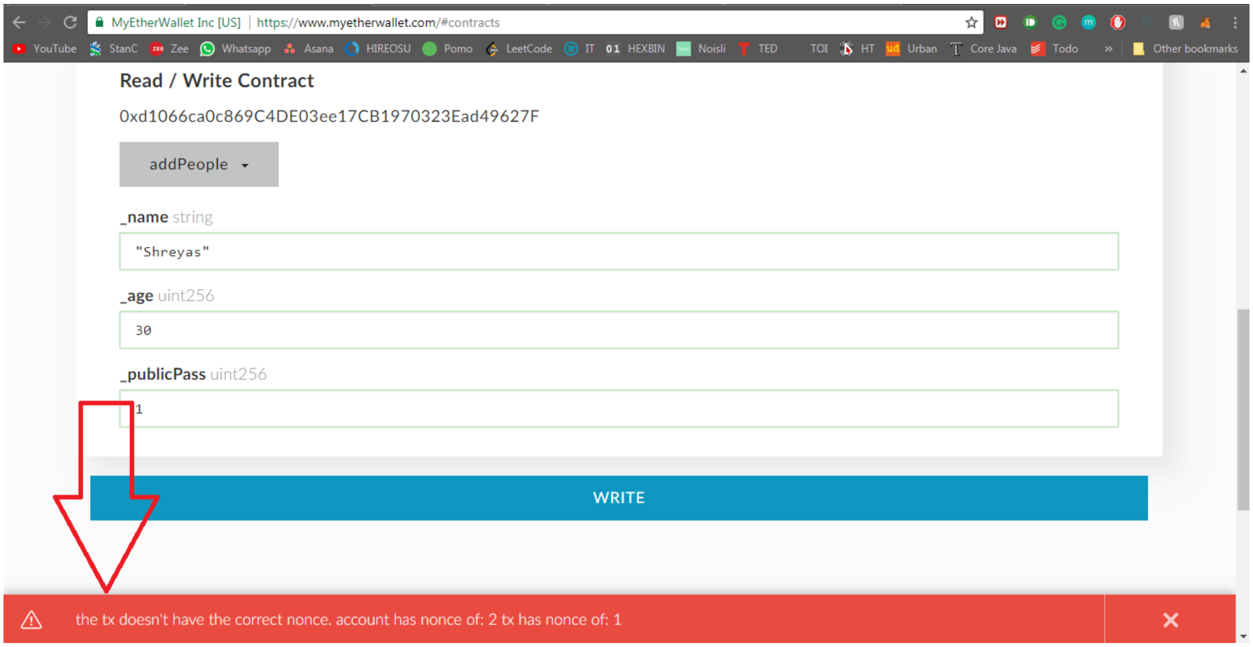
30

_publicPass uint256

1

Figure 30

If he tries to use the same function again, it gives an error,


MyEtherWallet Inc [US] | https://www.myetherwallet.com/#contracts
YouTube StanC Zee Whatsapp Asana HIREOSU Pomo LeetCode IT HEXBIN Noisli TED TOI HT Urban Core Java Todo Other bookmarks

Read / Write Contract

0xd1066ca0c869C4DE03ee17CB1970323Ead49627F

addPeople

_name string

"Shreyas"

_age uint256

30

_publicPass uint256

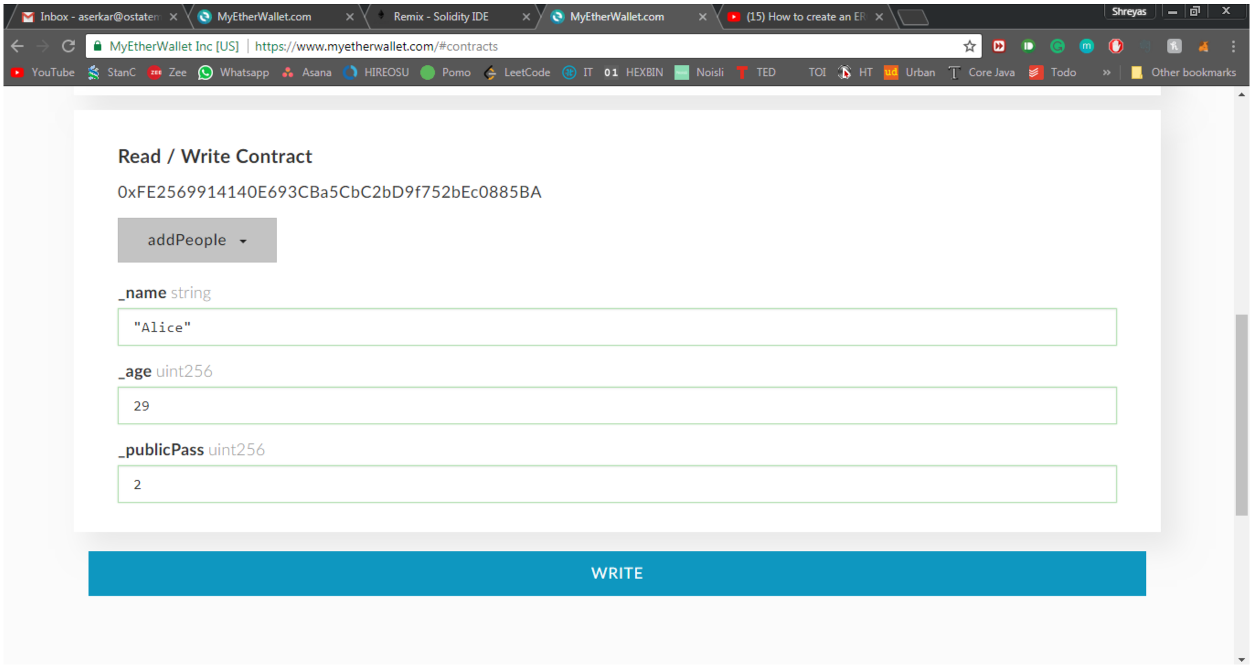
1

WRITE

the tx doesn't have the correct nonce. account has nonce of: 2 tx has nonce of: 1

Figure 31

PERSON2:


Inbox - aserkar@ostatem | MyEtherWallet.com | Remix - Solidity IDE | MyEtherWallet.com | (15) How to create an ER | Shreyas
MyEtherWallet Inc [US] | https://www.myetherwallet.com/#contracts
YouTube StanC Zee Whatsapp Asana HIREOSU Pomo LeetCode IT HEXBIN Noisli TED TOI HT Urban Core Java Todo Other bookmarks

Read / Write Contract

0xFE2569914140E693CBa5CbC2bD9f752bEc0885BA

addPeople

_name string

"Alice"

_age uint256

29

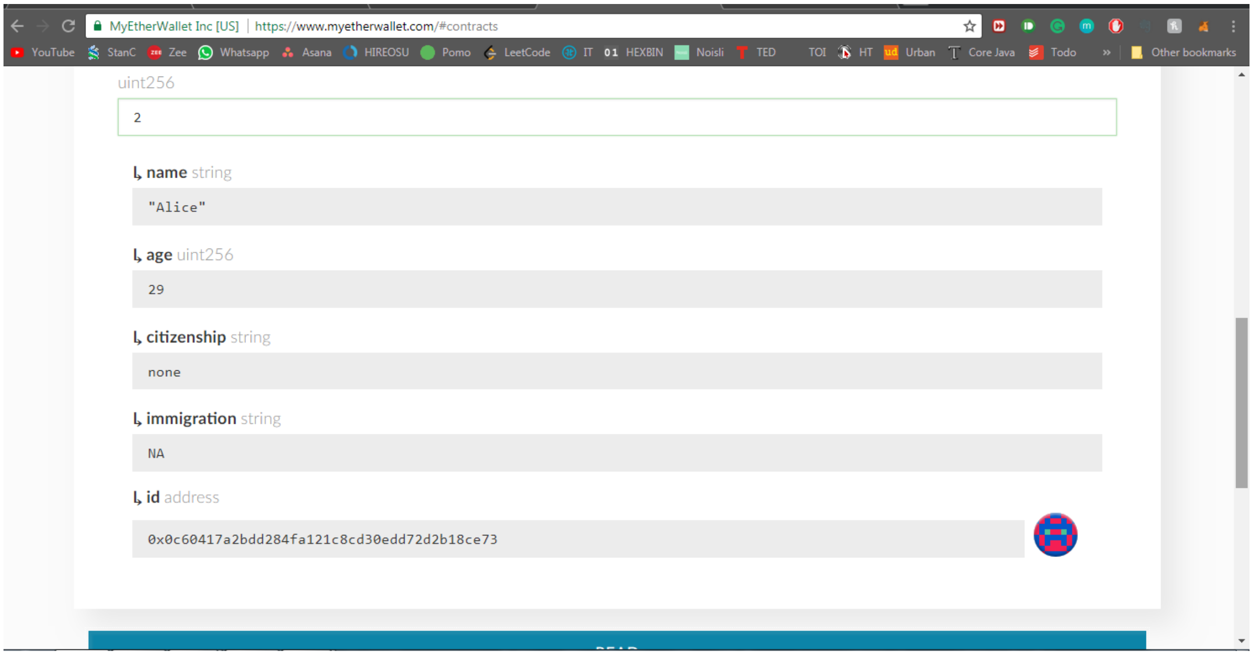
_publicPass uint256

2

WRITE

Figure 32

The contract were only able to view and modify their data, the other person data is hidden from the other.



uint256

2

↳ **name** string

"Alice"

↳ **age** uint256

29

↳ **citizenship** string

none

↳ **immigration** string

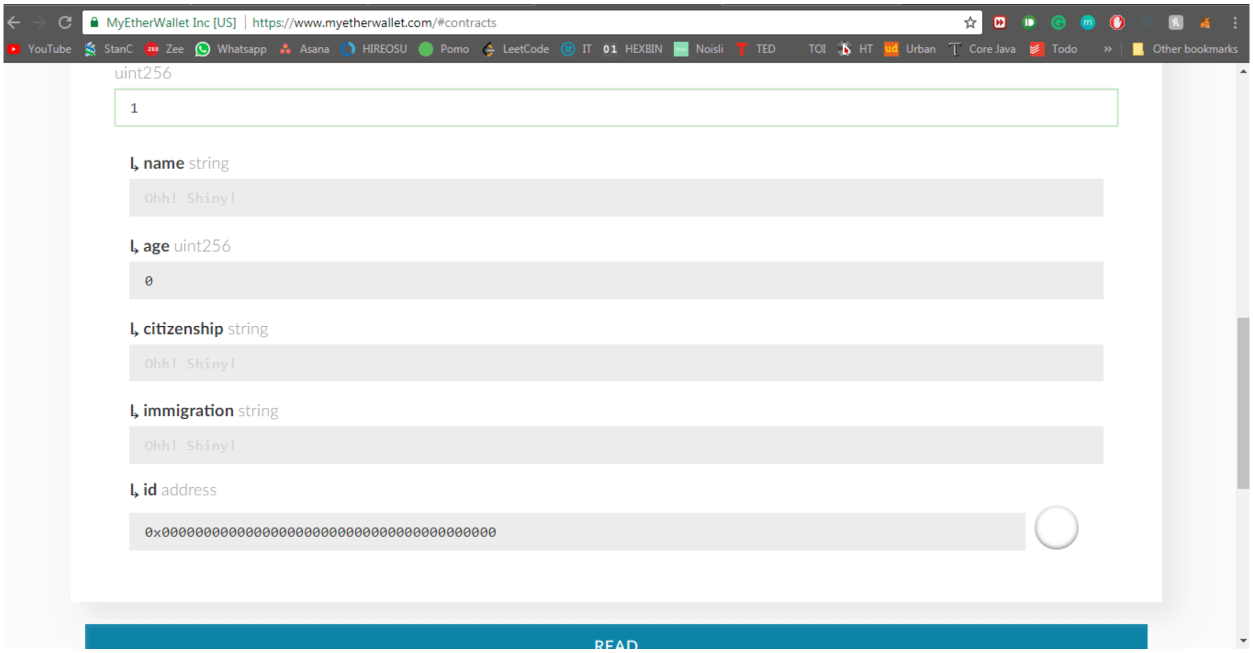
NA

↳ **id** address

0x0c60417a2bdd284fa121c8cd30edd72d2b18ce73

Figure 33

Below is the snapshot, if Alice tries to access Person 1’s data.



uint256

1

↳ **name** string

Ohh! Shiny!

↳ **age** uint256

0

↳ **citizenship** string

Ohh! Shiny!

↳ **immigration** string

Ohh! Shiny!

↳ **id** address

0x0000000000000000000000000000000000000000

Figure 34

Source Code

```
pragma solidity ^0.4.16;


// @author - Shreyas Aserkar
//
//Creative Component - A Study on Crypto identity
//
//Documents that cam be used for the below functions
//identity  -like passport or driver's license etc.
//Age - Birth certificate, U.S. passport etc.
//U.S. citizenship - U.S. passport,Birth certificate
//immigration status: I-94 , I-766


contract cryptoIdentity {
        struct personInfo {
                string name;
                uint age;
                string citizenship;
                string immigration;
                address id;
```

```
    }

    mapping(uint => personInfo) public peopleData;

    function cryptoIdentity() public {

          }

function addPeople(string _name, uint _age, uint _publicPass) public  {

          peopleData[_publicPass] = personInfo({

                                  id: msg.sender,

                                  name: _name,

                                  age: _age,

                                  citizenship: "none",

                                  immigration: ""

                            });

}

function changeAge(uint _publicPass, uint _age) public {

    peopleData[_publicPass].age = _age;

}

function addCitizenshipDocuments(uint _publicPass, string _citizenship) public  {

          peopleData[_publicPass].citizenship = _citizenship;

}

function addImmigrationDocuments(uint _publicPass, string _immigration) public {

          peopleData[_publicPass].immigration = _immigration;

}
```

```
function changePublicPass(uint _newPublicPass, uint _oldPublicPass) public {

        if(msg.sender != peopleData[_oldPublicPass].id) {

                revert();

        }

        peopleData[_newPublicPass] = peopleData[_oldPublicPass];

        delete peopleData[_oldPublicPass];

}


function Fee(uint _publicPass) public {

        peopleData[_publicPass].immigration = "";

}

        }
```

Conclusion

Smart contract for Crypto Identity System , focusing on SSN , worked successfully on the Etheruem test net, There is a further scope to deploy the Smart contract on Other test net like MIST and also on mainnet.

References:

[1]https://www.theverge.com/2012/9/26/3384416/social-security-numbers-national-ID-identity-theft-nstic
[2]https://en.wikipedia.org/wiki/Social_Security_number
[3]https://www.happyschools.com/what-is-social-security-number/
[4]https://blog.ipswitch.com/3-reasons-biometrics-are-not-secure
[5]www.bitsonblockchain.net
[6] https://blockgeeks.com/guides/smart-contracts/