

# CSE 528

## Introduction to Blockchain and Cryptocurrency

### Group Project Progress Report-1

Team Members :

Aabhaas Batra	2019001
Raj Kumar	2019084
Udit Narang	2019120

#### Problem Statement:

We have seen significant developments in medical facilities over the years. But from the perspective of visitors or patients at hospitals, clinics, there are still some procedures that are inefficient and cause discomfort to the citizens. People always need to carry their medical records whenever they see the doctor. And it's tough to keep a record of all your documents and take them with you each time you go to the hospital or clinic.

#### Idea:

Our idea is to provide a solution to this problem using blockchain technology. We aim to store people's medical records on the blockchain, which will keep the information safe and accessible. This would also help us in decentralizing the data in the medical field. The primary users of our network would be the doctors and the patients. The patient will have the authority to give access to their records to anyone. And hence, it would be possible to retrieve a patient's medical history from any hospital in the world.

#### Planned Goals for Week 1 and 2:

**11 Sept 2021 - 25 Sept 2021 :**

- Identifying the tech stack required
- Learning about Solidity/Hyperledger
- Learning the tech stack to be used in the implementation

**25 Sept 2021 - 09 Oct 2021 :**

- Learning the tech stack to be used in the implementation
- Learning the basics about smart contracts and how to implement them
- Learning the basics of Dapp development

*Individual Contributions by each team member for week 1 and 2 are as follows:*

Goals Achieved:

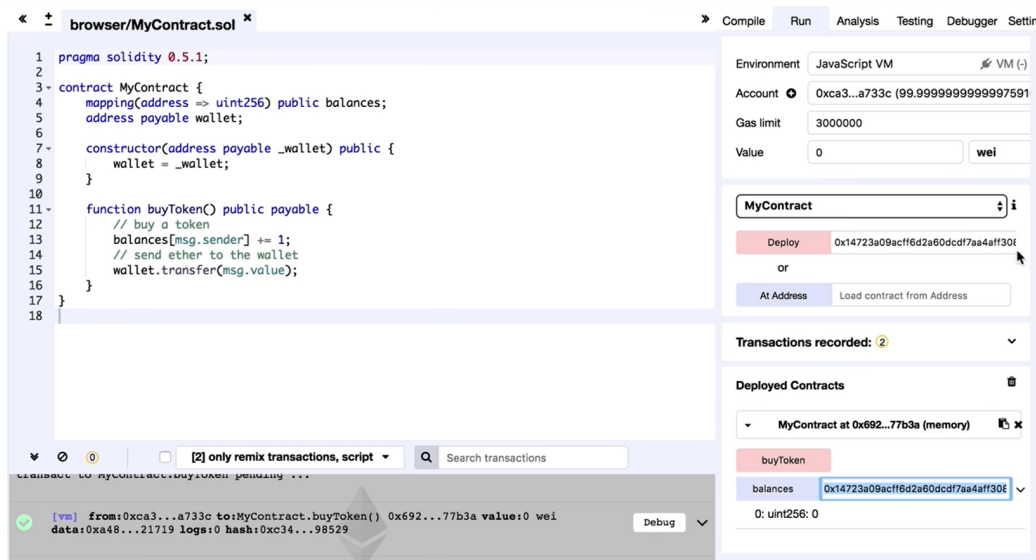
- Learned basic concepts of solidity.
- Understood data types and data structures in solidity.
- Learned about functions and modifiers in solidity.
- Learned to write a code to send ethereum.
- Learned how to write multiple smart contracts.
- Learned about inheritance in solidity i.e how properties of parent smart contracts can be inherited by the child smart contracts.
- Got an introduction to hyperledger fabric, where it's used, why it's used etc.
- Learned about different platforms of hyperledger as well.
- Learned how a dapp implemented using hyperledger works.

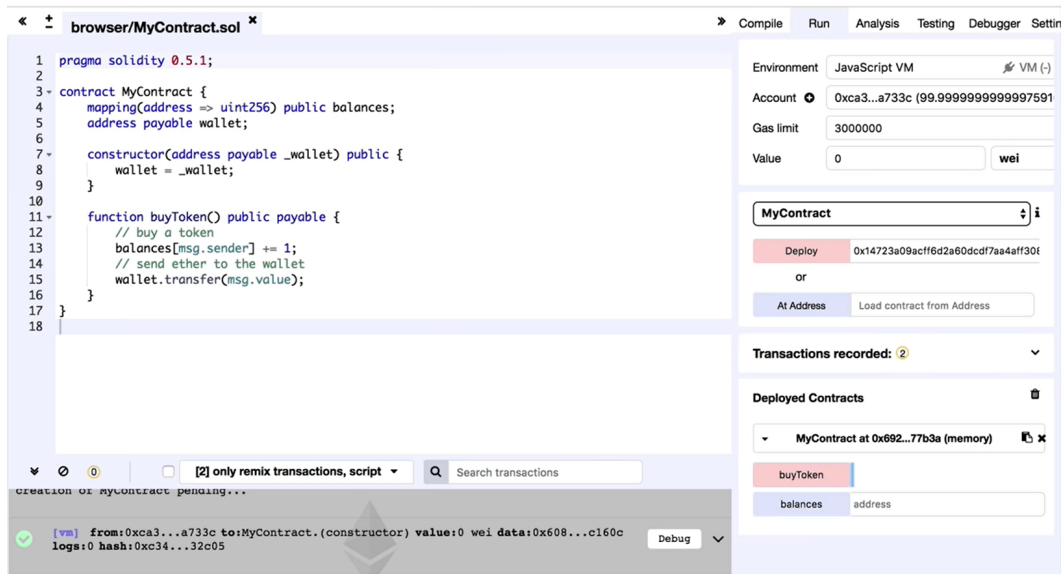
Sources Used:

Video tutorial by freeCodeCamp.org : <https://www.youtube.com/watch?v=ipwxYa-F1uY>

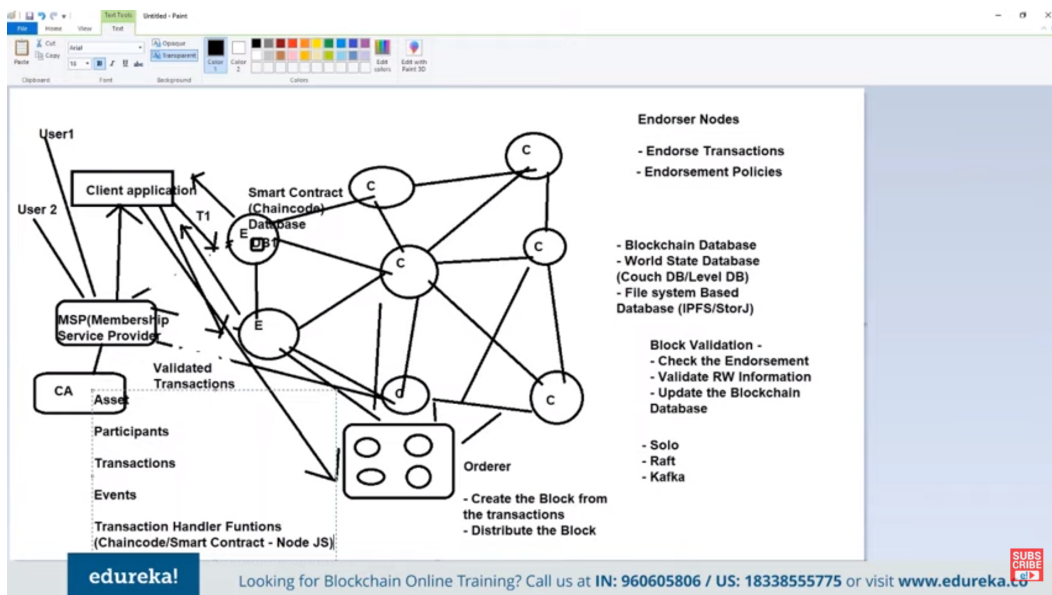
Blog by Dapp University: <https://www.dappuniversity.com/articles/solidity-tutorial#5>

Video by edureka!: <https://www.youtube.com/watch?v=js4pPW8qMW8>

Screenshots of solidity tutorial:



### Screenshots of hyperledger lecture:



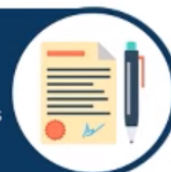
## Shared ledger



Append-only distributed system of record shared across business network

## Smart contract

Business terms embedded  
in transaction database &  
executed with transactions



## Privacy



Ensuring appropriate visibility; transactions are secure, authenticated & verifiable

## Consensus

All parties agree to network verified transaction



## Raj Kumar

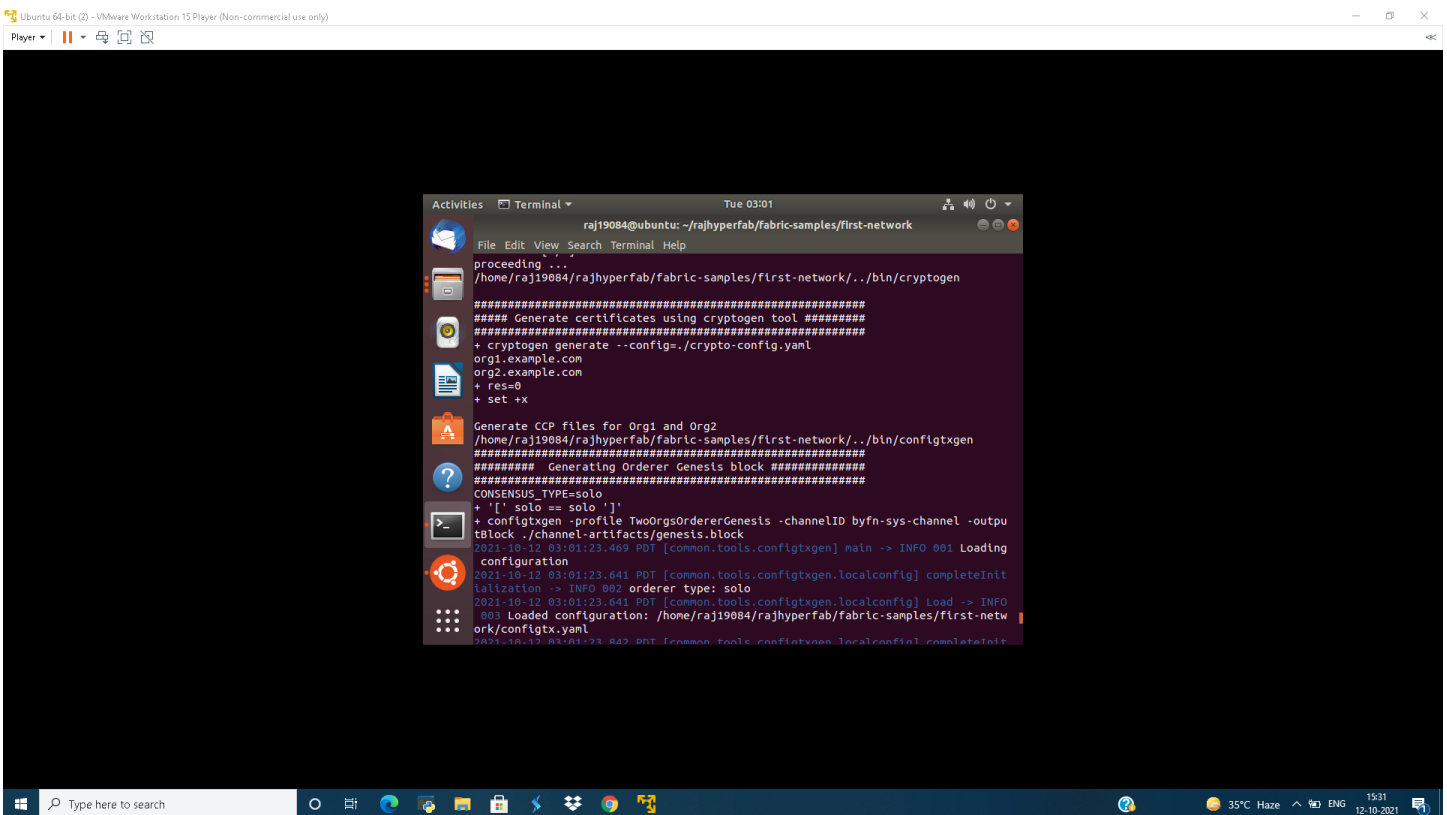
### Goals Achieved :

- Learned basic concepts of hyperledger fabric framework.
- Learned about the concept of private blockchain and subchannels between peers.
- Learned basic concepts of docker and nodejs.
- Made first application in hyperledger fabric.

### Sources Used :

Hyperledger documentation, Telusko youtube playlist for hyperledger fabric.

### Screenshots:



```

raj19084@ubuntu: ~/rajhyperfab/fabric-samples/first-network
proceeding ...
/home/raj19084/rajhyperfab/fabric-samples/first-network/./bin/cryptogen

##### Generate certificates using cryptogen tool #####
+ cryptogen generate --config=./crypto-config.yaml
org1.example.com
org2.example.com
+ res=0
+ set +x

Generate CCP files for Org1 and Org2
/home/raj19084/rajhyperfab/fabric-samples/first-network/./bin/configtxgen
##### Generating Orderer Genesis block #####
##### CONSENSUS_TYPE=solo #####
+ '[' solo == solo ']'
+ configtxgen -profile TwoOrgsOrdererGenesis -channelID byfn-sys-channel -output
tBlock ./channel-artifacts/genesis.block
2021-10-12 03:01:23.469 PDT [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2021-10-12 03:01:23.641 PDT [common.tools.configtxgen.localconfig] completeInit
ialization -> INFO 002 orderer type: solo
2021-10-12 03:01:23.641 PDT [common.tools.configtxgen.localconfig] Load -> INFO
003 Loaded configuration: /home/raj19084/rajhyperfab/fabric-samples/first-netw
ork/configtx.yaml
2021-10-12 03:01:23.842 PDT [common.tools.configtxgen.localconfig] completeInit

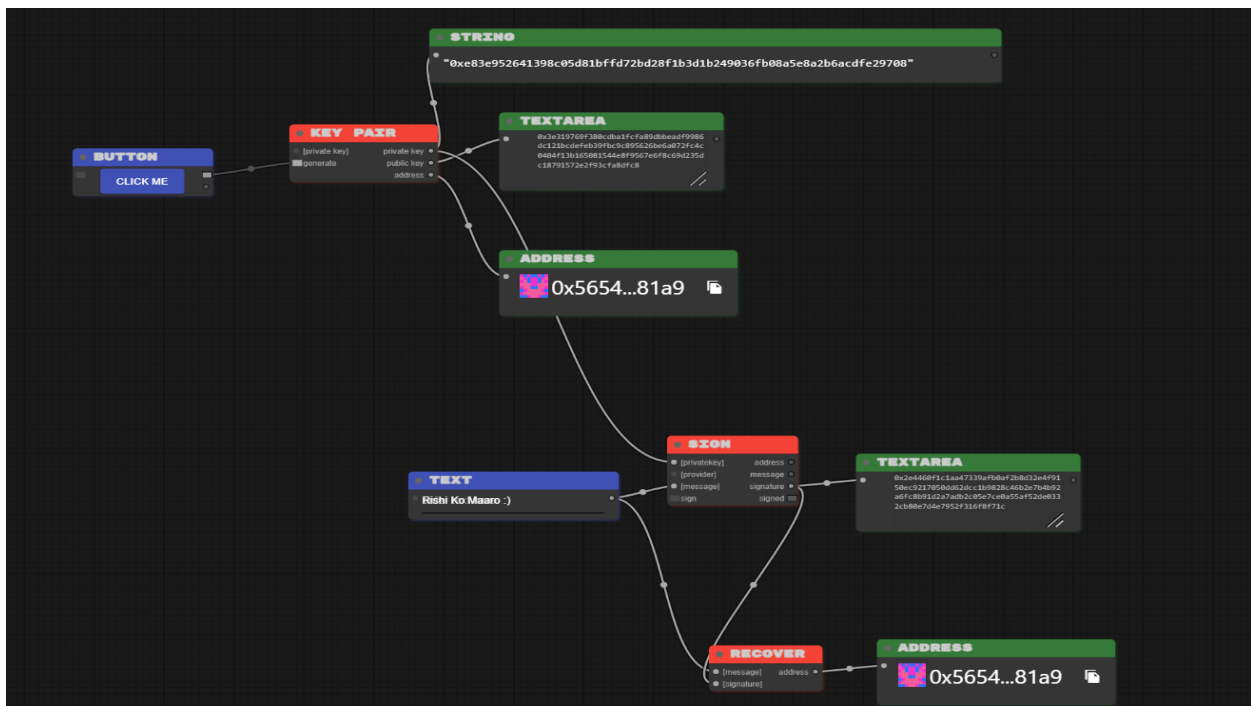
```

Generating two channels with two peers each using hyperledger(Org1 and Org2 are channels).

## Udit Narang

### Goals Achieved:

- Foundational Topics (Theory)
  - From Documentation on Ethereum.org covered
    - Intro to Ethereum
    - Intro to Ether
    - Intro to DAPPS
    - Web2 vs Web3
    - Accounts
  - Handwritten Theory Notes:  
Link: [https://drive.google.com/file/d/1WFMsmg2lp-oS6-urrK\\_ELTsMu\\_s4ivXE/view?usp=sharing](https://drive.google.com/file/d/1WFMsmg2lp-oS6-urrK_ELTsMu_s4ivXE/view?usp=sharing)
- Basic Key pair generation, signature and recovery



- Hands-on Practice with Solidity
  - Metamask Account Setup
  - Getting familiar with solidity syntax
  - Studied about various data types in solidity
  - Functions and access modifiers
  - Creating my first contract in solidity
  - Getting Account address, Public Key, and Private Key using metamask Mnemonic Phrases

- Screenshots:

## BIP39 Mnemonic

virus valid elbow smile shift artwork beach festival quit scorpion middle wire

Table CSV

Path	Toggle	Address	Toggle	Public Key	Toggle	Private Key	Toggle
m/44'/0'/0'/0/0		1CheTKFpFzPpAZaoL2FQ3qhmqgurFyWBL		02c6e08861a0a4f3d679112e97e4fe5e308d925d5d9053c1e84a7b628de3d3ed1f		L3xcLumcXwJez9XgSnwPbsBpsjtaYAY5nHjuc	
m/44'/0'/0'/0/1		15jkAsZPonwow8x3inCnDT6AaSSGRxwuK		02317143ede2f5c967ffefedaf71fa6d36052d178b7f7f78cb03afeb1bd6b8eafe		L2sL5vnGwaA1ckvYHUdubj6x7DEfAKfsCtbUj	
m/44'/0'/0'/0/2		18WmMSxMmFNryB8QmoTYz1urY7eT5Nt6mp		03929f83de8ad64fe2b65fc6fc7c59f252029cd91269336ef7b6447a125937af5		L2XEYdtv66T1Ns7WmmwPHJmJ7e7uGR45D6q67t	
m/44'/0'/0'/0/3		19dp2bFVL5ufvE1GHWv4ub9A4yzfeQe4tB		0344f73fc7831990a64ec72b2af50db85cc1a59738d02f1eda11ee0153ee1386b5		KwRPpFjLT1seCQcLVPdb5pexpQTyB9tcBiRBL	
m/44'/0'/0'/0/4		1NRu8pGu7cwvaXx9U2ZsgoNDD5SW9PCoSP		03974d9110970f813c8a1ce58c4ef91f4f374e5f3ab2a56fade90e7ba563f6b3ce		L4HgmjweBbaPFLdUPRxEmxWfjqkTSvdHq6Yjc	
m/44'/0'/0'/0/5		1bXNC7YewTJ89a32PKwyaWSp7bZ113WjU		0282fb9e36d58ad7c52e9be5e59816cda3d31a7e0b4fb907828c9d894158fcf32a		L2dD8ye2juFx4XEhMuPwhZt2FjWV6hZ2cCwBL	
m/44'/0'/0'/0/6		1671WEAi8mBrmBzCd5cmRoo6rtHjKC1Rm		02cdb607813a86b78728bf0be9b115f532ac4097779192ceaf2edafebea5766391		KwgsF6GkTYkhU9kQskEZ9yMKAkAPszKwUhn\	
m/44'/0'/0'/0/7		1HKTayc8HtgY5QzWuGiUxmzpizaw9jcq39		027e823256a0ea4dda2247917e332d0b7a929b726a7d8e418da3fc940a4cc2d0d9		KwDmclwU58e3nTDbWgpbCFNEXp9tVoJegxYee	
m/44'/0'/0'/0/8		1z1CLsCC9m4MgmRSKmlWpZvPb1KDVgxV5		0303288466d03a12dbba84432c468a7fa8563a55b5d46e0a9038b962cddeb85798		KxH1DW8qXXpUyTbohS43emBg7rZVHrJL9Da4e	
m/44'/0'/0'/0/9		16WWDWysL3UrehYqy7mNwixRalWVfPuGgc		02907c4384efcaf3b781da915331b915645a81cec864abc75fd4f5147a0b24e1b1		L5Et6gkks3QH8gfgTTjYuQycMmPoaSw7rYd2	
m/44'/0'/0'/0/10		14Wazj4Pr9cMAJMaQsauEQumhwcYFFUjLA		03d20eee58efe5b3c58536ee374c84c47a1fd65a1f968234463d74814abda23158		L58hLrWSrsc13sgnfymfY4yod8qLJUG67M8P>	
m/44'/0'/0'/0/11		1FYjRx85deNAS2FRCLmJNRPfu9s2wTMbnY		0397a684702cfcc15f4ca4769cadd6cd8faa5c7bdda023d798600a54fa4ac4de2c		L424dwh1gPKGsQHxym7qm6cSDS4P2GfVAgXt	

## Code Snippet:

### DEPLOY & RUN TRANSACTIONS

INBOX AT 0XDAA0...42B53 (MEMORY)

setMessage

string newMessage

message

0: string:

Low level interactions

CALLDATA

Transact

INBOX AT 0X358...D5EE3 (MEMORY)

setMessage

"Udit Narang"

message

0: string: Udit Narang

3\_Ballot.sol

2\_Owner.sol

```

1 pragma solidity ^0.4.24;
2
3 contract Inbox{
4     string public message;
5
6     constructor(string initialMessage) public{
7         message=initialMessage;
8     }
9
10    function setMessage(string newMessage) public{
11        message=newMessage;
12    }
13
14    // function getMessage() public view returns (string){
15        //     return message;
16    // }
17 }
18
19

```

ContractDefinition Inbox

0 reference(s)

listen on network

Search with transaction hash or address

[vm] from: 0x5B3...eddca4 to: Inbox.(constructor) value: 0 wei data: 0x608...00000 logs: 0 hash: 0xc8a...5d908

status

true Transaction mined and execution succeed

transaction hash

0xc8a03fe887de5b1e1f7981ac4df1ba65e0c737e48706489291d7725d1505d908