

# Solving Privacy and Security issues in Electronic Health Record (EHR) using Blockchain

## ***Abstract:***

The advantages of electronic health records in the clinical setting are numerous and important. In the 2012 edition of the Physician Sentiment Index™ More than two-thirds said an EHR can actually improve patient care. Poor EHR system design and improper use can cause EHR-related errors that jeopardize the integrity of the information in the EHR, leading to errors that endanger patient safety or decrease the quality of care and EHR systems faces problems regarding data security, privacy issues and information asymmetry. We have leveraged blockchain technology to transform the EHR systems and could be a solution for the above issues.

## ***Problems in existing EHR system:***

- ***Interoperability:*** It is the way for different information systems to exchange information between them.
- ***Information Asymmetry:*** which refers to one party having better access to information than the other party.
- ***Data Breaches:*** Data breaches in healthcare sector also calls for the need of a better platform.

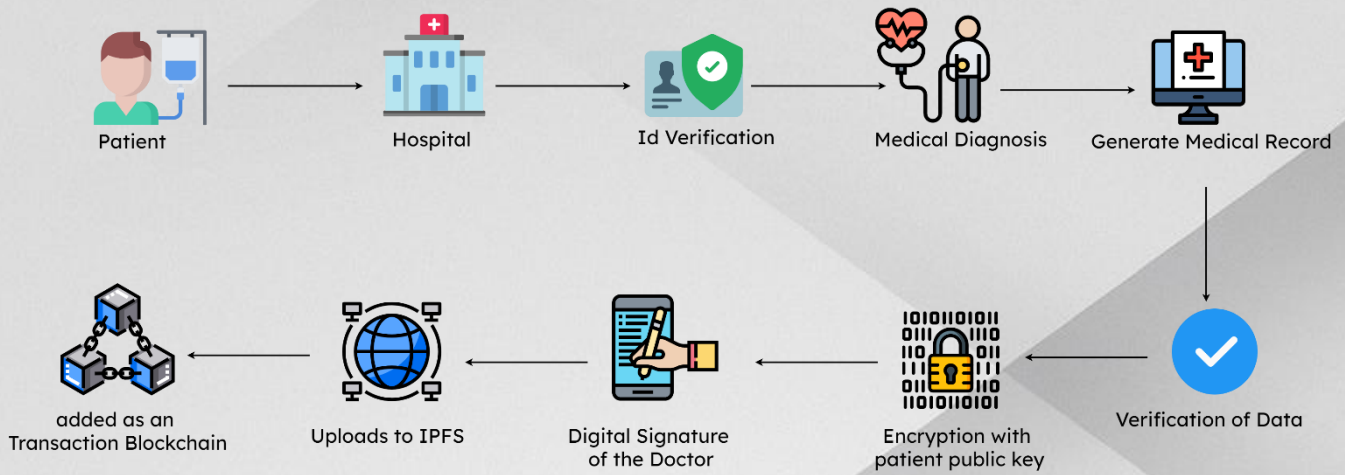
## **Solution:**

- We present a technique where we've leveraged the blockchain technology to transform the EHR systems and could be a solution for the above issues.
- The aim of our solution is firstly to implement blockchain technology for EHR and to give back the control of the Medical data to the patients. Moreover, we've also presented a solution for the scalability problem faced by the blockchain using *off-chain decentralized storage* for storing the EHR data.
- Another main issue in the traditional framework is, our EHR data is *accessed and shared* with unauthorized Personnel.
- Hence, we have designed a framework for sharing of EHR which is *solely controlled by the patient* to whom the EHR data belongs to and in order to keep the track of old EHR's we are going to implement git like, version control system on top of the IPFS.
- Data in our system can only be accessed by the patient's private key, even if someone obtains data from the decentralized storage, data will be unreadable and patients have full control over accessing their healthcare data, patients will control who sees their data and what they see. Hence, there's a reduced risk of errors, and better patient care. We believe, that the above solution will definitely improve our current Healthcare industry
- We have ensured decentralization in every flow of the system. Hence, there is *no single point of failure*.

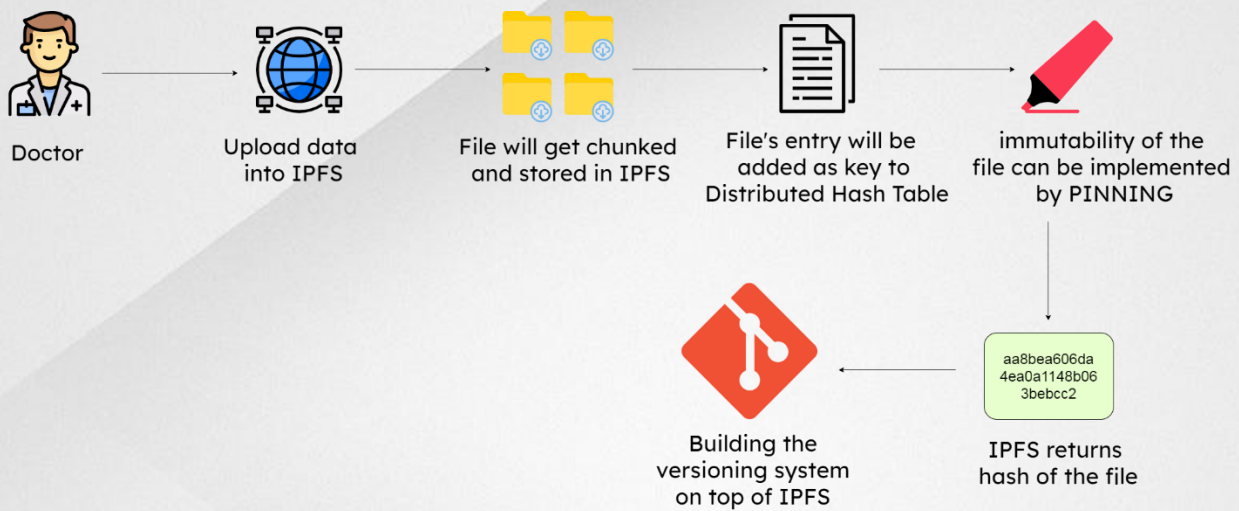
## **Detailed workflow:**

<https://prezi.com/q9x25xhgmsrv/ehr-implementation-using-blockchain/>

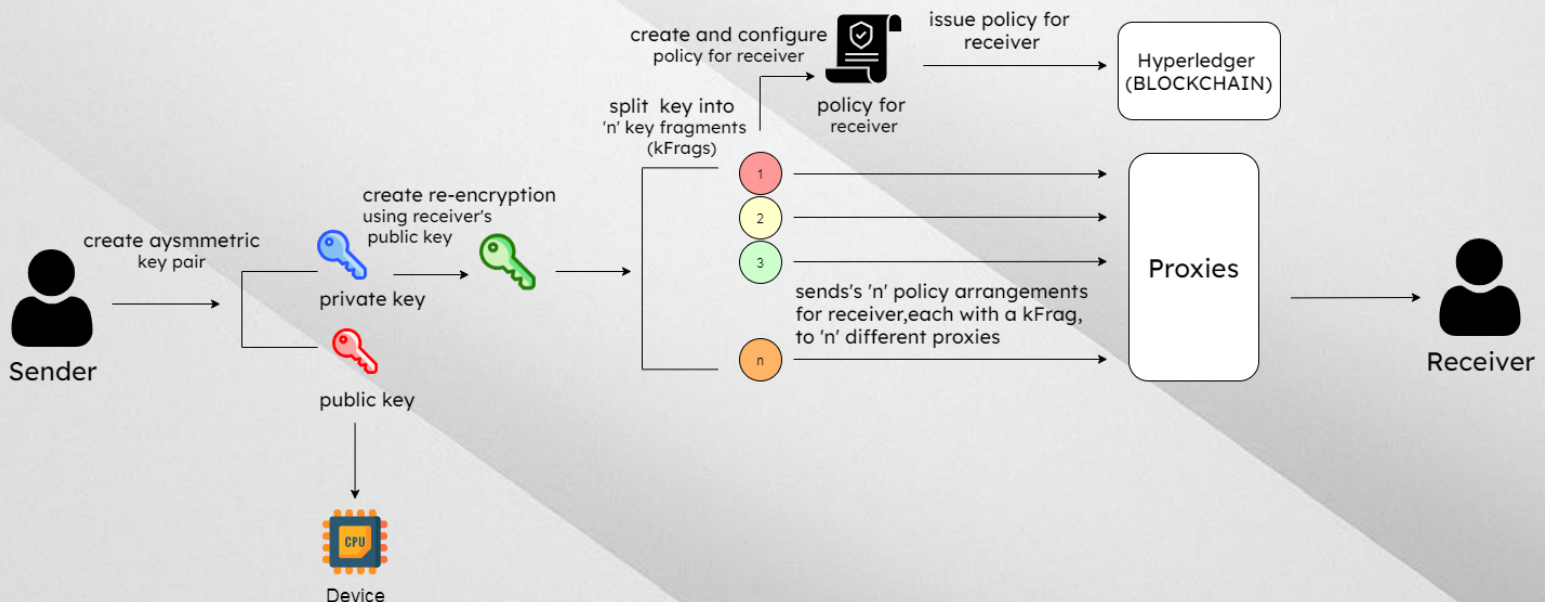
# Overall System workflow



## How data gets stored in our system?



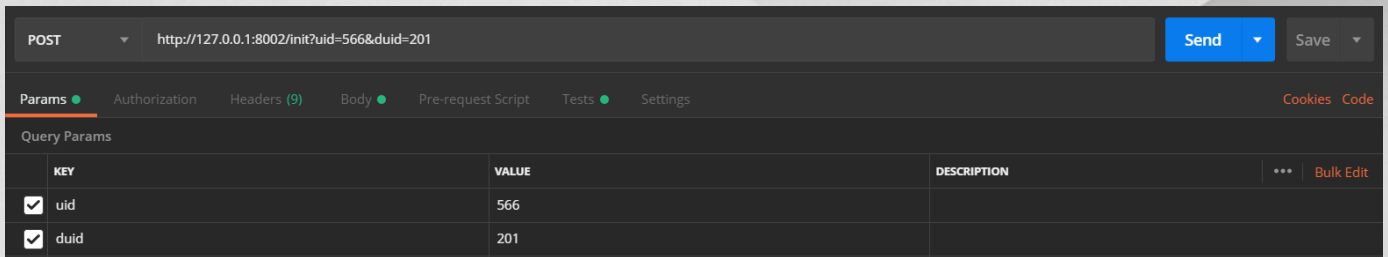
## How our data sharing mechanism works?





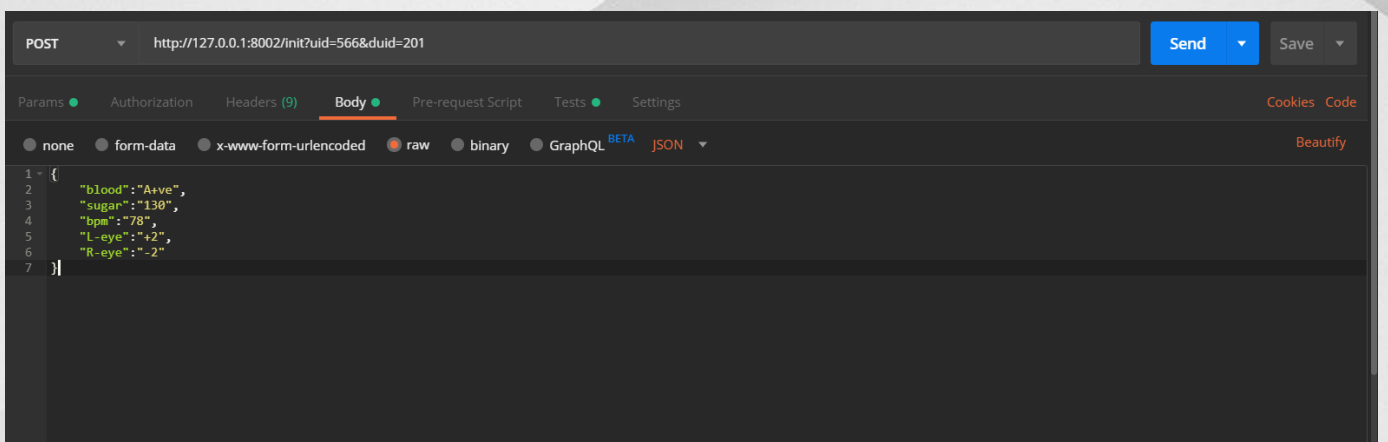
## Documentation and Screenshots:

### BACKEND:



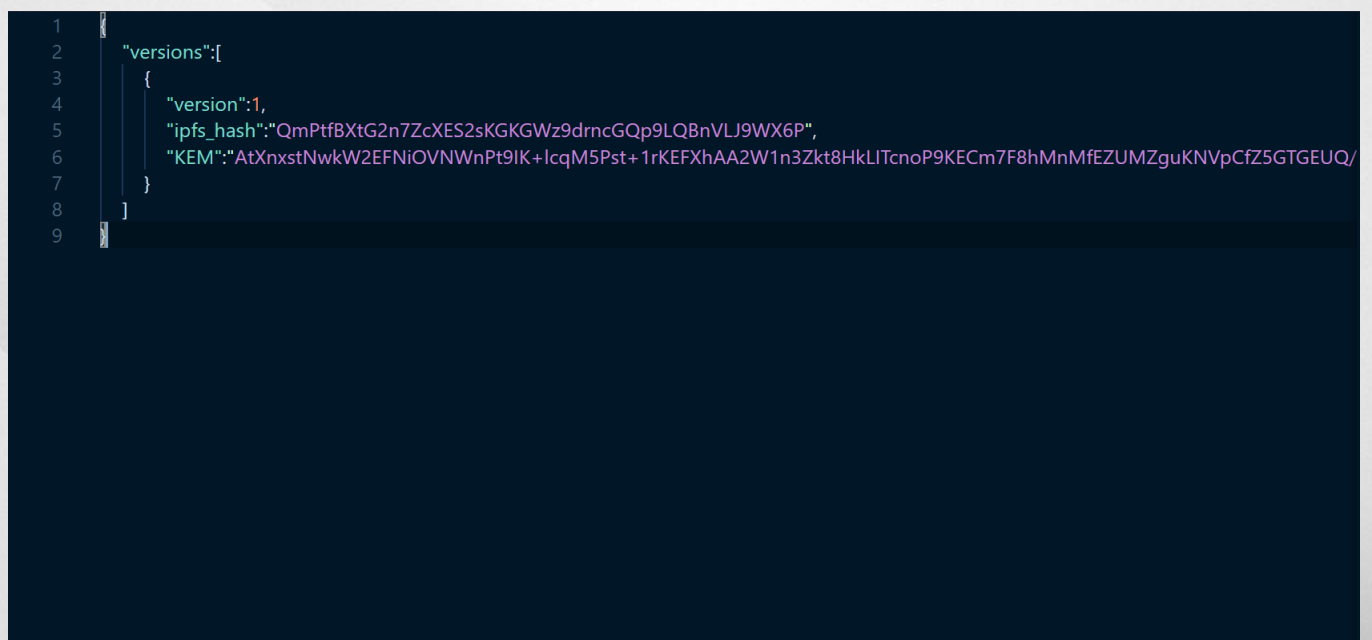
KEY	VALUE	DESCRIPTION
uid	566	
duid	201	

*Generating a new key pair for patient and doctor*



```
1 {
2   "blood": "A+ve",
3   "sugar": "130",
4   "bpm": "78",
5   "L-eye": "+2",
6   "R-eye": "-2"
7 }
```

*Sample EHR data*



```
1
2   "versions": [
3     {
4       "version": 1,
5       "ipfs_hash": "QmPtFBXtG2n7ZcXES2sKGKGWz9drncGQp9LQBnVLJ9WX6P",
6       "KEM": "AtXnxstNwkW2EFNiOVNWnPt9IK+IcqM5Pst+1rKEFXhAA2W1n3Zkt8HkLITcnoP9KECm7F8hMnMfEZUMZguKNVpCfZ5GTGEUQ/"
7     }
8   ]
9
```

*Version holder of a patient*

```
"8998": "QmadjQNj1BaYaaxzVQJgRyBQBT3APNboo4Xg3tWeQFX2th",
"999": "QmRoZNhCvG85rz8jGryvyRaGVEKDZ59LEKU9Ji7hmeenD9",
"500": "QmejTr9MaFYzdHq3SzqYghTUPPE23QUdYL4YboEGCkcSJ",
"100": "QmNxpLhPQeAKuDEarV8f3aSpMnaAvIT8uo7w6F9BYNHLq",
"997": "QmcUwg66VzR3ufiAkGM5kp773tAvW6YobvFKfWNxhhi22",
"222": "QmRcBLZSbeUCXfuaXAMFWeCSokJ64SDmn3SHsD997NmwrS",
"10000": "QmdG8W9eCPB76kcsQdBYJwF33EgHRUArpAH25ZioZ4SREM",
"400": "QmSJkuTN2qevDu6dc7yr3vt8pWk3oF8UC4LKk8np5mKpDe",
"1001": "QmQfWRqzNCxxJKm6UstzcU4FEy9QC4JE6rLuhuhQ3Z4Beh",
"123": "QmQp5VsKdzWPrFgBRFWFVsy1L9hxVE47PoZBA3XzyQJcCb",
"566": "QmTXu5NMZUkq4ZqjRnfBygLVQA9j7U6UPvQA9sDJ3pR14Q"
```

*EHR hash link of few patient (id:hash\_link)*

```
GET http://127.0.0.1:8002/mine Send
Pretty Raw Preview Visualize BETA JSON
1 {
2   "index": 7,
3   "message": "New Block Added",
4   "previous_hash": "f438b29730d9b90b6d443a5771380d0c50ec00c0c1d6becd06d1ddef3b272b9a",
5   "proof": 65350,
6   "transactions": [
7     {
8       "UUID": "8998",
9       "digital_sign_doctor": "MEUCIGEKdYphgFP9KbbIRoFedKVZ3wXMG1BCF-fX6HjXE+Pw5AiEAq1k1Hf8mJV7fQwJkMYnU0fAVM2abN2pJygorI87b4nM=",
10      "digital_sign_patient": "MEUCIQCVjb6ZPCsQRBKESqurXZ7nmSEHQ8UPEkyPTTzRqE91AIGKnUH7Hh/ozB04nCVm1bUb8H201bGhmUAjh7muvE3nVU=",
11      "ipfs_file_hash": "QmadjQNj1BaYaaxzVQJgRyBQBT3APNboo4Xg3tWeQFX2th",
12      "modification": "true",
13      "public_key_doctor": "AraPh1UxnY+B0rnnDtnnLmpDS5yXfotaxBHuvS681ERx7",
14      "public_key_patient": "AnmEKNTbhVJU3r7OI/xuN1C1BEW8pST1pX8K5d0fR508",
15      "time": "2020-01-21 04:17:18.291768",
16      "version_holder": "QmYadv7F2ndvqyhPEABLvK4Hutv4sXzmp8PYgJS52bAQLth"
17    },
18    {
19      "UUID": "566",
20      "digital_sign_doctor": "MEYCIQC+1XZYseDbEs8LQI1hbsBDjCKZ6S5r+kVhXps+1FUMC1QIhANZtyg3N2djFRJVCUDnp2tshSPITQzKSWrD/G1rLq15E",
21      "digital_sign_patient": "MEUCIh2qhsUAGMc2djK5yQ8/fghtRy/YBq9Zzg+vAUzUJCN+AiBVwO4+JDenqdMdSj/a41wgnFx7dp517a86U6grgCSKkQ=",
22      "ipfs_file_hash": "QmPtFBXtG2n7ZcXES2sKGKGWz9drncGQp9LQ8nVL39XG6P",
23      "modification": "false",
24      "public_key_doctor": "Ap6t745Ko3S5mYRAAtY2/1vP116V8vwuBTVM48g+0y80c",
25      "public_key_patient": "Ap6KXaIFEOxMw6JCfQLVhGqKJzVLSGs5EH4dIQjKUMs6",
26      "time": "2020-01-29 19:50:30.979582",
27      "version_holder": "QmcnWkQrR7eY3s19BswjE9wzRPXG636S5ziXa9M7tdzf4S"
28    }
29  ]
30 }
```

*Mining*

```

File "C:\Users\Nandu\Anaconda3\lib\sit
(base) C:\Users\Nandu\Desktop\projects\m
Enter port no : 800
* Serving Flask app "main_blockchain" (
(base) C:\Users\Nandu\Desktop\projects\m
Enter port no : 8002
* Serving Flask app "main_blockchain" (
* Environment: production
  WARNING: This is a development server
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:8002/ (Pr
checking
Generating key pair
checking
200_doc key pair exists and is readable
127.0.0.1 - - [29/Jan/2020 19:44:56] "GE
New Transaction initiated....
Encryption in progress
uploading.... to ipfs
hash of the file : QmQp5VsKdzWPrFgBRFWFV
hash_indices.json exists and is readable
updating.... hash indices
ipns hash=
[2020-01-29 19:45:20,595] ERROR in app:
Traceback (most recent call last):

```

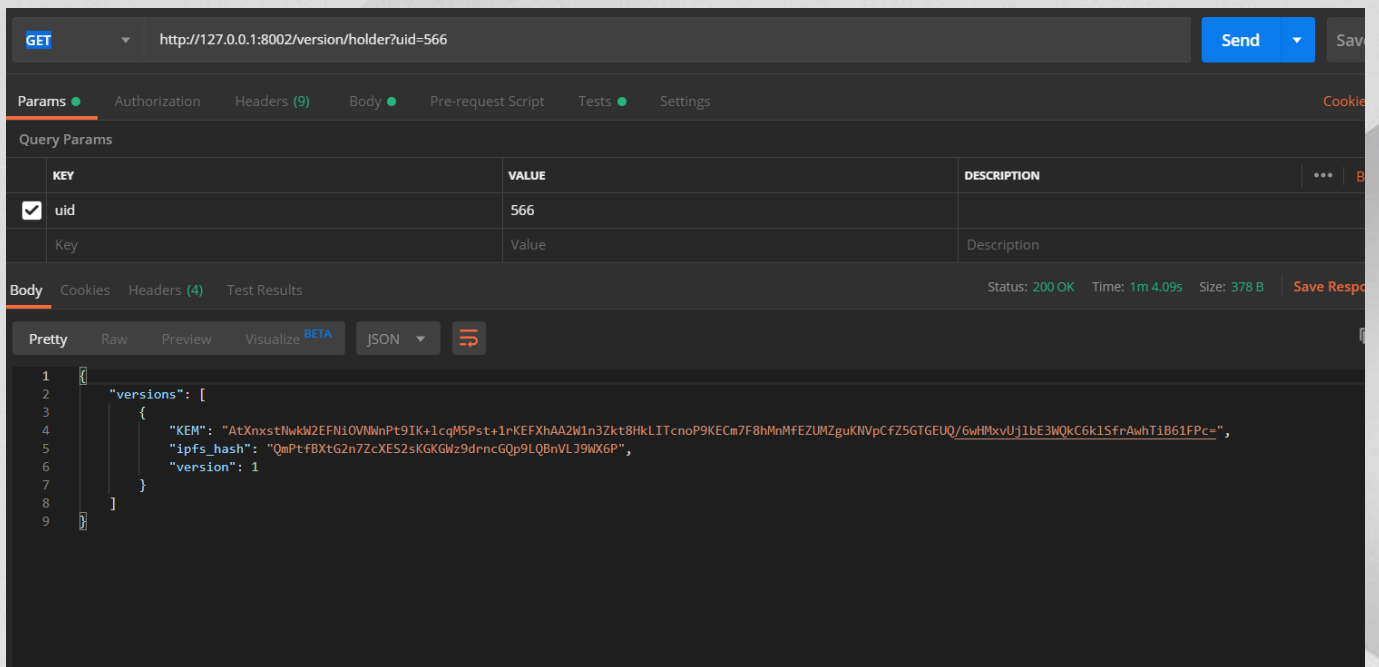
## Background logs-1

```

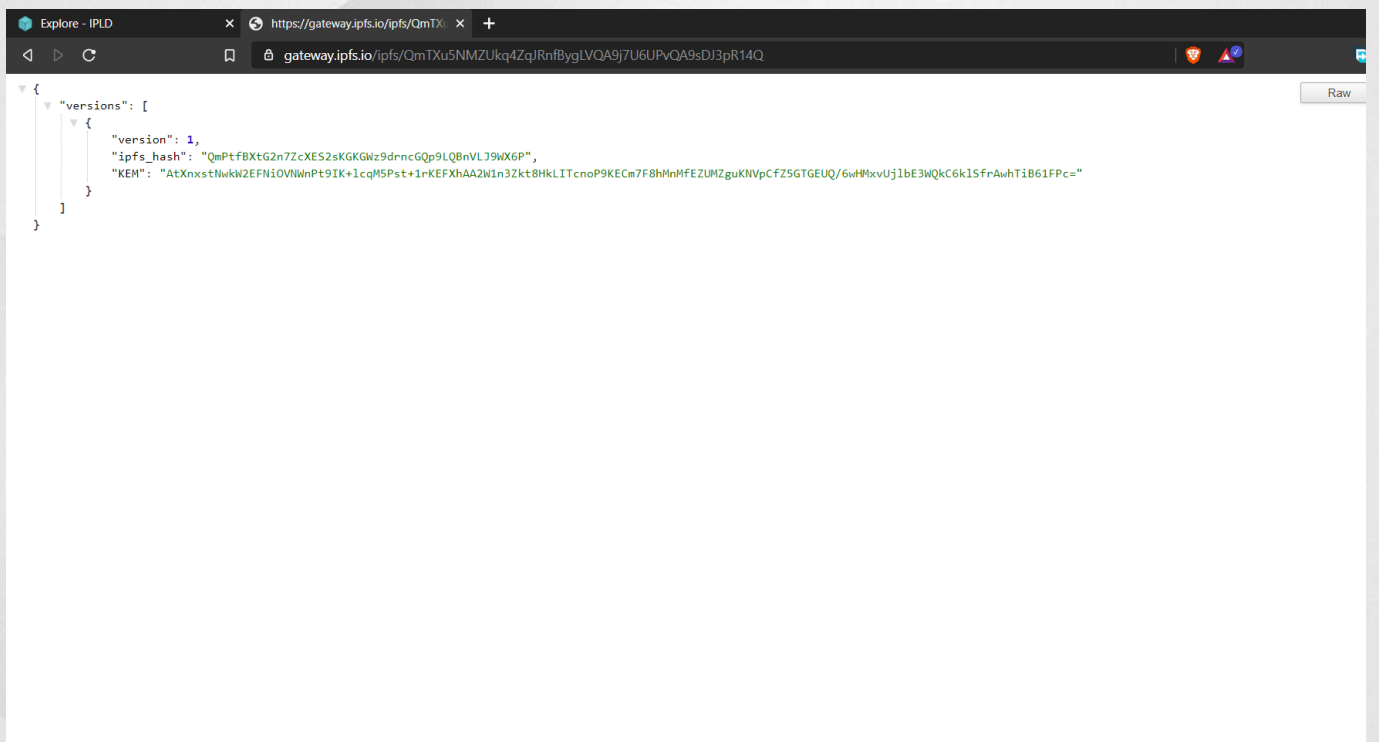
Generating key pair
127.0.0.1 - - [29/Jan/2020 19:48:29] "PO
New Transaction initiated....
Encryption in progress
uploading.... to ipfs
hash of the file : QmPtFBXtG2n7ZcXES2sKG
hash_indices.json exists and is readable
updating.... hash indices
ipns_hash_indices.json exists and is rea
updating.... ipns hash indices
ipns hash=
566_version_holder.json exists and is re
uploading.... to ipfs
hash of the file : QmTXu5NMZUkq4ZqJRnfBy
hash_indices.json exists and is readable
updating.... hash indices
The static ipns hash :
ipns key(version holder) and newly uploa
Signing the File
Signing the File
UmbralPublicKey:029eadeef8e4aa37
UmbralPublicKey:029e8a5da88510e
127.0.0.1 - - [29/Jan/2020 19:50:30] "POST /transactions/new HTTP/1.1" 201 -
127.0.0.1 - - [29/Jan/2020 19:55:18] "POST /mine HTTP/1.1" 405 -
127.0.0.1 - - [29/Jan/2020 19:55:21] "GET /mine HTTP/1.1" 200 -
checking version holder data
127.0.0.1 - - [29/Jan/2020 19:57:07] "GET /version/holder?uid=566 HTTP/1.1" 200 -

```

## Background logs-2



*Requesting the version holder of a patient*



*Viewing our data from the real IPFS endpoint*

```
GET http://127.0.0.1:8002/chain
Send Save

Pretty Raw Preview Visualize BETA JSON

1 {
2   "chain": [
3     {
4       "index": 1,
5       "previous_hash": "1",
6       "proof": 100,
7       "timestamp": 1579537881.4240881,
8       "transactions": []
9     },
10    {
11      "index": 2,
12      "previous_hash": "a7947e612ee820284ee746519579349591d76adc0df703df58c26ef0b67f6201",
13      "proof": 26698,
14      "timestamp": 1579538018.1222563,
15      "transactions": [
16        {
17          "UUID": "878",
18          "digital_sign_doctor": "MEUCIAYA4LPM/V+ItGznVnPV5HvBSOATZrK+10jDMhPO/1MvA1EA6u7hFT8sVDt5FP1DrUIEJ6+GDog8Ysq1sndvQy+sm9g=",
19          "digital_sign_patient": "MEUCIQD37FotELqnpMz9c9qHfo77wqv1AhOthrrcF9LzU6ZW9Q1gAp1cQFpAuoqIiRoHCSCjG5v1yZfh1nAxSgAm7951HFU-",
20          "ipfs_file_hash": "QmW2ized81dmrKCawsMzh4TzEgWPyQ33NuL9kJN1TRTAQJ",
21          "modification": "false",
22          "public_key_doctor": "AuCnHNB0u2QbJ39YCs6b8605ZmzAjiap8xkhozgoq8zi",
23          "public_key_patient": "AjrPUT7pG7g6f24tcrPnIN5m6ey7YKKEJdd1JNDUzyZ",
24          "time": "2020-01-20 22:03:31.688160",
25          "version_holder": "QmT2XyVYij4d1zUP8wDed6Shnmvdqsu4wnAYws991zLNDq"
26        }
27      ]
28    },
29    {
30      "index": 3,
31      "previous_hash": "1a04eFcd14449393f36780c1e02a4c9b8b4685606467f0f9f8bcaedd7a65b00d",
32      "proof": 8919,
33      "timestamp": 1579538018.1222563
34    }
35  ]
36 }
```

## Blockchain data of a single node

```
POST http://127.0.0.1:8002/share/new_policy
Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON

1 {
2   "s_uid": "566",
3   "n_uid": "8998",
4   "data_hash": "QmPtFbXtG2n77cXES2sKGGKwz9drncCQp9LQbNVLJ9Wx6P",
5   "public_key_digital_sign": "AuCnHNB0u2QbJ39YCs6b8605ZmzAjiap8xkhozgoq8zi",
6   "capsule": "AtXnxstlWkZEFNiOVMWnPt9IK+1cqM5Pst+1rKEFXhAA2W1n3Zkt8HklL1CnoP9KEcm7F8hMnMFEZUMZguKNVpCfZ5GTGEUQ/6wHxvUj1bE3WQkC6kKLSfrAwHt1B61FPc",
7   "remove": "0"
8 }

Body Cookies Headers (4) Test Results Status: 200 OK Time: 15ms Size: 207 B Save Response

Pretty Raw Preview Visualize BETA JSON

1 {
2   "message": "New policy has been added !",
3   "status": "success"
4 }
```

## Creating a new policy for data sharing

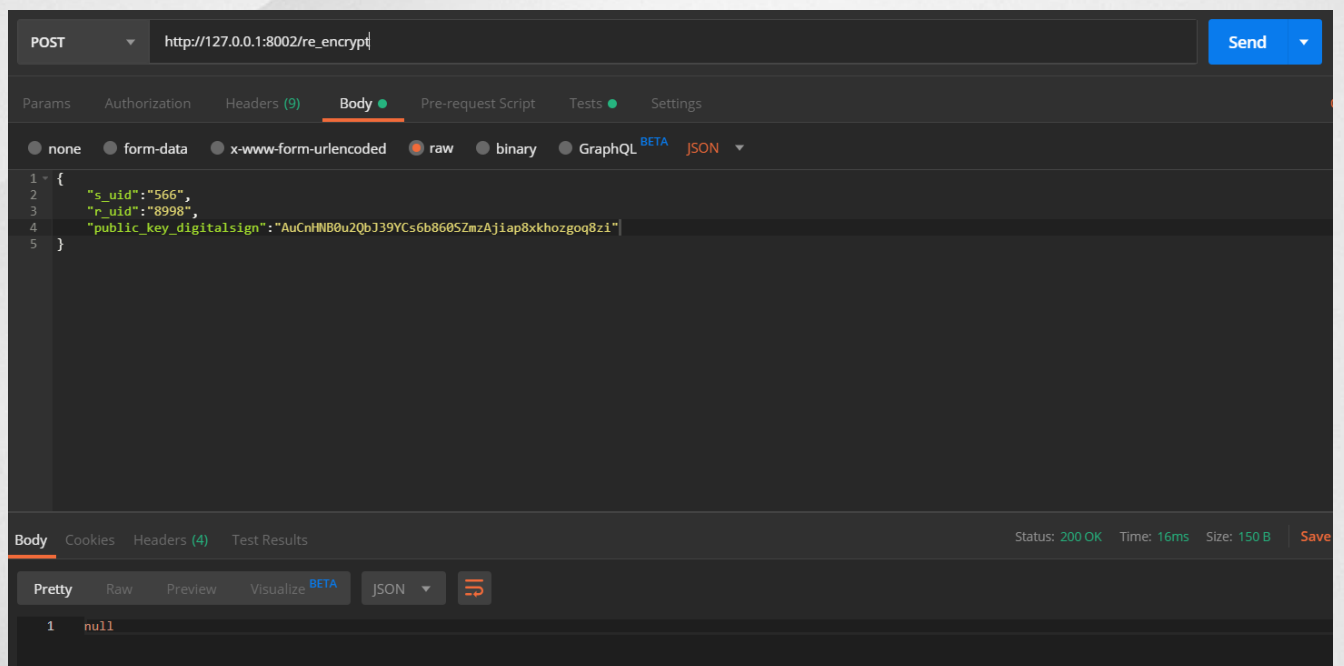


```

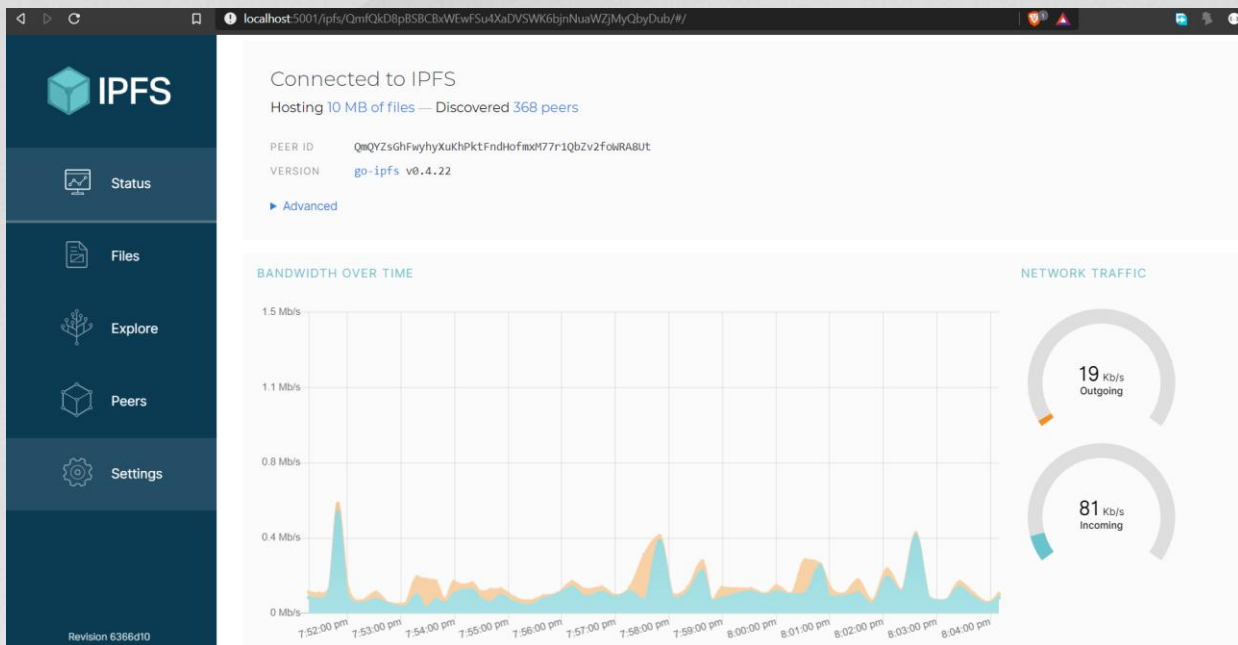
hash_indices.json exists and is readable
updating.... hash indices
The static ipns hash :
ipns key(version holder) and newly uploa
Signing the File
Signing the File
UmbralPublicKey:029eade8e4aa37
UmbralPublicKey:029e8a5da88510e
127.0.0.1 - - [29/Jan/2020 19:50:30] "POST /transactions/new HTTP/1.1" 201 -
127.0.0.1 - - [29/Jan/2020 19:55:18] "POST /mine HTTP/1.1" 405 -
127.0.0.1 - - [29/Jan/2020 19:55:21] "GET /mine HTTP/1.1" 200 -
checking version holder data
127.0.0.1 - - [29/Jan/2020 19:57:07] "GET /version/holder?uid=566 HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 19:58:40] "GET /chain HTTP/1.1" 200 -
data_sharing_policies.json exists and is readable
Modifying ploicies
127.0.0.1 - - [29/Jan/2020 20:01:45] "POST /share/new_policy HTTP/1.1" 200 -
in share
Policy not found !

```

## Background logs-3



## Requesting for data re-encryption



Connection details of a single Hospital (node)

## Sample Code screenshots (this is not full code):

```
blockchain.py ×
C: > Users > Nandakumar > Desktop > projects > main_module > blockchain.py

1  from time import time
2  import hashlib,json
3  import pickle
4  import hashlib
5  import json
6  from urllib.parse import urlparse
7  import requests
8  from config import *
9
10 class Blockchain:
11 > def __init__(self): #Initializing the genesis block ...
16
17 > def register_node(self, address): # Registering a new Hospital in our system...
25
26
27 > def valid_chain(self, chain): # Check whether the present Blockchain in our system is valid...
50
51 > def resolve_conflicts(self): # Resolve conflicts(different blockchain) among the available hospital
74
75 > def new_block(self, proof, previous_hash): # A block structure...
90
91 > def new_transaction(self, ipfs_file_hash, version_holder, digital_sign_doctor, digital_sign_patient
107
108
109 @property
```

Blockchain (code)-1

```
blockchain.py ×
C: > Users > Nandakumar > Desktop > projects > main_module > blockchain.py
14 > def __init__(self): # Initializing the genesis block...
16
17 > def register_node(self, address): # Registering a new Hospital in our system...
25
26
27 > def valid_chain(self, chain): # Check whether the present Blockchain in our system is valid...
50
51 > def resolve_conflicts(self): # Resolve conflicts(different blockchain) among the available hospital
74
75 > def new_block(self, proof, previous_hash): # A block structure...
90
91 > def new_transaction(self, ipfs_file_hash, version_holder, digital_sign_doctor, digital_sign_patient
107
108
109 @property
110 > def last_block(self): # Retrive the recent block...
112
113 @staticmethod
114 > def hash(block): # Generate hash for the provided data...
118
119 > def proof_of_work(self, last_block): # Consensus in our system...
130
131 @staticmethod
132 > def valid_proof(last_proof, proof, last_hash): # Check whether the proof is vaild...
136
```

## Blockchain (code)-2

```
blockchain.py  main_blockchain.py ×
C: > Users > Nandakumar > Desktop > projects > main_module > main_blockchain.py
27
28
29 @app.route('/init', methods=['GET','POST'])
30 > def key_gen(): # Generate key pair for the patient and doctor...
43
44 @app.route('/transactions/new', methods=['GET','POST']) # Create a new transaction
45 > def new_transaction(): ...
73
74 @app.route('/version/holder', methods=['GET','POST']) # Retrive the version holder of patient from IPFS
75 > def view_version_holder(): ...
86
87 @app.route('/update/data', methods=['GET','POST']) # Update the old EHR of an patient
88 > def update_data(): ...
111
112
113 @app.route('/mine', methods=['GET']) # Trigger Mining
114 > def mine(): ...
129
130
131 @app.route('/share/new_policy', methods=['GET','POST']) # Create a new policy for sharing data
132 > def share_data(): ...
149
150 @app.route('/view_policy', methods=['GET']) # View all the registered poolicies
151 > def share_datas(): ...
155
156
```

## Code for interacting with the blockchain-1

```
blockchain.py  main_blockchain.py X
C: > Users > Nandakumar > Desktop > projects > main_module > main_blockchain.py
155
156
157 @app.route('/re_encrypt', methods=['GET','POST']) # Requesting for Re-encryption (Receiver)
158 > def re_encrypt():...
164
165
166 @app.route('/chain', methods=['GET']) # View the whole blockchain on current node
167 > def full_chain():...
173
174
175 @app.route('/check/data', methods=['POST','GET']) # Retrives a patient's transaction from all the block
176 > def check_patient_data():...
191
192
193
194 @app.route('/block/explorer', methods=['POST','GET']) # Explore a particular block
195 > def block_explorer():...
204
205
206 @app.route('/nodes/register', methods=['POST','GET']) # Register a new hospital
207 > def register_nodes():...
222
223
224 @app.route('/retrive/latest', methods=['POST','GET']) # Retrive the recent EHR of a patient
225 > def retrives_latest():...
```

## Code for interacting with the blockchain-2

```
blockchain.py  main_blockchain.py  ipfs_integration.py ●
C: > Users > Nandakumar > Desktop > projects > main_module > ipfs_integration.py
1  import json
2  import encryp
3  import ipfshttpclient
4  import os
5  from json_check import json_file_check
6  from config import *
7  import chalk
8
9  > def get_details(uuid,details): # Processes the patient data...
16
17  > def upload_ipfs(uid,filename): # upload the data to ipfs...
32
33  > def download_and_decrypt(uid,file_hash): # download and decrypt the file from ipfs...
41
42
43
44  '''id=576776585
45  get_details(id)
46  encryp.gen_keypair(id)
47  encryp.encrypt_file(id,f"{id}_medical_report.json")
48  encryp.decrypt_file(id,f"{id}_medical_report.json")
49  sign=encryp.digital_sign(id,f"{id}_medical_report.json_encrypted")
50  hash=upload_ipfs(id,f"{id}_medical_report.json_encrypted")
51  download_and_decrypt(id,hash)'''
52
```

## Code for interaction with the IPFS



```
blockchain.py  main_blockchain.py  ipfs_integration.py  ipfs_versioning.py
C:\Users\Nandakumar\Desktop\projects\main_module> ipfs_versioning.py
1  from ipfs_integration import ipfs_integration
2  from config import *
3  from umbral.curve import Curve, SECP256K1
4  import ipfs_integration as ipfs
5  import os
6  from config import *
7  import encryp
8  from umbral.curve import Curve, SECP256K1
9  from umbral import pre
10 from umbral.params import UmbralParameters
11 import base64
12 import chalk
13
14 > def gen_ipnskey(uid): # Generate PKI inorder to create and maintain version holder...
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32 > def get_ipns_hash_from_index(uid): # Retrives the ipns values from the patient id...
33
34
35
36
37
38
39 > def ipfs_version_holder(uid): ...
40
41
42
43
44
45
46 > def update_medical_record(uid,initial_hash,capsule): # has the logic for updating a medical record...
47
48
49
50
51
52
53 > def ipfs_version_holder_updater(uid,details): #this gets new medical data and updates into ipns...
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97 > def retrieve_lastest(uid): ...
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
```

## Code for Handling the logics behind versioning of HER

```
logics.js - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
JS logics.js x
home > nandu > ehr_hyperledger > chaincode > newcc > JS logics.js > ...
1  'use strict';
2  const { Contract } = require('fabric-contract-api');
3
4  class testContract extends Contract {
5
6      async queryPatientEhr(ctx,patientId) {
7          let detailsAsBytes = await ctx.stub.getState(patientId);
8          if (!detailsAsBytes || detailsAsBytes.toString().length <= 0) {
9              throw new Error('Pateint with this Id does not exist: ');
10             }
11             let details=JSON.parse(detailsAsBytes.toString());
12
13             return JSON.stringify(details);
14         }
15
16         async addPateintEhr(ctx,patientId,ipfsHash,ipnsHash,digitalSig) {
17
18             let data={
19                 patientId,
20                 ipfsHash,
21                 ipnsHash,
22                 digitalSig
23             };
24
25             await ctx.stub.putState(patientId,Buffer.from(JSON.stringify(data)));
26         }
27     }
28 }
```

## Sample Hyperledger Fabric EHR Chaincode

```
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer

File Edit View Search Terminal Help

nandu@nandakumar:~/ehr_hyperledger$ cd basic-network/
nandu@nandakumar:~/ehr_hyperledger/basic-network$ ./
config/      generate.sh      start.sh      teardown.sh
crypto-config/ init.sh      stop.sh
nandu@nandakumar:~/ehr_hyperledger/basic-network$ ./start.sh

# don't rewrite paths for Windows Git Bash users
export MSYS_NO_PATHCONV=1

sudo docker-compose -f docker-compose.yml down
[sudo] password for nandu:
Removing cli ... done
Removing peer0.org1.example.com ... done
Removing ca.example.com ... done
Removing orderer.example.com ... done
Removing couchdb ... done
Removing network net_basic

sudo docker-compose -f docker-compose.yml up -d ca.example.com orderer.example.com peer0.org1.example.com couchdb
Creating network "net_basic" with the default driver
Creating orderer.example.com ... done
Creating ca.example.com ... done
Creating couchdb ... done
Creating peer0.org1.example.com ... done
sudo docker ps -a
CONTAINER ID        IMAGE                                     PORTS
STATUS
d5931b73efaa        hyperledger/fabric-peer               "peer node start --p..." 24 seconds ago
Up 17 seconds      0.0.0.0:7051-7053->7051-7053/tcp
77f8a3bff01b        hyperledger/fabric-couchdb           "tini -- /docker-ent..." 31 seconds ago
Up 24 seconds      4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp
06f95092aae4        hyperledger/fabric-ca                "sh -c 'fabric-ca-se..." 31 seconds ago
Up 23 seconds      0.0.0.0:7054->7054/tcp
c4b5ee5e2d33        hyperledger/fabric-orderer           "orderer"                  31 seconds ago
Up 27 seconds      0.0.0.0:7050->7050/tcp
f24ad983ebaa        dev-peer0.org1.example.com-myccl-1.0-384f11f484b9302df90b453200cfb25174305fce8f53f4e94d45ee3b6cab0ce9 "/bin/sh -c 'cd /usr..." 25 hours ago
Exited (255) 4 minutes ago
c3b9fe07b225        dev-peer0.org1.example.com-ehr-1.1-9e093e2f6ee3bc856f9f8652e9469fc44406e8a1a6e66edcf1f773e8b5a7d502 "/bin/sh -c 'cd /usr..." 26 hours ago
Exited (0) 26 hours ago
4342a9f131a9        dev-peer0.org1.example.com-t1-1.0-a4421cbbc2c0a2618a418f6f34d3af7c08aa604699688cc22093ddbeea6b38dd "/bin/sh -c 'cd /usr..." 26 hours ago
Exited (0) 26 hours ago
4599acd9c70e        dev-peer0.org1.example.com-t1-1.9-d01b960179840effb3b1e4e0358d718d60d51b86e09cb11af3c9fbdeb80f2ed8 "/bin/sh -c 'cd /usr..." 5 days ago
Exited (0) 5 days ago

# wait for Hyperledger Fabric to start
```

## Background logs-4 (Hyperledger Fabric Deployment)

```
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer

File Edit View Search Terminal Help

# wait for Hyperledger Fabric to start
# in case of errors when running later commands, issue export FABRIC_START_TIMEOUT=<larger number>
export FABRIC_START_TIMEOUT=10
# echo the FABRIC_START_TIMEOUT
sleep ${FABRIC_START_TIMEOUT}

# Create the channel
sudo docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp" peer0.org1.example.com peer channel create -o orderer.example.com:7050 -c mychannel -f /etc/hyperledger/configtx/channel.tx
2020-01-31 16:37:39.508 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-31 16:37:39.700 UTC [cli/common] readBlock -> INFO 002 Got status: &{NOT_FOUND}
2020-01-31 16:37:39.701 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2020-01-31 16:37:39.930 UTC [cli/common] readBlock -> INFO 004 Received block: 0
# Join peer0.org1.example.com to the channel.
sudo docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp" peer0.org1.example.com peer channel join -b mychannel.block
2020-01-31 16:37:40.343 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-31 16:37:41.123 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel

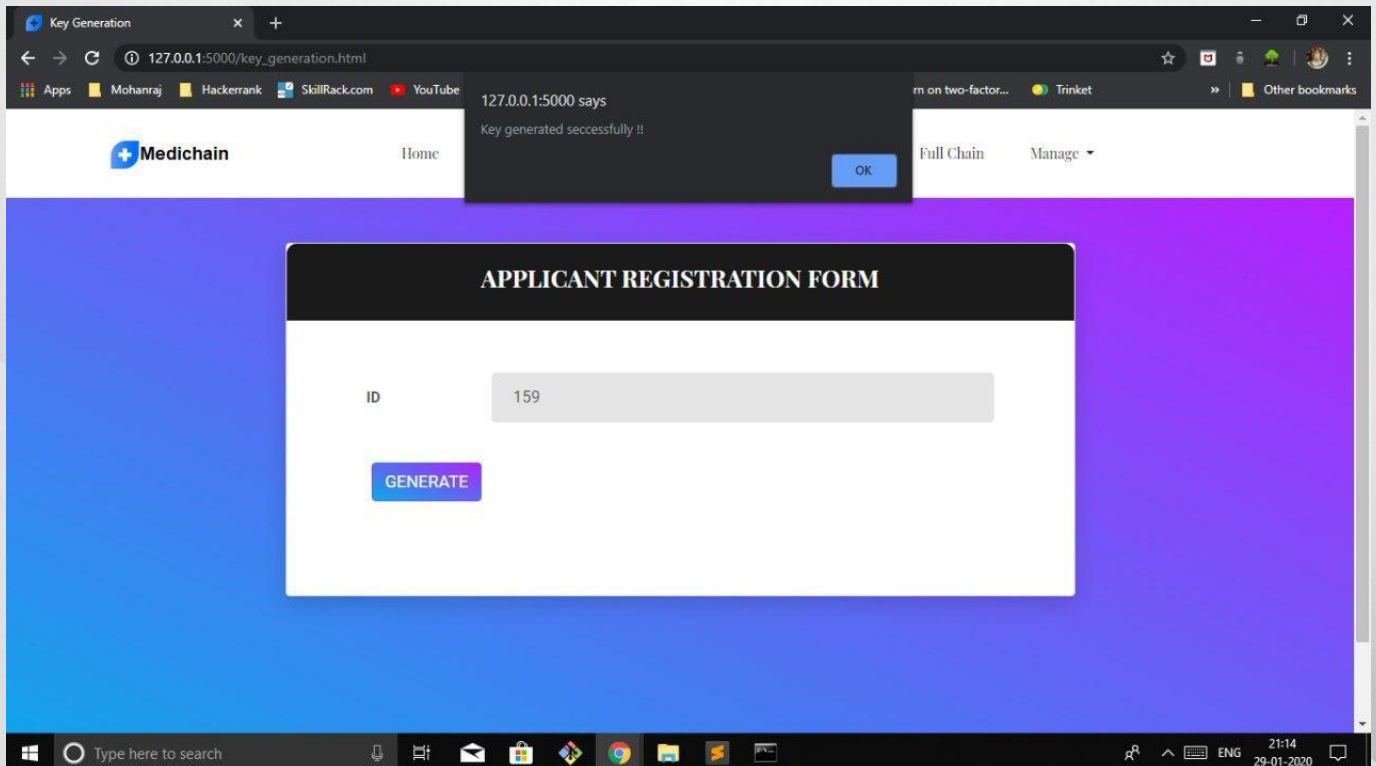
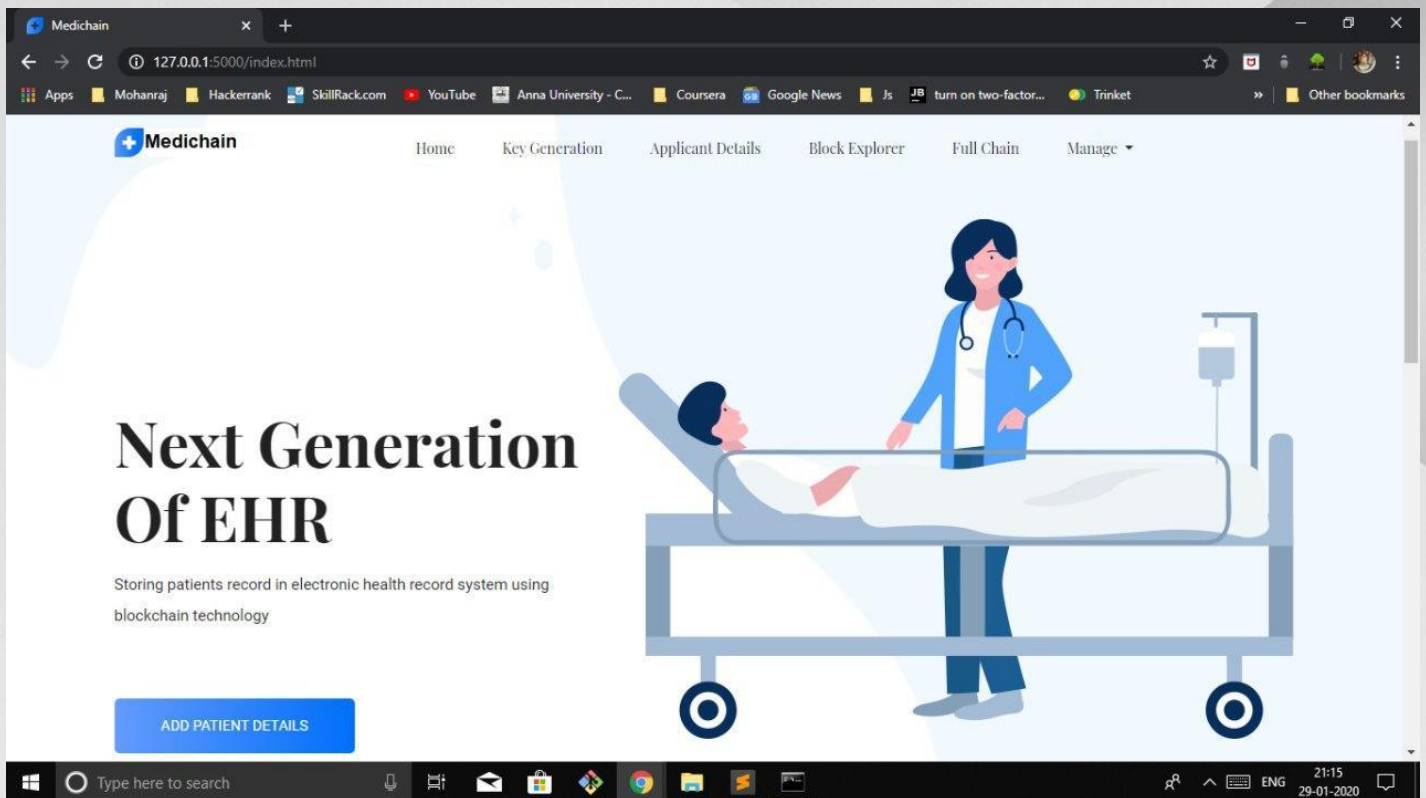
# Creating CLI Container
sudo docker-compose -f ./docker-compose.yml up -d cli
Creating cli ... done

nandu@nandakumar:~/ehr_hyperledger/basic-network$ docker exec -it cli bash
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode install -n mycc -v 1.0 -p "/opt/gopath/src/github.com/newcc" -l "node"

2020-01-31 16:40:07.312 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-01-31 16:40:07.312 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vsc
2020-01-31 16:40:07.312 UTC [container] WriteFolderToTarPackage -> INFO 003 rootDirectory = /opt/gopath/src/github.com/newcc
2020-01-31 16:40:07.373 UTC [chaincodeCmd] install -> INFO 004 Installed remotely response: <status:200 payload:"OK" >
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode instantiate -o orderer.example.com:7050 -C mychannel -n mycc -l "node" -v 1.0 -c '{"Args":[""]}'
2020-01-31 16:40:11.212 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-01-31 16:40:11.212 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vsc
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o orderer.example.com:7050 -C mychannel -n mycc -c '{"function": "addPatientEhr", "Args":["213", "B447C27A00E3A348881B0030177000CD", "AF9CBCEB8388E2170881D17F5CA88833C5C37ED6", "2CF3A75A61FBF56EBC9BE045D9592B"]}'
2020-01-31 16:42:10.037 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o orderer.example.com:7050 -C mychannel -n mycc -c '{"function": "queryPatientEhr", "Args":["213"]}'
2020-01-31 16:42:53.034 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:{"digitalSig":"2CF3A75A61FBF56EBC9BE045D9592B","ipfsHash":"B447C27A00E3A348881B0030177000CD","ipnsHash":"AF9CBCEB8388E2170881D17F5CA88833C5C37ED6","patientId":"213"}
root@989634065a7e: /opt/gopath/src/github.com/hyperledger/fabric/peer#
```

## Background logs-5 (Fabric Chaincode installation and invocation)

## Frontend (WebApp):





Medical Details

127.0.0.1:5000/medical\_details.html

AppsMohanrajHackerrankSkillRack.comYouTubeAnna University - C...CourseraGoogle NewsJsJBturn on two-factor...TrinketOther bookmarks

Medical Form

TYPE

☐ Update☒ New

UUID

159

DUID

999

Diagnosing Details

Analysis : Suspected the presence of tumor cells over the breast.

Diagnosis : Asked the patient to take MRI scan

Treatment : Based on the scan report it is identified a minute growth of cancer cells and advised to take chemotherapy.

Type here to search

ENG19:3129-01-2020

Full Chain

127.0.0.1:5000/full\_chain.html

AppsMohanrajHackerrankSkillRack.comYouTubeAnna University - C...CourseraGoogle NewsJsJBturn on two-factor...TrinketOther bookmarks

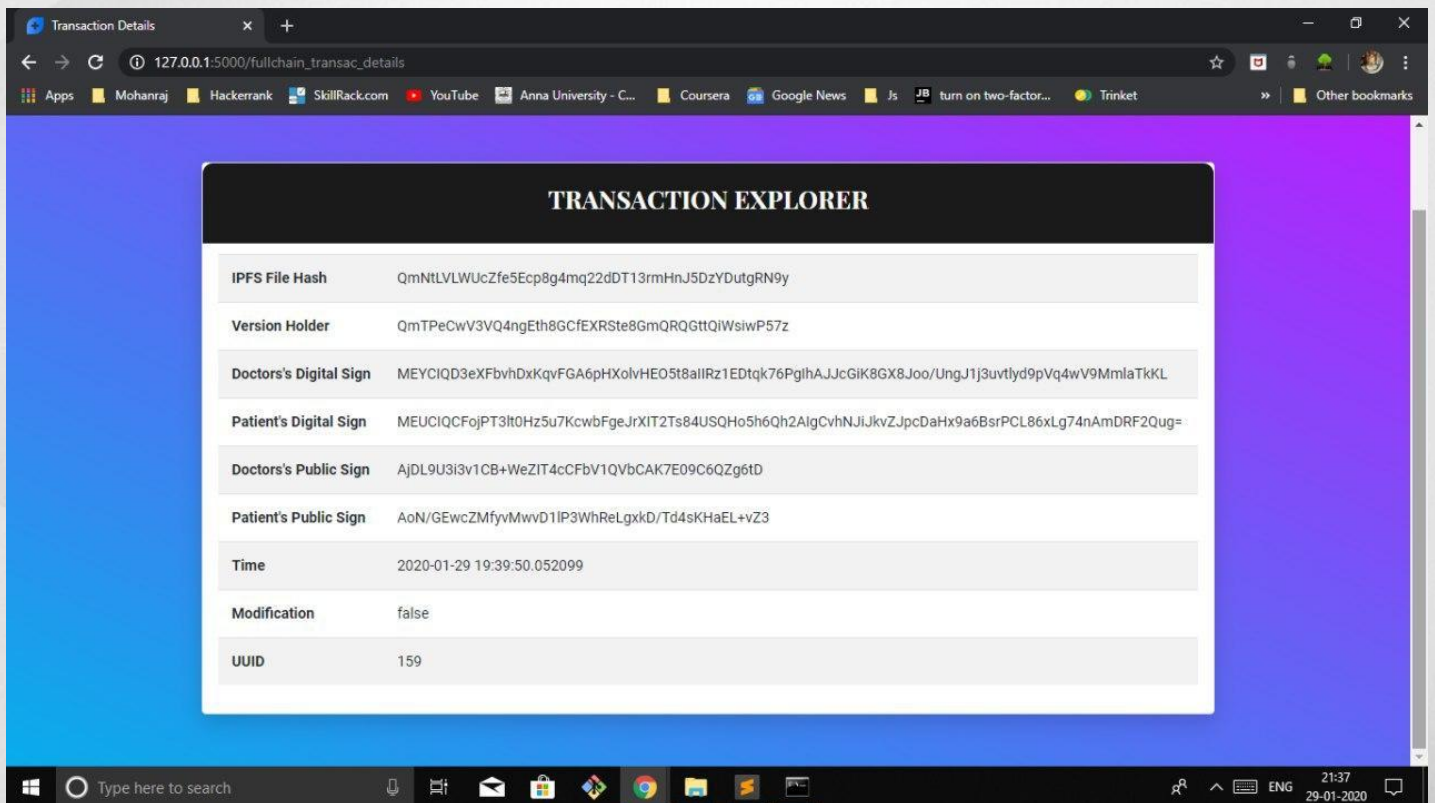
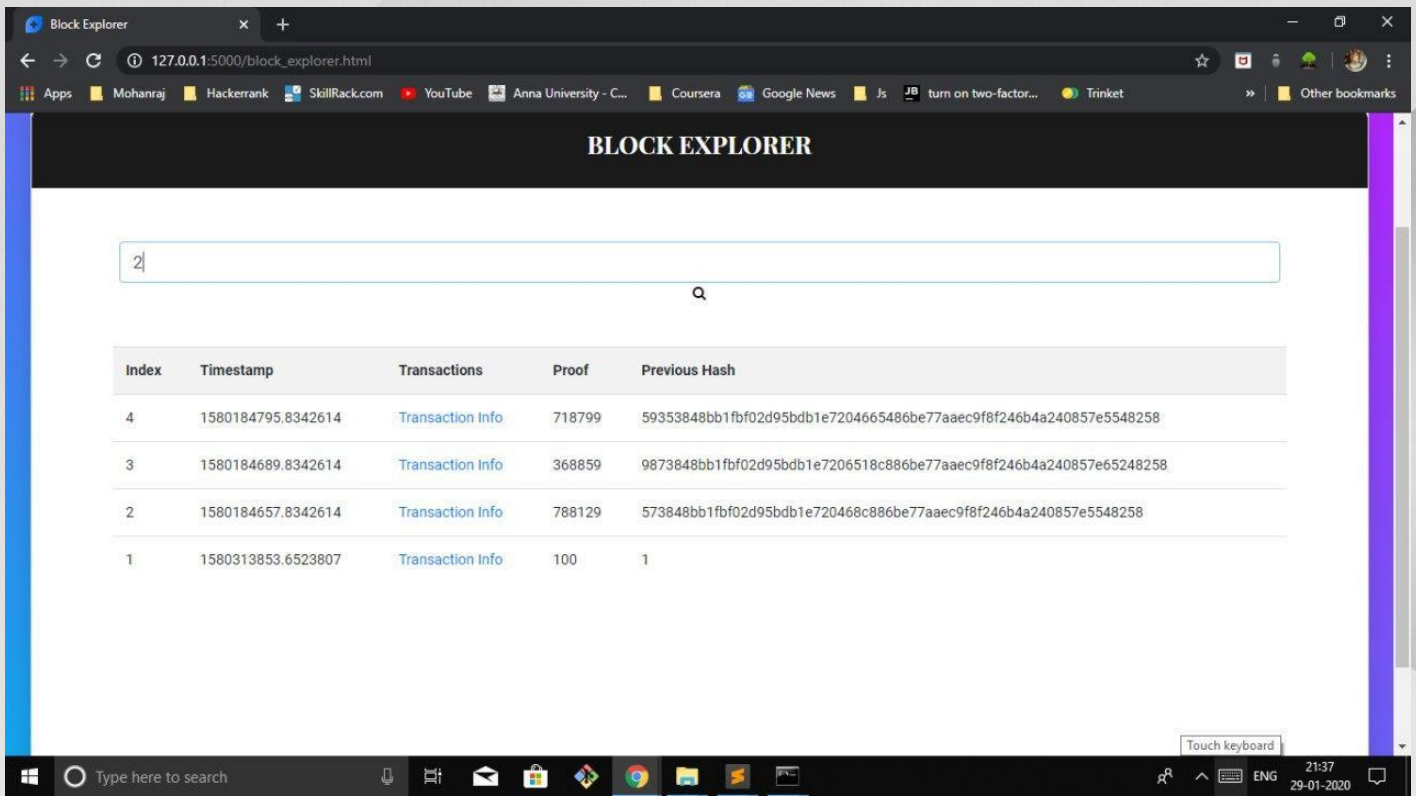
Block Explorer

Index	Timestamp	Transactions	Proof	Previous Hash
7	1580185259.8342614	<a href="#">Transaction Info</a>	89899	983848bb1fbf075s5bdb1e720468c886be77aaec9f8f2assb4a240857e5t
6	1580185178.8342614	<a href="#">Transaction Info</a>	988321	98848bb1fbf02d95bdb1e7204ada886be77aaec9f8f246b4a2408saddsd
5	1580184900.8342614	<a href="#">Transaction Info</a>	935892	355848bb1fbf02d95b35464.468c886be77aaec9f8f246b4a240857e5684
4	1580184795.8342614	<a href="#">Transaction Info</a>	718799	59353848bb1fbf02d95bdb1e7204665486be77aaec9f8f246b4a240857e
3	1580184689.8342614	<a href="#">Transaction Info</a>	368859	9873848bb1fbf02d95bdb1e7206518c886be77aaec9f8f246b4a240857e
2	1580184657.8342614	<a href="#">Transaction Info</a>	788129	573848bb1fbf02d95bdb1e720468c886be77aaec9f8f246b4a240857e55
1	1580313853.6523807	<a href="#">Transaction Info</a>	100	1

Type here to search

ENG21:3829-01-2020





```
CA\Users\Mohanraj\Desktop\Medichain Final-0.1\static\block_explorer\2block.json - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

2block.json
1 [
2   {
3     "ipfs_file_hash": "QmNtLVLWUcZf5Ecp8g4mq22dDT13rmHnJ5DzYDutgRN9y",
4     "version_holder": "QmTPeCwV3VQ4ngEth8GCFEXRSt8GmQRQgttQiWsiwP57z",
5     "digital_sign_doctor": "MEYCIQD3eXfbvhDxKqvFGA6pHx0lvHEO5t8aIIRz1EDtqk76PgIhAJJcGik8GX8Joo/UngJ1j3uvtlyd9pVq4wV9MmlaTkKL",
6     "digital_sign_patient": "MEUCIQCfFjPT31t0H2su7KcwbFgeJrXIT2Ts84U5QH05h6Qh2AIgCvhnJi7kvZJpcDaHx9a6BsrPCL86xLg74nAmDRF2Qug=",
7     "public_key_doctor": "AjdL9U3i3v1CB+WeZIT4cCFbV1QVbCAK7E09C6QZg6tD",
8     "public_key_patient": "AoN/GEwcZMfyvMwvD11P3WhReLgxxD/Td4sKHaEL+vZ3",
9     "time": "2020-01-29 19:39:50.052099",
10    "modification": "false",
11    "UUID": "159"
12  },
13  {
14    "ipfs_file_hash": "Qmft3FNmh4jckdcAf2jTBTv13okqwA6EMaTwaeWQ67jE6H",
15    "version_holder": "QmTPeCwV3VQ4ngEth8GCFEXRSt8GmQRQgttQiWsiwP57z",
16    "digital_sign_doctor": "MEUCIEhOH0AodL4UOUoJXERp6svnpivnA+8onRwtC4xH/W+GAiEAwwDdbZA5rpdwgyW9fHOVNphufAYZKxGt+a4sxyg8J7U=",
17    "digital_sign_patient": "MEUCIFavR1+ugWuAXYFH1t0RVRXIBzdEPEB+tyTxi18zf1hXaIB8mmM8BSLxebtY12A+GaXH9Ktwu3jwPVd6MncgaY0q+w==",
18    "public_key_doctor": "AjdL9U3i3v1CB+WeZIT4cCFbV1QVbCAK7E09C6QZg6tD",
19    "public_key_patient": "AoN/GEwcZMfyvMwvD11P3WhReLgxxD/Td4sKHaEL+vZ3",
20    "time": "2020-01-29 19:43:44.382974",
21    "modification": "true",
22    "UUID": "159"
23  },
24  {
25    "ipfs_file_hash": "QmUi2gQGXAmb7bSXdqBKcj9wWSE21YoH95gkJSCKDY9Z8S",
26    "version_holder": "QmTPeCwV3VQ4ngEth8GCFEXRSt8GmQRQgttQiWsiwP57z",
27    "digital_sign_doctor": "MEYCIQCcBIAXR0tSkX/RCTTvFBRxN/eUB//2JNrVTFPR2vodYAIhAM4PmHaTtmp2fmycl5rIOX6kFB1x0w1QpY3dn5bOKO/n",
28    "digital_sign_patient": "MEUCIC/HhnOYvvReD62yqi2d9G7w8zq6KVdpmo5yaF6UfpuJAiA+gcqeK9tah85dIw/oBjOdVE0i1awGWecChp+mF/tDGw==",
29    "public_key_doctor": "AxxwBI4/g80POOYvzSF7Z4Rzet03iedjntte5W9SPdeyW",
30    "public_key_patient": "AoN/GEwcZMfyvMwvD11P3WhReLgxxD/Td4sKHaEL+vZ3",
31    "time": "2020-01-29 19:45:56.038085",
32    "modification": "true",
33  }
34 ]
```

```
Command Prompt - python 06-Form-Example.py
C:\Users\Mohanraj\Desktop\Medichain Final-0.1>python 06-Form-Example.py
* Serving Flask app "06-Form-Example" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 443-956-587
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [29/Jan/2020 21:41:49] "GET /key_generation.html HTTP/1.1" 200 -
checking
159 key pair exists and is readable
{'status': 'success'}
127.0.0.1 - - [29/Jan/2020 21:41:55] "POST /index.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:42:00] "GET /applicant_details.html HTTP/1.1" 200 -
checking
159 key pair exists and is readable
checking
999 key pair exists and is readable
{'uid': '159', 'name': 'Chris Moris', 'dob': '1995-11-07', 'pnum': '7010165529', 'Acceptor_uid': '999'}
127.0.0.1 - - [29/Jan/2020 21:43:00] "POST /processing.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:02] "GET /index.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:08] "GET /medical_details.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:08] "GET /static/css/bootstrap-reboot.min.css HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:08] "GET /static/js/bootstrap.bundle.min.js HTTP/1.1" 200 -
checking
159 key pair exists and is readable
checking
999 key pair exists and is readable
exist2
{'typeOfTransaction': 'exist2', 'uid': '159', 'doctorUid': '999', 'data': 'Health Diagnosis'}
New Transaction Initiated....
Encryption in progress
159_public_key reading....
IPFS conn:
uploading.... to ipfs
hash of the file : QmeATG9vu8VC39Xcj1g7LATxHyZnNMGg1kLd2h8Ve7ARpB
hash_indices.json exists and is readable
updating.... hash indices
ipns hash= QmTPeCwV3VQ4ngEth8GCFEXRSt8GmQRQgttQiWsiwP57z
159_version_holder.json exists and is readable
IPFS conn:
```



```
Command Prompt - python 06-Form-Example.py
127.0.0.1 - - [29/Jan/2020 21:41:55] "POST /index.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:42:00] "GET /applicant_details.html HTTP/1.1" 200 -
checking
159 key pair exists and is readable
checking
999 key pair exists and is readable
{'uid': '159', 'name': 'Chris Moris', 'dob': '1995-11-07', 'pnum': '7010165529', 'Acceptor_uid': '999'}
127.0.0.1 - - [29/Jan/2020 21:43:00] "POST /processing.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:02] "GET /index.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:08] "GET /medical_details.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:08] "GET /static/css/bootstrap-reboot.min.css HTTP/1.1" 200 -
127.0.0.1 - - [29/Jan/2020 21:43:08] "GET /static/js/bootstrap.bundle.min.js HTTP/1.1" 200 -
checking
159 key pair exists and is readable
checking
999 key pair exists and is readable
exist2
{'typeOfTransaction': 'exist2', 'uid': '159', 'doctorUid': '999', 'data': 'Health Diagnosis'}
New Transaction initiated....
Encryption in progress
159_public_key reading....
IPFS conn:
uploading.... to ipfs
hash of the file : QmeATG9vu8VC39Xcj1g7LAtHxyZnNNGg1kLd2hBVe7ARpB
hash_indices.json exists and is readable
updating.... hash indices
ipns hash= QmTPeCwV3VQ4ngEth8GCFEXRSte8GmQRQgttQiWsiwP57z
159_version_holder.json exists and is readable
IPFS conn:
uploading.... to ipfs
hash of the file : Qm54PcUX96EAytEMMXeJXKch44rcbvm3LtzhiBUesVexvG
hash_indices.json exists and is readable
updating.... hash indices
The static ipns hash : QmTPeCwV3VQ4ngEth8GCFEXRSte8GmQRQgttQiWsiwP57z
ipns key(version holder) and newly upload hash comparison, result= True
Signing the File
Reading private key of 999
<umbral.keys.UmbralPrivateKey object at 0x04851838>
Signing the File
Reading private key of 159
<umbral.keys.UmbralPrivateKey object at 0x04851A78>
UmbralPublicKey:0230cbf54de2def
UmbralPublicKey:02837f184c1c64c
N=True
```

