

**BLOCKCHAIN FOR SECURE EHRs SHARING OF MOBILE CLOUD BASE E-HEALTH SYSTEM****K Rambabu, Cheboyina. Bhavya Sri****Assistant Professor(HOD) MCA&M. Teach, DEPT, Dantuluri Narayana Raju College, Bhimavaram, Andhra Pradesh****Email id:-[kattaramababudnr@gmail.com](mailto:kattaramababudnr@gmail.com)****PG Student of MCA, Dantuluri Narayana Raju College, Bhimavaram, Andhra Pradesh****Email id:-[bhavya43354@gmail.com](mailto:bhavya43354@gmail.com)****ABSTRACT**

As the healthcare industry becomes increasingly digitalized, mobile cloud-based e-health systems are becoming more prevalent, allowing for remote access to electronic health records (EHRs) from anywhere at any time. However, the sharing of EHRs through these systems presents significant security concerns, including the risk of data breaches and unauthorized access. Blockchain technology has emerged as a potential solution to these security concerns. Blockchain offers a decentralized and immutable ledger, which ensures data integrity, confidentiality, and security. By incorporating blockchain into mobile cloud-based e-health systems, EHRs can be shared securely between authorized parties, while maintaining patient privacy and preventing unauthorized access. This Project explores the potential of blockchain technology for secure EHR sharing in mobile cloud-based e-health systems. We examine the benefits of using blockchain, including its ability to provide a tamper-proof record of EHR access and the decentralization of trust. We also explore the challenges of implementing blockchain in healthcare, including regulatory barriers, technical limitations, and interoperability issues. The blockchain technology has the potential to significantly improve the security of EHR sharing in mobile cloud-based e-health systems. However, further research and development are necessary to overcome the challenges associated with its implementation and ensure its widespread adoption.

**1. INTRODUCTION****1. Brief Information about the project:**

Recently, there has been a growing interest in employing the blockchain technology to promote medical and e-health services. Blockchain with its decentralized and trustworthy nature has demonstrated immense potentials in various e-health sectors such as secure sharing of Electronic Health Records (EHRs) and data access management among multiple medical entities. Therefore, the adoption of blockchain can provide promising solutions to facilitate healthcare delivery and thus revolutionize the healthcare industry.

With the emergence of innovative technologies, including Mobile Cloud Computing (MCC) and Internet of Medical Things (IoMT), the healthcare industry has witnessed significant changes in e-health operations. Patients now can collect their personal health information at home based on mobile devices (such as smartphones and wearable sensors) and share on cloud environments where healthcare providers can access instantly to analyze medical records and provide timely medical supports. This smart e-health service allows healthcare providers remotely monitor patients and offer ambulatory care at home, which not only facilitates healthcare delivery but also brings economic benefits to patients. Further, the availability of complete EHRs on clouds also helps healthcare providers track patient health and offers proper medical services during diagnosis and treatment processes.

Besides all these great advantages, however, the trend of EHRs storage on clouds also poses security challenges which hinder the deployment of e-health applications on clouds. Among such security issues is secure EHRs sharing between patients and healthcare providers on mobile cloud environments. Unauthorized entities may gain malicious access to EHRs without consent of patients, which has detrimental impacts on data integrity, privacy and security of cloud e-health systems. Moreover, patients may find it difficult to track and manage their health records shared among healthcare providers on

clouds. It therefore is necessary to propose efficient access control solutions for mobile cloud EHRs sharing systems.

## **2. Motivation and contribution of Project:**

Generally, EHRs mainly contain patient medical history, personal statistics (e.g. age and weight), laboratory test results and so on. Hence, it is crucial to ensure the security and privacy of these data. In addition, hospitals in countries such as U.S. are subject to exacting regulatory oversight. There are also a number of challenges in deploying and implementing healthcare systems in practice. For example, centralized server models are vulnerable to the single-point attack limitations and malicious insider attacks, as previously discussed. Users (e.g. patients) whose data is outsourced or stored in these EHR systems generally lose control of their data, and have no way of knowing who is accessing their data and for what kind of purposes (i.e. violation of personal privacy). Such information may also be at risk of being leaked by malicious insiders to another organization, for example an insurance company may deny insurance coverage to the particular patient based on leaked medical history.

## **2. LITERATURE SURVEY AND RELATED WORK**

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating System and Language can be used for developing the tool.

### **An energy-efficient transaction model for the blockchain-enabled Internet of Vehicles (IoV),”**

The blockchain is a safe, reliable and innovative mechanism for managing numerous vehicles seeking connectivity. However, following the principles of the blockchain, the number of transactions required to update ledgers pose serious issues for vehicles as these may consume the maximum available energy. To resolve this, an efficient model is presented in this letter which is capable of handling the energy demands of the blockchain enabled Internet of Vehicles (IoV) by optimally controlling the number of transactions through distributed clustering. Numerical results suggest that the proposed approach is 40.16% better in terms of energy conservation and 82.06% better in terms of the number of transactions required to share the entire blockchain data compared with the traditional blockchain.

### **“On scaling decentralized blockchains,”**

The increasing popularity of blockchain-based cryptocurrencies has made scalability a primary and urgent concern. Analyze how fundamental and circumstantial bottlenecks in Bitcoin limit the ability of its current peer-to-peer overlay network to support substantially higher throughputs and lower latencies. Our results suggest that reparameterization of block size and intervals should be viewed only as a first increment toward achieving next-generation, high-load blockchain protocols, and major advances will additionally require a basic rethinking of technical approaches. offer a structured perspective on the design space for such approaches. Within this perspective, enumerate and briefly discuss a number of recently proposed protocol ideas and offer several new ideas and open challenges.

### **“A low storage room requirement framework for distributed ledger in blockchain,**

Traditional centralized commerce on the Internet relies on trusted third parties to process electronic payments. It suffers from the

weakness of the trust-based model. A pure decentralized mechanism called blockchain tackles the above problem and has become a hot research area. However, since each node in a blockchain system needs to store all transactions of the other nodes, as time continues, the storage room required to store the entire blockchain will be huge. Therefore, the current storage mechanism needs to be revised to cater to the rapidly increasing need for storage. Network coded (NC) distributed storage (DS) can significantly reduce the required storage room. This paper proposes a NC-DS framework to store the blockchain and proposes corresponding solutions to apply the NC-DS to the blockchain systems. Analysis shows that the proposed scheme achieves significant improvement in saving storage room.

#### **“Distributed storage meets secret sharing on the blockchain,**

Blockchain systems establish a cryptographically secure data structure for storing data in the form of a hash chain. Use a novel combination of distributed storage, private key encryption, and Shamirs secret sharing scheme to distribute transaction data, without significant loss in data integrity. Additionally, using Shamirs secret sharing scheme on the hash values and dynamic zone allocation, further enhance the integrity. In this Project highlight the tradeoff in storage cost and data loss probability with varying zone size choices. Then, formulate code design, given a probability of data recovery and targeted corruption, as an integer program. Using the coding scheme establish a mechanism to insure data, for instance in blockchain-based cloud storage systems, based on the value of the data, by understanding the costs involved for the service provider.

#### **“Efficient local secret sharing for distributed blockchain systems,**

Blockchain systems store transaction data in the form of a distributed ledger where each peer is to maintain an identical copy. Blockchain systems resemble repetition codes, incurring high storage cost. Recently, distributed storage blockchain (DSB) systems have been proposed to improve storage efficiency by incorporating secret sharing, private key encryption, and information dispersal algorithms. However, the DSB results in significant communication cost when peer failures occur due to denial of service attacks. In this letter, a new DSB approach based on a local secret sharing (LSS) scheme with a hierarchical secret structure of one global secret and several local secrets. The proposed DSB approach with LSS improves the storage and recovery communication costs.

### **3. EXISTING SYSTEM**

Blockchain is a paradigm-shifting technology that has emerged over the past decade, which is based on peer-to-peer communication technology, network theory, and cryptography. However, there are still some limitations in the existing blockchain framework that prevents its widespread adoption in the commercial world. One important limitation is the storage requirement, wherein each blockchain node has to store a copy of the distributed ledger. •us, as the number of transactions increases, this storage requirement grows quadratically, eventually limiting the scalability of a blockchain system.

### **4. PROPOSED SYSTEM**

In this, instead of saving entire transaction of blocks are saving only one block. To provide security to block author converting that block in to SHAMIR share and then all SHAMIR share will be distributed between all available nodes.

While reconstruction application will obtain all shares from nodes and then apply SHAMIR SECRET to recover original block data. If any share missed or return incorrect value then reconstruction will be failed. SHAMIR secret will work based on random polynomial and prime number while generating secret polynomial will be applied on block data and while getting original value will perform reverse polynomial.

### Advantages of Proposed System

- This can effectively work.
- Security is more.

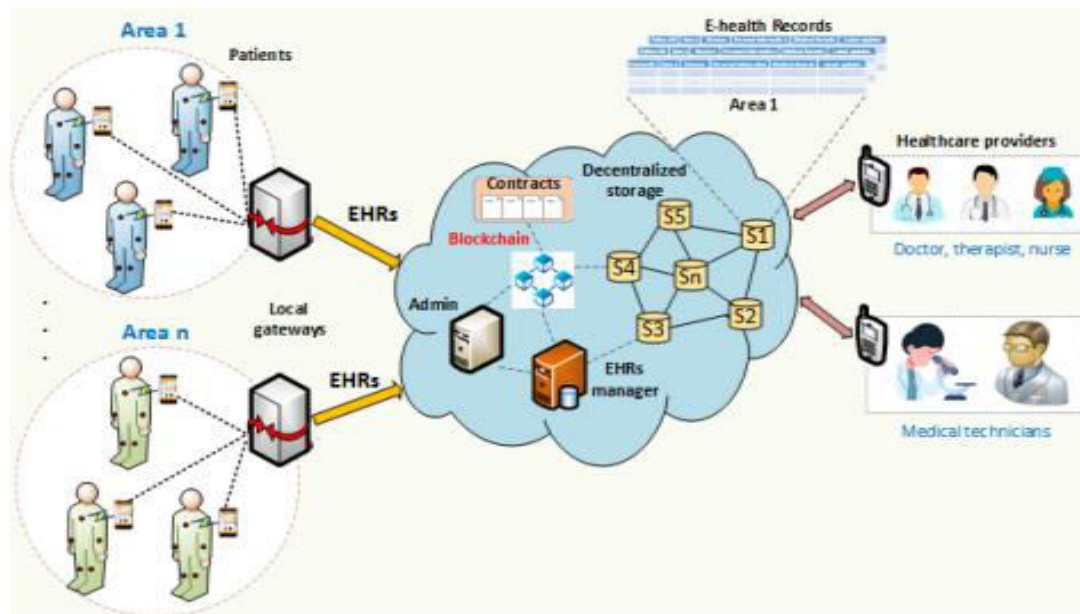


FIG1: SYSTEM ARCHITECTURE

## 5. METHODOLOGIES

The purpose is to exercise the different parts of the module code to detect coding errors. After this the modules are gradually integrated into subsystems, which are then integrated themselves too eventually forming the entire system. During integration of module integration testing is performed. The goal of this is to detect designing errors, while focusing the interconnection between modules. After the system was put together, system testing is performed. Here the system is tested against the system requirements to see if all requirements were met and the system performs as specified by the requirements. Finally accepting testing is performed to demonstrate to the client for the operation of the system.

For the testing to be successful, proper selection of the test case is essential. There are two different approaches for selecting test case. The software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called “black box testing”.

The focus here is on testing the external behavior of the system. In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. The two forms of testing are complementary: one tests the external behavior, the other tests the internal structure.

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Frequently the testing process starts with the test plan. This plan identifies all testing related activities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested; different units to be tested, and the manner in which the module will be integrated together. Then for different test unit, a test case specification document is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual results are compared with the expected outputs. The final output of the testing phase is the testing report and the error report, or a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the errors encountered and the action taken to remove the error.

#### MODULES:

- EHR Manager
- Admin
- Smart Contract
- IPFS
- Data Uploading
- Data sharing

#### DESCRIPTION:

##### **EHR Manager:**

This is an internal module responsible to manage all storage transaction and access control in Blockchain

##### **Admin:**

Responsible to deploy module on Blockchain

**Smart Contract:**

Allow us to execute operations on Blockchain such as storing hash codes. This module allows user to interact with Blockchain.

**IPFS:**

Allow us to store patient data and then returned hash code will be stored in Blockchain

**Data Uploading:**

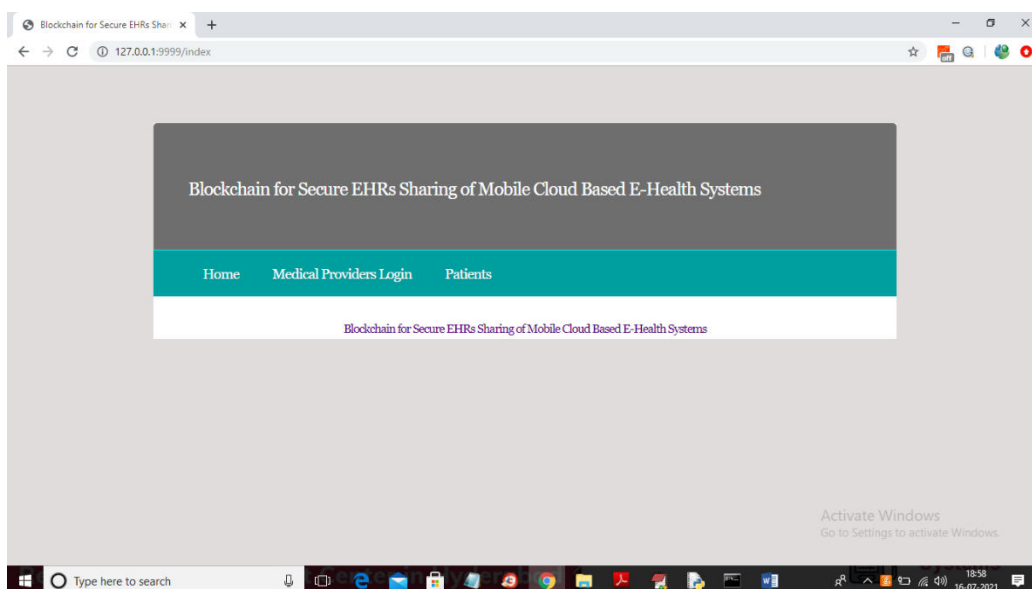
Patient can upload their data and will be stored at IPFS and Blockchain

**Data sharing:**

Doctors who have access to Blockchain can obtained hash code from it and then input that hash code to IPFS to get patient details

## 6. RESULTS AND DISCUSSION SCREEN SHOTS

**Test Screens**



**Fig1: Test screen for patient details to store on smart contract**

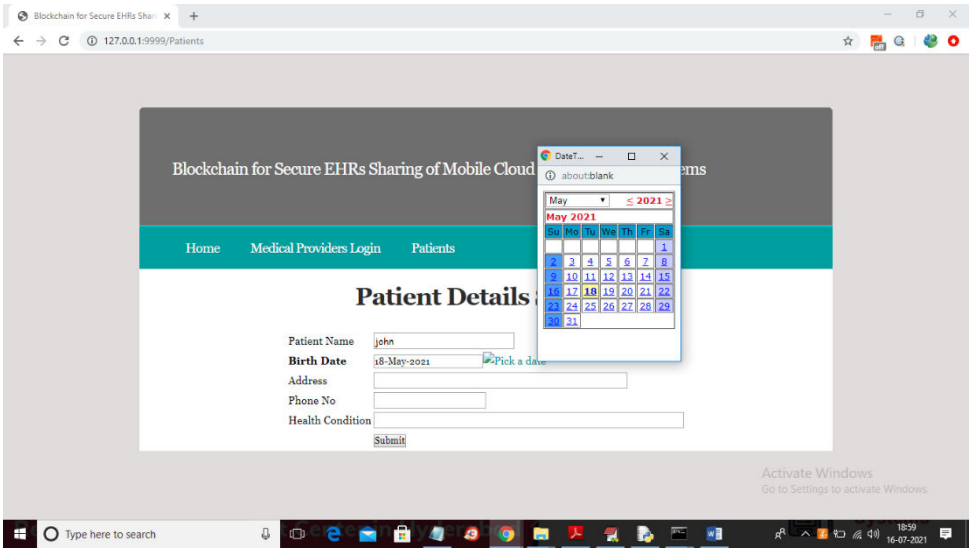


Fig: .2 Test screen for patient will enter his details and then select birth date

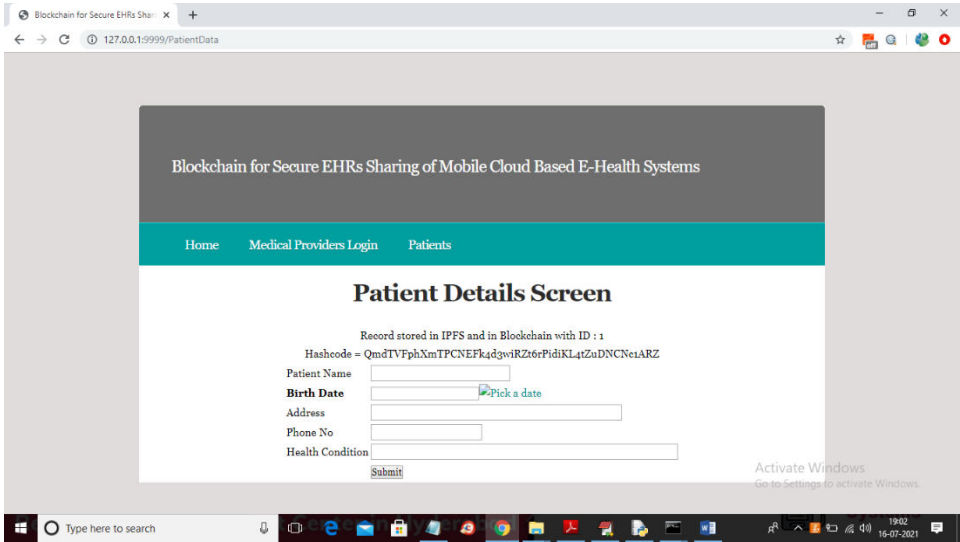


Fig : 3 Test screen for patient record id

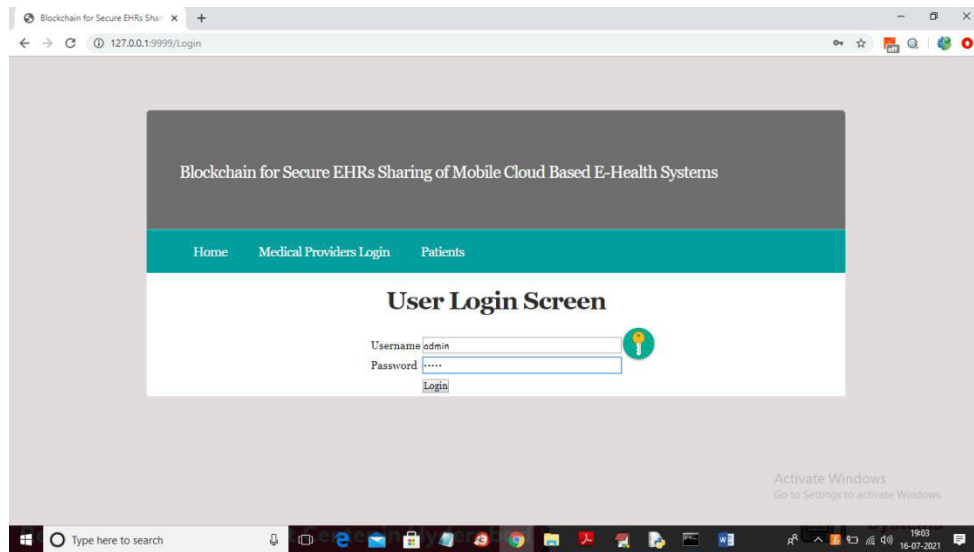


Fig: 4 Test screen for login

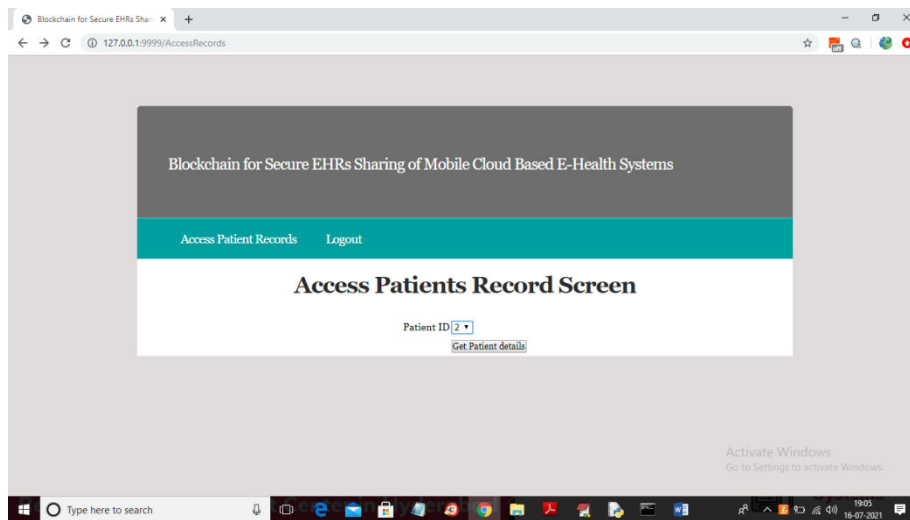


Fig: 5 Test screen for patient ID



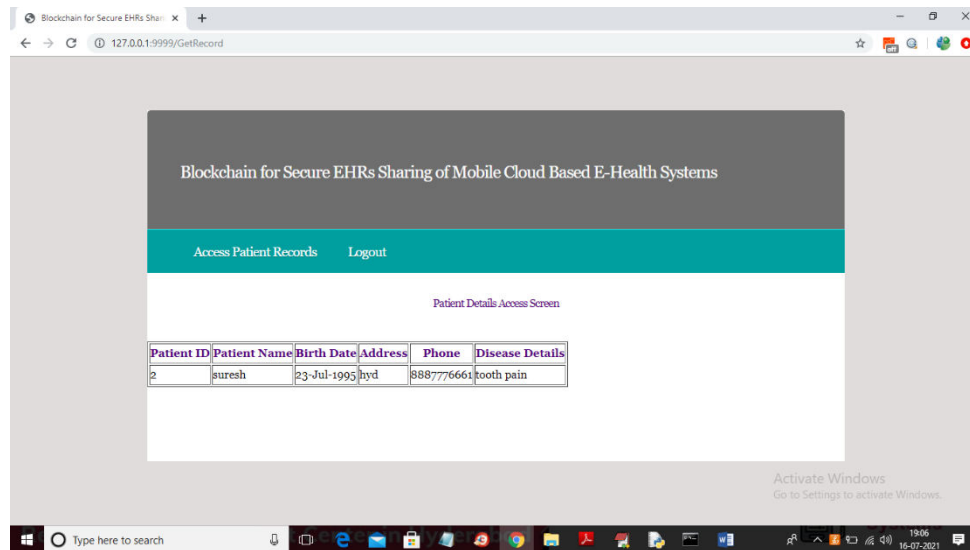
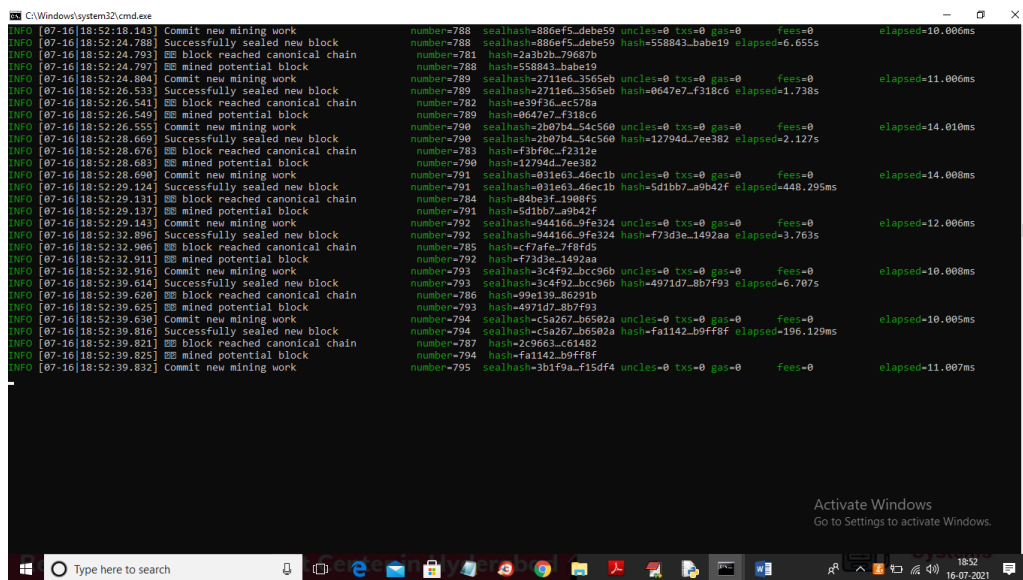


Fig: 6 Test screen for view patient details



### Screenshot for 'start\_eth'

DESCRIPTION: To run project you need to double click on 'start\_eth.bat' to start Ethereum tool and to get below screen and after starting this you need to wait few minutes till u get scrolling messages.

```

Select C:\Windows\system32\cmd.exe

E:\NewClient\SecureEHR>javac -cp ";lib/*" -d . *.java
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\NewClient\SecureEHR>java -cp ";lib/*" -Xmx1000M com.deploy
Transaction complete : 0xe9d0c5d272bee8daf9655568b2384d25b884e782b96056b321e7e8a92fe1a844 0x2af242047011f432a166561e14007c4362f88195
Deploying smart contract
Smart contract deployed to address 0x2af242047011f432a166561e14007c4362f88195
Initial value of counter in Smart contract: temp,temp
Smart Contract Ready to store data
Transaction complete : 0xe9d0c5d272bee8daf9655568b2384d25b884e782b96056b321e7e8a92fe1a844 0x2af242047011f432a166561e14007c4362f88195
Address 0x2af242047011f432a166561e14007c4362f88195
1 QmQxt04EkuB1fQejrEd2hEpjEZP6tVnBjx1rz9KM28sDmV Record saved in Ethereum
Address 0x2af242047011f432a166561e14007c4362f88195
2 QmdNrkYtEY4cgCeM17qkq43T15xe6g385qZ2h8YdsuWcaB7 Record saved in Ethereum
1#2
1#2
1#2
1#2
1, QmQxt04EkuB1fQejrEd2hEpjEZP6tVnBjx1rz9KM28sDmV
1#2
2, QmdNrkYtEY4cgCeM17qkq43T15xe6g385qZ2h8YdsuWcaB7

```

### Screenshot for 'initialize\_eth.bat'

DESCRIPTION: Smart Contract Ready to store data

```

C:\Windows\system32\cmd.exe

E:\NewClient\SecureEHR>ipfs init
initializing IPFS node at C:\Users\Admin\.ipfs
generating 2048-bit RSA keypair...done
peer identity: QmzEp5mMAABInCNSP7LpM7ahGoQnbDyxjoD3N3svPMdJL
to get started, enter:

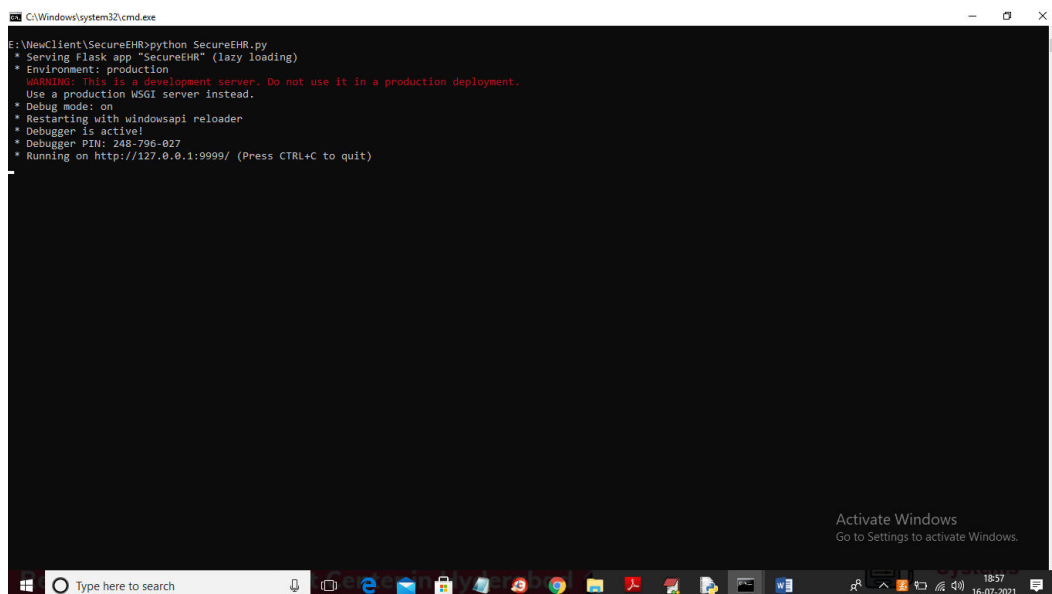
  ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuM1UhyvMcX9Ba8NuUH4vV/readme

E:\NewClient\SecureEHR>ipfs daemon
initializing daemon...
Swarm listening on /ip4/10.102.37.150/tcp/4001
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/169.254.131.210/tcp/4001
Swarm listening on /ip4/169.254.177.21/tcp/4001
Swarm listening on /ip4/169.254.221.206/tcp/4001
Swarm listening on /ip4/169.254.80.27/tcp/4001
Swarm listening on /ip4/172.23.81.17/tcp/4001
Swarm listening on /ip4/192.168.0.5/tcp/4001
Swarm listening on /ip6:::1/tcp/4001
Swarm listening on /p2p-circuit/ipfs/QmzEp5mMAABInCNSP7LpM7ahGoQnbDyxjoD3N3svPMdJL
Swarm announcing /ip4/10.102.37.150/tcp/4001
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/169.254.131.210/tcp/4001
Swarm announcing /ip4/169.254.177.21/tcp/4001
Swarm announcing /ip4/169.254.221.206/tcp/4001
Swarm announcing /ip4/169.254.80.27/tcp/4001
Swarm announcing /ip4/172.16.193.189/tcp/34335
Swarm announcing /ip4/172.23.81.17/tcp/4001
Swarm announcing /ip4/192.168.0.5/tcp/4001
Swarm announcing /ip6:::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready

```

### Screenshot for 'Start\_IPFS.bat'

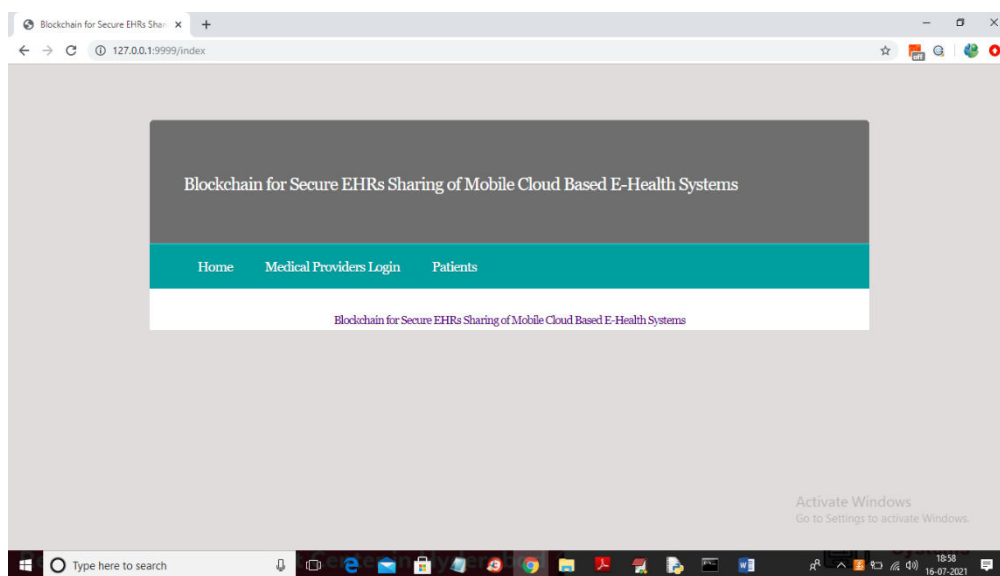
DESCRIPTION: IPFS server started and now double click on 'run.bat' file to run python FLASK server like below screen



```
C:\Windows\system32\cmd.exe
E:\NewClient\SecureEHR>python SecureEHR.py
* Serving Flask app "SecureEHR" (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 windowsapi reloader
* Debugger is active!
* Debugger PIN: 248-796-027
* Running on http://127.0.0.1:9999/ (Press CTRL+C to quit)
```

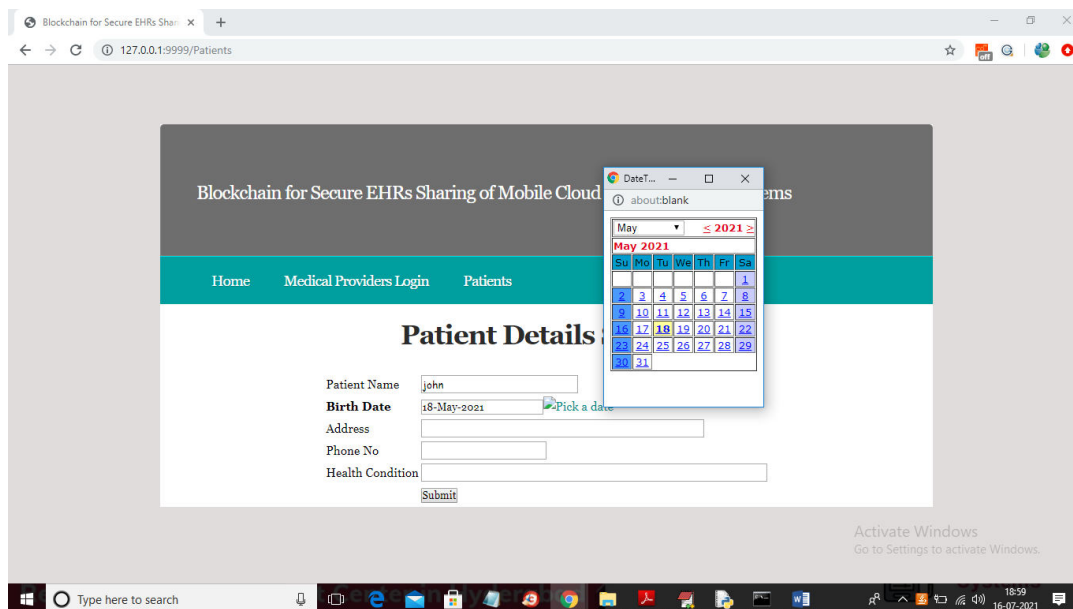
### Screenshot for 'run.bat' file'

**DESCRIPTION:** python server started and now open browser and enter URL as 'http://127.0.0.1:9999/index' and press enter key to get below page



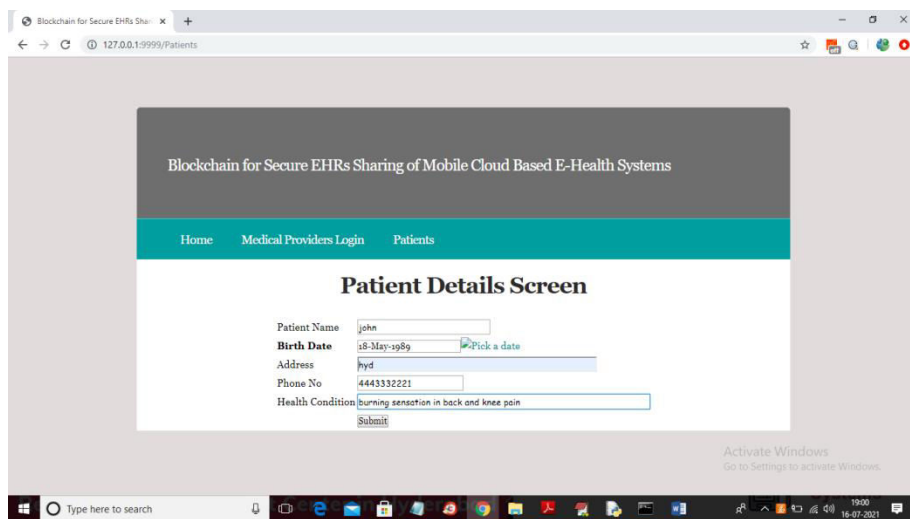
### Screenshot for Patients

**DESCRIPTION:** In above screen click on 'Patients' link to get below screen



Screenshot for Patients Details

DESCRIPTION: Patient will enter his details and then select birth date



Screenshot for Patients Details Screen Submit

DESCRIPTION: Entering patient data then click on 'Submit' button to get below output

```

C:\Windows\system32\cmd.exe
E:\NewClient\SecureEHR>javac -cp ".\lib\*" -d . *.java
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\NewClient\SecureEHR>java -cp ".\lib\*" -Xmx1000M com.deploy
Transaction complete : 0xe4025e0165d04bca36352c1c5d9b3b5458e8240c17e3935fbcf89ec6d77eb81
Deploying smart contract
Smart contract deployed to address 0x2af242047011f432a166561e14007c4362f88195
Initial value of counter in Smart contract: temp,temp
Smart Contract Ready to store data
Transaction complete : 0x08b6c5d272bee8daF9655568b2304d25b884e782b06056b321e7e8a92fe1a844 0x2af242047011f432a166561e14007c4362f88195
Address 0x2af242047011f432a166561e14007c4362f88195
1.QmQcto4ikuB1fgejEd2HepjEZP6tvm8jxrz9KM28sDwV Record saved in Ethereum
Address 0x2af242047011f432a166561e14007c4362f88195
2.QmDirkyEY4cgCeW17gkq431Sxe6g385q22h8ydsuicA87 Record saved in Ethereum
1#2
1#2
1#2
1#2
1.QmQcto4ikuB1fgejEd2HepjEZP6tvm8jxrz9KM28sDwV
1#2
2.QmDirkyEY4cgCeW17gkq431Sxe6g385q22h8ydsuicA87
Address 0x2af242047011f432a166561e14007c4362f88195
1.QmDirkyEY4cgCeW17gkq431Sxe6g385q22h8ydsuicA87 Record saved in Ethereum

```

Screenshot for Hash code returned by IPFS

DESCRIPTION: HASHCODE returned by IPFS and Blockchain and after successful storage will get below output

Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems

Home Medical Providers Login Patients

### Patient Details Screen

Record stored in IPFS and in Blockchain with ID : 1  
Hashcode = QmDirkyEY4cgCeW17gkq431Sxe6g385q22h8ydsuicA87

Patient Name

Birth Date  [Pick a date](#)

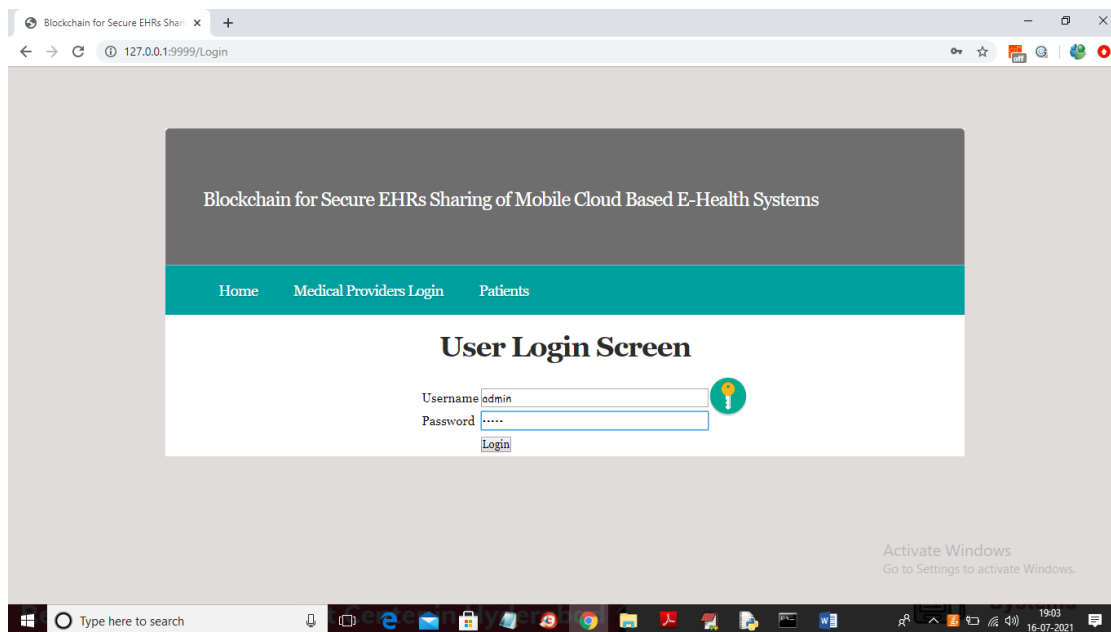
Address

Phone No

Health Condition

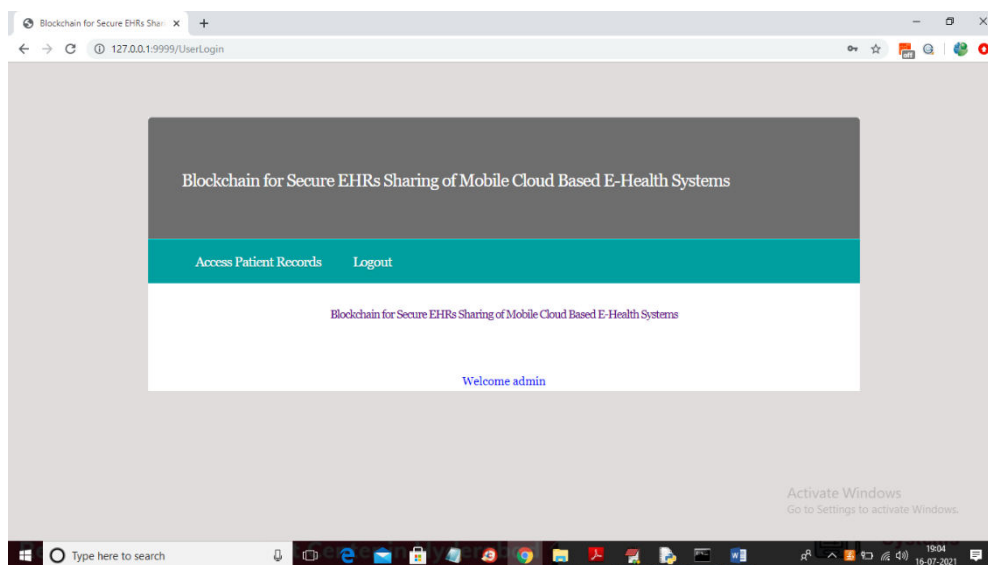
Screenshot for Hashcode

DESCRIPTION: Patient record id is generated and hashcode also and now click on 'Medical Providers Login' link to get below screen



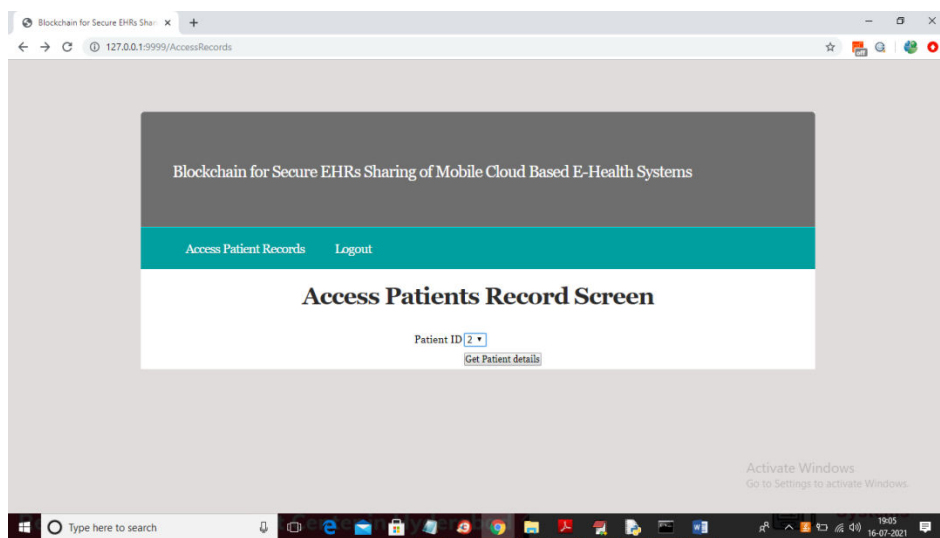
#### Screenshot for User Logi

**DESCRIPTION:** Medical providers can login by using username as 'admin' and password as 'admin' and then click on 'Login' button to get below screen



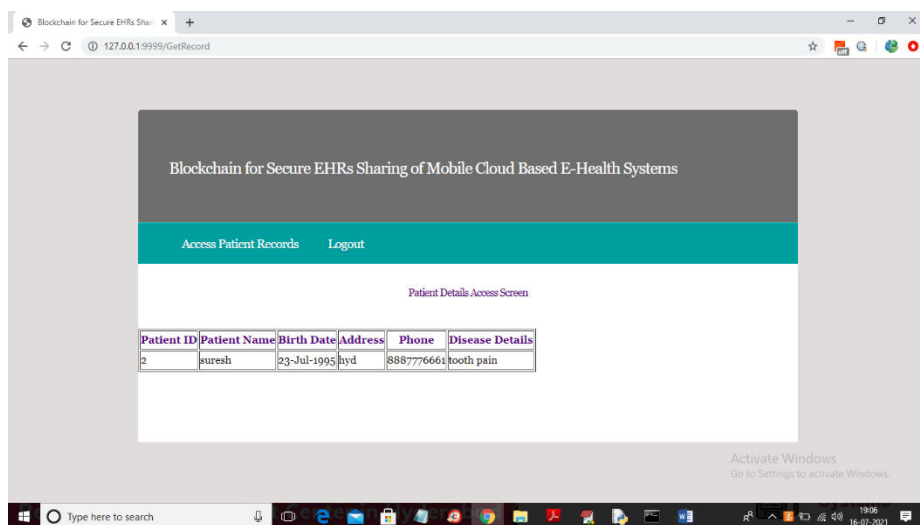
#### Screenshot for 'Access Patient Records'

**DESCRIPTION:** 'Access Patient Records' link to get below screen



#### Screenshot for 'Access Patient Records'

**DESCRIPTION:** Patient ID and then click on 'Get patient details' button to get below screen



#### Screenshot for patient details access screen

**DESCRIPTION:** Get patient details on the screen

## 7. CONCLUSION AND FUTURE SCOPE

In this project proposes a EHRs sharing scheme enabled by mobile cloud computing and blockchain. To identify critical challenges of current EHRs sharing systems and propose efficient solutions to address these issues through a real prototype implementation. In this work, our focus is on designing a trustworthy access control mechanism based on a single smart contract to manage user access for ensuring efficient and secure EHRs sharing. To investigate the performance of the proposed approach,

deploy an Ethereum blockchain on the Amazon cloud, where medical entities can interact with the EHRs sharing system via a developed mobile Android application. Integrate the peer-to-peer IPFS storage system with blockchain to achieve a decentralized data storage and data sharing. The implementation results show that our framework can allow medical users to share medical data over mobile cloud environments in a reliable and quick manner, in comparison to conventional schemes. In particular, our access control can identify and prevent effectively unauthorized access to the e-health system, aiming for achieving a desired level of patient privacy and network security. To provide security analysis and extensive evaluations on various technical aspects of the proposed system, showing advantages of our proposal over existing solutions. Based on the merits of our model, believe that our blockchain enabled solution is a step towards efficient management of e-health records on mobile clouds, which is promising in many healthcare applications.

## 8. REFERENCES

1. [1] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J. Amer. Med. Inf. Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017.
2. [2] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. 18th IEEE Int. Conf e-Health Net., Appl. Services*, Sep. 2016, pp. 1–3.
3. [3] W. J. Gordon and C. Catalini, "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability," *Computer. Struct. Biotechnol J.*, vol. 16, pp. 224–230, 2018.
4. [4] M. Hölbl, M. Kompara, A. Kamišalic, and L. N. Zlatolas, "A systematic review of the use of blockchain in healthcare," *Symmetry*, vol. 10, no. 10, p. 470, 2018.
5. [5] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Com. (PIMRC)*, Oct. 2017, pp. 1–5.
6. [6] M. Steichen, R. Norvill, B. F. Pontiveros, and W. Shbair, "Blockchain based, decentralized access control for IPFS," in *Proc. IEEE Blockchain*, Jul. 2018, pp. 1499–1506.