

A HEALTH DATA SYSTEM USING BLOCKCHAIN

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

K SANKARI (Reg.No - 39120102)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119**

MAY- 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

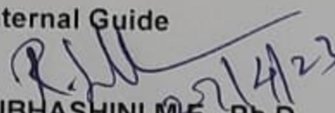
www.sathyabama.ac.in

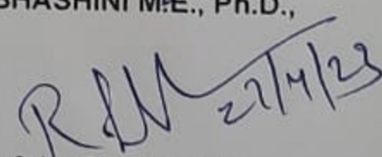
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of K SANKARI (Reg.No - 39120102) who carried out the Project Phase-2 entitled "A HEALTH DATA SYSTEM USING BLOCKCHAIN" under my supervision from January 2023 to May 2023.

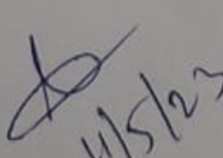
Internal Guide

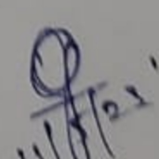

Dr. R. SUBHASHINI M.E., Ph.D.,


Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 11 MAY 2023.


Internal Examiner

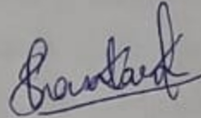

External Examiner

DECLARATION

I, **K SANKARI** (Reg No:39120102), hereby declare that the Project Phase - 2 Report entitled "**A HEALTH DATA SYSTEM USING BLOCKCHAIN**" done by me under the guidance of **DR. R. SUBHASHINI, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 11 MAY 2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. R. SUBHASHINI, M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R. SUBHASHINI, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

The advent of wireless technologies and the development of more and better globally connected mobile devices, leverage real time health monitoring. Mobile health (m-health) promises to deliver health services anytime and anywhere, improving user convenience and enabling faster diagnoses without the need to travel to healthcare facilities [5]. The use of m-health applications on mobile devices with the support of cloud computing is nowadays a technology trend that has many advantages, but also poses several challenges, especially on the data storage and privacy. Blockchain technology is an exponentially growing technology used in various research areas from finance, voting mechanisms, production chains, among others. This technology provides important characteristics such as immutability, non-repudiation, transparency, and reducing the need for intermediaries. Hence, this paper presents a novel approach for blockchain technology applied to m-health systems [7]. This proposal allows an easy and fast integration with other health systems or applications allowing a patient-user to access his/her electronic health record in a more secure way. The data is traceable throughout the system, however maintaining the necessary anonymity. Hence, a prototype for a blockchain-based solution using Hyperledger Fabric was developed. This implementation enables the creation of a chronologically organized and immutable health data record. To create an anonymous storage system, the proposed system uses two separate database components that maintain data traceability through sets of IDs stored in the blockchain. Furthermore, the development of the proposed system was evaluated in terms of performance and network configurations of the Hyper ledger Fabric.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF ABBREVIATIONS	vii
	LIST OF FIGURES	viii
1	INTRODUCTION	9
2	LITERATURE SURVEY	10
	2.1 Inferences from Literature Survey	10
	2.2 Existing System and Proposed System	11
	2.3 Open problems in Existing System	12
3	REQUIREMENTS ANALYSIS	13
	3.1 Feasibility Studies/Risk Analysis of the Project	13
	3.2 Software Requirements Specification Document	14
4	DESCRIPTION OF PROPOSED SYSTEM	15
	4.1 Selected Methodology or process model	15
	4.2 Architecture / Overall Design of Proposed System	17
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	17
	4.4 Project Management Plan	22
5	RESULTS AND DISCUSSION	23
	5.1 Results	23
	5.2 Conclusion	23
	5.3 Future Work	24
6	REFERENCE	25
7	APPENDIX	26
	A. SCREENSHOTS	26
	B. SOURCE CODE	28
	C. RESEARCH PAPER	37

LIST OF ABBREVIATIONS

WORD	ABBREVIATION
mhealth	Mobile health
EHR	Electronic Health Record
IoT	Internet of Things
AWS	Amazon Web Services
PHI	Protected Health Information
IPFS	Inter Planetary File System
API	Application Programming Interface

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.1	Architecture of Proposed System	17
5.1	Results	23
7.1	Screenshots	26

CHAPTER 1

INTRODUCTION

The continuous increase of the number of mobile devices all around the world in conjunction with significant improvements with wireless connectivity and mobile devices capabilities enabled the real-time monitoring of health-related events, transforming m-health in a hot research and development topic. [9] Mobile devices, at the moment, are capable of much more than just share data. These are equipped with multiple sensors, allowing them to work as a full-fledged monitoring device. Due to this technological evolution, health records can be transferred and updated, almost in real-time, between patients, healthcare professionals, and other authorized entities. M-health promotes the importance of self-care by the user, allowing patients to become in control of their health records and reliably self-diagnose their symptoms. [6] Furthermore, m-health offers monitoring capabilities, in real-time, of various biometric information, improving the patient's convenience and allowing for a faster diagnosis and treatment without the need for constant travel to a medical care facility. [10] The use of cloud computing in mobile applications is a very popular approach, however it poses several security challenges that can be mitigated with a correct implementation of blockchain technology. Blockchain technology adoption is growing at an exponential rate, being used in a wide area of studies beyond financial uses (popularized by the application on Bitcoin), such as vote mechanisms, supply chain monitoring, Internet of Things (IoT), data security, and others. [8] The outcome would be perfectly balanced community-wide information, with assured credibility from the point of data generation to the time of requirement, without manual human interference.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

Tomomitsu Motohash et al. a mhealth system using Blockchain was proposed. Blockchain technology intrinsic characteristics are perfectly fit for providing reliability and immutability to m-health data without any third-party contributor. However, the mobile devices used by the patients need to be authenticated and validated to avoid impersonation attacks to ensure the reliability of the data. The system purposed uses a client hash chain created in the patient mobile device and registered in the blockchain network. It was utilized Hyperledger Fabric v1.0 to implement the blockchain network. A private blockchain network was chosen for the management of medical data because of the node control of the stakeholders and since it is a private network it is possible to use different consensus protocols other than PoW allowing for the processing of more transactions. The authors tested this system in a m-health for insomnia treatment, where medical data was successfully registered and simulated illegal data was correctly identified.[1]

Dinh C. Nguyen et al. propose a novel Electronic Health Record (EHR) sharing framework combining blockchain and decentralized Inter Planetary File System (IPFS) on a mobile cloud platform. EHR on mobile cloud environments enables high flexibility and availability, facilitating medical data exchanges between patients and healthcare providers. Nevertheless, this flexibility and availability come with concerns about network security and data privacy. The objective of the work of the author was to guarantee high-security levels in the mobile cloud used to share EHR. For the implementation, it was deployed a private Ethereum blockchain network on Amazon Web Services (AWS) where various virtual machines were used as admin, as miners, and EHR manager. Since it is impossible to share and store large portions of data on a blockchain, causing scalability problems, a decentralized peer-to-peer IPFS was used to build a file system sharing platform in the blockchain network.[2]

Kristen Griggs et al. proposed the use of blockchain-based smart contracts for the management of medical sensors securely. To that end, a system using a private blockchain based on Ethereum was created, where the IoT sensors communicate with a smart device to execute smart contracts saving records of all events on the blockchain. The blockchain doesn't store confidential medical information, only storing the records that an event occurred. Medical data is stored in an EHR database, adding a new transaction to the blockchain stating the processing of the data.[3]

Aiqing Zhang et al. presented a blockchain-based secure and privacy-preserving Protected Health Information (PHI) sharing scheme to be used to improve diagnosis in ehealth systems. This implementation uses two different types of blockchain, private and consortium blockchain owned by a group of entities. The private blockchain was used to store, PHI while the consortium blockchain was used to secure the indexes of the PHI. The block generators are required proof of conformance to add a new block to the blockchain that is, the verifier needs to verify the block, checking if the PHI is generated by an authorized doctor.[4]

2.2 EXISTING SYSTEM AND PROPOSED SYSTEM

EXISTING SYSTEM

The main problem of the current health care is that the organizations hold multiple and fragmented medical records of patients. At present, medical records are extremely centralized and challenging to circulate. Obtaining critical data about a patient's health in front of a doctor shouldn't be as difficult as it is now.

PROPOSED SYSTEM

- A novel Blockchain-based m-health solution where anonymous health data is traceable through the system.
- The performance evaluation and analysis of the presented proposal.
- This system brings benefits regarding information ownership, data traceability,

and anonymity while enabling interoperability and integration with existing systems. The proposed system architecture is established by three crucial modules: a block chain module where all the new interventions events are stored, the database component where personal information and healthcare information is stored, and finally, the application and Application Programming Interface (API) where the users can communicate with the after-mentioned modules.

- The IOTA's Masked Authenticated Messaging extension was used, allowing the encryption and authenticating of data streams transmitted through the network as zero-value transactions that are transactions without the need for IOTA tokens. Masked Authenticated Messaging also allows post-quantum cryptography and forward transaction linking.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

A traditional health system or even a cloud-based mobile health system has several advantages, but also several disadvantages, including a single centralized point of failure. I.e., cloud or a single server, where all the data sent/received is processed poses as an attractive target for attackers, being subdued to security and privacy threats on health data. Health data can be susceptible to various threats, as clearly demonstrated by the increasing number of security threats and attacks on cloud-based systems and patient data since the first period of the COVID-19 crisis, including data breaches, tampering, and identity theft.

Currently, medical records are highly centralized and difficult to distribute. Getting important information about a patient's health in front of a doctor should not be as difficult as it is now.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him/her familiar with it. His/her level of confidence must be raised so that he/she is also able to make some constructive criticism, which is welcomed, as he/she is the final user of the system.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

SYSTEM REQUIREMENTS

1) HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 Ghz.
- Hard Disk : 40 GB.
- Monitor : 15 Inch VGA Color.
- Ram : 512 MB
- Keyboard : Standard Keyboard

2) SOFTWARE REQUIREMENTS

- Operating System : Windows XP.
- Platform : Python Technology
- Tool : Python 3.6
- Front End : Python Anaconda Script
- Back End : Spyder

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

ADVANTAGES OF PROPOSED SYSTEM

- Automate and streamline provider workflow
- Data can be devised and maintained by approved suppliers.
- Maintained in a sharable digital format across multiple organizations.
- Exchange data with parties such as laboratories, consultants, medical imaging centers, pharmacies, emergency departments, and clinics at school and in the workplace.
- Improves quality of care.
- Improves Diagnostics and Patient Outcomes.
- This method of real-time monitoring and recording is particularly important in critical groups of patients like chronic patients, elderly patients, patients with disabilities and young patients

AES CIPHER ALGORITHM

The hash of the original data combined with the encryption of the original text using the AES cipher algorithm and then is written in the blockchain directly.

On the other hand, if the data are multimedia files, the data encrypted in conjunction with the hash of the original data along with the encrypted index of the file's location is written in the block chain.

A single node manages to compromise the entire network while also allowing for better scalability.

Although the use of PoW allows for better scalability and security, the use of this consensus algorithm is of high consumption of time and computational power making this an implementation with high latency blockchain technology applied to m-health systems. This proposal allows an easy and fast integration with other health systems or applications allowing a patient-user to access his electronic health record in a

more secure way. The data is traceable throughout the system, however maintaining the necessary anonymity. Hence, a prototype for a blockchain-based solution using Hyperledger Fabric was developed.

This implementation enables the creation of a chronologically organized and immutable health data record. To create an anonymous storage system, the proposed system uses two separate database components that maintain data traceability through sets of IDs stored in the blockchain.

The use of m-health applications on mobile devices with the support of cloud computing is nowadays a technology trend that has many advantages, but also poses several challenges, especially on the data storage and privacy. Block chain technology is an exponentially growing technology used in various research areas from finance, voting mechanisms, production chains, among others. This technology provides important characteristics such as immutability, non-repudiation, transparency, and reducing the need for intermediaries.

Another advantage of a m-health system is the real-time motorization, all the time, providing extra information about the health condition of the patient allowing the medical professionals to realize a faster and better diagnosis. This method of real-time monitoring and recording is particularly important in critical groups of patients like chronic patients, elderly patients, patients with disabilities and young patients.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

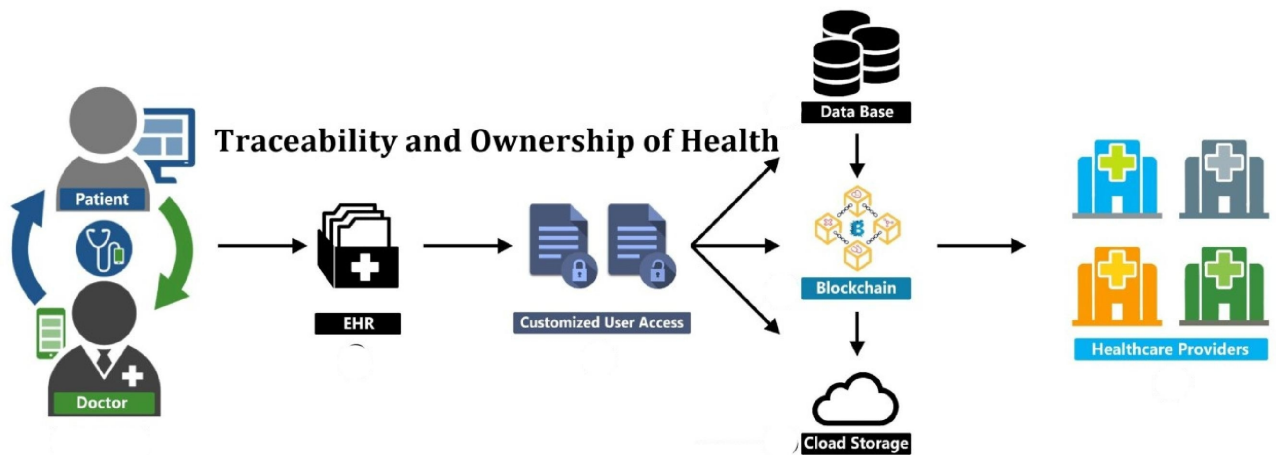


Fig 4.1 Architecture of proposed system

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

SOFTWARE DESCRIPTION

Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages.

The Anaconda distribution comes with over 250 packages automatically installed. Over 7500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI (graphical user interface), Anaconda Navigator, as a graphical alternative to the command line interface. Anaconda Navigator is included in the Anaconda distribution, and allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages and update them.

BLOCKCHAIN

The use of m-health applications on mobile devices with the support of cloud computing is nowadays a technology trend that has many advantages, but also poses several challenges, especially on the data storage and privacy. Blockchain technology is an exponentially growing technology used in various research areas from finance, voting mechanisms, production chains, among others.

Hence, a prototype for a blockchain-based solution using Hyper ledger Fabric was developed. This implementation enables the creation of a chronologically organized and immutable health data record.

SPYDER

A highly advanced Data Science Python platform. Created with Python for Python, this IDE boasts some immensely robust toolsets. With an editor, IPython Console, Variable Explorer, Advanced Plot Functionality, a built-in debugger, and object doc helper tools, the Spyder IDE is a promising choice for a large amount of Data Science tasks.

TRACEABILITY

This article presents a novel blockchain-based approach for m-health solutions, for a secure patient-user access and traceability of electronic health records. This implementation enables the creation chronologically organized and immutable health data records traceable throughout the system yet, maintaining the necessary anonymity. To create this anonymous storage system, the proposed system implements two separate database components (personal data and health data) that assures data traceability through sets of IDs stored in the blockchain module.

ELECTRONIC HEALTH RECORD

This system used a private hyper ledger Fabric network so that every electronic

health record sent to the network was capable to resist tampering and revision. The network was composed of four validating peers controlling the blockchain, and one membership service authenticating the client and the validating peers.

This proposal allows an easy and fast integration with other health systems or applications allowing a patient-user to access his electronic health record in a more secure way. The data is traceable throughout the system, however maintaining the necessary anonymity. Hence, a prototype for a blockchain-based solution using Hyperledger Fabric was developed.

INTERNET OF THINGS

The use of cloud computing in mobile applications is a very popular approach, however it poses several security challenges that can be mitigated with a correct implementation of blockchain technology. Blockchain technology adoption is growing at an exponential rate, being used in a wide area of studies beyond financial uses (popularized by the application on Bitcoin), such as vote mechanisms, supply chain monitoring, Internet of Things (IoT), data security, and others. Blockchain introduced a decentralized way to implement secure transactions between nodes in an untrustworthy network through a consensus algorithm validating the transactions for all the nodes of the network.

Blockchain introduced a decentralized way to implement secure transactions between nodes in an untrustworthy network through a consensus algorithm validating the transactions for all the nodes of the network.

The blockchain framework together with cryptographic algorithms enables the protection and immutability of the protected storage data. This system was implemented on the public Ethereum platform. According to the authors, the system displays effectiveness and efficiency during testing yet, there are still some storage optimization problems since each transaction contains a small amount of content, wasting some usable space.

If the data is a text file, it is used the SHA-256 algorithm to calculate the hash of the original data combined with the encryption of the original text using the AES cipher algorithm and then is written in the blockchain directly. On the other hand, if the data are multimedia files, the data encrypted in conjunction with the hash of the original data along with the encrypted index of the file's location is written in the block chain.

SIGNIFICANT IMPROVEMENTS

The use of cloud computing in mobile applications is a very popular approach, however it poses several security challenges that can be mitigated with a correct implementation of block chain technology. Block chain technology adoption is growing at an exponential rate, being used in a wide area of studies beyond financial uses (popularized by the application on Bit coin), such as vote mechanisms, supply chain monitoring, Internet of Things (IoT), data security, and others.

A health care provider employing medical devices or a patient using a m-health device needs to gather medical data about the patient. This medical data could fall into various categories as Patient/Disease registries, health surveys, or even more complex data like fully-fledged EHR. Before connecting to the proposed system, there needs to be a robust authentication and communication system to support the proposed system. To validate the system and guarantee security, privacy and the identity of the user, this authentication and communication system must implement several SOTA cryptography techniques based on elliptic curve cryptography, public-key cryptography, SSL and several key exchange protocols. The gathered data is sent to the API module that stores the data in a database module and generates a link to connect the block chain module to the health record database and stores the transaction identifier on another database implementation.

The possibility of having the availability of personal health records grants the patient more information and more ownership of the health records. In the case of auditing, a user with admin privileges on a peer can access and gather all the transaction records from the blockchain, working as an immutable log. In this case, the auditor cannot access any health care information, but can collect access records and intervention records with the identifiers of the users in the system.

Regarding performance, the proposed system utilizes a private blockchain that greatly improves the throughput and latency compared to a public solution. The chaincode invoked is also simple with reduced data without the need for substantial

computational power. Another important factor achieved by an implementation with Hyperledger Network is the separation between the ordering service (service responsible for a consistent and final blockchain state) and the peers, providing many advantages in terms of performance and scalability. The performance of the proposed system in a test environment was evaluated and is analyzed deeper/

The system requirements and concept. Namely, an m-health system integrated with blockchain technology. This system brings benefits regarding information ownership, data traceability, and anonymity while enabling interoperability and integration with existing systems.

The proposed system architecture is established by three crucial modules: a block chain module where all the new interventions events are stored, the database component where personal information and healthcare information is stored, and finally, the application and Application Programming Interface (API) where the users can communicate with the after-mentioned modules.

As previously mentioned, the IOTA's Masked Authenticated Messaging extension was also used, allowing the encryption and authenticating of data streams transmitted through the network as zero-value transactions that are transactions without the need for IOTA tokens. Masked Authenticated Messaging also allows post-quantum cryptography and forward transaction linking.

A traditional health system or even a cloud-based mobile health system has several advantages, but also several disadvantages, including a single centralized point of failure. I.e., cloud or a single server, where all the data sent/received is processed poses as an attractive target for attackers, being subdued to security and privacy threats on health data. Health data can be susceptible to various threats, as clearly demonstrated by the increasing number of security threats and attacks on cloud-based systems and patient data since the first period of the COVID-19 crisis, including data breaches, tampering, and identity theft. A mobile health system could integrate the benefits of blockchain technology to provide:

- (i) A distributed ledger without a single point of failure;
- (ii) privacy assurances by being possible to share data via transactions unreadable by other parties, and
- (iii) Confidentiality, integrity, and availability, inherent to the distributed nature of the blockchain technology, with signed blocks linked to each other via hashes the

performance of the proposed system and the Hyperledger Fabric network are evaluated. This performance evaluation was composed of various tests in order to test the configuration of the Hyperledger Network and the entire system. To guarantee the reproducibility of the data acquired during the system testing, all environment parameters of the test network utilized will be listed. These considerations were made based on the white paper by the Hyperledger Performance and Scale Working Group. This test network was implemented on a remote virtual server with 4 GB of RAM, 2 cores, and a processor clock speed of 3000 MHz running an installation of Ubuntu 20.04.2 LTS. Hence, it can be stated that referring to the geographic distribution of the nodes utilized in the test environment, all the nodes in the system are located in the same machine. In relation to the network model, a simple network with three organizations was utilized, each one with one node (three nodes total) on which all the transactions are broadcasted in between. The consensus protocol chosen was RAFT by being the easiest to implement and the only one fully supported by the documentation of Hyperledger Fabric.

4.4 PROJECT MANAGEMENT PLAN

January	Research, Base papers, Literature survey, study on Blockchain, IoT
February	Collecting the relevant data, Implementing the code, Front – end works
March	Back- end, Output

CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULTS

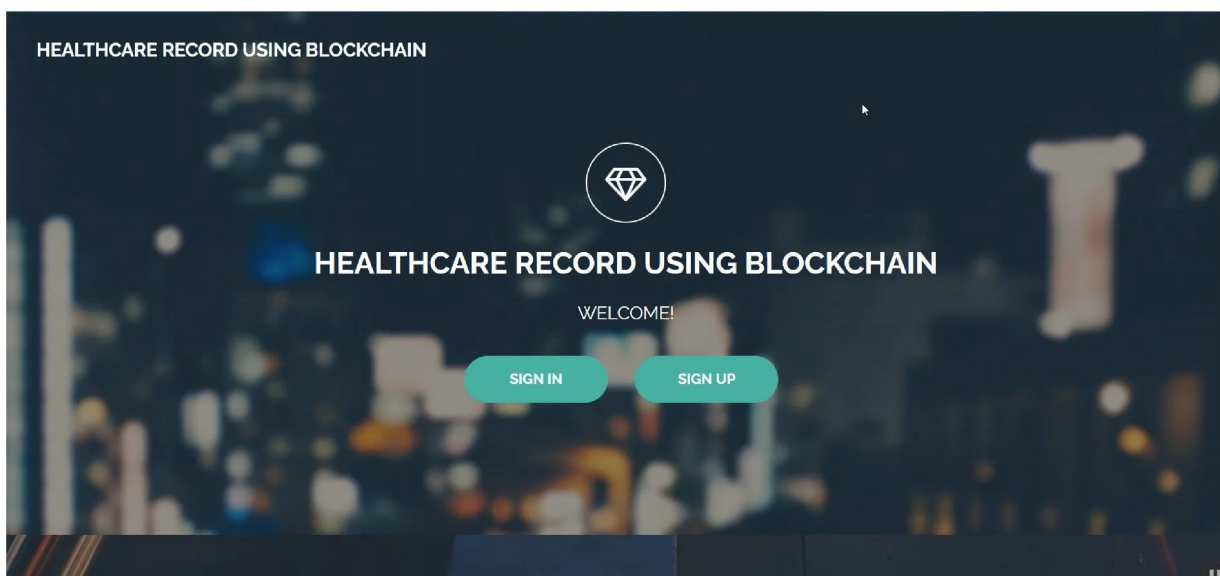


Fig 5.1 The main page of the project

5.2 CONCLUSION

Adopting the Blockchain to deploy the EHR solves the significant issues of accessibility and authority. The Blockchain allows ease of access to the records as it is available to any personnel authorized to access them. Since it is deployed on a Distributed network, it can be accessed from anywhere. However, there are a few issues that could still be addressed in the future. Since it is possible to distinguish the people engaged in the transactions, this could endanger their confidentiality and secrecy. Although certain block chains offer absolute anonymity, but some sensitive details should clearly not be distributed in this way. The system can be improved by allowing the medical records to be accessed quickly during emergencies. Its structure could be modified to fit a specific disease or modeled on established standards.

5.3 FUTURE WORK

- All stakeholders involved with the patients will have time-limited access to the EHR. The EHR is shared in the form of a Smart Contract.
- The information is always kept private and secure.
- A zero-knowledge protocol is a means by which one party (the prover) confirms to the other party (the verifier) without revealing any information apart from the fact that this specific statement is true. The use of zero-knowledge proof increases privacy.
- Blockchain being universal will ensure that the EHR framework will be compatible across multiple health applications.

CHAPTER 6

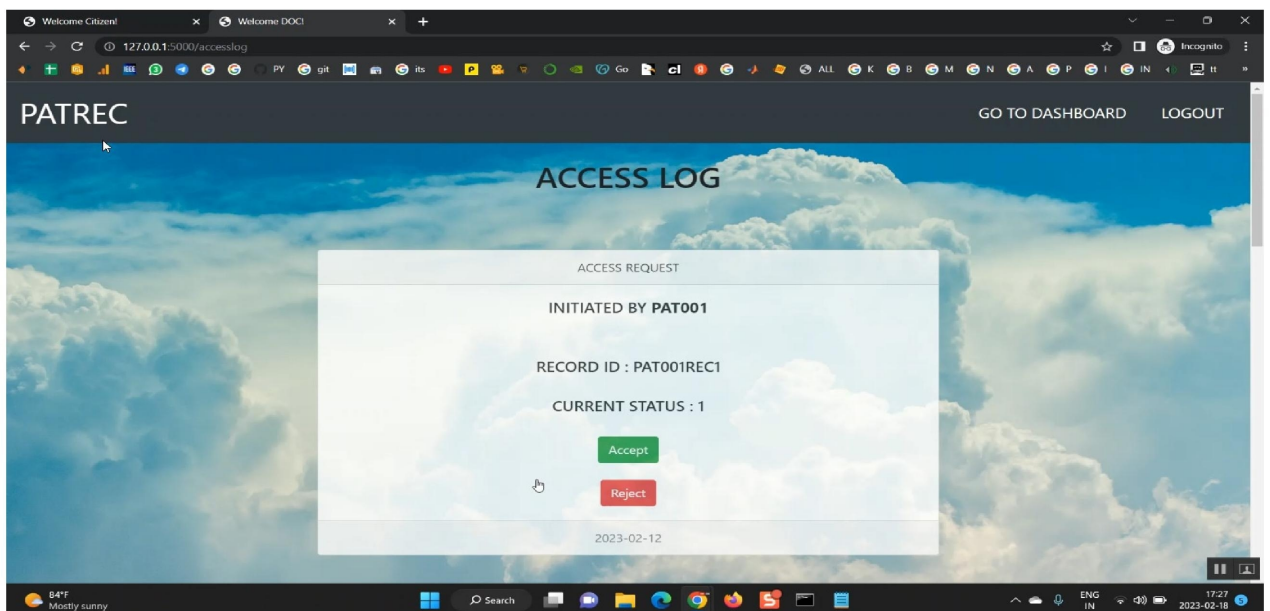
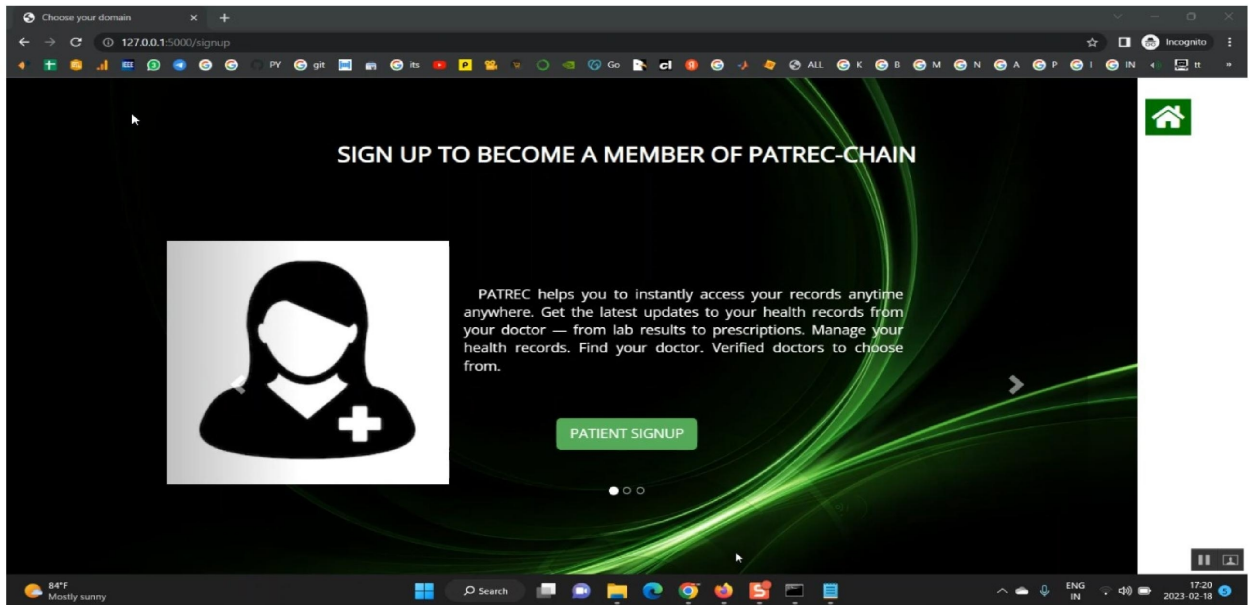
REFERENCES

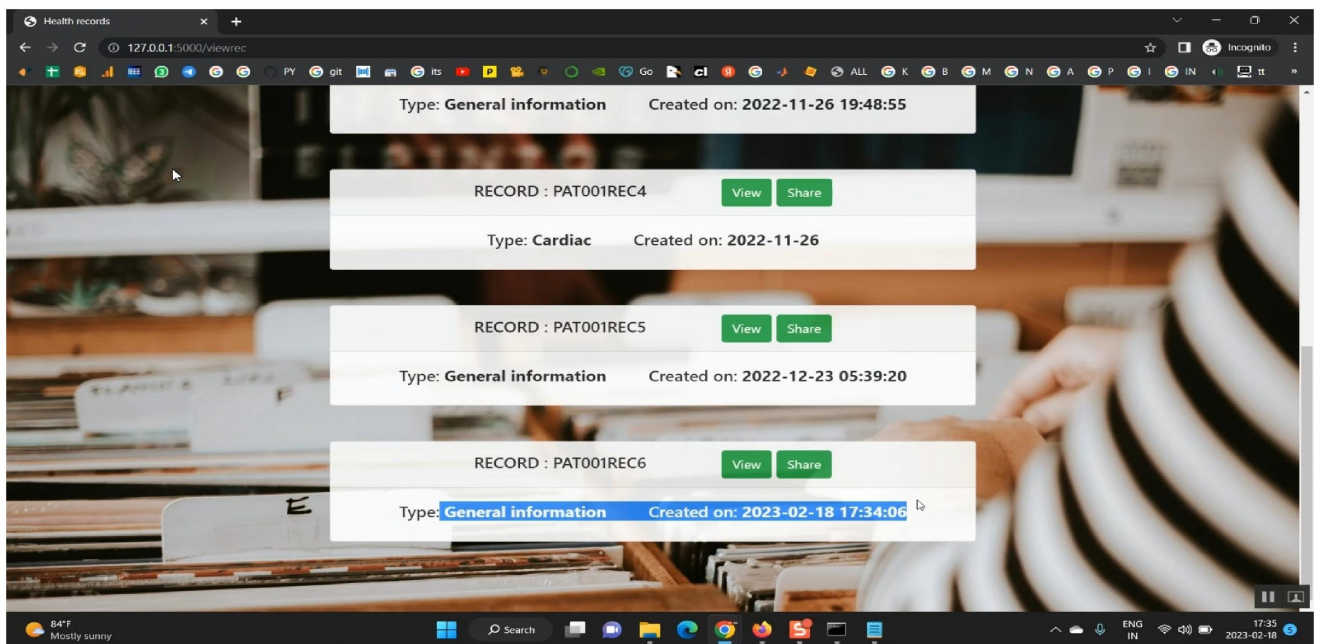
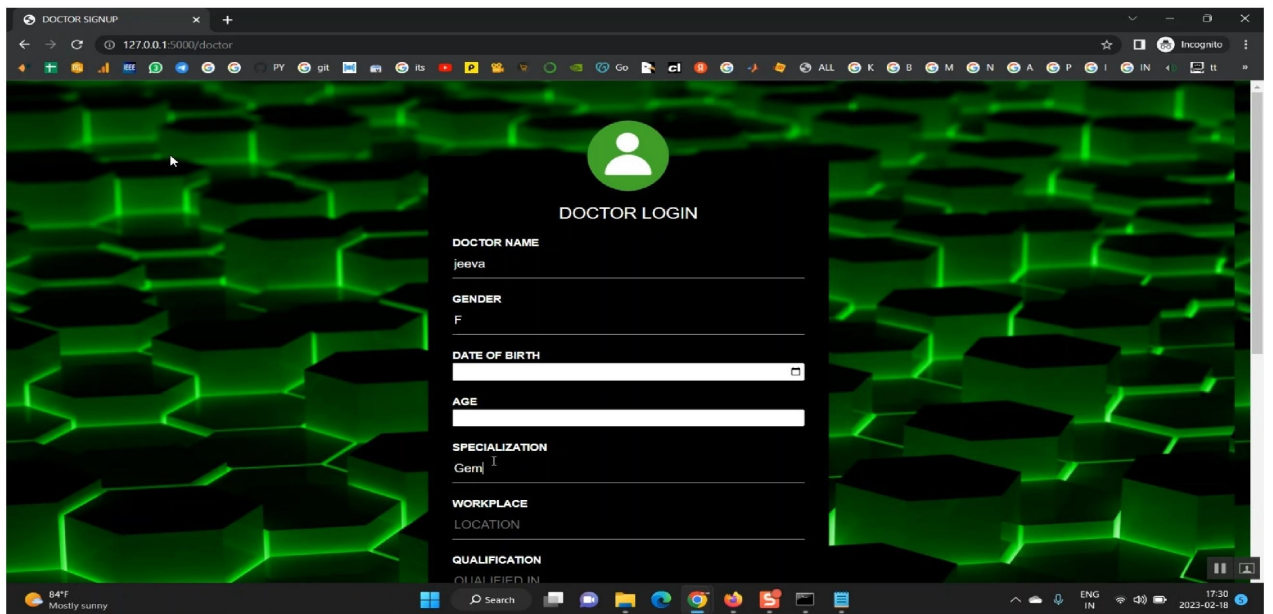
- [1] Tomomitsu Motohash et al. Secure and scalable mhealth data management using Blockchain combined with client hashing. *Journal of Medical Internet Research* (2019).
- [2] Dinh C. Nguyen. Blockchain for secure EHRs sharing of Mobile cloud based EHealth system. *IEEE Access* (2019).
- [3] Kristen N. Griggs, Olya Ossipova, Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring. *Journal of Medical Systems* (2018).
- [4] Aiqing Zhang and Xiaodong Lin. Towards secure and privacy-preserving data sharing in ehealth systems via blockchain consortium. *Journal of Medical Systems* (2018).
- [5] Santos, J., Silva, B. & Inácio, P. A Blockchain System for Mobile Health Applications and Services. *Universidade da Beira Interior*, (2019).
- [6] Steinhubl, S., Muse, E. & Topol, E. Can mobile health technologies transform health care? *Jama*. 310, 2395-2396 (2013).
- [7] Taralunga, D. & Florea, B. A Blockchain-Enabled Framework for mHealth Systems. *Sensors*. 21 (2021).
- [8] Kumar, S., Nilsen, W., Pavel, M. & Srivastava, M. Mobile Health: Revolutionizing Healthcare Through Transdisciplinary Research. *IEEE Computer Society*. 46 pp. 28-35 (2013,1).
- [9] Silva, B., Lopes, I., Rodrigues, J. & Ray, P. Sapo Fitness: A mobile health application for dietary evaluation. 2011 IEEE 13th International Conference on EHealth Networking, Applications And Services. 310, 375-380 (2011).
- [10] Lorenz, A. & Oppermann, R. Mobile health monitoring for the elderly: Designing for diversity. *Pervasive And Mobile Computing*. 5, 478–495 (2009).

CHAPTER 7

APPENDIX

7.1 SCREENSHOTS





7.2 SOURCE CODE

```

from django.shortcuts import render,redirect,get_object_or_404
from . models import *
from django.contrib import messages
import datetime
from django.db.models import Q
from django.db import connection
from django.db.models import Sum, Count
from django.conf import settings
from django.utils import timezone
import os
from cryptography.fernet import Fernet
import random
import string
from django.core.mail import send_mail
from django.core.mail import EmailMessage
def public_key(length):
    sample_string = 'd0LW25jG8feETs4WWpeCUA4AU1oPj7IAcCtKB1Cmuso=' #
define the specific string
    # define the condition for random string
    result = ''.join((random.choice(sample_string)) for x in range(length))
    return result
def private_key(length):
    sample_string = 'd0LW25jG8feETs4WWpeCUA4AU1oPj7IAcCtKB1Cmuso=' #
define the specific string
    # define the condition for random string
    result = ''.join((random.choice(sample_string)) for x in range(length))
    return result
import datetime
def home(request):
    return render(request,'index.html',{})
def register(request):
    if request.method == 'POST':
        username = request.POST.get('username')

```

```

address = request.POST.get('address')
mobile= request.POST.get('mobile')
email = request.POST.get('email')
password = request.POST.get('password')
fname = request.POST.get('fname')
lname = request.POST.get('lname')
country = request.POST.get('country')
city = request.POST.get('city')
zip = request.POST.get('zip')
user_type = request.POST.get('user_type')
crt = Register_Detail.objects.create(username=username,

address=address,mobile=mobile,password=password,email=email,fname=fname,lname=lname,

city=city,country=country,zip=zip,user_type=user_type)
if crt:
    messages.success(request,'Registered Successfully')
return render(request,'register.html',{})
def user_login(request):
    if request.session.has_key('username'):
        return redirect("user_dashboard")
    else:
        if request.method == 'POST':
            username = request.POST.get('username')
            password = request.POST.get('password')
            user_type = request.POST.get('user_type')
            post =
Register_Detail.objects.filter(username=username,user_type=user_type,password=password)
            if post:
                username = request.POST.get('username')
                request.session['username'] = username
                request.session['user_type']=user_type

```

```

        a = request.session['username']
        sess =
Register_Detail.objects.only('id').get(username=a).id
        request.session['user_id']=sess
        return redirect("user_dashboard")
    else:
        messages.success(request, 'Invalid Username or
Password')
        return render(request,'login.html',{})
def user_dashboard(request):
    if request.session.has_key('username'):
        return render(request,'user_dashboard.html',{})
    else:
        return render(request,'login.html',{})
def logout(request):
    try:
        del request.session['username']
        del request.session['user_id']
    except:
        pass
    return render(request, 'login.html', {})
def aa_login(request):
    if request.session.has_key('aa'):
        return redirect("aa_dashboard")
    else:
        if request.method == 'POST':
            username = request.POST.get('username')
            password = request.POST.get('password')
            post =
CA_Login.objects.filter(username=username,password=password)
            if post:
                username = request.POST.get('username')
                request.session['aa'] = username
                a = request.session['aa']

```

```

        sess = CA_Login.objects.only('id').get(username=a).id
        request.session['aa_id']=sess
        return redirect("aa_dashboard")
    else:
        messages.success(request, 'Invalid Username or
Password')
        return render(request,'aa_login.html',{})
def aa_dashboard(request):
    if request.session.has_key('aa'):
        return render(request,'aa_dashboard.html',{})
    else:
        return render(request,'aa_login.html',{})
def aa_logout(request):
    try:
        del request.session['aa']
        del request.session['aa_id']
    except:
        pass
    return render(request, 'aa_login.html', {})

def upload_file(request):
    if request.session.has_key('username'):
        uid = request.session['user_id']
        user_id = Register_Detail.objects.get(id=int(uid))
        if request.method == 'POST':
            name = request.POST.get('name')
            notes = request.POST.get('notes')
            a = request.FILES['file']
            crt =
Upload_File.objects.create(user_id=user_id,name=name,notes=notes,
                             file=a)
            if crt:
                messages.success(request, 'File Uploaded to Cloud.')
                cursor = connection.cursor()

```

```

        sql=""select f.file,f.id from app_upload_file as f order by
f.id DESC"" post = cursor.execute(sql)
        row = cursor.fetchone()
        a = str(row[0])
        b = str(row[1])
        directory = os.getcwd()
        file_name = directory+"/media/"
        img = file_name+a
        class Encryptor():

            def key_create(self):
                key = Fernet.generate_key()
                return key

            def key_write(self, key, key_name):
                with open(key_name, 'wb') as mykey:
                    mykey.write(key)

            def key_load(self, key_name):
                with open(key_name, 'rb') as mykey:
                    key = mykey.read()
                return key

            def file_encrypt(self, key, original_file,
encrypted_file):

                f = Fernet(key)

                with open(original_file, 'rb') as file:
                    original = file.read()

                encrypted = f.encrypt(original)

```



```

        with open(encrypted_file, 'wb') as file:
            file.write(encrypted)

def file_decrypt(self, key, encrypted_file,
decrypted_file):

    f = Fernet(key)

    with open(encrypted_file, 'rb') as file:
        encrypted = file.read()

    decrypted = f.decrypt(encrypted)

    with open(decrypted_file, 'wb') as file:
        file.write(decrypted)

encryptor=Encryptor()

mykey=encryptor.key_create()

encryptor.key_write(mykey, file_name+a+'.key')

loaded_key=encryptor.key_load(file_name+a+'.key')

encryptor.file_encrypt(loaded_key, img,
file_name+'enc_'+a)

encryptor.file_decrypt(loaded_key, file_name+'enc_'+a,
file_name+'dec_'+a)
        return render(request,'upload_file.html',{})
    else:
        return render(request,'index.html',{})
def files(request):
    if request.session.has_key('username'):
        uid = request.session['user_id']

```

```

        a = Upload_File.objects.filter(user_id=int(uid))
        return render(request,'files.html',{'b':a})
    else:
        return render(request,'index.html',{})
def delete(request,pk):
    if request.session.has_key('username'):
        uid = request.session['user_id']
        a = Upload_File.objects.filter(id=pk).delete()
        return redirect('files')
    else:
        return render(request,'index.html',{})
def search_file(request):
    if request.session.has_key('username'):
        a = Upload_File.objects.all().order_by('-id')

        return render(request,'search_file.html',{'b':a})
    else:
        return render(request,'index.html',{})
def all_users(request):
    if request.session.has_key('aa'):
        uid = request.session['aa_id']
        a = Register_Detail.objects.all()
        return render(request,'all_users.html',{'b':a})
    else:
        return render(request,'index.html',{})
def send_request(request,pk):
    if request.session.has_key('username'):
        uid = request.session['user_id']
        file_id = Upload_File.objects.get(id=pk)
        user_id = Register_Detail.objects.get(id=int(uid))
        a =
File_Request.objects.create(file_id=file_id,user_id=user_id,status='Pending')
        return redirect('requested_file')
    else:

```

```

        return render(request,'index.html',{})
def requested_file(request):
    if request.session.has_key('username'):
        uid = request.session['user_id']
        a = File_Request.objects.filter(user_id=int(uid))
        return render(request,'requested_file.html',{'b':a})
    else:
        return render(request,'index.html',{})
def send_key(request):
    if request.session.has_key('aa'):
        uid = request.session['aa_id']
        a = File_Request.objects.all()
        return render(request,'send_key.html',{'b':a})
    else:
        return render(request,'index.html',{})
def generate_aakey(request,pk):
    if request.session.has_key('aa'):
        uid = request.session['aa_id']
        pkey = public_key(20)
        b = private_key(25)
        a =
File_Request.objects.filter(status='Pending',id=pk).update(status='Send',
        public_key=pkey,private_key=b)
        recipient_list = [request.GET.get('email')]
        email_from = settings.EMAIL_HOST_USER
        b = EmailMessage(' Your SPK to Download Requested File',
        'SPK: ' + pkey,email_from,recipient_list).send()
        return redirect('send_key')
    else:
        return render(request,'index.html',{})

def download_file(request,pk):
    if request.session.has_key('username'):
        uid = request.session['user_id']

```

```

        if request.method == 'POST':
            pkey = request.POST.get('pkey')
            prkey = request.POST.get('prkey')
            detail =
File_Request.objects.filter(public_key=pkey,private_key=prkey,id=pk)
            if detail:
                return render(request,'download_file.html',
{'b':detail,'pk':pk})
            else:
                messages.success(request, 'You have entered wrong
keys pls check the keys.')

        return render(request,'download_file.html',{'pk':pk})
    else:
        return render(request,'index.html',{})
def view_keys(request,pk):
    a = File_Request.objects.filter(id=pk)
    return render(request,'view_keys.html',{'b':a})

```

C. RESEARCH PAPER