

BLOCKCHAIN HEALTHCARE SYSTEM

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering degree in Computer Science and Engineering

By

DIVYANSHU TRIPATHI (37110198)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING 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 - 600 119

March - 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC I 12B Status by UGC I
Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTM ENT OF COMPUTER SCIENCE AND ENGINEERING BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Mr **DIVYANSHU TRIPATHI (37110198)** who carried out the project entitled " **BLOCKCHAIN HEALTHCARE SYSTEM**" under my supervision from November 2020 to March 2021.

Mr. R. Sethuraman B.E.,M.E.,(Ph.D)

Internal Guide

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **DIVYANSHU TRIPATHI (37110198)** hereby declare that the Project Report entitled **BLOCKCHAIN HEALTHCARE SYSTEM** done by me under the guidance of **Mr. R. Sethuraman B.E.,M.E.,(Ph.D)** (Internal) and (External) at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree In **CSE**

DATE:

PLACE:

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 and **Dr.L.Lakshmanan M.E., Ph.D.,and Dr.S.Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering 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 **Mr. R. Sethuraman B.E.,M.E.,(Ph.D)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Blockchain have been an interesting research area for a long time and the benefits it provides have been used by a number of various industries. Similarly, the healthcare sector stands to benefit immensely from the blockchain technology due to security, privacy, confidentiality and decentralization. Nevertheless, the Electronic Health Record (EHR) systems face problems regarding data security, integrity and management. In this paper, we discuss how the blockchain technology can be used to transform the EHR systems and could be a solution of these issues. We present a framework that could be used for the implementation of blockchain technology in healthcare sector for EHR. The aim of our proposed framework is firstly to implement blockchain technology for EHR and secondly to provide secure storage of electronic records by defining granular access rules for the users of the proposed framework. Moreover, this framework also discusses the scalability problem faced by the blockchain technology in general via use of off-chain storage of the records. This framework provides the EHR system with the benefits of having a scalable, secure and integral blockchain-based solution with use of sha256 hashing algorithm.

LIST OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	iv
	LIST OF TABLES	
1	INTRODUCTION	
	1.1 INTRODUCTION TO PROJECT	1
		2
2	LITERATURE REVIEW	
	2.1 SECURING BLOCKCHAIN BASED ELECTRONIC HEALTH RECORD USING MULTILEVEL AUTHENTICATION	4
	2.2 HEALTH RECORD MANAGEMENT THROUGH BLOCKCHAIN TECHNOLOGY	4
	2.3 ACCESS CONTROL FOR ELECTRONIC HEALTH RECORDS WITH HYBRID BLOCKCHAIN-EDGE ARCHITECTURE	5
		7
	2.4 ACCESS CONTROL FOR ELECTRONIC HEALTH RECORDS WITH HYBRID BLOCKCHAIN- EDGE ARCHITECTURE	
		16
	2.5 SECURITY AND PRIVACY OF ELECTRONIC HEALTH RECORDS SHARING USING HYPERLEDGER FABRIC	

2.6 EACMS: EMERGENCY ACCESS
CONTROL MANAGEMENT SYSTEM FOR
PERSONAL HEALTH RECORD BASED ON
BLOCKCHAIN

17

	2.7 BLOCKCHAIN-BASED PERSONAL HEALTH RECORDS SHARING SCHEME WITH DATA INTEGRITY VERIFIABLE	17
		18
	2.8 AN EFFICIENT AUTHENTICATION SCHEME FOR BLOCKCHAIN-BASED ELECTRONIC HEALTH RECORDS	
	2.9 USING HEALTHCARE AUTHORITY AND PATIENT BLOCKCHAINS TO DEVELOP A TAMPER-PROOF RECORD TRACKING SYSTEM	18
	3.8 BLOCKCHAIN AI FRAMEWORK FOR HEALTHCARE RECORDS MANAGEMENT: CONSTRAINED GOAL MODEL	18
3	SYSTEM ANALYSIS	
	4.1 ECONOMIC L FEASIBILITY	19
	4.2 TECHNICAL FEASIBILITY	20
	4.3 SOCIAL FEASIBILITY	20
4	SOFTWARE REQUIREMENT SPECIFICATIONS	
	5.1 HARDWARE REQUIREMENTS	21

	5.2 SOFTWARE REQUIREMENTS	22
	5.3 LANGUAGE DESCRIPTION	22
5	SYSTEM DESIGN	
	5.1 UML DIAGRAM OF THE PROJECT	23
	5.2 USE CASE DIAGRAM	24
	5.3 SEQUENCE DIAGRAM	26
	5.4 ACTIVITY DIAGRAM	27
	5.5 FLOW CHART	29
	5.6 TESTING OBJECTIVES	30
	5.7 TYPES OF TESTS	30
	5.8 TEST STRATEGY AND APPROACH	33
6	OUTPUT SCREENSHOTS	35
7	CONCLUSION	51
	REFERENCES	52
	APPENDIX	53

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.2	ANACONDA NAVIGATOR	19
4.2	APPLICATION IN ANACONDA DISTRIBUTION	20
4.2	FEATURES OF ANACONDA	21
5.0	SYSTEM DESIGN	22
5.2	USE CASE DIAGRAM	25
5.3	SEQUENCE DIAGRAM	26
5.4	ACTIVITY DIAGRAM	28
5.5	FLOW CHART	29

CHAPTER-1

1.1 INTRODUCTION TO PROJECT

Many researchers have tried to apply artificial intelligence into the field of healthcare to assist doctors in the diagnosis and treatment of diseases. Besides, some medical decision-making systems based on processing to search treatment solutions in a large number of unstructured literatures and patient records such as the patients' medical history, symptoms, and laboratory tests to provide decision supports for doctors. Although smart healthcare systems have made impressive progress in recent years, situations in practical applications are still far from ideal. On the one hand, despite most doctors have a good knowledge of their own fields, more and more diseases require cross-border medical knowledge of experts from different backgrounds, thus it becomes necessary to have them work together using technological means. On the other hand, due to the regional and individual differentiations among patients, the demands for precise medical care, personalized diagnosis, and treatment are increasing, which makes the accurate patient picturing ever more important than before.

CHAPTER – 2

LITERATURE

REVIEW

1 Title : Securing Blockchain based Electronic Health Record using g Multilevel Authentication

Author : B.L. Radhakrishnan , A. Sam Joseph ,S.. Sudhakar

2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)

Electronic health records possess the patient's medication details and their health history. The health records attract the attention of the attackers' as it possesses invaluable information. Loss of electronic health record leads to a wrong medication or surgery. Healthcare systems provide fewer security measures to secure the health records. Blockchain is a distributed and decentralized ledger that plays a vital role in securing the data and transactions. Introduction of blockchain in the healthcare systems protects the health records from the attackers. However, the blockchain faces phishing, dictionary-based, cold wallets, and hot wallets attacks. This paper proposes a multilevel authentication-based scheme to protect the blockchain from the attacks mentioned above.

2 Title : Health Record Management through Blockchain Technology

Author : V M Harshini ; Shreevani Danai ; H R Usha ; Manjunath R Kounte

2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)

The world is moving towards progress, to achieve the desired progress, the world should

have a healthy population and health records are the projections of an individual's health over time. The centralised approach of maintaining the health records lead to data

breaches. According to 2017 Ponemon Cost of Data Breach Study, the cost of the data breach for healthcare organizations estimated to be \$380 per record. According to 2016 Breach Barometer Report, 27,314,647 patient records were affected. So we moved towards institution-driven approach of record maintenance, which didn't make much difference with the previously existing one. Since the patient have no control over the data, the chances of data being misused is high. So we need a patient-centered approach which is completely decentralised, which can identify data thefts, prevent data manipulation, and patient has the right in access control. Blockchain Technology serves as a best solution to address all the problems and fulfill the needs. Blockchain being a decentralised and distributed ledger it can also impact on billing, record sharing, medical research, identify thefts and financial data crimes in days to come. Implementation of smart contracts in health care can simplify things even better. Where invoking, record creation and validation will be done on Blockchain. This paper highlights on the patient-driven model of record maintenance using Blockchain technology where smart contracts can be incorporated in future days making it more potential in data exchange. Finding its huge scope, hoping that more researches will be carried out and practically implemented.

3 **Title :** Access Control for Electronic Health Records with Hybrid Blockchain-Edge Architecture

Author : Hao Guo ; Wanxin Li ; Mark Nejad ; Chien-Chung Shen
2019 IEEE International Conference on Blockchain (Blockchain)

The global Electronic Health Record (EHR) market is growing dramatically and expected to reach \$39.7 billions by 2022. To safe-guard security and privacy of EHR, access control is

an essential mechanism for managing EHR data. This paper proposes a hybrid architecture

to facilitate access control of EHR data by using both blockchain and edge node. Within the architecture, a blockchain-based controller manages identity and access control policies and serves as a tamper-proof log of access events. In addition, off-chain edge nodes store the EHR data and apply policies specified in Abbreviated Language For Authorization (ALFA) to enforce attribute-based access control on EHR data in collaboration with the blockchain-based access control logs. We evaluate the proposed hybrid architecture by utilizing Hyperledger Composer Fabric blockchain to measure the performance of executing smart contracts and ACL policies in terms of transaction processing time and response time against unauthorized data retrieval.

4 Title : Patient-centric medication history recording system using blockchain

Author: Ji Woong Kim , Ah Ra Lee , Min Gyu Kim , Il Kon Kim , Eun Joo Lee
2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)

Medication errors are one of the problems to be solved not only in Korea but also across the world. To prevent a medication accident in advance, the comprehensive management of individual medication history is essential. Currently, in the hospital-centric Personal Health Record system, if the patient is prescribed medicines at multiple hospitals, he/she needs to go to the hospital where a medicine was prescribed to get the corresponding needs such as prescription or medical certification to record the prescription information on his/her own. Alternatively, the patient needs to record the prescription information on his/her own. This is very time consuming, inconvenient, and even worse, not reliable method because of input errors. Therefore, in this study, we developed Patient-centric medication

history recording system using blockchain, which is directly capturing QR code printed on

the envelop by drug store based on prescription. All the information are stored using the hash value of the data in a blockchain and it prevents the tampering of the data. In addition, this system adopted Fast Healthcare Interoperability Resources, which is the international health information exchange standard, as way to improve interoperability.

5 Title : Security and Privacy of Electronic Health Records Sharing Using Hyperledger Fabric

Author: Tagrid Alshalali , Kenneth M'Bale , Darsana Josyula
2018 International Conference on Computational Science and Computational Intelligence (CSCI)

Electronic medical records and patient data sharing are critical and considered as core issues in health care. How to store patient's information securely, how to access the information and how to ensure the privacy of patients when sharing dical data among several health service providers or agents are critical considerations. To handle those crucial considerations, a blockchain-based technology called Hyperledger Fabric will be useful. Hyperledger Fabric is a permissioned blockchain technology that provides a way to secure the interactions among a group of identified participants. In this paper, we will show how the implementation of Hyperledger Fabric to store, manage and maintain electronic medical records can ensure the security and the privacy of patient data.

6 Title : EACMS: Emergency Access Control Management System for Personal Health Record Based on Blockchain

Author: Ahmed Raza Rajput , Qianmu Li , Milad Taleby Ahvanooey , Isma Masood, IEEE Access, Year: 2019

Personal health records (PHRs) are private and vital assets for every patient. There have

been introduced many works on various aspects of managing and organizing the PHR so far. However, there is an uncertain remaining issue for the role of PHR in emergencies. In a traditional emergency access system, the patient cannot give consent to emergency staff for accessing his/her PHR. Moreover, there is no secured record management of patient's PHR, which reveals highly confidential personal information, such as what happened, when, and who has access to such information. This paper proposes an emergency access control management system (EACMS) based on permissioned blockchain hyperledger fabric and hyperledger composer. In the proposed system, we defined some rules using the smart contracts for emergency condition and time duration for the emergency access PHR data items that patient can assign some limitations for controlling the PHR permissions. We analyzed the performance of our proposed framework by implementing it through the hyperledger composer based on the response time, privacy, security, and accessibility. The experiments confirm that our framework provides better efficiency compared with the traditional emergency access system.

7 **Title:** Blockchain-Based Personal Health Records Sharing Scheme With Data Integrity Verifiable

Author: Shangping Wang , Dan Zhang , Yaling Zhang, IEEE Access, Year: 2019

The sharing of personal health records can help to improve the accuracy of the doctor's diagnosis and to promote the progress of medical research. Currently, to reduce the maintenance cost of data, personal health records are usually outsourced to a third party such as the cloud service provider. In this case, patients may lose direct control over their personal health records and the semi-trusted cloud service provider may tamper with or

reveal personal health records. Therefore, ensuring the privacy and integrity of personal

health records and realizing the fine-grained access control are crucial issues when personal health records are shared. As a distributed architecture with decentralized and tamper-proof features, blockchain provides a new way to protect the personal health records sharing system. In this paper, we propose a new personal health records sharing scheme with data integrity verifiable based on blockchain. Aiming at the problems of privacy disclosure, limited keyword search ability and loss of control rights in the process of personal health record sharing, the new scheme uses searchable symmetric encryption and attribute-based encryption techniques to achieve privacy protection, keyword search, and fine-grained access control. Compared with the existing similar schemes, the new scheme allows patients to distribute attribute private key for users, avoiding many security problems caused by the existing of attribute authority in the scheme. Furthermore, the new scheme uses blockchain to manage keys in the scheme, avoiding the single point failure problem of centralized key management. In particular, the new scheme stores the hash values of encrypted personal health records in blockchain, and the related index set is stored in smart contract, which can further improve the efficiency of data integrity verification. Finally, performance evaluation and security analysis indicate that our scheme is secure and feasible for practical use.

8 **Title:** An Efficient Authentication Scheme for Blockchain-Based Electronic Health Records

Author: Fei Tang ; Shuai Ma ; Yong Xiang ; Changlu Lin
IEEE Access, Year: 2019

In traditional electronic health records (EHRs), medical-related information is generally

separately controlled by different hospitals and thus it leads to the inconvenience of

information sharing. Cloud-based EHRs solve the problem of information sharing in the traditional EHRs. However, cloud-based EHRs suffer the centralized problem, i.e., cloud service center and key-generation center. This paper works on creating a new EHRs paradigm which can help in dealing with the centralized problem of cloud-based EHRs. Our solution is to make use of the emerging technology of blockchain to EHRs (denoted as blockchain-based EHRs for convenience). First, we formally define the system model of blockchain-based EHRs in the setting of consortium blockchain. In addition, the authentication issue is very important for EHRs. However, existing authentication schemes for blockchain-based EHRs have their own weak points. Therefore, in this paper, we also propose an authentication scheme for blockchain-based EHRs. Our proposal is an identity-based signature scheme with multiple authorities which can resist collusion attack out of N from $N-1$ authorities. Furthermore, our scheme is provably secure in the random oracle model and has more efficient signing and verification algorithms than existing authentication schemes of blockchain-based EHRs.

9 **Title:** Using Healthcare Authority and Patient Blockchains to Develop a Tamper-Proof Record Tracking System

Author: Uttkarsh Goel ; Ron Ruhl ; Pavol Zavorsky

2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity),
IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and
IEEE Intl Conference on Intelligent Data and Security (IDS)

Blockchain technology combines the openness of the Internet with the security of cryptography to provide a way to verify key information and to establish a trust in applications such as peer-to-peer payment services. Blockchain can be used to track the

healthcare transactions and can ensure authorized access to patient records and

compliance with the health information protection laws. In this paper, a dual blockchain model for the healthcare sector is proposed. The model combines healthcare authority blockchains and private patient blockchains for a tamperproof permission tracking system which ensures increased security and privacy while improving record and permission redundancy. The proposed model and its possible implementation are discussed in the context of Canadian healthcare privacy legislation in Alberta.

10 **Title:** Blockchain AI Framework for Healthcare Records Management:
Constrained Goal Model

Author: Youssef Wehbe ; Mohamed Al Zaabi ; Davor Svetinovic
2018 26th Telecommunications Forum (TELFOR)

Healthcare-related technology has been growing exponentially, from electronic healthcare records (EHRs) and personal health trackers to population health management tools. Currently, data from these sources is largely archived, analyzed separately or using only cursory integration with other data sources. In this study, we propose a coupled AI-Blockchain EHR management system. The goal is to provide a platform that leverages blockchain and artificial intelligence (AI) for (i) secure EHR management, (ii) efficient data integration, and (iii) reliable computer-aided diagnoses. A goal-oriented modeling approach with the Constrained Goal Model (CGM) is used to elicit the system requirements. Questionnaire results for a case study in Abu Dhabi, UAE served for model validation and refinement for maximizing the number of system users.

CHAPTER-3

SYSTEM

ANALYSIS

EXISTING SYSTEM

In existing system a framework of parallel healthcare systems (PHSs) based on the artificial systems computational experiments and parallel execution (ACP) approach is proposed in this paper. PHS uses artificial healthcare systems to model and represent patients' conditions, diagnosis, and treatment process, then applies computational experiments to analyze and evaluate various therapeutic regimens, and implements parallel execution for healthcare processes.

DISADVANTAGE

It uses too simple algebraic structure.

Every block is always encrypted in the same way.

Hard to implement with software.

AES in counter mode is complex to implement in software taking both performance and security into considerations.

PROPOSED SYSTEM

SHA-256 is a one-way function that converts a text of any length into a string of 256

bits. This is known as a hashing function. In this case, it is a cryptographically secure hashing function, in that knowing the output tells you very little about the input. It is a

modified version of SHA1, which in turn is a modified SHA0. All three are now broken, to some extent.

Note 1: All variables are 32 bit unsigned integers and addition is calculated modulo 232

Note 2: For each round, there is one round constant $k[i]$ and one entry in the message schedule array $w[i]$, $0 \leq i \leq 63$

Note 3: The compression function uses 8 working variables, a through h

Note 4: Big-endian convention is used when expressing the constants in this pseudocode, and when parsing message block data from bytes to words, for example, the first word of the input message "abc" after padding is 0x61626380

Advantage

1. The chief advantage that SHA-256 has more efficient.
2. It is fast to compute, resistant to pre-image and second-preimage attacks, and is collision resistant.

FEASIBILITY STUDY

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

- Economic feasibility
- Technical feasibility

3. 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 being placed on the client. A feasibility study evaluates the project's potential for success.

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 familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER- 4

SOFTWARE REQUIREMENT SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

- ✓ System : Pentium IV 2.4 GHz
- ✓ Hard Disk : 1TB
- ✓ Floppy Drive : 1.44 Mb
- ✓ Monitor : 15 VGA Colour
- ✓ Mouse : Logitech
- ✓ Ram : 8Gb

4.2 SOFTWARE REQUIREMENTS

- ✓ Operating system : Windows 10
- ✓ IDE : Anaconda navigator
- ✓ Coding Language : python
- ✓ Backend : SQLite

LANGUAGE DESCRIPTION

About Python Language

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

It provides rich data types and easier to read syntax than any other programming languages

It is a platform independent scripted language with full access to operating system API's

Compared to other programming languages, it allows more run-time flexibility

It includes the basic text manipulation facilities of Perl and Awk

A module in Python may have one or more classes and free functions

Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows

For building large applications, Python can be compiled to byte-code

Python supports functional and structured programming as well as OOP

It supports interactive mode that allows interacting Testing and debugging of snippets of code

In Python, since there is no compilation step, editing, debugging and testing is fast.

Applications of Python Programming

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

Python GUI Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is

an

easy

task.

To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

pack() method:It organizes the widgets in blocks before placing in the parent widget.

grid() method:It organizes the widgets in grid (table-like structure) before placing in the parent widget.

place() method:It organizes the widgets by placing them on specific positions directed by the programmer.

About Opencv Package

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-

Python works, it is a Python wrapper around original C++ implementation.

And the support of Numpy makes the task more easier. Numpy is a highly optimized

library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

FEASIBILITY STUDY

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

Economic 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 being placed on the client. A feasibility study evaluates the project's potential for success.

SOCIAL FEASIBILITY

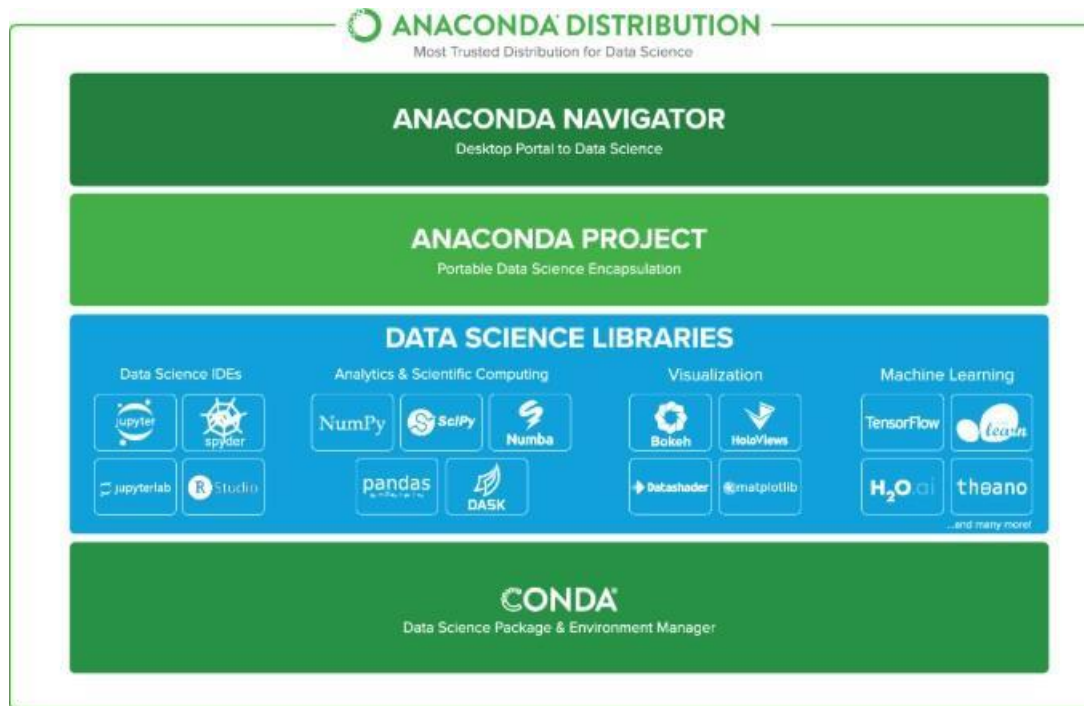
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends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

FEATURES OF ANACONDA NAVIGATOR

Anaconda

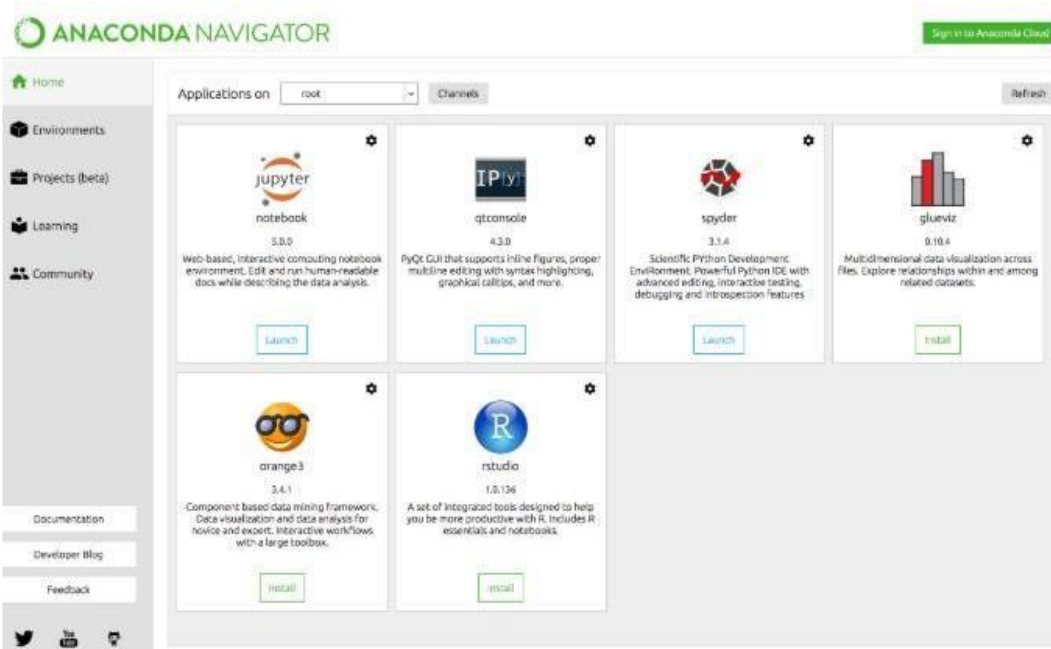
Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning.

The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source packages long with free community support.



What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.



Anaconda Navigator

Applications Provided In Anaconda Distribution

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

JupyterLab

Jupyter Notebook

Qt Console

Spyder

Glueviz

Orange

3

RStudio

Visual Studio Code

> JupyterLab: This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and

Architecture.

>Jupyter Notebook: This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data

analysis.

- > Qt Console: It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.
- > Spyder: Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
- > VS Code: It is a streamlined code editor with support for development operations like debugging, task running and version control.
- > Glueviz: This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.
- > Orange 3: It is a component-based data mining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large toolbox.
- > Rstudio: It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks.

5.3

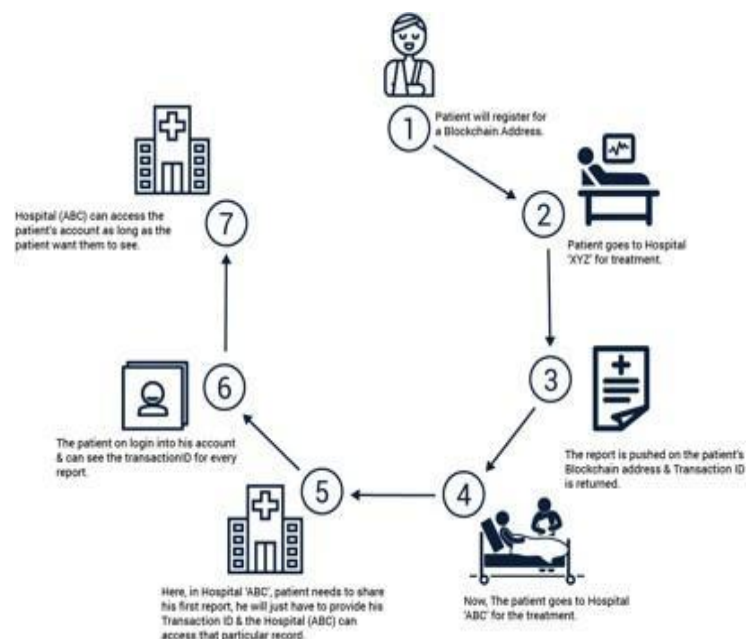


Compiled with Latest Python release: Anaconda 5.3 is compiled with Python 3.7,

taking advantage of Python's speed and feature improvements.

- **Better Reliability:** The reliability of Anaconda has been improved in the latest release by capturing and storing the package metadata for installed packages.
- **Enhanced CPU Performance:** The Intel Math Kernel Library 2019 for Deep Neural Networks(MKL 2019) has been introduced in Anaconda 5.3 distribution. Users deploying Tensorflow can make use of MKL 2019 for Deep Neural Networks. These Python binary packages are provided to achieve high CPU performance.
- **New packages are added:** There are over 230 packages which has been updated and added in the new release.
- **Work in Progress:** There is a casting bug in Numpy with Python 3.7 but the team is currently working on patching it until Numpy is updated.

5. SYSTEM DESIGN



5.1 UML DIAGRAM OF THE PROJECT:

UML is a standard language for specifying, visualizing, and documenting of software systems and created by Object Management Group (OMG) in 1997. There are three important type of UML modeling are Structural model, Behavioral model, and Architecture model. To model a system the most important aspect is to capture the dynamic behavior which has some internal or external factors for making the interaction. These internal or external agents are known as actors. It consists of actors, use cases and their relationships. In this fig we represent the Use Case diagram for our project.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

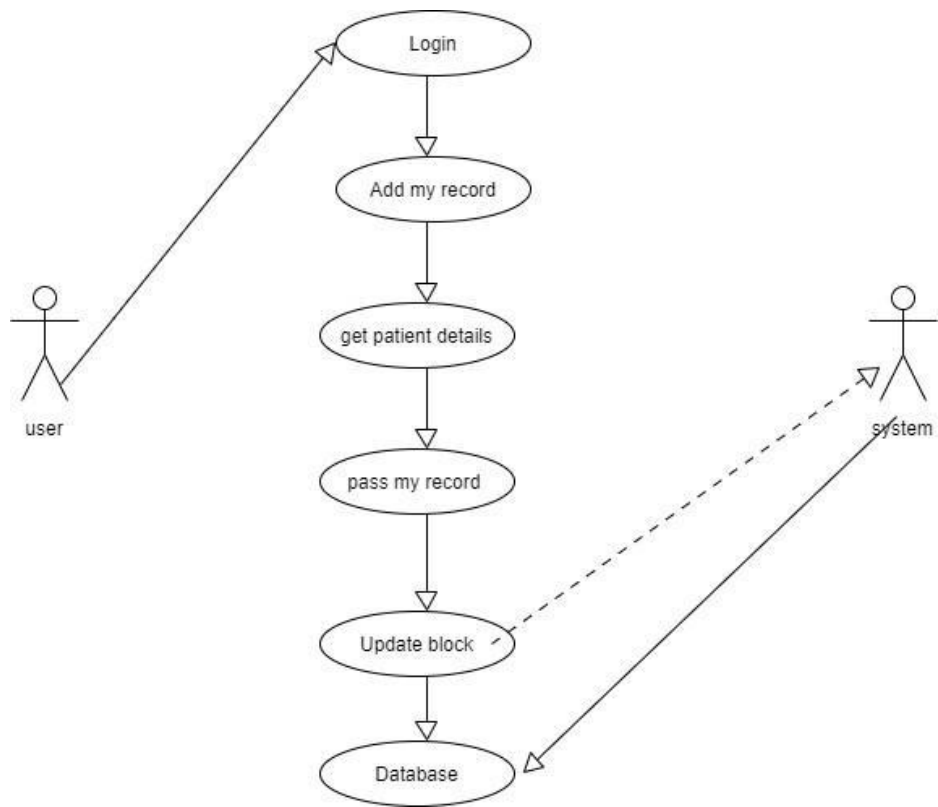
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

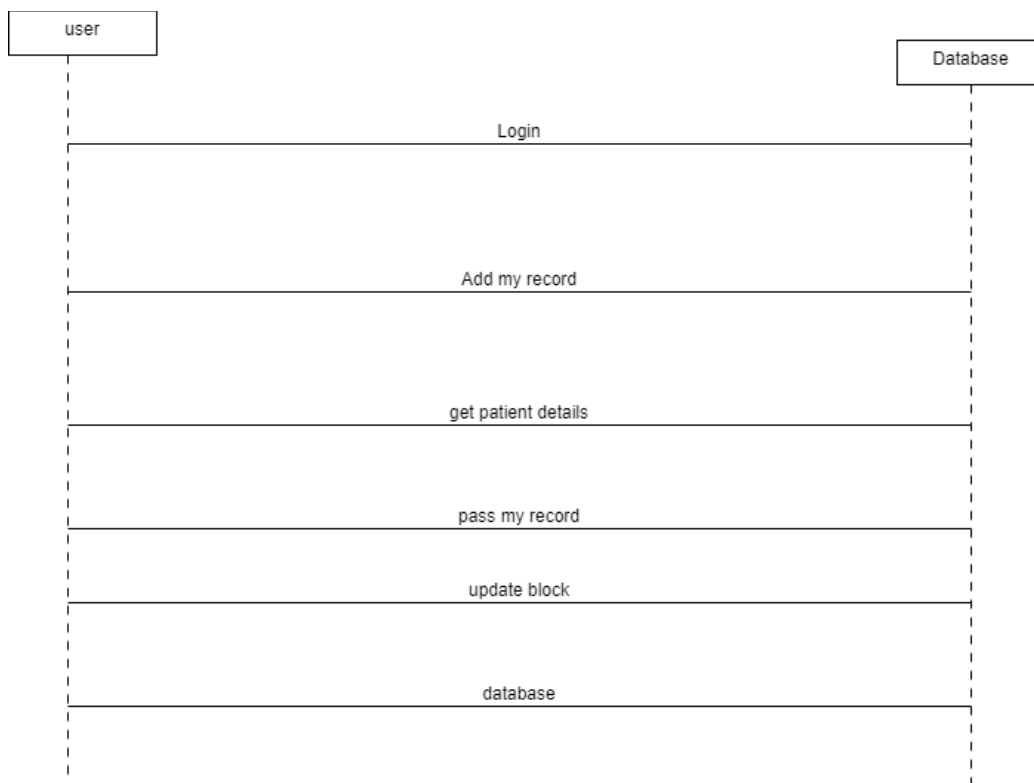
5.2 Use case diagram:

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components a user or another system that will interact with the system modeled. A use case is an external view of the system that represents some action the user might perform in order to complete a task.



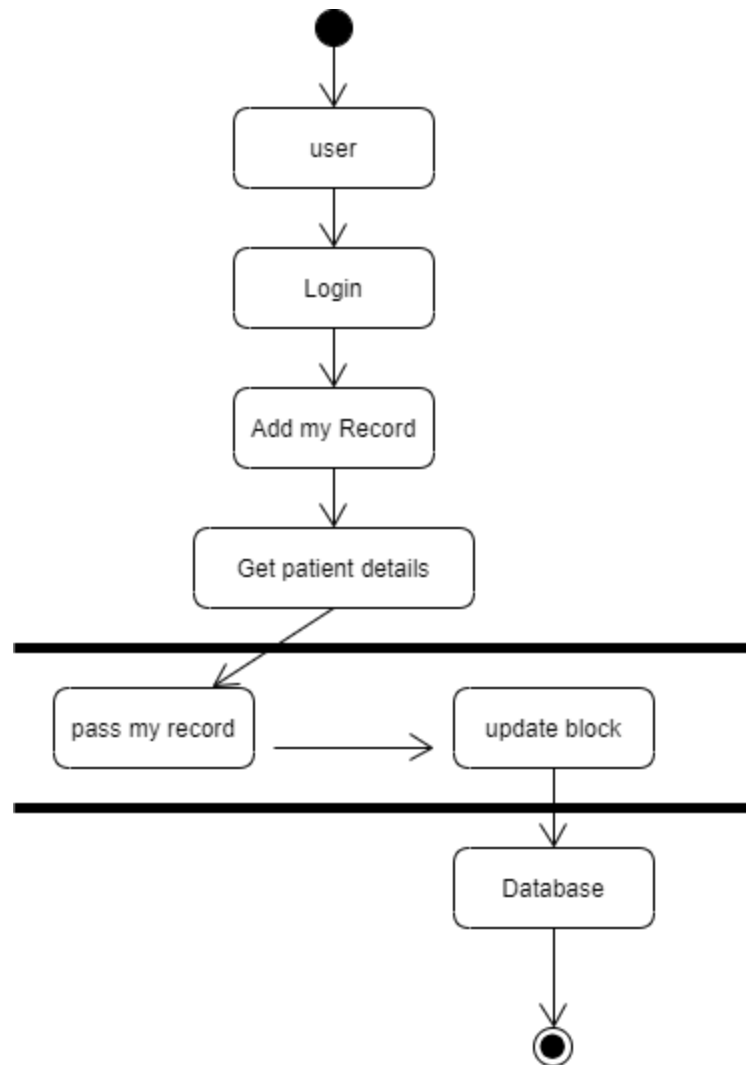
5.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



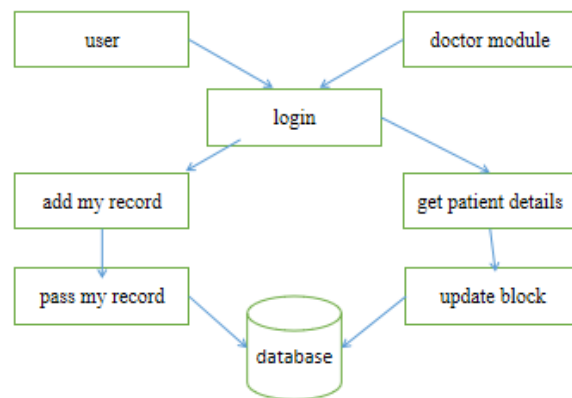
5.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



5.5 Flow Chart

Flow Chart



MODULES

register

login

add my record

pass my record

doctor module

get patient details

update block

SYSTEM TESTING

5.6 TESTING OBJECTIVES

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.7 TYPES OF TESTS

5.7.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.7.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more Concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.7.3 Functional test

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be

rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is Focused on requirements,

key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive

processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.7.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase

of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

5.8 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER-6

OUTPUT SCREENSHOTS

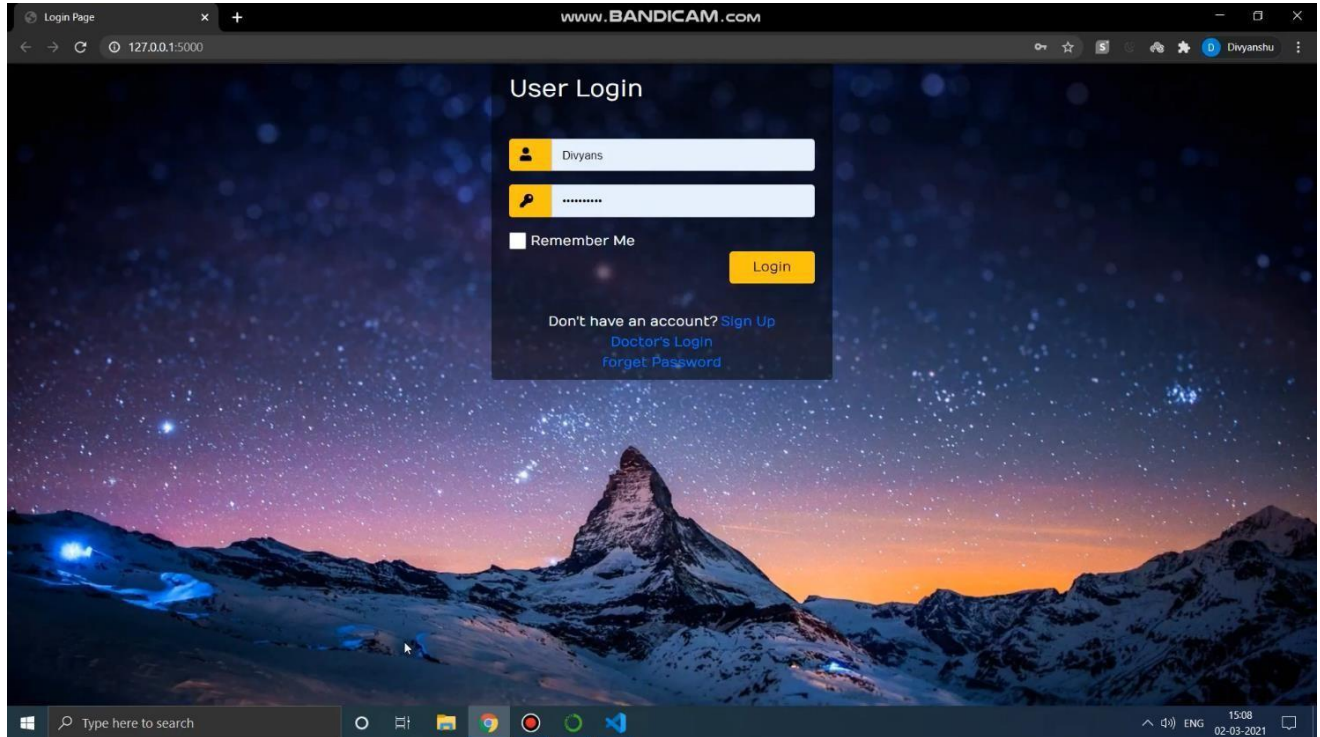
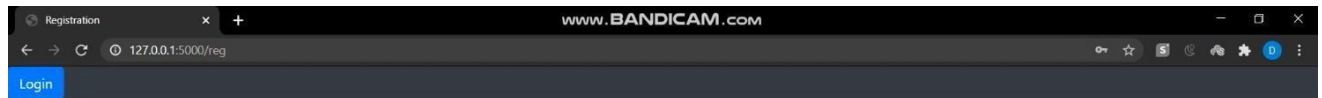


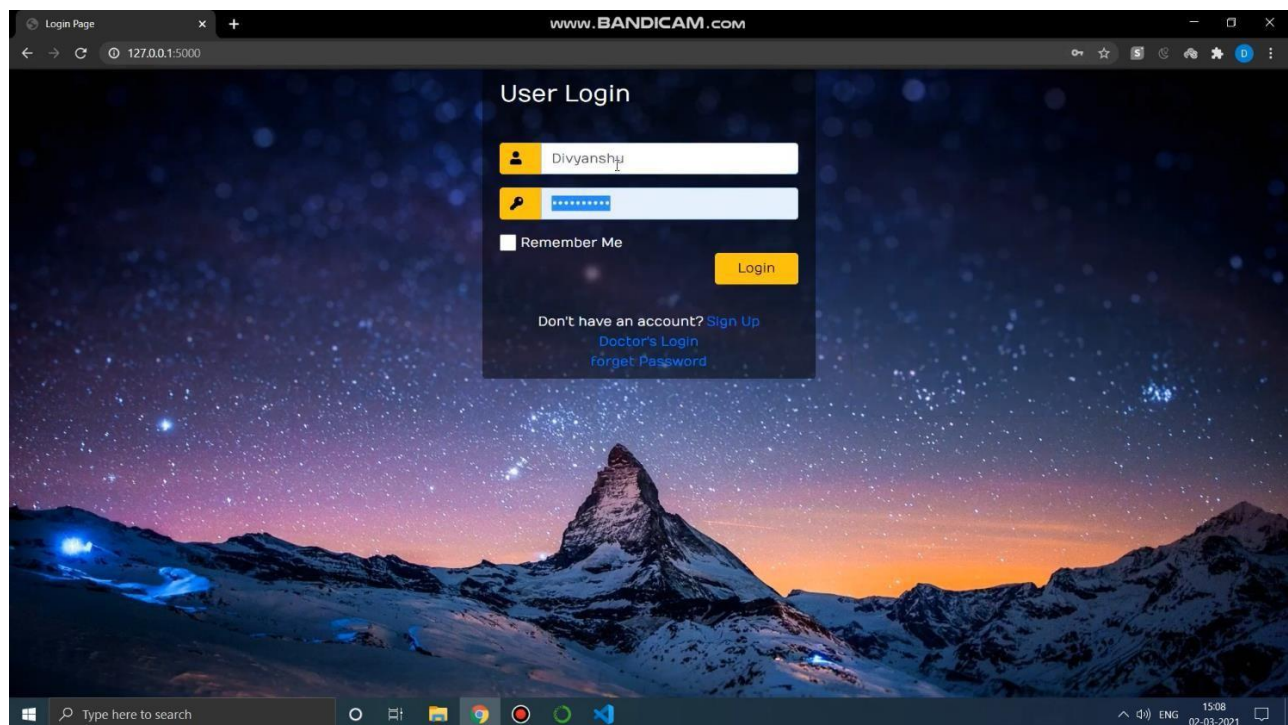
Fig 8.1-Home Page



Registration

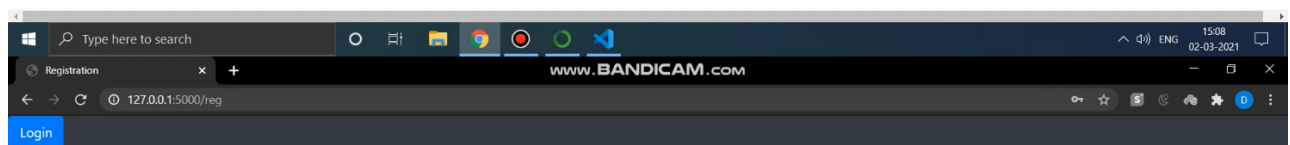
Username

Manage passwords...





Registration



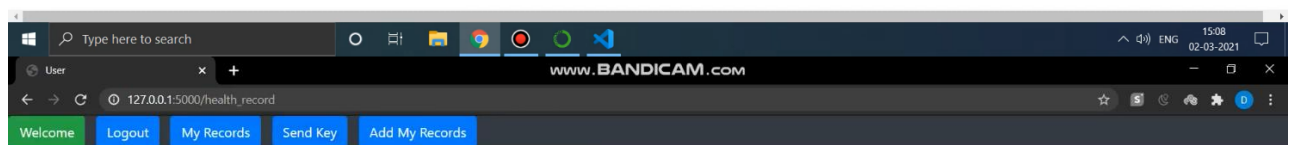
Registration





Registration

Divyanshu I
.....
Confirm password
Enter email Address
Submit



None	3
Time stamp	2021-02-02
Name	Divyanshu
Reason	Fever
Detail	stomach paining,cold
Previous hash	8d185dd1c429b8ff52f12462221bea4dfaa9f963b65887add9e5cacffd694a33
Hash	012df20d71bc6c97fd8babe78fa65d708ba3c101dfc5067e2523609b1e91cb7d



DB Browser for SQLite - C:\Users\Divya Tripathi\Desktop\patient_block_updated\block_chain.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: block_chain

	nonce	tstamp	name	reason	detail	phash	hash
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
7	3	2021-02-02	Himanshu	Fever	Weakness	7d71f6ea7af1...	265a969308e...
8	4	2021-02-03	Sudhanshu	Body Pain	weakness	265a969308e...	79615e7b087...
9	5	2021-02-03	Anwar	Chickenpox	Varicella- zos...	79615e7b087...	d7f0713abd5...
10	6	2021-02-03	Raghav	Common cold	nausea	d7f0713abd5...	9180e997eab...
11	10	2021-02-01	Himanshu	Common cold	Droplet Infect...	9180e997eab...	e5d801b9d56...
12	11	2021-02-02	Sudhanshu	Chickenpox	Air-borne dro...	e5d801b9d56...	1e266e0b1a5...
13	15	2021-02-03	Patient 1	Eye Swelling	pain	1e266e0b1a5...	8d09ceff8aa5...
14	13	2021-03-01	Divyanshu Tri...	Juindice	stomach paini...	8d09ceff8aa5...	4be8c0362e6...
15	17	2021-03-01	anmol	Chickenpox	headache	4be8c0362e6...	b991820ea68...
16	5	2021-03-04		Chickenpox	stomach paini...	b991820ea68...	a83ae363f0b...
17	99	2021-03-02	tha	mild fever	climate chang...	a83ae363f0b...	6da64b77357...
18	1	2021-03-02	Akash	Fever	weakness	6da64b77357...	b109f8a0220...
19	3	2021-03-02	Sonu	Common cold	cold	b109f8a0220...	4ab6c1e3adb...
20	3	2021-03-03	Divyans	Weakness	Body Pain	4ab6c1e3adb...	59b34ef73db...
21	13	2021-03-02	Lohit	eyepain	climate change	59b34ef73db...	e365b13742e...
22	12	2021-03-02	Arvind	Headache	air pollution	e365b13742e...	060626e6c94...
23	14	2021-03-02	Divyansh	Weakness	Blood Issue	060626e6c94...	0008be7e4a2...
24	21	2021-02-25	noor	Eye Swelling	dusting	0008be7e4a2...	88b981f206b...
25	22	2021-02-25	Sonu	Chickenpox	Shivering	88b981f206b...	314fd1b46e9...
26	23	2021-03-02	Sonuu	Fever	Weakness	314fd1b46e9...	9ee32d321cdf...
27	18	2021-03-02	febin	Common cold	climate change	9ee32d321cdf...	ae90300d15c...
28	19	2021-03-02	Rohan	Eye Swelling	air borne dro...	ae90300d15c...	fee2182d813...
29	20	2021-03-02	Divyanshu	Stomach Pain	Apetite Probl...	fee2182d813...	71fc291f48ec...

7 - 29 of 29

Go to: 1

Edit Database Cell

Mode: Text

64 char(s)

Type of data currently in cell: Text / Numeric

Remote

SQL Log Plot DB Schema Remote

UTF-8

DB Browser for SQLite - C:\Users\Divya Tripathi\Desktop\patient_block_updated\block_chain.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: block_chain

	nonce	tstamp	name	reason	detail	phash	hash
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
7	3	2021-02-02	Himanshu	Fever	Weakness	7d71f6ea7af1...	265a969308e...
8	4	2021-02-03	Sudhanshu	Body Pain	weakness	265a969308e...	79615e7b087...
9	5	2021-02-03	Anwar	Chickenpox	Varicella- zos...	79615e7b087...	d7f0713abd5...
10	6	2021-02-03	Raghav	Common cold	nausea	d7f0713abd5...	9180e997eab...
11	10	2021-02-01	Himanshu	Common cold	Droplet Infect...	9180e997eab...	e5d801b9d56...
12	11	2021-02-02	Sudhanshu	Chickenpox	Air-borne dro...	e5d801b9d56...	1e266e0b1a5...
13	15	2021-02-03	Patient 1	Eye Swelling	pain	1e266e0b1a5...	8d09ceff8aa5...
14	13	2021-03-01	Divyanshu Tri...	Juindice	stomach paini...	8d09ceff8aa5...	4be8c0362e6...
15	17	2021-03-01	anmol	Chickenpox	headache	4be8c0362e6...	b991820ea68...
16	5	2021-03-04		Chickenpox	stomach paini...	b991820ea68...	a83ae363f0b...
17	99	2021-03-02	tha	mild fever	climate chang...	a83ae363f0b...	6da64b77357...
18	1	2021-03-02	Akash	Fever	weakness	6da64b77357...	b109f8a0220...
19	3	2021-03-02	Sonu	Common cold	cold	b109f8a0220...	4ab6c1e3adb...
20	3	2021-03-03	Divyans	Weakness	Body Pain	4ab6c1e3adb...	59b34ef73db...
21	13	2021-03-02	Lohit	eyepain	climate change	59b34ef73db...	e365b13742e...
22	12	2021-03-02	Arvind	Headache	air pollution	e365b13742e...	060626e6c94...
23	14	2021-03-02	Divyansh	Weakness	Blood Issue	060626e6c94...	0008be7e4a2...
24	21	2021-02-25	noor	Eye Swelling	dusting	0008be7e4a2...	88b981f206b...
25	22	2021-02-25	Sonu	Chickenpox	Shivering	88b981f206b...	314fd1b46e9...
26	23	2021-03-02	Sonuu	Fever	Weakness	314fd1b46e9...	9ee32d321cdf...
27	18	2021-03-02	febin	Common cold	climate change	9ee32d321cdf...	ae90300d15c...
28	19	2021-03-02	Rohan	Eye Swelling	air borne dro...	ae90300d15c...	fee2182d813...
29	20	2021-03-02	Divyanshu	Stomach Pain	Apetite Probl...	fee2182d813...	71fc291f48ec...

7 - 29 of 29

Go to: 1

Edit Database Cell

Mode: Text

15 char(s)

Type of data currently in cell: Text / Numeric

Remote

SQL Log Plot DB Schema Remote

UTF-8

DB Browser for SQLite - C:\Users\Divya Tripathi\Desktop\patient_block_updated\block_chain.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: block_chain

	nonce	tstamp	name	reason	detail	phash	hash
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
4	3	2021-02-02	Divyanshu	Stomach Pain	Apetite Probl...	8d185dd1c42...	012df20d71bc...
5	3	2021-02-01	Akash	Dengue	Shivering	012df20d71bc...	1de65c81626...
6	3					1de65c81626...	7d71feea7af1...
7	3	2021-02-02	Himanshu	Fever	Weakness	7d71feea7af1...	265a969308e...
8	4	2021-02-03	Sudhanshu	Body Pain	weakness	265a969308e...	79615e7b087...
9	5	2021-02-03	Anwar	Chickenpox	Varicella- zos...	79615e7b087...	d70713abd5...
10	6	2021-02-03	Raghav	Common cold	nausea	d70713abd5...	9180e997eab...
11	10	2021-02-01	Himanshu	Common cold	Droplet Infect...	9180e997eab...	e5d801b9d56...
12	11	2021-02-02	Sudhanshu	Chickenpox	Air-borne dro...	e5d801b9d56...	1e266e0b1a5...
13	15	2021-02-03	Patient 1	Eye Swelling	pain	1e266e0b1a5...	8d09ceff8aa5...
14	13	2021-03-01	Divyanshu Tri...	Juindice	stomach paini...	8d09ceff8aa5...	4be8c036f2e6...
15	17	2021-03-01	anmol	Chickenpox	headache	4be8c036f2e6...	b991820ea68...
16	5	2021-03-04		Chickenpox	stomach paini...	b991820ea68...	a83ae363f0b...
17	99	2021-03-02	tha	mild fever	climate chang...	a83ae363f0b...	6da64b77357...
18	1	2021-03-02	Akash	Fever	weakness	6da64b77357...	b109f8a0220...
19	3	2021-03-02	Sonu	Common cold	cold	b109f8a0220...	4ab6c1e3adb...
20	3	2021-03-03	Divyans	Weakness	Body Pain	4ab6c1e3adb...	59b34ef73db...
21	13	2021-03-02	Lohit	eyepain	climate change	59b34ef73db...	e365b13742e...
22	12	2021-03-02	Arvind	Headache	air pollution	e365b13742e...	060626eec94...
23	14	2021-03-02	Divyansh	Weakness	Blood Issue	060626eec94...	0008be7e4a2...
24	21	2021-02-25	noor	Eye Swelling	dusting	0008be7e4a2...	88b981f206b...
25	22	2021-02-25	Sonu	Chickenpox	Shivering	88b981f206b...	314fd1b46e9...
26	23	2021-03-02	Sonuu	Fever	Weakness	314fd1b46e9...	9ee32d321cdf...

4 - 26 of 29

Go to: 1

Edit Database Cell

Mode: Text

Import Export Set as NULL

Type of data currently in cell: Text / Numeric
1 char(s)

Apply

Remote

Name	Commit	Last modified	Size
------	--------	---------------	------

SQL Log Plot DB Schema Remote

UTF-8

15:12 02-03-2021

Doctor

Inbox (36) - divyanshu.tripathi@... www.BANDICAM.com

127.0.0.1:5000/updates

Welcome Update Patient Logout

name

reason

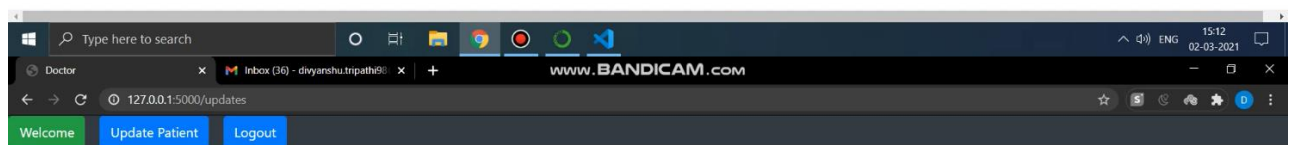
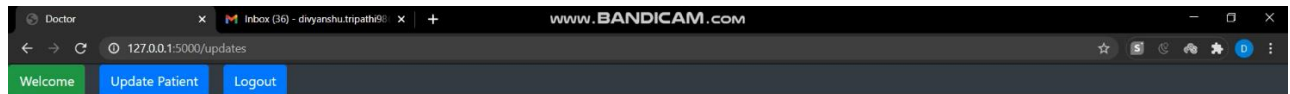
detail

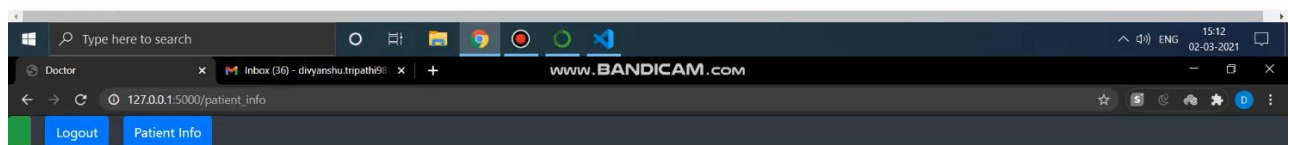
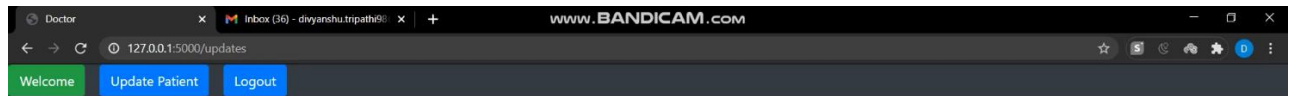
Submit

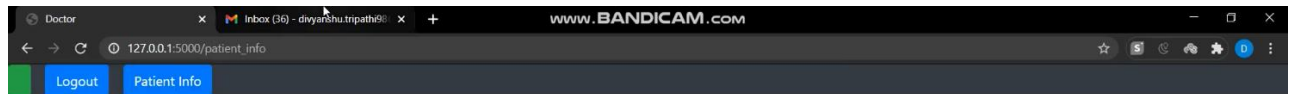
Patient info successfully updated....

Type here to search

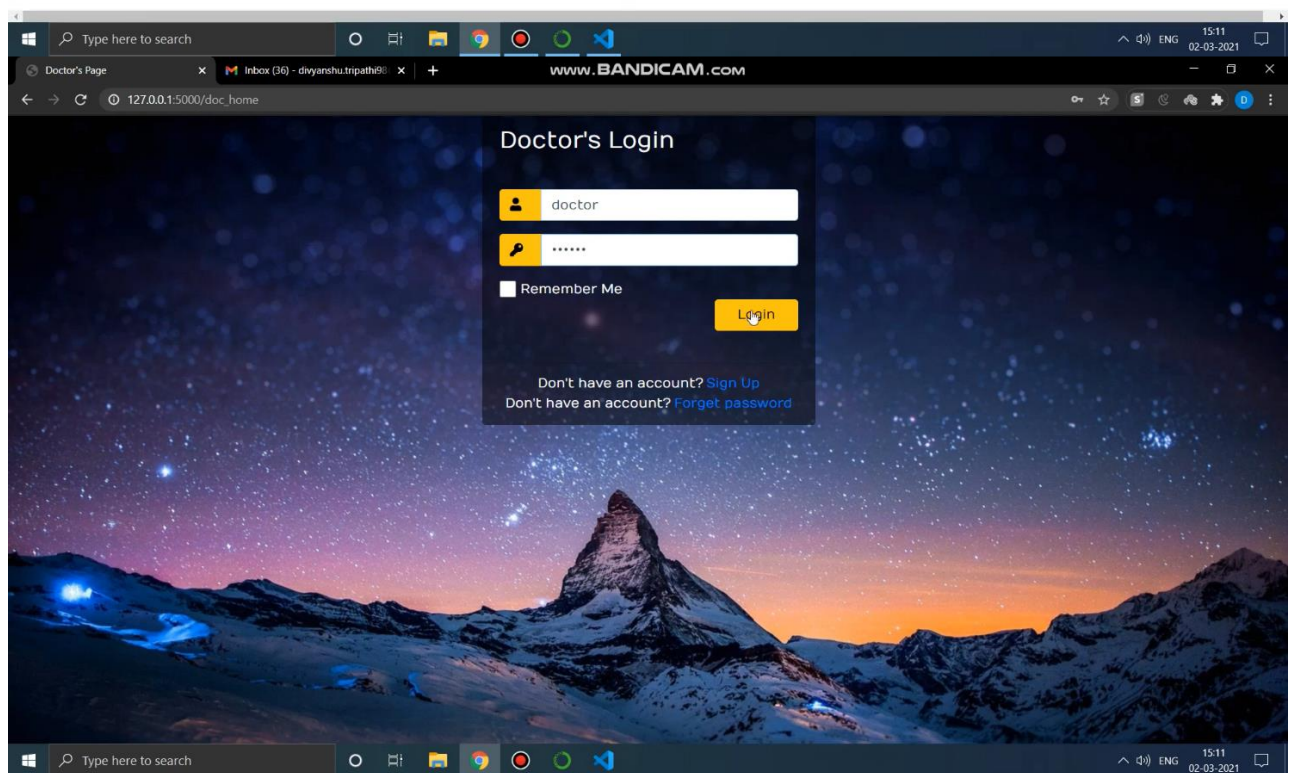
15:12 02-03-2021

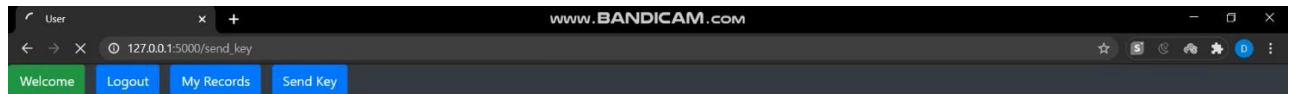




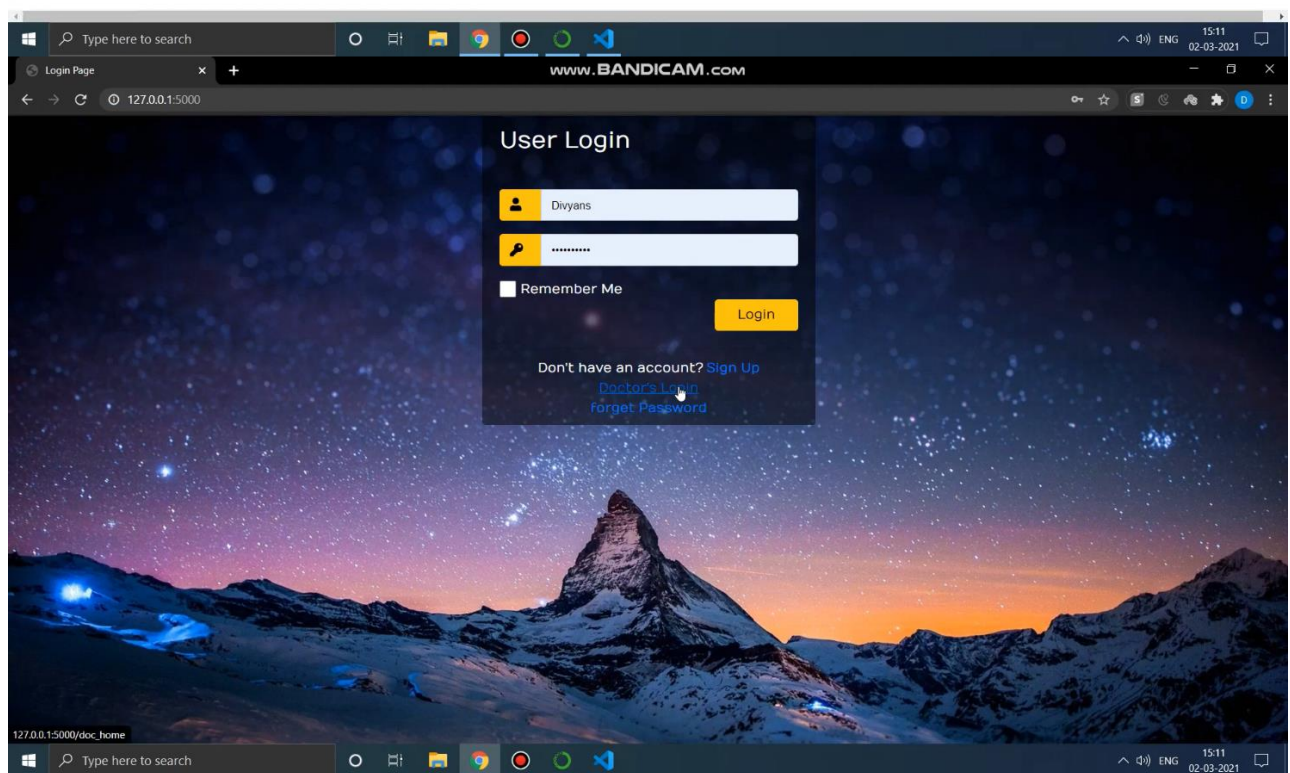


Enter OTP





divyanshu.tripathi98@gmail.com



DB Browser for SQLite - C:\Users\Divya Tripathi\Desktop\patient_block_updated\block_chain.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: block_chain

	nonce	tstamp	name	reason	detail	phash	hash
7	3	2021-02-02	Himanshu	Fever	Weakness	7d71f6ea7af1...	265a969308e...
8	4	2021-02-03	Sudhanshu	Body Pain	weakness	265a969308e...	79615e7b087...
9	5	2021-02-03	Anwar	Chickenpox	Varicella- zos...	79615e7b087...	d7f0713abd5...
10	6	2021-02-03	Raghav	Common cold	nausea	d7f0713abd5...	9180e997eab...
11	10	2021-02-01	Himanshu	Common cold	Droplet Infect...	9180e997eab...	e5d801b9d56...
12	11	2021-02-02	Sudhanshu	Chickenpox	Air-borne dro...	e5d801b9d56...	1e266e0b1a5...
13	15	2021-02-03	Patient 1	Eye Swelling	pain	1e266e0b1a5...	8d09ceff8aa5...
14	13	2021-03-01	Divyanshu Tri...	Juindice	stomach paini...	8d09ceff8aa5...	4be8c036f2e6...
15	17	2021-03-01	anmol	Chickenpox	headache	4be8c036f2e6...	b991820ea68...
16	5	2021-03-04		Chickenpox	stomach paini...	b991820ea68...	a83ae363f0b...
17	99	2021-03-02	tha	mild fever	climate chang...	a83ae363f0b...	6da64b77357...
18	1	2021-03-02	Akash	Fever	weakness	6da64b77357...	b109f8a0220...
19	3	2021-03-02	Sonu	Common cold	cold	b109f8a0220...	4ab6c1e3adb...
20	3	2021-03-03	Divyans	Weakness	Body Pain	4ab6c1e3adb...	59b34ef73db...
21	13	2021-03-02	Lohit	eyepain	climate change	59b34ef73db...	e365b13742e...
22	12	2021-03-02	Arvind	Headache	air pollution	e365b13742e...	060626e6c94...
23	14	2021-03-02	Divyansh	Weakness	Blood Issue	060626e6c94...	0008be7e4a2...
24	21	2021-02-25	noor	Eye Swelling	dusting	0008be7e4a2...	88b981f206b...
25	22	2021-02-25	Sonu	Chickenpox	Shivering	88b981f206b...	314fd1b46e9...
26	23	2021-03-02	Sonu	Fever	Weakness	314fd1b46e9...	9ee32d321cdf...
27	18	2021-03-02	febin	Common cold	climate change	9ee32d321cdf...	ae90300d15c...
28	19	2021-03-02	Rohan	Eye Swelling	air borne dro...	ae90300d15c...	fee2182d813...
29	20	2021-03-02	Divyanshu	Stomach Pain	Weakness	fee2182d813...	71fc291f48ec...

7 - 29 of 29

Go to: 1

Edit Database Cell

Mode: Text

Divyanshu

Type of data currently in cell: Text / Numeric
9 char(s)

Apply

Remote

Name	Commit	Last modified	Size
------	--------	---------------	------

SQL Log Plot DB Schema Remote

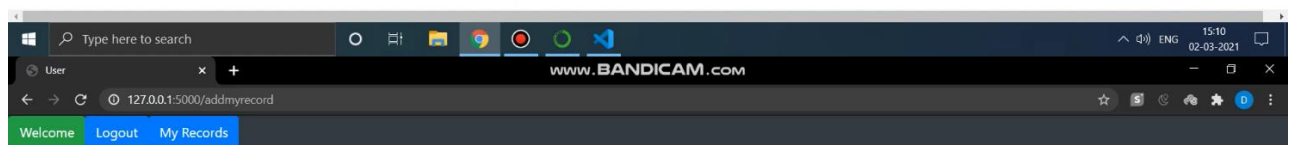
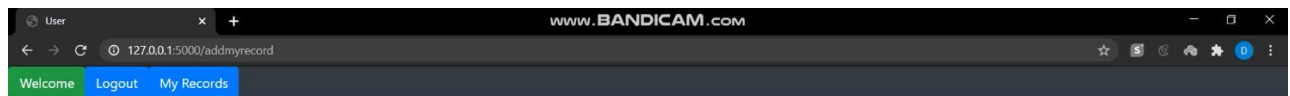
UTF-8

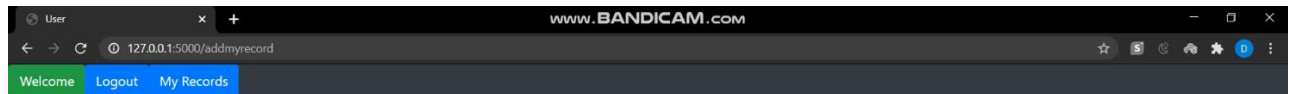
User

127.0.0.1:5000/health_record

Welcome Logout My Records Send Key Add My Records

None	3
Time stamp	2021-02-02
Name	Divyanshu
Reason	Fever
Detail	stomach paining,cold
Previous hash	8d185dd1c429b8ff52f12462221bea4d9a9f963b65887add9e5cacff694a33
Hash	012df20d71bc6c97fd8babe78fa65d708ba3c101dfc5067e2523609b1e91cb7d
None	20
Time stamp	2021-03-02
Name	Divyanshu
Reason	Stomach Pain
Detail	Weakness
Previous	fee2182d8132b1cdf835b6eeaa43f3ef852632d23a4f66b96aeb3f8a83ff0b727

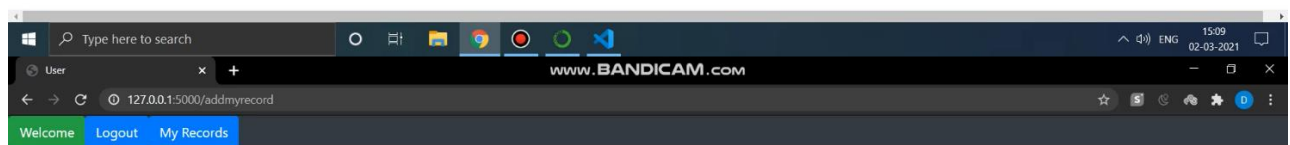




Add Records

Nonce	18
Time stamp	02-03-2021
name	febin
Reason	Common cold
detail	climate change

Submit

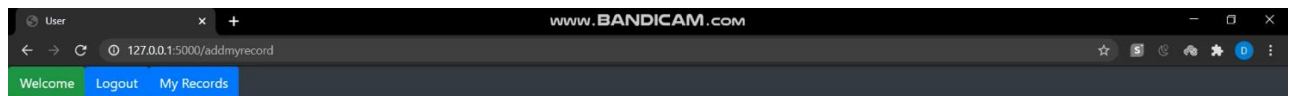


Add Records

Nonce	18
Time stamp	02-03-2021
name	febin
Reason	Common cold
detail	Common cold

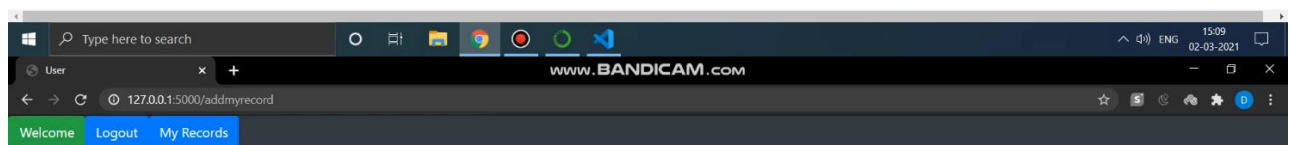
Submit





Add Records

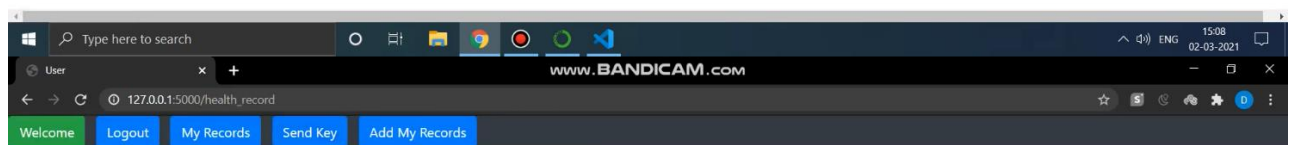
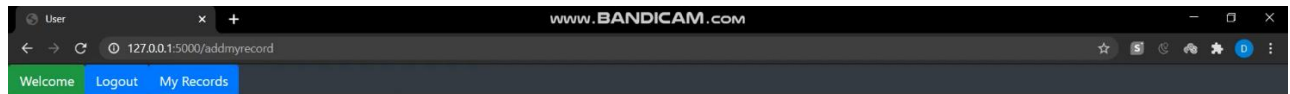
Nonce	18
Time stamp	02-03-2021
name	
Reason	
detail	
Submit	



Add Records

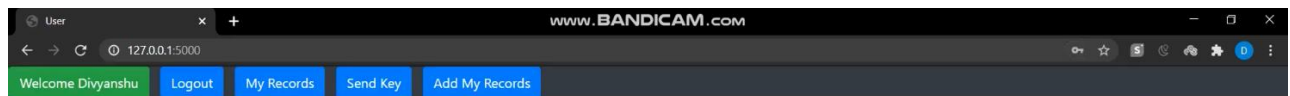
Nonce	18
Time stamp	dd-mm-yyyy
name	
Reason	
detail	
Submit	





None	3
Time stamp	2021-02-02
Name	Divyanshu
Reason	Fever
Detail	stomach pain, cold
Previous hash	8d185dd1c429b8ff52f12462221bea4dffa9f963b65887add9e5cacffd694a33
Hash	012df20d71bc6c97fd8babe78fa65d708ba3c101dfc5067e2523609b1e91cb7d





DB Browser for SQLite - C:\Users\Divya Tripathi\Desktop\patient_block_updated\block_chain.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: block_chain

	nonce	tstamp	name	reason	detail	phash	hash
7	3	2021-02-02	Himanshu	Fever	Weakness	7d71feea7af1...	265a969308e...
8	4	2021-02-03	Sudhanshu	Body Pain	weakness	265a969308e...	79615e7b087...
9	5	2021-02-03	Anwar	Chickenpox	Varicella- zos...	79615e7b087...	d70713abd5...
10	6	2021-02-03	Raghav	Common cold	nausea	d70713abd5...	9180e997eab...
11	10	2021-02-01	Himanshu	Common cold	Droplet Infect...	9180e997eab...	e5d801b9d56...
12	11	2021-02-02	Sudhanshu	Chickenpox	Air-borne dro...	e5d801b9d56...	1e266e0b1a5...
13	15	2021-02-03	Patient 1	Eye Swelling	pain	1e266e0b1a5...	8d09ceff8aa5...
14	13	2021-03-01	Divyanshu Tri...	Juindice	stomach paini...	8d09ceff8aa5...	4be8c036f2e6...
15	17	2021-03-01	anmol	Chickenpox	headache	4be8c036f2e6...	b991820ea68...
16	5	2021-03-04		Chickenpox	stomach paini...	b991820ea68...	a83ae363f0b...
17	99	2021-03-02	tha	mild fever	climate chang...	a83ae363f0b...	6da64b77357...
18	1	2021-03-02	Akash	Fever	weakness	6da64b77357...	b109f8a0220...
19	3	2021-03-02	Sonu	Common cold	cold	b109f8a0220...	4ab6c1e3adb...
20	3	2021-03-03	Divyans	Weakness	Body Pain	4ab6c1e3adb...	59b34ef73db...
21	13	2021-03-02	Lohit	eyepain	climate change	59b34ef73db...	e365b13742e...
22	12	2021-03-02	Arvind	Headache	air pollution	e365b13742e...	060626eac94...
23	14	2021-03-02	Divyansh	Weakness	Blood Issue	060626eac94...	0008be7e4a2...
24	21	2021-02-25	noor	Eye Swelling	dusting	0008be7e4a2...	88b981f206b...
25	22	2021-02-25	Sonu	Chickenpox	Shivering	88b981f206b...	314fd1b46e9...
26	23	2021-03-02	Sonuu	Fever	Weakness	314fd1b46e9...	9ee32d321cdf...
27	18	2021-03-02	febin	Common cold	climate change	9ee32d321cdf...	ae90300d15c...
28	19	2021-03-02	Rohan	Eye Swelling	air borne dro...	ae90300d15c...	fee2182d813...
29	20	2021-03-02	Divyanshu	Stomach Pain	Apetite Probl...	fee2182d813...	71fc291f48ec...

7 - 29 of 29

Go to: 1

Edit Database Cell

Mode: Text Import Export Set as NULL

fee2182d8132b1cdf835beee43f3e852632d23a4f66b96aeb3f8a83ff0b727

Type of data currently in cell: Text / Numeric
64 char(s)

Remote

Name	Commit	Last modified	Size
------	--------	---------------	------

SQL Log Plot DB Schema Remote

UTF-8

CHAPTER – 7

CONCLUSION AND FUTUREWORK

Blockchain enables the organization of check, characterization and data sharing while in the meantime giving information related to assurance, helpful resource saving and empowering for the patient and making masses social protection more intelligent. Taking the favored outlook of this technique, it achieves perfect security putting something aside for the patient. Compared to traditional EHR systems, which use client-server architecture, the proposed system uses blockchain for improving efficiency and security. Blockchain technology addresses interoperability challenges and is based on open standards, provides a shared distributed view of the health data and will also achieve widespread acceptance and deployment throughout all industries. Utilization of the proposed health blockchain that is described in this paper has the potential to engage millions of individuals, health care providers, health care entities and medical researchers to share the vast amount of genetic, diet, lifestyle, environmental and health data with a guaranteed security and privacy protection. Future reserch opportunities lie in a deeper analysis of the business processes in the manufacturing industry to further exploit the advantages of the blockchain technology

REFERENCES

- [1] G. Jetley and H. Zhang, "Electronic health records in IS research: Quality issues, essential thresholds and remedial actions," *Decis. Support Syst.*, pp. 113–137, 2019.
- [2] K. Wisner, A. Lyndon, and C. A. Chesla, "The electronic health record's impact on nurses' cognitive work: An integrative review," *Int. J. Nurs. Stud.*, vol. 94, pp. 74–84, 2019.
- [3] M. Hochman, "Electronic Health Records: a ``Quadruple Win," a ``Quadruple Failure," or Simply Time for a Reboot?," *J. Gen. Intern. Med.*, vol. 33, no. 4, pp. 397–399, Apr. 2018.
- [4] Q. Gan, "Adoption of Electronic Health Record System: Multiple Theoretical Perspectives," 2014 47th Hawaii Int. Conf. Syst. Sci., pp. 2716–2724, 2014.
- [5] T. Vehko et al., "Experienced time pressure and stress: electronic health records usability and information technology competence play a role," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, p. 160, Aug. 2019.
- [6] M. Reisman, "EHRs: The Challenge of Making Electronic Data Usable and Interoperable.," *P T*, vol. 42, no. 9, pp. 572–575, 2017.
- [7] W. W. Koczkodaj, M. Mazurek, D. Strzałka, A. Wolny-Dominiak, and M. Woodbury-Smith, "Electronic Health Record Breaches as Social Indicators," *Soc. Indic. Res.*, vol. 141, no. 2, pp. 861–871, 2019.
- [8] S. T. Argaw, N. E. Bempong, B. Eshaya-Chauvin, and A. Flahault, "The state of research on cyberattacks against hospitals and available best practice recommendations: A scoping review," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, pp. 1–11, 2019.
- [9] A. McLeod and D. Dolezel, "Cyber-analytics: Modeling factors associated with healthcare data breaches," *Decis. Support Syst.*, vol. 108, pp. 57–68, 2018.
- [10] L. Coventry and D. Branley, "Cybersecurity in healthcare: A narrative review of trends, threats and ways forward," *Maturitas*, vol. 113, pp. 48–52, 2018.

APPENDIX

```
from flask import
Flask,url_for,request,render_template,session from
flask_mail import Mail, Message
import sqlite3
import
blockchain1
from blockchain1 import
BlockChain,Block from random import
randint
```

```
app=Flask(_____
name_____)
mail=Mail(app)
app.secret_key="R
udy"
```

```
app.config['MAIL_SERVER']='smtp.gmail.
com' app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] =
'cloudcomp2021@gmail.com'
app.config['MAIL_PASSWORD'] =
'cloudcloudlearning@99' app.config['MAIL_USE_TLS']
= False app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
```

```
def
    random_with_N_digits(n):
        range_start = 10**(n-1)
        range_end = (10**n)-1
        return randint(range_start, range_end)
```

```
@app.route('/')  
,methods=["GET","POST"]) def login():  
    username=None  
    password=None  
    err="Invalid username and  
password" if  
request.method=='POST':
```

```

session['username']=request.form['un
ame']
username=request.form['uname']
password=request.form['pwd']
conn = sqlite3.connect("reg1.db")
r=conn.cursor()
r.execute("select name,pwd from reg where name=? and
pwd=?", (username,password))
rows=r.fetcha
ll() if
len(rows)!=0:
for i in rows:
if i[0]==username and i[1]==password:
return
render_template('user.html',username=username)
else:
return render_template('login.html',err=err)
else:
return
render_template('login.html',err=err) return
render_template('login.html')

```

```

@app.route('/reg',
methods=["GET","POST"]) def reg():
uname=None
pwd=None
cpwd=None
ema=None
msg="Register
successfully" if
request.method=='POST
':
uname=request.form['txt']
pwd=request.form['passw']

```

```
cpwd=request.form['cpass  
w'] ema=request.form['em']  
import sqlite3  
table_name = 'reg'  
conn = sqlite3.connect("reg1.db")
```

```

        c = conn.cursor()
        c.execute('create table if not exists ' + table_name + ' (name
varchar(50),pwd varchar(50),cpwd varchar(50),email varchar(50))')
        c.execute('insert into '+table_name+' values
(?,?,?,?)',(uname,pwd,cpwd,ema))
        conn.commit()
        conn.close()
        return
    render_template('registraion.html',msg=msg)
    return render_template('registraion.html')

```

```

@app.route('/health_record',methods=["GET",
"POST"]) def health_record():
    if request.method=='GET':
        conn=sqlite3.connect("block_chain.d
b") c=conn.cursor()
        c.execute("select * from block_chain where
name=?",(session['username'],))
        rows=c.fetchall(
        ) for i in rows:
            print(i)
        return render_template('user.html',rows=rows)

```

```

@app.route('/proudate')

```

```

def proudate():
    conn = sqlite3.connect("reg1.db")
    r=conn.cursor()
    r.execute("select name,pwd from reg ")
    rows=r.fetchall()
    print(rows)
    return render_template('user.html',rows=rows)

```

```
@app.route('/addmyrecord',methods=["GET","  
POST"]) def addmyrecord():
```

```
rname=None
rreason=None
rdetail=None
nonce=None
tstamp=None
final_li=[]
```

```
if request.method=='POST':
    nonce=request.form['nonce
    ']
    tstamp=request.form['tstam
    p']
    rname=request.form['name'
    ]
    rreason=request.form['reas
    on']
    rdetail=request.form['detail
    s'] u=[]
    u.append(nonc
    e)
    u.append(tstam
    p)
    u.append(rnam
    e)
    u.append(rreas
    on)
    u.append(rdetai
    l)
    conn =
    sqlite3.connect("block_chain.db") g
    = conn.cursor()
    g.execute("select nonce,tstamp,name,reason,detail from
    block_chain") rows=g.fetchall()
# print(rows)
```

```
li=[]
for i in rows:
    li.append(i)
li.append(u)

table_name="block_chain"
conn =
sqlite3.connect("block_chain.db") c
= conn.cursor()
c.execute('create table if not exists ' + table_name + ' (nonce
```

```
varchar(50),tstamp varchar(50),name varchar(50),reason  
varchar(50),detail varchar(50),phash varchar(50),hash varchar(50)))'
```

```
for i in  
    range(len(li)): if  
        li[i] not in  
        final_li:  
            final_li.append(li[i])  
print(final_li)  
c.execute("delete from block_chain")  
osa=BlockChain()  
for i in range(len(final_li)):
```

```
osa.addBlock(Block(final_li[i][0],final_li[i][1],final_li[i][2],final_li[i][3],fi  
nal_li[i]  
][4]))
```

```
for b in osa.chain:  
    c.execute('insert into '+table_name+' values  
(?,?,?, ?, ?, ?, ?)',(b.nonce,b.tstamp,b.name,b.reason,b.detail,b.prevh  
h,b.hash)  
)
```

```
conn.commit()  
conn.close()  
return render_template("add.html")
```

```
@app.route('/doc_home',  
methods=["GET","POST"]) def doc_home():  
    dname=None  
    dpwd=None  
    if request.method=='POST':  
        dname=request.form['dname']  
        dpwd=request.form['dpwd']  
        conn=sqlite3.connect("doc_lo  
g.db") c = conn.cursor()
```

```
c.execute("select username,password from doc_reg1 where  
username=? and password=?", (dname, dpwd))  
rows=c.fetchall()  
if  
len(rows)!=0:
```

```

        for i in rows:
            if i[0]==dname and i[1]==dpwd:
                return
            render_template('doc_page.html') else:
                return render_template('doc.html')
        else:
            return render_template('doc.html')

    return render_template('doc.html')

@app.route('/send_key',methods=["GET","POST"]) def send_key():
    e_mail=None
    OTP=None
    suc="Your OTP send successfully "
    if request.method=='POST':
        e_mail=request.form['e']
        msg = Message('Patient OTP', sender =
'cloudcomp2021@gmail.com', recipients =[e_mail])
        OTP=random_with_N_digit
s(4) msg.body = str(OTP)
        mail.send(msg)
        conn=sqlite3.connect("keys
.db") k=conn.cursor()
        k.execute("insert into keys
values(?,?)",(session['username'],OTP)) conn.commit()
        conn.close()
        return render_template('key.html',suc=suc)

    return render_template('key.html')

@app.route('/patient_info',methods=["GET","POST"]) def patient_info():

```

```

ckotp=None
nameli=[]
if request.method=='POST':
    ckotp=request.form['otp']
    conn=sqlite3.connect("keys.db")
    ch=conn.cursor()
    ch.execute("select name from keys where otp=?", (ckotp,))
    rows=ch.fetchall()
    for i in rows:
        nameli.append(i)
        print(i)
    conn=sqlite3.connect("block_chain.d
b")
    chk=conn.cursor()
    chk.execute("select
nonce,tstamp,name,reason,detail,phash,hash from block_chain
where name=?", (nameli[0]))
    lii=chk.fetchall()
    return

```

```

render_template('info.html',lii=lii) return

```

```

render_template('doc_page.html')

```

```

@app.route('/updates',
methods=["GET", "POST"]) def updates():
    pname=None
    preason=None
    pdetail=None
    succ="Patient info successfully updated    "
    if request.method=='POST':
        pname=request.form['pname']
        preason=request.form['preason']
        pdetail=request.form['pdetail']
        conn=sqlite3.connect("block_chain.d

```

```
b") c=conn.cursor()  
c.execute("update block_chain set reason=? ,detail=? where  
name=?", (preason, pdetail, pname))
```

```
conn.commit()
conn.close()
return
render_template('update.html',succ=succ) return
render_template('update.html')
```

```
@app.route('/changepassword',methods=["GET","
POST"]) def changepassword():
    emailid=None
    li=[]
    if request.method=='POST':
        emailid=request.form['em']
        conn=sqlite3.connect("reg1
.db") c = conn.cursor()
        c.execute("select name,pwd from reg where email=?", (emailid,))
        rows=c.fetchall()
        for i in rows:
            li.append(i)
            msg = Message('Patient OTP', sender =
'cloudcomp2021@gmail.com', recipients =[emailid])
            msg.body = "username: "+str(li[0][0])+" "+"Password:
            "+str(li[0][1]) mail.send(msg)
        return render_template('pwd.html')
```

```
@app.route('/dchangepassword',methods=["GET",
"POST"]) def dchangepassword():
    demail=None
    dli=[]
    if request.method=='POST':
        demail=request.form['demail']
        conn=sqlite3.connect("doc_lo
g.db") c = conn.cursor()
        c.execute("select username,password from doc_reg1 where
```

email=?",(demail,))

```
rows=c.fetchall(
) for i in rows:
    dli.append(i)
print(len(dli))
msg = Message('Patient OTP', sender =
'cloudcomp2021@gmail.com', recipients =[demail])
msg.body = "Doctor Name: "+str(dli[0][0])+" "+"Password:
"+str(dli[0][1]) mail.send(msg)
return render_template('dpwd.html')
```

```
if __name__ == '__
main__':
    app.run(debug=True)
```