A Project Report on

ACCESS CONTROL FOR HEALTHCARE DATA

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

in partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

A.L. Grace Evangelene (19BD1A05A5)

Ayush Bansal (19BD1A05A6)

Akanksha Pallikonda (19BD1A05B8)

Yasa Sai Snigdha (19BD1A05BT)

Under the guidance of
Mr. Suresh Gaddeboina
Assistant professor
Department of CSE

Department of Computer Science and Engineering

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH

3-5-1206, Narayanaguda, Hyderabad – 500029

2022-2023

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Accredited by NBA & NAAC, Approved By A.I.C.T.E., Reg by Govt of Telangana State & Affiliated to JNTU, Hyderabad)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **ACCESS CONTROL FOR HEALTHCARE DATA** being submitted by

A.L. Grace Evangelene (19BD1A05A5)

Ayush Bansal (19BD1A05B8)

Akanksha Pallikonda (19BD1A05B8)

Yasa Sai Snigdha (19BD1A05BT)

In partial fulfilment for the award of **Bachelor of Technology** in **Computer Science and Engineering** affiliated to the **Jawaharlal Nehru Technological University, Hyderabad** during the year 2022-23.

Internal Guide
(Mr. Suresh Gaddeboina)

Head of the Department
(Mr. PARA UPENDAR)

Submitted for Viva Voce Examination held on _____

External Examiner

Vision of KMIT

Producing quality graduates trained in the latest technologies and related tools and striving to make India a world leader in software and hardware products and services. To achieve academic excellence by imparting indepth knowledge to the students, facilitating research activities and catering to the fast growing and ever- changing industrial demands and societal needs.

Mission of KMIT

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modalplatforms and prepare students to become successful professionals.
- To establish industry institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create newenterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand lndia's challenges and to encourage them to develop effective solutions
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision & Mission of CSE

Vision of the CSE

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the CSE

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcateentrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions thatbenefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellenceand in fulfilling societal responsibilities.

PROGRAM OUTCOMES (POs)

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. Design/development of solutions: Design solutions for complex engineering problem and design system component or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to societal, health, safety. legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and

write effective reports and design documentation make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftment's.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, SocialNetworks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility

PROJECT OUTCOMES

P1: Understand the concept of blockchain.

P2: Implementation of Ethereum.

P3: Performing a transaction through Ethereum.

P4: Examine the outcome through the interface.

LOW - 1
MEDIUM - 2
HIGH - 3

PROJECT OUTCOMES MAPPING PROGRAM OUTCOMES

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1	3	2		2	3			1				3
P2	2	3	2	2	3	2		3	3		1	2
P3	1	2	3	2	3	2			2	2		2
P4	2	3	3	2	3	2	2	3	3	2	1	2

PROJECT OUTCOMES MAPPING PROGRAM SPECIFIC OUTCOMES

PSO	PSO1	PSO2
P1	2	3
P2	1	3
P3	2	2
P4	3	3

PROJECT OUTCOMES MAPPING PROGRAM EDUCATIONAL OBJECTIVES

PEO	PEO1	PEO2	PEO3	PEO4
P1	1	2	2	3
P2		3	2	
P3	1	2	2	
P4			3	3

DECLARATION

We hereby declare that the project report entitled **"ACCESS CONTROL FOR HEALTHCARE DATA**" is done in the partial fulfillment for the award of the Degree in Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhereelse.

A.L. Grace Evangelene (19BD1A05A5)

Ayush Bansal (19BD1A05A6)

Akanksha Pallikonda (19BD1A05B8)

Yasa Sai Snigdha (19BD1A05BT)

ACKNOWLEDGMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, B.E., M Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin**, Director and **Dr. D. Jaya Prakash**, Dean Academics for providing an excellent environment in the college.

We are also thankful to **Mr. Para Upendar**, Head of the Department for providing uswith both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Mr. Gaddeboina Suresh,** for his/her valuableguidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project. We sincerely thank our friends and family for their constant motivation during the project work.

A.L. Grace Evangelene (19BD1A05A5)

Ayush Bansal (19BD1A05A6)

Akanksha Pallikonda (19BD1A05B8)

Yasa Sai Snigdha (19BD1A05BT)

CONTENTS

ABSTRACT

Medical data contains multiple records of patient data that are important for subsequent treatment and future research. Although the recording of medical information has evolved from the initial paper records to electronic medical records (EMR), which are more convenient for data access and storage, more attention needs to be paid to protecting the privacy of data. Blockchain is widely used in the management of health-care data because of its decentralized and tamper-proof features. In blockchain data is saved at multiple nodes and this technique is called as decentralized storage. If one node is corrupted then user can recover data from other node. And each party data will be encrypted and then hashed and then stored in blocks. If an attacker tries to modify any block then that block hash code verification will fail and due to this technique no attacker will be able to alter block data. We are using block chain Ethereum tool to create a Smart Contract.

LIST OF FIGURES

CHAPTER -1

1 INTRODUCTION

Rental The ongoing popularity of Blockchain technology is increasing day by day and the diversity associated with its application, helps in advancement of research in different scientific fields with social relevance. In Spite of being new and still in the experimental stage, the Blockchain is being considered as a pathbreaking solution, which addresses several modern concerns in the technological field like identity management, decentralization, data ownership, decisions driven by trust and data. Blockchain works on a distributed peer-to-peer system where every participating node receives a copy of the same data and acts upon the same set of rules. Blockchain is a distributed ledger in which new data is appended as a chain of blocks and it grows continuously. Protection of user security and maintenance of ledger consistency are taken care of by cryptographic functions and distributed consensus mechanisms.

Big data refers to massive volumes of structured and unstructured data so large that it is difficult to process using traditional database or software techniques. Big data has 7 ‘V’ properties , that are Volume, Variety, Velocity, Variability, Veracity, Visualization and Value. Volume is about how large is the data in size. Velocity is the speed of accessing data. Variety is about heterogeneous sources and unstructured nature of the data. Variability says that the same data can have different meanings. Veracity is about the quality of data. Visualization consists of graphics, charts and other plots that help to understand the meaning of data and retrieve more details. And finally Value is about how data processing can be done to produce a meaningful outcome. Blockchain Technology can be used in several big data use cases like management of private/personal data, intellectual property resolution, supply chain management, IOT communication, healthcare etc. .

Any modification on any of the blocks will generate different hash values in other blocks in a cascading manner and different link relations. In this way the blockchain achieves immutability and security. Blockchain also involves consensus mechanisms in order to ensure synchronization among blocks and agreement between existing nodes to add a new transaction in the chain. Healthcare data is highly sensitive in nature, which consists of private information related to the diagnosis and treatment of patients. In traditional approach these data are stored by each hospital separately where sharing data is very time consuming. Also, health data is very limited to access due to the risk of disclosure of sensitive information. In order to address these issues, the objective of this paper is to integrate blockchain technology with healthcare records/data in order to provide better sharing of data without the fear of data tampering or security breach.

In blockchain each party data will be encrypted and then hashed and then store in blocks and each block will have chain of hashcode of one and other and each block hash code will be verify before storing new block and if attacker modify any block then that block hash code verification will be failed and due to this technique no attacker can be able to alter block data. In this project we are using Blockchain tool called Ethereum which contains Blockchain implementation and it store data in the form of blocks and this block contains encrypted and hash data. Each hash verification will be done with the help of PROOF OF WORK technique.

1.1 Scope

Nowadays, by developing the smart technologies, such as Internet of Things (IoT) and virtual and augmented reality, the majority of healthcare providers are interested in replacing the conventional healthcare systems with eHealth. The main objective of eHealth is to revolutionize health information management and improve the healthcare system through modern information and communication technologies (ICTs) (Pagliari et al., 2005). Delivering the health services and information, through smart IoT devices by using the Internet with the aim of reducing the cost, and enhancing the scalability and time efficiency is an application of eHealth (Sebestyen et al., 2014, Ahern et al., 2006). To achieve these potential benefits of eHealth, it is essential for healthcare providers and industries to regulate and optimize the electronic distribution and transmission of medical information to the physicians. Undoubtedly, the key component of eHealth systems is Electronic Health Record (EHR), which refers to storing a patient's paper chart digitally, with the purpose to promptly and securely make the information available for authorized third part (Iakovidis, 1998, Giannelos, 2019). EHRs, as a vital part of eHealth, may include all required information of a patient regarding medical history, diagnoses and treatments, laboratory results, and images (i.e., X-ray and MRI) (Shickel et al., 2018). They help to simplify the healthcare providers' work-flow by providing an access to evidence-based tools and practices to improve patient's care services.

CHAPTER -2

2.SYSTEM RERUIREMENT SPECIFICATIONS

2.1 What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

2.1 Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

2.2 Requirements Specification Document

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report,software document. A software document is primarily prepared for a project,

software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of SRS (Software Requirement Specification) document is to describe theexternal behaviour of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS.

This section introduces the requirement specification document for Word Building Game using Alexa which enlists functional as well as non-functional requirements.

2.3 Functional Requirement Specification

The System after careful analysis has been identified to be present with the following modules.

FR1) Authenticated access control — the proposed scheme has to ensure an efficient access control to outsourced data, where requesting entities are authenticated.

FR2) Management efficiency — the proposed scheme should offer efficient management processes.

FR3) Privacy through pseudonymity and unlinkability measuresl

FR4) Auditability — each data owner should have a transparent view over how data are collected, accessed and processed.

2.4 Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement

specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

1) The complexity of the user interface should be minimal.

2) The platform chosen for implementation should be scalable.

3) A key challenge is to protect private, confidential, and sensitive information.

4) It should be secured by design guaranteeing data security and privacy protection with role-based access.

2.6 Software Requirements:

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idel 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupiter (or)**
- **Google colab**

2.7 Hardware Requirement(min):

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system : windows, linux**
- **Processor : minimum intel i3**
- **Ram : minimum 4 gb**
- **Hard disk : minimum 250gb**

CHAPTER -3

3. LITERATURE SURVEY

3.1.1 Blockchain and Health IT: Algorithms, Privacy, and Data

Authors: Allison Ackerman Shrier, Anne Chang, Nadia Diakun-thibault, Luca Forni, Fernando Landa, Jerry Mayo, Raul van Riezen

Abstract: The President's Precision Medicine Initiative (PMI) is "enabling a new era of clinical care through research, technology, and policies that empower patients, researchers, and providers to work together toward the development of individualized care". 1 Its commitment to privacy and security in the setting of responsible data sharing and transparency is articulated in the "Privacy and Trust Principles" 1 and the "Data Security Policy Principles and Framework" 2 , developed by an interagency working groups including the Office of the National Coordinator for Health Information Technology in conjunction with multiple stakeholders. In this paper, we review the threats to the security, confidentiality, integrity, and availability of PMI data. PMI organizations can mitigate these challenges through a new system architecture in development at MIT-the OPAL/Enigma project 3-which creates a peer-to-peer network that enables parties to jointly store and analyze data with complete privacy, based on highly optimized version of multi-party computation with a secret-sharing. An auditable, tamper-proof distributed ledger (a permissioned blockchain) records and controls access through smart contracts and digital identities. We conclude with an initial use case of OPAL/Enigma that could empower precision medicine clinical trials and research. MIT's OPAL/Enigma challenges traditional data security paradigms. Centralized databases cannot assure security and data integrity, regardless de-identification and controlled access requirements. Safe, vetted queries that are distributed to private, encrypted databases assure that organizations and participants can share health care data with cryptographic guarantees of privacy with various stakeholders, assuring momentum for a new era of medical research and practice.

3.1.2 Blockchain-Based Data Preservation System for Medical Data

Authors: Li, H., et al.,

Abstract: Medical care has become an indispensable part of people’s lives, with a dramatic increase in the volume of medical data (e.g., diagnosis certificates and medical records). Medical data, however, is easily stolen, tampered with, or even completely deleted. If the above occurs, medical data cannot be recorded or retrieved in a reliable manner, resulting in delay

treatment progress, even endanger the patient's life. In this paper, we propose a novel blockchain-based data preservation system (DPS) for medical data. To provide a reliable storage solution to ensure the primitiveness and verifiability of stored data while preserving privacy for users, we leverage the blockchain framework. With the proposed DPS, users can preserve important data in perpetuity, and the originality of the data can be verified if tampering is suspected. In addition, we use prudent data storage strategies and a variety of cryptographic algorithms to guarantee user privacy; e.g., an adversary is unable to read the plain text even if the data are stolen. We implement a prototype of the DPS based on the real world blockchain-based platform Ethereum. Performance evaluation results demonstrate the effectiveness and efficiency of the proposed system.

3.1.3 MedRec: Using Blockchain for Medical Data Access and Permission Management

Authors: A. Azaria, A. Ekblaw, T. Vieira and A. Lippman Abstract: Years of heavy regulation and bureaucratic inefficiency have slowed innovation for electronic medical records (EMRs). We now face a critical need for such innovation, as personalization and data science prompt patients to engage in the details of their healthcare and restore agency over their medical data. In this paper, we propose MedRec: a novel, decentralized record management system to handle EMRs, using blockchain technology. Our system gives patients a comprehensive, immutable log and easy access to their medical information across providers and treatment sites. Leveraging unique blockchain properties, MedRec manages authentication, confidentiality, accountability and data sharing- crucial considerations when handling sensitive information. A modular design integrates with providers' existing, local data storage solutions, facilitating interoperability and making our system convenient and adaptable. We incentivize medical stakeholders (researchers, public health authorities, etc.) to participate in the network as blockchain "miners". This provides them with access to aggregate, anonymized data as mining rewards, in return for sustaining and securing the network via Proof of Work. MedRec thus enables the emergence of data economics, supplying big data to empower researchers while engaging patients and providers in the choice to release metadata. The purpose of this short paper is to expose, prior to field tests, a working prototype through which we analyze and discuss our approach.

3.2 Existing System

Though In traditional approach these data are stored by each hospital separately where

sharing data is very time consuming. Also, health data is very limited to access due to the risk of disclosure of sensitive information.

The following are the drawbacks of the existing manual System:

1. Time consuming
2. Limited

3.3 Proposed System

In order to address these issues, the objective of this paper is to integrate blockchain technology with healthcare records/data in order to provide better sharing of data without the fear of data tampering or security breach.

Advantages of proposed system:

1. Better sharing

2.Enhanced Security

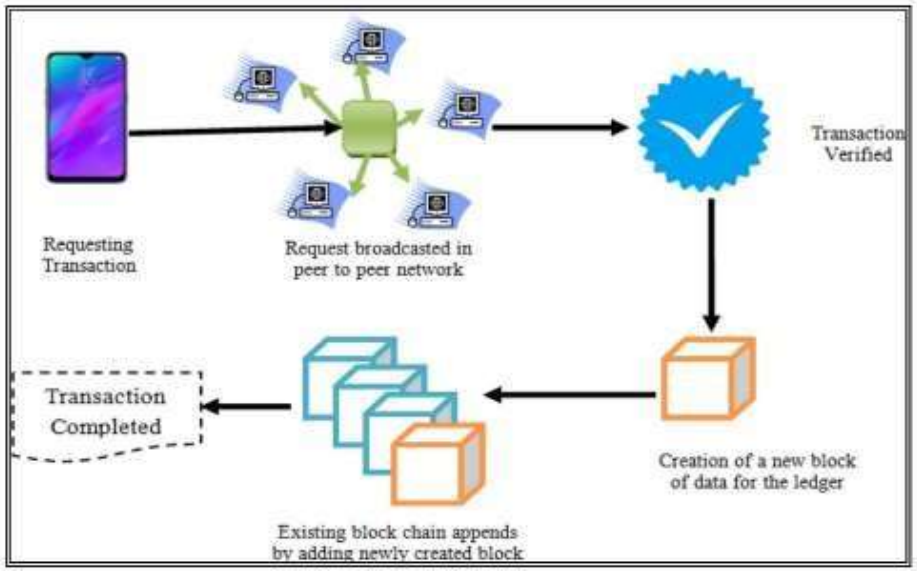
3.Fatser medical credentialing

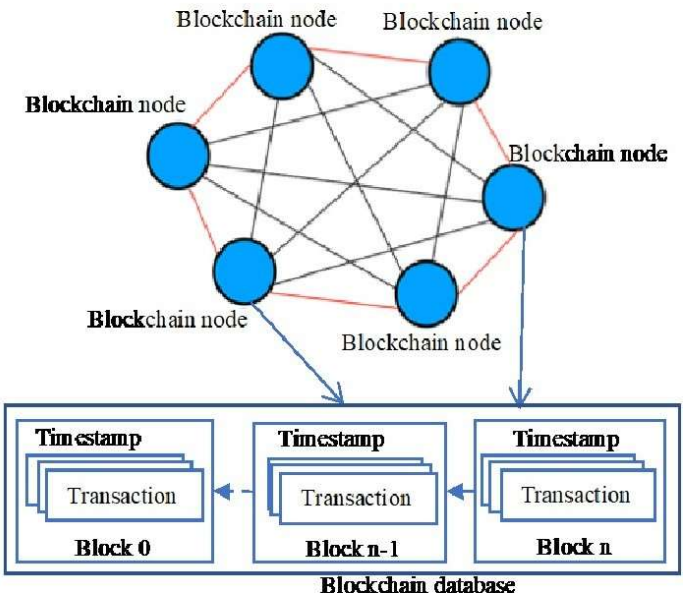
4.Commitments enforced via smart contracts.

CHAPTER -4

4 SYSTEM DESIGN

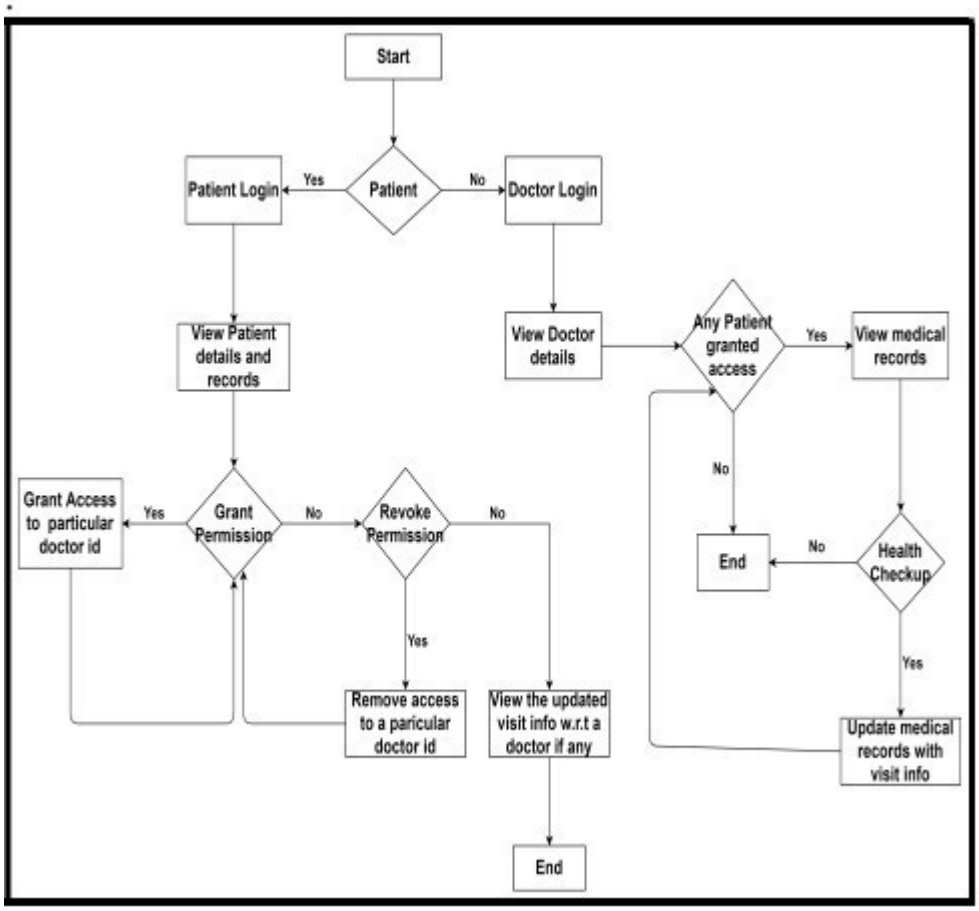
4.1 System Architecture





Flow Chart:

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.



4.2 Introduction to UML

Unified Modeling Language allows the software engineer to express an analysismodel using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describethe system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

1. **User Model View**

   This view represents the system from the users' perspective. The analysis representation describes a usage scenario from the end-users' perspective.

2. **Structural Model View**

   In this model, the data and functionality are arrived from inside the system. This model view models the static structures.

3. **Behavioural Model View**

   It represents the dynamic of behavioural as parts of the system, depicting he interactions of collection between various structural elements described in the user model and structural model view.

4. **Implementation Model View**

   In this view, the structural and behavioural as parts of the system are represented as they are to be built.

5. **Environmental Model View**

   In this view, the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

4.3 UML Diagrams

4.3.1 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified. In brief, the purposes of use case diagrams can be as follows:

a. Used to gather requirements of a system.
b. Used to get an outside view of a system.
c. Identify external and internal factors influencing the system.
d. Show the interacting among the requirements are actors.

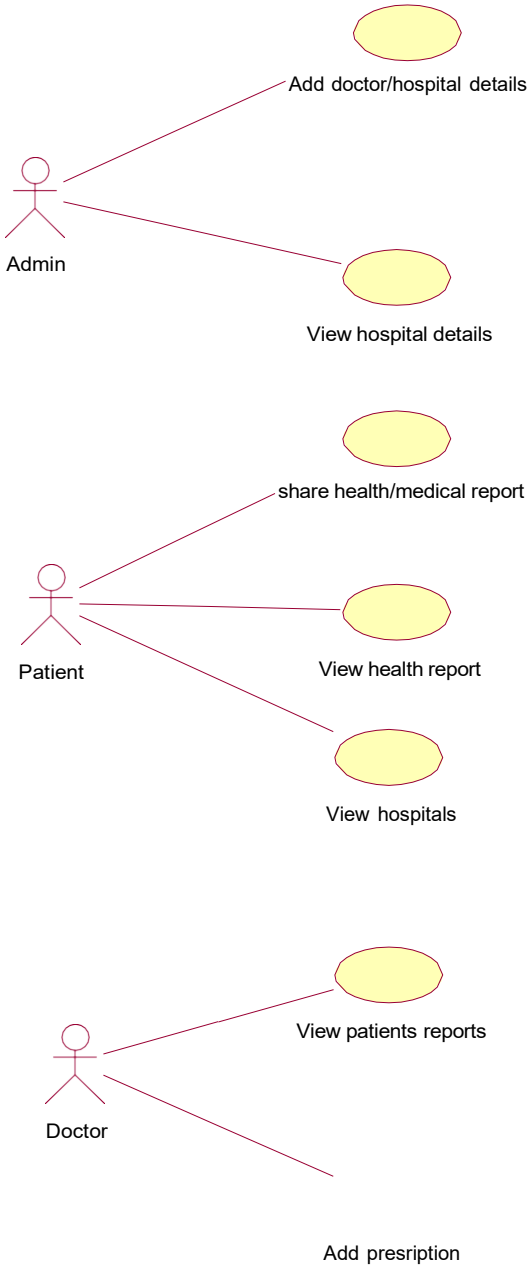


Fig 4.2.1 – Use Case Diagram

4.3.2 Class Diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

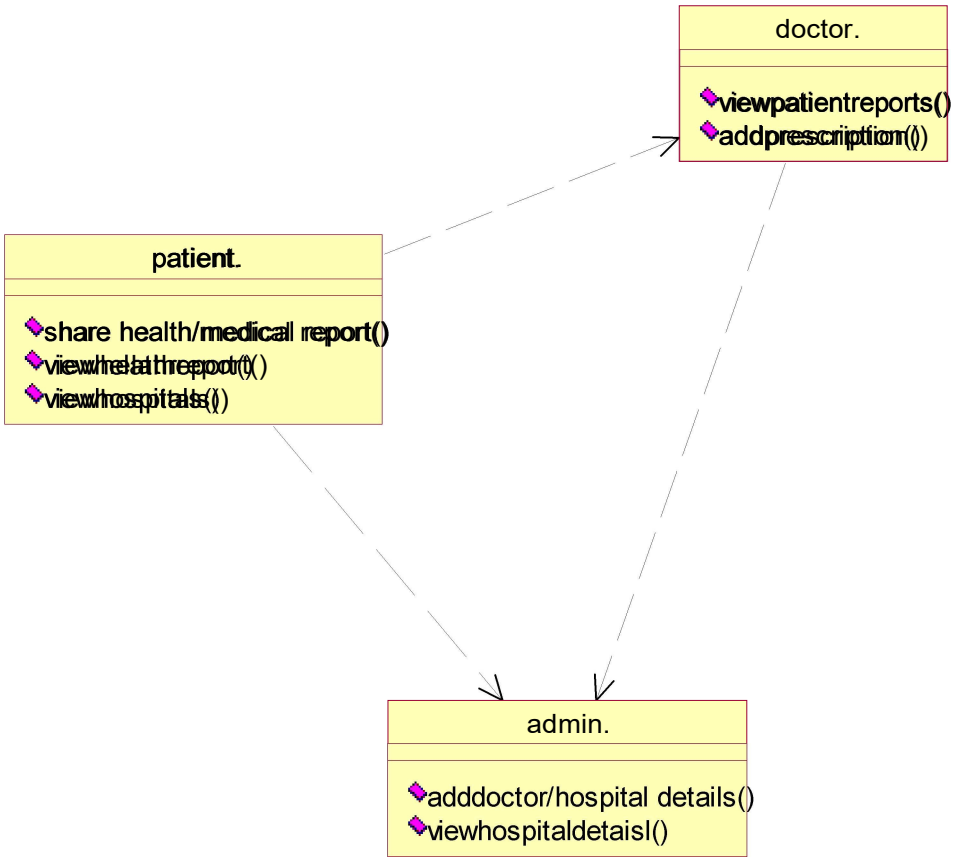


Fig 4.2.2 – Class Diagram

4.3.3 Activity Diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

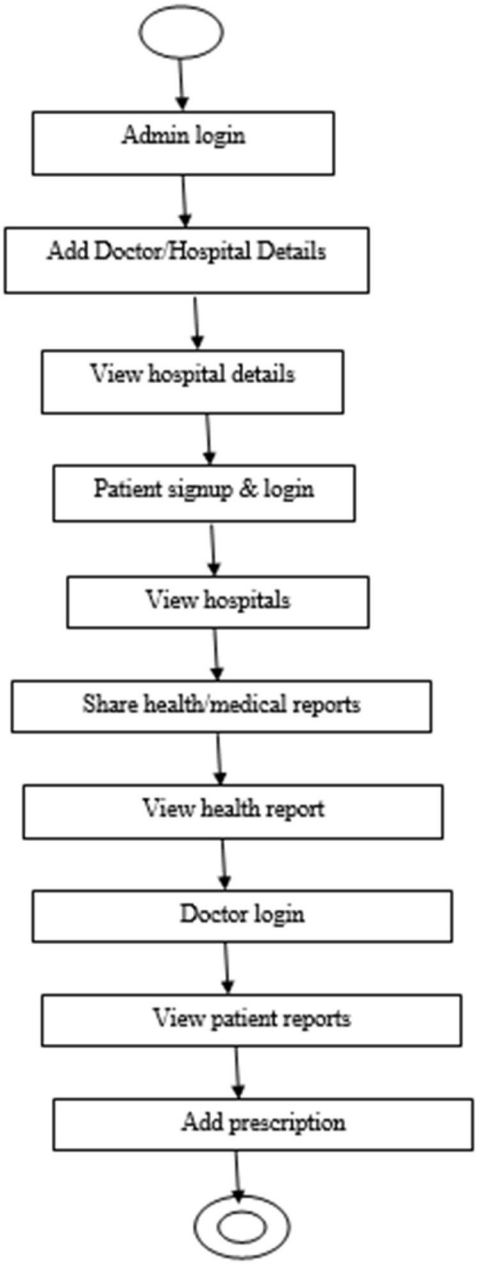


Fig 4.2.3 – Activity Diagram

4.3.4 Sequence Diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

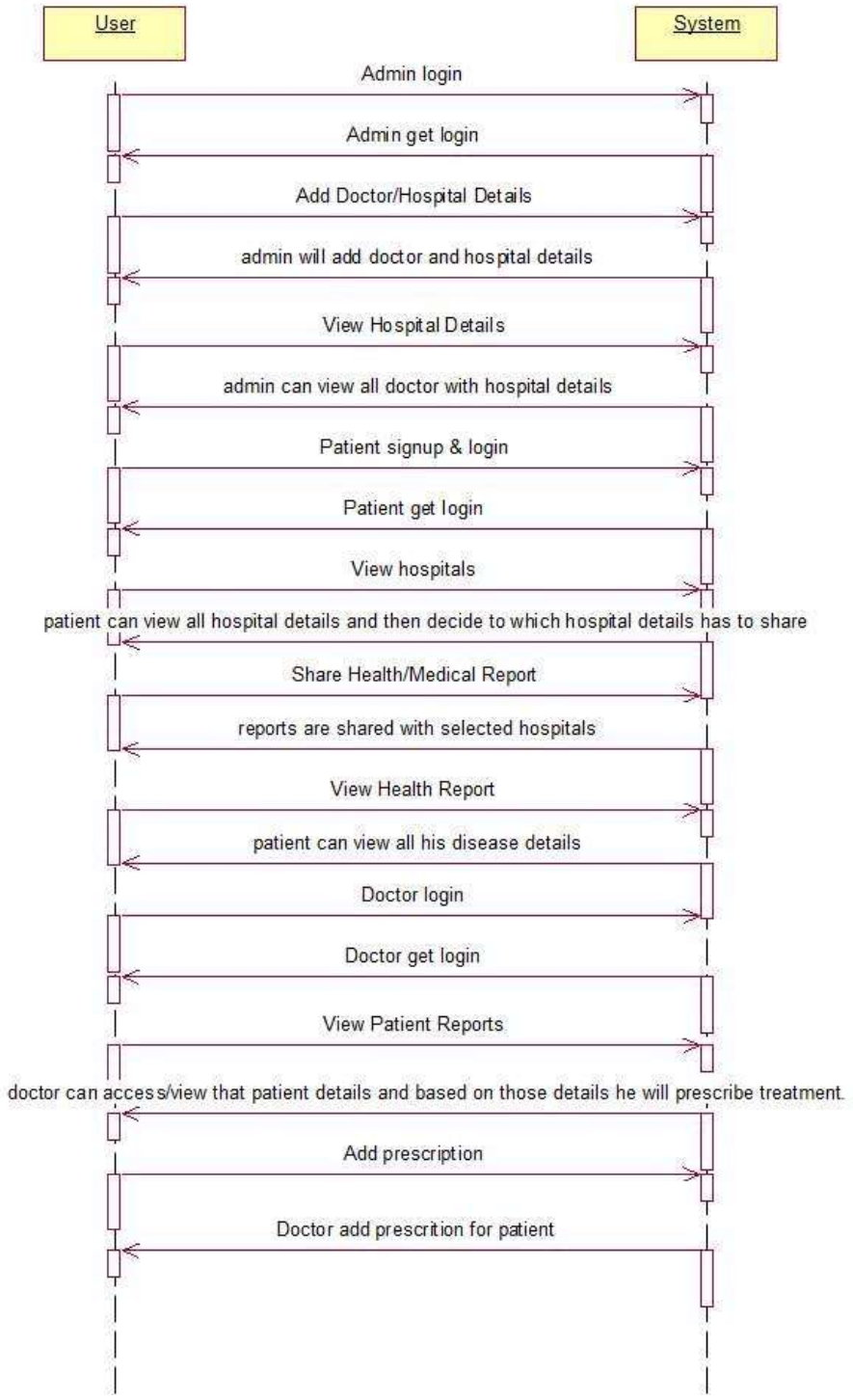


Fig 4.2.4 – Sequence Diagram

4.3.5 Collaboration Diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

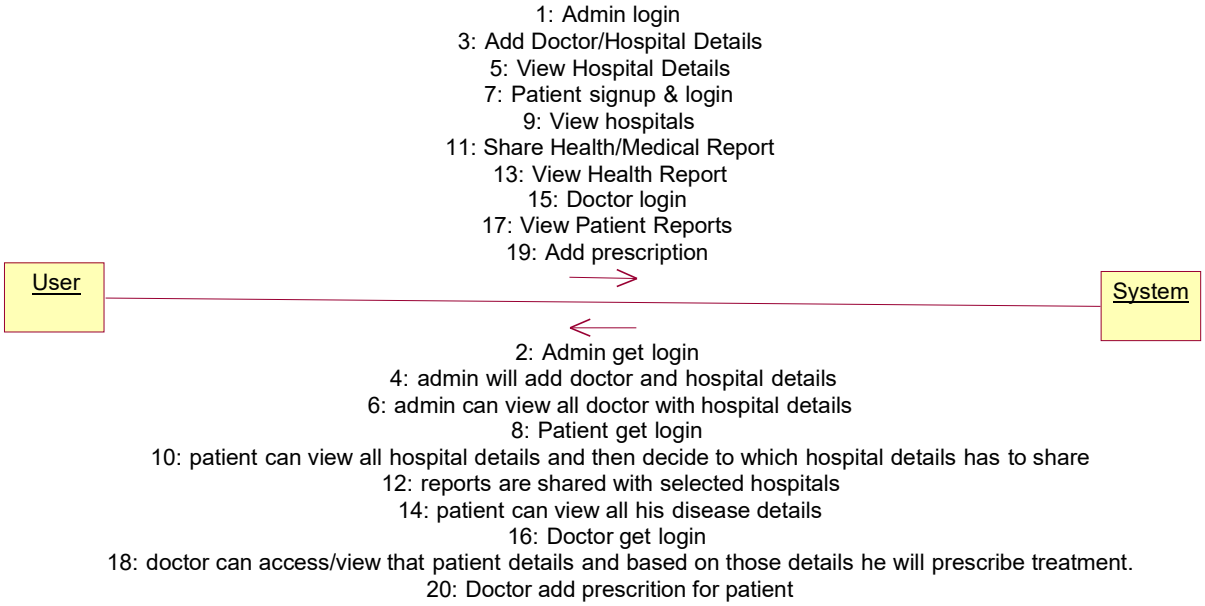
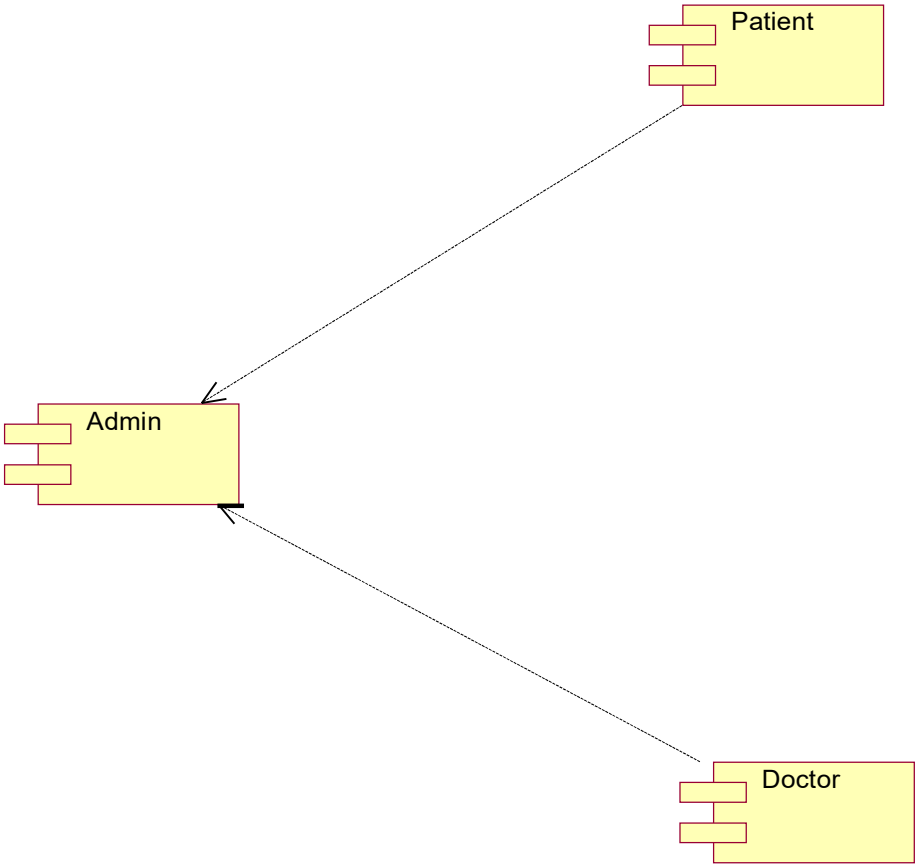


Fig 4.2.5 – Collaboration Diagram

4.3.6 Component Diagram:

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



4.3.7 Deployment Diagram:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

CHAPTER -5

5. IMPLEMENTATION

5.1 Introduction

The Blockchain gaining its popularity in almost all fields due to its inbuilt support for data integrity, reliability and data security. Healthcare records often contains sensitive information such as family diseases and if attackers hack this data then patient sensitive disease information will be leak/alter as existing application maintaining data in single centralized server and if this server hack or attack then server will not give any information.

To overcome from this issue author of this paper introducing Blockchain technology to manage healthcare records as this Blockchain will store all records as Blocks/transaction and then associate each block with unique Hashcode and maintain this information at multiple nodes or Blockchain Server. If one node/server down then users can get services from other working nodes and this service is not available in existing single centralized server.

Blockchain data is consider as immutable as data change in one node will generate different Hashcode while verification. This failure verification helps Blockchain in knowing about data leak or alter.

Data stored at multiple nodes can be access/share by multiple users by assigning permissions to them. In propose paper author is proposing that patient can share data with multiple hospitals who has access permission to this Blockchain nodes.

To implement this project we have used SOLIDITY code to define Blockchain functions to manage healthcare details such as Patient disease report, doctor details etc. We are using Python TRUFFLE Ethereum tool to deploy above Solidity code on Blockchain and once deploy then using python we can call that solidity contract to access Blockchain functions to store and retrieve patient details. Below screen showing Solidity code for patient functions

5.2 Algorithm

```
import json
import os
import socket
import subprocess
import tkinter as tk
```

```
from tkinter.constants import *
from tkinter import ttk, messagebox
from ttkthemes import ThemedTk


class App:
    def __init_(self, root):
        self.root = root
        self.path = 'data/peers.json'
        self.proceses = []
        self.assigned_peers, self.unassigned_peers = self.get_peers()

        self.get_ip()

        self.root.iconbitmap('background.ico')

        self.root.wm_title('EHR | Administrator Panel')

        # self.root.geometry("400x600")
        # setting window size
        self.get_window_size()
        self.root.resizable(False, False)

        tabControl = ttk.Notebook(self.root)
        self.tab1, self.tab2 = ttk.Frame(tabControl), ttk.Frame(tabControl)
        tabControl.add(self.tab1, text='Assigned Nodes', compound=TOP)
        tabControl.add(self.tab2, text='Unassigned Nodes', compound=TOP)
        tabControl.pack(expand=1, fill="both")

        self.tab_frame1 = self.get_notebook_frame(self.tab1)
        self.tab_frame2 = self.get_notebook_frame(self.tab2)
        self.get_widgets(self.tab_frame1, self.tab_frame2)

        ttk.Button(tabControl, text='Refresh',
                   command=lambda: self.refresh_details()).pack(padx=20, pady=20, side=BOTTOM)
```

```
    def get_widgets(self, tabframe1, tabframe2):
        ttk.Label(tabframe1, text="Manage {} available peer nodes" .format(len(self.assigned_peers))
if self.assigned_peers else "No assigned nodes to manage").pack(
            side=tk.TOP, padx=40, pady=20)

        ttk.Label(tabframe2, text="There are {} pending peer nodes"
.format(len(self.unassigned_peers)) if self.unassigned_peers else "No unassigned nodes to
assign").pack(
            side=tk.TOP, padx=20, pady=20)

        for index, peer in enumerate(self.assigned_peers):
            self.assigned_frame = ttk.Frame(tabframe1)
            self.assigned_frame.pack(fill="both", expand=1)
            ttk.Label(self.assigned_frame, text="{}. Peer Node {}".format(index+1,
peer['assigned_port'],)).grid(
                padx=40, pady=10, column=1, row=0)
            ttk.Button(self.assigned_frame, text="Start Node", command=lambda peer=peer:
self.start_node(peer['assigned_port'])).grid(
                padx=70, pady=10, column=2, row=0)
            ttk.Separator(self.assigned_frame, orient='horizontal').grid(
                padx=60, pady=10, row=1, columnspan=3, sticky="ew")

        for index, peer in enumerate(self.unassigned_peers):
            self.unassigned_frame = ttk.Frame(tabframe2)
            self.unassigned_frame.pack(fill="both", expand=1)
            ttk.Label(self.unassigned_frame, text="{}. Peer Node {}".format(index+1,
peer['assigned_port'],)).grid(
                padx=40, pady=10, column=1, row=0)
            ttk.Button(self.unassigned_frame, text="Assign Node", command=lambda peer=peer:
self.assign_port(peer)).grid(
                padx=70, pady=10, column=2, row=0)
            ttk.Separator(self.unassigned_frame, orient='horizontal').grid(
                padx=60, pady=10, row=1, columnspan=3, sticky="ew")

    def assign_port(self, peer_details):
        with open(self.path) as json_file:
            data = json.load(json_file)
```

```
            assigned, unassigned = data['assigned'], data['unassigned']
            assigned.append(peer_details)
            unassigned.remove(peer_details)
        with open(self.path, mode='w') as f:
            json.dump(data, f, indent=4)

        for widget in self.tab2.winfo_children():

            widget.destroy()
        self.unassigned_frame.destroy()
        self.refresh_details()

    def refresh_details(self):
        self.get_ip()
        self.assigned_peers, self.unassigned_peers = self.get_peers()
        for widget in self.tab1.winfo_children():
            widget.destroy()
        for widget in self.tab2.winfo_children():
            widget.destroy()

        self.tab_frame1 = self.get_notebook_frame(self.tab1)
        self.tab_frame2 = self.get_notebook_frame(self.tab2)
        self.get_widgets(self.tab_frame1, self.tab_frame2)
        self.root.after(600000, self.refresh_details)

    def get_window_size(self):
        width, height = 400, 600
        difwidth = self.root.winfo_screenwidth()-width
        difheight = self.root.winfo_screenheight()-height
        alignstr = '%dx%d+%d+%d'.format() % (width, height,
                                    difwidth/2, difheight/2)
        root.geometry(alignstr)

    def get_ip(self):
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
        s.connect(("8.8.8.8", 80))
        self.ip_address = s.getsockname()[0]
        s.close()

    def get_peers(self):
        if os.path.exists(self.path):
            with open(self.path) as json_file:
                loaded = json.load(json_file)
                return loaded['assigned'], loaded['unassigned']

    def start_node(self, peer):
        self.pid = subprocess.Popen(
            "python node.py -p {} --host {}".format(peer, self.ip_address), shell=True).pid
        self.proceses.append(self.pid)

    def on_closing(self):
        if messagebox.askokcancel("Quit", "Do you want to quit?"):
            for pid in self.proceses:
                subprocess.Popen("TASKKILL /F /PID {} /T".format(pid))
            root.destroy()

    def get_notebook_frame(self, frame):
        s = ttk.Style()
        bg = s.lookup('TFrame', 'background')
        canvas = tk.Canvas(frame, height=500, background=bg)
        scrollbar = ttk.Scrollbar(
            frame, orient=VERTICAL, command=canvas.yview)
        canvas.bind('<Configure>', lambda e: canvas.configure(
            scrollregion=canvas.bbox(ALL), highlightthickness=0))
        canvas.focus_set()
        canvas.focus("")
        tab_frame = ttk.Frame(canvas, height=200)
        canvas.create_window((0, 0), window=tab_frame, anchor=NW)
        canvas.pack(side=LEFT, fill="both", expand=1)
        scrollbar.pack(side=RIGHT, fill=Y, anchor=NE)
```

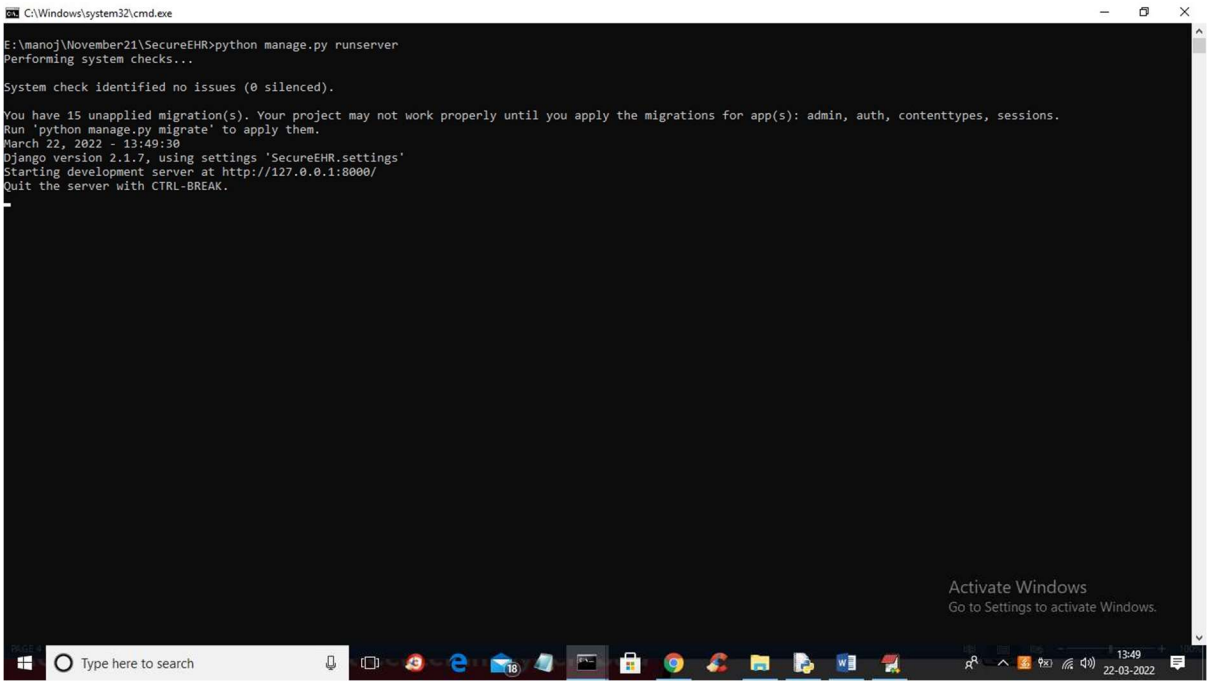
```
        return tab_frame



# adapta
# yaru
if __name___== "__main__":
    root = ThemedTk(theme="adapta")
    style = ttk.Style()
    style.configure('.', focuscolor='', highlightbackground='',
                    font=('Segoe UI', 10))
    app = App(root)
    app.start_node(2)
    root.protocol("WM_DELETE_WINDOW", app.on_closing)
    root.mainloop()
```

CHAPTER -6

6 SCREENSHOTS



To run project first double click on 'run.bat' file to start python Django server and to get below screen. In above screen python server started and now open browser and enter URL as 'http://127.0.0.1:8000/index.html' and press enter key to get below home page.

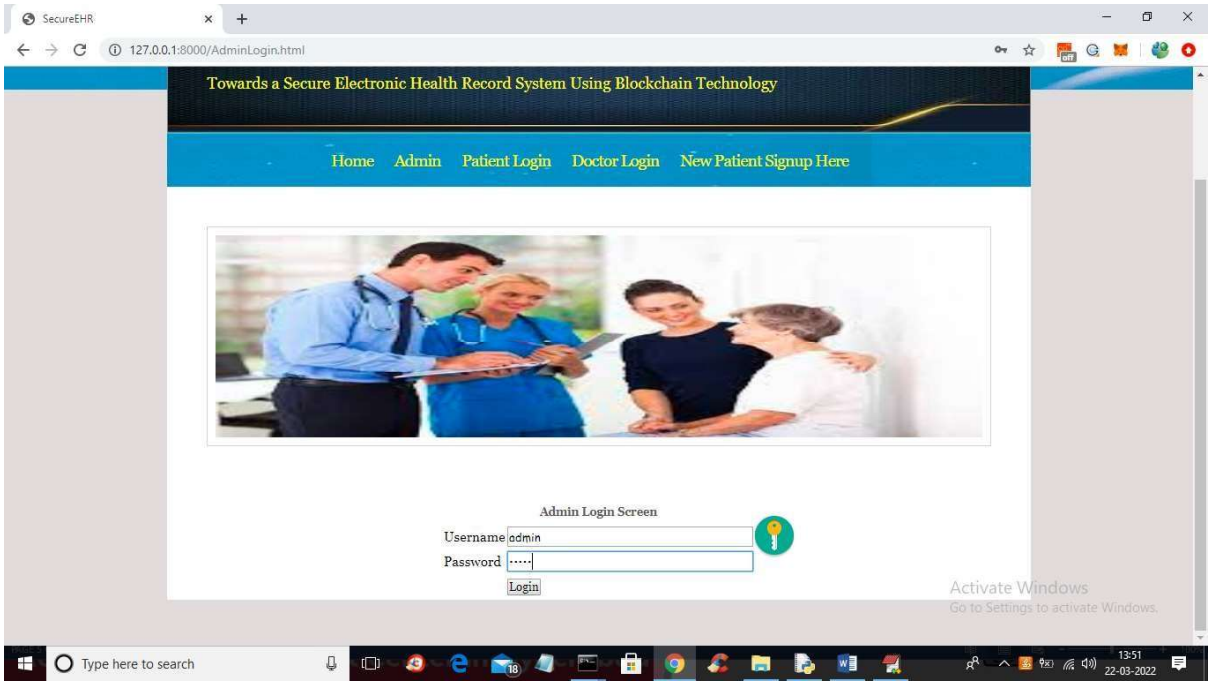
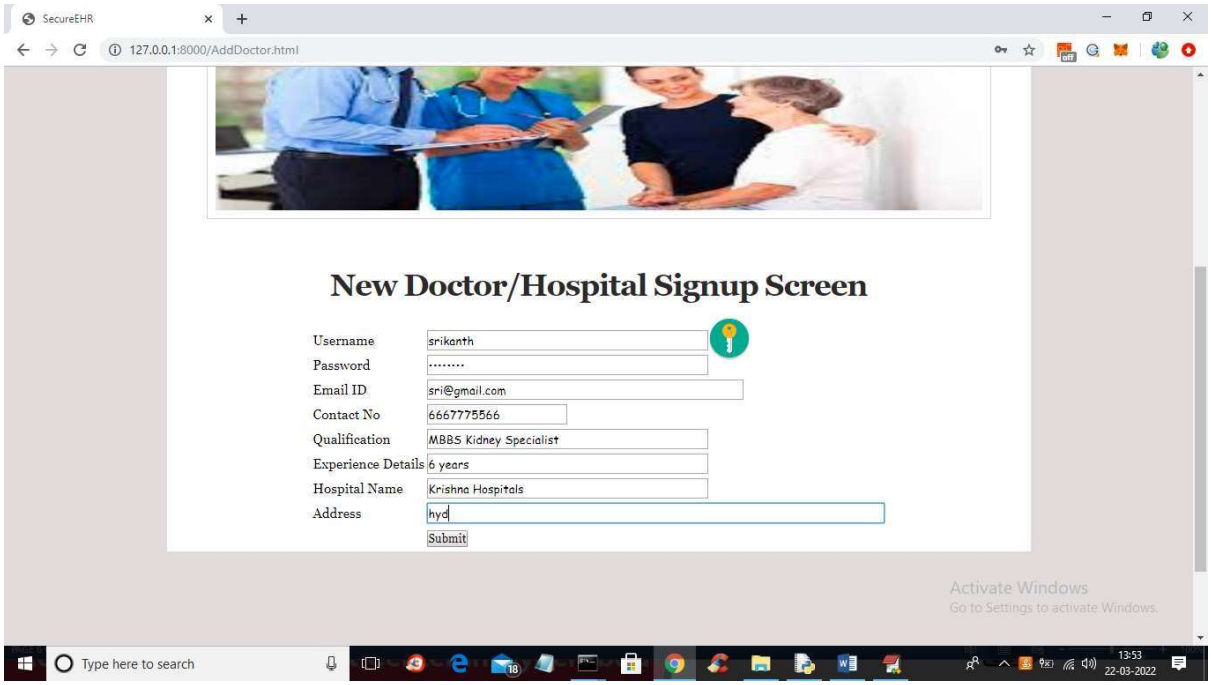


Fig.6.1 Login page



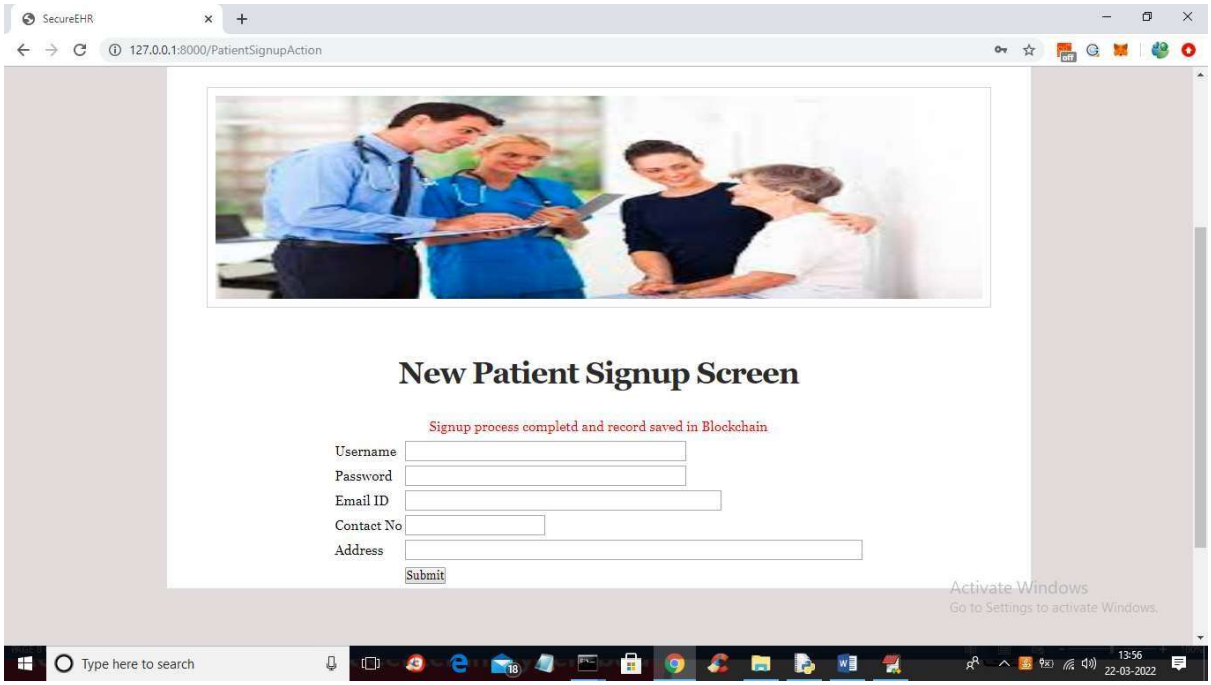


Fig.6.2 Registration page

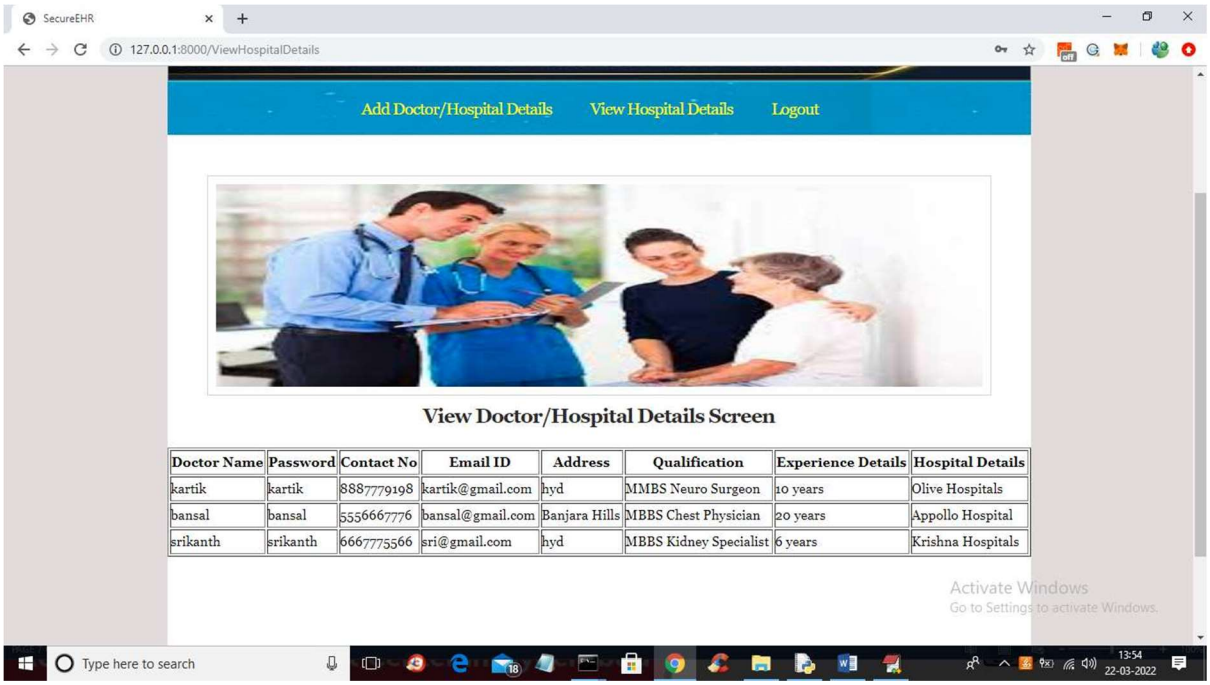


Fig.6.3 View Hospital Details (Patient's view)

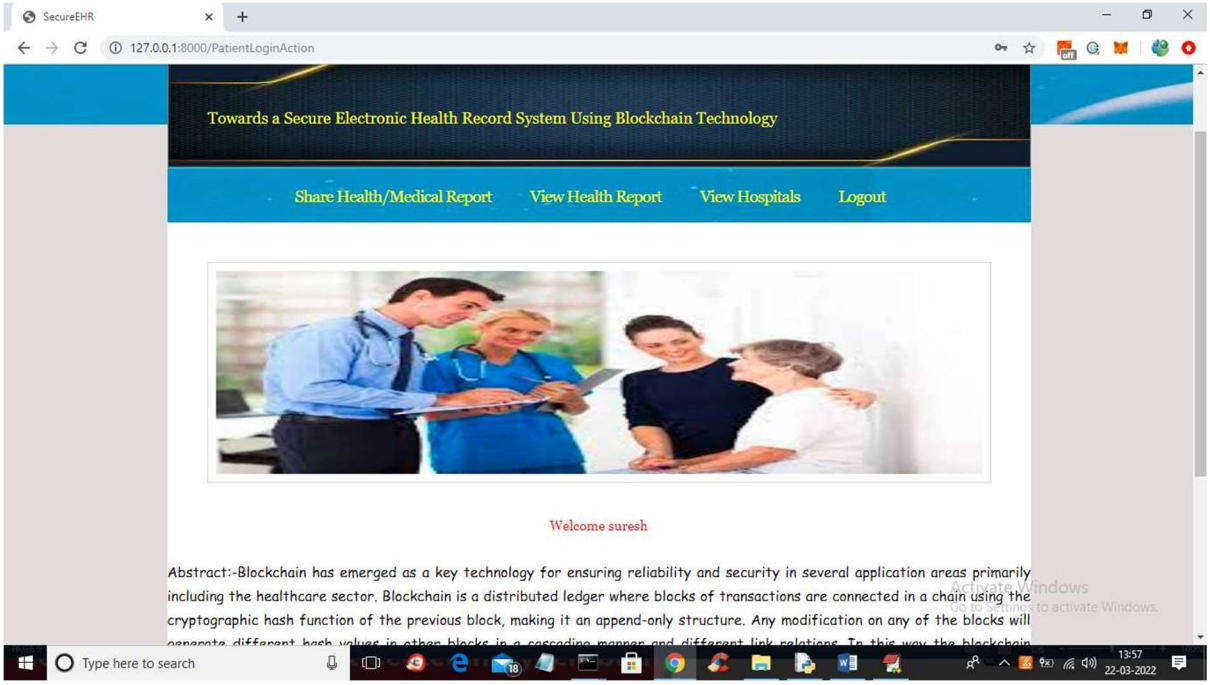


Fig.6.4 Home Page

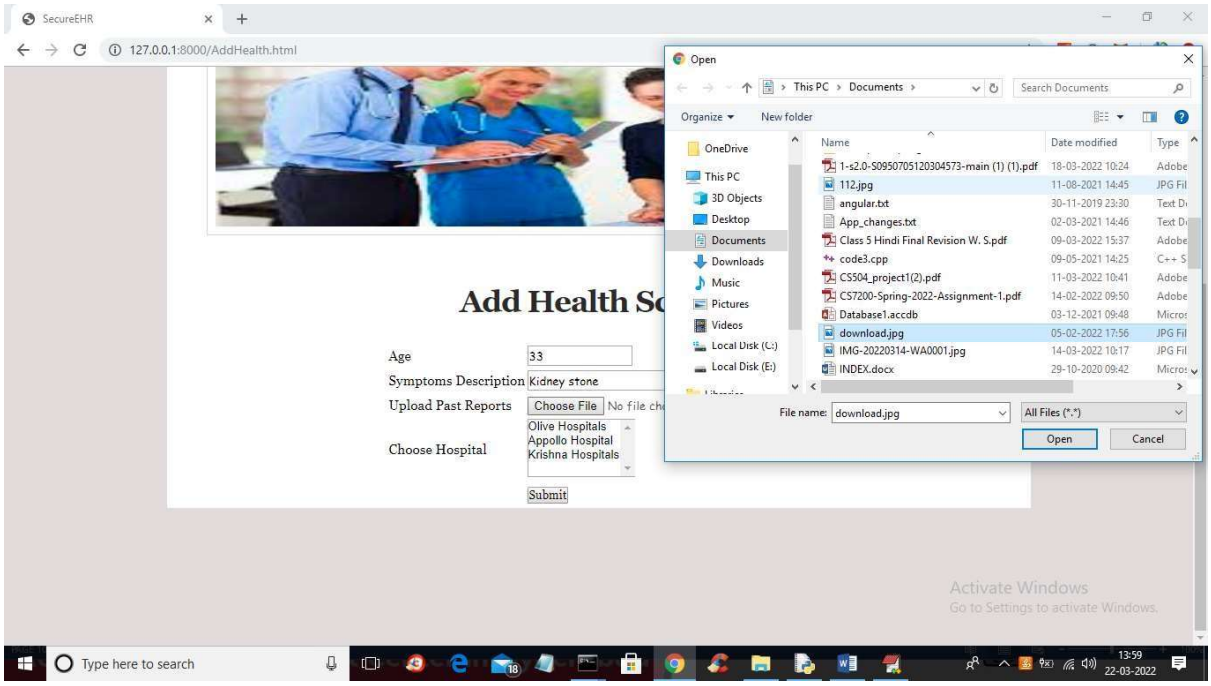


Fig.6.5 Share Health/Medical Report (Patient ‘sView)

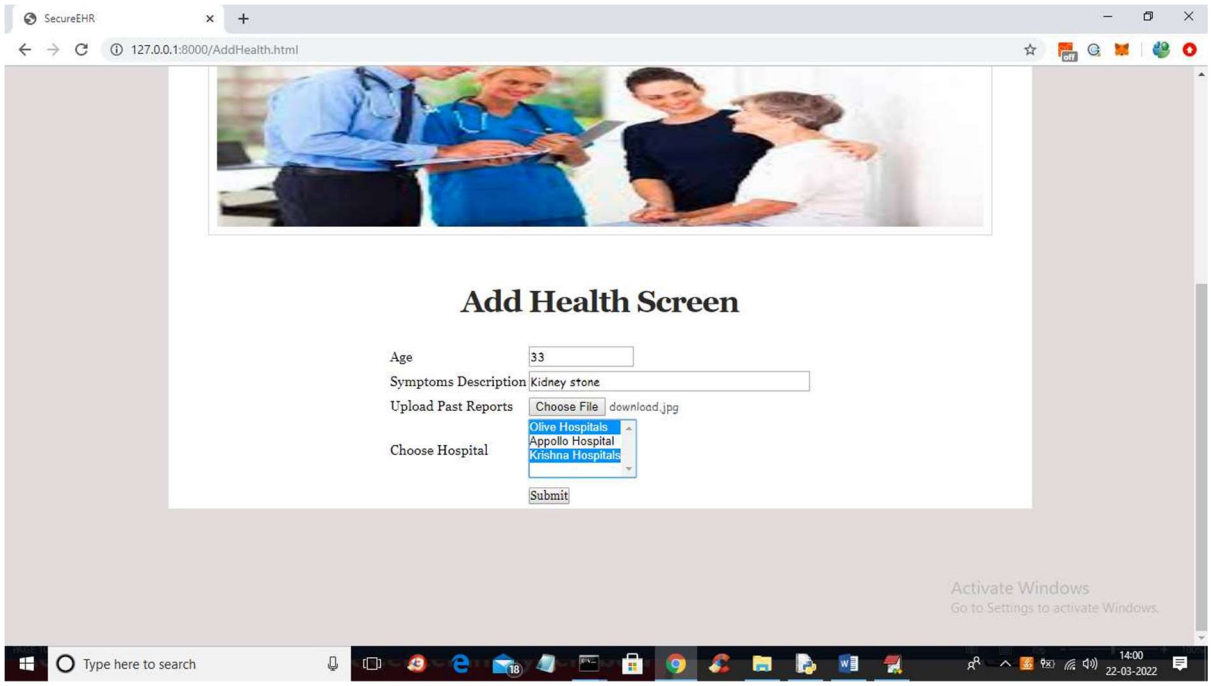
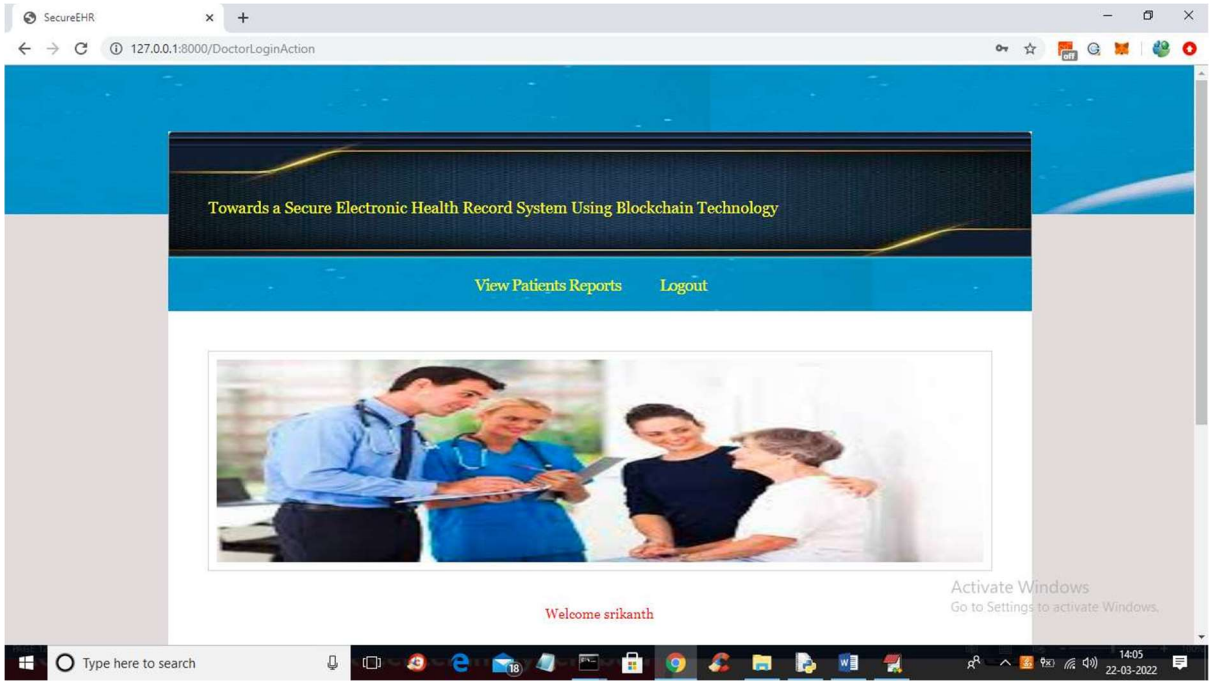


Fig.6.6 Share Details (Patient’s View)

Fig.6.7 Past report details (Patient's View)



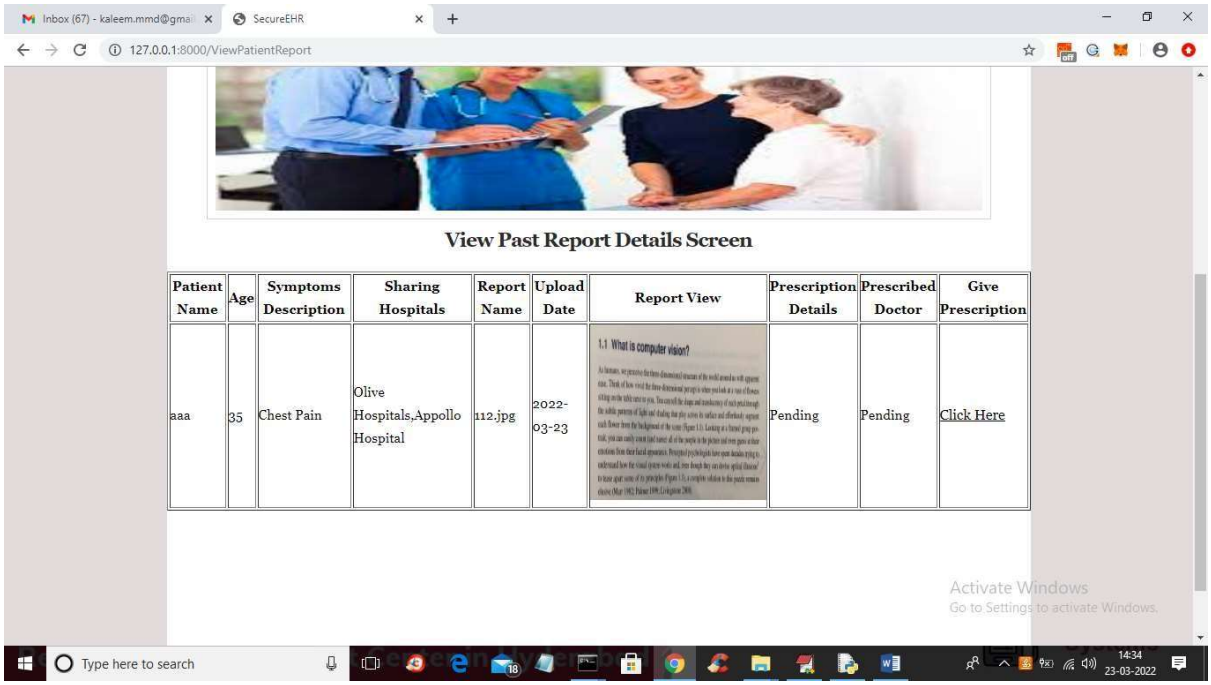


Fig.6.8 View Patient Reports (Doctor's View)

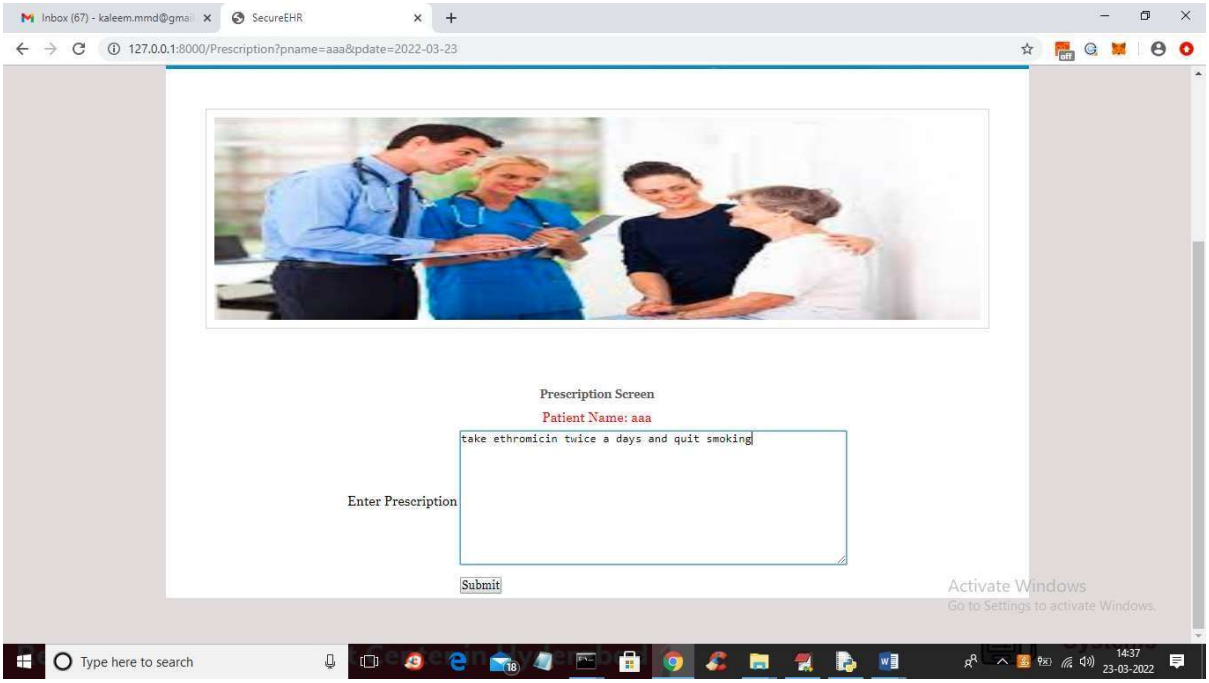


Fig.6.9 Adding Prescription

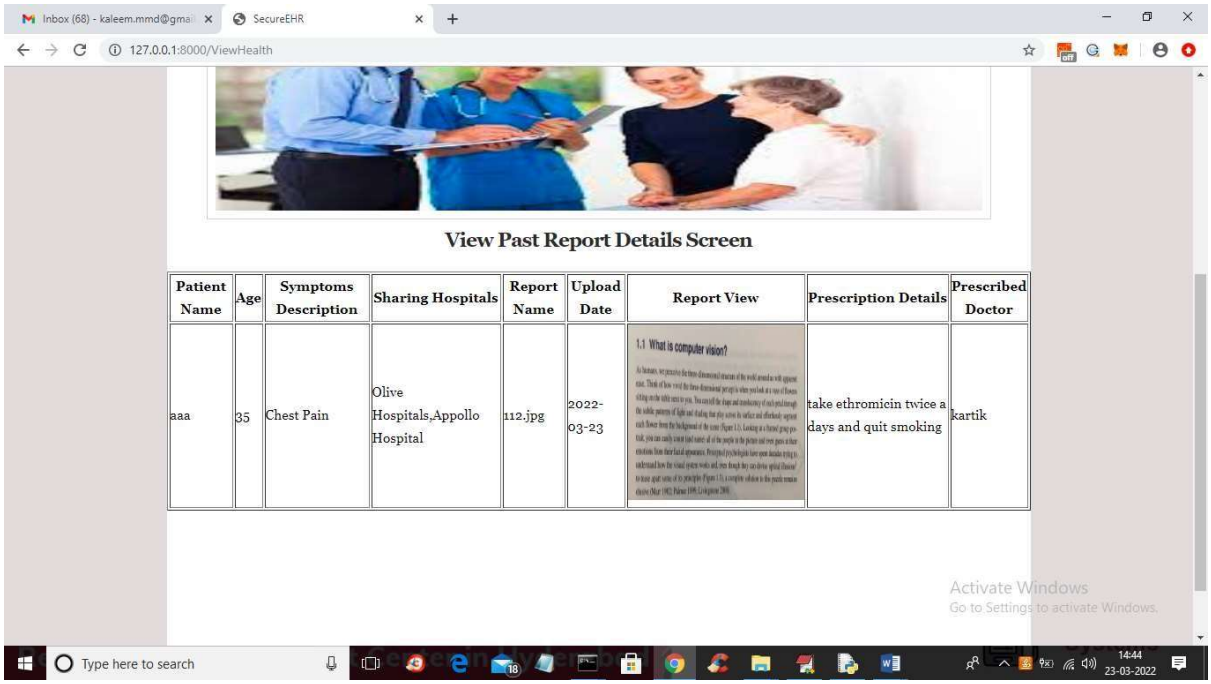


Fig.6.10 Viewing Prescription (Patient's View)

CHAPTER -7

7.TESTING

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification

7.1 SOFTWARE VALIDATION

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – "Are we developing the product which attemptsall that user needs from this software?".
- Validation emphasizes on user requirements.

7.2 SOFTWARE VERIFICATION

Verification is the process of confirming if the software is meeting the business requirements and is developed adhering to the proper specifications and methodologies.

- Verification ensures the product being developed is according to design specifications.
- Verification answers the question– "Are we developing this product by firmly following all design specifications?"
- Verifications concentrate on the design and system specifications.

7.3 TARGET OF THE TEST ARE

- Errors -These are actual coding mistakes made by developers. In addition, there is adifference in output of software and desired output, considered as an error.
- Fault - When error exists fault occurs. A fault, also known as a bug, is a result of anerror which can cause system to fail.
- Failure - failure is said to be the inability of the system to perform the desired task.Failure occurs when fault exists in the system.

7.4 BLACK-BOX TESTING

It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

7.5 Black-Box Testing Techniques

- Equivalence class - T he input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
- Boundary values - T he input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
- Cause-effect graphing - In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
- Pair-wise Testing - The behavior of software depends on multiple parameters. In pair wise testing, the multiple parameters are tested pair-wise for their different values.
- State-based testing - The system changes state on provision of input. These systems are tested based on their states and input.

7.6 WHITE-BOX TESTING

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing

In this testing method, the de sign and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

The following are some White box testing techniques

- Control-flow testing - The purpose of the control-flow testing to set up a test case which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
- Data-flow testing - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

7.7 TESTING LEVELS

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified. Testing separately is done just to make sure that there are no hidden bugs or issues left in the software.

7.8 Unit Testing

While coding, the programmer performs some tests on that unit of program to know if 64 it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

7.9 Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passes and data updating etc.

7.10 System Testing

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

- Functionality testing - Tests all functionalities of the software against requirement.
- Performance testing - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.
- Security & Portability - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

7.11 Acceptance Testing

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears

or works, it may be rejected.

- Alpha testing - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would reactto some action in software and how the system should respond to inputs.
- Beta testing - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute 65 problems, which were skipped to attend.

7.12 Regression Testing

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code.

7.13 End-to-End Testing

End-to-End testing is a type of Software testing that not only validates the software system under test but also check its integration with external interfaces. It uses actual production like data and test environment to simulate real-time settings. End-to-End testing is also called Chain testing. End-to-End design framework consists of three parts: Build User functions, Build conditions, Build test cases.

7.14 TEST CASES:

S.NO	INPUT	If available	If not available
1	Admin login	admin can login to application	There is no process
2	Add Doctor/Hospital Details	admin will add doctor and hospital details	There is no process

3	View Hospital Details	admin can view all doctor with hospital details	There is no process
4	Update quantity	supplier can update quantity for the product in Blockchain	There is no process
5	Patient signup & login	patient can login to application	There is no process
6	View Hospitals	patient can view all hospital details	There is no process
7	Share Health/Medical Report	reports are shared with selected hospitals	There is no process
8	Doctor login	Doctor can login to application	There is no process
9	View Patient Reports	doctor can access/view that patient details	There is no process
10	Add prescription	Doctor can add prescription for patient	There is no process

CHAPTER -8

8 FUTURE ENHANCEMENTS

The future work consists of two aspects. Basically the proposed system is a prototype to get a idea of functioning of blockchain while it is integrated with EHR. First thing is to add more participants like hospital, insurance service provider etc. Next thing is to scale up the system and maintain off-chain database so that the real prescription in the form of image etc. can be stored. Also search will be continued for a better blockchain like hybrid or layered to get leverage from it.

We noted a significant need to enforce appropriate access control mechanisms for the IoT systems from unauthorized entities. It requires the implementation of lightweight, flexible, and controlled access control mechanisms that fulfil the specific characteristics (e.g., dynamic, large-scale, etc.) of an IoT system. Most access control proposals deployed by conventional access control mechanisms must be re-examined for their feasibility in adopting lightweight security mechanisms and flexibility.

With the rapid development of the IoT, research in the IoT landscape has become an important issue, and it continues to grow every day. However, security is an essential concern for IoT systems. Among others, the demand for securing access control is paramount. We have seen that conventional access control mechanisms cannot efficiently manage access control policies and enforce authorization decisions for large-scale IoT systems. That said, IoT systems require a unique solution for access

CHAPTER -9

9 CONCLUSION

In this work an architecture and a methodology based to create an EMR system based on blockchain is proposed. Also the blockchain type that can be used for this purpose is identified with justification. And the proposed methodology is implemented using Hyperledger fabric and Hyperledger composer tools. Here patients can actively control his medical record. Also the security and privacy issues have been taken care of. The patient decides on who gets access to his/her medical records via doctorId and may revoke access permission for the same. On the other hand, the doctor may access the patient's medical history to read and analyse the patient's situation better, given all medical history logs are updated timely on the blockchain network. The doctor may update the current visit with prescriptions and medical procedures, if any, on blockchain. The limitations of the paper can be divided into three aspects mainly. Firstly the system is not scalable. Secondly the participants are limited to doctors and patients only. And finally the information to be stored by the database is very limited. Only some basic diagnosis information is considered and that also is only text based. The future work consists of two aspects. Basically the proposed system is a prototype to get a idea of functioning of blockchain while it is integrated with EHR. First thing is to add more participants like hospital, insurance service provider etc. Next thing is to scale up the system and maintain off-chain database so that the real prescription in the form of image etc. can be stored. Also search will be continued for a better blockchain like hybrid or layered to get leverage from it.

CHAPTER -10

10 REFERENCES

i. [1] King Yip, “BlockChain and alternative payment models,” White paper, 2016.

ii. [2] Blockchain and Health IT: Algorithms, Privacy, and Data. Prepared by: Allison Ackerman Shrier, Anne Chang, Nadia Diakun-thibault, Luca Forni, Fernando Landa, Jerry Mayo, Raul van Riezen.

iii. [3] Li, H., et al., Blockchain-Based Data Preservation System for Medical Data J J. Med. Syst. 2018. 42(8): p. 1- 13.

iv. [4] Nguyen, D.C. , Pathirana, P.N. , Ding, M. , Seneviratne, A. , 2019. Blockchain for secure ehrs sharing of mobile cloud based e-health systems. IEEE Access .

v. [5] Nitesh Gupta, Anand Jha, and Purna Roy, “Adopting Blockchain Technology for Electronic Health Record Interoperability”, 2016.

vi. [6] Laure A., Linn Martha B., Koo, M.D, “Blockchain For Health Data and Its Potential Use in Health IT and Health Care Related Research”, 2016

vii. [7] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," 2016 2nd International Conference on Open and Big Data (OBD), Vienna, 2016, pp. 25-30, doi: 10.1109/OBD.2016.11.

viii. [8] Al Omar, A. , Rahman, M.S. , Basu, A. , Kiyomoto, S. , 2017. Medibchain: a blockchain based privacy preserving platform for healthcare data. In: International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage. Springer, pp. 534–543.

ix. [9] Lee, S.H. , Yang, C.S. ,2018. Fingernail analysis management system using microscopy sensor and blockchain technology. International Journal of Distributed Sensor Networks 14 (3) . 1550147718767044.

x. [10] Zheng, X. , Mukkamala, R.R. , Vatrapu, R. , Ordieres-Mere, J. , 2018. Blockchain-based personal health data sharing system using cloud storage. In: 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom). IEEE, pp. 1–6.