Decentralized Healthcare Network Platform

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Shankha Shubhra Sarkar 18BCE2453

Anindya Sen 18BCE2382

Under the guidance of

Prof. Krishnamoorthy A

School of Computer Science & Engineering

VIT, Vellore

June, 2022

DECLARATION

We hereby declare that the thesis entitled “Decentralized Healthcare Network Platform” submitted by us, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* to VIT is a record of bonafide work carried out by us under the supervision of Prof. Krishnamoorthy A.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 03.06.2022

Shankha Shubhra Sarkar

Shankha Shubhra Sarkar

Anindya Sen

Anindya Sen

CERTIFICATE

This is to certify that the thesis entitled “Decentralized Healthcare Network Platform” submitted by Shankha Shubhra Sarkar (18BCE2453) and Anindya Sen (18BCE2382), School of Computer Science and Engineering, VIT University, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by them under my supervision during the period, 03. 01. 2022 to 02.06.2022, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 03.06.2022

Prof. Krishnamoorthy A

Faculty Guide

Internal Examiner **External Examiner**

Head of Department

Department of Software Systems

ACKNOWLEDGEMENTS

Since the project's inception, we have worked tirelessly to bring it to fruition. It would not have been possible without each other's support and understanding, as well as the support and help of our family, professors, and many other individuals and organizations. We would like to express our sincere gratitude to each and every one of them. We are grateful to Dr. Vairamuthu S, Head of Department, Department of Software Systems, and Dr. Ramesh Babu K, Dean of SCOPE and VIT University Management, for allowing us to work at the school of computer science and engineering under the supervision of highly qualified and professional individuals.

We would like to express our heartfelt gratitude to Prof. Krishnamoorthy A for his invaluable and precise assistance throughout the project, which was incredibly beneficial to our team's research and project. We'd also want to express our appreciation to the professors for their valuable input during the project reviews, which allowed us to develop it even further. We would also want to express our deepest gratitude to VIT University for providing us with wonderful four years and the chance to learn new skills and grow further.

Shankha Shubhra Sarkar

Shankha Shubhra Sarkar

Anindya Sen

Anindya Sen

EXECUTIVE SUMMARY

We built a health care network platform that connects all medical institutions. All healthcare services are now focused on hospitals, where most data are still saved manually and controlled in hard copy. The patient must also manually gather services and contact transportations. If the patient is too sick, family or nurses undertake these activities for a fee. This project integrates all Apps- or web-based service automation.

Modern hospitals have internal management systems, but most system administrators lack cybersecurity knowledge. We're adopting the most up-to-date server software with secure over-the-air update capabilities, enabling all hospitals who utilize our services to update themselves in the case of a patch or service upgrade. Hospitals run most ambulances. Emergency response is reliant on patient-hospital communication, which is inefficient. We're developing an autonomous ambulance network to react to SOS calls from neighboring patients. By validating their medical information, users may purchase drugs online. Most pharmacies are independent stores without the digital infrastructure to deliver online services to patients. Using our technology, small to large pharmacists may quickly set up and manage their digital inventory, including online and emergency services to the nearest patient or hospital.

CONTENTS

List of Figures

List of Tables

List of Abbreviations

IHFS	Inter Hospital File System
SOS	Save Our Souls
ISM	Interpretive Structural Modeling
PHR	Personal Health Reports
EHR	Electronic Health Records
GCP	Google Cloud Platform
IPFS	Interplanetary File System
IPNS	Interplanetary Name System
QR	Quick Response code
RAID	Redundant Array of Independent Disks
UML	Unified Modeling Language
VM	Virtual Machine
npm	Node Packet Manager
SUMO	Simulation of Urban Mobility
NS-3	Network Simulator 3
OSM	OpenStreetMap
GPS	Global Positioning System
V2I	Vehicle to Infrastructure Communication

Symbols and Notations

1 INTRODUCTION

In today's digitalized healthcare environment, keeping the best outcomes for patients at the center of all activity increasingly depends on the smart use of medical data.

The exponential growth in health data from a variety of sources, such as electronic medical records and image databases, makes it difficult to integrate information for optimized decision-making that meets the highest possible standards of care. A digital platform that assembles and structures a wide variety of data in a user-friendly format has the potential to remedy this situation. If the platform can provide interoperability between different systems, it becomes even more useful.

In order to maintain the healthcare services with the help of the latest and greatest technologies that can provide the best support and infrastructure for making treatments more accessible and easier, we are creating this decentralized healthcare network platform. This will bring majority of the medical related services to the single platform. Also, this platform will run decentralized utilizing the existing computer infrastructure that the hospitals have. Users will have full control over their data including all the other internal users. Authentications will be done by state-of-the-art blockchains. Users can access all these services through single terminal and automated processes will guide them along the way without any false scam.

1.1 Objective

Primary objective of this project is to make healthcare easily accessible. In order to get there, we need to solve couple of secondary objectives. Such as –

- User can get hospital services, ambulance services, pharmaceuticals services and all other healthcare related stuff through a single platform.
- This platform needs to be accessible through app or website from anywhere of the world.
- Platform needs to be very secure as medical records require utmost privacy for anyone.
- User needs to be able to share or give certain access permission or revoke to any other parties as they please.
- Hospitals need to have their own management services maintained and easily maintainable including their internal users with their own database access and control.

1.2 Motivation

Health data from many sources, such as electronic medical records and imaging databases, is growing exponentially, making it challenging to combine information for optimal decision-making that meets the highest possible standards of care. A user-friendly digital platform that collects and organizes a broad range of data might help to alleviate this problem. The platform's utility increases exponentially if it can facilitate communication across various types of systems. We're building this decentralized healthcare network platform to keep healthcare services up to date with the most cutting-edge technology and to give the finest support and infrastructure to make treatments more accessible and easier.

1.3 Background

1.3.1 Theoretical Background

New technologies, including artificial intelligence, big data analysis, mobile applications and devices, and cloud storage, among others, have the power to significantly contribute to improving any health system. However, for many health service users, the majority of hospitals are far from being able to offer these technological benefits with what they have at present.

- **Patient’s Perspective:**

Currently, patients must gather all communications and services themselves. Patients must manually find a hospital, call an ambulance if need, sign in, claim insurance, be checked out, buy prescription, etc. If the patient is too sick, family members is the hope else nurses do these activities for a fee. This project will automate all of these services using apps or web services.

- **Hospital’s Perspective:**

Currently, all healthcare services are focused in hospitals, where data is manually recorded and handled in hard copy in the worst-case scenario. However, many contemporary hospitals have internal management systems that are frequently hacked due to system managers' lack of cybersecurity knowledge. Over the air or secure channel updates will be implemented to enable all hospitals who utilize our services to update themselves in case of patch or service upgrade.

- **Pharmaceutical’s Perspective:**

Our country's pharmacies are mostly privately operated and heavily regulated. Also, they keep their inventory manually, either by hardcopy or memories them, or by checking availability each time someone wants to purchase a drug, causing confusion and time waste. We will give pharmacists with a simple to manage online service platform. They can conveniently manage their inventory and react to neighboring emergency medical needs.

- **Ambulance Service Providers:**

Meanwhile, hospitals run majority of the ambulance services. And they can only respond to emergencies if patients and hospitals communicate, which is inefficient. To tackle this problem, we will use a completely autonomous ambulance network that can react to any patient's SOS. The software may also link users to nearby ambulances.

- **Data storage and maintenance:**

Existing service providers using a centralized system. In which they control the user data and offer simple backups. But subscribing and maintaining them is rather costly. But we use our own decentralized platform architecture to store, preserve, and retrieve data via current service holders' installations. This will provide comprehensive backups and the latest security offered by blockchain.

1.3.2 Survey of the existing models/works

They examined Blockchain papers in "Blockchain Research in Healthcare: A Bibliometric Review and Current Research Trends" [1]. Studies on academic output included bibliometric analysis of academic institutions, nations, and renowned writers. Blockchain technology may help address many current healthcare issues. In healthcare, blockchain technology may increase data security and decentralization. The tamper-proof recording mechanism of blockchain makes it perfect for secure healthcare data systems that safeguard patient privacy.

"Research on the Application of Blockchain in Smart Healthcare: Constructing a Hierarchical Framework" [2] proposes a decision-making and experimental evaluation laboratory (DEMATEL). ISM is used to partition the hierarchy and construct a hierarchical theoretical framework. As a consequence of the system's design, medical record management, and doctor management. A growing structure of internal and external regulation, medical insurance, and environmental governance protects stakeholders'

interests.

They designed a blockchain-based privacy-preserving and sharing system that uses access control mechanisms and encryption to safeguard users' personal data in "MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain" [3]. No patient data is uploaded to a trustworthy third party. An efficient approach is aided by the ledger's bread crumb pattern.

A blockchain-based architecture for protecting electronic health record systems [4] is proposed. The idea is to use blockchain technology to preserve data records and improve system compatibility. The authors suggested a new blockchain incentive system. It may thus be used for various electronic record systems requiring data protection. There is a Java app and Hyperledger Fabric. It shows their smart contracts and incentive scheme.

The researchers presented the current state-of-the-art in non-financial blockchain applications like healthcare based on study on "Blockchain technology, improvement recommendations, security problems on smart grid and its use in healthcare for sustainable development" [5]. According to them, HDG, a secure mobile app paradigm for digitizing medical data, also had an influence. However, issues like a 51% attack need alternate working techniques like Proof-of-Stake.

For secure and transparent healthcare data exchange, the authors of "Securing and authenticating healthcare records with blockchain technology" [6] proposed a distributed blockchain system. These variables are used to store patient data. A logged-in user can only obtain reports. The data is safeguarded since it is copied across several nodes. Md5 was used instead of SHA-1 or other more efficient algorithms.

"Chatbot for Healthcare System Using Artificial Intelligence" [7] is meant to provide quality replies promptly. It relieves the answer supplier by providing the solution directly to the user. The user may save time by calling doctors or experts. The authors extracted the keyword from the user query using N-gram and TF-IDF. Weigh each phrase to achieve the best answer.

"Medbot: Conversational Artificial Intelligence Powered Chatbot for Delivering Tele-Health after COVID-19" [8] describes a serverless application that gives preventive measures and symptoms for the most frequent diseases in rural India. Telehealth in India may be given using a Google Cloud Platform conversational bot dubbed “Aapka Chikitsak” (GCP).

The study "Decentralized safe storage of medical records utilizing Blockchain and IPFS: A comparative analysis with future perspectives" [9] compared the two technologies.

In an emergency, how may non-Consortium Blockchain parties access data? No insider threats or access control. HTTP and DNS must be changed for true decentralization. IPFS and IPNS (InterPlanetary Name System) may revolutionize the paradigm.

The authors of "Sharing COVID-19 patient data securely and distributed Using IPFS and Consortium Blockchain" [10] offered a distributed on- and off-chain storage solution to protect sensitive patient data such as identification data. It just saves hashes of patients' clinical records to save space. Authorized peers get fair services without a third-party mediator. The authors claim that the concept may be extended with more peers and gigabytes of report sharing.

A blockchain architecture for secure and easy exchange of patient Personal Health Reports (PHR) between health organization actors is discussed in the study "Inter-Planetary File System Enabled Blockchain Solution for Securing Healthcare Records" [11]. It also leverages IPFS for faster PHR retrieval. Their technology employs blockchain-based compute nodes to store client-centric patient health data. JMeter analyzes a participant's data read/write access and database retrieval. As shown in the simulation results, the system can analyze large data sets quickly and efficiently.

The hyper ledger fabric technology proposed in "Securing Pharmaceutical Supply Chain using Blockchain Technology" [12] ensures data interchange, storage, transparency, and traceability across the supply chain. Smart contracts governed sender-receiver interactions on Ethereum. Using blockchain for tracking and monitoring prevents counterfeit pharmaceuticals from entering the supply chain and reaching consumers. To generate a QR code for each product, an ID was assigned. An affordable and workable solution was identified. Consumer feedback helped assess and rank supply chain participants.

1.3.3 Gaps identified in the Survey

- **Centralized System:**

  Almost all of the existing services that trying to solve the similar issues are centralized. This causes large infrastructure, costly, inefficient system.

- **Vulnerable to Cyber Attack:**

  As majority of the hospitals doesn’t have cybersecurity specialists, they prone to attain a single time installation system. This leaves old systems open to exposed vulnerabilities.

- **Single Point of Failure:**

  Storing all data in a single server with traditional onsite backups are dangerously vulnerable to single point of failure. Any sort of damage can cause permanent data loss or data leak.

- **Privacy Concern:**

  Current centralized system doesn't ensure privacy as they own the user data and experiment as they want.

- **Compatibility with Existing or Multiple Systems:**

  Existing systems tends to make a single type of service which may not be compatible with existing infrastructure or not even utilize the existing systems properly.

2 PROJECT DESCRIPTION AND GOALS

The suggested system as a whole is modeled using our IHFS network. The next paragraphs detail the overall system architecture, UML sequence, and implementation methodologies.

2.1 Primary Targets

- **Decentralized System:**

  Data will be stored in a decentralized our very own IHFS with will completely use the existing systems without requiring installation of any new large infrastructures.

- **Secure Storage:**

  The whole network will be secured by the state-of-the-art blockchain based security. This will prevent any kind of data integrity related issues.

- **Automated Software Management:**

  Hospitals along with other nodes will get over the air software updates and management services like OS.

- **Secure Automated Backups:**

  IHFS will provide regular RAID backups as well as offsite backups to prevent any permanent data loss.

- **Owner with full control of the data:**

  Owners of the medical records will have full control over the access, modification or monitoring.

- **Utilizing Existing Services Compatibility:**

  As this service will include all services under single roof, whole healthcare departments will be in perfect sync.

2.2 Architecture of the Proposed System

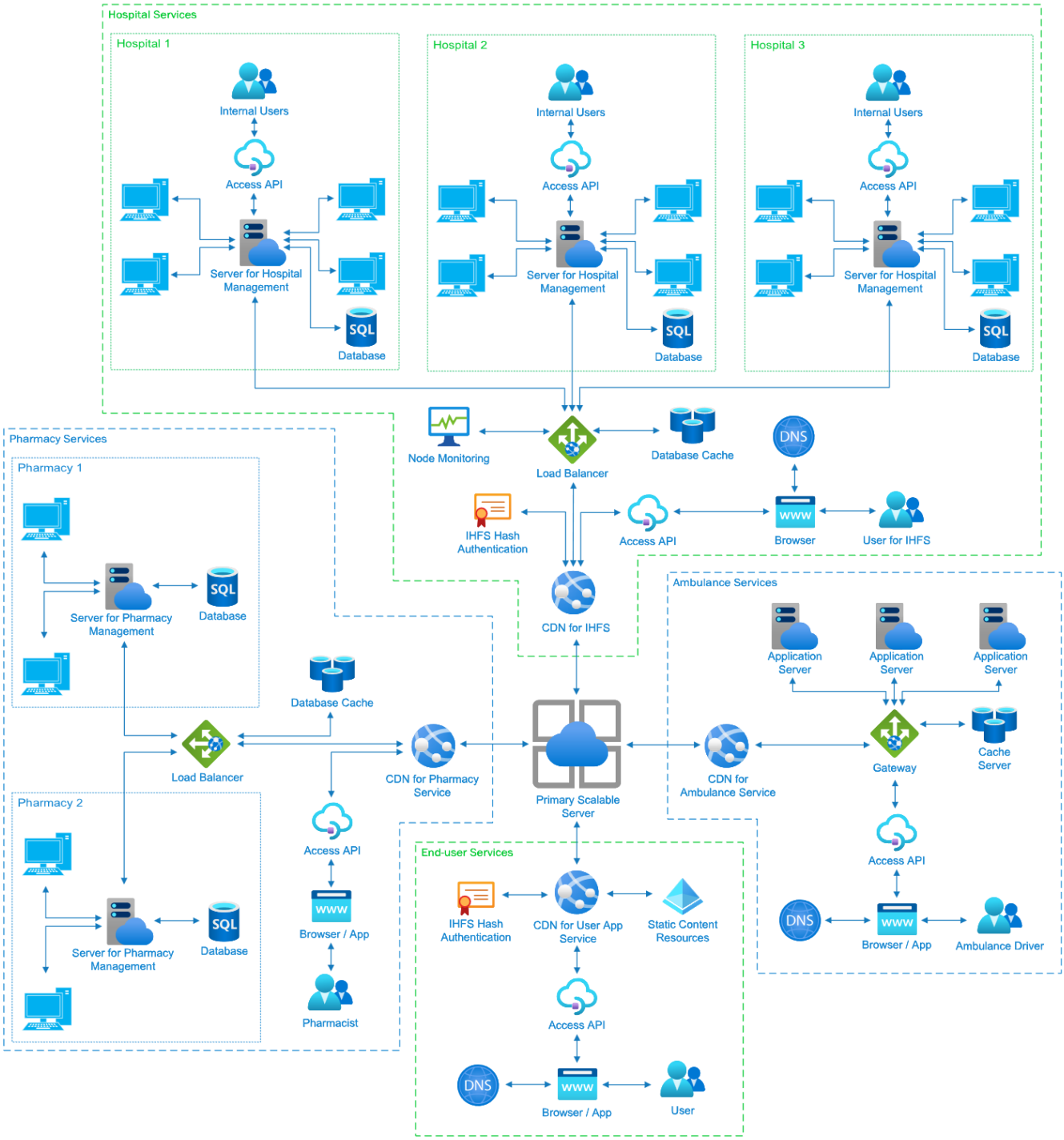

Figure 1 Architecture of the Decentralized Healthcare Network Platform

2.3 System Architecture Analysis

- **End user services:**

All of the medical services that a patient requires are integrated into the overall infrastructure. Declaring an emergency, looking for a hospital, calling an ambulance, purchasing medicine, paying costs, maintaining medical records, and so on. Currently, the patient must collect all communications and services by themselves manually. Patients must manually locate a hospital, call an ambulance, if necessary, admit themselves, claim insurance, get checked out, purchase medicine, and so on. If the patient is too ill, these tasks are performed by relatives or nurses at an additional cost. We will integrate all of these services' automation based on Apps or web services in this project. Patients have to keep the medial record with them safely and present to doctors every time. In order to sort this problem, we will be using decentralized storage system for easy and secure access.

- **Hospital Internal Network:**

A complete hospital management system will be included in this module. Currently, all healthcare services are concentrated in hospitals, where the bulk of data is still stored manually, and data is managed in hard copy in the worst-case scenario. On the other hand, many modern hospitals have internal management systems, but they are frequently vulnerable to hacking because most system administrators lack cybersecurity expertise. To address this issue, we will implement the most up-to-date server applications with the most secure channel or over-the-air update capability, allowing all hospitals who use our services to update themselves in the event of a patch or service upgrade.

- **Pharmaceutical Services:**

This module will create a separate network between all the pharmacy and their stocks. They can provide online shopping service as well as hospital supplies. In general scenario most of the pharmacy in our country are privately owned and there's much of a regulation in local shops. Also, they maintain their inventory in traditional way, either using hardcopies or memories or in worst case they check availability each time anyone wants to buy some medicine which creates confusions and time waste. We will provide pharmacies easy to host site framework that they can manage their own shop very easily including online service providing. They can keep track of their inventory easily including responding to nearby emergency medical products requirements.

- **Ambulance Services:**

Registered ambulances will have own network & will be able to answer on patient's emergency declarations. Meanwhile, many ambulance services are maintained by hospitals. And they can only response to emergency based on communication done between patients and hospital which is way to inefficient. In order to solve this issue, we will be using a complete independent ambulance service network that can respond to emergency SOS from any patients nearby. Users can use the app to connect with the nearest ambulances too.

- **Data storage and maintenance:**

Our state-of-the-art decentralized file management system will be implemented in this module. Existing service providers trying to solve similar problems have centralized system. Where they own the user data and provide traditional encryption security along with simple backup plan. But also, those are quite expensive to subscribe and maintain. But we are working with our own decentralized platform design to store, maintain and access data using existing installations of the service holders. This will arrange full backups along with state-of-the-art security provided by the blockchain.

3 TECHNICAL SPECIFICATION

3.1 Specification Analysis

Requirement analysis is concerned with determining the requirements or conditions necessary to meet the needs of the new or altered product or project.

3.1.1 Functional Specifications

User tasks need the implementation of product features or services. It is essential that the stakeholders be aware of the project's goals and objectives.

- **Product Perspective**

Distributed servers, Fast network connectivity, Proper Status Monitoring of the system.

- **Product features**

Fast, Convenient, and Low maintenance Distributed Database, Easy but Secure Access.

- **User characteristics**

Little to no training for Users familiar with Smart Phones or Internet Browsers. App will provide hands on tutorial on first launch with optional repetition.

- **Assumption & Dependencies**

As it is a platform-independent distributed system, just one terminal is required to access the service. Hospitals will need some setup, but users will have plug and play experience.

- **Domain Specification**

This project development in accordance with openEHR (Open Electronic Health Record)

- **User Specification**

As mostly users will be patients, a fully customizable UI with human machine interface for disabled people such as coded sound/vibration signals with proper SOS service.

3.1.2 Non-Functional Specifications

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

3.1.2.1 Product Specifications

A product specifications document defines the product we are building.

- ***Efficiency***

The system provides acknowledgment in just one second once the 'patient's information is checked. The system needs to support at least 1000 people at once. The user interface acknowledges within five seconds. The system offers the efficiency for data backup.

- ***Reliability***

As the system is completely distributed through blockchain, it’s available all the time.

- ***Portability***

A single handheld terminal such as smartphone is enough to operate all the emergency services along with the transactions and monitoring.

- ***Usability***

  Any modifications like insert, delete, update, etc. for the record can be synchronized quickly and executed only by the proper authenticated user. Every kind of users from separate modules have their own interconnected portals for better usability.

3.1.2.2 Organizational Specifications

- **Implementation Specifications (in terms of deployment)**

  Primary scalable server on the cloud needs to be deployed for the start of the service and monitoring. Once enough hospitals are subscribed for the IHFS, the system will become self-sustained and all the databased will be maintained by the distributed file system. Hospitals will require to setup an intranet among the computers situated within & rest of the service holders will require to setup apps in their computer or phones for connectivity to the service.

- **Engineering Standard Specifications**

  We will use ISO 18308:2011 (Health informatics - Requirements for an electronic health record architecture) for this project. ISO 18308:2011 outlines the standards for an EHR architecture, a system that processes, maintains, and transmits EHR data. These EHRs must be clinically valid and trustworthy, ethically sound, satisfy current legal requirements, promote excellent clinical practice, and permit data analysis for a variety of objectives.

3.1.2.3 Operational Requirements

- ***Economic:***

  Our solution is economically viable since it does not need a centralized server, which eliminates the enormous expenses associated with their maintenance.

- ***Environmental:***

  Due to the decentralized nature of our solution and the fact that it is only maintained online, it eliminates the massive paper consumption associated with medical file recording systems. No additional electricity is required to maintain any centralized server. As a result, our product has little to no effect on the environment and will instead seek to improve our environment.

- ***Social:***

  Users will be able to communicate with their physicians and other healthcare providers. Customers will be just a few buttons clicks away from enjoying the service, eliminating the inconvenience of scheduling appointments and interfering with social life.

- ***Political:***

  Our product will adhere to all the government's policies in the jurisdiction in which we operate. Any civilian can observe our operations and cast doubt on our authority.

- ***Ethical:***

  Since all data will be kept in a decentralized system, no one, including us, will have access to the users' personal or sensitive information.

- ***Health and Safety:***

  Our solution is aimed at enhancing the healthcare system. Without a doubt, it satisfies the need.

- ***Sustainability:***

  To begin, we will use Ether to execute transactions, since it is a more sustainable cryptocurrency than Bitcoin. In the future, we want to introduce our own cryptocurrency that will include all of the sustainability objectives.

- ***Legality:***

  We do not collect, store, or exchange any user data. All of our efforts will be completely open source and lawful.

- ***Inspectability:***

  Our initiative is entirely open source. Anybody can inspect our project or algorithm and utilize them for the greater good of society.

3.2 System Specifications

3.2.1 Hardware Specifications

End User H/W

Any kind of internet connected PC or Mobile with Browser or IHFS App.

Subscribed Hospitals H/W

IHFS connected Central Server system, Individual desk computers and Stuff computers in the hospital connected to intranet of the central server.

Subscribed Pharmaceuticals

IHFS connected PC for small shop, IHFS connected central terminal with intranet among the shop's other PCs for larger pharmacy shops.

Ambulance Drivers H/W

Internet connected Ambulance Service Providing App installed Smart Phone.

3.2.2 Software Specifications

Programming Languages:

JavaScript, Python, Java, Solidity, HTML, CSS

Technologies:

REST API, Django, Microsoft Azure Cloud Engine, Ethereum VM, IPFS

Software & IDE:

VS Code, Cloud SSH, Remix, IntelliJ IDEA

4 DESIGN APPROACH AND DETAILS

All of a patient's required medical services are incorporated into the overall infrastructure. Declaring an emergency, locating a hospital, summoning an ambulance, procuring medication, paying associated charges, and preserving medical records, among other things. This module will comprise a comprehensive hospital management system. This module will establish a distinct network between each pharmacy and its stock. They may give both internet shopping and healthcare supplies. Ambulances that are registered will have their own network and will be able to respond to patient emergency declarations. This module will implement our cutting-edge decentralized file management system.

4.1 Design Approach

4.1.1 Proposed System Model: UML Sequence Diagram

On this UML sequence diagram, it's explained how an admin system is configured on the network by admin. Admin represents a whole internal hospital management system as a service holder. Upon successful configuration users from that hospital can upload their documents and healthcare related documents to the IHFS and automatically secured by blockchain. Out centralized monitor service will keep an eye on the process execution along with admin to check how many users they have under them and their whole database

maintenance along with backups and all other provided data and transaction monitoring services.

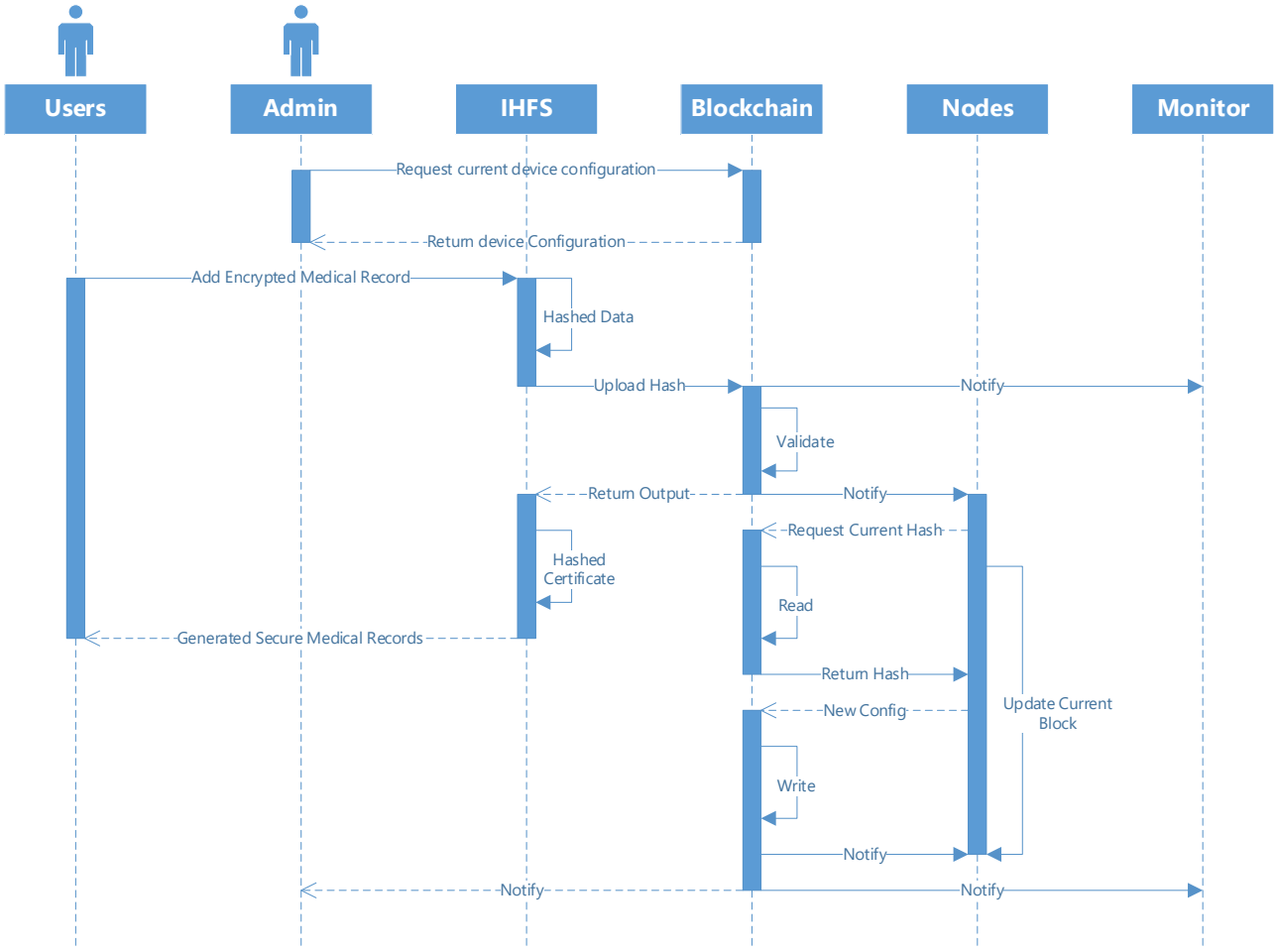

Figure 2 UML Sequence of User and Admin interaction with IHFS and Blockchain

4.2 Codes and Standards

A code is a set of rules and specifications for the correct methods and materials used in a certain product, building or process. Codes can be approved by proper administration. The main purpose of codes is to protect the public by setting up the minimum acceptable level of safety for products and processes. For this project we have used ISO 18308:2011 (Health informatics - Requirements for an electronic health record architecture) for this project. ISO 18308:2011 outlines the standards for an EHR architecture, a system that processes, maintains, and transmits EHR data. These EHRs must be clinically valid and trustworthy, ethically sound, satisfy current legal requirements, promote excellent clinical practice, and permit data analysis for a variety of objectives.

As per the implementation, we have maintained a well-defined and standard style of

coding. Some of these standards are Limited use of global variables, Standard headers for different modules, Naming conventions for local variables, global variables, constants and functions, Proper indentation, Code should be well documented, Length of functions should not be very large, Error return values and exception handling conventions.

We have used Django to build our complete web app and it's API structure. So, we have followed official Django coding standards. These standards are -

Python Style:

- All files are formatted using the black auto-formatter.
- String variable interpolation used by f-strings, or str.format(), with the goal of maximizing code readability.
- Underscores, not camelCase, is used for variable, function and method names.
- InitialCaps is used for class names.

```
1.  from django import forms
2.  from django.contrib.auth.models import User
3.  from . import models
4.
5.  #for admin signup
6.  class AdminSigupForm(forms.ModelForm):
7.      class Meta:
8.          model=User
9.          fields=['first_name','last_name','username','password']
10.         widgets = {
11.         'password': forms.PasswordInput()
12.         }
13.
14. #for doctor related form
15. class DoctorUserForm(forms.ModelForm):
16.     class Meta:
17.         model=User
18.         fields=['first_name','last_name','username','password']
19.         widgets = {
20.         'password': forms.PasswordInput()
21.         }
22. class DoctorForm(forms.ModelForm):
23.     class Meta:
24.         model=models.Doctor
25.         fields=['address','mobile','department','status','profile_pic']
26.
```

Template Style:

- In Django template code, one (and only one) space is given between the curly brackets and the tag contents.

```
1.  <!DOCTYPE html>
2.  <html lang="en" dir="ltr">
3.    <head>
4.      <meta charset="utf-8" />
```

```
    <title>Decentralized Healthcare Network Platform</title>
  </head>

  <body>
    {% include "hospital/navbar.html" %}
    {% block content %}

    {% endblock content %}
    {% include "hospital/footer.html" %}
  </body>
</html>

```

View Style:

- In Django views, the first parameter in a view function is called request.

```
from django.shortcuts import render,redirect,reverse
from . import forms,models
from django.db.models import Sum
from django.contrib.auth.models import Group
from django.http import HttpResponseRedirect
from django.core.mail import send_mail
from django.contrib.auth.decorators import login_required,user_passes_test
from datetime import datetime,timedelta,date
from django.conf import settings

# Create your views here.
def home_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'hospital/index.html')

#for showing signup/login button for admin(by sumit)
def adminclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'hospital/adminclick.html')

#for showing signup/login button for doctor(by sumit)
def doctorclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'hospital/doctorclick.html')

#for showing signup/login button for patient(by sumit)
def patientclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'hospital/patientclick.html')

```

Model Style:

- Field names are all lowercase, using underscores instead of camelCase.
- The class Meta is added after the fields are defined, with a single blank line separating the fields and the class definition.
- If choices are defined for a given model field, each choice is defined as a list of tuples, with an all-uppercase name as a class attribute on the model.

```
1.  from django.db import models
2.  from django.utils import timezone
3.  from django.contrib.auth.models import AbstractUser
4.  from django.db import models
5.  from django.db.models.signals import post_save
6.  from django.dispatch import receiver
7.
8.  from django.db.models import BooleanField, ExpressionWrapper, Q
9.  from django.db.models.functions import Now
10.
11.
12. class Patients(models.Model):
13.     gender_category=(
14.         ('Male','Male'),
15.         ('Female','Female'),
16.     )
17.     admin = models.OneToOneField(CustomUser,null=True, on_delete = models.CASCADE)
18.     reg_no=models.CharField(max_length=30,null=True,blank=True,unique=True)
19.     gender=models.CharField(max_length=7, null=True, blank=True,
    choices=gender_category)
20.     first_name=models.CharField(max_length=20,null=True,blank=True)
21.     last_name=models.CharField(max_length=20,null=True,blank=True)
22.     dob=models.DateTimeField(auto_now_add= False, auto_now=False,null=True,
    blank=True)
23.     phone_number=models.CharField(max_length=10,null=True,blank=True)
24.     profile_pic=models.ImageField(default="patient.jpg",null=True,blank=True)
25.     age= models.IntegerField(default='0', blank=True, null=True)
26.     address=models.CharField(max_length=300,null=True,blank=True)
27.     date_admitted=models.DateTimeField(auto_now_add=True, auto_now=False)
28.     last_updated = models.DateTimeField(auto_now_add=False, auto_now=True)
29.
30.     def __str__(self):
31.         return str(self.admin)
32.
33.
34. class AdminHOD(models.Model):
35.     gender_category=(
36.         ('Male','Male'),
37.         ('Female','Female'),
38.     )
39.     admin = models.OneToOneField(CustomUser,null=True, on_delete = models.CASCADE)
40.     emp_no= models.CharField(max_length=100,null=True,blank=True)
41.     gender=models.CharField(max_length=100,null=True,choices=gender_category)
42.     mobile=models.CharField(max_length=10,null=True,blank=True)
43.     address=models.CharField(max_length=300,null=True,blank=True)
44.     profile_pic=models.ImageField(default="admin.png",null=True,blank=True)
45.     created_at = models.DateTimeField(auto_now_add=True)
46.     updated_at = models.DateTimeField(auto_now=True)
47.     date_employed=models.DateTimeField(auto_now_add=True, auto_now=False)
48.     objects = models.Manager()
49.     def __str__(self):
50.         return str(self.admin)
51.
52.
53. class Pharmacist(models.Model):
54.     gender_category=(
55.         ('Male','Male'),
56.         ('Female','Female'),
57.     )
58.     admin = models.OneToOneField(CustomUser,null=True, on_delete = models.CASCADE)
59.     emp_no=models.CharField(max_length=100,null=True,blank=True)
60.     age= models.IntegerField(default='0', blank=True, null=True)
61.     gender=models.CharField(max_length=100,null=True,choices=gender_category)
62.     mobile =models.CharField(max_length=10,null=True,blank=True)
63.     address=models.CharField(max_length=300,null=True,blank=True)
```

```
    profile_pic=models.ImageField(default="images2.png",null=True,blank=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
    def __str__(self):
        return str(self.admin)

```

```
from django import forms
from django.contrib.auth.models import User
from . import models

#for admin signup
class AdminSigupForm(forms.ModelForm):
    class Meta:
        model=User
        fields=['first_name','last_name','username','password']
        widgets = {
        'password': forms.PasswordInput()
        }

#for doctor related form
class DoctorUserForm(forms.ModelForm):
    class Meta:
        model=User
        fields=['first_name','last_name','username','password']
        widgets = {
        'password': forms.PasswordInput()
        }
class DoctorForm(forms.ModelForm):
    class Meta:
        model=models.Doctor
        fields=['address','mobile','department','status','profile_pic']

```

4.3 Constraints, Alternatives and Tradeoffs

4.3.1 Constraints of the product

- Internet Connection Dependency

  One of the main constraints of this product is complete dependency on the internet. Any kind of services access and modification require all the parties involved to have an internet connection. But while using IHFS, users situated in that particular hospital area they have registered, connected to hospital's internal network such as hospital provided Wi-Fi, will have access to the services even without the outside connectivity. But if the user is trying to access the services from outside the perimeter, then they will require at least 1G level internet connectivity for text/document based services and 2G level for image based services at the minimum.

- Hardware Requirement for patients

  Users will be required to have some sort of digital devices that can connect to internet and browse web apps. Smartphones, laptops, PCs along with other Smart

Devices fall in this category. But again, without any of these devices' users can't access the services outside of the hospital perimeter, but they can still get their work done by the hospital and pharmacy reception.

- Hardware Requirement for hospitals

One of the main constraints of the IHFS including the whole Platform is that subscribed hospitals need to have hardware that are interconnected to the network which will eventually create the IHFS. This hardware needs to be good quality with decent amount of storage and as fast internet speed connection possible.

- User Training for patients

Although the platform is very easy and user friendly, users still do need to go through the process of learning the system and work flow and make themselves comfortable with the digital process replacing the old paper based system.

- User Training for Hospitals

This is a very important part as the whole decentralized system can work properly, if each node works perfectly. That is why doctors and receptionists who will be directly working with the services need to have proper training so that they can teach the patients as well.

- Decentralized Nodes Maintenances

As the whole system depends on the hospitals hardware and their performances, regular maintenance and their monitoring is required. Although the IHFS will have its own server management service running 24/7 for possible issues and warning, it is a constraint to have it monitored for safekeeping.

4.3.2 Alternatives of the approach

- Centralized System

Our product is completely unique on this approach, so there isn't any alternative decentralized service or similar product on the market. But, among existing approaches centralized system is there in the competition. Majority of the healthcare sector that turned to digital are actually using centralized system. But it lacks our state of the art blockchain based security and reliability of the IHFS.

- Old Paper based Approach

The old paper based completely analog or semi digital approach, where documents are made in computer but printed copy is used for the services, is also an alternative

approach. But they lack all the real time and emergency connectivity services provided by fully web based services.

4.3.3 Tradeoffs

- Internet Dependence

  One of the main of few tradeoffs is internet dependency. As without internet connectivity no online based service can work. But considering the other option is to travel to the hospital or pharmacy every single time it's pretty good.

- Paper based Medical Records

  Last tradeoff is to leave the paper based medical records or prescriptions for the digital ones. This could feel uninviting for some old generation users but also tempting for new generations. But our service has option to have the records printed and get the facilities as current semi-digital approach systems provide.

5 SCHEDULE, TASKS AND MILESTONES

The project has been completed over a period of four months in phases.

Phase 1:

- We concentrated on research throughout this stage. We reviewed research articles to have a better understanding of the product we intended to create.
- The topic was selected and approved.
- For the project, a literature review was conducted.
- To close the gap identified in our survey, we attempted to create a file system and an effective emergency vehicle management system.

Phase 2:

- The design of our modules is completed.
- To better comprehend the project, we developed diagrams and frameworks.
- We determined on the project's technical specifications as well as the tools we'll employ to construct it.

Phase 3:

- We constructed the server engine and backend for the emergency ambulance system android app.
- For our review 2, we presented the app and initial integration of the IHFS.

Phase 4:

- For prospective publication in SN Computer Science, we created a research article titled "Route Planning Service for Emergency Vehicles with Increased Accuracy and Efficiency for Online Platforms."
- Using the feedback from review 2, we make minor revisions and concentrate on finishing it.
- We finished the final construction of the project, as well as the thesis documentation.

Table 1 Schedule, tasks, and milestones for the completion of the project

	Task Mode	Task Name	Work	Duration	Start	Finish	Predecessors	Resource Names
1		Start	0 hrs	0 days	06 January 2022 8:00 AM	06 January 2022 8:00 AM		
2		**Requirement Analysis**	**72 hrs**	**4.75 days**	**06 January 2022 8:00 AM**	**12 January 2022 3:00 PM**		
3		Market Research	16 hrs	1 day	06 January 2022 8:00 AM	06 January 2022 5:00 PM	1	Anindya Sen,Shankha Shubhra Sarkar
4		Paper Research	48 hrs	3 days	07 January 2022 8:00 AM	11 January 2022 5:00 PM	3	Anindya Sen,Shankha Shubhra Sarkar
5		Brain Storming	4 hrs	0.25 days	12 January 2022 8:00 AM	12 January 2022 10:00 AM	4	Anindya Sen,Shankha Shubhra Sarkar
6		Feature List	4 hrs	0.5 days	12 January 2022 10:00 AM	12 January 2022 3:00 PM	3,5	Shankha Shubhra Sarkar
7		Requirement Complete	0 hrs	0 days	12 January 2022 3:00 PM	12 January 2022 3:00 PM	3,4,5,6	
8		**Project Process Diagram**	**4 hrs**	**0.5 days**	**12 January 2022 3:00 PM**	**13 January 2022 10:00 AM**		
9		Feature Prioritaization	1 hr	0.13 days	12 January 2022 3:00 PM	12 January 2022 4:00 PM	7	Anindya Sen
10		Architecture	3 hrs	0.38 days	12 January 2022 4:00 PM	13 January 2022 10:00 AM	9	Shankha Shubhra Sarkar
11		Diagram Complete	0 hrs	0 days	13 January 2022 10:00 AM	13 January 2022 10:00 AM	9,10	
12		**Design**	**40 hrs**	**4.5 days**	**13 January 2022 10:00 AM**	**19 January 2022 3:00 PM**		
13		Interface Desgin	24 hrs	3 days	13 January 2022 10:00 AM	18 January 2022 10:00 AM	11	Shankha Shubhra Sarkar
14		Software Design	8 hrs	1 day	18 January 2022 10:00 AM	19 January 2022 10:00 AM	13	Anindya Sen
15		Design Specification	8 hrs	0.5 days	19 January 2022 10:00 AM	19 January 2022 3:00 PM	14	Anindya Sen,Shankha Shubhra Sarkar
16		Design Complete	0 hrs	0 days	19 January 2022 3:00 PM	19 January 2022 3:00 PM	13,14,15	
17		**Development**	**536 hrs**	**37 days**	**19 January 2022 3:00 PM**	**11 March 2022 3:00 PM**		
18		Develop Frontend	80 hrs	10 days	19 January 2022 3:00 PM	02 February 2022 3:00 PM	16	Anindya Sen
19		Develop Server Engine	80 hrs	10 days	19 January 2022 3:00 PM	02 February 2022 3:00 PM	16	Shankha Shubhra Sarkar
20		Develop Backend	320 hrs	20 days	02 February 2022 3:00 PM	02 March 2022 3:00 PM	16,19	Anindya Sen,Shankha Shubhra Sarkar
21		Integrate System Modules	32 hrs	4 days	02 March 2022 3:00 PM	08 March 2022 3:00 PM	20	Shankha Shubhra Sarkar
22		Perform Initial Testing	24 hrs	3 days	08 March 2022 3:00 PM	11 March 2022 3:00 PM	21	Anindya Sen
23		Development Complete	0 hrs	0 days	11 March 2022 3:00 PM	11 March 2022 3:00 PM	18,19,21,22,20	
24		**Testing**	**120 hrs**	**9 days**	**11 March 2022 3:00 PM**	**24 March 2022 3:00 PM**		
25		Perform System Testing	16 hrs	2 days	11 March 2022 3:00 PM	15 March 2022 3:00 PM	23	Anindya Sen
26		Document Issues Found	8 hrs	1 day	15 March 2022 3:00 PM	16 March 2022 3:00 PM	25	Shankha Shubhra Sarkar
27		Correct Issues Found	96 hrs	6 days	16 March 2022 3:00 PM	24 March 2022 3:00 PM	26	Anindya Sen,Shankha Shubhra Sarkar
28		Testing Complete	0 hrs	0 days	24 March 2022 3:00 PM	24 March 2022 3:00 PM	25,26,27	
29		**Deployment**	**58 hrs**	**4.25 days**	**24 March 2022 3:00 PM**	**30 March 2022 5:00 PM**		
30		Manage Resources	8 hrs	1 day	24 March 2022 3:00 PM	25 March 2022 3:00 PM	28	Shankha Shubhra Sarkar
31		Onsite Installation	24 hrs	3 days	25 March 2022 3:00 PM	30 March 2022 3:00 PM	30	Anindya Sen
32		Setup CDN	24 hrs	3 days	25 March 2022 3:00 PM	30 March 2022 3:00 PM	30	Shankha Shubhra Sarkar
33		Import Live Data	2 hrs	0.25 days	30 March 2022 3:00 PM	30 March 2022 5:00 PM	31,32	Anindya Sen
34		Deployment Complete	0 hrs	0 days	30 March 2022 5:00 PM	30 March 2022 5:00 PM	30,31,32,33	
35		Project Outcome	128 hrs	8 days	31 March 2022 8:00 AM	11 April 2022 5:00 PM	34	Anindya Sen,Shankha Shubhra Sarkar
36		End	0 hrs	0 days	11 April 2022 5:00 PM	11 April 2022 5:00 PM	35	

6 PROJECT DEMONSTRATION

We have used python as our primary language and that is why we have chosen Django as our primary web server framework. For front-end we have used Django Templates along with npm libraries for JS integration on the rendered pages. We have used default Django provided sqlite3 database as it is easy and convenient for sharing over IHFS. User can have their medical records locally on their node or upload to blockchain for permanent storage.

Primary Hospital Website:

Figure 3 Hospital Primary Website Homepage

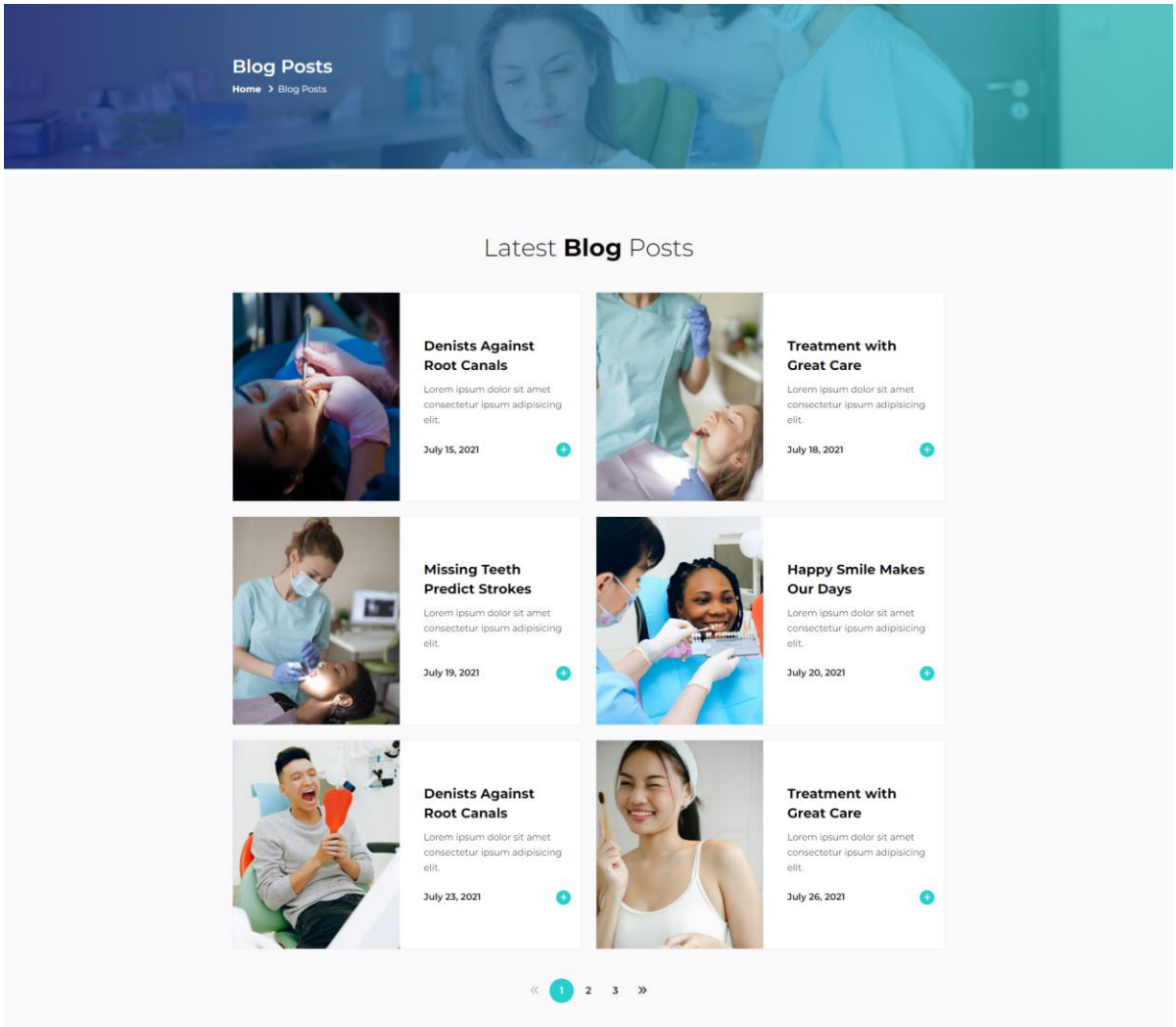

Figure 4 Demo Blogs Page for Primary Website

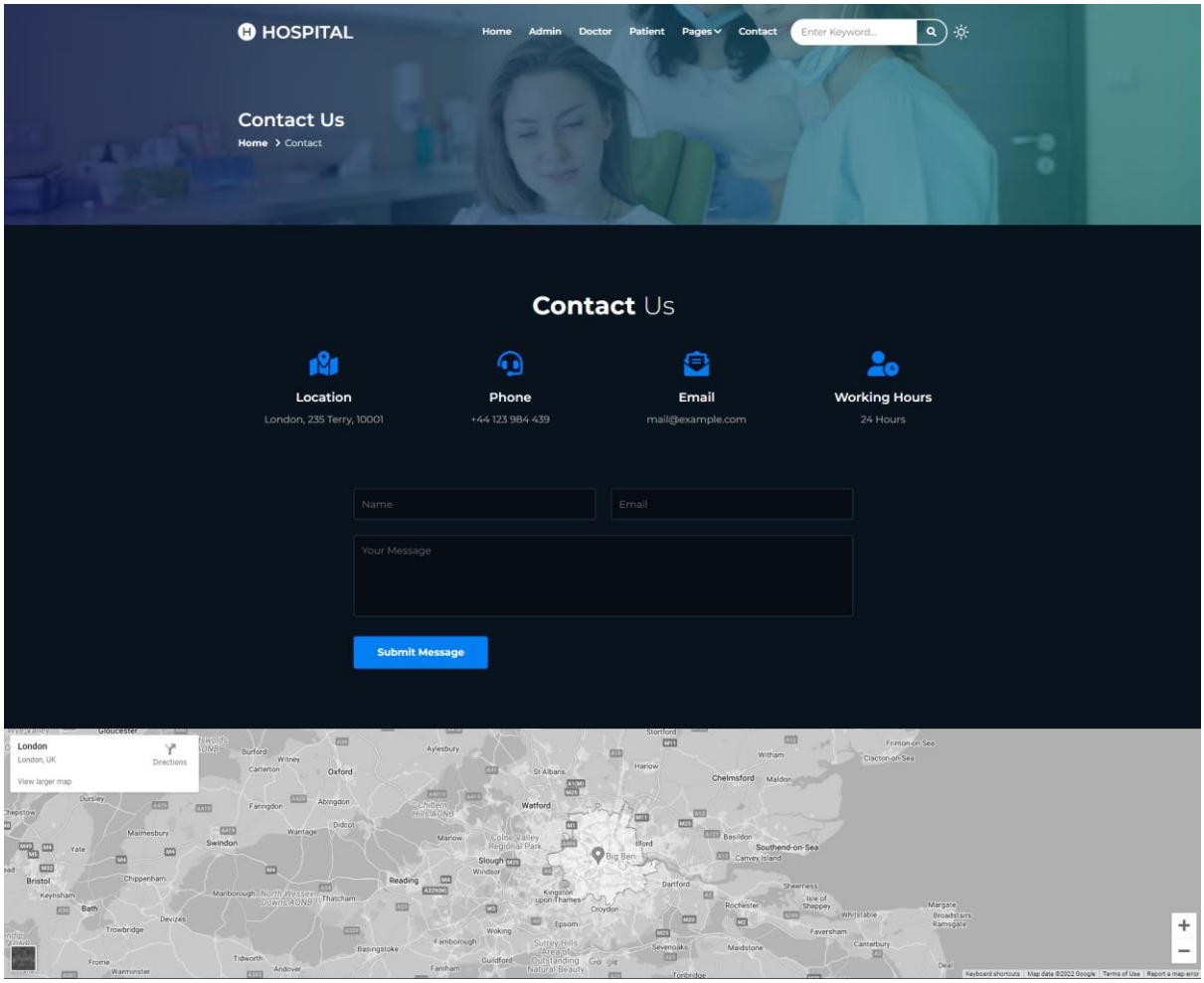

Figure 5 Contact Us Page with Dark Mode Activated

Patient Module:

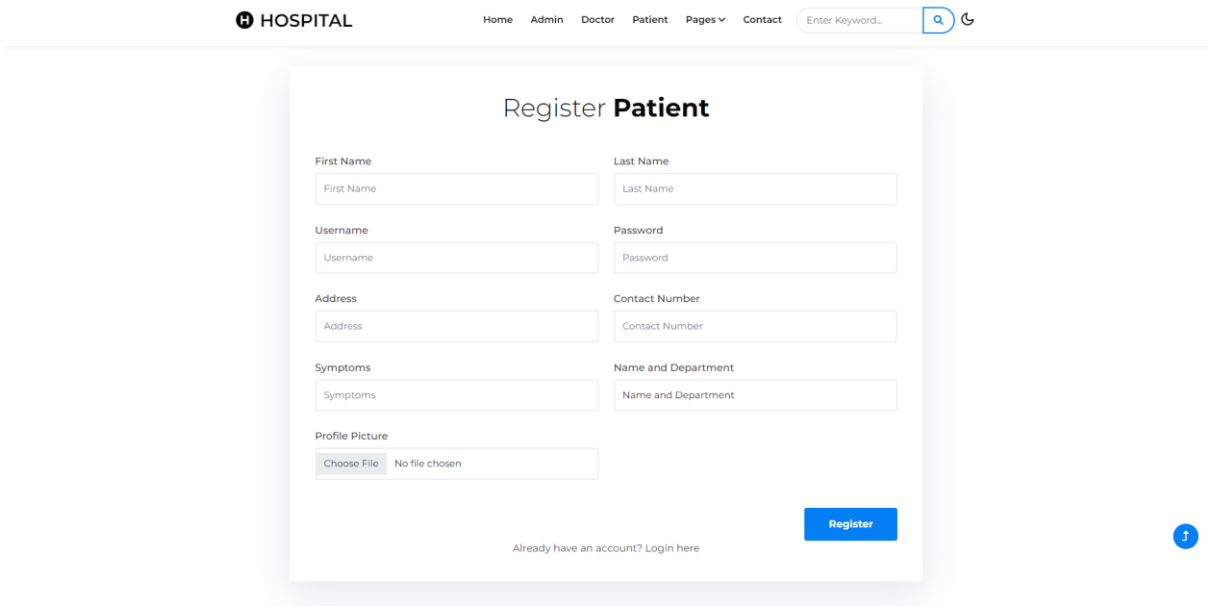

Figure 6 Patient Registration Form

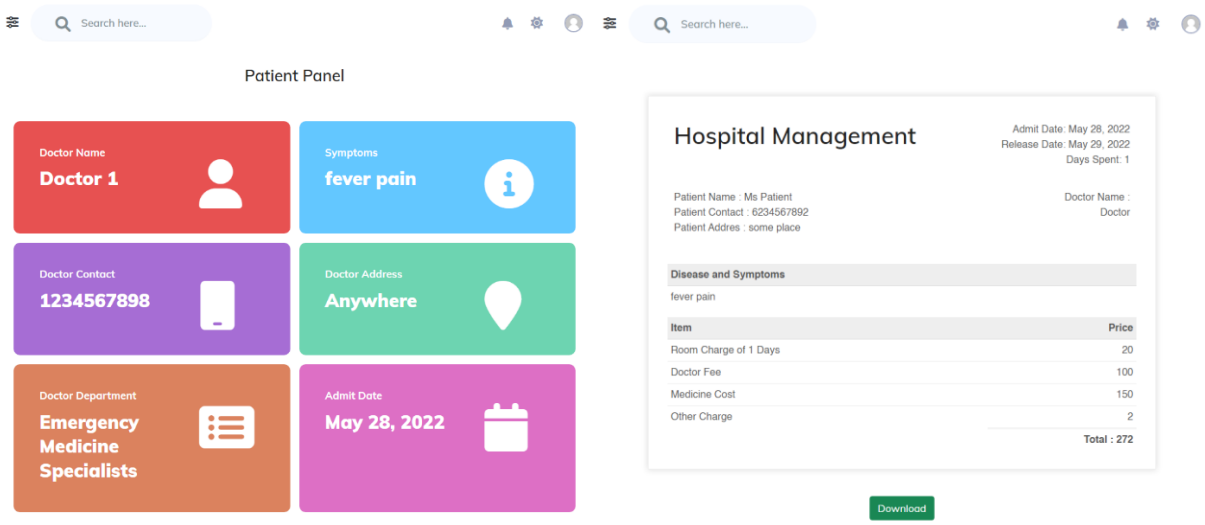

Figure 7 Patient Dashboard Cards (left) and Patient Discharge Receipt (right)

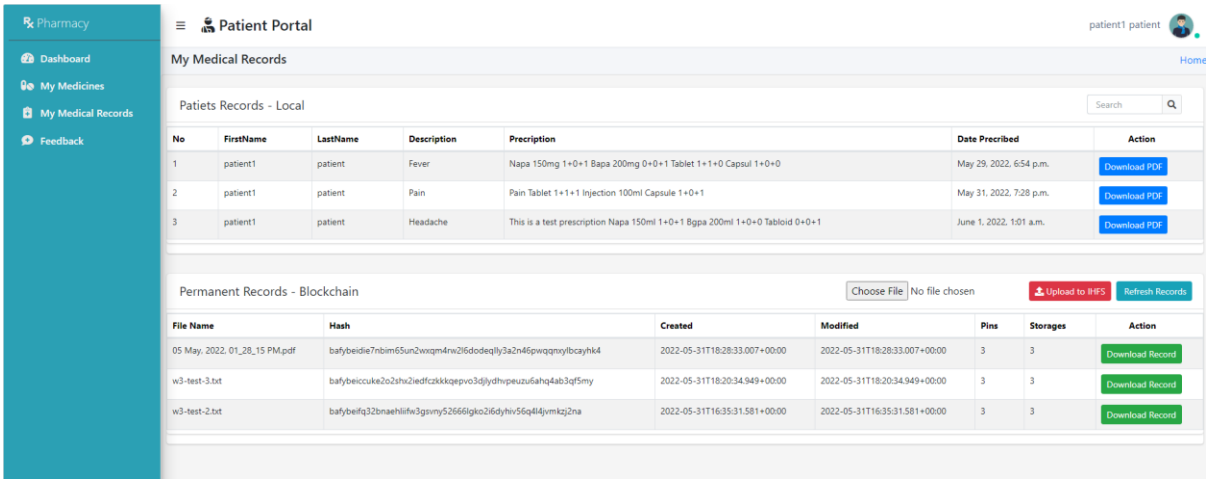

Figure 8 Patient Medical Records Stored Locally (Top Table) and Permanently on Blockchain (Bottom Table)

Doctor Module:

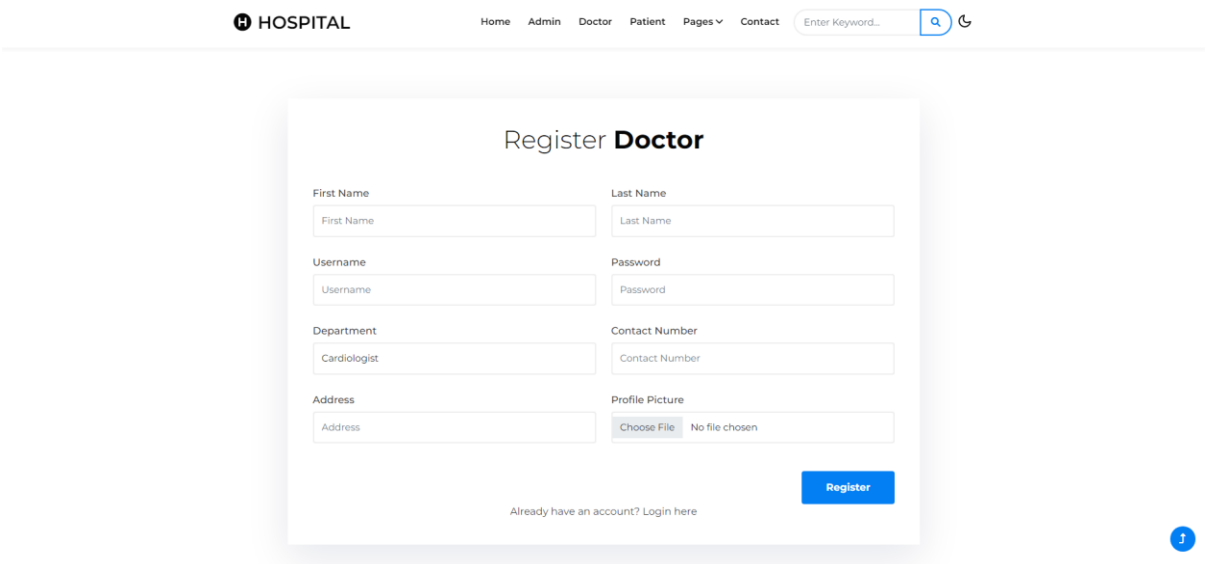

Figure 9 Doctor Registration Form

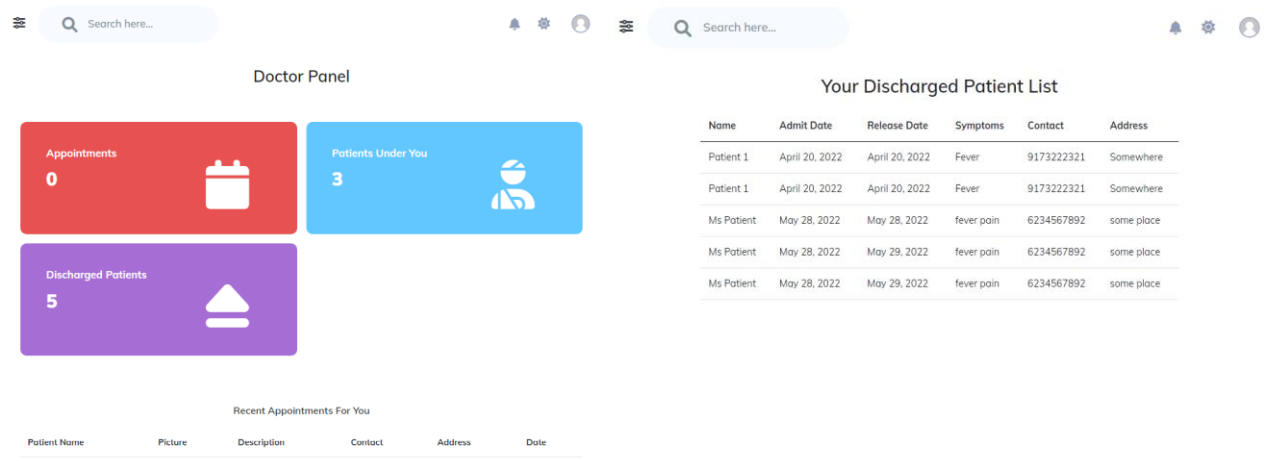

Figure 10 Doctor Dashboard Cards (left) and Previous Patients List (right)

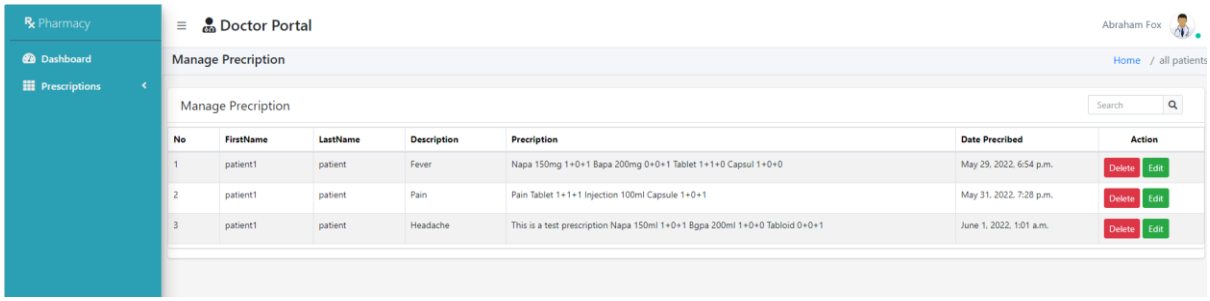

Figure 11 Doctor Manage Previous Prescriptions

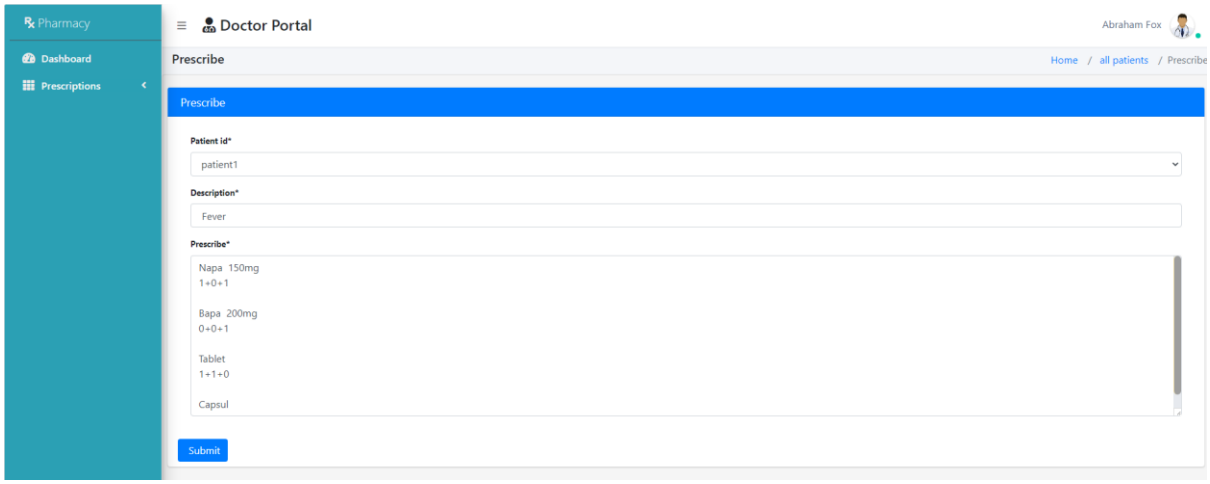

Figure 12 Doctor Prescription Form

Pharmacy Module

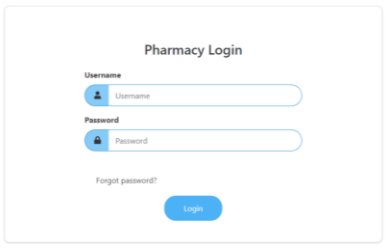

Figure 13 Pharmacy Login for Admin, Pharmacist and Clark

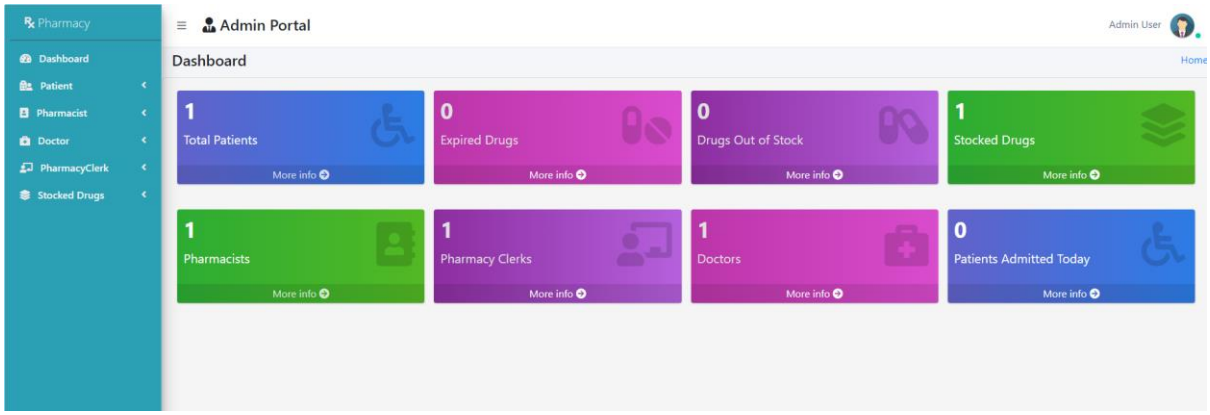

Figure 14 Pharmacy Admin Dashboard

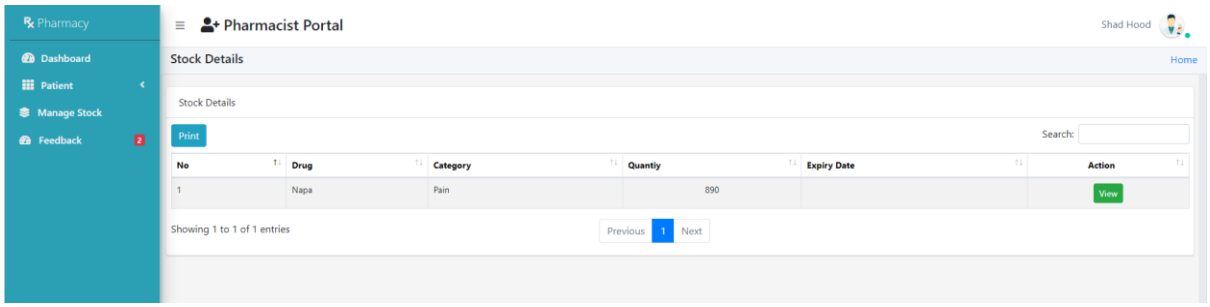

Figure 15 Pharmacist Portal for Managing Stock

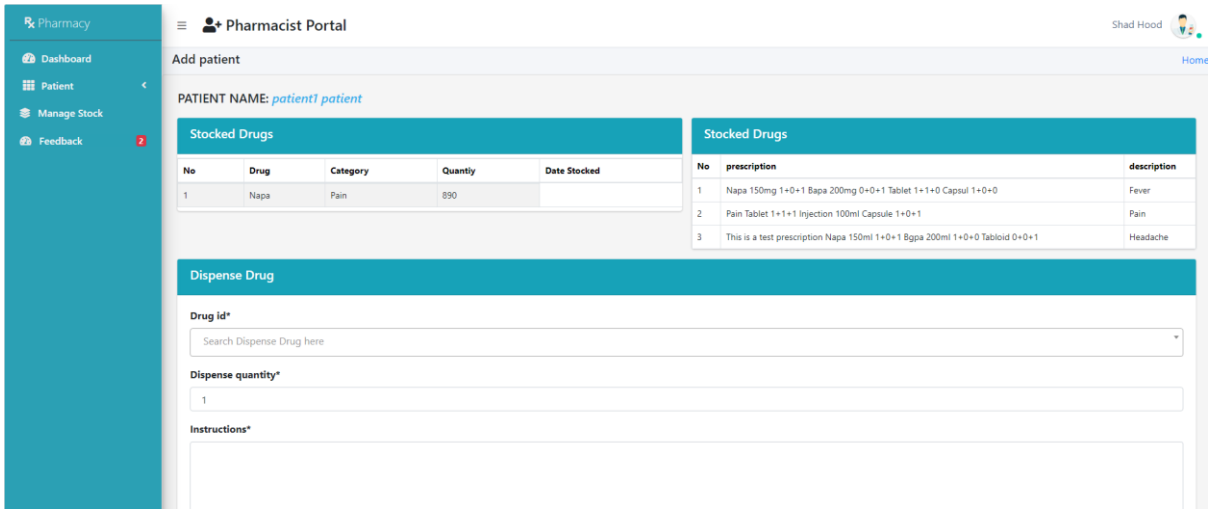

Figure 16 Pharmacist Portal for Dispensing Drug based on Prescription

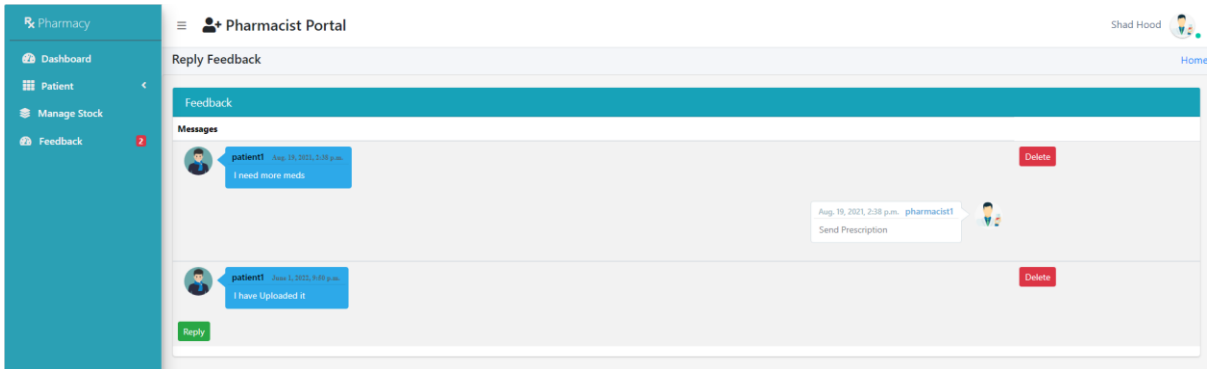

Figure 17 Pharmacist Portal for Chat and Feedback with Patient

Hospital Admin Module

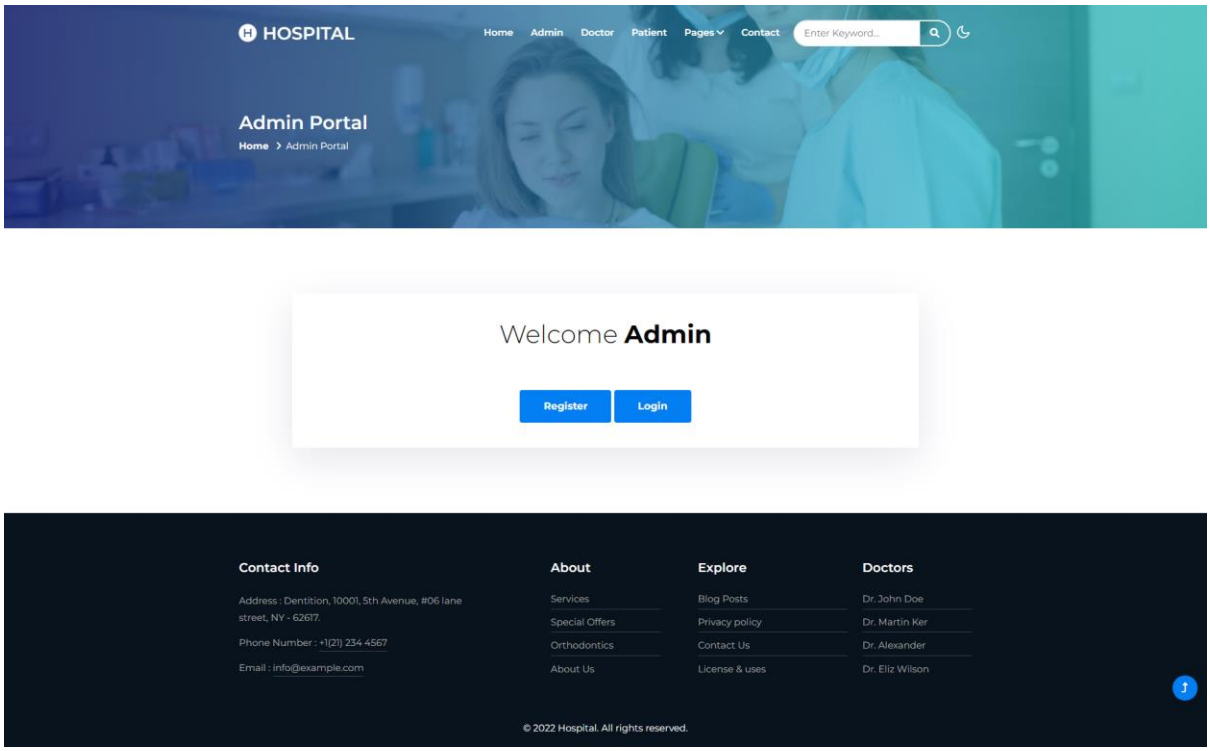

Figure 18 Hospital Admin Portal

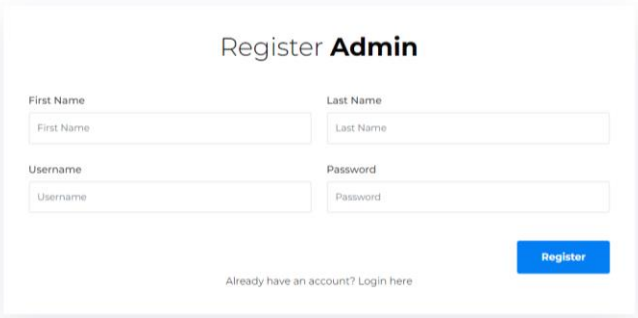

Figure 19 Hospital Admin Registration

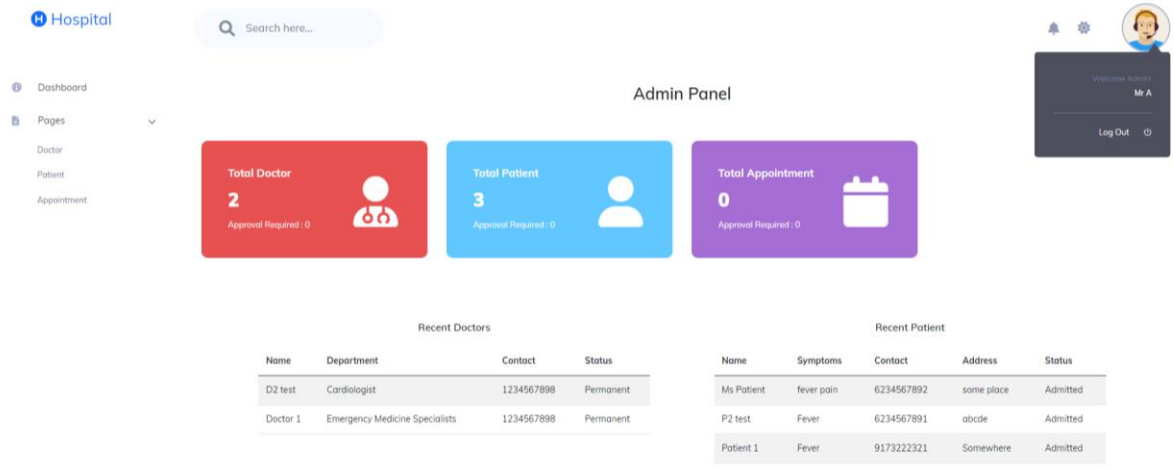

Figure 20 Hospital Admin Dashboard Cards

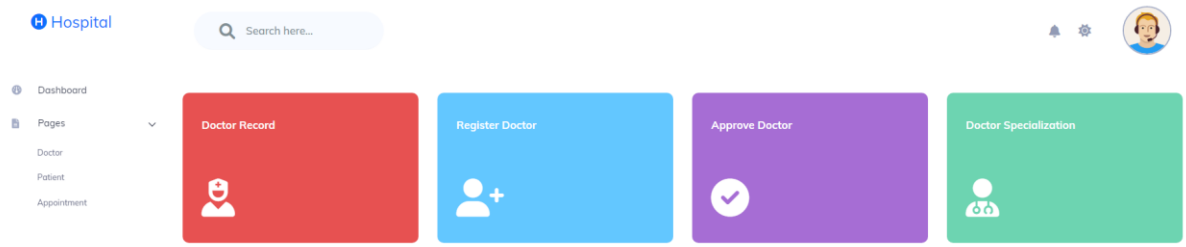

Figure 21 Hospital Admin Doctor Management Cards

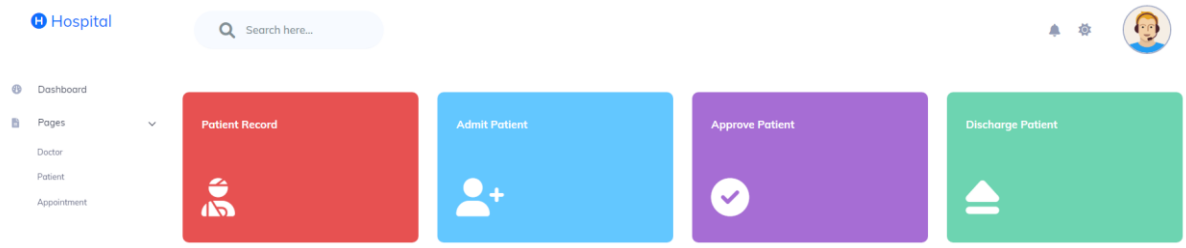

Figure 22 Hospital Admin Patient Management Cards

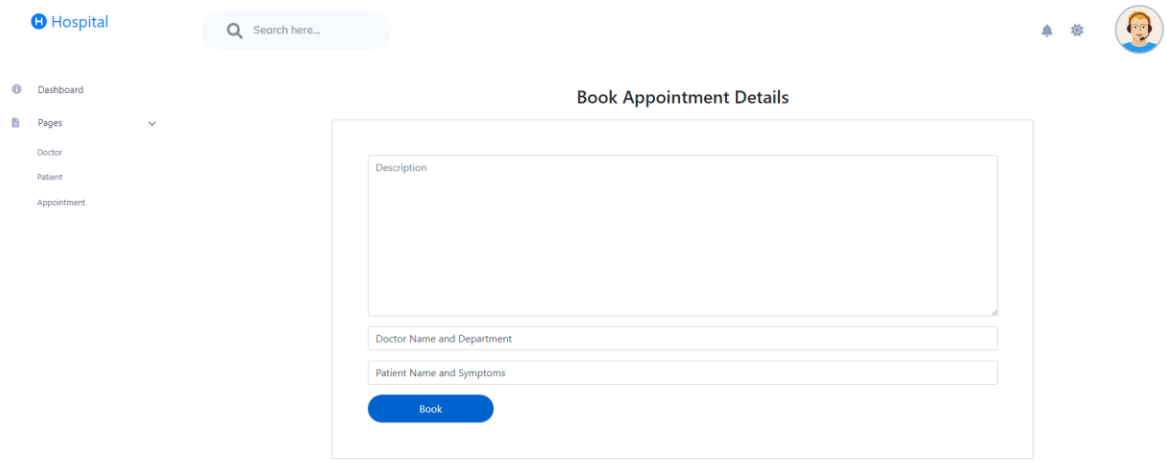

Figure 23 Hospital Admin Appoint Form

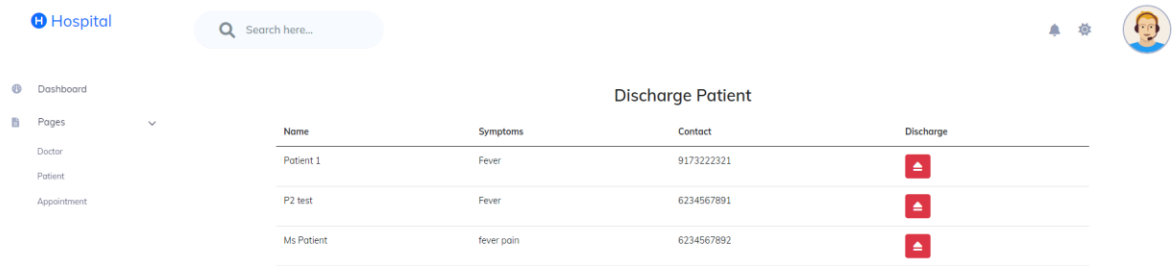

Figure 24 Discharge Patient

Ambulance Module

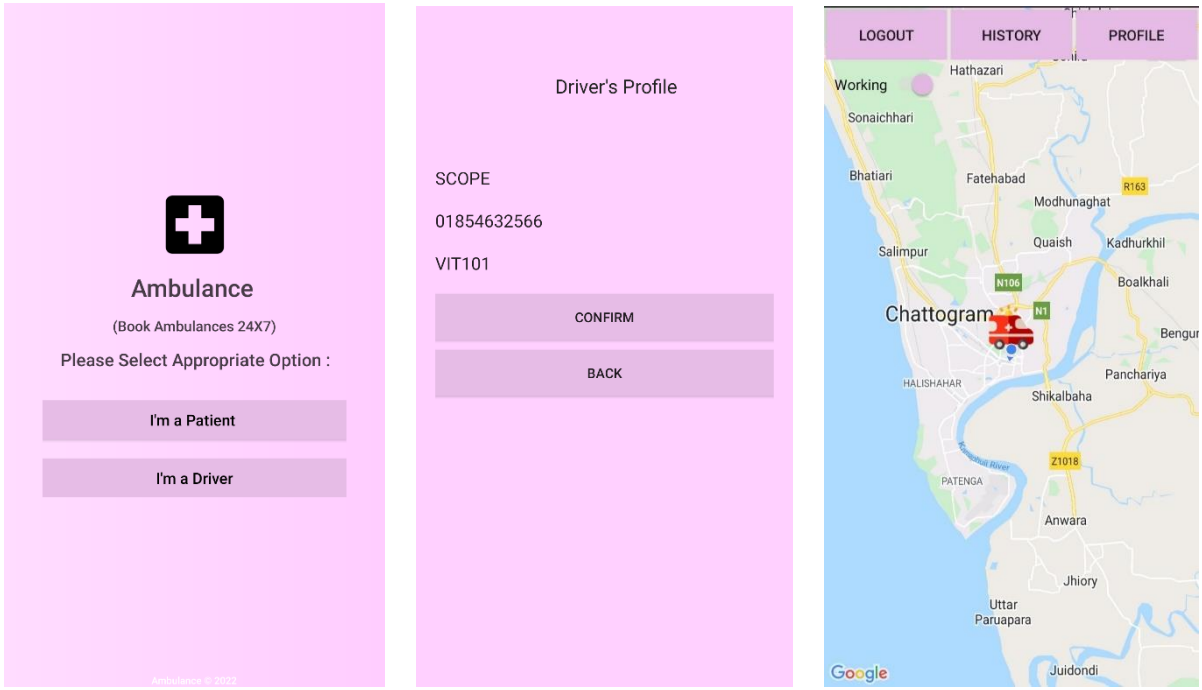

Figure 25 Ambulance App - Login Page (left); Driver Profile Page (mid); Driver Home Page (right)

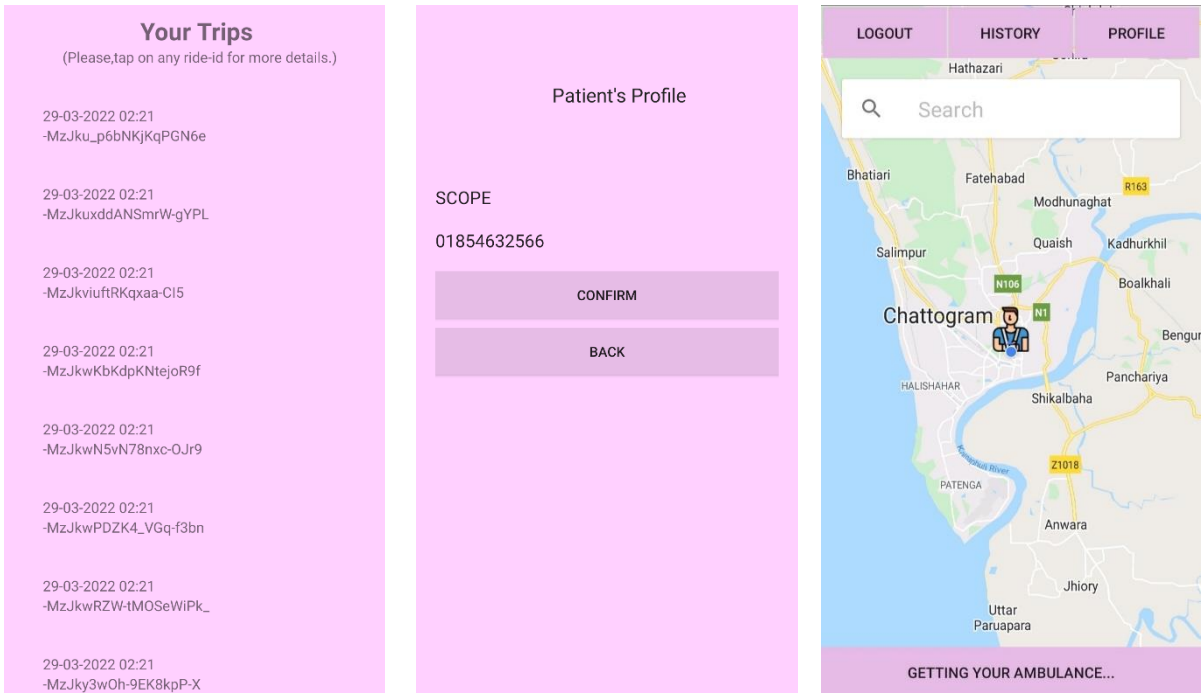

Figure 26 Ambulance App - Trip History Page (left), Patient Profile Page (mid), Patient Home Page (right)

7 RESULT AND DISCUSSION

In reality, it is quite difficult to execute a huge scare decentralized project of this kind and track all relevant data as a project prototype for implementation IHFS and all the user database

related work. That is why for the database we have used IPFS for prototyping and for the ambulance service we have used our proposed system on simulator for calculating shortest path and communicating real-time data from the ambulance with the hospital and other respected parties. In order to track data like journey distance, speed, energy consumption, and speed. We have simulated our architecture using "Simulation of Urban MObility (SUMO)", "Network Simulator 3 (NS-3)", and "OpenStreetMap (OSM)". We have built a SUMO configuration file, osm.sumocfg, using OSM to later generate a trace file for simulation on NS-3. We ran SUMO on both the Dijkstra and the A* algorithms. Then, for 50 minutes, we ran the trace files on NS-3 for eight total nodes and obtained our performance results for eight of those nodes, including total travelled distance and time, as well as all fuel and emission metrics. These results have been put on a Time-Fuel chart as Fig. 27. Throughout the experiment, eight cars were dispatched in a variety of time intervals, including one gap as non-peak hour to cut the local ad hoc connection and use standalone GPS and V2I communications. By comparing these results to previous efforts, we can see that our performance was nearly as efficient as [13], despite the fact that we used three distinct technologies to determine accurate real-time location along with the default navigation algorithms used by major navigation services [14].

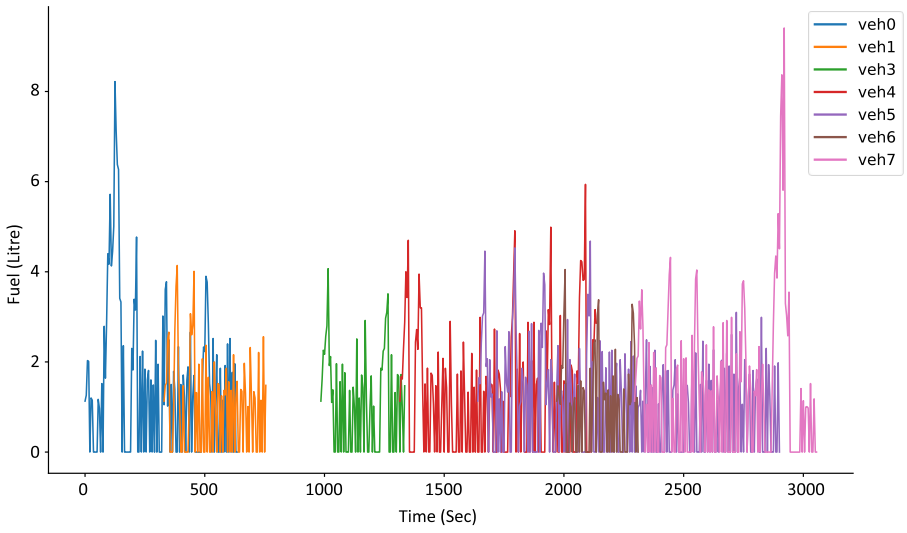

Figure 27 Simulation Result Plot of proposed network architecture ran on our Institution Map

8 SUMMARY

As a consequence of the continuous development, we have already achieved some of the promised achievements, and more are anticipated in the near future. Using state-of-the-art blockchain technology, we have built a totally decentralized system in which data is saved in

our proprietary file system called IHFS, which is entirely protected from data security and integrity concerns. This prevents any form of data integrity issues from occurring. We have developed a route planning system for ambulance services that is both efficient and quick. Once the whole service is up and running with a small number of subscriber hospitals, the database-related tasks will become entirely self-sustaining, and it will monitor all occurrences, including backups, on a continuous basis. Our secure pipeline access via the App provides customers with simple but very secure access to the network, as well as a straightforward method of granting or revoking access to other parties that are engaged in the transaction. It also makes use of the hospitals' current system architecture as well as the extremely popular smartphone infrastructure, which is perfectly compatible with the system and does not need the construction of any new major infrastructure to maintain the system operational. This will help up finalizing the platform which will connect all the major healthcare services.

9 REFERENCES

[1] A. Rejeb, H. Treiblmaier, K. Rejeb and S. Zailani, "Blockchain research in healthcare: a bibliometric review and current research trends," *Journal of Data, Information and Management,* vol. 3, p. 109–124, 2021.

[2] Y.-Z. Jiang, X. Du, B. Chen, M. Ma and Y. Zhang, "Research on the Application of Blockchain in Smart Healthcare: Constructing a Hierarchical Framework," *Journal of Healthcare Engineering*, vol. 2021, p. 6698122, 2021.

[3] K. Fan, S. Wang, Y. Ren, H. Li and Y. Yang, "MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain," *Journal of Medical Systems*, vol. 42, p. 136, 2018.

[4] G. Yang, C. Li and K. E. Marstein, "A blockchain-based architecture for securing electronic health record systems," *Concurrency Computat Pract Exper,* vol. 33, p. e5479, July 2021.

[5] F. Alam Khan, M. Asif, A. Ahmad, M. Alharbi and H. Aljuaid, "Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development," *Sustainable Cities and Society,* vol. 55, p. 102018, 2020.

[6] P. Pandey and R. Litoriya, "Securing and authenticating healthcare records through blockchain technology," *null*, vol. 44, p. 341–356, July 2020.

[7] L. Athota, V. K. Shukla, N. Pandey and A. Rana, "Chatbot for Healthcare System Using Artificial Intelligence," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020.

[8] U. Bharti, D. Bajaj, H. Batra, S. Lalit, S. Lalit and A. Gangwani, "Medbot: Conversational Artificial Intelligence Powered Chatbot for Delivering Tele-Health after COVID-19," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020.

[9] S. Kumar, A. K. Bharti and R. Amin, "Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions," *Security and Privacy*, vol. 4, p. e162, September 2021.

[10] R. Kumar and R. Tripathi, "A Secure and Distributed Framework for sharing COVID-19 patient Reports using Consortium Blockchain and IPFS," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2020.

[11] R. K. Marangappanavar and M. Kiran, "Inter-Planetary File System Enabled Blockchain Solution For Securing Healthcare Records," in *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, 2020.

[12] V. Lingayat, I. Pardikar, S. Yewalekar, S. Khachane and S. Pande, "Securing Pharmaceutical Supply Chain using Blockchain Technology," *ITM Web Conf.*, vol. 37, 2021.

[13] I.-C. Chang, H.-T. Tai, D.-L. Hsieh, F.-H. Yeh and S.-H. Chang, "Design and Implementation of the Travelling Time- and Energy-Efficient Android GPS Navigation App with the VANET-Based A* Route Planning Algorithm," Chengdu, 2013.

[14] D. R. Lanning, G. K. Harrell and J. Wang, "Dijkstra's Algorithm and Google Maps," in *Proceedings of the 2014 ACM Southeast Regional Conference*, New York, NY, USA, 2014.

[15] R. S. Kaurav, R. R. Rout and S. Vemireddy, "Blockchain for Emergency Vehicle Routing in Healthcare Services: An Integrated Secure and Trustworthy System," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2021.

[16] A. Finogeev, S. Vasin, L. Gamidullaeva and D. Parygin, "Blockchain and Smart Contracts for Support the Interaction between the Actors in the Regional Innovation

System," in *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, 2018.

[17] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016.