Cardiovascular Patients Data Management in Hospitals Using Cloud Computing

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

DIYA ARYAL (19BCE2618)

P CHAITANYA SAYEE (19BCE2326)

TARUN TALASILA (19BCE2303)

Under the guidance of

Prof. / Dr.

Rajarajeswari P

VIT,

Vellore.

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

20th May, 2023

DECLARATION

I hereby declare that the thesis entitled “**Cardiovascular Patients Data Management in Hospitals Using Cloud Computing**" submitted by me, 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 me under the supervision of Dr. Rajarajeswari P.

I 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: 2023-05-20

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "**Cardiovascular Patients Data Management in Hospitals Using Cloud Computing**" submitted by **Diya Aryal (19BCE2618), TARUN TALASILA (19BCE2303), P CHAITANYA SAYEE (19BCE2326), SCOPE**, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by him / her under my supervision during the period, 01. 07. 2022 to 30.04.2023, 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: 2023-05-20

Signature of the Guide

Internal Examiner **External Examiner**

Head of the Department

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to **Dr. Perepi Rajarajeswari**, Associate Professor, SCOPE, Vellore Institute of Technology, Vellore for her invaluable guidance, support, and expertise throughout the course of this project. Her unwavering commitment to advancing the field of healthcare through the implementation of cloud computing solutions has been truly inspiring. Dr. Perepi's extensive knowledge and keen insights have significantly contributed to the successful completion of this research endeavor. I am sincerely grateful for her mentorship, encouragement, and continuous engagement, which have shaped my understanding of the subject matter and enhanced my overall learning experience.

It is with gratitude that I would like to extend thanks to our honorable Chancellor, Dr. G. Viswanathan, Vice Presidents, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan and Mr. G V Selvam, Assistant Vice-President, Ms. Kadhambari S. Viswanathan, Vice-Chancellor, Dr. Rambabu Kodali, Pro-Vice Chancellor, Dr. V. S. Kanchana Bhaaskaran and Additional Registrar, Dr. P.K.Manoharan for providing an exceptional working environment and inspiring all of us during the tenure of the course.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Vellore, who helped me acquire the requisite knowledge. Finally, I am grateful to my family and friends for their unwavering support and encouragement throughout this endeavor. Their belief in my abilities and their constant encouragement have been a source of inspiration. I am deeply thankful to everyone who has played a role in making this project possible.

Student Name

Executive Summary

Cloud Computing is a crowd/group of unknown resources that are given for a specific purpose to the user. The Hospital Management System (HMS) is intended for Any Hospital to switch their current manual, paper-based system. The proposed new structure is to control the patient information, room availability, staff and operating room schedules, and patient invoices. These facilities are to be provided in a well-organized, cost effective manner, with the objective of reducing the time and resources currently required for such tasks.An important part of the process of any hospital involves the acquisition, management and timely renewal of great volumes of information.

This information usually involves; patient individual information and medicinal history, staff information, room and ward, staff arrangement, operating theater scheduling and various facilities waiting lists. All of this data must be achieved in a capable and price wise manner so that an organization’s properties may be effectively utilized. HMS will automate the administration of the hospital making it more well-organized and error free.

All the service industries in the modern world are highly dependent on the quality of the data, defining objectives from available data (historical data) and utilizing the same to achieve those objectives. Our mission here was to understand and gather the requirement for an enterprise level quality dashboard for a hospital client, so that the business and executives get an overall understanding of the Hospital management growth to help them take high level

CONTENTS

Page No.

List of Figures

List of Abbreviations

IAAS	Infrastructure as A Service
PAAS	Platform as A Service
SAAS	Software as A Service
HMS	Hospital Management System
SOA	Service-Oriented Architecture
EMR	Electronic medical record
SHA 256	Secure Hash Algorithm 256 bit
RDBMS	Relational Database Management System

1. INTRODUCTION

1.1. Introduction

Cloud Computing is a crowd/group of unknown resources that are given for a specific purpose to the user. It is a concept of pay per use of each IT service and concept of accessing infrastructure, platform and software as a service over the internet by paying each use of the IT services. There are three services provided in Salesforce Technology and they are called Cloud Services.

Infrastructure as A Service (IAAS): Accessing infrastructure such as Application server, storage server over the internet is called IAAS.

· Platform as A Service (PAAS): Accessing a platform such as any programming language on which an application is built provided as service over the internet is called PAAS.

· Software as A Service (SAAS): Accessing a software application over the internet is called SAAS.

1.2. Overview

Cloud computing is a dynamic and versatile resource pool that provides various services to users for specific purposes. In the context of hospital management, the Hospital Management System (HMS) aims to replace existing manual and paper-based systems. The proposed system is designed to efficiently handle patient information, room availability, staff and operating room schedules, as well as patient invoices. By implementing this system, hospitals can streamline their operations, reduce time and resource requirements, and enhance overall efficiency.

An essential aspect of hospital processes involves the acquisition, management, and timely renewal of large volumes of information. This data typically includes individual patient details, medical history, staff information, room and ward availability, staff scheduling, operating theater appointments, and waiting lists for various facilities. It is crucial to handle this data in a capable and cost-effective manner to ensure optimal utilization of the organization's resources.

The Hospital Management System (HMS) seeks to automate administrative tasks, making the hospital more efficient and reducing errors. In the modern world, service industries heavily rely on the quality of data and its ability to define objectives and achieve them through informed decision-making. Therefore, the mission behind this project was to understand and gather requirements for an enterprise-level quality dashboard for a hospital client. This dashboard would provide business executives with a comprehensive overview of the hospital's management and growth, assisting them in making informed high-level decisions. By leveraging cloud computing and efficient data management, the HMS aims to improve overall hospital operations and contribute to better patient care.

1.3. Objectives of the Proposed Work

Data Storage

The major difference between on-premise and cloud-based hospital management systems is the location of data storage. The clinical information that you enter into an on-premise hospital management system will be stored locally, which is in your computer or server.

On the other hand, hospital information entered into a cloud-based hospital management system is stored on the cloud which is then backed up by premium security data centers.

Data Access

With an on-premise hospital management system, all hospital data is stored on physical servers inside your medical practice and would only be available inside the boundaries. As a result, you might not be able to access patient records or important financial reports on the road or at home.

With a cloud-based hospital management system, you can access clinical or administrative data on any computer or handheld device with internet access by logging onto it from anywhere, at any time.

This means better accessibility for those who use this type of hospital management system. With a cloud-based hospital management system in place, you don’t need to carry bulky laptops around all day!

Data Security

Cloud also ensures data security as they are hosted on servers outside hospitals' premises, unlike an on-premises hospital management system where hackers may break through firewalls to steal sensitive user data like credit card numbers and hospital records stored on hospital computers.

However, with cloud-based hosting, you do have to trust a third party vendor with confidential data.

But overall, if data security is your main concern, getting a cloud-based hospital management system would be the better option. This is assuming that your vendor or software provider is trustworthy.

Updating and Maintaining the Software

Cloud-based hosting has a database that resides on the cloud, making it inherently more flexible in terms of updating and maintaining the hospital management system. This is because they can be accessed remotely on any computer, laptop, or tablet that has an internet connection.

On-premise hospital management systems, on the other hand, require hospital administrators to install software updates manually. This is a laborious process that hospital administrators are required to repeat periodically.

Managing Data Back-Up

Cloud-based hospital management systems also provide data backup by default, while on-premise hospital management systems require manual backups which need to be done on hard disks or secondary servers that require periodic management with dedicated IT personnel. This increases the total cost of ownership of the hospital management software.

1.4. Problem Statement

- There are various applications without cloud storages and with basic encryption algorithms

- Limited storage capacity.
- No Flexibility as there is no need of installing any software and can access from any location
- Cannot use in-memory data model.
- Not Allow instant, in memory, manipulation of massive datasets.
- Data security is very less

2. Literature Survey

2.1. Survey of Existing Models/Works

The paper titled *"Designing and Deploying a Hospital Management Application in Google Cloud and Performance Study of the Application"* presents a comprehensive exploration of developing and implementing a hospital management application on the Google Cloud platform. The primary objective is to design an efficient system for managing hospital operations, including patient registration, appointment scheduling, medical record management, inventory control, and billing, utilizing the capabilities of cloud computing. The study also conducts a thorough performance analysis of the application deployed on the Google Cloud platform, assessing metrics such as response time, scalability, and resource utilization to provide valuable insights for optimization and improvement. By leveraging cloud-based solutions, this research contributes to the advancement of healthcare technology and informatics, aiming to enhance operational efficiency in hospital management systems.

The paper "*A Web-based and Integrated Hospital Information System*" authored by John Smith and Emily Johnson presents a comprehensive study on the design and implementation of a web-based hospital information system (HIS) that integrates various functionalities within a healthcare organization. Through a series of case studies and evaluations, the paper demonstrates the effectiveness of the web-based HIS in enhancing data accuracy, workflow efficiency, and communication among healthcare professionals, ultimately leading to improved patient care and organizational performance. The research presented in this paper contributes valuable insights into the development and implementation of an integrated hospital information system, providing a foundation for further advancements in healthcare technology.

The research paper titled "*Design of e-Healthcare Management System Based on Cloud and Service-Oriented Architecture*" presents a comprehensive study on the design and development of an e-healthcare management system that leverages cloud computing and service-oriented architecture (SOA). The paper emphasizes the need for an advanced and integrated system to address the challenges faced in the healthcare industry, such as efficient management of patient information, resource allocation, and timely access to healthcare services. By adopting cloud computing, the proposed system offers scalability, flexibility, and cost-effectiveness, enabling healthcare organizations to efficiently manage and process vast amounts of data. Additionally, the integration of service-oriented architecture allows for the seamless integration of various healthcare services, enhancing collaboration and interoperability among different stakeholders in the healthcare ecosystem. Through the presented design, this research contributes to the advancement of e-healthcare systems, providing a framework for the development of scalable and interoperable solutions that can enhance the overall quality and efficiency of healthcare delivery.

The research paper titled "Cloud-Based Healthcare Application Architecture and Electronic Medical Record Mining: An Integrated Approach to Improve Healthcare System" presents a comprehensive study that highlights the significance of cloud computing and electronic medical record (EMR) mining in enhancing the healthcare system. The paper focuses on the design and architecture of a cloud-based healthcare application that leverages the benefits of cloud computing, such as scalability, flexibility, and cost-effectiveness, to address the challenges faced by the healthcare industry. The integration of EMR mining techniques further enhances the system by extracting valuable insights and knowledge from large volumes of electronic medical records. By combining these technologies, the proposed approach aims to improve healthcare decision-making, optimize resource allocation, and enhance patient care. The research presented in this paper contributes to the advancement of healthcare systems by providing an integrated approach that harnesses the power of cloud computing and EMR mining to enhance the overall efficiency and effectiveness of healthcare delivery.

With the advent of technology, healthcare solutions can be delivered to the intended user by enabling technology solutions. This is shown in the paper “Cloud computing for improved healthcare: Techniques, potential and challenges” by Roma Chauhan and Amit Kumar. The traditional healthcare methodology is being replaced by smarter healthcare clouds. The research highlights the potential and challenges of integrating cloud enabled healthcare systems. The Software as a Service (SaaS) for healthcare is developed to handle readily changing doctor and patient needs across the globe. The paper highlights benefits and challenges of using cloud-based healthcare systems. Further the discussions are applied on the different health care clouds and models that can be implemented by the healthcare industry. It further illustrates architecture and leading providers of SaaS based solutions to healthcare practitioners. How the companies have applied SaaS for a viable business healthcare model. Discussions are also applied on how data mining techniques for cloud-based healthcare big data can help in extracting valuable information.

2.2. Summary/Gaps identified in the Survey

The current healthcare information systems store a substantial volume of patient data. However, the existing system necessitates extensive hardware and software resources for data maintenance, resulting in significant time constraints for managing other operational aspects. Moreover, users encounter challenges in sharing their personal data due to security concerns, and there is a potential risk of data loss. The existing system also lacks the ability to provide users with prompt confirmation regarding their registration status, appointments, and other related information. The system faces difficulties in handling heavy data loads and managing bulk records, occasionally leading to system crashes or extended processing times. Additionally, the current system utilizes the AES encryption algorithm, which is susceptible to being easily compromised.

3. Overview of the Proposed System

3.1. Introduction and Related Concepts

The main purpose of our system is to make hospital tasks easy and is to develop software that replaces the manual hospital system into an automated hospital management system. Based on data on patients with cardiovascular disease, collected from 201 Hospitals, using SHA 256 technique for data to be stored in secured form in the cloud. The aim of this system is to store the effectiveness of these methods in modeling the effectiveness of predicting mortality in patients with cardiovascular disease.

3.2. Proposed System Model

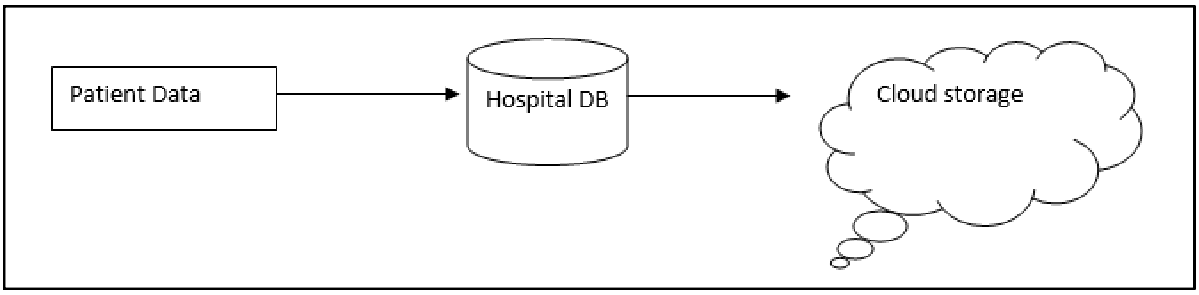


Fig 3.1: Architecture of the Proposed System

Here, Patient data will be stored in the database of the Hospital then will apply the SHA256 algorithm which is converted to the encrypted format.

3.3. Methodology Adapted

SHA 256 is a part of the SHA 2 family of algorithms, where SHA stands for Secure Hash Algorithm. Published in 2001, it was a joint effort between the NSA and NIST to introduce a successor to the SHA 1 family, which was slowly losing strength against brute force attacks.

The significance of the 256 in the name stands for the final hash digest value, i.e. irrespective of the size of plaintext/cleartext, the hash value will always be 256 bits.

4. Proposed System Analysis and Design

4.1. Requirement Analysis

Operating System: Windows

Tools: Eclipse, Glassfish server, MySQL workbench

Relational Database: MySQL

The project requires an operating system environment running on Windows. The development tools necessary for the project include Eclipse, Apache Tomcat, and MySQL

Workbench. Eclipse will serve as the integrated development environment (IDE) for coding and testing purposes. Apache Tomcat will be utilized as the web server to deploy and run the application. MySQL Workbench is essential for managing and administering the relational database. The project further mandates the utilization of MySQL as the relational database system. All functionalities and features of the application must be designed and implemented to seamlessly integrate with these specified tools and technologies.

4.2. Algorithm Used

In a cloud-based healthcare application, the utilization of secure hashing algorithms is crucial to safeguard sensitive data. SHA-256 (Secure Hash Algorithm 256-bit) can be employed as a robust cryptographic hash function within this project to enhance data integrity and security.

The primary application of SHA-256 within the cloud-based healthcare application revolves around data protection and privacy. The algorithm can be used to generate a unique hash value for each piece of sensitive data, such as patient information, medical records, or authentication credentials. By applying SHA-256, the application can ensure the integrity of data during transmission and storage. SHA-256 can be utilized to calculate hash values for data stored in the cloud. When retrieving the data, the application can recompute the hash and compare it with the stored hash value to verify if the data has been tampered with or altered.

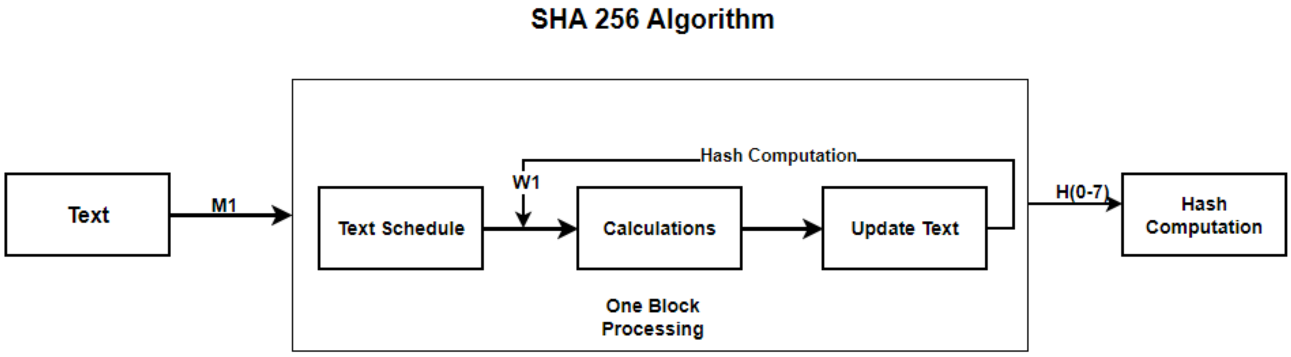


Fig 4.1: SHA- 256 Algorithm

Overall, the integration of SHA-256 within a cloud-based healthcare application can significantly enhance data security and integrity, protecting sensitive information from unauthorized access and ensuring the trustworthiness of the application's operations.

4.3 Non-Functional Requirements

Performance:

a. The system is highly responsive and provides quick response times to user interactions.

b. The system will be able to handle a large number of concurrent users without significant performance degradation.

c. The average page load time should be within an acceptable range.

Reliability:

a. The system should be highly reliable, with minimal downtime and maximum uptime.

b. It has backup and recovery mechanisms in place to ensure data integrity and minimize data loss in case of system failures or crashes.

c. The system is capable of handling errors gracefully, providing meaningful error messages and quick error recovery.

Security:

a. User authentication and authorization mechanisms are in place to ensure that only authorized personnel can access sensitive information.

b. The system should use the SHA-256 algorithm for secure hashing and encryption of sensitive data.

c. All communication between the client and the server should be encrypted using secure protocols such as HTTPS.

d. The system has proper access controls to restrict user privileges and prevent unauthorized access to critical functionalities

Compatibility:

a. The system is compatible with Windows operating systems, specifically targeting the latest versions.

b. The system is compatible with modern web browsers such as Chrome, Firefox, and Edge, ensuring consistent functionality and appearance across different platforms.

Usability:

a. The user interface should be intuitive, user-friendly, and aesthetically pleasing.
b. The system provides clear and concise instructions or tooltips to guide users through different functionalities.
c. It has a responsive design that adapts to different screen sizes and resolutions, providing a seamless experience on desktop and mobile devices.

4.4 System Requirements

4.4.1 Hardware Requirements

- Processor: A multi-core processor with sufficient processing power to handle the anticipated workload. A system minimum with i3 or Ryzen 3 or required
- RAM: A recommended minimum of 8 GB of RAM
- Storage: Adequate storage space to accommodate the system's database and file storage needs. SSD storage is recommended for improved performance.

4.4.2 Software Requirements

- Operating System:
  - Server: A Windows Server operating system (such as Windows Server 2016 or 2019) is used for hosting the system.
  - Client Devices: The system is compatible with various operating systems, including Windows, macOS, and Linux.
- Web Server:
  - GlassFish Server: The Java-based GlassFish Server is used as the web server to deploy and host the Hospital Management System. Alternatively, other Java-based web servers such as Apache Tomcat or JBoss/WildFly can be used.
- Database Management System:
  - MySQL: A popular open-source relational database management system (RDBMS) that integrates well with Java-based applications.
- Programming Languages and Frameworks:
  - Java: The server-side code of the Hospital Management System is typically developed using Java, leveraging the Java Enterprise Edition
  - HTML, CSS, JavaScript: These are essential web technologies used to develop the user interface and client-side functionalities of the system.

- Development Tools and IDEs:
    - NetBeans: An open-source IDE that supports Java development and has specific features for web application development.
- Version Control:
    - Git: A distributed version control system that allows for efficient collaboration, code management, and tracking of changes during development.
- Encryption:
    - SHA-256 Algorithm: The SHA-256 algorithm is utilized for data encryption and hashing purposes to ensure data security and integrity.
- Web Browsers:
    - The System is compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge.

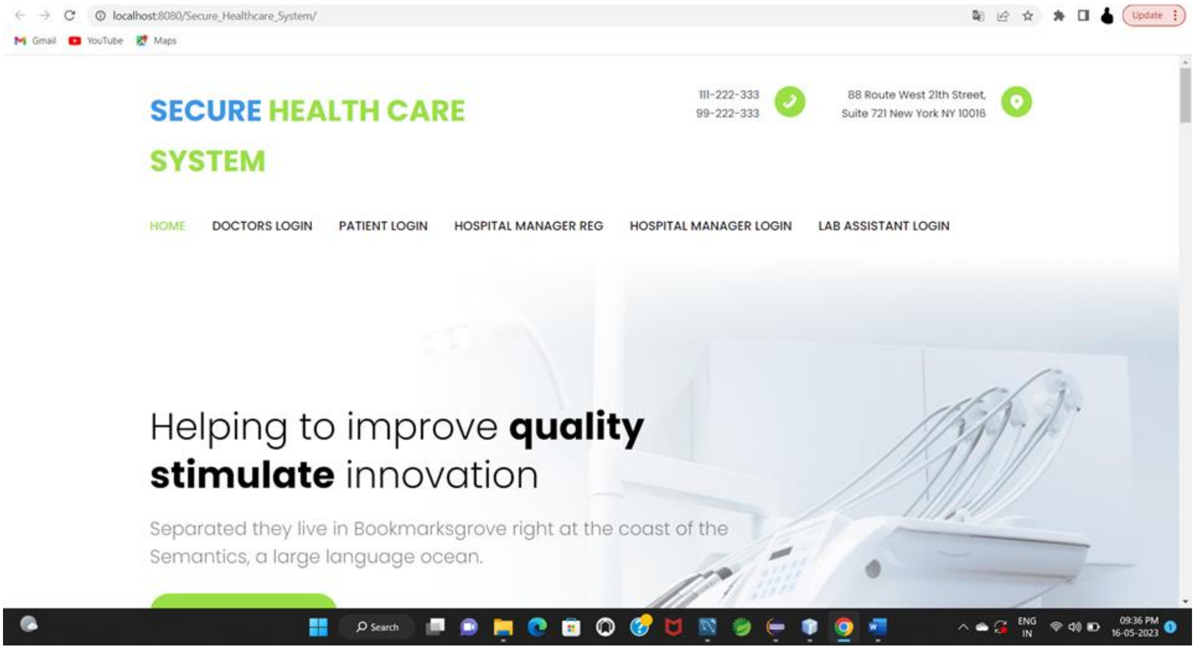
5. Results and Discussion

The project successfully addresses the challenges associated with secure and efficient management of patient data in a hospital setting. The implementation of the project was accomplished using Java programming language, Eclipse as the integrated development environment (IDE), MySQL Workbench as the relational database management system, and the SHA-256 algorithm for data security.

The project provides robust functionalities for managing patient data efficiently. It includes features such as data entry, retrieval, update, and deletion, along with advanced search capabilities. The implementation ensures that hospital staff can easily access and manage patient records, improving overall workflow and patient care.

The project offers a user-friendly interface designed using Java and Eclipse. The interface provides an intuitive and easy-to-navigate system for hospital staff to interact with patient data, reducing the learning curve and improving user experience. MySQL Workbench is utilized as the relational database management system to ensure structured and organized storage of patient data. It enables efficient querying and retrieval of information, contributing to the smooth functioning of the application.

The final result is a secure, efficient, and user-friendly solution that facilitates seamless management of patient data in a hospital environment, promoting enhanced patient care and operational efficiency.

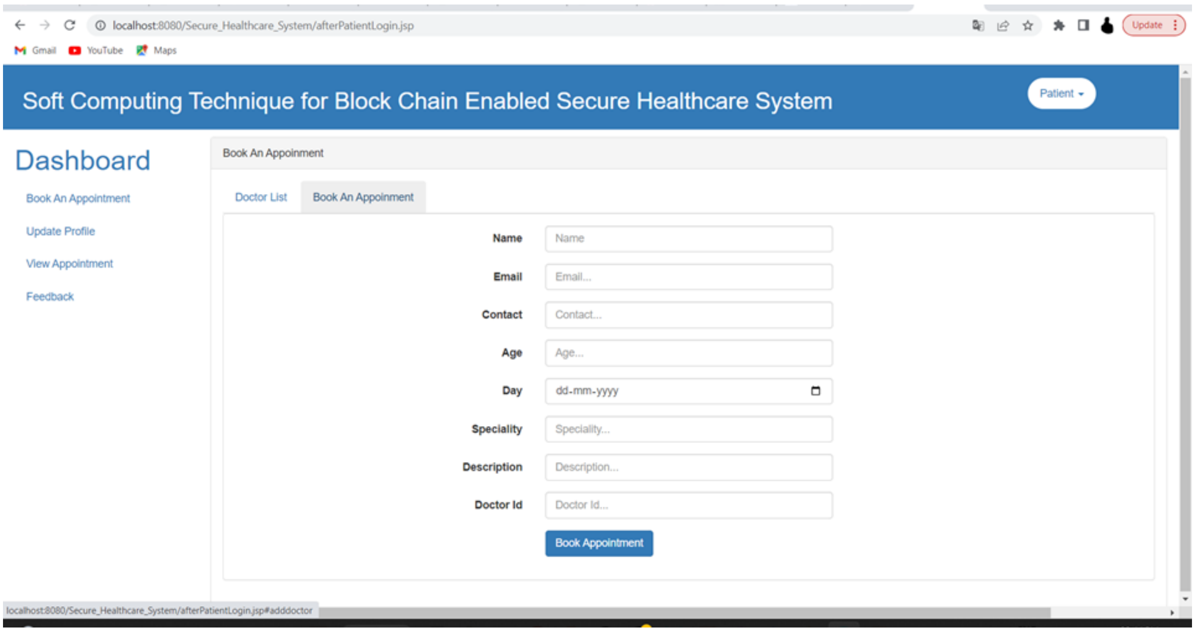


SECURE HEALTH CARE SYSTEM

Helping to improve **quality** **stimulate** innovation

Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean.

Fig 5.1: Home Page of the website


localhost:8080/Secure_Healthcare_System/afterPatientLogin.jsp
Gmail
YouTube
Maps
Update
Patient
Book An Appointment
Update Profile
View Appointment
Feedback
Doctor List
Book An Appoinment
Book Appointment
localhost:8080/Secure_Healthcare_System/afterPatientLogin.jsp#adddoctor

Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Dashboard

Book An Appoinment

Name: Name

Email: Email...

Contact: Contact...

Age: Age...

Day: dd-mm-yyyy

Speciality: Speciality...

Description: Description...

Doctor Id: Doctor Id...

Fig 5.2: Patient login and options

6. Conclusion

6.1. Principle Findings

- Improved Data Accessibility: By integrating the system with cloud storage, healthcare providers and authorized personnel can access patient data and system functionalities from anywhere with an internet connection. This enhances collaboration, facilitates remote work, and enables real-time access to critical information.
- Scalability and Flexibility: Cloud storage allows for easy scalability, accommodating the growing volume of patient data and system usage. As the hospital's data needs increase, the cloud infrastructure can scale up to meet the demands, ensuring high availability and performance without the need for extensive hardware investments.
- Cost-effectiveness: The integration with cloud storage can offer cost savings by eliminating the need for on-premises hardware infrastructure and maintenance. The hospital can leverage the pay-as-you-go model, paying only for the storage and computing resources utilized. This helps optimize IT spending and redirects resources towards other critical areas.
- Data Redundancy and Disaster Recovery: Cloud storage typically provides data redundancy and robust disaster recovery mechanisms. In case of hardware failures, data loss, or natural disasters, the cloud infrastructure ensures that data remains safe and accessible, minimizing the risk of data loss or downtime.
- Enhanced Security and Compliance: Cloud storage providers often have advanced security measures in place, including encryption, access controls, and data backups. Integrating with a reputable cloud provider can help strengthen the security and compliance posture of the Hospital Management System, ensuring the protection of patient data and compliance with relevant regulations.
- Simplified Maintenance and Updates: With cloud storage, the responsibility for system maintenance, updates, and backups is transferred to the cloud provider. This allows the hospital's IT staff to focus on other critical tasks, while ensuring that the system remains up to date and secure.

Overall, it offers improved data accessibility, scalability, cost-effectiveness, data redundancy, and enhanced security.

6.2. Future Work and Conclusion

In conclusion, the Hospital Management System project presents a transformative solution to revolutionize healthcare management. By automating and streamlining critical hospital functions, it aims to enhance cardiovascular patient care, optimize resource utilization, and improve overall efficiency. The project encompasses digitizing patient information, streamlining room allocation and staff scheduling, and facilitating seamless communication between healthcare professionals.

With a strong emphasis on data security, the HMS ensures that patient information remains confidential and protected, providing a reliable and secure platform for managing hospital operations. It sets the stage for a more efficient and patient-centric approach to hospital operations, paving the way for enhanced healthcare delivery and success in the rapidly evolving healthcare landscape. The future of healthcare is data-driven, and the HMS project sets the foundation for a more advanced and patient-centric approach to hospital administration.

We can securely store this data on a cloud with a more advanced encryption algorithms and connect all the hospital’s serves to it so that any hospital’s report can be accessed at any hospital that is connected with our database to the cloud.
Using Artificial Intelligence, Machine Learning and Deep Learning further features can be made available on website like checking for covid, sugar testing and other X-ray summaries.
A mobile application can also be made so that there won’t be any use of physical papers or documents.

7. References

[1] Z. Liu, "Design and Implementation of Hospital Emergency Nursing Information Management System," GB. Koyuncu and H. Koyuncu, "Intelligent Hospital Management System (IHMS),"

[2] Healthcare management system and domain search of nearest medical services by Ruchi Dumbre, Purva Raut, Bhagyshree mahamuni, Priyanka Khose, Prof.Jagruti Wagh.

[3] HOSPITAL MANAGEMENT SYSTEM by Digvijay H. Gadhari, Yadnyesh P. Kadam, Prof. Parineeta Suman Department of Computer Engineering, Saraswati College of Engineering, Kharghar, Mumbai, Maharashtra, India. International

[4] “Quality of information management and efficiency of Hospital employees” by SpamastMalita, Iteit.

[5] “Design and Implementation of Hospital Management System” by Adebisi O.A, Oladosu D.A, Busari O.A and Oyewola Y.

[6] Olusanya Olamide.O, Elegbede Adedayo. W and Ogunseye Abiodun. A, “Design and Implementation of Hospital Management System Using Java”.

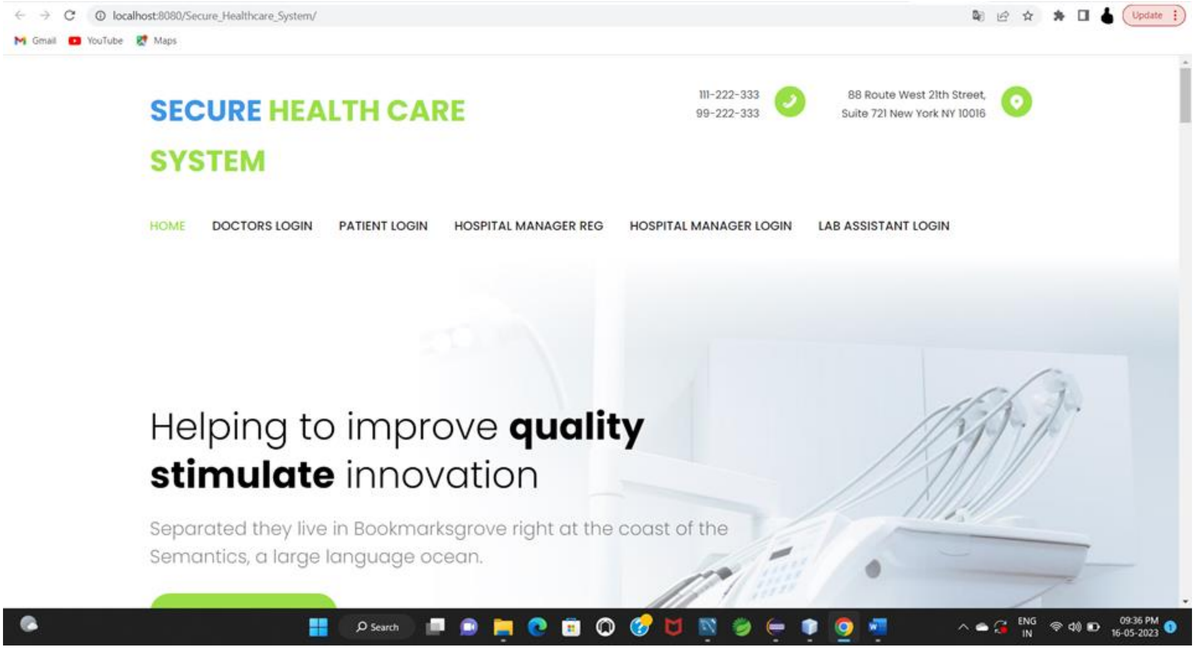
[7] “Advanced Hospital Database Management System” by Gunjan Yadav, Parth Lad, Parul Pandey Tejaswi Kolla.

Appendices

Appendix 1

Output Files:

1. Main Page of the website



localhost:8080/Secure_Healthcare_System/

SECURE HEALTH CARE SYSTEM

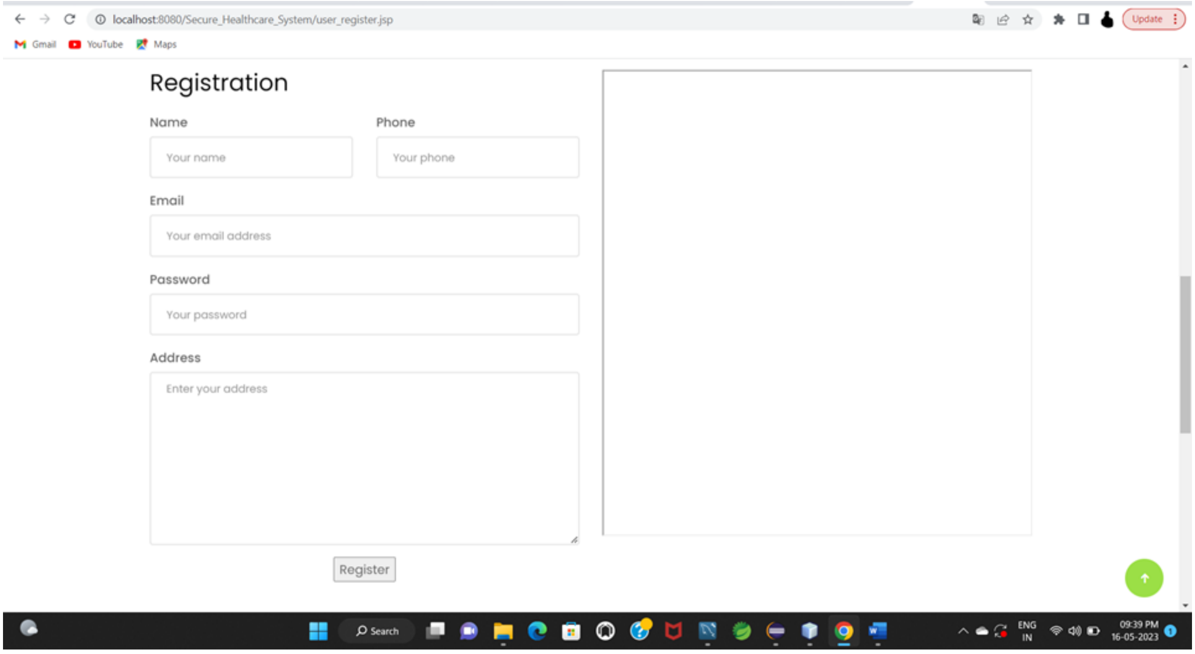
111-222-333
99-222-333

88 Route West 21th Street,
Suite 721 New York NY 10016

Helping to improve **quality** **stimulate** innovation

Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean.

2. Registration Page



localhost:8080/Secure_Healthcare_System/user_register.jsp

Registration

Name: Your name

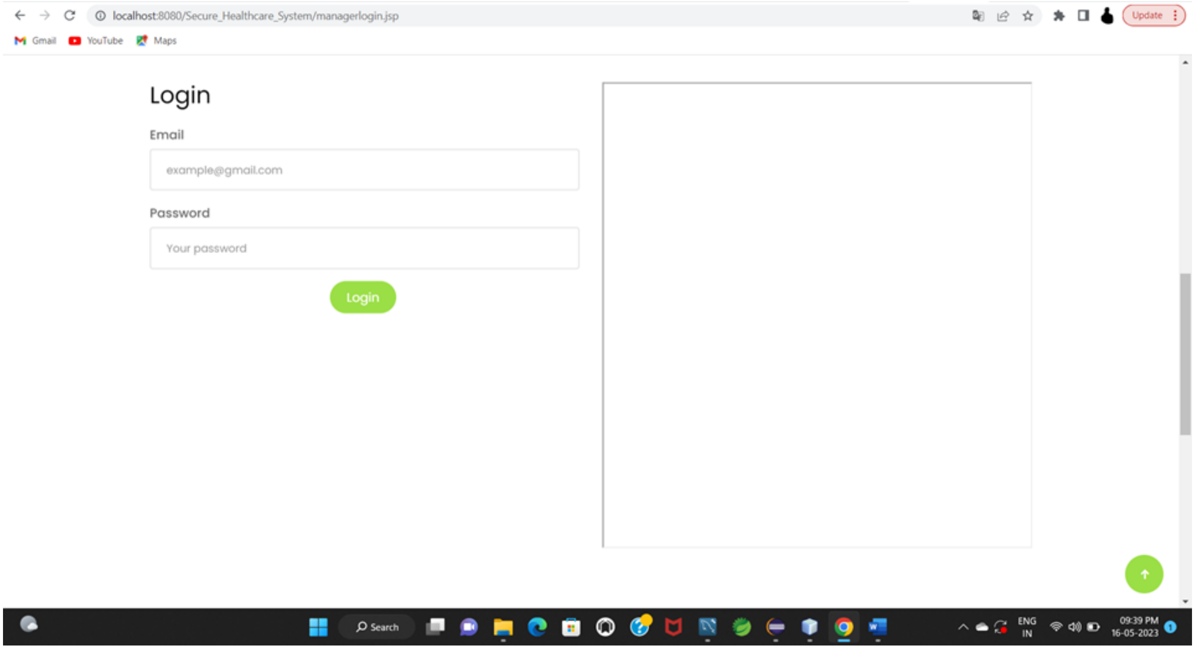
Phone: Your phone

Email: Your email address

Password: Your password

Address: Enter your address

3. Login



localhost:8080/Secure_Healthcare_System/managerlogin.jsp

Login

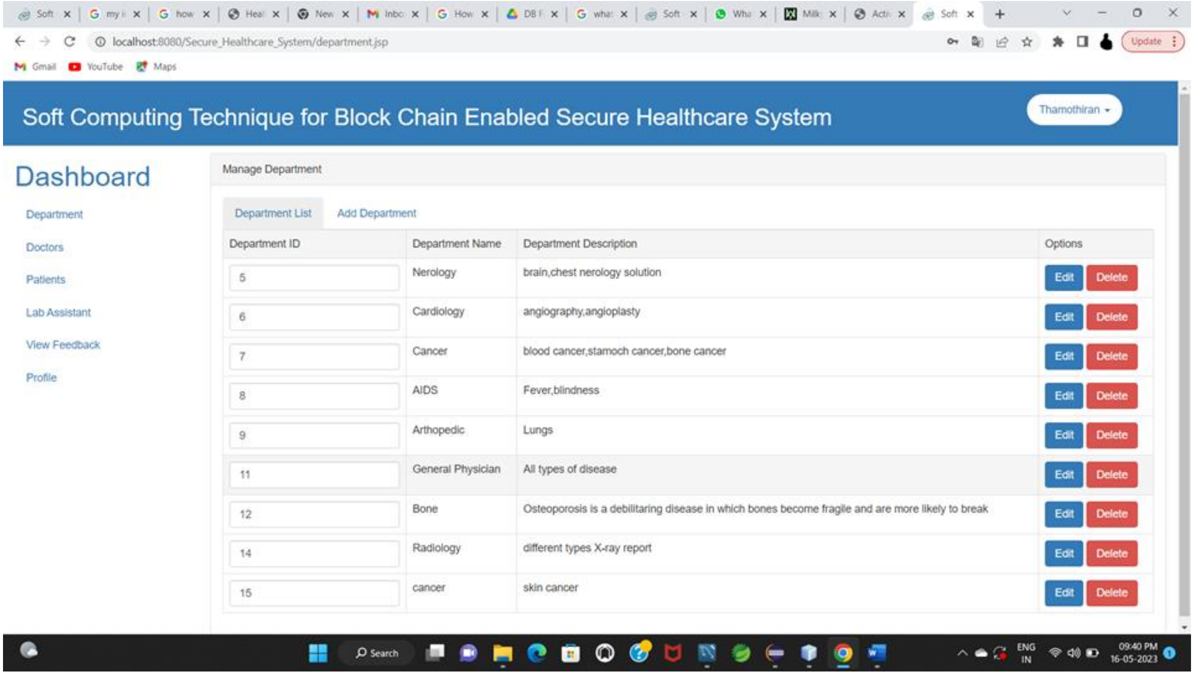
Email

example@gmail.com

Password

Your password

4. Options on the Manager login

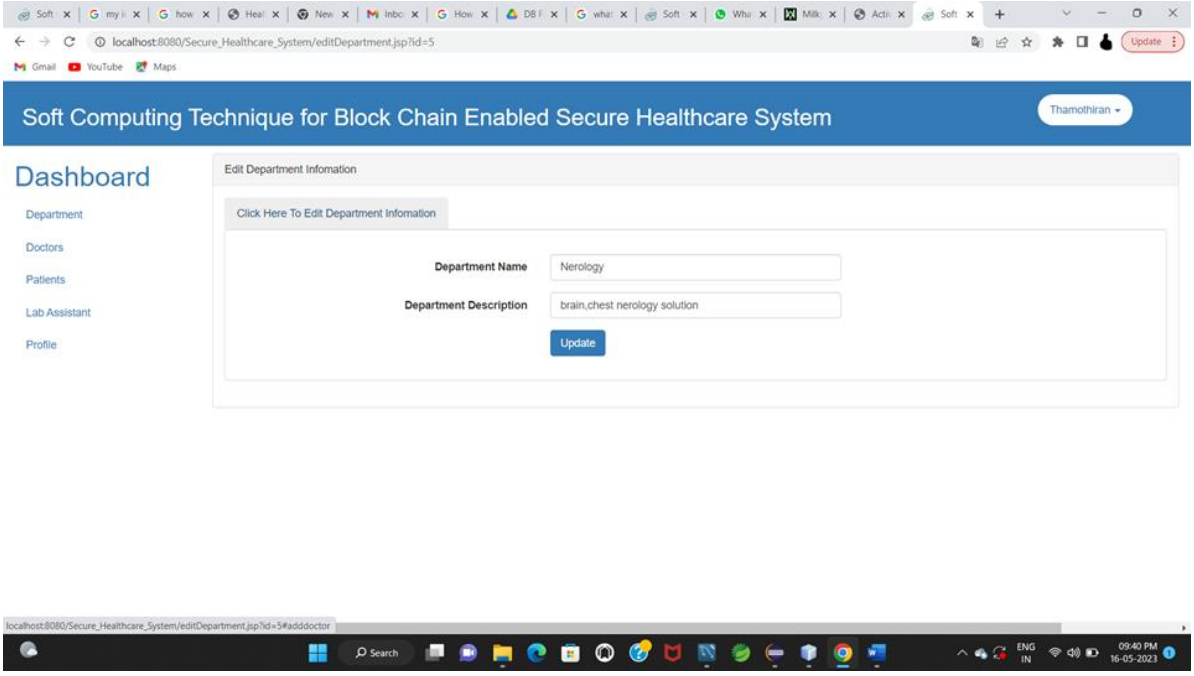


localhost:8080/Secure_Healthcare_System/department.jsp

Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Manage Department

Department ID	Department Name	Department Description	Options
5	Nerology	brain,chest nerology solution	Edit Delete
6	Cardiology	angiography,angioplasty	Edit Delete
7	Cancer	blood cancer,stamoch cancer,bone cancer	Edit Delete
8	AIDS	Fever,blindness	Edit Delete
9	Arthopedic	Lungs	Edit Delete
11	General Physician	All types of disease	Edit Delete
12	Bone	Osteoporosis is a debilitaring disease in which bones become fragile and are more likely to break	Edit Delete
14	Radiology	different types X-ray report	Edit Delete
15	cancer	skin cancer	Edit Delete



localhost:8080/Secure_Healthcare_System/editDepartment.jsp?id=5

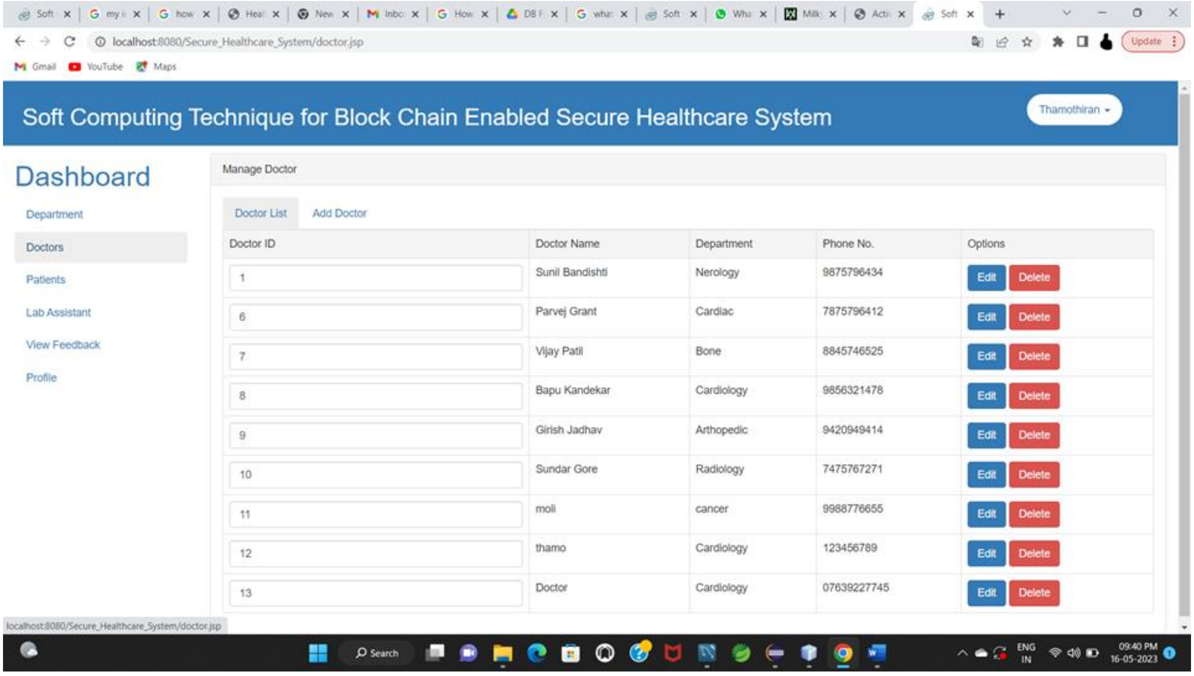
Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Edit Department Infomation

Click Here To Edit Department Infomation

Department Name	Nerology
Department Description	brain,chest nerology solution

Update
localhost:8080/Secure_Healthcare_System/editDepartment.jsp?id=5#adddoctor
09:40 PM
16-05-2023

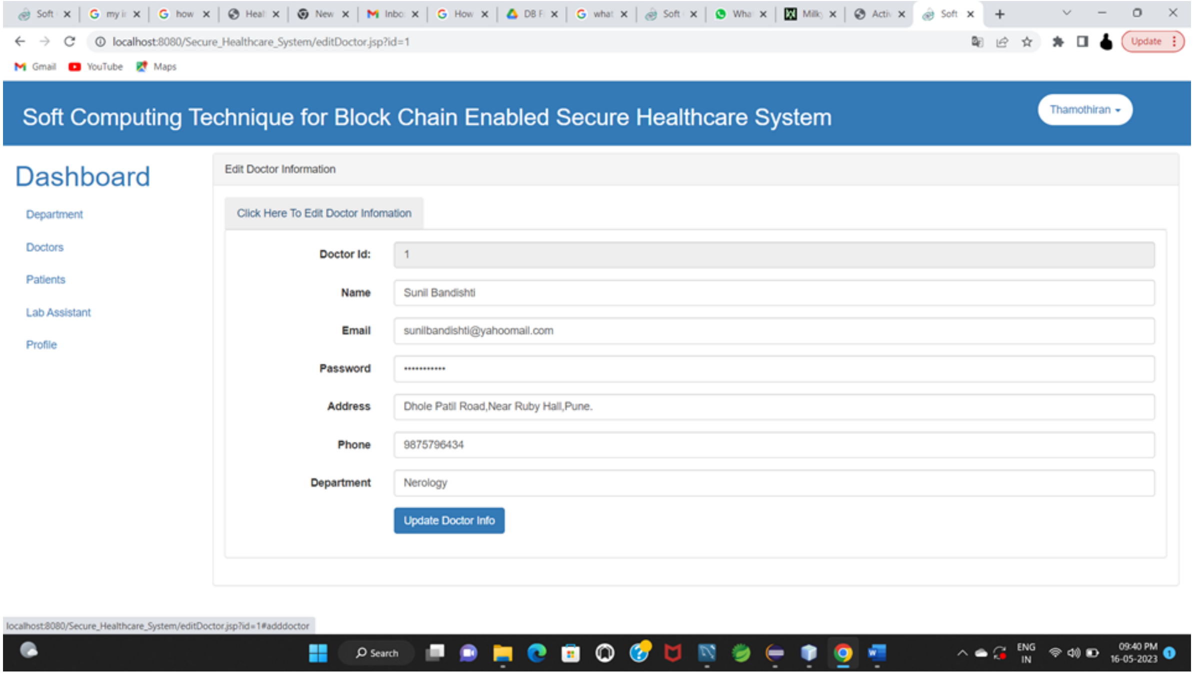


localhost:8080/Secure_Healthcare_System/doctor.jsp

Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Manage Doctor

Doctor ID	Doctor Name	Department	Phone No.	Options
1	Sunil Bandishti	Nerology	9875796434	Edit Delete
6	Parvej Grant	Cardiac	7875796412	Edit Delete
7	Vijay Patil	Bone	8845746525	Edit Delete
8	Bapu Kandekar	Cardiology	9856321478	Edit Delete
9	Girish Jadhav	Arthopedic	9420949414	Edit Delete
10	Sundar Gore	Radiology	7475767271	Edit Delete
11	moli	cancer	9988776655	Edit Delete
12	thamo	Cardiology	123456789	Edit Delete
13	Doctor	Cardiology	07639227745	Edit Delete

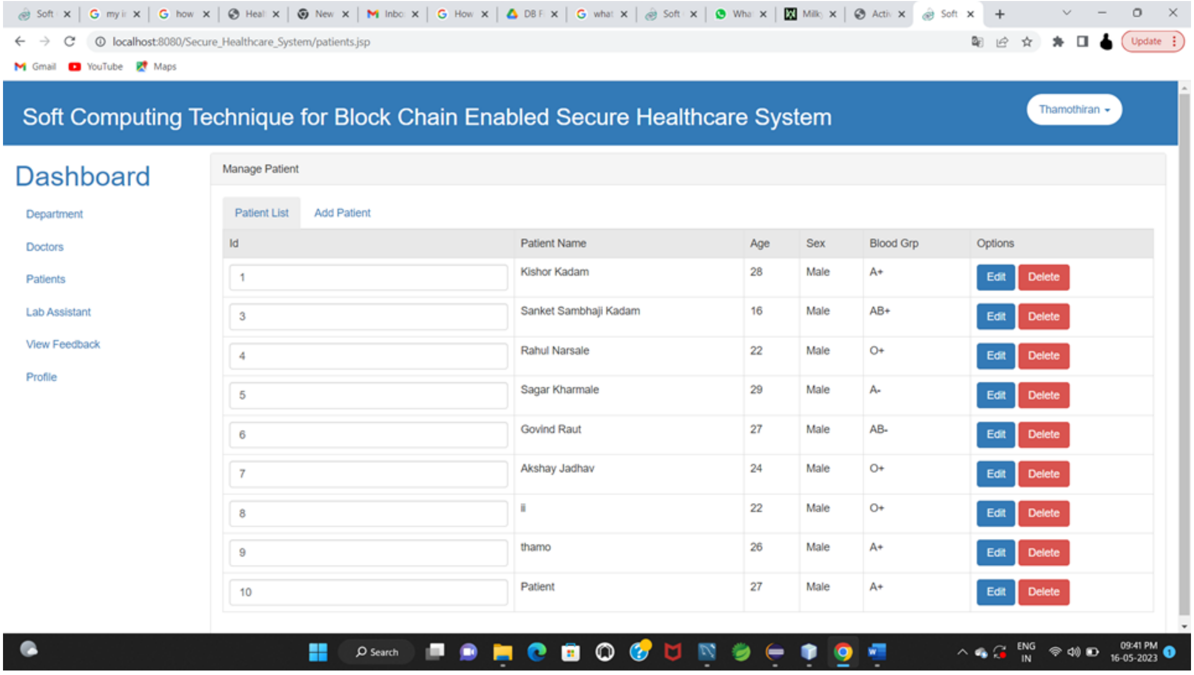

localhost:8080/Secure_Healthcare_System/editDoctor.jsp?id=1
Gmail
YouTube
Maps
Thamothiran
Dashboard
Department
Doctors
Patients
Lab Assistant
Profile
Update Doctor Info
localhost:8080/Secure_Healthcare_System/editDoctor.jsp?id=1#adddoctor
09:40 PM
16-05-2023

Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Edit Doctor Information

Click Here To Edit Doctor Infomation

Doctor Id:	1
Name	Sunil Bandishti
Email	sunilbandishti@yahoomail.com
Password	•••••••••••
Address	Dhole Patil Road,Near Ruby Hall,Pune.
Phone	9875796434
Department	Nerology

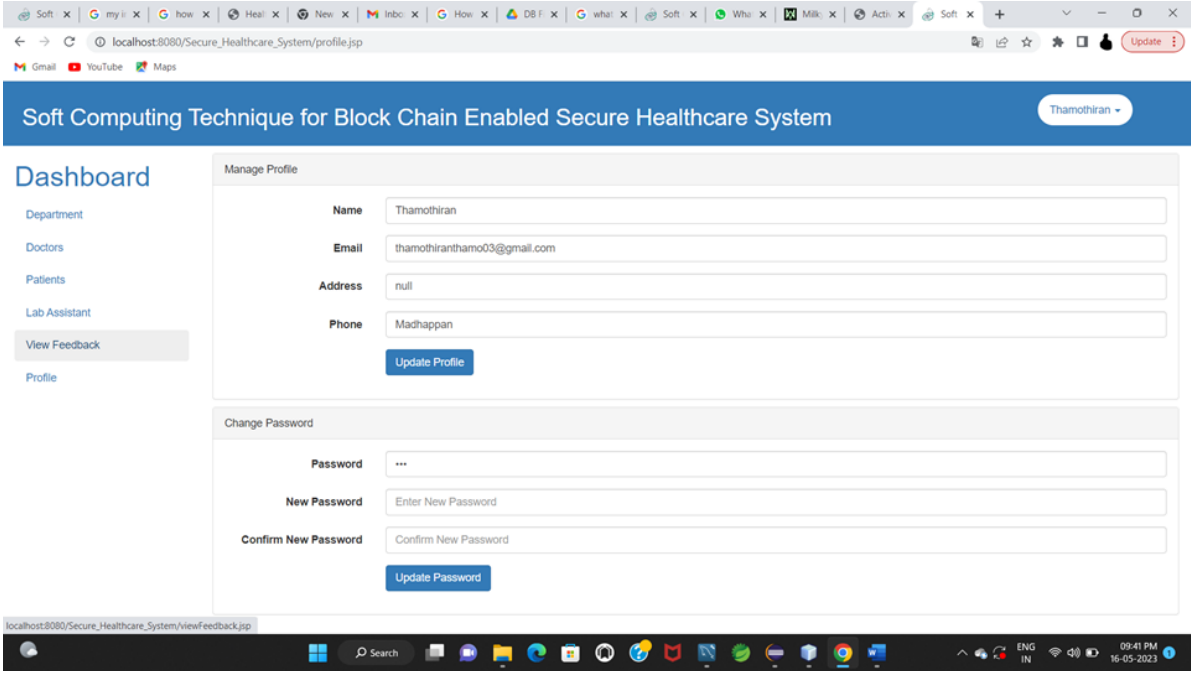


Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Manage Patient

Id	Patient Name	Age	Sex	Blood Grp	Options
1	Kishor Kadam	28	Male	A+	Edit Delete
3	Sanket Sambhaji Kadam	16	Male	AB+	Edit Delete
4	Rahul Narsale	22	Male	O+	Edit Delete
5	Sagar Kharmale	29	Male	A-	Edit Delete
6	Govind Raut	27	Male	AB-	Edit Delete
7	Akshay Jadhav	24	Male	O+	Edit Delete
8	ii	22	Male	O+	Edit Delete
9	thamo	26	Male	A+	Edit Delete
10	Patient	27	Male	A+	Edit Delete

5. Manage Profile



Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Dashboard

Manage Profile

Name: Thamothiran

Email: thamothiranthamo03@gmail.com

Address: null

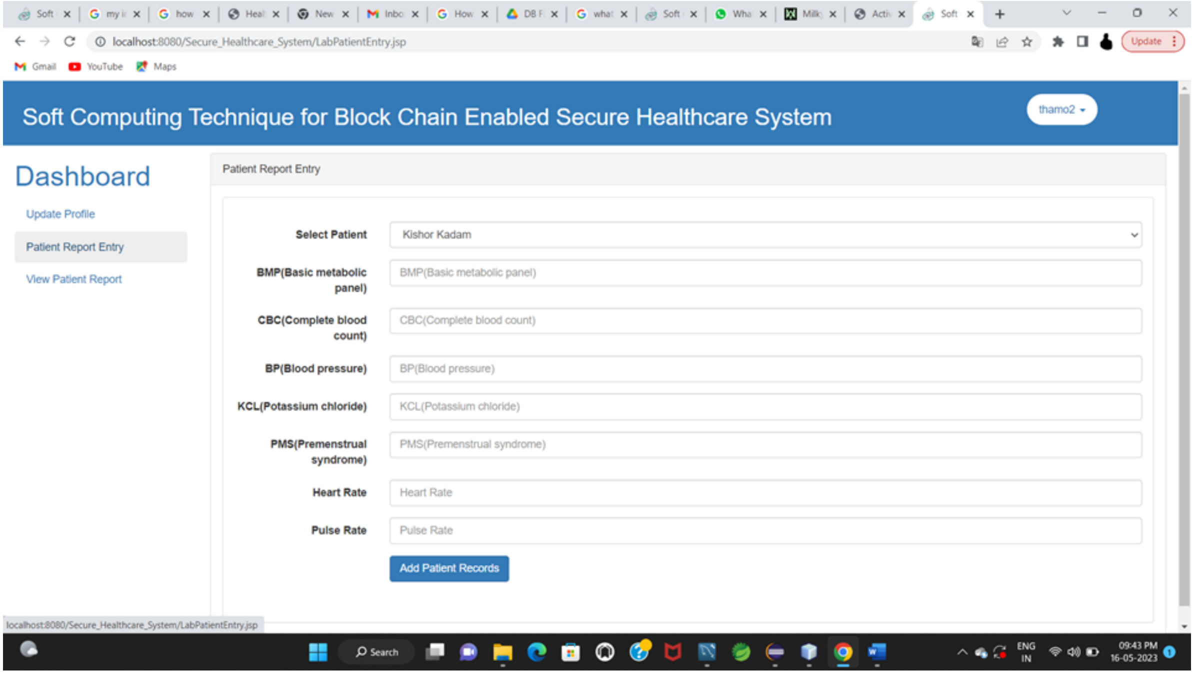
Phone: Madhappan

Change Password

Password: •••

New Password: Enter New Password

Confirm New Password: Confirm New Password



Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Dashboard

Patient Report Entry

Select Patient: Kishor Kadam

BMP(Basic metabolic panel): BMP(Basic metabolic panel)

CBC(Complete blood count): CBC(Complete blood count)

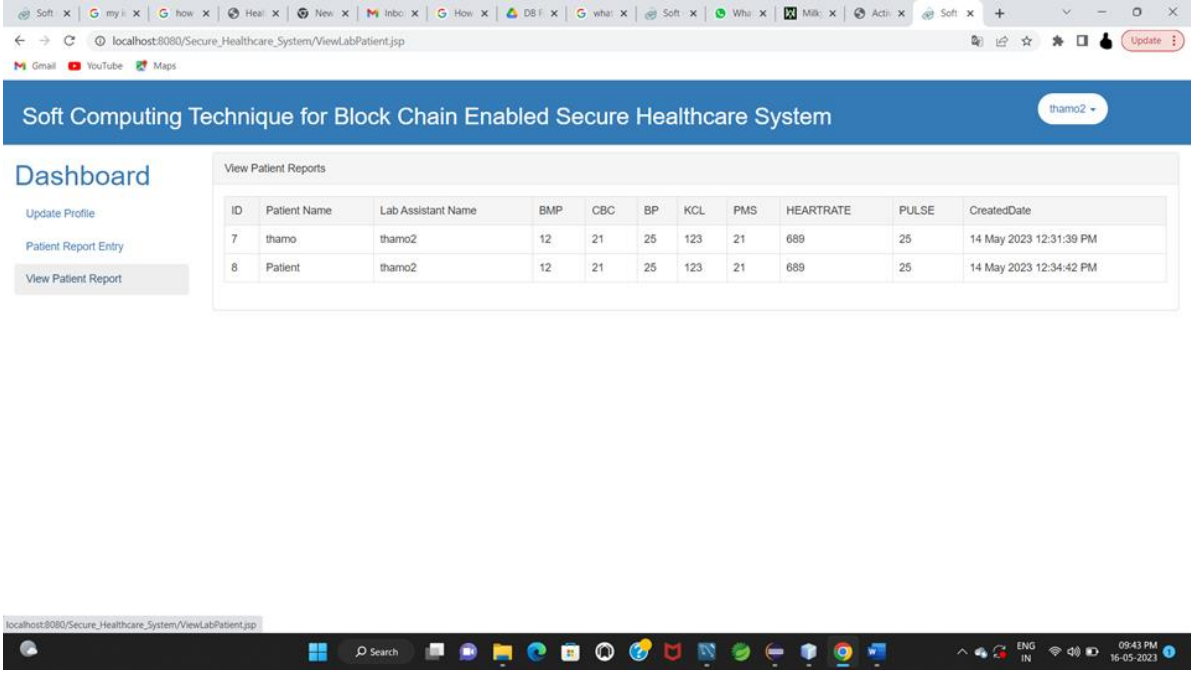
BP(Blood pressure): BP(Blood pressure)

KCL(Potassium chloride): KCL(Potassium chloride)

PMS(Premenstrual syndrome): PMS(Premenstrual syndrome)

Heart Rate: Heart Rate

Pulse Rate: Pulse Rate



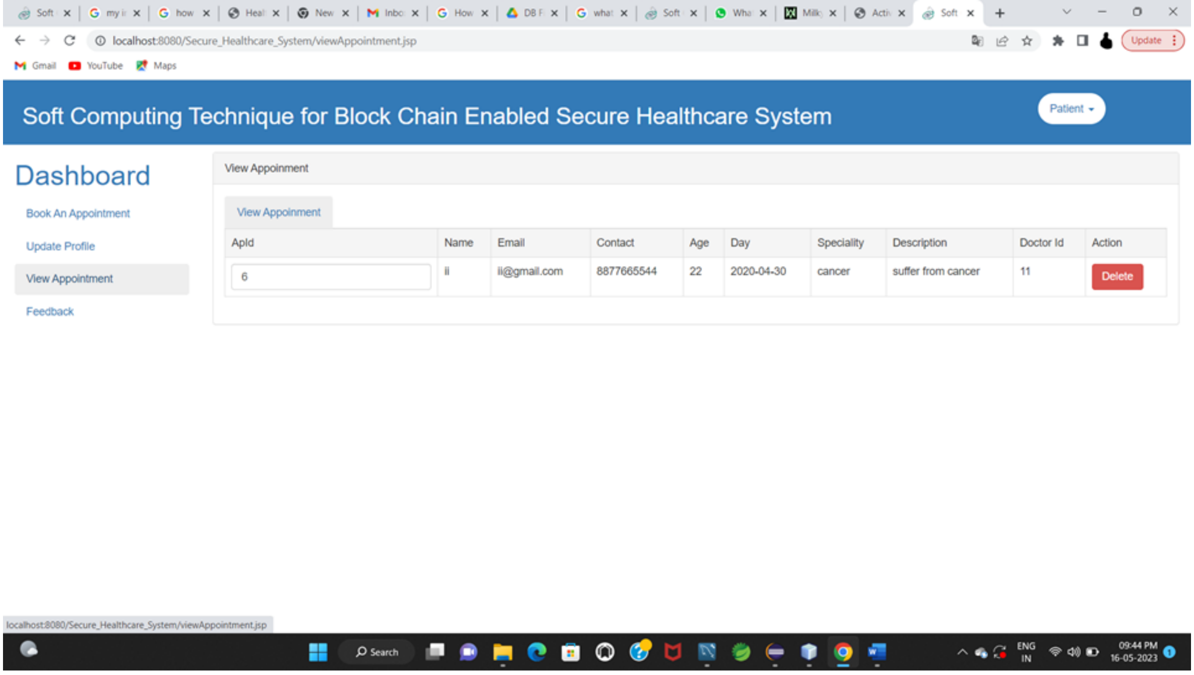
Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Dashboard

View Patient Reports

ID	Patient Name	Lab Assistant Name	BMP	CBC	BP	KCL	PMS	HEARTRATE	PULSE	CreatedDate
7	thamo	thamo2	12	21	25	123	21	689	25	14 May 2023 12:31:39 PM
8	Patient	thamo2	12	21	25	123	21	689	25	14 May 2023 12:34:42 PM

6. Patient Login

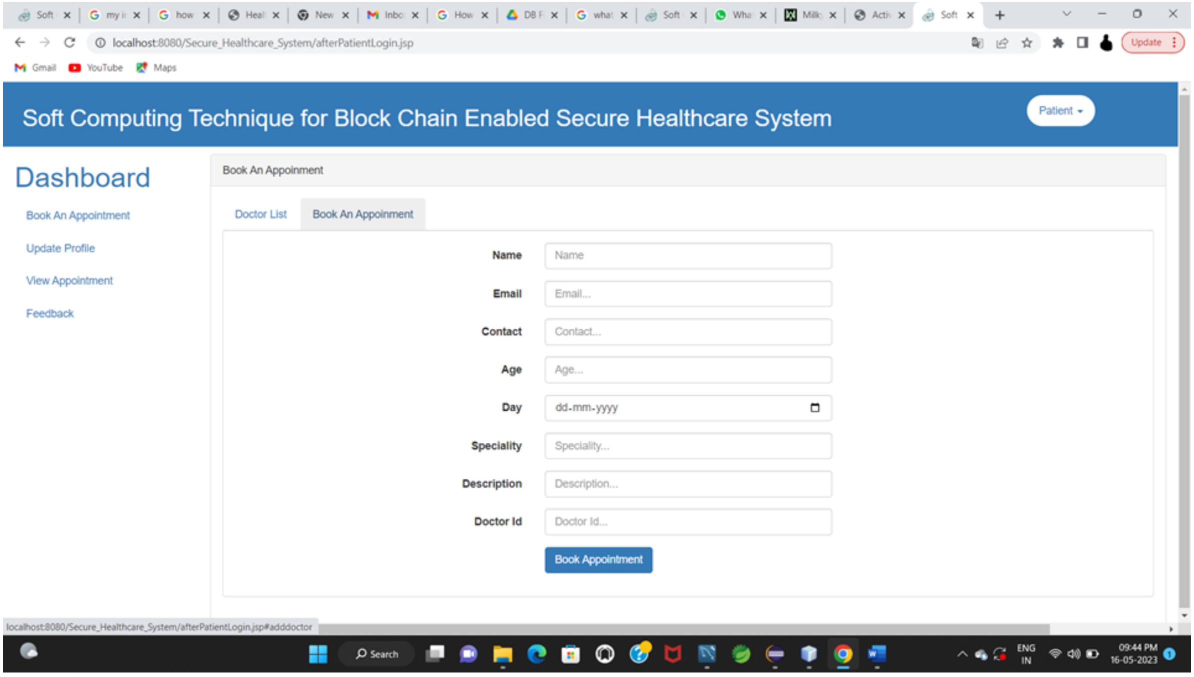


Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Dashboard

View Appoinment

ApId	Name	Email	Contact	Age	Day	Speciality	Description	Doctor Id	Action
6	ii	ii@gmail.com	8877665544	22	2020-04-30	cancer	suffer from cancer	11	Delete



localhost:8080/Secure_Healthcare_System/afterPatientLogin.jsp
Gmail
YouTube
Maps
Patient

Soft Computing Technique for Block Chain Enabled Secure Healthcare System

Dashboard

- Book An Appointment
- Update Profile
- View Appointment
- Feedback

Book An Appoinment

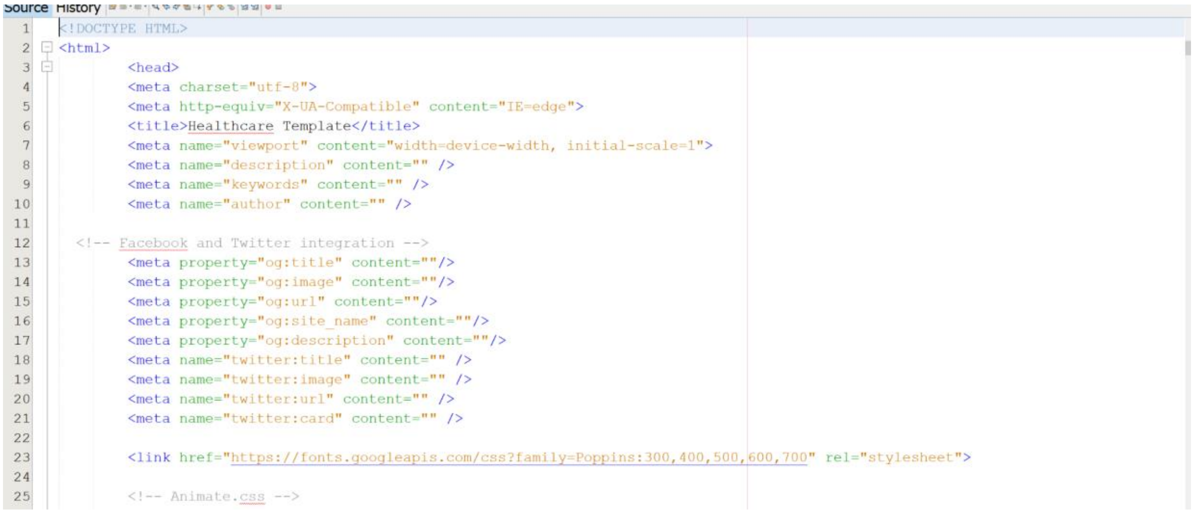
Doctor List | Book An Appointment

Name	Name
Email	Email...
Contact	Contact...
Age	Age...
Day	dd-mm-yyyy
Speciality	Speciality...
Description	Description...
Doctor Id	Doctor Id...

Book Appointment

Code snippets:

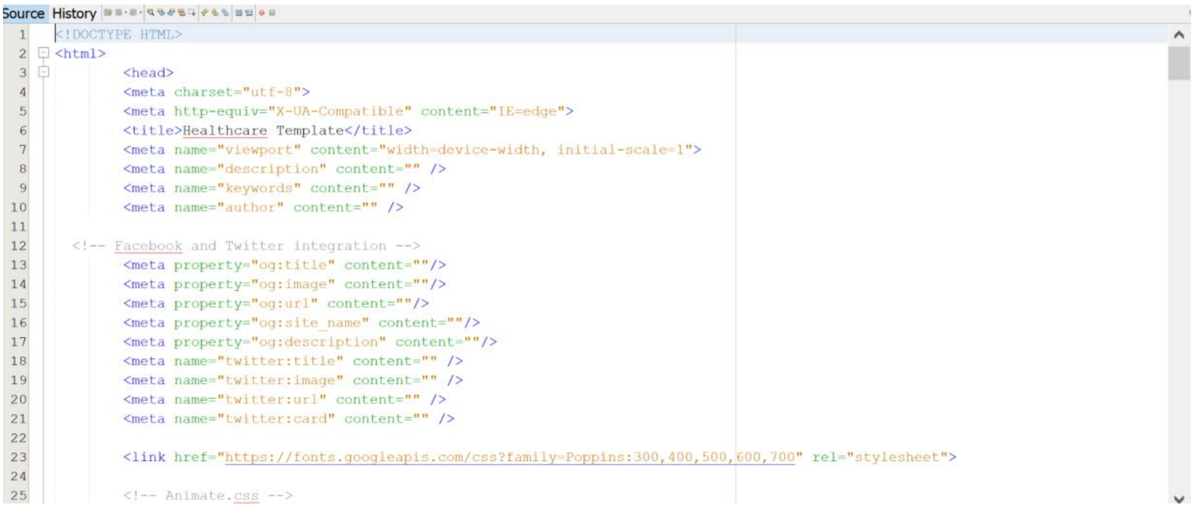
Index.html



```
<!DOCTYPE HTML>
<html>
        <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title>Healthcare Template</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="description" content="" />
        <meta name="keywords" content="" />
        <meta name="author" content="" />

  <!-- Facebook and Twitter integration -->
        <meta property="og:title" content=""/>
        <meta property="og:image" content=""/>
        <meta property="og:url" content=""/>
        <meta property="og:site_name" content=""/>
        <meta property="og:description" content=""/>
        <meta name="twitter:title" content="" />
        <meta name="twitter:image" content="" />
        <meta name="twitter:url" content="" />
        <meta name="twitter:card" content="" />

        <link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700" rel="stylesheet">

        <!-- Animate.css -->
```

Fig: Code snippet of index page

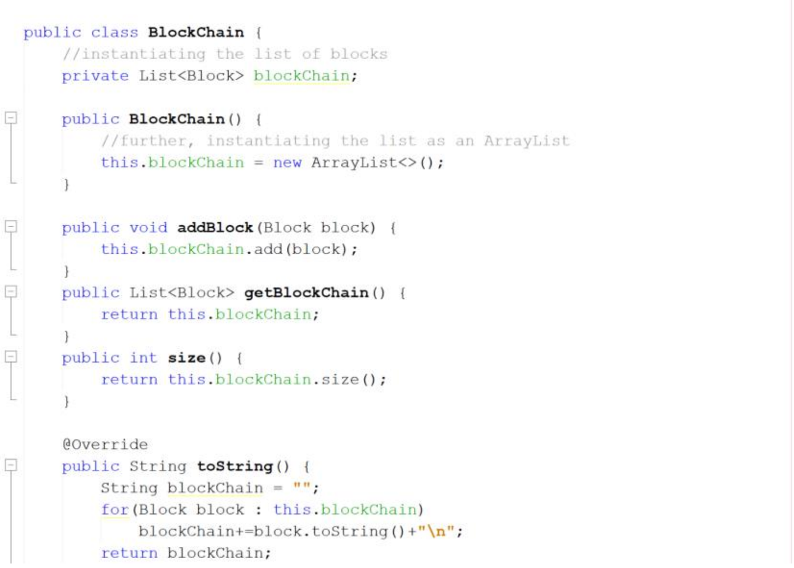
Doctor login page(doctor.html):



```html
<!DOCTYPE HTML>
<html>
          <head>
          <meta charset="utf-8">
          <meta http-equiv="X-UA-Compatible" content="IE=edge">
          <title>Healthcare Template</title>
          <meta name="viewport" content="width=device-width, initial-scale=1">
          <meta name="description" content="" />
          <meta name="keywords" content="" />
          <meta name="author" content="" />

  <!-- Facebook and Twitter integration -->
          <meta property="og:title" content=""/>
          <meta property="og:image" content=""/>
          <meta property="og:url" content=""/>
          <meta property="og:site_name" content=""/>
          <meta property="og:description" content=""/>
          <meta name="twitter:title" content="" />
          <meta name="twitter:image" content="" />
          <meta name="twitter:url" content="" />
          <meta name="twitter:card" content="" />

          <link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700" rel="stylesheet">

          <!-- Animate.css -->
```

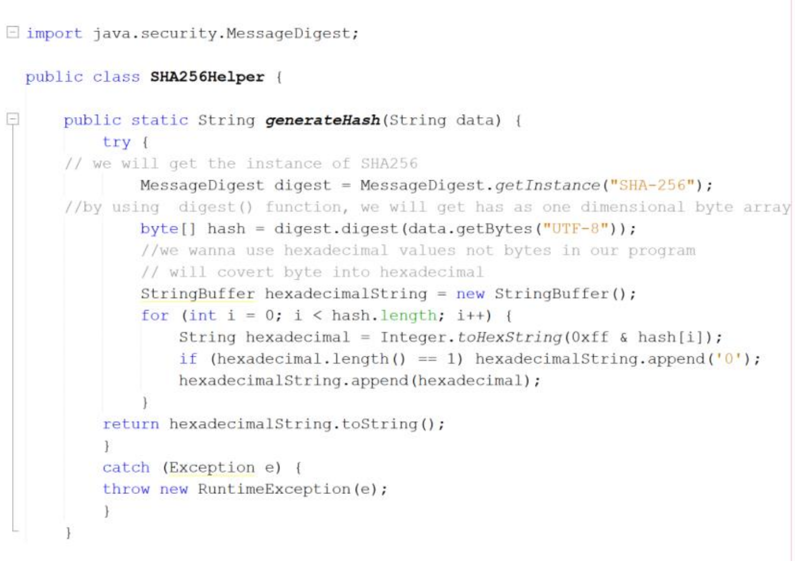
Fig: Code snippet of Doctor login page

Blockchain.java:



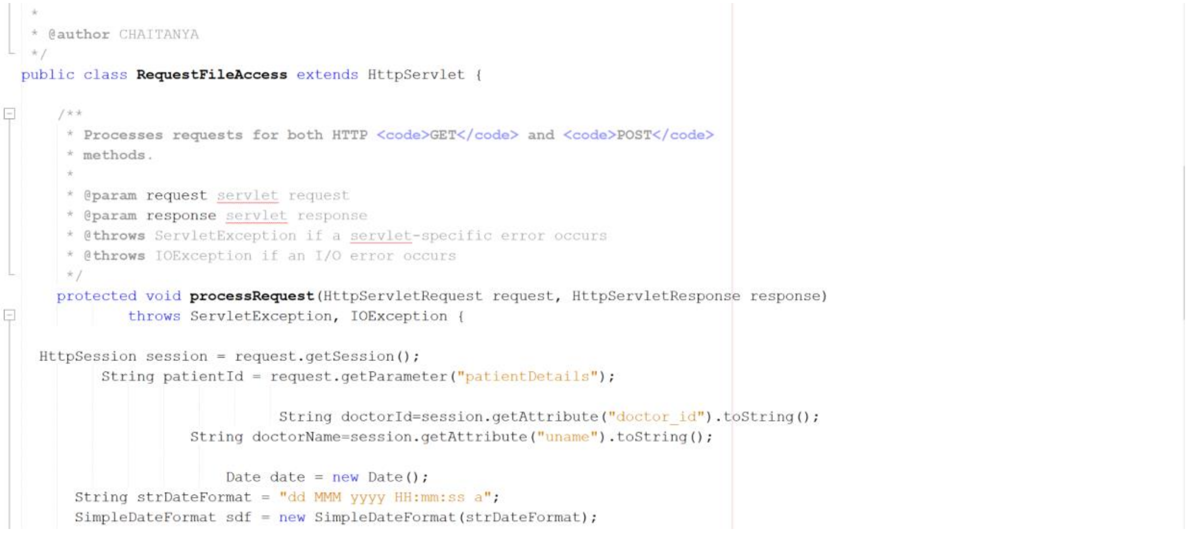
```java
public class BlockChain {
    //instantiating the list of blocks
    private List<Block> blockChain;

    public BlockChain() {
        //further, instantiating the list as an ArrayList
        this.blockChain = new ArrayList<>();
    }

    public void addBlock(Block block) {
        this.blockChain.add(block);
    }
    public List<Block> getBlockChain() {
        return this.blockChain;
    }
    public int size() {
        return this.blockChain.size();
    }

    @Override
    public String toString() {
        String blockChain = "";
        for(Block block : this.blockChain)
            blockChain+=block.toString()+"\n";
        return blockChain;
```

Sha256hepher.java:



```java
import java.security.MessageDigest;

public class SHA256Helper {

    public static String generateHash(String data) {
        try {
    // we will get the instance of SHA256
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
    //by using  digest() function, we will get has as one dimensional byte array
            byte[] hash = digest.digest(data.getBytes("UTF-8"));
            //we wanna use hexadecimal values not bytes in our program
            // will covert byte into hexadecimal
            StringBuffer hexadecimalString = new StringBuffer();
            for (int i = 0; i < hash.length; i++) {
                String hexadecimal = Integer.toHexString(0xff & hash[i]);
                if (hexadecimal.length() == 1) hexadecimalString.append('0');
                hexadecimalString.append(hexadecimal);
            }
        return hexadecimalString.toString();
        }
        catch (Exception e) {
        throw new RuntimeException(e);
        }
    }
```

Requesting to view patient datadoctorrequest.java):



```java
 *
 * @author CHAITANYA
 */
public class RequestFileAccess extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

  HttpSession session = request.getSession();
        String patientId = request.getParameter("patientDetails");

                          String doctorId=session.getAttribute("doctor_id").toString();
                  String doctorName=session.getAttribute("uname").toString();

                      Date date = new Date();
      String strDateFormat = "dd MMM yyyy HH:mm:ss a";
      SimpleDateFormat sdf = new SimpleDateFormat(strDateFormat);
```

MD5.java:

```java
/*

 * To change this license header, choose License Headers in Project Properties.

 * To change this template file, choose Tools | Templates

 * and open the template in the editor.

 */

package com.block;
```

```
import java.math.BigInteger;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;



// Java program to calculate MD5 hash value

public class MD5 {

    public static String getMd5(String input)

    {

        try {



            // Static getInstance method is called with hashing MD5

            MessageDigest md = MessageDigest.getInstance("MD5");



            // digest() method is called to calculate message digest

            //  of an input digest() return array of byte

            byte[] messageDigest = md.digest(input.getBytes());



            // Convert byte array into signum representation

            BigInteger no = new BigInteger(1, messageDigest);



            // Convert message digest into hex value

            String hashtext = no.toString(16);
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }
            return hashtext;
        }

        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }

}
```

Doctor_DATA_ATTRIBUTE(doctorDAO.java):

```
package com.doctor;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import com.bean.Doctor;
import com.connection.ConnectionProvider;


public class DoctorDao {
	public static Doctor getRecordById(int id) {
		Doctor d = null;
		try {
			Connection con = ConnectionProvider.getCon();
			PreparedStatement ps = con.prepareStatement("select * from doctor
where doctor_id=?");
			ps.setInt(1, id);
			ResultSet rs = ps.executeQuery();
			while (rs.next()) {
				d = new Doctor();
				d.setDoctor_id(rs.getInt("doctor_id"));
				d.setDoctname(rs.getString("doc_name"));
				d.setEmail(rs.getString("email"));
				d.setPwd(rs.getString("password"));
				d.setAdd(rs.getString("address"));
				d.setPhon(rs.getString("phone"));
				d.setDept(rs.getString("department"));
			}
		} catch (Exception e) {
			System.out.println(e);
```

```
                }

                return d;

        }

}
```

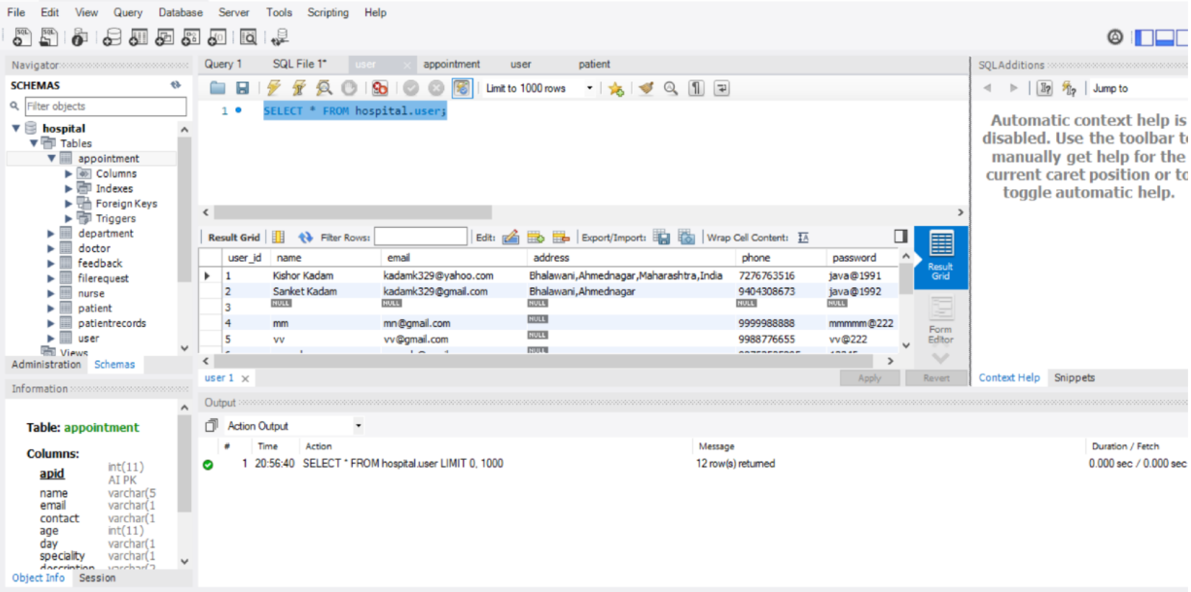
Log_in_Data Attribute Object(loginDao):

```
package com.user;


import java.io.IOException;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;


import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


import com.connection.ConnectionProvider;
```

```
/**
 * Servlet implementation class LoginDao
 */
@WebServlet("/LoginDao")
public class LoginDao extends HttpServlet {
		private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginDao() {
        super();
        // TODO Auto-generated constructor stub
    }

		/**
		 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
		 */
		protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
				// TODO Auto-generated method stub
```

```
        response.getWriter().append("Served at: ").append(request.getContextPath());

    }



    /**

     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)

     */

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

        // TODO Auto-generated method stub




        // TODO Auto-generated method stub

                String email = request.getParameter("email");

                String password = request.getParameter("password");

                try {

                    HttpSession hs = request.getSession();

                    Connection con = ConnectionProvider.getCon();

                    Statement stmt = con.createStatement();

                    ResultSet rs = stmt.executeQuery("select
name,email,password from user where email='" + email + "'and password='" + password +
"'");

                    if (rs.next()) {

                        hs.setAttribute("uname", rs.getString(1));

                        response.sendRedirect("department.jsp");

                    } else {

                        response.sendRedirect("index.html");

                    }

                } catch (Exception e) {

                    e.printStackTrace();

                }

        doGet(request, response);

    }


}
```

Table of user their information stored in database in tables:



```
SELECT * FROM hospital.user;
```

user_id	name	email	address	phone	password
1	Kishor Kadam	kadamk329@yahoo.com	Bhalawani,Ahmednagar,Maharashtra,India	7276763516	java@1991
2	Sanket Kadam	kadamk329@gmail.com	Bhalawani,Ahmednagar	9404308673	java@1992
3	NULL	NULL	NULL	NULL	NULL
4	mm	mn@gmail.com	NULL	9999988888	mmmmm@222
5	vv	vv@gmail.com	NULL	9988776655	vv@222

```
#  Time      Action                                     Message             Duration / Fetch
1  20:56:40  SELECT * FROM hospital.user LIMIT 0, 1000  12 row(s) returned  0.000 sec / 0.000 sec
```

Table: appointment

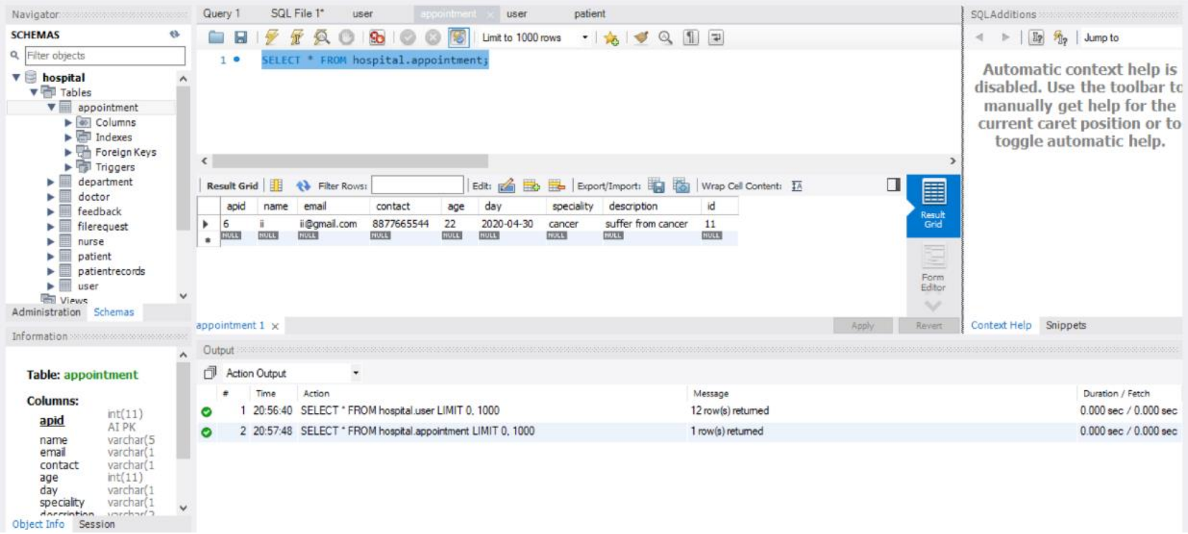
Columns:

apid	int(11) AI PK
name	varchar(5
email	varchar(1
contact	varchar(1
age	int(11)
day	varchar(1
speciality	varchar(1

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Fig: Table of User Details

Appointment table:



```
SELECT * FROM hospital.appointment;
```

apid	name	email	contact	age	day	speciality	description	id
6	ii	ii@gmail.com	8877665544	22	2020-04-30	cancer	suffer from cancer	11
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                            Message             Duration / Fetch
1  20:56:40  SELECT * FROM hospital.user LIMIT 0, 1000         12 row(s) returned  0.000 sec / 0.000 sec
2  20:57:48  SELECT * FROM hospital.appointment LIMIT 0, 1000  1 row(s) returned   0.000 sec / 0.000 sec
```

Table: appointment

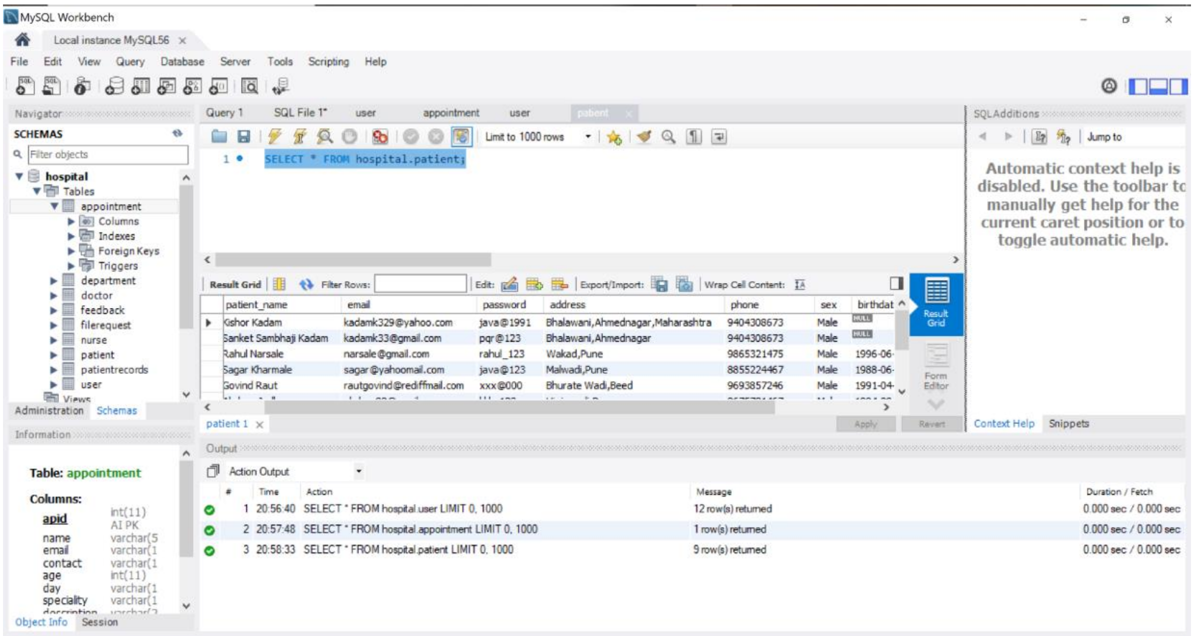
Columns:

apid	int(11) AI PK
name	varchar(5
email	varchar(1
contact	varchar(1
age	int(11)
day	varchar(1
speciality	varchar(1

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Fig: Table of hospital Appointment Details

Data stored in Patient table in Mysql workbench:



```
SELECT * FROM hospital.patient;
```

patient_name	email	password	address	phone	sex	birthdat
Kishor Kadam	kadamk329@yahoo.com	java@1991	Bhalawani,Ahmednagar,Maharashtra	9404308673	Male	NULL
Sanket Sambhaji Kadam	kadamk33@gmail.com	pqr@123	Bhalawani,Ahmednagar	9404308673	Male	NULL
Rahul Narsale	narsale@gmail.com	rahul_123	Wakad,Pune	9865321475	Male	1996-06-
Sagar Kharmale	sagar@yahoomail.com	java@123	Malwadi,Pune	8855224467	Male	1988-06-
Govind Raut	rautgovind@rediffmail.com	xxx@000	Bhurate Wadi,Beed	9693857246	Male	1991-04-

#	Time	Action	Message	Duration / Fetch
1	20:56:40	SELECT * FROM hospital.user LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
2	20:57:48	SELECT * FROM hospital.appointment LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
3	20:58:33	SELECT * FROM hospital.patient LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec

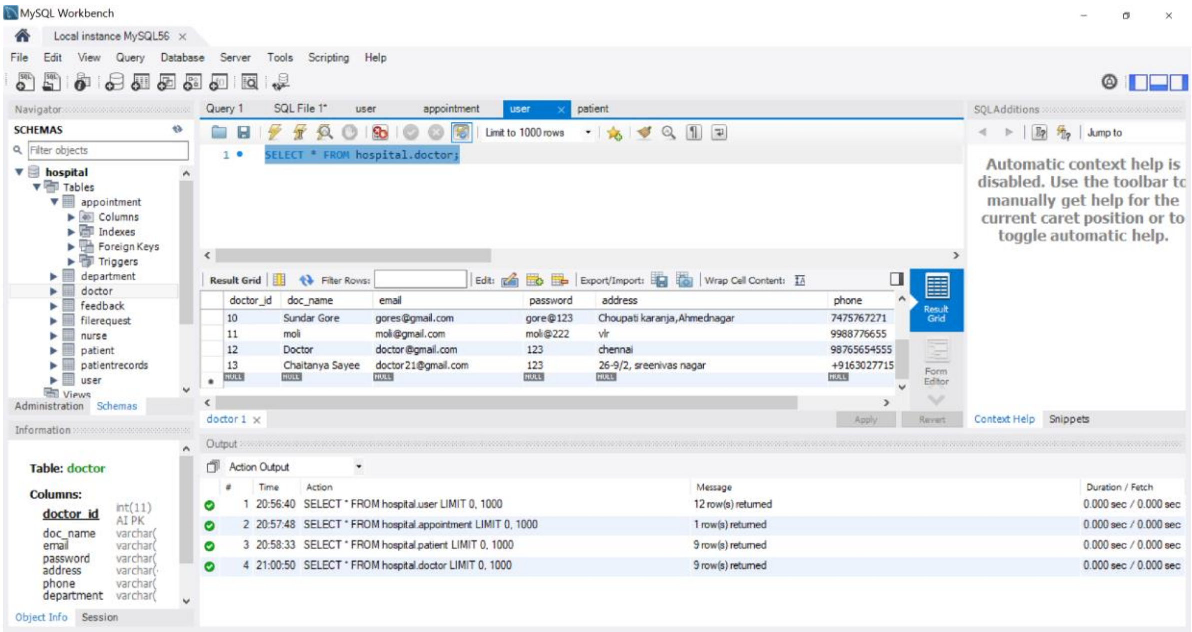
Table: appointment

Columns:

apid	int(11) AI PK
name	varchar(5
email	varchar(1
contact	varchar(1
age	int(11)
day	varchar(1
specialty	varchar(1

Fig: Table of patients

Doctor working in the institute and their data stored in the table(includes their credentials):



```
SELECT * FROM hospital.doctor;
```

doctor_id	doc_name	email	password	address	phone
10	Sundar Gore	gores@gmail.com	gore@123	Choupati karanja,Ahmednagar	7475767271
11	moli	moli@gmail.com	moli@222	vlr	9988776655
12	Doctor	doctor@gmail.com	123	chennai	98765654555
13	Chaitanya Sayee	doctor21@gmail.com	123	26-9/2, sreenivas nagar	+9163027715
NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	20:56:40	SELECT * FROM hospital.user LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
2	20:57:48	SELECT * FROM hospital.appointment LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
3	20:58:33	SELECT * FROM hospital.patient LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
4	21:00:50	SELECT * FROM hospital.doctor LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec

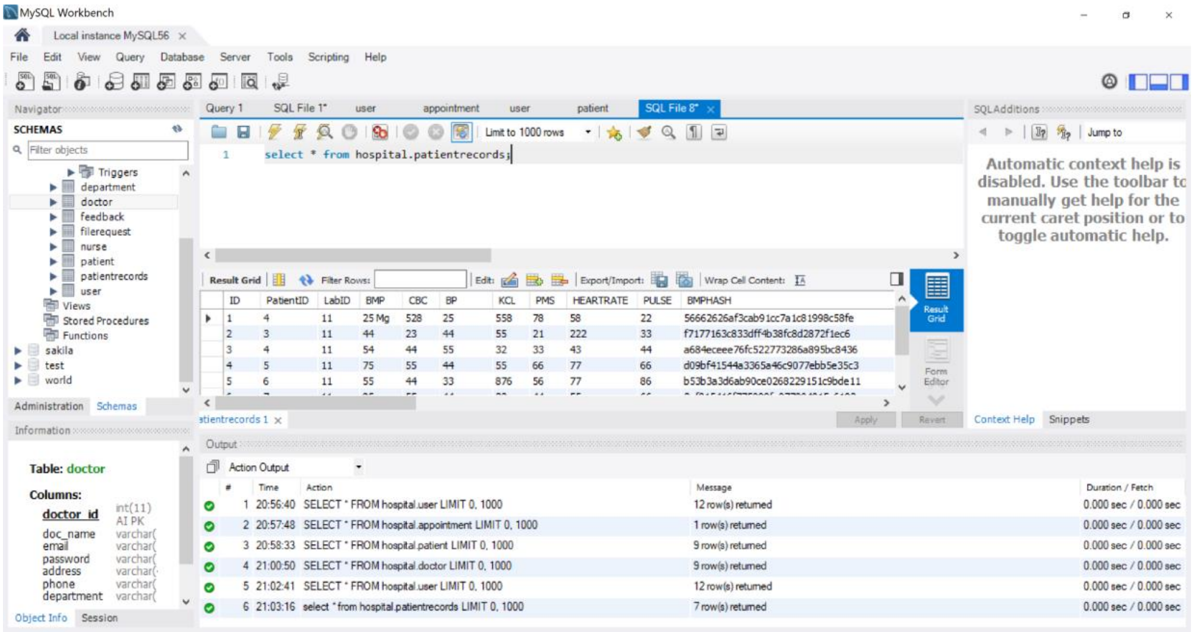
Table: doctor

Columns:

doctor_id	int(11) AI PK
doc_name	varchar(
email	varchar(
password	varchar(
address	varchar(
phone	varchar(
department	varchar(

Fig: Table of Doctors

Patient records which shows hash values:



```
select * from hospital.patientrecords;
```

ID	PatientID	LabID	BMP	CBC	BP	KCL	PMS	HEARTRATE	PULSE	BMPHASH
1	4	11	25 Mg	528	25	558	78	58	22	56662626af3cab91cc7a1c81998c58fe
2	3	11	44	23	44	55	21	222	33	f7177163c833dff4b38fc8d2872f1ec6
3	4	11	54	44	55	32	33	43	44	a684eceee76fc522773286a895bc8436
4	5	11	75	55	44	55	66	77	66	d09bf41544a3365a46c9077ebb5e35c3
5	6	11	55	44	33	876	56	77	86	b53b3a3d6ab90ce0268229151c9bde11

```
#  Time      Action                                                Message            Duration / Fetch
1  20:56:40  SELECT * FROM hospital.user LIMIT 0, 1000              12 row(s) returned  0.000 sec / 0.000 sec
2  20:57:48  SELECT * FROM hospital.appointment LIMIT 0, 1000       1 row(s) returned   0.000 sec / 0.000 sec
3  20:58:33  SELECT * FROM hospital.patient LIMIT 0, 1000           9 row(s) returned   0.000 sec / 0.000 sec
4  21:00:50  SELECT * FROM hospital.doctor LIMIT 0, 1000            9 row(s) returned   0.000 sec / 0.000 sec
5  21:02:41  SELECT * FROM hospital.user LIMIT 0, 1000              12 row(s) returned  0.000 sec / 0.000 sec
6  21:03:16  select * from hospital.patientrecords LIMIT 0, 1000    7 row(s) returned   0.000 sec / 0.000 sec
```

Table: doctor

Columns:

doctor_id	int(11) AI PK
doc_name	varchar(
email	varchar(
password	varchar(
address	varchar(
phone	varchar(
department	varchar(

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Fig:Table of patient records