

CONFIDENTIALITY OF PATIENT MEDICAL RECORDS

A PROJECT REPORT

Submitted by,

B.L.N.S. LAHARI

-20201CSE0370

SD. AMRUTHAVALLI

-20201CSE0386

LINGUTLA THANUSHA

-20201CSE0416

Under the guidance of,

Dr. PRAKASH SHANMURTHY

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At



PRESIDENCY UNIVERSITY

BENGALURU

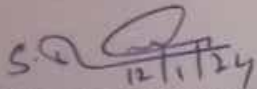
JANUARY 2024

PRESIDENCY UNIVERSITY

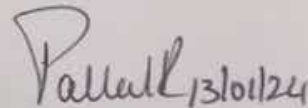
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

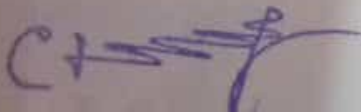
This is to certify that the Project report “**CONFIDENTIALITY OF PATIENT MEDICAL RECORDS**” being submitted by **B.L.N.S.LAHARI, SD.AMRUTHAVALLI, LINGUTLA THANUSHA** bearing roll number(s) **20201CSE0370, 20201CSE0386, 20201CSE0416** in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science & Engineering is a bonafide work carried out under my supervision.



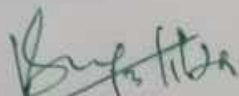
Dr. PRAKASH SHANMURTHY
Assistant Professor
School of CSE
Presidency University



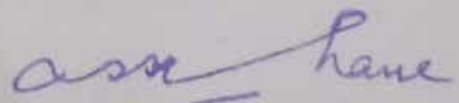
Dr. PALLAVI
Associate Professor & HoD
School of CSE
Presidency University



Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University



Dr. L. SHAKKEERA
Associate Dean
School of CSE&IS
Presidency University



Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Confidentiality of Patient Medical Records** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Prakash Shanmurthy, Assistant Professor, School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME

B.L.N.S.Lahari

SD.Amruthavalli

Lingutla Thanusha

ROLL NO

20201CSE0370

20201CSE0386

20201CSE0416

SIGNATURE

B.L.N.S.Lahari

SD

L-Thanusha

ABSTRACT

As the integration Personal Medical Records and cloud computing transforms the management of health information, the focus is on storing patient information. This document present about framework designed to improve the safeguarding and confidentiality of confidential information. Based on Attribute-Based Encryption with Policy-Controlled Ciphertext Access and Proxy-Based Data Transformation for Encryption our method allows providers to not only define the process easily manage them, but also the content. Control payload using traditional encryption techniques. Incorporating a policy definition process allows for careful investigation and provides a comprehensive accounting process for changes in access policy. Through performance evaluation, our system demonstrates good performance and reliability by creating the ability to pose threats against unauthorized access, enabling patient data private healthcare to improve in the challenging health environment.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science and Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L**, School of Computer Science and Engineering, Presidency University and **Dr. Pallavi**, Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr. Prakash Shanmurthy**, School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators **Mr. Zia Ur Rahman, Mr. Peniel John Whistely**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

B.L.N.S.LAHARI

S D.AMRUTHAVALLI

LINGUTLA THANUSHA

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 1	Test Cases	19
2	Table 2	Model Building	19

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 1	Workflow of Proposed System	8
2	Figure 2.1	Use Case Diagram	11
3	Figure 2.2	Class Diagram	11
4	Figure 2.3	Sequence Diagram	12
5	Figure 2.4	Collaboration Diagram	12
6	Figure 2.5	Deployment Diagram	13
7	Figure 2.6	Activity Diagram	13
8	Figure 2.7	Component Diagram	14
9	Figure 2.8	ER Diagram	14
10	Figure 2.9	DFD Diagram	15
11	Figure 3.1	Home Page	32
12	Figure 3.2	Module Page	32
13	Figure 3.3	Patient Login Form	32
14	Figure 3.4	Patient Registration Page	33
15	Figure 3.5	Appointment Form	33
16	Figure 3.6	Doctor Login	33
17	Figure 3.7	Doctor Registration	34
18	Figure 3.8	Doc Home Page	34
19	Figure 3.9	View Appointments	34
20	Figure 3.10	Upload Files	35
21	Figure 3.11	View Files	35
22	Figure 3.12	View Report	35
23	Figure 3.13	Management Login	36
24	Figure 3.14	Doctor Requests	36
25	Figure 3.15	Proxy Login	37
26	Figure 3.16	View Request	37
27	Figure 3.17	Authority Login	38
28	Figure 3.18	Authority Request	38

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	ACKNOWLEDGMENT	v
	LIST OF TABLES	vi
	LIST OF FIGURES	vii
1.	INTRODUCTION	1-3
	1.1 Motivation	1
	1.2 Problem Statement	1
	1.3 Objective of the Project	1
	1.4 Scope	2
	1.5 Project Introduction	2-3
2.	LITERATURE REVIEW	4-5
	2.1 Advantages	4
	2.2 Disadvantages	5
3.	RESEARCH GAPS OF EXISTING METHODS	6-7
4.	PROPOSED METHODOLOGY	8
	4.1 Proposed System	8
	4.2 Workflow of Proposed System	8
5.	OBJECTIVES	9
6.	SYSTEM DESIGN AND IMPLEMENTATION	10-16

	6.1 Introduction to design	10
	6.2 UML Diagrams	11-15
	6.3 Implementation	15-16
7.	TIMELINE FOR EXECUTION OF PROJECT	17
8.	OUTCOMES	18
9.	RESULTS AND DISSCUSSION	19
10.	CONCLUSION	20

CHAPTER-1

INTRODUCTION

1.1 Motivation:

The primary goal of this project is to develop a streamlined and user-friendly policy update scheme. The objective is to significantly enhance the confidentiality and integrity of PMRs, thereby contributing to the improvement of healthcare data management and access control. By devising an efficient system, we seek to provide individuals with greater control over their personal health information, fostering a more secure and user-centric approach to managing sensitive medical data. This streamlined scheme aims to mitigate the complexities associated with existing methods, ensuring a seamless and secure framework for the dissemination and management of health-related information. Ultimately, our project endeavors to advance the current state of PMR systems, promoting enhanced security, user empowerment, and overall efficiency in healthcare data handling.

1.2 Problem Statement:

The escalating reliance on third-party providers for the storage and management of personal medical records (PMRs) has underscored the pressing challenge of maintaining data security and privacy. Existing policy update schemes for PMRs often prove unwieldy and resource-intensive, prompting the need for a more streamlined approach. This research endeavors to develop a nimble access control policy framework for externally managed personal medical records, aiming to enhance data privacy and security while minimizing computational overhead and system complexity. By addressing the shortcomings of current methods, the proposed scheme seeks to strike a balance between the imperative of robust access control policies and the practical necessity for efficiency in the dynamically evolving landscape of outsourced PHRs.

1.3 Objective of the Project:

The objective of this project is to secure the patient's confidential data in cloud by using Encryption technique. Transfer that data to the patients securely to the patient. Patient will decrypt the data using key and also the doctors and the hospital also views the patient data form emergency purpose.

1.4 Scope

This project aims to develop and implement for Confidentiality of Patient Medical Records. Focusing on overcoming the critical challenges associated with access control policy management, the scheme will prioritize the security and privacy of sensitive health data. Central to the project is the design of efficient algorithms and mechanisms that facilitate seamless policy updates while minimizing computational overhead. Compatibility with existing PHR systems will be a key consideration, with a strong emphasis on user-friendliness, scalability, and robust security measures. The overarching objective is to empower individuals with a secure and efficient method for managing their health information within outsourced PHR environments.

1.5 Project Introduction

In the rapidly evolving landscape of data sharing, particularly within cloud storage systems, maintaining continuous server availability is essential for seamless access to shared information. Cloud providers' economic benefits and efficient resource management have led to the widespread outsourcing of critical data, including sensitive patient health records. To address security concerns in this context, data owners commonly employ encryption as a fundamental layer of protection. However, for environments dealing with sensitive health information, more advanced security measures are imperative. Attribute-Based Encryption with Policy-Controlled Ciphertext Access and Policy-Driven Attribute-Based Encryption (PD-ABE) has emerged as a robust solution, offering fine-grained access control capabilities. Despite the advantages of Attribute-Based Encryption with Policy-Controlled Ciphertext Access, challenges arise in scenarios involving attribute revocation or policy updates, leading to overheads like ciphertext re-encryption and key re-distribution. This paper introduces an innovative approach tailored to Personal Medical Records (PMRs) that efficiently updates Attribute-Based Encryption with Policy-Controlled Ciphertext Access rules without necessitating re-encryption processes on the data owner's end. The model allows data owners, including patients, to selectively share health data with authorized entities, utilizing symmetric encryption for data and encrypting the symmetric key using Attribute-Based Encryption with Policy-Controlled Ciphertext Access. The introduction of a proxy re-encryption (PRE) protocol effectively manages ciphertext re-encryption, reducing computational costs and streamlining policy updates. The paper also introduces a policy

CHAPTER-2

LITERATURE REVIEW

We introduce a groundbreaking paradigm in the realm of identity-based encryption (IBE) termed as "Fuzzy Identity-Based Encryption" (Fuzzy IBE). Within the framework of Fuzzy IBE, identity is perceived as a nuanced and descriptive process. This innovative scheme empowers a third party to decipher a ciphertext symbol encrypted with ω_0 if the private key linked to the identity represented by ω converges in meaning, determined by the "set overlap" distance measure between the symbols ω and ω_0 . Fuzzy IBE showcases its prowess by leveraging biometric login as the foundational identity element for encryption. The inherent strength of Fuzzy IBE lies in its capacity for error tolerance, attributed to the incorporation of biometric identification, which invariably introduces noise into every model. Additionally, our research underscores the versatility of fuzzy IBE in the context of attribute-based encryption. In this paper, we present two innovative models of Fuzzy IBE schemes, conceptualized as individual encryptions of varying qualities that coalesce to form a comprehensive (fuzzy) identity. Essentially, our IBE concept not only exposes potential breaches but also fortifies against impending attacks. Of paramount significance, our infrastructure circumvents dependence on randomness, and we fortify our concepts in alignment with the selected identity security model. In essence, our Fuzzy IBE framework epitomizes a symbiotic blend of cutting-edge artificial intelligence and advanced encryption methodologies, ensuring a unique and secure approach to identity-based encryption.

2.1 Advantages

- 1. Empowering Users:** Robust policy update mechanisms empower individuals, granting them greater authority over who can access their personal health records, fostering patient autonomy.
 - 2. Swift Access Adjustments:** Users can promptly adjust access policies, enabling timely updates in response to changing circumstances or personal preferences.
 - 3. Facilitated Collaboration:** Certain approaches facilitate the secure sharing of health records among healthcare providers, fostering improved care coordination and minimizing redundant tests and treatments.
 - 4. Improved Accessibility:** Remote execution of policy updates is often feasible, simplifying individuals' management of their health information, even from a distance.
 - 5. Alleviated Administrative Load:** Lightweight methodologies alleviate the administrative burden on healthcare organizations, allowing them to prioritize patient care over intricate policy management.
-

- 6. Streamlined Record Sharing:** They simplify the process of sharing health records with authorized parties, such as specialists or family members, ensuring quicker access to crucial information.
- 7. Tailored Customization:** Many approaches offer extensive policy customization, tailoring access levels based on different information categories and user roles.
- 8. Scalability:** Effective methods can seamlessly scale to accommodate a growing user base and their expanding health records without compromising performance or security.

2.2 Disadvantages

- 1. Security Vulnerabilities :**Insufficiently robust policy update methods can pose significant security threats, potentially resulting in unauthorized data breaches and compromises of privacy.
- 2. Data Integrity Risks:** Inadequate methods may fail to sufficiently guard against data tampering or unauthorized alterations to health records, putting the overall integrity of the information at risk.
- 3. Privacy Shortcomings:** Certain approaches may lack adequate privacy safeguards, potentially allowing unauthorized access to sensitive health data by external parties.
- 4. Compliance Challenges:** Existing methods may fall short of meeting evolving legal and regulatory requirements, exposing healthcare entities and individuals to potential compliance issues.
- 5. Complex Implementation:** The implementation and usage of certain policy update methods can be intricate, potentially discouraging active management of health records by users.
- 6. Interoperability Hurdles:** Ensuring seamless data sharing between healthcare providers and users can be challenging due to interoperability issues among different systems and platforms.
- 7. Reliability Concerns:** The reliability of lightweight methods may vary, leading to potential disruptions in policy management and data sharing processes.
- 8. Financial Implications:** Deploying effective policy update methods may incur costs related to software, infrastructure, and staff training, potentially acting as a barrier for smaller healthcare providers or individuals with limited resources.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Ensuring the confidentiality of patient medical information is an important consideration in managing medical information and protecting personal health information. Although there is good competition in treatment and data protection, the current system still faces significant research and challenges.

A significant research gap in the security field of cybersecurity threats, which requires continuous improvement of privacy measures. Advances in technology have introduced new strategies for criminals to attempt to gain unauthorized access to patient information. Research into developing security systems and updates that can deal with threats and vulnerabilities is an urgent need.

Interoperability of electronic health record (EHR) systems becomes another challenge and poses a threat to patient privacy. The lack of formal procedures and conflicting communication between different medical facilities creates opportunities for data deletion and confidentiality. Research efforts should focus on developing internationally accepted standards and procedures to ensure secure data exchange across the healthcare ecosystem.

In addition, user experience and training programs regarding human factors in patient privacy management should be investigated. Even with effective security measures in place, human error and lack of awareness significantly increase the risk of data breach. Research efforts should utilize effective educational strategies to educate practitioners on the importance of privacy and best practices for protecting personal information.

Measuring accessibility and privacy in existing systems is a difficult task. Achieving the balance between ensuring authorized parties have timely access to patient information and preventing unauthorized access requires the search for new methods. This includes advanced authentication and management systems that increase the security of patient information without compromising usability.

In summary, identifying gaps in cybersecurity adaptation, interoperability, user awareness and access control is important to ensure the confidentiality of the patient's medical records.

Collaboration between scientists and practitioners is essential to develop solutions that will not only solve current problems but also reduce future threats, primarily to medical records.

Existing System:

Current cloud service providers frequently offer parallel or public key encryption as optional features to enhance data privacy. Nevertheless, the prevailing solution proves inadequate for the data outsourcing landscape, primarily due to the significant managerial overhead associated with encryption combinations and the substantial maintenance costs incurred in handling diverse ciphertexts using conventional encryption solutions

Drawbacks:

- **Elevated Complexity:** The existing paradigm is plagued by intricate processes, introducing a considerable level of intricacy into data security management.
- **Diminished Computational Efficiency:** The computational performance of the current solution is suboptimal, impeding the seamless execution of encryption tasks and potentially affecting overall system efficiency.
- **Dependency on Skilled Personnel:** Successful implementation and maintenance of the encryption infrastructure demand a skilled workforce, adding another layer of resource dependence.

CHAPTER-4

PROPOSED METHODOLOGY

4.1 Proposed System:

Our pioneering approach revolves around the integration of Ciphertext Policy Behavior-Based Encryption (CP-ABE) and Proxy Re-Encryption (PRE). Additionally, we have implemented cutting-edge policy versioning technology, ensuring comprehensive traceability for any changes made to policies. To substantiate the robustness of our system, we have conducted a thorough performance evaluation.

Key Advantages:

- Precision is exceptional.
- Complexity is kept to a minimum..
- Computational efficiency is notably high
- Skilled personnel are not a prerequisite.

4.2 work Flow of Proposed system:

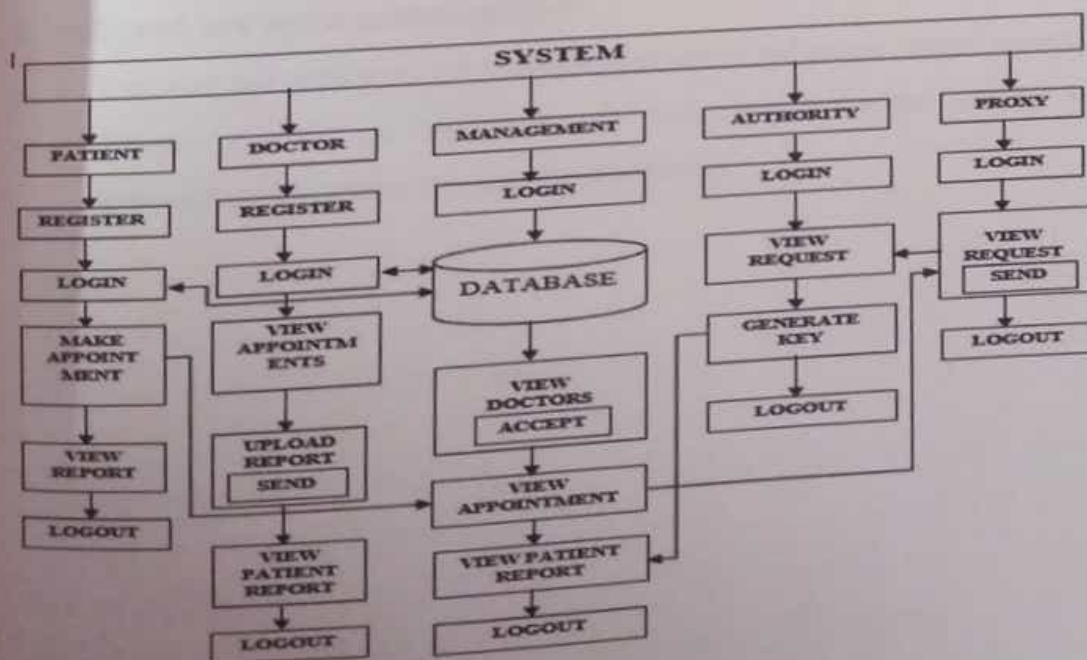


FIG-1

CHAPTER-5

OBJECTIVES

1. Establish privacy protections:

- Establish and implement comprehensive safeguards to ensure the confidentiality of patient medical information.
- First of all, protect sensitive health information against evolving cybersecurity threats.

2. Solve interoperability issues:

- Explore and implement custom procedures to ensure seamless communication between electronic health records (EHR).
- Secure interactions, prevent data leaks and protect patient privacy.

3. Increase user awareness and education:

- Implement effective educational strategies to ensure healthcare professionals are aware of privacy best practices.
- Educate staff on the importance of protecting patient information.

4. Implement new access control systems:

- Discover and implement advanced authentication and access control systems.
- Ensure security and availability by balancing authorized access to patient information and preventing unauthorized access.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 Embarking on Design Excellence:

Within the realm of information systems, raw data undergoes a transformative journey to yield meaningful output. Designers, akin to artisans, harness tools like PCs, MICR, OMR, and more as their creative palette. The selection of these tools plays a pivotal role in the design process, where due consideration must be given to the intricacies of the equipment involved. It is imperative to recognize that the calibre of input directly dictates the finesse of the output. In the crafting of a well-conceived document or display, certain hallmark features come to the fore. It should work well for a specific purpose, such as storing, recording and preserving information.

- He makes sure he does it right and correctly.
- It should be easy to write and understandable.
- The designed entity should seamlessly cater to its designated purpose, be it the storage, recording, or preservation of information.
- It should focus on users in terms of transparency, consistency and simplicity.
- All these goals are achieved by knowing the following basic design principles: -
 - What strategies should be available?
 - How end users react to different content in text and screens.

Goals of strategic planning:

Goals of strategic planning are:-

- Create strategic information and strategic programs
- Reduce idea costs
- Create information to provide information or create other information access How to Understand
- Information entry, information entry, interface screens, etc. create.
- Implement appropriate controls and develop quality control strategies.

6.2 UML Diagrams

6.2.1 Use Case Diagram:

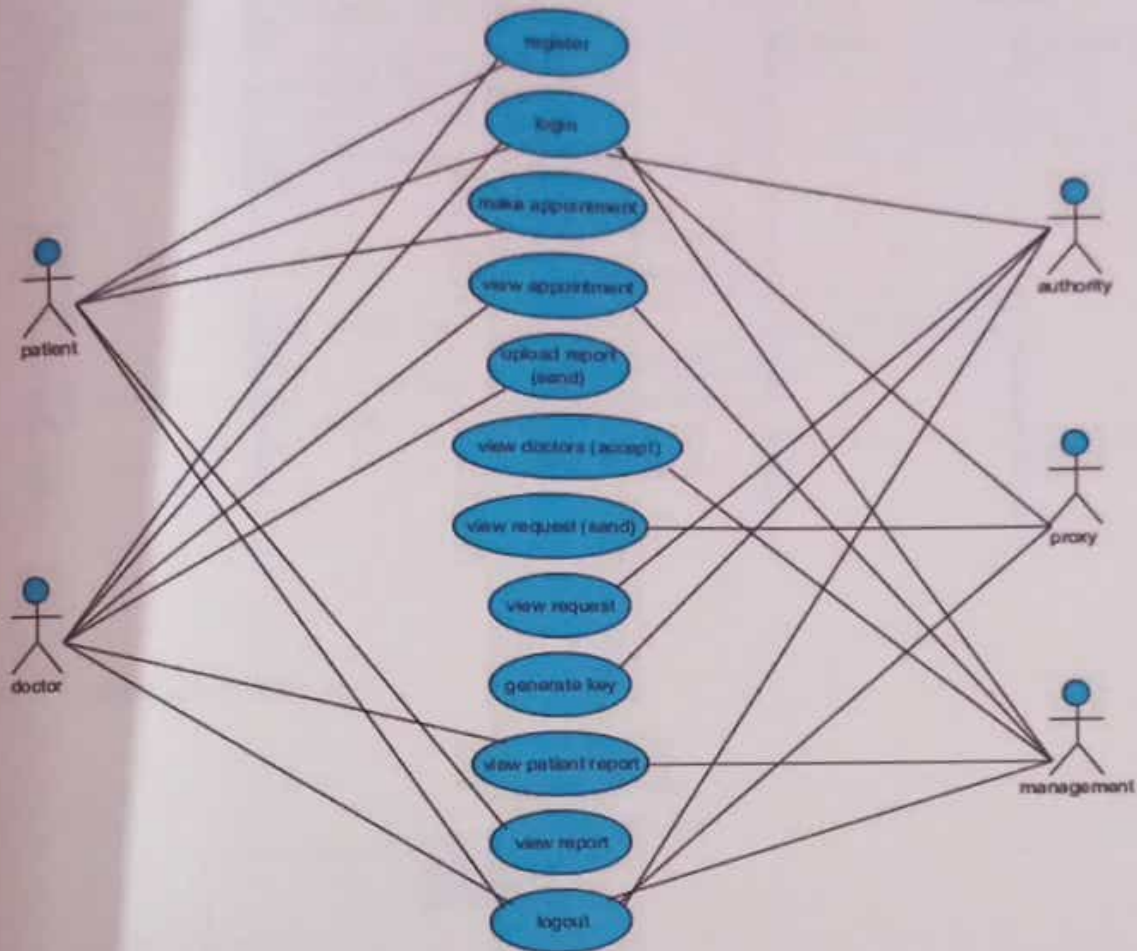


Fig-2.1

6.2.2 Class Diagram:

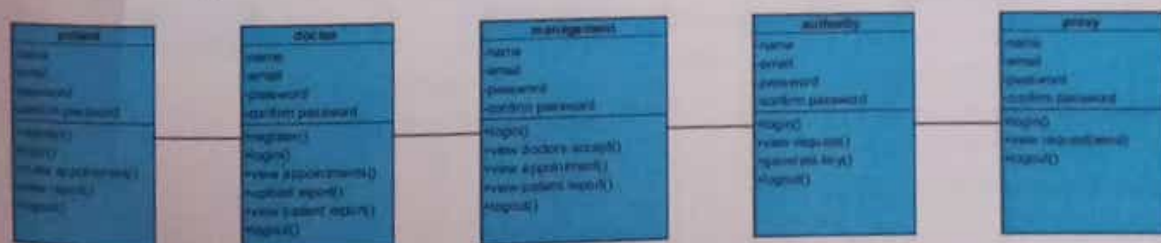


Fig-2.2

6.2.3 Sequence Diagram:

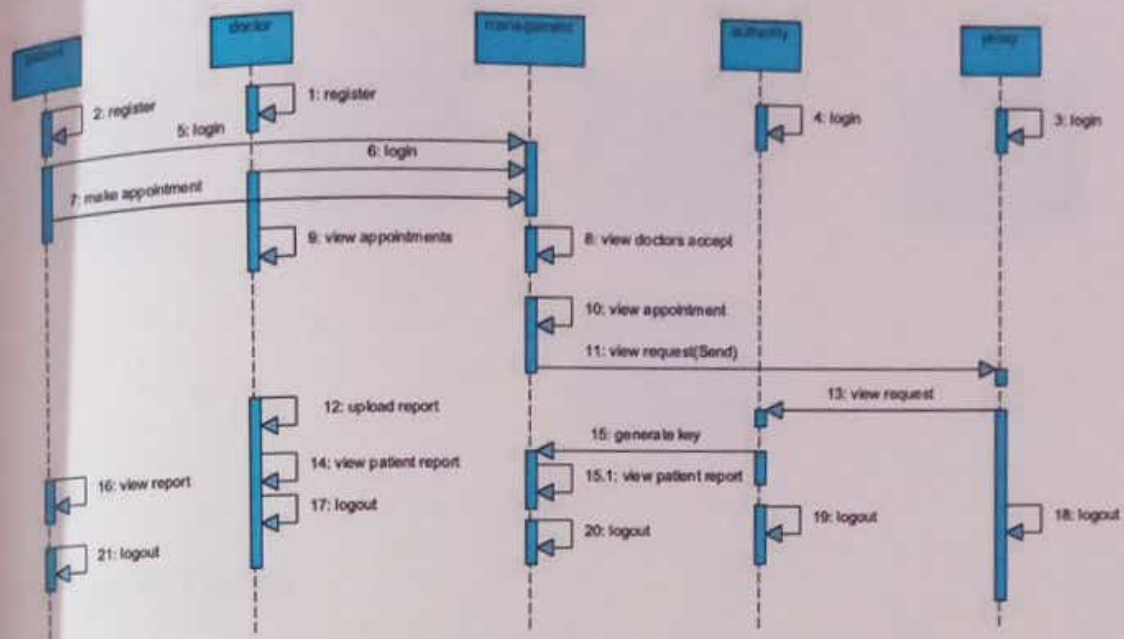


Fig-2.3

6.2.4 Collaboration Diagram:

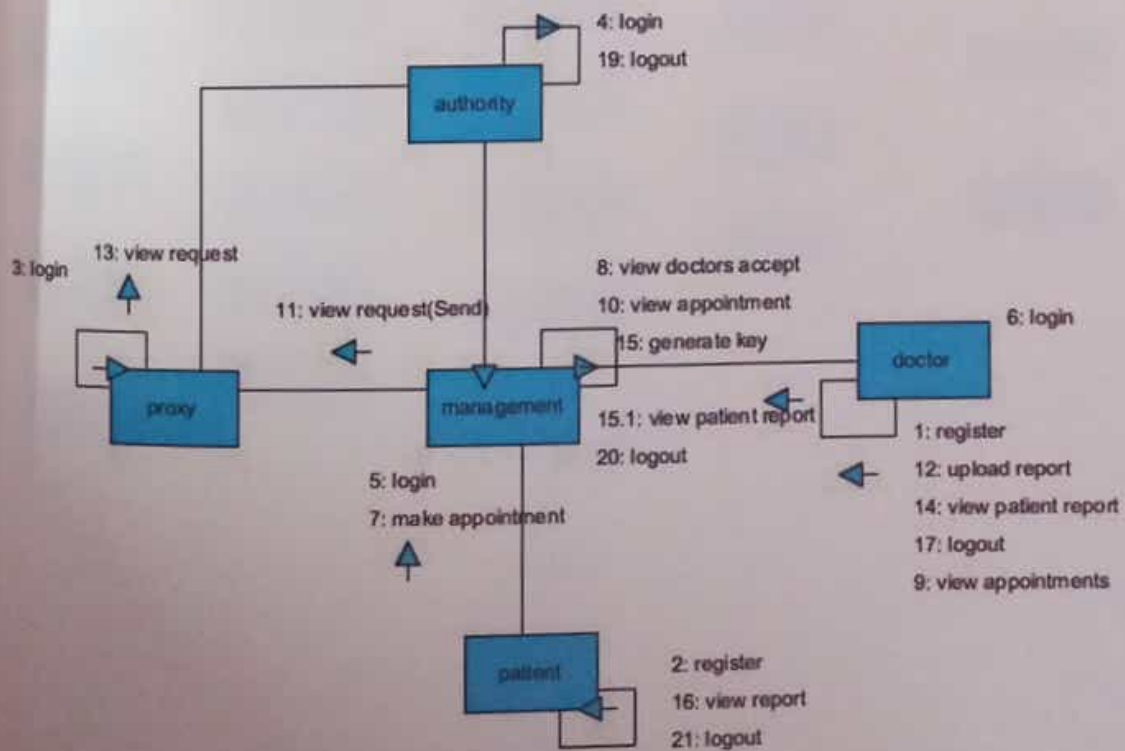


Fig-2.4

6.2.5 Deployment Diagram:

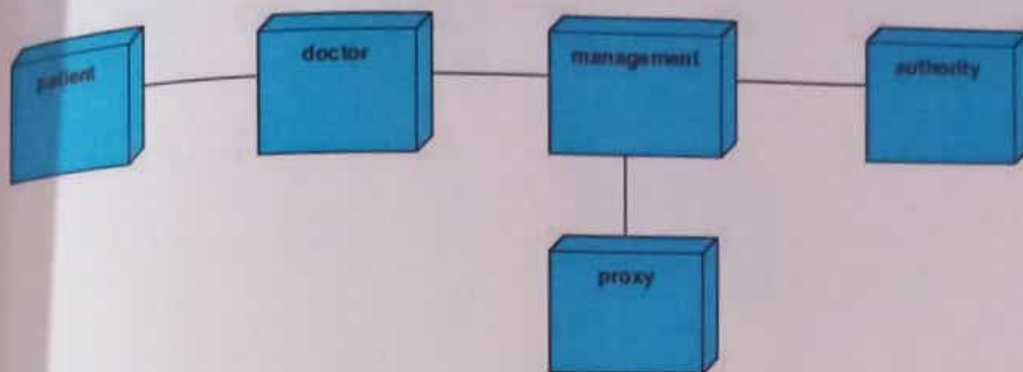


Fig-2.5

6.2.6 Activity Diagram:

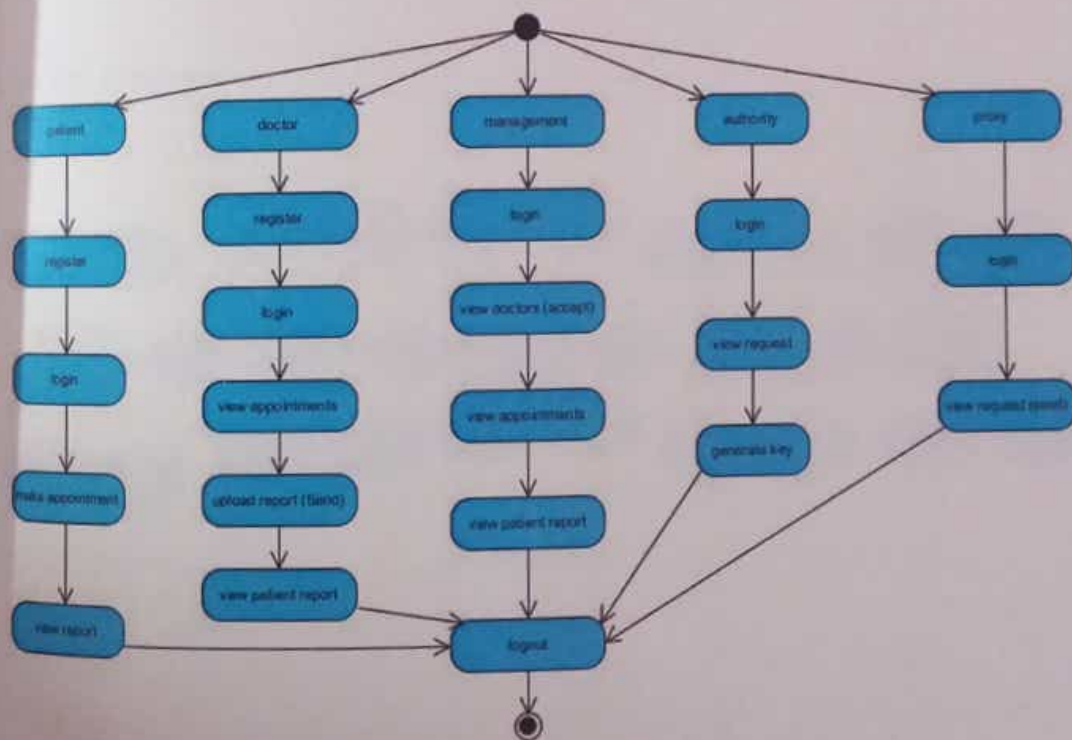


Fig-2.6

6.2.7 Component Diagram:

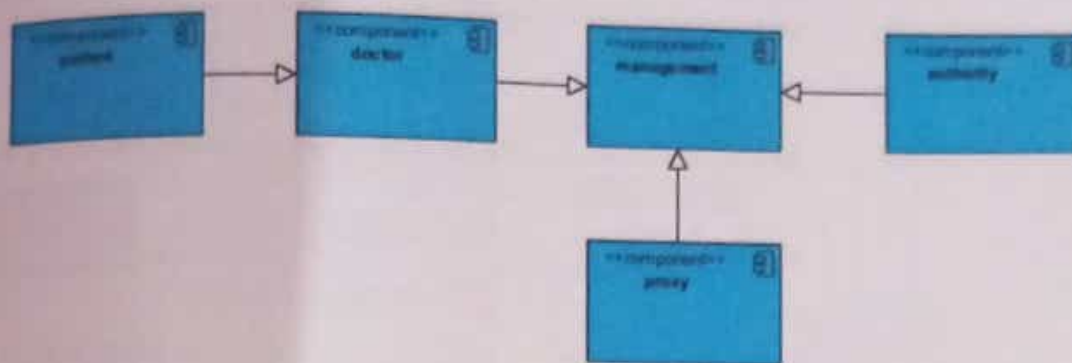


Fig-2.7

6.2.8 ER Diagram:

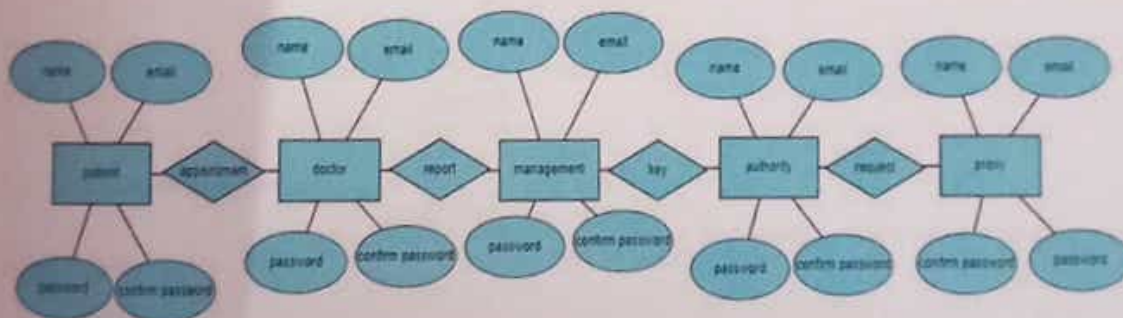


Fig-2.8

6.2.9 DFD Diagram:

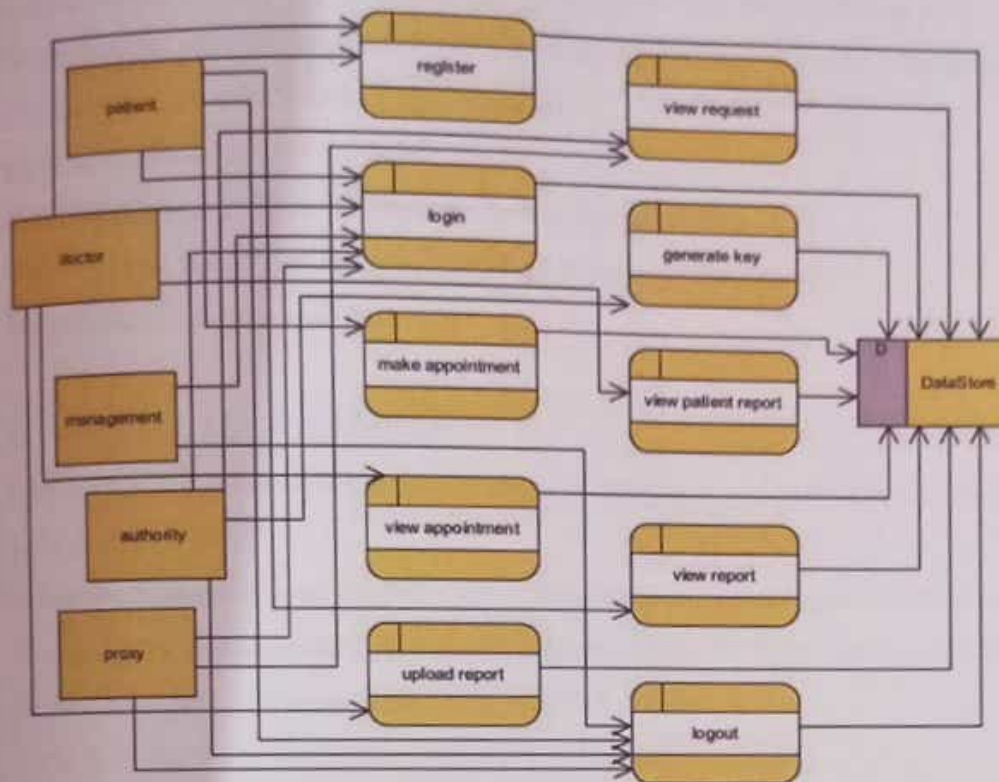


Fig-2.9

6.3 Implementation

6.3.1 Modules

1. PATIENTS:

Login: Patients must log in using valid credentials with which they signed up.

Registration: All patients must register.

Now: Patients will make an appointment with symptoms.

Refer to Chart: The patient will review the chart after the physical examination.

View health history: Patients can view their past medical records.

Exit: Finally, exit the system.

2. PHYSICIANS:

Login: Physicians must access the valid information they are registered with.

Registration: All patients must register and the administrator must accept the offer.

View appointments: View all appointments

Upload Report: Upload a report.

Send information: Send information to the representative

View Patient Report: Patient will view all reports of the patient.

Exit: Finally, exit the system.

3. HOSPITAL MANAGEMENT:

Login: Administrators can log in with false information and view patients' medical records without a key.

Schedule Appointment: Access a list of appointment requests submitted by patients.

Review Doctor Requests: Examine all incoming requests from doctors. Communicate; Dispatch the patient's message to the respective doctor.

View Report: View the patient's emergency report.

Exit: Finally, exit the system.

4. AUTHORITY:

Login: Authorization will be logged in with default details

Retrieve Inquiry: Access all inquiries initiated by the agent.

Generate Key: Formulate a key to be dispatched to the designated and authorized recipient

Log Out: Finally, log out of the system.

5. PROXY SERVER:

Login: The site will be logged in using the default details

View Requests: View all requests from the provider.

Send Request: Submit the request to the organization.

Exit: Finally, log out of the system.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)



CHAPTER-8

OUTCOMES

Complying with privacy when managing patient information can have some great benefits:

1. Trust and Patient Engagement:

Patients are more likely to interact openly and trust their doctors, creating an environment conducive to accurate information sharing and collaborative communication.

2. Compliance and Mitigation:

Strict compliance with data protection requirements to comply with the law, protect healthcare organizations from legal consequences, and reduce concerns about criminal information.

3. Enhanced Data Security:

Privacy measures reduce the risk of unauthorized access, protecting patient information from possible leaks, identity theft and unauthorized access.

4. Honesty and Integrity Policy:

Based on standards of medical ethics before patient confidentiality and promotes the honesty and ethical practices of medical professionals.

5. Improve Healthcare and Outcomes:

Access to confidential and accurate medical information allows doctors to make informed decisions, resulting in personalized care and ultimately better outcomes and health.

CHAPTER-9

RESULTS AND DISCUSSIONS

TEST CASES

Input	Output	Result
Input text files	File Upload or not	Success

TABLE-1

MODEL BUILDING

S.NO	Test Cases	I/O	Expected O/T	Actual O/T	P/F
1	Read data	File data	Data read successfully	Data read success	P
2	Performing Encryption on file data	Encryption has to perform on file data	Encryption has to perform on file data	Encryption successfully completed	P
3	Generating key pair	Key has to generate	Key will generate	Key generated successfully	P
4	Cipher text	File data Encrypted data will Decrypt	Data should be Decrypt	Data decrypted successfully	P

TABLE-2

CHAPTER-10

CONCLUSION

In conclusion, the meticulous maintenance of confidentiality in patient medical health records yields significant benefits for both healthcare providers and patients. The results indicate enhanced trust, legal compliance, reduced data breach risks, improved patient safety, and a commitment to ethical standards.

The discussion emphasizes the ongoing need to balance access and security, leveraging technological advances, investing in staff training, addressing interoperability challenges, and empowering patients. These considerations are vital in navigating the dynamic landscape of healthcare information management.

As healthcare evolves, the conclusion underscores the imperative for a continuous commitment to confidentiality, adapting policies, and embracing innovations to safeguard patient data. Upholding the principles of privacy and trust not only ensures compliance with legal standards but also fosters a resilient and patient-centric healthcare system.

REFERENCES

1. U. S. Varri, S. K. Pasupuleti, and K. Kadambari, "Key-escrow free attribute-based multi-keyword search with dynamic policy update in cloud computing," in Proc. 20th IEEE/ACM Int. Symp. Cluster, Cloud Internet Comput. (CCGRID), Melbourne, VIC, Australia, May 2020, pp. 450–458.
2. Terry, N. P., Francis, L., & Ensor, J. E. (2019). Privacy, Security, and Information Sharing: The Interlinked Triad. *Journal of the American Medical Informatics Association*, 26(10), 945–951.
3. Rothstein, M. A. (2010). Is Deidentification Sufficient to Protect Health Privacy in Research? *The American Journal of Bioethics*, 10(9), 3–11.
4. Malin, B., & Sweeney, L. (2004). How (Not) to Protect Genomic Data Privacy in a Distributed Network: Using Trail Re-identification to Evaluate and Design Anonymity Protection Systems. *Journal of Biomedical Informatics*, 37(3), 179–192.
5. El Emam, K., & Jonker, E. (2016). The Case for De-Identifying Personal Health Information. *Annals of Internal Medicine*, 164(12), 855.
6. Agha, Z., Schapira, R. M., & Laud, P. W. (2009). Patient satisfaction with physician-patient communication during telemedicine. *Telemedicine Journal and E-Health*, 15(9), 830–839.
7. McGraw, D., & Dempsey, J. X. (2009). What Patients and Health Care Organizations Want from On-line Health Records. *Journal of the American Medical Informatics*.
8. Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. In *ACM SIGMOD International Conference on Management of Data* (pp. 439–450).
9. Poon, C. C. Y., & Zhang, Y. T. (2008). Biometrics-based privacy-preserving remote health monitoring — A review. *IEEE Journal of Biomedical and Health Informatics*, 13(6), 1473–1481.
10. Car, J., Black, A., Anandan, C., & Cresswell, K. (2011). The impact of eHealth on the quality and safety of health care: A systematic overview. *PLoS Medicine*, 8(1), e1000387.
11. G. Hsieh, and R. J. Chen, "Design for a secure interoperable cloud-based Personal Health Record service," In 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2012, pp.472-479.

APPENDIX-A

PSUEDOCODE

```
import os
from flask import *
import mysql.connector
import pandas as pd
import random
from flask_mail import *

db = mysql.connector.connect(user='root', port=3307, database='lightweight')
cur = db.cursor()
app = Flask(__name__)
app.secret_key = '!@#H%S$BV#AS><)SH&BSGV*(_Sjnkxcb9+_ )84JSUHB&*%$^+= '

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/patient', methods=['POST', 'GET'])
def Patientlog():
    if request.method == 'POST':
        name = request.form['Name']
        password = request.form['Password']
        cur.execute(
            "select * from patient_reg where Name=%s and Password=%s", (name,
password))
        content = cur.fetchone()
        print(content)
        age = content[-3]
        db.commit()
        if content is None:
            msg = "Credentials Does't exist"
            return render_template('patientlog.html', msg=msg)
        else:
            return render_template('patienthome.html', name=name, age=age)
    return render_template('patientlog.html')

@app.route('/patientreg', methods=['POST', 'GET'])
def Patientreg():
    if request.method == 'POST':
        name = request.form['Name']
        age = request.form['Age']
        email = request.form['Email']
        password1 = request.form['Password']
```

```

password2 = request.form['Con_Password']
if password1 == password2:
    sql = "select * from patient_reg where Name='%s' and Email='%s'"

% (
    name, email)
cur.execute(sql)
data = cur.fetchall()
db.commit()
print('----', data)
if data == []:
    sql = "insert into patient_reg(Name, Age, Email, Password)
values(%s,%s,%s,%s)"
    val = (name, age, email, password1)
    cur.execute(sql, val)
    db.commit()
    return render_template('patientlog.html')
else:
    warning = 'Details already Exist'
    return render_template('patientreg.html', msg=warning)
error = 'password not matched'
flash(error)
return render_template('patientreg.html')

@app.route('/proceed')
def proceed():
    return render_template('proceed.html')

@app.route('/patientreq', methods=['POST', 'GET'])
def patientreq():
    if request.method == 'POST':
        Name = request.form['Name']
        doc = request.form['Doc']
        Age = request.form['Age']
        Symptoms = request.form['symptoms']
        AppointmentDate = request.form['AppointmentDate']
        Time = request.form['Time']
        sql = "insert into patientreq
(Name, Type, Age, symptoms, AppointmentDate, Time) values
('%s', '%s', '%s', '%s', '%s', '%s')" % (
            Name, doc, Age, Symptoms, AppointmentDate, Time)
        cur.execute(sql)
        db.commit()
        msg = "Your appointment request Sent to Management"
        return render_template('patienthome.html', msg=msg)
    return render_template('patienthome.html')

```



```

@app.route('/hospitalmanagement', methods=['POST', 'GET'])
def hospital_management():
    if request.method == 'POST':
        name = request.form['Username']
        password = request.form['passcode']
        print(name, password)
        if name == "Hospital" and password == 'management':
            return render_template('managementhome.html')
        return render_template('management.html')

@app.route('/viewappointments')
def view_appointments():
    sql = "select Id,Name,Type,Age,AppointmentDate,Time from patientreq where status='pending'"
    data = pd.read_sql_query(sql, db)
    db.commit()
    return render_template('viewappointments.html', cols=data.columns.values, rows=data.values.tolist())

@app.route('/accept_request/<x>/<y>/<z>')
def acceptreq(x=0, y='', z=''):
    print(x, y)
    print(z)
    sql = "select Name,Department,Email from docreg where Department='%s' " % (
        z)
    data = pd.read_sql_query(sql, db)
    db.commit()
    print(data)
    if data.empty:
        flash('Doctot is not available')
        return redirect(url_for('view_appointments'))
    else:
        sql = "update patientreq set status='accepted' where status='pending' and Id='%s' and Name='%s' " % (
            x, y)
        cur.execute(sql)
        db.commit()
        return render_template('acptreq.html', cols=data.columns.values, rows=data.values.tolist())

@app.route('/Connect/<x>/<y>/<z>')
def mergereq(x='', y='', z=''):
    print(x)
    print(y)
    print(z)

```



```

sql = "select name,Type,Age from patientreg where status='accepted' and
Type='%s'" % (
    y)
cur.execute(sql)
da = cur.fetchall()
db.commit()
dat = [j for i in da for j in i]
print(dat)
print(dat[0], dat[2], dat[1])

```

```

sql = "insert into
connectdata(Patientname,patientAge,Type)values('%s','%s','%s')" % (
    dat[0], dat[2], dat[1])
cur.execute(sql)
db.commit()

```

```

return redirect(url_for('view_appointments'))

```

```

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

```

```

    if request.method == 'POST':

```

```

        dept = request.form['Department']

```

```

        name = request.form['Name']

```

```

        age = request.form['Age']

```

```

        number = request.form['Number']

```

```

        email = request.form['email']

```

```

        password = request.form['password']

```

```

        conpassword = request.form['conpassword']

```

```

        if password == conpassword:

```

```

            print("True")

```

```

            sql = "select * from docreg"

```

```

            cur.execute(sql)

```

```

            data = cur.fetchall()

```

```

            db.commit()

```

```

            for i in data:

```

```

                if email in i[5]:

```

```

                    msg = "Email already Exist's"

```

```

                    return render_template('doctorreg.html', msg=msg)

```

```

            else:

```

```

                sql = "insert into

```

```

docreg(Name,Department,Age,Number,Email>Password)

```

```

values('%s','%s','%s','%s','%s','%s')" % (

```

```

    name, dept, age, number, email, password)

```

```

    cur.execute(sql)

```

```

    db.commit()

```

```

    msg = "Your Request Sent to Management"

```

```

    return render_template('doctorreg.html', msg=msg)

```

```

else:

```

```

msg = "password doesn't Match"
return render_template('doctorreg.html', msg=msg)

return render_template('doctorreg.html')

@app.route('/Doc_requests')
def Docrequests():
    sql = "select Name,Department,Age,Number,Email from docreg where
status='pending'"
    data = pd.read_sql_query(sql, db)
    db.commit()
    return render_template('Doc.html', cols=data.columns.values,
rows=data.values.tolist())

@app.route('/acpt_doc/<x>/<y>')
def acceptdoc(x='', y=''):
    sql = "update docreg set status='accepted' where status='pending' and
Name='%s' and Email='%s'" % (
    x, y)
    cur.execute(sql)
    db.commit()
    """
    Have to Complete Email Code

    """
    sender_address = 'lahariuma11@gmail.com'
    sender_pass = 'sukqmsdpqpparoap'
    content = "Your Request Is Accepted by the Management You Can Login Now"
    receiver_address = y
    message = MIMEMultipart()
    message['From'] = sender_address
    message['To'] = receiver_address
    message['Subject'] = "Confidentiality of Patient Medical Records "
    message.attach(MIMEText(content, 'plain'))
    ss = smtplib.SMTP('smtp.gmail.com', 587)
    ss.starttls()
    ss.login(sender_address, sender_pass)
    text = message.as_string()
    ss.sendmail(sender_address, receiver_address, text)
    ss.quit()
    return redirect(url_for('Docrequests'))

@app.route('/Docs')
def Docs():
    sql = "select Name,Department,Age,number,Email from docreg where
status='accepted'"

```

```

data = pd.read_sql_query(sql, db)
db.commit()
return render_template("docs.html", cols=data.columns.values,
rows=data.values.tolist())

@app.route('/doctor_log', methods=['POST', 'GET'])
def doctorlog():
    if request.method == 'POST':
        Docname = request.form['Docname']
        passcode = request.form['Docpasscode']
        sql = "select * from docreg where status='accepted' and name='%s'" %
        (
            Docname)
        cur.execute(sql)
        data = cur.fetchall()
        db.commit()
        print(data)
        email = data[0][-3]
        session['doc'] = email
        if data != []:
            i = [i for i in data]
            session['dept'] = i[0][2]
            if Docname in i[0][1] and passcode in i[0][-2]:
                msg = "Doctor Login Success"
                return render_template("docrequest.html", msg=msg)
            else:
                msg = "Details doesn't exist"
                return render_template("doctorlog.html", msg=msg)
        return render_template('doctorlog.html')

@app.route('/view_patient')
def viewpatient():
    sql = "select * from connectdata where Type='%s'" % (session['dept'])
    cur.execute(sql)
    data = cur.fetchall()
    db.commit()
    print(data)
    if data == []:
        msg = "You dont have any appointments "
        return render_template("viewpatient.html", msg=msg)
    Name = data[0][1]
    Age = data[0][2]
    Type = data[0][3]
    return render_template('viewpatient.html', name=Name, age=Age, type=Type)

@app.route("/patient_access/<a>/<b>")

```

```

def patientaccess(a='', b=0):
    sql = "select Email from patient_reg where Name='%s' and Age='%s'" % (a,
    b)
    cur.execute(sql)
    data = cur.fetchall()
    db.commit()
    print(data)
    if data != []:
        Email = data[0][0]
        session['email'] = Email
        return render_template("uploadfile.html", email=Email)
    --"patient mail access code""
    msg = "Your Appointment is accepted "
    # return render_template("patientaccess.html", msg=msg)
    return render_template("uploadfile.html")

@app.route('/upload_file', methods=['POST', 'GET'])
def uploadfile(email=''):
    print(email)
    if request.method == 'POST':
        filedata = request.files['filedata']
        n = filedata.filename
        data = filedata.read()
        print(data)
        path = os.path.join(
            "uploadfiles/", n)
        filedata.save(path)
        status = "accepted"
        sql = "insert into reports(FileName,FileData,patientEmail,Status)
values(%s,AES_ENCRYPT(%s,'sec_key'),%s,%s)"
        val = (n, data, session['email'], status)
        cur.execute(sql, val)
        db.commit()

        msg = "file Uploaded successfully"
        return render_template('uploadfile.html', msg=msg)
    return render_template('uploadfile.html')

@app.route('/view_files')
def viewfiles():
    sql = "select FileName,Filedata,PatientEmail from reports where
PatientEmail='%s' and status='accepted'" % (
        session['email'])
    data = pd.read_sql_query(sql, db)
    db.commit()
    return render_template('files.html', cols=data.columns.values,
rows=data.values.tolist())

```



```
@app.route('/performs')
def performs():
    sql = "update reports set status='updated' where status='accepted' and
    PatientEmail='%s'" % (
        session['email'])
    cur.execute(sql)
    db.commit()
    return redirect(url_for('viewfiles'))
    # return render_template('performs.html')

@app.route('/authority', methods=['POST', 'GET'])
def authority():
    if request.method == 'POST':
        name = request.form['Username']
        password = request.form['passcode']
        if name == 'Authority' and password == 'auth':
            return render_template('authhome.html')

    return render_template('authority.html')

@app.route('/vr')
def vr():
    sql = "select Id,FileName,PatientEmail from reports where
    status='updated'"
    data = pd.read_sql_query(sql, db)
    db.commit()
    return render_template('vr.html', cols=data.columns.values,
    rows=data.values.tolist())

@app.route('/proxy_server', methods=['POST', 'GET'])
def proxyserver():
    if request.method == 'POST':
        name = request.form['Username']
        password = request.form['passcode']
        if name == "proxy" and password == "server":
            return render_template('proxylog.html')

    return render_template('proxy.html')

@app.route('/Generate_Key/<c>/')
def generatekey(c=0):
    x = random.randrange(000000, 999999)
    print(x)
```

```

print(c)
sql = "update reports set Key1='%s',status='done' where Id = '%s' and
status='updated' " % (
    x, c)
cur.execute(sql)
db.commit()

return redirect(url_for('vr'))

```

```

@app.route('/all_requests')
def allrequests():
    sql = "select Id,FileName,PatientEmail,Key1 from reports where
status='done' and PatientEmail='%s'" % (
    session['email'])
    data = pd.read_sql_query(sql, db)
    db.commit()
    return render_template('all.html', cols=data.columns.values,
rows=data.values.tolist())

```

```

@app.route('/sentmail/<e>/<k>')
def sentmail(e='', k=0):
    sender_address = 'lahariuma11@gmail.com'
    sender_pass = 'sukqmsdpqpparoap'
    content = str(k)
    print(content)
    receiver_address = e
    message = MIMEMultipart()
    message['From'] = sender_address
    message['To'] = receiver_address
    message['Subject'] = "Confidentiality Of Patient Medical Records"
    message.attach(MIMEText(content, 'plain'))
    ss = smtplib.SMTP('smtp.gmail.com', 587)
    ss.starttls()
    ss.login(sender_address, sender_pass)
    text = message.as_string()
    ss.sendmail(sender_address, receiver_address, text)
    ss.quit()
    sql="update reports set status='complete' where status='done' and
PatientEmail='%s'"%(session['email'])
    cur.execute(sql)
    db.commit()
    return redirect(url_for("allrequests"))

```

```

@app.route('/view_report', methods=['POST', 'GET'])
def viewreport():
    try:

```

```
sql = "select * from reports where Status='complete' and
patientEmail='%s'" % (
    session['email'])
data = pd.read_sql_query(sql, db)
db.commit()
if request.method == 'POST':
    keyvalue = request.form['keycvalue']
    sql = "select * from reports where Status='complete' and
patientEmail='%s'" % (
        session['email'])
    cur.execute(sql)
    data = cur.fetchall()
    db.commit()
    print(data)
    if keyvalue in data[0][-1]:
        sql = "select AES_DECRYPT(FileData, 'sec_key') from reports
where PatientEmail='%s'" % (
            session['email'])
        cur.execute(sql)
        data = cur.fetchall()
        db.commit()
        data = data[0][0].decode()
        return render_template('views.html', data=data)
    return render_template('report.html', cols=data.columns.values,
rows=data.values.tolist())
except:
    msg = "Your reports are not available"
    return render_template('patienthome.html', msg=msg)

@app.route('/logout')
def logout():
    return redirect(url_for('/'))

if __name__ == "__main__":
    app.run(debug=True)
```

APPENDIX-B

SCREENSHOTS

1. HOMEPAGE

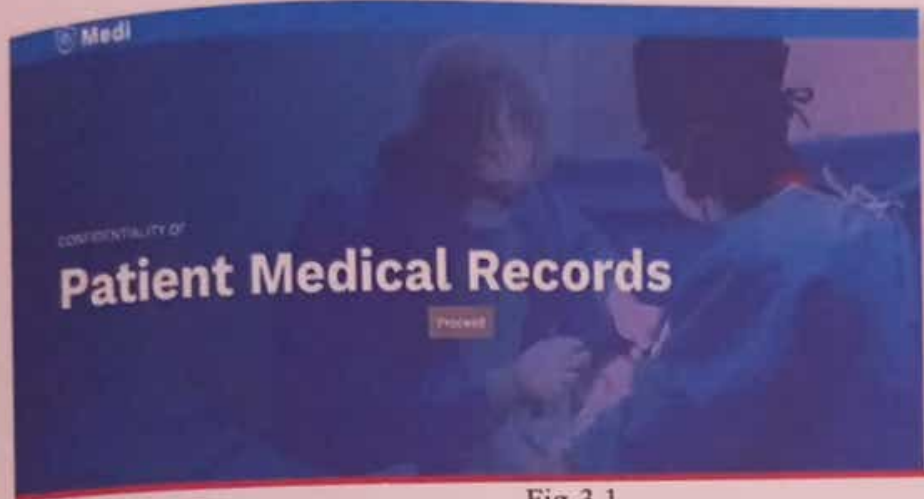


Fig-3.1

2. MODULES PAGE

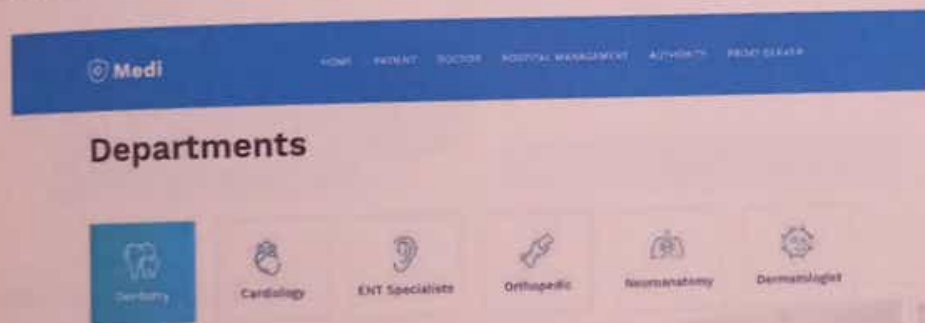


Fig-3.2

3. PATIENT LOGIN FORM



Fig-3.3


4. PATIENT REGISTRATION PAGE



The screenshot shows a web application interface for patient registration. On the left, there is a background image of a person's arm in a blue medical sleeve. On the right, a white form titled "REGISTRATION FORM" is displayed. The form contains several input fields: "First Name", "Last Name", "Email", "Phone", "Address", and "City". Below these fields is a blue button labeled "Register".

Fig-3.4

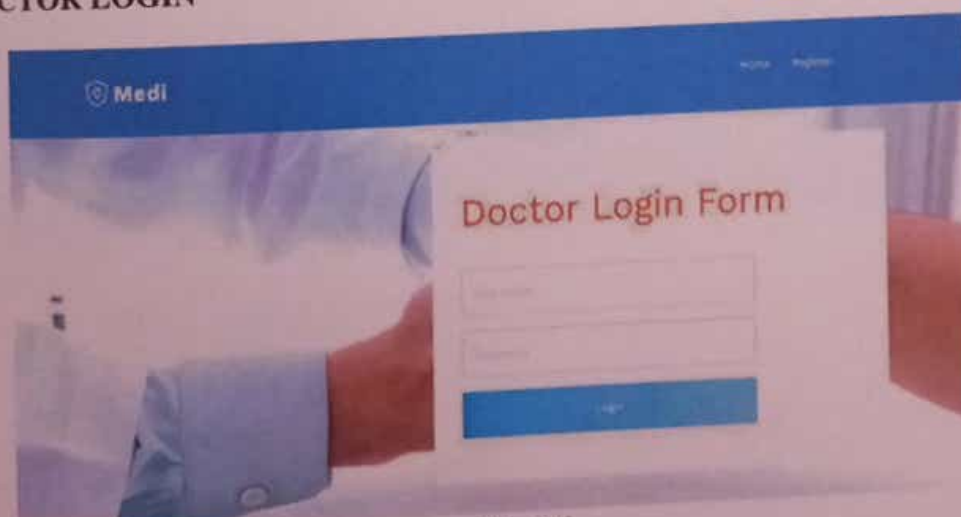
5. APPOINTMENT FORM



The screenshot shows a web application interface for appointments. At the top, a blue header bar contains the "Medi" logo and navigation links for "Home", "About Us", and "Contact". Below the header, a white form titled "Appointment" is shown. The form includes a "patient" dropdown menu, a "Please select a doctor" dropdown, and a "Book" button. There are also fields for "Appointment Date" and "Appointment Time".

Fig-3.5

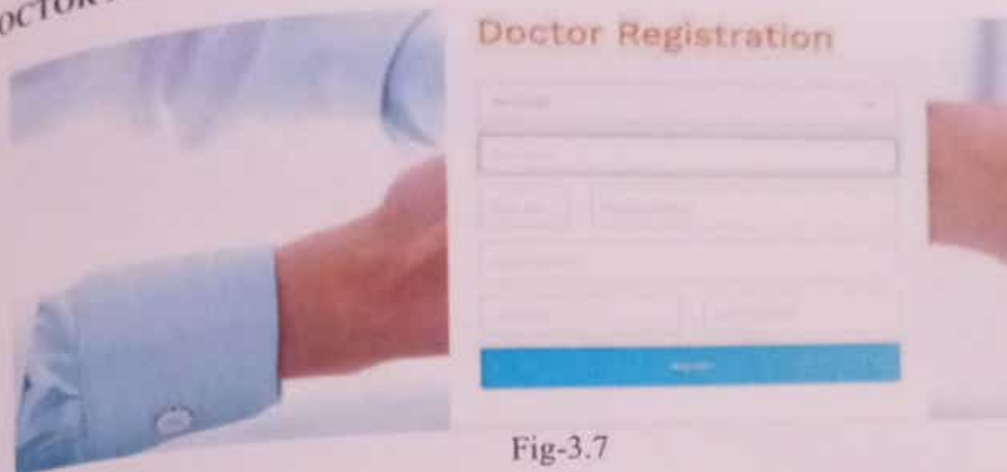
6. DOCTOR LOGIN



The screenshot shows a web application interface for doctor login. At the top, a blue header bar contains the "Medi" logo and navigation links for "Home" and "Register". Below the header, a white form titled "Doctor Login Form" is displayed. The form has two input fields: "Username" and "Password". Below these fields is a blue button labeled "Login".

Fig-3.6

7. DOCTOR REGISTRATION FORM



The screenshot shows a web form titled "Doctor Registration". It includes a profile picture upload area on the left. The form fields include "Email", "Password", "Confirm Password", "First Name", "Last Name", "Specialty", "Address", and "Phone Number". A blue "Register" button is at the bottom.

Fig-3.7

8. DOC HOME PAGE

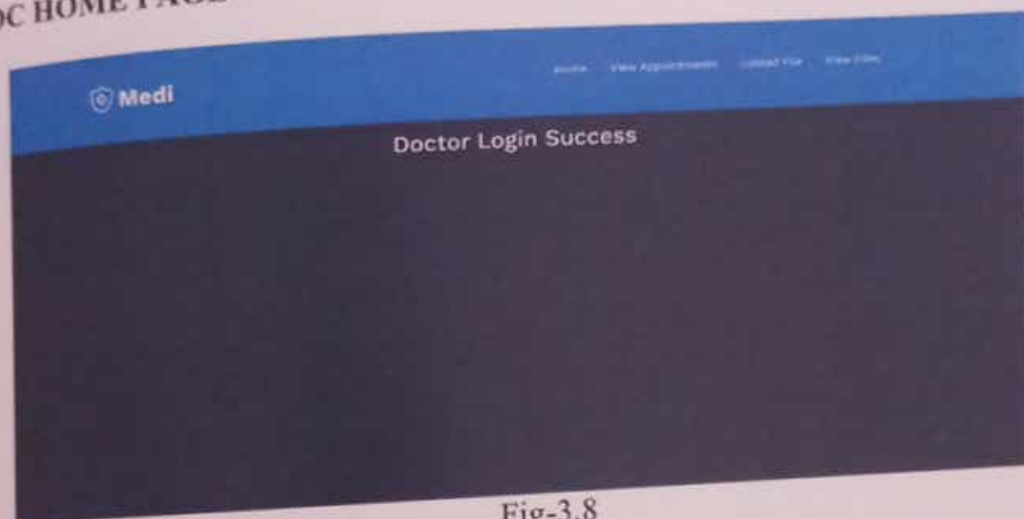


Fig-3.8

9. VIEW APPOINTMENTS



Fig-3.9

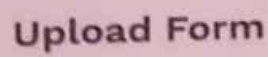
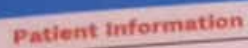


Fig-3.10

11. VIEW FILES



Patient Information		
PatientName	PatientEmail	Author
John Doe	john.doe@gmail.com	Prof. Dr. Smith

Fig-3.11

12. VIEW REPORTS

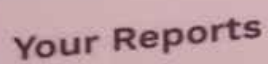


Fig-3.12

13. MANAGEMENT LOGIN

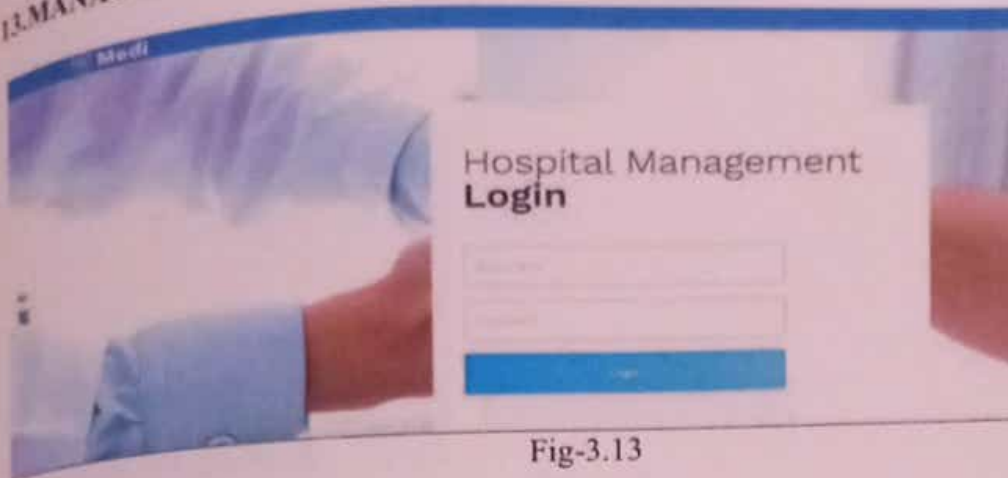


Fig-3.13

14. DOCTOR REQUEST

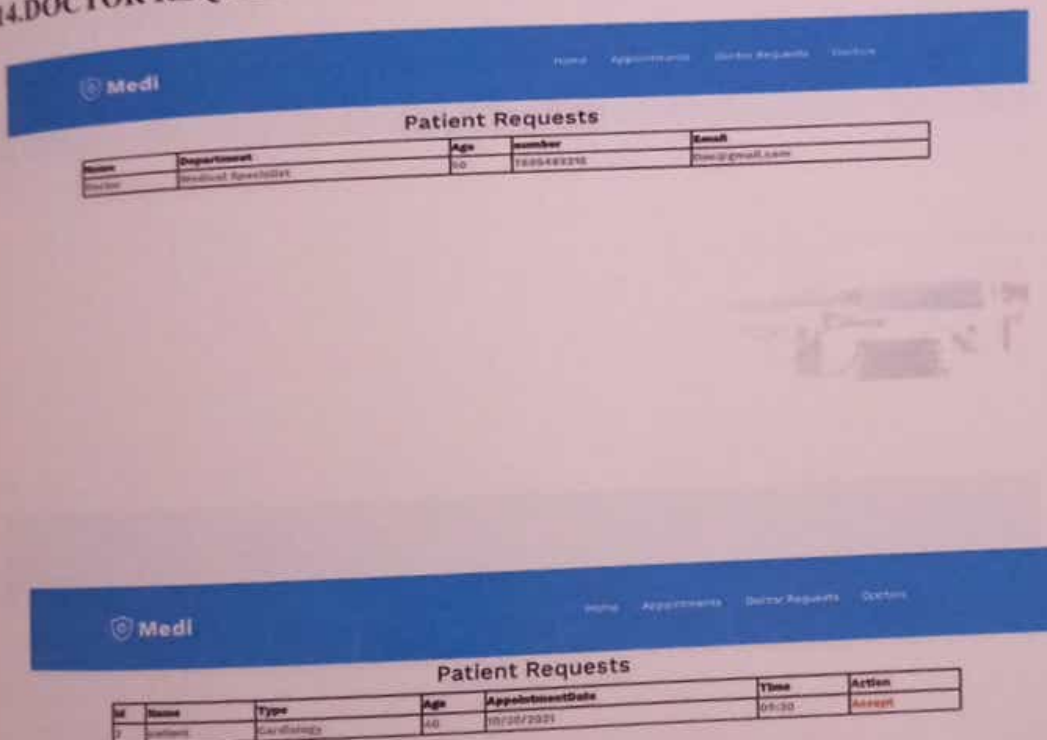


Fig-3.14

15. PROXY LOGIN

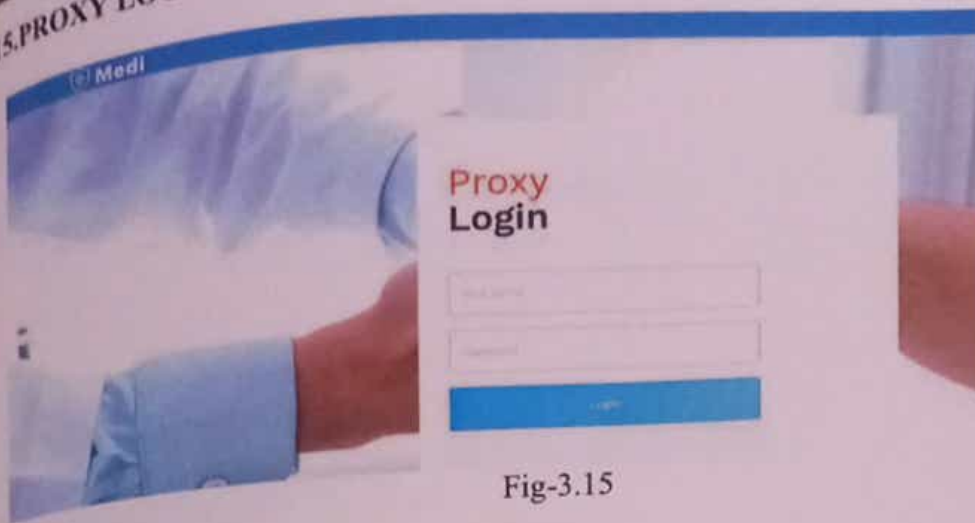


Fig-3.15

16. VIEW REQUEST

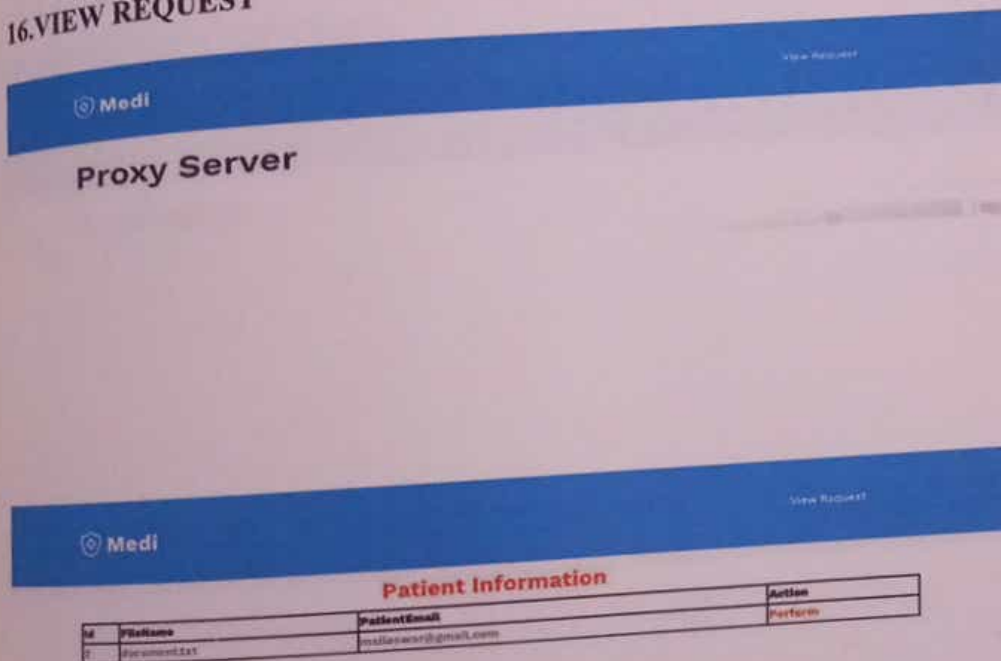


Fig-3.16

17. AUTHORITY LOGIN

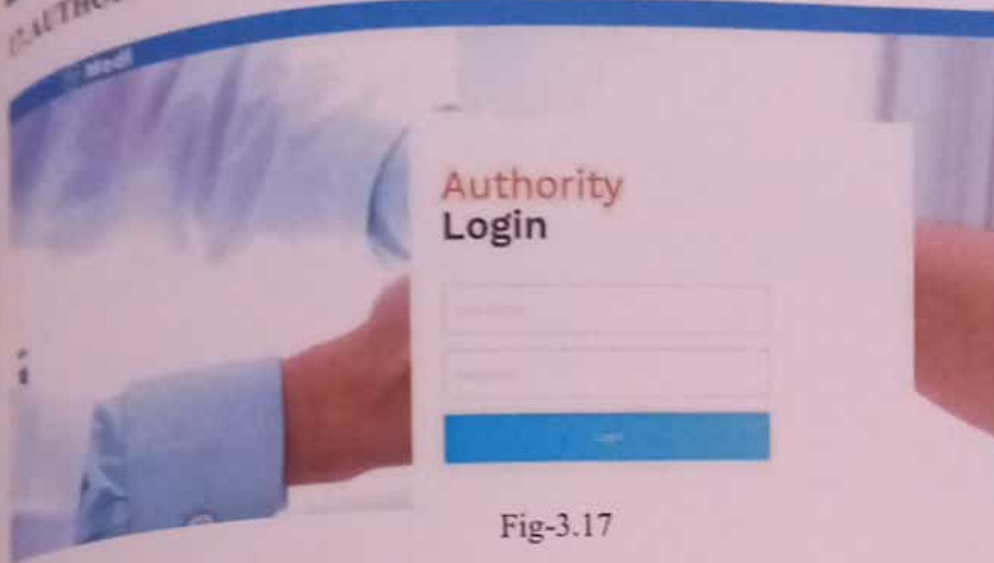


Fig-3.17

18. AUTHORITY REQUESTS



Fig-3.18

APPENDIX-C

ENCLOSURES



SUSTAINABLE DEVELOPMENT GOALS

17 GOALS TO TRANSFORM OUR WORLD



The Project work carried out here is mapped to SDG-3 Good Health and Well-Being:

Good Health and Well-Being is crucial for addressing global health challenges. The project work carried here contributes to the well-being of the human society. This project involves improving the management of patient medical records, it can certainly contribute to enhancing healthcare efficiency and patient outcomes. Keep up the impactful work!

12/01/2024, 15:50

Submission Summary

Conference Name

IEEE International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications

Track Name

Blockchain and Cyber Physical Systems;

Paper ID

271

Paper Title

Confidentiality of Patient Medical Records

Abstract

As the integration Personal Medical Records and cloud computing transforms the management of health information, the focus is on storing patient information. This document present about framework designed to improve the safeguarding and confidentiality of confidential information. Based on Attribute-Based Encryption with Policy-Controlled Ciphertext Access and Proxy-Based Data Transformation for Encryption our method allows providers to not only define the process easily manage them, but also the content. Control payload using traditional encryption techniques. Incorporating a policy definition process allows for careful investigation and provides a comprehensive accounting process for changes in access policy. Through performance evaluation, our system demonstrates good performance and reliability by creating the ability to pose threats against unauthorized access, enabling patient data private healthcare to improve in the challenging health environment.

Created on

12/01/2024, 15:41:13

Last Modified

12/01/2024, 15:41:13

Authors

Bysani Lakshmi Narasimha Sai Lahari (Presidency University) <

bysanilahari@gmail.com> ✓

Prakash Shanmurthy (Presidency University) < research4prakash@gmail.com> ✓

SD Amruthavalli (Presidency University) < amruthakeerthana83@gmail.com> ✓

Lingutla Thanusha (Presidency University) < lthanusha2002@gmail.com> ✓

Submission Files

Confidentiality of Patient Medical Records.pdf (609.9 Kb, 12/01/2024, 15:40:30)

G66

ORIGINALITY REPORT

4%

SIMILARITY INDEX

3%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

www.journaltoocs.ac.uk
Internet Source

1%

2

www.coursehero.com
Internet Source

1%

3

Abhijeet Borade, Rashmi Agarwal. "Chapter 17 Securing Outsourced Personal Health Records on Cloud Using Encryption Techniques", Springer Science and Business Media LLC, 2023
Publication

1%

4

www.perseus.tufts.edu
Internet Source

<1%

5

idr.mnit.ac.in
Internet Source

<1%

6

www.researchgate.net
Internet Source

<1%

Exclude quotes

Off

Exclude matches

Off