**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JNANA SANGAMA”, BELAGAVI - 570018, KARNATAKA**

****

**A Project Report**

**On**

**"Blood Donation Management System: Reducing the Gap Between Donors and Recipients with Compassion and Efficiency"**

***In the partial fulfilment of the requirement for the award of degree***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

***Submitted by***

|  |  |
| --- | --- |
| **PAVAN CHANDRAPPA HOTTIGOUDRA** | **(4VZ22CS019)** |
| **SHIVANI VEERESH LINGADAHALLI** | **(4VZ22CS026)** |
| **SOURABHA SHRENIKRAJA HALLI** | **(4VZ22CS029)** |
| **VIKAS** | **(4VZ22CS031)** |

***Under the guidance of***

**Dr. G. F. Ali Ahammed**

**Program Coordinator,**

**Dept. of Computer Science and Engineering,**

**VTU Regional Center, Mysore.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Centre for Post Graduate Studies, Sathagalli, Mysuru – 570029.**

**2024 – 2025**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JNANA SANGAMA”, BELAGAVI - 570018, KARNATAKA**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Centre for Post Graduate Studies, Sathagalli, Mysuru – 570029.**

**2024 – 2025**

**CERTIFICATE**

This is to certify that the Project work entitled **"Blood Donation Management System: Reducing the Gap Between Donors and Recipients with Compassion and Efficiency"** is a bonafied work carried out by **Pavan Chandrappa Hottigoudra, Shivani Veeresh Lingadahalli, Sourabha Shrenikraj Halli, Vikas** bearing USN **4VZ22CS019, 4VZ22CS026, 4VZ22CS029, 4VZ22CS031** at Department of **Computer Science and Engineering, Visvesvaraya Technological University, Centre for Post Graduate Studies, Mysuru** in partial fulfilment for the award of **Bachelor of Technology in Computer Science and Engineering, Visvesvaraya Technological University, Belagavi during the academic year 2024-2025.** It is certified that all the corrections/suggestions indicated during Internal Assessment have been incorporated in the report. The Mini Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Technology degree.

**Signature of the Guide**

Dr. G. F. ALI AHAMMED

Program Coordinator,

Dept. of CS&E,

VTU, CPGS Mysuru - 570029

**Signature of Program Coordinator**

Dr. G. F. ALI AHAMMED

Program Coordinator,

Dept. of CS&E,

VTU, CPGS Mysuru - 570029

**Name of the Examiners:**  **Signature with date**

**1.**

**2.**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground our efforts with success.

We express our sincere thanks to **Dr. G F ALI AHAMMED, Program Coordinator, PG Studies, Department of Computer Science & Engineering, VTU Regional Centre, Mysuru** for his kind help and constant encouragement and for providing us necessary facilities for carrying out this work successfully.

In particular, We would like to take this opportunity to express our Honor, Respect, Deepest Gratitude and Genuine Regards to our guide, **Dr. G F ALI AHAMMED**, **Program Coordinator**, Department of Computer Science and Engineering, VTU Regional Centre, for giving us all guidance required for our project apart from being a constant source of inspiration and motivation.

We owe our special thanks to **Our Parents** for their moral support and warm wishes and finally We would like to express appreciation to all **Our Friends** for their unconditional support which helped us to complete this work successfully.

|  |  |
| --- | --- |
| Pavan Chandrappa Hottigoudra | 4VZ22CS019 |
| Shivani Veeresh Lingadahalli | 4VZ22CS026 |
| Sourabha Shrenikraja Halli | 4VZ22CS029 |
| Vikas | 4VZ22CS031 |

**DECLARATION**

We Pavan Chandrappa Hottigoudra, Shivani Veeresh Lingadahalli, Sourabha Shrenikraja Halli, Vikas bearing USN 4VZ22CS019, 4VZ22CS026, 4VZ22CS029, 4VZ22CS031 hereby declare that this project work entitled **"Blood Donation Management System: Reducing the Gap Between Donors and Recipients with Compassion and Efficiency"**, is a bonafide work carried out by me under the guidance and supervision of Dr. G. F. Ali Ahammed, Program Coordinator, Department of Computer Science and Engineering, VTU Center for PG Studies, Mysore. This project work is submitted to Visvesvaraya Technological University, Belagavi in partial fulfilment of the requirements for the award to degree of Bachelor of Technology in Computer Science and Engineering during the academic year 2024-2025.

|  |  |
| --- | --- |
| Pavan Chandrappa Hottigoudra | 4VZ22CS019 |
| Shivani Veeresh Lingadahalli | 4VZ22CS026 |
| Sourabha Shrenikraja Halli | 4VZ22CS029 |
| Vikas | 4VZ22CS031 |

**ABSTRACT**

In an era where cybersecurity threats are rapidly increasing, ensuring the privacy and security of digital communication has become a major concern. This project proposes a secure messaging application that is fully decentralized and built upon blockchain technology to enhance confidentiality, integrity, and authenticity of communication.

Unlike traditional messaging platforms that rely on centralized servers — which are not only expensive but also vulnerable to data breaches, censorship, and privacy violations — this system eliminates single points of failure by leveraging peer-to-peer networking. Messages are transmitted directly between users, ensuring that sensitive data is not routed or stored through a central authority.

The application integrates end-to-end encryption (E2EE) along with blockchain-based identity authentication and decentralized storage mechanisms. To enable real-time communication, WebRTC is used for direct, secure peer-to-peer messaging. Furthermore, the system adopts a customized variation of the Proof of Authority (PoA) consensus algorithm to support secure and reliable group messaging in decentralized environments.

This research addresses the stark absence of a comprehensive decentralized communication framework by combining the strengths of blockchain, WebRTC, and Decentralized Applications (dApps). It offers a robust, private, and censorship-resistant platform that enhances user trust and experience while placing the highest value on privacy.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | Introduction | 1-3 |
| 2 | System Analysis   1. Existing System 2. Proposed System | 4-6 |
| 3 | System Requirements   1. Hardware Specification 2. Software Specification 3. Functionality Requirements 4. Non-Functionality Requirements | 7-8 |
| 4 | System Architecture   1. ER Diagram 2. UseCase Diagram 3. Activity Diagram | 9-11 |
| 5 | System Design   1. Database Table Structure | 12-14 |
| 6 | Implementation   1. Packages 2. Functions | 15-18 |
| 7 | Testing | 19-21 |
| 8 | Results | 22-29 |
| 9 | Conclusion | 30 |
| 10 | References | 31 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| Fig.1 | Home page | 22 |
| Fig.2 | Blood bank registration | 23 |
| Fig.3 | Donor registration | 23 |
| Fig.4 | Reset password | 23 |
| Fig.5 | Donor dashboard | 24 |
| Fig.6 | Admin details | 25 |
| Fig.7 | Donars list | 25 |
| Fig.8 | Blood bank list | 25 |
| Fig.9 | Ongoing requests list | 26 |
| Fig.10 | Fulfilled requests list | 26 |
| Fig.11 | Blood bank dashboard | 27 |
| Fig.12 | Blood request form | 28 |
| Fig.13 | Request Message to donor | 29 |
| Fig.14 | Certificate | 29 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| Table 5.1 | Donar registration table | 12 |
| Table 5.2 | Blood request table | 13 |
| Table 5.3 | Blood bank registration table | 14 |
| Table 5.4 | Admin table | 14 |
| Table 7.1 | User authentication test case | 19 |
| Table 7.2 | Donar dashboard test case | 19 |
| Table 7.3 | Blood bank dashboard test case | 20 |
| Table 7.4 | Blood request management test case | 20 |
| Table 7.5 | Donar registration test case | 21 |
| Table 7.6 | Blood bank registration test case | 21 |
| Table 7.7 | Password reset with SMS OTP test case | 21 |

**CHAPTER 1**

**INTRODUCTION**

In today's rapidly evolving digital landscape, securing personal communication has become more critical than ever.  
 "Conventional messaging apps are based on centralized servers and are prone to data leakage, eavesdropping, and unauthorized interception."   
Our system — a **Quantum-Secure Messaging Application** — is designed to redefine privacy by combining **post-quantum cryptography**, **decentralized identity management**, and **peer-to-peer networking** into one seamless, efficient platform.  
 "Decentralized operation make use of peer-to-peer networks, this ensures that no network failure can do due to central node failure."   
"Blockchain serves as an unalterable ledger which allows messaging to take place in a decentralized manner."

At its core, the application leverages **Kyber**, a post-quantum encryption algorithm, to ensure that even the most powerful quantum computers cannot compromise private conversations. Every message is encrypted with **AES-256-GCM**, protected through Kyber-secured key exchange, and distributed securely using **IPFS (InterPlanetary File System)**, **WebSockets**, and **libp2p's decentralized peer-to-peer protocol**.  
 "IPFS establishes a permanent distributed network by using content-based reference."   
"This decentralized architecture eliminates single points of failure, enhances data redundancy, and ensures high availability of information even if some nodes go offline."   
This future-proof cryptographic foundation guarantees end-to-end confidentiality, forward secrecy, and resilience against quantum threats.

Rather than relying on centralized servers, the application uses a fully decentralized architecture based on Web3 principles.  
 "Peer-to-peer networks distribute resources, with each node acting as both provider and consumer, thus enhancing scalability and reducing dependency on a single messaging server."   
Technologies like **IPFS**, **WebSockets**, and **Kademlia Distributed Hash Tables (kadDHT)** power secure storage, message routing, and global peer discovery, ensuring users can communicate directly without intermediaries.  
 "Random connections between peers carry out resource discovery and information dissemination."   
This design not only enhances security and privacy but also improves network scalability and fault tolerance.

User identity management is decentralized and quantum-secure. Each user is assigned a **Decentralized Identifier (DID)** linked to their mobile number through a secure, OTP-based authentication system.  
 "Each Internet-connected computer needs to install and run a computer application specific to the ecosystem they wish to participate in."   
Critical metadata, such as DIDs and public keys, are securely managed via MongoDB (or optionally, MySQL) under strict encryption policies, ensuring real-time access without exposing sensitive information.  
 "Data and applications are kept on a single, central server in centralized servers, making them vulnerable to security flaws and hacking."

The backend is powered by **Node.js with Express.js**, handling **API orchestration, WebSocket communication management, encryption services, peer session management, and message routing**.  
 The frontend is built using **React.js** for a responsive, dynamic user experience, coupled with **Redux.js** for efficient state management across authentication, messaging, and encryption workflows.  
 "The proposed solution uses WebRTC, a modern peer to peer web communication technology as the underlying framework that facilitates secure communication."   
HTML5 and CSS3 standards are employed to ensure a fast, adaptive, and user-friendly interface.

The system optionally supports **WebRTC** for secure peer-to-peer **voice and video calls**, maintaining the same quantum-resistant security for call metadata.  
 "WebRTC can only connect two peers if their public IPs are known."   
WebSockets are simultaneously leveraged to maintain **real-time, low-latency messaging** between peers for text and metadata communication.  
 Waku v2 integration for enhanced decentralized message relay is also available as an optional feature for scaling under large network loads.

By automating critical cryptographic operations, enabling decentralized direct messaging, and minimizing server reliance, the system ensures that user conversations remain private, resilient, and future-proof — even against the advent of quantum computing.  
 "Decentralized chat applications, therefore, offer a safer and more private solution for those wishing to connect online."

This messaging application marks a significant leap forward in digital privacy, empowering users with complete control over their communications and setting a new standard for secure, decentralized interactions in the Web3 era.

**Aim :**

We’re building a Quantum-Secure Messaging App that takes privacy and security to the next level. It uses cutting-edge post-quantum cryptography, decentralized identity management, and peer-to-peer networking to keep your conversations safe. The app relies on Kyber for quantum-resistant key exchanges and AES-256-GCM for strong end-to-end encryption. Instead of depending on centralized servers, it uses IPFS and libp2p to store and route messages in a fully decentralized way.

Users will manage their identities with Decentralized Identifiers (DIDs), secured through simple OTP-based authentication. Beyond messaging, the app will also support quantum-secure voice and video calls, ensuring scalable, secure communication for the future. Our goal is to create a powerful, resilient platform that keeps your privacy intact, even as quantum computing becomes a reality.

**Motivation :**

We’re developing a Quantum-Secure Messaging App because secure, private communication is more important than ever—especially with the rise of quantum computing threats. Our goal is to give users full control over their conversations by using post-quantum cryptography, decentralized identity management, and peer-to-peer networking.

By cutting out centralized servers, we’re working to eliminate risks like data breaches and ensure that sensitive information stays private and protected, even against future quantum attacks.

With technologies like Kyber encryption, AES-256-GCM, IPFS, and libp2p, we're building a system that not only boosts user trust and privacy but also sets a new standard for secure, decentralized communication.

At the heart of this project is a simple mission: to protect personal conversations, keep digital interactions authenticpro, and drive innovation toward a future where communication stays safe—no matter what new technologies emerge.

Problem statement :

With the rise of quantum computing, existing encryption methods are becoming increasingly vulnerable, putting the privacy of digital communication at serious risk. At the same time, centralized messaging platforms expose users to threats like data breaches, censorship, and surveillance. This project addresses these challenges by creating a secure, decentralized messaging application that uses post-quantum cryptography and peer-to-peer networking to ensure long-term privacy, data ownership, and resilience against both current and future threats.

Scope of the project :

This project focuses on building a truly secure and private messaging platform that’s ready for the future. It will provide users with real-time, end-to-end encrypted messaging, secured using quantum-resistant technologies like Kyber and AES-256-GCM.

Instead of relying on traditional centralized servers, all data and communication will be handled through decentralized peer-to-peer networks using IPFS and libp2p, giving users complete control over their information and reducing the risks of hacking or data breaches.

Users will manage their identities through Decentralized Identifiers (DIDs) with simple, secure OTP-based authentication, further enhancing trust and privacy. The app will support not just text messaging, but also optional quantum-secure voice and video calls, making it a full-featured communication platform.

In short, this project aims to deliver a future-proof solution that protects conversations from today’s threats and the even bigger risks posed by quantum computing in the future. It will set a new standard for secure, decentralized communication and give people confidence that their private conversations will stay truly private.

**Objective :**

Protect Communication Against Quantum Threats:

Build a messaging platform that uses post-quantum cryptography to stay secure even against future quantum computer attacks.

Ensure Complete Privacy and User Control:

Give users full ownership of their conversations and identities by removing reliance on centralized servers and using decentralized technologies.

Deliver End-to-End Encryption:

Use strong, proven encryption methods like Kyber for key exchange and AES-256-GCM for securing messages, ensuring that only intended recipients can read the conversations.

Implement Decentralized Identity Management:

Let users create and manage their digital identities safely through Decentralized Identifiers (DIDs) secured with simple OTP-based authentication.

Enable Real-Time, Peer-to-Peer Communication:

Support fast, direct messaging without intermediaries by using IPFS and libp2p for decentralized data storage and routing.

Support Quantum-Secure Voice and Video Calls:

Extend security beyond text by offering optional voice and video communication that's protected against quantum threats.

Set a New Standard for Privacy and Trust:

Create a platform that not only meets today’s privacy needs but also inspires confidence and trust for the future of digital communication.

**LITERATURE SURVEY**

### **1. [Shivani.pdf — *Decentralized Secure Messaging Application Using Blockchain*]**

**Existing System:**

This paper presents a decentralized messaging platform built on blockchain technology, replacing centralized servers with a peer-to-peer network. By leveraging blockchain’s immutable ledger and encryption capabilities, it strengthens communication security and ensures continuity even if some nodes go offline. Messages are distributed across the network, making tampering highly challenging. However, the system still relies on classical cryptographic methods, which may become vulnerable as quantum computing advances.

**Proposed System:**

To address these concerns, our proposed Quantum-Secure Messaging Application integrates post-quantum cryptography, using Kyber for key exchanges. It also builds on decentralized principles by utilizing IPFS for storage and libp2p for peer-to-peer communication, while implementing Decentralized Identifiers (DIDs) for secure identity management. This approach ensures greater resilience against quantum threats while maintaining decentralization and enhancing forward secrecy.

### **2. [Decentralized-Chat-Application-Using-Blockchain-Technology.pdf]**

**Existing System:**

The system described focuses on building a decentralized chat platform through blockchain, eliminating reliance on central servers. It uses Ethereum and the Whisper protocol to enable encrypted peer-to-peer messaging. Although it successfully tackles server dependency and improves user privacy, it continues to depend on conventional cryptographic techniques that are expected to be broken by future quantum computers.

**Proposed System:**

Our solution strengthens this model by incorporating quantum-safe encryption, specifically Kyber for secure key exchanges and AES-256-GCM for data encryption. Rather than relying solely on blockchain frameworks like Ethereum, we integrate IPFS and libp2p to create a more scalable, efficient, and quantum-resistant peer-to-peer messaging system designed to withstand both current and emerging cyber threats.

### **3. [SendingNetworkAdvancingtheFuture.pdf]**

**Existing System:**

SendingNetwork introduces a robust decentralized messaging protocol using libp2p, dynamic relay nodes, and Proof of Relay/Availability mechanisms. It enhances scalability and privacy through encryption schemes like the Double Ratchet Algorithm for group communications. Although the system greatly advances decentralized infrastructure, it still depends on classical cryptographic standards, leaving it vulnerable to future quantum decryption methods.

**Proposed System:**

Our Quantum-Secure Messaging Application builds upon these foundations by embedding post-quantum encryption techniques at the core, particularly Kyber for key establishment. It further enhances decentralized communication using IPFS for distributed storage and DIDs for secure, user-controlled identity. This results in a highly scalable and quantum-resilient messaging platform that ensures privacy even against next-generation computing attacks.

### **4. [2504.10365v1.pdf — *Staggering and Fragmentation for Improved Large Message Handling in libp2p GossipSub*]**

**Existing System:**

This paper focuses on optimizing message transmission in libp2p's GossipSub protocol by introducing fragmentation and staggering techniques. These improvements significantly reduce network latency and improve bandwidth usage for decentralized applications. However, while it addresses efficiency challenges, it does not address encryption robustness, especially concerning quantum-era threats.

**Proposed System:**

Our system retains the scalability benefits offered by GossipSub optimizations while embedding quantum-resistant security measures. By integrating Kyber-based encryption for key exchanges and AES-256-GCM for message protection, along with decentralized content storage via IPFS and secure identity management through DIDs, the application ensures both high performance and strong, future-proof communication security.

### **5. [A\_Decentralized\_Application\_for\_Secure\_P.pdf]**

**Existing System:**

The paper proposes a decentralized messaging solution that employs WebRTC for direct peer-to-peer communication and uses a blockchain with Proof of Authority (PoA) for authentication. This setup ensures communication persists even when users go offline by involving temporary relay nodes. However, its reliance on classical encryption makes it potentially vulnerable to future quantum computing capabilities.

**Proposed System:**

In our design, WebRTC remains a core component for real-time peer-to-peer connections but is strengthened through quantum-safe encryption methods. Kyber is used for secure key exchanges and AES-256-GCM for data confidentiality. Additionally, IPFS decentralized storage and OTP-verified DIDs provide stronger security layers, ensuring communication remains confidential and tamper-proof even in a quantum-empowered environment.

### **6. [paper.pdf — *Secure Messaging WebApp Using Blockchain Technology*]**

**Existing System:**

This system presents a secure messaging application leveraging blockchain for decentralized storage and smart contracts for authentication and key management. It offers end-to-end encryption and ensures data integrity. Nonetheless, its reliance on traditional asymmetric cryptography leaves it susceptible to decryption by quantum computers in the near future.

**Proposed System:**

Our Quantum-Secure Messaging Application upgrades this approach by utilizing Kyber, a post-quantum encryption algorithm selected by NIST, for key exchanges. Messages are further protected with AES-256-GCM encryption. IPFS enables decentralized storage, and user identities are managed securely with DIDs, making the platform highly resistant to both classical and quantum cyberattacks.

### **7. [20 June 2024 IJERCSE.pdf — *Decentralized Messaging Web Application*]**

**Existing System:**

The paper outlines a decentralized messaging web application using blockchain to ensure data integrity, availability, and resistance to censorship. It successfully addresses challenges associated with centralization, such as single points of failure. However, its dependence on classical encryption methods could make it vulnerable to attacks from future quantum computers.

**Proposed System:**

Our proposed system extends the decentralization benefits while future-proofing security by incorporating Kyber for quantum-safe key exchanges and AES-256-GCM for data encryption. Combined with IPFS storage and libp2p networking for peer discovery, and secure DIDs for identity management, the system ensures robust, tamper-resistant communication even against emerging quantum threats.

### **8. [Secure\_Peer-to-Peer\_Communication\_using\_Private\_Network\_Blockchain\_Technology.pdf]**

**Existing System:**

This paper discusses a secure peer-to-peer communication model built on a private blockchain network using Go-Ethereum (GETH) and Proof of Authority (PoA) consensus. While the system effectively tackles issues of censorship, data security, and centralization, its encryption and identity verification methods still rely on classical cryptographic techniques, posing risks in the quantum computing era.

**Proposed System:**

Our application strengthens this approach by embedding Kyber-based quantum-resistant encryption for secure key exchanges and employing IPFS for decentralized storage. It also introduces OTP-secured Decentralized Identifiers (DIDs) for tamper-proof identity management. Together, these enhancements create a communication network that remains secure and decentralized, even against the capabilities of future quantum computers.

**Problem Statement**

Today’s digital communication is increasingly threatened by cyberattacks, data breaches, and emerging technologies like quantum computing. Most messaging platforms rely heavily on centralized servers or traditional cryptographic techniques, leaving them vulnerable to surveillance, censorship, and future quantum decryption attacks. Since these systems are not designed to withstand the capabilities of quantum computers, there is a pressing need for a messaging platform that ensures privacy, protects user identities, and maintains communication resilience — all without depending on a central authority.

**Existing Systems (Literature Review Summary)**

**Decentralized Blockchain-Based Messaging**

Research such as *Shivani.pdf* and *Decentralized Chat Application Using Blockchain* introduced peer-to-peer messaging using blockchain technology, Ethereum, and the Whisper protocol to eliminate centralized control and strengthen security.

**Libp2p-Based Messaging Networks**

Projects like *SendingNetwork* took a step further by integrating libp2p for peer-to-peer networking, incorporating group encryption and relay systems to improve scalability and reliability.

**GossipSub Optimization**

The paper *2504.10365v1* focused on optimizing decentralized communication by proposing message fragmentation and staggering techniques to improve GossipSub protocol performance for large messages.

**WebRTC and Blockchain Hybrid Systems**

Solutions such as *A\_Decentralized\_Application\_for\_Secure\_P.pdf* and *Secure Peer-to-Peer Communication Using Private Blockchain* combined WebRTC technology and private blockchain networks (like GETH) to enable direct, secure messaging between users.

**Secure Messaging with Blockchain and Smart Contracts**

Systems discussed in *paper.pdf* used blockchain-based smart contracts and AES encryption to offer secure messaging features but remained reliant on traditional cryptography methods.

**Limitations of Existing Systems**

**Lack of Quantum Resistance**

Most current systems use classical cryptographic algorithms like RSA and ECC, which are vulnerable to quantum attacks through Shor’s and Grover’s algorithms.

**Partial Decentralization**

Many platforms still have elements of centralization, such as reliance on smart contract hosting services or bootstrap nodes, limiting their resilience and autonomy.

**Weak Identity Management**

Identity handling often relies on semi-centralized databases and lacks full decentralized identifiers (DIDs), which makes users susceptible to identity theft and tracking.

**Absence of Secure Real-Time Media Communication**

Only a few systems address encryption for voice and video calls, and none currently offer quantum-secure protection for real-time media streams.

**Scalability and Efficiency Issues**

Solutions heavily dependent on blockchain often face bottlenecks such as high transaction fees, limited throughput, and network congestion, restricting their scalability.

**Proposed System: Quantum-Secure Messaging Application**

To overcome the gaps identified in the existing systems, the proposed project introduces a **Quantum-Secure Messaging Application** that provides secure, decentralized, and quantum-resistant communication. Key features include:

**Post-Quantum Cryptography**

The system implements **Kyber**, a NIST-recommended algorithm, to ensure that key exchanges are safe against quantum decryption attempts, securing the future of private communications.

**AES-256-GCM Encryption**

All message content is encrypted with AES-256-GCM, offering strong, efficient, and authenticated encryption to preserve message integrity and confidentiality.

**Decentralized Storage & Networking**

**IPFS** is used for decentralized message distribution, and **libp2p** with **Kademlia DHT** is employed for peer discovery, eliminating the need for central servers and improving fault tolerance.

**Decentralized Identity Management (DIDs)**

Each user is assigned a Decentralized Identifier linked to their mobile number, authenticated through secure OTP verification. Metadata and cryptographic keys are protected in encrypted MongoDB/MySQL databases.

**Quantum-Secure Real-Time Communication**

Optional support for **WebRTC** enables encrypted voice and video calls, securing both data and metadata with quantum-resistant protections.

**Scalable Communication Relay**

**Waku v2** is optionally integrated to handle message relay efficiently over large decentralized networks, improving scalability without compromising privacy.

**Modern Architecture**

The system uses **React.js** and **Redux** for the frontend and **Node.js** with **Express.js** for backend services, ensuring a highly responsive, scalable, and secure communication experience.

# **CHAPTER 4**

# **SYSTEM REQUIREMENT SPECIFICATION**

This chapter outlines the system requirements necessary to develop and deploy the **Quantum-Secure Messaging Application**. These specifications ensure optimal performance, high security, decentralization, and real-time communication capabilities even under quantum-era threats.

## **4.1 Hardware Specification**

To ensure smooth operation of the decentralized, encrypted messaging platform, the following hardware specifications are recommended:

* **Processor**: Intel Core i3 / AMD Ryzen 3 or higher
* **RAM**: Minimum 6GB or more (for handling real-time encryption, WebRTC media streams, and decentralized networking)
* **Storage**: SSD preferred, 256GB minimum (for caching IPFS data and running local nodes)
* **Network**: Stable broadband internet connection for peer-to-peer communications
* **Display**: 13-inch or larger screen, 1366x768 resolution minimum (recommended Full HD for better UI experience)
* **Keyboard & Mouse**: Standard USB or wireless peripherals

## **4.2 Software Specification**

The development and deployment of the application require the following software environment:

* **Frontend Development**: React.js, Redux.js (for user interface and state management)
* **Backend Development**: Node.js with Express.js (for server-side API orchestration and encryption services)
* **Cryptography**: Kyber (for quantum-safe key exchange), AES-256-GCM (for message encryption)
* **Networking**: libp2p and Kademlia DHT (for decentralized peer-to-peer networking)
* **Storage**: IPFS (InterPlanetary File System) for decentralized message storage
* **Identity Management**: DID (Decentralized Identifiers) linked with OTP-based verification
* **Voice/Video Communication**: WebRTC (for secure peer-to-peer real-time communication)
* **Scalable Relay Option**: Waku v2 (for efficient decentralized message relaying)
* **Database**: MongoDB / MySQL (for secure metadata storage, encrypted at rest)
* **Code Editor**: Visual Studio Code (preferred)
* **Browser**: Latest versions of Google Chrome or Firefox (for testing WebRTC and libp2p integrations)
* **Version Control**: Git with GitHub/GitLab for collaborative development

## **4.3 Functionality Requirements**

### **1. Quantum-Secure Messaging**

* Enable real-time, end-to-end encrypted text messaging with Kyber-secured key exchanges and AES-256-GCM encryption.
* Ensure complete forward secrecy and tamper-resistance across all communications.

### **2. Peer-to-Peer Networking**

* Implement decentralized messaging via libp2p protocols without centralized servers.
* Achieve peer discovery using Kademlia DHT for efficient routing.

### **3. Decentralized Storage**

* Utilize IPFS to store message content in a distributed manner.
* Ensure message retrieval and publishing are resistant to censorship and node failures.

### **4. User Identity Management**

* Assign users unique DIDs linked to their mobile numbers through OTP authentication.
* Securely store and manage public keys and metadata in encrypted databases.

### **5. Secure Voice and Video Calling**

* Integrate WebRTC to provide real-time, quantum-safe encrypted voice and video communication.

### **6. Scalability and Message Relay**

* Support optional Waku v2 integration to relay encrypted messages efficiently over large networks.

### **7. Cross-Platform Compatibility**

* Ensure the application works across major operating systems (Windows, Linux, macOS) and is mobile-optimized for Android/iOS.

## **4.4 Non-Functional Requirements**

### **Performance**

* The application should support sending/receiving encrypted messages with minimal latency.
* Handle up to 100+ simultaneous peer connections with smooth encryption and decryption operations.

### **Reliability**

* Guarantee consistent delivery of messages even with fluctuating network conditions.
* Resilient against single-point failures due to decentralized storage and networking.

### **Usability**

* Provide an intuitive and responsive user interface using React.js.
* Offer seamless onboarding with simple DID creation and OTP-based authentication.

### **Scalability**

* Allow for easy expansion to accommodate more users, additional decentralized storage nodes, and enhanced WebRTC group calling features.
* Modular codebase to easily integrate new cryptographic standards or scaling solutions.

### **Security**

* All encryption and decryption should happen locally on the user’s device to maintain absolute privacy.
* Ensure protection against browser vulnerabilities (e.g., XSS, CSRF) during WebRTC signaling and UI interactions.

### **Maintainability**

* Codebase must be modular, well-documented, and ready for future updates such as multi-language support, new DID standards, or next-generation post-quantum algorithms.
* Ensure easy debugging, monitoring, and logging with minimal overhead on decentralized communications.

**CHAPTER 4**

**TECHNICAL FEASIBILITY**

The proposed **Quantum-Secure Messaging Application** is highly feasible given the rapid progress in **post-quantum cryptography**, **decentralized networking**, and **web-based secure communication technologies**. With the availability of advanced tools and standards, the technical implementation of a secure, scalable, and resilient messaging platform is both practical and timely.

At the core of the system is **Kyber**, a NIST-approved **quantum-resistant encryption algorithm**, which provides secure key exchanges immune to quantum-based attacks. Combined with **AES-256-GCM**, a robust symmetric encryption standard, the application ensures high-speed **end-to-end message encryption** with minimal computational overhead.

To eliminate reliance on centralized servers, the system leverages **IPFS** for **decentralized storage** and **libp2p** for **peer-to-peer communication**. These technologies enable secure and scalable data routing, while **Kademlia DHT** ensures efficient peer discovery across distributed networks. All cryptographic and network operations are performed locally, further enhancing data privacy and resilience.

**Decentralized Identifiers (DIDs)** are used to manage user identities, supported by a simple and secure **OTP-based authentication** mechanism. This avoids the need for central user databases, while still enabling verifiable and secure login processes. Metadata and keys are securely stored in **encrypted MongoDB or MySQL databases** for fast, protected access.

The system supports **real-time voice and video communication** using **WebRTC**, secured through the same quantum-resistant principles. To handle high user loads and ensure message relay efficiency, **Waku v2** is optionally integrated, enabling robust **decentralized message relaying**.

The application is developed using a modular and maintainable tech stack, including **React.js**, **Redux**, **Node.js**, and **Express.js**, ensuring scalability, cross-platform compatibility, and smooth user experience. Development and maintenance are handled in **Visual Studio Code**, with modular code design facilitating future enhancements such as **multi-language support** or **biometric integration**.

Minimum hardware requirements are modest — a device with at least **8GB RAM** and a **multi-core processor** is sufficient to run the application smoothly. This ensures accessibility across a wide user base without demanding high-end systems.

In conclusion, the proposed system is technically viable and well-aligned with current advancements in secure communication. It offers a robust, privacy-focused platform that is resistant to both current and emerging cybersecurity threats, especially those posed by future quantum computing capabilities.

**References**

1. **ChatGPT**: [https://chat.openai.com](https://chat.openai.com/)
2. **MDN Web Docs**: [https://developer.mozilla.org](https://developer.mozilla.org/)
3. **React**: [https://reactjs.org](https://reactjs.org/)
4. **React Router DOM**: [https://reactrouter.com](https://reactrouter.com/)
5. **Axios**: [https://axios-http.com](https://axios-http.com/)
6. **html2pdf.js**: <https://github.com/eKoopmans/html2pdf.js>
7. **jsPDF**: <https://github.com/parallax/jsPDF>
8. **Node.js**: [https://nodejs.org](https://nodejs.org/)
9. **Express.js**: [https://expressjs.com](https://expressjs.com/)
10. **dotenv**: <https://github.com/motdotla/dotenv>
11. **mysql**: <https://github.com/mysqljs/mysql>
12. **Nodemailer**: [https://nodemailer.com](https://nodemailer.com/)
13. **Nodemon**: [https://nodemon.io](https://nodemon.io/)
14. **Twilio**: [https://www.twilio.com](https://www.twilio.com/)
15. **otp-generator**: <https://www.npmjs.com/package/otp-generator>