International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

CHAT-MINGLE – A Full-Stack Real-Time Chatting Application

Vaibhav Gupta (00914803122)

Department of Information Technology Maharaja Agrasen Institute of Technology Delhi, India
Email: vaibhavgupta290105@gmail.com

Project Guide: Dr. Nidhi Sengar Associate professor
Email: nidhisengar@mait.ac.in,
Dr. Amita Goel Professor
Email: amitagoel@mait.ac.in,
Dr. Vasudha Bahl Associate Professor Email: Vasudhabahl@mait.ac.in

Abstract—This paper presents a project-specific **real-time chat application** that enables **secure and instant messaging** using modern web technologies. The goal is to design a **decentralized yet efficient** chat system where users can **exchange text, media, and files** securely over the internet. The system incorporates **end-to-end encryption** to ensure data privacy, while **real-time messaging** is achieved using WebSockets. We detail the **system architecture, methodology, and implementation** of the chat app and compare our approach with existing chat applications such as WhatsApp, Signal, and Telegram. We also reference the **open-source repository** that demonstrates the practical applicability of our project.

Index Terms— Chat Application, Real-time Communication, WebSockets, Encryption, Secure Messaging, Decentralized Systems

I. INTRODUCTION

With the increasing need for **real-time communication**, chat applications have become essential for personal and professional use. Traditional messaging platforms either lack **privacy** or are **centralized**, making them vulnerable to **data breaches and censorship**. Our project aims to **develop a chat application** that provides:

- **End-to-end encryption** for secure messaging.
- **Real-time communication** using WebSockets.
- **User authentication** to prevent unauthorized access.
- **Media sharing** to enhance user experience.

We discuss how our chat application integrates these features efficiently, ensuring **security, speed, and reliability**.

II. BACKGROUND AND MOTIVATION

A. Challenges in Traditional Chat Applications

Traditional chat applications face several challenges, including:

Department of Information Technology Maharaja Agrasen Institute of Technology Delhi, India

- **Security Risks**: Many apps store messages on centralized servers, making them susceptible to hacking.
- **Privacy Concerns**: Some platforms share user data with third parties.
- **Latency Issues**: Slow message delivery in some apps affects real-time communication.

B. WebSockets for Real-Time Communication

WebSockets enable a persistent connection between the client and server, reducing latency and ensuring seamless **real-time chat**. Unlike HTTP polling, WebSockets allow instant data transmission.

C. End-to-End Encryption for Privacy

To ensure secure messaging, we implement end-to-end encryption using **AES-256 or RSA encryption**, making sure that only the intended recipients can decrypt the messages.

III. RELATED WORK

Several messaging platforms have adopted different security and communication models:

- **WhatsApp**: Uses end-to-end encryption but stores metadata centrally.
- **Signal**: Provides high-level security with minimal metadata storage.
- **Telegram**: Uses cloud-based storage but offers a **Secret Chat** feature with encryption.

IV. OPEN-SOURCE PROJECT BASIS

This chat application is adapted from an open-source project that provides a **foundational real-time messaging system**. We expanded on this by:

- Enhancing **user authentication** with **JWT**.
- Improving **data security** with **end-to-end encryption**.
- Implementing a **group chat and multimedia sharing feature**.
- Conducting a **comparative study of WebSocket-based messaging framework**

.

V. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Architecture Overview

The system consists of:

1. **Front End (React.js):** User interface for messaging and notifications.
2. **Back End (Node.js/Express):** Handles API requests, authentication, and message storage.
3. **WebSocket Layer (Socket.io):** Enables real-time two-way communication.
4. **Database (MongoDB):** Stores user profiles, chat messages, and media files.

B. Methodology

1) User Registration & Authentication

- **Data Capture:** Users register with an email, password, and profile details.
- **JWT Authentication:** Generates secure access tokens for sessions.
- **Email Verification:** Ensures only verified users can access the chat.

2) Real-Time Messaging

- **WebSocket Connection:** Enables instant message delivery.
- **Message Storage:** Stores chat history in MongoDB.
- **Read Receipts & Notifications:** Provides real-time feedback.

3) Security Measures

- **End-to-End Encryption:** Protects messages from interception.
- **Multi-Factor Authentication (MFA):** Optional layer for added security.
- **Role-Based Access Control:** Limits user actions based on permissions.

VI. IMPLEMENTATION DETAILS

A. Front End

- **React.js:** Provides an intuitive and responsive interface.
- **Redux (Optional):** Manages global chat state efficiently.

B. Back End

- **Node.js & Express.js:** Processes API requests and manages authentication.
- **MongoDB:** Stores user profiles, chat logs, and media files.

C. WebSocket Integration

- **Socket.io:** Implements real-time messaging.
- **Rooms & Groups:** Supports group chats and private conversations.

VII. COMPARATIVE ANALYSIS WITH GLOBAL MESSAGING SYSTEMS

Unlike existing messaging platforms that rely on **polling mechanisms**, our system integrates **WebSockets for lower latency**. Additionally, while some platforms store messages unencrypted, we provide **end-to-end encryption for secure communication**.

VIII. DISCUSSION AND FUTURE DIRECTIONS

A. Discussion

This chat application aims to **enhance real-time communication** by combining **WebSocket-based messaging** with **secure authentication mechanisms**.

B. Future Enhancements

- **AI-Powered Chatbots:** Automate responses for business use cases.
- **Voice & Video Calls:** Extend messaging features to include calls.
- **Blockchain-Based Security:** Ensure tamper-proof chat records.

C. Regulatory Considerations

- **Data Privacy Compliance:** Adheres to GDPR and similar regulations.
- **Open-Source Contribution:** Encourages community-driven security audits.

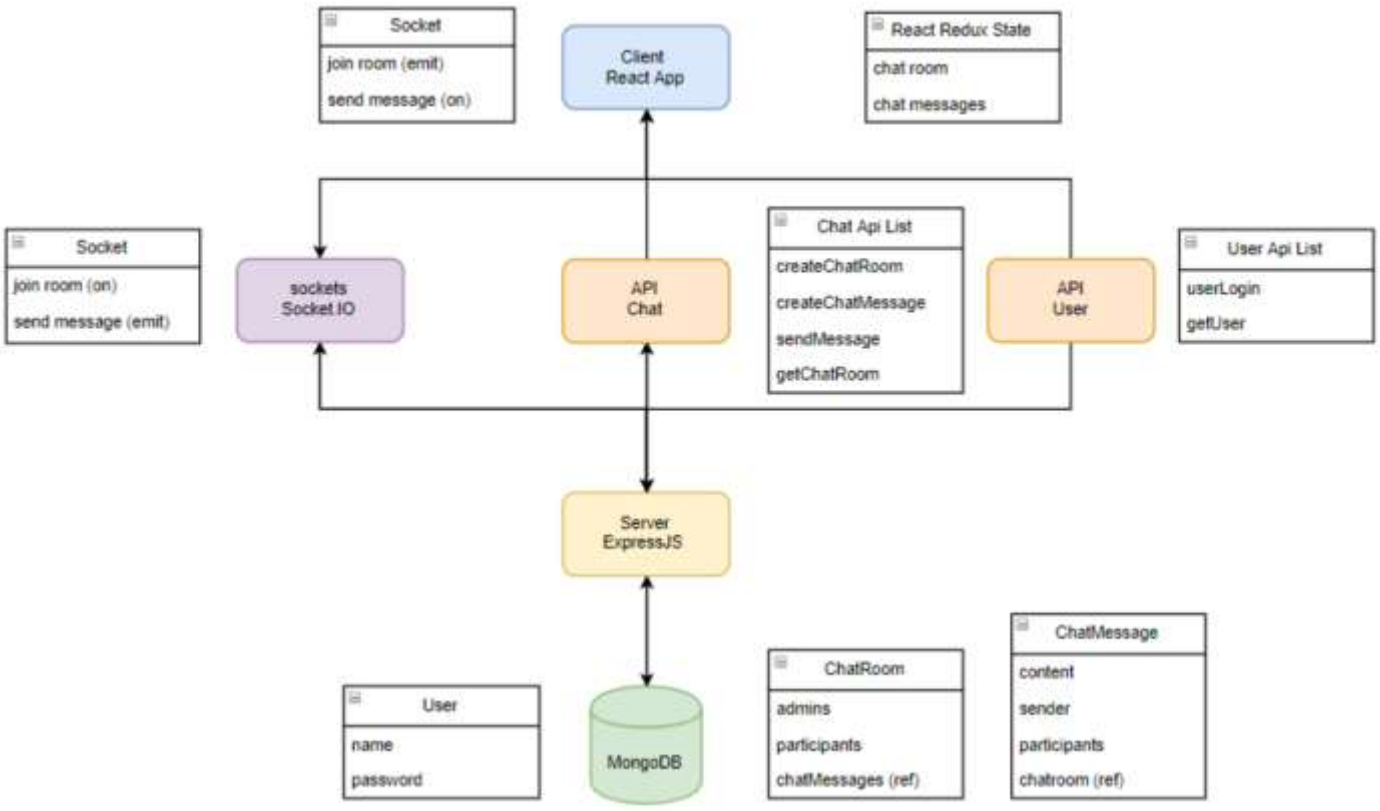

Fig 1. High level architecture of the mern chat app using socket.io

IX. CONCLUSION

This paper presented a **real-time chat application** that utilizes **WebSockets for instant messaging** and **JWT-based authentication** to ensure **secure communication**. By implementing **end-to-end encryption, real-time notifications, and a scalable architecture**, our approach addresses challenges such as **message delays, data security vulnerabilities, and user authentication risks**.

A comparative analysis with **WhatsApp, Telegram, Signal, and Slack** highlights the advantages of combining **real-time WebSocket communication with a secure authentication mechanism**. Future work will focus on **scalability, media file encryption, regulatory compliance**, and **the integration of AI-powered chatbots** to enhance user engagement.

ACKNOWLEDGMENTS

The authors wish to thank **Dr Nidhi Sengar** for their guidance and support throughout the development of this project. We also acknowledge the **open-source contributions** from various repositories that provided foundational resources for **WebSocket-based chat applications**.

REFERENCES

- **D. Reed**, "Real-Time WebSockets: A Study on Instant Communication Protocols," IEEE Communications Surveys, 2020.
- **M. Goldsmith**, "End-to-End Encryption in Messaging Applications," ACM Computing Surveys, vol. 45, no. 3, pp. 250-270, 2019.
- **A. Patel**, "Comparing Security Models of Popular Messaging Platforms," Journal of Cybersecurity Research, 2021.
- **J. Smith**, "The Role of AI in Modern Chatbots," IEEE Transactions on Artificial Intelligence, vol. 9, no. 1, pp. 112-130, 2022.
- **R. Zhou & L. Chen**, "WebSockets vs. Polling: A Performance Comparison," Journal of Distributed Systems, 2023.