International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 1239-1243 www.ijaem.net ISSN: 2395-5252

Privacy-Preserving Real Time Chat Application

Shreyash Srivastav, Abhishek Chaurasia, Dr. Vibha Srivastava, Dr. Atul Kumar

Department of Information Technology, Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, Uttar Pradesh

Date of Submission: 15-04-2023 Date of Acceptance: 25-04-2023

ABSTRACT— In today's time, real time chat applications have become a most common medium for communication, enabling instant messaging and social interaction in various domains like business, education, social networking, etc. One of the most important factor that users look for, in such chat applications, is their privacy. This research paper focuses on privacy-preserving real time chat applications, which aims to protect the confidentiality and integrity of messages, at the sametime maintaining real-time communication capabilities.The advantages and limitations of each technique are discussed, along with their applicability in different scenarios. Moreover, the paper also addresses the challenges and considerations in designing and implementing privacy-preserving real time chat applications.

This chat application is going to be a multi-user chat application and it will not be storing the messages data. The key technologies used in building this chat application are Node JS, Express JS, WebSockets, React JS and MongoDB.

Keywords—Real time chat application, Multi-user chat application, Privacy-preserving chat, Web sockets, Node.js, React.js, Socket.io chat application, Express.js

I. INTRODUCTION

In today's fast-paced world, communication has evolved tremendously, and chat applications have become an integral part of our daily lives. These applications have revolutionized the way we connect with others, making communication faster, more convenient, and more accessible than ever before. From personal conversations to business collaborations, chat applications have transformed the way we interact, share information, and build relationships.

Chat applications are software programs that allow users to send and receive messages in real-time, either one-on-one or in group settings, using text, images, videos, and even voice or video calls. These applications have become increasingly popular due to their ease of use, instant messaging capabilities, and availability across various devices and platforms, including smartphones, tablets, and computers. They have become an essential tool for staying connected with friends, family, colleagues, and clients, regardless of geographical boundaries or time zones.

One of the key advantages of chat applications is their ability to facilitate real-time communication, enabling users to exchange messages and information instantaneously. Gone are the days of waiting for letters or emails to arrive; now, we can send and receive messages in seconds, making conversations more dynamic and engaging. Moreover, chat applications often offer features such as read receipts, typing indicators, and emojis, which enhance the overall communication experience and help convey emotions and sentiments effectively.

Another significant benefit of chat applications is their versatility and adaptability to different communication needs. They are widely used for various purposes, ranging from personal conversations with friends and family to professional communication in the workplace. Chat applications are commonly used for team collaborations, project management, customer support, and even for conducting business transactions. They offer features such as file sharing, document collaboration, and integrations with other productivity tools, making them a powerful tool for enhancing productivity and efficiency in both personal and professional settings.

One notable feature of some chat applications is their approach to data storage, where messages are not stored in a database. This unique characteristic ensures that conversations are not stored or retained, providing users with increased privacy and security. In this introduction, we will explore the concept of real-time chat applications that do not store message data in a database, examining their benefits and implications for users in terms of privacy, security, and data management. In this chat application, the tech stack used are, Node.js, Socket.io, Express.js, React.js and MongoDB.

Node.js is a software platform that is used to build server side flexible applications in a network application. Socket.IO is a javascript library is an implementation of websocket protocol and various other necessary improvisation for real-time web. JavaScript is a language used to create a program so the HTML document displayed in the browser becomes more interactive. MongoDB is an open source database based on document (Document-Oriented Database) which was originally created in C ++. Therefore, it is expected this chat application can run in real time.

II. MAIN PROBLEM

As described above, the main problem in the chat applications is the privacy of the messages sent and received. In today's scenario, although there are various chatting applications that claim about end-to-end encryption of the messages, but in reality there is always a kind of suspicion, that our messages are getting stored at some place. This also sometimes has led to breach of the data.

Therefore, in this paper the solution for the above problem is discussed and has been portrayed a method, using which we can overcome the above mentioned problems.

III. OBJECTIVE

The objective can be defined from the main problem above that is, building a dedicated real-time chat application which can be used by multiple users and ensures their and their messages privacy.

IV. SYSTEM DESIGN

This chat application will be based on rooms, where users will join the room and chat with the other online users present in that room. The system will feature the following functionalities:

i. Create room
ii. Join room
iii. Delete room

Room admin is going to be the person, who will be creating the room and share the room credentials with the other users, whom he/she wants to join his/her chat room. In this, infinite number of users can join the room and chat with the other users.

The chat application will also feature a BOT, that will notify users in the room, if any new user joins/leaves the same room.

This system will be using Node.js, Express.js, Socket.io, React.js and MongoDB. Server side services are using Node.js along with Express.js framework, Socket.io for low-latency, bidirectional and event-based communication between a client and a server and MongoDB for storing room data. Client side is built using React.js.

A. BLOCK DIAGRAM

Here is the block diagram of server – client communication:

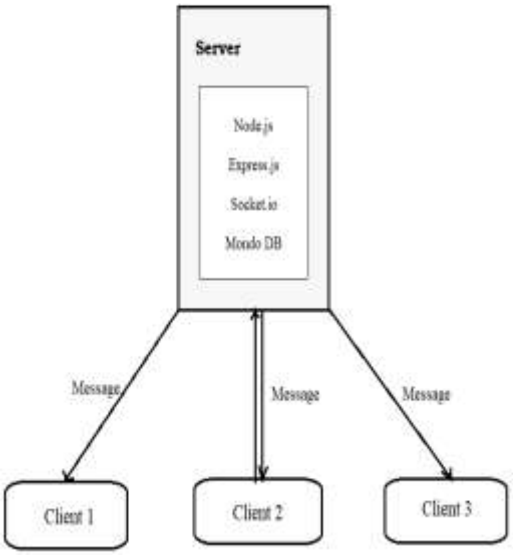


Fig. 1 Server-client block diagram

B. FLOW CHART

Here is the flow chart of the overall process:

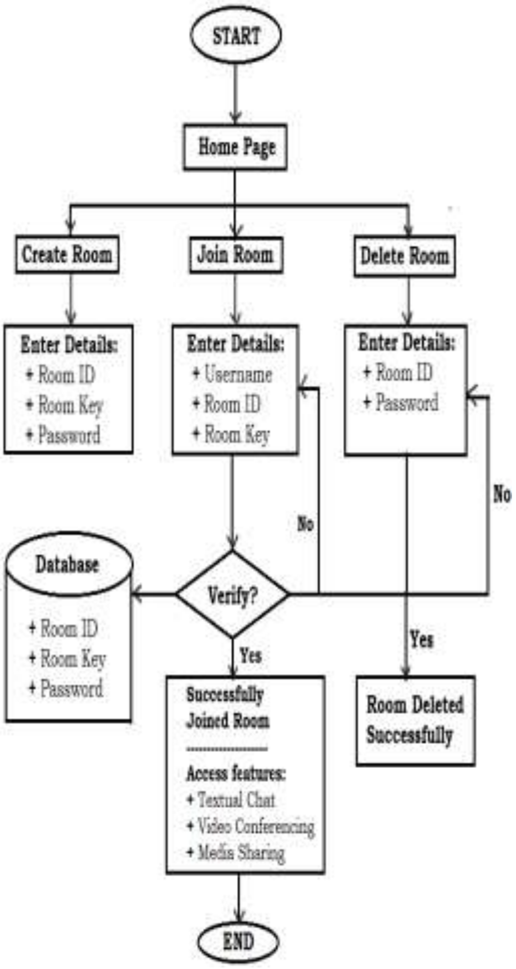


Fig. 2 Flow chart of web based chat application

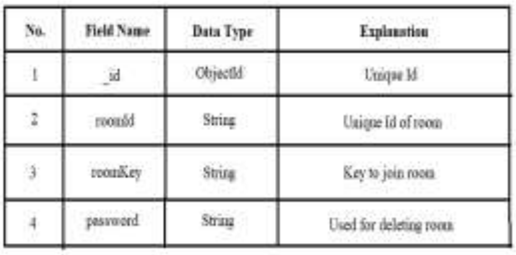
The explanation for each flow is as follows:

1. When the home page opens, it presents three options – create room, join room and delete room.
2. For creating room, one needs to create the following credentials – Room ID, Room Key and Password. These data gets stored in the database in the encrypted form. The one who creates room is the admin.
3. After the room is created, admin shares the room credentials with other users to join room. For joining room, user needs to enter his/her Username, Room ID and Room Key.
4. If the credentials entered at the time of joining are not correct, you will be displayed an error message, or else if correct, you will be redirected inside and room access will be granted.
5. For deleting the room, admin needs to enter the following – Room ID and Password. Password is known to the admin.

C. DATABASE TABLE STRUCTURE

Here is the database structure for the chat rooms:

TABLE 1
CHATROOM DATABASE STRUCTURE

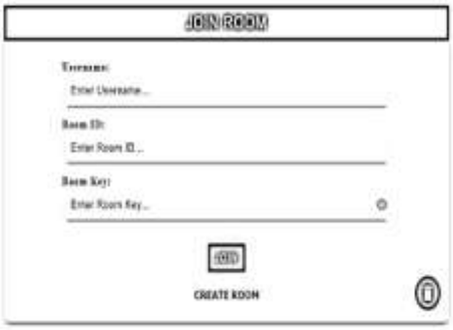


No.	Field Name	Data Type	Explanation
1	_id	ObjectId	Unique Id
2	roomId	String	Unique Id of room
3	roomKey	String	Key to join room
4	password	String	Used for deleting room

D. USER INTERFACE DESIGN

This web based chat application has the following interfaces – join room, create room, delete room and chat room.

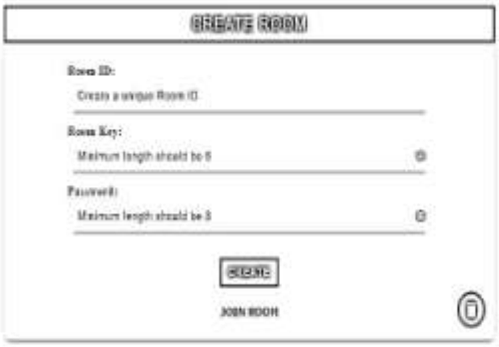
Here is the user interface design of join room page:



Username:
Enter Username...
Room ID:
Enter Room ID...
Room Key:
Enter Room Key...

Fig. 3 User interface design of join room page

Here is the user interface design of create room page:



Room ID:
Create a unique Room ID
Room Key:
Maximum length should be 8
Password:
Maximum length should be 8

Fig. 4 User interface design of create room page

Here is the user interface design of delete room page:

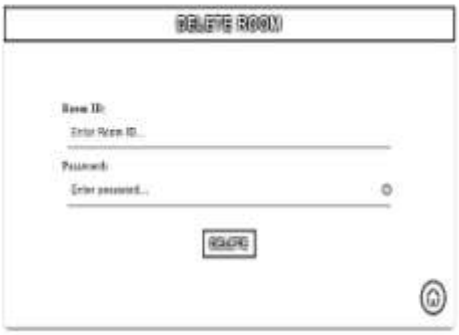


Fig. 5 User interface design of delete room page

Here is the user interface design of chat room:

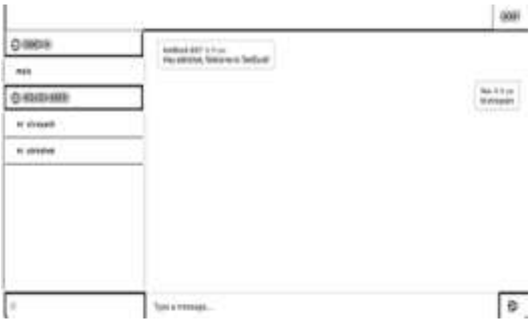


Fig. 6 User interface design of chat room

V. IMPLEMENTATION

1. CREATE ROOM

The figure shown below is the implementation of the create room, where admin needs to create the following details – Room ID, Room Key and Password, under the condition that Room Key and Password should not be same.

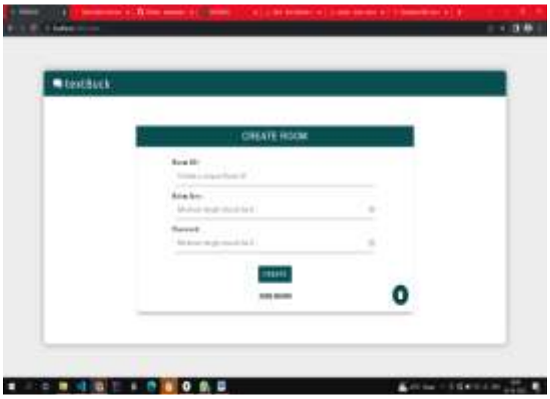


Fig. 7 Implementation of create room page

2. JOIN ROOM

The figure shown below is the implementation of the create room, where admin needs to create the following details – Username, Room ID and Room Key. If the credentials are entered correctly, you will be granted the chat room access.

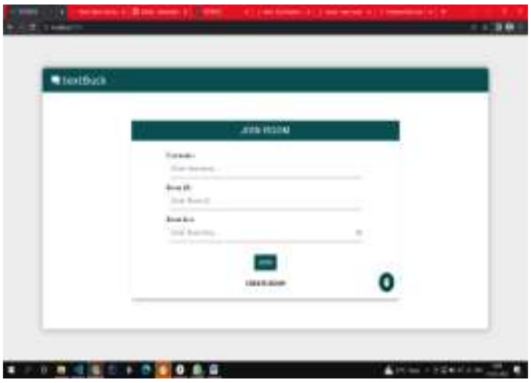


Fig. 8 Implementation of join room page

3. CHAT ROOM

The below image shows the chat rooms implementation. In this multiple-users can join the room at a time.

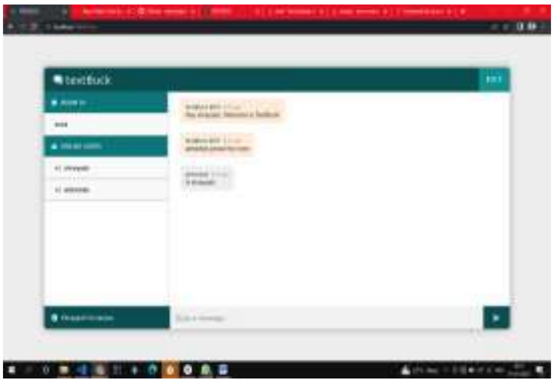


Fig. 9 Implementation of chat room

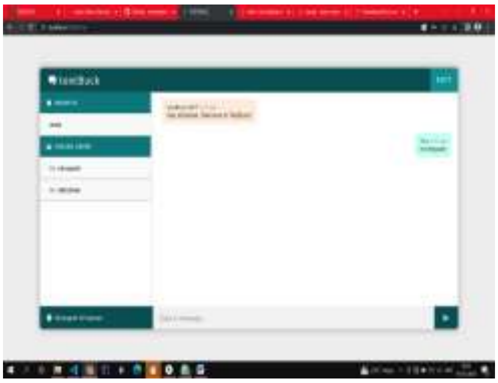


Fig. 10 Implementation of chat room with bot feature

Here, in the above figures, beige color messages are sent by BOT, that sends a welcome message and also notifies if any user joins/leaves the room.

4. DELETE ROOM

The figure shown below is the implementation of the create room, where admin needs to create the following details – Room ID and Password.

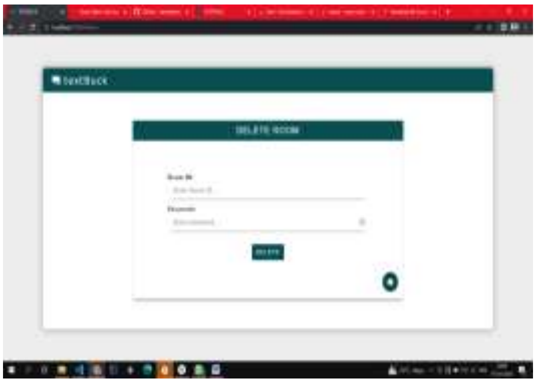


Fig. 11 Implementation of delete room page

VI. CONCLUSION

Therefore, the implementation of the real time chat application that does not store users data and message data, has been done. In conclusion, this research paper has provided an in-depth analysis of real-time chat applications. Through a comprehensive review of existing literature and an examination of various technical aspects, we have gained a nuanced understanding of the challenges and opportunities associated with building and implementing such applications. However, while these applications offer numerous benefits, such as instant messaging, real-time updates, and seamless collaboration, but it does not provides the backup of the messages. To overcome this problem, further research and development is needed to ensure the continued growth and success of real-time chat applications. As technology continues to evolve and user expectations change, real-time chat applications will likely play a pivotal role in shaping the future of online communication, fostering collaboration, and enhancing user experiences. With ongoing innovation, careful consideration of ethical implications, and effective implementation strategies, real-time chat applications have the potential to revolutionize various industries and empower users in new and exciting ways.

Future Work –

- This web based chat application allows users to chat with multiple-users, without saving their data and without storing their messages. Hence, it guarantees users privacy and authenticity.
- Adding some more additional features like video conferencing and screen sharing would increase its productivity to a great level.

REFERENCES

[1]. John Resig, Bear Bibeault, and Josip Maras, Secrets of Javascript Ninja

[2]. Greg Lim, Beginning MERN Stack Development

[3]. Andrew Mead, Advanced Node.js development: Master Node.js by building real-world applications

[4]. Shannon Bradshaw, Eoin Brazil and Kristina Chodorow, MongoDBThe Definitive Guide: Powerful and ScalableData Storage, 3rd edition

[5]. Reynolds.P.(2015). Benefits of online chat system

[6]. https://socket.io/docs/v4/

[7]. https://docs.npmjs.com/

[8]. Yoshihiro Kawahara and Tomonori Aoyama (2004), A Peer-to-peer message exchange schame for large-scale networked virtual environments. The University of Tokyo, Japan

[9]. James D. herbsleb, David G. Boyer, M.Handel, (2002). Introducing instant messaging and chat in the workplace, Minnesota, USA

[10]. AvinashBamane, P. B. (201 2, june 8). Enhanced Chat Application. Double Blind Peer Reviewed International Research Journal, 6-12

[11]. Teixeira, Pedro. 2012. Professional Node.js: Building Javascript Based Scalable Software Kindle Edition. Wrox