

# Real Time Chat Box System using MERN

**Khushi Srivastava<sup>1</sup>, Shashwat Srivastava<sup>2</sup>, Dr. Amit Kr. Pathak<sup>3</sup>**

Students, Department of Information Technology<sup>1,2</sup>

Associate Professor, Department of Information Technology<sup>3</sup>

Raj Kumar Goel Institute of Technology, Ghaziabad, India

**Abstract:** *Chat applications have brought about a revolutionary transformation in the way individuals communicate and interact in the digital age. This research paper aims to delve into the evolution, features, and impact of chat applications on contemporary communication. The study commences with an overview of the historical development of chat applications, tracing their origins from rudimentary text-based platforms to the sophisticated and feature-rich applications that are ubiquitous today. Moreover, this research paper investigates the ramifications of chat applications on various spheres of society, encompassing personal relationships, business communication, education, and healthcare. It examines both the advantages and challenges associated with the utilization of chat applications, including concerns pertaining to privacy, security, and information overload. Overall, this paper provides insights into the significance of chat applications in enhancing communication efficiency and connectivity in today's digital era.*

**Keywords:** Real Time Chat Box, MERN

## I. INTRODUCTION

Research in the field of internet-based real-time collaboration, encompassing text and multimedia interactions, has gained significant attention. However, there is a lack of well-defined applications in this domain. The term "collaborative application" is now used generically to describe any software program that allows users to connect with one another in real-time or near-real-time to share information via text or video. Consequently, numerous programs claim to be collaborative. One popular feature or program, known as a chat room, allows internet users, including those who are constantly online, to engage in direct communication regardless of their geographical distance. For such a feature to accommodate a large user base, it must be capable of solving problems and remain accessible across different platforms and time zones.

Numerous programmers are currently developing chatrooms, each with their own set of advantages and disadvantages. Prior to initiating our project, we conducted an extensive examination of the available messaging systems. While we knew the existence of numerous message services and chat programs, we needed to thoroughly investigate their tools to evaluate if they matched our development needs.

During our investigation, it became apparent that none of the existing platforms aligned perfectly with our objectives. Some lacked crucial features that we think were important, while others gave areas of potential growth or development. We had a look on platforms like Slack, WhatsApp, Telegram, Messenger, Discord, Skype and more. These applications boast billions of users worldwide and are prominent players in the industry. They consistently generate substantial profits and employ large teams to develop new features and keep up with competitors. Moreover, these applications implement diverse measures and protocols to safeguard user data privacy, which is of paramount importance in the face of escalating data theft incidents.

After conducting background research, it became evident that real-time chat applications varied in terms of features. Our objective is to consolidate all essential features, such as sending invitations, displaying online indicators, typing notifications, message storage in a database, chat functionality, audio and video calling, and screen sharing, into a single application. Contrary to common belief, the availability of multiple applications proved beneficial for our project. By examining users' experiences and insights shared through blogs and development updates, we gained valuable knowledge for our own development process. Notably, companies like Slack frequently publish updates and valuable information that informed our decision-making process. Additionally, we often relied on internet research to explore alternatives and determine the most essential features to incorporate.

Throughout our research, we extensively examined the below websites :-

- Flowdock ([www.flowdock.com](http://www.flowdock.com))
- Gitter ([gitter.com](http://gitter.com))
- Hangouts ([hangouts.google.com](http://hangouts.google.com))
- Discord ([discord.com](http://discord.com))
- Messenger ([messenger.com](http://messenger.com))
- Rocket.chat ([rocket.chat](http://rocket.chat))
- Skype ([web.skype.com](http://web.skype.com))
- Slack ([slack.com](http://slack.com))
- Telegram ([web.telegram.org](http://web.telegram.org))
- WhatsApp ([web.whatsapp.com](http://web.whatsapp.com))

This thorough exploration allowed us to gather insights.

Among the applications we examined, Gitter and Slack were the primary focus. While services like Rocket.chat provided basic features such as code sharing, it was apparent that these platforms did not cater specifically to professionals. Consequently, they lacked the essential tools required by professionals in their respective fields. We believe that professionals require a dedicated environment tailored to their needs, where code sharing becomes an enjoyable experience and where seamless integration with their source code repository is possible.

It became evident that our research should primarily focus on Slack and Gitter. The other sites, while providing some general insights, were not immediately applicable to our study. Slack and Gitter are two well-known systems that place extreme value on developers and productivity., have demonstrated long-standing success. Although none of them were open-source, our program will distinguish itself from these two platforms by incorporating unique aspects and features.

In a global context, developers are actively striving to enhance the user experience and optimize their product offerings. Engineers from around the world are diligently working to improve application experiences and streamline development workflows, ensuring timely delivery of applications. Web development stacks play a crucial role in rapidly creating web applications, serving as a software developer's response to the evolving industry requirements. These stacks leverage current frameworks, such as JavaScript, to simplify the development process. Well-known stacks like MEAN and MERN have emerged from the JavaScript ecosystem, although there exist numerous other stacks to cater to diverse needs in web development.

Both the MEAN and MERN stacks are built using open-source components and provide a comprehensive foundation for developing sophisticated web applications that enable user interaction. One of the notable advantages of utilizing these stacks is the integration of JavaScript, which eliminates concerns about misinterpretation or grammatical errors while coding. Moreover, the MERN stack offers increased flexibility in web application development. It comprises four key components: MongoDB, Express.js, React, and Node.js. These elements, when combined, enable programmers to develop dynamic and advanced web applications.

## II. LITERATURE REVIEW

This research investigated the structure and design of a real-time collaboration programme that allows many users to see and produce coordinated versions of files and other materials through a web browser. In addition, the application enables the integration of files with the document being changed. The example showed how users can collaborate on a rich-text document with an embedded synchronized video player Smart Object. The proposed solution distinguishes itself from other collaboration choices by emphasizing user-friendly chat elements and utilizing a flexible design that allows experts to simply include novel experiences utilizing the Stream Container API..

Encoding user messages on the server is essential for maintaining communication privacy. When messages are stored or received by unauthorized sources, encryption ensures that they remain in an unreadable format without any meaningful information. Decrypting these encrypted messages requires the use of a secret key, also known as the

cypher key, which is only accessible by authorized systems. Only users with the relevant authorization can access this key, which is required for message decoding. Even if unauthorized parties obtain access to the encrypted messages, they will be unable to decrypt the initial message without the secret key. Within the system, the message exchanged between users gets encrypted. The sender encrypts the message before transmission, and only the intended recipient with the cipher key can decrypt it. This robust encryption process ensures that unauthorized messages cannot traverse system, as a result, both system security and user privacy are preserved.

In a research article published by R. Gayathri and C. Kalieswari in 2020 in the International Journal of Engineering and Advanced Technology (IJEAT) [15], the authors present their work on a chat application that offers an enhanced and versatile platform for discussions. The application is developed using state-of-the-art technologies to ensure a reliable system. The key advantages of this system feature group chat, enhanced security, real-time collaboration, and instantaneous messaging capabilities. This chat application holds significant potential in the market, particularly for businesses seeking secure and private communication solutions. Additionally, the research article mentions that future updates will incorporate new elements like conference calls and video chat based on community feedback, as well as additional functionalities such as location sharing as per user requirements.

The paper titled "Designing and implementing a real-time web-based chat server" by Diotra Henriyan, Devie Pratama Subiyanti, and Rizki Fauzian was published in 2016 in the International Conference on Engineering and Technology (ICSET) [1]. To appeal to a broad user base, a good chat programme should have a live forum and support many sites. The authors emphasize the use of programming languages such as MongoDB for creating the website and Node.js for establishing a robust server foundation. The necessity of these components in constructing a real-time web-based chat server is emphasized in the study paper.

The article "Online Chat Application [14]" discusses a permission-based programme that allows users to access their accounts from any mobile device, at any location, and at any time. The authors emphasize the need for users to initiate a conversation request before engaging with another user. Communication is only possible when both parties grant approval. Notably, the application does not require OTP (One-Time Password) authentication; instead, users can log in using their email address and password. The article showcases the effectiveness of internet communication, enabling users to engage in rapid and efficient conversations. It highlights the significance of conversational applications in modern life and their impact on the technological landscape. This research demonstrates the value and benefits of such applications in facilitating convenient and streamlined communication experiences.

Despite the previously discussed paper, an article [15] introduces a discussion system for private networks or organizations. This system prioritizes the security of shared personal information and network interactions. It ensures the security of data kept in the backend. The established communication system is bidirectional, including parts from various conventional systems as needed. It facilitates convenient and rapid communication between persons by allowing for both group and private conversations. Furthermore, this method allows for infinite data transfer with no size constraints, allowing for connections with others at any time and from any location. The system also offers ample space for storing communication data, accommodating various forms of information exchange.

Another networking program [16] introduces various features, such as user-friendly social networking, Image backup, image theft warnings, landmark identification, and image information retrieval are all available. This program utilizes the React framework to create interactive interfaces for users. It also includes real-time messaging capabilities on web browsers, there is no longer any requirement for additional third-party account programmes to establish visual communication. The Express.js framework is used in conjunction with React.js and Node.js to create the application. It also incorporates the Mongo DB database for efficient data storage and retrieval.

With the advancement and improvement of the internet, network chatting tools have become increasingly popular for communication. These applications allow for simple communication over long distances. To ensure that users utilize an application properly, it should be cross-platform and provide real-time communication. No additional user information from third-party sources is necessary to establish visual communication using a web-based real-time chatting programme.

Point-to-point connections between servers permit data transport, and text communication is exchanged between these servers. The use of the virtual space concept in the React framework greatly improves performance as compared to PHP-developed apps, resulting in approximately six times higher performance.

### **III. DESIGNS AND IMPLEMENTATION**

This study's research design depends on qualitative analysis. Instead of examining many research works, we have subjectively picked key contributions in the field of real-time chat application development. These apps often mix socket.io with frontend languages such as HTML, CSS, and others. It is critical to ensure that the project is adaptable and scalable, as there is a potential need for future expansion. Extensive literature reviews have been conducted to identify various methodologies employed by developers in the development of chat applications.

The project design will adhere to the MERN stack development approach. This entails utilizing MongoDB as the database, Express.js as the backend framework, React for frontend development, and Node.js as the runtime environment. The MERN stack provides a solid foundation for creating a robust and scalable real-time chat application. For implementation, this project makes use of React.js as the frontend framework. Facebook created React.js, is chosen primarily for its reusable component feature. This feature enables efficient rendering of components without the need to recreate or load them repeatedly. Take Instagram as an example, where the stories feature utilizes reusable components that are rendered for each story. By employing React.js in the frontend, the application can achieve faster performance by reusing components as needed. This approach not only improves speed but also enhances extensibility, allowing for easy addition of new components when required.

Socket.io is a popular JavaScript-based library for online applications. It comprises a server library written in Node.js and a client library that runs in web browsers. Socket.io generally uses the WebSocket protocol from the client's perspective, but it may also fall back to alternative approaches such as Adobe Flash sockets, AJAX long polling, and others to preserve a consistent interface. Socket.io provides a full range of sophisticated features, including the ability to handle asynchronous I/O, broadcast messages to many sockets, save client-specific data, and create server-side rooms to connect numerous connections.

The Node Package Manager (npm), a tool for managing Node.js modules, makes it simple to install Socket.io. Peer instances for the sender and recipient occur in various web browsers in the context of a chat application. A "signaling server" is frequently required prior to establishing a peer-to-peer connection to permit the exchange of signal data between the two browsers. The simple peer library is used for peer-to-peer WebRTC connections.

In summary, Socket.io is a versatile library that enables real-time communication in web applications, providing a range of capabilities and compatibility with different protocols and fallback options.

In addition to Socket.io, the backend of the application utilizes Node.js. Node.js is a JavaScript-based runtime built on the Chrome V8 engine. It serves as the foundation for developing scalable web servers and remains at the core of the application's architecture. The Node Package Manager (npm) is used as the repository for all the required packages, enhancing the development process.

Node.js is critical in enabling the construction of real-time JavaScript applications, making it an essential component of our real-time chat platform. It simplifies the handling of real-time messages, allowing users to communicate in real time. As a result, Node.js is required for the chat application's operation. The routing API in our project is provided by Express.js, a Node.js framework. It plays a key role in the communication flow between the frontend, API, and backend. The messages are sent from the frontend to the API, which is then responsible for processing and forwarding them to the backend. Express.js enhances the performance of the application and provides convenient integration with third-party plugins, enabling faster execution of tasks.

Based on previous literature, the real-time chat applications varied in terms of speed and compatibility with different devices. However, by utilizing these major components in our application, we aim to significantly improve its performance compared to previous applications. The website developed using this approach is expected to deliver faster and more efficient real-time communication.

The backend of the application is connected to the database, and after considering various options available for real-time chat applications, MongoDB has been chosen for our Chatterbox chat application. MongoDB is a NoSQL document-oriented database that offers several advantages. It is a cross-platform programme, assuring platform compatibility and making it a versatile alternative for a variety of applications. Scalability is one of the primary reasons for choosing MongoDB. As a NoSQL database, it can handle unstructured data efficiently and allows for easy scalability as per the application's needs. This scalability feature enables us to extend the application as much as required without compromising its performance. Another advantage of using MongoDB is that it is always available.

Unlike other applications, we don't need to host it as a separate server, and it is maintained 24/7. This ensures continuous access to the database, enhancing the reliability of the application.

The flexibility of MongoDB's schema is another factor that influenced our decision. Its NoSQL nature and flexible schema enable us to manage data effectively, accommodating changes and updates in the application's structure with ease.

In summary, MongoDB has been selected as the database for our real-time chat application due to its cross-platform compatibility, scalability, 24/7 availability, and flexible schema, providing a reliable and efficient foundation for the application.

#### IV. CONCLUSION

In conclusion, this research paper has delved into the development and design aspects of real-time chat applications. Through a qualitative analysis and an extensive review of pertinent literature, essential methodologies and technologies have been identified for constructing such applications. The project design revolves around the employment of a MERN stack, with React.js serving as the frontend language. The reusable component feature of React.js enhances the application's performance and extensibility, rendering it a fitting choice for modern chat application development. Socket.io, a JavaScript-based library, proves instrumental in establishing real-time communication between the server and client. Its versatility in employing various techniques, including WebSockets and AJAX long polling, ensures dependable and efficient communication across diverse platforms. On the backend, Node.js serves as a runtime environment, providing scalability and facilitating the development of real-time applications based on JavaScript. Express.js, a Node.js framework, acts as the routing API, seamlessly connecting the frontend, API, and backend components. For efficient data storage, MongoDB, a NoSQL document-oriented database, has been selected. Its scalability, flexibility, and cross-platform compatibility make it an ideal choice. MongoDB enables efficient management of unstructured data and obviates the need for hosting a separate server, ensuring uninterrupted availability.

Collectively, this research emphasizes the significance of considering various technologies and methodologies when developing real-time chat applications. By harnessing the strengths of React.js, Socket.io, Node.js, Express.js, and MongoDB, developers can create versatile, extensible, and efficient chat applications. The findings of this research contribute to the advancement of chat application development and underscore the importance of real-time communication in today's ever-evolving technology landscape.

#### V. ACKNOWLEDGMENT

I am delighted to offer the research report completed during my final year of B. Tech. I'd like to offer my heartfelt gratitude to Dr. Amit Kumar Pathak, Associate Professor at Raj Kumar Goel Institute of Technology, Ghaziabad, for his invaluable insight and skills, as well as for sharing his pearls of wisdom with me throughout the project. He was crucial in determining the path. I am incredibly grateful to him for this. I'd like to express my heartfelt gratitude to "The Department of Information Technology, RKGIT" for their invaluable assistance in completing the research. Although it is impossible to name everyone individually, please know that your efforts, no matter how large or small, were critical to the completion of this research study.

#### REFERENCES

- [1] Diotra Henriyan, Dvie Pratama Subiyanti, Rizki Fauzian, Dian Anggraini, M. Vicky Ghani Aziz, Ary Setijadi Prihatmanto. (2016). Design and Implementation of Web Based Real Time Chat Interfacing Server.
- [2] Rohitha Pasumarty, Raja Praveen K. N. (2021). Secure Chatroom Application using Advanced Encryption Standard Algorithm.
- [3] Bogdan Ionescu, Cristian Gadea, Bogdon Solomon, Mircea Trifan, Dan Ionescu (2015). A chat-centric collaborative environment for web-based real-time collaboration.
- [4] Schillinger, F. and C. Schindelhauer.(2020). Partitioned Private User Storages in End-to-End Encrypted Online Social Networks.

- [5] T. Mei, Y. Rui, S. Li and Q. Tia. (2012). "Multimedia Search Reranking: A Literature Survey", ACM Computing Surveys.
- [6] K. Istvan, A. Guth and R. Klamma. (2013). "Shared editing on the web: A classification of developer support libraries", Collaborative Computing: Networking Applications and Worksharing (Collaborate.com) 9th Int. Conf on, pp. 468-477.
- [7] C. Sun, S. Xia, D. Sun, D. Chen, H. Shen and W. Cai.(2006). "Transparent adaptation of single-user applications for multi-user real-time collaboration" in ACM Transactions on Computer-Human Interaction (TOCHI), ACM, vol. 13, no. 4, pp. 531-582.
- [8] Dr. Abhay Kasetwar, Ritik Gajbhiye, Gopal Papewar, Rohan Nikhare, Priya Warade(2022)."Development of Chat Application".
- [9] Nita Thakare, Nitin Deshmukh, Anshul Vairagade, AyushNagarare, Himanshu Kamane, Rajat Mohod (2022)."Real Time Chatting Web Application".
- [10] "simple peer"[Online].  
Available: <https://www.npmjs.com/package/simple-peer>.
- [11] "WebRTC based peer to peer voice,video calling and messaging web app build with MERN stack," [Online].  
Available: <https://github.com/saalikmubeen/talkhouse>.
- [12] "What is Material UI in React?" ,[Online]. Available: <https://www.educative.io/answers/what-is-material-ui-in-react>
- [13] "How To Manage State in React with Redux" Online].
- [14] Zhengyou Wang, Yan Sun (2018). "How to Design the Registration and Login Function of APP" Journal of Software Engineering and Applications
- [15] In Jhalak Mittal, Arushi Garg, Shivani Sharma, 'OnlineChat Application', Jhalak Mittal, International Journal of Research in Engineering, IT and Social knowledges, ISSN 2250-0588, Impact Factor 6.565, Volume 10 Issue 04, April 2020, go-between 10-16
- [16] In R. Gayathri, C. Kalieswari, 'Multi-User Chatting Application', International Journal of Engineering and Advanced Technology (IJEAT) ISSN 2249 – 8958, Volume- 9 Issue- 5, June 2020