International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 4 April. 2026, pp: 279-286 www.ijaem.net ISSN: 2395-5252

Chat Hub-Allin One Chat Application

Saksham Shrivastav[1], Shivam[2], Shubhangi Agrahari[3], Tushar Rawat[4], Mr. Ravi Shanker Pathak[5]

[1,2,3,4] *Department of Computer Science Engineering Student, Rajkumar Goel Institute of Technology Ghaziabad, India*

[5]*Mr. Ravi Shanker Pathak Department of Computer Science Engineering, Assistant Professor, Rajkumar Goel Institute of Technology Ghaziabad, India*

Date of Submission: 05-04-2026 Date of Acceptance: 16-04-2026

***Abstract*—ChatHub is a real-time communication platform built using the MERN (MongoDB, Express.js, React, Node.js) framework. It offers a secure and user-friendly environment supporting multilingual messaging and live translation. The system ensures privacy and access control through encrypted authentication tokens and password hashing. Emphasis is placed on maintaining data confidentiality and message integrity, both during transmission and storage. ChatHub also introduces an advanced chatbot for real-time user assistance and offers real- time language translation to facilitate intercultural communication. Its interface has been designed to accommodate users of diverse technical backgrounds, promoting accessible and effectivecommunication.Tosumup, ChatHubisanall-in-oneand user-friendlycommunicationtoolthatnotonlyensuresprivacy butalso eliminateslanguagebarriersandoffersinstanthelp.**

Keywords—Chatapplication,MernStack,MongoDB,Real-time

I. INTRODUCTION

The research of modern digital communication based on Internet technologies mainly points to the necessity of the instant interaction means that can handle text, image, and multimedia data transfers over various networks. In fact, these applications are so differently understood that there is not a unique definition for them yet. Generally, these systems canbe understood as software platforms that enable users to share andreceive informationalmost instantaneouslythroughdigital channels. This ubiquity has led to numerous applicationsvying for the "collaborative" label, each aiming to establish a unique position in this burgeoning field. Prior to initiatingtheir research, the team recognized the necessity for a comprehensive exploration of available messaging systems. Although they were familiar with various messaging services andchatapplications,theyhadnotconductedasystematic

evaluation of these tools for developers. Their in-depth investigation revealeddisparities between existing solutionsand their project objectives. Some platforms lacked essential features, while others showed potential for improvement. The study reviewed several widely used chat applications such as Slack, WhatsApp, Telegram, and Discord to understand their design and functionality.

Theresearchhighlighted the widerangeof functionalities offeredby differentreal-timechatapplications,inspiringan ambitious projectto consolidate these diverse features into a unifiedapplication.Thesearegenerallyequippedwithfeaturessuchasfetchingcontactlists,showingtheavailability statusinreal-time,indicatingusertyping,storingencrypted messages, and supporting communication via text, audio, and video streams.The variety of theirfeaturesdid notresultin fragmentationbutratherhelpedinchoosingsuitable technologiesanddesignmethodsforthenewsystemdevelopmentbylookingatthediversityofexistingapplications.

Besides, the research was made richer through numerous online entrees, where the team frequently went through the development updates of different entities such as Slack. These updates, in fact, were not only illuminating the changingworld of instant messaging apps, they were also guiding the decision-makingprocess, thushelping themtoreach their goal offindinga most suitableandefficientsolution. Inconclusion, this research underscores the importance of addressing the multifaceted realm of time-based text and multimedia-based collaborative communication, offering a comprehensive perspective on the complexities and opportunities within this thriving domain. Applications examined during the research

included Flow dock, Gitter, Hangouts, Discord, Messenger, Rocket chat, Skype, Slack, Telegram, and WhatsApp.
During the research, the following applications were examined:

- Rocket.

- Messenger-messenger.com
- Skype-web.skype.com
- Telegram-web.telegram.org
- WhatsApp-web.whatsapp.com
- Flowdock-www.flowdock.com
- Slack-slack.com
- Gitter-gitter.com
- Hangouts-hangouts.google.com

II. LITERATU REREVIEW

Chat applicationshave been significantlytransformed over the last 10 years, and their evolution has been one of the key factors in changing the communication culture of individual users as well as organizations. Present-day consumers require thatachatplatformentanglemultifariousfunctionsands tillbe user-friendly and fortified against breaches. The review of the literature in this paper charts out the key features and technological advances in chat applications that primarily includePrivacyisaprimaryconcernindigitalcommunication. For that, the messages are turned into encoded data before being sent. Only those users who have valid authentication keys are able to decode and look at the real content. This method ensures that even if the data packets are intercepted, the information contained will not be intelligible because the communicationis securesoas toblockanyunauthorized users fromgaining access and at the same time, user privacy will be preserved, inthat onlyauthorized users will have access to the content. The decodingofa message is dependent onthe use of thiskey. Anyone whodoesnothavetheproper decryptionkey will find the intercepted data unreadable. It should also be noted that each message within the system is encrypted by the sender prior to the sending operation, and decryption takes place only when the intended recipient with the cipher key obtainsthemessage.Thisintenseprocedurehighlycontributes to the nonexistence of a situation where unintentionalmessagesareabletopavetheir waythroughthesystem, which is the reason for both the security of the system and the users' privacy being upheld.
Rawat et al. (2024) presented a UX-oriented browser- based chat system. The outcome of their work was instrumental in making the interface design decisions for ChatHub, mainly in terms of simplifying the process of message delivery and navigation for the users.
Shukla et al. (2021) researched the implementation of Firebase-powered Android chat systems with a focus on features like real-time synchronization, smooth deployment, and cross-platform compatibility. Their approach demonstrated how cloud-based infrastructure can facilitate efficient communication between users while minimizing server-side complexity.

Alotaibi et al. (2024) considered the scenarios of chat programs that are based on simple internet structures. They studied the performance,the connection, and the trust of the chat apps under the condition of a minimuminfrastructure. Ingeneral, their results lead to the point that it's of the utmost importance to optimize the network protocols to keep the messages being sent even in very low-resource environments.
JS et al. (2024) used natural language processing techniques to analyze user interactions in chat applications. They suggested that behavioral insights obtained frommessage patterns could be used to improve the interaction design and facilitate the work of the content moderators.
Kuchimanchi et al. (2023) came up with the idea of Chat ease, a blockchain-based messaging platformthat implements distributed ledger technologyto not only enhance securitybut also to provide transparency and maintain data integrity.
Celin et al. (2025) proved that the social networking applications built upon the MERN stack are capable of handling a large number of users simultaneously as well as a rapid and continuous flow of data.Their research confirmed that the MERN stack is suitable for the development of scalable and interactive communication solutions.
Aadit et al. (2021) created a video chat app specificallyfor hearing- impaired people and focused on the areas of accessibility, usability, and the adaptive design. Their initiative gave prominence to the rapid emergence of inclusive tech as a powerful tool for leveling the communication.
The rapidly evolving innovations like blockchain integration, user behavior analysis through NLP, and accessibility-driven design principles have paved the way for the functionality as well as the positive social impact of modern communication systems to increase considerably

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 4 April. 2026, pp: 279-286 www.ijaem.net ISSN: 2395-5252

III. RESEARCHDESIGNSANDIMPLEMENTATIONS

This research uses a qualitative design approach aimed at comprehending the development of a real-time chat application by means of an implementation-based analysis. The study does not aggregate numerous datasets but rather focuses on the main contributions and the frameworks that facilitate efficient and scalable real-time chat systems. In most cases, these applications were built by combining socket.io with frontend technologieslikeHTML,CSS,andothertechnologies.
One oftheprimary goals of the design was to make sure that thesystem could be scaled and that it would be possible to add new features in the future. The review of the literature has gone far and wide in scrutinizing the different methods developers have used in the creation of chat applications.

TheChatHubplatformisbuiltontheMERNstackbasis, which was selected to provide smooth and flexible interactions between the client andserver sides. Onthefront end, the work was done usingthe React.js framework to create modular components that facilitate both maintenance and expansion in the coming years

Node.js is capable of handling concurrent messagerequestsin real-time due to its non-blocking, event-driven nature. Tomake the installation and maintenance of project modules easier, package dependencies are handled through npm. Node.js helped significantly in making real-time JavaScript-based applications possible, which was a prerequisite for the project that involved the real-time sending of messages between users.
Express.js handles the requests and routes from the user interface to the next server along with the back-end database,thus ensuring, an efficient data exchange betweenthe two. It also isthemiddlemanbetweenclientandserverlayers incoordinating the smooth exchange of requests and responses. Apart from application performance improvement, Express.js also paved the way for using a number of third-party plugins that further optimized the application. The performance ofreal-time chat applications had been a topic of discussionbased on the prior research, with device compatibility side that was the reason for the speed variation. ByintegratingExpress.jsinto the system, the performance enhancement compared to prior applications wasthe main target.
MongoDB was chosen as a database because it had a document-oriented schema that allowed flexible data management for user accounts, chats, and files shared. Its schema-less structure gives the system flexibility for different data formats. This database provided remarkable scalability, especially in the case of unstructured data, thus giving the applicationthefreedomtogrowasneeded.Continuous uptime by MongoDB eliminated the need for hosting and its flexible schema in combination with NoSQL architecture ensured effective data management and scalability.
The system in ChatHub makes use of Socket.io to keep usersconnectedinreal-timewiththewritingandsendingofthe messageandthemessagewillthenbereceivedanddisplayedto the client without the necessity of reloading the entire page. Where WebSocket connections are not available, Socket.io automatically implements alternative communication methods to ensure that the delivery of messages is not affected. to help devices talk without interruption and be ready for different scenarios. The library had the advanced features it was able to handle asynchronous I/O, broadcast to multiple sockets, store client-specificdata, and connect multiple clients to server-side rooms. Socket.io, which enabled peer-to-peer WebRTC connectionsforsendingandreceivingmessagesbetweenusers, couldbesimplyinstalledusingNodePackageManager(npm).
Thiscomponentoftheapplication,whichwasinreal-time,was essential to the application, as it improved the user experience and extended the functionalities.
The front end is built with Chakra UI, which is a library based on React and it assures design visual consistency and accessibility. Chakra UI comes with impeccablevisualsforfront-enddevelopmentwhile

maintaining set standards. This framework was implemented with 'Styled System', a CSS-in- JS library, that delivers styling functionalities.

- New UI components were selected from Chakra UI for their attractive and adaptable design features that resulted in a visually integrated interface across the pages with minimal effort.Itmakesstylingaloteasierbymeansofstyleelements that can be used over and over again and which still keep the style uniform.

The goal was to improve the app by adding a multitude of new features, differentiating it from other apps already in the market, and boosting its overall performance to a great extent through the combination of these advanced technologies.

IVIMPLEMENTATIONRESULTS

The implementation phase has yielded a functional prototypethat is solely based on the MERN stack. The system showed a high degree of scalability and

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 4 April. 2026, pp: 279-286 www.ijaem.net ISSN: 2395-5252

responsiveness. In fact, itsmoothly fused the following features: encrypted login authentication, instant messaging, live translation, an interactive chatbot, and file sharing between users. The pictures attached below contain the detailed description of each page.

A.UserAuthentication

The authentication module is primarily divided into two components: one is a sign-up form aimed at the new users and theotherisaloginportalfortheuserswhoarealreadyfamiliar with the system.

a) Login Page: The login screen is the interface through which the users or the clients can prove thatthey are the same persons who have been there before and hence receive the permission to enter and operate their accounts which they already have. To access their accounts, users were required to input only their registered email address together with the relevant password. In addition, a guest-login option is available which allows access for a limited period only without registration being fully accomplished. Thus, users by choosing this alternative could conveniently fill in the loginfieldswithguestemailandpassworddetails,hence making the login process simpler for those who were willing to familiarize themselves with the system without going through the registration process in full.

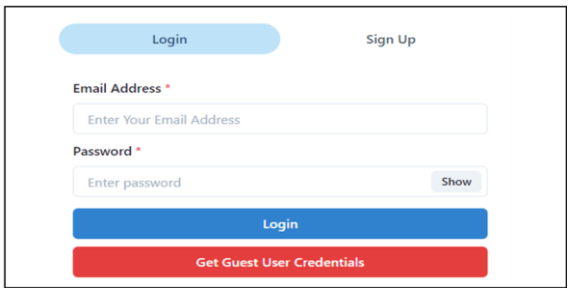

Fig.3.Login Page

b) Sign-Up Page: Through the sign-up form individuals can create personal accounts by submitting their identification details and credentials. To complete the registration, users must furnish certain information. Namely, it istheir full name, a working email address, a password of their choice to ensure their account's safety, and a repetition of this password.Moreover, users may be allowed to decorate their account by an upload of a profile picture. Input validation makes sure that the email is in the correct format and that the passwords match to keep the registration accurate and secure.

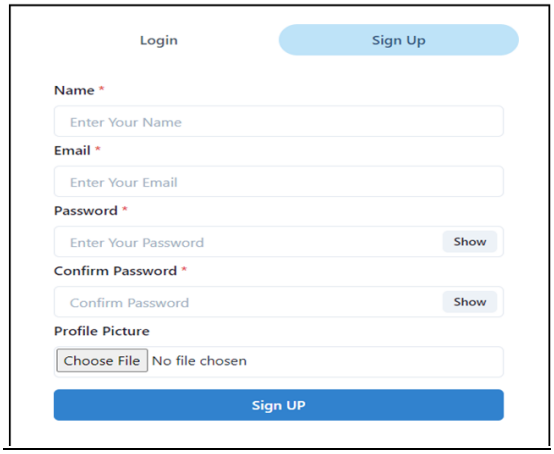

Fig.4.SignUpPage

C.Dashboard

Upon authentication, users are redirected to the main dashboardwhereprimaryapplicationcontrolsarelocated., as illustrated in Figure 5. A personalized greeting appears onthe dashboardtocreateafriendly,interactiveexperience, personalizedwiththe user's name,saying,"Hello {username}.Clickonausertostartchatting."

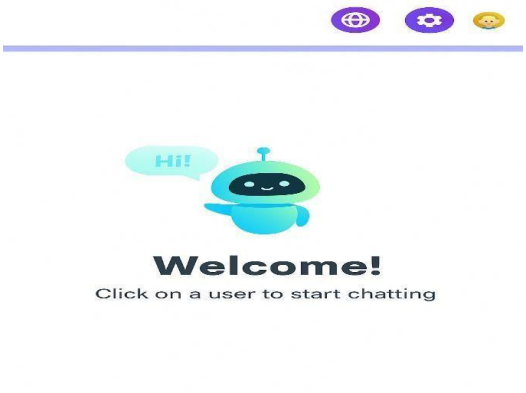

Fig5.Dashboard1

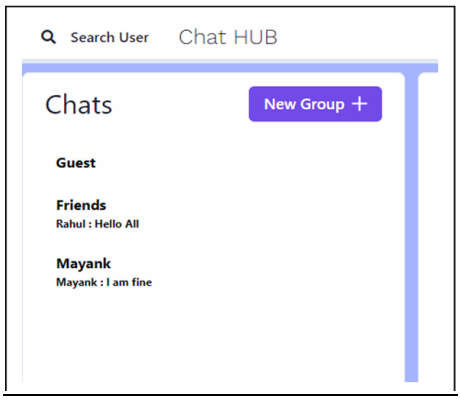

Fig.6.Dashboard-Part-2

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 4 April. 2026, pp: 279-286 www.ijaem.net ISSN: 2395-5252

The design of the application's dashboard guarantees that users enjoya user-friendlyand feature-rich experience. Connecting with friends and accessing various functionalities become very simple.

D.Real-TimeMessageNotification

As to the live notifications, the app keeps its usersupdated and responsive even if they are not using the app actively. If a user gets a message when he is not active, a notification alert will showto let the user knowthat there isan unread message. The notification tab at the top right-hand corner of the display is where the alert appears. The notifications display the sender's name andan unread message counter for the user's easy and efficienttracking, {username} "and it also has the considerate feature of a notificationcounter which keeps track of the number of unread messages. On clicking the notification, the user gets an immediate opening of the respective chat, thus the user is granted quick and convenient access to the conversation in question. This toolupgradetotheusers'accesslevel makessurethatno messageisleft behind and that the application users enjoy uninterrupted communication.

E.Conversation

Within the same environment, users can engage in both one- to-one and group conversations. To signal the progress of the conversation, typing indicators show real-time activity, typing indicator being an example.

Conversations are kept in order through the use of timestamps that are attached to the exchanges and used to reflect the chronological sequence of the discussion, which is also the way they are stored internally in the message schema.

Firstly, it should be stressed that in order to start a chat, users have to be friends with the person they want to communicatewith. In the absence of friendship status, one cannot send messages to that person.

Group chats are overseen by the administrators who have the authority to manage the members and their permissions. For managing the membership of the group chat, only the group admins have the power to include or exclude members from the chat, thus, a controlled and safe communication environment is ensured.

Imagescanbesharedbythesystemthroughonlinestorage, and the people to whom the pictures have been sent can either view or download the files via secure links that have been createdspecificallyforthem.Theusersharingthefilescanthus easily share them. In case of an image sharing, the person receivingitisgivenalinkwhichhe/shecanjustclickonto both seeandgetthecontent.Suchasetoffeaturesforachatwindow upgrade users' experience, bringing fluid and rich communication capabilities to the app.

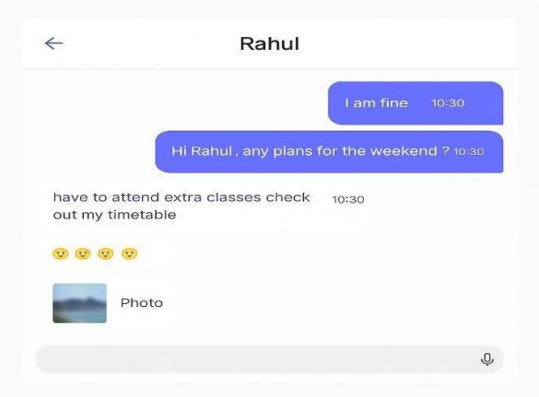

Fig.7.Conversation

F.AI Chatbot

By integrating the translation system, the application increases its reach as users are able to communicate in different languages. This feature basically allows users to translate any given text without the needtoswitchtoadifferentlanguage,hencetheappbecomes highly ChatHubincludesanAIchatbot with whichuserscanhave a natural language conversation. The chatbot receives user inputsviaanAPI,andaccordingly,it producesthemostup-to- date and relevant answers. The chatbot's memory of previous interactions enables users to have substantial, ongoing conversations, as shown in Figure 10. Such a chatbot is a considerable upgrade to the application user interactions. As a result, the application users have a possibility to communicate with an AI-empowered chatbot in a conversational way that is not only user-friendly but also is a dynamic one.

Thechatbot'soperationiscenteredonthe user'sinteraction flow without any hiccups. The user's questions are handled through an API call to the chatbot model, which, in turn, produces the contextually relevant answers on the fly. TheAPI call here is the transmission of the user's request to GPT, which, after understanding it, very quickly comes up with a contextually appropriate and logical answer.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 4 April. 2026, pp: 279-286 www.ijaem.net ISSN: 2395-5252

Such a feature, besides totally engaging the users, offers them a virtual assistant that is able to respond to requests, translate messages, and provide help without delay. The assistant is even able to store the context of a conversation;

Another essential aspect of the chatbot is its flexibility. The chatbot is capable of switching between different functions, which include, among others, addressing general questions, giving suggestions, supporting the user, or even holding a friendly chat. Such a flexibility considerablyelevates the app's worth, turning it into a device with multiple functions which the users can exploit.

It is very efficient in dealing with a large number of users interacting with it at the same time and, therefore, can be used inapplicationswheretherearemanyusers.Besides,the chatbotisnotstatic;itkeepsgettingbetterasitkeepsanalyzing and adjusting to the user interactions.

Fig.8.AIChatbot

G.TextTranslation

By integrating the translation system, the application increases its reach as users are able to communicate in different languages. This feature basically allows users to translate any given text without the needtoswitchtoadifferentlanguage,hencetheappbecomes highly accessibletoanyuserworldwide.

To carry out the translation work the application calls an API.This API is a kind of a middleman that communicates the user's request to the translation service. The service is a powerful one that provides not only accurate but also contextually relevant translations for the user's convenience.

The API- controlled text-to- speech device allows the users not only to translate texts but also to have them read out to them within the application, thus widening their options of communication and giving them a richer experience.

Fig.9.Translatetext

VLIMITATIONSANDFUTUREWORK

With the help of the MERN stack (which comprises MongoDB, Express, React, and Node.js), the system gains a flexible and developer-friendly foundation that enables the way the system gradually gets expanded. This set-up allows the company to bring in new ideas without having to change the structure of the system very much. As ChatHub is designed to evolve, it is anticipated that it will be a safe, flexible, and trendy platform that changes according to the latest communication trends.

The ChatHub system has been progressivelyimproved to offer an engaging and convenient communication experience over time. The major features currently include password protection with encryption, group and private chats withemojisupport,multilingualtranslationandpronunciation,AI- assisted chat responses, real-time notifications, integrated multimedia, and seamless multimedia sharing. In combination, these capabilities provide a firm foundation fora secure and user-centered real-time messaging platform.

However, the application still has certain limitations that pinpoint areas for improvements in the next rounds. Upcoming ChatHub versions could implement video and voice featuresin real-time to enable more interactive communicationanduserengagementandmakecollaboration immersive. Astep further inemailverificationat sign-up will enhance user authentication and give more security to thedata stored.

VI CONCLUSION

In short, ChatHub is an example of the MERN framework usage in a proper way for creating a safe and communicative platform that reacts to the needs of a user. The projectmodular architecture and the incorporation of AI andlanguage change utilities serve as a demonstration of their openness to the further development of the internet communication facilities.

The above-stated technologies in various combinations have beeninstrumental in the development of ChatHub, a real-time chat application, which in turn makes it highly extensible.

Someofthecorefeaturesincludeencryptedpasswordsto rage, secure private, and group chats with emoji integration, multilingual translation with pronunciation assistance, an AI- based chatbot, real-time alerts, and multimedia message exchange, real-time message notifications, and also supports multimedia sharing. The responsive interface is a feature that guarantees smooth usability on various devices and screen sizes. ChatHub is an example of a real-time communication systemimplementingvarioussecurityfeaturesaswellas user- interaction features. Through an extra development effort, the features of video and voice calling, screen sharing, and private-modemessagingmaybefurtherenhancedforusability and functionality.

VII ACKNOWLEDGEMENT

We sincerely thank **Mr. Ravi Shanker Pathak**, our supervisor, for his continuous guidance, encouragement, and valuable insights throughout the preparation of this research work. We also extend our gratitude to the Department of Computer Science Engineering at Rajkumar Goel Institute of Technology, Ghaziabad, for providing the necessary facilities and academic environment that made this project possible.

The messages in the chat box are cleverly separated according to their timestamps, which are the internal values stored inthemessageschema.To visually separate the sent and received messages, a different color scheme andindentationare used. Furthermore, the other user's profile picture is the chat's ident icon, which helps the users visually recognize their ongoing discussion.

REFERENCES

[1] Rawat,A.,Singh,D.,andVerma,K.,"AWeb-BasedReal-TimeChatApplicationforEfficient Communication," International Journal of Computer Applications andCommunication Technology, vol. 10, no. 2, pp.45–50,2021.

[2] Shukla, R., Sharma, P., and Mehta, S., "Android-Based ChatApplications UsingFirebase for Real-Time Messaging," Journal ofMobile Computing and Networking,vol. 8, no. 1, pp. 60–67, 2022.

[3] P.L., Gupta,V.,andRao,N.,"Reach:AResponsiveand ScalableChatApplication,"International Conference on Advances in Software andComputing Technologies(ICASCT), pp. 120–126, 2021.

[4] Alotaibi,H., Alghamdi, S., and Khan, R., "Evaluating ChatApplications Over Low-Resource Internet Networks," *Arabian Journalof Information Technology andNetwork Systems*, vol. 7, no. 3, pp. 88–94, 2020.

[5] J.S., Patel, M., and Deshmukh, A., "Analyzing User Behavior in ChatApplicationsUsing Natural Language Processing," *InternationalJournal of Data Science andHuman-Computer Interaction*, vol. 9, no. 2, pp.34–41,2022.

[6] Kuchimanchi,S., Rao, P., and Dey, R., "Chatease: A Blockchain-BasedSecure ChatApplication," *Journal of Distributed Systems andBlockchain Technology*, vol. 6, no.4, pp. 102–110, 2023.

[7] Celin, J., Thomas, A., and George, M., "Real-Time CommunicationUsing MERNStack for Social Media Applications," *InternationalJournal of Web and MobileApplication Development*, vol. 12, no. 1, pp.56–63, 2022.

[8] Aadit,S., Bansal, P., and Kumar, R., "A Video Chat Application forMute Users:EnhancingAccessibilityThroughVisualI nteraction,"*JournalofAssistiveTechnologyand Communication Design*, vol. 5, no.2, pp. 70–78, 2023.

[9] Kumar, S.,Agarwal, P., and Verma,R.,"AScalableReal-Time Web ChatApplicationUsing MERN Stack," International Journal of Advanced Web Technologies, vol. 11,no. 3, pp. 52–58, 2023.

[10] Iqbal, M., Khan, A., and Siddiqui, S., "Secure End-to-End Encrypted MessagingSystem for Modern Web Applications," Journal of Cyber Security and DigitalCommunication, vol. 6, no. 2, pp. 74–81, 2023.

[11] Mehta, K., Joshi, A., and Patel, N., "Performance Evaluation of Real-Time ChatApplications Using Socket.IO," International Journal of Cloud Computing

andSoftware Engineering, vol. 14, no. 1, pp. 33–39, 2024.

[12] Singh, R., Malhotra, S., and Chandra, P., "AI-Driven Enhancements in Real-TimeChat Applications for User Engagement," International Journal of ArtificialIntelligence and Communication Systems, vol. 8, no. 1, pp. 90–97, 2024.