ChatterBox- A Real Time Chat Application

Nidhi Zala
ID: 1160480
Department of Computer Science
Lakehead University
Thunder Bay, Ontario
nzala@lakeheadu.ca

Vinita Agrawal
ID :1159203
Department of Computer Science
Lakehead University
Thunder Bay, Ontario
vagrawal@lakeheadu.ca

Dr Jinan Fiaidhi
Department of Computer Science
Lakehead University
Thunder Bay, Ontario
jfiaidhi@lakeheadu.ca

***Abstract*—A chat operation is a location or software application that allows Internet users to connect directly with one another. With the help of this online application, people may interact effectively even when they are far apart. To be utilised by many people, this web based application is supposed to be real-time and multiplatform. The process of creating this web application starts with the gathering of relevant data that will be used in the web application. It makes it possible for users to exchange resources, make audio and video calls, screen-sharing etc. Chatterbox's tool enables users to chat or contact with others online. The backend server is developed with the help of Node.js also with an express framework and a database called Mongo Db. We developed this web application using the MERN stack development which states MongoDb as the backend Express.js, React.js and Node.js for the development making the application different from others as it is easily extensible and manageable. These are the emerging technologies that are widely employed in many different businesses, including Instagram and Facebook and much more. Particularly the application is more secure and comes with the option of screenshare which other applications mostly don't have making it handy and easily reliable as well. Comparatively speaking, it is more secure and trustworthy than any other conventional systems. Also, with further system demands for running higher ends of the project, multiple features can be added to the application due to the extensible code and the application can be extended.**

***Index Terms*—chat application, Mern Stack, MongoDb, real-time**

I. Introduction

Internet-based simultaneous and cooperation which is time based on text and multimedia has become major area of research. Applications for the same are currently not defined in a good manner. The term "collaborative application" is currently used to refer to any software programme that allows users to connect to one another in order to connect information in writing or through video in real-time or very close to real-time. As a result, several programmes nowadays make the claim to be collaborative. Internet users who are online or those who are constantly using the internet can talk directly with one another using a feature or programme called a chat room. Users of chat apps can communicate even when they are far apart. To be accessed by many people, this feature must be problem solving time and platform independent.

The chatroom is currently being created by a lot of programmers. Numerous chat applications have their own pros and cons.We examined the available messaging systems before beginning the development of the project. We already knew that there were a number of messaging services and chat programmes. We had never, however, examined their tools in-depth to see whether they were enough for developers. We quickly became aware that none of the locations were moving in our direction. Some of them lacked characteristics that we thought were essential, while others offered room for improvement. We looked into various platforms like Gitter, Slack, Whatsapp, Telegram, Messenger, Discord, Skype, Flowdock, etc. There are billions of users of the listed applications worldwide. These businesses rank among the strongest in the industry. They make more profit each year and hire a vast group of users to work on new features for their releases to keep up with other companies. These applications use a variety of features and procedures to protect the privacy of their users' data. Today, the most common crime is data theft, which is committed by the majority of people. These days, a lot of instances involving the loss of personal data are being filed. Hence, entities need to protect data security against data breaches. The chatting system should also provide simultaneous operations like send and receive. Both transmitting and receiving are possible in this application.

Based on the background research, the problem with real time chat applications is that different application have different features. We are trying to bring all features like sending invitation, online indicator, notify on typing, storage of messages in database, chatting, audio and video call, screen sharing in one application.Contrary to popular opinion, having a few applications available is a good thing. Based on their experience, we were able to gather insights for what to develop and how, and choose the technologies and techniques to implement. Checking their blogs was usually all that was required. Companies like Slack usually publish updates on their development.Several times, we had to look on the internet to learn about our alternatives and choose the one we felt was necessary.

During the research, we looked into the following applications:
- Flowdock - www.flowdock.com
- Gitter - gitter.com

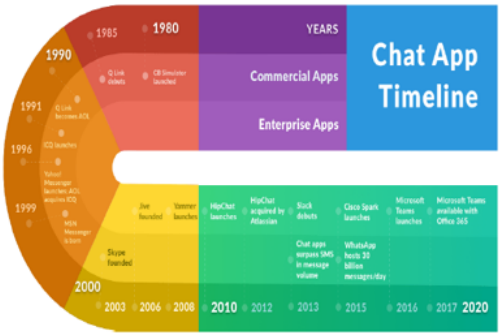

Figure 1. Chat App Timeline

- Hangouts - hangouts.google.com
- Discord - discord.com
- Messenger - messenger.com
- Rocket.chat - rocket.chat
- Skype - web.skype.com
- Slack - slack.com
- Telegram - web.telegram.org
- Whatsapp - web.whatsapp.com

The ones that concentrated most on were Gitter and Slack. Although simple functions like code sharing were also accessible in services like Rocket.chat, it was obvious that these platforms were not designed with professionals in mind, which frequently left them in absence of the necessary tools. They do, in our opinion, require a well defined environment that is created only for them, which makes code sharing fun, and offers the option of being connected with their source code repository.

It seemed obvious at this time that Slack and Gitter should be the focus of our research. The other platforms were still useful for trying to gather a few overall ideas, but they had nothing to do with our study. Both Slack and Gitter are well-known platforms with a developer and productivity focus that have demonstrated success over a significant period of time. Aside from being open-source, which none of them was, there are a few additional aspects that will set our programme apart from these other two.

Now Globally, developers are working to improve the the experience of user of their products. Engineers from all around the world are working to enhance the application's experience and development workflow in order to design apps that can be delivered on time and for the release of application. Stacks may be used to create web apps quickly. Web development stacks are essentially a software developer's answer to current demands. Accept and use current frameworks, such as JavaScript, to simplify their job. MEAN and MERN are well-known stacks that sprang from JavaScript, while there are many others [8].

Both stacks are constructed from open source parts and offer a complete foundation for creating intricate web apps that let users interact with the website. JavaScript unites the two, which is another significant benefit of utilising any stack. By encoding in JavaScript, one may code without worrying about misunderstanding or grammatical mistakes. A further advantage of using MERN to create web applications is the increased flexibility it offers. The four components can be considered that makes the MERN stack, known as MongoDB, Express.js, React, and Node.js.

II. Literature Review

The structure and design of a real-time collaboration application were discussed in this study, enabling many other users to observe and create synchronised versions of files and other material through a web browser. Additionally, it can combine files with the document that is being edited. It was demonstrated how users might work together on a rich-text document with an embedded synchronised video player Smart Object. The suggested solution differs from current collaboration choices due to its emphasis on easily accessible chat features and flexible architecture that enables professionals to swiftly add new experiences via the Stream Container API.

Encrypting the communications that a user sends or recieves on the server is one method of maintaining the privacy in communication. Even though messages are kept on hold or received by sources who are not given access to the information and do not have the permission to do so, encryption makes it possible that they remain in an unreadable format that does not make sense. Before they take on the actual message and actual meaning of the communication in question, these encrypted messages need to be decrypted. Only systems with permission to use a secret key known as the cypher key are able to perform this decryption operation. Users with authorization only are able to access this key. The decoding of messages requires this use of the key.One even if they gain access to the encrypted messages, unwanted sources cannot decrypt the original message without this secret key. Every message that is exchanged between users within in the system is encrypted [3]. They are encrypted by the sender before being transmitted, and are only decrypted after they get to the appropriate recipient who has the cypher key. This ensures that non intended messages do not pass through the system, preserving both the system's security and the privacy.

R. Gayathri, C. Kalieswari published their work in 2020 in the International Journal of Engineering and Advanced Technology (IJEAT) [15]. The chat application offers a better and more adaptable programme for debate, according to this research article.Developed with cutting-edge technologies to offer a dependable system. The system's primary benefits include group chat, improved security, real-world collaboration, and instant messaging. For the majority of businesses looking to have private applications, this app may

locate the greatest market demand. Based on community requests, more programme features like conference calls and video chat will also be implemented. Sharing locations, etc., according to necessity.

In 2016, the International Conference on Engineering and Technology published "Designing and implementing a real-time web-based chat server" by Diotra Henriyan, Devie Pratama Subiyanti, and Rizki Fauzian (ICSET) [1]. According to this study report, a chat application should have a live forum and be multi-site to accommodate a large user base. The programming language is used to create the MongoDB website and the Node.js server with a clear foundation.

In "On-line Chat Application [14]"- the writers have provided a note of permission application that enables users to access their accounts from any mobile phone, anywhere, at any time. Before communicating with another user, a user must first issue a conversation request. Users only send communications when they have approval from them; otherwise, they do not. Otp is not required, thus the user must connect to their application using their email address and password.This article has shown that internet communication enables users to communicate with others in a rapid and efficient way. This essay demonstrates the value of conversational applications in modern life and their influence on the technology landscape.

In contrast to the above discussed paper, an article [15] presents a discussion system for private networks or associations. This method ensures that any shared private information and communications across the network remain secure. Additionally, it securely retains data used in the back side. It provides a communication system that is two way, adding more elements from other conventional systems as needed and allowing both group conversation and private discussion, allowing for simple and quick contact between individuals. This ensures limitless data transfer without size restrictions, enabling connections with others at any time and from any location. in order to transport various train forms through the system. infinite space for communication data storage.

Another networking programme [16] offers features including social networking in easy language, picture backup, image theft alerts, landmark identification, and information on any image. It has been seen to operate with the framework called react to create interfaces that a user can interact with and real-time messaging applications on web browser that do not require any additional third-party account programmes in order to establish visual communication. The application has been built utilizing express js framework with React.js and Node.js,followed by the Mongo DB database.

Maximum number of people are preferring network chatting tools for communication as the internet develops and improves. These applications make it easier to communicate across enormous distances. For an application to be utilised by people properly it must be cross platform and real time. No additional user information from a third party is required for the web-based real-time chatting application to establish visual communication.The data transmission is made easy by connecting servers with point-to-point connections, and the text communication is transmitted between servers. React framework's implementation of the virtual space notion improves performance over currently available PHP-developed applications by a factor of roughly six.

III. Research Questions

- **1) What purpose does the application suffice in the field?**

  A) Recently many applications have been developed with have features like posting images and audio and video, reach time chatting is something must but something which allows to make call, share screen, and video calls makes it more versatile and stands a good space in the industry where user can get a one stop access to these features.

- **2) How is the approach different from the previously developed products?**

  A) The approach followed for the development of the web application is MERN stack development which uses the current latest technologies like MongoDB, Express.js, React.js and Node.js for the development procedure making the project different from the pre-developed apps. Also these development styles are easily extensible of proper system requirements are met, so the application can be easily extended with new features.

- **3) How is privacy and confidentality maintained in this application as compared to other?**

  A) Nowadays the files exchanged as well as the images that are shared are sometimes not protected. Therefore, this application provides the feature of sharing the screen which helps the user to present files/images without sending them.

IV. Research Designs and Implementation

The research design is based on qualitative analysis. Rather than considering the approach for using multiple research works, we have qualitatively selected some major work done in the field for developing real-time chat applications. Most of the applications are developed by combining socket.io with the frontend work languages bet it Html, CSS and much more. The need to make the project more versatile and extensible was a requirement as most of it would need to be extended in the near future. The literature studies are

extensively reviewed to find multiple methodologies that were being used by the developers for developing the chat application.

The project design would follow a MERN stack development. Here the basic flow for the project is shown in the below image.

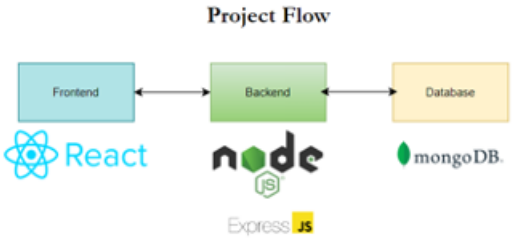

Fig. 1. Project Flow

As shown in the figure 1 the project considers React Js as the frontend language for the implementation. React has been developed by Facebook. The major reason behind using React for the development is its reusable component feature. For instance,consider an example of Instagram. The stories available on Instagram are not made or loaded again and again it's the re-usable component that renders again for each story passing by. Similarly for making the application faster, we use the methodology of using React js in the frontend so that we can reuse the components one after the other when required thus making it easily extensible as well.

Socket.io is one of the javascript based library for web applications. It is made up of a server library for Node.js and a client library that runs in the browser.The client point of view of socket.io is a library that primarily utilises the WebSocket protocol, but may also fall back on a range of other techniques as necessary, including Adobe Flash sockets, AJAX long polling, and others while maintaining the same interface. It offers a wide range of sophisticated capabilities, including handling asynchronous I/O, broadcasting to many sockets, storing data related to particular clients, and linking several connections with a server-side room.With the help of the tool node packaged modules (npm), socket.io may be conveniently installed.Peer instances for the sender and receiver would be in different browsers. Up until a peer-to-peer connection is created, a "signalling server" (often built using websockets) may be required to exchange signal data between two browsers.Therefore, for peer-to-peer WebRTC connections simplepeer library is used.

Furthermore, as the flow displays, Node Js is used for the backend. Nodejs is a javascript-based runtime built on the chrome virtual engine. Considering a scalable web server, the flow mentioned will always remain the heart of the developed application. Further, we use npm which is the node js package manager, the repository for all the packages. Node js is the major reason which helps to develop javascript-based real-time applications which is much vital for our real-time chat application. The application will take real-time messages and send them from one person to another. Thus Node js plays a pivotal role in that.

Considering Express js which is the framework for the Node js it works as the routing API. The messages will be sent to Frontend $\leftrightarrow$ API $\leftrightarrow$ Backend. Thus, Express js helps in this scenario. It provides high performance to the application which also provides some third-party plugins to the application for the faster working of the application. Looking after the previous literature, depending on the application compatibility with the device the real-time chat application was either slow or fast but with these major components used to build the application, the website would work much faster than the previous applications developed till date.

Finally, the backend is connected to the database of the application. There are a wide variety of applications used till date as the backend of a real-time chat application, like Firebase, Default Django admin database and much more. But MongoDB is selected for developing our chatterbox real-time chat application. Basically, it is a NoSQL Document oriented application. Also, it is a cross-platform application which works across multiple platforms thus making it a versatile platform for use in any of the applications. It is super scalable as it is NoSQL based, and it also works for the unstructured data Thus it will allow us to make it as extendable as we want our application to be. One of the major reasons for selecting MongoDB as the database for our application is that it will always be on. Like the other apps, we don't have to host it as a server and it is also maintained 24/7. Because of its NoSQL and Flexible schema, it is highly scalable and thus can be well managed.

For the UI "Material UI" library is used. It is open-source front-end framework for react components. It was created with Less. A CSS language extension that is compatible is called Less. The aim of Material UI is to form visuals that are at frontend while maintaining quality of digital experience. It is based on Google. With an importance on how the components gives shadows and reflect light, it creates textures that are different and clean. The following are a few major benefits of using Material UI while designing:

1) Some frontend frameworks are challenging to work with since they are not properly documented. On the other hand, Material UI includes thorough documentation that makes it simple to learn the framework.

2) The produced application or website looks visually appealing since all of the components follow the same design and colour scheme.

For state management ”Redux” library is used.A well-liked data store for JavaScript and React apps is Redux. It respects to the fundamental element that data binding should be held as a central repository and flow in a single way. Redux’s achievement is due to its straightforward design philosophy and modest implementation.Redux functions based on a few ideas. The store is a single object that has fields for each type of data selection. By deploying an action that specifies how the data should change, you may update the data. Then, using reducers, you understand actions and change the data. Reducers are functions that operate on data and produce a new state rather than modifying the old one.

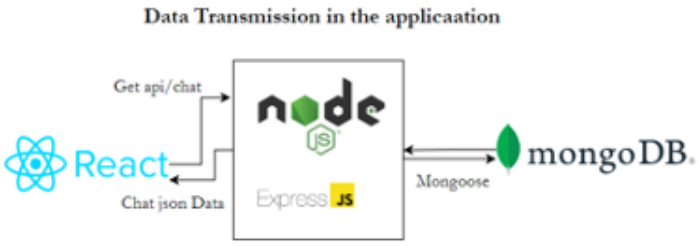

Fig. 2. flowchart

The flow chart in Figure 2 will delineate how the project would work fully with all the components involved in the development cycle.Thus with the following methodology, a successful real-time chat application is developed which is faster than the previously developed chat applications and have more features in comparison to the previous literature mentioned regarding the developed apps. Thus with these new technologies, we made an attempt to add various features to make the application stand out from the other developed applications which also works much better than them.

V. Implementation Results

Focusing on the implementation, as stated above the application works on technologies like MERN stack which is highly extensible and the best to be used. Considering the system limitation at this point a basic model for real time chatting is developed with some major features like audio call, video call, secure login and registration and so on. Each page is explained in detail below in the images.

1) **Registration and Login Page**
The user must get a unique ID for the account in order to indicate his/her independent existence in accordance with the registration and login features. The user might need to keep track of his/her activity, including his/her usage history, likes, favourite content, etc. Additionally, the user must connect with other users in order to get their concrete facts or detailed information as necessary. Figure 4 represents the view of the registration or sign up page. It asks user to create an account by entering Username, Email address and password followed by a sign up button. It also says if the user already have an account , he/she can click on the login option. The login option will redirect the user to a new page called login page as shown in figure 6. This page allows user to login with username and password. The login page also has an register here option which will redirect the user to registration page.

Registration Page:

Username: input field = text
Email: input field = email
Password: input field = password

IF (Email == blank or password == blank):
“Error message”
Else:
“Registration Successful”

Fig. 3. Register page Pseudo code

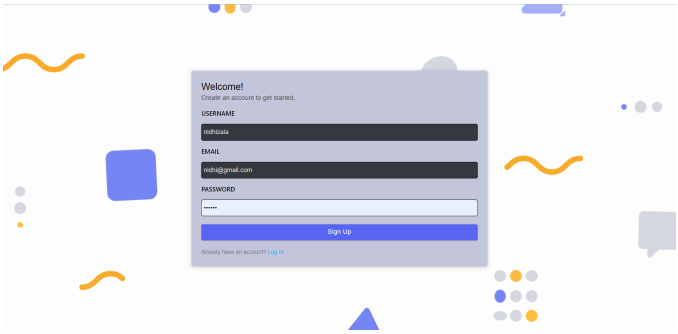

Fig. 4. Register page

2) **Dashboard**
After logging in, the user will be directed to a dashboard as shown in figure 7. The dashboard has various tabs to indicate various features of the application. A heading ”To start chatting - select a friend for conversation” is seen at the center of the display.To select a friend for chatting, there is a list of added friends on the left side of the display.It also shows a green dot to indicate that the friend is online. There is ”Add friend” button on the top left corner of the display which allows the user to invite a friend to use this web application. There is a

Login Page:

Get login_id
Get password

IF (login_id = = Entered_username and password = = Entered_ password):
Login Successful
Redirects to Dashboard Screen

ELSE:
Login Failed

End

Fig. 5. Login page Pseudo code

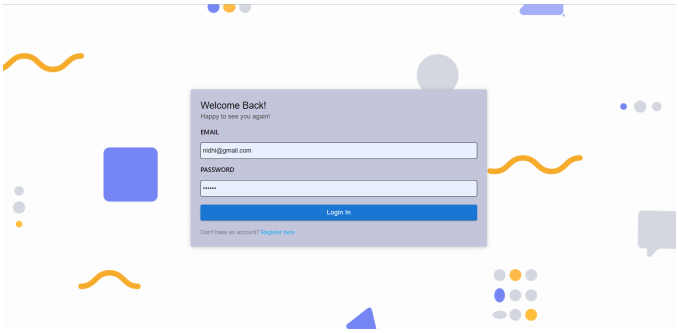

Fig. 6. Login Page

list of private messages and all invitations sent followed by the list of friends. There is a three line menu button at the top right corner of the display which indicates the username of the account and also has an option to logout.

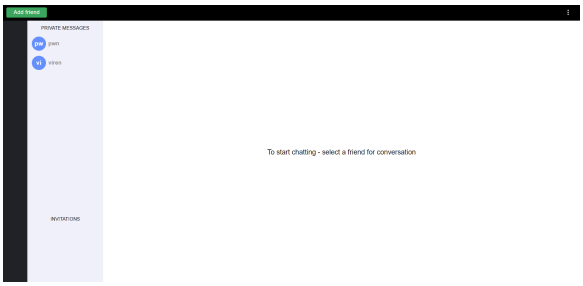

Fig. 7. Dashboard

3) **Invitation Sent,Invitation via Email and Invitation Accepted:**
With the help of Add friend button on the dashboard user can send an invitation request to anybody via valid email address. As the user clicks on the Add Friend button it will ask for the email address of the user that is to be invited as shown in figure 9. Once the email address is entered, click on Invite button the user's name will be displayed in the list of invitations on the bottom left of the dashboard display. As soon as the user accepts the invitation request the user is then added to the list of friends displayed at left side of the dashboard display as shown. As shown in figure 10 that user has sent an invitation to jay which is displayed in list of invitations at the bottom left side of the dashboard display. When jay accepted the invitation as shown in figure 12 his name is now displayed in the list of added friends at the left side of the dashboard display. This is how securely one can invite a friend and add to the chat list.Later allowing the user to chat, audio call, video call etc to the friend in the list.

FriendInvitation Module:

Enters the username of the friend

IF (Username == Entered Username)
Passes to the controller for authentication

If (Entered Username == database.Username and Status of friend == pending)

"Alert message = friend request already send"

ElseIf (Entered Username == database.Username and Status of friend == Friend)

"Alert message = Already added to your friend list "

Else
"Status changes to pending"

Request send successfully

Else:
"Please enter a valid username/Email"

Fig. 8. Invite Friend psuedo code

4) **Conversation**
The web application allows user to chat with each other in a private chat as shown in figure 10. The chat box also notifies when the other user is typing. The chat box separates messages according to date and time. The messages sent are in blue color box theme and the messages received are in white color box theme. Also, the messages also indicate time of receiving and sending.Just like a normal messaging application the user can open a chat and chat further with the person he is friend with. For having the conversation the person must be the friend else he wont be able to send any messages to the person. As shown above the invitation must be sent via email. Thus this way a secure system allows you to chat with people. Futhermore, more features like stickers or gifs could be

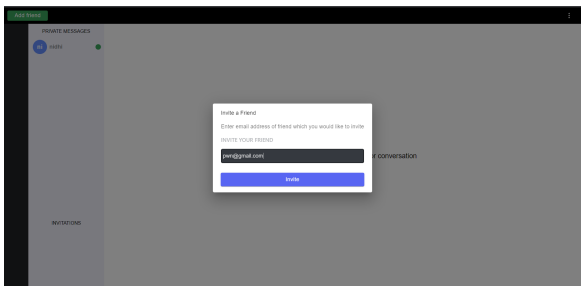

Fig. 9. Invitation via email

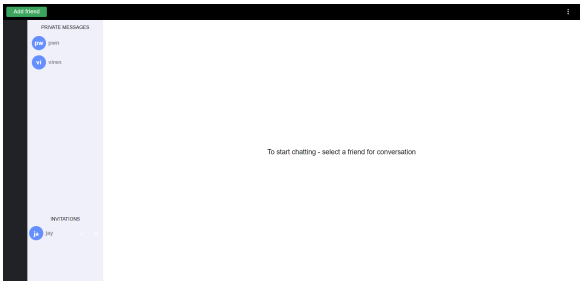

Fig. 10. Invitation sent

Friend Accepted Module:

IF (Current_user Accepts the invitation):
"Status changes to Friends"

Else

"Status changes to Rejected"

Fig. 11. Invitation accepted psuedo

added to chatterbox for making it similar to application.

5) **Audio call, video call and screen sharing**

Our web application has a new feature of screen sharing as shown in figure 10.Sharing screen's contents with another device is referred to as screen sharing or desktop sharing. This gives user total control over how much of their desktop is shown while still ensuring user's privacy. It can encompass just a window or all the objects that are on a screen. User may display friends, employees, or clients any media that is on the device to share screen without ever sending any files; this includes presentations, documents, photographs, and even films. Additionally, the receiver may watch while the user interacts with the shared device in real-time, navigating the interface and making changes, while simultaneously viewing the content on it.This web application also allow user to make audio call as well as video call.They way displayed in Figure 12,

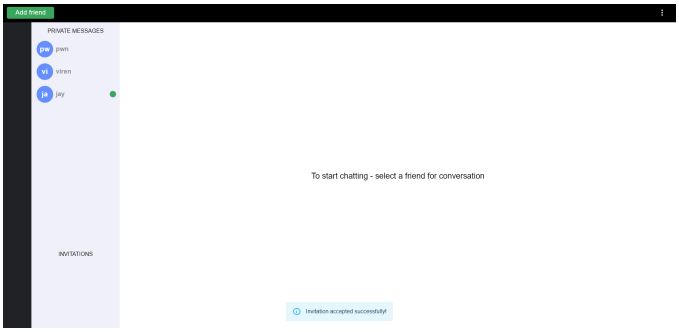

Fig. 12. Invitation accepted

```
If (Click == User_id):
    open chat section
    get_active_connection = (userId) (connected user)
    get online_users
        {fetch Socket_IO}
    Establish Sockets Connections
    Crease Socket Server = (Server) {Localhost : 3000}
    Console.log (Socket)
    socket.on("Direct message")
        Notify person is typing
        check connected sockets
        display message
```

Fig. 13. Conversation pseudo code

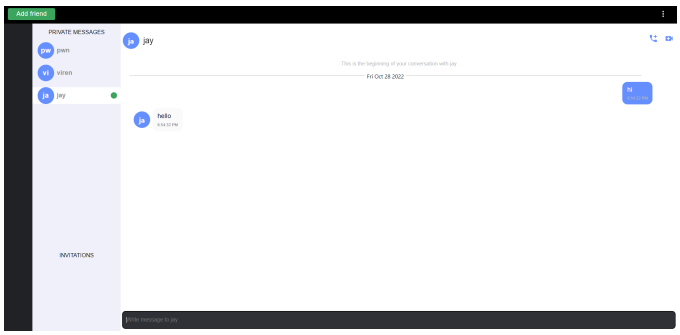

Fig. 14. Conversation

user is receiving an incoming call which gives him an option to pick up audio, video or end call.

VI. Limitations and Future Work

With the advent of time a lot of new technologies have come up delineating multiple features for a Real time chat application. the current chat application chatterbox connects some vital features for instance be it adding friends using Invites, delivering messages, audio calls, Video calls and on the top an option for screen sharing. Considering the system load and the acquired time some limitations are still visible in the application. Group chatting and the same features for the group chatting can be added for making

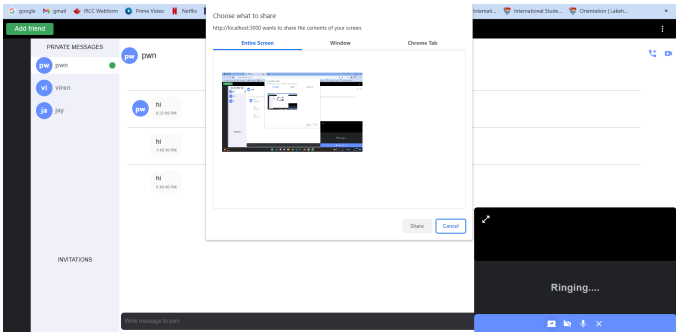

Fig. 15. Screen Share

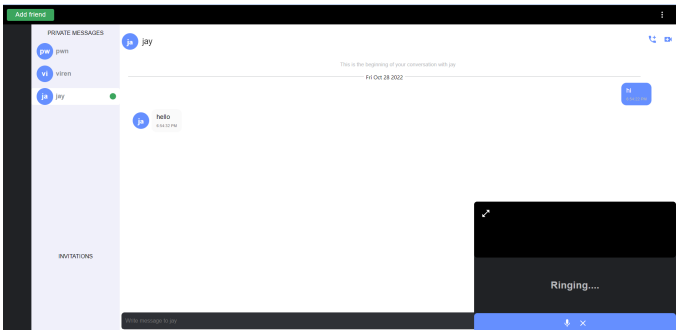

Fig. 16. Audio call

the application a complete package. Furthermore, email authentication while registering could be added for making the application more secure and some features like read more for long messages, hide messages, private chatting all makes the project a complete package which could be added on later.

Furthermore, the project works with MERN stack technology which makes it easily extensible and developer friendly so adding new features to it can be easier than other usual projects that uses firebase and normal html, css for functionalities. Thus, new features can be added and the app can be made more dynamic and secure for the user in future.

VII. Conclusion

To conclude, the real time chat application chatterbox has been developed with much recent technologies like ”MERN” stack where M-MongoDb, E-Express.js, R-React.js, N-Node.js making it more extensible. The applications has features like audio call, video call, screen sharing, private messages and some secure invitation method to add friends in the application. It also comes with an option of screen share while video call making it more handy for a user. Thus, the application is a complete package for user real time chatting with some major features to use and if the developer resources increase the application can be extended by adding group chatting models and hidden chatting features to make it more useful for a user.

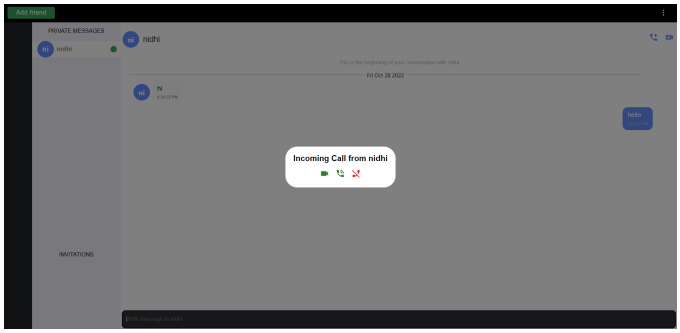

Fig. 17. Incoming Call

Acknowledgment

We would like to express our gratitude to our professor, Dr. Jinan Fiaidhi, who has always been our driving force behind completing the paperwork. Our profound gratitude to the Management, as well as the computer science department staff at Lakehead University in Thunder Bay, Ontario, Canada, for their encouragement and motivation.

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 lonescu, Cristian Gadea, Bogdon Solomon, Mircea Trifan, Dan lonescu (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 (Collaboratecom) 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 ,Ayush Nagarare , 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 messag-ing 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]. Available: https://www.digitalocean.com/community/tutorials/how-to-manage-state-in-react-with-redux

[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 ofResearch 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

[17] Akhilesh Sarjit M S, Srivishak V, Shiddarth S,Saravana Kumar P, Preethi D,' Web conversation using React Framework', International Journal of Trend in Scientific Research and Development (IJTSRD) Volume4 Issue- 3, April 2020