ISSN No:-2456-2165

PRIVY: A Mern Chat App with Video Conferencing & Screen Sharing

M Dolie Ciri (B.Tech), MDS Vishnu Vardhan (B. Tech), M Surya Teja (B. Tech)
Computer Science & Systems Engineering Andhra University,
Visakhapatnam-530003, AP, India

Dr. M. Ramjee
Adjunct Professor
Department of Information Technology & Computer Applications,
Andhra University, Visakhapatnam-530003, AP, India

Abstract:- Messaging is now part of everybody's life (1) in the world due to its ease and convenience for instant communication and simple to use. Almost all social networks in the world have their chatting facilities. However, there are many common featuresin the existing chatting applications. Here a new chat application system by the name PRIVY is proposed by the authors with new features *Video Conferencing and screen sharing facility*.

This PRIVY chat application with meetings feature is a comprehensive communication tool designed for real-time messaging and audio/video conferencing. The application offers a modern and intuitive approach to communication, providing a seamless user experience. It is built using MongoDB (2), ExpressJS (3), ReactJS (4), and NodeJS (5), making it a powerful and flexible tool.

The chat feature allows users to communicate via text messaging in real time, with the ability to create groups and add multiple users. The meeting feature enables users to host or join audio/video calls, with the option to mute/unmute audio or video and share their screen. The application's interface is clean and modern, providing a user-friendly experience.

This PRIVY chat application with meetings feature is ideal for remote teams, students, and individuals who require reliable and efficient communication tools. The application provides a versatile and flexible way to collaborate and stay connected, with its seamless user experience and intuitive design.

Keywords:- *MERN, chat application, meetings, real-time messaging, audio/video conferencing, MongoDB, ExpressJS, ReactJS, NodeJS, user experience, remote teams, collaboration, and efficiency*.

I. INTRODUCTION

The English meaning of *chat*(6) is to talk in a friendly and informal way. However, the term *chat* is also used when there is an exchange of conversation electronically between two or more people over the internet. Now, the term *chat* is also used when there is communication between people through the apps that allow maintaining this exchange. A chat application makes it easy to communicate with people anywhere in the world by sending and receiving messages in real-time.

Different chat applications is providing different features to facilitate the users for different types of communications. Here the authors proposed a new chat application by the name PRIVY, with two new features viz: video conferencing and screen sharing facility.

II. EARLY CHAT SYSTEMS

The history of chatting can be traced back to the early days of computer networking and the development of the Internet. Here's a brief overview (7) of the key milestones in the history of chatting.

- In the 1960s and 1970s, chat systems began to emerge on early computer networks. One of the first notable examples was the compatible Time-Sharing (CTSS) developed at the Massachusetts Institute of Technology (MIT) in 1961, which allowed multiple users to chat in real time.
- Internet Relay chat (IRC) was developed in 1988 by Jarkko Oikarinen. It became widely popular in the early 1990s and allowed users to join chat rooms and communicate through text-based messages in real time.
- Instant Messaging (IM) gained popularity in the 1990s with the introduction of services like AOL Instant Messenger (AIM), ICQ and MSN Messenger. These platforms allowed users to send real-time text messages to individuals or groups and often featured additional functionalities like file sharing, voice chat and online presence indicators.
- Web-based chat popularity increased with the development of the World Wide Web, and web-based chat flatforms started to emerge. Chatrooms and chat services embedded within websites allowed users to engage in real-time conversations without the need for dedicated software. Many websites, particularly in the late 1990s and early 2000s, offered chat services as a way to interact with visitors.
- Social Media Chat gain popularity in the 2000s brought chat functionalities to a broader audience. Services like Facebook, Twitter, and LinkedIn integrated chat features, enabling users to communicate with their connections in real time. These platforms expanded the concept of chatting by incorporating multimedia elements like photos, videos and emojis.
- Mobile chat Applications are becoming popular all over the world with the adoption of smartphones. Apps like WhatsApp, WeChat, Line, and Telegram gained

popularity, offering instant messaging, voice calls and video calls on mobile devices. These apps often leveraged internet connectivity or data plans to provide a cost-effective means of communication.

- Chatbots and AI-Assisted Chat have become increasingly prevalent in recent years. AI-powered virtual assistants, like Apple's Siri, Google Assistant, or Amazon's Alexa, incorporate chat-based interfaces to interact with users and offer various services.
- Messaging platforms like WhatsApp, Facebook Messenger, and Slack have become dominant in the chat landscape. These platforms combine various features, including text messaging, voice calls, video calls, group chats, and integrations with other services. They often provide end-to-end encryption and focus on user privacy and security.

III. PRIVY-MERN CHAT APPLICATION

The existing methods of remote communication and collaboration are often inefficient, inconvenient, and unreliable, particularly for users who require real-time communication and interactive features. Traditional messaging and conferencing tools often lack the necessary functionality to support complex user interactions and can be prone to technical issues and delays. Furthermore, many existing tools require users to switch between multiple platforms and applications to access the full range of features they need, leading to confusion, frustration, and reduced productivity.

To address these challenges, we propose the development of a PRIVY chat application using the MERN stack that allows users to connect in real time. It is designated to provide a seamless experience for users who want to communicate with their peers or colleagues, collaborate on projects, or simply stay in touch with their friends and family. This chat application with meetings provides users with a single, integrated platform for real-time communication, collaboration, and video conferencing. The application will leverage the latest technologies and best practices in software development, including Socket.io for real-time messaging, JWT and bcrypt.js (8) for cure authentication, Chakra-UI (9) for quality UI, and MongoDB for efficient data storage and retrieval. The app features several functionalities, including real-time chatting with profile details, group creation, deletion, and updating for group chats. It also allows users to initiate video meetings with mute and unmute options for audio and video, as well as screen-sharing capabilities.

The MERN stack is an acronym for MongoDB, Express JS, ReactJS, and NodeJS. It is a popular web development stack used to build scalable and efficient web applications. Each of these components plays a crucial role in the development of Privy, allowing for a robust and reliable application.

Fig. 1: Acronym (M-MongoDB, E-Express JS, R-ReactJS, and N-NodeJS)

IV. FEATURES OF THE PRIVY

The proposed system is a MERN stack chat application with meetings that provides users with a comprehensive and user-friendly platform for remote communication, collaboration, and video conferencing. The system will leverage the latest technologies and best practices in software development to provide users with fast, efficient, and reliable communication and collaboration tools. The system will be designed with user experience in mind, providing a clear and intuitive user interface that is easy to navigate and use.

There are three important facilities in Privywhich are narrated hereunder.

A. Communication:

The facility feature of Privy is to provide a reliable and effective communication tool for users. This includes real-time chatting, profile details, and group creation, deletion, and updating features.

- *Real-time Chatting:*

The real-time chatting feature of Privy allows users to communicate with each other in real time. This feature is essential for effective collaboration and is a key feature of the project.

- *Profile Details:*

The profile details feature provides users with important information about their peers, including their names, profile pictures, and status. This information makes it easier for users to identify and communicate with the right people.

- *Group Creation, Deletion, and Updating:*

The group creation, deletion, and updating features of Privy allow users to organize their conversations and collaborate more effectively. These features are designed to make communication more efficient and streamlined.

B. Collaboration:

The second facility of Privy is to provide users with a range of collaboration tools that make it easier to work together on projects. This includes meetings with mute and unmute options and screen-sharing capabilities.

ISSN No:-2456-2165

➢ *Meetings with Mute and Unmute Options:*

The meetings with mute and unmute options feature allows users to control the audio and video during their meetings. This feature is particularly useful for large group meetings, where it can be difficult to manage multiple speakers at once.

➢ *Screen Sharing:*

The screen-sharing feature of Privy allows users to share their screens with other participants in their meetings. This feature makes it easier to collaborate on projects or share important information, particularly in remote teams or online classes.

C. User Experience:

The third feature of Privy is to provide a seamless and user-friendly experience for users. This includes easy navigation, intuitive interface design, and fast and responsive performance.

➢ *Easy Navigation:*

The easy navigation objective of Privy is to provide users with a simple and intuitive interface that makes it easy to find the features they need. This objective is essential for a seamless user experience.

➢ *Intuitive Interface Design:*

The intuitive interface design objective is to create an interface that is easy to understand and use, even for users who are not familiar with the app. This objective is essential for making the app accessible to a wide range of users.

➢ *Fast and Responsive Performance:*

The fast and responsive performance objective of Privy is to ensure that the app runs smoothly and quickly, even on slower devices or connections. This objective is essential for providing a positive user experience and ensuring that users can communicate and collaborate effectively.

V. SYSTEM DESIGN AND ARCHITECTURE

The process of defining a system's constituent parts, including its architecture, modules, and components, as well as its many interfaces and the data it processes, is known as system design. The primary focus of systems design is on establishing the architecture, parts, modules, interfaces, and data for a system to meet predetermined requirements.

The system architecture provides a high-level understanding of how the system functions. The following is an illustration of how this system functions:

The chat page is the main hub for communication in this system, where users input their messages. From there, users can navigate to different sections of the page using the search, my chats, and menu options. The search option allows users to look for other users with whom they can have an individual chat. My chats option leads to individual chats, group chats, and the ability to create a new group. Selecting the create group option takes users to a page where they can create a new group, which will then appear in the group chat section. The menu option leads to three different pages: meet, profile modal, and log out. The meet page allows users to join a meeting or conference, the profile modal allows them to view and edit their personal information, and the log-out option logs them out of the system. Clicking on the close button while in a meeting or conference will bring the user back to the chat page. Finally, selecting the log-out option will take the user back to the input screen where they can log in again.

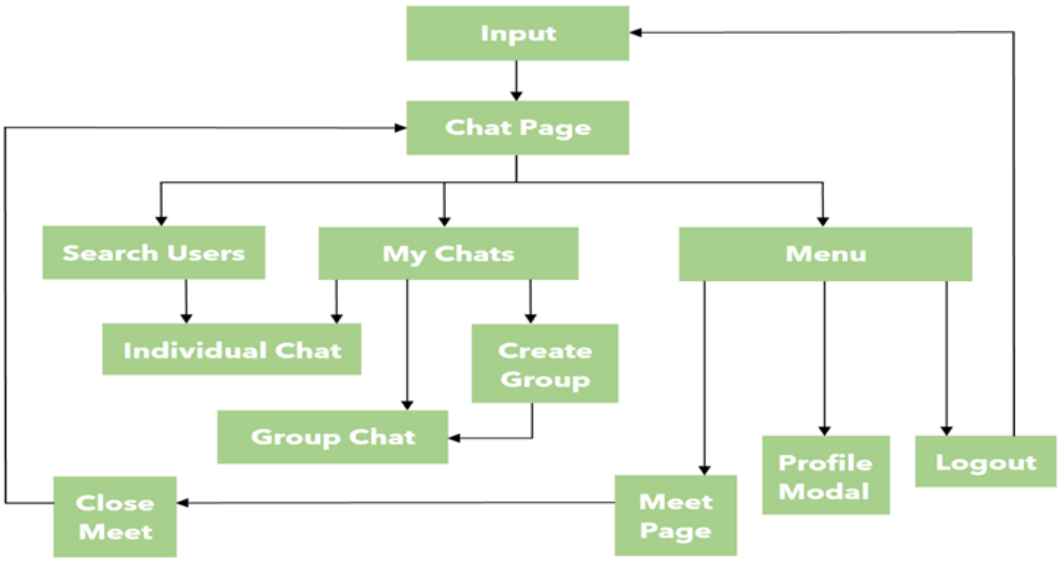

Fig. 2: Individual Chat Features

From Fig (2), the individual chat feature allows users to communicate one-on-one with another user. Once a user selects an individual chat from my chats section, they will be taken to the chat interface where they can input and send messages to the other user. Additionally, by clicking on the individual chat menu icon, users can access the interlocutor's profile modal, where they can view more information about the user, such as their profile picture, name, and email.

ISSN No:-2456-2165

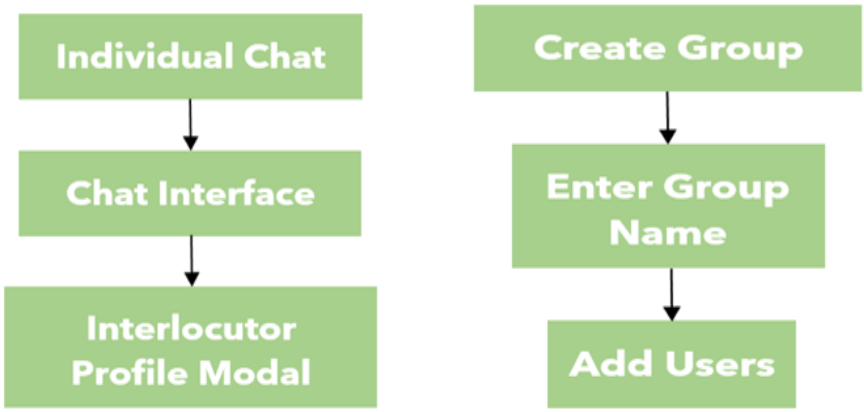

Fig. 3: Individual and Group structure

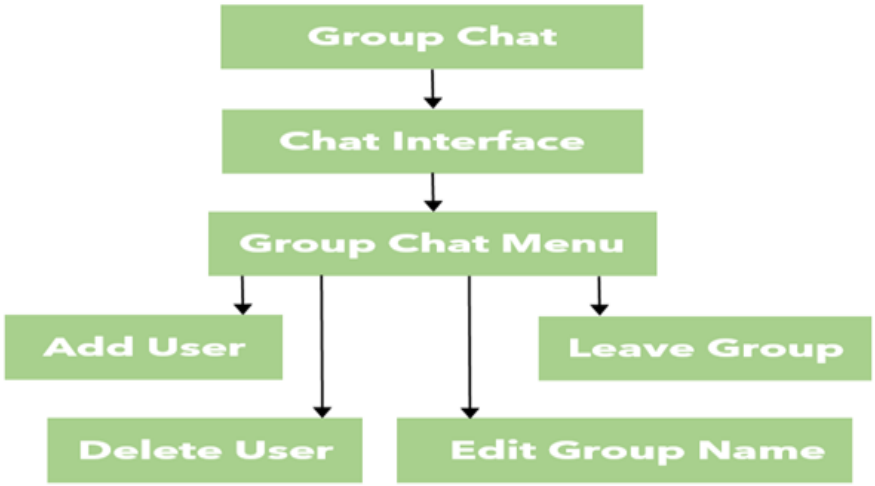

Fig. 4: Group chat Structure

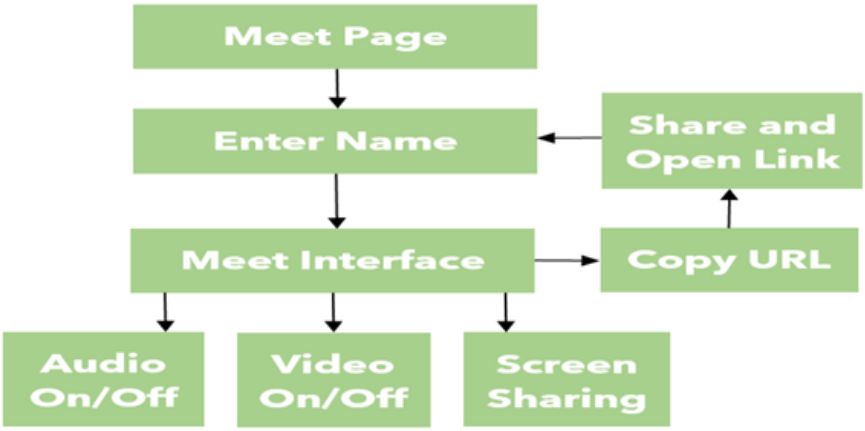

Fig. 5: Meet Page Structure

VI. SYSTEM IMPLEMENTATION

System implementation is the process of converting a system design into an operational system by building and deploying it in a production environment. It involves developing and integrating software, hardware, and other necessary components into a complete system that is ready to be used by end users.

During system implementation, the system design is translated into code, tested, and integrated with other components to form a complete system. This process includes tasks such as programming, testing, documentation, training, and deployment. The main goal of system implementation is to ensure that the system is fully functional, meets the user's requirements, and is easy to use.

ISSN No:-2456-2165

A. Server-side implementation:

- Set up a Node.js server using the Express.js framework: Node.js(10) is a JavaScript runtime that allows us to run JavaScript on the server side. Express.js is a popular framework built on top of Node.js that provides a variety of useful tools and features for building web applications, including routing, middleware, and error handling.
- Use MongoDB as the database and set up a connection using Mongoose: MongoDB is a NoSQL database that stores data in JSON-like documents. Mongoose is an Object Document Mapper (ODM) that allows us to define data models and interact with MongoDB using JavaScript.
- Implement user authentication using Bcrypt.js and JWT: Bcrypt.js is an authentication middleware that provides a variety of authentication strategies, including local authentication (using email and password), social authentication (using third-party providers like Google or Facebook), and token authentication (using JWT). JWT (JSON Web Tokens) is a standard for securely transmitting information between parties as a JSON object.
- Implement real-time communication using Socket.io: Socket.io (11) is a library that enables real-time, bidirectional communication between the server and clients. It uses web sockets as the underlying transport mechanism and provides a simple API for sending and receiving messages.

B. Client-side implementation:

- Use React.js to create the front-end user interface: React.js is a popular JavaScript library for building user interfaces. It uses a component-based architecture and provides a simple and intuitive way of building complex user interfaces.
- Use Redux to manage application state: Redux is a state management library that provides a predictable way of managing application state. It uses a unidirectional data flow and a single source of truth to ensure that the application state is always consistent.
- Use Socket.io client library to handle real-time communication with the server: The Socket.io client library is used on the client side to handle real-time communication with the server. It provides a similar API to the server-side library and allows for easy communication between the client and server.
- Implement the user interface for creating and joining chat rooms, sending messages, initiating audio and video calls, etc.: The user interface is implemented using React components and styled using CSS or a CSS framework like Chakra UI.
- Using Chakra UI in client-side implementation can help developers save time and effort in building UI components from scratch. The library's customizable and accessible components can be easily integrated into a React application, providing a solid foundation for building a responsive and user-friendly UI.

VII. RESULTS (REFERENCE 12-17)

A. Sign Up page:

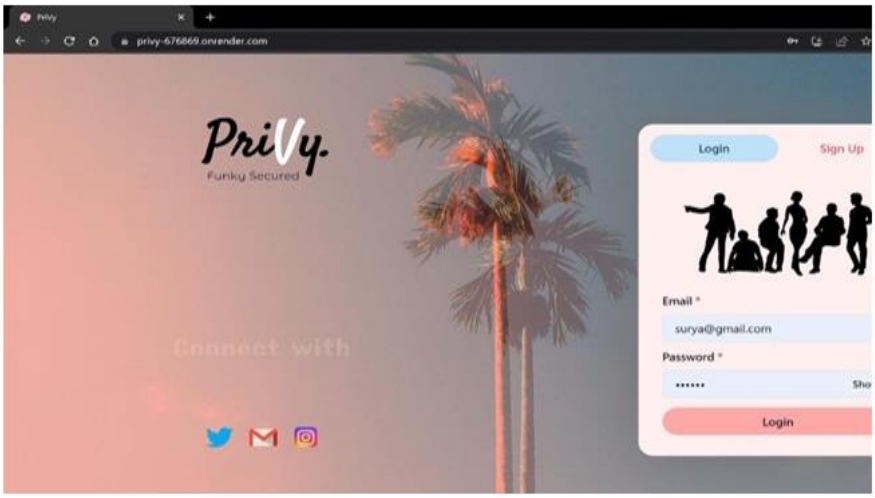

Fig. 6: Sign-up page

ISSN No:-2456-2165

B. Login page:

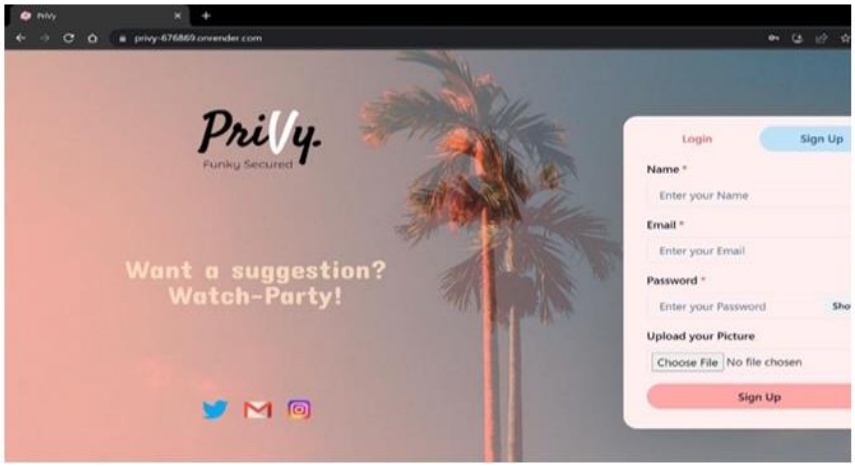

Fig. 7: Login page

C. Chat page:

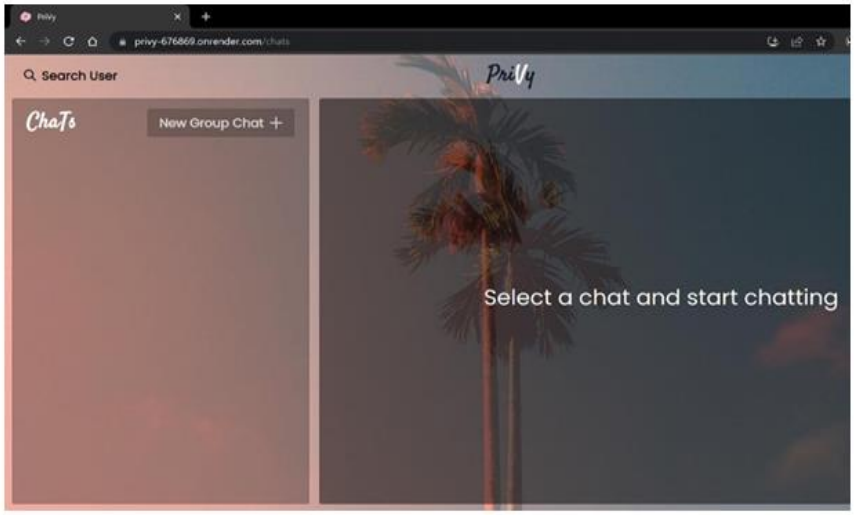

Fig. 8: Chat page

D. Search user:

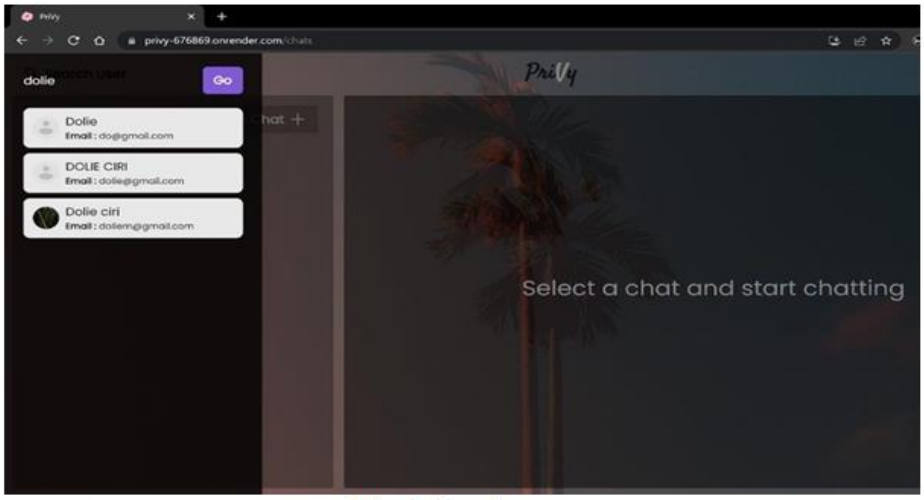

Fig. 9: Search user

E. Individual chat:

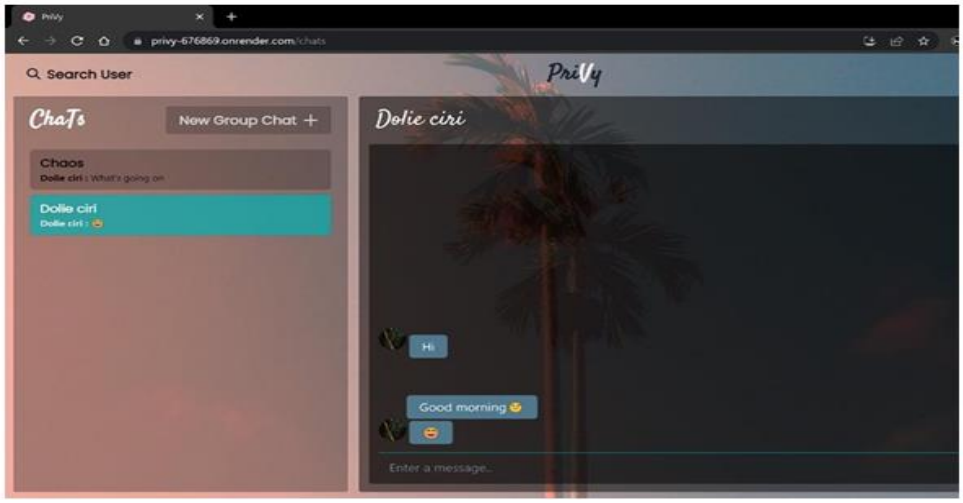

Fig. 10: Individual chat

F. Interlocutor profile:

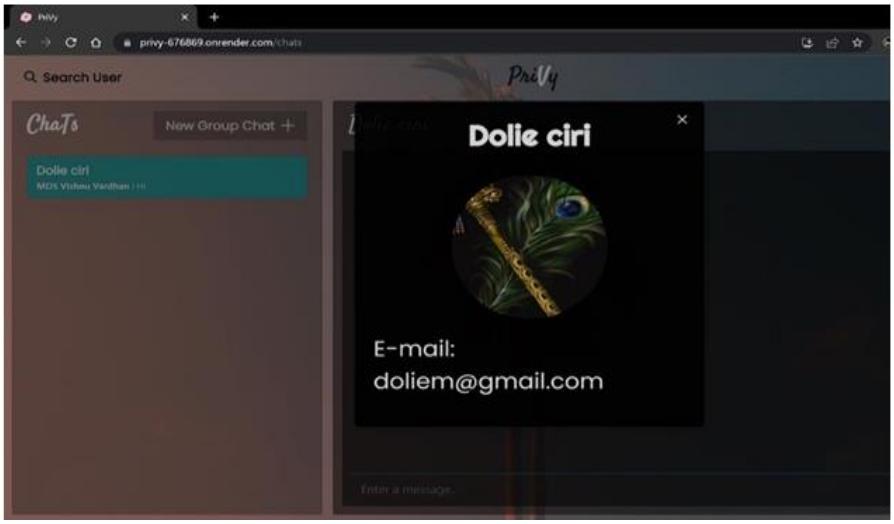

Fig. 11: Interlocutor profile

G. Create group chat:

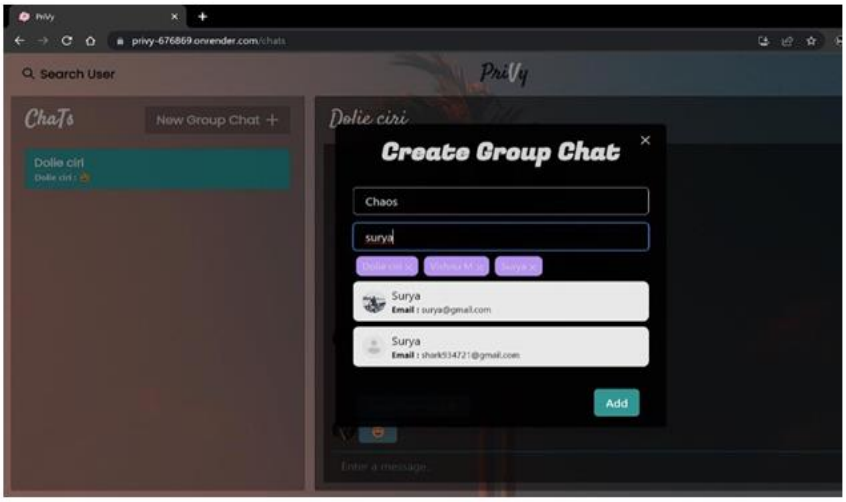

Fig. 12: Create a group chat

H. Group chat:

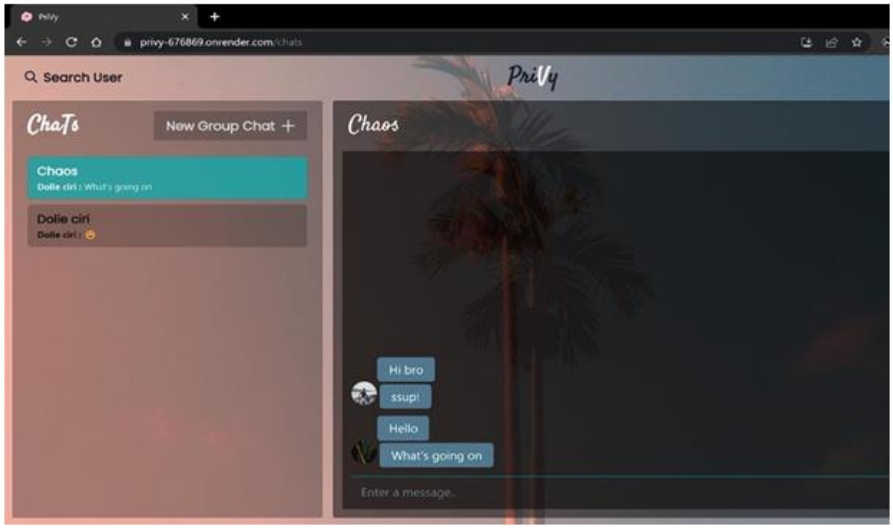

Fig. 13: Group chat

I. Edit group chat:

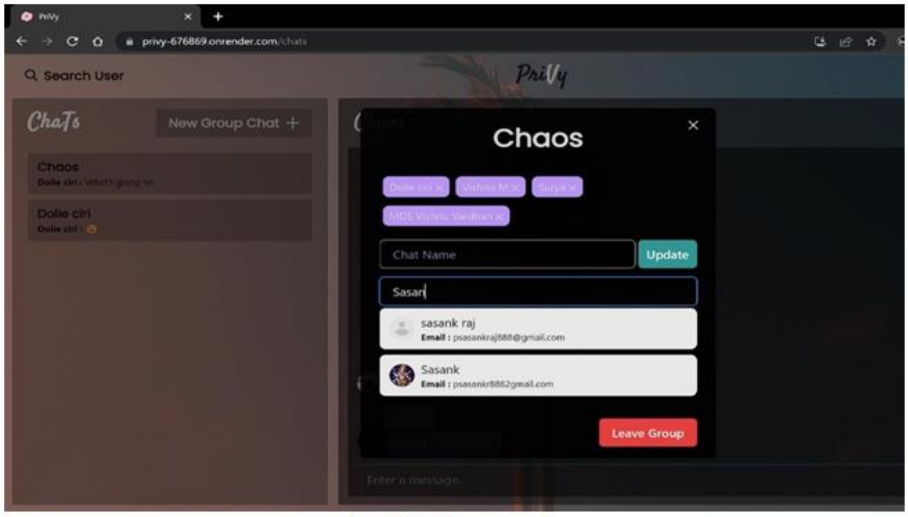

Fig. 14: Edit group chat

J. Notifications:

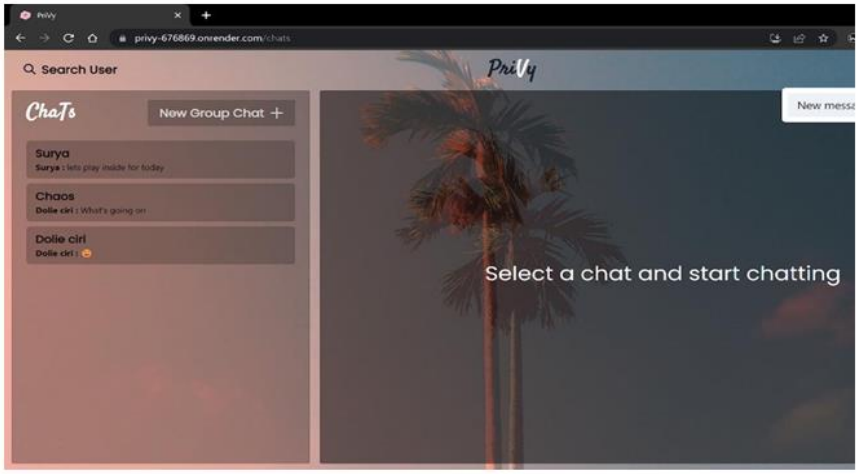

Fig. 15: Notifications

K. Meet page:

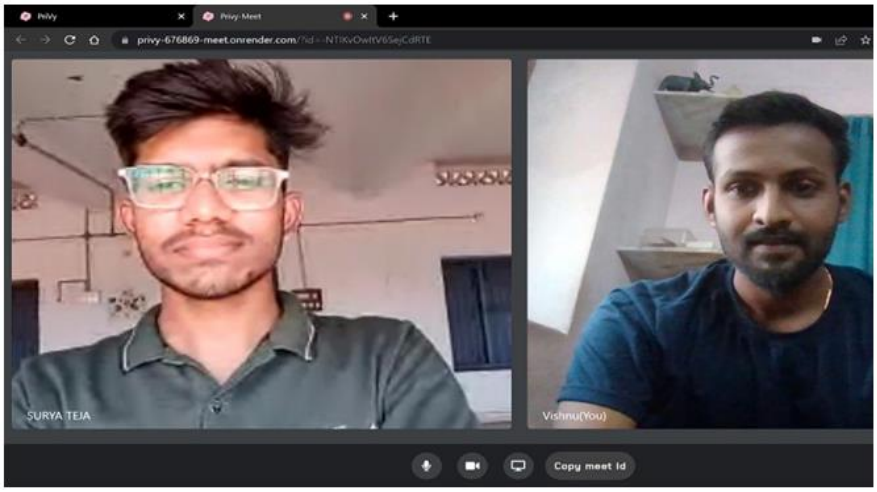

Fig. 16: Meet page

L. Menu:

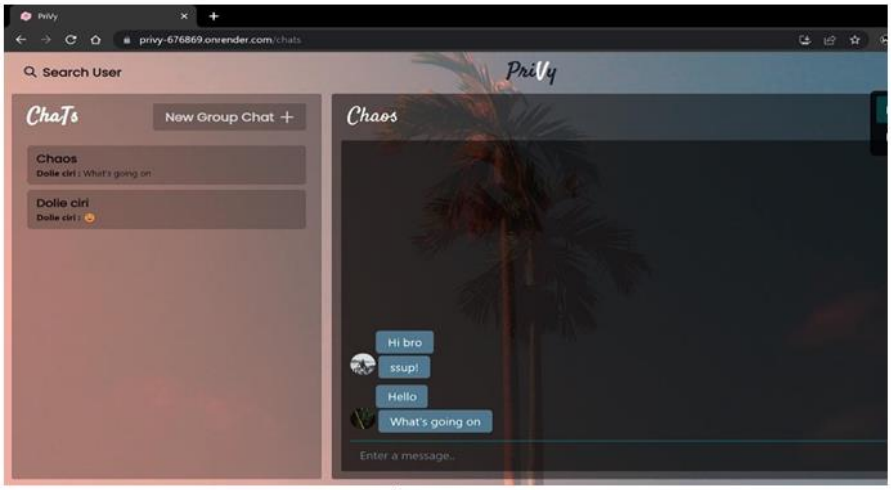

Fig. 17: Menu

M. My profile:

Fig. 18: My Profile

N. Logout:

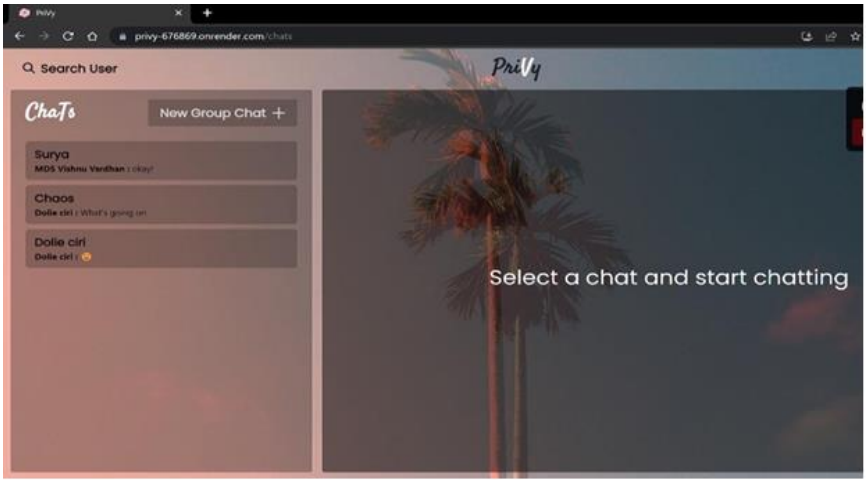

Fig. 19: Logout

VIII. CONCLUSION

Video conferencing is a powerful tool that allows users in improving communication, collaboration, and productivity. With video conferencing, users can see the facial expressions and body language of the other person, which is not possible through text-based communication. Our chat app comes with an integrated video conferencing feature that allows users to make video calls.

Moreover, our chat app also provides the option to share files, which is another essential feature that is missing in many of the existing chat app platforms.

In addition, our chat app provides a secure platform for users to communicate with their contacts. We have integrated features like end-to-end encryption and authentication using JWT and bcrypt.js to ensure the confidentiality and privacy of the messages exchanged between the users.

When compared to other chat app platforms, our chat app stands out due to its added features of video conferencing and file sharing. Our app is also more secure, making it a safe platform for users to communicate with their contacts.

Furthermore, our chat app is built using modern web technologies like React, Node.js, and Socket.io, which make the app fast, responsive, and scalable. We have also used Chakra UI, which is a simple and modular UI component library that allows us to design and develop a visually appealing and responsive user interface.

In conclusion, our chat app is a unique platform that provides an integrated solution for messaging, video conferencing, and file sharing. The app is secure, user-friendly, and built using modern web technologies.

REFERENCES

[1.] https://www.digitalcitizenship.nsw.edu.au/articles/what-are-the-benefits-of-chat-and messaging#:~:text=Key%20message,and%20help%20each%20other%20out.
[2.] MongoDB retrieved from https://www.mongodb.com/what-is-mongodb .
[3.] Express.js. retrieved from https://expressjs.com/ Mardan, A. (2018). Express.jsGuide: The Comprehensive Book on Express.js.
[4.] React. (n.d.). Getting Started. Retrieved from https://reactjs.org/docs/getting- started.html Wieruch, R. (2020). The Road to React: Your journey to master plain yet pragmatic React.js.
[5.] Node.js. (n.d.). About Node.js. Retrieved from https://nodejs.org/en/about/ Haverbeke, M. (2018). Eloquent JavaScript: A Modern Introduction to Programming.
[6.] https://dictionary.cambridge.org/dictionary/english/chat
[7.] https://towcenter.gitbooks.io/guide-to-chat-apps/content/introductionthe_dawn_of/a_brief_history.html
[8.] Bcrypt.js. (n.d.). Bcrypt.js. Retrieved from https://github.com/kelektiv/node.bcrypt.js#readme Krishnan, K. (2018).
[9.] Mastering Node.js 11.x: Master the art of building scalable and powerful web applications using Node.js.
[10.] Chakra UI. (n.d.). Chakra UI. Retrieved from https://chakra- ui.com/docs/getting-started Brown, B. (2021). Learning React: A Hands-On Guide to Building Web Applications Using React and Redux.
[11.] JSON Web Tokens. (n.d.). JSON Web Tokens. Retrieved from https://jwt.io/introduction/ Granger, J. (2017). Securing Node.js: Protecting Your Code and Your Users.
[12.] Socket.IO. (n.d.). Socket.IO. Retrieved from https://socket.io/ Griffiths, D., &
[13.] Griffiths, D. (2014). Head First HTML5 Programming: Building Web Apps with JavaScript.

[14.] Firebase. (n.d.). Firebase. Retrieved from https://firebase.google.com/docs Quinlan, B. (2017).
[15.] Firebase Essentials for Android: Get to grips with Firebase essentials for Android and learn to use it to build high- performance, feature- packed apps.
[16.] Dotenv retrieved from https://www.npmjs.com/package/dotenv Cahill, K. (2018). Web Development with Node and Express: Leveraging the JavaScript Stack. . (n.d.). Font Awesome. Retrieved from https://fontawesome.com/ Mall, D. (2018). Bootstrap 4 Site Blueprints Volume II: Build a responsive website and landing page using Bootstrap 4.
[17.] Redux. (n.d.). Redux. Retrieved from https://redux.js.org/ Goyvaerts, S. (2017). React Native Blueprints: Build nine projects by leveraging the power of React Native.
[18.] Web Vitals. (n.d.). Web Vitals. Retrieved from https://web.dev/vitals/ Cushing, B. (2021). Mastering Performance: Faster Websites, Applications, and APIs. NPM. (n.d.). What is NPM? Retrieved from https://docs.npmjs.com/about-npm/ Osmani, A. (2018). The Modern JavaScript Collection.
[19.] R. Alkadhi, T. Lata, E. Guzman, and B. Bruegge. Rationale in Development Chat Messages: An Exploratory Study. In Proc. of the 14th Work. Conf. on Min. Softw. Repos., MSR'17, pages 436-446, 2017.
[20.] “Signalling and video calling,” MDN Web Docs. At https://developer.mozilla.org/enUS/docs/Web/API/WebRTC_API/Signal ing_and_video_calling.