WEB- BASED CHAT APPLICATION USING REACT

Ajay Kumar Sahu1**[*]**, Vishesh Vishwakarma2, Shubham Yadav3, Amanat Anand4

[1*]ajay4989@gmail.com, [2]visheshofficial3@gmail.com, [3]shubham7524yadav@gmail.com, [4]gniot.amanatanand.2m@gmail.com

[1]Greater Noida Institute Of Technology, Knowledge Park, Greater Noida-201310, India

[2]Greater Noida Institute Of Technology, Knowledge Park, Greater Noida-201310, India

[3]Greater Noida Institute Of Technology, Knowledge Park, Greater Noida-201310, India

[4]Greater Noida Institute Of Technology, Knowledge Park, Greater Noida-201310, India

Abstract: A chat application is a useful tool in the present time because it enables people to communicate with one another from anywhere and in real-time without inconvenience. This web-based chat tool also eliminates the need for users to call each other every time they have a question about their work. Nowadays it is being used a lot in every field whether it is online banking, online e-commerce, or business. The best feature of the app is that together we can send a similar text to several people at the same time because we may create a group of a specific number of people or send messages to a large number of users, including Sender Message wishes to convey. The biggest example of this is when you're on a website, such as a bank, and you encounter bank account-related issues, you can use the website's chat feature to send an instant message to an agent so that the question can be answered quickly and efficiently. In today's time, apps like WhatsApp, Skype, and Messenger are the biggest examples of chatting apps. WhatsApp itself has been downloaded over 1 billion times on the Play store alone. The popularity of the chat application can be gauged from this fact. Today's chatting apps have become so advanced that with their help we can easily make video calls, and send audio files, documents and locations too. By lowering average contact costs, live chat lowers overall contact center expenses. This increases efficiency by allowing live chat agents to manage many chats at once, minimizing the need for additional representatives (Goel, A. K., 2022). In this research paper, we have mentioned a react-based web chat application and also we have compared react based chat application with another technology " PHP "that will help us to understand which technology is suitable for developing chat applications. There's no need for third-party client software with this web-based real-time chatting tool. The app is built with React. js, and Node.js with the Express package, as well as a Mongo DB database. Text messages are sent to and received from the server, and via a point-to-point connection between the servers. Transfer of data from is enhanced. The concept of virtual environments has been adopted due to the use of the React framework, which increases performance by a factor of about 6 over traditional PHP-based apps.

Keywords:Chat,Socket.Io,MERNMultipleUser,Node.js,ChatApplication, MEAN, reactjs.php.

1. INTRODUCTION

Communication has been simpler now than ever before new advancements in information technology. There are programs that enable in communication by sending words, pictures, data, and so on from one people to the other. Several similar applications are being utilized to connect with the large population. Typically, these applications are aimed at the broader public and benefit community overall. There are few programmers that enhance communication within associations such as universities, industries, as well as corporations by limiting the number of end user and keeping the information transferred among the members of the organization secret. As a result, the purpose of our project, the online chatting app, is to solve this issue and provide people with enhanced experience that maintains texts at a distance and restricted within a barrier. This article provides an online infrastructure for personal conversations which allows clients to connect in real time in a shared environment. The paradigm presented here varies from previous collective circumstances in that it provides novel computations, standards, and techniques for the production, synchronization, and extensive and other information connections (Gackenheimer, C, 2015). The suggested technique is based on using the WebPages as a mixed content provider with messaging platform, capable of extracting or pushing material to a large number of clients from varied application behaviors on one or more website pages. The study's approach is built on integrating clients into a complex communication process that has been supplemented with purportedly "Smart Objects" that provide further distinctive usefulness. (Gupta, K., 2021). To give clients with ease, this conversation program will be developed as a web and mobile application. This messaging app was built with React.js, Node.js and the express framework, as well as a Mongo DB database. Servers provide text messages, and data transfer is simplified by point-to-point links between servers. The virtual space concept is adopted since the react framework enhances performance by a factor of six when compared to normal PHP-based apps. This app is designed for users at companies and institutions that have their own infrastructure of server in order to give users with privacy. Aside from the application's specific usage, it might also be used in a more general sense to facilitate services to the common public. This application was created with React.js, HTML, SASS, MongoDB, Node.js, Express (Shekhawat, A., 2019). . Node.js is a programming framework for developing server-side adaptable applications for network applications.

HTML is the accepted industry mark-up specifically for web browser pages. SASS is a pre-processor supporting CSS styles. The React.JS technology is utilized since this project is meant for mobile apps. It is a JavaScript library used to create user interfaces (Sheiko, D., 2017). React can also be used to create single-page or mobile applications.

According to the above description, the issue is identifying how to construct and create a customized mobile phone app and web-based chat framework focused at a certain section or the general public in genuine time to make the messaging service and the sharing process simpler and faster (Tharanidharan, S. K., 2022).

2. LITERATURE SURVEY

Previous chat applications, such as Yahoo Messenger, relay on PHP and MySQL to manage servers and handle client requests. PHP is a server-side, cross-platform, HTML-enabled utility for creating web pages

(Kumar, T. S., 2021). MySQL is a free and open-source framework for managing relational databases. It is the most used administration solution for PHP databases. Oracle created and maintains MySQL. However, because of their synchronous nature, PHP will be a highly time-consuming operation for massive database retrieval (Verma, D. 2022). jQuery, a JavaScript-based server design tool that employs the DOM (Document Object Model) capability to explore a document, select DOM components, generate animations, handle events, and construct Ajax applications, is another related technology used in hangouts (Biswas, N., 2021) jQuery is also used by developers to build plug-ins that act upon the JavaScript library Developers may utilize this to build abstractions for low-level interaction and visual effects, complicated effects, and high-level, visual style panels (Kane, A., 2016). The modular nature of the jQuery framework enables the building of complex dynamic web pages and Web apps. Because the React framework may increase speed, the notion of virtual space boosts productivity by updating the space produced for that specific request rather than calling to update the entire work area.

3. OBJECTIVE

The aim of this paper as described by the above-mentioned major challenge, is to develop a good real-time multi-platform chat application that anyone can use to make their lives simpler by exchanging information and chatting with one another effortlessly and fast and also it will manage the details of user chat history their chat profile automatically using react technology. Whenever the user wants to see the past data, it should be available as a chat history. Before proceeding to the next level, the scope of application should be specified. The following are the application scopes:

1. This chat application's design and structure are accessible for all internet-based mobile and other device users.

2. Node.js, socket.io, and other JavaScript technologies were used to create this application.

3. The program's database is maintained in the MongoDB NOSQL database.

4. Only maximum 100 users can use the app at a time.

4. SYSTEM DESIGN

In this chat application two servers are used to build the backend interface: client and server. The server handles database retrieval, database maintenance, and client-side services, while the client handles page management and the application interface. This can be understood by Client 1 will be notified by the server that a message has been sent, and the client 2 will be able to access that message via the browser. Client 1 will be informed by the server that a message has been delivered, and the client 2 will be able to showcase that message using the browser. When a client sends a message, it is forwarded to the host, who passes it to the other service users in the room at the moment (Singh, T., Sahu, A. K., 2022). A website consists of collection of pages made available via an interconnected network and accessible to everyone on the internet anywhere in the world. First, communication between the client and server ends with Node.js (K. Suresh Kumar, 2022) acting as an intermediary via JSON. JSON (JavaScript Object Notation) is a human-readable and writable lightweight data transport format as well as readily translated and created (produced) by a machine. MongoDB, Express.js, React, and Node.js deliver the services. This channel allows for dual-path communication between the client and server. Because the MERN stack supports the use of a single language on both the server and client sides, every configuration is implemented solely using JavaScript. If the user launches the web chat application and does any actions on the website, JSON will commence the service, and after receiving the relevant data from the server, the results will be shown on the website.

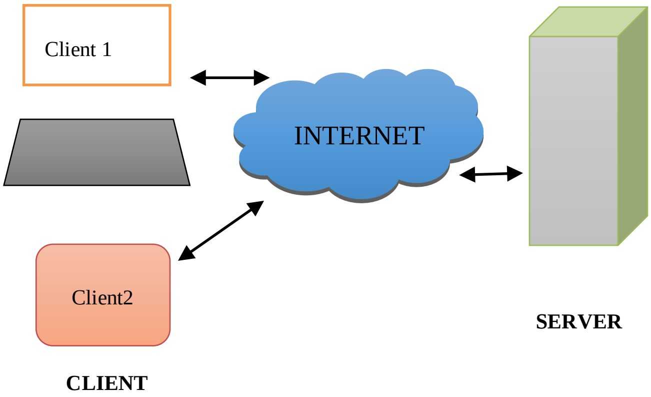

Fig 1: - Server Client Communication

The following procedure explains user interface design: -

1. When you first visit the website, you will see the main page, which contains choices for current user login and new user registration. If the user's information is already recorded in the database, the login page will prompt them to enter it. If the user chooses the signup option, they will be directed to a one-of-a-kind user signup page where their information will be captured and stored in the database.
2. Before beginning a chat after login into the site, the user must have a contact. As a result, they may find them immediately from the app. Following that, the user can add another person about their friend list while waiting for confirmation of the other user's request.
3. Furthermore, the message database must be searched for prior chat history, and if it exists, it should be displayed.
4. At last, users will be able to communicate with one another.

Figure 2 describe the workflow of the sequence of action that how the process will take place in the system. With the help of this flow chart, we can visualize the whole system.

The flow chart below displays the full chat application process for better comprehension.

1. When the software is run, the user is sent to the home page, where access to the main page is restricted. This may be accomplished by generating a new account for new users while keeping existing accounts for existing members.
2. After successfully checking in, the client is taken to the user's home page or a chat room with contacts shown. You may access their chat room by simply clicking on the name of the listed contact.
3. The group chat will display any information collected by the user through any of his or her contacts. To transfer the data, the user requires first visit the chat room of a specified contact.

	Field name	DataType	Explanation
1	_id	ObjectId	UserId
2	username	String	Email
3	password	String	Password
4	googleId	String	GoogleId
5	googleDisplayName	String	UserGoogleA/cName

5. IMPLIMENTATION

1. Registration:-Registration is the first step to enter in this chat application. It is a process that requires the user to provide confidential details including their name, email address, and phone number (Sahu, A. K., & Kumar, A., 2021). But in our chat application we will only use the email address for user signup. We will Google account to sign in this app because Google offers a signup process that is much faster and simpler than the traditional signup process. This process can be completed in two ways:

Table 1 -Sign in User Data Structure

Table 1 describe about the field that a user needs to field in order to get registered at very first time. Using the table 1 we can understand various fields and their accepted data type.

1.1 Enter your mail id. -User will enter their email id on login box.

1.2 Choose password: -After entering email id user need to choose their password for authenticating the app in future

2. Login to Your Account: -A login is a system procedure that allows a user to input their username and password to access a specific account. Before login user must ensure that they have already been registered their self by signing in the app.

2.1 Enter your mail id*:* - In order to login, user will have to enter their registered email address.

2.2 Enter your password: -After entering email address user will enter their password in password box.

2.3 If valid user moves step 3: - If the user's credential match with the database then it will be redirected to chatting page

2.4 If not a valid user go-to step 1: -In case of not being matched in database, user will have to repeat the step 1 from initial.

Table2-login User Data Structure

	CPU Time	System Time	RAM
PHP	100.65s	102.20	2497420KB
Node.js	1.54s	1.64s	93340KB

Table 2 describe that we have used string data type that can be enter into the username and password field at the time of login.

3. Find Friends to Start Chatting: -After successful login the user will redirect to chat page where they can see the already registered or available users on the app.

4. Chatting: -The user can initiate its chatting with another available user. User need to type their message in message box. After typing the message user hit the enter button by keyboard or send button by mouse the message will be delivered to the selected user. User can easily chat with friends and check their messages too.

5. Groups: -The user can also create group to chat with more than one user at a time. The message will be delivered to ever member of group with more than one people at a same time.

6. Logout: - To leave or terminate the chatting user can logout from their account. After logout user will be prompt to the login page. Figure 3 depicts the work flow of a chat room. The first step in using the chat system is to login. The login is linked to a condition; if the user has not yet registered, he must do so before he may login. They can proceed to the find friend, send message, initiate chat, and logout functions after successfully logging in.

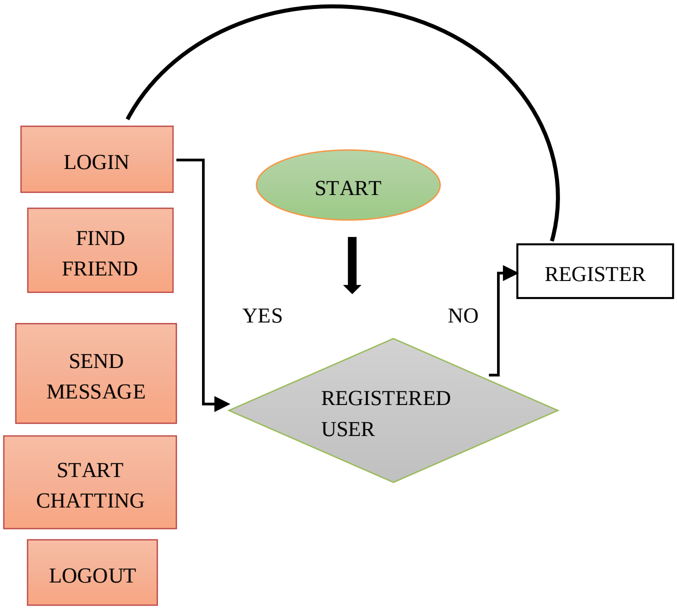

Fig 3: - Chat room work flow chart

6. TEST RESULT

We collected data in table 3 and table 4 after testing an app based on Node.js, React, and MongoDB with PHP and MySQL (speed parameter). Table 3 contains multiple CPU performance test data from both php-based and react-based applications at the same time. Table 4 compares the resources for php-based vs. react-based apps (Sahu, A.K., & Kumar, A., 2019).We can effectively analyze the performance of both technologies using the two tables below.

Table3: CPU performance test data

Table 4: Resource usage

	CPU Time	System Time	RAM
PHP	100.65s	102.20	2497420KB
Node.js	1.54s	1.64s	93340KB

In figure 4 we have created a graphical representation of performance of compared test data. From the figure 4 we can easily conclude that which technology is performing best.

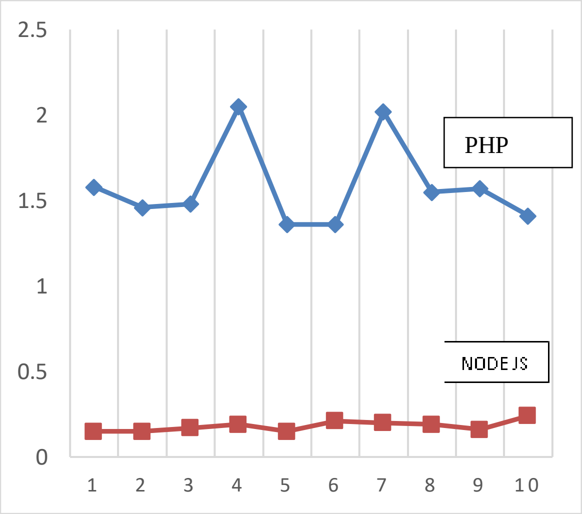

Fig4: - Graphical representation of performances

6. CONCLUSION

The outcomes of this research demonstrate that the chat app created with React and MongoDB, Node.js outperforms the application built with PHP and MySQL in real-time, with a runtime of around a second. Node.js exceeds PHP in terms of process time by more than 30 times and is better in terms of RAM use. Apart from that Real Time Chat Application" is more interesting and imaginative than creating a website about a company or a catalog. As a result, I can say that socket.io reduced the task of designing and implementing our own chat app. I can claim that this task taught me far more than I anticipated. Chat programs are in great demand and are utilized everywhere in today's society, minimizing our difficulties and saving us time. As I previously stated, we can use chat applications to send its same message to several users at the same time, removing the need to call someone every time. Finally, I'd want to emphasize that creating a chat app with Node.js and Socket.io is perfect for two reasons. The first is that it saves time by allowing you to type in seconds, and the second is that you can always learn more while developing this application and will have a good handle on the technology by the time it is finished.

No.	Chat with PHP and MySQL(in seconds)	Chat with Node.js, MongoDB and React(in seconds)
1	1.58	0.15
2	1.46	0.15
3	1.46	0.17
4	2.05	0.19
5	1.36	0.15
6	1.36	0.21
7	2.02	0.20
8	1.56	0.19
9	1.57	0.18
10	1.41	0.24

There is always scope and potential for improvement in every application. Our aim is only engaging with text-based exchanges for now. Other chat software provides services comparable to our product, however they are complicated to use it and have complex layouts. In both human relationships and human-computer interactions, making a strong first impression is critical. The project's goal is to create a chat service Web application with a high-quality user engagement. We are certain that by integrating these services, we will be able to make the application more successful in the future, based on the knowledge gained while developing this application.

- File Transfer
- Video Message
- Audio Call
- Video Call
- Group Call
- Adding Authorization service to this application
- Creating and managing users in a database.
- It will be extended to Web Support.
- Increasing the application's efficacy by including voice-based conversation

REFERENCES

Goel, A. K., Gupta, S., Singh, C. K., & Agrawal, K. K. (2022). Web-ChatLine: An Innovative Chatting Platform. Materials Proceedings, 10(1), 6.

Gackenheimer, C., & Gackenheimer, C. (2015). Introducing flux: An application architecture for react. Introduction to React, 87-106.

Gupta, K., Btech, C. S. E., Srivastavaa, N., & Goswami, V. (2021). CZAT–A Web Application Based RealTime Chat App.

Shekhawat, A. (2019). WEB BASED CHAT APPLICATION (Doctoral dissertation, GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI).

Sheiko, D. (2017). Cross-platform Desktop Application Development: Electron, Node, NW. js, and React. Packt Publishing Ltd.

Tharanidharan, S. K., Yaswanth, P. S., Sastry, M. S. V. C. M., & Sindhwani, M. (2022). Real Time Web Based Multilingual Chat Application. In Futuristic Sustainable Energy and Technology (pp. 33-42). CRC Press

Kumar, T. S., Reddy, V., DL, S., & Rananavare, L. (2021). INTERNET CHAT APPLICATION. International Journal of Advanced Research in Computer Science, 12.

Verma, D. (2022). A comparison of web framework efficiency: performance and network analysis of modern web frameworks.

Biswas, N., & Biswas, N. (2021). Creating a Video Chat Site. Foundation Gatsby Projects: Create Four Real Production Websites with Gatsby, 349-423.

Kane, A. (2016). ITHub Chat Application (Doctoral dissertation, Dublin, National College of Ireland).

Singh, T., Sahu, A. K., Dubey, S., Sharma, M. P., Verma, S., & Kumar, C. (2022), "Treatment of thyroid disease through Machine Learning Predictive Model", International Journal of Health Sciences, 6(S8), 3176–3188. https://doi.org/10.53730/ijhs.v6nS8.12813, Scopus & Web of Sciences. http://sciencescholar.us/journal/index.php/ijhs.

K. Suresh Kumar, Vinay Kumar Nassa, Dipesh Uike, Ashima kalra, Ajay Kumar Sahu, Vijay Anant Athavale, V. Saravanan (2022), "A Comparative Analysis of Blockchain in Enhancing the Drug Traceability in Edible Foods Using Multiple Regression Analysis", Journal of Food Quality, vol. 2022, 6 pages. https://doi.org/10.1155/2022/1689913. Hindawi, ISSN: 1687-5265, Impact Factor. 3.200, indexing: Scopus & SCI. https://www.hindawi.com/journals/cin/.

"A Review on Smart and Intelligent Techniques for Digital Tourism" in the International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), Volume 8, Issue 6, November December 2022. Link https://ijsrcseit.com/CSEIT228651, UGC Journal No. : 64718, Impact Factor: 7.254. DOI: https://doi.org/10.32628/CSEIT228651.

Sahu, A. K., & Kumar, A. (2021), "A Novel Verification Protocol to Restrict Unconstitutional Access of Information From Smart Card", International Journal of Digital Crime and Forensics (IJDCF); Vol. 13(1): pp. 65-78. doi:10.4018/IJDCF.2021010104 (ESCI, Scopus Journal).

Sahu, A.K., & Kumar, A. (2019), "SPAS: An Authentication Scheme to Prevent Unauthorized Access of Information from Smart Card", Pertanika Journal of Science and Technology; Vol. 27(1): pp. 175-192. ISSN: 0128-7680. http://www.pertanika.upm.edu.my/(Scopus Journal).