International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 6, Issue 11, ISSN No. 2455-2143, Pages 33-35
Published Online March 2022 in IJEAST (http://www.ijeast.com)

WEB CHAT STATION – A REVIEW

Shreenath Jagtap, Farhan Pathan and Sujal Jadhav
Department of Information and Technology
Jaywantrao Sawant polytechnic, Pune

Abstract: **In the modern world, people are communicating more over the internet than in person and many of people have choose network chatting tools for communication. Applications such as these facilitate communication over great distances. Therefore, this application must both be real-time and multi-platform to be used by many users. The web-based real-time chatting application does not require any additional third-party program, and the visual communication could be established without difficulty. The text communication is transferred through the servers and the data transmission is facilitated through point to point connection between servers.**

Keywords: **Chatting, Real-time, Multiplatform.**

I. INTRODUCTION

With the development in information technology, communication has become easier like never before. There are applications that help in the process of communication by relaying texts, images, files, etc. from one person to the other. Several such applications do exist that serve as a means to communicate to a large population. Such applications are often aimed at the general public and serve the society as a whole. There are very few applications that facilitate communication within organizations such as institutes, industries, and companies etc. that limit the number of users and keep the content being transferred among the users of the organization private. Therefore, this project, the web chatting application, is aimed at to overcome this problem and to provide users with a much better platform that keeps the texts at bay and confined within a boundary. This Web Application is made by using React JS, this is a JavaScript library for building user interfaces React can be used as a base in the development of single-page or mobile applications. And we used an API called Chat engine which handles all the chat which we will discuss below in methodology.

II. RELATED WORK

In the world of the web, there are Web Application that are used to send messages and share pictures with a large number of people. Some of these applications, such as Spark Matrix are designed with a picture-sharing logic. This means that when a user shares a picture, all the other users who have that picture will receive the message. This can be a great way to keep in touch with family and friends, but also a way to gather a large amount of information about people. Helping the people who do not have a compatible device for more storage and space that could handle the database for content sharing and file receiving our application is totally web based so the client doesn't need any compatibility of the device for storage and database only the thing they need is a proper internet connection and they are all set to go!

III. EXISTING SYSTEM

Below is the existing system which we are referring, the backend interface is implemented with two servers: client and Server. Server side is responsible for the database retrieval, database maintenance and their services to the client side, whereas the page maintenance and the application interface will be performed by the client side. Website is a set of pages provided through interconnected network that is available for everyone to access around the globe through network facility. First, the communication between client end and server end through Node.js which acts as intermediate or by the usage of JSON. JSON (JavaScript Object Notation) is a lightweight data interchange format, readable and writable by humans, as well as easily translated and made (generated) by the computer The services are provided by Mongo DB, Express.js, React and Node.js. Dual path communication between the client side and server side is possible through this channel. Here, every configuration is implemented only through JavaScript, because in MERN stack both server and client sides support the usage of common language. If the user starts using web chat application and perform any steps in the page, this will initiate a request to JSON, JSON would then initiate the service, then after obtaining the required information from the server end, it will provide the results in the page.

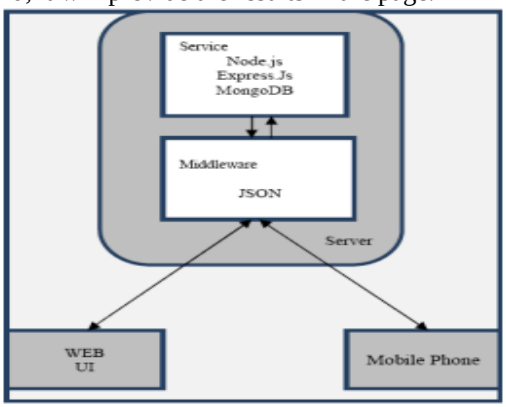

Fig 1: Existing Architecture

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 6, Issue 11, ISSN No. 2455-2143, Pages 33-35
Published Online March 2022 in IJEAST (http://www.ijeast.com)

IV. METHODOLOGY

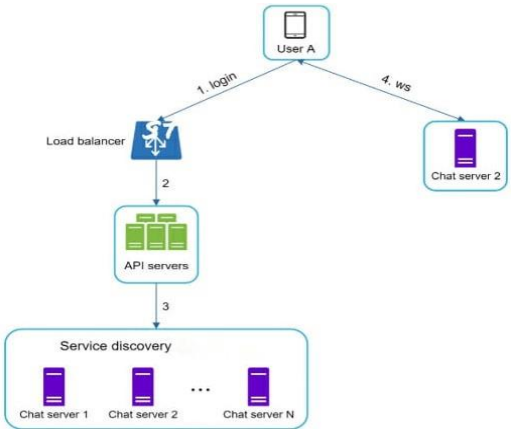

Fig 1: Proposed Architecture

Here, we propose our method for the system of Chat Station with the help of React JS aligned with the API called Chat Engine which helps the system to send messages and have data security compatibility integrity with the backup of data/content shared both by the sender and receiver through our system.

In Fig.1Proposed Architecture we can see our system where the API servers allocates the right data to the authorized client server which will receive the data with full integrity on the lowest medium available on their device. Here the API servers does the wholesome job of data maintainability of data managing all the messages /content that are to be sent to the specific client.

Our system helps the client to connect with the receiver as fast as possible where the API then sub-divide the request as per the service required the Chat Engine API here is more faster than node JS and Mongodb providing the client with at most consistency in the network connection for sharing content, messages, data, document, etc.

V. RESULT

Result of our system:

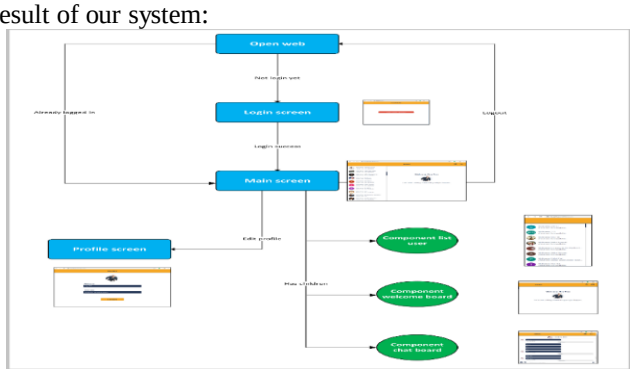

Fig. 1 User interface design

Login Page:

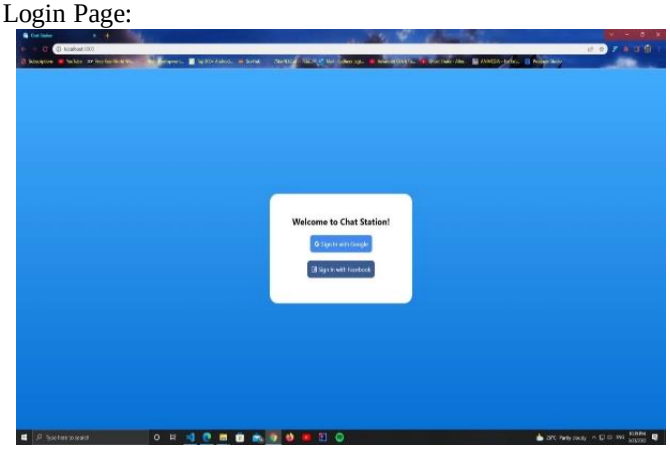

Fig. 2 Login Page

Home Page:

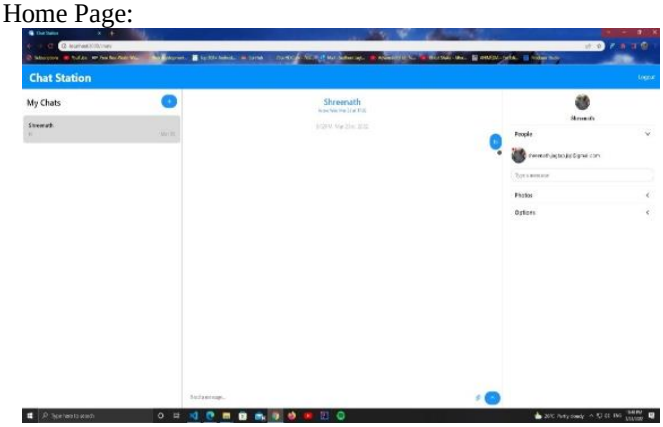

Fig 3 Home Page

So, here are some results of the system we propose to you of our web-based chat application in fig.1 We can see the design of the user interface in fig.2 It symbolises the login page of our system and fig,3 Symbolizes the home page or the main interface for chatting

VI. CONCLUSION

On running the tests, the chat application that we have developed using chat engine, React.js is faster in real-time with a speed less than a second compared to the application developed using PHP and My SQL. Node.js is faster than PHP and is more structured than PHP in terms of RAM usage. As well as it is web based so there is no need to have more storage on your device in order to have a backup you only need is an internet connection.

VII. REFERENCES

[1]. Croucher, T.H., & Wilson, M. (2012). Node: Up and Running. United States
[2]. Keissling, Manuel. 2012. The Node Beginner Book. Lulu.com, United States
[3]. Teixeira, Pedro. 2012. Hands-on Node.js. Wrox.

[4]. Sidik, B. (2011). JavaScript. Bandung: Informatika
[5]. Purnomosidi, B. (2013). Penbangan Sistem Informasi Penegelolaan Inventaris Barang Divisi PustekinBerbasis Web. Bandung: Politeknik Telkom.
[6]. Tim A. Majchrzak University of Agder, Kristiansand, Norway Andreas Biørn-Hansen Westerdals, Oslo, Norway Tor-Morten Grønli Westerdals, Oslo, Norway
[7]. Tim A. Majchrzak, Benjamin Ruland and Till Weber ¨ Department of Information Systems, University of Munster, M ¨ unster, Germany
[8]. A study of internet instant messaging and chat protocols published on 14 August 2006 by R. B. Jennings, E. M Nahum, D. P Olshefski, D Saha, Zon-yin Shae, Christopher J. Waters (http://ieeexplore.ieee.org/document/1668399/)
[9]. Robert W. Sebesta: Programming the World WideWeb, 8th Edition Pearson Education, 2015