International Journal of Innovative Research in Engineering
Volume 3, Issue 2 (March-April 2022), PP: 52-55
www.theijire.com
ISSN No: 2582-8746

Chat application using MongoDB, Express.js, React.js, Node.js (MERN) stack

Apoorva Srivastava[1], Gargi Srivastava[2], Bhargavi Pandey[3], Shubham Srivastava[4]

[1,2,3] UG students, Department of computer science, Institute of Technology and Management, Gida Gorakhpur, India.

[4] Assistant Professor, Department of computer science, Institute of Technology and Management, Gida Gorakhpur India.

How to cite this paper: Apoorva Srivastava1, Gargi Srivastava2, Bhargavi Pandey3, Shubham Srivastava4, "Chat application using MongoDB, Express.js, React.js, Node.js (MERN) stack", IJIRE-V3I02-52-55.

Abstract: *With the development and enhancement in internet, more and more people have been choosing chatting tools for communication. Applications such as these facilitates communication over great distances. Therefore this application must both be real time and multi-platform to be used by many users. Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers. The technology has been available for years but the acceptance was quite recent. Our project is an example of chat server. It is made up of two applications – the client application, which runs on the user's web browser and server application, runs on any hosting servers on the network. To start chatting client should get connected to server where they can do private and group chat. Security measures were taken during the last one.*

The programming tools used in building this application is React.js, Node.js with Express framework and Mongo DB database. The text communication is transferred through and from servers and the data transmission is facilitated through point to point connection between servers. The advantage of building chat application with MERN is the fact that one can benefit from its enhanced flexibility. One can basically avoid any syntax error or any confusion by just coding in one programming language, Java script. While there are many, MEAN and MERN are just two of the popular stack that have evolved out of Java script. Both stacks are made up of open source components and offer an end-to-end framework for building comprehensive web apps that enable browsers to connect with databases. The common theme between the two is Java script and this is also the key benefit of using either stack. In order to understand MERN stack, we need to understand the four components that make up the MERN stack, namely- Mongo DB, Express.js, React and Node.js.

Keywords: *MEAN,MERN,MongoDB,Express.js,React.js, Node.js,Javascript.*

I. INTRODUCTION

With the growth and 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 serves the society as a whole. There are very few applications that facilitate communication within organizations such as institutes, industries, companies etc. that limit the number of users and keeps 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 paper proposes a chat driven online framework that empowers continuous cooperation between clients in a shared environment.

This chatting application will be built as a web-based and mobile application so as to provide the users with flexibility. This chatting application is aimed for the users in organizations and institutions with their very own servers so as to provide privacy for the users. But apart from the specifics usage of the application, this can also be put to generic usage so as to extend the services to the general public. This application is developed and built using HTML, CSS, Java script, React.js, Node.js, Express and Mongo DB. HTML is the standard mark-up language for documents designed to be built in a web browser. CSS is a computer language for laying out and structuring web page. Java script is an object oriented computer programming language commonly used to create interactive effects within web browsers. Mongo DB makes use of records which are made up of documents that contain a data structure composed of field and value pairs. Documents are the basic unit of data in Mongo DB. Express does is that it enables you to easily create web applications by providing a slightly simpler interface for creating your request endpoints, handling cookies, etc. than vanilla Node. As this project is being aimed at for mobile applications, React.JS library is put to use. 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. Node.js is an open-source and cross-platform JavaScript runtime environment. Node.js with Express.js can also be used to create classic web applications on the server-side.

Scope:

The scope of the project should be broken-down and the system should be declared before advancing further. The scope are as follows:

1. The design and construction of this application is aimed at building a web-based application and mobile application.
2. This system is developed using React.js and Node.js.
3. The database of this system implemented using MongoDB.

II. TECHNOLOGY USED

HTML, CSS and Java script

1. Mongo DB: Cross-platform Document-Oriented Database
2. Express: Back-End Framework
3. React: Front-End Library
4. Node.js: JS Runtime Environment

HTML: The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

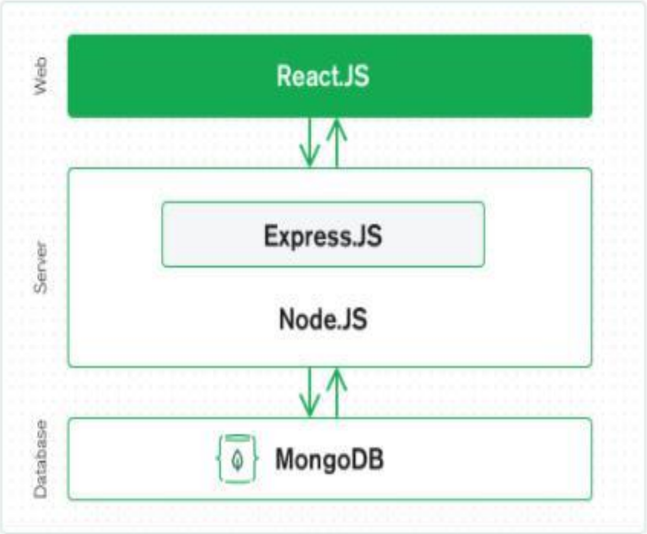
CSS: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Java script: JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

React.js Front End: The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

Express.js and Node.js Server Tier: The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, un opinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

Mongo DB Database Tier: If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you’re going to want a database that’s just as easy to work with as React, Express, and Node.



III. SYSTEM DESIGN

Firstly, we will be doing the requirement and analysis part of the development cycle, in which we will find out our project's final goals and what will get in the long run. Then, we will move onto the designing phase of the cycle which will be deciding the designing and look of it. Then we'll deep dive into the implementation part which consists of frontend and back-end development.

We'll first start from the back-end of the project with the help of the Java script and try to build the minimum valued product first which will set the base of our working project and then we will add other backlogs of the product and try to develop in their sprint. If a few backlogs will not be completed in their mentioned sprint, then we'll add it to the next sprint and try to finish it in that one.

During the development of the back-end of the product, we will be using only the React JS for the front-end which will be showing the info on the browser. Database part will be developed parallelly with the back-end development and Mongo DB will be used for the database.

If the back-end of the project gets complicated, we will move on to the front-end development fully. It will be developed by HTML, CSS , JavaScript and React js.

When this all gets done, we will run some tests in the testing phase of the project and then will manage the documentation which we've maintained during the whole implementation and development phase.

The steps are as follows:

1. While accessing the site, main page will pop up first, with options for existing user login and new user signup. If User details are already present in the database, user details will be asked for logging in the login page. If the signup option is selected then it will redirect to a new user register page, where the user details are collected and then stored in database.
2. After logging in to the portal, user must have a contact before starting conversation. So, they can locate them directly from the app. User can then add friends to their friend list, and wait for confirmation of friend's request.
3. 3. Moreover, the chat database must be checked for previous chat history and if present, chat history is displayed. At last, chat will be possible between the users.

The user interface design is explained in the process below:

- On opening the application, the user is first taken to the home page that requires user credentials to access the main page. This can be obtained by creating an account for the new users and the old users can continue with the previous accounts.
- On successful login, the user will be directed to the home page or the chat room where the user is displayed with all the contacts. Chat room of specific contacts can be accessed by just clicking on the contact name being displayed.
- The data received by the user through any of the user's contact will be displayed in the chat room. To send the data, the user will have to enter the chat room of a specific contact to send the data.

IV. OBJECTIVE

Our goal is to build a web chat that holds a single chat room. Any user can connect to it from an open window/tab, is able to upload an image to use as avatar during the chat, each connected user will see instantly any message sent, and new connected users will see the last 10 messages sent.

Mongo DB makes use of records which are made up of documents that contain a data structure composed of field and value pairs. Documents are the basic unit of data in Mongo DB.

Express does is that it enables you to easily create web applications by providing a slightly simpler interface for creating your request endpoints, handling cookies, etc. than vanilla Node.

React is an open-source, component-based JavaScript library used to create quick and interactive interfaces or UI components for users for web and mobile-based applications. It is a declarative and highly accurate library that provides reusable code, which enhances the efficiency and flexibility of single-page applications.

Node.js is an open-source and cross-platform JavaScript runtime environment. Node.js with Express.js can also be used to create classic web applications on the server-side.

V. CONCLUSION

As a conclusion, I can say that this undergoing project is giving great experience. Thanks to this project, I am acquiring deeper knowledge concerning my technical skills but I also personally benefited. Currently MERN stack is a common technologies of web applications and chat applications, and one of the most popular technology for development used by developers worldwide. If we surf internet we can see millions of websites, applications and games built with MERN and MEAN stack. I am learning to live in a different environment from the one I am used to. Indeed, I am growing more independent in work and also in everyday life, realizing that I could do more things than I thought like learning new things by myself.

There are huge opportunities available for the students who want to work in this field. Many private and public organizations hire web designer and app designer for their online work and development. With the rapid advent of online industry, the demand of web development and app development professionals is increasing and this has created a huge job opportunity for the aspirants in the upcoming days. Also an experienced person in this field can also work as a freelancer; there are many online companies which provide online projects to the individuals.

VI. FUTURE SCOPES

Especially if I don't have any experience. Finding work can be a real challenge. A successful learning can help me turn an experience into a career opportunity.

- To be work IT company.
- Can work as a Software Engineer.
- Can work as a Web Designer.
- Can work as a Web Developer.
- Can work as a QA Tester.

References

1. *Shubham Patil, Saurav Daware, Ameya Bhagat, Prof. Jayant Sawarkar, Department of Computer Engineering, Datta Meghe college of Engineering, Airoli, Navi Mumbai, Maharashtra, India.*
2. *Sourabh Mahadev Malewade, Archana Ekbote , Information Technology Department VCET, Vasai, Palghar, India.*
3. *Akhilesh sarjit M S, Srivishak V, Shiddarth S, Sarvana Kumar P, Preethi D, Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Erode, Tamil Nadu, India.*
4. *Tanya agarwal, Swapnil Upadhyay, Shreeansh Srivastava, Rohit Sharma , Poornima college of Engineering, India.*
5. *Nithin Katla, Gautham Kumar M, Pidugu Rohith Raj, Dr. S Shitharth , Dept. of CSE, Vardhaman college of Eng. , Hyderabad.*
6. *Noor Sabah , Jamal M , Kadhim and Ban N. Dhannoon, Department of computer science , Al-Nahrain Univerversity.*
7. *Sumit Kumar Mishra, Ankit Kumar, Ankush Kaundal, Upasna Joshi, Department of Computer Science and Engineering, Delhi Technical Campus, New Delhi, India*
8. *Suman Kharel, Jukka Jauhiainen , Degree Programme in Information Technology, Oulu University of Applied Sciences*
9. *Bishal Kumar, Kumar Sagar, M Arvindhan, Ankit kr. Tiwari, Galgotias University Greater, Noida, India.*
10. *Lavish Mangal, Pushpendre Pratap Singh, Keshav Gupt, Jai Kumar, Saijal Gupta, Department of Information Technology, Dr. Akhilesh Das Gupta Institute of Technology and Management, New Delhi, India.*