
DEV THREADS

Prof. Archana Senapati*¹, T.Y.M Sai*², Himanshu Sahu*³, Sneha Yadav*⁴, Sakshi Singh*⁵

*¹Assistant Professor, Department Of Computer Science, Shri Shankaracharya Technical Campus,
Junwani (Bhilai), India.

*^{2,3,4,5}UG Student, Department Of Computer Science, Shri Shankaracharya Technical Campus,
Junwani (Bhilai), India.

DOI : <https://www.doi.org/10.56726/IRJMETS51977>

ABSTRACT

In today's rapidly evolving technological landscape, effective collaboration among developers is crucial for innovation and problem-solving. Developer creativity and hassle-fixing rely heavily on effective collaboration. So one can encourage understanding sharing and teamwork among developers this paper provides DevThreads a web software constructed with the MERN stack. Functions like creating post, real-time chat, profanity filtering, special markdowns that support code and simple CRUD operations on posts are all a part of DevThreads which ambitions to improve developer collaboration. Using a complete approach this look at explores the implementation of DevThreads and their effect on developer productivity and engagement. The outcomes spotlight DevThreads capability as a useful platform for the developer network by way of demonstrating the enhancements in collaboration and increasing efficiency.

Keywords: MERN STACK, Mongo DB, Express JS, React JS, Node JS Platform, JWT Tokens.

I. INTRODUCTION

This document covers the journey of a full-stack development of a project from a frontend to backend using MERN stack. Inside the speedy-paced global of software program development, collaboration among builders is important for fulfilment but, traditional verbal exchange channels inclusive of email and forums regularly lack the actual-time interaction and capability required for powerful collaboration and also to increase efficiency. To cope with this challenge, we built DevThreads, a web application in particular designed to enhance developer collaboration and efficiency. Constructed using the MERN stack (Mongo Database, express javascript framework, React javascript framework, Node javascript framework), DevThreads contains a number of features to facilitate exchange of text via personal message and expertise in sharing code snippets, and network engagement among developers. On this paper, we speak the incentive in the back of DevThreads, its key features and the method hired to evaluate its impact on developer collaboration and efficiency. The existing systems such as Stackoverflow has many drawbacks such as users cant chat privately on the platform. So the main goal of this project was to eliminate the drawbacks of already existing systems such as Stackoverflow and create a userfriendly platform where developers can connect people around the world. The security of DevThreads includes JWT authentication. The JSON web token is commonly called as JWT. Json web tokens are lightweight alphanumeric threads as the throughput time to process these is very less as they are self contained. Json web tokens ease the task of authentication and enhances the security of the website. The chat applications are also being used for carrying out some unethical practices like online harassment so there is a need to filter the abusive words and to curb this issue. DevThreads makes use of javascript methods to overcome this problem. DevThreads also aims to stop spamming attempts by users by implementing a cooldown timer to prevent users from repeatedly creating multiple posts and replies. Moreover DevThreads also supports searching by using keywords which help users to navigate to their interested post.

II. METHODOLOGY

Dev Threads is a web application made using MERN (MONGODB, Express js. React.js, Nodejs) stack which aims to provide a collaborative platform for programmers. The technology used for developing this platform are

- React.js is used to create the user interface of the application.
- Express.js is used to build the backend RESTFUL API endpoint,
- Material ui is used to in the front for the ui designing
- MongoDB is used to store application data that ensures compatibility with the operating system's server.

While developing DevThreads, The primary method for validating correctness and ensuring functionality involved the user of JSON Web Tokens (JWT).JWT provide a secure and efficient ways of verifying the legitimacy of users and approving their access to different features within the application.

Dev Threads produce interactions using its integrated messaging system, where users can communicate privately with each other. It encourages deepen collaboration. Networking with the community and empowers peer developer to learn and to connect in a spirited environment of sharing knowledge.

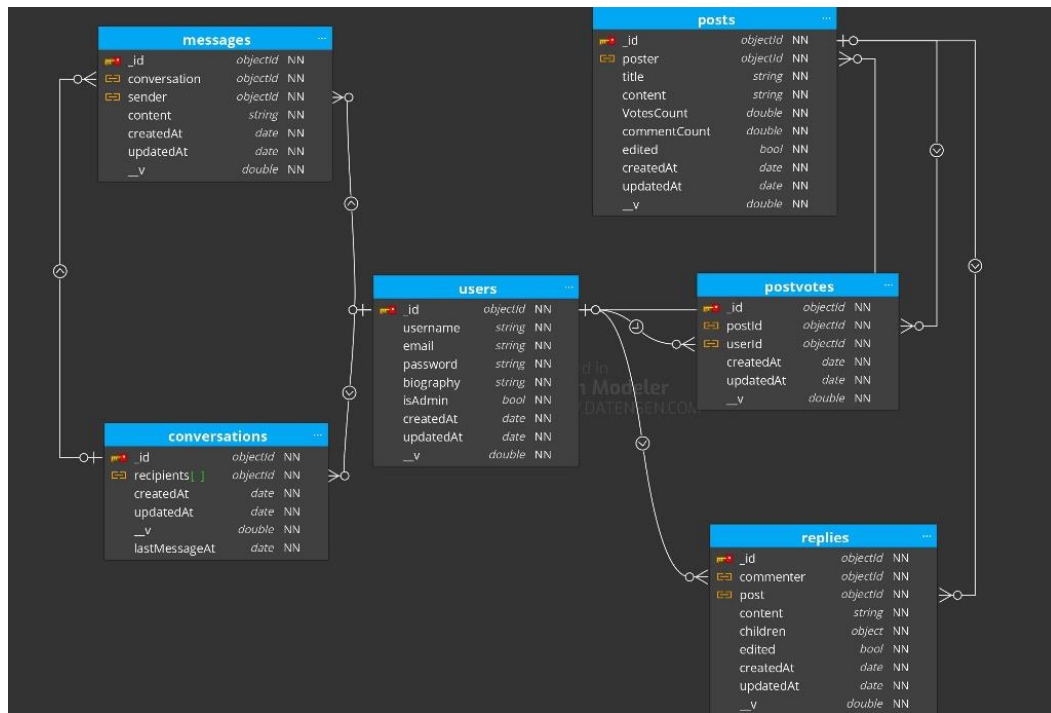


Fig 1: Data model of the project

The image depicts the data model and data flow of the project this includes –

1. Messages Entity:

- a) Contains essential attributes like `_id`, `communique`, `sender`, `content`, `createdAt`, and `updatedAt`.
- b) These help effective communication within the platform for developers.

2. Posts Entity:

- a) Includes details such as `_id`, `poster`, `name`, `content`, `Votes Count`, `comment Count`, `edited`, `createdAt`, and `updatedAt`.
- b) This provides a structure for users to posts their coding, queries, feedbacks, solutions.

3. User:

- a) Holds attributes like `_id`, `username`, `email`, `password`, `biography`, `isAdmin`, `createdAt`, and `updatedAt`.
- b) Crucial for user customization and authentication.

4. Conversations:

- a. Contains `_id`, `recipients`, `createdAt`, `updatedAt`, `double`, and `lastMessageAt`.
- b. Provides a proper conversation method for developer to connect with peers.

5. Replies:

- a) Nested within posts, introduces attributes
- b) like `_id`, `commenter`, `post`, `content`, `children`, `edited`, `createdAt`, `updatedAt`, and `v`.
- c) This contains nested replies which empowers the code feedback and better approaches.

6. Post votes :

- a) It includes `user_id`, `postvotes` `createdAt`, `updatedAt`
- b) This helps developers to rate the post the more the votes the better the post.

JWT authentication: Java web tokens (JWT) plays vital role in authentication of the user, while login it ensure the user is valid ,Let's see how Jwt works

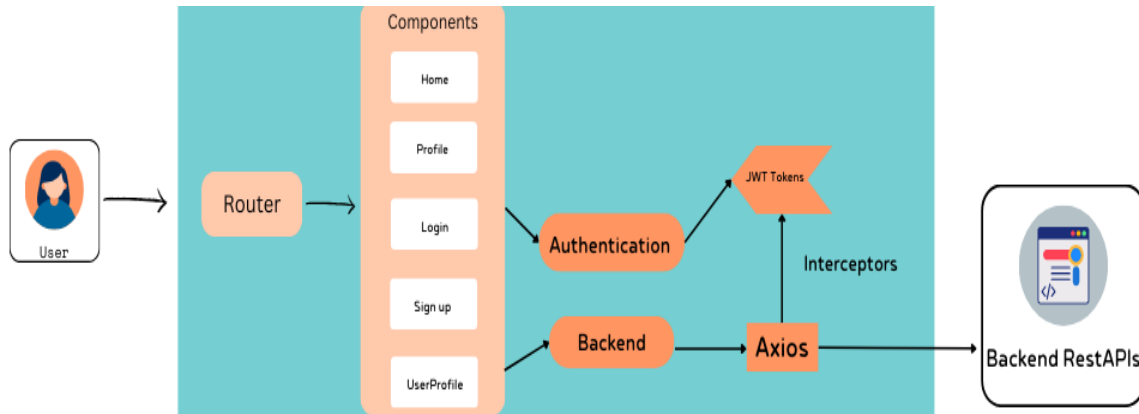


Fig 2: JWT and Axios authentication

The image represents JWT-Axios authentication in the project. First, the- user enters their login details into a form on the website. The server checks if the details are correct. If so, it creates a JSON Web Token (JWT) and sends it back to the browser.

The browser securely stores the JWT. For future API requests, Axios, an HTTP client tool, helps the browser communicate with the server. The requests include the JWT in the header for authorization. When getting a request, the server verifies the JWT and checks the user's permissions before processing. If authentication succeeds and the user has approval, the server sends a response.

Architecture

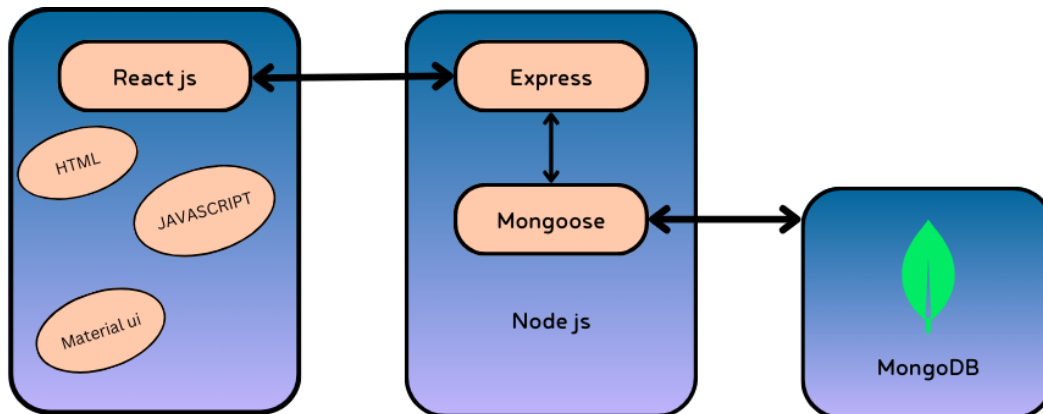


Fig 3: Front end and backend interaction

Web or front-end tier: React.js primarily manages this tier, which is the highest level of the MERN stack. It is among the most widely used front-end JavaScript libraries available for free that are used to create web apps. It is well known for producing client-side dynamic applications. With React, you can use single components to create complex interfaces. Additionally, it links such intricate interfaces to the backend server's data. Web apps and mobile applications (React Native) are made with React. React saves a lot of time and provides several advantages by facilitating the reusability and easy support of code. With it, users may easily construct massive web applications that update the page's data without requiring a page reload.

Server: Server is also known as the middle tier, is the logical step below the top layer and is mostly managed by Express.js and Node.js, two elements of the MERN stack. Because Express.js, which runs inside the Node.js server, maintains the server-side framework, these two components handle it concurrently. One of the most popular JavaScript frameworks for backend development is Express.js. It makes it considerably simpler and easier for developers to put up reliable web servers and APIs. It also enhances Node.js HTTP objects with useful features. However, Node.js is a very significant component in and of itself. It is a cross-platform runtime environment that is open-source that allows JavaScript code to be executed outside of a browser.

Database: It is one of the most critical ranges of the MERN Stack and is particularly handled by way of MongoDB; the main role of a database is to store all the statistics related to your application, as an example - content, statistics, records, user profiles, votes and so on. MongoDB is not primarily based on the desk-like relational database structure. On the other hand, it gives an altogether special mechanism for the retrieval and storage of records. Mongo DB is the most popular NoSQL (NoSQL or Non Structured Query Language) database, an open-supply record-oriented database. The term 'NoSQL' typically means a non-relational database that does not require a fixed schema or proper relational tables to store the necessary statistics in it. MongoDB shops the information in a one of a kind layout aside from the relational tables, consisting of rows and columns.

III. RESULTS

As a result the proposed system will tends to providing a platform for developers to collaborate, share ideas, and work together on code, your project can foster a sense of community and teamwork

The following are the screenshot of our application "DevThreads" -

1. SignUp page

The figure represents the interface of 'Sign Up Page' of our application. As, the main motivation of our application is to provide an easy and interactive interface to developers. Here, developers could Sign Up easily with the information and relevant data of the user such as name, valid email id and password .The screenshot highlights the design element, input field and features of the sign up page.

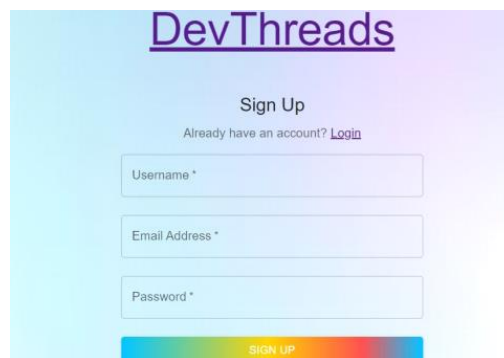


Fig 4: SignUp page

2. Login page

After the successful Sign in of the user. The registered user can directly login by adding the desired credentials which are needed for the authentication of the user. The input fields include email address and password. This will allow the user to gain access to their profile in the application and to interact with other functionalities.

Figure 5 represents the 'Login' page of the application.

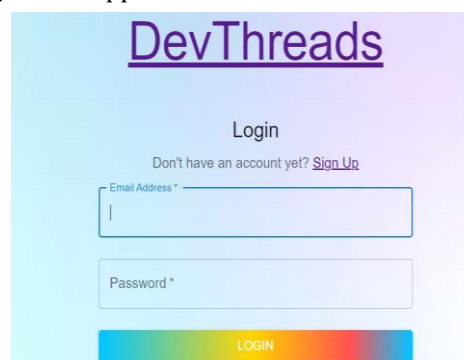


Fig 5: Login page

3. User Profile

Figure 6 represents the User Profile of the application . The user can reach the user profile by clicking on the user profile symbol which is the first icon from the top right corner. The user profile will have different sections which include Bio of the user which could be edited anytime by the user . A profile representing the user image

,number of posts added and the total votes recieved by the other user making it very interactive. The lower section of the profile represents the people who are connected with the user.The user can see all of thier posts and can also sort them accprding to thier relevance. Each post will contain three section that is POSTS ,VOTES and COMMENT recieved.

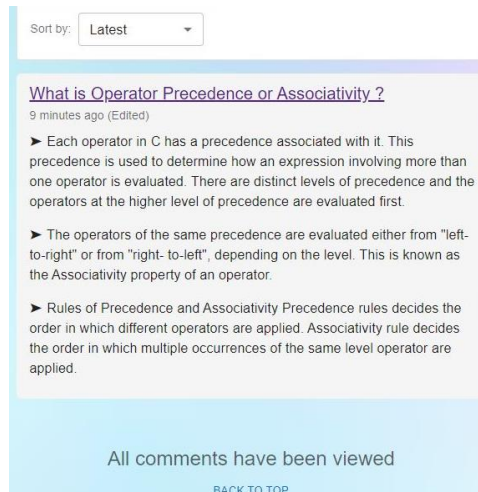


Fig 6: User page

4. Personal Chat View :

Figure 7 represents the Personal Chat View of the user. The user can anytime access thier Chat with the other user. Personal Chat view will consist of two section .first if the List of the chats with other user .Second is the Chat of the selected user and the user accessing it. The user can clear chat , send messages ,read messages from the other user, can see thier profile status. It demonstrates that how a user can easily enjoy the conversation feature within the application.

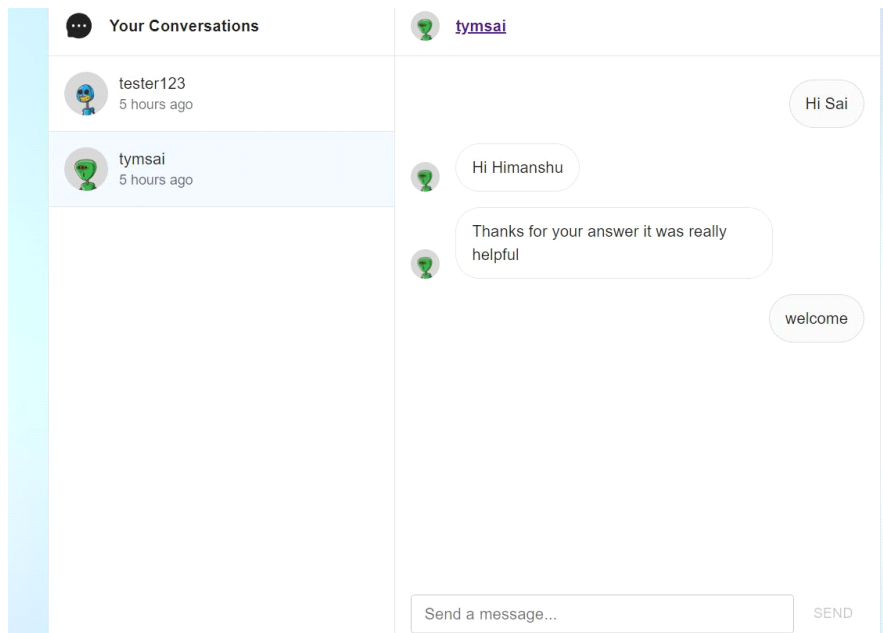


Fig 7: Personal chat view

5. Real-Time Chat

Figure 8 represents the Real-Time Chat between two users, The screenshot captures the real-time Conversation between Himanshu and T.Y.M SAI within the application. The figure represents the vital elements which include exchanged messages ,user avatar, post search option along with home, message view and profile options and communication between users with a good flow could be easily achieved by the application.

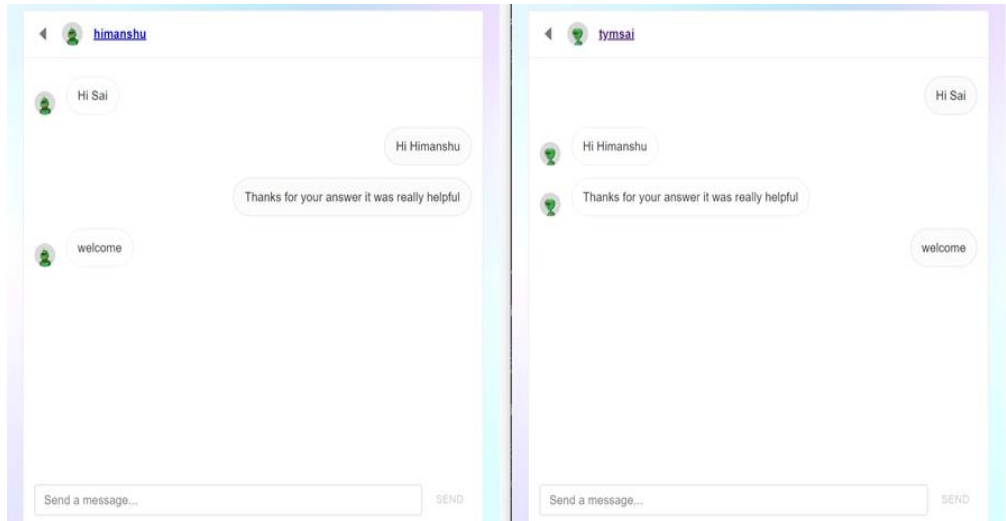


Fig 8: Real time chat

6. Profanity Filtering

Figure 9 represents the profanity filtering feature of the application which is a unique feature. The application automatically detects abusive words and replace those words with * symbol. The user also cannot post posts/queries at the same time. Along with that one cannot spam .All these features provide an extra edge to the user.

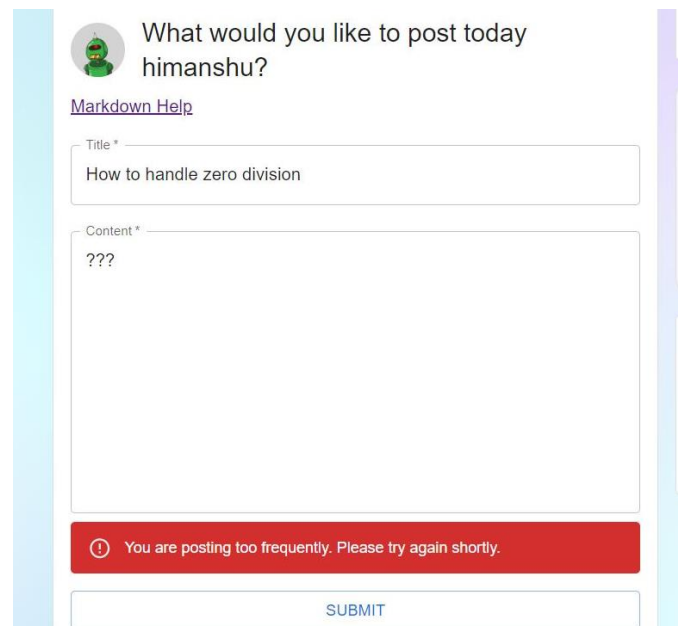


Fig 9: Profanity

IV. CONCLUSION

All in all Dev Threads is a brilliant MERN stack application and its improvement strategy, well established in present day age and client driven design, and aptitude sharing inside the programming domain.

The MERN stack, with its strong parts - MongoDB, Express.js, React.js, and Node.js, holds up the spine of Dev Threads. MongoDB's NoSQL shape ensures adaptability in realities the executives, on the indistinguishable time as Express. Js grants the coming of a strong backend, pondering the consistent coordination of abilities that frame board insights. Node.js, helps in server-component runtime, guarantees fast and non-closing off execution, adding to the general responsiveness of the programming program application.

The frontend, worked with React.js, delivers a natural and interactive interface single page application. Dev Threads isn't just a Q & A stage; it's miles an environmental factors intended to motivate support, cooperation,

and information change. Verification instruments and the blend of JSON Web Tokens (JWT) highlight the privateness between users. It's not just a question-and-answer site; it's a vibrant hub where developers encourage and support one another.

ACKNOWLEDGMENT

We extend our sincere gratitude to all our mentors and faculty members who has contributed to this research on DevThreads.

Special thanks to our project guide Professor Archana Senapati for her invaluable guidance and crucial support throughout the research work. We acknowledge Shri Shankaracharya Technical Campus for providing all the necessary resources and useful facilities. Additionally, we thank our friends and family for their unwavering encouragement.

Thank you to everyone who played a role in the completion of this research paper.

V. REFERENCES

- [1] Vasan Subramanian, and Springerlink (Online Service. Pro MERN Stack : Full Stack Web App development with Mongo, Express, React, and Node. Berkeley, Ca, Apress, 2019.
- [2] Wilson, Eddy. MERN QUICK START, 2018.
- [3] D. T. Andrew Hunt, The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley.
- [4] Professional, first ed., october 1999.
- [5] Faryn, F., (2020) "Mongodb manual 3.4."
- [6] Cadenhead, Tyson. Socket.IO Cookbook. Packt Publishing Ltd, 15 Oct. 2015.
- [7] Boduch, Adam. React Material-UI Cookbook. Packt Publishing Ltd, 30 Mar. 2019.
- [8] Baiskar, Yogesh. "MERN: A Full-Stack Development." International Journal for Research in Applied Science and Engineering Technology, vol. 10, no. 1, 31 Jan. 2022, pp. 1029–1035, <https://doi.org/10.22214/ijraset.2022.39982>.
- [9] Abhishek Bedare, et al. "Lifeline Messenger Real Time Chat Application: Using MERN Stack." Social Science Research Network, 1 Jan. 2023, [papers.ssrn.com /sol3/ papers.cfm? abstract_id=4447248](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4447248), <https://doi.org/10.2139/ssrn.4447248>.
- [10] Desai, Krutika, and Jinan Fiaidhi. "Developing a Social Platform Using MERN Stack." INDIGO (University of Illinois at Chicago), 12 Dec. 2022.
- [11] <https://doi.org/10.36227/techrxiv.21699764.v1>