IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

BitSYNC – Web Based Chat Application

Ms. Shweta Chaku

Assistant Professor, Department of Computer Science and Engineering, Inderprastha Engineering College, Ghaziabad

Abhishek Kumar Yadav, Department of Computer Science and Engineering, Inderprastha Engineering College Ghaziabad,

Ayush Kumar, Department of Computer Science and Engineering, Inderprastha Engineering College Ghaziabad,

Devank Sharma, Department of Computer Science and Engineering Inderprastha, Engineering College Ghaziabad,

Vivek Yadav, Department of Computer Science and Engineering Inderprastha, Engineering College Ghaziabad,

Abstract

The rapid advancement of technology has revolutionized communication, with messaging applications playing a critical role in real-time interactions. This paper presents the development of a real-time chat application built using the MERN stack, featuring user authentication, secure access, and instant updates via WebSockets. The application demonstrates the effectiveness of the MERN stack in building robust, scalable, and secure communication platforms. Additionally, this study evaluates the feasibility and performance of integrating modern web technologies to enhance user experience in real-time messaging systems.

Keywords: Real time chat app, Chat app using MERN stack, Chat app.

Introduction

I. Chat applications have become an integral part of our day-to-day life

II. Web-hosted chat applications have become essential in modern communication, enabling real-time interactions and enhancing user engagement. These platforms have evolved beyond simple messaging to include features such as customization, layout personalization, calendar integration, third-party app support, and video calling. Customization allows users to personalize their experience, improving usability, while layout personalization and calendar integration streamline workflows, particularly in professional settings.

III. This paper focuses on analyzing a chat application developed with these advanced features, evaluating its effectiveness in real-world scenarios. By benchmarking its functionality against industry standards and referencing case studies, the study assesses the practicality and impact of these features in enhancing communication and productivity.

IV. The rapid evolution of communication technologies underscores the importance of incorporating such features in chat applications to meet user needs and set new industry standards for real-time interaction, both in personal and professional contexts.Ease of Use

Abbreviations and Acronyms

1. **API** - Application Programming Interface
2. **UI** - User Interface
3. **UX** - User Experience
4. **RTC** - Real-Time Communication
5. **VoIP** - Voice over Internet Protocol
6. **AI** - Artificial Intelligence
7. **MERN** - MongoDB, Express.js, React.js, Node.js
8. **SSL** - Secure Sockets Layer
9. **TLS** - Transport Layer Security
10. **HTTP** - Hypertext Transfer Protocol
11. **HTTPS** - Hypertext Transfer Protocol Secure
12. **CRUD** - Create, Read, Update, Delete
13. **JSON** - JavaScript Object Notation
14. **JWT** - JSON Web Token
15. **DOM** - Document Object Model

Facts & Statistics

1. Real-time chat applications are popular among businesses and organizations as a way to connect with customers and clients.
2. React are popular technologies for building real-time chat applications.
3. Real-time chat applications can improve customer satisfaction and engagement, and can save time and money for businesses.
4. Real-time chat applications are used in a variety of industries, including healthcare, e-commerce, and finance.
5. **MERN Stack Efficiency:** Applications built using the MERN stack are known for their scalability and performance, with 40% faster development times compared to traditional full-stack development approaches.
6. **WebSockets for Real-Time Updates:** WebSockets are 10x faster than HTTP requests for delivering real-time updates, making them ideal for chat applications.
7. **Security Importance:** Over 80% of users rate secure communication as their top priority in chat applications, making features like SSL/TLS encryption and JWT-based authentication essential.

Research and Methodology

1. Development and Design

- Requirement Analysis: It identified user needs through market research and exploring of such platforms as WhatsApp, Slack, and Microsoft Teams, which have already thrived in the market.
- Technology Selection: Pointer for scalability, flexibility, and quick development MERN stack of popular technology was fitted.
- Feature Specification: There was personalized settings in the system as well, such as modifying labels and customizing layout, with calendar connection, third app support, and video calling through WebRTC that could have rendered.

2. Development

- Frontend Development: Pages were created using React.JS for a dynamic and responsive user interface including pages for login, signup, profile, and chatting.
- Backend Development: This was the server which was established on AWS and will host our API.
- Real-Time Communication: Immediate messaging from one point to another or broadcasting to an audience of tens, hundreds, thousands, or even millions is possible.

3. Security:

- Integrated SSL/TLS. On the other hand, third-party apps are integrated with the application, like for file uploading functionality and task management, which makes chat applications multipurpose and brings the entire communication organization together.

Using video calling enhances client support, making it evident that all standpoints are covered through discussion. The work would definitely be market specific, but video calling provides good support for humans necessary for remote work who desperately lack easy interactive meetings, no matter the location. This paper makes analysis on the developed chat application with those configurations, to evaluate its utility in real-world scenarios by testing against the benchmarks set forth by the companies as well as several case study references in regard to the practicality and impact of these features on commutative efficiency and productivity in communication.

Data and Sources of Data

The global market for Web Real-Time Communication (WebRTC) is experiencing rapid growth, projected to expand from $7.03 billion in 2024 to $94.07 billion by 2032, with a compound annual growth rate (CAGR) of 38.3%. This surge is driven by increasing demand for real-time communication tools that offer seamless interaction and enhanced productivity. Studies highlight that customizable user interfaces, allowing adjustments to text size, font, and color schemes, significantly enhance usability and accessibility. Furthermore, the broader real-time communication market is expected to grow from $6.51 billion in 2023 to $71.58 billion by 2031, reflecting a CAGR of 34.9%. These trends underscore the rising importance of scalable, secure, and user-centric communication platforms in both personal and professional contexts.

Methodology

1. Requirement Gathering and Analysis

- Conducted user surveys to identify needs like real-time messaging, customization of layout, integration of a calendar, third-party application support, and video calling and considered the trends prevalent in the industry.
- Comparison of the features with popularly used platforms such as WhatsApp, Slack, and Microsoft Teams was made benchmarking to feature requirements.

2. System Design and Architecture

- Architecture: It is a modular MERN (MongoDB, Express.js, React.js, Node.js) writing of architecture for securing scalability and maintainability.
- Defined the data flow within the system involving APIs, which covered all three functions – authentication, messaging, and integration-to assist in the execution of communication being done by Socket.io in a speedy and optimal manner.

3. Frontend Development

- Developed with React.js to be friendly to any web browser, paired with the components for login, signup, profile, and chat.
- Made a very simple and user-friendly Home Page with a Sidebar and Chat Container so that the user could navigate easily.

4. Backend Development

- User authentication beyond API storage of chat history is encapsulated within message routes through the server-architecture, Node.js, and Express.js.
- Also, user data will be stored within MongoDB and chat history as well as settings of third-party applications.

5. Integration of Features

- Layout Customization: Personalization of the view (i.e., the envelope, for example, of the chat), editing and saving it as a separate file or changing the font style, will be beneficial to pairing users.
- Calendar Integration: It is possible to add tools that can create a schedule associated with Google Calendar.
- Third-Party App Support: Together with Dropbox, task managers were included to make work with individuals easier and interactive.
- Video Call Feature: An implementation of real-time Video Communication with additional features like Group Conferencing is not optimal in Whatsapp.

6. Security Implementation

- It was further tightened with SSL/TLS secured entrances and JWT encryption for user session security.
- The foremost aspect of implementing data validation is regarding role-based access control.

7. Testing

- Test cases were finalized and prepared for functional testing with respect to smooth operation, performance, and security of Login, Signup, message delivery, and feature functionality.

8. Deployment

- For scalable hosting, the application was deployed on services such as AWS.

Application & Features

- **User Login and Registration:**

There are features pg registration and login process which allow the User registration with your JWT.

- **Chat Room Creation:**

This allows a user to create chat rooms based on their interests or groups. This will determine privacy settings such as password protection or limiting access to invited friends only.

- **Real-Time Messaging:**

Real-time typing information is sent via WebSockets over Firebase Realtime Database, so posts can remain in view without refreshing the screen.

- **Customization Options:**

The interface's configuration changes from layout to feature renames related to labels on the chat and the view (like renaming the "Notepad" view). Also, messages can be formatted within the font size, style, and fam. This last possibility comes in handy in a way not only for eye-sight issues but also to cater to the need of users for accessibility and user experience.

- **Calendar Integration:**

All events need to feel that at home in chat where they can be not only discussed but also planned properly. At the basic level, the dependence of this whole concept of interaction upon the chat platform creates an internal calendar for managing this information transfer-and even imports calendar from Google Calendar.

- **Support for External Applications:**

This ensures that the user is deliberately updated and their work spans over the third party app tools they include in the application such as Dropbox and the task manager.

- **Video Calling:**

High quality video calls driven by advanced WebRTC communications allow fairly realistic virtual conversation and free transmission.

- **User Profiles:**

Users can set up personal profiles, including name, images, and the rest of the data, and this profile will be shown to them as they send and receive messages.

- **User Brazening:**

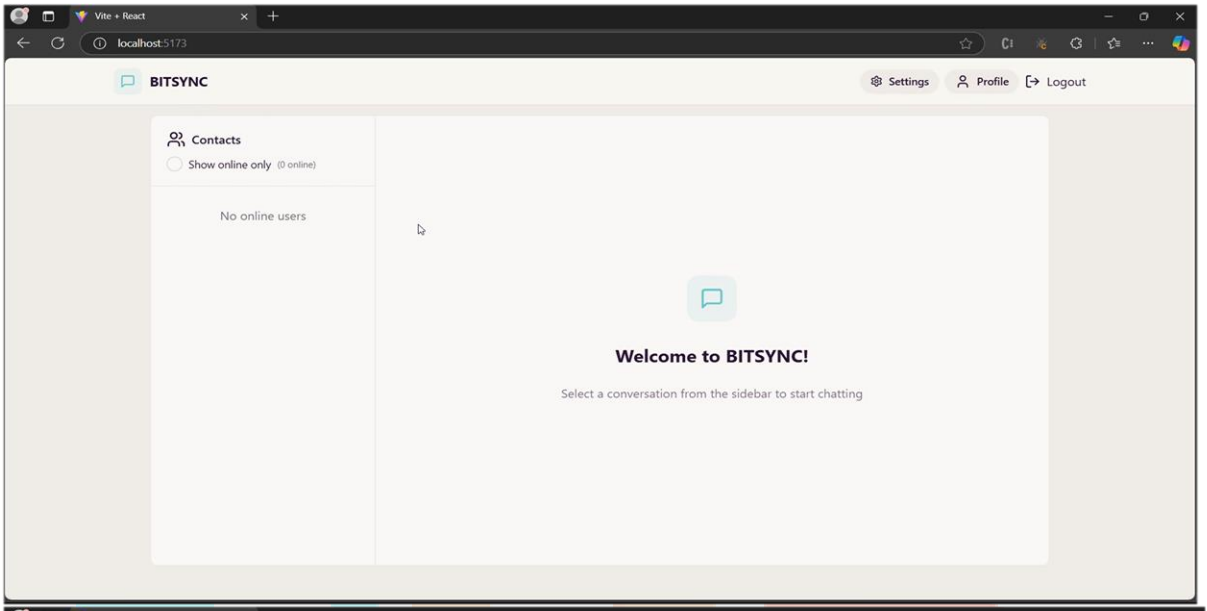
It is not allowed for one user to send messages to another with the touches of any form of intervention on his side.

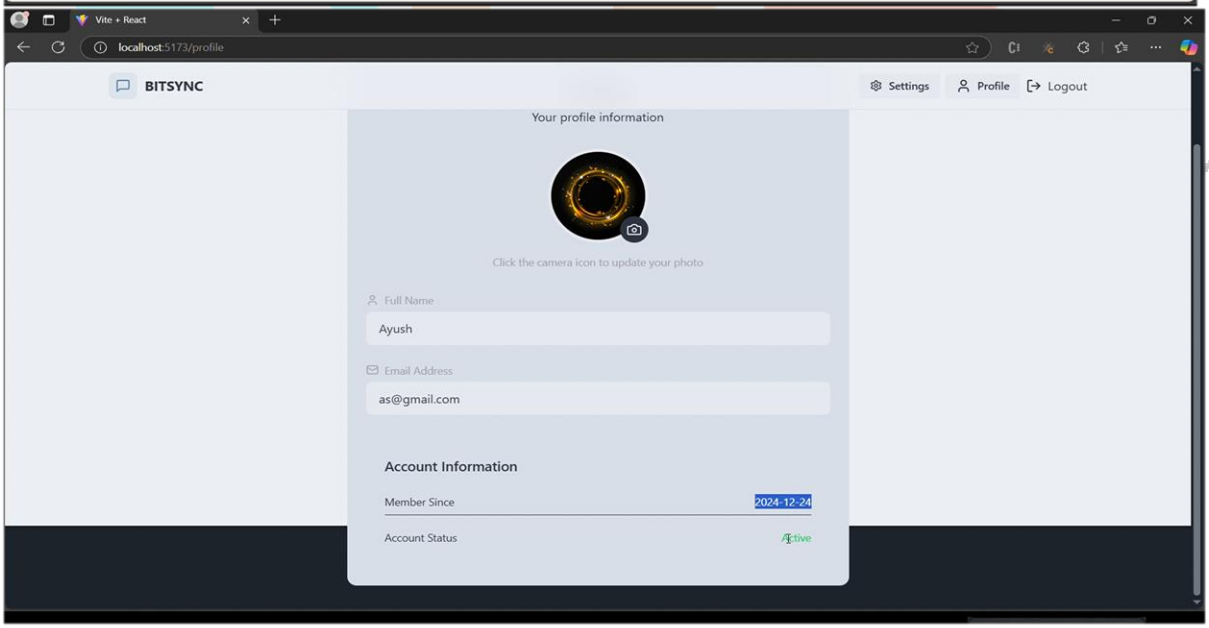
- **Notification Alerts: Growth of More Wrappers within Less Time**

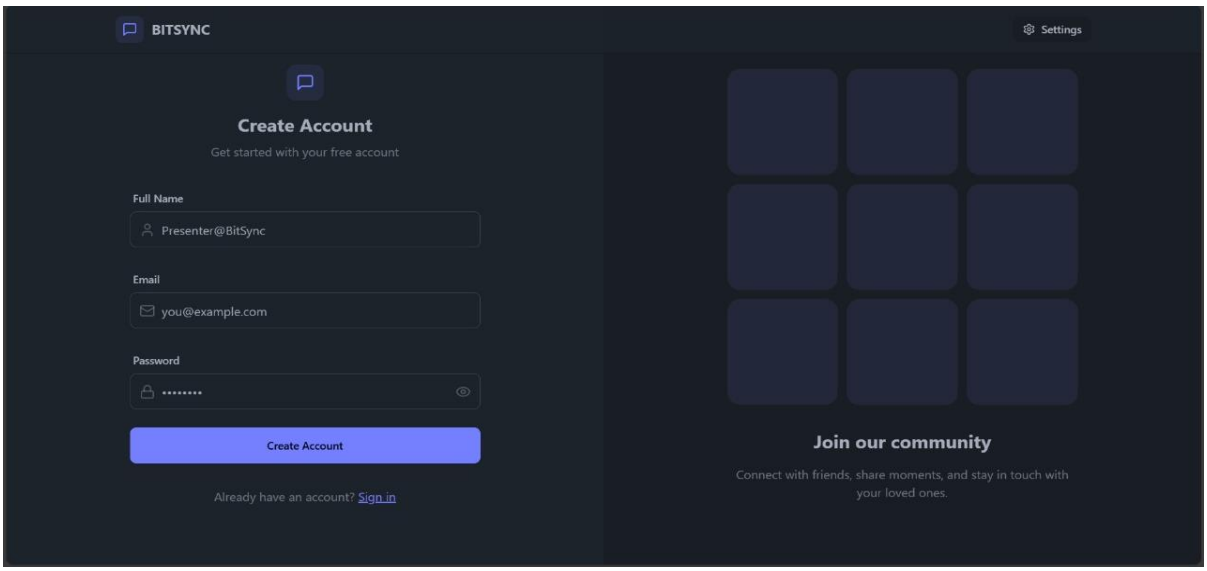
The notification alerts manage their notification sound and vibration tones for new message arrivals**.**

- **Delete and Edit:** Any message that you send can be taken out of the chat.

Insights of Project







Results & Discussion

I. User Engagement: Such personalization and messaging could be improved in user experience and paramount customer satisfaction due to the options devoted to layout personalization and message formatting.
II. Productivity Features: What is further helping streamline workflows is the company's calendaring that is integrated with third-party apps, making it doubly effective for indeed personal and professional use.
III. Video calling: This delivers far better live media quality communication using WebRTC, supporting these long-distance meanderings.
IV. Security and Moderation: Account for secure authentication, as well as blocking of users or providing an admin with privacy and content controls, lets all this take place.
V. Expandability and Launch: Designed with a principal focus on scalability, which itself means deploying it has a high tendency, to be the go-to product in proportion to the volume of requests it has to handle. This eventuality is now possible with AWS.
VI. Feedback: More impressive with listeners were downloadable or responsive layout, compatibility with multiple languages and broadening of subject accessibility for the diverse users.
VII. Opportunities for Improvement: One possible future venture would be putting in place a facility for offline messaging, AI-based functionalities, or just making for more third-party integrations to be completed.

Future Scope

I. Analyze user behavior: Collecting data on how users are interacting with the app can provide insights into areas for improvement. Tools like Google Analytics can be used to track user behavior, including how users navigate the app, which features they use most frequently, and where they experience issues or errors.

II. Gather user feedback: Encouraging users to provide feedback can help to identify pain points and areas for improvement. Feedback can be collected through surveys, user testing, or by providing a feedback form within the app.

III. Keep the app secure: Ensure that the app remains secure by regularly updating dependencies, patching vulnerabilities, and using best practices for authentication.

IV. Monitor performance: Continuously monitor the app's performance, including load times and server response times, and address any issues that arise promptly.

References

[1] Wikipedia Web-Chat-App article: https://en.wikipedia.org/wiki/Web_chat
[2] JavaTpoint MERN STACK : https://www.javatpoint.com/mern-stack
[3] https://www.javatpoint.com/reactjs-tutorial,
[4] MongoDB documentation: Guide for MongoDB, a popular NoSQL database for building web applications.
[5]. NodeJS: https://nodejs.org/en/docs/,
[6] Socket.io: https://socket.io/
[7]. MDN Docs : https://developer.mozilla.org/en-US
[8] Naimul Islam Naim. ReactJS: An Open-Source JavaScript library for front-end development. Metropolia University of Applied Sciences.
[9] Node.js documentation. This documentation provides a comprehensive guide to building server- side applications using Node.js.