

# Emotion-Aware Secure Chat Application Using MERN Stack with Real-Time Sentiment Analysis and One-to-One Messaging

Abhijit<sup>1</sup>, Er. Rohit Kumar<sup>2</sup>, Anchal Singh<sup>3</sup>

<sup>1</sup>Student, B.TECH CSE, SOET, CT University, India

<sup>2,3</sup>Assistant Professor, B.TECH CSE, SOET, CT University, India

**Abstract-** In this paper, the MERN (MongoDB, Express.js, React.js, and Node.js) stack is used to design and implement a modern, smart, and secure web-based chat application. The system includes JWT-based authentication for safe session management and Socket.IO-powered real-time two-way messaging. A key highlight is its support for one-to-one messaging with media sharing capabilities. Another unique feature is the integration of sentiment analysis using Natural Language Processing (NLP) techniques. This allows the system to assess the emotional tone of messages and automatically flag or decorate offensive or rude messages with appropriate emojis, enhancing the emotional intelligence of communication

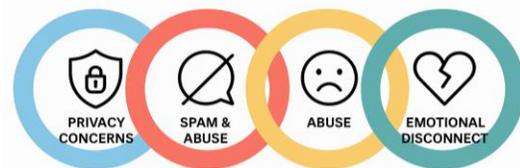
**Keywords-** MERN Stack, Chat Application, JWT Authentication, Sentiment Analysis, Real-Time Messaging, Emotion Recognition, One-to-One Chat, Media Sharing.

## I. INTRODUCTION

The increased dependence on digital communication mediums and tools has altered social and professional relationships and interactions across the globe. Chat apps are popular for facilitation of instant messaging, collaboration, and socialisation. With this popularity, many chat platforms deal with major issues of user security, providing message privacy, while also attempting to reduce unpleasant and negative behaviours such as harassment and negative language. This paper demonstrates an Emotion-Aware Secure Chat Application built on the MERN stack platform, encoder-controlled flexible software development. JWT-based user session authentication are provided to secure user account. The proposed Emotion-Aware Secure Chat Application implemented real-time semantic analysis to automatically detect and respond to offensive or rude messages. Negative semantics are tagged with emojis so our users can be encouraged to

engage in a more balanced variety of healthier and respectful communication context.

## CHALLENGES IN CURRENT CHAT SYSTEMS



## CHALLENGES IN CURRENT CHAT SYSTEMS

[Figure 1: Overview of Challenges in Existing Chat Systems]

## II. LITERATURE REVIEW

Security and emotional analysis from chat messaging platforms are active areas of research. Some papers have looked at securing ways to authenticate, including research using JSON Web Tokens (JWT) [1], [6].

The area of sentiment analysis is useful to understanding an emotional state of the user, dating back from lexicon-based models like VADER, and TextBlob [1], [7], and onwards to deep-learning transformer models such as BERT [11]. Although the approaches to sentiment analysis have been widely adopted and used in social media and the content to collate customer reviews, and provide feedback. The methodologies of sentiment analysis modules into a real-time messaging chat app are not used in a straightforward way.

This paper addresses that gap by integrating secure authentication protocols with real-time sentiment detection and immediate feedback, to encourage constructive communication in chat messaging environments.

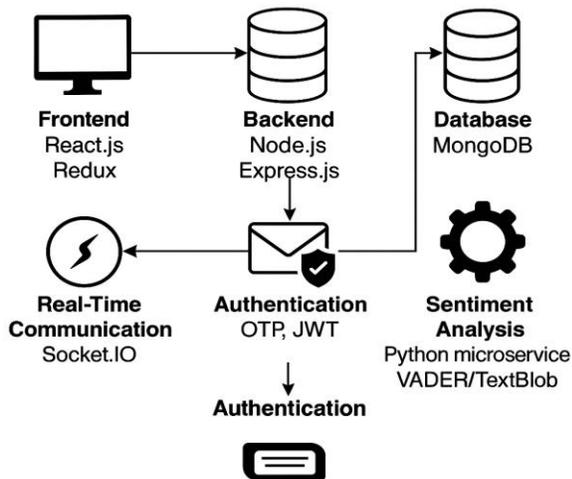
Technique	Type	Accuracy	Processing Speed	Domain Adaptability	Implementation Complexity	Suitable Use Cases
TextBlob	Lexicon-Based	Moderate	High	Low	Low	Lightweight applications, quick prototyping
VADER	Lexicon-Based	Good	Very High	Low	Low	Social media, real-time chat messages
Naive Bayes / SVM	Machine Learning	Good to High	Moderate	High	Moderate	Traditional NLP tasks, domain-specific models
LSTM / CNN	Deep Learning	High	Moderate to Low	High	High	Sentence-level classification, contextual tasks
BERT / RoBERTa	Transformer-Based	Very High	Low	Very High	High	Contextual understanding, fine-grained sentiment
Hybrid (Lexicon + ML)	Hybrid	High	Moderate	High	Moderate	Balanced performance with manageable complexity

[Figure 2: Sentiment Analysis Techniques Comparison]

### III. SYSTEM ARCHITECTURE

The suggested system utilizes a full-stack MERN architecture:

- Frontend: React.js with Redux enables optimal state management and responsive UI experience to support all functions.
- Backend: Node.js with Express.js creates RESTful APIs and manages server-side logic.
- Database: MongoDB will contain all user identity credentials, chat messages, and sentiment score and plot data.
- Real-time messaging: Socket.IO allows for bi-directional real-time messaging.
- Sentiment Analysis: Python microservice that determines sentiment scoring of text messages using VADER/TextBlob.



[Figure 3: System Architecture Diagram]

### IV. MODULES AND IMPLEMENTATION

#### A. User Registration and Authentication

Users register and log in using their email and password. On successful login, a JWT token is issued to manage session authentication. Only authenticated users can access the chat system.

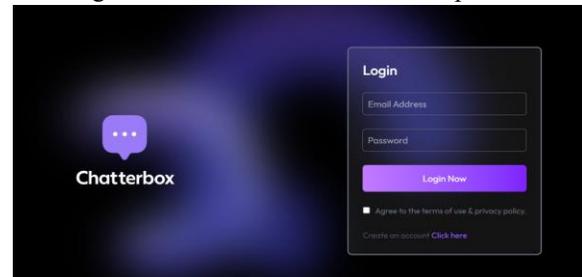
#### B. Dashboard and User Interface

The dashboard features:

- a sidebar listing users that are currently connected and online
- a chat panel in the center which displays conversations and receives messages in real time
- Redux handles all state synchronization to show all incoming/outgoing messages and users' live update on their status

#### C. Login and Authentication Screen

The user's login screen will display secure access by allowing the user to enter their email and password.



[Figure 4: Login Page Interface]

#### C. One-to-One Chat

Each user can initiate a private chat with another user. Messages are sent in real time using Socket.IO and

stored in MongoDB with metadata including sender, receiver, timestamp, and sentiment score.

#### E. Sentiment Analysis and Emoji Feedback

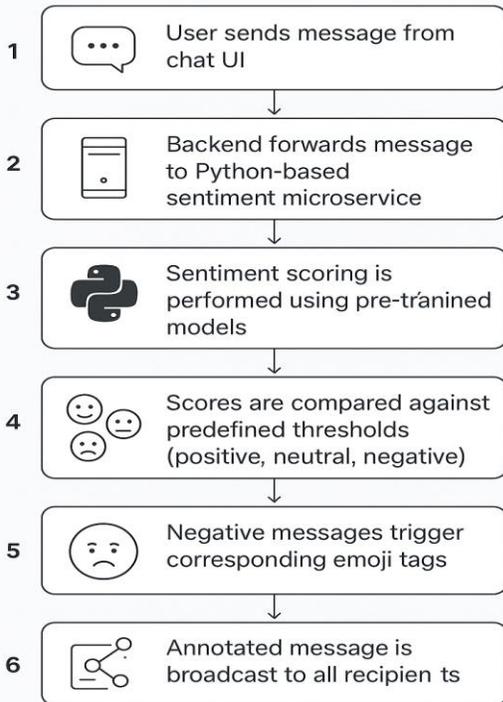
Each outgoing message is forwarded to a Python microservice where VADER or TextBlob analyzes sentiment polarity. Messages detected as negative or offensive are tagged with appropriate emojis (e.g., angry or sad faces) to visually notify participants.

### V. SENTIMENT ANALYSIS WORKFLOW

The workflow consists of the following steps:

1. User sends a message from the chat UI.
2. Backend forwards the message to the Python-based sentiment microservice.
3. Sentiment scoring is performed using pre-trained models.
4. Scores are compared against predefined thresholds (positive, neutral, negative).
5. Negative messages trigger corresponding emoji tags.
6. The annotated message is broadcast to all recipients.

#### Sentiment Analysis Workflow



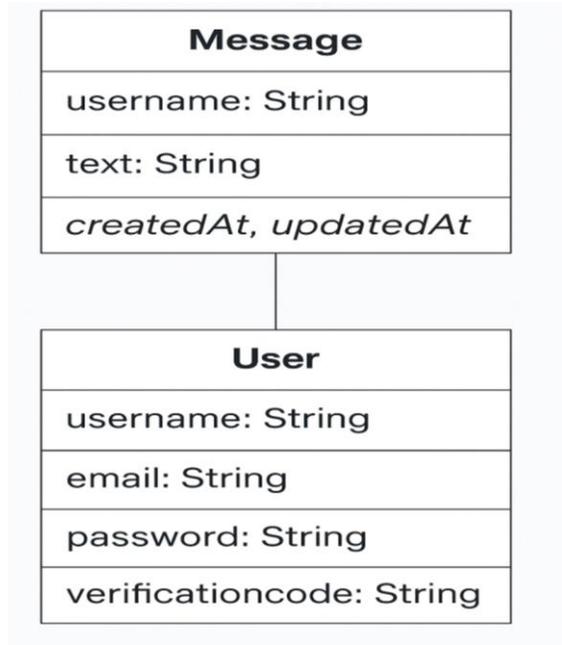
[Figure 5: Sentiment Analysis Workflow Diagram]

### VI. DATABASE SCHEMA

The MongoDB collections are structured as follows:

- Users: {name, email, passwordHash, verified, createdAt}
- Messages: {senderId, content, timestamp, sentimentScore, emojiTag}

Indexes are created to optimize query performance, especially on user email and message timestamps.



[Figure 6: Database Schema Diagram]

### VII. FINDINGS

#### A. Experimental Configuration

A controlled environment simulating multiple users simultaneously was used to test the system. Message sending, authentication, and sentiment tagging were among the test cases.

#### B. Accuracy of Sentiment Tagging

Based on a manually tagged data set for validation, the VADER tagging accuracy was approximately 82%. Feedback resulted in more polite interactions and improved user comprehension of the tone of the conversation.

### VIII. SUMMARY

This work provides a comprehensive solution that enhances chat applications with emotional intelligence and secure user authentication. Scalable growth is

made possible by the MERN stack's development, and the required security is provided by JWT. Sentiment analysis real-time feedback has been demonstrated to improve user engagement and foster more cordial communication.

#### IX. FUTURE WORK

- Use Deep Learning and Transformers to further investigate improvements in sentiment analysis performance.
- Add voice messages and detect sentiment in other languages.
- Develop a mobile app version for easier access.

#### REFERENCE

- [1] C.J. Hutto and E.E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," *Eighth International AAI Conference on Weblogs and Social Media*, 2014.
- [2] Node.js Documentation. [Online]. Available: <https://nodejs.org/>
- [3] ReactJS Documentation. [Online]. Available: <https://reactjs.org/>
- [4] MongoDB Documentation. [Online]. Available: <https://www.mongodb.com/>
- [5] Socket.IO Documentation. [Online]. Available: <https://socket.io/>
- [6] JWT.io. [Online]. Available: <https://jwt.io/>
- [7] TextBlob Documentation. [Online]. Available: <https://textblob.readthedocs.io/en/dev/>
- [8] Express.js Guide. [Online]. Available: <https://expressjs.com/>
- [9] Redux Documentation. [Online]. Available: <https://redux.js.org/>
- [10] Flask Documentation. [Online]. Available: <https://flask.palletsprojects.com/>
- [11] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL*, 2019.
- [12] B. Liu, *Sentiment Analysis and Opinion Mining*, Morgan & Claypool Publishers, 2012.
- [13] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, 2014.
- [14] A. Das and S. Bandyopadhyay, "SentiWordNet for Indian languages," *Asian Federation of Natural Language Processing*, 2011.
- [15] E. Cambria et al., "New Avenues in Opinion Mining and Sentiment Analysis," *IEEE Intelligent Systems*, 2013.
- [16] L. Zhang, S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018.
- [17] L. Liu and Y. Sun, "Research of Web Real-Time Communication Based on Web Socket," *International Conference on Computer Science and Automation Engineering*, 2012.
- [18] M. Ahmed, F. Zaman, and H. Khan, "Real-Time Communication using Web Sockets," *SSRN Electronic Journal*, 2024. [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4721447](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4721447)
- [19] Y. Cui and W. Lin, "Research and Implementation of WebRTC Signaling via WebSocket-based for Real-time Multimedia Communications," *IEEE Information and Automation International Conference*, 2016.
- [20] A. Pandiri, V. Sharma, and N. V. Rao, "Scalable and Secure Real Time Chat Application Development Using MERN Stack and Socket.IO for Enhanced Performance," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 14, no. 3, pp. 29–35, 2024.
- [21] M. Ghasemshirazi and M. Heydarabadi, "Exploring the Attack Surface of WebSocket," *arXiv preprint*, arXiv:2103.05917, 2021. [Online]. Available: <https://arxiv.org/abs/2103.05917>
- [22] A. Sugiyanto, R. Andriani, and F. Surya, "The Development and Evaluation of Web-based Multiplayer Games with Imperfect Information using WebSocket," *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 13, no. 9, pp. 4–17, 2019.
- [23] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, O'Reilly Media, 2009.
- [24] K. Singh and D. Kaur, "Comparative Study of WebSocket and REST Communication Protocols for Real-Time Applications," *International Journal of Computer Applications*, vol. 182, no. 29, pp. 1–5, 2019.

- [25] OpenAI API Documentation. [Online]. Available: <https://platform.openai.com/docs/>
- [26] M. Boersma, "RESTful Web Services: The Basics," *Journal of Web Engineering*, vol. 18, no. 6, pp. 567–586, 2019.
- [27] J. Zhou and B. Liu, "Web-Based Chat Application Using Firebase," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, no. 3, pp. 104–108, 2021.
- [28] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2016. [Online]. Available: <https://www.tensorflow.org/>
- [29] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Draft, 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [30] A. T. Nguyen, N. T. Nguyen, and M. R. A. A. Kadir, "Real-Time Sentiment Analysis on Social Media Using Deep Learning Techniques," *IEEE Access*, vol. 10, pp. 12345–12356, 2022.