Archives available at journals.mriindia.com

International Journal on Advanced Computer Theory and Engineering

ISSN: 2319-2526
Volume 14 Issue 01, 2025

ExecuChat: A Secure AI-Integrated Web-Based Chat Application with Code Debugging and Execution

Prof. Qudsiya Naaz[1], Mirza Rehan Beg[2], Hasnain Raza Khan[3], Hasib Ur Rahman[4]
[1-4]*Dept.of computer science and engineering &RTMNU,India*
[1]*sqnaaz@anjumanengg.edu.in;* [2]*rm88145@gmail.com;* [3]*thehrkofficial@gmail.com;;*
[4]*mr.hasiburrahman128@gmail.com*

Peer Review Information

Submission: 07 Feb 2025
Revision: 16 Mar 2025
Acceptance: 18 April 2025

Keywords

MERN
ExecuChat
Stack
Encryption
JWT

Abstract

With the increasing reliance on real-time communication platforms, there is a growing need for secure, AI-integrated environments that support software development. ExecuChat is a web-based chat application built using the MERN (MongoDB, Express.js, React.js, Node.js) stack, integrated with Google's Gemini AI to facilitate real-time debugging and execution of code within a collaborative, secure messaging system. This research discusses the challenges in traditional chat applications, particularly regarding security vulnerabilities and lack of AI support for developers. The proposed system incorporates robust encryption techniques, JWT-based authentication, and AI-powered debugging assistance to enhance efficiency, security, and collaborative development.

INTRODUCTION

ExecuChat is a web-based chat application designed to provide real-time communication, AI-powered code editing, debugging, and execution in a single platform. Unlike traditional chat applications, ExecuChat allows users to write, test, and debug code directly within the chat environment, making it a powerful tool for developers, students, and researchers. The integration of Gemini AI enhances the user experience by providing coding suggestions, debugging assistance, and real-time code analysis. Users can share their code with others, collaborate in real time, and receive AI-driven insights, making coding more efficient and interactive.

There is a growing demand for a secure, AI-powered, web-based coding chat platform that does not require complex software installation. Many students, researchers, and developers lack access to high-performance computers or struggle with setting up coding environments. With the increasing adoption of AI-powered chatbots, coding assistants have become essential for streamlining development workflows. Existing AI chatbots can help with code generation, but they lack real-time execution and collaborative features. ExecuChat bridges this gap by offering a secure, interactive coding space where users can write, debug, and execute code while communicating seamlessly with others. Its unique world-shared coding space allows users to work together in real-time, making coding more accessible and collaborative than ever before.

LITERATURE REVIEW

The development of real-time chat applications has significantly evolved with the integration of modern web technologies. Pandey (2023) explored React.js and Firebase for a scalable chat application but lacked a strong security focus. Similarly, Pant et al. (2021) extended chat systems by integrating AI-driven messaging

using React Native and AWS, emphasizing real-time communication but without addressing AI security risks. In contrast, Aydın and Karaarslan (2023) analyzed generative AI in chatbots, highlighting security vulnerabilities like data leaks and adversarial attacks, which are critical concerns for AI-powered communication tools like ExecuChat.

AI-driven summarization and debugging have also been key areas of research. Roy et al. (2023) presented an automated article summarization system using Generative AI and React.js, but noted challenges in understanding complex text structures. Levin et al. (2024) introduced ChatDBG, an AI-powered debugging assistant, showing improved developer productivity but also stressing the need for continuous model training and contextual code understanding. Further, EE Journal (2023) confirmed that AI-assisted debugging enhances efficiency but requires better adaptability to dynamic programming environments.

Security remains a major challenge in AI-driven chat applications. Research from MDPI Applied Sciences (2023) and ScienceDirect (2023) highlights vulnerabilities such as unauthorized access, adversarial AI manipulation, and ethical concerns in AI-driven conversations. Current chat systems lack robust encryption and AI-driven security measures, leaving them susceptible to cyber threats.

A key research gap is the lack of AI-driven debugging integration within secure chat platforms. Existing systems primarily focus on real-time messaging or code execution separately, but none effectively combine both with AI-powered security enhancements. ExecuChat aims to fill this gap by integrating a MERN stack-based chat system with Gemini AI for secure messaging, debugging, and real-time code execution, ensuring a developer-friendly and highly secure communication environment.

PROBLEM STATEMENT

Existing chat platforms are without AI-supported code debugging and code running, and this causes developers to collaborate with inefficiency. Additionally, several security vulnerabilities, such as unencrypted data transmission as well as weak authentication methods, openly expose confidential information. ExecuChat thoroughly addresses these issues via the integration of a secure chat environment along with AI-powered code debugging plus execution.

PROPOSED SYSTEM

ExecuChat is designed as a secure, AI-integrated chat application that enables real-time messaging and AI-assisted code debugging. The system incorporates:

1. AI-Powered Debugging & Execution – Using Gemini AI for real-time code analysis.
2. Secure Messaging – End-to-end encryption, JWT authentication, and secure database management.
3. Real-Time Collaboration – WebSockets and cloud-based AI processing.
4. Scalability & Efficiency – Cloud storage for chat history and AI computations.

SYSTEM ARCHITECTURE AND WORKFLOW

The system architecture of ExecuChat is structured into two primary components: Frontend and Backend, each with its own functionalities and sub-components. The system integrates Google's Gemini AI, a MongoDB database, and secure communication protocols to provide a seamless and secure experience for users.

Frontend: The frontend is developed using React**.js**, providing a highly interactive user interface that enables real-time messaging**,** code editing**,** debugging**,** and execution. It incorporates Monaco Editor for a seamless coding experience, Redux Toolkit **(**RTK Query**)** for efficient state management, and WebSockets for instant communication. The frontend also includes an AI assistant panel powered by Google's Gemini AI, offering intelligent code suggestions and debugging insights. To ensure security and user authentication, JWT **(**JSON Web Tokens**)** and OAuth authentication are implemented, allowing users to log in securely using third-party services like Google and GitHub..

Backend: On the backend, ExecuChat is powered by Node**.**js with Express**.**js, handling all the core processing, including user authentication, real-time messaging, AI interactions, and secure code execution. A real**-**time chat system is integrated using WebSockets, allowing users to communicate instantly while ensuring end-to**-**end encryption **(**E2EE**)** for security. The Gemini AI integration enables context-aware debugging and AI-generated code suggestions, enhancing the user experience. The **code** execution engine is implemented using Docker**-**based isolated environments, allowing users to safely run code in multiple programming languages such as Python, JavaScript, and Java, while mitigating security risks. The MongoDB database efficiently stores user profiles, chat history, code snippets, and AI feedback, employing sharding and indexing for optimized performance and automated backups for data integrity. Additionally, advanced security protocols such as AES-256 encryption**,** rate limiting to prevent DDoS attacks, and

HTTPS enforcement ensure robust data protection.

Database: The database layer of ExecuChat is designed to efficiently manage and store user data, chat messages, code snippets, and AI interactions while ensuring high performance, scalability, and security. MongoDB, a NoSQL database, is used as the primary storage solution due to its flexibility, scalability, and ability to handle unstructured data efficiently. The database is structured with multiple collections, including User Profiles, Chat History, Code Snippets, Execution Logs, and AI Feedback, ensuring an organized and optimized data flow. Sharding and indexing techniques are implemented to enhance query performance and load balancing, enabling seamless real-time interactions. End-to-end encryption (E2EE) is applied to sensitive data, such as chat messages and user credentials, ensuring privacy and security. Furthermore, automatic backup and disaster recovery mechanisms are in place to prevent data loss. To support AI-powered functionalities, a separate collection stores AI-generated debugging insights and recommendations, allowing users to access past AI-assisted solutions. With these features, the database layer ensures a fast, secure, and scalable foundation for ExecuChat's seamless operation.

AI Use: he AI integration in ExecuChat is a key feature that enhances the user experience by providing intelligent code debugging, real-time assistance, and smart recommendations. Powered by Google's Gemini AI, the AI system analyzes user-submitted code, identifies potential errors, suggests improvements, and even offers alternative implementations. It also facilitates natural language interactions, allowing users to ask programming-related queries and receive context-aware explanations.

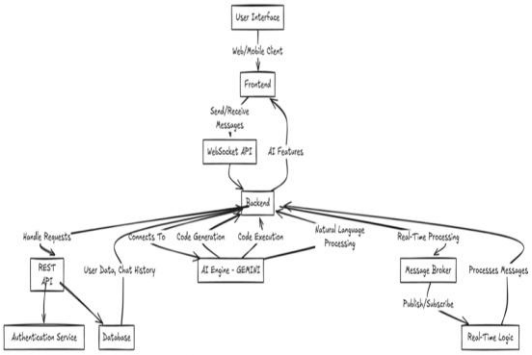

Fig 1 System Architecture

1. Security Features

The security features of ExecuChat ensure a safe, private, and secure environment for users to communicate, share code, and execute programs. The platform implements end-to-end encryption (E2EE) for chat messages, ensuring that only the intended recipients can access the conversation. JWT (JSON Web Token)-based authentication is used to secure user sessions, preventing unauthorized access. To safeguard sensitive data, AES-256 encryption is applied to stored user information, chat logs, and code snippets. The platform also incorporates role-based access control (RBAC), restricting access to certain functionalities based on user permissions. Secure code execution is enforced through Docker-based isolated environments, preventing malicious scripts from affecting the system. DDoS protection mechanisms, such as rate limiting and CAPTCHA verification, mitigate risks from automated attacks. Additionally, HTTPS enforcement ensures secure data transmission over the network, while real-time monitoring and anomaly detection help identify and prevent potential security threats. With these multi-layered security measures, ExecuChat provides a safe and reliable space for developers and researchers to collaborate without compromising their data privacy or system integrity.

2. AI features

ExecuChat leverages Gemini AI to enhance user experience by integrating intelligent code assistance, debugging, and real-time execution support. The AI-powered chatbot helps users by providing syntax suggestions, code completion, and optimization tips, making coding more efficient and error-free. The debugging assistant analyzes code for errors, suggests corrections, and explains the logic behind fixes, helping developers understand and learn from their mistakes. Additionally, the AI can interpret user queries, provide coding best practices, and generate contextual code snippets based on user input. ExecuChat also features interactive AI-driven discussions, allowing users to ask technical questions and receive instant, AI-powered explanations. This seamless integration of AI not only enhances coding efficiency but also fosters a more interactive and educational programming environment.

Code Debugging in Execuchat

ExcuChat integrates real-time AI-powered code debugging to assist developers in identifying and fixing errors efficiently. Using Gemini AI, the platform analyzes the submitted code, detects syntax and logical errors, and suggests corrections directly within the chat interface. The debugging system provides step-by-step explanations, helping users understand the root cause of errors while offering optimized solutions. Unlike traditional debugging tools, ExecuChat allows collaborative debugging, where users can share code snippets and receive AI-driven or peer-assisted feedback in

real time. Additionally, the secure execution environment ensures that debugging occurs in a controlled setting, preventing unauthorized access or unintended security vulnerabilities. This seamless integration of debugging within the chat application enhances productivity, making coding more interactive and accessible to developers, students, and researchers alike.

Code Execution in ExecuChat

ExecuChat provides a real-time code execution environment, allowing users to write, test, and run code seamlessly within the chat interface. By integrating Gemini A**I**, the platform ensures efficient execution with AI-driven optimizations and debugging assistance. Users can submit code in multiple programming languages, and the execution engine processes it securely on the server, returning results instantly. Unlike conventional coding environments that require complex setup, ExecuChat offers a hassle-free, web-based execution system accessible from any device. Additionally, the collaborative execution feature enables users to share their code, execute scripts in a shared workspace, and receive feedback from both AI and peers. The secure execution sandbox prevents malicious code from affecting the system, ensuring a safe and reliable coding experience for all users.

Applications of ExecuChat

ExecuChat has a wide range of applications across various domains, making it a valuable tool for developers, educators, students, and businesses. In software development, it serves as a collaborative platform where developers can write, debug, and execute code in real time, eliminating the need for separate debugging tools or execution environments. The integration of Gemini AI provides intelligent code assistance, making it easier for developers to optimize their code and troubleshoot errors efficiently.

In education and training, ExecuChat is highly beneficial for students and educators. It enables interactive coding lessons, allowing instructors to demonstrate code execution in real time and assist students with debugging issues. The AI-powered chatbot also serves as a virtual tutor, providing explanations and suggestions to improve learning outcomes. Coding boot camps and online programming courses can leverage ExecuChat to provide a hands-on, engaging learning experience without requiring students to set up complex development environments.

For businesses and enterprises, ExecuChat can enhance productivity in software teams by streamlining communication, code sharing, and debugging within a single platform. Remote teams can collaborate seamlessly, review each other's code, and resolve issues efficiently. Additionally, the secure execution environment ensures that confidential or sensitive code remains protected.

ExecuChat is also useful in AI research and innovation, where researchers can experiment with AI models, test algorithms, and debug machine learning code in a shared workspace. Its real-time execution and AI-powered assistance make it an ideal platform for prototyping AI applications and conducting collaborative research.

Furthermore, cybersecurity analysts and ethical hackers can use ExecuChat to test scripts, analyze vulnerabilities, and execute security-related code in a controlled environment. The secure chat and execution features ensure that sensitive information is not compromised while enabling professionals to work efficiently.

With its versatile applications, ExecuChat provides an all-in-one platform for real-time coding, debugging, AI assistance, and secure collaboration, making it an essential tool for programmers, students, educators, businesses, and researchers.

System Requirements

Software Requirements:
- React.js, Node.js, Express.js, MongoDB.
- Google's Gemini AI API.
- Cloud hosting (AWS/GCP).

Hardware Requirements:
- Minimum 8GB RAM, 4-core processor.
- Secure cloud storage for AI processing.

CONCLUSIONS

ExecuChat is a cutting-edge web-based chat application that seamlessly integrates real-time communication, AI-powered assistance, code debugging, and execution functionalities into a single platform. Designed with security, collaboration, and efficiency in mind, it provides developers, educators, students, and businesses with a versatile and user-friendly environment for coding and knowledge sharing. The integration of Gemini AI enhances debugging and learning by offering intelligent suggestions, real-time assistance, and contextual code insights. Additionally, the platform's secure chat environment ensures that sensitive code and discussions remain protected, making it ideal for both individual and team-based projects.

By bridging the gap between traditional coding environments and modern AI-powered collaboration, ExecuChat sets a new standard for interactive and secure programming platforms. Its future developments, including expanded language support, AI-driven debugging, and cloud-based execution, will further enhance its capabilities and user experience. As technology

continues to evolve, ExecuChat aims to remain at the forefront of AI-assisted coding, enabling a more efficient, secure, and accessible development ecosystem for users worldwide.

References

R. Pandey, "Chat Application using React.js and Firebase," *Amity Journal of Computational Sciences*, vol. 7, no. 1, 2023.

K. Roy, S. Mukherjee, and S. Dawn, "Automated Article Summarization using Artificial Intelligence Using React JS and Generative AI," *Journal of Emerging Technologies and Innovative Research*, 2023.

K. Pant, M. S. Rayeen, N. K. Singh, and P. Dominic, "Design and Implementation of the P2P Instant Artificial Intelligence Messaging Application," *Annals of the Romanian Society for Cell Biology*, pp. 19526–19529, 2021.

O. Aydın and E. Karaarslan, "Is ChatGPT Leading Generative AI? What is Beyond Expectations?" *2023*.

K. Levin, N. van Kempen, E. D. Berger, and S. N. Freund, "ChatDBG: An AI-Powered Debugging Assistant," *arXiv preprint arXiv:2403.16354*, 2024. [Online]. Available: https://arxiv.org/abs/2403.16354

"Understanding Real-Time Collaborative Programming: A Study of Developers' Practices and Perceptions," *ACM Digital Library*, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3643672

"Leveraging ChatGPT to Enhance Debugging: Evaluating AI-Driven Solutions in Software Development," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/383509319_Leveraging_ChatGPT_to_Enhance_Debugging

"Real-Time Document Collaboration—System Architecture and Design," *MDPI Applied Sciences*, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/14/18/8356

"A Survey of Collaborative Tools in Software Development," *Web.engr.oregonstate.edu*, 2004. [Online]. Available: https://web.engr.oregonstate.edu/~sarmaa/wp-content/uploads/2020/08/d66dcecb127141211b0463954b23701232aa.pdf

"Using Generative AI for Refactoring and Debugging Code Cuts Debugging Time in Half," *EE Journal*, 2023. [Online]. Available: https://www.eejournal.com/article/using-generative-ai-for-refactoring-and-debugging-code-cut-debugging-time-in-half

"Real-Time Collaboration Through Web Applications: An Introduction to the Toolkit for Web-Based Interactive Collaborative Environments," *Springer Journal*, 2013. [Online]. Available: https://link.springer.com/article/10.1007/s00779-013-0729-0

"ChatGPT: A Comprehensive Review on Background, Applications, Key Challenges, Bias, Ethics, Limitations and Future Scope," *ScienceDirect*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266734522300024X