



## Building a Real-time Communication Platform: A Full Stack WhatsApp Clone with Video and Voice Call Functionality

<sup>1</sup>Sarita Singh, <sup>2</sup>Abhishek Kumar Patel, <sup>3</sup>Krishna Chandra Maurya, <sup>4</sup>Naveen Kumar, <sup>5</sup>Nashrah

[ssingfcs@rkgit.edu.in](mailto:ssingfcs@rkgit.edu.in), [airforceabhishek562@gmail.com](mailto:airforceabhishek562@gmail.com), [kcmaurya2001@gamil.com](mailto:kcmaurya2001@gamil.com), [newnaveen24kumar@gmail.com](mailto:newnaveen24kumar@gmail.com), [nashrak031@gmial.com](mailto:nashrak031@gmial.com),

<sup>\*1</sup> Assistant Professor, Department Of Computer Engineering, RKGIT, Ghaziabad, Uttar Pradesh, India.

<sup>\*2,3,4,5</sup> Students, Department Of Computer Engineering, RKGIT, Ghaziabad, Uttar Pradesh, India.

### ABSTRACT :

In this research paper, we propose the development of a full-stack WhatsApp clone enriched with video and voice call functionalities. Leveraging Next.js, Tailwind CSS, Socket.io, Prisma, Node.js, Express, PostgreSQL, Firebase, and other libraries, our aim is to create a comprehensive real-time chat application that rivals the original platform. The integration of ZEGOCLOUD's video and voice call SDK ensures seamless and secure communication experiences. Key features include Google login via Firebase, flexible database management with Prisma, real-time messaging with Socket.io, and crystal-clear voice and video calls. Additional functionalities such as voice notes with live audio waveforms, emoji support, image sharing, online/offline status indicators, message search, and message read status enhance the user experience. Through this endeavor, we aim to explore the intricacies of building a sophisticated communication platform while highlighting the potential of modern technologies in shaping the future of messaging applications.

**KEYWORDS:** Comprehensive chat application, Modern communication platform, Real-time interaction, Next.js framework, Tailwind CSS styling and Socket.io integration.

### INTRODUCTION:

Realtime Chat App (WhatsApp Clone) using Next.js, Socket.io, Tailwind CSS, Node.js, and Zegocloud. The Realtime Chat App is a cutting-edge web application that replicates the core functionalities of WhatsApp, offering users an immersive and responsive platform for seamless real-time communication. Built with a modern tech stack including Next.js, Socket.io, Tailwind CSS, Node.js, and integrated with Zegocloud for data storage and synchronization, the app delivers a secure and user-friendly messaging experience.

The Realtime Chat App represents more than just a digital communication tool; it embodies a vision for the future of online interactions. Through a meticulous integration of cutting-edge technologies, this project not only replicates the familiar features of WhatsApp but elevates the user experience to new heights. Imagine a world where communication transcends the barriers of time and distance. With the Realtime Chat App, this vision becomes a reality. Users can create accounts securely, ensuring a personalized and protected space for their digital conversations. The authentication process, fortified with JWT, lays the foundation for a robust and secure user environment.

Real-time messaging adds a layer of dynamism to conversations. The instantaneous updates mimic face-to-face interactions, fostering a sense of immediacy and connection. No longer confined by delays or asynchronous exchanges, users can experience the flow of conversation in its true, uninterrupted form. The contact list management feature offers more than just a directory of names. Users can effortlessly track the online/offline status of their contacts, creating a more connected experience. Message status indicators, such as 'sent,' 'delivered,' and 'read,' provide users with insights into the progression of their conversations, bringing transparency to the communication process.

Multimedia sharing goes beyond conventional text messages, allowing users to share the richness of moments through images, videos, and various media files. Group chats facilitate virtual gatherings, enabling users to create and manage group conversations effortlessly. The responsive design ensures a seamless and visually appealing interface, adapting to the diverse array of devices used in today's digital landscape.

Message history retrieval ensures that no conversation is lost in the digital ether. Users can revisit past interactions, creating a sense of continuity even after logging out. Instant notifications keep users informed, ensuring that they never miss a beat, even

when the app is not actively open. Profile customization features allow users to express their individuality. Setting profile pictures and status messages adds a personal touch to their virtual presence. The implementation of end-to-end encryption underscores the commitment to privacy and data security, instilling confidence in users that their conversations are protected.

By leveraging Zegocloud for data storage and synchronization, the Realtime Chat App offers a seamless cross-device experience. The tech stack, comprising Next.js, Socket.io, Tailwind CSS, Node.js, and Express.js, represents a harmonious marriage of efficiency and innovation, forming the backbone of a reliable and scalable application architecture. While challenges such as real-time functionality, encryption, Zegocloud integration, and UI design were encountered, each obstacle was met with expertise and determination. The result is not just a chat application; it's a testament to the dedication to delivering a flawless user experience.

In conclusion, the Realtime Chat App is an exploration into the future of digital communication. It doesn't merely replicate existing functionalities; it reshapes the landscape, setting a new standard for real-time, secure, and user-centric interactions. Join us on this journey as we redefine the art of digital conversation, one message at a time.

---

## EXISTING SOFTWARE

WhatsApp, as one of the pioneers in the field, boasts an expansive user base globally due to its simplicity and cross-platform compatibility. Its end-to-end encryption ensures that messages and calls remain private and secure. Telegram, on the other hand, emphasizes speed and versatility. The platform allows users to create channels for broadcasting to large audiences, groups for community interactions, and boasts a range of multimedia features. Signal, known for its privacy-centric approach, has gained popularity for its commitment to open-source development and privacy-focused features, making it a go-to choice for those who prioritize confidentiality.

In the professional realm, Slack has emerged as a leader in team collaboration. Its interface allows users to organize discussions into channels, integrate with various third-party apps, and efficiently share files and information within a team. Microsoft Teams seamlessly integrates with the broader Microsoft 365 suite, providing a comprehensive platform for workplace collaboration, including video conferencing, document collaboration, and project management.

These existing chat applications not only facilitate communication but also integrate additional functionalities to cater to diverse user needs. The continual evolution of these platforms, driven by user feedback and technological advancements, underscores the dynamic nature of the communication software landscape.

---

## RELATED WORK

The landscape of chat-based platforms and mobile learning techniques has been extensively explored in prior research, showcasing a wide array of applications and methodologies. The study by Kim et al. (2020) delves into the realm of personalized recommendations within the dining sector. Their restaurant recommendation app stands out for its integration of collaborative filtering and AI chatbots. By leveraging collaborative filtering, the app tailors suggestions based on user preferences and behaviors, while AI chatbots enhance the user experience by providing dynamic and interactive information on dining options. This intersection of recommendation systems and conversational agents represents a novel approach to enhancing user engagement and satisfaction in the context of dining choices.

Fathy El's research (2020) shifts the focus to mobile learning techniques, specifically investigating the effectiveness of WhatsApp in improving students' writing skills. The experimental group, which utilized WhatsApp technology, exhibited advancements in their writing abilities compared to the control group following traditional teaching methods. This underscores the potential of mobile messaging platforms as educational tools, offering a dynamic and accessible channel for skill development beyond conventional teaching approaches.

In a similar vein, Alonzo and Oo's study (2023) explores the collaborative aspects of using Messenger for research collaboration. By employing WhatsApp technology, the experimental group engaged in collaborative research endeavors, highlighting the platform's versatility beyond traditional communication. The study sheds light on the evolving role of messaging apps in fostering collaboration, particularly in academic and research-oriented settings.

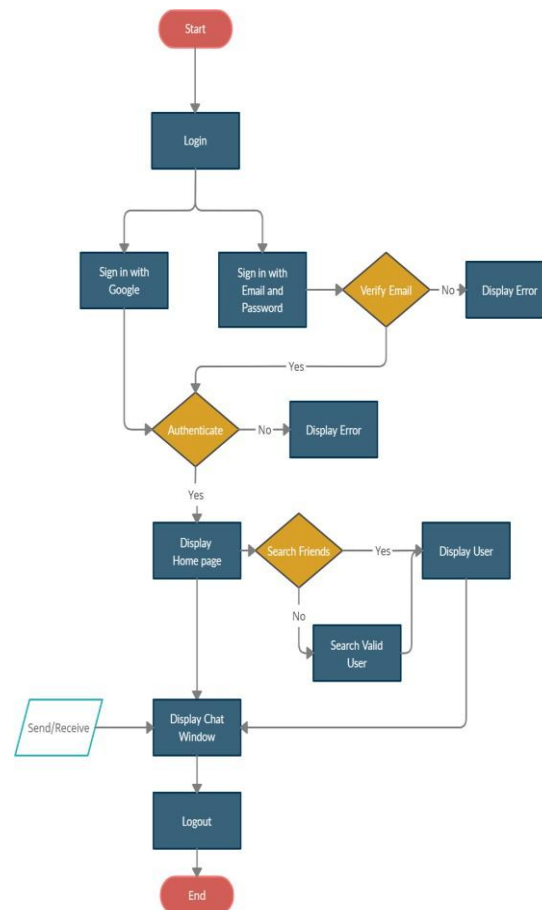
Transitioning to the technical aspects, Maurya et al. (2022) contribute to the field with the development of an Android chat application. Their implementation involves the integration of Google Firebase for authentication and a PHP server for overall application functionality. This technical framework ensures a secure and seamless user experience, underlining the importance of robust backend infrastructure in the success of chat applications.

Jagtap et al.'s work (2022) further emphasizes the technical underpinnings of chat applications by focusing on the implementation of chat rooms using Firebase's real-time database. The adoption of this database technology ensures swift and efficient updates as messages are sent and received, enhancing the real-time nature of the communication platform.

Expanding on the technical front, Bedare et al. (2023) contribute insights into the MERN stack's role in real-time chat applications. The MERN stack, built on JavaScript, emerges as a comprehensive solution for developing web applications. This stack, comprising MongoDB, Express.js, React.js, and Node.js, facilitates end-to-end development, ensuring a seamless connection between browsers and databases. Bedare et al. highlight the stack's

robustness in building feature-rich applications, showcasing its potential for creating dynamic and responsive real-time chat platforms. Collectively, these studies provide a multifaceted view of the applications of chat technologies and mobile learning platforms, demonstrating the intersection of innovative methodologies and their potential implications across various domains. The research not only underscores the importance of user-centric approaches but also emphasizes the critical role of robust technical foundations in the successful implementation of chat-based applications.

## METHODOLOGY



**Figure : Flow-chart**

The methodology for developing the WhatsApp clone with enhanced features involves a systematic approach aimed at meeting project objectives while ensuring quality, security, and usability. Initially, a comprehensive requirement analysis is conducted to identify both functional and non-functional requirements, laying the foundation for subsequent development phases. This analysis involves close collaboration with stakeholders to understand their needs and expectations, ensuring that the final product aligns with user expectations and market demands.

Following this, appropriate technologies are carefully selected, including Next.js and Tailwind CSS for front-end development, Node.js with Express for back-end, Socket.io for real-time communication, Prisma for database management, Firebase for authentication, and ZEGOCLOUD SDK for voice/video calling. The selection of these technologies is based on their suitability for the project requirements, as well as considerations such as scalability, maintainability, and community support.

The project setup involves configuring the development environment and establishing the initial project structure. This includes setting up version control systems, IDEs, and dependencies, as well as defining coding standards and best practices to ensure consistency and maintainability throughout the development process.

Feature implementation proceeds iteratively, starting with basic functionalities such as user authentication and messaging, and gradually integrating more complex features like voice/video

calling and message search. This iterative approach allows for frequent feedback loops, enabling the development team to adapt and refine the product based on evolving requirements and user feedback.

Integration and testing are carried out throughout the development process, ensuring seamless interaction between components and validating the application's functionality through unit tests, integration tests, and end-to-end testing.

User feedback is continuously solicited and incorporated into iterative development cycles, driving improvements in usability and feature set. This feedback may be gathered through user testing sessions, surveys, or feedback forms integrated into the application itself. By prioritizing user feedback, the development team can ensure that the final product meets the needs and expectations of its intended audience.

Performance optimization measures are implemented to enhance system responsiveness and minimize latency, while security measures such as data encryption and secure communication protocols are employed to safeguard user data and privacy. This involves conducting thorough security audits and implementing best practices for secure coding and infrastructure configuration.

This documentation includes setup instructions, architecture overview, API documentation, and codebase structure, ensuring that the application is well-documented and easy to understand for both internal and external stakeholders. Finally, deployment to a production

environment and establishment of a maintenance plan ensure the application's ongoing stability, performance, and relevance in meeting user needs. This involves setting up monitoring and logging systems to track application performance and detect issues in real-time, as well as implementing

regular updates, bug fixes, and feature enhancements based on user feedback and changing requirements. By following this methodology, the development team can ensure the successful delivery of a high-quality WhatsApp clone with enhanced features,

meeting the needs of its users and stakeholders.

## RESULT



Figure 1: Login Page

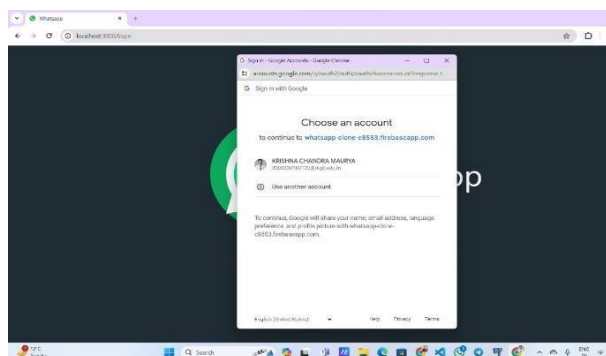


Figure 2: Login with google



Figure 3: Onboard Page

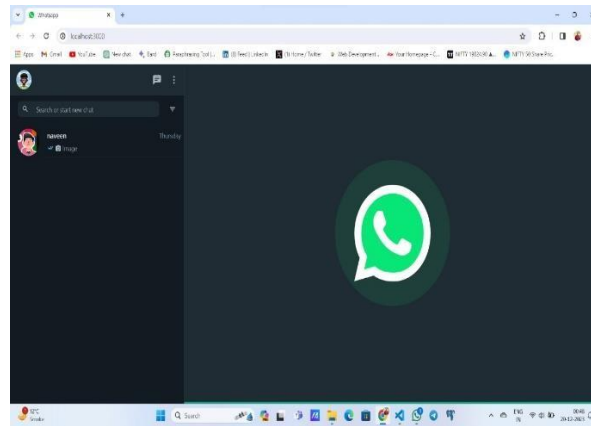


Figure 4: Dashboard

---

## CONCLUSION

In conclusion, the Realtime Chat App project encompasses a comprehensive set of functional and non-functional requirements. By addressing the user authentication, messaging, contact management, group chat, user profile, message history, notifications, security, Zegocloud integration, interface design, performance, scalability, and usability aspects, the application aims to provide a secure, efficient, and user-friendly real-time messaging platform. The successful implementation of these requirements will result in a robust application that meets the expectations of users for seamless and secure communication, demonstrating the capabilities of Next.js, Socket.io, Tailwind CSS, Node.js and Zegocloud in a practical and impactful manner.

---

## FUTURE SCOPE

Looking forward, the WhatsApp clone project holds promise for several exciting developments. Advanced communication features like real-time language translation and improved voice/video call quality are on the horizon. Integration of social media elements and cross-platform compatibility will enhance user engagement. Strengthening security measures and adopting emerging technologies like blockchain and AI will bolster privacy and innovation. Scalability improvements and community collaboration are key for sustained success. Embracing continuous iteration based on user feedback ensures the project remains at the forefront of communication technology.

---

## REFERENCE :

1. Heeyoung Kim, Sunmi Jung and Gihwan Ryu, "A Study on the Restaurant Recommendation Service App Based on AI Chatbot Using Personalization Information", International Journal of Advanced Culture Technology Vol.8 No.4 263-270, 2020,
2. <https://doi.org/10.17703/IJACT.2020.8.4.2.63>
3. Fattah, Fathy , "The Effectiveness
4. of Using Whatsapp Messenger as One of Mobile Learning Techniques to Develop Students' Writing Skills", Journal of Education and practice, 2020, <https://eric.ed.gov/?id=EJ1083503#:~:text=This%20research%20is%20limited%20to.group%20outperformed%20the%20control%20group.>
5. ~:text=This%20research%20is%20limited
6. %20to.group%20outperformed%20the%20control%20group.
7. Ayush, Saurabh Sudhakar, Jitendra Thakare, "Android Chat Application", Social Science Research Network, May 2020,
8. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4109045](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4109045)
9. Shreenath Jagtap, Farhan Pathan and Sujal Jadhav, "Web Chat Station – A review", IRJET,2022, [https://www.researchgate.net/publication/361106310\\_WEB\\_CHAT\\_STATION\\_-\\_A\\_REVIEW](https://www.researchgate.net/publication/361106310_WEB_CHAT_STATION_-_A_REVIEW)
10. Dennis Alonzo, Cherry Zin Oo, "The Use of Messenger for Research Collaboration: An auto-ethnographic study", Frontiers Original Research article, January 2023, <https://doi.org/10.3389/fpsyg.2022.1076340>
11. Nikhil Chaudhari, Sushma Shinkar, Priyanka Pagare, "Chatting Application with Real
12. TimeTranslation", IRJET, 2020, [https://www.academia.edu/38241655/IRJET\\_Chatting\\_Application\\_with\\_Real\\_Time\\_Translation](https://www.academia.edu/38241655/IRJET_Chatting_Application_with_Real_Time_Translation)

13. Soumya Dhanappa . Sahnaz Parveen . Sanya Yadav . Naveen Chandra Gowda, “Enhancing Real-Time Communication: A WebRTC-Based Video Chat Application”, International Journal of Computational Learning&Intelligence,2024, <https://milestoneresearch.in/JOURNALS/index.php/IJCLI/article/view/138>
14. Ömer Özgün Üngüder Südstraße, “Real-Time Chat Application with ADALM-PLUTO Software Defined Radio”, Hochschule Rhein-Waal Rhine-Waal University of Applied Sciences,2023, <https://opus4.kobv.de/opus4-rhein-waal/frontdoor/index/index/docId/1852>
15. Muath Abdullah SAEED , Iraq Naktal Moaid EDAN, “DESIGN AND IMPLEMENT VIDEO AND CHAT APPLICATION USING MESH NETWORK”, MINAR International Journal of Applied Sciences and Technology,2022,<https://www.minarjournal.com/design-and-implement-video-and-chat-application-using-mesh-network-222>
16. Muhammed Burak KILIÇ, “Encryption Methods and Comparison of Popular Chat Applications”, Advances in Artificial Intelligence Research (AAIR), 2021, <https://dergipark.org.tr/en/pub/aaair/issue/64655/959499>