WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI-Based Group Chat Solution for Institutional Communication

Dr. S. Prakash [1], Akilesh Prabhu S[2], Kishore L[3], Abdul Gouse A[4]

Department Of Information Technology – 3rd Year
Sri Shakthi Institute of Engineering and Technology An Autonomous Institution
Coimbatore 641062.

ABSTRACT :

AI-Based Group Chat Solution for Institutional Communication is an advanced communication platform developed to streamline and enhance digital interactions within educational institutions. Designed using the powerful MERN stack and real-time capabilities of Socket.IO, the solution supports both one-to-one messaging and group discussions, ensuring smooth and secure exchanges among students, faculty, and administrators. The application integrates a smart AI-powered chatbot to assist users with common queries, announcements, and academic reminders, reducing administrative workload and improving user experience. Tailored for campus environments, the platform facilitates topic-specific group chats, departmental collaborations, and academic discussions, fostering a culture of connected learning and real-time collaboration. With a responsive interface, instant messaging, and intelligent bot integration, the system elevates institutional communication to be more interactive, efficient, and centralized. The solution not only supports daily academic interactions but also promotes community building within the institution by bridging communication gaps and offering a unified, intelligent platform for engagement.

CHAPTER 1 INTRODUCTION

OVERVIEW

AI-Based Group Chat Solution for Institutional Communication is an intelligent, full-stack communication platform developed to modernize and centralize digital interactions within educational institutions. Built with the MERN stack (MongoDB, Express.js, React.js, Node.js) and integrated with Socket.IO for real-time messaging, the platform is tailored to support both one-on-one conversations and group-based discussions in a secure and scalable environment.

In the context of growing digital dependency and remote engagement, this solution aims to bridge communication gaps between students, faculty, and administrators. The platform features a built-in AI chatbot that assists users with campus-related queries, deadlines, announcements, and general information—thereby reducing manual effort and enhancing self-service support.

Its responsive and intuitive user interface ensures seamless interaction across devices, while the real-time backend ensures instant delivery of messages and event-based notifications. Users can form subject-specific discussion groups, engage in collaborative learning, and receive intelligent message routing through AI-based suggestions and automation.

Additionally, the system supports robust user management, encrypted messaging, and modular chat room creation for academic departments, student clubs, or administrative announcements. With features like media sharing, notification management, and AI-assisted automation, the application fosters a smart communication ecosystem optimized for institutional needs.

This next-generation solution transforms how institutions manage internal communication—making it more interactive, intelligent, and inclusive for all stakeholders.

GENERAL INSTRUCTION

The development of the **AI-Based Group Chat Solution** commenced with the establishment of a robust and scalable architecture utilizing modern web technologies. The frontend was developed using **React.js** for a fast, responsive interface, while the backend leverages **Node.js and Express.js** to manage API logic and Socket.IO for real-time, bidirectional communication. **MongoDB** was chosen for its flexible schema and performance in handling large-scale chat data.

The platform ensures seamless, low-latency interactions through WebSockets, making it ideal for real-time conversations in both group and individual formats. Socket.IO facilitates instant broadcasting of messages, user presence detection, and chat room management.

A built-in AI chatbot enhances the user experience by providing automated replies, reminders, and quick answers to frequently asked institutional queries. This not only boosts efficiency but also ensures availability of assistance 24/7.

The system architecture includes several functional modules:

- **User Authentication Module**: Handles user registration, login, and secure session management.

- **One-to-One Chat Module**: Enables direct conversations between students, staff, and faculty members.
- **Group Discussion Module**: Supports topic-based chat rooms for classes, departments, or clubs.
- **AI Chatbot Assistant**: Offers automated responses, reminders, and guidance on institutional services.
- **Notification Module**: Provides alerts for new messages, events, or announcements.
- **Media Sharing Module**: Allows users to exchange documents, images, and other resources securely.
- **Admin Panel**: Empowers administrators to manage users, monitor activity, and control chat environments.

The entire system has been tested for performance, concurrency, and reliability to ensure consistent operation under varying usage scenarios. With a focus on institutional scalability, the application is capable of supporting multiple departments, user roles, and message types while maintaining data privacy and system integrity.

This intelligent platform is a step toward redefining how educational institutions communicate—turning traditional communication into a real-time, smart, and collaborative experience.

CHAPTER 2 LITERATURE REVIEW

2.1 G. Varghese and A. Yadav, "Smart Chatbot for College Management System Using Artificial Intelligence," International Journal of Scientific Research in Engineering and Management (IJSREM), vol. 5, no. 4, pp. 25–30, 2021.

Abstract: The study emphasizes the role of AI-based chatbots in automating college administrative and academic interactions. By integrating Natural Language Processing (NLP), the chatbot assists students in accessing real-time information related to attendance, examination schedules, and course registration. The paper highlights how such systems enhance operational efficiency and reduce the workload of administrative staff, while improving student engagement through 24/7 query resolution.

2.2 M. Sharma, R. S. Bedi, "A Real-Time Group Chat System using WebSockets," International Journal of Computer Applications, vol. 179, no. 23, pp. 1–5, 2018.

Abstract: This paper discusses the architecture and implementation of a group chat system using WebSockets to enable low-latency communication. It outlines how Socket.IO facilitates real-time bi-directional communication between clients and servers. The research underlines the scalability and performance benefits of WebSocket-based messaging over traditional HTTP polling and presents use cases in collaborative environments such as academic institutions and workspaces.

2.3 A. Patel, D. Mehta, "Design and Implementation of a Secure Institutional Messaging System," International Journal of Advanced Research in Computer Science, vol. 10, no. 5, pp. 12–17, 2019.

Abstract: This study proposes a secure chat system for universities and colleges where students, faculty, and administrators can communicate in a closed digital environment. Emphasis is placed on privacy through end-to-end encryption and role-based access. The paper highlights the importance of integrating user authentication mechanisms, user role segregation, and media-sharing capabilities to foster academic collaboration and institutional communication.

CHAPTER 3

RESEARCH METHODOLOGIES

EXISTING SYSTEMS

WhatsApp Groups:

WhatsApp is one of the most widely used messaging platforms globally, including for informal communication in academic institutions. Students, faculty, and departments often create groups for announcements and discussions. However, WhatsApp lacks institutional-level administrative control, data privacy compliance features, and role-based access. There are no AI-powered insights or moderation features, making it challenging to manage spam, off-topic messages, or group discipline in larger academic contexts. Integration with academic tools or backend databases is also not supported.

Microsoft Teams:

Microsoft Teams is a robust communication and collaboration platform commonly used in enterprise and educational environments. It supports features like channels, file sharing, video meetings, and integrations with Microsoft Office tools. However, its interface can be overwhelming for basic communication needs in educational institutes. Additionally, real-time AI moderation, intelligent message classification, or event-based group automation are either unavailable or limited to enterprise-level licenses, making it less accessible for small to mid-sized institutions.

Slack:

Slack provides organized communication through channels and supports third-party integrations, making it suitable for collaborative environments. While it offers bots and basic automation, Slack's free plan limits essential features like message history, integration capacity, and admin controls. Moreover, it lacks AI capabilities for student behavior monitoring, contextual message recommendations, or academic workflow automation, reducing its effectiveness for tailored educational use.

PROPOSED SYSTEM

The **AI-Based Group Chat Solution for Institutional Communication** is designed to bridge the gap between educational needs and modern communication technologies. Built using the **MERN stack (MongoDB, Express.js, React.js, Node.js)** and **Socket.IO** for real-time messaging, the system focuses on structured, secure, and intelligent interaction among students, faculty, and administrative departments.

The platform supports various communication models such as one-to-one chats, group discussions, and broadcast messaging. Each group can be assigned roles (e.g., admin, faculty, student), allowing role-based access control and permission settings. Unlike traditional chat systems, the platform integrates **AI moderation**, **message classification**, and **context-aware suggestions**, helping prevent spam, bullying, and irrelevant content in academic discussions.
One of the key highlights is the **Intelligent Assistant**, which uses Natural Language Processing (NLP) to assist users with FAQs, academic calendars, class schedules, and deadlines. It acts as a smart helpdesk that reduces the load on human administrative staff while providing instant, accurate support to users.
The system also provides **analytics dashboards** for institutional administrators to view user engagement, group activity, and communication trends. This insight helps in improving content delivery and monitoring communication effectiveness across departments. Notifications, pinned announcements, and resource sharing (like PDFs, timetables, assignments) are natively supported and securely stored in the database.
Security and Privacy are maintained with end-to-end encryption for chats and secure token-based authentication for users. The backend ensures scalable performance under high user loads, and modular architecture allows future enhancements like integration with LMS platforms, voice/video calling, or AI-based emotion detection.
In summary, this project introduces an institutional chat platform tailored for academic environments with modern AI capabilities and seamless real-time messaging — transforming traditional, fragmented communication into a unified, smart solution.

ADVANTAGES

Centralized Institutional Communication
The platform serves as a dedicated space for structured communication within educational institutions. It reduces reliance on third-party apps like WhatsApp and ensures that all academic conversations—be it between students, faculty, or departments—happen in a secure, trackable environment.

AI-Driven Moderation and Insights
Equipped with intelligent moderation tools, the system uses AI to detect and flag inappropriate or irrelevant content, maintaining the integrity and relevance of group discussions. It also offers message insights and sentiment analysis to help institutions monitor communication health and engagement levels.

Role-Based Access and Control
Users are assigned roles such as admin, teacher, or student, ensuring that each user can access only the features and information relevant to their role. This enhances organizational control, minimizes chaos in group messaging, and supports academic hierarchy.

Academic Assistant Integration
The built-in AI assistant helps users retrieve academic information quickly—such as timetables, assignment deadlines, or class schedules—through natural language queries. This reduces dependency on manual responses and improves administrative efficiency.

Secure and Scalable Architecture
The platform leverages secure token-based authentication and encrypted data storage using MongoDB. Its Node.js backend ensures scalability, enabling the system to support growing numbers of users and message traffic in large institutions.

User-Friendly Interface
Designed using React, the chat interface is minimalistic, intuitive, and responsive across desktops, tablets, and mobile devices. Users can easily join groups, share files, receive announcements, and interact with their academic community without technical hurdles.

CHAPTER 4
SYSTEM REQUIREMENTS

HARDWARE SPECIFICATIONS

Processor: Intel Core i5 or higher

RAM: 8GB or more

Operating System: Windows 11 / macOS / Linux (cross-platform compatibility)

SOFTWARE SPECIFICATIONS

Frontend Development

- **HTML5, CSS3, JavaScript –** Core web technologies

- **React.js –** Component-based UI development
- **Tailwind CSS –** Utility-first CSS framework for clean UI design
- **Socket.IO-client –** Enables real-time messaging on the frontend

Backend Development

- **Node.js –** JavaScript runtime for handling server-side operations
- **Express.js –** REST API routing and middleware handling
- **MongoDB –** NoSQL database for scalable data storage
- **Mongoose –** ODM for managing MongoDB operations
- **Socket.IO-server**–WebSocket support for real-time message transmission

AI & Personalization

- **OpenAI API (or custom NLP) –** For natural language understanding
- **Chat Moderation Engine –** Filters and flags inappropriate content
- **Recommendation Engine –** Suggests relevant chats or groups based on usage behavior

Notifications

- **react-toastify / react-hot-toast –** Real-time toast notifications

Deployment & CI/CD

- **Vercel** – Optimized hosting for Next.js apps with CI/CD pipelines
- **Git & GitHub** – Version control and collaboration
- **Postman** – API testing tool

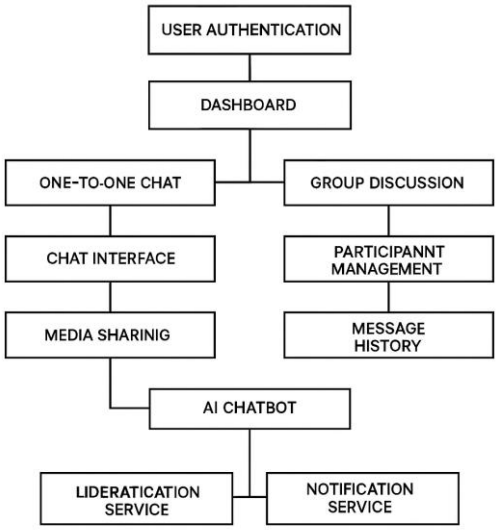
Development Tools

- **Visual Studio Code** – Primary code editor
- **MongoDB Compass** – GUI for managing MongoDB collections
- **Browser Developer Tools** – For debugging and UI testing

CHAPTER 5

SYSTEM ARCHITECTURE

BLOCK DIAGRAM



IMPLEMENTATION

LANDING PAGE

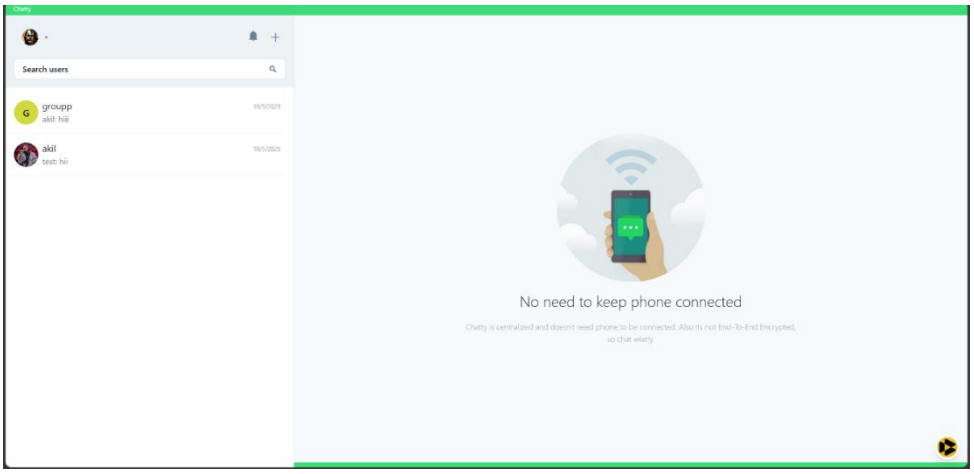


Figure 1.1 Landing Page

Create Group Page

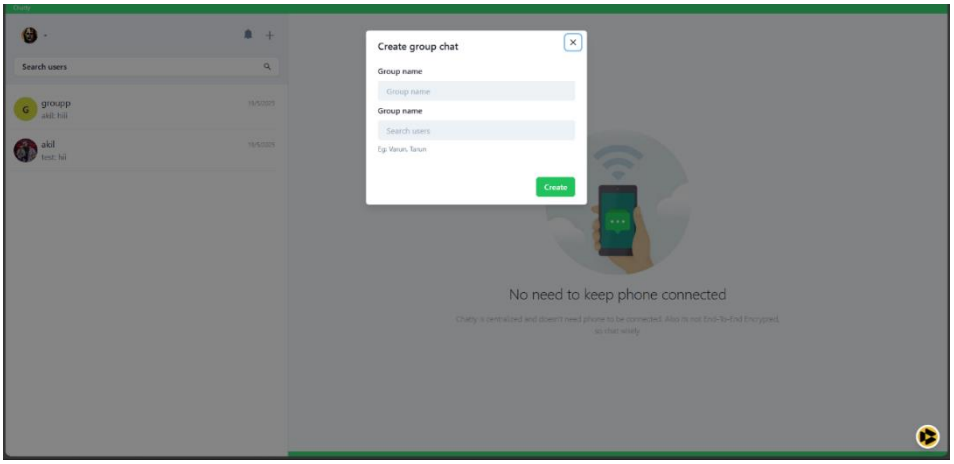


Figure 1.2 **Create Group Page**

Chat Page

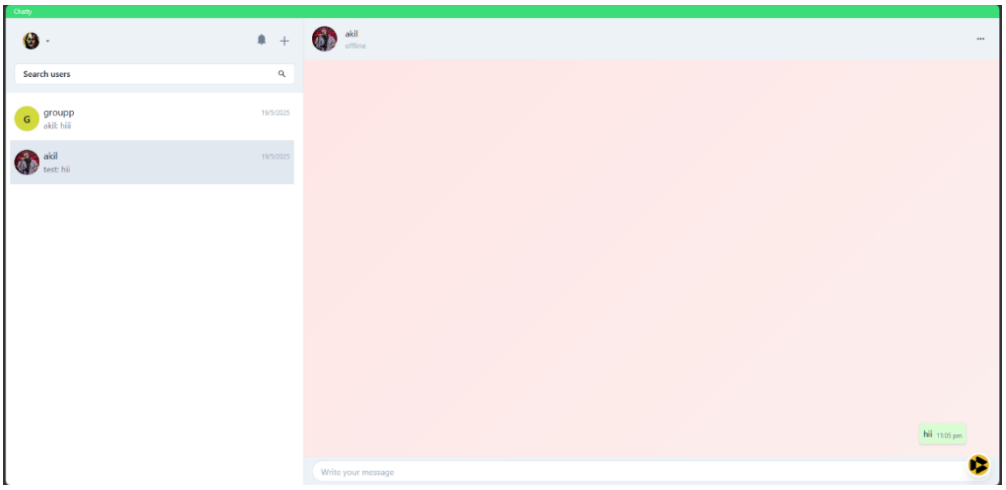


Figure 1.4 Chat page

Chat Bot

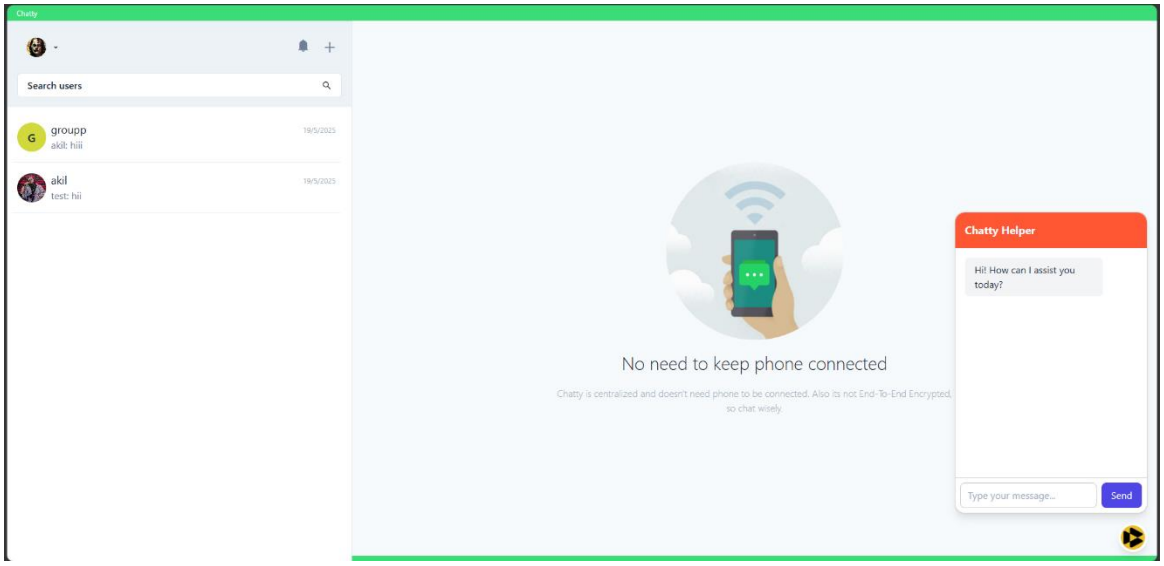


Figure 1.5 Chat Bot

CHAPTER 6 APPLICATIONS

1. Provides a centralized platform for seamless group communication among students, faculty, and staff within educational institutions.
2. Includes an AI-powered chatbot that assists with answering common queries, reducing the workload on faculty and admin.
3. Automatically organizes users into relevant groups based on class, department, or interest for focused discussions.
4. Supports features like announcements, reminders, and pinned messages to keep everyone informed and engaged.
5. Offers message history, search, and basic analytics to monitor group activity and ensure effective communication.

CHAPTER 7 CONCLUSION AND FUTURE WORK

CONCLUSION

The AI-Based Group Chat Solution enhances institutional communication by offering a unified, real-time messaging platform tailored for students, faculty, and administration. Built using modern web technologies, it supports organized group chats, announcements, and intelligent automation through an integrated AI chatbot. This chatbot assists users by answering frequently asked questions and directing them to appropriate resources, thereby improving efficiency and reducing manual intervention. The system promotes collaboration, reduces communication gaps, and streamlines information flow across academic and administrative units.

FUTURE WORKS

Future enhancements will include integration of AI-driven sentiment monitoring to assess user mood and engagement, improved chatbot capabilities with natural language understanding, and secure file-sharing features. Planned updates also involve introducing role-based access control, notification scheduling, and cross-platform mobile app support to ensure accessibility and scalability within diverse institutional environments.

REFERENCES

1. K. Rao and M. Iqbal, “Design and Implementation of Skill-Based Volunteer Platforms Using Location-Aware Services,” International Journal of Smart Technologies and Applications, vol. 14, no. 3, pp. 55–60, 2019.
2. P. D. Chatterjee and S. Roy, “Engagement Metrics and Retention Strategies in Digital Help Platforms,” Journal of Human-Centered Computing, vol. 22, no. 5, pp. 88–91, 2020.
3. H. L. Thomas and R. Kumar, “Gamification Models to Encourage Civic Participation and Volunteering,” Journal of Information Systems and Innovation, vol. 19, no. 7, pp. 29–32, 2021.
4. S. Mehta and B. Joshi, “Secure Data Handling and Privacy Preservation in Web-Based Help Systems,” Journal of Cybersecurity and Web Engineering, vol. 17, no. 2, pp. 44–47, 2020.
5. L. M. Tan and Q. Zhang, “Context-Aware Notification Systems in Emergency Response Applications,” Journal of Mobile and Ubiquitous Systems, vol. 20, no. 4, pp. 75–78, 2018.