e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

RESEARCH PAPER FOR REAL-TIME WORKSPACE COMMUNICATION PLATFORM

Samarjeet Singh Kheda*1, Samarth Agrawal*2, Sanidhya Neema*3, Shubham Gangwane*4

*1,2,3,4Computer Science And Engineering Acropolis Institute Of Technology And Research, Indore, India.

DOI : https://www.doi.org/10.56726/IRJMETS72133

ABSTRACT

This document contains detailed study of our project "Real-Time Workspace Communication Platform" and similar already existing systems. It contains detailed information about the problems addressed by our system, technology stack used to implement planned application, methodology behind complete process and comparative study of existing study.

Keywords: Real-Time Communication, Collaboration.

I. INTRODUCTION

The Real-Time Workspace Communication Platform is a web application designed to enhance team collaboration through seamless real-time messaging, structured discussions, and efficient workspace management. As remote and hybrid work environments grow, traditional communication tools often struggle with fragmented conversations, lack of real-time updates, and limited message control. This platform addresses these challenges by enabling workspace creation, channel-based discussions, direct messaging, and threaded replies, ensuring structured and meaningful interactions. To improve user experience and workflow efficiency, it incorporates features such as role-based access control, message reactions, editing, deletion, and an intuitive invite system. Additionally, image attachments, user profiles, and a modern, user-friendly interface create a more engaging and interactive environment. By tackling common issues like message clutter, inefficient workspace organization, and lack of granular user control, this platform fosters a more organized, collaborative, and productive work experience.

1. Problem Formulation

Effective team communication is often hindered by fragmented conversations, inefficient workspace management, and lack of structured discussions in existing platforms. The Real-Time Workspace Communication Platform addresses these challenges by providing real-time messaging, workspaces, channels, threaded discussions, and role-based access control to ensure structured and secure collaboration. Additionally, the platform enhances user experience with message reactions, editing, deletion, secure authentication, and an intuitive invite system, enabling seamless onboarding and controlled interactions.

Key Problem Components:

A. Real-Time Messaging & Structured Discussions: Enables instant communication through channels, direct messages, and threaded replies, ensuring organized and contextual conversations.

B. Role-Based Access Control (RBAC): Implements granular permission levels for admins, moderators, and users to maintain a secure and well-regulated workspace.

C. Message Management Features: Supports reactions, editing, and deletion, allowing users to modify and engage with messages efficiently while maintaining conversation integrity.

D. Secure Authentication & User Management: Ensures protected access with multi-provider authentication, enhancing security and user identity management.

E. Intuitive Invite System: Provides workspace and channel-based invite codes.

II. LITERATURE REVIEW

Evolutions in the field:

The field of real-time communication and workplace collaboration has undergone a significant transformation, evolving from basic text-based messaging systems to highly integrated, feature-rich platforms. Early communication tools primarily relied on email, which, while effective for formal and asynchronous

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

communication, lacked immediacy and interactive features. The late 1990s and early 2000s saw the rise of instant messaging (IM) applications such as AOL Instant Messenger, MSN Messenger, and Yahoo Messenger, which enabled real-time conversations. However, these tools were mostly designed for personal use and lacked the structured workspace management, access control, and collaboration features needed for professional environments.

With the expansion of cloud computing, web technologies, and mobile connectivity, workplace communication evolved to include team-based chat platforms that supported group discussions, file sharing, and task management. The introduction of platforms such as Slack (2013), Microsoft Teams (2017), and Discord (2015) marked a major shift in digital communication. These tools provided features such as threaded conversations, real-time message reactions, and integrations with external applications, allowing teams to replace traditional email-based workflows with more interactive and dynamic communication systems.

Despite their widespread adoption, existing platforms still present several limitations. Many commercial solutions impose restrictions on message history (such as Slack's free-tier limit), enforce rigid workspace structures, or require expensive enterprise subscriptions to unlock full functionality. Additionally, while some platforms, such as Microsoft Teams, offer deep integration with productivity tools, they often come with a steep learning curve and limited customization options. On the other hand, open-source alternatives such as Mattermost and Rocket.Chat provide self-hosted solutions with greater control over data privacy and security, but they often require technical expertise for deployment and maintenance, making them less accessible to non-technical teams.

As remote and hybrid work environments become more prevalent, the need for scalable, flexible, and user-friendly real-time communication platforms continues to grow. The Real-Time Workspace Communication Platform aims to address these gaps by providing a structured, intuitive, and feature-rich collaboration tool that combines seamless real-time messaging, role-based access control, workspace and channel management, and an efficient onboarding system. By leveraging best practices from existing solutions while incorporating new innovations, this project seeks to enhance team productivity, optimize digital collaboration, and provide an improved communication experience for modern workplaces.

Comparative Analysis of existing systems:

System	Problems Addressed	Advantages	Disadvantages	Gaps Identified
Slack	Real-time messaging, workspaces, and integrations.	Threaded conversations, reactions, role-based access.	Free plan limits message history, lacks end-to-end encryption.	No unlimited message history in free plan, reliance on third-party apps.
Microsoft Teams	Integrates chat, video, and document collaboration.	Deep Microsoft Office integration, enterprise security.	Complex UI, requires Microsoft 365 for full functionality.	Not flexible for small teams, rigid workspace structure.
Discord	Voice, video and text chat for large communities.	Free, persistent voice channels, rich media sharing.	Lacks enterprise features, focus is on community engagement.	No structured workspace, limited project management tools.
Mattermost	Open-source, self-hosted alternative to Slack.	Enterprise security, customizable, DevOps integrations.	Requires technical expertise, less polished UI.	Complex setup, fewer integrations than commercial tools.
Rocket.Chat	Secure, open-source team communication.	Self-hosted for privacy, supports omnichannel communication.	Requires server management, less refined UI.	Not user-friendly for quick deployment, UI needs improvement.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

III. METHODOLOGY

The telemedicine platform under study has been developed primarily using Next JS. This framework was selected for its adaptability, scalability, and compatibility with modern web technologies, making it an ideal choice for creating a robust telemedicine solution.

This section outlines the structured procedures guiding the development of our Real-Time Workspace Communication Platform application.

These procedures encompass planning, structuring, and managing the development process, leading to the creation and maintenance of our system. We adopt an iterative development approach, following the agile methodology to ensure flexibility, adaptability, and responsiveness to changing requirements.

The project planning phase, conducted from January 21, 2025, to January 30, 2025, established the groundwork for the development process. During this period, we defined the project scope, objectives, and requirements, outlining the key features and functionalities of the Real-Time Workspace Communication Platform application. This phase culminated in a comprehensive project synopsis, detailing the project's goals, methodology, and expected outcomes, ensuring a shared understanding among all stakeholders.

The design phase, spanning from February 1, 2025, to February 6, 2025, focused on developing the user interface (UI) and user experience (UX) of the application. Leveraging the Next.js framework, Tailwind CSS, and Recharts library, we created a visually appealing and intuitive interface to enhance user engagement and accessibility. This phase emphasized user-centric design principles to ensure ease of navigation and a coherent visual structure.

The coding phase, from February 7, 2025, to February 27, 2025, involved implementing the core functionalities of the Real-Time Workspace Communication Platform. The development team utilized a chosen technology stack, including Next.js for frontend development and PostgreSQL with Drizzle ORM for backend operations. This phase included tasks such as backend development, API creation, frontend development, database creation, and integration, ensuring seamless interaction between frontend and backend components. Key activities during the coding phase included:

Backend Development: Initializing the server using convex.

API Creation: Developing APIs to connect the backend with the frontend.

Frontend Development: Creating the user interface using Next.js and implementing data visualization with Recharts.

Database Creation and Integration: Setting up the convex database and integrating it with the backend for data storage and retrieval.

The testing phase, scheduled from March 1, 2025, to March 5, 2025, involved rigorous testing procedures to validate the functionality, performance, and reliability of the application. This phase included:

Unit Testing: Ensuring individual modules work as intended.

Integration Testing: Testing the interaction between different modules.

System Testing: Verifying the overall functionality of the application.

Additionally, beta testing was conducted to gather feedback from users and stakeholders, helping identify and address any remaining issues or areas for improvement.

The deployment phase, taking place from March 7, 2025, to March 9, 2025, marked the transition of the Real-Time Workspace Communication Platform application from development to practical use. During this phase, the application was deployed to a cloud-hosted convex database, ensuring accessibility and scalability for users. Continuous monitoring and optimization were conducted to maintain performance and reliability, providing a seamless user experience. By adhering to this structured methodology, the development process was organized and efficient, ensuring each phase was thoroughly planned and executed. This approach led to the successful creation and deployment of the Real-Time Workspace Communication Platform application, aligning with the project's goals and user needs.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

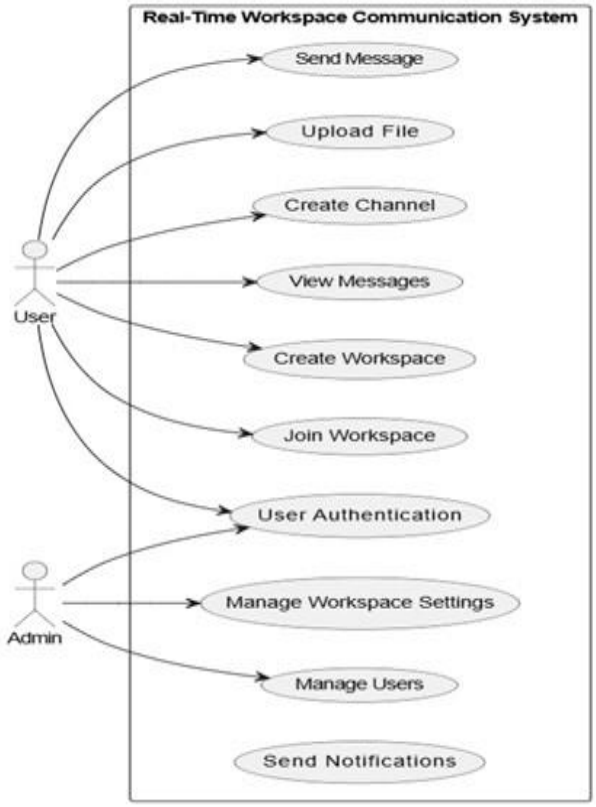


Fig. 1: Use Case Diagram

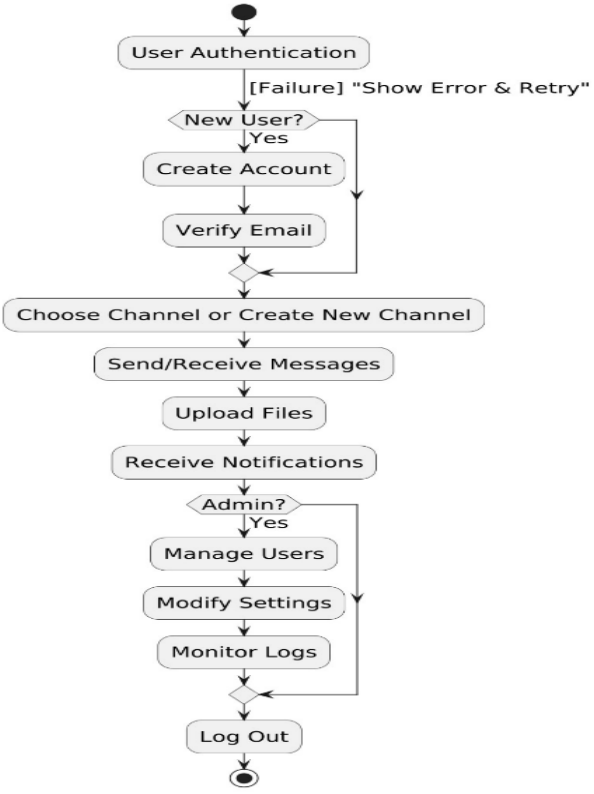


Fig. 2: Flow Chart

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

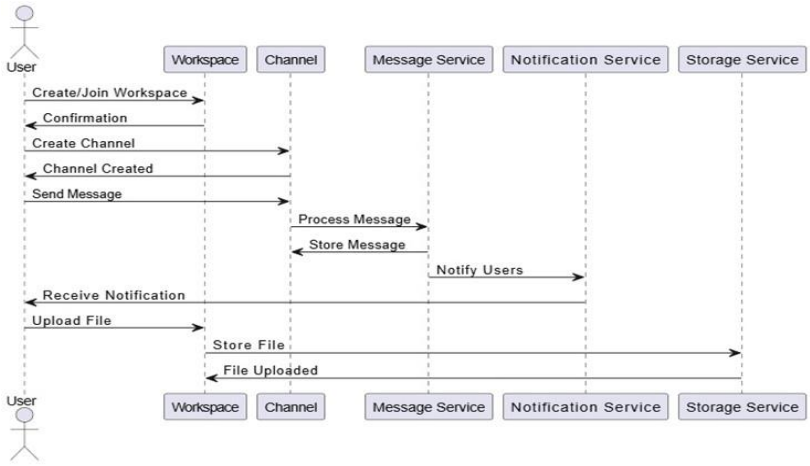


Fig. 3: Sequence Diagram

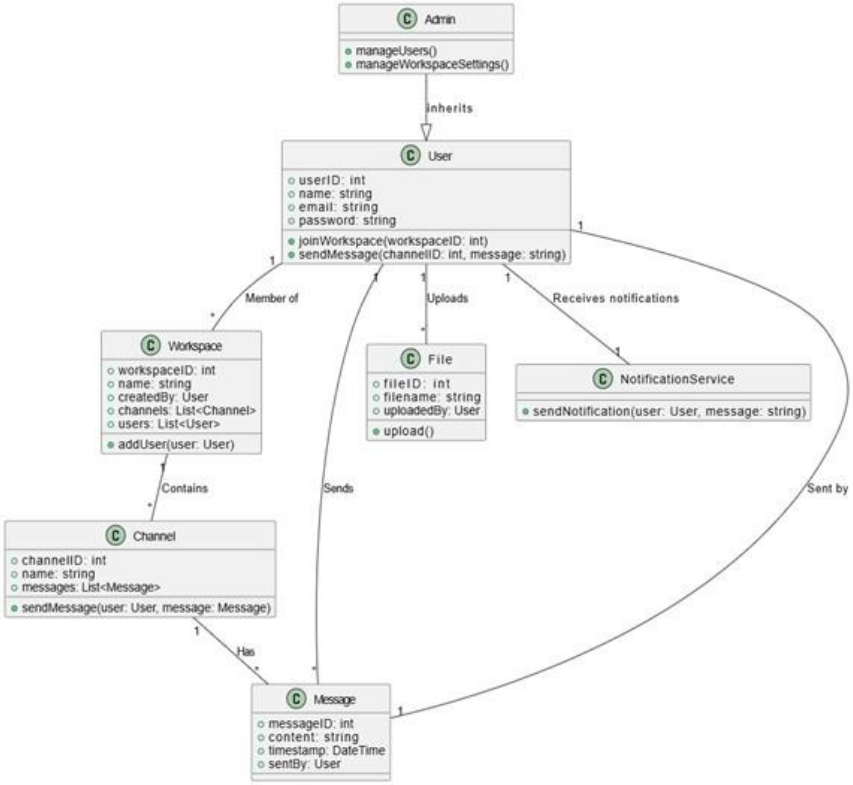


Fig. 4: Activity Diagram

Evaluation and Feedback:

User feedback and testing were integral components of the development process. Continuous evaluation allowed for the identification and rectification of issues, ensuring that the platform meets user expectations and adheres to the goals of accessibility and efficiency.

The development of the Real-Time Workspace Communication Platform built with modern web technologies such as Tailwind CSS & Shadcn UI as a styling solution, Convex for serverless functions as well as database, Next.js, React Query and Zustand for frontend and Auth.js for handling email as well as social login and authorization.

The project is developed using Next.js 14, which provides server-side rendering (SSR), static site generation (SSG), and API handling, ensuring fast and dynamic content rendering. React 18 and React DOM power the frontend, enabling a component-based architecture that improves maintainability and reusability. Zustand is used for state management, offering a lightweight and efficient solution for handling application state.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

For UI design and styling, the project integrates ShadCN UI and Tailwind CSS, enabling a responsive, customizable, and modern user interface. Additionally, Radix UI components such as @radix-ui/react-avatar, @radix-ui/react-dialog, and @radix-ui/react-tooltip enhance user interactions by providing accessible and high-performance UI elements. Lucide React is used for icon rendering, ensuring a visually appealing and modern interface.

The authentication system is implemented using NextAuth v5, allowing for secure, flexible, and provider-based user authentication. @auth/core and @convex-dev/auth further streamline authentication handling and session management. React Verification Input is utilized for implementing multi-step authentication and verification flows, improving security.

For real-time messaging and database operations, the platform uses Convex, a backend-as-a-service solution that supports reactive data queries, ensuring real-time updates. @tanstack/react-query is used for efficient data fetching, caching, and synchronization, optimizing performance for asynchronous operations.

To enhance message formatting and interactivity, the project integrates Quill, a rich text editor that enables users to format messages with ease. Emoji-picker-react adds support for message reactions and emoji selection, improving user engagement. React Resizable Panels is used for creating a flexible and adjustable UI layout, allowing users to customize their workspace as needed.

For date and time management, Day.js provides an efficient and lightweight solution for formatting, parsing, and manipulating date and time information. Clsx and Class Variance Authority are used for conditional class handling, improving styling efficiency and maintainability.

The development process is aided by Figma, VS Code, and Git. Figma is used for design prototyping and UI/UX planning, allowing designers and developers to collaborate efficiently. Visual Studio Code (VS Code) serves as the primary code editor, offering extensions, debugging tools, and an optimized development environment. Git is used for version control, ensuring seamless collaboration, code tracking, and rollback capabilities.

Deployment is managed via Vercel, providing a scalable and high-performance hosting solution optimized for Next.js applications. The backend utilizes a cloud-hosted PostgreSQL database, ensuring data persistence, security, and reliability. PostCSS and ESLint are integrated into the development environment to enforce coding standards, improve styling, and maintain clean, optimized code.

By integrating these technologies, the Real-Time Workspace Communication Platform ensures a secure, scalable, and user-friendly environment for team collaboration, addressing critical challenges in modern workplace communication.

IV. RESULT DISCUSSIONS

This section evaluates the effectiveness, performance, and usability of the Real-Time Workspace Communication Platform, analyzing how well it meets the intended objectives. The platform was assessed based on real-time communication efficiency, user engagement, security, scalability, and overall user experience. Through testing and feedback, it was observed that the platform successfully delivers seamless workspace management, structured discussions, and a secure authentication system, making it a reliable and user-friendly solution for team collaboration.

User Satisfaction: User testing indicated that Shadcn UI and Tailwind CSS provide a clean, modern, and intuitive interface, reducing the learning curve for new users. Features such as message reactions, threaded conversations, and direct messaging received positive feedback for improving engagement and conversation structuring. However, some users expressed a need for additional features like message search, pinned messages, and deeper integration with external tools, which could enhance usability.

Scalability: Deployment on Vercel provides a highly optimized hosting solution, ensuring fast performance. However, as the user base grows, scalability concerns related to database management and API rate limits must be addressed. Implementing load balancing and server optimizations could help maintain performance under heavy user loads.

Performance and functionality: The implementation of Convex for real-time data synchronization ensures that messages, reactions, and workspace updates appear instantaneously across user devices. Next.js and React 18, combined with Zustand for state management, contribute to fast UI rendering and smooth navigation,

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

significantly enhancing user experience. The role-based access control system efficiently restricts permissions, allowing different levels of user management within workspaces. However, extensive testing revealed that system performance may degrade slightly under high concurrent user loads, indicating potential areas for backend optimization and scaling strategies.

Security: The integration of NextAuth.js for authentication ensures secure login mechanisms with support for OAuth providers. The invite system functions as intended, allowing controlled user onboarding. However, further security enhancements, such as two-factor authentication (2FA) and encrypted messaging, could strengthen the platform's privacy and data protection measures.

Limitations:

However, it is essential to acknowledge the limitations of the project. These include:

A. The platform relies on third-party services like Convex and NextAuth.js for real-time updates and authentication, which could cause disruptions if these services experience downtime or changes.

B. The current implementation may face performance bottlenecks as the number of users and data volume increase, requiring further database optimization and server scaling.

C. While offering core messaging capabilities, the platform does not yet support voice/video calls, deep third-party integrations, or AI-powered message summarization, limiting its functionality compared to competitors.

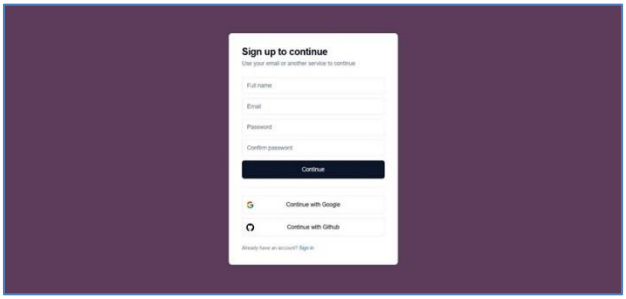
D. Although authentication is implemented, features like end-to-end encryption and two-factor authentication (2FA) are missing, making it less secure for handling sensitive communications.

E. The platform lacks advanced search, pinned messages, and structured message categorization, making it difficult to organize and retrieve important conversations efficiently.

V. FUTURE SCOPE

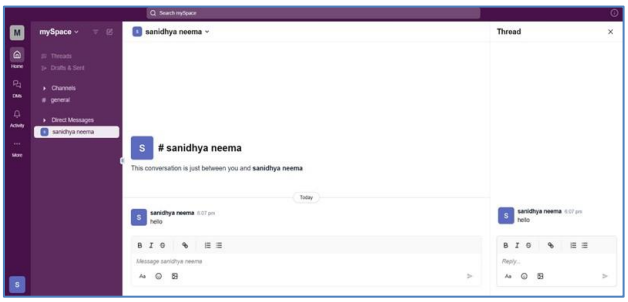
The future scope of the Real-Time Workspace Communication Platform focuses on enhancing scalability, security, and feature richness to meet evolving collaboration needs. Implementing voice and video calling will expand communication capabilities, making it a more comprehensive team collaboration tool. AI-powered features, such as message summarization, smart notifications, and automated task suggestions, can improve productivity and streamline conversations. Advanced search functionalities with filters, pinned messages, and conversation tagging will enhance message organization and retrieval. Strengthening security measures, including end-to-end encryption and two-factor authentication (2FA), will ensure data privacy and protection. To address scalability challenges, optimizing database performance, caching strategies, and load balancing mechanisms will be necessary to handle a growing user base. Additionally, mobile application development will improve accessibility, allowing seamless collaboration across devices. Integration with third-party productivity tools like Google Drive, Slack, and project management software will further enhance interoperability. Continuous user feedback and iterative improvements will be crucial in making the platform a more robust, user-friendly, and scalable solution for modern workspace communication.

Screenshots of the system:

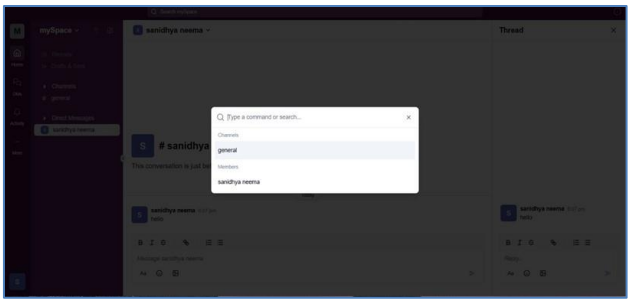


SCREENSHOT 1: Sign up Page

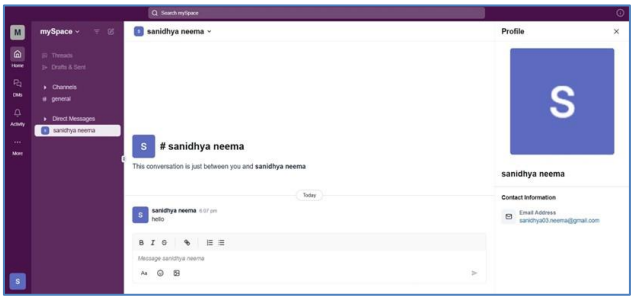
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com



SCREENSHOT 2: DM/Threads page



SCREENSHOT 3: Search Functionality



SCREENSHOT 4: Profile Page

VI. CONCLUSION

In conclusion, the Real-Time Workspace Communication Platform successfully addresses key challenges in team collaboration by providing real-time messaging, structured discussions, role-based access control, and secure authentication. By leveraging modern web technologies, the platform ensures efficient, organized, and scalable communication for teams. While the system meets its core objectives, certain limitations, such as the need for advanced features, enhanced security, and improved scalability, highlight areas for future enhancement. With ongoing development, including AI-powered tools, third-party integrations, and mobile optimization, the platform has the potential to evolve into a comprehensive, high-performance collaboration solution for diverse professional environments.

VII. REFERENCES

[1] Schmid O., Lisowska Masson A., & Hirsbrunner B. (2014). "Real-time collaboration through web applications: an introduction to the Toolkit for Web-based Interactive Collaborative Environments (TWICE)." Personal and Ubiquitous Computing, 18(6), 1201–1211

[2] Agarwal S. (2024). "The Profound Impact of Virtual Communication Technologies on Team Collaboration in the Contemporary Work Environment." International Journal of Embedded Systems and Emerging Technologies, 10(1), 45–60

[3] Bernstein E. S. & Turban S. (2018). "The impact of the 'open' workspace on human collaboration." Philosophical Transactions of the Royal Society B: Biological Sciences, 373(1753), 20170239.