

Vivekanand Education Society's Institute of Technology



Department of Computer Engineering

Group No.: 30

Date :- 2/08/2024

Project Synopsis (2024-25) - Sem V

Commune - Igniting ideas,connecting teams

Prof.Abha Tiwari

Asst. Professor, CMPN

Raheni Ajwani

2022.raheni.ajwani@ves.ac.in

Yash Janyani

2022.yash.janyani@ves.ac.in

Tarun Gulwani

2022.tarun.gulwani
@ves.ac.in

Abstract:

This mini-project aims to develop a clone of the popular team communication platform, Slack. The objective is to provide a cost-effective and customizable alternative that offers essential features such as real-time messaging, channels, direct messaging, file sharing, and notifications. This project will showcase the integration of modern web technologies to create a seamless and efficient communication tool.

By mimicking key functionalities of Slack, this project will demonstrate an understanding of real-time communication, user authentication, scalability, and user experience design. The resulting application will cater to small teams and startups, offering a flexible and robust platform for team collaboration. The development of this Slack clone will delve into various technical aspects, including the use of WebSockets for real-time communication, secure user authentication mechanisms, and responsive user interface design. Additionally, the project will explore how to build a scalable architecture capable of handling increasing user loads and data traffic.

Introduction:

In the digital age, effective communication tools are vital for team collaboration and productivity. Platforms like Slack have revolutionized how teams communicate, offering a suite of features that facilitate seamless interaction and information sharing. However, the cost associated with such platforms can be a barrier for small teams and startups. Moreover, the need for customization to fit specific organizations by standard solutions.

This project involves creating a Slack clone, a web-based application designed to replicate the core functionalities of Slack. The clone will enable real-time messaging, creation and management of channels, direct messaging, file sharing, and notifications. The aim is to provide a flexible and scalable communication tool that can be tailored to the unique needs of different organizations. By focusing on both the technical implementation and user experience, this project aims to deliver a robust communication platform that not only addresses the cost and customization issues of existing tools but also enhances team productivity and collaboration. The Slack clone will be an open-source project, making it accessible and modifiable for different use cases, thus providing a valuable resource for small teams and startups.

Problem Statement:

The problem addressed by this project is the need for an efficient and user-friendly communication tool for teams. While Slack offers a comprehensive solution, it is often costly for small teams and startups. Additionally, some organizations require customized solutions that align with their specific needs and workflows. The problem can be summarized as follows:

High Cost: Existing communication tools like Slack can be expensive for small teams and startups.

Lack of Customization: Many existing solutions do not offer sufficient customization options to meet the specific requirements of different organizations.

Complex User Interfaces: Some communication platforms have steep learning curves, making it difficult for new users to quickly become proficient.

Features comparison table:

Features	Commune	Slack	Microsoft Teams	Discord	Rocket chat
Customization	Yes	No	No	No	No
Status Update	Yes	Yes	No	Yes	Yes
Guest Account	Yes	No	No	Yes(Limited)	No
Messages and notification limit	Unlimited	Unlimited	Unlimited	Unlimited	Limited

Proposed Solution:

To address the identified problem, this project proposes the development of a Slack clone that offers the following solutions:

Cost-Effective Communication Tool: Open-Source Solution: By creating an open-source Slack clone, small teams and startups can have access to a powerful communication tool without incurring high costs.

Free Basic Version: Providing a free basic version with essential features can help small teams get started without financial barriers.

Customizable Features: Modular Architecture: Designing the application with a modular architecture will allow organizations to easily add or remove features as needed.

Custom Themes: Offering the ability to customize the look and feel of the platform to match the branding of different organizations.

Plugin Support: Enabling third-party plugins and integrations to extend the functionality of the platform.

Core Functionalities: Real-Time Messaging: Implementing real-time messaging with support for text, images, and file attachments to facilitate instant communication.

Direct Messaging: Providing a direct messaging feature for private conversations between team members.

Notification System: Implementing a notification system to keep users informed about important updates and messages.

Methodology:

1. Design Thinking principles to empathize with users, define problems, ideate solutions, prototype, and test.
2. Continuous Integration and Deployment (CI/CD) pipelines to automate testing, building, and deployment processes.
3. Microservices Architecture: Architecture is based on microservices, which allows for scalability, flexibility, and independent development of individual components.
4. Creating a user-friendly and intuitive interface, prioritizing user experience and feedback.

Hardware , Software and tools Requirements:

Backend:

1. Programming Language: Node.js (or alternative: Python, Ruby, or Go)
2. Framework: Express.js (or alternative: Django, Ruby on Rails, or Gin)
3. Database: MongoDB (or alternative: PostgreSQL, MySQL)
4. Real-time Communication: WebSockets, Pusher, or Firebase Realtime Database
5. Authentication: OAuth, Auth0, or Okta

Frontend:

1. Programming Language: JavaScript (with HTML/CSS)
2. Library: React.js
3. UI Components: Material-UI (or alternative: Bootstrap, Tailwind CSS, or Semantic UI)
4. State Management: Redux

Additional Tools:

1. Version Control: Git
2. Package Manager: npm (or alternative: yarn)

Proposed Evaluation Measures:

Functionality (40%): Verify user authentication, channel management, messaging, file sharing, and real-time updates through test scenarios and user feedback.

Performance (20%): Measure page load times, message delivery speed, and API response times using load testing tools.

User Experience (20%): Conduct usability tests and gather user feedback on interface design, messaging clarity, and ease of management.

Security (10%): Perform security audits and penetration tests to ensure data encryption, secure storage, and input validation.

Scalability (5%): Test app scalability under increased user load and review architecture for resource efficiency and feature extensibility.

Code Quality (5%): Conduct code reviews, use static analysis tools, and implement automated testing to maintain clean, readable, and maintainable code.

Conclusion:

In conclusion, our app will demonstrate a functional and user-friendly communication platform for teams and communities. By leveraging React JS for the frontend and Node JS for the backend, we will create a scalable and efficient application that meets the core requirements of a modern communication tool.

With features like real-time messaging, file sharing, and direct messaging. While there's always room for improvement and expansion, our app serves as a great starting point for further development and customization.

References:

1. P. Chatterjee, K. Damevski , N. A. Kraft and L. Pollock, "Software-related Slack chats with disentangled conversations", *Proceedings of MSR 2020 (International Conference on Mining Software Repositories)*, pp. 588-592, 2020.
2. M. Elsner and E. Charniak, "You talking to me? A corpus and algorithm for conversation disentanglement", *Proceedings of ACL-HLT 2008 (Association for Computational Linguistics: Human Language Technologies)*, pp. 834-842, 2008.
3. P. Chatterjee, K. Damevski, L. Pollock, V. Augustine and N. A. Kraft, "Exploratory study of Slack Q&A chats as a mining source for software engineering tools", *Proceedings of MSR 2019 (International Conference on Mining Software Repositories)*, pp. 490-501, 2019.
4. D. Shen, Q. Yang, J.-T. Sun and Z. Chen, "Thread detection in dynamic text message streams", *Proceedings of SIGIR 2006 (International Conference on Research and Development in Information Retrieval)*, pp. 35-42, 2006.
5. V. Stray and N. B. Moe, "Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack", *Journal of Systems and Software*, vol. 170, pp. 110717, 2020.
6. M. Raglianti, R. Minelli, C. Nagy and M. Lanza, "Visualizing Discord servers", *Proceedings of VISSOFT 2021 (Working Conference on Software Visualization)*, pp. 150-154, 2021.
7. Rocket chat app