

25/02/2025

LiveDocs: A Peer-to-Peer Real-Time Collaborative Document Editor Using WebRTC and CRDTs

Team Members:

Charankumar E G D [921321205029]

Arunprasad S [921321205015]

Dharani Dharan R [921321205032]

Mentor:

Mrs. A. Sangeetha, M.E., Ph.D.,
Assistant Professor - Information Technology

Abstract

LiveDocs is a decentralized, real-time collaborative document editor that enables seamless collaboration without centralized servers. It leverages WebRTC for peer-to-peer (P2P) communication and Yjs (CRDTs) for distributed synchronization, ensuring low-latency collaboration, fault tolerance, and scalability. Key features include real-time editing, role-based permissions, instant notifications, and JWT-based authentication for secure access. By eliminating centralized infrastructure, LiveDocs reduces server costs, enhances scalability, and improves fault tolerance, making it a powerful alternative for teams and enterprises needing real-time collaboration.

Problem Statement

Traditional real-time document editors face significant challenges due to high latency and dependence on centralized servers for synchronization and storage. This reliance introduces scalability limitations, single points of failure, and potential privacy risks. As user demand increases, latency issues degrade the collaborative experience, making real-time editing less efficient. To address these challenges, a decentralized approach is required to ensure low-latency, fault-tolerant, and scalable real-time document collaboration without reliance on a central server.

Objective

Develop a decentralized editor to eliminate reliance on central servers and enhance scalability.

Leverage WebRTC-based P2P communication for low-latency and efficient real-time collaboration.

Integrate Yjs (CRDTs) to enable seamless, conflict-free data synchronization across users.

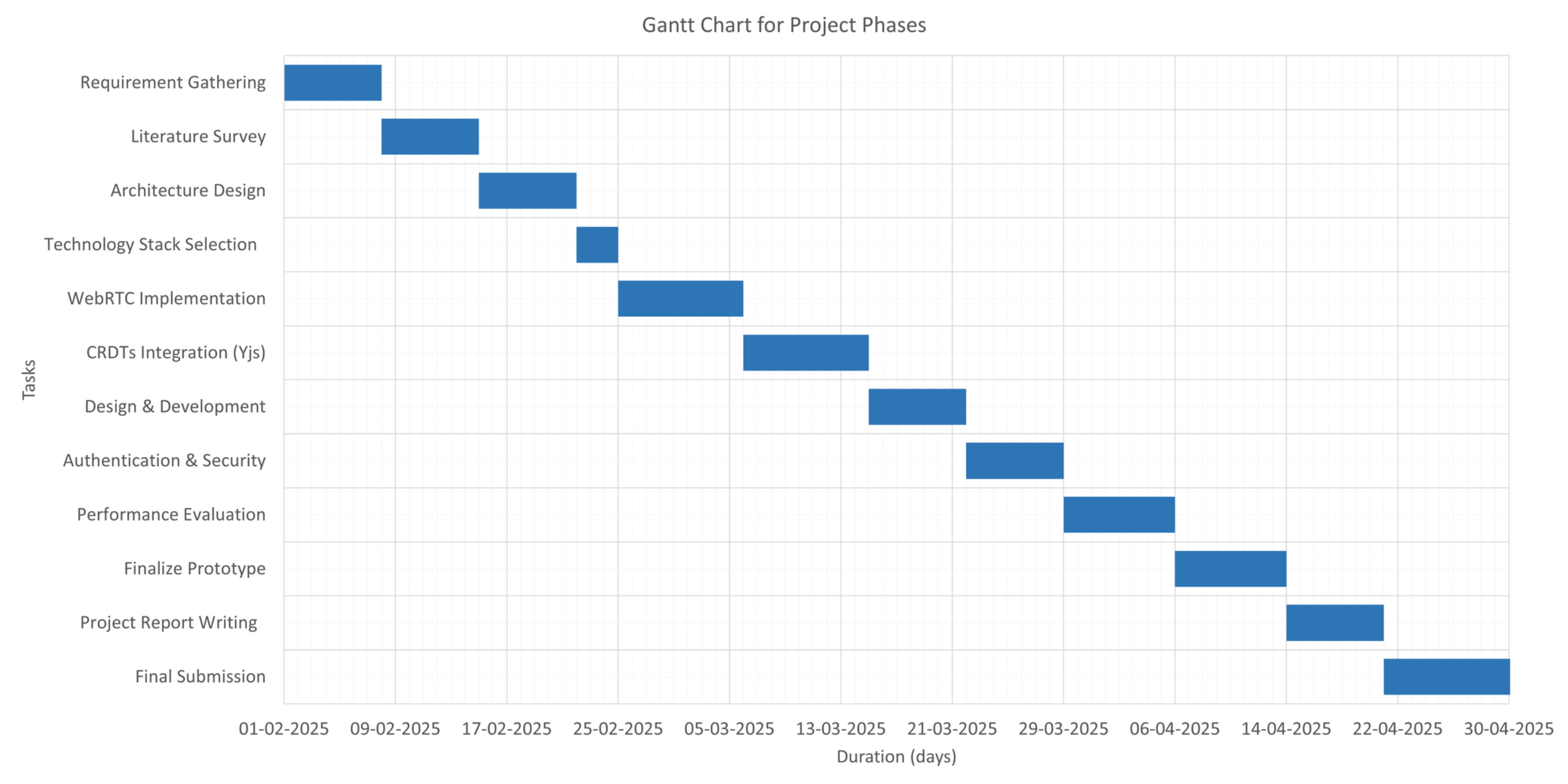
Provide an intuitive UI with modern design for a smooth and responsive editing experience.

Ensure security with JWT authentication, role-based access, and real-time notifications.

Literature Survey

Title	Author(s) & Year	Approach	Limitations	Link
Performance of real-time collaborative editors at large scale: User perspective	Quang-Vinh Dang, Claudia-Lavinia Ignat - 2016	Evaluates large-scale collaborative editors using user interaction data and system logs to assess responsiveness and consistency.	Performance degradation observed under high concurrency, limited consideration for network partition scenarios.	https://ieeexplore.ieee.org/document/7497258
A Personal Distributed Real-time Collaborative System	Michalis Konstantopoulos, Nikos Chondros, Mema Roussopoulos - 2020	A distributed peer-to-peer (P2P) collaborative editing system with decentralized storage, reducing reliance on central servers.	Consistency management becomes complex when multiple users edit simultaneously; relies heavily on stable peer connectivity.	https://ieeexplore.ieee.org/document/10117106
Collaborative Code Editors - Enabling Real-Time Multi-User Coding and Knowledge Sharing	Khushwant Viridi, Anup Lal Yadav, Azhar Ashraf Gadoo, Navjot Singh Talwandi - 2023	Examines collaborative code editors, focusing on WebSocket communication, OT algorithms, and conflict resolution, with a prototype evaluation and comparison to existing platforms.	Persistent issues like latency, synchronization, and conflict resolution remain, requiring further innovation for larger-scale effectiveness.	https://ieeexplore.ieee.org/document/10325722
SecureC2Edit: A Framework for Secure Collaborative and Concurrent Document Editing	Shashank Arora, Pradeep K. Atrey - 2023	Proposes a secure framework for concurrent collaborative editing, integrating cryptographic controls to ensure content integrity.	Increased processing overhead due to encryption; real-time performance is affected in low-bandwidth environments.	https://ieeexplore.ieee.org/document/9036818

Gantt Chart



Proposed Methodology

Use Next.js for the frontend with a custom collaborative editor.

Apply Yjs (CRDTs) for real-time updates and conflict resolution.

Leverage WebRTC for direct peer-to-peer communication and low latency.

Implement JWT authentication for secure access control.

Store documents in PostgreSQL for reliable backups and fast access.

Enable collaboration features like presence, comments, roles, and notifications.

Optimize WebRTC and CRDTs for low latency, scalability, and efficient sync.

Tools & Frameworks



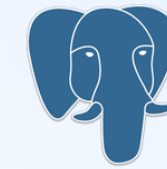
Next.js



TypeScript



Tailwind CSS



PostgreSQL



Prisma



WebRTC



WebSocket



CRDTs (Yjs)



JWT



Git

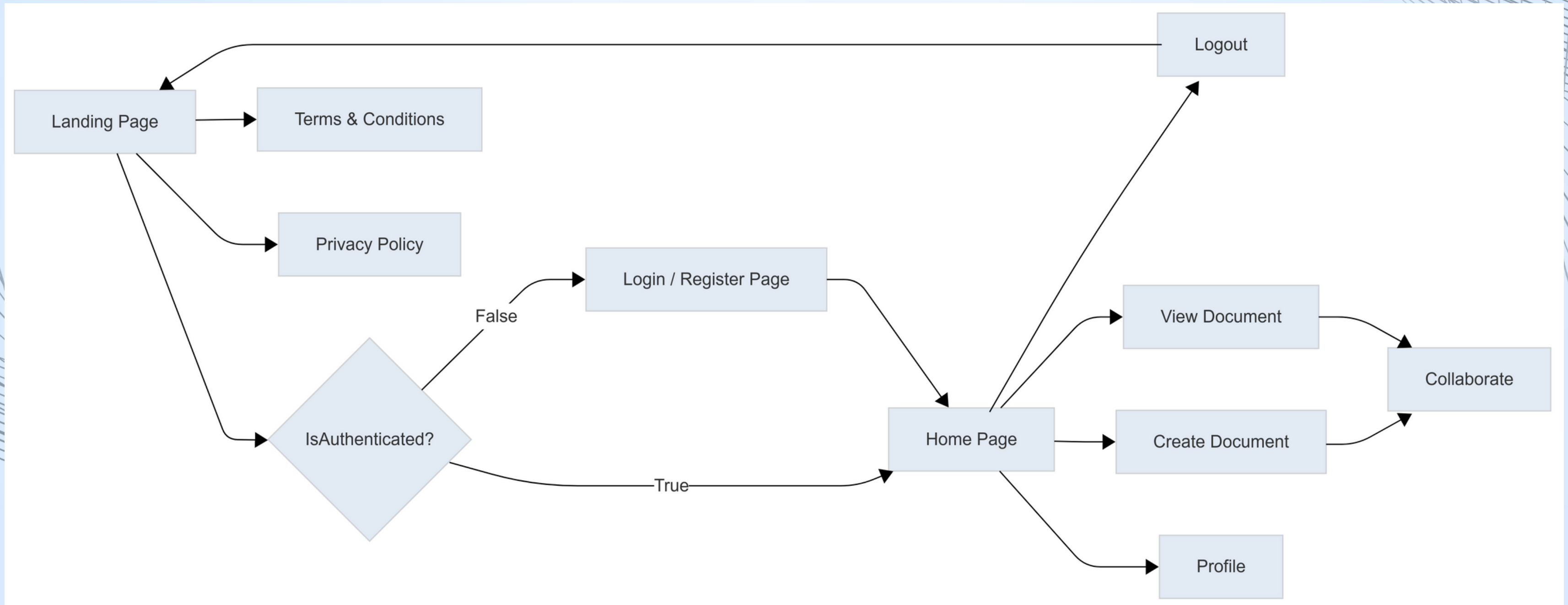


GitHub



Postman

Flow Chart



Prototype

The background of the slide features a light blue gradient. In the top-left corner, the word "Prototype" is written in a bold, black, sans-serif font. The bottom-left and top-right corners are decorated with thin, grey, wavy lines that create a sense of motion and depth.

The background features a light blue gradient with abstract, flowing, wavy lines in a slightly darker shade of blue, creating a sense of movement and depth. These lines are concentrated on the left and right sides, framing the central text.

Thank you