Besly John Jacob, 2026, 14:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

CodeCollab: Real Time Collaborative Code Editing

Besly John Jacob, Aparna Ashok, Fathima Noora, Lakshmi H, Dr. Rani Saritha R
Department Of Computer Application Saintgits College Of Engineering Kottayam, Kerala ,India

Abstract- Collaborative programming has become an essential practice in both academic and professional software development; however, existing tools often require multiple platforms for editing, communication, and execution, leading to workflow interruptions and increased setup effort. CodeCollab is proposed as a web-based real-time collaborative coding platform that brings these functionalities into a single, integrated environment. The system supports simultaneous code editing with live cursor visibility, user presence tracking, role-based permissions, contextual chat, and in-editor code execution. The application is developed using React, TypeScript, and Vite on the frontend, while Firebase is used for authentication, real-time synchronization, and data persistence. Code execution is enabled through integration with the Piston API, allowing users to compile and run programs without local configuration. Experimental evaluation shows that the platform provides stable synchronization, minimal latency, and an intuitive collaborative experience. CodeCollab reduces configuration overhead and improves productivity, making it suitable for classrooms, workshops, and distributed development teams.

Keywords: Real-time collaboration, collaborative code editor, web-based IDE, Firebase, rolebased access control, code execution, pair programming.

I. INTRODUCTION

In modern software development and computer science education, collaboration plays a critical role in improving code quality, learning outcomes, and development speed. Despite this, collaborative programming is often affected by fragmented tool usage, complex version control mechanisms, and the absence of integrated communication features. Developers and students frequently rely on separate applications for writing code, discussing ideas, and executing programs, which results in frequent context switching and reduced efficiency. To overcome these challenges, this paper presents CodeCollab, a real- time collaborative code editing platform accessible through a web browser.

The proposed system combines code editing, communication, and CodeCollab allows multiple users to edit the same codebase simultaneously while observing real-time cursor movement, presence indicators, and synchronized file updates. The platform incorporates role-based access control, project invitations, and integrated chat to support structured and secure collaboration. Additionally, code execution is provided through the Piston API, enabling users to compile and test programs directly within the editor. This paper discusses the system design, implementation details, and performance evaluation of CodeCollab.

Proposed System

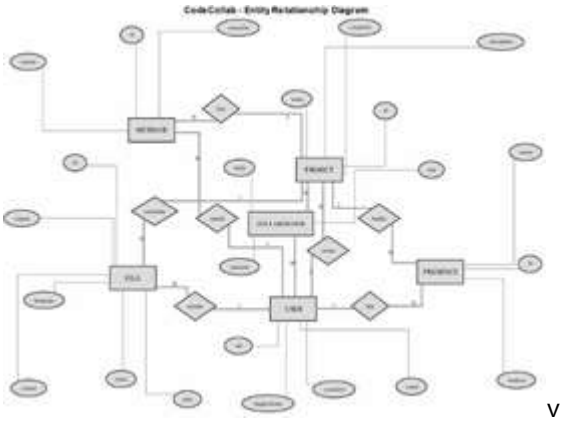
CodeCollab is implemented as a full-stack web application designed to support seamless real-time collaboration. The frontend is developed using React and TypeScript, providing a responsive and interactive coding interface. A Node.js and Express-based backend supports execution requests and auxiliary system operations, while Firebase services handle authentication, data storage, and real-time synchronization. The platform enables concurrent code editing, allowing multiple users to work on shared files while observing instant updates, cursor positions, and active participants.

A structured project and file management system allows users to create, organize, and maintain files within collaborative workspaces. Access control is enforced using a role- based permission model consisting of owner, editor, and viewer roles, ensuring secure and controlled collaboration. To facilitate communication, an in- context chat feature is integrated directly into the coding environment, enabling users to discuss changes without leaving the editor. Code execution is supported through integration with the Piston API, which provides secure, sandboxed execution for multiple programming languages. Firebase Authentication and Firestore are used to manage user identities, store project data, and propagate updates in real time. Together, these components form a reliable

and efficient collaborative coding system suitable for educational and team-based development scenarios.

System Architecture

The architecture of CodeCollab follows a modular design that clearly separates frontend, backend, and real-time data services. The frontend layer, built using React and TypeScript, is responsible for rendering the code editor, handling user interactions, and displaying collaboration indicators. The backend layer, implemented with Node.js and Express, processes execution requests and supports system-level operations. Firebase services play a central role by managing user authentication, storing project and file data, and synchronizing updates across connected clients in real time. Communication between components is achieved through lightweight APIs and Firestore listeners, enabling efficient data flow, scalability, and secure collaboration among users.



v

Fig 1: System Architecture

Result and Discussion

Experimental testing indicates that CodeCollab performs reliably under typical collaborative usage conditions. File synchronization and cursor updates occur with minimal delay, even when multiple users edit the same document simultaneously. The integrated code execution feature produces consistent results across supported programming languages, enabling rapid testing and iteration. Chat messages and presence indicators are updated in real time, contributing to an effective collaborative experience. Overall, the system demonstrates stable performance, responsive interaction, and dependable synchronization, confirming its suitability for classroom environments and team-oriented development tasks.

Limitations

The effectiveness of real-time collaboration in CodeCollab is influenced by network connectivity, and users with limited bandwidth may experience synchronization delays. Executing large or computationally intensive programs through the Piston API can result in longer execution times during periods of high demand. Certain advanced features commonly available in desktop-based IDEs, such as comprehensive debugging tools and full version control integration, are currently limited. Dependence on third-party services may also lead to temporary disruptions due to service outages or rate restrictions.

Future Enhancement

Future development of CodeCollab may include integration with distributed version control systems such as Git, enhanced debugging support, and improved offline collaboration through advanced synchronization techniques. The inclusion of AI-based assistance for code explanation, error detection, and automated testing could further enhance usability. Additional improvements may involve mobile platform support, multilingual interfaces, advanced analytics for instructors, and stronger security mechanisms such as multi-factor authentication and activity monitoring.

II. CONCLUSION

CodeCollab provides a unified platform that integrates real-time collaborative coding, project management, communication, and execution into a single web-based environment, reducing the need for multiple disconnected development tools and improving teamwork efficiency in both academic and professional settings. By combining synchronous editing, organized file handling, in-context chat, and sandboxed code execution, the system supports smooth collaboration and eliminates common workflow barriers faced by distributed teams and learners. Built using React,

TypeScript, Firebase, and a Node.js/Express backend, CodeCollab demonstrates the advantages of a scalable, modern architecture that ensures secure authentication, reliable real-time synchronization, and structured project data management. The integration of the Piston API further streamlines coding activities by allowing users to run programs without local setup, enhancing accessibility and productivity. Although the platform delivers strong functionality, future improvements could include deeper version control support, advanced debugging tools, offline editing, enhanced analytics, and expanded AI-assisted coding features. Overall, CodeCollab provides a robust foundation for modern collaborative development, showcasing how cloud- backed services and browser-based tools can significantly enhance usability, reduce setup overhead, and improve learning and development outcomes.

REFERENCES

1. G.Sun, C. Chen and K. Zhang, "Real-time collaborative editing: a systematic literature review, " Journal of systems and software, Vol.168, 2020.
2. "Firebase documentation ,"Google.[Online]. Available:
3. "Piston API documentation ,"Piston [Online].Available
4. React documentation ,Facebook Open Source[Online] Available: