IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

The Real-Time Collaborative Code Editor

L Abhishek[*1] and Prof. Sandeepa K M[*2]
[*1] Student, Department of MCA
[*2]Assistant Professor, Department of MCA
Maharaja Institute of Technology Mysore, Karnataka, India

Abstract: *Real-time collaboration is essential for today's software development. Developers can effectively work together regardless of where they are located. In this project, CodeSync as real-time collaborative coding editor is developed which allows the users to co-edit the code at the same time with synchronized cursor activity and changes pushed in an instant.*
The proposed system is developed with React.js along with Monaco Editor as front-end components. Node.js with Express.js as backend components, Socket.IO for real-time communication, MongoDB for storing data, and Redis for storing sessions and broadcasting. The work has adopted the iterative development methodology which is that at first implementation of front-end part followed by implementation of real time part and at last implementation of back-end part followed by extensive evaluation. Through extensive experiment of proposed system we show that this application leads to effective speedy synchronization, secure real-time collaboration with session control and data persistence that can have awesome experience and the file sharing, synchronization, screen share can be eradicated while doing remote collaborative programming with the implemented application and thus proposed system demonstrate that a lean collaborative coding environment can develop online for students, teacher and pair and code sharing.

Keywords: *Real-time collaboration*.

1. INTRODUCTION

Building software is generally a team effort, where program developers will often be in different geographical locations, resulting in a development by a remote-working team. There are plenty of good versioning-control tools, with Git and GitHub being two readily available examples, that cater towards managing changes within a piece of code but aren't well suited for code collaboration over the web; file-sharing or screen sharing typically results in a higher risk of version conflicts, thus lower productivity.
Recently, researchers began to explore the viability of collaborative document editing over the web by utilising technologies such as WebSockets, or Operational Transformation (OT).
However, lightweight collaborative editors for students and smaller, casual teams have not yet been put into practice that allow for collaboration inside a browser. CodeSync has been created as a web application where multiple people simultaneously edit the same body of code. It was built utilising the React.js, Monaco editor, Node.js, Express.js, Socket.IO, MongoDB and Redis stacks to deliver a rapid communication protocol, robust data persistence, and a pleasant coding experience.

METHODOLOGY

This research focuses on developing **CodeSync**, a real-time collaborative code editor that enables multiple users to edit source code simultaneously through a web browser. The system is designed using a client-server architecture to provide instant synchronization, low-latency communication, and reliable data storage. Modern web technologies are integrated to create a scalable and user-friendly collaborative coding environment.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

System Architecture

The application follows a client-server architecture consisting of a React.js frontend, a Node.js and Express.js backend, and Socket.IO for real-time communication. MongoDB is used for persistent data storage, while Redis manages active sessions and message broadcasting.

Real-Time Collaboration

Real-time synchronization is achieved using Socket.IO and WebSockets. Any code changes made by one user are instantly transmitted to all connected users. Live cursor tracking and room-based collaboration allow multiple participants to work together without manual refreshing.

Data Storage and Testing

MongoDB stores user, session, and code information securely, while Redis improves performance by handling temporary session data. The system was tested for functionality, real-time synchronization, performance, usability, and security to ensure reliable collaborative coding.

MODELING AND ANALYSIS

The proposed **CodeSync** system is designed using a client-server architecture to support real-time collaborative programming. It integrates modern web technologies to provide fast, reliable, and synchronized code editing among multiple users. The architecture ensures efficient communication, secure data management, and a responsive user experience.

A. System Model

The **CodeSync** application consists of four major components that work together to enable collaborative coding. The **frontend** is developed using **React.js** and **Monaco Editor**, providing an interactive interface with syntax highlighting and live code editing. The **backend** is built using **Node.js** and **Express.js**, which handle user authentication, project management, and server-side operations. **Socket.IO** enables real-time synchronization of code changes and cursor positions between connected users. **MongoDB** stores user information, project details, and source code, while **Redis** manages active sessions and improves real-time message delivery.

B. Materials and Technologies Used

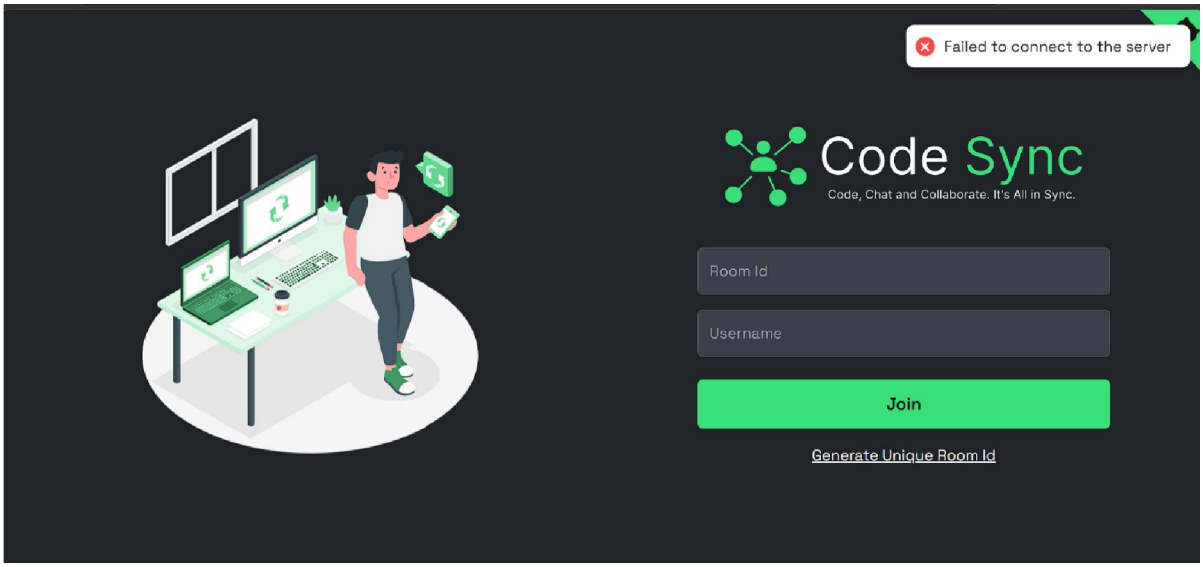
The following technologies were used to develop the proposed system:
React.js – Used to build the responsive and interactive user interface.
Monaco Editor – Provides an advanced code editor with syntax highlighting and auto-completion.
Node.js – Executes server-side application logic.
Express.js – Handles routing, APIs, and backend services.
Socket.IO – Enables real-time communication and live code synchronization.
MongoDB – Stores user accounts, project details, and source code.
Redis – Manages active sessions and improves application performance through caching.

C. Analysis

The developed system was evaluated based on synchronization speed, responsiveness, data consistency, and usability. Functional testing confirmed that multiple users can edit code simultaneously without conflicts. Performance analysis showed minimal latency during real-time updates, while MongoDB and Redis ensured reliable data storage and efficient session management. Overall, the proposed architecture provides a scalable, efficient, and reliable solution for real-time collaborative software development

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

SNAPSHOTS



II. RESULTS AND DISCUSSION

The proposed **Real-Time Collaborative Code Editor** was successfully developed and tested to evaluate its functionality, performance, and usability. The application enables multiple users to edit the same source code simultaneously through a shared coding session. During testing, all code modifications were synchronized instantly among connected users using **Socket.IO**, ensuring a smooth collaborative experience with minimal latency.

The system was evaluated under different scenarios, including multiple user connections, simultaneous code editing, session management, and data storage. The application performed reliably without synchronization conflicts, while **MongoDB** provided secure storage of user and project information. **Redis** improved the efficiency of session handling and real-time message delivery, resulting in faster communication between connected clients.

III. CONCLUSION

CodeSync is a successful development of real-time collaborative code editor implemented through browser, where multiple users can contribute and edit source code together within the collaborative space. By using technology such as, React.js, Node.js, Express.js, Socket.IO, MongoDB, and Redis CodeSync will enable reliability of data synchronisation, securing session and data efficiently storing it. Thus the developed app provides better teamcollaboration experience by nullifying the effort of sharing file and delay of communication in coding. Overall CodeSync can be used to establish productive session and be utilized as a pair programming tool, coding assessment tool, collaborative programming and in remote programming lessons.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest gratitude to our Internal guide Prof. Sandeepa K M, Assistant Professor, Department of MCA, Maharaja Institute of Technology Mysore without whose constant motivation, guidance and support, this endeavour would not have been possible. I also extend my sincere appreciation to our External guide Mr. Madhu M, Project Manager, Torus Solutions, Mysuru, for your valuable inputs and support throughout the internship. I am also thankful to the Department of MCA and the Management of Maharaja Institute of Technology Mysore for giving the required infrastructure to complete this project. In addition, I also thank my family members and my friends for standing by my side all throughout.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

REFERENCES

[1]. C. A. Ellis and S. J. Gibbs, "Concurrency Control in Groupware Systems," in *Proc. ACM SIGMOD*, 1989, pp. 399–407.

[2]. M. Ressel, D. Nitsche-Ruhland, and R. Gunzenhäuser, "An Integrating, Transformation-Oriented Approach to Concurrency Control and Undo in Group Editors," in *Proc. ACM CSCW*, 1996, pp. 288–297.

[3]. Imine, "Flexible Concurrency Control for Real-Time Collaborative Editors," in *Proc. 28th Int. Conf. Distributed Computing Systems Workshops (ICDCSW)*, Beijing, China, 2008, pp. 423–428, doi: 10.1109/ICDCS.Workshops.2008.91.

[4]. R. Li and D. Li, "A New Operational Transformation Framework for Real-Time Group Editors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 3, pp. 307–319, 2007