iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72014

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Real-Time Collaborative Code Editor: A Web based Synchronous Programming Platform

Mohd Saad Ansari[1], Soham Babdi[2], Vidhee Bandkar[3], Ansh Mishra[4], Prof. Vijay Jumb[5]
[1, 2, 3, 4]*Students,* [5]*Professor, Department of Computer Engineering, Xavier Institute of Engineering, Mumbai, India*

Abstract: ***As collaborative development gains prominence in both academic and professional software environments, the need for intuitive, accessible, and real-time programming platforms becomes essential. This paper presents a minimalistic, browser-based code editor that enables synchronous programming collaboration through live code sharing and execution capabilities. Built using the MERN (MongoDB, Express.js, React.js, Node.js) stack and Socket.IO for real-time communication, the platform allows multiple users to write, view, and execute code together seamlessly. Designed with simplicity and performance in mind, the system enables peer coding, interviews, and learning scenarios with instant feedback and interactive features. This paper details the architectural framework, user-centric interface, and implementation of the proposed system, supported by a review of existing tools and their limitations.***

Keywords: Real-Time Collaboration, Web IDE, Code Synchronization, MERN Stack, Online Compiler, Collaborative Programming

I. INTRODUCTION

Software development today is rarely a solitary activity. With the increasing adoption of remote work, open-source collaboration, and online education, developers often find themselves in need of tools that support live, synchronous coding. However, traditional IDEs are often designed for solo use, and collaboration typically requires complex setups, third-party integrations, or external communication channels. To bridge this gap, we propose a lightweight and easily accessible real-time collaborative code editor, deployable in any modern browser.

The platform allows developers, students, and interviewers to work on a shared codebase simultaneously, with changes reflected across all clients in real time. Moreover, it includes an embedded compiler for languages like C, enabling users to write and execute code without leaving the interface. By eliminating installation barriers and enabling multi-user interactions, the system offers a scalable and inclusive coding environment.

II. LITERARY SURVEY

Various platforms attempt to provide collaborative environments for coding. Tools like Replit, CodePen, and Visual Studio Code Live Share offer some level of collaboration, yet come with trade-offs. Replit, while comprehensive, requires user registration and suffers from interface clutter for users seeking simplicity. CodePen is tailored for front-end prototyping and lacks multi-language support and real-time chat integration. VS Code Live Share requires software installation, extensions, and often doesn't cater well to beginners.

Previous research also highlights the difficulty in managing consistency in real-time editors, with approaches like Operational Transformation (OT) and Conflict-Free Replicated Data Types (CRDTs) being explored to ensure consistency across clients. However, many open platforms avoid this level of sophistication due to performance trade-offs.

Our platform simplifies this problem space by implementing Socket.IO for real-time synchronization and targeting straightforward user experiences, particularly useful in classrooms, coding bootcamps, and interviews.

III. METHODOLOGY

A. System Overview

The collaborative code editor functions as a single-page application accessible through any modern web browser. The system allows users to either create or join existing rooms, enter code collaboratively, and compile it through an integrated backend compiler API. The architecture enables live synchronization, chat support, and output display, all in one interface.

B. System Architecture and Technologies

1) Frontend: Built with React.js to manage the user interface dynamically. It supports syntax highlighting, code editing, and user cursors.
2) Backend: Powered by Node.js and Express.js, handling authentication, socket events, and routing.
3) Database:MongoDB stores session metadata, user info, and project history.
4) Real-Time Sync:Socket.IO enables real-time bidirectional communication between clients and server.
5) Code Execution: A REST-based code execution API allows users to compile and run code (currently C).

C. Key Modules and Process Flow

1) Homepage and Room Management: Users can create or join a room using unique IDs.
2) Editor Interface: All clients in a room can simultaneously edit code with changes reflected live.
3) Compiler Integration: Upon clicking "Run", the code is sent to a backend server or third-party compiler API and the result is shown in a shared output panel.
4) Chat Box: Supports real-time textual communication among participants.
5) Error Handling: The editor ensures graceful handling of disconnections, partial updates, and compiler failures.

D. System Requirements

1) Hardware: At least 4GB RAM, standard browser compatibility, and an active internet connection.
2) Software Stack: React.js, Node.js, Express.js, MongoDB, Socket.IO, Docker (optional for compiler API containerization).

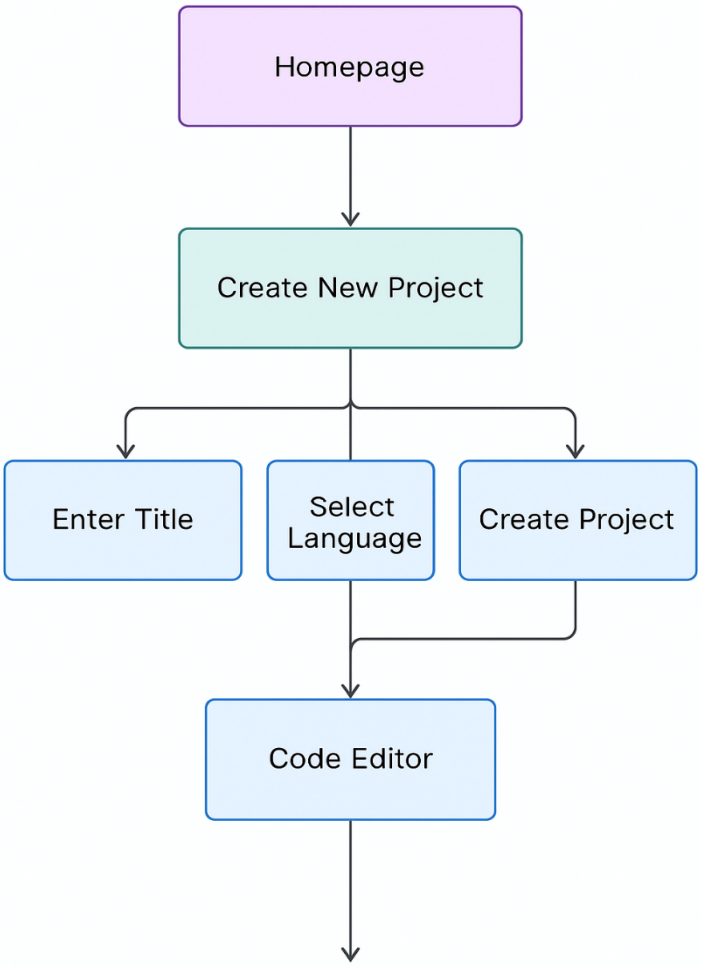


Figure 1: Architecture Diagram

IV. CONCLUSION

The platform was tested with multiple users joining the same session across different machines and networks. The average sync latency observed was below 150ms, even with concurrent editing. Testers praised the minimal interface, ease of entry (no installation required), and seamless collaboration. The live execution feature performed consistently across standard C programs, returning output within 2 seconds.

Experiments also included simulated network drops to test fault recovery. The system successfully retained last-synced states and re-established connections without data loss. These outcomes affirm the viability of the platform for practical collaborative use.

This project demonstrates a functional, real-time, collaborative coding platform that balances performance, usability, and technical depth. By focusing on accessibility and minimalism, it empowers users to engage in synchronous code editing and execution without software overhead. The project is particularly valuable for remote instruction, pair programming, and coding interviews.

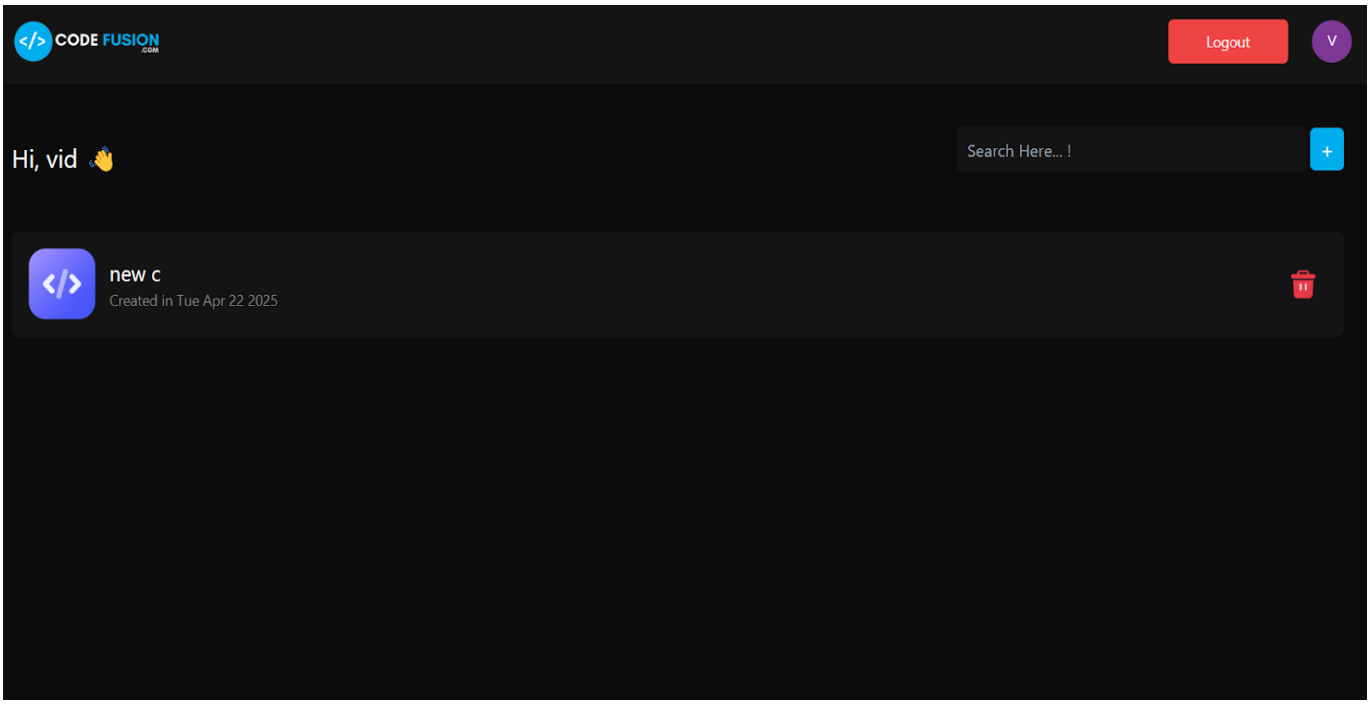


Figure 2: Homepage

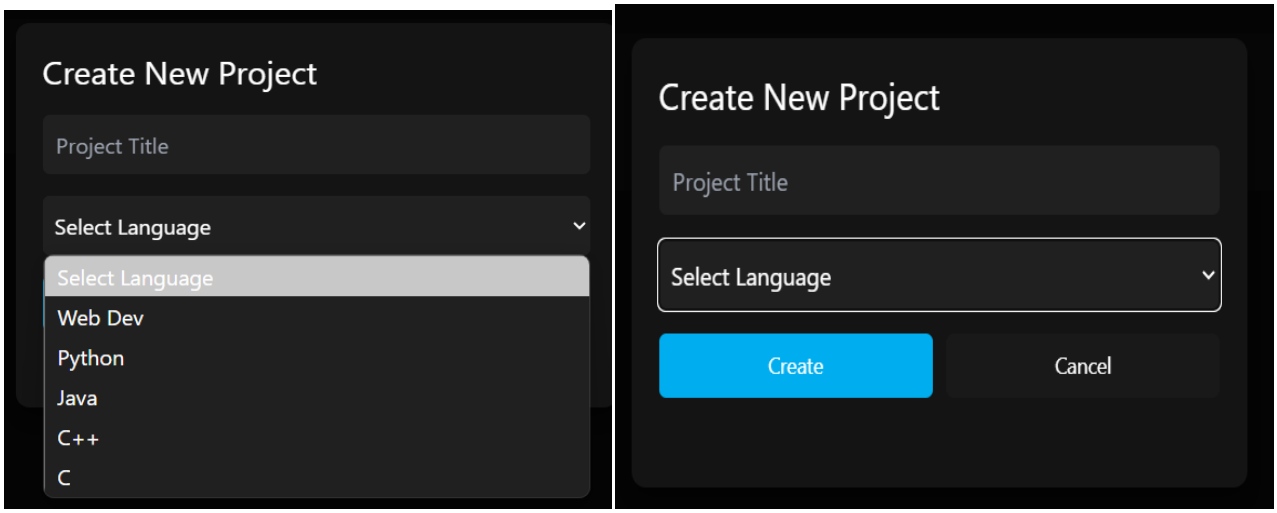


Figure 3: Create a Project

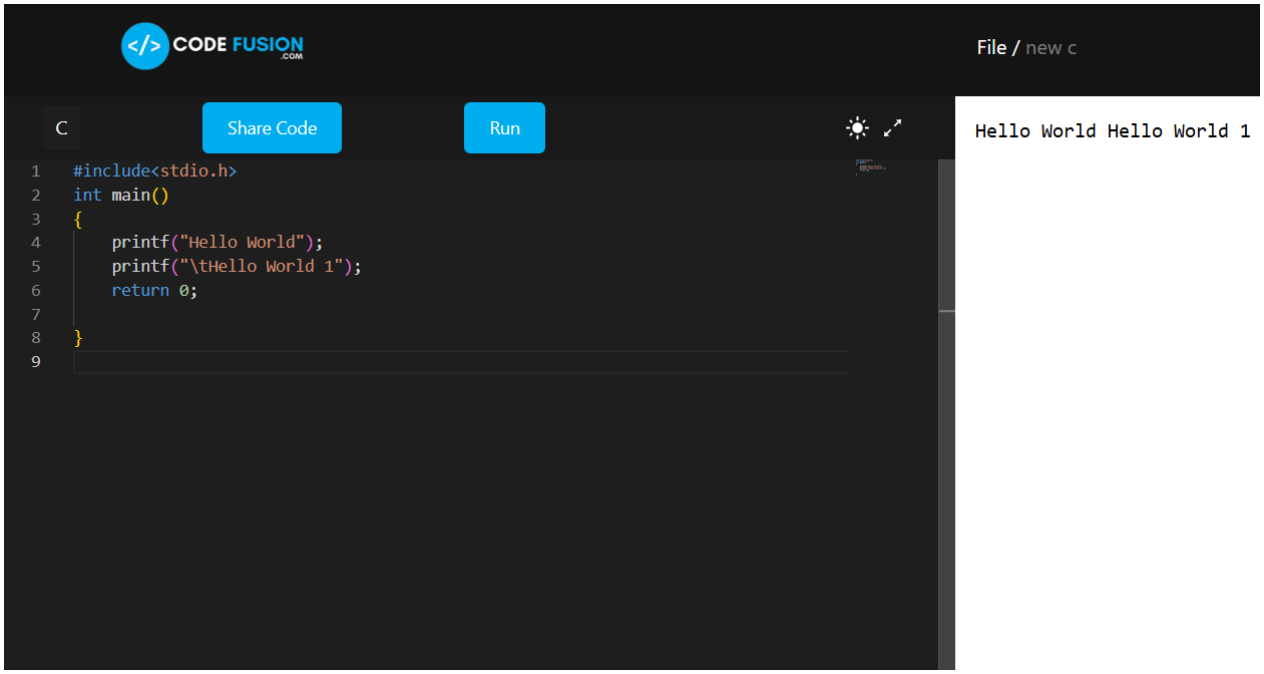


Figure 4: Code Execution System

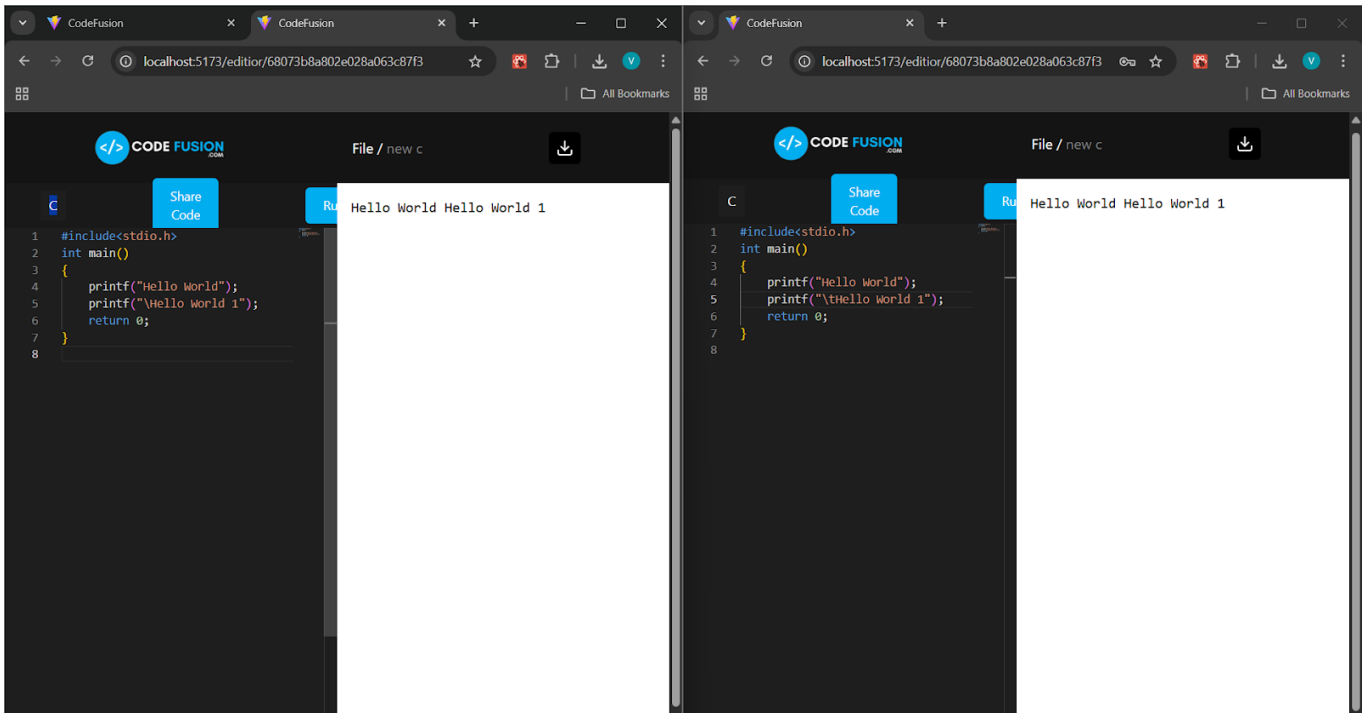
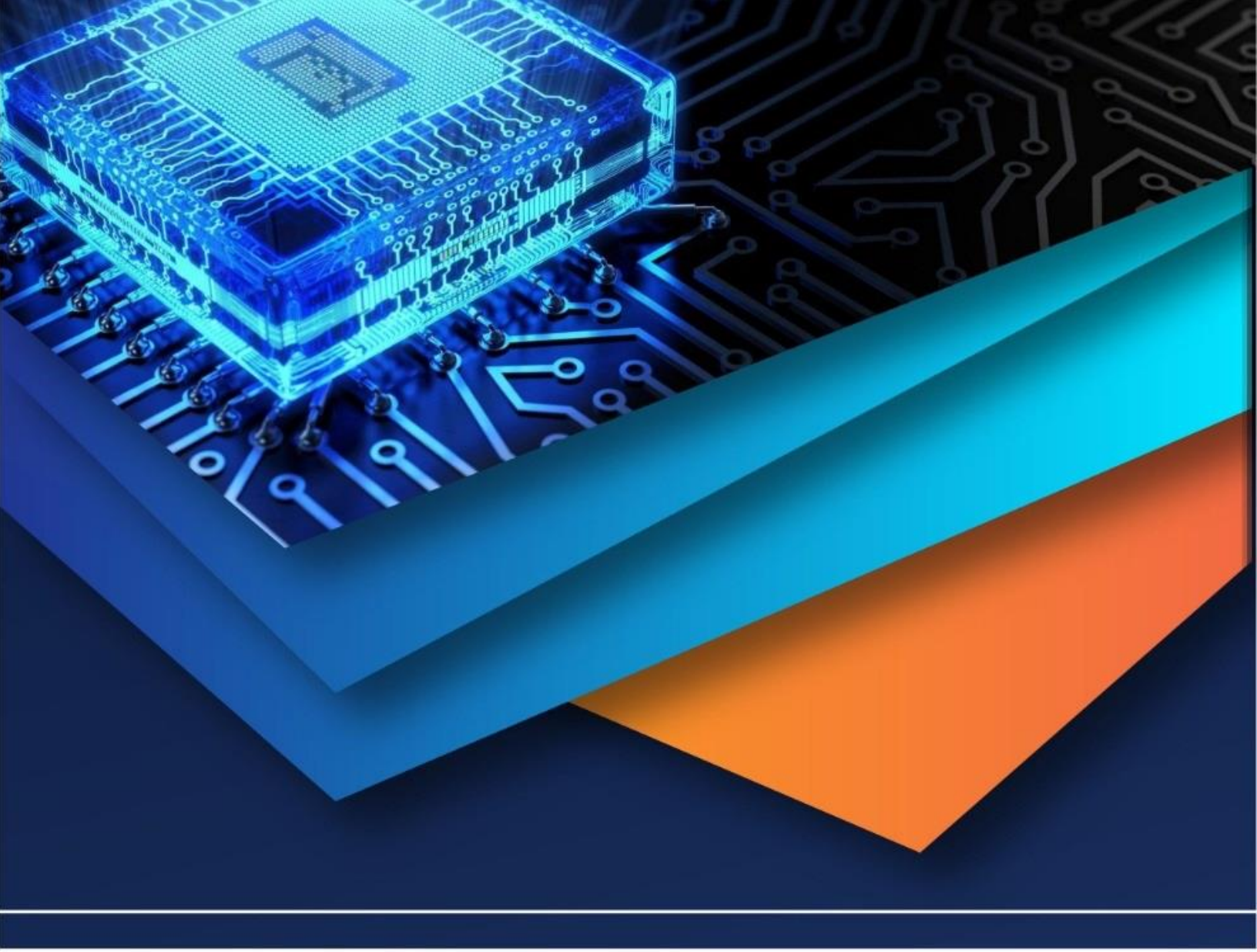


Figure 5: Real-Time Collaboration

REFERENCES

[1] Goldman, M., Little, G., & Miller, R. C. (2011). Real-time collaborative coding in a web IDE. Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11).
[2] Levin, S., &Yehudai, A. (2015). Collaborative Real Time Coding or How to Avoid the Dreaded Merge. arXiv preprint arXiv:1504.06741.
[3] Kurniawan, A., Soesanto, C., &Wijaya, J. E. C. (2015). CodeR: Real-time Code Editor Application for Collaborative Programming. Procedia Computer Science, 59, 510–519.
[4] Li, K., Zhang, W., & Zhang, Y. (2019). CoVSCode: A Novel Real-Time Collaborative Programming Environment for Lightweight IDE. Applied Sciences, 9(21), 4642.

[5] Khan, M. B., Kushwaha, C. S., Rani, R., Verma, A., &Bahad, P. (2023). Design and Development of Real-time Code Editor for Collaborative Programming. International Journal of Scientific Research in Computer Science and Engineering, 11(3).
[6] Dhawan, A., Singh, A., Gupta, C., Goel, P., &Teotia, S. (2024). Collaborative Landscape of Software Development: RTC Code Editor. International Research Journal on Advanced Science Hub, 6(5).
[7] Wang, A. Y., Wu, Z., Brooks, C., &Oney, S. (2024). "Don't Step on My Toes": Resolving Editing Conflicts in Real-Time Collaboration in Computational Notebooks. arXiv preprint arXiv:2404.04695.
[8] Jatana, N., Singh, M., Gupta, C., Dhand, G., Malik, S., Dadheech, P., Aneja, N., &Aneja, S. (2024). Differentially Processed Optimized Collaborative Rich Text Editor. arXiv preprint arXiv:2407.03027.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY