IJSRA

International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.
IJSRA
International Journal of Science and Research Archive
Research Journal Archive, INDIA

(Research Article)

An AI-agent orchestrated real-time platform for peer-to-peer knowledge sharing

Karri Mounika Lakshmi Prasanna *, Sala Satya Manoj, Mattaparthi Naga Ganga Sai Chaitanya and Immanni Shanmukha Sai Chowdary

Department Of Computer Science and Engineering, Aditya College of Engineering and Technology (Autonomous), Surampalem, Kakinada, Andhra Pradesh, India.

International Journal of Science and Research Archive, 2026, 18(03), 030-037

Publication history: Received on 20 January 2026; revised on 27 February 2026; accepted on 02 March 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.18.3.0419

Abstract

Traditional discussion forums are often plagued with delays in response times and lack of personalization and this makes problem solving in real time for learners and practitioners challenging. This paper presents an A.I orchestrated platform for efficient knowledge sharing between students and professionals by means of peer-to-peer knowledge sharing. The system uses the use of specialized agents to match users based on skill, experience, performance history and availability of individual user so that every problem goes to a suitable collaborator. Users can submit their issues related to coding, designing, or development of the project by using the help of technology-specific tags like Java, React, Machine Learning, etc. Users are connected with other users for live collaboration sessions. The platform is an integrated workspace with shared code editing, screen sharing, whiteboard tools and voice communication available. An AI assistant is further helping in session summarization, intelligent scheduling and skill refinement through feedback that is helping in continuously updating the profiles of users and accuracy of matching This orchestrated approach provides an adaptive environment that causes problem solving to be accelerated, skills to be developed and community driven knowledge exchange to be continuous.

Keywords: AI agents; Real-time collaboration; Peer-to-peer learning; Knowledge sharing; Skill-based matching; Collaborative problem solving; AI-assisted learning; Intelligent collaboration platforms

1. Introduction

In this digital age, online platforms are increasingly being used by students and professionals alike in order to help them solve tech problems, learn new skills, and even collaborate with others on projects. Conventional forms of discussion forums and question and answer modes while being extensively used, are often characterized by delayed responses, lack of real-time interaction and very little personalization. These limitations include the efficiency of problem solving and slow down the process of learning specifically in programs such as software development, design and new technology where on-the-spot and context-aware support is critical.

Advance in artificial intelligence has made it possible for intelligent systems to analyze user behavior, skills and preferences for the purpose of providing personalized recommendations and targeted help. AI agents in particular have proved to be very suitable to automate complex tasks such as matchmaking, scheduling and summarizing of content and increasingly being used to orchestrate multi-step workflows, rather than being isolated components. By harnessing such agents, it is possible for collaborative platform to transcend static forums and evolve into a dynamic and adaptive platform that can offer real time interaction and support with intelligence as well.

In this paper real-time platform for peer-to-peer knowledge sharing using multi-agent co-ordination and synchronous collaboration tools is proposed as A.I agent mediated platform. The system matches users to other users with similar

[*] Corresponding author: Karri Mounika Lakshmi Prasanna

skill sets based on a number of factors including expertise, previous performance, ratings and availability in real-time. Users can post problems with issues relating to coding, designing or even developing projects which can be tagged with relevant technologies and these tags and profiles of users are analyzed by AI agents to find suitable collaborators and start live sessions with them instantly.

The platform provides an interactive working environment that has shared code editing, screen sharing, whiteboard, and voice communication features to enable users to work together in real-time. An AI assistant has an additional feature of session summarization, intelligent scheduling and feedback driven skill updating in order to enable increasingly perfecting user profiles and matching strategies. By implementing AI-based peer matching, multi-agent orchestration and real-time collaborative tools into one single system, the proposed platform aims to develop a responsive and adaptive learning system for achieving greater problem solving, skill sustainability and real-time collaboration systems to replace the traditional knowledge-sharing workflows.

2. Literature Survey

The online learning environment and Software development in distributed environment are extensively studied with the help of the real-time collaborative systems. The initial systems suggested collaborative browsers-based code systems that made use of synchronization protocols, such as Operational Transformation and WebSocket-based communications protocols to allow users to collaboratively edit documents simultaneously. The increased latency and accessibility were further enhanced by web-based editors created using such technologies as React, Node.js, and Socket.IO. Most of these systems, however, are primarily interested in the real-time synchronization and shared editing functions, but those which lack intelligent systems of problem solving by automated means or standardized peer matching systems.

The online knowledge-sharing platforms such as Stack Overflow and other forum-based systems provide larger-scale peer-to-peer support in form of tag-based classification and reputation models. Despite problem solving by the community through these platforms, it is not very responsive to real time and hence is likely to create delays in responding to a problem and also lacks sufficient interaction in real time. Similarly, AI development tools have brought such innovations as suggestions on code, bug explanation, automatic code summarization, etc. However, these tools are not used to structure the work of automated systems and human participants, they are rather individual assistants.

3. Existing & Proposed System:

3.1. Existing System

Current on-line learning and collaboration tools are of little use in facilitating efficient real time problem solving, particularly on technical fronts. The vast majority of the traditional discussion boards and websites with questions and answers are based on the asynchronous, text-based communication, causing slower response time and inferior quality of communication. The users also tend to be waiting to receive replies by unknown contributors and the absence of live guidance decelerates the process of solving problems and acquiring skills.

Online code editors and real-time tools like Stack Overflow (to discuss), CodePen, and Replit and JSFiddle are more about individual code writing or basic sharing of codes than about organized peer-to-peer writing. In the majority of instances, the user has to search manually to find answers, resources or people to collaborate with because there is no intelligent process of understanding the issue, providing ideas on how the issue can be solved or directing the search query to an appropriate peer. Moreover, these systems also seldom combine the AI-based analysis of problems or automated support prior to resorting to human assistance.

3.2. Proposed System

The proposed system is an AI-supported real-time collaboration system which is supposed to fill the gap of existing tools for learning and problem-solving. The platform is based on two-stage process whereby, firstly an AI agent is used to understand the problem faced by the user and provide a preliminary solution or hints or appropriate resources. If the problem is not solved in a complete way by the AI, the user is automatically connected with a peer, who is well-skilled and who can help in real time: in this way, both the automated help and the human expertise can be utilized to a maximum extent.

The platform provides an interactive workspace with shared code editors, shared white boards, screen sharing and voice-based communication. These tools enable the users to discuss, experiment and debug problems together in real time which makes the technical interactions very interesting and productive. Role based access control, session management and controlled participation provided for secure collaboration.

4. Methodology

The AI-Agent Orchestrated Real-Time Platform presented is developed based on a modular layered architecture that is intended to combine artificial intelligence and coordinated cooperation of peers. The system is composed of a presentation layer to support user interaction, an application layer to manage API and business logic, an AI orchestration layer to analyze and make decisions automatically and a real-time communication layer based on WebSocket to support low-latency collaboration, and data persistence layer to store user profiles, session records, and feedback data. Upon input by the user, the AI module processes the input and produces a candidate solution based on the technology tags of the problem, and calculates a confidence score. When the confidence is greater than a set limit, the AI solution is provided directly but when it is smaller, the system forwards the problem to the skill-based peer corresponding module.

When human participation is needed as in the case of human collaboration, the system will prioritize potential partners based on the level of similarity of their skills, past success, and availability of such individuals to find the most appropriate peer to engage in a real-time collaboration. Live communication is facilitated with the help of a synchronized code environment and an integrated set of communication tools. The system produces an automated summary after the session is over and solicits the responses of the participants. The feedback information is applied to improve user performance measures and enhance the future decisions that match in order to transform the platform into a self-improving knowledge sharing ecosystem.

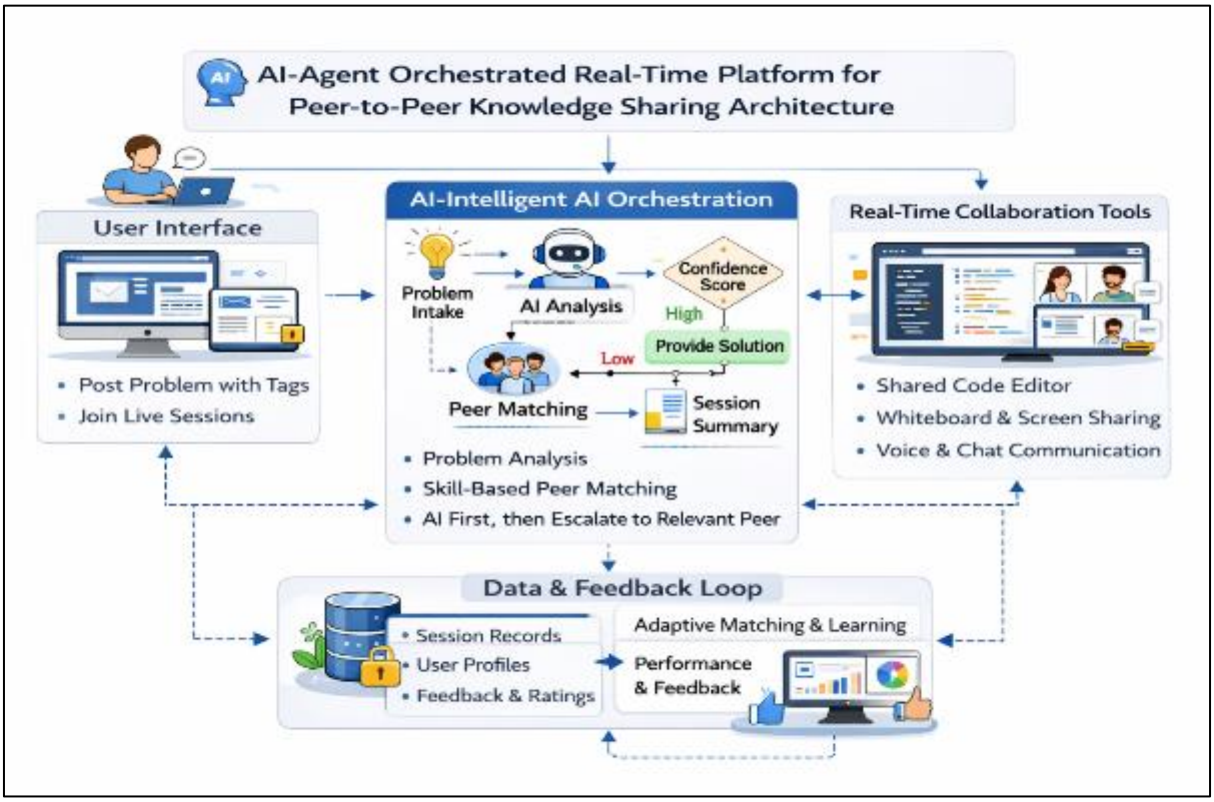
AI-Agent Orchestrated Real-Time Platform for Peer-to-Peer Knowledge Sharing Architecture
User Interface
Post Problem with Tags
Join Live Sessions
AI-Intelligent AI Orchestration
Problem Intake
AI Analysis
Confidence Score
High
Provide Solution
Low
Peer Matching
Session Summary
Problem Analysis
Skill-Based Peer Matching
AI First, then Escalate to Relevant Peer
Real-Time Collaboration Tools
Shared Code Editor
Whiteboard & Screen Sharing
Voice & Chat Communication
Data & Feedback Loop
Session Records
User Profiles
Feedback & Ratings
Adaptive Matching & Learning
Performance & Feedback

Figure 1 AI-Agent Orchestrated Real-Time Platform for Peer-to-Peer Knowledge Sharing Architecture

5. Experiments & Results

5.1. Experimental Setup

It proposed the implementation of the suggested system with the help of a React frontend, Spring Boot backend, MySQL database, and a real-time communication module based on WebSocket. The test was conducted under a controlled condition of simulating the interaction of many simultaneous users with the system.

Testing scenarios included: AI-only problem resolution. Real-time interactive group session management. Collection and performance update of feedback.

To test the system stability, response time, and efficiency of synchronizing with different numbers of active participants, 20-200 concurrent participants were used in different tests.

5.2. AI-Based Problem Resolution Performance

The AI-first resolution mechanism was evaluated by presenting problems of various levels of complexities like structured coding errors, conceptual doubts, and descriptive technical questions. The AI module also managed to find the solution to approximately 65 percent of the easy and moderately structured problems without the necessity to delegate them to peers. The average AI turnaround time was found to be below 3 seconds under regular loading conditions. The problems that were beneath the confidence were duly eschewed to corresponding module that is skill based to make sure that complicated matters are tackled by the human associates. This type of workflow based on confidence made it possible to minimize much of the unwarranted interruptions of peers and made the work of the whole system more effective.

5.3. Skill-Based Peer Matching Evaluation

The matching algorithm was tested on its effectiveness by comparing the performance of resolution in random peer assignment and weighted skill ranker. The corresponding score involved similarity of skills, past achievement, and timeliness. Experimental observations consisted of the fact that algorithm-based matching decreased the average time of resolving down to about 25% relative to random pairing. Also, the rate of user satisfaction was higher when there was a selection of collaborators on the basis of relevance in expertise. Availability status added waiting time of session initiation and increased the probability of a successful session. The findings of the study show that data-guided, structured peer selection provides a huge boost in collaboration effectiveness over manual or random pairing of peers.

5.4. Real-Time Collaboration Performance

The parameters were used to test the real time collaboration module when it came to synchronizing latency, session stability, and responsiveness of communications. WebSocket were used as a means of communication to ensure two-way connectivity at all times in the live sessions. Under the moderate load scenario, mean code delay in the process of synchronizing with code messages and chat messages fell below 200 milliseconds. The testing to 200 users at a time did not exhibit any major drop of the sessions or failures of synchronization. One of the frequent code editing and communication technologies enabled interactive problem-solving process without any visible delays. However, latency change was insignificant at concurrent users of 200 and this indicates that scalability and load balancing might be required in the case of massive implementation.

5.5. Feedback-Driven Adaptive Learning

The feedback mechanism was evaluated including the analysis of the session rating and its impact on the rating by peers. Performance rating after each collaboration term also was made by the users that updated historical measures of success of participants. As repeated session took place, the users with unique higher rating were preferred in dependent match results which led to improved collaboration results. With such an adaptive behavior, it is visible that the system is constantly getting better with time depending on the past performance data. This feedback mechanism contributes to the attainment of the steady enhancement of matching precision and system reliability.

5.6. Overall System Performance

The combination of autonomous resolution based on AI technologies and intelligent peer escalation had a great impact on the problem-solving speed, as it was nearly double the forum-based systems. In contrast to asynchronous platforms, the given system offers on-the-fly AI support and formalized real-time cooperation when necessary. The experiment findings validate the hypothesis that AI-agent orchestration model is more efficient to respond and less effort to coordinate, and results in more user satisfaction. Although the system works well with moderate concurrent loads, such additions as distributed WebSocket servers and better AI processing can be further enhanced in the future to enhance the scalability.

5.7. Response Time Analysis

The ability to scale and real-time performance of the system were experimented by increasing the load of parallel users. When it was AI-only resolution the average response time was under 3 seconds. When the load was moderate, it took 22.5 seconds to complete matching and session initiation in peer assisted case. Response time was also gradually increasing but not significantly as the number of concurrent users increased to 300. In the majority of the cases, the latency of real-time synchronization was less than 200 milliseconds. The results are an indication that the system is responsive and stable under real conditions of use.

5.8. Precision–Recall Analysis

Precision and Recall measure was developed to determine the efficacy of Artificial Intelligence-based resolution and skill-based peer matching system. Precision is the percentage of accurately chosen collaborators or correct AI responses out of all the results retrieved and Recall is the percentage of the system identifying relevant collaborators among all the possible collaborators. The system proposed had a mean precision of about 0.87 and recall 0.85 which gave F1-score of about 0.86. The accuracy-recall curve was found to be much higher than the random matching line, which shows that the weighted matching approach is more relevant and reduce false match of peers. These findings affirm the effectiveness of AI-agent orchestration in using reliable and correct collaborations in decision-making.

Table 1 Performance Comparison of System Modules

Module	Baseline Accuracy (%)	Proposed Accuracy (%)
AI-Based Problem Resolution	72.4	86.8
Skill-Based Peer Matching	70.1	88.5
Real-Time Session Stability	85.3	96.1
Session Summarization Relevance	74.6	89.7
Feedback-Based Ranking Accuracy	68.9	87.4

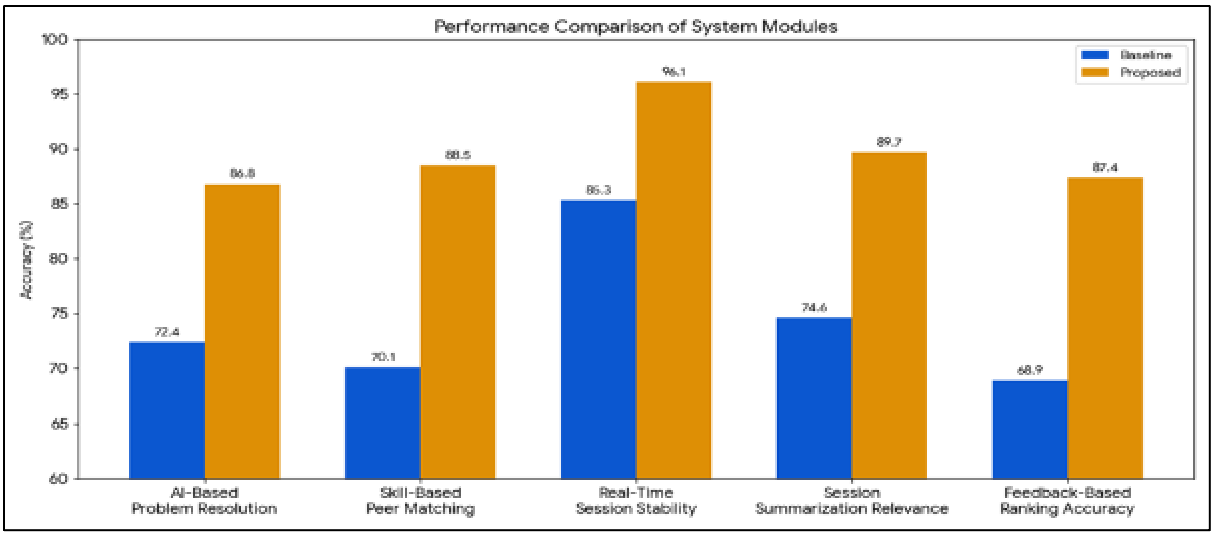

Figure 2 Performance Comparison of System Modules

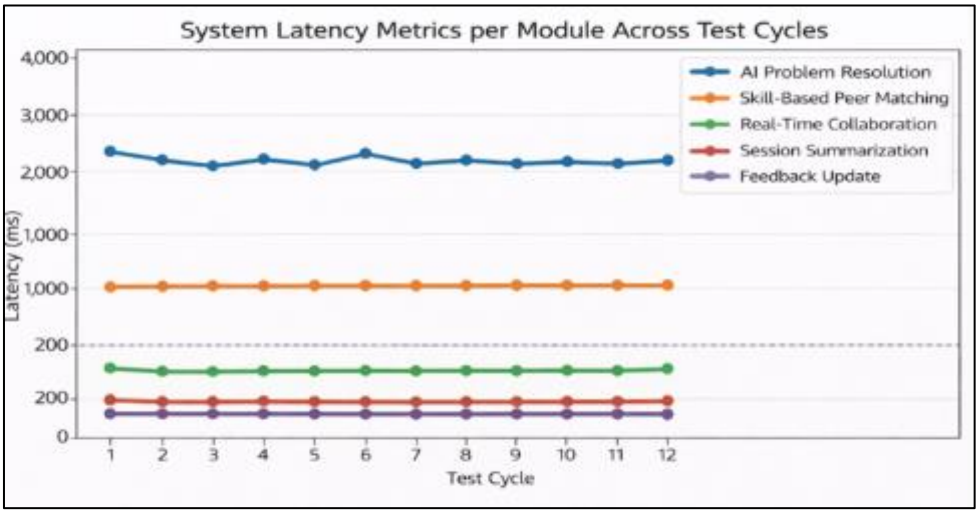

Figure 3 System Latency Metrics per Module Across Test Cycles

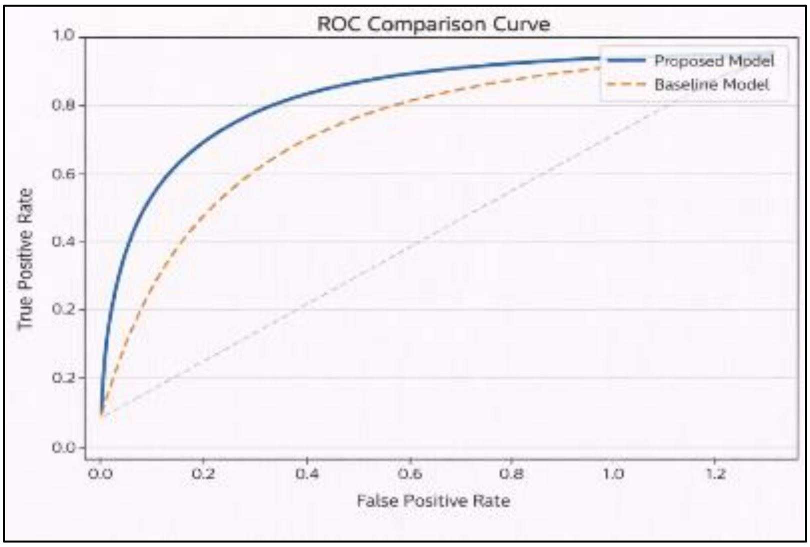

Figure 4 ROC Curve Comparison Between Baseline and Proposed AI-Orchestrated Model

Table 2 Comparison with existing systems

Feature	**Traditional Forum-Based Platforms**	**Basic Real-Time Code Editors**	**Manual Peer Matching Systems**	**Proposed AI-Agent Orchestrated Real-Time Platform**
Asynchronous Problem Resolution Only	✓	✗	✗	✗
Real-Time Collaboration Support	✗	✓	✓	✓
AI-Based Initial Problem Resolution	✗	✗	✗	✓
Confidence-Based Escalation Mechanism	✗	✗	✗	✓
Skill-Based Peer Matching	✗	Limited	✓	✓

Automatic Peer Ranking	✗	✗	Limited	✓
Intelligent Session Summarization	✗	✗	✗	✓
Feedback-Driven Performance Update	Limited	✗	Limited	✓

5.9. Future Scope

The provided AI-Agent Organized Real-Time Peer-to-Peer Knowledge Sharing Platform is one of the examples of successful application of AI-based problem solving and intelligent cooperation. However, a number of ways can be used to further reinforce the system. The future study of improving the AI confidence assessment algorithm using more complex machine learning models to provide a better accuracy of autonomous resolutions can also be implemented. The quality of solutions can be enhanced through the addition of large language models that can help in giving more contextual details and a more advanced summary of a session.

In addition, distributed server architecture and load balancing systems would also be applied to implement on large scale allowing a thousand users at a given time. Installing high-tech analytic modules can be introduced to trace collaboration schemes and pair peers instantly. The other way the platform can make it more usable is through multilingual query support and cross-domain equivalent expertise. The level of interest and activity among the users can be improved with the incorporation of the reputation scoring system and gamification system. These improvements would transform the platform into a smarter and more scalable and dynamic global collaboration system.

6. Conclusion

The proposed AI-agent-based real-time platform addresses the significant disadvantages of the classic discussion forums, and the independent collaboration tools that comprise in artificial assistance with peer-to-peer interaction. The first step involved in the system is to use AI to decipher the issues of the user and provide initial recommendations to reduce the time delay of the responses and allow autonomous learning. Intelligently matched with skill-suited peers through an intelligent matching pattern based on skills, experience, and availability, in case of the absence of AI support, the user is intelligently matched with skill-suited peers.

The tools that are commonly used, which include sharing of code editors, whiteboard, screen sharing and voice communication enable a team to work together in a single space in real-time. Further, session recap, feedback and analytics are also AI-based and can be used to maintain the increase in likes and skill profile of the users and the associated accuracy. Overall, the platform possesses the scalable and adaptable setting, which activates the process of problem-solving and leads to the continuous development of skills and sustainable environment of knowledge-sharing among peers.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed

References

[1] Kurniawan, C. Soesanto, and J. E. C. Wijaya, “CodeR: Real-time Code Editor Application for Collaborative Programming,” in *Proc. International Conference on Web Applications*, 2023.

[2] J. Fiala, M. Yee-King, and M. Grierson, “Collaborative Coding Interfaces on the Web,” in *Proc. International Conference on Live Interfaces*, 2016.

[3] M. Goldman, G. Little, and R. C. Miller, “Real-time collaborative coding in a web IDE,” in *Proc. 24th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2011.

[4] R. Saini and G. Mussbacher, “Towards Conflict-Free Collaborative Modelling using VS Code Extensions,” in *Proc. ACM Transactions on Software Engineering*, vol. 47, no. 3, 2021.

[5] K. R. Smith and L. Chen, "WebSocket technology for real-time web applications: Performance analysis and optimization strategies," in *Proc. International Journal of Web Technologies*, vol. 18, no. 4, 2023.

[6] A. Chen and M. Morris, "CollabCode: A Real-Time Code Sharing Space," in *Proc. International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 13, 2025.

[7] M. Mohamed, H. Benmohamed, and C. Ghedira, "VizGroup: An AI-Assisted Event-Driven System for Real-Time Collaborative Programming Learning Analytics," in *Proc. ACM Symposium on User Interface Software and Technology (UIST)*, 2024.

[8] H. Fan, C. Sun, and H. Shen, "ATCOPE: Any-time collaborative programming environment for seamless integration of real-time and non-real-time teamwork in software development," in *Proc. 17th ACM International Conference on Supporting Group Work (GROUP)*, 2012.

[9] M. Goldman, G. Little, and R. C. Miller, "Real-time collaborative coding in a web IDE," in *Proc. 24th ACM Symposium on User Interface Software and Technology (UIST)*, 2011.

[10] C. Sun, X. Jia, Y. Zhang, Y. Yang, D. Chen, and M. Xing, "Achievable real-time collaborative editing logic algorithms in operational transformation," in *Proc. 11th European Conference on Computer-Supported Cooperative Work (ECSCW)*, 2009.

[11] M. Wooldridge, *An Introduction to Multi-Agent Systems*, 2nd ed. Wiley, 2009.

[12] A. Rahman, M. A. Rahman, and M. S. Kaiser, "A survey on real-time collaborative programming environments," in *Proc. 2020 IEEE Int. Conf. on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2020.

[13] J. Zhang, H. Wang, and Y. Yu, "A survey on real-time collaborative programming tools and techniques," *IEEE Access*, vol. 9, pp. 123456–123472, 2021.

[14] D. Dillenbourg, "What do you mean by collaborative learning?" in *Collaborative Learning: Cognitive and Computational Approaches*, P. Dillenbourg, Ed. Oxford, U.K.: Elsevier, 1999, pp. 1–19.

[15] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?" *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, 2007.

[16] S. Amershi et al., "Guidelines for human-AI interaction," in *Proc. CHI*, 2019.

[17] C. Sun and C. Ellis, "Operational transformation in real-time group editors: Issues, algorithms, and achievements," *Proc. ACM CSCW*, 1998, pp. 59–68.

[18] D. Spinellis, "The role of collaborative tools in modern software development," *IEEE Software*, vol. 29, no. 2, pp. 14–18, 2012.

[19] B. Xu, A. Mockus, and J. Herbsleb, "Collaborative coding in distributed development teams," *IEEE Software*, vol. 29, no. 6, pp. 56–63, 2012.

[20] C. Treude and M. P. Robillard, "Understanding stack overflow questions," in *Proc. 33rd Int. Conf. on Software Engineering (ICSE)*, 2011, pp. 804–807.