

# A Human–AI Collaborative Project Management System with Agent-Based Intelligence

P. Naveen Raj<sup>1</sup>; M. Madesh<sup>2</sup>; M. S. Ramesh<sup>3</sup>; S. Venkata Lakshmi<sup>4</sup>

<sup>1,2,3</sup>UG Scholar, CSE Department, K.L.N. College of Engineering, Sivagangai, Tamil Nadu, India

<sup>4</sup>Associate Professor, CSE Department, K.L.N. College of Engineering, Sivagangai, Tamil Nadu, India

Publication Date: 2026/03/21

**Abstract:** Modern software development environments require continuous coordination among team members, timely issue resolution, and accurate project planning to achieve successful project outcomes. However, many existing project management tools still rely heavily on manual monitoring and human-driven decision making. Such reliance often leads to delayed responses to issues, inaccurate scheduling of tasks, inefficient distribution of workload, and an overall reduction in productivity. To address these challenges, this work proposes an AI-assisted collaborative framework in which an intelligent agent functions as a virtual team member that supports development activities. The system continuously observes team communication, task progress, issue updates, and code changes in order to develop contextual awareness of ongoing project activities. Based on this contextual understanding, the system generates suggestions only when relevant conditions are detected, such as errors, delays, or inconsistencies. By analysing historical project data together with developer workload patterns, the framework predicts potential delays and recommends adaptive scheduling and planning strategies in advance. A human-in-the-loop mechanism is incorporated to ensure that developers retain complete control over final decisions. This collaborative approach improves productivity, enhances planning accuracy, and supports effective cooperation between human intelligence and artificial intelligence without replacing human expertise.

**Keywords:** AI-Assisted Project Management, Agent-Based Intelligence, Human–AI Collaboration, Intelligent Software Engineering, Context-Aware Recommendation, Predictive Task Scheduling, Code Analysis Automation, Human-in-the-Loop Systems, Developer Productivity, Knowledge-Driven Collaboration.

**How to Cite:** P. Naveen Raj; M. Madesh; M. S. Ramesh; S. Venkata Lakshmi (2026) A Human–AI Collaborative Project Management System with Agent-Based Intelligence. *International Journal of Innovative Science and Research Technology*, 11(3), 1733-1739. <https://doi.org/10.38124/ijisrt/26mar1009>

## I. INTRODUCTION

Continuous collaboration, prompt decision-making, and efficient coordination are essential for modern software development, particularly as remote and dispersed teams become more typical. As projects become increasingly complex, relying solely on human monitoring for scheduling, problem solving, and task management becomes inefficient and prone to delays. Conventional project management systems mainly help with tracking and communication, but they don't have the intelligence to help developers in an emergency. The creation of intelligent frameworks that examine project data, team interactions, and code artifacts to offer decision assistance and prediction insights has been made possible by recent developments in artificial intelligence, agent-based systems, and big language models [1], [3], and [4]. Human-AI collaboration research focuses on contextual awareness and adaptive support, with the goal of augmenting rather than replacing human capabilities [2], [7], [10]. Agile management, defect prediction, and decision support have all employed intelligent agents to boost productivity and planning accuracy [6, 9, 11]. However, many existing systems operate independently, lack real-time

connection with communication platforms, and lack reliability and human-in-the-loop control techniques [8], [12]. Recent studies on cognitive agents and multi-agent frameworks highlight the possibility for proactive risk identification and adaptive scheduling [4], [5], [15], whereas context-aware systems merely provide assistance when needed [13], [14].

## II. METHODOLOGY

The proposed system follows a structured and sequential workflow to provide intelligent assistance in collaborative software development environments, as illustrated in Fig. 1. The initial stage in the procedure is to continuously monitor developer contact through chat interfaces, problem discussions, or collaborative platforms. A virtual AI assistant works silently in the background, monitoring messages and interacting without disrupting normal communication. An initial trigger detection algorithm examines incoming messages to identify predefined terms and semantic patterns linked to errors, flaws, or development issues. When no triggering condition is detected, the AI assistant remains dormant, ensuring minimal interference.

The system collects contextual information from the message, such as pertinent repository references and file-level details, when it detects a trigger. The captured repository data is then used to retrieve the source code from the version control system. An AI-based analysis engine examines the recovered code and previous project data to identify any errors, logical inconsistencies, or implementation risks.

recommendations to improve work scheduling and resolve issues discovered. These recommendations are provided via a human-in-the-loop review stage, which allows developers to accept, modify, or reject the AI-generated insights. Finally, validated decisions and interaction outcomes are stored in a knowledge repository to support continuous learning and improve the accuracy of subsequent recommendations.

### III. PROCESS FLOW

The proposed approach uses a structured sequential process to provide intelligent support for collaborative software development. The procedure starts when the user launches the web application, which acts as the main interface for project operations, as shown in Fig. 2. The system gathers team conversations, task updates, and cooperative exchanges created during development once it is accessed. In order to preserve a consistent history of team activities and advancement, these communications and task records serve as the original project data.

Following data collection, the system conducts two steps of parallel analysis. To find development intentions like implementation problems, help requests, code mistakes, or workflow conflicts, an AI intent detection module first analyse messages and job descriptions. In order to identify potential delays or scheduling hazards, a deadline analysis component simultaneously examines task timeframes, workload distribution, and past completion patterns. The system makes AI-driven recommendations based on these evaluations that are meant to enhance task performance, address issues found, and facilitate improved project planning. By presenting these suggestions to the user for assessment, a human-in-the-loop approach guarantees that developers continue to participate in decision-making. Lastly, project data is safely kept along with authorized proposals and produced results. In collaborative team environments and sophisticated contemporary project workflows, this structured method fosters continual learning, increases efficiency, and fortifies cooperation between developers and intelligent systems throughout ongoing software development projects.

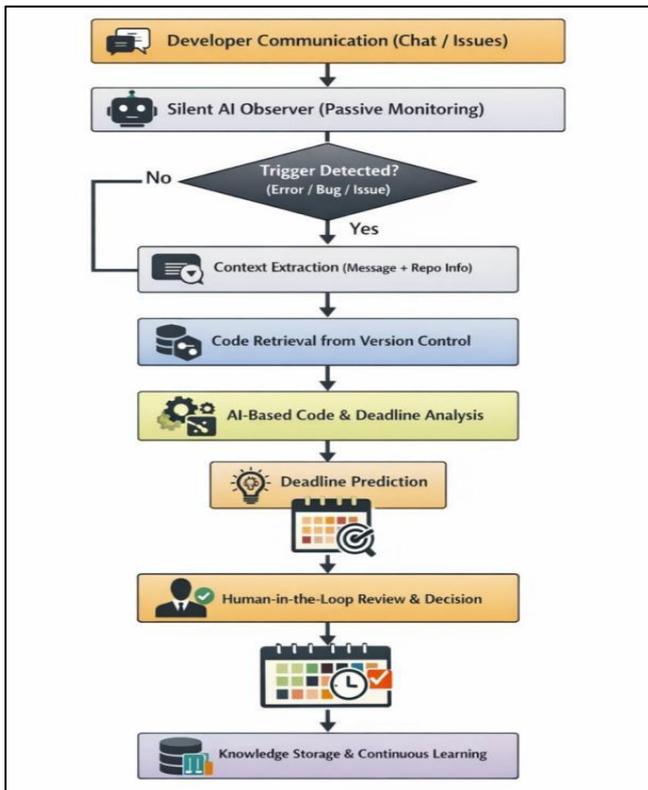


Fig 1 Methodological Workflow

The system simultaneously performs deadline analysis by evaluating task complexity, past completion trends, and developer effort in order to predict potential delays and provide revised timelines. Based on the combined code and deadline analysis, the system generates concise and useful

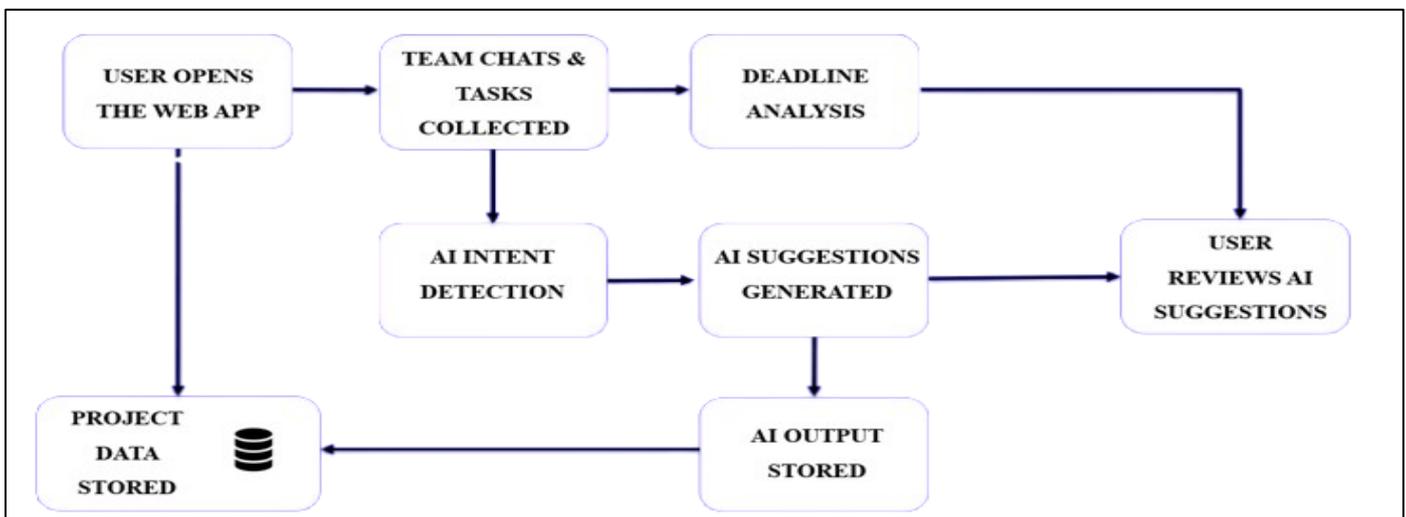


Fig 2 Process Flow

**IV. IDENTITY AND ACCESS CONTROL**

The collaborative development platform incorporates identification and access control techniques to provide safe and role-based system interaction. Authorization policies are implemented according to predetermined roles, such as developers, project managers, and administrators, and users gain access to the system using secure login credentials. The degree of access to project data, task details, AI-generated recommendations, and stored history records within the system is determined by these roles. At the same time, limiting unauthorized data exposure during cooperative development activities, the access control mechanism guarantees the protection of sensitive project information. Furthermore, secure session management and token-based authentication enhance system dependability, especially in distributed development settings where numerous users engage with the platform at once. Access logs and permission validation mechanisms help to maintain accountability by tracking user actions and tracing system interactions. While retaining flexibility for team-based workflows, this structured identity management system guarantees that collaboration, AI-assisted recommendations, and project monitoring activities take place inside clearly defined security limits.

**V. AGENT-BASED INTELLIGENCE AND REASONING**

Agent-based intelligence is the system's main analytical skill, allowing it to operate as an autonomous yet supportive virtual team member. Intelligent agents continuously examine communication streams, task updates, and code artifacts to find important patterns and insights. They use contextual analysis, machine learning models, and rule-based reasoning to identify any errors, inconsistencies, or process bottlenecks. Instead of responding to every interaction, the agents operate selectively, turning on only when relevant trigger conditions, such as error reports, issues, or work delays, are detected. This selective activation minimizes disruption to human workflows while maintaining situational awareness. The reasoning process also takes into account historical project data, including past task outcomes, developer performance trends, and previous AI recommendations, to produce context-aware and practical recommendations. By combining proactive detection,

predictive reasoning, and adaptive learning, agent-based intelligence enhances decision-making, aids in predictive planning, and provides focused support. This improves teamwork and overall project efficiency without replacing human authority.

**VI. PREDICTIVE PLANNING AND HUMAN-IN-THE-LOOP COLLABORATION**

Predictive planning in the proposed system is achieved through the analysis of workload trends, development velocity, and historical task completion data. By examining task complexity and past performance patterns, the algorithm estimates realistic completion timelines for ongoing development activities. This analysis enables the system to anticipate possible delays and identify scheduling risks before they significantly affect project progress. If any delay is identified, the system adjusts the project timeline or task sequence accordingly to maintain workflow continuity. Early detection of such risks supports better planning and allows development teams to manage tasks more effectively. The predictive insights assist teams in allocating resources efficiently, balancing workloads, and maintaining realistic expectations for project milestones while reducing the likelihood of deadline overruns. The system includes a human-in-the-loop cooperation mechanism to guarantee safe use of AI. Users are given the opportunity to examine, approve, alter, or reject AI-generated recommendations before they are put into practice. While ongoing user feedback enhances system performance and future recommendations, this method guarantees that human authority and accountability remain vital to the decision-making process.

**VII. RESULT AND DISCUSSION**

An AI-assisted collaborative environment that enhances project coordination, communication, and decision-making is demonstrated by the built system. The platform offers an intuitive user interface that incorporates intelligent support into standard procedures. Only authorized developers and administrators are able to access project information thanks to the secure login interface depicted in Fig. 3, which uses role-based control and authenticated credentials.

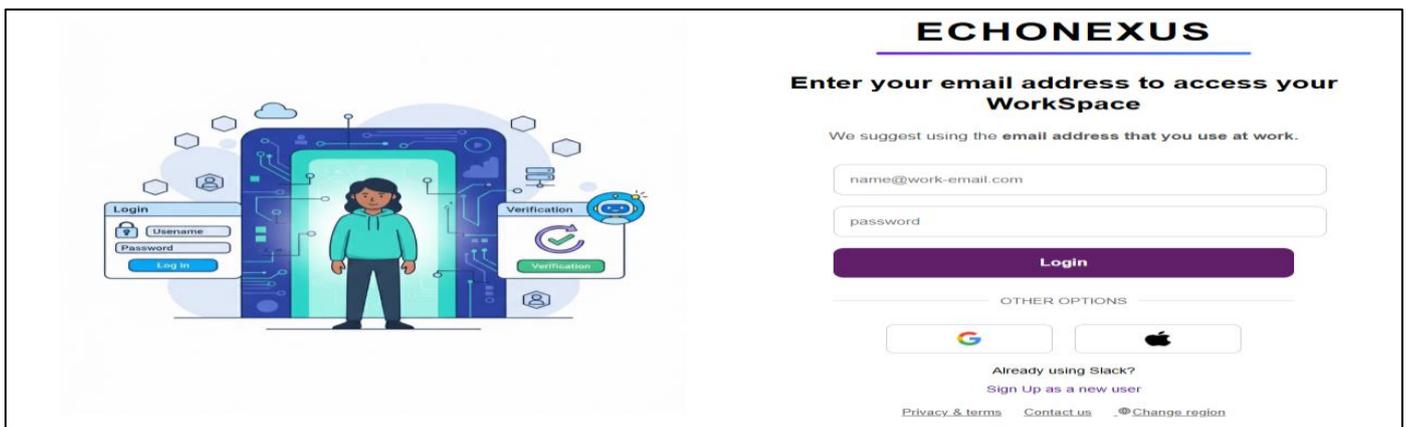


Fig 3 Login Interface

Managers and team members can efficiently keep an eye on development activities thanks to the system's centralized dashboard interface.

The dashboard provides important data including commit activity, project growth, team performance, and task

status in an organized visual layout, as seen in Fig. 4. This interface makes it simple for users to monitor development updates, assess progress, and maintain improved workflow collaboration.

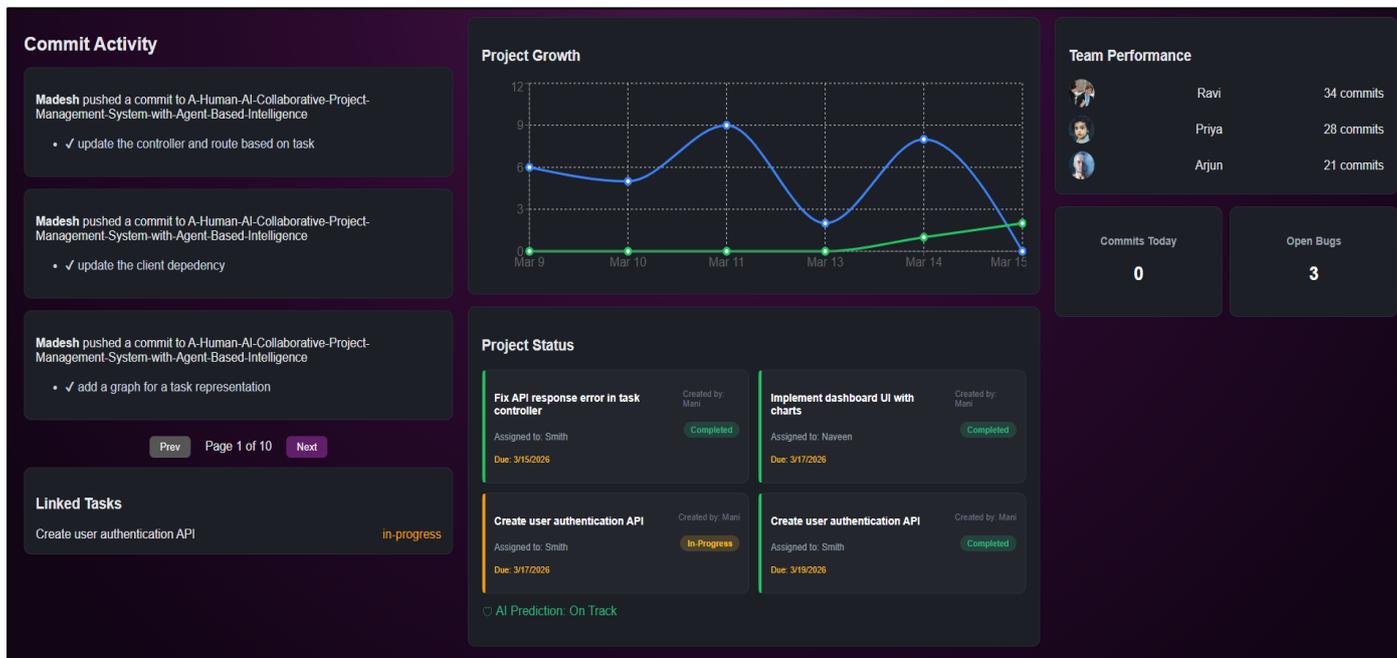


Fig 4 Dashboard

The chat interface, shown in Fig. 5, allows team members to communicate in real time. It serves as the primary point of contact for developers, allowing them to plan development activities, resolve issues, and provide progress reports. A silent AI assistant keeps an eye on the discussion

and only activates when specific trigger conditions, such errors, delays, or problem-related terms, are detected. Once activated, the assistant analyses the conversation's context and makes relevant suggestions to aid developers while maintaining constant contact.

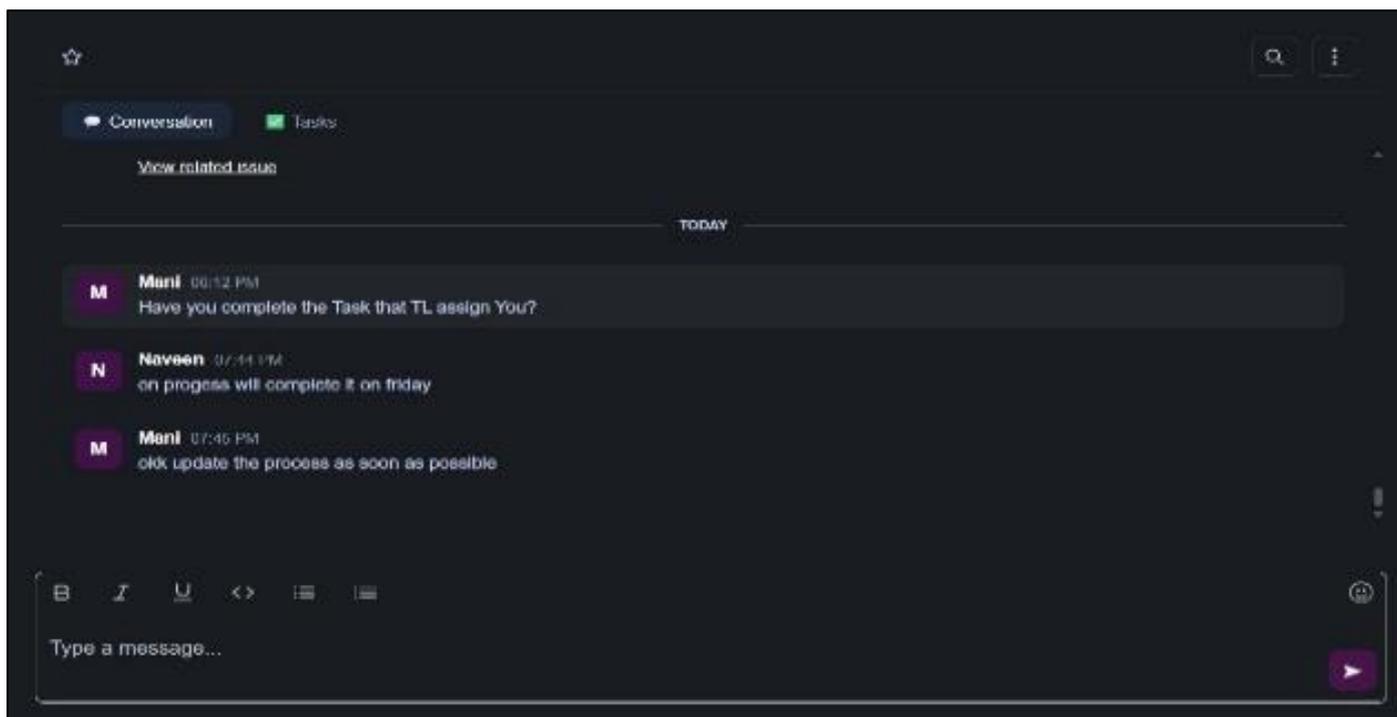


Fig 5 Chat Screen

The project listing module, depicted in Fig. 6, allows users to monitor and manage several projects using a single dashboard. By providing an organized summary of both ongoing and completed projects, it enables users to switch

between multiple workspaces efficiently. In addition to improving accessibility, this well-organized representation makes project administration and monitoring easier.

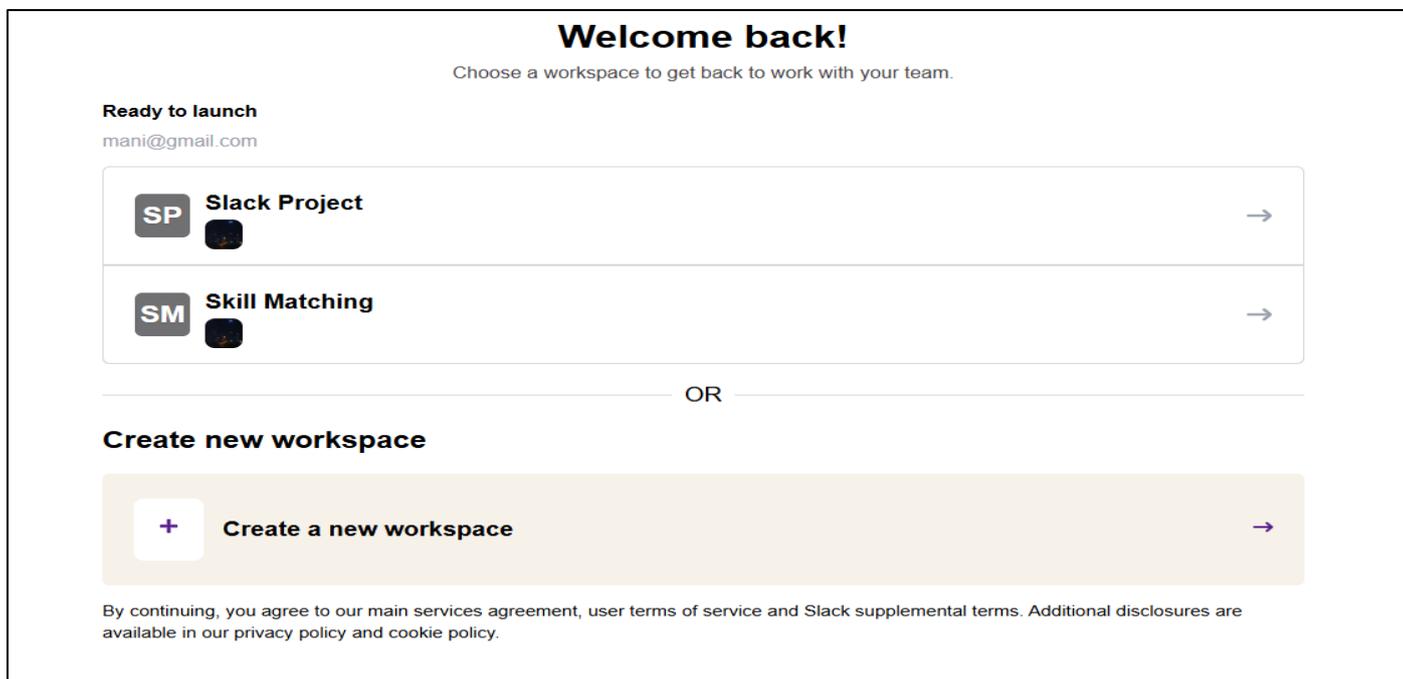


Fig 6 Project Listing

The platform also facilitates AI-based task allocation, in which appropriate developers are automatically assigned tasks based on their current workload and skill set. The system assesses available team members and chooses the best

developer with suitable experience and less ongoing duties, as illustrated in Fig. 7. This guarantees a fair division of the task and boosts team productivity.

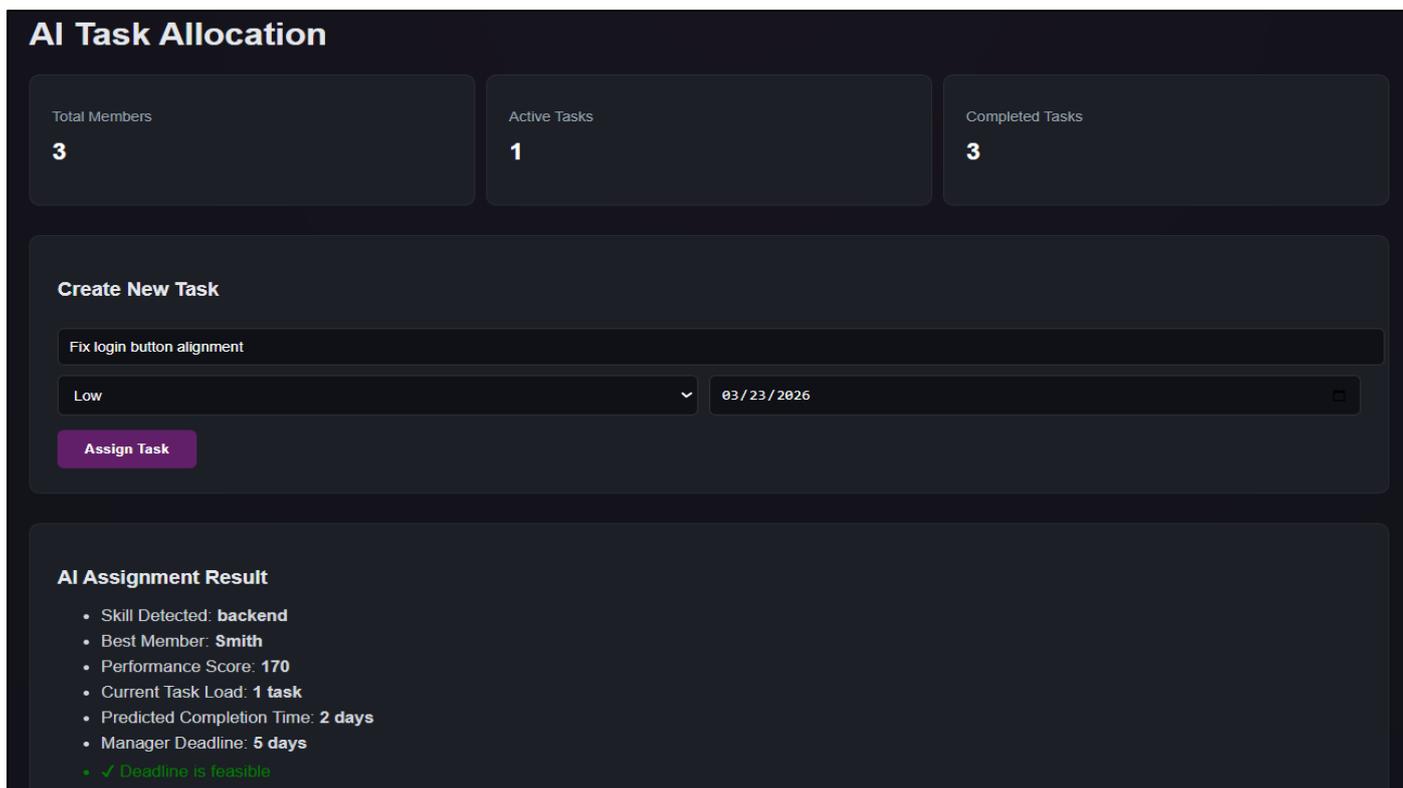


Fig 7 Task Allocation

The AI-generated suggestions exhibited in the group chat, as illustrated in Fig. 8, provide suggestions when scheduling or development challenges are found. These

suggestions appear directly in the chat, allowing developers to review and respond while maintaining regular communication.

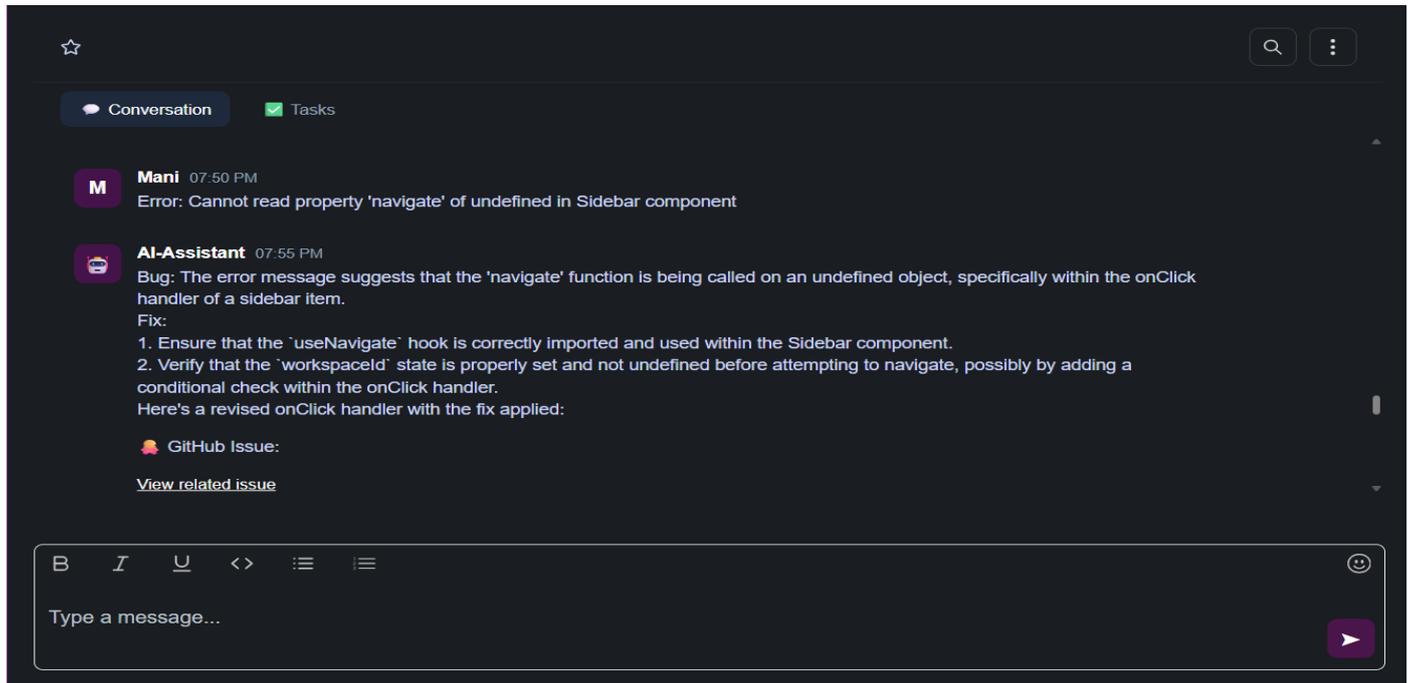


Fig 8 Suggestion of AI

The solution that has been put into place successfully integrates collaborative development elements with intelligent support, increasing productivity and decreasing the need for human monitoring. In contemporary software development environments, integrating AI-driven insights with human oversight results in a balanced workflow that promotes effective communication, well-informed decision-making, and improved project coordination.

## VIII. CONCLUSION

The proposed system presents a framework for AI-assisted collaboration designed to support developers during software development activities. The virtual AI assistant runs in the background and only activates in response to pertinent events, such as problems, errors, or work delays. The system creates practical suggestions that help developers find and fix issues more successfully by examining communication messages, task updates, and code modifications. In order to enhance planning precision, job coordination, and effective project execution, predictive modules simultaneously assess possible delays and provide adaptive scheduling techniques. The integration of a human-in-the-loop approach ensures that developers retain full control over final decisions, thereby maintaining accountability, transparency, and trust in the system. A knowledge base is used to retain approved interactions and ideas, allowing for ongoing learning and the incremental improvement of subsequent recommendations. The framework illustrates how, in contemporary software development settings, the combination of intelligent agents, context-aware analysis, and predictive planning may boost

teamwork, increase output, and facilitate effective task management.

## REFERENCES

- [1]. Parrish, Michael D., and Steven Corns. "An ai-based conceptual framework to improve program management of complex systems." *IEEE Engineering Management Review* (2024).
- [2]. Kamboj, Payal, Ayan Banerjee, and Sandeep KS Gupta. "Expert knowledge driven human-ai collaboration for medical imaging: a study on epileptic seizure onset zone identification." *IEEE Transactions on Artificial Intelligence* 5, no. 10 (2024): 5352-5368.
- [3]. Hoda, Rashina, Hoa Dam, Chakkrit Tantithamthavorn, Patanamon Thongtanunam, and Margaret-Anne Storey. "Augmented agile: Human-centered ai-assisted software management." *IEEE Software* 40, no. 4 (2023): 106-109.
- [4]. Cinkusz, Konrad, Jarosław A. Chudziak, and Ewa Niewiadomska-Szynkiewicz. "Cognitive agents powered by large language models for agile software project management." *Electronics* 14, no. 1 (2024): 87.
- [5]. Gvilava, N. T. "AUTONOMOUS INTELLIGENT AGENTS IN DECISION SUPPORT SYSTEMS FOR CRITICAL INFRASTRUCTURE." *Professional Bulletin: Information Technology and Security* 1 (2025): 55-61.
- [6]. Rahman, Md Mostafizur, Md Mostafijur Rahman, Maria Khatun Shuvra, Md Mashfiqur Rahman, and Najmul Gony Md. "AI-Powered Defect Prediction: From Code Smells to Failure Forecasting." (2025).

- [7]. Takerngsaksiri, Wannita, Jirat Pasuksmit, Patanamon Thongtanunam, Chakkrit Tantithamthavorn, Ruixiong Zhang, Fan Jiang, Jing Li, Evan Cook, Kun Chen, and Ming Wu. "Human-in-the-loop software development agents." In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 342-352. IEEE, 2025.
- [8]. Hoda, Rashina. "Keynote on Augmented Agile: Human-centred AI-assisted Software Project Management at FinanSE Workshop (ICSE 2024)." In *Proceedings of the 1st IEEE/ACM Workshop on Software Engineering Challenges in Financial Firms*, pp. 17-18. 2024.
- [9]. Cinkusz, Konrad, and Jarosław Chudziak. "Agile Software Management with Cognitive Multi-Agent Systems." In *17th International Conference on Agents and Artificial Intelligence*. SciTePress, 2025.
- [10]. Sen, Prothit, and Sai Mihir Jakkuraju. "Modeling AI-Human Collaboration as a Multi-Agent Adaptation." *arXiv preprint arXiv:2504.20903* (2025).
- [11]. Pal, Dheeraj Kumar Dukhiram, Subrahmanyasarma Chitta, Venkata Sri Manoj Bonam, Pranadeep Katari, and Shashi Thota. "AI-Assisted Project Management: Enhancing Decision-Making and Forecasting." *J. Artif. Intell. Res* 3 (2023): 146-171.
- [12]. Klieger, Benjamin, Charis Charitsis, Miroslav Suzara, Sierra Wang, Nick Haber, and John C. Mitchell. "ChatCollab: Exploring Collaboration Between Humans and AI Agents in Software Teams." *arXiv preprint arXiv:2412.01992* (2024).
- [13]. Augusto, Juan Carlos. "Contexts and context-awareness revisited from an intelligent environments perspective." *Applied Artificial Intelligence* 36, no. 1 (2022): 2008644.
- [14]. Liu, Zhangqi. "Human-AI co-creation: a framework for collaborative design in intelligent systems." *arXiv preprint arXiv:2507.17774* (2025).
- [15]. Mo, Tianjun, Ziyi Jiang, and Qichang Zheng. "Interactive AI agent for code refactoring assistance: A study on decision-making strategies and human-agent collaboration effectiveness." *Academia Nexus Journal* 4, no. 1 (2025).