

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 11 ISSUE : 07 Print / Issue Publication Date: November 2026

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2026.v11i07.006

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

DEVNEXUS AI: AN AI-POWERED UNIFIED SOFTWARE ENGINEERING PLATFORM

Jagdish Zate, Sai Sathe, Prathmesh Mahadik, Shamsundar Aghav
Department of Computer Science and Engineering,
MIT College of Railway Engineering & Research,
Barshi, Maharashtra, India

Prof. S. S. Mule
Assistant Professor, Department of Computer Science and Engineering,
MIT College of Railway Engineering & Research,
Barshi, Maharashtra, India

Abstract— **Modern software development involves multiple interconnected activities, including source-code review, project coordination, documentation, team collaboration, and task management. Managing these activities through separate tools can increase workflow fragmentation and reduce project visibility. This paper presents DevNexus AI, an AI-powered unified software engineering platform designed to integrate essential development activities within a centralized workspace. The system combines project and team management, AI assisted code review, preliminary security screening, automated documentation generation, project-aware conversational assistance, and task and sprint tracking. The implementation uses React and Vite for the frontend, Node.js and Express for backend services, and PostgreSQL for persistent data storage. Role-based access control and JWT-based authentication support controlled project collaboration. The AI-assisted components use configurable language-model integrations to provide contextual software engineering assistance. Functional evaluation includes recorded backend regression tests, frontend tests, browser-based workflows, and database verification. The results demonstrate successful execution of tested application workflows, although AI detection accuracy, scalability, and production security require further independent evaluation. DevNexus AI demonstrates the feasibility of consolidating software engineering activities into an extensible, AI-assisted development environment.**

Keywords— **Artificial Intelligence, Software Engineering, AI-Assisted Code Review, Project Management, Automated Documentation, Large Language Models, Software Development.**

I. INTRODUCTION

Software engineering projects require continuous coordination between developers, project managers, and other stakeholders. Activities such as code inspection, issue identification, documentation, task assignment, and sprint planning are often performed using separate applications. Although specialized tools support individual activities effectively, switching between multiple platforms can introduce fragmented information, inconsistent project context, and additional coordination effort.

Recent advances in artificial intelligence, particularly large language models, have introduced new opportunities for assisting developers with source-code understanding, documentation generation, and software maintenance. However, AI-generated recommendations may be incomplete or inaccurate, making human verification essential.

DevNexus AI is developed as a unified software engineering platform that combines conventional project management functionality with AI-assisted development services. It provides a centralized workspace for managing projects, assigning team responsibilities, reviewing code, generating documentation, interacting with project-aware AI assistance, and tracking tasks and sprints. The primary contribution of this work is the design, implementation, and functional verification of an integrated software engineering environment. Rather than claiming superior AI detection accuracy, the study examines whether these capabilities can operate together through a consistent application architecture and controlled project workflows.

II. RELATED WORK

Recent advances in artificial intelligence have significantly influenced software engineering, particularly in automated code analysis, repository-level understanding, documentation generation, and intelligent development assistance. Several studies have explored the application of

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

large language models (LLMs) to improve software development workflows.

Jimenez et al. [1] introduced SWE-bench, a benchmark for evaluating language models on real-world GitHub issues. Yang et al. [2] proposed SWE-agent, demonstrating how agent-computer interfaces enable automated software engineering tasks. Zhang et al. [3] presented RepoCoder, which uses iterative retrieval and generation to improve repository-level code completion.

Hong et al. [4] introduced MetaGPT, a multi-agent framework for collaborative software development. Chen et al. [5] investigated the capabilities of large language models for code generation. Luo et al. [6] developed RepoAgent, an LLM-powered framework for automated repository-level code documentation.

Lewis et al. [7] introduced Retrieval-Augmented Generation (RAG), which combines information retrieval with language generation to produce context-informed responses. Qian et al. [8] presented ChatDev, demonstrating communication-driven collaboration among AI agents for software development.

Pearce et al. [9] examined security weaknesses in AI generated code, emphasizing the importance of security assessment and human verification. Zhang et al. [10] proposed CodeAgent, integrating external tools with AI agents to support repository-level coding tasks. Xia et al. [11] investigated LLM-based software engineering agents, highlighting their capabilities and practical challenges.

Li et al. [12] explored large-scale pre-training for automated code review activities, while Kim et al. [13] examined automated code review practices and developer experiences in industrial environments. Dingsøyr et al. [14] reviewed agile software development methodologies, emphasizing their evolution and relevance to collaborative development. Kalliamvakou et al. [15] investigated the opportunities and limitations of mining GitHub repositories for software engineering research.

These studies demonstrate significant progress in AI assisted software engineering. However, DevNexus AI focuses on integrating selected capabilities—AI-assisted code review, security-oriented analysis, automated documentation, contextual assistance, and collaborative project management—within a unified web-based development environment.

TABLE I. RELATED WORK AND DESIGN IMPLICATIONS

Reference	**Approach or contribution**	**Relevance and limitation**
[1]	SWE-bench	Real-world software issue benchmark
[2]	SWE-agent	Automated software engineering agent
[3]	RepoCoder	Repository-level code completion
[4]	MetaGPT	Multi-agent software development
[5]	Evaluating LLMs Trained on Code	AI-based code generation
[7]	Retrieval-Augmented Generation	Context-aware information retrieval
[9]	Copilot Security	Security analysis of AI-generated code

Research Gap

Existing research demonstrates substantial progress in AI assisted code generation, repository-level analysis, automated documentation, and intelligent development agents. However, these capabilities are frequently studied as individual technical problems rather than as components of an integrated collaborative engineering environment.

The practical problem addressed by DevNexus AI is the fragmentation of project information and development activities across separate tools. This work investigates the feasibility of combining AI-assisted engineering services with project membership management, role-based access control, task coordination, sprint tracking, and persistent project records.

The contribution is an implemented integrated platform and evidence from functional verification. The study does not establish that DevNexus AI outperforms specialized tools or improves developer productivity without controlled comparative evaluation.

III. STUDY METHOD

A. Research Design

This study follows a design-and-implementation methodology to develop and evaluate DevNexus AI as an integrated software engineering platform. The work involves identifying development workflow requirements, designing a modular application architecture, implementing core functional modules, and validating the resulting system through automated and browser-based testing.

B. System Development

The frontend is developed using React, Vite, and Tailwind CSS to provide an interactive user interface. The backend uses Node.js and Express for API processing, while PostgreSQL maintains persistent records of users, projects,

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

memberships, tasks, sprints, comments, and related activities.
AI-assisted features use configurable language-model integrations for code review, documentation generation, and contextual assistance. Authentication and project-level authorization are implemented to control access to shared resources.

C. Evaluation Method

System verification includes backend regression testing, frontend unit testing, mocked browser scenarios, real backend browser acceptance tests, and database-level checks. The recorded development milestone reported 288 passing backend regression tests, 54 mocked browser tests, seven frontend unit tests, nine real-backend browser acceptance groups, and 38 focused live database test nodes.
These results provide evidence of functional correctness for the tested workflows. They do not independently establish AI-generated output accuracy, vulnerability-detection performance, production scalability, or deployment security.

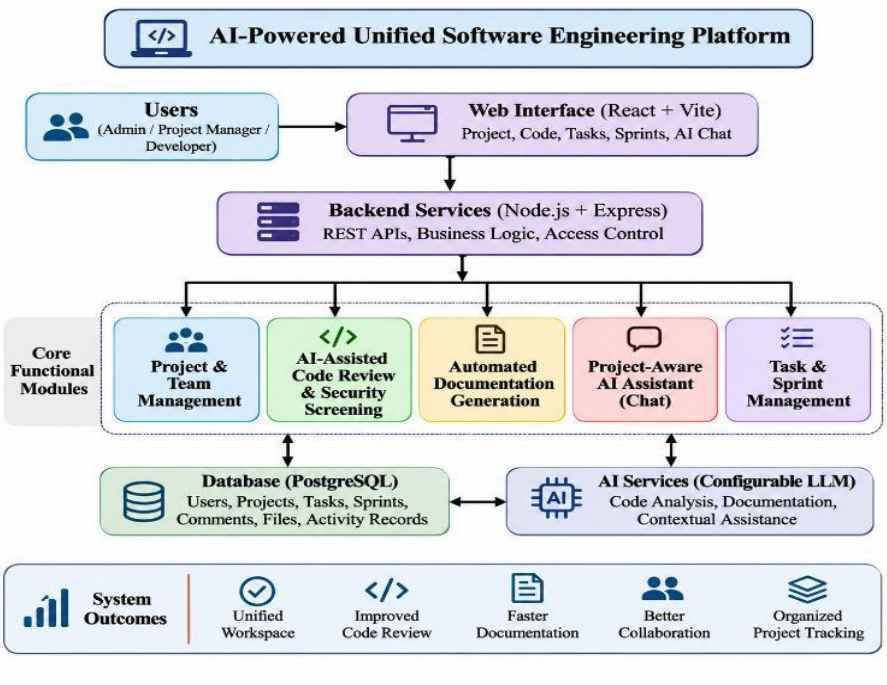


Fig. 1. System architecture and workflow of the proposed AI-powered unified software engineering platform.

Evaluation design. The evaluation of DevNexus AI follows a development-stage verification approach designed to assess functional correctness, system integration, authorization enforcement, and database persistence. The evaluation combines backend regression testing, frontend unit testing, mocked browser scenarios, real-backend acceptance testing, and PostgreSQL database verification. Each testing category examines a different aspect of system behavior, providing complementary evidence of implementation reliability.

Backend verification. Automated regression tests examine REST API operations, authentication, request validation, project membership management, role-based authorization, task assignment, and application business logic. At the recorded development milestone, 288 backend regression tests passed, demonstrating successful execution of the tested server-side operations.

Frontend verification. Seven frontend unit tests and 54 mocked browser tests passed. These tests examine interface behavior, component interactions, navigation, and responses to controlled API conditions. Mocked testing helps identify interface-level defects independently of live backend availability.

Integration verification. Nine real-backend browser acceptance groups were completed to evaluate selected workflows across the frontend, backend, and database. These scenarios provide evidence that application operations can be executed through the user interface and that resulting changes are reflected in persistent records.

Database verification. PostgreSQL validation included 38 focused live database test nodes covering selected database operations, membership relationships, task assignments,

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

comments, and activity tracking. These checks support the correctness of the tested persistence workflows within the configured development environment.

Evaluation boundary. The reported results originate from a recorded development milestone and do not constitute an independent production assessment. The evaluation does not establish AI-output accuracy, vulnerability-detection precision or recall, concurrent-user performance, or resistance to adversarial attacks. Further evaluation using independently prepared code samples, controlled benchmarks, load testing, and security assessment is required to determine the platform's effectiveness beyond functional integration.

IV. SYSTEM ARCHITECTURE

DevNexus AI adopts a modular, layered web application architecture that integrates project management, collaborative development workflows, and AI-assisted software engineering capabilities within a unified environment. The architecture consists of four principal layers: the presentation layer, application services layer, data persistence layer, and AI integration layer. This separation enables independent development, maintenance, and extension of individual system components.

The presentation layer is implemented using React, Vite, and Tailwind CSS, providing interactive interfaces for project administration, team collaboration, code review, documentation generation, AI-assisted chat, and task and sprint management. The application services layer uses Node.js and Express to expose RESTful APIs, process requests, enforce business rules, and implement JWT-based authentication and role-based access control.

PostgreSQL serves as the primary relational database, maintaining persistent information about users, projects, memberships, tasks, sprints, comments, and activity records. The AI integration layer communicates with configurable language-model services to support source-code analysis, contextual assistance, and automated documentation generation. AI-generated responses are returned through backend services for presentation and review.

The system follows a request-response workflow in which user actions are processed by the frontend, validated by backend services, and directed to the appropriate database operations or AI-assisted modules. Project-level authorization checks restrict access to protected resources. The overall system architecture is presented in Fig. 1.

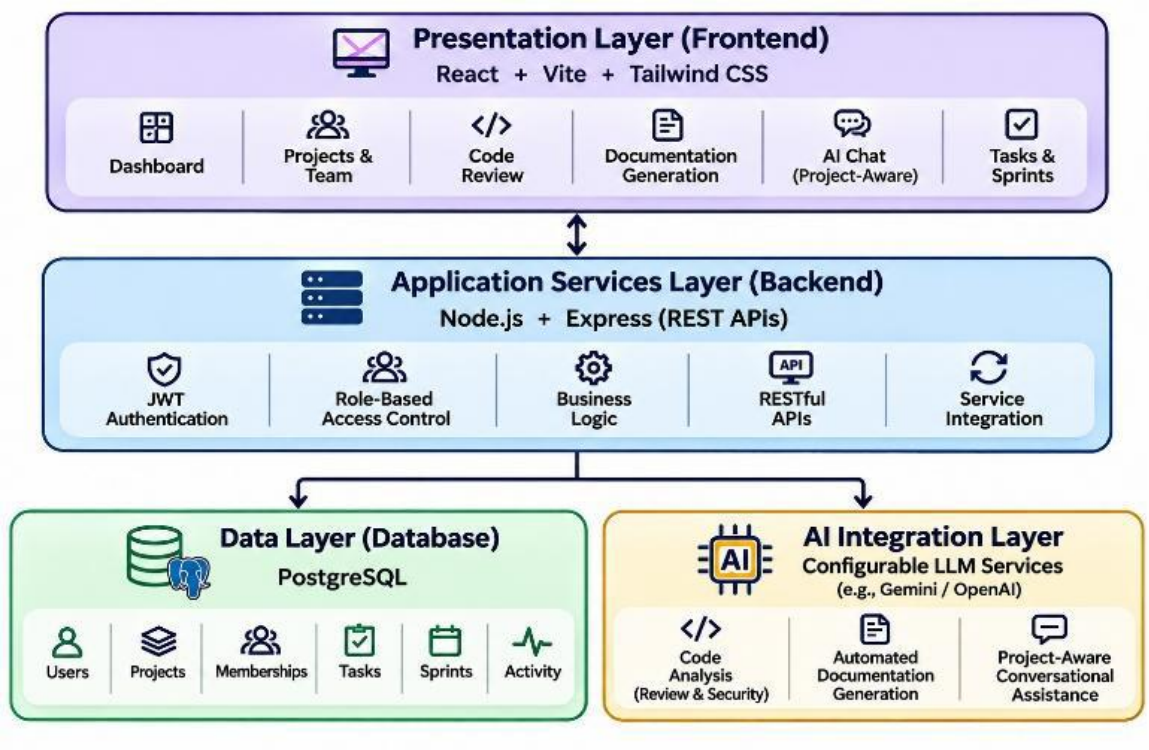


Fig. 2. Layered system architecture of DevNexus AI illustrating the presentation layer, backend application services, PostgreSQL database, and configurable AI integrations.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

TABLE II. COMPONENT RESPONSIBILITIES AND DOCUMENTED STATUS

Component	**Main functions**	**Documented status**
Web Interface	Project dashboard, team collaboration, code review, AI chat, documentation and task management.	Implemented and browser-tested.
Backend API	REST endpoints, request validation, authentication, authorization and business logic.	Implemented; regression tests passed.
Database	Persistent storage of users, projects, memberships, tasks, sprints and activity records.	PostgreSQL operations verified.
AI Integration	Configurable language-model services for code analysis, documentation and contextual assistance.	Integrated; output accuracy not independently established.
Component	**Main functions**	**Documented status**
Task and Sprint Services	Task assignment, status updates, sprint coordination, comments and activity tracking.	Selected workflows verified.

Security and data governance. DevNexus AI incorporates authentication, project-level authorization, and relational data management to support controlled access to software engineering resources. JWT-based authentication protects backend API operations, while role-based access control assigns permissions according to project membership. The system distinguishes Owner, Manager, Developer, and Viewer roles, enabling controlled project administration and collaboration.

PostgreSQL maintains persistent records of users, projects, memberships, tasks, sprints, comments, and activity information. Backend validation and authorization checks are applied to protected operations. AI-assisted requests are processed through configured service integrations, and generated responses require developer review before being treated as reliable technical findings.

Although these mechanisms provide foundational application security, the recorded evaluation does not establish resistance to all security threats. Independent penetration testing, comprehensive access-control auditing, and production deployment validation remain necessary for broader operational use.

V. SYSTEM WORKFLOW AND IMPLEMENTATION

A. Authentication And Project Access Control

DevNexus AI implements a controlled authentication and authorization workflow to manage access to project resources. Users register and authenticate through the web interface, while backend services validate credentials and issue JWT-based authentication tokens. Protected API endpoints require valid authentication before processing requests.

Project-level authorization is enforced through four membership roles: Owner, Manager, Developer, and Viewer. These roles determine the operations available to each user, including project administration, membership management, and access to development resources. Authorization checks are performed by backend services rather than relying solely on frontend interface restrictions.

B. AI-Assisted Code Review

The code review module enables developers to submit source code for AI-assisted examination. After request validation, the backend communicates with a configured language-model service and processes the returned analysis. Review information is presented through the user interface to assist developers in identifying potential defects, maintainability concerns, and security-related issues.

The module provides advisory analysis rather than a formally validated vulnerability-detection system. AI generated findings may contain omissions or inaccuracies and therefore require human verification.

C. Automated Documentation Generation

The documentation module supports the generation of project-related technical documents using AI-assisted processing. Users select a document type and initiate a generation request through the application

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

interface. Backend services process the request using available project context and return generated content for review.

D. Project-Aware Conversational Assistance

The AI chat module provides contextual software engineering assistance by incorporating selected project or repository information into model requests. This allows responses to reflect relevant development context while keeping project interaction within the unified workspace.

E. Task and Sprint Coordination

The task and sprint management module supports task creation, assignment, status tracking, sprint organization, comments, and activity history. Backend services validate operations and persist changes in PostgreSQL. Authorized team members can retrieve updated project information through the application interface, supporting coordinated development workflows. The module also maintains task-related activity records, allowing authorized team members to monitor project progress and review updates. By integrating task assignments, sprint planning, and activity tracking within a centralized workspace, DevNexus AI supports better coordination and provides consistent visibility into ongoing development activities.

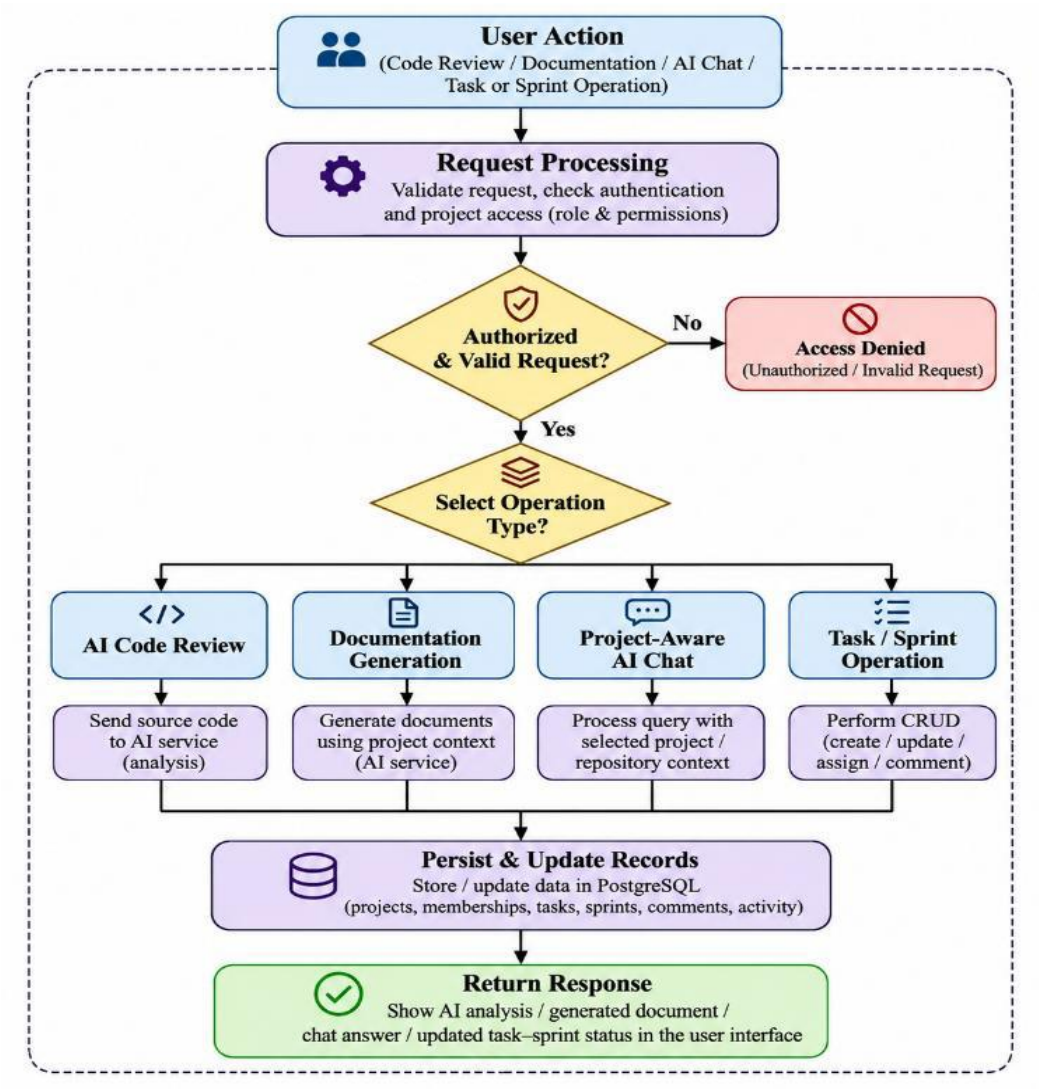


Fig. 3. Workflow of user request validation, authorization, and operation processing in DevNexus AI.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

TABLE III. SYSTEM RESPONSE TYPES AND PROCESSING OUTCOMES

Response type	**Trigger**	**Required interpretation**
Access denied	Invalid authentication or insufficient project permissions.	Rejects the protected operation.
Code review	User submits source code for analysis.	Returns AI-assisted review findings or feedback.
Documentation	User requests a project document.	Returns generated documentation for review.
AI chat	User submits a project-related question.	Provides a response using available project context.
Response type	**Trigger**	**Required interpretation**
Task update	Authorized user creates or modifies a task.	Updates the task and persists the change.
Sprint operation	User assigns or removes tasks from a sprint.	Updates sprint membership and associated records.
Service error	Request processing or an external AI service fails.	Returns an error response rather than a successful result.

VI. INTERFACE AND DEPLOYMENT STATUS

Interface behaviour. DevNexus AI provides a web-based interface designed to integrate multiple software engineering activities within a centralized workspace. The React-based frontend includes project dashboards, team management, AI assisted code review, automated documentation, project aware conversational assistance, and task and sprint coordination. Users interact with these modules through a consistent navigation structure, while available operations are controlled according to project membership and assigned permissions.

The interface supports project creation, member administration, task assignment, sprint management, comments, and activity tracking. AI-assisted features allow developers to submit source code for review, request technical documentation, and obtain contextual assistance. Generated responses are displayed within the application and remain subject to developer verification.

Deployment Status. The platform has been developed and evaluated primarily within a local development environment using React, Node.js, Express, and PostgreSQL. Backend services expose REST APIs for communication with the frontend, while configurable AI integrations support selected analysis and generation operations.

The recorded development milestone included successful backend regression testing, frontend verification, real backend browser acceptance testing, and database-level checks. These results support the functionality of tested application workflows but do not establish production deployment readiness.

Further work is required to validate deployment configuration, recovery procedures, network security controls, scalability, and operational monitoring before the platform can be considered suitable for broader production use.

VII. EVALUATION RESULTS

DevNexus AI successfully passed 288 backend tests, 54 browser tests, seven frontend unit tests, nine real-backend acceptance groups, and 38 database test nodes. The results confirm the functionality of tested project management, authorization, collaboration, and data persistence workflows.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

AI accuracy and production scalability require further evaluation.

TABLE IV. DEVNEXUS AI FUNCTIONAL VERIFICATION RESULTS

Testing Method	Tests / Groups / Nodes	Result
Backend regression	288 tests	Passed
Mocked browser	54 tests	Passed
Frontend unit	7 tests	Passed
Real-backend acceptance	9 groups	Passed
Live database verification	38 nodes	Passed

The evaluation results demonstrate successful verification of DevNexus AI across multiple testing levels. A total of 288 backend regression tests, 54 mocked browser tests, seven frontend unit tests, nine real-backend acceptance groups, and 38 live database test nodes passed. These results provide evidence of functional correctness, system integration, and database persistence within the tested environment.

The verified workflows include authentication, role-based access control, project membership management, task assignment, sprint coordination, comments, and activity tracking. Browser acceptance tests further confirmed selected interactions between the frontend, backend services, and PostgreSQL database. Although these findings demonstrate the feasibility of the integrated platform, AI-generated output accuracy, vulnerability-detection effectiveness, and production scalability require further evaluation.

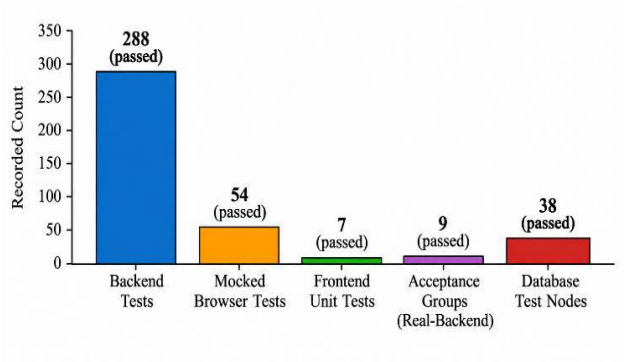


Fig. 4. Functional verification results of DevNexus AI across backend, frontend, browser, and database testing.

TABLE V. SUMMARY OF VERIFIED SYSTEM WORKFLOWS

Module	Verification outcome
Authentication	Passed
Role-based access control	Passed
Project and team management	Passed
Task and sprint management	Passed
Comments and activity tracking	Passed
Database persistence	Passed
AI-assisted services	Successfully integrated and functionally verified.

Functional Evaluation. The functional evaluation of DevNexus AI was conducted through multiple verification methods to assess the correctness and integration of its software engineering modules. The recorded development milestone reported 288 passing backend regression tests, 54 mocked browser tests, seven frontend unit tests, nine successful real-backend browser acceptance groups, and 38 passing live database test nodes. Backend verification covered authentication, API request validation, project membership management, role-based authorization, and task-related operations. Frontend and browser tests examined interface behavior, navigation, and selected user interactions. The results demonstrate successful execution of tested workflows and provide evidence of functional stability within the configured development environment.

System Reliability and Integration Analysis. The integration assessment focused on communication between the React-based frontend, Node.js and Express backend, and PostgreSQL database. Real-backend browser acceptance scenarios verified selected workflows involving project membership updates, task assignment, sprint coordination, comments, and activity tracking. Database-level checks examined persistence operations and relationships between project records. The combined verification approach helped identify inconsistencies between user-interface behavior and stored application data. Successful completion of these tests supports the reliability of the evaluated workflows, although it does not establish system availability or performance under production workloads.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 43-52
Published Online November 2026 in IJEAST (http://www.ijeast.com)

AI Integration and Implementation Assessment. DevNexus AI combines conventional project-management functionality with AI-assisted code review, automated documentation generation, and project-aware conversational assistance. The AI modules communicate with configurable language-model services to process source code, generate technical content, and provide contextual development support. These capabilities demonstrate the integration of AI assisted operations within a centralized software engineering environment. However, successful execution of AI requests does not independently establish the correctness of generated outputs or the accuracy of security-related findings. Further evaluation using independently labeled code samples, controlled documentation-quality assessments, and representative development tasks would provide stronger evidence of AI effectiveness. The current findings primarily establish functional integration and provide a foundation for subsequent improvements in system performance, scalability, and security.

Evaluation plan

Future evaluation of DevNexus AI will assess AI-assisted code review accuracy, documentation quality, and contextual chat reliability using verified test cases. Additional testing will measure response time, scalability, access control, and system recovery to establish overall performance and deployment readiness.

TABLE VI. PROPOSED EVALUATION FRAMEWORK FOR DEVNEXUS AI

Evaluation area	**Test method / measure**	**Evidence to report**
AI Code Review	Analyse source code with known defects and vulnerabilities.	Detection accuracy, precision, recall.
Documentation	Evaluate generated technical documents against project requirements.	Correctness, completeness, relevance.
AI Chat	Test project-related questions using verified reference answers.	Response accuracy, contextual relevance.
Access Control	Verify authorized and unauthorized project operations.	Permission enforcement results.
Performance	Measure response times under varying workloads.	Latency, throughput, error rates.
Database Reliability	Verify data persistence and recovery operations.	Data consistency and recovery results.

VIII. DISCUSSION AND LIMITATIONS

The evaluation results demonstrate the feasibility of integrating AI-assisted development capabilities with conventional software engineering workflows within a unified platform. Successful backend, frontend, browser, and database verification provides evidence of functional integration, role-based authorization, and reliable data persistence under controlled development conditions.

Despite these findings, several limitations remain. The evaluation primarily focuses on functional correctness rather than comparative developer productivity or AI-generated output quality. Although the platform supports AI-assisted code review, automated documentation, and contextual assistance, their accuracy has not been independently established through standardized benchmarks.

Furthermore, testing was conducted mainly within a local development environment. Production-level performance, concurrent-user scalability, advanced security assessment, and disaster recovery require additional validation. The recorded results therefore demonstrate implementation feasibility rather than comprehensive production readiness.

IX. CONCLUSION AND FUTURE WORK

DevNexus AI is an AI-powered unified software engineering platform integrating project management, team collaboration, AI-assisted code review, security screening, automated documentation, and task and sprint management. Developed using React, Node.js, Express, and PostgreSQL, the platform provides a centralized and secure development environment.

Functional evaluation recorded 288 passing backend tests, 54 mocked browser tests, seven frontend unit tests, nine real backend acceptance groups, and 38 database test nodes, demonstrating successful system integration and workflow verification.

Future work will focus on improving AI accuracy, vulnerability detection, repository-context retrieval, scalability, and production security. Overall, DevNexus AI demonstrates the feasibility of integrating AI-assisted capabilities with collaborative software engineering workflows.

ACKNOWLEDGMENT

The authors sincerely thank Prof. S. S. Mule for valuable guidance, encouragement, and technical support throughout the development of DevNexus AI. They also express gratitude to the Department of Computer Science and Engineering, MIT College of Railway Engineering and Research, Barshi, for providing the academic support and resources necessary for the successful completion of this research.

X. REFERENCES

[1]. Jimenez Carlos E., Yang John, Wettig Alexander, Yao Shunyu, Pei Kexin, Press Ofir, and Narasimhan Karthik R. (2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues?, in Proceedings of the International Conference on Learning Representations (ICLR), arXiv:2310.06770.

[2]. Yang John, Jimenez Carlos E., Wettig Alexander, Lieret Kilian, Yao Shunyu, Narasimhan Karthik, and Press Ofir (2024). SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, in Advances in Neural Information Processing Systems (NeurIPS), vol. 37, DOI: 10.52202/079017-1601.

[3]. Zhang Fengji, et al. (2023). RepoCoder: Repository-Level Code Completion Through Iterative Retrieval and Generation, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), (pp. 2471–2484), DOI: 10.18653/v1/2023.emnlp-main.151.

[4]. Hong Sirui, et al. (2024). MetaGPT: Meta Programming for a Multi Agent Collaborative Framework, in Proceedings of the International Conference on Learning Representations (ICLR), arXiv:2308.00352.

[5]. Chen Mark, et al. (2021). Evaluating Large Language Models Trained on Code, arXiv preprint, arXiv:2107.03374.

[6]. Luo Qinyu, et al. (2024). RepoAgent: An LLM-Powered Open Source Framework for Repository-Level Code Documentation Generation, in Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), (pp. 436–464), DOI: 10.18653/v1/2024.emnlp demo.46.

[7]. Lewis Patrick, et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, (pp. 9459– 9474), arXiv:2005.11401.

[8]. Qian Chen, et al. (2024). ChatDev: Communicative Agents for Software Development, in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), (pp. 15174–15186), DOI: 10.18653/v1/2024.acl-long.810.

[9]. Pearce Hammond, Ahmad Baleegh, Tan Benjamin, Dolan-Gavitt Brendan, and Karri Ramesh (2022). Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions, in Proceedings of the IEEE Symposium on Security and Privacy (SP), (pp. 754–768), DOI: 10.1109/SP46214.2022.9833571.

[10]. Zhang Kechi, Li Jia, Li Ge, Shi Xianjie, and Jin Zhi (2024). CodeAgent: Enhancing Code Generation with Tool-Integrated Agent Systems for Real-World Repo-Level Coding Challenges, in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), (pp. 13643–13658), DOI: 10.18653/v1/2024.acl-long.737.

[11]. Xia Chunqiu Steven, Deng Yinlin, Dunn Soren, and Zhang Lingming (2025). Demystifying LLM-Based Software Engineering Agents, Proceedings of the ACM on Software Engineering, DOI: 10.1145/3715754. Presented at the ACM International Conference on the Foundations of Software Engineering (FSE 2025).

[12]. Li Zhiyu, Lu Shuai, Guo Daya, et al. (2022). Automating Code Review Activities by Large-Scale Pre-training, in Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), (pp. 1035–1047), DOI: 10.1145/3540250.3549081.

[13]. Kim Hyungjin, Kwon Yonghwi, Joh Sangwoo, Kwon Hyukin, Ryou Yeonhee, and Kim Taeksu (2022). Understanding Automated Code Review Process and Developer Experience in Industry, in Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), (pp. 1398–1407), DOI: 10.1145/3540250.3558950.

[14]. Dingsøyr Torgeir, Nerur Sridhar, Balijepally VenuGopal, and Moe Nils Brede (2012). A Decade of Agile Methodologies: Towards Explaining Agile Software Development, Journal of Systems and Software, Vol. 85, Issue 6, (pp. 1213–1221), DOI: 10.1016/j.jss.2012.02.033.

[15]. Kalliamvakou Eirini, Gousios Georgios, Blincoe Kelly, Singer Leif, German Daniel M., and Damian Daniela (2014). The Promises and Perils of Mining GitHub, in Proceedings of the 11th Working Conference on Mining Software Repositories (MSR), (pp. 92–101), DOI: 10.1145/2597073.2597074.

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

AUTHORS

- Jagdish Zate
- Sai Sathe
- Prathmesh Mahadik
- Shamsundar Aghav
- Prof. S. S. Mule

ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS

H-INDEX	i10-INDEX	IJEAST CITATIONS	PEER-REVIEWED & OPEN ACCESS
33	154	9,647 (IJEAST CITATIONS)	PUBLISHED MONTHLY

PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.

AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.

OPEN ACCESS

Free and unrestricted access to research for all.

GLOBAL REACH

Connecting researchers and professionals worldwide.

TIMELY PUBLICATION

We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

IJEAST

editor@ijeast.com
www.ijeast.com
India

2455-2143