

AI - ASSISTED BUG TRACKING AND MANAGEMENT SYSTEM

Sasikala P

Assistant Professor
Department of computer science
Sri Shakthi Institute of
Engineering and technology
Coimbatore, India
sasikalapcse@srishakthi.ac.in

Mahalakshmi K

Department of computer science
Sri Shakthi Institute of
Engineering and technology
Coimbatore, India
mahalakshmik22cse@srishakthi.ac.in

Paruthiya R

Department of computer science
Sri Shakthi Institute of
Engineering and technology
Coimbatore, India
paruthiyar22cse@srishakthi.ac.in

ABSTRACT - The AI-Assisted Bug Tracking and Management System is an intelligent, full-stack software solution designed to modernize and automate the process of identifying, tracking, prioritizing, and resolving software defects through the integration of artificial intelligence and scalable web technologies. Built using a robust technology stack including React 18, Vite, Node.js, Express.js, FastAPI, and MongoDB, the system delivers a secure, responsive, and high-performance platform for efficient bug lifecycle management. The architecture follows a modular microservices-inspired design, where a Node.js + Express backend manages core application logic, powered by FastAPI handles intelligent bug analysis, classification, and recommendation workflows. The frontend, developed with React 18 and enhanced using Tailwind CSS and React Router, delivers a responsive and intuitive interface for seamless team collaboration. Real-time notifications are powered by Socket.IO, while secure authentication is enforced using JWT-based role-based access control. By combining AI-driven insights with modern DevOps practices and a highly interactive and The AI-Assisted Bug Tracking and Management System enhances developer productivity, and improves overall software quality through intelligent automation and seamless collaboration.

Keywords:

AI-Based Bug Tracking, Bug Classification, Real-Time Tracking, Automated Prioritization, Defect Management, Socket.IO, MERN Stack, Role-Based Access Control, JWT Authentication, Natural Language Processing, Duplicate Detection.

I. INTRODUCTION

The AI-Assisted Bug Tracking and Management System represents a modern and intelligent approach to software defect management by integrating artificial intelligence, real-time collaboration, and scalable full-stack technologies. This system is designed to streamline the process of identifying, reporting, categorizing, and resolving software bugs efficiently, reducing manual effort and improving development productivity. Built using a robust technology stack including React 18, Node.js, Express.js, MongoDB, and FastAPI, the system ensures high performance, scalability, and seamless user experience.

At the core of the platform is an AI-powered analysis engine capable of interpreting bug descriptions, logs, and user inputs to automatically classify issues, predict severity levels, detect duplicate bugs, and suggest potential fixes. The system incorporates real-time communication using Socket.IO, enabling instant updates on bug status and team collaboration. Additionally, it provides features such as file uploads, report generation, and intelligent dashboards for monitoring bug trends. Secure authentication using JWT and role-based access control ensures data protection and controlled system access, making it suitable for teams of all sizes. While the current system successfully integrates AI-based bug analysis, real-time tracking, and efficient workflow management, several future enhancements are planned to further improve system intelligence, scalability, and user experience.

II. PROBLEM STATEMENT

A. Challenges in Modern Bug Tracking Systems:

In today's fast-paced software development environment, managing bugs efficiently has become a critical challenge for development teams. Traditional bug tracking systems often rely heavily on manual processes, where bugs are reported, analyzed, and prioritized by humans. This approach can be time-consuming, error-prone, and inconsistent, especially in large-scale projects with complex codebases. Developers and testers frequently struggle to accurately classify bugs, determine their severity, and assign priorities. Misclassification or delayed identification of critical issues can lead to increased debugging time and reduced software quality. Without instant updates and notifications, communication gaps arise between team members, leading to delays in bug resolution. Furthermore, the absence of AI-driven insights makes it difficult to identify patterns, predict potential issues, or suggest effective solutions. The lack of an integrated online solution creates fragmented workflows, limited collaboration, and reduced productivity. These challenges highlight the need for a modern, AI-powered bug tracking platform that automates classification, prioritization, and resolution to improve team efficiency and software quality.

B. Limitations of Existing Bug Tracking Systems:

Despite advancements in software development tools, many existing bug tracking systems fail to provide an intelligent, efficient, and user-friendly environment for managing software defects. Most traditional systems rely heavily on manual input, lacking advanced features such as automated bug classification, real-time analysis, and intelligent prioritization. This results in increased workload for developers and slower resolution times. Existing solutions also struggle with scalability and integration. They often lack support for modern technologies, real-time collaboration, and seamless integration with AI services or development tools. These limitations highlight the need for an advanced bug tracking system that incorporates artificial intelligence, automation, and real-time communication to improve efficiency, accuracy, and software quality.

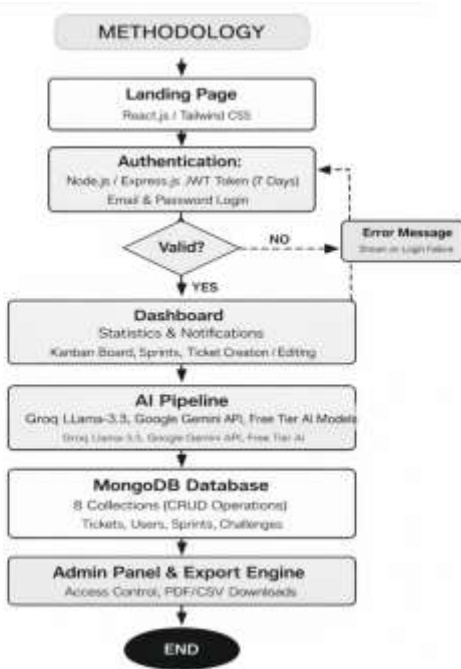
III. EXISTING SYSTEM

In traditional software development environments, bug tracking is often handled using basic tools or standalone systems that rely heavily on manual processes. Developers and testers typically use separate platforms to report, monitor, and resolve bugs, which leads to fragmented workflows and inefficient communication. These systems lack a unified architecture that integrates all stages of the bug lifecycle into a single, streamlined platform. Most existing solutions do not incorporate intelligent automation or AI-based analysis. As a result, bugs must be manually reviewed, categorized, and prioritized, which is time-consuming and prone to human error. Developers often struggle to identify duplicate issues or understand complex bug reports due to inconsistent data and lack of structured input formats. Additionally, traditional systems do not provide real-time updates or seamless collaboration features. Team members must rely on emails or external communication tools to stay updated on bug status, leading to delays in resolution and reduced productivity. There is also limited support for centralized data storage, making it difficult to access bug history, track changes, or maintain proper documentation.

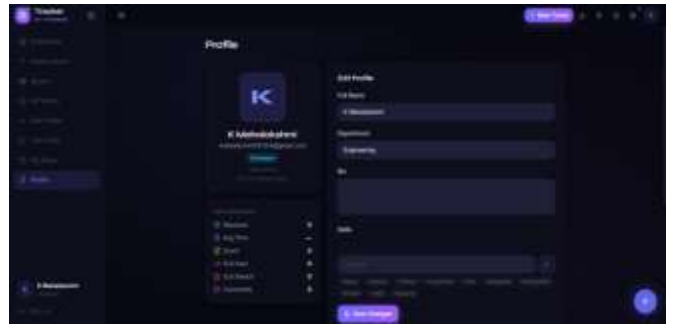
IV. PROPOSED SYSTEM

The proposed system, AI-Assisted Bug Tracking and Management System, is designed to overcome the limitations of traditional bug tracking methods by providing an intelligent, automated, and web-based platform for efficient defect management. It enables users to report, track, and resolve bugs through a centralized interface without relying on manual processes. The system integrates artificial intelligence to automatically classify bugs, detect duplicates, assign priority levels, and provide resolution suggestions, reducing developer workload and improving accuracy. Using natural language processing, it analyzes bug descriptions to generate meaningful insights. The platform offers real-time updates and notifications, ensuring seamless communication and collaboration among developers, testers, and administrators. It features a user-friendly interface, role-based access control, and secure data handling to ensure efficient and safe usage.

V. METHODOLOGY

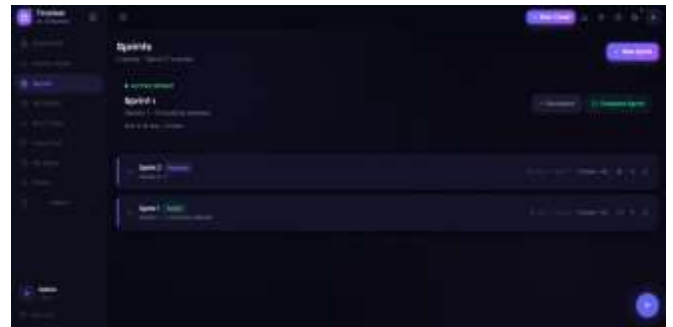


VI. EXPERIMENTAL RESULT

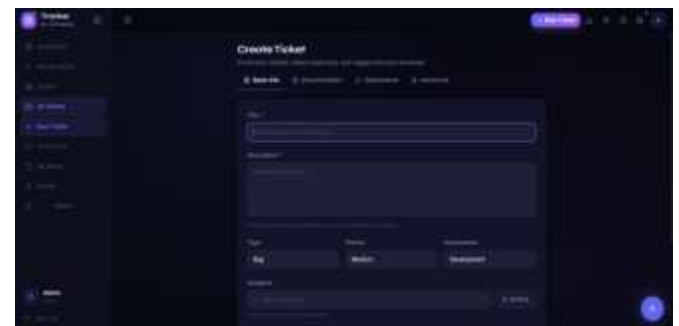


PROFILE PAGE - Displays user information, performance metrics, and role details in a structured dashboard.

SPRINTS PAGE - Displays active and completed sprints with timeline, progress and



ticket details.

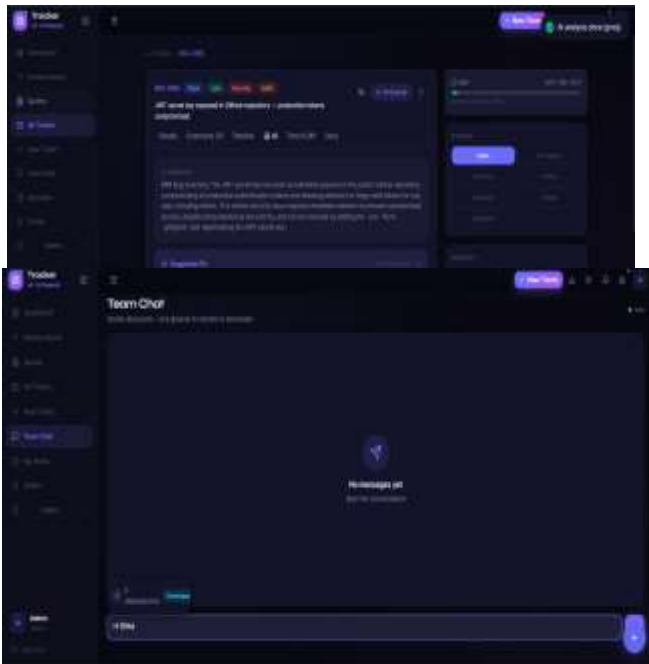
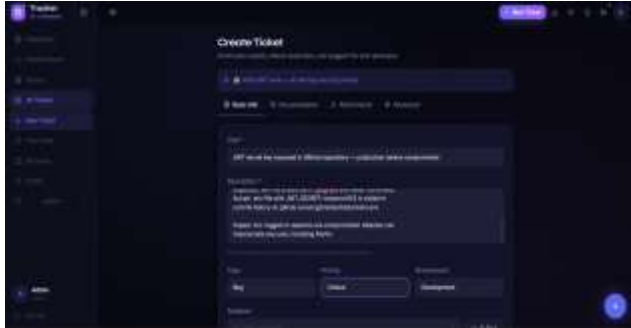


The methodology of the AI-Assisted Bug Tracking and Management System follows a structured, layered process involving the frontend interface, backend services, AI processing pipeline, and database management components. Each layer plays a crucial role in ensuring secure, efficient, and intelligent handling of bug reports within the system. The interaction between these components enables real-time bug tracking, accurate analysis, and seamless collaboration among users. Once authenticated, users are directed to the Dashboard, which provides features such as bug statistics, notifications, Kanban board management, ticket creation, and editing capabilities. This layer enables users to efficiently manage and monitor bug-related activities in real time. A Python FastAPI microservice handles AI inference requests from the Node.js backend, including bug classification, duplicate detection, and fix generation.

The system utilizes an AI Pipeline to analyze bug descriptions, classify issues, detect duplicates, and generate intelligent recommendations, ensuring automated and accurate bug processing. All bug-related data is managed in the MongoDB database, where tickets, user data, and changes are stored with efficient retrieval and updates. The Admin Panel provides functionalities such as access control, report generation, and data export in PDF/CSV formats. Overall, the workflow ensures a secure, scalable, and efficient bug management system.

TICKET CREATE PAGE - Allows user to report new bugs with detailed information such as title, description, priority and environment.

AI CHATBOT PAGE - Provides an intelligent assistant to help users with bug - related queries, errors, and system guidance in real time.



TICKET CREATION PAGE - This AI - powered interface automatically detects high - risk security threats and suggests priority levels in real - time.

SOLUTION PREDICTION PAGE - This system generates instant, high - confidence technical fixes and root cause analyses for complex security vulnerabilities.

TEAM COLLABORATION - This integrated chat hub enables real - time global discussions, keeping every team member aligned and connected.

VII. CONCLUSION

The development of the AI-Assisted Bug Tracking and Management System successfully demonstrates how modern web technologies and artificial intelligence can enhance the efficiency of software defect management. By integrating an intelligent AI pipeline with a user-friendly web-based interface, the system automates bug classification, duplicate detection, and prioritization, reducing manual effort and improving accuracy. Real-time updates, notifications, and collaboration features ensure faster communication among team members, enabling quick identification and resolution of issues. This seamless process improves productivity and supports better software quality.

Overall, the platform serves as a practical and scalable solution for developers, testers, and organizations by providing a centralized environment for managing bugs effectively. It simplifies the entire bug lifecycle—from reporting to resolution—while ensuring secure data handling and structured workflows. The system’s intuitive design and intelligent automation make it suitable for both small teams and large-scale development environments.

In conclusion, this project successfully delivers an advanced and efficient bug tracking system that overcomes the limitations of traditional methods. By combining AI-driven analysis, real-time collaboration, and modern technologies, it significantly reduces debugging time and enhances development efficiency. The system also lays a strong foundation for future enhancements such as advanced analytics, predictive bug detection, and deeper AI integration, making it a powerful tool for improving overall software reliability and performance. The system enhances software development by providing intelligent insights and automated decision-making for bug resolution. It improves team collaboration through real-time updates, notifications, and shared workflows, ensuring smooth communication among developers and testers.

The integration of AI reduces human error and significantly speeds up the debugging process. Its scalable architecture allows it to handle efficiently multiple users and large volumes of bug data. Overall, the project contributes to building reliable and high-quality software through smart and automated defect management.

VIII. FUTURE WORK

Future improvements can include integrating advanced AI models for predictive bug detection, enabling real-time team collaboration features, and enhancing analytics dashboards for deeper insights into bug trends. Additional features such as automated testing integration, CI/CD pipeline support, and mobile application development can further expand the system's capabilities. Moreover, incorporating AI-driven chat assistants and personalized recommendations can improve user experience and make the system more intelligent and adaptive.

IX. REFERENCES

- [1] Abid, A., Farooqi, M., & Zou, J. "Gradio: An Open-Source Python Library for Building AI Web Interfaces." *Proceedings of Machine Learning Systems*, 2022.
- [2] Amershi, S. et al. "Software Engineering for Machine Learning: A Case Study." *IEEE/ACM International Conference on Software Engineering*, 2019.
- [3] Chen, J., et al. "Large Language Models for Automated UI Design and Intelligent Systems." *arXiv Preprint*, 2023; arXiv:2311.09721.
- [4] Thompson, J., et al. "Automated Version Control and Change Tracking in Modern Software Systems." *ACM Transactions on Software Engineering*, 2024.
- [5] Wilson, A., et al. "Real-Time Communication Systems in Collaborative Web Applications." *IEEE Software*, 2024.
- [6] Martinez, R., et al. "NoSQL Databases for Dynamic Data Management: A Study on MongoDB." *International Journal of Database Systems*, 2023.
- [7] Kumar, V., et al. "Efficient Backend Development Using Express.js for Scalable Applications." *Journal of Web Engineering*, 2023.
- [8] Rodriguez, L., et al. "Utility-First CSS Frameworks in Modern Web Applications." *Frontend Engineering Journal*, 2022.
- [9] Zhang, H., et al. "AI-Assisted Software Development: Trends, Tools, and Future Directions." *ACM Computing Surveys*, 2025.
- [10] Xu, L., & Zhao, K. "Scalable REST API Architectures for AI-Driven Applications." *IEEE Cloud Computing*, 2023.
- [11] Newman, S. "Building Microservices: Design, Patterns, and Practices." *O'Reilly Media*, 2022.
- [12] Luo, M., et al. "A Scalable Architecture for AI-Assisted Software Development Platforms." *Software: Practice and Experience (Wiley)*, 2023.
- [13] Lin, T., et al. "A Survey of Large Language Models for Text-to-Code and Software Automation." *arXiv Preprint*, 2023.
- [14] Sharma, N., et al. "Resource Management in AI-Based Systems." *International Journal of Artificial Intelligence Research*, 2024.
- [15] Gomez, H., et al. "CI/CD Pipelines and Automated Deployment in Modern Applications." *Web Engineering Conference (ICWE)*, 2024.
- [16] OpenAI. "GPT-4 Technical Report." *arXiv Preprint*, 2023; arXiv:2303.08774.
- [17] MongoDB Inc. "MongoDB Documentation." *Official Docs*, 2024.