

CodeXpert (An Integrated Web based platform for collaborative project Management and Code Analysis)

Aditya Kumar Gond¹, Abhishek Kr Chaudhary², Shivansh Dwivedi³, Aditya Singh⁴, Prashant Chaturvedi⁵

¹*Aditya Kumar Gond, Buddha Institute of Technology*

²*Abhishek Kumar Chaudhary, Buddha Institute of Technology*

³*Shivansh Dwivedi, Buddha Institute of Technology*

⁴*Aditya Singh, Buddha Institute of Technology*

Abstract—Coding environments don't have integrated communication which in turn requires you to switch to apps like WhatsApp also we see that for AI based coding assistance you have to leave the platform which in turn causes data loss and time inefficiency.

Index Terms—Codexpert, MERN stack, code review, code commenting, bug detection, team management, task assignment, task tracking, collaboration, web application

I. INTRODUCTION:

They see themselves either fighting the steep learning curves of large scale enterprise tools such as Jira, or else they turn to the simple solutions of apps like WhatsApp and basic spreadsheets. This over use of unstructured tools in turn causes break up of workflows, disorganized task updates, and a wide scale lack of accountability which in turn increases the risk of miscommunication and lost project data.

To fill this void we present CodeXpert a new kind of middle way platform which is very easy for the average user to get into but also robust for the technical professional. Based in the MERN (MongoDB, Express, React, Node.js) stack, CodeXpert creates a single, full featured environment that has agile project management and in depth code analysis. At the heart of CodeXpert is our AI based innovation. Where other tools may just be a classic task board, in CodeExpert we have included a fully automated 'Bug Finder' tool and an intelligent code review feature. This not only increases the speed at which you develop but also has the active benefit of getting developers to write better, cleaner and more efficient code. Supported by Socket.IO for real time team synchronization and JSON Web Tokens (JWT) for secure access, CodeXpert provides a modern seamless experience. In

this paper we look at what it takes to put AI directly into the project management workflow which in turn improves team productivity and also we see how it plays a role in promoting professional software engineering practices among green field developers.

II. LITERATURE REVIEW AND RESEARCH GAPS:

EXISTING SYSTEMS AND LITERATURE: In the past few years we have seen a great shift in which the software development lifecycle has adopted two very different sets of tools. On the one hand we have project management platforms like Jira, Trello, and Asana which have become the norm for tracking progress. They allow teams to organize tasks, assign roles, and run workflows. Also we have the growth of Artificial Intelligence which has brought in stand alone tools (for instance ChatGPT or GitHub Copilot) that are dedicated to code optimization, bug detection, and automatic reviews. While both groups of tools are very effective in what they do we see little to no integration between the two.

III. IDENTIFICATION OF RESEARCH GAPS: In our review of present day collaborative software we see a large gap. Although the idea of using AI for code improvement and update is not new, we note the lack of a single unified platform which also handles team management and technical development. Presently there is a void in which developers do not have a single space to put in that they are working on a project, report in detail to their team leaders' progress, and at the same time use AI to fix code issues within the same environment. We see that developers are forced to leave their project management dashboard which they are in the middle of to use a separate AI tool for debugging which they then have to leave again to go

back and update their task status. This breaks focus, slows the development process down and also we see that value time is wasted.

IV. PROPOSED SOLUTION (BRIDGING THE GAP): To solve for this issue we came up with CodeXpert. Our platform which we are very proud of does what no other does it brings together collaborative project tracking and AI based code analysis in one package. We also put in features which enable team members to talk to each other, report to management, and debug code all in the same place which removes the need for constant context switching, thus we see great improvements in flow and also large time savings for the development team.

SYSTEM ARCHITECTURE: CodeXpert is designed into a Three Tier Architecture which puts the user interface, backend logic, and database in separate layers. This in turn gives us high performance at which we can handle heavy AI tasks at the same time as real time team activities.

FRONTEND: Dynamic Single Page Application (SPA).

BACKEND: Centralized server which handles API routing and AI interactions.

DATABASE: Flexible storage for user data and unformatted code logs.

TECHNOLOGY STACK (MERN): The platform is developed with MERN stack for which we use to implement quick and flexible data management

MONGODB: A NoSQL database we use for it's flexibility in storage of many different data types which includes task boards and raw code snippets.

EXPRESS JS NODE JS: we use this to put together a very responsive server environment which scales well for many at once, think team updates and AI API calls.

REACT JS: we use this to build out an interactive, component based user interface.

V. SECURITY AND AI INTEGRATION:

SECURITY: We use JSON Web Tokens (JWT) for user auth and role based access. **AI Code Engine:** We have built in AI which is the main feature. As a user submits code we secure it routing it to a Generative AI API (e.g. Gemini/GPT). The AI takes on the role of an automatic code review which identifies bugs and suggests.

VI. IMPLEMENTATION DETAILS

SECURITY AND AUTHENTICATION: User authentication and session management is done via JSON Web Tokens (JWT). Upon a user's login account details are checked and a token issues out to confirm their identity and role which in turn protects all sensitive API endpoints from unauthorized access. As for the Real Time Engine (Socket.IO) we put in place a solution for delayed updates and muddled communication. It is a bi directional broadcaster. When a team member reports in on project status the server sends out a notice which in turn the UI for all connected users updates in real time, which also removes the need for manual page refresh.

INTERACTIVE UI (KANBAN BOARD): The task management platform uses the react-beautiful-dnd library for a seamless drag and drop experience. As you move a task from one column to another (for instance from 'To-Do' to 'In-Progress' the backend PATCH request is sent out which in turn updates the task's state in the database almost instantaneously which provides for quick turn around.

AI INTEGRATION AND ANALYTICS:

AI CODE REVIEW: Code snippets from users are sent to the Generative AI API through dedicated backend routes. The AI evaluates the logic and returns structured JSON responses which also include suggested improvements. In terms of Dashboard Metrics we implemented MongoDB's Aggregation Pipeline which in turn reports performance metrics very efficiently at the database level thus reducing server load and bandwidth use which in turn improves what the user sees on the dashboard.

VII. RESULTS AND DISCUSSION: **System Performance and Real-Time Sync:** The system was put through test in simulated team settings to study the performance of the MERN architecture and Socket.IO integration. We found that real time task updates (for instance dragging a task from 'To-Do' to 'Completed' were reported to all active users within an average delay of 50 milli seconds. Which in turn proves that the platform does indeed do away with the communication delays which are present in the conventional, refresh based task boards. **AI Code Analysis Study:** To determine the AI's performance

we did a comparison between manual debugging and CodeXpert's AI assisted debugging.

EFFICIENCY: Standard during the manual review which took 10-15 minutes we found and described by the AI engine in under 5 seconds.

ACCURACY: The AI model did in fact find issues in 92% of the code we tested and also provided action items for code improvement along with the fix.

VIII.USER ACCEPTANCE AND DISCUSSION: A we ran a User Acceptance Test (UAT) which included a small group of computer science students and interns. What we found was a large scale of context switching reduction. Users reported that they see great value in having the code base, task board and AI debug assistant all in one place which in turn greatly improved focus and workflow continuity.

DISCUSSION: The results support our initial hypothesis. We put together Project management and AI developed tools in CodeXpert which we found to be a very efficient "in between" solution. We mitigated the large learning curve of enterprise tools like Jira and at the same time gave in depth structure and technical utility which casual apps like WhatsApp do not.

IX.CONCLUSION: CodeExpert is a solution which fills the key gap between what an individual does technically well and what the organization as a whole puts out in terms of productivity. We have designed a dual layered architecture which we present as a full scale environment for today's software development. At the start we introduce the Gemini API which we put forward as a very advanced entry point for independent developers which includes real time error detection, auto comments, and in depth code optimization which we put in place before any code goes into the shared repo. Transitioning beyond individual utility, the post-login environment introduces a robust framework designed to mitigate the inherent bottlenecks of collaborative engineering. By establishing a structured hierarchy of **Team Leaders** and **Members**, the platform formalizes accountability and streamlines communication. The inclusion of granular record tracking and performance monitoring further transforms CodeExpert from a simple coding tool into a high-level management asset.

Ultimately, this research demonstrates that code quality and team synchronization are not independent variables but are "two sides of the same coin." By unifying AI-driven technical support with role-based management, CodeExpert fosters a disciplined development culture capable of scaling to meet the demands of complex, high-stakes projects. Future work will explore the scalability of these features in cross-functional environments and the potential for predictive analytics in team performance.

X.APPENDIX:

A.1 THE AI INSTRUCTION SET (PROMPTING): To get high quality reviews we do not just ask the AI to "fix code". We give it a specific persona. The backend puts the user's code in a system prompt which makes the Gemini API act as a Senior Security Auditor. This in turn gives feedback which is not only that the code works but that it is also secure and efficient.

A.2 REAL TIME SYNC VIA SOCKET.IO: To fix the "WhatsApp-switching" issue which we brought up in our intro we used WebSockets. As a Team Lead moves a task card on the Kanban board a small data packet is sent out to all connected team members. This all happens behind the scenes which in turn updates everyone's screen instantly without them having to hit "Refresh."

APPENDIX B:

DATABASE DESIGN (MONGODB): We have categories of users which we tag by role. This is what puts the "Management" tab at a Leader's disposal and not for a regular Member.

TASK LOGS: As we go along we document which tasks are moved along and which bugs the AI finds. It is from this data that we produce the "Progress Reports" at project end.

XI.ACKNOWLEDGMENT: We would like to thank our project coordinator and mentors at the Buddha Institute of Technology which we had at great support and very useful guidance through out the development of CodeXpert. Also we are very grateful to our Department of Computer Science and Engineering for providing the required infrastructure and resources that made this research a reality.

Special appreciation goes to our peers and the student focus group which we had for User Acceptance Testing; we had from them very frank input which we used to improve the AI integration and real time synchronization features. Also we would like to thank

our families and friends for their constant support and patience through the many late nights of coding and documentation. This project is a product of a team effort, we are thankful to all that played a role, big or small in it.

XII. REFERENCES:

[1] Meta Open Source, "React - A JavaScript library for building user interfaces," React Official Documentation, 2024. [Online]. Available: <https://react.dev/>

[2] OpenJS Foundation, "Node.js v20.x Documentation," Node.js Official Site, 2024. [Online]. Available: <https://nodejs.org/en/docs/>

[3] MongoDB, Inc., "MongoDB Atlas: The multi-cloud developer data platform," MongoDB Official Documentation, 2024. [Online].

Available: <https://www.mongodb.com/atlas>

[4] Auth0, "JSON Web Tokens (JWT) - Introduction and Implementation," JWT Official Website, 2024. [Online]. Available: <https://jwt.io/introduction>

[6] Google DeepMind, "Gemini API Documentation: Generative AI for Developers," Google Cloud, 2024. [Online]. Available: <https://ai.google.dev/docs>

[7] A. K. Sharma and R. Singh, "Comparative Analysis of Agile Project Management Tools in Remote Software Development," *International Journal of Computer Applications*, vol. 182, no. 45, pp. 12-18, 2023.