iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XII **Month of publication:** December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76642

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

Collaborative Web Designing Platform Using Liveblocks

Mrs. Shylaja L N[1], Hindu K R[2], Naveen R[3], Shalini H S[4], Sthavan Jain[5]
[1]*Associate Professor, Dept. of Artificial Intelligence and Machine Learning, Bahubali college of Engineering, Shravanabelagola, Hassan, Karnataka, India*
[2, 3, 4, 5]*UG Student, Dept. of Information Science and Engineering, Bahubali college of Engineering, Shravanabelagola, Hassan, Karnataka, India*

Abstract: ***Collaboration between designers and developers in modern web application development is often fragmented, leading to inefficiencies, delays, and repetitive manual implementation. This paper presents a Collaborative Web Designing Platform using Liveblocks, a web-based system that enables real-time multi-user UI design combined with AI-assisted frontend code generation. The platform allows multiple users to collaboratively design interfaces on a shared canvas with instant synchronization, presence awareness, and conflict-free updates. Visual designs and textual prompts are automatically converted into production-ready frontend code using modern frameworks such as Next.js. A live sandbox preview enables immediate validation and iterative refinement of generated applications. Experimental evaluation demonstrates improved productivity, reduced development time, and enhanced collaboration compared to traditional design-to-development workflows. The proposed system provides a scalable and efficient solution for modern collaborative web development environments.***
Keywords: Collaborative Web Design, Liveblocks, Design-to-Code Automation, Real-Time Collaboration, AI Code Generation, Next.js

I. INTRODUCTION

Web application development typically involves separate phases for design and implementation, often handled by different teams using disconnected tools. This separation results in communication gaps, repeated manual coding, and slower development cycles. Designers create static prototypes, which developers later translate into code, introducing delays and inconsistencies.

To address these challenges, this paper proposes a Collaborative Web Designing Platform using Liveblocks, which unifies real-time UI design and automated code generation in a single environment. The platform enables multiple users to collaboratively design interfaces while simultaneously generating corresponding frontend code. By integrating Liveblocks for real-time synchronization and AI-powered automation, the system eliminates traditional handoff barriers and accelerates the overall development lifecycle.

II. LITERATURE REVIEW

Recent research highlights the importance of real-time collaboration and automation in modern software development. Model-driven web engineering approaches have explored collaborative modeling environments to involve both technical and non-technical stakeholders. Studies on UI-to-code generation demonstrate the potential of automating frontend development from design prototypes using AI and large language models.

Existing collaborative code editors focus mainly on shared coding environments, while design tools often lack real-time multi-user interaction or reliable code generation. Although several systems attempt partial automation, limitations remain in responsiveness, scalability, and integration. The proposed platform builds upon these works by combining real-time collaborative design, AI-based code generation, and live execution within a unified system.

III. SYSTEM ARCHITECTURE

The proposed Collaborative Web Designing Platform using Liveblocks is designed using a modular and scalable architecture that integrates real-time collaboration, AI-based code generation, and live application preview within a single environment. The architecture ensures seamless interaction between users while maintaining low latency, consistency, and reliability. It follows a layered approach where each component is independently responsible for specific functionalities, enabling easier maintenance, extensibility, and future enhancements. The architectural components are as follows:

1) *User Interface Layer:* Provides an interactive and intuitive design environment where users can create and modify web interfaces using a drag-and-drop canvas. This layer is developed using React and Fabric.js, enabling smooth manipulation of UI elements such as shapes, text, and images. The responsive interface ensures an efficient user experience across devices while allowing designers and developers to work simultaneously on shared projects.
2) *Collaboration Layer:* Is responsible for enabling real-time multi-user interaction within the platform. Liveblocks is used to manage shared state synchronization, presence awareness, and live cursor tracking. This layer ensures that any design change made by one user is instantly reflected for all collaborators without conflicts, maintaining consistency and supporting effective teamwork in distributed environments.
3) *Application Logic Layer:* Is implemented using Next.js and handles core system functionalities such as project creation, routing, session management, and communication between frontend and backend components. This layer acts as the central controller, coordinating user actions, collaboration events, and data flow across the system.
4) *AI Code Generation Layer:* Automates the conversion of visual designs and user prompts into structured, production-ready frontend code. By leveraging large language models and prompt engineering techniques, this layer generates optimized Next.js components that accurately represent the designed interfaces. This automation significantly reduces manual coding effort and accelerates the design-to-development workflow.
5) *Data Management Layer:* Ensures reliable storage and retrieval of project-related information. PostgreSQL is used to store design metadata, user prompts, generated code, and version history. This layer provides data persistence, supports version tracking, and enables users to revisit or modify previous designs efficiently.
6) *Execution and Preview Layer:* This layer enables instant validation of the generated frontend code through live, in-browser previews. The platform leverages the E2B sandbox provider to run code inside secure, isolated environments backed by Docker images that are dynamically pulled and instantiated on demand. Using these containerized sandboxes, users can execute, preview, test, and iteratively refine applications in real time—directly within the browser—without requiring local setup or external deployment.

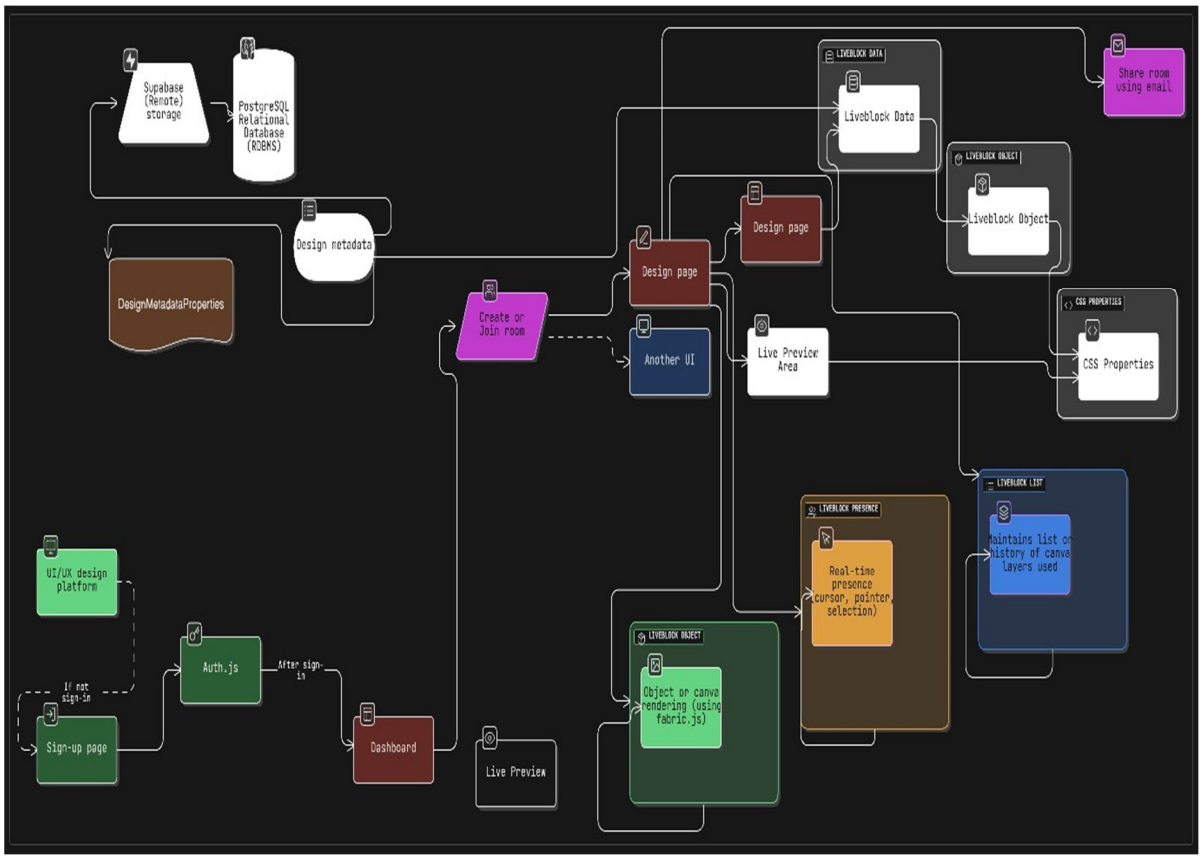


Fig. 1 Collaborative UI Design and Real-Time Editing Architecture.

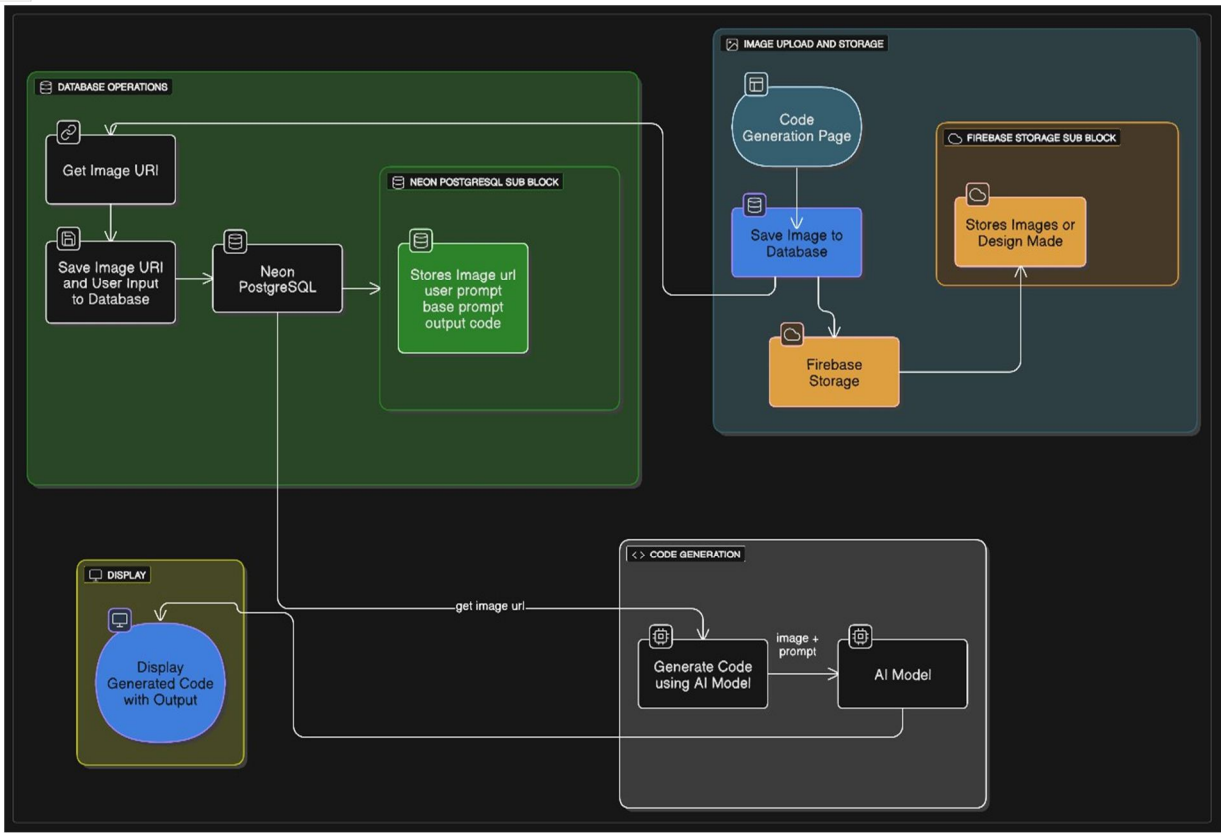


Fig. 2 Code Generation using Firebase, Neon (PostgreSQL), LLM and WebContainer.

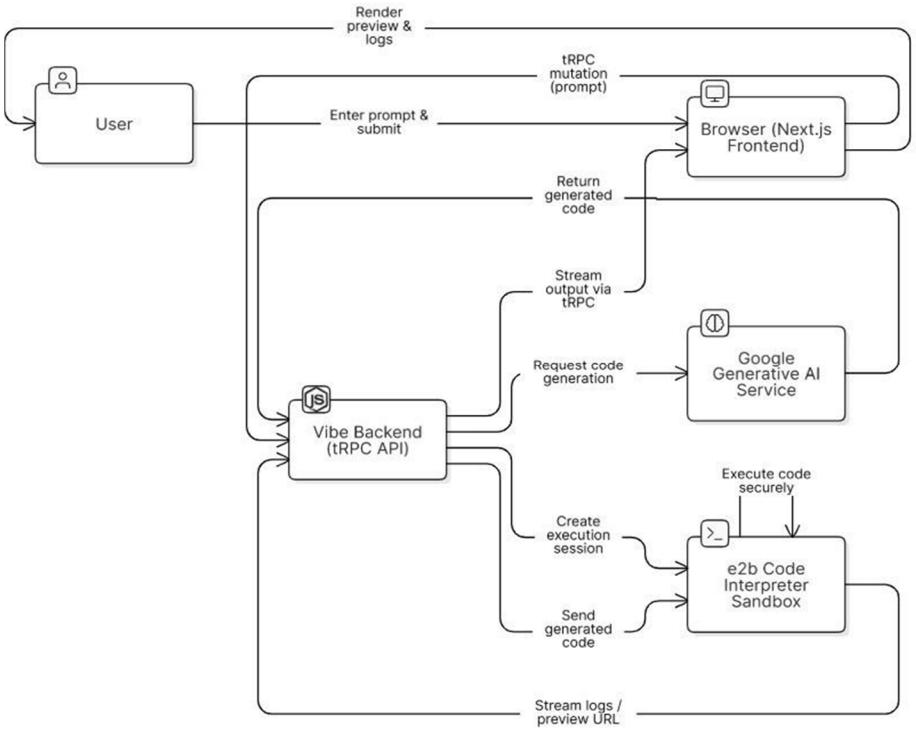


Fig. 3 AI-powered code generation and secure execution flow.

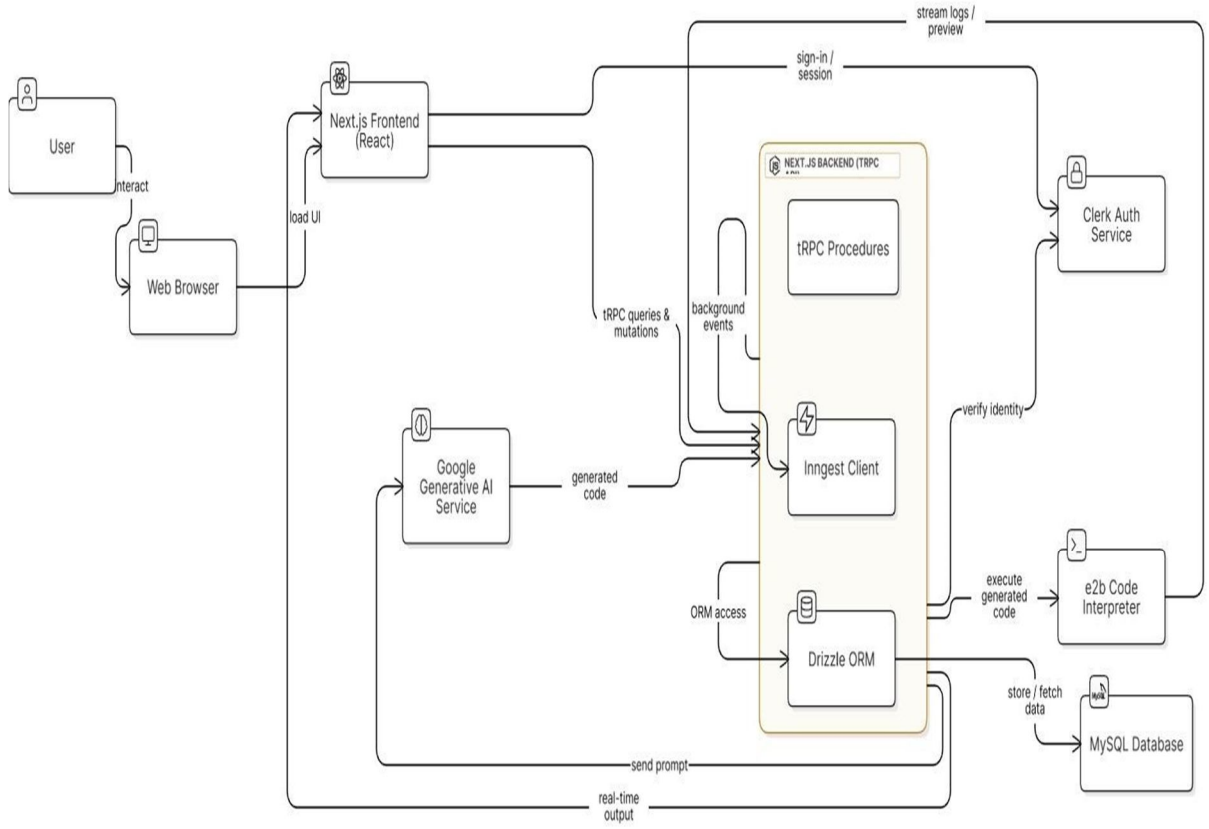


Fig. 4 Backend Orchestration and Event-Driven Processing Architecture

IV. IMPLEMENTATION

The platform allows users to create or join collaborative rooms where design changes are instantly synchronized across participants. Each user's actions, such as adding or modifying UI components, are reflected in real time. The AI engine converts finalized designs into structured Next.js components, organizing files and dependencies automatically. Users can preview the generated application instantly and iteratively refine both design and code within the same environment.

V. RESULTS AND PERFORMANCE ANALYSIS

The system was tested under multiple collaborative scenarios involving concurrent users. Results show that design synchronization latency remained minimal, and generated code accurately reflected visual designs. Compared to traditional workflows, the platform significantly reduced manual coding effort and improved iteration speed. The live preview feature enabled rapid validation, reducing debugging and rework time.The outcomes of the proposed system are presented below.

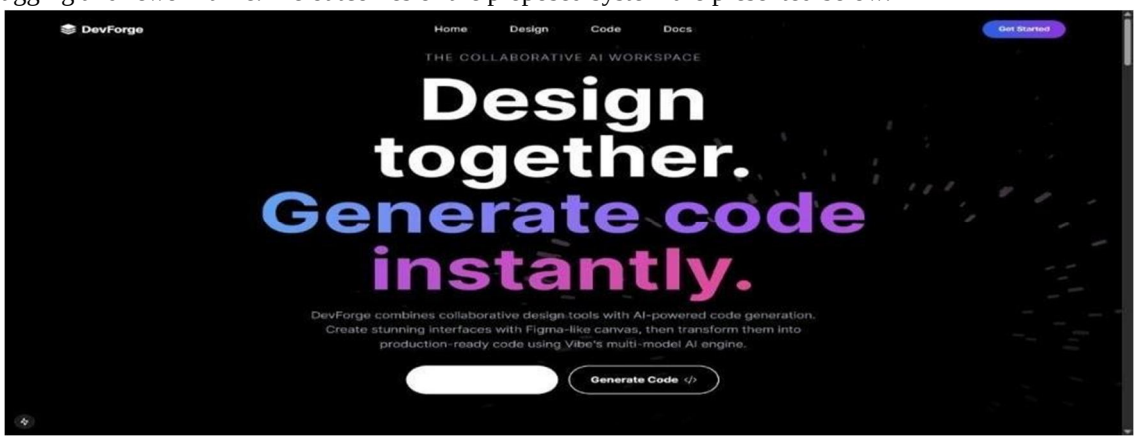


Fig. 5 Home page.

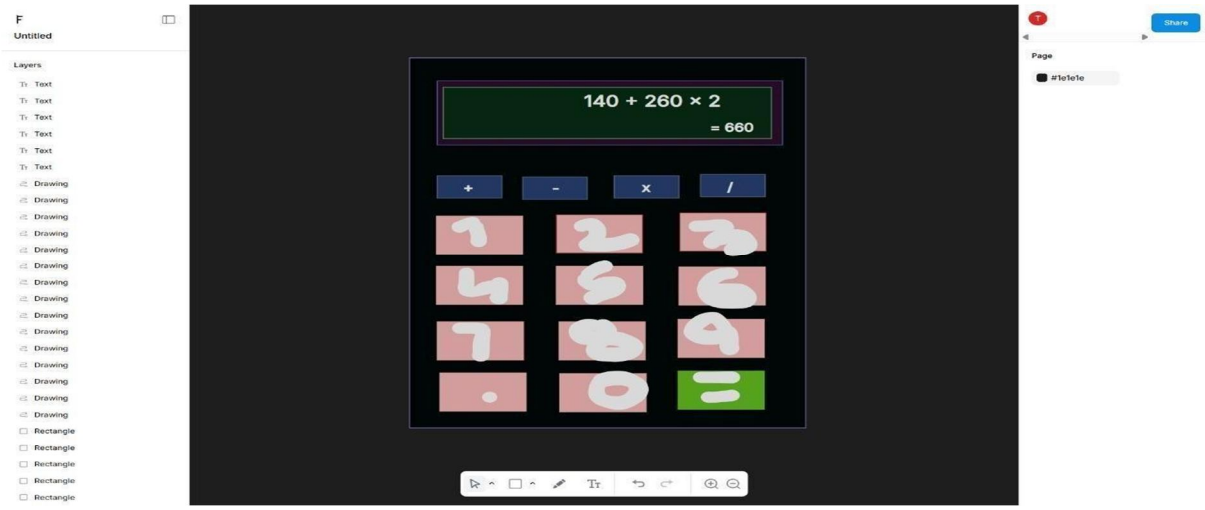


Fig. 6 Designing

Fig. 7 Main Landing Page

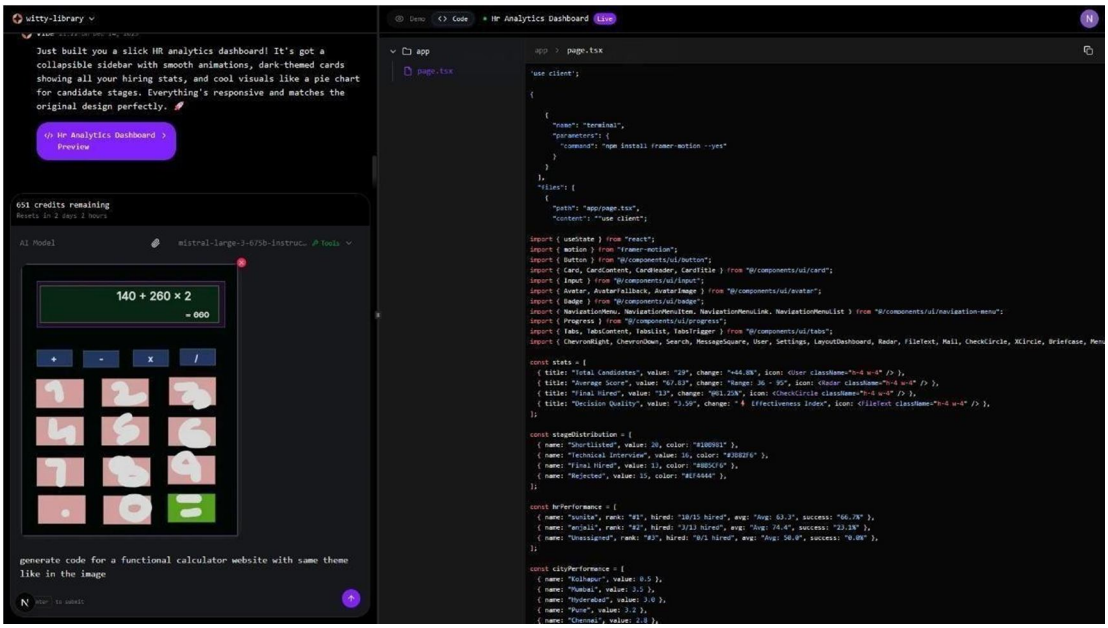


Fig. 8 Code and Preview

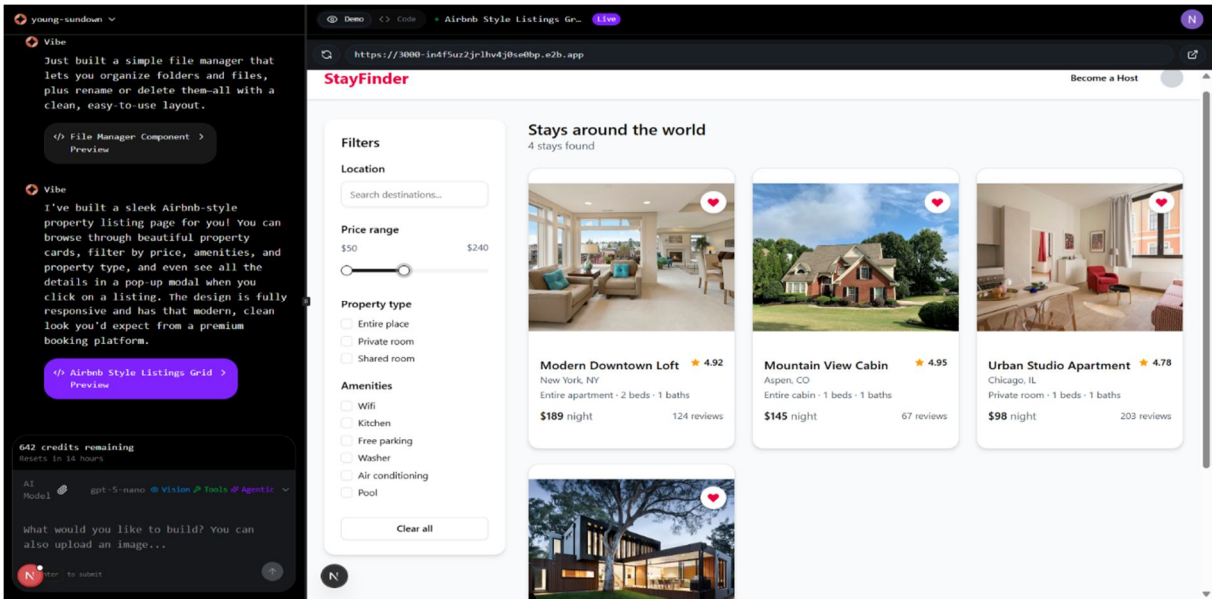


Fig. 9 Realtime Live Preview Page

VI. CONCLUSION

This paper presented a Collaborative Web Designing Platform using Liveblocks that integrates real-time multi-user design with AI-assisted code generation. By eliminating the traditional separation between design and development, the platform improves productivity, collaboration, and development speed. The system demonstrates strong potential as a scalable solution for modern web development workflows.

VII. FUTURE WORK

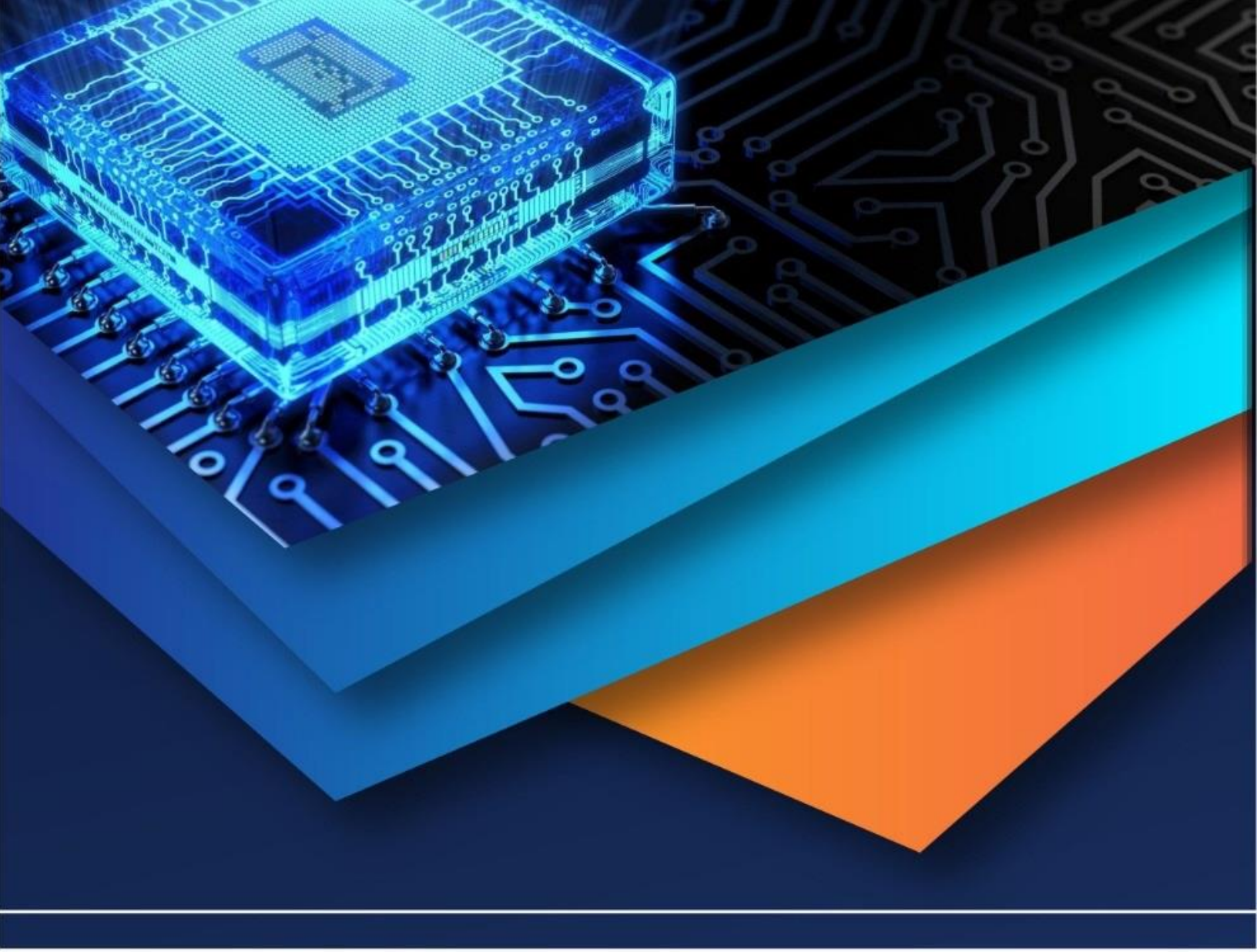
Future enhancements include extending backend code generation for full-stack applications, integrating advanced AI models for improved UI understanding, and supporting additional frontend frameworks and deployment pipelines.

VIII. ACKNOLWEDGEMENT

The authors express their sincere gratitude to Mrs. Shylaja L N, Associate Professor & Head, Department of Artificial Intelligence and Machine Learning, Bahubali College of Engineering, for her valuable guidance and continuous support throughout this work. The authors also thank the faculty of the Department of Information Science and Engineering for their encouragement and academic support. Finally, heartfelt thanks are extended to their parents and peers for their motivation an d cooperation during the completion of this research.

REFERENCES

[1] M. B. Khan, C. S. Kushwaha, R. Rani, A. Verma, and P. Bahad, “Designing and Development of Real-Time Code Editor for Collaborative
[2] Programming,” International Journal of Computer Applications, 2023.
[3] Y. Dong, X. Jiang, Z. Jin, G. Li, and S. N., “Self-Collaboration Code Generation via ChatGPT,” arXiv Preprint, 2024.
[4] H. Zhang and Y. Su, “WebCode2M: A Real-World Dataset for Code Generation from Webpage Designs,” Proceedings of the International Conference on Software Engineering, 2025.
[5] S. Xiao, Y. Chen, J. Li, L. Chen, L. Sun, and T. Zhou, “Prototype2Code: End-to-End Front-End Code Generation from UI Design Prototypes,”
[6] Proceedings of the ACM Conference, 2024.
[7] T. Zhou and Y. Zhao, “Bridging Design and Development with Automated Declarative UI Code Generation,” IEEE Transactions on Software Engineering, 2024.
[8] M. Finnie, “Automatic Code Generation from Design Patterns,” IBM Systems Journal, 1996.
[9] P. de Lange and P. Nicolaescu, “Engineering Web Applications Using Real-Time Collaborative Modeling,” International Conference on Web Engineering, 2017.
[10] A. Frick, “From Design to Code: A Study on Generating Production Code from User Interface Design Software,” Master’s thesis, 2020.
[11] B. Combemale, J. Gray, and B. Rumpe, “Model-Based Code Generation Works: But How Far Does It Go? On the Role of the Generator,” IEEE Software, 2024.
[12] A. Neumann, “Integrating Web-Based Collaborative Live Editing into Model-Driven Engineering,” Journal of Web Engineering, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY