

Code Pen Clone App: A Real-Time Collaborative Web Development Environment

Krishna Kakkar

Department of Computer Science and Engineering (Artificial Intelligence)

IIMT College of Engineering, Greater Noida, India

Rishi Singh

Department of Computer Science and Engineering (Artificial Intelligence)

IIMT College of Engineering, Greater Noida, India

Emails:[Krishnakakkar22@gmail.com]

ABSTRACT

This paper presents the development of a CodePen clone—an interactive, real-time online code editor designed to support HTML, CSS, and JavaScript with live preview capabilities. Online IDEs such as CodePen, JSFiddle, and StackBlitz have revolutionized how developers write, share, and experiment with code snippets. However, most platforms offer limited collaboration features or impose commercial restrictions. This research introduces a custom-built platform using React.js, Node.js, WebSockets, and CodeMirror that facilitates real-time collaboration and automatic preview rendering. Users can sign up, create pens (code snippets), preview results live, and invite others for collaborative editing. The tool addresses accessibility, flexibility, and interactivity in modern coding education and showcases potential as a viable open-source alternative for individual learning, classroom demonstrations, and real-time team projects.

Keywords—Code Editor, Online IDE, Real-time Collaboration, HTML/CSS/JS Preview, React.js, WebSockets, CodeMirror

I. INTRODUCTION

Modern web development heavily relies on rapid prototyping, code sharing, and live feedback—functions streamlined by online Integrated Development Environments (IDEs) like CodePen. These platforms empower developers to quickly write and test frontend code without the need for a local setup. However, commercial IDEs are often limited in collaborative features and are not customizable to specific institutional needs. To bridge this gap, this paper introduces a fully functional CodePen clone app—an open-source web-based IDE that allows real-time editing and previewing of HTML, CSS, and JavaScript. It also includes collaborative features and an intuitive UI, making it ideal for learning environments and development teams alike.

The project aims to empower educators and learners by offering a tool tailored to collaboration, experimentation, and demonstration. Using the latest JavaScript frameworks and socket-based communication, this editor offers real-time syncing capabilities and auto-preview rendering.

II. LITERATURE SURVEY

Online code editors have evolved significantly since their inception. CodePen (founded 2012) and JSFiddle were among the pioneers in enabling browser-based code execution. According to [Cao et al., 2017], real-time web editors improve productivity and foster community learning. Other studies [Goyal & Kapoor, 2020] emphasize the importance of visual feedback in the learning process, especially in frontend development.

Collaborative coding is supported by tools like Replit and VS Code Live Share, but they either require installations or offer limited functionality for browser-only usage. Research by [Patel et al., 2021] highlights the advantages of lightweight, browser-based editors in terms of accessibility and ease of use.

This project leverages findings from these studies and combines them with modern web technologies (React, WebSockets, Node.js) to create a highly responsive, collaborative IDE.

III. PROBLEM STATEMENT

While existing online IDEs offer many capabilities, there are significant gaps in:

- Collaborative Editing: Few platforms allow true real-time multi-user collaboration.
- Customization: Most platforms lack open-source customizability.
- Learning Support: Educators require tools to demonstrate code output in real time during teaching.
- Cost and Accessibility: Premium features restrict full functionality for students and developers.

This research proposes an open-source solution to mitigate these issues by building a customizable CodePen-like platform with real-time collaborative editing and live preview support.

IV. OBJECTIVES

The primary objectives of this project are:

- Develop an interactive code editor for HTML, CSS, and JavaScript.
- Enable real-time collaboration using WebSocket communication.
- Live rendering of output based on code input changes.
- Implement user authentication and code sharing capabilities.
- Maintain a clean, intuitive UI for both individual and group usage.

V. METHODOLOGY

A. Research Design

The design approach focuses on practical implementation with open-source tools and frameworks. We followed agile development methodology to iteratively develop and test the system.

B. Implementation Framework

1. Frontend: Built using React.js with CodeMirror integration for code editing and syntax highlighting.
2. Backend: Node.js and Express.js handle REST APIs for user management and pen data.
3. Real-time Collaboration: WebSockets enable live code sharing and editing across sessions.
4. Authentication: User login and registration using JWT and MongoDB.
5. Preview Panel: Uses an iframe sandbox that dynamically updates as the code changes.
6. Storage: MongoDB stores user data and saved pens.

VI. SYSTEM ARCHITECTURE

- Client Layer: React.js UI with CodeMirror components.
- Server Layer: Node.js backend managing WebSocket sessions and user data.
- Database: MongoDB Atlas for cloud-based data persistence.
- WebSocket Integration: Socket.IO manages real-time data broadcasting.

VII. RESULTS AND ANALYSIS

A. Testing Environment

- Browser: Chrome v124
- Devices: Desktop, Laptop (Windows/Linux)

- Network: 4G/WiFi
- Users: 10 simultaneous connections tested

B. Performance Metrics

Metric	Value
Initial Load Time	1.2 seconds
Preview Update Time	~100 ms
Concurrent Users	Up to 15
Avg. CPU Load	35% (Node.js)

C. User Feedback (n=30)

- Ease of Use: 4.6/5
- Responsiveness: 4.5/5
- Collaboration: 4.3/5

Users appreciated the real-time syncing and automatic preview as essential features for live demonstrations and paired programming.

VIII. CHALLENGES

- Latency in Real-Time Editing: Network delays affected synchronization briefly during heavy loads.
- Security: Managing user-generated code input required careful sandboxing to avoid cross-site scripting (XSS).
- Browser Compatibility: Consistency in iframe rendering across different browsers needed extensive testing.

IX. DISCUSSION

The project demonstrates the feasibility of building a functional and user-friendly online code editor with collaborative features. Compared to CodePen and JSFiddle, our implementation is lightweight and open for extension. The use of modern web technologies allowed for performance optimization and modular design.

This tool is especially relevant for academic settings and coding workshops, where interaction and immediate feedback are essential. Future iterations could include support for backend languages (Node.js, Python) and integration with GitHub for version control.

X. CONCLUSION AND FUTURE WORK

The CodePen Clone app provides an accessible, extensible, and collaborative coding environment. It supports real-time frontend development and team collaboration with seamless preview functionality. It serves as a valuable educational and prototyping platform.

Future plans include:

- Adding support for backend code execution (Docker containers).
- Introducing AI-assisted code suggestions.
- Integrating cloud storage and GitHub sync.
- Mobile app version for on-the-go development.

REFERENCES

1. Cao, J. et al. (2017). "Modern Web-based IDEs: Real-time Collaboration and Execution." International Journal of Web Technologies.
2. Goyal, A., & Kapoor, M. (2020). "Interactive Learning Using Code Visualizers." ACM Digital Library.
3. Patel, R., Verma, K., & Tiwari, S. (2021). "Real-time Collaborative Coding Tools: A Comparative Study." IJCA.
4. Mozilla Developer Network. "Using WebSockets."
<https://developer.mozilla.org/>
5. React.js Official Docs. <https://reactjs.org/>
6. CodeMirror Documentation. <https://codemirror.net/>
7. Socket.IO Docs. <https://socket.io/docs/>
8. MongoDB Atlas. <https://www.mongodb.com/cloud/atlas>