International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

REAL TIME COLLABORATIVE CODE EDITOR

Ms. P.Anita M.E
Assistant Professor, Department of CSE
Coimbatore Institute of Technology
Coimbatore – 641 014

Sakithyan A	Dilip S	Nikhil R	Rithish S	Maneesh Aravind S
UG Student	UG Student	UG Student	UG Student	UG Student

Department of CSE, Coimbatore Institute of Technology

Coimbatore – 641 014

ABSTRACT

This project presents a real-time collaborative code editor, uniquely enhanced with a voice chat feature to facilitate seamless communication among team members. Built using Express.js for the backend and Next.js for the frontend, and utilizing a Firebase database for efficient data management, the system offers a user-friendly and interactive interface.

Key functionalities include a login page where users can sign up with their name, email ID, and password, and an interactive dashboard for managing projects. Projects can be categorized as public or private, with public projects accessible to anyone and private projects restricted to the owner. Users can invite teammates to collaborate on projects via unique project-specific links. The system supports up to 10 participants in a single room, allowing multiple users to work together effectively.

The collaborative environment is powered by WebSockets for real-time communication and incorporates the Gemini API for additional team interaction and query resolution. WebRTC is utilized for the voice chat feature, enabling users to have live audio discussions while working on code. Additionally, users can compile the code directly within the editor, supporting languages such as C, JavaScript, Python, and Go.

Overall, this project aims to enhance productivity and teamwork in coding projects by integrating robust real-time collaboration tools and features, providing a comprehensive and interactive platform for developers.

KEYWORDS: Collaborative Realtime Code Editor, Next.js, Node.js, Socket.io, Express.js, Web Application.

INTRODUCTION

Effective collaboration is vital in modern software development, especially in remote work environments where traditional tools often lack real-time interactivity and communication features. Real-time collaborative code editors address these challenges by allowing multiple users to write and edit code simultaneously, providing immediate feedback and reducing code conflicts.

These platforms leverage technologies like WebSockets for persistent, bi-directional communication and WebRTC for peer-to-peer audio interactions, facilitating seamless teamwork.

This project develops a real-time collaborative code editor with integrated voice chat, using Express.js and Next.js for the application framework and Firebase for data management. The goal is to enhance productivity and teamwork by providing a robust, scalable solution that meets the needs of modern software development teams.

PROBLEM STATEMENT

In the rapidly evolving field of software development, effective collaboration among team members is critical for the successful completion of projects. Traditional code editors and version control systems often lack the real-time interactivity and communication features necessary for dynamic teamwork. This limitation results in inefficiencies and miscommunications, particularly when team members are working remotely.

To address these challenges, there is a need for a real-time collaborative code editor that not only supports simultaneous coding by multiple users but also includes integrated communication tools. Specifically, the absence of seamless voice communication within existing collaborative coding platforms hampers the ability of team members to discuss and resolve issues quickly. Therefore, the problem is to develop a comprehensive platform that facilitates real-time collaboration, provides robust project management features, and integrates effective communication tools, including voice chat, to enhance the overall efficiency and productivity of development teams.

LITERATURE SURVEY

The literature survey explores current advancements in real-time collaborative code editors and integrated communication tools. It highlights the importance of synchronization in collaborative programming environments to prevent code duplication and enhance efficiency. Existing platforms like Google Docs and EtherPad demonstrate the effectiveness of real-time text editing for collaboration, which is crucial in programming contexts.

Integrating voice chat using WebRTC enhances collaborative coding experiences by allowing real-time audio communication without additional software installations. This capability supports peer-to-peer communication with low latency and high-quality audio, facilitating effective team coordination and decision-making during code development.

Firebase's Realtime Database provides synchronized, cloud-hosted JSON storage suitable for real-time collaborative applications. It ensures data consistency across multiple clients and supports offline capabilities, enhancing reliability and continuous collaboration. Firebase also offers robust security features through authentication and access controls, crucial for protecting sensitive data in collaborative environments.

WebRTC's signaling mechanism, facilitated by protocols like JSEP and STUN/TURN servers, enables secure peer-to-peer connections for real-time media exchange. It employs encryption protocols like DTLS and SRTP to ensure data confidentiality and integrity, crucial for maintaining secure communications in collaborative coding platforms.

ARCHITECTURE

The **FRONTEND** encompasses several key user interfaces. The Register interface allows new users to create an account, while the Login interface is for returning users. After logging in, users are taken to the Dashboard, the main interface where they can access various features. Within the Dashboard, users can navigate to User Settings, where they can manage their personal settings, and Add File, where they can add new files by specifying the file name, access type, and language. The Code Editor is another crucial interface, providing a space for users to write and edit code.

On the **BACKEND**, the application handles essential logic, computation, and database interactions. When a file is added, the backend determines whether the file is private. If the file is not private, the backend creates a socket connection based on a unique URL, enabling real-time collaboration. This backend functionality allows multiple users to join a room via the unique URL and engage in audio chat using WebRTC. For code compilation, the backend integrates with glot.io, while additional API integrations are managed

through the Gemini API. The application also utilizes the Firebase Real-time Database to store user files, settings, and other dynamic data.

The **LOGIN** process is straightforward. When a user attempts to log in, the system first checks if they are a new user. New users are directed to the Register page, while returning users proceed to the Login page. Upon successful authentication, users are taken to the Dashboard, where they can access various functionalities. From the Dashboard, users can manage their settings, view files, and access the Trash of Deleted Files, which temporarily stores deleted files for potential recovery.

A notable feature is the **ROOM** join concept. When a user adds a file and it is marked as not private, the system creates a unique URL for the file, facilitating real-time collaboration. Multiple users can join the room using this URL, and they can communicate through audio chat enabled by WebRTC. This feature enhances the collaborative aspect of the application, making it easier for users to work together seamlessly.

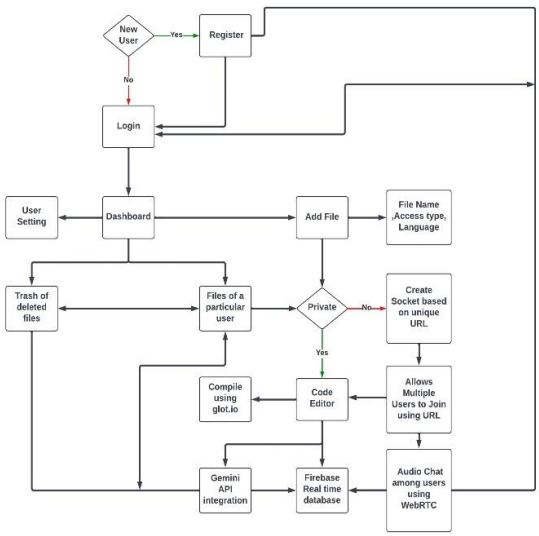

Figure 1 – OVERALL ARCHITECTURE

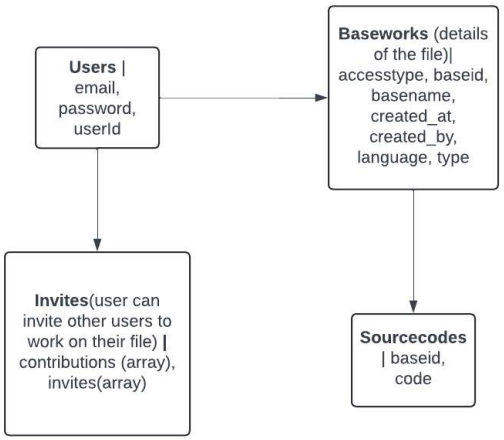

Figure 2 – DATABASE FLOWCHAT

COMPONENTS USED

The collaborative Realtime Code Editor web application is built using a combination of technologies that work together seamlessly to provide an efficient and user-friendly platform for developers to collaborate on code. The user interface is built using Next.js, which provides a scalable and modular framework for building interactive components such as the code editor, chat box, and file explorer.

Node.js is used for the server-side implementation, providing an efficient platform for executing server-side code and handling user authentication, file management, and real-time communication through Socket.io.

Node.js

Node.js is a scalable network application builder that uses an asynchronous event-driven JavaScript engine. Most connections can be managed simultaneously in the following "hello world" example. The call back is invoked with each connection, but if there is no work to be done, Node.js will sleep

Next.js

This system offers several powerful features for streamlined development: Hot Code Reloading automatically updates the application upon code changes, ensuring seamless development iterations. Automatic Code Splitting optimizes page loading by bundling only necessary imports, enhancing performance. It boasts ecosystem compatibility with JavaScript, Node.js, and React, ensuring flexibility across environments. Additionally, Server Rendering efficiently renders React components on the server, enhancing user experience. Styled-JSX simplifies styling by enabling CSS to be directly written within JavaScript code, streamlining the development process.

Express.js

Express.js is a fast, flexible and minimalist web framework for Node.js. It's effectively a tool that simplifies building web applications and APIs using JavaScript on the server side. Express is an open-source that is developed and maintained by the Node.js foundation.

Express.js offers a robust set of features that enhance your productivity and streamline your web application. It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node HTTP objects and facilitates the rendering of dynamic HTTP objects.

Socket.io

Socket.io is a widely used library for Node.js that enables real-time communication between a client and server via WebSocket connections. It simplifies handling real-time events and data exchange between a client and server by providing an abstraction layer over the WebSocket API.

WebRTC(WebReal-Time Communications)

WebRTC(WebReal-TimeCommunications) is an open source project that enables real-time voice, text and video communications capabilities between web browsers and devices. WebRTC provides software developers with application programming interfaces (APIs) written in JavaScript.

Developers use these APIs to create peer-to-peer (P2P) communications between internet web browsers and mobile applications without worrying about compatibility and support for audio-, video- or text-based content.

With WebRTC, data transfer occurs in real time without the need for custom interfaces, extra plugins or special software for browser integration. WebRTC enables real-time audio and video communication simply by opening a webpage.

WebSocket Protocol

WebSocket is a computer networking protocol that enables full-duplex communication over a single TCP connection. In 2011, the IETF standardized the WebSocket protocol as RFC 6455. WebSocket differs from HTTP, both of which are on the OSI model's layer 7 and rely on TCP on layer 4. According to RFC 6455, WebSocket "is designed to run over HTTP ports 443 and 80 and to support HTTP intermediaries and proxies," thereby making it HTTP compliant..

The WebSocket protocol enables real-time data transfer between the server and a web browser or other client application by allowing interaction with less overhead than half-duplex alternatives like HTTP polling. This is achieved by allowing the server to transmit content to the client without waiting for a request from the client and by enabling messages to be exchanged while the connection is open.The client and server can engage in a continuous two-way dialogue in this way.

RESULTS

The implementation of the real-time collaborative code editor significantly enhanced team productivity and communication. Using Next.js, Node.js, Express.js, and Socket.io provided a seamless, responsive user experience, enabling multiple users to edit code simultaneously. The WebRTC-powered voice chat feature facilitated real-time verbal communication, crucial for quick issue resolution and collaborative decision-making.

User feedback praised the intuitive interface, project management capabilities, and support for multiple

programming languages (C, JavaScript, Python, Go). Performance tests confirmed the system's ability to support up to 10 participants per room with consistent real-time updates and voice communication. Firebase Realtime Database ensured data synchronization and security.

Overall, the project successfully addressed remote collaborative coding challenges, offering a robust platform that enhances teamwork and efficiency in software development.

REFERENCES

[1] Aditya Kurniawan, Christine Soesanto, Joe Erik Carla Wijaya "CodeR: Real-time Code Editor Application for Collaborative Programming", (ICCSCI2020).

[2] Tiwari, A., Gupta, H., Mittal, S., & Rawat, M. (2022). VOICE CHAT WEB APP USING WEBRTC. IJRDO -Journal of Computer Science Engineering, 8(11), 23-28.https://doi.org/10.53555/cse.v8i11.5444

[3]FIREBASE:https://firebase.google.com/docs/database

[4]WebRTC:https://upcommons.upc.edu/bitstream/handle/2117/106694/alex.albas_117680.pdf

[5] Anwar Saif, Saeed Mohammed, Saleh Mohammed, Osama Ali, Majdi Abdu,"C-MPE: A Collaborative Multiprogramming Development Environment for .Net Framework". (SJITN) Vol 8 No.2 (2020)

[6] Rutvik Manish Kumar Patel, "Distributed & Collaborative Software Engineering" (IRJET) Vol: 5 No.9 (2018) [4] Khanh Nguyen Trong, Doanh Nguyen Ngoc, "Towards a Collaborative IDE for Novice Programmers" (IJITEE) , Vol 6 No.5, (2016)

[7] Yang S, Poonawala A, Jiang T and Schneider B. (2023). Can Synchronous Code Editing and Awareness Tools Support Remote Tutoring? Effects on Learning and Teaching. Proceedings of the ACM on Human-Computer Interaction. 7:CSCW2. (1-30). Online publication date: 28-Sep-2023.