Project Report: Coll-IDE - Collaborative IDE for Competitive Programming

Abstract

Our project is a collaborative coding environment designed to meet the unique needs of competitive programmers, hackathon participants, and teams engaged in group coding contests like ICPC. Built with cutting-edge technologies such as React.js, Socket.IO, MongoDB, and Express.js, our project offers a feature-rich platform that facilitates real-time collaboration, code editing, test case management, and direct submissions to online judges.

This project report provides an overview of our project's architecture, features, implementation details, and the technologies used. It highlights the motivation behind the project, the challenges addressed, the novelty and application innovation, as well as the future directions for development.

Through our project, programmers can collaborate seamlessly, brainstorm ideas effectively, and compete with confidence in coding competitions. Whether it's practicing for contests, participating in hackathons, or working on group projects, our project empowers teams to achieve their coding goals efficiently and collaboratively.

Table of Contents

1 Contribution of each team member

1. Garv Sethi :
Enrollment no. - 22115057
e-mail - garv_s@cs.iitr.ac.in
Mobile no. - 9654690583
Contribution -

- Designed the Frontend for the Application
- Implemented the whiteboard functionality and syntax highlighting

2. Mmukul Khedekar :
Enrollment no. - 22114054
e-mail - mmukul_sk@cs.iitr.ac.in
Mobile no. - 8657526108
Contribution -

- Designed the Frontend for the Application
- Designed the login and signup page

3. Granth Gaud :
Enrollment no. - 22114035
e-mail - granth_g@cs.iitr.ac.in
Mobile no. - 7506955864
Contribution -

- Password Hashing and Authentication Using MongoDB
- File Storage and Retrieval Using MongoDB

4. Swapnil Garg :
Enrollment no. - 22115150
e-mail - swapnil_g@cs.iitr.ac.in
Mobile no. - 7060220490
Contribution -

- Web-scraping for Test Case fetching using Axios
- Web-automation for Direct submission using Puppeteers

5. Sukhman Singh :
Enrollment no. - 22114097
e-mail - sukhman_s@cs.iitr.ac.in
Mobile no. - 9415860440
Contribution

- Implemented the Backend of the project
- Sockets and code execution

2 Introduction

- **Background:**

  The genesis of our collaborative coding project is deeply rooted in the evolving programming landscape and the burgeoning demand for tools tailored to facilitate collaborative coding efforts. With the surge in popularity of competitive programming and hackathons over the years, individuals and teams worldwide have increasingly participated in coding competitions. However, existing tools often fall short in fostering effective collaboration and streamlining the development process.

  Traditional Integrated Development Environments (IDEs) typically cater to individual coding workflows, lacking features that facilitate real-time collaboration or integration with online coding platforms like Codeforces. Furthermore, the absence of built-in tools for brainstorming and idea sharing further complicates the collaborative process, especially in group contests such as the International Collegiate Programming Contest (ICPC).

  Recognizing these challenges, our collaborative coding project emerged to address the need for a comprehensive development environment specifically tailored for competitive programming and hackathons. By amalgamating features such as real-time collaboration, seamless integration with Codeforces for test case management and submission, and a built-in whiteboard for idea generation and discussion, our project aims to bridge the gap between traditional IDEs and the demands of modern programming competitions.

  Our project draws inspiration from the increasing emphasis on teamwork and problem-solving skills in the programming community, as well as the desire to provide programmers with the necessary tools to thrive in competitive environments. Through continuous development and iteration, our project strives to become the go-to platform for teams seeking to enhance their coding capabilities, collaborate more effectively, and achieve success in competitive programming and hackathon challenges.

- **Motivation of Project:** The motivation behind our collaborative coding project stemmed from the need to enhance the collaborative aspect of programming, particularly in competitive programming and hackathons. Traditional IDEs often lack robust collaboration features, making it challenging for teams to work seamlessly together, especially in high-pressure environments like coding contests or hackathons. By developing our project, the aim was to address these limitations and provide a platform that facilitates real-time collaboration among programmers, enables efficient sharing of ideas, and streamlines the process of accessing and submitting test cases, all within a single environment. This not only enhances the overall experience for participants but also fosters a more productive and cohesive working environment, ultimately improving outcomes in competitive programming and hackathon scenarios.

- **Application Domain:** The application domain of our collaborative coding project encompasses various aspects of competitive programming, hackathons, and collaborative coding environments. Specifically, it targets:

1. **Competitive Programming**: Our project is tailored to meet the needs of competitive programmers who participate in coding competitions on platforms like Codeforces. It provides features such as fetching test cases directly from Codeforces, enabling efficient coding, testing, and submission processes within the same interface.
2. **Hackathons**: The platform is designed to support hackathon participants by facilitating collaborative coding sessions. Teams can work together in real-time, brainstorm ideas using the built-in whiteboard feature, and streamline the development process during intense hackathon events.
3. **Group Contests like ICPC**: Our project caters to group contests such as the International Collegiate Programming Contest (ICPC), where teams of programmers collaborate to solve complex problems under time constraints. The platform allows multiple users to work together seamlessly, enhancing teamwork and productivity during group coding sessions.
4. **Collaborative Coding**: Beyond specific events or competitions, our project serves as a general-purpose environment for collaborative coding endeavors. It enables programmers to work together on projects, share code snippets, discuss ideas, and iterate on solutions in real-time, fostering a collaborative coding culture.

3 Proposed System

- **Overview:**

  1. **Code Editor**
     - The system provides a text area for writing and editing C++ Code.
     - Syntax highlighting and font size zoom are implemented.
     - Code auto-completion and suggestions based on previous written code are provided.
  2. **Input/Output Area**
     - An input area is included for users to provide test case inputs.
     - An output area displays the results of the code execution.
  3. **Collaboration**
     - Two sharing links are generated: one for read-only access and one for editor access.
     - Other users who are logged in can use the respective links to join the work.
  4. **Whiteboard/Scribble Area**
     - A collapsible whiteboard area is present with basic pen and eraser drawing tools.
     - It is compatible with both mouse and stylus input and saves the state along with the particular file.
  5. **User Authentication and File Management**
     - Users can create accounts and log in securely.
     - The system supports file saving, loading old files, and a simple file explorer for each user.
  6. **Direct Submission to Online Judges**
     - Users have a submit button where they can enter the link to the problem they want to submit their code to.
     - Once the link is entered, users do not have to provide the link again upon clicking the button.

- **Design Diagram:**

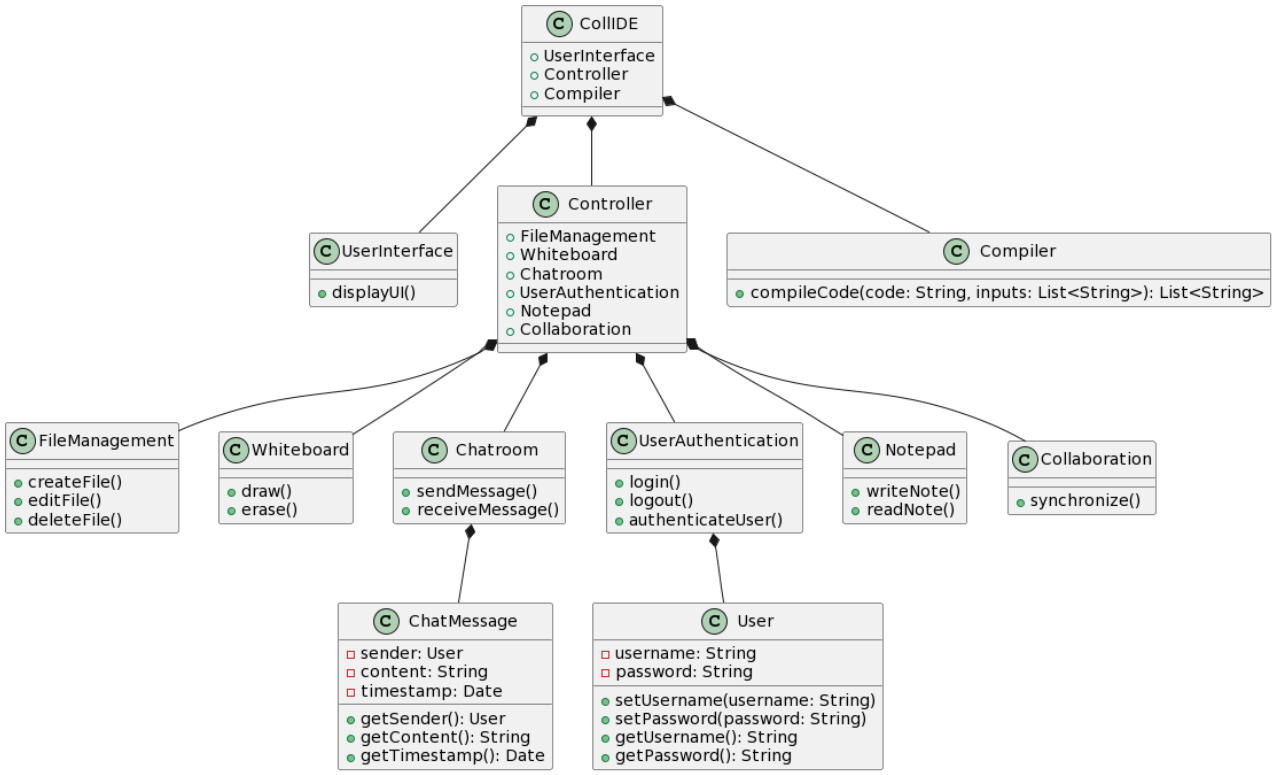

Figure 1: Class Diagram

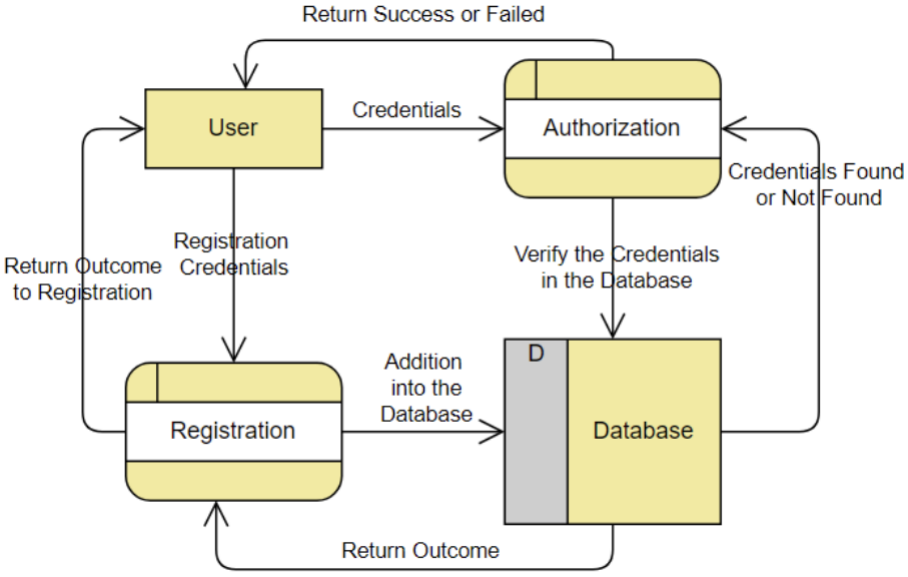

Figure 2: User authentication and registration DFD

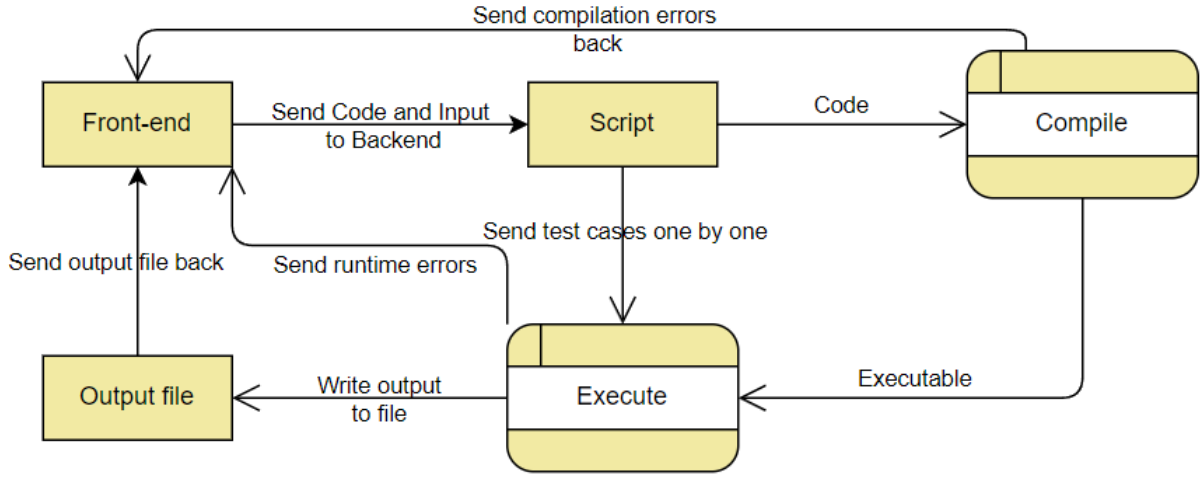

Figure 3: DFD for code execution

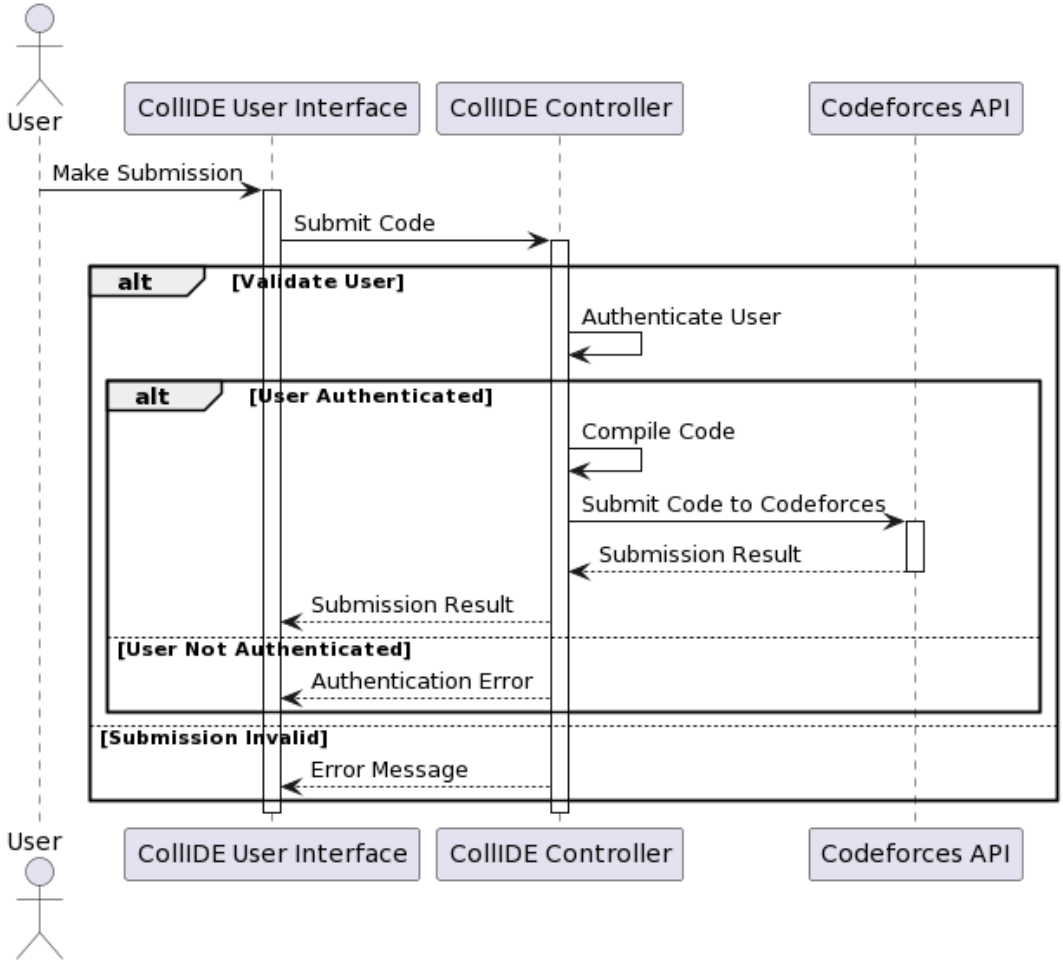

Figure 4: Sequence Diagram for direct submission

- **Algorithms:** In our implementation of Coll-IDE, we have utilized a variety of algorithms to ensure the security, efficiency, and reliability of the system:

    1. **Hashing Algorithm**: We have employed a robust hashing algorithm, such as bcrypt, for securely hashing and storing user passwords in the database, ensuring the confidentiality of user credentials.
    2. **Authentication Algorithms**:
        - For authentication, we have implemented token-based authentication using JWT (JSON Web Tokens) or session-based authentication. This allows us to securely authenticate users and manage their sessions throughout their interaction with the system.
    3. **User Management Algorithms**:
        - Our system incorporates algorithms for user registration, including validation of user input, checking for uniqueness of usernames or email addresses, and enforcing password strength requirements to ensure the integrity of user data.
    4. **Socket Communication Algorithms**:
        - We have implemented robust socket communication algorithms, utilizing either TCP/IP or UDP protocols depending on the requirements of the system. These algorithms facilitate efficient and reliable communication between the client and server components of Coll-IDE.
    5. **Database Management Algorithms**:
        - Our system incorporates algorithms for performing CRUD (Create, Read, Update, Delete) operations on user data stored in the database. These algorithms ensure efficient management and retrieval of user information while maintaining data consistency and integrity.
    6. **Concurrency and Multithreading Algorithms**:
        - To handle multiple client connections concurrently, we have implemented thread pooling algorithms, managing a pool of worker threads to efficiently handle client requests. Additionally, we employ concurrency control algorithms to prevent data corruption or inconsistencies in a multi-user environment.

- **Novelty/Application Innovation:**

    1. **Real-time Collaboration**:
        - Coll-IDE introduces real-time collaboration features specifically tailored for competitive programming and hackathons. Unlike traditional IDEs, Coll-IDE allows multiple users to work simultaneously on the same codebase, enhancing teamwork and productivity.
    2. **Integration with Online Judges**:

- The direct submission feature to online judges like Codeforces streamlines the process of testing and submitting code for competitive programming contests, saving time and effort for users.

3. **Whiteboard Functionality**:
   - The built-in whiteboard/scribble area provides a platform for brainstorming and idea sharing, fostering creativity and collaboration among users.
4. **User Authentication and File Management**:
   - Coll-IDE offers secure user authentication and robust file management capabilities, ensuring data privacy and seamless user experience.
5. **Tailored for Competitive Programming and Hackathons**:
   - Coll-IDE is specifically designed to meet the needs of competitive programmers and hackathon participants. By providing features such as real-time collaboration, direct submission to online judges, and a built-in whiteboard, Coll-IDE streamlines the coding process for these high-pressure environments.
6. **Enhanced Collaboration**:
   - The collaborative features of Coll-IDE enable teams to work together efficiently, whether they are participating in coding contests or collaborating on projects. By facilitating communication and idea sharing, Coll-IDE fosters a collaborative coding culture that leads to better outcomes.
7. **Versatile Use Cases**:
   - While Coll-IDE is primarily targeted towards competitive programming and hackathons, its features make it applicable in various other scenarios as well. From group contests like ICPC to general collaborative coding endeavors, Coll-IDE offers a versatile platform for programmers to collaborate and innovate.
8. **Seamless Integration**:
   - Coll-IDE seamlessly integrates with existing online coding platforms like Codeforces, enhancing the user experience and providing a familiar environment for users.

4 Implementation

- **Implementation Details/Tools and Technology Used:**

  1. **Frontend Framework**:
     - React.js: Used for building the frontend user interface of Coll-IDE, providing a responsive and interactive coding environment.
  2. **Backend Framework**:
     - Express.js: Used to develop the backend server of Coll-IDE, handling HTTP requests, managing user sessions, and interacting with the database.
  3. **Websockets**:
     - Socket.IO: Utilized for real-time communication between the client and server, enabling collaborative editing features in Coll-IDE.
  4. **Database**:
     - MongoDB: Used as the database management system for storing user accounts, files, session data, and other relevant information.
  5. **Web Scraping and Automation**:
     - Cheerio: Employed for parsing and manipulating HTML data retrieved from web pages, particularly for extracting test cases from Codeforces.
     - Axios: Used for making HTTP requests to Codeforces API endpoints to fetch test cases and handle direct submissions.
     - Puppeteer: Utilized for automating web browser interactions, such as submitting code to Codeforces and simulating user actions during web scraping tasks.

  These tools and technologies work together to provide a comprehensive and efficient coding environment in Coll-IDE, offering features like collaborative editing, test case fetching, direct submissions, and more.

5 System Screenshots

- **Real-time Collaboration:**

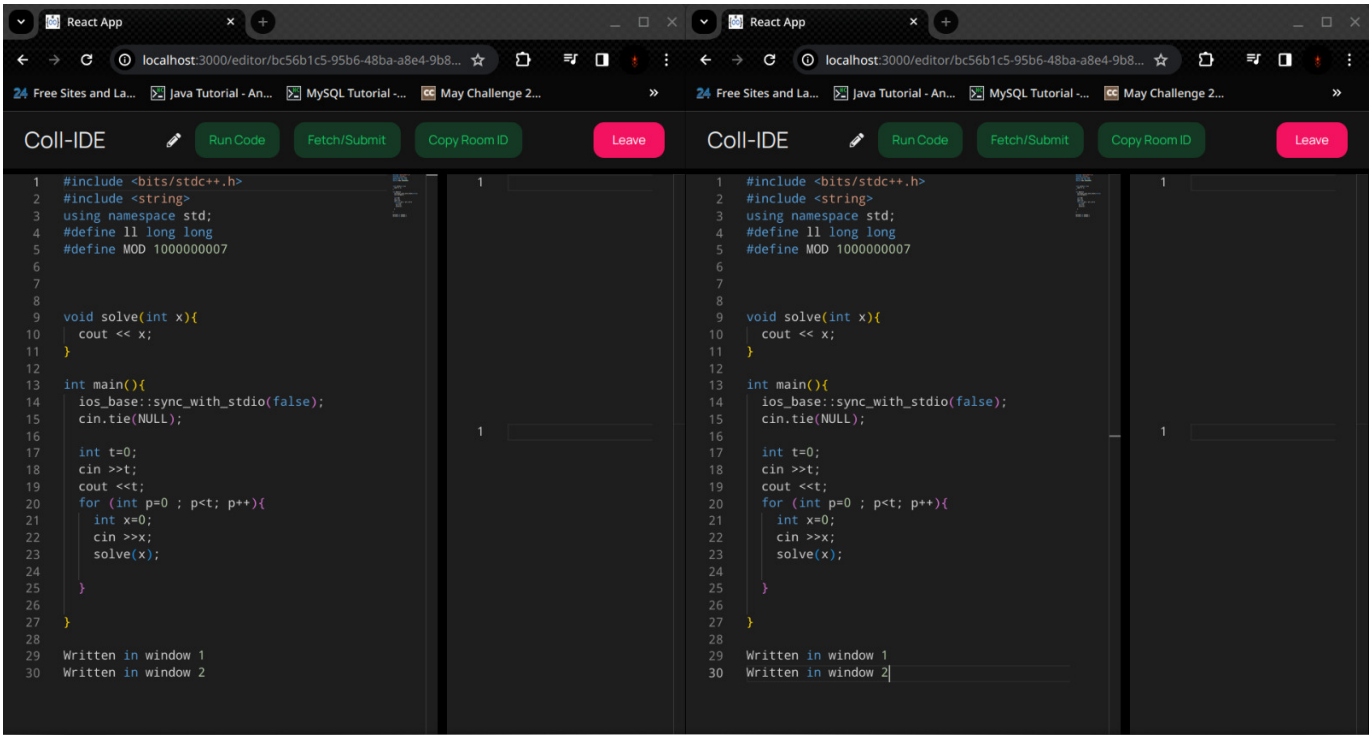

Figure 5: Collaborative Coding

- **Whiteboard:**

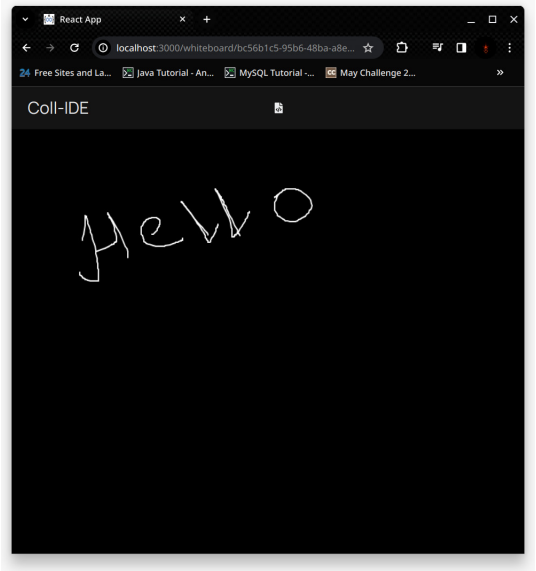

Figure 6: Whiteboard Functionality

- **Code Editor:**

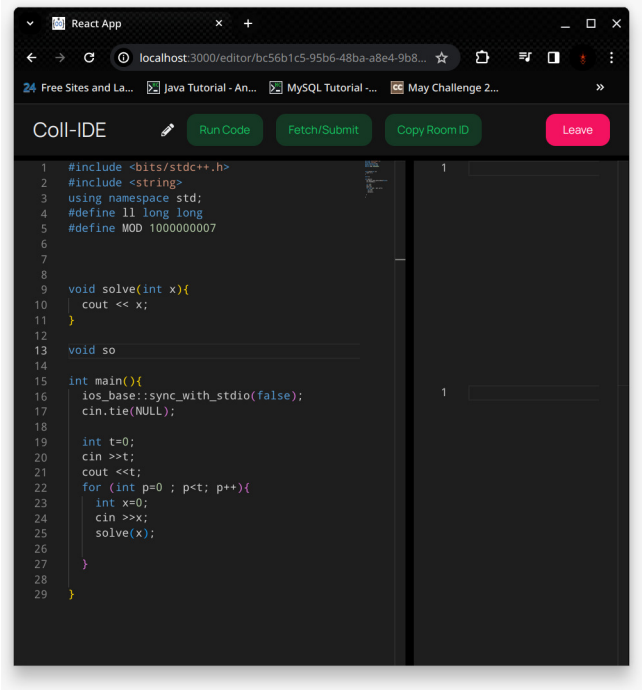

Figure 7: Code Editor

- **Input and Output Tabs:**

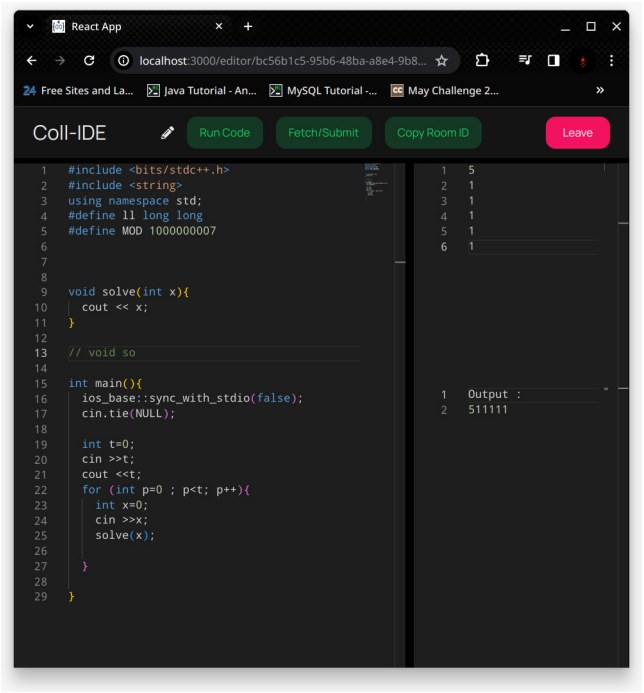

Figure 8: I/O Tabs

- **Codeforces Integration:**

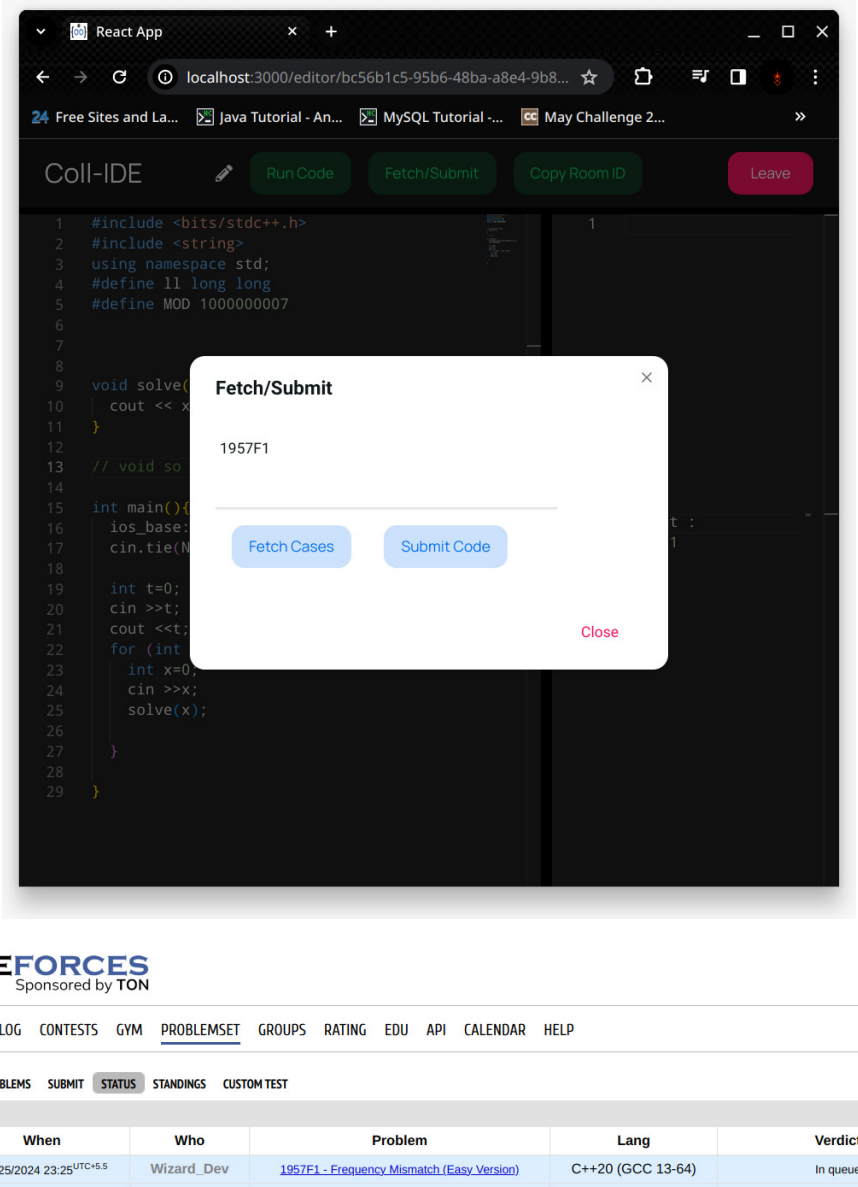

Figure 9: Fetch test cases and submit directly to codeforces

- **Multi-user Rooms:**

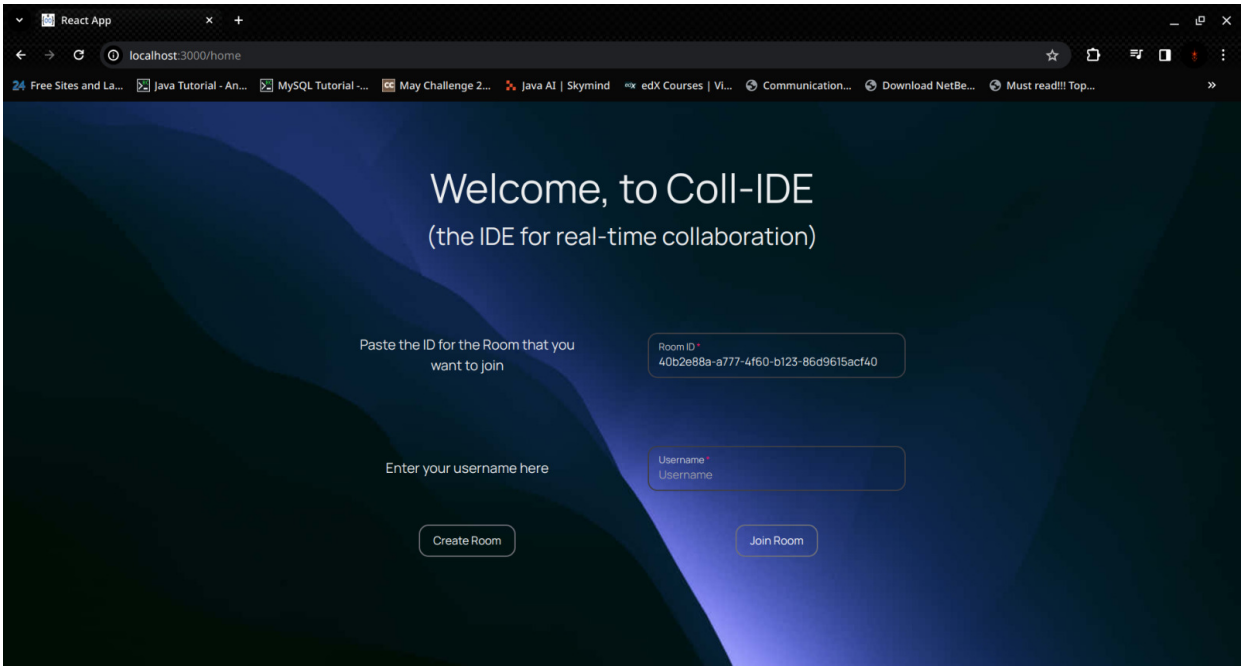

Figure 10: Users joining different rooms with room Ids

- **Signup and Login**

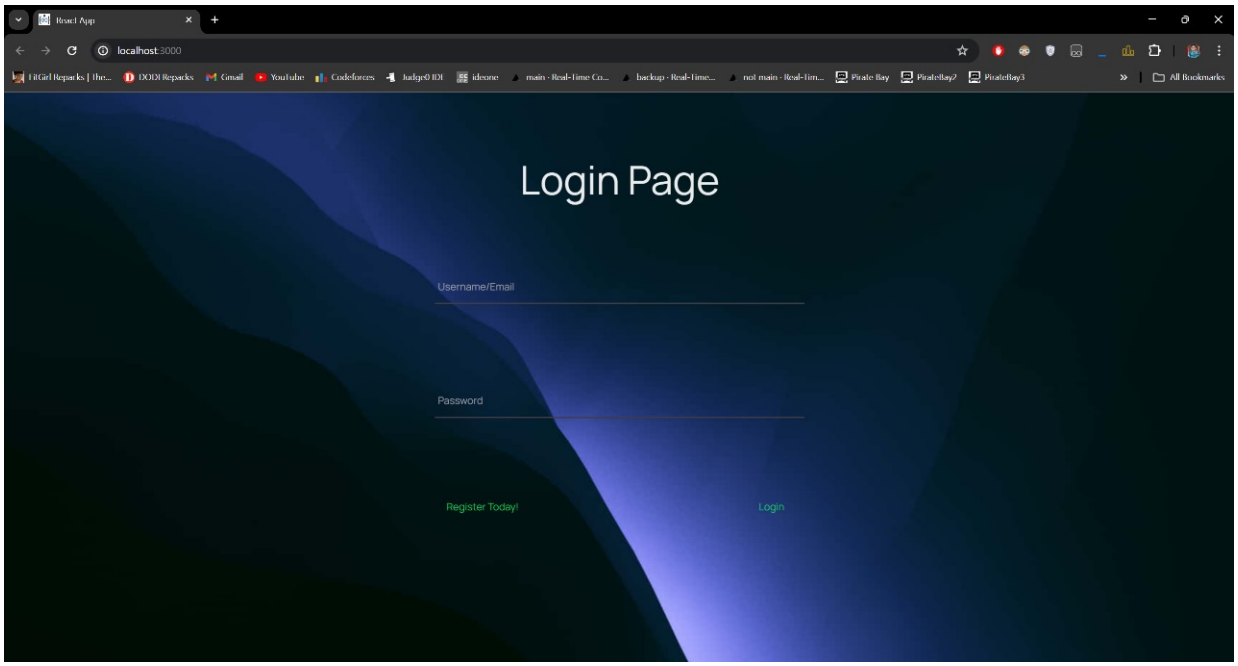

Figure 11: Sign Up and Login facility

6 Comparison with existing similar systems

For the purpose of our comparison we are choosing famous online editors called Replit and Ideone.

Table 1: Comparison with popular Online IDEs

Feature	Coll-IDE	Replit	IdeOne
Collaborative Editing	Yes	Yes	No
User Authentication	Yes	Yes	Yes
Fetching Test Cases from Codeforces	Yes	No	No
Direct Submission to Codeforces	Yes	No	No
Support for multiple languages	No	Yes	Yes
Support for Rooms	Yes	Yes	No
Support for Chatrooms	In Development	Yes	No
Support for Whiteboards	Yes	No	No
Separate I/O Sections	Yes	No	Yes

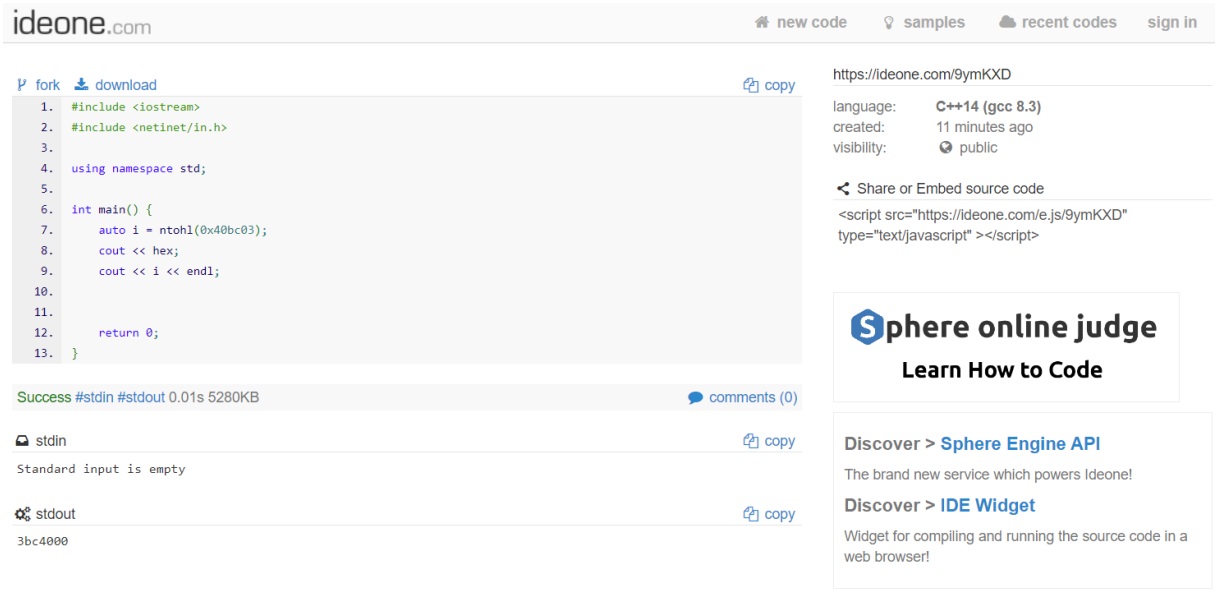

Figure 12: Ideone Interface

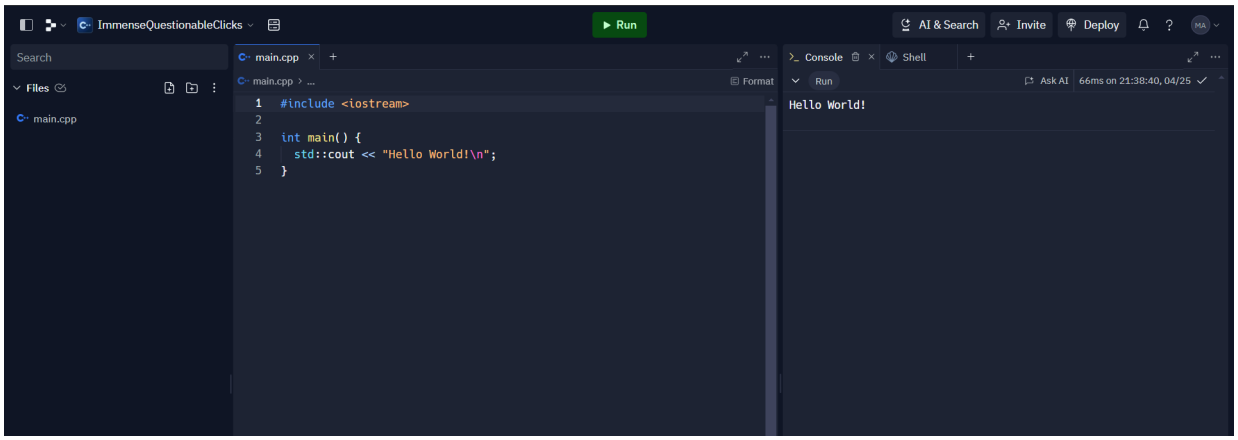

Figure 13: Replit Interface

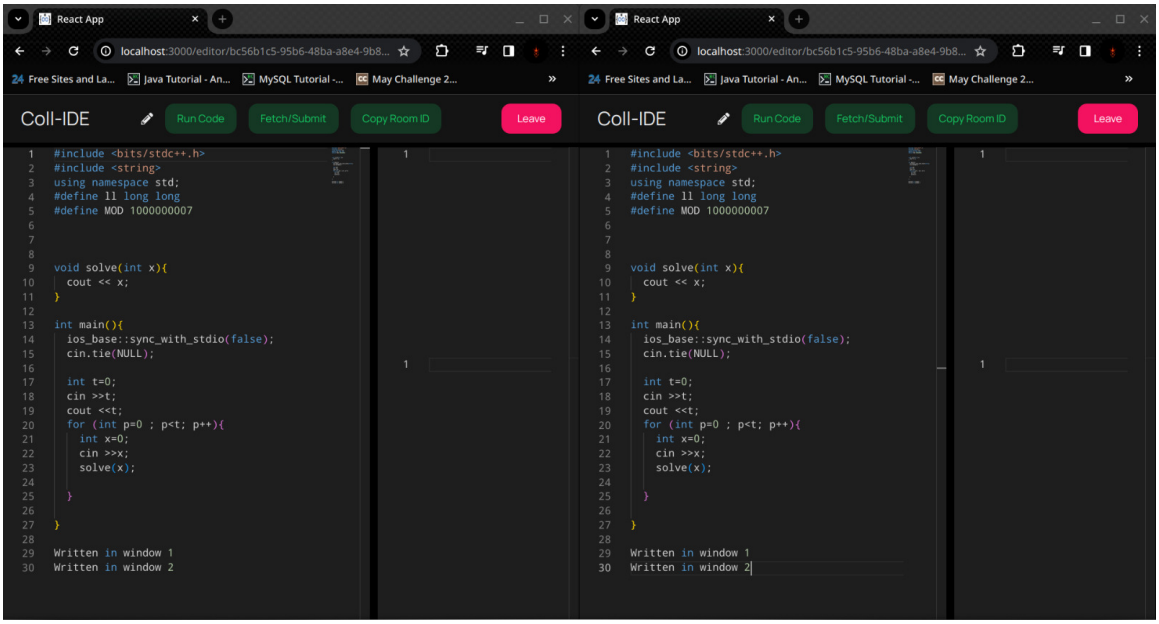

Figure 14: Coll-IDE Interface

7 Concluding Remarks

In conclusion, Coll-IDE represents a comprehensive and innovative solution for collaborative coding environments, tailored particularly towards competitive programming, hackathons, and group coding contests like ICPC. By leveraging cutting-edge technologies and integrating advanced features, Coll-IDE offers a seamless and efficient platform for programmers to collaborate, brainstorm ideas, and tackle coding challenges together.

With its intuitive user interface built using React.js and supported by Socket.IO for real-time collaboration, Coll-IDE fosters teamwork and enhances productivity by enabling multiple users to edit code simultaneously and communicate in real-time. The integration with MongoDB ensures robust data storage and management, while authentication mechanisms powered by Passport.js and JWT provide secure access control and session management.

Coll-IDE's integration with external platforms like Codeforces, facilitated by Cheerio, Axios, and Puppeteer, streamlines the process of fetching test cases and handling direct submissions, simplifying the workflow for competitive programmers. Additionally, the inclusion of a collaborative whiteboard/scribble area enhances creativity and facilitates idea sharing among users.

Overall, Coll-IDE stands as a testament to innovation in collaborative coding environments, offering a versatile and powerful toolset for programmers to collaborate, learn, and succeed together. As development continues and new features are added, Coll-IDE remains committed to providing a seamless and enriching experience for its users, driving forward the advancement of collaborative coding practices in the digital age.