International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 11, pp 196-198, November-2025
https://doi.org/10.47001/IRJIET/2025.911023

CodeMate: An AI Powered Coding Platform

[1]Prof. S. L. Vidhate, [2]Sanjana Patil, [3]Sakshi Mathure, [4]Mangesh Khapare, [5]Aniruddha Kedar

[1]Assistant Professor, Department of MCA, MET's Institute of Engineering, Nashik, Maharashtra, India
[2,3,4,5]PG Student, Department of MCA, MET's Institute of Engineering, Nashik, Maharashtra, India

***Abstract* - This paper presents a comprehensive analysis and implementation of Code Mate, an AI-powered platform designed to fundamentally enhance coding education in university and technical environments. Existing coding platforms focus primarily on problem-solving but lack real-time collaboration tools and mechanisms to maintain the integrity of student work. Code Mate solves these gaps by generating unique coding problems with AI for each attempt, enabling real-time pair programming, enforcing robust anti-cheating mechanisms, and providing guided hints at strategic intervals. The platform leverages modern web technologies and OpenAI integrations to deliver a secure, inter active, and highly adaptive environment that engages learners and educators alike. Code Mate stands out for its focus on teamwork, integrity, and the ability to track progress, offering a scalable solution for coding practice and assessments. This paper details the system's architecture, methodology, functional modules, evaluation strategy, limitations, and future roadmap.**

Keywords: AI coding platform, collaborative learning, pair programming, anti-cheat system, real-time coding, education technology, OpenAI, student assessment.

I. INTRODUCTION

1.1 Background and Problem Statement

The surge in demand for coding proficiency—driven by the proliferation of software-centric careers—has exposed the limitations of traditional coding practice platforms for students. Most existing systems enable problem-solving in isolation, often neglecting essential collaborative features true measures of fairness. This structural deficit is exacerbated by the absence of real-time teamwork, adaptive assistance, and effective anti-cheating mechanisms. Students require environments that not only challenge them with unique, personalized coding problems but also foster teamwork and encourage honest effort. Major limitations observed in legacy or semi-modern coding platforms include:

1) Repetitive Questions and Easy Access to Solutions: Most platforms recycle questions from static databases, making it possible for students to find answers online and undermining genuine skill development.

2) Lack of Real-Time Collaboration: Platforms seldom allow synchronous teamwork, preventing students from 2nd Sanjana Patil Department of Computer Engineering Institute Name City, Country email@institute.com experiencing pair programming—a proven method to enhance technical and social skills [3], [4].

3) Insufficient Integrity Controls: Traditional environments fail to prevent copying, screen-switching, or unauthorized resource access, leading to unfair assessments [6].

4) Minimal Adaptive Guidance: Static hints or step explanations do not align with individual learning speed, leaving students frustrated if they get stuck [5].

5) Limited Progress Tracking and Motivation: Without dashboards, leaderboards, or project modes, students lack visibility and motivation to improve [1].

6) Manual Reporting, Administration Difficulties: Instructors struggle to obtain real-time metrics, and course administration is often manual or paper-based [2].

1.2 Motivation and Research Objectives include:

- Delivering a secure, engaging, and collaborative coding environment.
- Empowering students to choose language and difficulty, and experience pair programming.
- Generating unique problems per session via OpenAI, minimizing unfair tactics.
- Blocking unauthorized behavior by disabling copy-paste and tracking screen activity.
- Unlocking guided hints after 15 minutes, balancing per severance and help.
- Tracking user progress and providing actionable feedback.
- Enabling future expansion through competitions, project modes, and AI mentorship.

II. RELATED WORK

The emergence of AI-powered educational platforms and real-time collaborative code editors has transformed digital learning [1], [2]. Research shows that adaptive coding environments and pair programming tools enhance skill acquisition and teamwork [3], [4]. AI-driven assessment

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 11, pp 196-198, November-2025

https://doi.org/10.47001/IRJIET/2025.911023

modules and anti cheat solutions—often employing computer vision, behavioral analytics, or system monitoring—are increasingly adopted to ensure test integrity and individualized progress [6].

III. SYSTEM ARCHITECTURE AND DESIGN

A. Overall Architecture and Technology

Stack Code Mate is architected as a cloud-based, modular web application with a three-tier structure: frontend, backend, and database:

- Frontend: HTML, CSS, JavaScript, React.js, Tailwind CSS.
- Backend: Node.js RESTful APIs.
- Database: Firebase Realtime Database.
- Real-Time Collaboration: WebSockets/WebRTC for synchronized editing [7], [8].
- AI Integration: OpenAI API for question generation and context-sensitive hints [5].
- Anti-Cheat System: Clipboard monitoring and screen activity detection [6].
- Notifications: SMTP or third-party services for email alerts.

B. Database and Component Diagrams

Firebase collections track Users, Sessions, Problems, Hints, Collaborations, Violations. Component and UML diagrams detail module interactions, activity sequences, and anti-cheat workflows.

IV. IMPLEMENTATION AND FUNCTIONALITY

A. User Module Features

- Secure OAuth and email/password login.
- Language and difficulty selection (Easy, Medium, Hard).
- AI-driven problem generation each session.
- Real-time collaborative editor with operational transformation.
- Pair programming invites and notifications.
- Timed AI hints unlocked after 15 minutes.
- Progress dashboard: solved problems, streaks, badges.
- Automated invite, notification, and tracking visualization.

B. Administrator Module Features

- System monitoring: usage metrics and violation logs.
- Question analytics: topic distributions and difficulty trends.
- Integrity controls: review and block violations.
- User management: CRUD operations for users and roles.
- Reporting tools: generate detailed engagement and fair ness reports.
- Model Training & Tuning: Administrators upload datasets—curated problems or anonymized submissions—to fine-tune the OpenAI model. The admin UI allows adjustment of hyperparameters (temperature, max tokens), monitoring of training progress, version history, and performance metrics.
- Manual and automated alerts for repeated integrity issues.

C. Reporting Capabilities

Administrators can generate:

- Problem completion reports.
- Pair programming effectiveness analytics.
- Integrity and cheating event visualizations.
- Longitudinal learning trend reports.
- Custom filters by user, session, date, problem type, and language.

V. TESTING AND EVALUATION

A. Test Methodology and Results

- Black-Box Testing: login flows, submissions, invites, hint logic.
- Integrity Testing: simulated copy-paste and screen switching attempts.
- Collaborative Scenarios: pair vs solo engagement, aligned with research [4].
- Negative Scenario Validation: conflict resolution via operational transformation [3].
- Usability Studies: feedback on hints and dashboards.
- Performance Assessment: scalability to 500 concurrent users.

B. Feasibility Analysis

- Technical: stable open-source frameworks.
- Economic: minimal API and hosting costs.
- Operational: multi-device support and real-time analytics.
- Legal/Social: GDPR-aligned data handling.
- Social: fosters teamwork and fair assessment.

VI. LIMITATIONSAND FUTURE ENHANCEMENTS

Current limitations:

- Internet dependency for real-time features.
- No offline problem bank.
- Basic anti-cheat; advanced behavioral analytics planned.
- Web-based only; mobile application forthcoming.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 11, pp 196-198, November-2025

https://doi.org/10.47001/IRJIET/2025.911023

- Manual pair configuration; dynamic matching planned.

Proposed enhancements:

- Native mobile app with offline support.
- Advanced proctoring: webcam and keystroke analytics.
- Automated model retraining pipelines.
- Feedback-driven AI tuning.
- Gamified leader boards and project modes.

VII. CONCLUSION

Code Mate integrates AI-powered problem generation, real time collaboration, and robust integrity controls into one plat form. Initial evaluation shows enhanced engagement, fairness, and scalability. Future work will focus on personalization, advanced proctoring, and expanded analytics to redefine coding education.

REFERENCES

[1] Codio Blog, "Teaching with OpenAI and Other Generative AI APIs in Codio," 2024.
[2] AlgoCademy, "The Revolutionary Impact of AI on Coding Education," 2024.
[3] M. Goldman, "Software Development with Real-Time Collaborative Editing," PhD Thesis, MIT, 2012.
[4] S. Faja, "Evaluating Effectiveness of Pair Programming as a Teaching Tool," Information Systems Education Journal, vol. 12, no. 3, 2014.
[5] Coursera, "OpenAI API for Beginners," 2024.
[6] IRJMETS, "Anti-Cheating System for Examination Hall Using Deep Learning," vol. 7, no. 4, 2025.
[7] J. Doe, "WebSockets for Real-Time Communication," Web Dev Journal, 2024.
[8] Monaco Editor, Microsoft (2023).* rowser-Based Code Editor. [https://microsoft.github.io/monaco-editor/] (https://microsoft.github.io/monaco-editor/

Citation of this Article:

Prof. S. L. Vidhate, Sanjana Patil, Sakshi Mathure, Mangesh Khapare, & Aniruddha Kedar. (2025). CodeMate: An AI Powered Coding Platform. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(11), 196-198. Article DOI https://doi.org/10.47001/IRJIET/2025.911023

****

