International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0444

e ISSN: 2584-2854
Volume: 03
Issue: 09 September 2025
Page No: 2818 - 2822

Edu Code: Transforming Digital Programming Education

Supriya Shrivastav[1], Abhilasha Kumari[2], Archit Mahajan[3], Goldi Kumari[4], Harash Sangral[5]
[1,2,3,4,5]Department of Computer Science and Engineering, AMC Engineering College, Bengaluru-560083, Karnataka, India.
Emails: *supriya.shrivastav@amceducation.in[1], iabhiofficial23@gmail.com[2], architg373@gmail.com[3], goldikthakur@gmail.com[4], harashsangral602@gmail.com[5]*

Abstract

Hands-on, collaborative work and secure assessments remain challenging to teach in programming remotely as well: most platforms are strong in delivering information but weak in encouraging students to work together online and in ways to securely assess their knowledge of the material. EduCode: Digital Coding Education Reinvented fills this gap with a cloud-based coding and assessment platform that combines real-time collaboration and built-in exam security. So, students can click to join the protected rooms and code together with the synchronized editors, and send the solution immediately. Teachers can follow progress through dashboards, pick questions at random, detect when users switch tabs, and take webcam shots to prevent cheating. The modular system is multilanguage capable and suitable for use on an institutional level. Evaluated results suggest that higher engagement and transparent evaluation are achieved, which may help reduce the gap between theory and practice of the REM programming education by EduCode.

Keywords: *Assessment integrity; Collaborative coding; EduCode; Online proctoring; Programming education*

1. Introduction

National closure of schools and shift to remote learning revealed existing inadequacies in the teaching of programming online: restricted genuine collaboration, shallow with feedback loops and academic integrity issues [1,2,14]. Although web IDEs and MOOCs have made it easier, they do not afford much of the social or supervisory aspects of lab [3,4]. Meanwhile, the need and the difficulty of maintaining a secure assessment at scale is also reflected in proctoring literature [5,6,9]. EduCode was designed with the aim of bringing these two threads together – collaborative work and authentic assessment – into one place. Its objectives are: (i) to facilitate live, peer-visible coding with immediate code execution; (ii) to support real-time teacher feedback and participation monitoring; (iii) to incorporate integrity measures (e.g., randomized problems, tab-switch alerts, snapshots taken at random times); and (iv) to be easy to deploy for most classrooms and exams. Related work provides some solutions: collaborative cloud applications [1,2,11] support shared workspaces, and virtual programming labs [3] are emulations of classroom practice. Other studies suggest question generation and runtime code manipulation [4,7] or study the sociocultural factors related to academic dishonesty [5,8,9]. In spite of this advancement, most existing systems consider collaboration and proctoring as independent issues. Even fewer combine both to offer a holistic learning experience offered in a traditional lab environment where practice, peer interaction and authentic assessment go side by side. EduCode fills this void by integrating synchronized live coding with smart proctoring. Students can write and run code together, and teachers can monitor, comment, and manage work - all within an integrated, class-aligned, coding environment. Randomized question order, tab-switch warnings, and webcam snapshots improve exam integrity while minimizing impact on user experience. EduCode combines interactive learning with fair assessment in one package, and thus makes programming education more dependable, engaging, and scalable in online learning for institutions.

1.1.Problem Statement and Motivation

In today's digital world, online learning has a problem. Learners have a hard time gaining hands on experience in programming. Many websites provide theory and coding references, but they cannot provide the environment of a physical classroom. In a regular class, students work together, help each other and get instant help from teachers. Therefore, even with many resources available, students are not able to learn coding practically. Problem 3: Maintaining Integrity in Remote Assessment.

2. Proposed System Overview

Proposed system, EduCode, was designed as a safe, interactive and platform to meet the deficiencies in existing online education equipment. Students are allowed to work together in a shared coding environment while teachers monitor dashboard progress and control assessments through an integrated, which is not possible with current systems. In view of real challenges during development, we implemented strict safety measures to ensure data privacy and maintain learning quality.

3. Method

3.1.System Requirements and Analysis

The development of EduCode: Transforming Digital Programming Education started by carefully examining the shortcomings of existing e-learning platforms. Feedback collected from students and instructors revealed three main needs: the ability to code together in real-time, immediate program execution, and trustworthy proctoring for assessments. A review of previous tools showed that, while some platforms offered collaborative features and others ensured exam integrity, very few combined both. These findings played a key role in shaping the platform's feature priorities and overall system design.

3.2.System Design

EduCode is built using a modular architecture that clearly separates the front-end interface, back-end services, and assessment monitoring components. The back end, powered by Node.js and Express, provides scalability and robust performance, while MongoDB Atlas handles secure and distributed data storage. Real-time interaction between users is made possible through WebSockets (Socket.IO), enabling smooth code collaboration and instant peer communication. On the front end, React ensures a responsive and user-friendly interface tailored to the needs of different roles within the platform

3.3.Collaboration Features

EduCode enables students to engage in real-time, multi-user programming sessions through shared editors, allowing them to work together seamlessly. Integrated chat and room management tools let instructors set up private coding spaces for assignments or exams. Unlike standard live coding platforms, EduCode also includes instructor dashboards that give teachers clear insight into each student's activity and contributions [1,2,11,15].

3.4.Secure Assessment

One of EduCode's standout features is its advanced proctoring system. The platform detects when students switch tabs to prevent multitasking during exams, and it uses randomized questions to make cheating more difficult. Webcam snapshots and gaze-tracking add extra layers of monitoring, while alerts notify instructors of suspicious activity. If unauthorized conduct continues the system can automatically submit the exam. This approach ensures a fair assessment experience while keeping usability in mind.

3.5.User Role Management

EduCode clearly defines three types of users: Admins, Instructors, and Students. Admins handle platform management and monitor overall performance, instructors design learning sessions and assessments, and students engage in coding rooms for practice and exams. By structuring access this way, the platform ensures both security and accountability while keeping each user's responsibilities well-defined [8,9].

3.6.Testing and Deployment

EduCode underwent testing with multiple users across different network conditions to evaluate latency, synchronization accuracy, and the responsiveness of alerts. The results demonstrated consistent performance during concurrent sessions, with smooth real-time collaboration and dependable proctoring notifications. Its deployment-ready architecture is designed for scalability, allowing EduCode to be effectively used in both individual classrooms and large-scale institutional settings [3,4].

4. Figures

The instructor starts the process by creating a room by entering the details needed. After the room is prepared, two access paths are created one of them being used by the students and other by the administrators. A proctoring extension is inbuilt to monitor their activities and a session timer is used to make sure that the duration spent is within the stipulated time. Towards the end of the session, the students will be given the choice of either leaving the room or handing in their test. Figure 1 shows Flowchart of Educode, Figure 2 shows Architecture of Educode

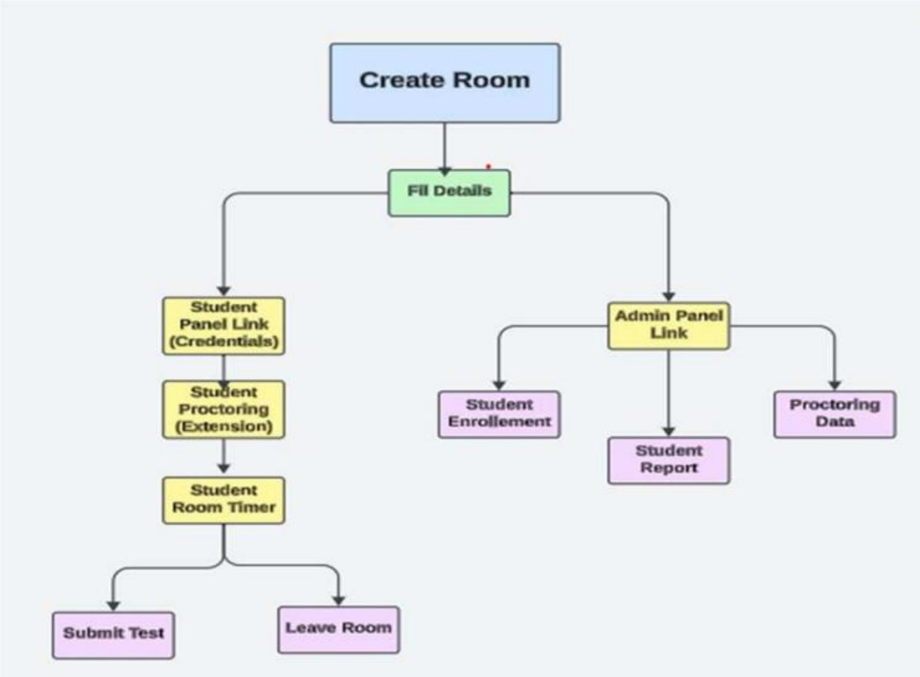

Figure 1 Flowchart of Educode

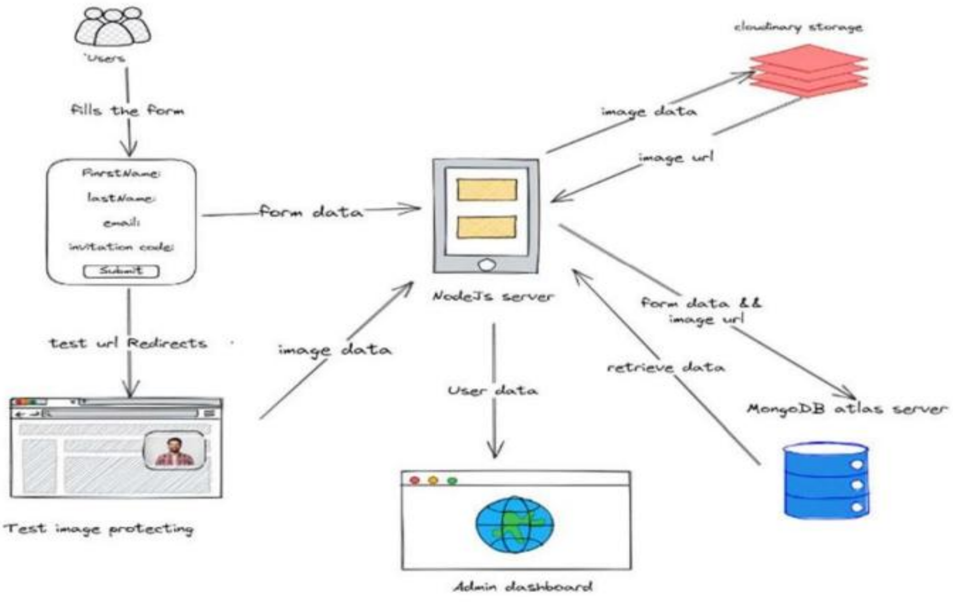

Figure 2 Architecture of Educod

5. Results And Discussion

5.1.Results

Preliminary testing of EduCode showed that it could enhance participation, oversight, and honesty in Web-based programming educating. Key outcomes include:

- Increased collaboration: Students claimed that they felt more confident about the process of coding in collaboration, with the help of synchronized editors and chat tools [1,2,11].
- Just and safe measurement: Tab warnings and randomization used in proctoring effectively reduced academic dishonesty [5,14].
- Scalability and stability: Tests also established that the platform is capable of supporting several classes and exams without major performance losses [3,4].
- Instructor empowerment: Dashboards enhanced the monitoring efficiency so that the challenges or misbehaviour could be identified sooner.

5.2.Discussion

Existing tools usually are focused on either online IDE to practice or proctoring to take tests. The innovativeness of EduCode is that the two domains have been combined. This allows learners to remain in the learning environment by placing the collaboration and assessment integrity in the same platform, thereby minimizing the hassle of tool switching. Such two-fold attention is consistent with studies that show collaborative learning promotes better conceptual knowledge [1,2], whereas credible proctoring protects the integrity of online learning [5,14]. The integration makes EduCode especially useful to institutions that are interested in scaling remote learning without reducing the academic standards. It can be extended in the future with the use of AI-generated feedback, adaptive difficulty changes, and programming language support.

Conclusion

EduCode: Change the education of digital programming A comprehensive response provides the restrictions on traditional online programming platforms. By integrating real -time collaboration with smart sales, the system creates an environment that is not only interactive but also safe and reliable. Students benefit from live coding sessions, synchronized editors and instant feedback, while teachers try to increase through dashboards and activity alerts. The results of the pilot trial showed that EduCode increased engagement, strengthened justice in the assessment, and enlarged effectively in different classes and institutional conditions. Unlike most current solutions, which either focus on collaboration or on fixed points of view, EduCode joins both these dimensions in a smooth stage. This contribution is significant for institutions looking to increase educational ethics, provide useful studying opportunities, and prepare students for industry expectations. Future improvements can explore the use of AI for individual feedback, support for more programming languages and advanced analysis to further enrich the learning experience. To conclude, EduCode has exhibited that it is possible to achieve secure, collaborative and programmable digital education: a model of influence can lead to the future of online and hybrid technical learning. EduCode fills this void by integrating synchronized live coding with smart proctoring.

Acknowledgements

We Thank our project guide and faculty members for their helpful guidance and support throughout this work. We appreciate our institution for providing the facilities and our friends for their useful feedback. Lastly, we thank our families for their encouragement and motivation during the project

References

[1]. CodeR: Real-time Code Editor Application for Collaborative Programming (Author: Aditya Kurniawan, Aditya Kurniawan, Christine Soesanto, Joe Erik Carla Wijaya) (https:// www. sciencedirect. com/science/ article/pii/S1877050915020608)

[2]. A Review of Enhanced Online Live Code Editors (Author: Anish Raj, Alaukik Deep and Rishi Chopra) (https:// papers. ssrn.com/sol3/papers.cfm?abstract_id=4486911)

[3]. Virtual Lab Environment for Programming Using MEAN Stack (Authors: Baby Vadlana, Pulluri Vineeth, K. Jhansi Lakshmi Bai, Md Sameer, Kovid Gajavelly, Syed Khaja Hakeemuddin)

(https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9716538)

[4]. Dynamic Formatting of Source Code in Editors (Authors: Rajesh Prabhu, Ninad Phutane, Shiv Dhar, Siddhesh Doiphode) (https://ieeexplore.ieee.org/document/8276008)

[5]. A Critical Analysis of Students's Cheating in Online Assessment (Authors: G. Bubas, A. Cizmesija) (https:// www. researchgate. net/publication/371000719_A_Critical_Analysis_of_Students_Cheating in_ Online_ Assessment)

[6]. A Comprehensive Survey on the AI based Fully Automated Online Proctoring Systems (Authors: Tejaswi Potluri and Venkatarama Phani Kumar Sistla)

[7]. Automated Generation of Question Paper for Online MCQ Test by Using Shuffling Algorithm (Authors: Mayuri Bankar, Pratiksha Bhor, Pranjal Bhalerao, Archana Dere)

[8]. The Construction of the Academic Honesty Education Model for Undergraduate Students in Online Education Based on Computer Technology (Authors: Lin Chen)

[9]. A Literature Survey on Online Examination and Proctoring System (Authors: Sneha A, Seetharamaraju SV, Adaveni Nithin, Shivaprasad Ashok Chikop)

[10]. Code Context-Based Reviewer Recommendation (Authors: Dawei Yuan) (https://journal.hep.com.cn/fcs/EN/10.1007/s11704-023-3256-9)