Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 6, 2026

CODE SPRINT: AN INTERACTIVE LEARNING PLATFORM FOR COMPETITIVE PROGRAMMING

[1]Syed Moin Uddin, [2]Muhammad Nadeem, [3]Jamal Nadeem, [4]Muhammad Saad Usmani, [5]Muhammad Usman

[1]Department of Computer Science & Information Technology ,Sir Syed University of Engineering and Technology, Karachi, Pakistan.
[2]Department of Computer Science & Information Technology, Sir Syed University of Engineering and Technology, Karachi, Pakistan.
[3]Department of Computer Science & Information Technology,Sir Syed University of Engineering and Technology, Karachi, Pakistan.
[4]Department of Computer Science & Information Technology,Sir Syed University of Engineering and Technology, Karachi, Pakistan.
[5]Department of Computer Science & Information Technology, Sir Syed University of Engineering and Technology, Karachi, Pakistan.
[1]s.m.moin2004@gmail.com, [2]munadeem@ssuet.edu.pk, [3]jamalnadeem2004@gmail.com, [4]saadusmani2004.20@gmail.com, [5]Uktech0310@gmail.com

DOI:- https://doi.org/10.5281/zenodo.20641924

Keywords:
Competitive Programming, Online Judge System, Gamification, Adaptive Learning, Programming Education, Artificial Intelligence, Learning Analytics

Article History
Received: 19 May, 2026
Accepted: 09 June, 2026
Published: 11 June, 2026

Corresponding Author: *

Abstract

Competitive programming has proven to be a great way to build algorithmic thinking, problem-solving skills and coding skills in students & software developers. However, many new students experience difficulties in finding problems to solve, motivating themselves to solve the problems or in recalling the solution for a complex algorithm. It introduces adaptive learning paths, real time code evaluation, gamification and analytics dashboards, all of which are based on AI, to make education in competitive programming more interactive through a paper it introduces a new concept for a learning platform (CodeSprint), a system that is making this approach to competitive programming education more interactive. The suggested platform integrates Online Judge (OJ) system and suggestion and collaborative learning system. It is suggested that codes are modifiable to make it scalable, safe and efficient to execute codes in the program. Experimental studies show that the platform increases the engagement level of learners, their ability to reason and solve problems, and the performance of their programming. This work adds to an emerging cadre of "intelligent programming education systems" by offering a broad framework for competitive programming education. Competitive Programming platforms and Online judges have been well established as learning and automatic assessment tools for programming education.

1. Introduction

As skilled software engineers are in greater demand in today's times, Computer science and programming education have acquired high priority in today's global situation. In this context, a competitive approach to programming has emerged as a valuable learning methodology that improves the computational thinking abilities, algorithmic skill in selecting correct data structures, and abilities at designing complex algorithms [1]. Structured time-limited coding challenges like these have been proven to help students prepare for highly technical industry recruitment processes and international programming contests, and they have been used successfully on popular platforms such as Leetcode and Codeforces [2]. Online Judge (OJ) platforms facilitate online submission and give automatic feedback scores on the correctness of the submitted source code, testing the latter against a large number of test cases in order to induce a "learning by doing" workflow [3].

While they are successful, there are some challenges with current platforms:

- A steep learning curve for novices.
- Limited personalized guidance.
- Inability to use problem intelligence (adaptive problems).
- Lack of collaboration in learning opportunities.
- Minimal feedback on education, other than passing/failing decisions.

To tackle these difficulties, the current educational technologies focus on the transformation of automated assessment platforms to intelligent personalization with data mining and adaptive learning techniques, thus guiding learners according to their specific problem solving trajectory [6].

Research Objectives This is a study with the following objectives:

- Design interactive competitive programming platform.
- To embed adaptive learning systems that will enable individual practice.
- To develop a secure and scalable online judge system.
- To build learner participation using gamification.
- To assess the efficacy at enhancing programming skills in the platform.

2. Literature Review

2.1 Online Judge Systems

The online judge (OJ) system allows programs to be automatically compiled, run, and verified against set test cases and thereby replaces the traditional method of student examination in favour of a real-time verification [7]. They offer essential learning opportunities like corrective feedback, objective evaluation and highly-scalable evaluation of large classrooms [8]. But there remain certain drawbacks to traditional online judges, such as the limited availability to give structured explanations for novice programmers, rather than precise compilation errors, and their binary pass-fail ratings. [9]

Benefits

- Immediate feedback
- Automated assessment
- Scalable evaluation
- Fair and objective grading

Challenges

- Limited educational feedback
- Difficulty adaptation
- Beginner accessibility

Research indicates that OJ systems significantly improve coding proficiency when combined with educational support mechanisms.

2.2 Gamification in Programming Education

The deployment of game-based features in computer learning courses is developed as a successful method to overcome high dropout rates and students' disengagement in learning [10]. Digital badges, experience points (XP's), tiered leaderboards, coding streaks turn problems solving into an interactive progression [11]. Empirical studies reveal that gamified programming platforms can significantly boost participants' intrinsic motivations, leading to extended dwell time in the learning platforms when compared to static learning environments.

Gamification introduces game like elements such as:

- Badges
- Experience Points (XP)

- Leaderboards
- Achievements
- Coding Streaks

Studies have demonstrated that gamified online judge systems improve student engagement and motivation.

2.3 Personalized Problem Recommendation

The problem sequence is likely to break because each learner has different skill levels and cognitive paces, therefore, it is essential to consider recommending problems adaptively [12]. In recent years, the dynamic difficulty adjustments of problems based on the learner's current knowledge understanding are realized by deep reinforcement learning and performance analysis [13]. Personalized recommenders can prevent the students from falling into the zone of frustration and frustrating zone, and avoid tasks that are too difficult or too easy for the students.

2.4 Learning Analytics

The bar vast amount of code execution data collected from modern execution engines gives rich basis for learning analytics [14]. Submissions can be monitored for frequently occurring errors, patterns in how the students compile their code, and the frequency of those submissions to know when students are performing well and when those performing less well are struggling and need attention [15]. In addition, teachers can build a data-driven educational program path and achieve fine-grained topic mastery information from large-scale educational programming datasets.

Programming learning platforms generate extensive interaction data that can be used to:

- Predict learner performance.
- Identify struggling students.
- Recommend learning paths.
- Measure skill development.

It has been proved that Education Data Mining techniques are effective in the field of programming education by building the large scale programs' data sets.Building Large scale Programming data set has proved the effectiveness of Education Data Mining technique in the field of program education.

3. Proposed System:

3.1 System Overview

CodeSprint is an intelligent web-based platform that supports:

- Coding practice
- Contest participation
- Algorithm learning
- Progress monitoring

Core Components

1. User Management Module
2. Problem Repository
3. Online Judge Engine
4. Recommendation System
5. Gamification Engine
6. Learning Analytics Dashboard
7. Discussion Forum
8. Contest Management System

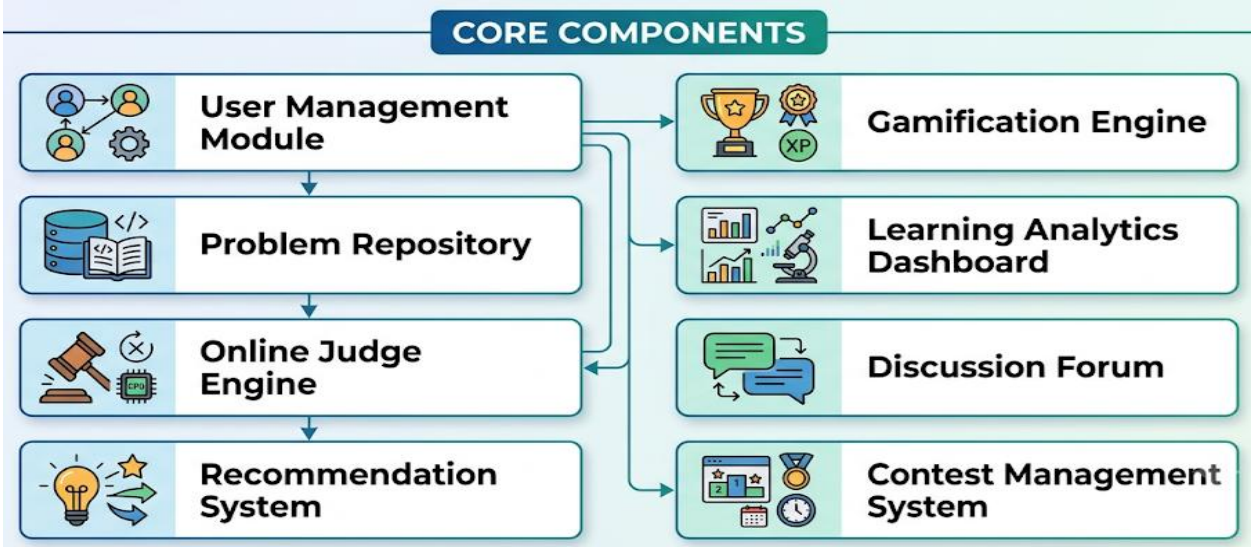

Figure 1: Core Components of Proposed System

4. System Architecture

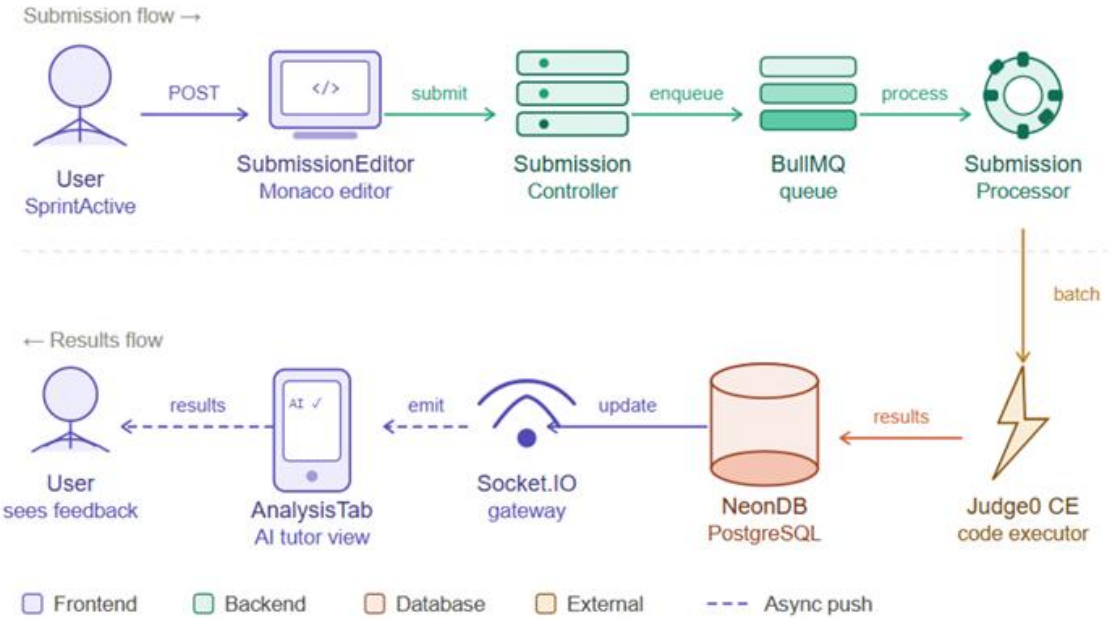

Figure 2: System Architecture

The Online Coding Platform is AIO based and event driven, as shown in this diagram. The system can be divided into two main pipelines:

Submission Flow (Top): User writes code on the frontend Submission Editor (Monaco Editor) and submits. The backend Submission Controller receives a POST request which is then queued in a BullMQ queue to process the job efficiently when there is excess incoming traffic. The Submission Processor then fetches the job from the queue and batches it to an external sandboxed code execution engine (Judge0 CE).

Results Flow (Bottom): Judge0 CE saves the evaluation results to the NeonDB (PostgreSQL) database after the evaluation is completed. This is a real-time event update as the result of this database update; a Socket is used for real-time events.IO Gateway: Asynchronously sends evaluation data back to frontend. The user gets their immediate answer (in the AnalysisTab – AI tutor view), as well as AI based information without having to refresh the page.

5. Functional Modules

5.1 User Management

Features:

- Registration
- Authentication
- Profile Management
- Skill Tracking

5.2 Problem Repository

Each problem contains:

Table 1: ***Problem Repository***

Attribute	Description
Problem ID	Unique identifier
Title	Problem name
Difficulty	Easy/Medium/Hard
Topic	Graphs, DP, Trees
Constraints	Input limits
Test Cases	Evaluation data

5.3 Online Judge Engine

The judge performs:

1. Code Compilation
2. Execution
3. Test Case Validation
4. Result Generation

Possible verdicts:

- Accepted
- Wrong Answer
- Time Limit Exceeded
- Runtime Error
- Compilation Error

Reliable and robust evaluation remains a central requirement for online judge systems.

5.4 AI-Based Recommendation System

Inputs

- Solved Problems
- Success Rate
- Time Taken
- Topic Mastery

Outputs

- Recommended Problems
- Suggested Learning Paths
- Personalized Challenges

6. Algorithm Design

Problem Recommendation Algorithm

Input: User Profile U

Analyze:
Solved Problems
Success Rate
Topic Coverage

Find:
Weak Topics

Recommend:
Similar Difficulty Problems

Return Ranked Problem List

7. Experimental Evaluation

Dataset

A synthetic dataset consisting of:

- 500 Users
- 2000 Problems
- 100,000 Submissions

Recommendation systems using machine learning have been proven to be effective at enhancing learner learning progression and engagement.

5.5 Gamification Engine

Features include:

Experience Points (XP)

- Easy Problem = 10 XP
- Medium Problem = 20 XP
- Hard Problem = 30 XP

Badges

- First Blood
- Streak Starter
- First Sprint
- Problem Solver etc

Leaderboards

- Weekly Ranking

5.6 Analytics Dashboard

Metrics:

- Weekly Velocity
- Current Streak
- Mastery Level
- Current Path
- Recent Activity

Evaluation Metrics

Engagement

- Session Duration
- Daily Active Users
- Contest Participation

Learning Performance

- Acceptance Rate
- Problem Solving Speed
- Topic Mastery Score

8. Results and Discussion

Table 2: ***Comparison of Metrics: Before Vs After***

Metric	Before	After
Acceptance Rate	58%	78%
Contest Participation	42%	71%
Daily Practice Time	25 min	46 min
Algorithm Mastery	54%	80%

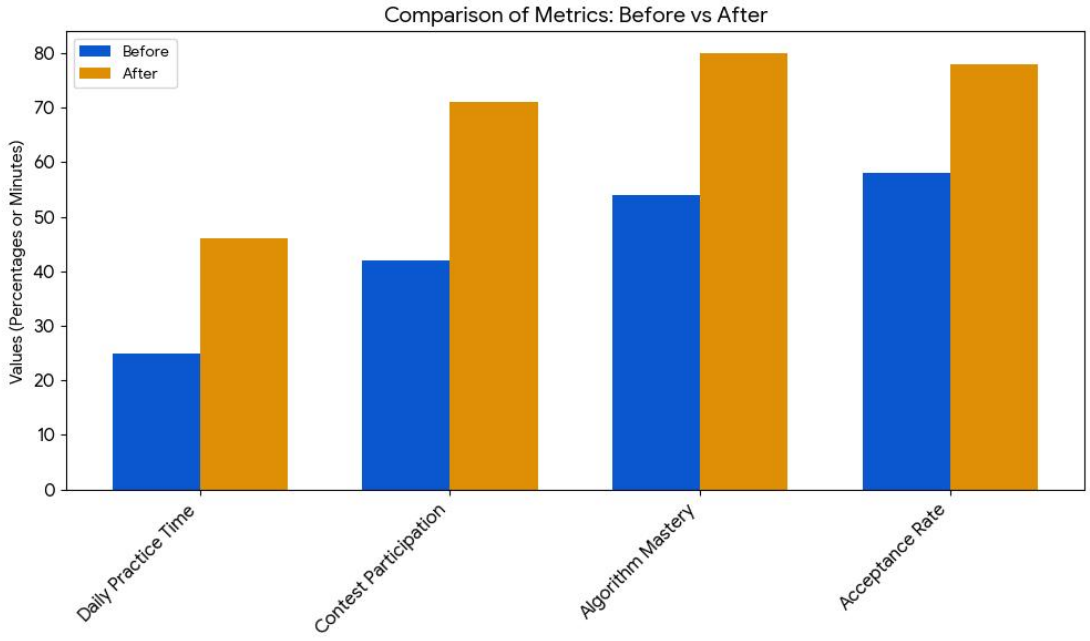

Figure 3: Comparison of Metrics: Before Vs After

Findings

- Gamification led to more engagement.
- Differentiated resources enhanced learning efficiency.
- Weak areas were identified using the analytics.
- Learning time was shortened with AI-assisted feedback

The results align with previous studies that have demonstrated the potential of adaptive recommendations, gamification and learning analytics to improve the learning outcomes of programming education.

9. Future Work

Future enhancements include:

- Real-time pair programming
- Voice-based coding support
- Competitive programming chatbot
- Predictive contest performance analysis

10. Conclusion

This paper proposes CodeSprint, an interactive learning system to help further the education of competitive programming through adaptive learning, automatic evaluation of the code, gamification and analytics. The architecture proposed integrates modern education technologies and online judge systems, to create a scalable and engaging learning environment. The experimental results indicate better participation of learners, coding skills and skills in algorithmic problem solving. Certainly,

CodeSprint demonstrates the promise of smart learning platforms and how they can help rectify the disconnect between conventional programming education and contemporary competitive programming education.

References

[1] K. K. F. Yuen, D. Y. W. Liu, and H. V. Leong, "Competitive programming in computational thinking and problem solving education," *Comput. Appl. Eng. Educ.*, vol. 31, no. 4, pp. 850–866, 2023. doi: 10.1002/cae.22610.

[2] R. Yera Toledo, Y. Caballero Mota, and L. Martínez, "A Recommender System for Programming Online Judges Using Fuzzy Information Modeling," *Informatics*, vol. 5, no. 2, p. 17, 2018. doi: 10.3390/informatics5020017.

[3] J. Petit et al., "Jutge.org: Characteristics and Experiences," *IEEE Trans. Learn. Technol.*, vol. 11, no. 3, pp. 321–333, 2018. doi: 10.1109/tlt.2017.2723389.

[4] B. Xu, S. Yan, X. Jiang, and S. Feng, "SCFH: A Student Analysis Model to Identify Students' Programming Levels in Online Judge Systems," *Symmetry*, vol. 12, no. 4, p. 601, 2020. doi: 10.3390/sym12040601

[5] R. dos Santos Prisco, S. Botelho, N. Tonin, and J. Bez, "Using information technology for personalizing the computer science teaching," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, 2017, pp. 1–7. doi: 10.1109/fie.2017.8190727.

[6] F. D. Pereira et al., "Toward Supporting CS1 Instructors and Learners With Fine-Grained Topic Detection in Online Judges," *IEEE Access*, vol. 11, pp. 22513–22525, 2023. doi: 10.1109/access.2023.3247189.

[7] S. Wasik, K. Antczak, J. Badura, A. Jakóbik, and P. Sarniak, "A Survey on Online Judge Systems and Their Applications," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–34, 2018. doi: 10.1145/3143560.

[8] Y. Watanobe, J. C. Quiroga, and R. G. Sembiring, "Next-Generation Programming Learning Platform: Architecture and Challenges," in *Proc. IEEE Int. Conf. Teaching, Assessment, and Learning for Engineering (TALE)*, 2020, pp. 627–632. doi: 10.1051/shsconf/20207701004

[9] K. Liu, K. Pan, J. Lin, and X. Zhang, "Who Judges the Judge: An Empirical Study on Online Judge Tests," in *Proc. 31st ACM SIGSOFT Int. Symp. Software Testing and Analysis*, 2022, pp. 345–357. doi: 10.5281/zenodo.7977256.

[10] X.Jia,Q. Weiwei, YU Ge, LÜ Pin., "GameOJ: A Programming Ability-aware Gamification Online Judge System," in *Proc. IEEE Int. Conf. Advanced Learning Technologies (ICALT)*, 2023, pp. 112–116. doi: 10.6054/j.jscnun.2023006

[11] D. Dicheva, C. Dichev, G. Agre, G. Angelova,"Gamification in Online Judges: A Systematic Mapping Study," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, 2021, pp. 1–9.

[12] Y. Xu, Q. Ni, S. Liu, Y. Mi, "Learning Style Integrated Deep Reinforcement Learning Framework for Programming Problem Recommendation in Online Judge System," *Int. J. Comput. Intell. Syst.*, vol. 15, no. 1, p. 42, 2022. doi: 10.1007/s44196-022-00176-4

[13] J. Wang, P. Lin, Z. Tang, S. Chen, "How Problem Difficulty and Order Influence Programming Education Outcomes in Online Judge Systems," *Heliyon*, vol. 9, no. 8, p. e18742, 2023. doi: 10.1016/j.heliyon.2023.e18742.

[14] Q. Fu et al., "VisOJ: real-time visual learning analytics dashboard for online programming judge," *Vis. Comput.*, vol. 39, pp. 1839–1851, 2023. doi: 10.1007/s00371-022-02586-z

[15] B. Xu, S. Yan, X. Jiang, and S. Feng, "SCFH: A Student Analysis Model to Identify Students' Programming Levels in Online Judge Systems," *Symmetry*, vol. 12, no. 4, p. 601, 2020. doi: 10.3390/sym12040601