

Article

Not peer-reviewed version

Creating a Platform for Competitive Programming Training with Integration of Automatic Decision Checking System

[Nureles Almazbek uulu](#)^{*} and Zhenishbek Orozakhunov

Posted Date: 2 May 2025

doi: [10.20944/preprints202504.2557.v1](https://doi.org/10.20944/preprints202504.2557.v1)

Keywords: competitive programming; full-stack development; judge0; spring boot; angular



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Creating a Platform for Competitive Programming Training with Integration of Automatic Decision Checking System

Almazbek uulu Nureles ^{1,*} and Zhenishbek Orozakhunov ²

¹ Student of Computer Science, International Alatoo University, Department of Computer Science, Ankara 1/8, Bishkek, Kyrgyzstan, 720048

² PhD Student (Engineering and Informatics faculty) Employment, Ankara 1/8, Bishkek, Kyrgyzstan, 720048

* Correspondence: nureles.almazbekuulu@alatoo.edu.kg

Abstract: Many students in Kyrgyzstan, especially beginners, face difficulties accessing competitive programming platforms that are tailored to their needs. Existing systems primarily cater to advanced users or international audiences, making it harder for newcomers to engage effectively with algorithmic problem-solving. This project proposes the development of a competitive programming platform designed specifically for beginner learners. The system includes interactive problem statements, real-time code execution using external APIs and automated evaluation of submitted solutions. The platform is built using a modern full-stack architecture with Angular, Spring Boot, and PostgreSQL, and integrates third-party services such as Judge0 for multi-language code execution. The goal is to encourage algorithmic thinking and programming skill development through a user-friendly and localized interface. Future enhancements will include a broader problem set, gamification elements, and educational resources such as tutorials and solution explanations.

Keywords: competitive programming; full-stack development; judge0; spring boot; angular

Classifier: 004.4 (Software Engineering) + 371.3 (Educational Technology)

Introduction

Competitive programming could be a challenging learning process for newcomers if current platforms cater more to sophisticated users or presuppose previous experience with algorithmic thinking. Although popular systems provide extensive collections of problems and accommodate various languages, they do not necessarily come with simple-to-use, streamlined interfaces and linear learning pathways. That could deter new users, especially those who are just starting to learn data structures, algorithms, and code optimization techniques.

Emerging trends in web development, cloud APIs, and educational platform design have enabled the delivery of novel possibilities for interactive programming learning. By combining real-time code execution, automatic grading, and intuitive interfaces, one can now build platforms that have the potential to guide step-by-step skill acquisition in programming.

This project is for creating an interactive competitive programming platform for beginners. The system has problem statements with live interaction, test case feedback, and server-side code verification logic based on external execute APIs. Angular is on the frontend, Spring Boot as backend services, and PostgreSQL for storing persistent data in the database. The system is built in such a way that users will have the ability to gain confidence and problem-solving skills step by step through guided problem-solving and feedback.

Hypothesis: Real-time feedback and step-by-step guided problem-solving competitive programming web application for learners will have higher user interaction and quicker learning outcomes than traditional practices of coding exercises.

Literature Review

Instructional technology history has witnessed a paradigmatic shift in problem-solving and programming education. Adaptive learning environments, immediate feedback systems, and individualized learning have maximized learning and enhanced interactivity, especially for novice learners. Bloom's (1984) initial research on one-to-one tutoring laid the groundwork for the effect of personalized learning strategies. Modern adaptive learning platforms like Khan Academy and Duolingo mirror aspects of this personalization with varying degrees of difficulty aligned to student performance to facilitate scalable and student-driven learning.

Aside from these innovations, the majority of mass-market platforms cater to broad, global groups and assume a minimum level of familiarity with ideas and sink many new beginners. This shortcoming has been recognized by studies invoking adaptive platforms that value learners' heterogeneous educational backgrounds and unique learning patterns (Choi et al., 2019; Raza & Alruily, 2021). Adaptive systems not only enhance access but also enhance learner engagement and satisfaction.

De facto algorithm practice and technical interview preparation coding competition platforms such as Codeforces, HackerRank, and LeetCode are. However, their steep learning curves, lack of onboarding assistance, and competitive design scare off beginners (Nguyen et al., 2020). Practice entails design solutions for new centering, such as problem explanation scaffolding, interactive hints, and short feedback loops to counteract cognitive loads and facilitate stepwise learning.

The integration of live code execution APIs like Judge0 and Sphere Engine has helped modern education software to provide interactive problem-solving sessions. The platforms have facilitated enhanced automated feedback, verified test cases, and the ability to support multiple languages within sandbox environments (Kulkarni & Deshpande, 2021). Their integration lowers backend complexity and enhances the user experience by providing instant feedback and error detection.

Gamification also turned out to be the most powerful impeller of web-based learning. Proof also indicates that such features such as XP points, achievement badges, progress bars, and streak counters do significantly increase activity as well as learning frequency on the users' sides (Anderson et al., 2020). Remaining competitive and friendly, however, still remains top notch, particularly in training environments geared towards new trainees.

In addition, graphical code runtimes like Python Tutor and intelligent tutoring systems (ITS) offer more support in terms of code behavior explanation and debugging assistance (Guo, 2013; Alvarado et al., 2022). These types of tools allow deep understanding by bridging the syntax and logic gap. Furthermore, integrating AI-based adaptive mechanisms into ITS enables content recommendation, difficulty adjustment, and performance monitoring, which are crucial for personalization.

An iterative design-science approach has been used in earlier education platform research to incrementally improve user interface and functionality with real-world feedback. Agile development cycles like sprint releases allow platforms to evolve based on user needs effectively. Feedback from stakeholders during ongoing development ensures critical requirements like usability, offline capability, content organization, and analytics for learning are addressed at different stages of the development lifecycle.

Finally, knowledge management systems—storing and showing to be reused subsequent material—allow scalability and ongoing improvement. Material gathered anonymously concerning students' performance can be used to make training strategies more subtle and reveal root misconceptions.

In summary, the combination of the features of personalized learning, gamification features, instant feedback mechanisms, and adaptive development procedures forms a firm basis to form an efficient rival of a learning coding portal for new coders. According to data from existing portals and research studies, the targeted system will provide new coders with an attractive, motivating, and friendly interface to learn coding.

Methods

This subsection describes the system architecture, development process, and testing methodologies used in building the competitive programming platform.

1. System Architecture and Technology Stack

The platform has been developed as a full-stack web application with modular architecture. The backend has been developed using **Spring Boot (Java)** and **Angular 18** with components separated for the frontend. **PostgreSQL** has been used as the database to keep users, problems, submissions, and performance stored.

Code execution is managed by the integration of **Judge0 API**, which provides feature-rich real-time evaluation of code provided by the user in a variety of programming languages. RESTful interaction enables communication between frontend, backend, and external code execution service.

2. Implementation Steps

Backend Development

The backend consists of multi-layered architecture consisting of:

- **Controller Layer** for handling HTTP requests.
- **Service Layer** for business logic like test evaluation and grading.
- **Repository Layer** based on Spring Data JPA for data storage.

If utilized as microservices, inter-service communication relies on FeignClient and Spring Cloud Eureka for load balancing and service discovery.

Frontend Development

Angular frontend includes:

- **Dashboard:** Available coding problems and user stats are shown.
- **Code Editor:** Code editor within a web application to write code and run it.
- **Result Panel:** Test case outcome, instantaneous feedback, and run status are rendered.

Angular reactive properties and routing modules manage the state and navigation across the platform with ease.

3. User Interface Design

Principles of user-centered design, which are founded on usability research, were used to create the interface (Ismailova & Ermakov, 2024). Paper sketching and wireframes were used to create the initial structure, pattern of navigation, integration of the editor, and areas for giving feedback on the results.

Readability, simplicity, and features for new users were assured to provide a new face. Syntax highlighting enforced new face with attractive designs which were desktop as well as mobile friendly.

Polish was established through initial prototypes and refined through refinement by small user groups so as to increase usability and readability visually. Feedback loops helped increase the readability as well as the responsiveness of workflows.

4. Evaluation Methods

Pilot test was given to students lacking competitive programming experience. The following were tracked:

- First problem time.
- Attempt/submits.
- Subjective difficulty and user satisfaction.

Surveys and observation were utilized for measuring the effectiveness of the platform to engage learners and anchor problem-solving. Long-term retention, learning analytics, and adaptive content delivery will be tested in future studies.

Results

The product that was built is an interactive, guided environment to code and learn with submission of on-the-fly code and comments. Focusing on usability and responsiveness, the product enhances learning efficacy for new and intermediate students.

The key achievements are:

- **Ergonomic Submission of Code:** The users are able to code, execute, and submit solutions using one unified editor, and the result is displayed almost in real time.
- **Feedback-Driven Learning:** The mechanism identifies errors and defective test cases, allowing users to gradually enhance their code.
- **Scalable Content Management:** Problem sets can be authored, tagged, and mapped along axes of difficulty, subject, or language and easily scaled by admins.

Though formal usability testing has yet to be undertaken, initial user response has pointed towards ease of use, clear UI, and instant feedback that will go a long way towards making the learning process simpler. These early indicators are supported by findings of more recent research into education technology, which have suggested that instant reward and visual simplicity lead to more engagement and better retention of problem-solving strategies.

Conclusion

This project demonstrates how technology can organically enhance programming education through the use of an interactive and dynamic competitive programming environment. The environment is suitable for real-time code reviewing, learning through problem sets, and gameful interaction to facilitate intense user interaction.

With its user-friendly front-end and performance-based back-end, the site bridges the gap between learning from textbooks and coding exercises. As it is, before the chance for official usability testing, the design concept—a user-centered design concept based on best practices of education technology—demonstrates great potential for student engagement, problem-solving recall improvement, and general interest in computer programming.

This project illustrates the tremendous potential of composite tools and feedback systems to craft an organic, interactive world for students to learn fundamental and advanced programming principles.

References

1. Ali, Z. (2020, February). Artificial intelligence (AI): A review of its uses in language teaching and learning. *IOP Conference Series: Materials Science and Engineering*, 769(1), 012043. IOP Publishing. <https://doi.org/10.1088/1757-899X/769/1/012043>
2. Google. (n.d.). *Google | mobile_object_labeler_v1 | Kaggle*. <https://www.kaggle.com/models/google/mobile-object-labeler-v1>
3. Hajizadeh, M., Sabokrou, M., & Rahmani, A. (2023). MobileDenseNet: A new approach to object detection on mobile devices. *Expert Systems with Applications*, 215, 119348. <https://doi.org/10.1016/j.eswa.2022.119348>
4. Haristiani, N. (2019, November). Artificial Intelligence (AI) chatbot as language learning medium: An inquiry. *Journal of Physics: Conference Series*, 1387(1), 012020. IOP Publishing. <https://doi.org/10.1088/1742-6596/1387/1/012020>
5. Heil, C. R., Wu, J. S., Lee, J. J., & Schmidt, T. (2016). A review of mobile language learning applications: Trends, challenges, and opportunities. *The EuroCALL Review*, 24(2), 32–50. <https://doi.org/10.4995/eurocall.2016.6402>
6. Ismailova, E., & Ermakov, A. (2024). Analysis of user experience data and methodology of application to improve the development of user interface. *Preprints*, 2024051624. <https://doi.org/10.20944/preprints202405.1624.v1>

7. Ivić, V., & Jakopec, T. (2016). Using mobile application in foreign language learning: A case study. *Libellarium*, 9(2), 121–132. <https://doi.org/10.15291/libellarium.v9i2.240>
8. Liu, P. L., & Chen, C. J. (2023). Using an AI-based object detection translation application for English vocabulary learning. *Educational Technology & Society*, 26(3), 5–20. <https://www.jstor.org/stable/26942344>
9. Machine learning with ARCore. (n.d.). *Google for Developers*. <https://developers.google.com/ar/develop/machine-learning>
10. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510–4520). <https://doi.org/10.1109/CVPR.2018.00474>
11. Son, J. B., Ružić, N. K., & Philpott, A. (2023). Artificial intelligence technologies and applications for language learning and teaching. *Journal of China Computer-Assisted Language Learning*. Advance online publication. <https://doi.org/10.4018/JCALL.2023010101>
12. Topsakal, O., & Topsakal, E. (2022). Framework for a foreign language teaching software for children utilizing AR, voicebots and ChatGPT (large language models). *The Journal of Cognitive Systems*, 7(2), 33–38.
13. Уровень знания кыргызского и русского языка в регионах КР — результаты переписи. (2023, September 29). *Sputnik Кыргызстан*. <https://ru.sputnik.kg/20230929/uroven-znaniya-kyrgyzskogo-russkogo-yazykov-regionah-kr-1079010702.html>
14. Wan Daud, W. A. A., Ghani, M. T. A., Rahman, A. A., Yusof, M. A. B. M., & Amiruddin, A. Z. (2021). ARabic-Kafa: Design and development of educational material for Arabic vocabulary with augmented reality technology. *Journal of Language and Linguistic Studies*, 17(4), 1760–1772.
15. YOLOV8: State-of-the-Art Computer Vision Model. (n.d.). <https://yolov8.com/>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.