

Criterion A

Scenario

My client is my IT/computer science teacher in the Russian school. During the interview I understood the problem he faces in the daily teaching basis - the students have to train for competitive programming competitions, and the trainings require carefully made problemsets which are organised and accessible for the students. The school doesn't have a centralised system to store the trainings and navigate between them, as well as providing interesting study information. This decreases the overall student interest in the competitive programming field and makes it more difficult for my teacher to provide helpful materials to the students. The solution that is currently in use is just manual checking of people's submissions (which is slow and error-prone) and rarely asking the students to make the trainings as the teacher himself doesn't have time.

My teacher suggested to create several programming trainings on different topics and come up with a rational way of structuring the links to them, as well as providing additional learning materials and information (see Appendix 1 for details). There is currently no need in a very advanced system, but I had to make my product as flexible as possible. The main problems that can arise in the process of development are the old technical services which should be used and the specific requirements of the field of competitive programming, for instance, the need to be specific in terminology and understanding of the topic level.

Rationale

My response to my teacher's request is to make a set of trainings on a world-class competitive programming training platform named Codeforces and to create a modern, nice-looking website to set up easy-to-use structured links to the coding trainings, as well as some additional resources and functions.

The students will be able to see all the trainings structured by difficulty level and topics covered and all the links to them, as well as navigate additional learning information and helpful resources on both competitive and practical programming. The real-time scoreboard for school competitive programming team members will also be included to increase the competitiveness level and thus boost the learning process. A built-in browser-based tool for problem recommendation will be implemented as a part of the website. The website will be open-source and editable for everyone with teacher's (or my) approval.

The website will be made using Jekyll, a fast, adaptive, and modern static site generator. This way I can host the website for free on GitHub Pages and not overcomplicate the product, limiting myself to JavaScript code and Markdown/HTML/CSS editing without any backend. The stack used to make the trainings themselves is the Polygon platform which is a native Codeforces problemsetting tool.

The final result should provide my teacher with a combined product of the coding trainings on codeforces.com and a website to organise the links to those trainings, provide additional learning resources and simplify access to different tools by combining them into one website. It will reduce the time it takes to prepare the lessons and collect the learning material and guarantee that the content is high-quality and accessible. This will improve the quality and entertainability of my client's classes and let the teacher edit the website and trainings himself because extensibility is one of the core criteria of the structure. The successfulness of the project will be determined by testing in accordance with the Test Plan (see Appendix 2 – Test Plan) and the success criteria below.

Success criteria

1. Accessibility - the website should present all the information in a beautiful way on all devices and browsers. Should be publicly available without downloading (e.g., through the browser).
2. Performance - the website should be fast to load and responsive to the user's actions.
3. Educational value - the trainings must teach the students some problem-solving skills and be a practical way of checking the knowledge.
4. User interface - the website and trainings should be easy to navigate and use to focus on the knowledge and practice part.
5. Content - the trainings and the website should provide a variety of content and information on different topics. Should be easily extensible and maintainable.
6. Internal structure - it should be easy to extend, change and remove information, as well as give the admins full control over the content.

7. Adaptivity - the website should look good on various devices, from laptops and PCs to mobile phones.