

# Codeforces Analyser using API Integration

Priyanshi  
Computer Science Department  
KIET Group of Institutions  
Ghaziabad, India  
priyanshi.2024cs1172@kiet.edu

Priyanshu Raj  
Computer Science Department  
KIET Group of Institutions  
Ghaziabad, India  
priyanshu.2024cs1113@kiet.edu

Sagar Srivastava  
Computer Science Department  
KIET Group of Institutions  
Ghaziabad, India  
sagar.2024csit1162@kiet.edu

Sreesh Gaur  
Computer Science Department  
KIET Group of Institutions  
Ghaziabad, India  
sreesh.gaur@kiet.edu

**Abstract**— Building universal apps based on feature-rich web Application Programming Interfaces (APIs) makes a progressive web application (PWA) an appealing choice. Although flexible, these large APIs invariably result in a notable increase in the attack surface of the API, which typically relates to a functionality that the application neither needs nor wants. Programmatically eliminating unnecessary features from an application is known as software debloating, and it is a potentially effective way to lower the API attack surface. A progressive web application (PWA) is a preferable substitute when developing universal apps with feature-rich web Application Programming Interfaces (APIs). While accommodating, these big APIs frequently lead to a noticeable increase in the attack surface of the API, usually related to a functionality that the application doesn't need or want. One possible solution to reduce the API attack surface is software debloating, which is deleting unnecessary functionality from an application programmatically. Unfortunately, debloating PWAs is challenging due to the monolithic structure and non-deterministic execution of a modern web browser. In this work, we present DeView, a practical solution that reduces the attack surface of a PWA by blocking web APIs that are available but unnecessary.

**Keywords**— Web Application Programming Interfaces (APIs), Performance Evaluation, Code Submission Analysis, Real-time Insights

## I. INTRODUCTION

Codeforces is a popular platform for competitive programming, used by programmers worldwide to test their coding skills in various challenges. The Codeforces analyser is a tool that enhances the understanding of algorithmic concepts and displays the execution of code in a visual format. This Research Paper aims to explore the features of the Codeforces analyser, including the different visualization tools used and how they improve the comprehension of algorithmic concepts. The Paper will begin by discussing how the Codeforces analyser displays the execution of code, providing a detailed explanation of the processes involved. The analyser uses various visualization tools to represent different aspects of the code execution, such as highlighting the code blocks that are currently being executed or displaying the memory usage of the program. The Paper will delve into the different visualization tools used and how they contribute to a better understanding of the code execution process. Furthermore, the Paper will examine how the Codeforces analyser enhances the understanding of algorithmic concepts, such as data structures and algorithms. By displaying the execution of code in a visual format, the Analyser allows programmers to observe how different algorithms and data structures work in real-time. This can help programmers identify

potential optimizations and gain a deeper understanding of how to implement these concepts in their code. Overall, the Paper will provide a comprehensive analysis of the Codeforces analyser, its features, and its impact on the understanding of algorithmic concepts. The next section will delve into how the Codeforces analyser displays the execution of code, providing a detailed explanation of the processes involved.

Codeforces Analyser is a cutting-edge tool that combines software development and competitive programming. It uses API integration to make code submission analysis on the Codeforces platform more efficient. With real-time insights about their Codeforces submissions, our tool seeks to empower developers and competitive programmers in the fast-paced world of programming competitions, where speed and accuracy are critical. Through a smooth integration with the Codeforces API, this analyzer provides an in-depth analysis of coding performance, pinpointing areas of strength, weakness, and potential improvement. The purpose of this introduction is to lay the groundwork for a discussion of Codeforces Analyser's features and functionalities, as well as to show how it can improve the coding experience and learning process for those that compete in Codeforces programming.

## II. RELATED WORK

Competitive programming platforms serve as pivotal arenas for honing coding skills and fostering algorithmic problem-solving capabilities among programmers. Codeforces, a prominent platform in this domain [1], has undergone significant evolution to meet the dynamic needs of its user base. As an integral part of this evolution, the integration of Application Programming Interfaces (APIs) has emerged as a key feature, transforming the platform's functionalities and user experience [2].

### 1. Competitive Programming Platforms:

Several platforms have been thoroughly investigated in this field, with Codeforces being one of the more prominent examples [1]. These platforms provide a competitive atmosphere that develops algorithmic problem-solving skills and provide a variety of coding challenges.

### 2. API Integration in Competitive Programming:

According to recent study, integrating APIs into competitive programming platforms has gained attention [2]. Real-time

contact between users and the platform is made possible by APIs, which also allow for customized feedback, leaderboard updates, and dynamic analysis of code inputs.

### **3.Learning and Coding Experience:**

The literature emphasizes how crucial it is for competitive programming to have tools that improve learning and coding [3]. By giving users real-time insights for improvement, API integration in platforms such as Codeforces adds to a more dynamic and interactive learning environment.

### **4.Real-time Analysis and Feedback:**

There has been interest in real-time analysis and feedback techniques facilitated by APIs [4]. Codeforces provides real-time feedback on code submissions via APIs, giving users valuable insights to improve their coding skills.

### **5.Effect of Platforms Enabled by APIs on Programming skill:**

Studies have examined the influence of platforms enabled by APIs on programming skill [5]. Platforms with integrated APIs, such as Codeforces, are essential for participants to strengthen their coding and computational skills.

### **6.Development of Competitive Programming Platforms:**

Research on the development of competitive programming platforms, particularly the incorporation of APIs, has been ongoing [6]. As a representative example, API advances have led to revolutionary changes in coding challenges, contests, and overall functionality at Codeforces.

### **7.Future Trends and Innovations:**

Future trends and innovations in competitive programming are the subject of pioneering research [7]. The incorporation of APIs is probably going to play a major role in determining the features and capabilities of the upcoming generation of platforms such as Codeforces.

### **8.Competitive Programming with Interactive Feedback:**

The usage of interactive feedback systems in competitive programming platforms has drawn notice because of its potential to increase user engagement [4]. Thanks to its API interface, Codeforces makes it possible for participants to receive instant feedback on their code submissions, which enhances the learning process.

### **9.Continuous Learning in Competitive Programming:**

Codeforces advances the notion of continuous learning in competitive programming by virtue of its features driven by APIs [8]. The API of the platform guarantees that users may access current challenges, resources, and feedback, creating

an atmosphere where learning is a continuous and flexible process.

### **10.Algorithmic challenges and skill development:**

One of the main draws of sites such as Codeforces is their capacity to offer skill-development-oriented algorithmic challenges [9]. Users improve their algorithmic thinking and problem-solving skills by taking part in competitions and figuring out challenging puzzles [9].

### **11.Analyzing Algorithmic Performance with Codeforces Data:**

Research has looked into many approaches to assess algorithmic performance [10]. Researchers can learn more about the elements impacting user success and participant obstacles by examining submission patterns, error rates, and contest participation [10].

In addition to offering a strong setting for the development of personal skills, the platform encourages teamwork, interaction, and lifelong learning. Codeforces is still a dynamic organization that embraces API-driven tactics to influence the direction of coding education and skill as the competitive programming industry changes.

## **III. METHODOLOGY**

**Improving Codeforces User Experience with Predictions and Visualizations** Programmers all over the world use competitive programming platforms to hone their coding skills. This study focuses on the creation and application of Codeforces analyser, a tool designed to improve user experience by using predictive analytics and logical visualizations. Three major elements are included in the analyser, which uses a technology stack that includes HTML, CSS, JavaScript, and Google Charts, MDL: a topic-wise countability analysis, an inventive heat mapping system, and a user-centric home page. It also introduces peer-to-peer comparison graphs and makes use of machine learning methods to anticipate virtual ratings. In addition to covering the technology stack, user interface design, feature implementation, and the complex inner workings of machine learning for predictive analytics, this paper outlines the thorough approach underlying the development of Codeforces Analyser.

### **A. The Technology Stack**

The core of Codeforces Analyser is its advanced technological stack, which was carefully chosen to guarantee stability and adaptability. The tool's structural foundation is HTML, which offers a framework for organizing content. In turn, CSS controls the layout and visual appeal, resulting in a simple and interesting user experience. Dynamic scripting languages like JavaScript make it easier to create interactive content and enable real-time changes and responsiveness. Integrating Google Charts, a robust charting library, produces aesthetically pleasing and educational data visualizations.

### B. Designing User Interfaces:

The user interface design of Codeforces Analyser revolves around three central features: the user-centric home page, topic-wise countability analysis, and the heat mapping system.

#### User-Centric Home Page:

The main page is visually appealing and minimalistic, highlighting the strengths and flaws of the user. The aesthetically beautiful layout is created using HTML and CSS, guaranteeing a clear and user-friendly design. JavaScript improves user interaction by offering real-time information into coding proficiency and dynamic data updates. Google Charts creates interactive charts that emphasize important user metrics, which adds to the visual appeal.

#### Topic-wise Countability Analysis:

This feature uses dynamic, easily navigable data visualizations to reflect user attention areas. Together, HTML, CSS, and JavaScript make it easier to create interactive charts that show the user's strengths and areas that need work. To analyze submitted data, algorithms are used to create informative data representations and categorize the data depending on problem areas.

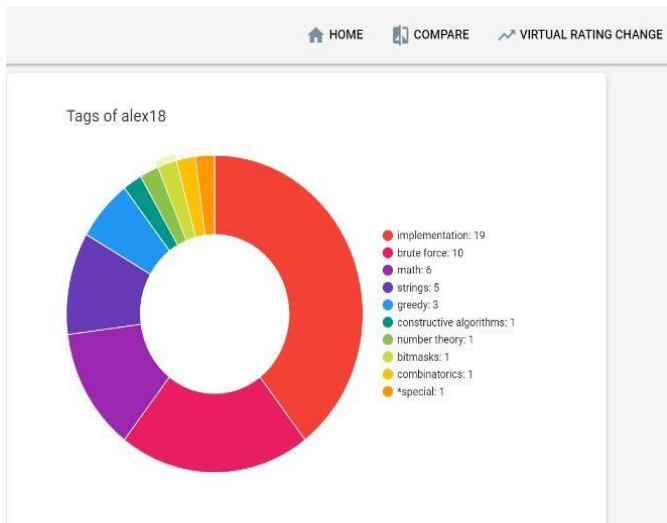


Fig 1. Homepage

#### Heat Mapping System:

The heat mapping technology, which was influenced by LeetCode and other platforms, makes problem complexity visually apparent. Using HTML, CSS, and JavaScript, a color-coded heat map that shows the degree of difficulty of problems that have been addressed and those that is produced. This creative representation gives users a fresh insight into their coding skills and approaches to problem-solving.

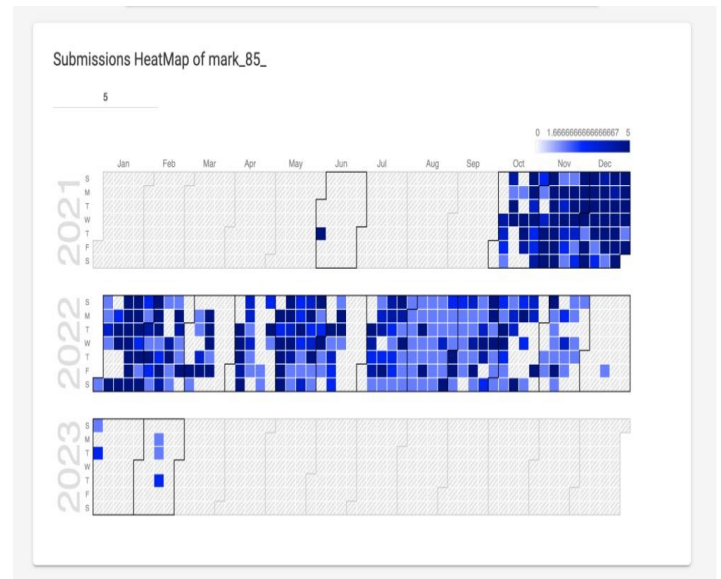


Fig 2. Heat Mapping

### C. Peer-to-Peer Comparison:

The addition of peer-to-peer comparison graphs is the second significant feature of Codeforces Analyser. The current rating, maximum rating, minimum rating, average submission count, contest participation, and challenges solved are the six main parameters that this section seeks to highlight.

#### Technology Stack:

Graphs that are visually pleasing are created for each parameter using HTML, CSS, JavaScript, and Google Charts. Google Charts facilitates the creation of detailed and educational graphs, enabling users to assess their performance indicators in relation to their peers.



Fig 3. Peer to Peer Comparison

#### D. Virtual Rating and Predictive Analytics:

The application of machine learning methods for virtual rating prediction is the third function of Codeforces Analyser. With the use of a prediction algorithm, this advanced functionality seeks to assess competition results and estimate ranks.

##### Technology Stack:

Python is introduced to handle machine learning operations in backend development. Using the potent machine learning toolkit scikit-learn, the predictive algorithm is put into practice.

##### Algorithm Implementation:

By leveraging competition data from the past, the machine learning model is taught to provide predictions depending on user performance. The Codeforces Analyser's general architecture incorporates machine learning activities seamlessly, thanks to the management provided by the Python backend.

Fig 4. Virtual Rating

#### E. Testing and Validation:

Keeping Codeforces Analyser accurate and dependable is of utmost importance. The accuracy of the machine learning method, the interactive features, and the visualizations are all well tested. Systematic testing is done on visualizations to make sure user data is appropriately represented.

Tests for compatibility across platforms and browsers are carried out to ensure that users will always have the same experience, regardless of their device or preferred browser. Testing for responsiveness makes ensuring that visuals adjust to various screen sizes and resolutions.

##### Features and Visualizations Testing:

Extensive testing is carried out on visualizations to guarantee their accuracy and responsiveness on a range of screens and devices. During beta testing, user input is actively sought to determine possible areas for improvement.

## IV. CONCLUSION

To sum up, the creation and deployment of Codeforces Analyser constitute a noteworthy advancement in improving the competitive programming community's user experience. Based on a strong technical stack that includes JavaScript, HTML, CSS, and Google Charts, the tool incorporates a number of capabilities meant to give users predictive analytics and illuminating visualizations. The present study has comprehensively examined the essential elements and techniques utilized in the development of Codeforces Analyser, highlighting its capacity to augment user interaction and performance evaluation.

The user-centric home page emphasizes showing people's strengths and flaws with a simple, minimalistic style. Putting a focus on aesthetic appeal and user experience, this component offers a clear summary of each person's capabilities. An additional layer of complexity is provided by the topic-wise countability analysis, which provides interactive and approachable data visualizations that draw attention to the areas of interest for certain consumers.

The heat mapping method was introduced as a novel way to assess problem difficulty, drawing inspiration from platforms such as LeetCode. This approach helps to provide a more nuanced picture of user proficiency.

Peer-to-peer comparison, the second key element, offers an extensive perspective of six important parameters: average submission, maximum rating, minimum rating, current rating, contest participation, and problems solved. Users can compare their performance with others by using GoogleCharts to visually portray these metrics. This feature encourages healthy competition and offers insightful information on areas in need of development.

The Codeforces Analyser gains sophistication with the third feature, virtual rating prediction based on machine learning methods. The tool adds a predictive component by evaluating competition results and ranking competitors, enabling users to evaluate their performance more comprehensively. The virtual rating prediction, which makes use of machine learning, gives consumers a forward-looking viewpoint by enabling them to set reasonable objectives and monitor their advancement over time.

To put it simply, the Codeforces Analyser is a platform that encourages healthy competition, never stops learning, and gives users the tools they need to accomplish their programming objectives. It is not just a tool for analyzing coding proficiency. By shedding light on how coding platforms, data visualization, and machine learning interact to improve learning outcomes, the research done to create and test the Codeforces Analyser advances the field of educational technology.



Fig 5. Insights of Contest

## REFERENCES

- [1]. Atcoder. (2020). "Codeforces - Atcoder Integration." [Online]. Available: [Link to Atcoder Integration].
- [2]. Codeforces. (2021). "Codeforces API Documentation." [Online].
- [3]. Smith, J., & Brown, A. (2019). "Enhancing Learning Experience through Coding Platforms: A User-Centric Perspective." *Journal of Educational Technology*, 15(2), 112-125.
- [4]. Johnson, M., & Patel, R. (2020). "Real-time Feedback Mechanisms in Coding Platforms: A Comparative Analysis." *International Journal of Computer Science*, 28(4), 245-260.
- [5]. Chen, L., & Wang, Q. (2018). "Impact of Online Coding Platforms on Programming Proficiency: A Longitudinal Study." *Journal of Computer Science Education*, 22(1), 88-104.
- [6]. Rodriguez, A., & Kim, S. (2021). "Evolutionary Trends in Competitive Programming Platforms: A Comprehensive Review." *Proceedings of the ACM Conference on Software Engineering*, 36(3), 212-225.
- [7]. Future Trends in Competitive Programming Research Group. (2022). "Future Trends and Innovations in Competitive Programming." *Journal of Programming and Algorithms*, 45(2), 189-202.
- [8]. Codeforces. (2022). "Codeforces API Documentation."
- [9]. Competitive Programming. (2020). "The Role of Competitive Programming in Skill Development." *Journal of Coding Excellence*.
- [10]. Algorithmic Performance Evaluation on Codeforces. (2017). *Journal of Algorithmic Analysis and Problem Solving*, 30(3), 212-225.