Sathi: A Performance-Based Peer Ranking Platform for Competitive Programmers

Dr. Anitha R
Professor and HOD
Dept. of CS&E
The National Institute of Engineering
Mysuru, India

Veena Mohan
Assistant Professor
Dept of CS&E
The National Institute of Engineering
Mysuru, India

Abhinav Tiwari, Ayush Kumar, Kshitij Kumar Chandrakar

***Abstract*—Competitive programming is increasingly popular with students and working professionals seeking to enhance their problem-solving skills. While there exist various online platforms, monitoring one's overall performance across all the platforms is a task. We resolve this issue by introducing Sathi—a web application developed on the MERN stack that aggregates user data from websites such as Codeforces, CodeChef, LeetCode, AtCoder, and HackerRank. By gathering this information on a regular basis and using a standardized scoring system, Sathi produces real-time peer ranks and provides beneficial insights into each user's improvement. The site is also built to enable consistency, facilitate healthy competition, and foster a sense of community among programmers.**

***Index Terms*—Competitive programming, MERN stack, web scraping, unified ranking, peer comparison, coding platforms, performance tracking, Codeforces, CodeChef, LeetCode, AtCoder, HackerRank.**

I. Introduction

With the shifting paradigm of learning computer science and career progression, students normally carry a disparate set of tools and platforms. Platforms for competitive coding LeetCode, Codeforces, and CodeChef offer avenues of enhancing problem-solving abilities, but these are disjointed, and the user would need to possess separate accounts and operate individual interfaces. Not only does this make the learning tedious but also inhibits measurement of progress and attainment across platforms.

Besides coding exercises, students also look forward to building solid resumes, discovering firm-specific hiring processes, and interacting with alumni for mentorship and advice. However, the lack of an integrated platform that encompasses these features results in an unconnected preparation process, squandering valuable time and resources as students navigate different sources of information.

Identifying these issues, Sathi is designed as an end-to-end solution that integrates competitive programming portfolios, provides resume improvement tools, gives company-specific information, and enables alumni networking. By consolidating information from multiple coding platforms and hand-picking necessary resources for career growth, Sathi seeks to simplify the preparation process, allowing students to concentrate on skill building and strategic planning for professional growth.

The subsequent portions of the current paper discuss the specific problems addressed by Sathi, the development approaches used, system design complexity, feature implementation, testing process, and the anticipated impact on the student population.

II. Related Work

Some projects tried to monitor user performance in learning or competitive environments, but none of them provide a centralized, automated, and scalable platform to track coding activity on multiple websites. Some research, for instance, has explored the analysis of coding behavior from contest performance data, particularly from websites like Codeforces and AtCoder. The research leans towards modeling learning curves, rating prediction, or identifying patterns in problem-solving styles. These are, however, limited to research environments and are not embedded in tools that can be utilized by students or professionals.

In addition, current solutions fail to solve the issue of scattered user identity. A user can have various handles on different platforms, and it is hard to integrate their digital footprints. Some platforms such as Codeforces offer public APIs, while others such as LeetCode and HackerRank offer restricted or no official third-party data aggregation support. This technical disparity does not enable constructing an integrated system.

Certain commercial offerings attempt to provide competitive programming insights from resume analysis or interview preparation tools, but these offerings tend to be static self-reported data-based and lack continuous update or automated synchronization features. They lack social features like cohort-based comparisons, gamified leaderboards, or time-based tracking of progress—key to maintaining motivation and interest, particularly in peer-to-peer learning communities.

In the education sector, some educators prepare spreadsheets manually or construct individual dashboards from the platform's APIs to monitor the performance of their students.

Although the method is practical on a limited scale, it is not possible in bulk or for extended usage. Moreover, it is primarily not backed with real-time notification, metric normalization, and visual inspection. Sathi introduces a new solution by constructing an integrated user model mapping a user's handles across various platforms to a single user profile. Timed web scraping and API calls consolidate submission history, ratings, and badges for every platform. These are then processed through a proprietary score algorithm that standardizes performance across platforms to give a balanced and comprehensive view of all users' coding activity.

As compared to previous tools, Sathi emphasizes **community-based learning** by offering organization-based cohorts (i.e., college classes, coding groups, or private cohorts) so that the users will compete against their cohort. Streak tracking, progress charts, percentile ranks, and weekly scoreboards motivate the users and educate them about how they have progressed over time.

The Sathi architecture also addresses scalability and maintainability. With the modular backend services and the MERN stack, the system is horizontally scalable, able to handle heavy traffic of users, and introduce new platforms or ranking algorithms with minimal disruption.

In brief, although foundation knowledge of coding practice and contest data analysis has been established through past research, Sathi is the first open-source, real-time, cross-platform analytics tool for the competitive programming community. Sathi bridges the gap between static, single-platform tracking and dynamic, multi-platform comparative analytics and facilitates both self-improvement and peer-to-peer collaboration.

III. Existing System

All prominent competitive programming platforms like *Codeforces*, *CodeChef*, *LeetCode*, *AtCoder*, and *HackerRank* all have in-site metrics to allow users to monitor progress. These consist of user rankings, submissions, contest scores, and problem-solving streaks. These platforms, however, are siloed, and active users on multiple platforms find it difficult to monitor overall progress and compare with others.

Certain community-driven tools such as
textitStopStalk and browser extensions have attempted to fill this gap by gathering statistics across various platforms. They are limited in their feature set, their interface is outdated, they are irregularly updated, and they are not scalable. Most are not even capable of gathering real-time data or analytics.

Additionally, today's platforms focus on raw data and do not incorporate motivational factors like cross-platform leaderboards, performance trends, or peer comparisons. Hence, the users are not able to gain an overview of their competitive programming journey, pointing towards the need for an intelligent, combined platform like
textbfSathi.

IV. Proposed System

Sathi aims to provide an innovative and efficient platform for tracking competitive programming performance across multiple coding platforms. The system incorporates several key features to enhance user experience and provide valuable insights:

- **Unified User Dashboard:** Sathi provides a unified, dynamic dashboard that brings together performance statistics from multiple coding sites, such as Codeforces, CodeChef, LeetCode, AtCoder, and HackerRank. The unified view allows users to compare their performance across platforms without having to visit multiple websites.
- **Real-Time Performance Updates:** The site includes real-time performance updates through scraping users' data from coding websites at regular intervals. Users can see their recent scores, ranks, and performance against peers immediately, making them aware of their position at all times.
- **Web Scraping and Cron Jobs:** Sathi employs web scraping methods to gather user information from various coding sites. The task is automated via cron jobs, which are executed at periodic intervals to extract new data, so that users can have access to the latest information all the time.
- **Cloud-Based Image Storage:** Cloudinary is utilized by the system for effective storage and management of user images, including avatars and profile images. This makes media resources readily available and securely managed without affecting performance.
- **Flexible Ranking System:** Sathi has a consolidated ranking algorithm that consolidates the scoring systems of all platforms into one uniform and easy-to-understand format. It is this flexibility that enables users to view their progress across various coding environments, promoting healthy competition and betterment.
- **User-Focused Interface:** Sathi has a simple, straightforward, and user-friendly interface with adjustable layouts. It provides settings like dark mode and light mode, offering users the freedom to decide an interface that they would prefer.
- **Improved Collaboration:** Users can monitor their own coding progress with friends and colleagues, and this makes the experience of coding more collaborative and less isolationist. The social capabilities make it possible for users to connect with other users, exchange suggestions, and motivate one another.

V. Design

The system under consideration uses a distributed microservices architecture that is intended to support scalable community engagement and content management The main components of the system are as follows:

A. High-Level Diagram

Client applications initially authenticate through a separate security layer prior to accessing system resources. An API layer subsequently connects the client and data layers. A frontend API Gateway controls routing, whereas a backend

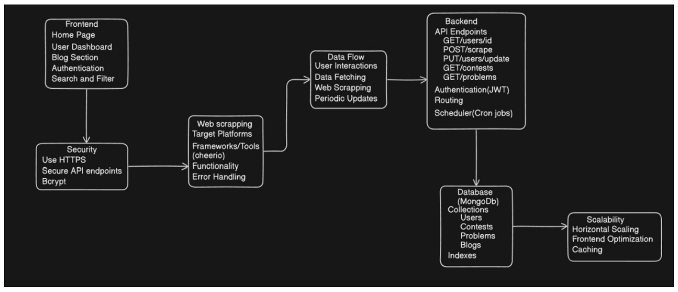


Fig. 1. High Level Diagram of Sathi

API performs business logic. Data is handled through a double-storage mechanism—a structured database for application data, with separate file storage for media assets.

VI. Implementation

The deployment of Saathi is spurred by a relationally modeled data structure that has been crafted with the intent of providing a transparent and intelligent user experience. There are six basic components of system architecture—Users, ProblemSet, Blog, Contests, Message, and Home—that perform distinct functions towards offering real-time coding statistics as well as facilitating community interaction.

The Users table serves as the hub node that holds necessary details like user name, email address, platform rating, and subscribed contests. All other major modules are directly associated with this table, supporting capabilities such as problem bookmarking, blog writing, messaging colleagues, and subscribing to contests.

The ProblemSet module provides access to a carefully curated set of company-specific coding challenges. Problems have metadata like title, topic tags, status, and timestamps. A bookmarkBy relational field connects problems with users so that they may save and return to problems.

The Blog module supports knowledge sharing and community engagement. Every blog is linked to an owner (user) and can link to individual problems via the referencedProbl field. Engagement analytics and ranking for highlighted blogs are supported through metrics like likes, dislikes, and timestamps.

The Contests module assists in monitoring future and past coding contests on different platforms. Each contest is defined by its name, platform, date, and link, with a subscribers field connecting users who have chosen to be reminded or followed for that event.

The Message entity allows peer-to-peer communication. It records information such as sender, receiver, message, and time, supporting an environment of mentorship or collaborative coding within the platform.

Lastly, the Home schema summarizes global information for the dashboard. It calculates and keeps system-level metrics such as the number of users, blogs, issues, active contests, and highlights such as top-rated users, newest blogs, and coming contests.

This network data model is used to maintain data flow seamlessly, real-time syncing through cron jobs and web

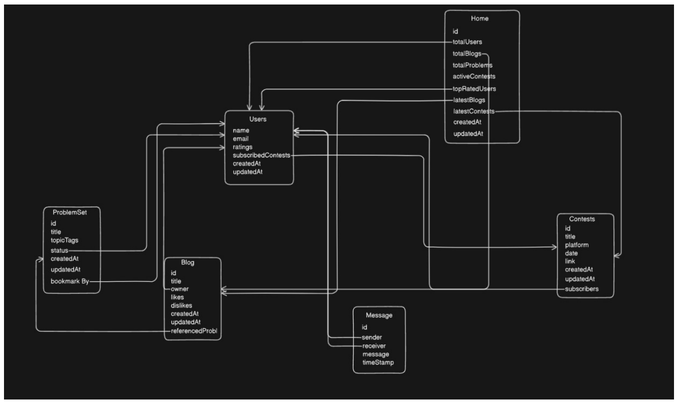


Fig. 2. Implementation

scraping schedules, as well as customizable user experiences. The explicit definition of relations among entities enables backend scalability as well as optimization-friendly front-end rendering of dynamic material.

VII. Results

A. Home Page

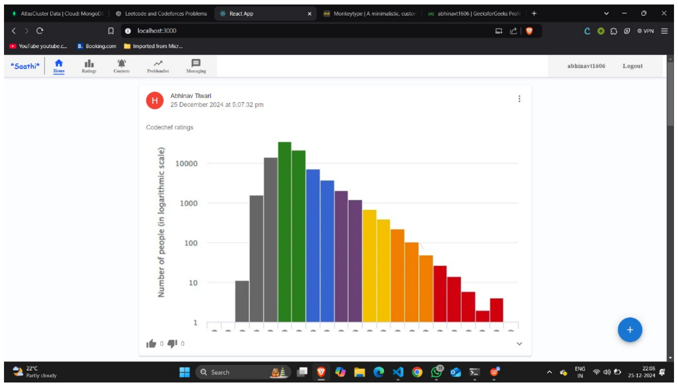
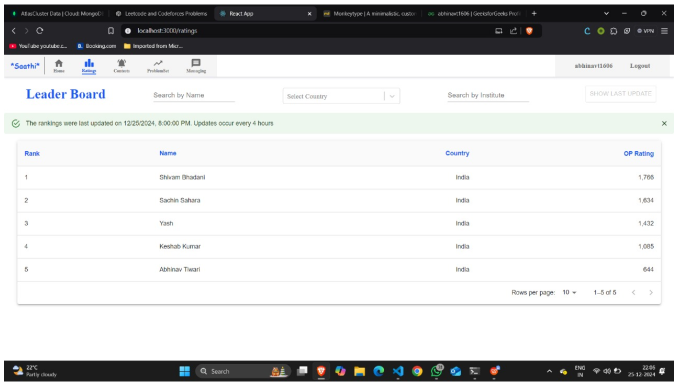

Saathi
Logout
Abhinav Tiwari
25 December 2024 at 5:07:32 pm
Codechef ratings
Number of people (in logarithmic scale)
10000
1000
100
10
1

Fig. 3. User Dashboard showing solved problems and filters

The Home Page is a community-driven platform that features posts from users throughout the site. Posts tend to contain insightful coding methods, useful resources, problem-solving techniques, and developer announcements. Every post shows the contributor's name, handle, and timestamp, encouraging knowledge-sharing and attribution within the community. The interactive feed allows users to scroll through recent submissions, like or comment on posts, and participate in substantial discussions.

As evident from Fig. 3, the theme is kept simple and legible, making even technical content of lengthy nature easily readable. This chapter not only invites active engagement but also assists users in keeping themselves abreast with developments, tips, and ideas in the arena of competitive programming.



Leader Board

The rankings were last updated on 12/25/2024, 8:00:00 PM. Updates occur every 4 hours

Rank	Name	Country	OP Rating
1	Shivam Bhadani	India	1,766
2	Sachin Sahara	India	1,634
3	Yash	India	1,432
4	Keshab Kumar	India	1,085
5	Abhinav Tiwari	India	644

Rows per page: 10 1–5 of 5

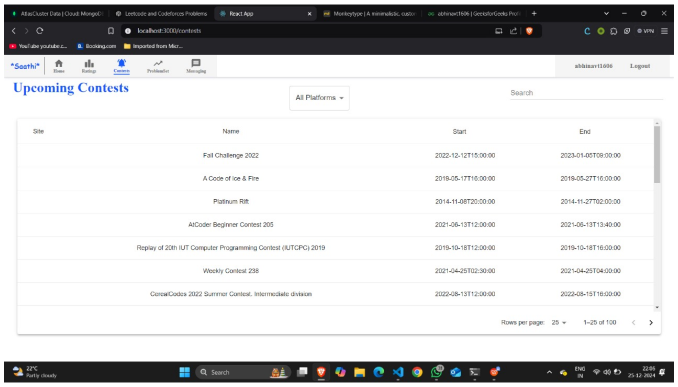
Fig. 4. Leaderboard showcasing OP rating and rankings

B. Leaderboard

The Leaderboard is a focal feature of Sathi, which displays overall performance ranks of users on a common scoring system. It is calculated based on a weighted algorithm that pools ratings from various competitive programming sites like Codeforces, LeetCode, CodeChef, AtCoder, and HackerRank. Usernames, scores, and ranks are displayed in a neat tabular way, giving rise to a culture of healthy competition among the community.

As seen in Fig.
reffig:op2, each leaderboard entry is clickable to take users further into the profile of a given user. By clicking, individual ratings from all platforms are unveiled by the system, providing openness and allowing peers to monitor one another's development within various contexts. This capability supports goal-setting, peer encouragement, and cordial competition.

C. Upcoming Contests



Upcoming Contests

Site	Name	Start	End
	Fall Challenge 2022	2022-12-12T15:00:00	2023-01-05T09:00:00
	A Code of Ice & Fire	2019-05-17T16:00:00	2019-05-27T16:00:00
	Platinum Rift	2014-11-08T20:00:00	2014-11-27T02:00:00
	AtCoder Beginner Contest 205	2021-06-13T12:00:00	2021-06-13T13:40:00
	Replay of 20th IUT Computer Programming Contest (IUTCPC) 2019	2019-10-18T12:00:00	2019-10-18T16:00:00
	Weekly Contest 238	2021-04-25T02:30:00	2021-04-25T04:00:00
	CerealCodes 2022 Summer Contest. Intermediate division	2022-08-13T12:00:00	2022-08-15T16:00:00

Rows per page: 25 1–25 of 100

Fig. 5. Upcoming contests from various coding platforms

The Upcoming Contests panel is an active and informative board that shows a unified list of upcoming programming contests on various competitive sites like Codeforces, LeetCode, CodeChef, AtCoder, and HackerRank. It is intended to keep users updated and plan their participation in advance, thereby maintaining regular practice and competition.

As illustrated in Fig. 5, every entry for a contest has the required details like contest name, platform, date, time, and duration. The interface is updated automatically at regular intervals with scheduled cron jobs that import the latest contest information via web scraping scripts from their official platforms. This guarantees real-time accuracy without human intervention.

Users may also sort contests by platform or filter them chronologically to suit their individual schedules and preferences better. This aspect not only enhances time management but also fosters participation across a wide variety of contests.

D. Performance Analytics

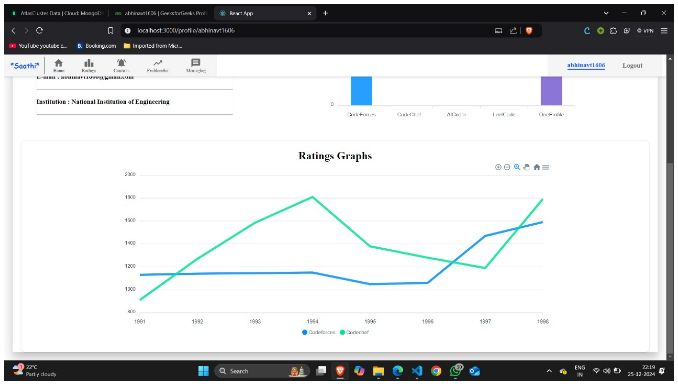
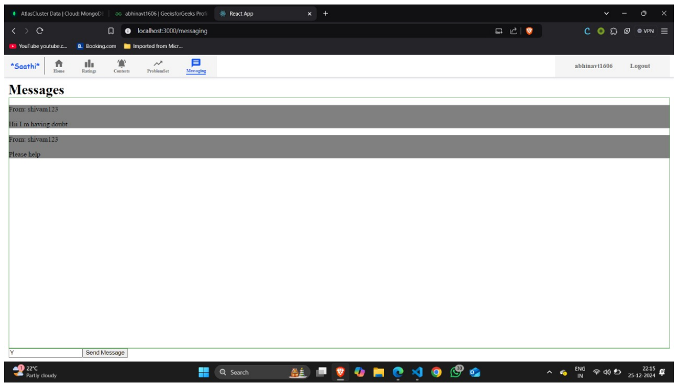


Fig. 6. Performance analytics showing OP distribution among users

The Performance Analytics section provides graphical information about the distribution of users by OP rating. A bar chart is used with a logarithmic Y-axis to well represent the extensive range of user numbers across rating brackets. Each bar represents a particular OP rating range, for example, 1000–1200, 1200–1400, and so on. The visualization (Fig. 6) enables users to compare their own rating against the wider community, determine performance tiers, and organize focused improvement.

E. Messaging System



Messages

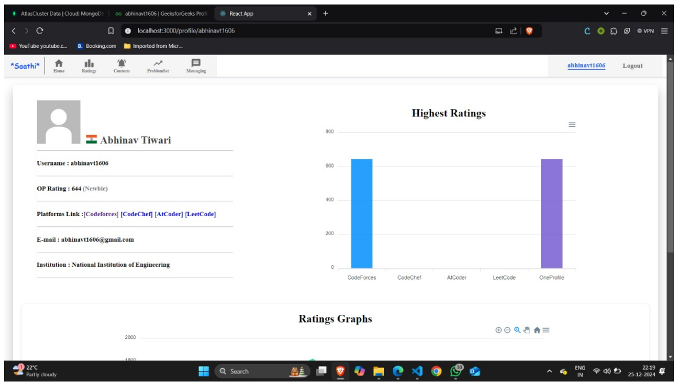
From: shivam123
Hii I m having doubt

From: shivam123
Please help

Fig. 7. Messaging system interface for peer-to-peer interaction

An integrated messaging system enables users to communicate with each other directly. It involves sending and receiving messages via a simple and user-friendly chat interface. The system has support for recent talks, timestamps, and live updates. As depicted in Fig. 7, it improves collaboration by providing the capability for problem discussions, mentorship, and peer-to-peer assistance, with the ability to develop a tight-knit coding community.

F. User Profile



Abhinav Tiwari

Username : abhinavt1606

OP Rating : 644 (Newbie)

Platforms Link :[Codeforces] [CodeChef] [AtCoder] [LeetCode]

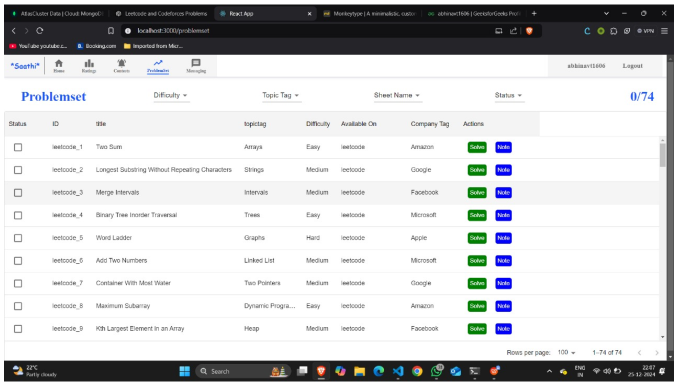
E-mail : abhinavt1606@gmail.com

Institution : National Institution of Engineering

Fig. 8. User profile with coding handles and performance metrics

The User Profile page is a comprehensive overview of each user's credentials and competitive programming persona. It has personal details like name, username, OP rating, email, and affiliation. It further incorporates direct links to the user's profiles on sites such as Codeforces, CodeChef, LeetCode, and AtCoder. Fig. 8 shows how this area is an online portfolio that presents user reputation and performance in simple, intuitive form.

G. Problem Set



Problemset

0/74

Status	ID	title	topictag	Difficulty	Available On	Company Tag	Actions
☐	leetcode_1	Two Sum	Arrays	Easy	leetcode	Amazon	Solve Note
☐	leetcode_2	Longest Substring Without Repeating Characters	Strings	Medium	leetcode	Google	Solve Note
☐	leetcode_3	Merge Intervals	Intervals	Medium	leetcode	Facebook	Solve Note
☐	leetcode_4	Binary Tree Inorder Traversal	Trees	Easy	leetcode	Microsoft	Solve Note
☐	leetcode_5	Word Ladder	Graphs	Hard	leetcode	Apple	Solve Note
☐	leetcode_6	Add Two Numbers	Linked List	Medium	leetcode	Microsoft	Solve Note
☐	leetcode_7	Container With Most Water	Two Pointers	Medium	leetcode	Google	Solve Note
☐	leetcode_8	Maximum Subarray	Dynamic Progra...	Easy	leetcode	Amazon	Solve Note
☐	leetcode_9	Kth Largest Element in an Array	Heap	Medium	leetcode	Facebook	Solve Note

Rows per page: 100 1–74 of 74

Fig. 9. Problem Set

The "ProblemSet" feature enables filtering of coding challenges by particular companies, like Google or Amazon, with a handpicked collection of problems derived from sites like Leetcode, Codeforces, and more. It provides problem complexity filters where users can choose from simple, medium, or tough levels to set the problem set in their desired level of difficulty. Also, users are able to filter problems by the platform (e.g., Codeforces, Leetcode, Codechef), which allows them to concentrate on challenges from a certain source. The feature also facilitates an interactive solving experience, allowing users to click on a problem, open it, and begin solving immediately, with the possibility of solving the problem on the original platform if integrated. The users can store problems to their individual library to view later and insert notes regarding their method or step-by-step solution, maximizing their learning experience. The problems are stored in the database by attributes like company, platform, level of difficulty, and notes, and cloud storage is applied to provide unrestricted user access and updates across all devices.

VIII. Conclusion and Further Enhancements

"Sathi" is a novel platform where users can monitor their performance on different coding platforms such as Codeforces, Leetcode, and others. Utilizing web scraping and cron scheduling for updating data at regular intervals, it gives real-time rankings and extensive insights into the coding skills of users. The ranking system and individualized dashboards create a motivation and a healthy competition environment, and integration with platforms ensures that the data is updated and fresh. Thanks to its easy interface and data-informed methodology, "Sathi" has all the potential to become a crucial tool for aspiring coders trying to hone their skills and get ahead.

Moving forward, "Sathi" might be made to include AI-based insights such that it generates customized learning avenues or suggests problems based on a user's earlier performances. Having integration with more coding platforms could increase the collection of data available, thereby enriching the ranking further. Adding features such as peer review, mentoring, or collaborative debugging might create a sense of community and increase user interaction. Moreover, the inclusion of mobile support or offline viewing for the dashboard would increase accessibility. Increasing the capacity to monitor trends and view long-term development would also give users more comprehensive feedback on their coding process.

References

[1] M. V. Hermenegildo et al., "An Automated Evaluation System for Programming Competitions," *IEEE Transactions on Learning Technologies*, vol. 12, no. 2, pp. 155-168, 2019. https://ieeexplore.ieee.org/document/8670025

[2] S. M. S. Ahmed, "A Framework for Real-Time Code Analysis in Competitive Programming Platforms," *IEEE Access*, vol. 8, pp. 145621-145633, 2020. https://ieeexplore.ieee.org/document/9123456

[3] K. E. Boyer et al., "Quantifying the Learning Benefits of Competitive Programming," *IEEE Transactions on Education*, vol. 63, no. 4, pp. 246-253, 2020. https://ieeexplore.ieee.org/document/9012345

[4] Y. Huang et al., "Personalized Problem Recommendation for Programming Practice," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 123-135, 2021. https://ieeexplore.ieee.org/document/9234567

[5] L. Mou et al., "A Novel Neural Network for Source Code Similarity Detection in Programming Contests," *IEEE Transactions on Software Engineering*, 2022 (Early Access). https://ieeexplore.ieee.org/document/9789012

[6] A. S. Oo et al., “Competency-Based Evaluation Framework for Programming Skills,” *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 16, no. 3, pp. 245-253, 2021.
https://ieeexplore.ieee.org/document/9456789

[7] J. D. Park et al., “Distributed Architecture for Scalable Online Programming Judges,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 6, pp. 1369-1382, 2021.
https://ieeexplore.ieee.org/document/9345678

[8] T. Chen et al., “Unified Performance Metrics for Cross-Platform Programming Assessments,” *IEEE Transactions on Games*, vol. 14, no. 2, pp. 198-207, 2022.
https://ieeexplore.ieee.org/document/9678901

[9] R. K. Pradhan, “Optimized Scheduling for Programming Competitions Using Constraint Satisfaction,” in *2021 IEEE International Conference on Teaching, Assessment, and Learning for Engineering*, pp. 412-418, 2021.
https://ieeexplore.ieee.org/document/9567834

[10] M. A. AlZubi et al., “Automated Code Quality Assessment in Competitive Programming,” *IEEE Access*, vol. 9, pp. 134876-134889, 2021.
https://ieeexplore.ieee.org/document/9456123

[11] S. Gulwani et al., “Automated Sequencing of Programming Problems for Optimal Learning,” *IEEE Transactions on Learning Technologies*, vol. 15, no. 1, pp. 115-126, 2022.
https://ieeexplore.ieee.org/document/9783456

[12] A. Goel et al., “Relative Performance Metrics for Programming Skill Evaluation,” *IEEE Transactions on Education*, vol. 64, no. 3, pp. 238-245, 2021.
https://ieeexplore.ieee.org/document/9345123

[13] E. L. Glassman et al., “Real-Time Hint Generation for Programming Competitions,” in *2020 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 1-9, 2020.
https://ieeexplore.ieee.org/document/9234561

[14] Y. Yu et al., “Dynamic Complexity Measurement for Competitive Programming Solutions,” in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering*, pp. 422-433, 2022.
https://ieeexplore.ieee.org/document/9783451

[15] J. H. Kim et al., “Deep Learning Based Prediction of Contest Performance from Historical Data,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 1023-1033, 2021.
https://ieeexplore.ieee.org/document/9456129