CS349 - Database and Information Systems Lab

CodeForces Revamped

Ananya Rao (22B0980)
Deeksha Dhiwakar (22B0988)
Sharvanee Sonawane (22B0943)
Sanchana Arun (22B1034)

April 2024

1 Project Idea

The idea for our project is to build a clone of the popular website CodeForces. It is a competitive programming platform that hosts regular contests and provides a vast archive of algorithmic problems for users to practice and enhance their coding skills. It features a rating system that categorizes participants from Newbie to Legendary Grandmaster, fostering a competitive learning environment. Users can engage in virtual contests, solve problems with editorial support, and participate in community discussions through blogs. It also offers a "Gym" for training with custom problem sets, making it an essential resource for aspiring competitive programmers.

We intend to replicate the essential functionalities of CodeForces, and then extend it to support several novel features such as an inbuilt plagiarism checker, an option to generate model solutions using AI when creating a problemset, a version control system that enables users to access previous iterations of their code (similar to the timeline feature of VS Code), isolated sandbox environments for a better coding experience, an inbuilt coding IDE featuring intellisense and autocomplete for multiple languages and many more.

We also plan to revamp the user interface by improving the design and layout and making it more intuitive and user friendly.

2 Progress Made

ER Diagram

We have formulated the below ER diagram that our database schema will follow (implemented using PostgreSQL).

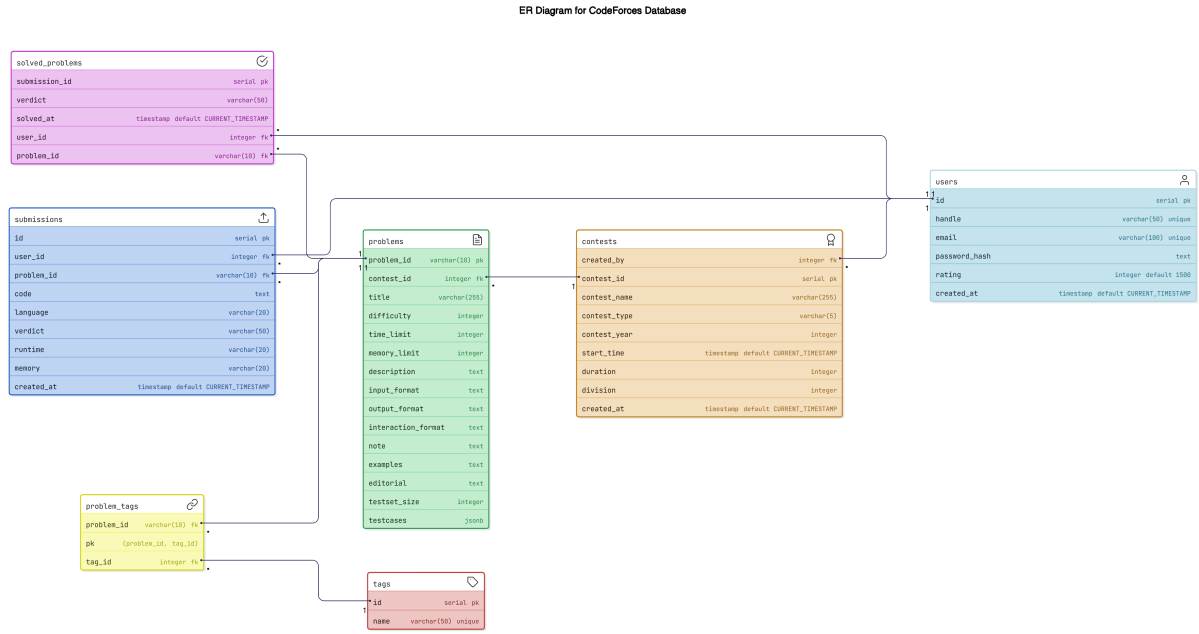


Figure 1: ER Diagram

Dataset

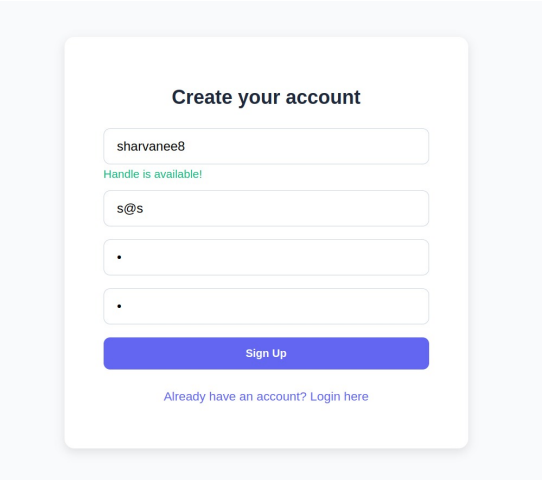
In this project, for problems and contests we use the publicly available dataset from this link. It contains 10024 rows of data, where each row has the details of a CodeForces problem such as the problem ID, contest ID, difficulty, tags, description, testcases etc.

To populate the `users` table, we generate fake user data (including email ID, username, location, joining date etc) using a Python script.

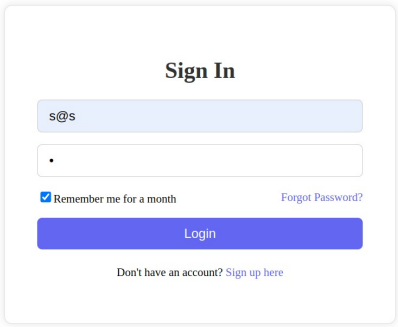
For submission data, we use the publicly available dataset from this link. It has been generated based on actual user submissions till the end of 2024, using aliases for all the users to protect their anonymity. It contains several details of the submissions such as the code, verdict, runtime, programming language etc.

Signup/Login/Logout

The pages in focus have been implemented. We aim to add *Google authenticator* as well to facilitate the signup/login process for users who already have an email id registered with *Google*



(a) Sign-up



(b) Login

Once the user has registered (or logged in), they are redirected to the dashboard page

Dashboard

Following is an image of the dashboard. It has a Navigation bar at the top which has buttons for important pages like `Contests`, `Problem-Set`, `Profile` etc.

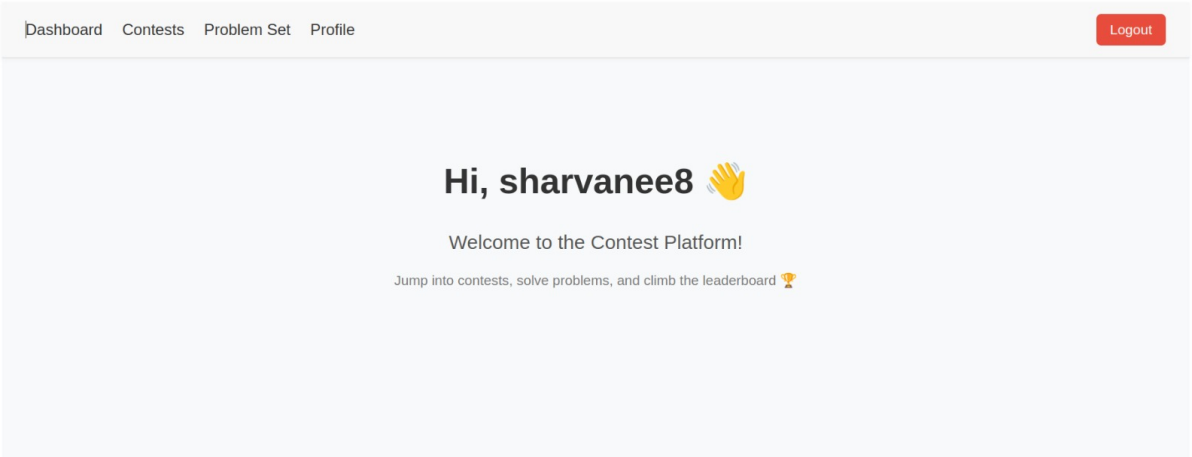


Figure 3: Dashboard

Contests

Here we show the `Contests` page which has a list of all active and inactive contests. The name of the contests are hyperlinks to the contest page. The contest list shows important details like start-time, and duration of individual contests

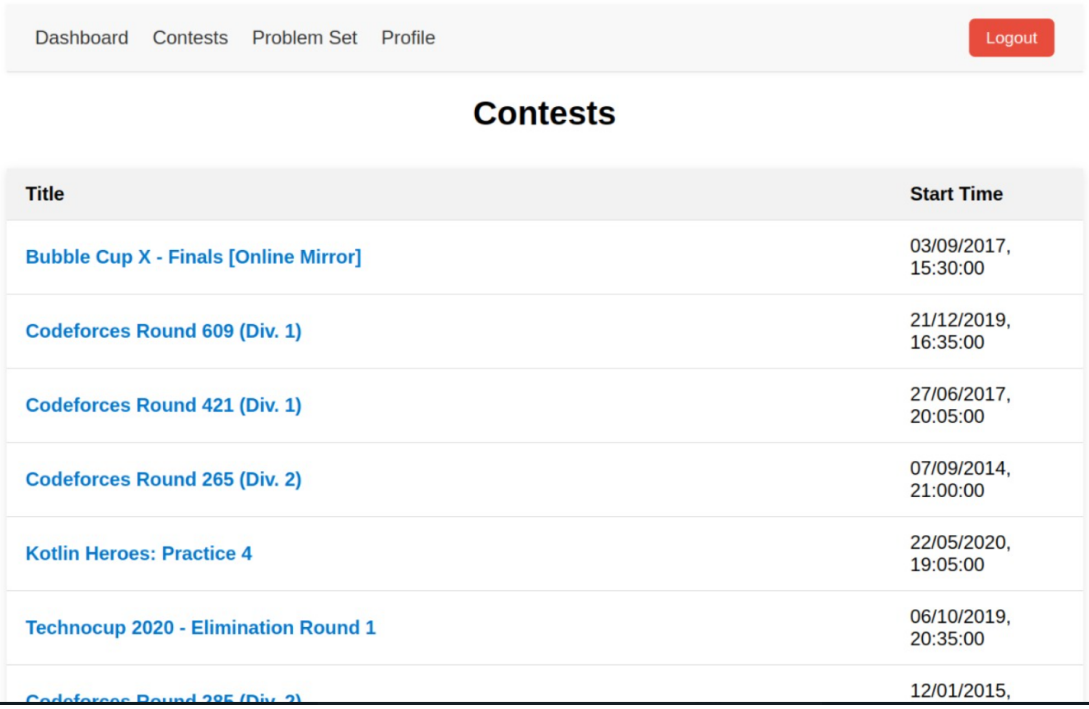


Figure 4: Contest List

Following is the page for a contest. It shows the list of all the problems which are a part of the contest and has hyperlinks to all of them.

Future Modifications: We aim to make different interfaces for active and inactive contests with active contest page being more interactive, providing live score-tables, rankings with other relevant features. Similarly, in inactive contests, we aim to provide details about the contest, problems' difficulties and hints to solve them.

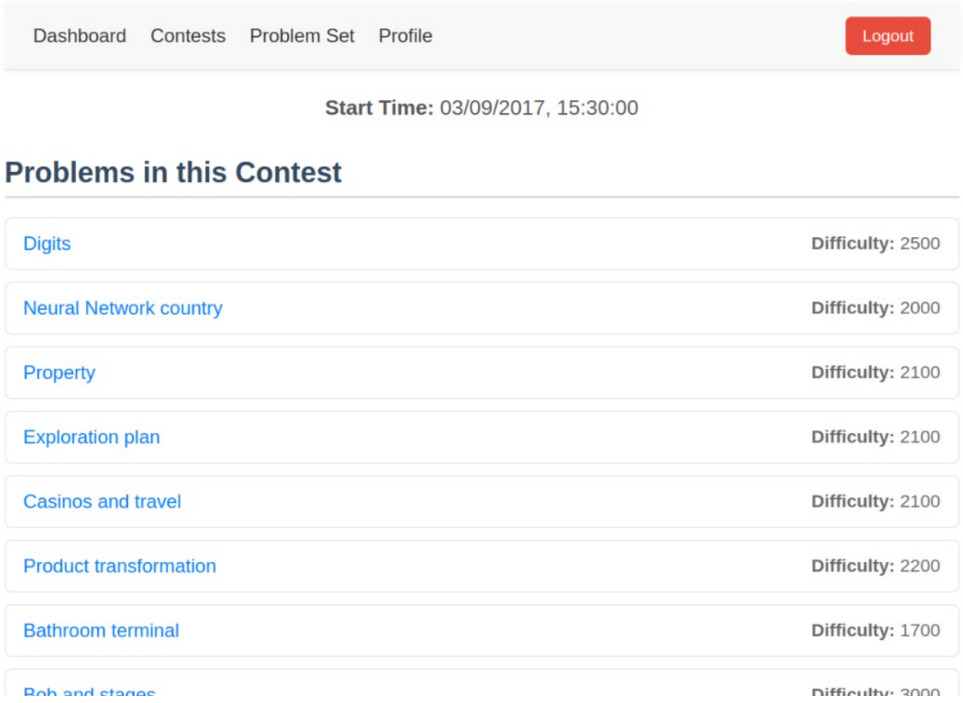


Figure 5: Contest Page

Problems

The `Problem-Set` page displays a list of all problems ever inserted in the database. It especially displays the problem title, difficulty and tags for coders to choose a question of their choice and need.

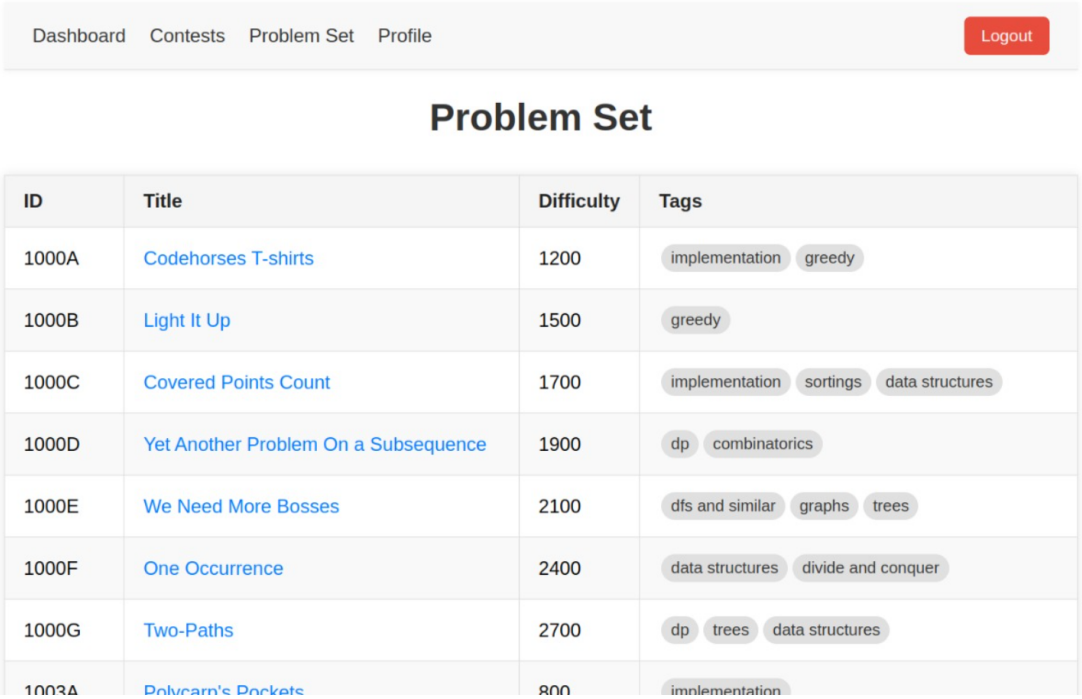


Figure 6: Problem-Set

The hyperlink to any Problem takes us to the `Problem` page, which display details like problem description, input/output format, tags, extra notes and any relevant link or information required to help users.

Future Modifications: We aim to make two different pages to show inactive problems and active problems (those which are included in active contests); each with slightly different but relevant characteristics.

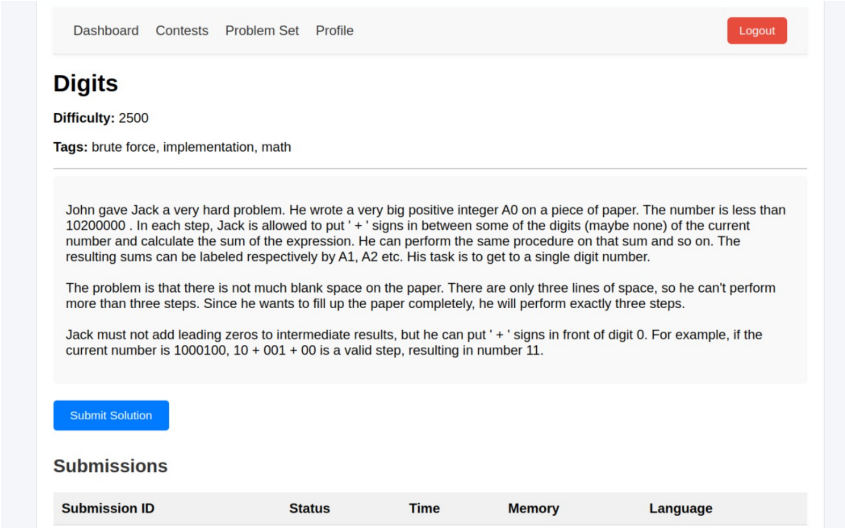
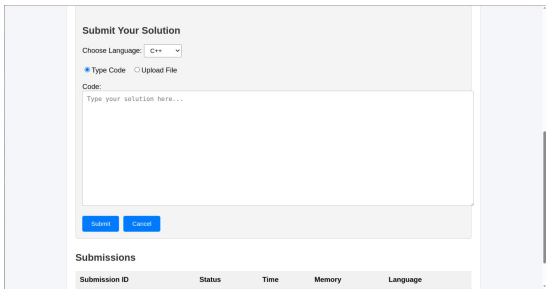


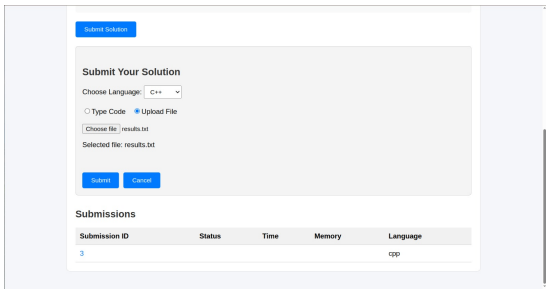
Figure 7: Problem Page

We have also added a submission feature in each problems page, for users to submit their solution

through either a valid file or directly through an editor, and a list of submissions for easy tracking of submissions.



(a) Submit code



(b) Submit file

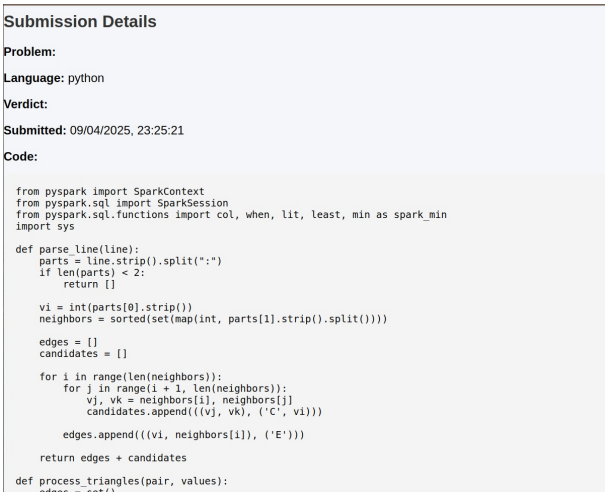


Figure 9: Submission details

Future Modifications: We plan to implement Sandboxing to enable a safer and more convenient environment for users to run their programs. We also aim to add built-in compilers in the Sandbox container for direct compilation and testing.

Profile

The profile page shows personal information (handle, rating, problems solved etc.) of a user along with a hyper link for adding contests.

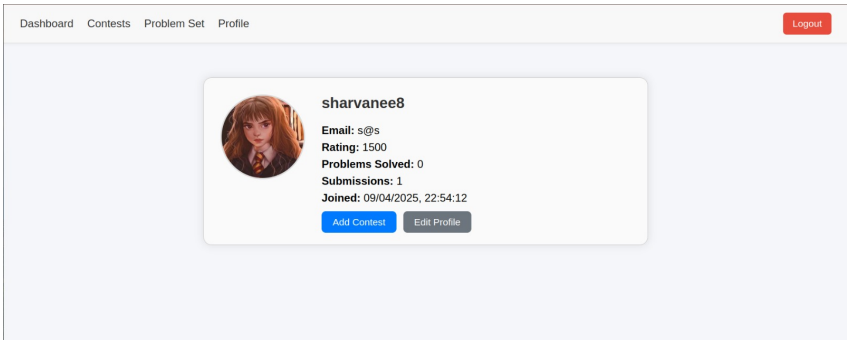


Figure 10: Profile

Future Modifications: We aim to add activity calendar and detailed performance report for users to track their progress and activities on our platform. We also plan to make the rating system more

dynamic and use it to grant various privileges to users (like adding contests, making problems and organizing hackathons)

2.1 Add-Contest

This page allows a user to organize a contest on our platform. The number and description of problems depend on the creator and there is a dynamic support for adding as many problems as required.

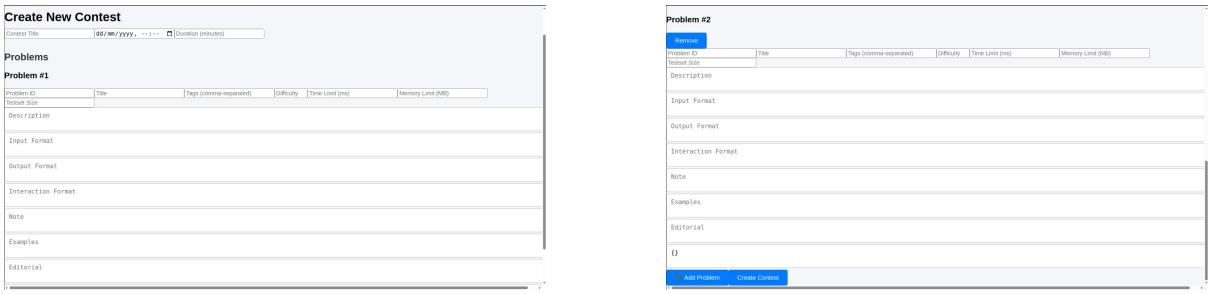


Figure 11: Add-Contest

3 Plan for Final Submission

The remaining tasks to be completed by our final submission are listed out below:

- Inbuilt plagiarism and AI code detector to automatically flag unethical submissions and restrict users with multiple offences.

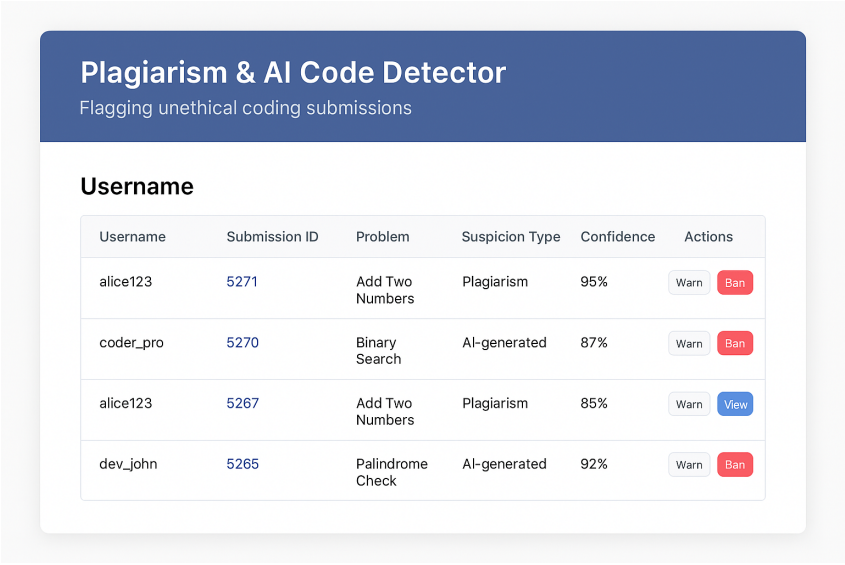


Figure 12: Plagiarism Detector

- Improved inbuilt coding IDE that supports features inspired from VS Code such as autocomplete and intellisense.

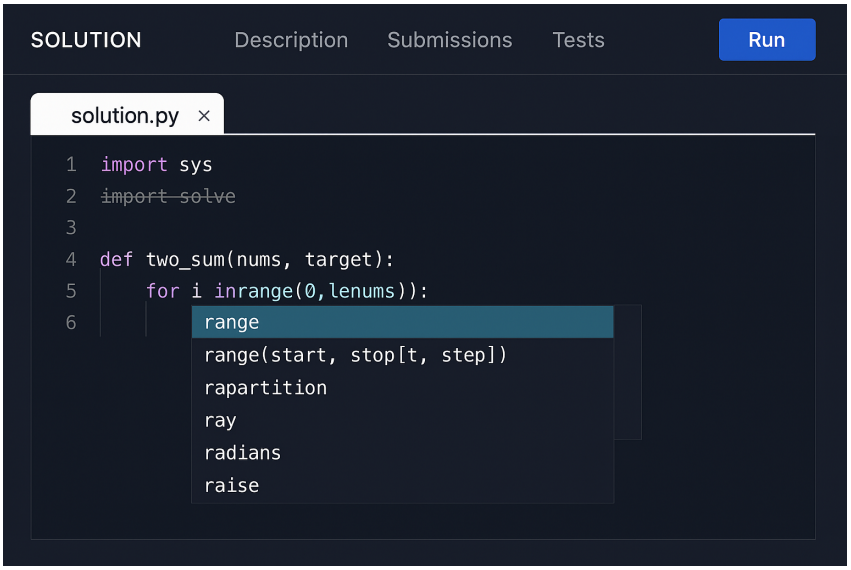


Figure 13: Autocomplete feature

- Version control feature that allows users to access previous iterations of their code (inspired from VS Code timeline feature).

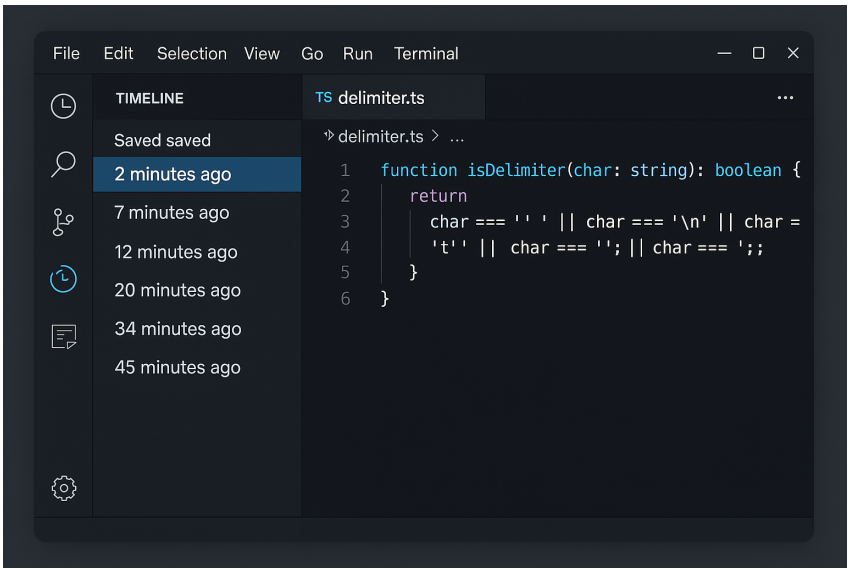
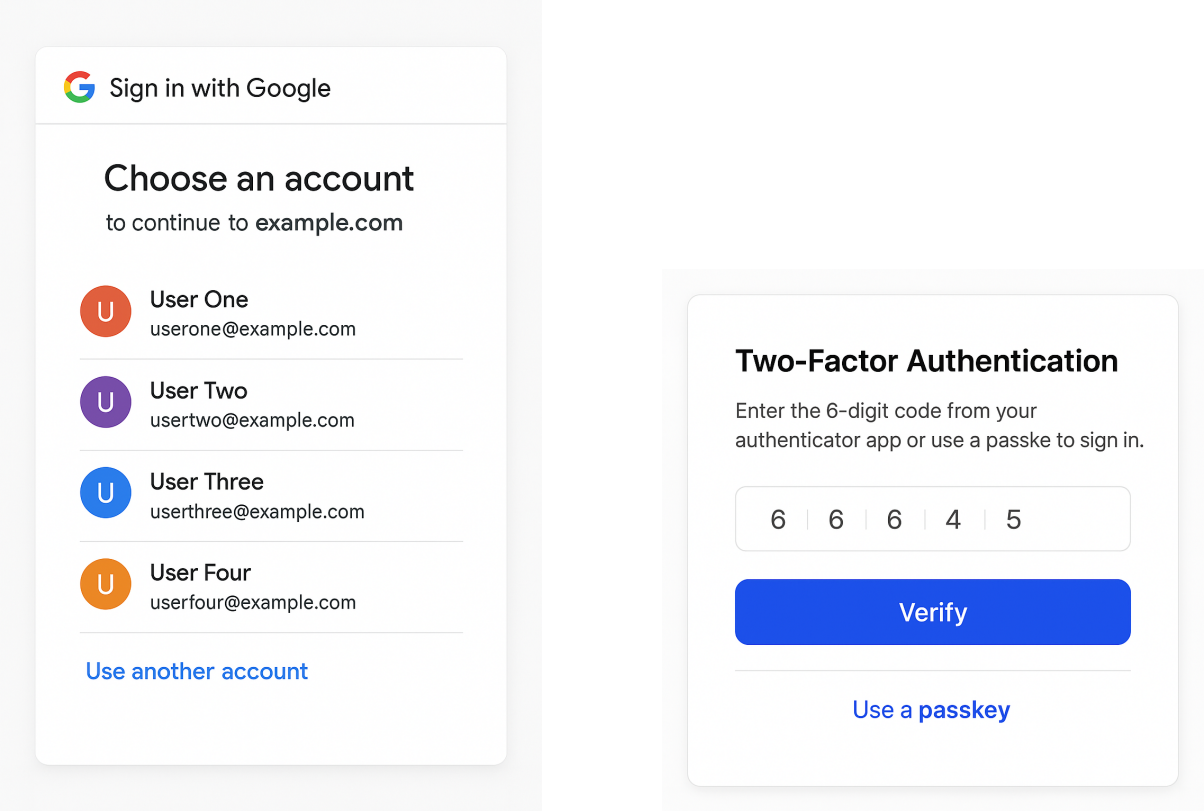


Figure 14: Version control

- Option to use an AI model to generate model solutions and testcases when creating problems.
- Google authenticator feature for two-factor-authentication, added support for passkeys.



(a) Sign-up (b) Authenticator

Figure 15: Google Authentication

- Separate interfaces for an interactive page for active contests with live scoreboards and rankings, and a detailed page for inactive contests with problem insights, difficulty levels, and hints.

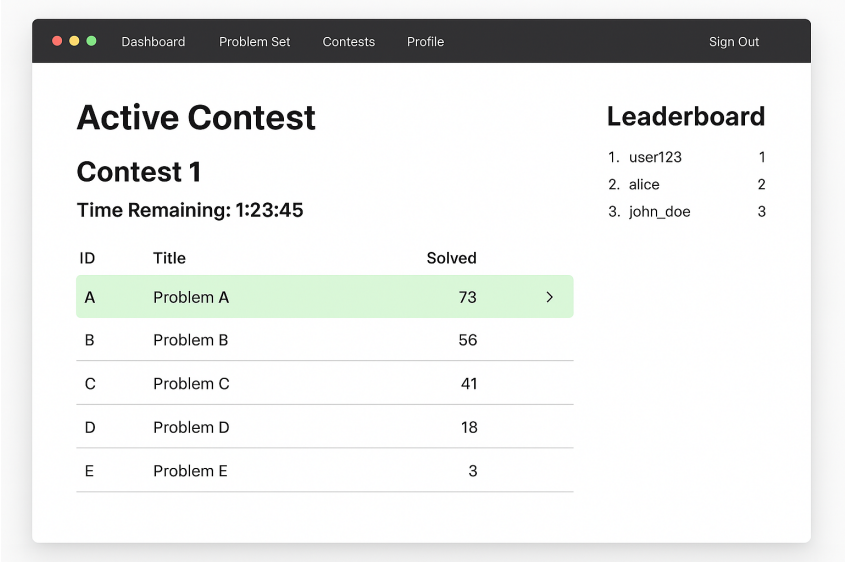


Figure 16: Interactive contest page

- Activity tracking, performance reports (SWOT analysis for example), and a dynamic rating system to unlock user privileges.

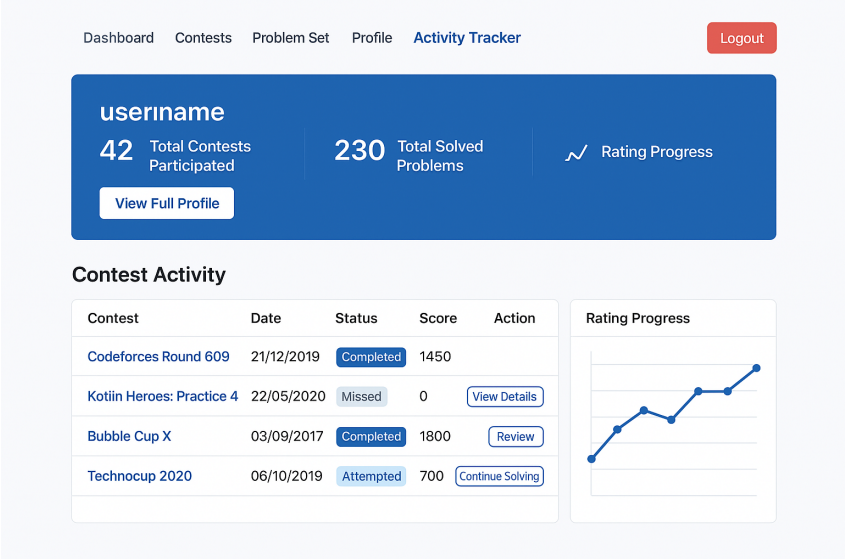
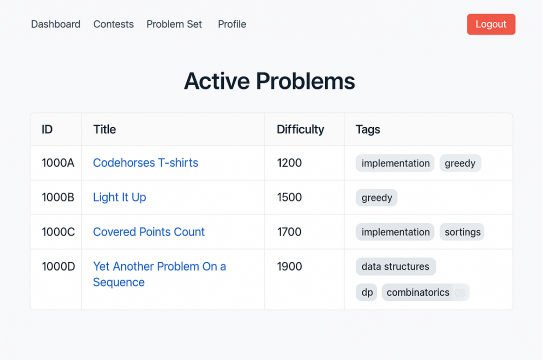


Figure 17: Activity tracking and performance reports

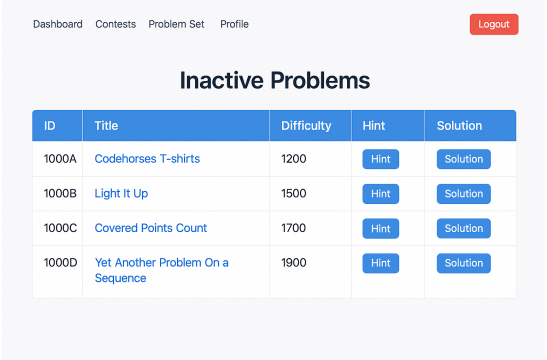
- Sandboxing for safe code execution, with built-in compilers for direct testing and compilation.

Figure 18: Sandboxing

- Active and inactive problems will be shown on separate pages, each tailored with relevant features.



(a) Active problems



(b) Inactive problems