

Team #18

Project Name: DSA VS

Group Members: Lawrence Liu, Josh Cho, Tyler Gantry, Joshua Jackson

Problem Statement:

Currently, the barrier of entry for interviews are “LeetCode” style DSA questions, which are often isolating, repetitive, and lacking instant feedback. DSA VS aims to “gameify” this process, making a once tedious task seem fun through a “1v1” style battleground.

Background Information:

a) Audience

Data Structures and Algorithms is imperative to most individuals studying Computer Science. As a result of this, there are many online resources that contain many programming questions which allows developers and programming enthusiasts to improve their coding skills. While resources such as LeetCode are great, is almost entirely individually based and lacks an interactive, competitive environment.

The idea behind “DSA VS” is to create a competitive 1v1 system where users can have real-time contests against each other. This format can appeal to those who enjoy competition and feel motivation from working with their peers. Instead of solving problems in isolation, users can compete in timed coding challenges, allowing them to improve their coding skills in friendly competition.

b) Similar platforms

Initially, “[DSA.VS](#)” was created due to one of the member’s interests in online, competitive Tetris. Through this, the initial scheme of a 1v1 Elo-based system was created. In terms of the actual DSA portion, there are many applications such as LeetCode, HackerRank and similar websites that allow users to either solve problems individually, or participate in coding contests that have large groups, both of which aren’t focused on one-on-one personalized matches.

c) Limitations

While similar websites exist that allow programmers to practice their problem solving skills, their main limitation is that they don’t have any sort of feature that allows you to directly compare and test your skill and speed of implementing solutions to different problems.

Additionally, the multiplayer aspect adds another layer of competitiveness and pressure that better simulates how you would be feeling in an interview situation. We hope to have high quality problems similar to LeetCode or CodeWars and be able to compete with them in that market but at the same time offer our own niche unique to our platform.

Functional Requirements:

1. As a user, I would like to create an account on [DSA.VS](#)
2. As a user, I would like to be able to login using a password on [DSA.VS](#)
3. As a user, I would like to be able to change the password connected to my account on [DSA.VS](#)
4. As a user, I would like to be able to set a username on [DSA.VS](#)
5. As a user, I would like to be able to change my password on [DSA.VS](#)
6. As a user, I want the ability to customize my profile for my account.
7. As a user, I want the ability to “friend” other users for later contact.
8. As a user, I want the ability to “block” other users from communication.
9. As a user, I want the ability to message other users.
10. As a user, I would like to be able to enter the queue for matchmaking on [DSA.VS](#)
11. As a user, I would like to be able to challenge a specific user
12. As a user, I would like to be able to see my current Elo on [DSA.VS](#)
13. As a user, I would like to be able to get matched with another user with a similar Elo score in real time.
14. As a user, I would want to view the history of my previous matches to review and see how I’ve improved
15. As a user I would want the matches to last no longer than 25 minutes
16. As a user, I would want to see a timer to keep track of how much time I have left
17. As a user, I would like to be able to sort my past problems by category
18. As a user, I would like to be able to see the number of test cases my opponent has passed at any given time during a match on [DSA.VS](#)
19. As a user, I would want the code that I have written to execute properly when given test cases.
20. As a user, I would like for my Elo to increase or decrease depending on the outcome of a match.
21. As a user, I would want a variety of problems within the problem set
22. As a user, I would want the problems within the problem set to be correct and able to be solved within a given time period
23. As a user, I would want to be able to queue for an “AI free for all mode”
24. As a user, I would want my opponent to play fairly, without the use of cheating in the “non-AI” mode
25. As a user, I would want a working report system to ban users that are abusing the platform
26. As a user, I want the ability to see my statistics for different modes and types of problems
27. As a user, I want the ability to sort and visualize my different stats in different formats.
28. As a user, I would want to see how my current Elo stacks up to others with a ranking system through an active leaderboard.
29. As a user, I would want to report any issues/bugs I find to ensure they’re fixed

30. As a user, I would want to be able to spectate a match (if time allows)
31. As a user, I would want a whiteboard-like scratch area to help visualize my solution (if time allows)
32. As an employer, I would like to see the current level of programmers to be able to gauge the requirements for jobs.
33. As an instructor, I would like to evaluate a student's coding performance without the use of AI.
34. As an employer, I would like to see the relative focus students have on certain problems
35. As an employer, I would like to see what programming languages students excel at
36. As a contributor, I would like to be able to submit problems to the [DSA.VS](#) problem set
37. As a contributor, I would like to submit individual test cases for a given problem

Nonfunctional Requirements:

Security

- All sensitive user data (emails, match results, etc.) should be encrypted to protect privacy and secure data
- Multi-factor authentication should be implemented for 95% to prevent unauthorized access, ensuring user accounts are secure
- User-submitted code should be properly sanitized in order to prevent vulnerabilities like SQL injection
- Sessions should time out after 15 minutes of inactivity to prevent unauthorized access

Architecture

- Code execution should be done in a sandboxed environment to ensure any malicious code has no effect

Performance

- The site should be able to handle up to 20 simultaneous users and matches at a time without lags or crashes
- The platform should have < 1 second of latency to prevent disruptions to the competitive experience
- The system should be able to handle up to 20 concurrent matches or more, allowing performance to remain stable as user demand increases

Usability

- User satisfaction with the interface should be 4.5/5 or higher to allow for easy navigation and coding without issues
- Accessibility features like a screen reader will allow the platform to be used by individuals with disabilities