DSA CODELAB: A Unified Platform for Problem Solving, Code Evaluation, and DSA Concept Visualization

Dheeraj Shukla, Ansari Mohd Saad, Divya Patil, Tanuj Patil, Vaibhav Patil

Department of Computer Engineering, D.N. Patel College of Engineering, Shahada, Dist. Nandurbar, Maharashtra, India

DOI: https://doi.org/10.51244/IJRSI.2025.12060076

Received: 19 June 2025; Accepted: 24 June 2025; Published: 09 July 2025

ABSTRACT

Despite the growing demand for strong problem-solving skills in computer science education, many existing platforms lack an integrated environment that combines problem statements, code execution, and real-time feedback in one interface. Traditional classroom teaching often fails to provide an interactive and scalable solution for hands-on coding practice. While some platforms offer code execution or learning content, few effectively merge both into a unified, student-centric tool. To address this, we present DSA CodeLab, a full-stack web-based platform primarily designed to provide a compiler-driven problem-solving interface, where users can write, test, and submit code for predefined DSA problems. The platform supports real-time code execution through JDoodle API, automated evaluation, and immediate feedback. Additionally, DSA CodeLab offers a secondary layer of DSA learning support, including a theory module and algorithm animations that visually demonstrate concepts like sorting, searching, and recursion. By combining practical coding tasks with conceptual aids, DSA CodeLab ensures a comprehensive, scalable, and user-friendly learning experience tailored for students, educators, and aspiring developers.

Keywords: Online Judge, Code Execution Platform, Interactive Learning, Algorithm Visualization, Real-Time Feedback

INTRODUCTION

In the era of rapid technological advancement, programming education has evolved significantly, yet many students still face challenges in bridging the gap between theoretical concepts and practical implementation. Traditional learning environments often lack real-time interaction and immediate feedback, which are essential for mastering data structures and algorithms (DSA) [1]. Furthermore, most online platforms either emphasize problem-solving without conceptual clarity or provide theory without interactive practice, making it difficult for learners to gain a comprehensive understanding.[2] [3] Recognizing these limitations, we introduce DSA CodeLab, a unified, full-stack web platform that aims to transform the way DSA is taught and learned. The platform is primarily designed to offer a compiler-based problem-solving interface where students can read a problem statement, write and execute code, and instantly receive feedback. This core module supports multiple programming languages through the integration of the JDoodle API[4], ensuring broad accessibility and real-time performance. In addition to the problem-solving environment, DSA CodeLab also includes supporting educational tools, such as a theory section and animated algorithm visualizations inspired by systems like JSAV-based interactive viewer [5] and mainstream platforms including VisuAlgo and Galles's Data Structure Visualizations [6] [7].These features help reinforce core concepts by visually demonstrating the inner workings of sorting, searching, recursion, and other DSA techniques. The architecture of the platform leverages modern web technologies including React.js for a responsive frontend, Spring Boot for secure backend APIs, RabbitMQ for managing concurrent tasks, and Docker for deployment and scalability. The platform was developed with both students and educators in mind, offering administrative tools for monitoring user submissions, managing problem sets, and evaluating student progress. With its combination of real-time code execution, automated evaluation, and visual learning aids, DSA CodeLab provides a scalable and interactive solution for enhancing DSA education in academic institutions and self-paced learning environments.

LITERATURE SURVEY

The development of interactive web applications has emerged as a crucial area in educational technology, enabling user engagement through responsive, browser-based platforms. A web application typically follows a multi-tier architecture, consisting of client-side development, server-side logic, and database management [8]. This structure forms the foundation of many modern learning environments and online coding systems.

In the domain of client-side development, HTML (HyperText Markup Language) plays a fundamental role in defining the structure and semantics of web content. CSS (Cascading Style Sheets) is used to style and visually organize these elements, ensuring that interfaces remain accessible and visually appealing. The importance of interactivity and user responsiveness is addressed through JavaScript, a powerful scripting language that adds behavior, control, and logic to web interfaces. Within this scope, React.js, a JavaScript library, has gained prominence due to its component-based architecture and virtual DOM manipulation, which allow the efficient development of dynamic and reusable UI components. This modular approach has been widely adopted in educational platforms to support live code rendering and user interaction. [9] [10]

On the server side, Spring Boot, a Java-based application

framework, simplifies the construction of RESTful services and server-side logic [11] [12]. The use of Spring Boot ensures rapid development, streamlined dependency management, and modular scalability. It plays a central role in handling API requests, user authentication, and communication with external code execution services in platforms such as DSA CodeLab. By automating configurations and minimizing boilerplate code, Spring Boot enhances both productivity and maintainability in back-end development.[13] [14]

For data persistence and retrieval, PostgreSQL is utilized as a relational database system known for its robustness, ACID compliance, and support for complex queries. Its structured schema and support for concurrent transactions make it particularly well-suited for managing large volumes of student submissions, test cases, and evaluation results within educational environments. PostgreSQL’s flexibility in integrating with web frameworks ensures seamless connectivity between the application layer and storage engine.

Together, these technologies form the technical backbone of the DSA CodeLab platform. The frontend—composed of HTML, CSS, JavaScript, and React—delivers a fluid and engaging user interface, allowing learners to navigate problem statements, write code, and view visualizations effortlessly. The backend—built with Spring Boot and PostgreSQL—handles user data securely and manages real-time code execution and result processing. This layered full-stack configuration enables the platform to support both its primary objective of problem-solving through a compiler interface and its secondary modules, including interactive theory explanations and algorithm animations.

LeetCode: Practice and Improve Coding Skills through Online Judge System

Summary: LeetCode is a widely used online judge platform that offers curated problem sets, coding contests, and company-specific interview questions. Its real-time evaluation system and detailed performance analysis serve as a benchmark for platforms like DSA CodeLab [15] .

Key Contribution:

Encouraged structured problem-solving in competitive and interview-based coding.

Offered time and memory analysis, enabling performance tracking.

VisualAlgo: Visualising Data Structures and Algorithms through Animation

Summary: VisuAlgo provides interactive visualizations for fundamental algorithms and data structures. It inspired the visualization module in DSA CodeLab by demonstrating how animations can enhance conceptual understanding [16] .

Key Contribution:

Used animations to break down algorithm flow step-by-step.Provided design

Improved learning through interactive traversal and sorting visual tools.

HackerRank: A Skill-Based Platform for Learning and

Hiring Programmers

Summary: HackerRank offers a challenge-driven coding environment with auto-evaluation and leaderboards. Its modular contest setup helped shape the competitive module of DSA CodeLab [17] .

Key Contribution:

Integrated learning paths with practical problem sets.

Gamified learning experience with badges and scores.

JDoodle: Online Compiler and IDE for 70+ Languages

Summary: JDoodle is a lightweight, browser-based compiler API used by DSA CodeLab to execute code across multiple languages. Its API-first design enables fast, scalable integration with online coding platforms [18] .

Key Contribution:

Provided real-time execution support via REST API.

Enabled language-independent coding environments.

CodeForces: Competitive Programming and Online Contests Platform

Summary: Codeforces is a globally recognized platform for competitive programming, known for its regular contests and strong community. It influenced the problem difficulty classification and timed contest format in DSA CodeLab [19].

Key Contribution:

Introduced rating-based contests and editorial discussions.

Fostered a global community of competitive programmers.

Objective

The objectives of the DSA Codelab project using are:

Provide a Unified Coding Platform: Develop an integrated environment for solving DSA problems, writing code, and compiling it in real-time.

Real-time Code Execution: Implement JDoodle API integration to allow users to execute code instantly and view results or errors.

Simplified User Interface: Design an intuitive and clean interface to enhance the user experience, making it easy to navigate between problem descriptions, code editor, and test cases.

Problem Selection: Allow users to choose from a variety of DSA problems and practice solving them directly on the platform.

Enhance Learning with Visual Aids: Offer theoretical explanations and animations for data structures and algorithms to help users understand complex concepts

Improve Feedback Mechanism: Provide immediate feedback to users on their code execution, making debugging and learning more efficient.

Accessibility: Ensure the platform is entirely web-based, making it accessible from any device with a stable internet connection.

Scalability: Design the platform to accommodate a growing number of users and coding problems as the system evolves.

Implementation

The DSA Codelab platform is designed using (React.js, Spring Boot) to deliver an efficient, scalable, and interactive environment for learning and practicing Data Structures and Algorithms (DSA). The architecture ensures robust functionality, user-friendly interfaces, and a seamless coding experience.

Frontend Architecture

The frontend of the DSA CodeLab platform is developed using React.js, offering a dynamic and user-friendly interface optimized for learning Data Structures and Algorithms. Leveraging React's component-based architecture, the UI is modularized into key elements such as a problem listing panel with filters for difficulty and topic, an integrated code editor (e.g., Monaco Editor) that supports syntax highlighting, autocompletion, and customizable themes, and a results panel that provides real-time feedback on test case outputs, runtime errors, and execution metrics. The frontend communicates with the backend via RESTful APIs to dynamically fetch problem data, send user-submitted code for evaluation, receive execution results, and maintain session state and progress tracking.

Backend Architecture

The backend of DSA CodeLab is built using Spring Boot, which handles the core functionalities of the platform and ensures secure, efficient operations. It is responsible for processing HTTP requests from the frontend, managing routing, and generating appropriate responses. The system integrates with external compiler APIs such as JDoodle or Judge0 to execute user-submitted code in real time and return outputs. User authentication and session management are secured using JWT (JSON Web Tokens) or OAuth protocols, ensuring protected access to user-specific data. The backend also includes a feedback mechanism that evaluates code submissions against predefined test cases, providing users with instant output, error messages, and execution metrics. Furthermore, it supports progress tracking by recording solved problems, submission history, and performance data, which can also contribute to personalized statistics or leaderboard features.

Database Management

The platform utilizes PostgreSQL, a robust and scalable relational SQL database, to efficiently manage and store core data components. It maintains comprehensive user data, including profile information, authentication credentials, and activity logs to support secure access and personalized experiences. The database dynamically handles problem sets, organizing them by difficulty level, topic, and associated constraints, thereby enabling efficient filtering and retrieval during user interactions. Additionally, it records submission history, capturing timestamps, submitted code, execution results, and feedback for each attempt. This structured storage model ensures data consistency, supports analytics, and enables smooth integration with both the frontend and backend components of the platform.

METHODOLOGY

We are developing an interactive learning platform called DSA CodeLab, focused on helping students practice and understand Data Structures and Algorithms more effectively. To achieve this, we have developed a compiler-based problem-solving interface where users can read coding problems, write their solutions, and get real-time feedback. For the frontend, we used React.js to create a fast and responsive interface, with components like the problem listing section, an embedded code editor, and a results panel that displays execution outputs and errors. The backend is built using Spring Boot, which handles the core logic, user authentication, and API interactions. We have integrated the JDoodle API for real-time code execution, allowing users to run their code in multiple programming languages. To manage user sessions, problem data, and submissions, we're using PostgreSQL as our database. We also added RabbitMQ for asynchronous task processing, so that the platform stays fast and responsive even during high usage. Apart from the coding interface, we included a DSA theory section for concept explanations and algorithm animations built with HTML, CSS, and JavaScript, to help users visualize how algorithms like merge sort, binary search, and recursion work step by step. With this tech stack, we aim to provide a complete and user-friendly platform that supports both practical coding and conceptual learning.

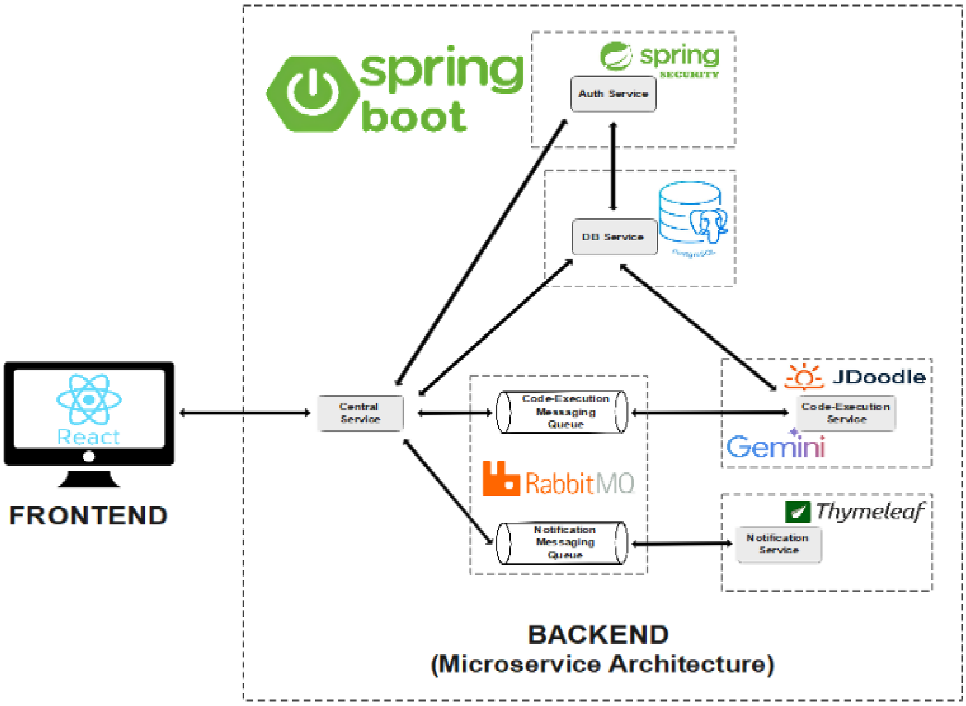


Fig 1. System Architecture

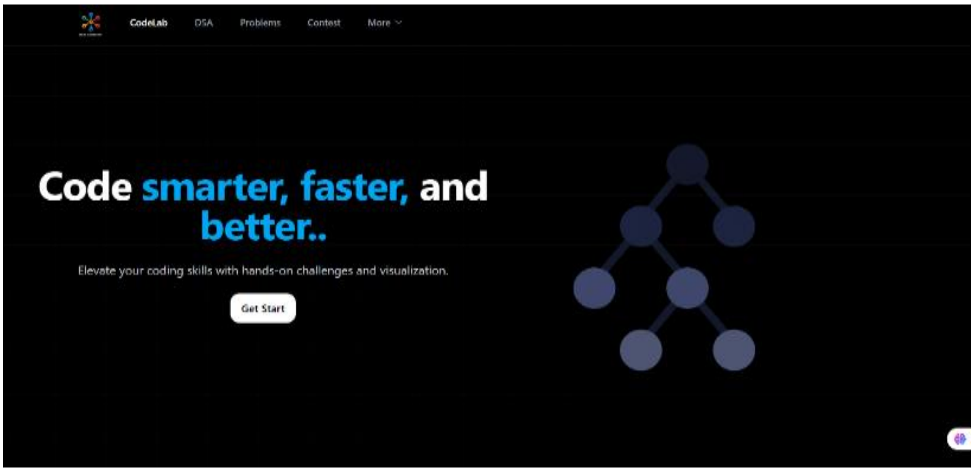
As shown in Figure 1, at the client side, users interact with the platform through a browser-based interface developed using React.js, which is responsible for rendering the user interface, capturing inputs, and communicating with the backend. The frontend includes components like the problem listing, code editor, and output display, and interacts with backend services via API calls.

When a user logs in or submits code, the request is routed through a central service which manages all communication between frontend and backend microservices. Authentication requests are handled by the Auth Service, built with Spring Security, which validates user credentials against the PostgreSQL database through the DB Service. Once authenticated, users can access protected modules such as problem-solving and submission history.

For code execution, the central service forwards the user's code to the Code-Execution Messaging Queue, implemented using RabbitMQ. This queue triggers the Code Execution Service, which integrates with third-party APIs like JDoodle or Gemini to run the code and return execution results. Similarly, the Notification Messaging Queue handles background events and passes them to the Notification Service, which uses Thymeleaf for rendering messages.

RESULTS

Landing Page– Entry Point and Platform Overview



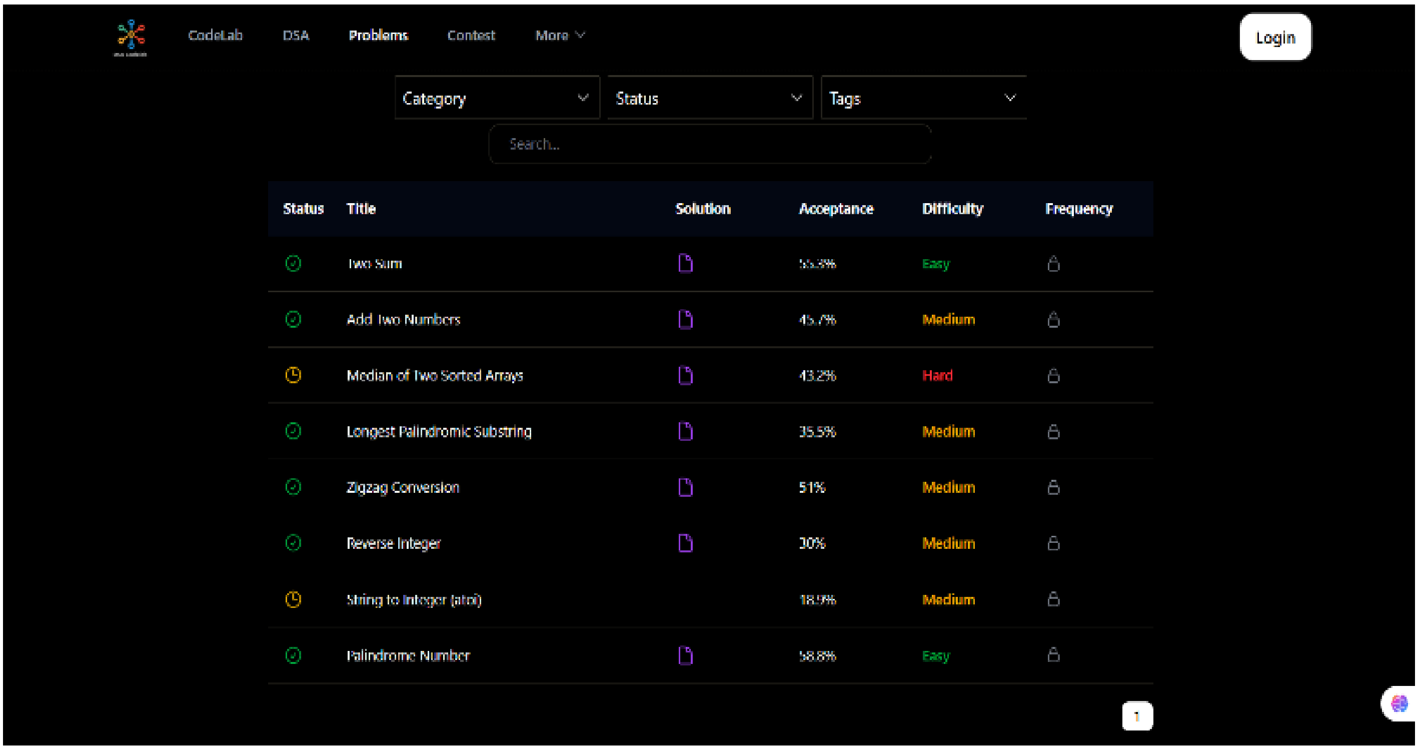
Code smarter, faster, and better..

Elevate your coding skills with hands-on challenges and visualization.

Fig. 2 Homepage

Above Figure 2 illustrates the Landing Page of our platform DSA CodeLab, which serves as the initial entry point for users interested in practicing data structures and algorithms. From this page, users can navigate to different sections of the platform, such as DSA problems, contests, and learning modules. The page also includes a clear call-to-action button to help users get started quickly. The layout is minimal and visually engaging, providing a brief introduction to the platform's purpose. This helps users understand the platform's goal and smoothly transition into the core functionalities.

Problem Page–

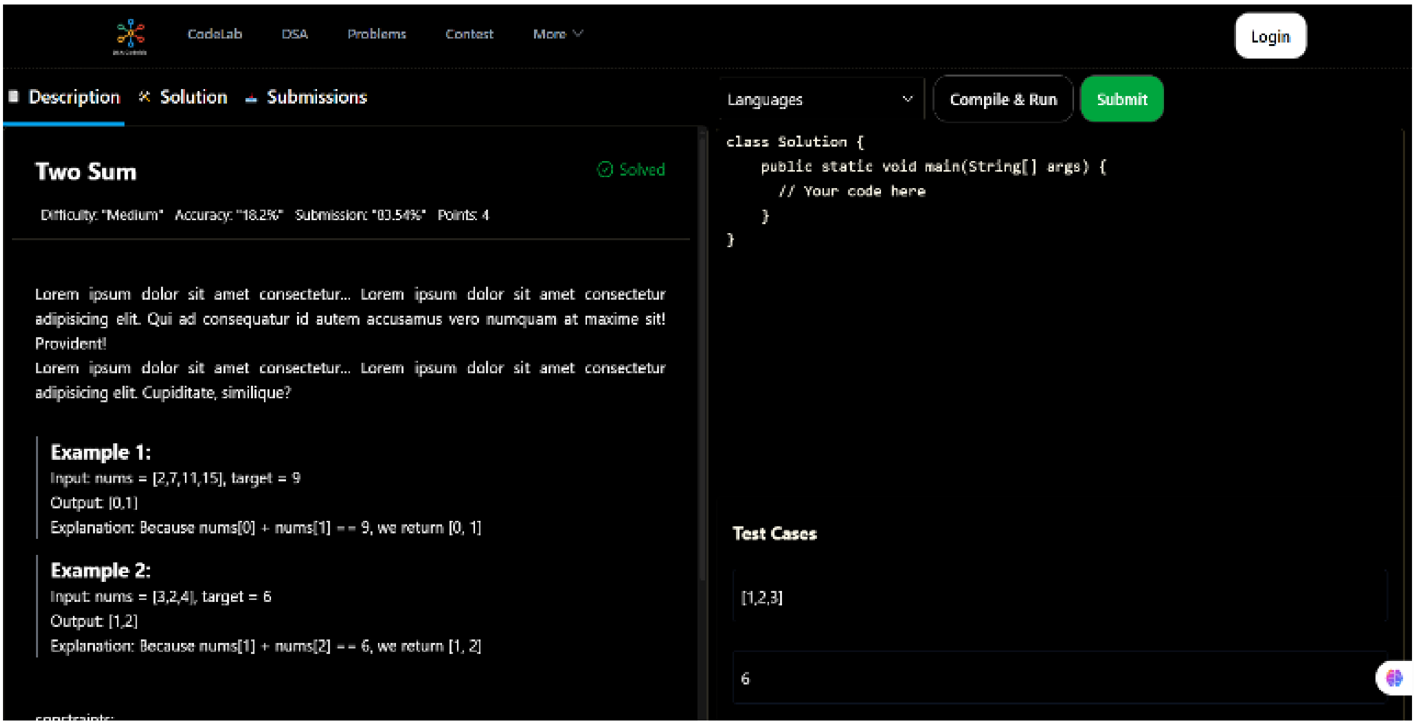


Status	Title	Solution	Acceptance	Difficulty	Frequency
✓	Two Sum	🗎	55.3%	Easy	🔒
✓	Add Two Numbers	🗎	45.7%	Medium	🔒
◷	Median of Two Sorted Arrays	🗎	43.2%	Hard	🔒
✓	Longest Palindromic Substring	🗎	35.5%	Medium	🔒
✓	Zigzag Conversion	🗎	51%	Medium	🔒
✓	Reverse Integer	🗎	30%	Medium	🔒
◷	String to Integer (atoi)		18.9%	Medium	🔒
✓	Palindrome Number	🗎	58.8%	Easy	🔒

Fig.3 List of Problems Page

Above Figure 3 shows the Problems Page of the platform, which lists all available coding problems. Users can view problem titles along with their difficulty, acceptance rate, and status. Filters like category, status, and tags help users find specific problems easily. Each problem also links to its solution, allowing users to practice and learn efficiently.

Integrated Compiler Page:



Two Sum

Difficulty: "Medium" Accuracy: "18.2%" Submission: "03.54%" Points: 4

Lorem ipsum dolor sit amet consectetur... Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui ad consequatur id autem accusamus vero numquam at maxime sit! Provident!
Lorem ipsum dolor sit amet consectetur... Lorem ipsum dolor sit amet consectetur adipisicing elit. Cupiditate, similique?

Example 1:
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1]

Example 2:
Input: nums = [3,2,4], target = 6
Output: [1,2]
Explanation: Because nums[1] + nums[2] == 6, we return [1, 2]

constraints:

```
class Solution {
    public static void main(String[] args) {
      // Your code here
    }
}
```

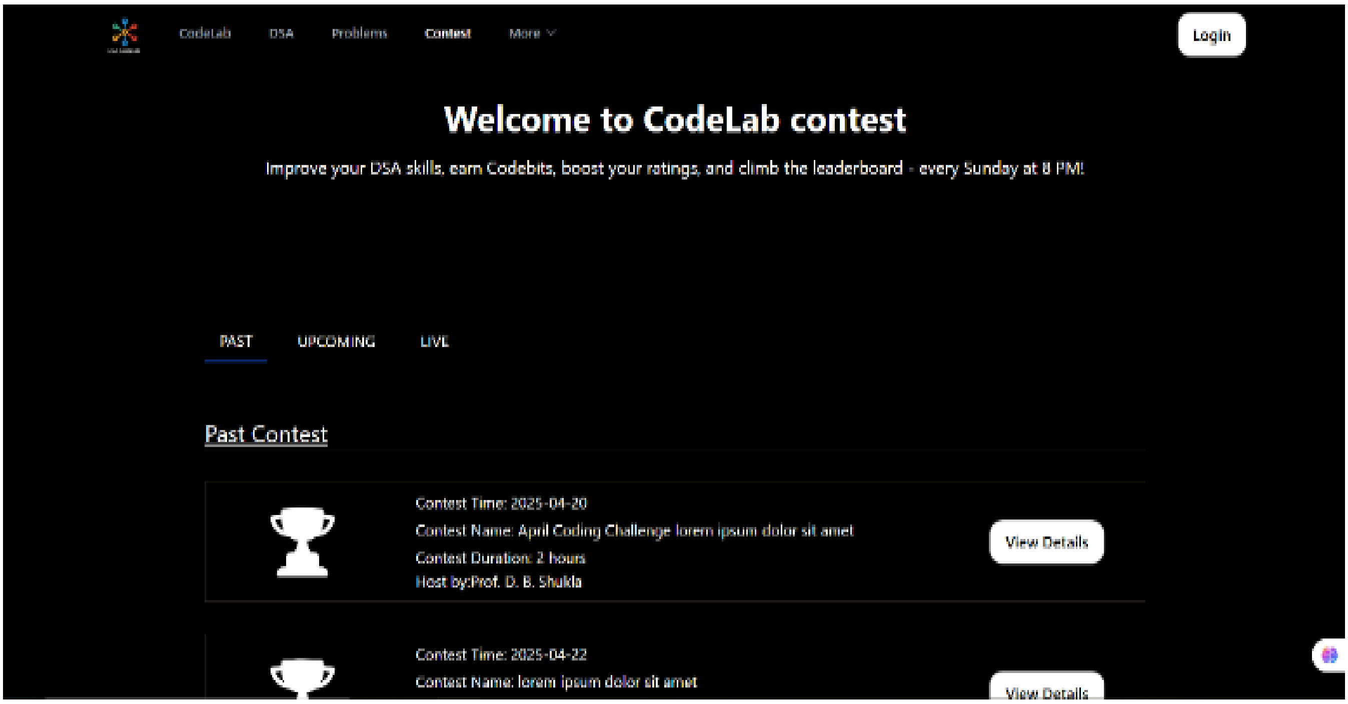
[1,2,3]

6

Fig.4 Integrated Compiler

Figure 4 showcases the coding interface of our platform, DSA CodeLab, designed to enhance the problem-solving experience for learners. This interface allows users to read problem descriptions, analyze test cases, and write code—all within a single, intuitive layout. The left panel presents the problem statement, difficulty level, sample inputs/outputs, and explanations, while the right panel offers an interactive coding environment supporting multiple programming languages. Additionally, users can compile, run, and submit their solutions seamlessly, making the platform ideal for practicing Data Structures and Algorithms in a structured and user-friendly environment.

Contest Page:



Welcome to CodeLab contest

Improve your DSA skills, earn Codebits, boost your ratings, and climb the leaderboard - every Sunday at 8 PM!

Past Contest

Contest Time: 2025-04-20
Contest Name: April Coding Challenge lorem ipsum dolor sit amet
Contest Duration: 2 hours
Host by:Prof. D. B. Shukla

Contest Time: 2025-04-22
Contest Name: lorem ipsum dolor sit amet

Fig 5. Contest Page

Figure 5 shows the Contest Page of the platform, where all available and past contests are listed. Users can view contest names, start times, durations, and participation status. From this page, users can register for upcoming contests or explore previous contests to view rankings and problems. It helps users stay engaged with regular challenges and track their progress through competitive practice.

CONCLUSION

The DSA CodeLab platform offers an effective and user-friendly environment for learning and practicing Data Structures and Algorithms. By integrating coding challenges with interactive visualizations, the platform enhances both conceptual understanding and practical skills. Immediate feedback, guided problem-solving, and detailed test case analysis contribute to improved learning outcomes. With its intuitive interface and responsive design, DSA CodeLab supports academic learning, interview preparation, and competitive programming, making it a valuable tool for learners at various levels.

REFERENCES

1. IJIRT, “Data Structures & Algorithms Visualizer: Interactive animations & real-time feedback,” IJIRT, vol. 11, no. 11, Nov. 2024.[Online].Available:https://ijirt.org/publishedpaper/IJIRT169087_PAPER.pdf
2. Christy et al., “Data Structures Visualization,” IJNRD, Apr.2024. [Online].Available:https://www.ijnrd.org/papers/IJNRD2404264.pdf
3. J. Lin and H. Zhang, “Data Structure Visualization on the Web,” ResearchGate, 2022. [Online].Available:https://www.researchgate.net/publication/350199044_Data_Structure_Visualization_on_the_Web
4. JDoodle, “Online Compiler and IDE – JDoodle,” 2024. [Online].Available:https://www.jdoodle.com
5. P. Perháč and S. Šimoňák, “Interactive System for Algorithm and Data Structure Visualization,” ResearchGate, 2022.[Online].Available:https://www.researchgate.net/publication/362297100
6. D. Galles, “Data Structure Visualizations,” University of SanFrancisco. [Online].Available:https://www.cs.usfca.edu/~galles/visualization/
7. S. Halim, “VisuAlgo: Visualising Data Structures and Algorithms,” VisuAlgo.net, 2024. [Online]. Available: https://visualgo.net
8. "Spring Boot – Architecture," GeeksforGeeks, [Online]. Available:https://www.geeksforgeeks.org/spring-boot-architecture/. Accessed on: 18 June 2025.
9. "Code, Compile, and Debug React Apps," NextLeap, [Online]. Available: https://nextleap.com/react-apps. Accessed on: 18 June 2025.
10. "The interactive way to master modern React," React.gg, [Online]. Available: https://react.gg. Accessed on: 18 June 2025.
11. "Spring Boot – Architecture," GeeksforGeeks, [online]. Available: https://www.geeksforgeeks.org/spring-boot-architecture/. Accessed on: 18 June 2025.
12. "Spring Boot," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Spring_Boot. Accessed on: 18 June 2025.
13. "Spring Boot with PostgreSQL: A Step-by-Step Guide," Medium, [Online]. Available: https://medium.com/@spring-boot-postgresql. Accessed on: 18 June 2025.
14. "Using Postgres Effectively in Spring Boot Applications," HackerNoon, [Online]. Available: https://hackernoon.com/postgres-in-spring-boot. Accessed on: 18 June 2025.
15. "LeetCode – The World's Leading Online Programming Learning Platform," LeetCode, [Online]. Available: https://leetcode.com. Accessed on: 18 June 2025.
16. "VisuAlgo: Visualising Data Structures and Algorithms," VisuAlgo.net, [Online]. Available: https://visualgo.net. Accessed on: 18 June 2025.
17. "HackerRank – Online Coding Tests and Technical Interviews," HackerRank, [Online]. Available: https://www.hackerrank.com. Accessed on: 18 June 2025.
18. JDoodle, “Online Compiler and IDE – JDoodle,” 2024. [Online]. Available: https://www.jdoodle.com. Accessed on: 18 June 2025.
19. "Codeforces – Programming competitions and contests," Codeforces.com, [Online]. Available: https://codeforces.com. Accessed on: 18 June 2025.