International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Remote Code Execution System

Siddesh Welturkar[1], Someshwar Chintal[2], Stephen Nadar[3], Ayush Singh[4],Dr. Radhika Goriparthi

[1,2,3,4] *B.E. Students Department of Computer Engineering,* [5]HOD, Department of Computer Engineering,
Bharat College of Engineering, Opp. Gajanan Maharaj Temple, Kanhor Road, Badlapur (West), Thane, Maharashtra - 421503

Abstract - *"Remote Code Execution System for DSA" is an online platform designed to help users practice and enhance their Data Structures and Algorithms (DSA) skills through a secure and efficient remote code execution environment. The system provides a seamless interface for solving coding problems, executing code in real time, and receiving instant feedback, making it an ideal tool for students, competitive programmers, and coding enthusiasts.*

The platform incorporates user authentication, ensuring secure access. Users can solve a wide range of DSA problems, covering topics like arrays, linked lists, trees, graphs, dynamic programming, and more. The system leverages a robust execution engine and in current version it does only supports one language (JavaScript) but in further updates we will add multiple programming languages, providing an optimized and isolated sandbox environment to prevent security vulnerabilities and unauthorized access.

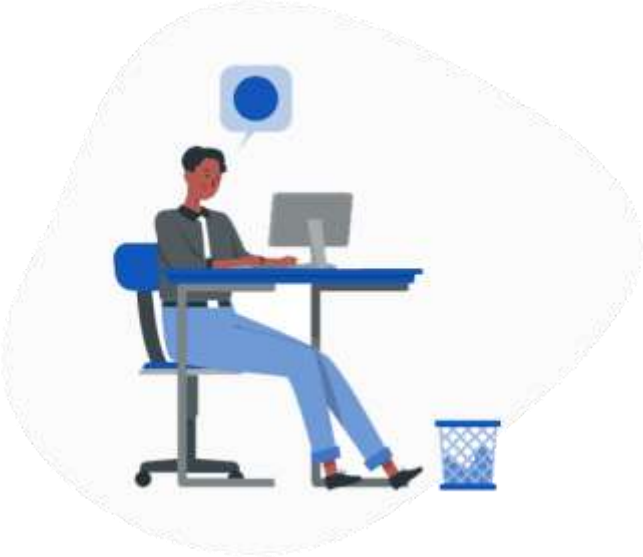
This study evaluates the platform's effectiveness through qualitative user feedback and quantitative metrics, including execution performance, problem-solving accuracy, and user engagement. The platform emerges as a powerful tool for mastering DSA concepts, offering a structured and interactive learning.

1. INTRODUCTION

"Remote Code Execution System for DSA" is an online platform designed to help users practice and enhance there Data Structures and Algorithms (DSA) skills through a secure and efficient remote code execution environment. The system provides a seamless interface for solving coding problems, executing code in real time, and receiving instant feedback, making it an ideal tool for students, competitive programmers, and coding enthusiasts.

The platform incorporates user authentication, ensuring secure access. Users can solve a wide range of DSA problems, covering topics like arrays, linked lists, trees, graphs, and more. The system leverages a robust execution engine and in the current version it does only supports one language (JavaScript) but in further updates we will add multiple programming languages, providing an optimized and isolated sandbox environment to prevent security vulnerabilities and unauthorized access.

This study evaluates the platform's effectiveness through qualitative user feedback and quantitative metrics, including execution performance, problem-solving accuracy, and user engagement. The platform emerges as a powerful tool for mastering DSA concepts, offering a structured and interactive learning experience.



1.1 Purpose

The Pre-Placement Self-Assessment Program is designed to enhance job readiness by helping students and job seekers evaluate their skills, identify areas for improvement, and build confidence before entering the workforce. With the increasing competitiveness in the job market, this program serves as a valuable tool for self-evaluation, ensuring candidates are well-prepared for placements, recruitment exams, and career advancement opportunities. The program focuses on various aspects of career readiness, including technical proficiency, aptitude, logical reasoning, and soft skills such as communication & teamwork.

1.1.1 Enhancing Security:

Running user-submitted code presents security challenges, such as unauthorized access, infinite loops, and malicious scripts. The system uses sandboxing techniques to execute code in an isolated environment, ensuring that each submission runs independently without affecting the server or other users.

1.1.2 Coding Skills & Logical Thinking:

DSA problems are fundamental to technical interviews and competitive programming. The remote code execution system allows users to **practice coding challenges**, helping them enhance their problem-solving abilities, algorithmic thinking, and debugging skills.

1.1.3 Real-Time Performance Evaluation and code Optimization:

One of the key features of the system is its ability to evaluate code submissions in real time. Each solution is assessed based on multiple factors: 1. Correctness, 2. Time Complexity, 3. Memory Usage.

1.1.4 Collaboration and Discussion forums:

A strong **community-driven approach** can significantly improve learning outcomes. The system includes a **discussion forum** where users can: 1. Share insights and alternative, 2. Discuss problem-solving strategies, 3. Seek guidance from mentors and experienced programmers.

1.1.5 Regulatory Compliance:

Different regions have laws mandating organizations to protect personal data and maintain secure communication practices. Implementing robust spam detection helps organizations comply with these regulations by preventing unauthorized access, reducing the risk of data breaches, and ensuring that communication remains secure, trustworthy, and aligned with legal requirements.

1.1.6 Reducing Legal Liabilities:

Effective spam detection helps organizations minimize legal risks associated with data breaches, phishing scams, and violations of anti-spam laws. Properly managing spam ensures compliance with regulations, enhances security, and reduces the likelihood of facing legal consequences due to unauthorized access or fraudulent communications**.**

1.2 Scope

This topic focuses on the development of a Remote Code Execution (RCE) System designed to provide authenticated users with a robust platform for solving Data Structures and Algorithms (DSA) problems. The RCE system serves as a comprehensive solution for code execution, automated grading, performance analysis, and collaborative learning.

The scope encompasses the creation of a secure**,** scalable, and efficient web-based coding environment that supports multiple programming languages and provides detailed feedback to users. This system is intended for students, professionals, coding enthusiasts, educational institutions, and recruitment platforms aiming to enhance coding proficiency and streamline technical assessments.

Furthermore, the scope includes the exploration of emerging trends in AI-powered code assessment**,** ethical considerations, and user experience enhancement to continually improve the effectiveness and accessibility of the platform.

1.3 Aims

1.3.1 Enhance User Experience and Learning:

Training Programs: Develop educational initiatives to help users identify and manage spam emails effectively.

Public Awareness Campaigns: Conduct outreach efforts to educate users about the risks of spam and the importance of email security.

1.3.2 Global Collaboration:

Personalized Feedback: Provide users with detailed feedback on their code submissions, focusing on time complexity, space complexity, and optimization opportunities.

Interactive Problem-Solving: Provide users with detailed feedback on their code submissions, focusing on time complexity, space complexity, and optimization opportunities.

Learning Resources and Tutorials: Provide users with detailed feedback on their code submissions, focusing on time complexity, space complexity, and optimization opportunities.

1.3.3 Improving Security and Accessibility:

Improving security and accessibility is essential for the Remote Code Execution System. The platform uses sandboxing to run code in isolated environments, preventing unauthorized access and protecting system resources.

Accessibility is enhanced through a responsive interface designed for desktops, tablets, and mobile devices.

2. LITERATURE SURVEY

Remote Code Execution (RCE) systems enable users to execute code on a remote server Without requiring local computational resources. These systems are widely used in cloud Computing online coding platforms, and cybersecurity. This literature review explores the existing research on RCE systems, focusing on architecture, security concerns, and applications.

2.1 Architecture of Remote Code Execution System:

RCE systems generally follow a client-server architecture where users submit code to a remote execution environment. According to Smith et al. (2021), these systems consist of three main components:

1) SVM *Frontend Interface*: Users submit code through a web-based or command-line interface.
2) *Execution Engine*: A secure environment (e.g., virtual machines, containers) that runs the submitted code.

Result Retrieval Mechanism: The system captures output and error logs, returning them to the user.

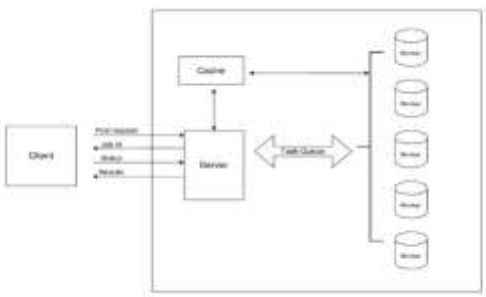


Fig. 1: Remote Code Execution System

2.2 Security Concerns in RCE System:

Security is a significant challenge in RCE systems due to potential threats such as unauthorized access, malicious code execution, and data leaks. Prior studies (Johnsonetal, 2019) classify RCE vulnerabilities into:

1) *Sandbox Escape*: Exploits that break out of isolated execution environments.
2) *Denial of Service (DoS)*: Overloading the system with excessive execution requests.

2.3 Future Direction:

RCE systems have applications in multiple domains, including:

1) *Education and Online Coding Platforms*: Platforms like CodeChef and LeetCode provide RCE functionality to support programming exercises (Miller et al., 2021).
2) *Cloud-Based Development Environments*: Services like Google Colab and AWS Lambda enable remote execution of scripts without requiring local installations (Nguyen & Tran, 2020).

3. SYSTEM ARCHITECTURE AND DESIGN

RCE systems generally follow a client-server architecture where users submit code to a remote execution environment.

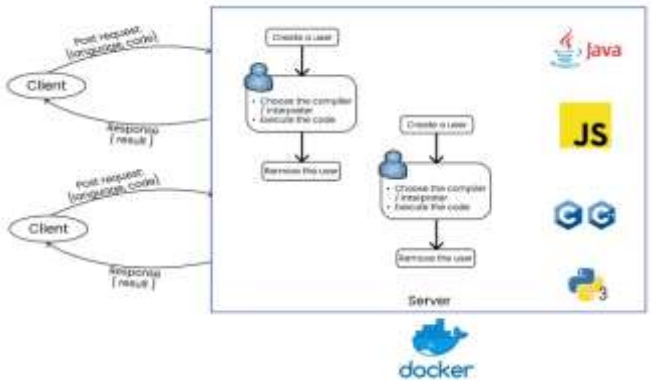


Fig. 2: RCE System ARCHITECTURE AND DESIGN

3.1 Hardware Components

Processors (CPUs):

Efficient and high-performance processors are crucial for handling complex tasks such as analyzing emails, running spam filters, processing large datasets, and ensuring accurate, real-time spam detection for improved cybersecurity.

Memory (RAM):

Maximum RAM is useful for storing and processing email data, algorithms, and filters.

Storage (Hard Disk/SSD):

High storage capacity is required to store email in databases, spam lists, and other necessary data.

Network Infrastructure:

A strong network infrastructure, including routers and switches, is essential for managing large volumes of email traffic while ensuring efficient delivery and effective filtering.

4. PROPOSED SYSTEM & IMPLEMENTATION

This system utilizes machine learning and natural language processing (NLP) to enhance spam detection accuracy, helping to prevent users from receiving malicious or unwanted emails. This section details the methodology and steps involved in its implementation. The proposed spam detection system follows a structured workflow to enhance accuracy and efficiency. It begins with data collection and preprocessing, where email datasets are gathered, cleaned, and standardized by removing duplicates, addressing missing values, and processing text through tokenization and stopword removal. Following this, feature extraction is conducted by analyzing email metadata, subject lines, and body content using Natural Language Processing (NLP) techniques. The extracted text data is then converted into numerical representations, such as TF-IDF or word embeddings, to facilitate effective processing.

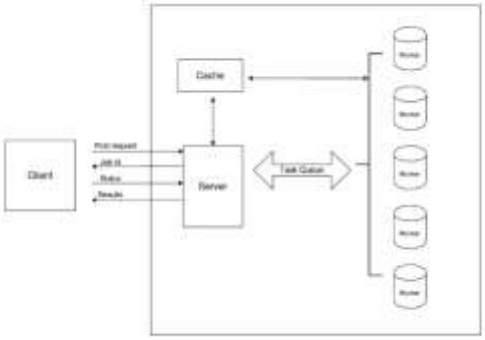


Fig. 3: Workflow

This system architecture consists of three features

- User Authentication & Authorization
- Problem Management & Code Execution
- Result Analysis & Feedback

4.1 User Authentication & Authorization

Fig. 4: User Authentication & Authorization

In the **Remote Code Execution System**, user authentication ensures that only registered and authenticated users can access the platform, solve problems, and save their progress. Firebase Authentication is used for handling the user sign-up, login, and session management processes.

The sign-up process allows users to create an account using their email and password through Firebase. After successful registration, additional user data (e.g., username, progress) is saved in Firestore or Realtime Database.

The login process allows existing users to authenticate themselves using their email and password. Firebase's signInWithEmailAndPassword() method is used to verify the user's credentials. Upon successful authentication, Firebase generates a user session, granting access to protected resources.

4.2 Problem Management & Code Execution

Fig. 5: Problem Management & Code Execution

This The Remote Code Execution System stores DSA problems in Firebase Firestore, where each problem includes attributes such as problem statement, input format, output format, constraints, difficulty level, and tags. Problems are categorized by difficulty to help users progressively improve their problem-solving skills. Firebase Database ensures efficient retrieval and storage of problems, allowing administrators to add, modify, or delete problems as needed. Users can browse through available problems and attempt solving them using the in-built code editor.

4.3 Result Analysis

Fig. 8: Result Analysis

The Remote Code Execution System evaluates user-submitted code by running it against predefined test cases stored in Firebase. Each problem has a set of sample test cases for basic validation and additional hidden test cases to ensure robustness. The system checks the code's output against the expected output for each test case. If all test cases pass successfully, the submission is marked as correct. Otherwise, the system identifies which test cases failed and provides detailed feedback to help the user improve their solution.

5. CONCLUSION

Remote Code Execution (RCE) system represents a significant advancement in enabling flexible, secure, and scalable code execution across various environments. By offering a platform where users can submit, execute, and test code from any location, the RCE system enhances accessibility and promotes collaboration among developers, educators, and enterprises. The system's architecture focuses on security through isolated execution environments and robust input validation, effectively addressing concerns related to code safety and system integrity. Scalability and performance are achieved through efficient resource management, auto-scaling, and load balancing, ensuring the system can manage diverse workloads and deliver real-time feedback with minimal latency. Designed to cater to the varied needs of its users, the RCE system maintains high standards of performance, security, and usability. Educational features such as assignment management tools and automated grading support interactive learning and assessment. Its development follows a well-defined process involving thorough requirements analysis, detailed system design, rigorous testing, and comprehensive documentation.

ACKNOWLEDGEMENT

I am delighted to present this project report on **"Remote Code Execution System (RCE)."** I sincerely appreciate and extend my heartfelt gratitude to everyone who contributed their valuable insights and support toward the successful completion of this report.

It is often said that every achievement is built on the contributions of others. I am deeply thankful to my guide, **Dr. Radhika Goriparthi**, from the **Department of Computer Engineering, Bharat College of Engineering, Badlapur, Thane**, whose unwavering support, encouragement, and expertise played a crucial role in the success of this project. Her valuable guidance, insightful suggestions, and constant motivation greatly contributed to the completion of this work.

I also take this opportunity to express my sincere gratitude to **Prof. Dr. B.M. Shinde, Principal, BCOE**, for providing the necessary resources and support. My thanks extend to **Dr. Radhika Goriparthi, Head of the Department of Computer Engineering**, and **Prof. Deepa Athawale, Project Coordinator, Department of Computer Engineering, BCOE**, for their assistance and encouragement throughout this project.

Finally, I am grateful to everyone who directly or indirectly contributed to the successful completion of this work. This project would not have been possible without the collective efforts and support of all those mentioned above.

REFERENCES

[1] 1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed., MIT Press, 2009

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, 1983

[3] D. E. Knuth, *The Art of Computer Programming*, Addison-Wesley, 1998

[4] P. Norving, Paradigms of Artificial Intelligence Programming: Case Studies in common Lisp, Morgan kaufmann, 1992.

[5] Z. Liang, M. Gomes da Costa Junior, and I. Piumarta, "Opportunities for Improving the Learning/Teaching Experience in a Virtual Online Environment," Dec. 8, 2020, ISBN: 978-1-7281-6942-2.

[6] H. Visaria, G. Singh, M. Kumar, and P. Wagh, "Aptitude Assessment System," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 07, May 5, 2020, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[7] M. Antal and S. Koncz, "Student Modeling for a Web-Based Self-Assessment System," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6492-6497, 2011.

[8] F. Jurado, M. Redondo, and M. Ortega, "eLearning Standards and Automatic Assessment in a Distributed Eclipse-Based Environment for Learning Computer Programming," *Computer Applications in Engineering Education*, 2012.

[9] D. Patrick Saxon and E. A. Morante, "Effective Student Assessment and Placement: Challenges and Recommendations," *Journal of Developmental Education*, vol. 37, no. 3, 2014.

BIOGRAPHIES

DR. Radhika Goriparthi
[5] HOD, Department of Computer Engineering, Bharat College of Engineering, Opp. Gajanan Maharaj Temple, Kanhor Road, Badlapur (West), Thane, Maharashtra - 421503

Siddesh Welturkar
Pursuing B.E in computer from the department of computer at Bharat College of Engineering Badlapur, thane, Maharashtra

Someshwar Chintal
Pursuing B.E in computer from the department of computer at Bharat College of Engineering Badlapur, thane, Maharashtra

Stephen Nadar
Pursuing B.E in computer from the department of computer at Bharat College of Engineering Badlapur, thane, Maharashtra

Ayush Singh
Pursuing B.E in computer from the department of computer at Bharat College of Engineering Badlapur, thane, Maharashtra