JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION OF ONLINE PLATFORM FOR CONDUCTING CODING INTERVIEWS

[1]Ruchika Dilip Khekare, [2]Abhimanyu Prakash, [3]Pranshu Verma, [4]Payal Parshive, [5]Prof. P. J. Chate

[1]B.E. Student, Department of Computer Engineering, SPPU, Pune, P.O. Box 411002,India,
[2]B.E. Student, Department of Computer Engineering, SPPU, Pune, P.O. Box 411002,India,
[3]B.E. Student, Department of Computer Engineering, SPPU, Pune, P.O. Box 411002,India,
[4]B.E. Student, Department of Computer Engineering, SPPU, Pune, P.O. Box 411002,India,
[5]Assistant Professor, Department of Computer Engineering, SPPU, Pune, P.O. Box 411002,India,

Abstract : The use of online platforms for taking coding interviews has become increasingly popular in recent years, especially in light of the COVID-19 pandemic that has forced many companies to transition to remote work. In this paper, we present an online platform for taking coding interviews that utilizes WebRTC, WebSockets, and a compiler to enable real-time communication, code sharing, and testing. We discuss the design and implementation of the platform, including the integration of WebRTC for video and audio communication, WebSockets for data exchange, and a compiler for code testing. We also evaluate the performance and usability of the platform using a sample set of users and provide insights into the benefits and limitations of our approach.

***Keywords*- Design, Architecture, Technology, Evaluation, Security.**

I. INTRODUCTION

Technical interviews are an essential component of the software development industry. They are conducted to assess a candidate's technical skills and ability to solve real-world problems. Traditionally, technical interviews have been conducted in person, but with the advent of remote work, online technical interviews have become increasingly popular. Online technical interviews, however, come with their own set of challenges. One of the most significant challenges is the lack of a real-time coding environment that can simulate an in-person interview. To address this challenge, we propose the implementation of an online coding interview platform with a compiler, WebRTC, and WebSockets. The platform is designed to provide a real-time coding environment for candidates to write and test code while interacting with interviewers through video and chat features. The platform aims to make the technical interview process more efficient, effective, and engaging for both candidates and interviewers. The process of taking coding interviews has traditionally been conducted in-person, either at the company's offices or at an external location. However, the COVID-19 pandemic has forced many companies to transition to remote work, which has made it challenging to conduct in-person interviews. As a result, online platforms for taking coding interviews have become increasingly popular. These platforms enable remote communication and collaboration between the interviewer and interviewee, but they often lack features such as real-time code sharing and testing. In this paper, we present an online platform for taking coding interviews that utilizes WebRTC, WebSockets, and a compiler to enable real-time communication, code sharing, and testing.

II. LITERATURE SURVEY

Wang Lei, "Micro Service Architecture and Practice" [1] This book discusses the concept of microservice architecture and its practical applications. It explores the design principles, benefits, and challenges associated with microservices. The author provides insights into the architecture's scalability, modularity, and fault tolerance. S. Li, Y. Zhuo, S. Feng, and C. Yi, "The application of course-oriented online judge in 'Data Structure'" [2] This paper focuses on the implementation of an online judge system for a specific course on data structures. It highlights the benefits of using an online judge to evaluate and assess students' programming assignments. The authors discuss the system's design, functionality, and its impact on improving learning outcomes. Laurie Williams and Robert Kessler, "Pair Programming Illuminated" [3] In this book, the authors delve into the practice of pair programming, where two programmers work together on the same code. They explore the benefits, challenges, and best practices associated with pair programming. The book also provides insights into the impact of pair programming on code quality and productivity. LeetCode [4] is an online platform that offers a collection of coding challenges and interview questions. It provides a wide range of algorithmic problems to help developers enhance their coding skills and prepare for technical interviews. The platform supports multiple programming languages and offers solutions from the community. K. I. ZinnahApu, N. Mahmud, F. Hasan, and S. H. Sagar, "P2P video conferencing system based on WebRTC" [5]

This paper focuses on the development of a peer-to-peer video conferencing system using WebRTC (Web Real-Time Communication) technology. It discusses the system architecture, signaling protocols, and multimedia data transmission techniques. The authors highlight the advantages and challenges of implementing a WebRTC-based video conferencing system. International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017 [6] This conference paper presents a study on electrical, computer, and communication engineering. While the specific details are not provided, it may contain research papers related to various aspects of these fields, including software engineering, networking, and algorithms. CodeEval Inc. [7] CodeEval is an online platform that provides coding challenges and competitions to assess programmers' skills. It offers a collection of problems across different domains and programming languages. Participants can submit their solutions, which are then evaluated and ranked based on their performance. HackerRank FAQ [8] HackerRank is an online platform that offers coding challenges, tutorials, and interview preparation materials. The FAQ section provides answers to commonly asked questions about the platform's features, supported languages, and usage guidelines. It can be a helpful resource for understanding the platform's functionalities. HackerEarth Tutorial [9] HackerEarth is an online platform that provides tutorials, practice problems, and coding competitions. The tutorial section offers resources on various topics, including time and space complexity analysis. It provides guidance on understanding and analyzing the efficiency of algorithms and data structures.

III. DESIGN AND ARCHITECTURE:

1. Design of the Platform

The platform's architecture is designed to be scalable, fault-tolerant, and secure. The platform is built on a microservices architecture that allows for the separation of concerns and the deployment of individual services independently. The platform consists of the following microservices:

1. Authentication Service: The Authentication Service is responsible for managing user authentication and authorization.

2. Interview Service: The Interview Service is responsible for managing the technical interview process. It provides a real-time coding environment for candidates to write and test code, while interacting with interviewers through video and chat features.

3. Compiler Service: The Compiler Service is responsible for compiling and executing code written by candidates in the Interview Service. It supports multiple programming languages, including Java, Python, C++, and JavaScript.

4. Notification Service: The Notification Service is responsible for sending notifications to candidates and interviewers about the status of their interviews.

5. Database Service: The Database Service is responsible for managing the platform's data storage. It stores user data, interview data, and code submissions.

The platform's design patterns are based on the principles of microservices, event-driven architecture, and reactive programming. The platform uses WebRTC and WebSockets to provide real-time communication between candidates and interviewers.

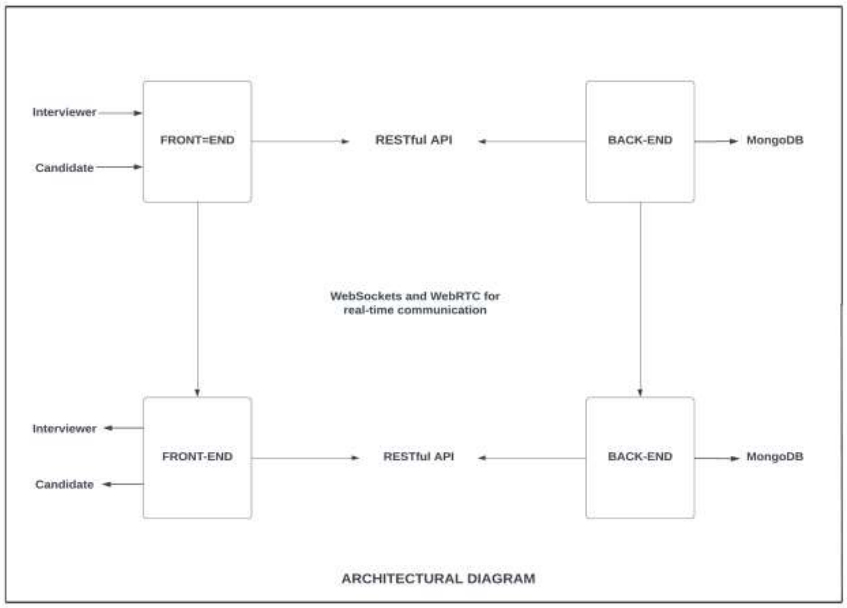

Fig. 2. High-level Architecture of the platform.

2. Technologies Used:

The platform is implemented using the following technologies:

1. Node.js: Node.js is used to implement the platform's microservices. Node.js provides a scalable and event-driven architecture that is well-suited for microservices.

2. Docker: Docker is used to containerize the platform's microservices. Containerization allows for the deployment of individual services independently and improves the platform's scalability and fault tolerance.

3. Kubernetes: Kubernetes is used to orchestrate the deployment of the platform's microservices. Kubernetes automates the deployment, scaling, and management of containerized applications.

4. WebRTC: WebRTC is used to provide real-time communication between candidates and interviewers. WebRTC is a free and open-source project that enables real-time communication capabilities in web browsers and mobile applications.

5. WebSockets: WebSockets are used to provide real-time communication between candidates and interviewers. WebSockets provide a persistent connection between the client and the server, allowing for real-time bidirectional communication.

3. Candidate Evaluation Design:

Our platform works on two principles:

1. Usability: The platform's usability is essential to its success. It should be easy to use and provide an intuitive user experience for both candidates and interviewers. A well-designed platform should be user-friendly, with clear and concise instructions, minimal distractions, and a simple and effective interface. The usability of the platform can be improved by conducting user testing and gathering feedback from users to identify areas for improvement.
2. Security: The security of the platform is crucial to protect user data and credentials. The Authentication Service should implement secure user authentication and authorization protocols, and the platform should be protected against common web application attacks such as cross-site scripting (XSS) and SQL injection. Additionally, user data and code submissions should be encrypted to prevent unauthorized access. Regular security audits should be conducted to identify and address any vulnerabilities in the platform. By prioritizing security, the platform can build trust with users and ensure the protection of sensitive data.

This system is built on a B/S architecture and includes the Web portion and a security sandbox environment as service components. A single application will often have inferior scalability, tougher resource optimisation, greater maintenance costs, etc. as the system gets more complicated. The Micro Service architectural (MSA), which promotes more adaptable, lightweight, loosely connected architectural design, is so proposed [1].

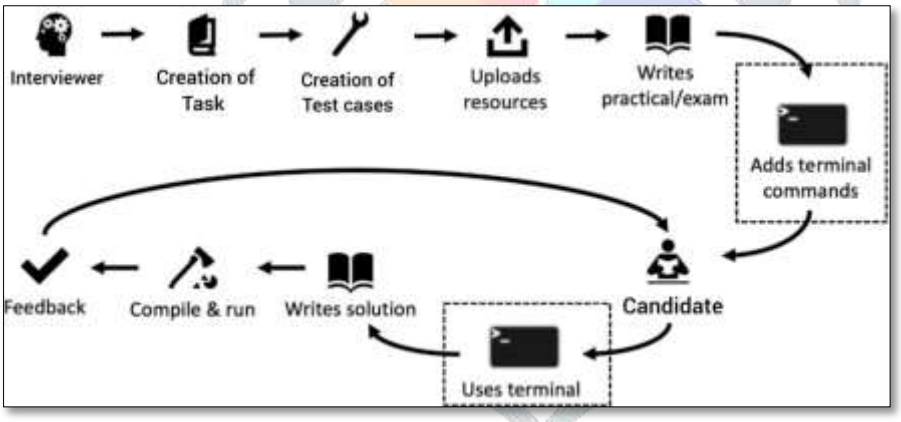

Fig. 2. Typical workflow in creating and using the platform.

4. Security:

First off, security is provided by the backend running on a different server host as an unprivileged user. [2]. achieve security in the proposed online coding interview platform, the following measures can be taken:

1. Secure user authentication and authorization: The Authentication Service component of the platform should implement secure user authentication and authorization protocols. This includes using strong password policies, implementing two-factor authentication, and using secure protocols such as HTTPS to encrypt communication between the user and the platform.
2. Protection against web application attacks: The platform should be protected against common web application attacks such as cross-site scripting (XSS) and SQL injection. This can be achieved by using input validation and sanitization, encoding data inputs, and implementing secure coding practices.
3. Data encryption: User data and code submissions should be encrypted to prevent unauthorized access. This can be achieved using encryption algorithms such as AES to protect sensitive data at rest and in transit.
4. Role-based access control: Access to the platform's resources should be restricted based on user roles. This includes limiting access to the Interview Service, Compiler Service, and Database Service based on the user's role.
5. Regular security audits: Regular security audits should be conducted to identify and address any vulnerability in the platform. This includes penetration testing, vulnerability scanning, and code reviews to ensure the platform is secure and up to date.

By implementing these measures, the proposed online coding interview platform can achieve a high level of security and protect user data and credentials.

IV. FUTURE SCOPE

The implementation of an online coding interview platform with a compiler, WebRTC, and WebSockets provides a solid foundation for future enhancements and expansions. Some potential future scopes for the platform could include:

1. Enhanced Collaboration Features: The platform can be further improved by incorporating collaborative coding features, such as allowing multiple candidates or interviewers to work together on a coding problem simultaneously. This could include features like shared code editing, collaborative debugging, and interactive whiteboarding.

2. AI-powered Evaluation and Feedback: Integrating artificial intelligence (AI) capabilities into the platform can enable automated code evaluation and provide intelligent feedback to candidates. AI algorithms can analyze code quality, identify common programming mistakes, and offer suggestions for improvement, providing more comprehensive and immediate feedback to candidates.

3. Advanced Code Execution Environments: The platform can be expanded to support a wider range of programming languages and provide advanced code execution environments, including support for frameworks, libraries, and specific development environments. This would enhance the platform's versatility and make it suitable for a broader range of technical interviews.

4. Integration with Learning Management Systems: The online coding interview platform can be integrated with learning management systems (LMS) to seamlessly connect the interview process with educational programs or coding bootcamps. This integration would allow for better tracking of candidate progress, alignment with curriculum objectives, and personalized interview preparation.

5. Analytics and Reporting: Implementing analytics and reporting capabilities would enable tracking and analysis of interview performance metrics, including success rates, time taken for each coding problem, and areas of improvement. These insights can help optimize the interview process and provide valuable data for interviewers and candidates.

6. Integration with Applicant Tracking Systems (ATS): Integrating the platform with popular Applicant Tracking Systems would streamline the recruitment workflow by seamlessly transferring interview results and candidate data. This integration would simplify the hiring process and enhance the platform's compatibility with existing HR systems.

7. Mobile Application: Developing a mobile application version of the platform would enable candidates and interviewers to conveniently access and participate in coding interviews using their mobile devices. This would enhance accessibility and provide a more flexible user experience.

V. CONCLUSION

The implementation of an online coding interview platform with a compiler, WebRTC, and WebSockets can improve the efficiency, effectiveness, and engagement of technical interviews for both candidates and interviewers. The platform's architecture and design patterns are designed.

VI. ACKNOWLEDGEMENTS

We express our deep gratitude to Prof. P. J. Chate for her invaluable moral and technical support throughout this project. Her guidance has been instrumental in shaping our research. Additionally, we would like to extend our thanks and appreciation to the Principal, Head of the Department (H.O.D.), and the entire teaching staff for their constant motivation and technical assistance, which has been crucial in the successful execution of this work.

VII. REFERENCES

[1] Wang Lei, Micro Service Architecture and Practice, Electronic Industry Press, pp. 75-101, (2015).

[2] S. Li, Y. Zhuo, S. Feng and C. Yi, The application of course-oriented online judge in "Data Structure", *2012 IEEE Symposium on Robotics and Applications (ISRA)*, pp. 687-690, (2012).

[3] Laurie Williams and Robert Kessler. *Pair Programming Illuminated*. Addison Wesley, 2002.

[4] "LeetCode," https://leetcode.com/, accessed: 2020-08-09.

[5] K. I. ZinnahApu, N. Mahmud, F. Hasan and S. H. Sagar, "P2P video conferencing system based onWebRTC,"2017.

[6] International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, pp. 557-561\

[7] CodeEval Inc.CodeEval.http://www.codeeval.com/

[8] https://www.hackerrank.com/faq/

[9] https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/