

Cloud Native Online Code Judging System

Mr. Roshan Kosare, Mr. Vaibhav Lakshane, Prof. Mr. Ravindra Kale,

Bhavesh Katare, Mr. Sahil Kolhe, Mr. Karan Amnerkar

Computer Science and Engineering

G H Raison Institute of Engineering & Technology, Nagpur, Maharashtra, India

Abstract: In the age of digitalization, technology has become an essential part of our daily lives. Cloud computing has emerged as a significant technological advancement that has transformed the way we work and store data. One of the significant applications of cloud computing is online code judging systems, which enable users to test their programming skills and improve their coding abilities. However, traditional online code judging systems have scalability, performance, and availability limitations. To address these limitations, a cloud-native approach is required. This paper proposes a cloud-native online code-judging system that provides a scalable, high-performance, and reliable platform for evaluating programming skills. The system will be designed using microservices architecture, enabling it to scale horizontally and handle a large number of requests simultaneously. The system will be deployed on a cloud platform, providing a cost-effective solution and enabling users to access the system from anywhere in the world.

Keywords: Cloud Computing, Microservices Architecture, Online Code Judging System, Scalability, Performance, Reliability

I. INTRODUCTION

Online code judging systems have become very popular in recent years, and they are widely used for evaluating programming skills. These systems are designed to provide an interactive platform to users where they can write and test their code. The system evaluates the code and provides feedback to the user, highlighting errors and suggesting improvements. These systems are usually required a web-based approach, and they require a good server with all the required resources to run the code and evaluate it. However, traditional online code judging systems have scalability, performance, and availability limitations. To address these limitations, a cloud-native approach is required.

The objective of this research is to build a system that will help students to increase their programming ability by solving coding problems that are based on core fundamentals of computer programming in the system. This system will provide all the essential tools for coding and testing the code written by the user. It will run the user's code and also test the user's code based on a question that the user is trying to solve. Based on the result shown by the system user can understand on which aspect of the skills he or she needs to work on. [1-4].

II. BACKGROUND

Cloud computing has emerged as a significant technological advancement that has transformed the way we work and store data. It has enabled users to access computing resources and services over the internet, providing a cost-effective solution to store and access data. Cloud computing has also enabled the development of cloud-native applications, which are designed to run on cloud platforms and take advantage of cloud computing's scalability, performance, and reliability. Microservices architecture is a key component of cloud-native applications, enabling them to scale horizontally and handle a large number of requests simultaneously.

Objective: The objective of this research is to develop a cloud-native online code-judging system that provides a scalable, high-performance, and reliable platform for evaluating programming skills. The system will be designed using microservices architecture, enabling it to scale horizontally and handle a large number of requests simultaneously. The system will be deployed on a cloud platform, providing a cost-effective solution and enabling users to access the system from anywhere in the world.

III. MATERIALS AND METHODS

3.1 Methodology:

The proposed system will be designed using a microservices architecture, where each service will be responsible for a specific task. The system will be deployed on a cloud platform, providing a scalable and reliable solution to users. The following are the key components of the system:[5].

User Interface: The user interface will be developed using web technologies, providing an interactive platform for users to write and test their code.

Code Execution Service: The code execution service will be responsible for running the user's code and evaluating it. It will also send feedback and the result of the user's solution. It will be designed as a stateless service, enabling it to scale horizontally and handle a large number of requests.

Problem Service: The problem service will be responsible for storing problems, test cases of problems. This problem will then be accessed by code execution service for accessing test cases of problem and by front end for showing problem statement and all other details of problem to user.

Database Service: The database service will be responsible for storing user data, such as code snippets, evaluation results, and user feedback. It will be designed as a stateful service, ensuring data consistency and durability.

Authentication Service: The authentication service will be responsible for authenticating users and managing user sessions. It will be designed as a stateless service, enabling it to scale horizontally and handle a large number of requests.

3.2 Proposed System

The core of this system is three parts: docker, React JS Frontend and Nest JS Backend with microservices architecture, which will be introduced one by one.

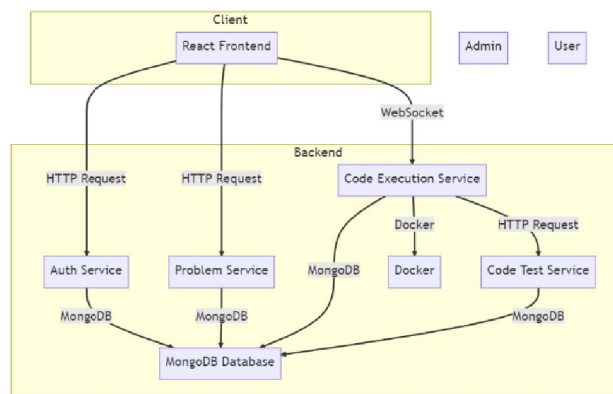


Fig 1. System design of cloud-native code judging system

3.2.1 Docker

Docker is an open-source platform that simplifies application deployment and management. It uses containerization to package applications with their dependencies, making them portable and easy to run consistently across different environments. With Docker, developers can define the application stack in a Dockerfile, ensuring reproducibility. It also offers tools like Docker Compose and Docker Swarm for managing complex systems and scaling applications. Docker Hub serves as a repository for sharing pre-built container templates, streamlining the setup of development environments. Overall, Docker provides a lightweight and efficient solution for deploying and managing applications.

In our application docker is used for running multiple instances of our services and running users-submitted code in a separate environment as users can submit malicious code that can harm our system. This technique of running untrusted code is called code sandboxing. [6-7]

3.2.2 React JS Frontend of Application

React.js, also known as React, is a popular JavaScript library used for building user interfaces (UIs) for web applications. Developed by Facebook, React allows developers to create interactive and dynamic UI components that can efficiently update and render changes to the user interface.

One of the key features of React is its component-based architecture. Developers can create reusable UI components that encapsulate their own logic and state. These components can be composed and combined to build complex user interfaces. React's virtual DOM (Document Object Model) efficiently updates and renders only the necessary changes to the UI, resulting in improved performance and responsiveness.[9]

The front end of our app consists of many different modules such as code editor, sign in, and sign up module all of this is implemented using react node library. There are many open-source packages available for the rapid development of web applications such as Axios for making HTTP requests to servers, tailwind CSS package, and react-icon package all of these are free packages that can be used by the developer for free for development the development of frontend and backend system using node.

3.2.3 Nest JS API and Microservices

Nest.js is a progressive Node.js framework used for building efficient and scalable server-side applications. It leverages modern JavaScript, TypeScript, and object-oriented programming principles to provide a robust and developer-friendly environment.

One of the key features of Nest.js is its modular architecture. It follows the module-based structure, allowing developers to organize their code into reusable and self-contained modules. This promotes code reusability, maintainability, and testability, making it easier to scale and evolve applications over time. [11-13].

There are several different microservices are there in this system such as authentication service, and code execution service all these services are developed using the nest microservice framework which makes it easy and rapid implementation of the microservice.

Nest.js also embraces dependency injection, which helps manage dependencies and promotes loose coupling between modules. It simplifies the process of integrating external services, libraries, and databases into applications, making them more flexible and extensible.

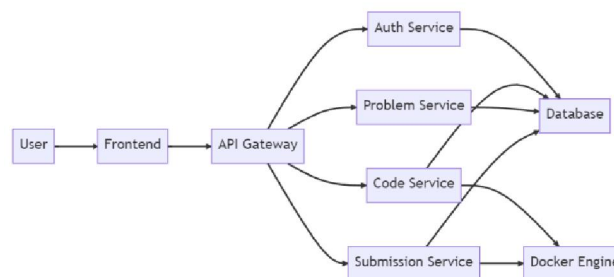


Fig 2. Flow chart of system cloud native online code judging system

IV. RELATED WORK

Several online code judging systems have been developed in recent years, such as Codechef, Codeforces, and HackerRank. These systems have provided a platform for users to test their programming skills and improve their coding abilities. However, these systems have limitations in terms of scalability, performance, and availability. To address these limitations, cloud-native approaches have been proposed in recent years, such as the cloud-based online code judging system proposed by Li et al. (2021).

V. CONCLUSION

In conclusion, the development of a cloud-native online code judging system has proven to be a highly beneficial project for the modern era of software development. By leveraging cloud technology, this system provides a flexible and scalable solution for conducting online coding competitions, assessments, and interviews.

With the use of containerization and microservices, the system can efficiently manage multiple programming languages, provide an isolated and secure runtime environment for each submission, and support automatic scaling to handle a large number of submissions concurrently. Additionally, the use of cloud-based databases and storage services ensures data persistence and easy data access from anywhere at any time.

Overall, this cloud-native online code judging system has the potential to significantly enhance the process of assessing and evaluating programming skills for individuals and organizations alike. With its flexibility, scalability, and security features, this system can facilitate fair and efficient coding competitions and interviews, which can help organizations identify the best talent in the market.

VI. RESULT

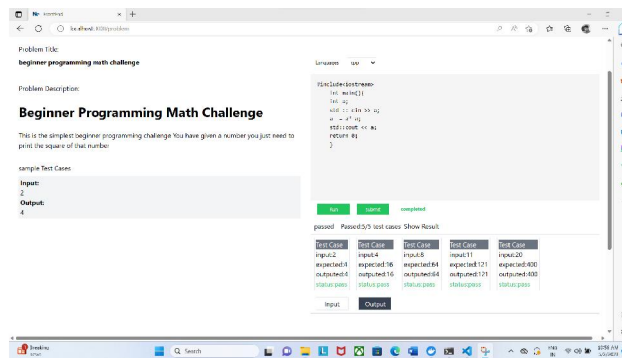


Fig 3. Output of System

REFERENCES

- [1]. Zhou, W., Pan, Y., Zhou, Y., & Sun, G. (2018, May). The framework of a new online judge system for programming education. In Proceedings of ACM turing celebration conference-China (pp. 9-14)..
- [2]. Wu, H., Liu, Y., Qiu, L., & Liu, Y. (2016, July). Online judge system and its applications in c language teaching. In 2016 International Symposium on Educational Technology (ISET) (pp. 57-60). IEEE.
- [3]. Pan, G. C., Liu, P., & Wu, J. J. (2022, June). A Cloud-Native Online Judge System. In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 1293-1298). IEEE.
- [4]. Z. W. Yang and Y. S. Zeng, "Research on ACM online evaluation system based on Linux," Computer and Modernization, no. 6, pp. 166–169,2010.
- [5]. F. S. Guan, "Research on automatic source code evaluation system based on cluster technology," Silicon Valley, no. 16, pp. 48–49, 2009
- [6]. Kan, C. (2016, January). DoCloud: An elastic cloud platform for Web applications based on Docker. In 2016 18th international conference on advanced communication technology (ICACT) (pp. 478-483). IEEE.
- [7]. Li, Y., & Xia, Y. (2016, December). Auto-scaling web applications in hybrid cloud based on docker. In 2016 5th International conference on computer science and network technology (ICCSNT) (pp. 75-79). IEEE.
- [8]. [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- [9]. Aggarwal, S. (2018). Modern web-development using reactjs. International Journal of Recent Research Aspects, 5(1), 133-137.
- [10]. <https://github.com/nestjs/nest>
- [11]. Gonzalez, D. (2016). Developing Microservices with node. js. Birmingham, UK: Packt Publishing.
- [12]. De Santis, S., Florez, L., Nguyen, D. V., & Rosa, E. (2016). Evolve the Monolith to Microservices with Java and Node. IBM Redbooks.

- [13]. Ghiya, P. (2018). TypeScript Microservices: Build, deploy, and secure Microservices using TypeScript combined with Node. js. Packt Publishing Ltd.