Coder's Battle Room Using Cloud Computing

Sushmita Hubli [1], Sachin Ghuge [2] & Dr. S. V. Shinkar [3]

[1]Student; SCTR'S Pune Institute of Computer Technology, Department of Electronics and Telecommunication, Pune, Maharashtra, India, hublisushmita7@gmail.com
[2]Student; SCTR'S Pune Institute of Computer Technology, Department of Electronics and Telecommunication, Pune, Maharashtra, India, ghugesachin0806@gmail.com
[3]Assistant Professor; SCTR'S Pune Institute of Computer Technology, Department of Electronics and Telecommunication, Pune, Maharashtra, India, svshinkar@pict.edu

Abstract:

In the ever-evolving landscape of competitive programming and coding contests, the utilization of cloud computing has emerged as a transformative force, revolutionizing the way programmers participate and compete. This survey paper, titled "Coder's Battle Room Using Cloud Computing," explores the wide-ranging applications of cloud computing in coding competitions conducted across diverse online platforms. Coding contests have gained immense popularity, offering participants opportunities to showcase their problem-solving skills and coding acumen. With the increasing scale and complexity of these contests, cloud computing has provided an essential infrastructure to support the demands of both organizers and contestants. This paper provides a comprehensive overview of the key components and functionalities of cloud-based coding battle rooms discussion, shedding light on their architecture, advantages, and prospects. By examining the state-of-the-art practices in this field, this survey offers insights into how cloud computing is shaping the competitive coding landscape, facilitating global participation, and fostering innovation in algorithmic problem-solving. As the demand for coding contests continues to grow, understanding the symbiotic relationship between coding platforms and cloud computing becomes essential for both participants and organizers. This survey serves as a valuable resource for researchers, practitioners, and enthusiasts seeking to navigate the exciting realm of coding competitions in the cloud era.
Keywords: Cloud Computing, Competitive Programming, Contest Platforms, Data Structures and Algorithms, Online Platforms, Optimal approach.

1 Introduction

The world of competitive programming has undergone a profound transformation in recent years, ushering in a new era where algorithmic prowess and coding acumen are celebrated on a global scale. Coding contests, ranging from the fiercely competitive ACM ICPC (International Collegiate Programming Contest) to online platforms like Codeforces, LeetCode, and HackerRank, have become the crucibles in which the finest programmers refine their skills and showcase their problem-solving abilities. However, what sets apart this contemporary realm of competitive coding is the infusion of cloud computing, a technological phenomenon that has fundamentally reshaped the way these contests are orchestrated, experienced, and celebrated. In the not-so-distant past, coding contests were predominantly local affairs, often held within the confines of educational institutions or tech company campuses. Participants would assemble in physical spaces, armed with their trusty laptops, to engage in timed battles of logic and ingenuity. While these contests served as valuable breeding grounds for talent, they had inherent limitations-geographical constraints, limited computational resources, and logistical complexities, to name a few. Enter the era of serverless cloud computing a paradigm shift that has transcended boundaries and unlocked unprecedented potential in the world of coding competitions. Cloud computing involves accessing and utilizing various IT resources such as compute power, storage, databases, applications, and more via the Internet on a pay-as-you-go basis, allowing for on-demand consumption as needed [1]. In this survey paper, we embark on an insightful journey into the convergence of competitive coding discussion and cloud computing, illuminating the transformative impact this partnership has had on the coding contest landscape. This study looks at online discussions from both static and dynamic angles. The outcomes show the features of group knowledge creation process online discussion in a blended learning environment. The benefits and drawbacks of this study can be utilized to raise the process of group knowledge building's efficacy [12]. Our journey in this survey paper begins by exploring the architectural underpinnings of the Coder's Battle Room. We dissect the various components and technologies that collaborate to provide participants with a seamless and exhilarating coding question discussion experience. From virtual machines and containerization to distributed computing clusters, the infrastructure supporting these digital arenas is a testament to the power and versatility of cloud computing. However, the allure of cloud-powered coding contests extends far beyond the mere convenience of virtual battlefields. It encompasses a multitude of advantages that have redefined the very nature of competitive programming. Universal accessibility is chief among them, as participants from diverse corners of the globe can engage in the same contest, levelling the playing field and fostering cross-cultural collaboration. Cost-effectiveness is another critical facet, as cloud resources are procured on-demand, eliminating the need for expensive on-premises hardware and infrastructure maintenance [21]. To truly understand the magnitude of this transformation, we will undertake an extensive exploration of the state-of-the-art platforms and technologies that are pushing the boundaries of what is possible in cloud-based coding contests. We shall embark on comparative analyses of these platforms, offering a critical assessment of their features, user experiences, and the innovations they bring to the table. Moreover, this survey paper will cast its gaze into the future. By discerning emerging trends and technological advancements, we aim to provide a glimpse of the evolving landscape of coding competitions in the cloud era. What new paradigms and tools will shape the contests of tomorrow? What research opportunities lie on the horizon? These are questions we shall strive to answer. In summation, this survey serves as a comprehensive guide for researchers, practitioners, educators, and coding enthusiasts who seek to navigate the captivating realm of coding contests as it stands today - a realm where cloud computing breathes new life into age-old competitions, redefining what is possible and paving the way for an exciting future of collaborative coding excellence on a global scale. Let us embark on this journey into the heart of the "Coder's Battle Room," where code becomes art, and the cloud empowers dreams.

2 Literature Survey

2.1 Evolution of Cloud Computing:

Cloud computing has progressed through several stages to meet varying needs. Early models such as Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) provided users with a broad range of control and flexibility. IaaS offered virtualized computing resources over the internet, allowing users to manage and configure their own virtual machines. PaaS, on the other hand, abstracted some of the infrastructure management, enabling developers to focus more on application development. However, these models often required significant management overhead and could lack flexibility for dynamically changing workloads. The advent of serverless computing marked a significant shift. Introduced with AWS Lambda in 2014, serverless computing provides a model where developers write code without worrying about the underlying infrastructure. Instead, resources are automatically scaled based on demand, and users are billed only for the compute time consumed by their functions. This shift promised greater efficiency and cost savings, making it particularly attractive for applications with highly variable workloads.

2.2 Serverless in Competitive Coding Platforms:

In the context of competitive coding platforms, serverless computing has shown promise due to its ability to handle unpredictable workloads without the need for pre-provisioned resources. Competitive coding platforms often experience spikes in activity during contests or high traffic periods, and serverless models are well-suited to scale dynamically in response to such spikes. Research has demonstrated that serverless architectures can manage these fluctuations effectively by automatically provisioning resources as needed. However, there are noted limitations, such as cold start latency, which can affect the responsiveness of the platform, and the constraints on execution time which may not be suitable for tasks requiring long processing times. Despite these challenges, serverless computing's benefits in cost efficiency and scalability make it a compelling choice for many applications in this domain.

2.3 Comparative Studies:

Comparative studies highlight that serverless computing generally offers a cost-effective and efficient model, particularly for applications with unpredictable or highly variable demand. Serverless models operate on a pay-as-you-go basis, which can significantly reduce costs compared to maintaining a fixed set of servers.

This model is advantageous for applications that experience fluctuating workloads, as it eliminates the need for over-provisioning and ensures that users only pay for the resources they actually use. Nevertheless, ongoing challenges include security concerns, such as the need for robust mechanisms to protect function code and data, and the requirement for better management tools to monitor and control serverless environments effectively.

Despite these challenges, the evolution of serverless computing continues to drive innovation and efficiency, presenting new opportunities and considerations for developers and organizations.

3 Proposed Methods

AWS Lambda, Azure Functions, and Google Cloud Functions are serverless computing services provided by the respective cloud platforms. They allow developers to build, deploy, and run applications without managing the underlying infrastructure. Here's a detailed explanation of each:

A. AWS Lambda

   AWS Lambda is offered by Amazon Web Services (AWS). It is a cloud service that lets developers run code without having to set up or manage any servers. In a serverless architecture, developers can focus solely on writing and deploying their functions, leaving the infrastructure, scaling, and maintenance tasks to AWS. Lambda functions are event-driven and can be triggered by various AWS services or HTTP requests.

   They are written in languages like Python, Node.js, Java, and others, and can execute code in response to events such as changes in data, file uploads, or scheduled tasks. Lambda functions are stateless, meaning they don't retain information between executions, which promotes scalability and simplifies development. Developers pay only for the compute time consumed by their functions, making Lambda a cost-effective solution for running code in response to specific events.

   Additionally, Lambda integrates seamlessly with other AWS services, enabling the creation of sophisticated, serverless architectures. Overall, AWS Lambda empowers developers to build scalable and efficient applications without the need to manage traditional server infrastructure.

B. Azure Functions

   Azure Functions is a cloud service from Microsoft Azure that allows developers to build, launch, and expand apps without having to manage any servers. Function-as-a-Service (FaaS) is the core concept, allowing developers to focus solely on code logic rather than worrying about the underlying infrastructure.

   Azure Functions provides support for multiple programming languages, such as C#, JavaScript, Python, and Java, giving developers a high degree of flexibility. Events like HTTP requests, database updates, scheduled timers, or messages from Azure services such as Azure Queue Storage or Azure Service Bus can trigger functions within Azure.

   This event-driven model enables the creation of highly responsive and scalable applications, where functions are executed on-demand in response to specific events. Azure Functions are crafted to operate without retaining state and seamlessly integrate with additional Azure services like Azure Storage, Cosmos Database, and Logic Applications. This seamless integration allows developers to build complex workflows and applications using a combination of serverless components.

The pay-as-you-go pricing model of Azure Functions ensures cost-effectiveness, as users are billed only for the resources consumed during the execution of functions. Additionally, Azure Functions provides robust monitoring and logging capabilities, allowing developers to track and troubleshoot issues efficiently.

The development experience is further enhanced with features like local debugging, continuous integration and deployment (CI/CD) support, and integration with popular development tools such as Visual Studio Code and Azure DevOps.

With its simplicity, scalability, and event-driven architecture, Azure Functions empowers developers to focus on writing code that delivers business value without the burden of managing infrastructure.

C. Google Cloud Functions

Google Cloud Functions is a cloud service from Google Cloud Platform (GCP) that lets developers create and launch functions that run automatically when triggered by events, and can scale up or down based on demand.

This serverless architecture eliminates the need for developers to manage infrastructure, enabling them to focus on writing code. Cloud Functions offers support for a variety of programming languages, such as Node.js, Python, Go, and Java, giving developers the flexibility to select their preferred language.

Developers can create functions triggered by various GCP services, such as Cloud Storage, Cloud Pub/Sub, and HTTP requests. These functions execute in lightweight, stateless containers, ensuring quick startup times and efficient resource utilization. Additionally, Cloud Functions integrates seamlessly with other GCP services, allowing developers to build complex workflows and applications by connecting different components of the Google Cloud ecosystem.

The pay-as-you-go pricing model ensures cost efficiency, as users only pay for the actual compute resources consumed during function execution. Monitoring and logging are built into Cloud Functions, enabling developers to gain insights into function performance, troubleshoot issues, and ensure reliability.

Launching is made easier using the gcloud command-line tool or through continuous integration/continuous deployment (CI/CD) pipelines.. Overall, Google Cloud Functions provides a scalable and efficient solution for building serverless applications, allowing developers to focus on writing code without the overhead of managing infrastructure.

Its integration with other GCP services and support for multiple programming languages make it a versatile choice for a wide range of applications and use cases.

4 Results & Discussion

4.1 Work Overview

In the "Coder's Battle Room using Cloud Computing" paper work, MongoDB Atlas Cloud was chosen as the database platform to store and manage information.

The focus of this image is on the structure and content of the database table for the questions. MongoDB is a NoSQL database system known for its flexibility and scalability, making it an ideal choice for cloud-based applications.

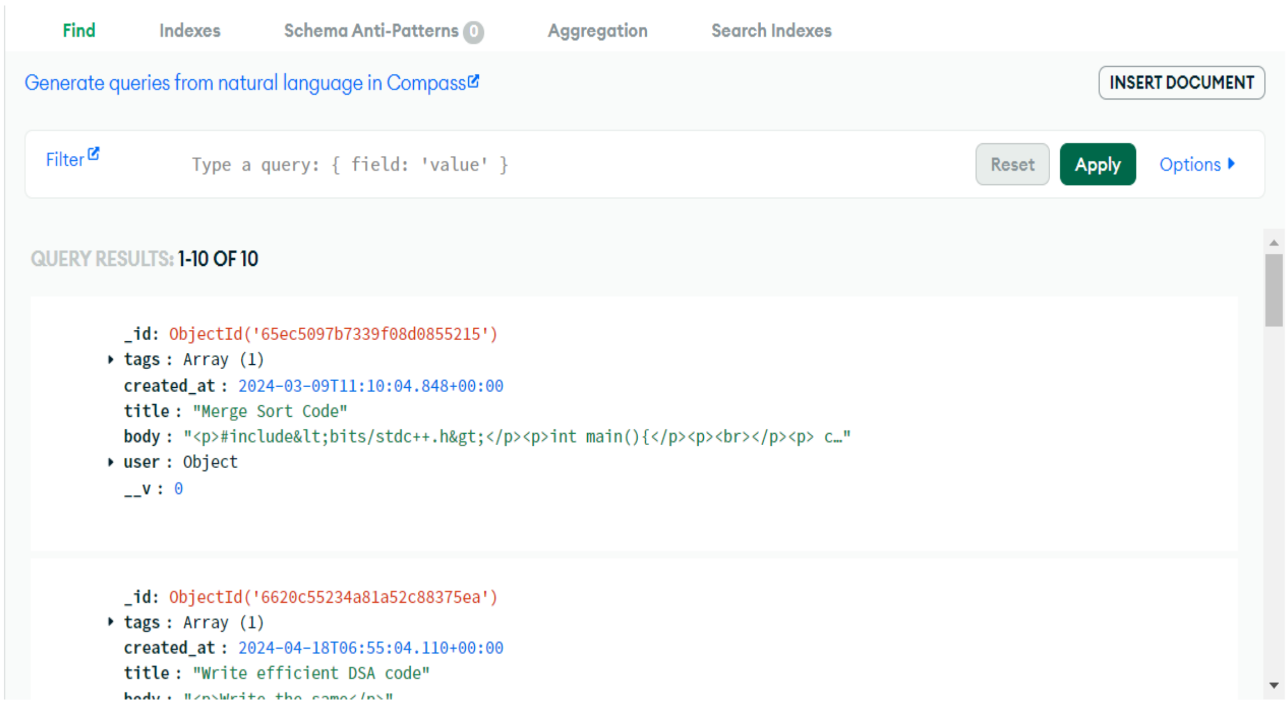

Fig. 1 Database content of Question table

The "Coder's Battle Room" work stores questions in MongoDB Atlas Cloud, using a flexible structure with key fields like "_id" "tags," "created_at," "title," "body," and "user" The "user" field establishes a link to the user collection, enabling connections between questions and their authors.

Indexes on fields like "_id" and "tags" optimize query performance, ensuring fast retrieval. Data management and security are addressed through role-based access controls and regular backups. MongoDB's scalability and flexibility support the platform's growth, allowing it to adapt to a larger user base without compromising performance. This setup provides a solid foundation for a reliable and scalable DSA discussion platform.

We have also created additional tables to support various interactions on the platform. The "comments" table stores comments that users post on specific questions submitted by others, allowing for deeper engagement and discussion.

The "answers" table is designed to capture user responses to questions, with each answer linked to a specific question ID for easy retrieval and organization. Similarly, the "posts" table holds information about articles and other content that individual users contribute to the platform, enabling a broader scope of user-generated content.

These tables work together to foster a robust and dynamic environment for discussion, feedback, and content sharing within the "Coder's Battle Room."

The work has resulted in the development of a fully functional platform where users can engage in discussions, post questions, provide answers, publish articles, and collaborate on problem-solving.

The platform's intuitive interface and comprehensive feature set cater to the diverse needs of users, enabling seamless interaction and knowledge sharing.

The platform can make a significant impact on the programming community by democratizing access to knowledge, fostering collaboration, and empowering individuals to improve their problem-solving skills and advance their careers.

By providing a supportive and inclusive environment for learning and growth, the platform has contributed to the professional development of countless users.

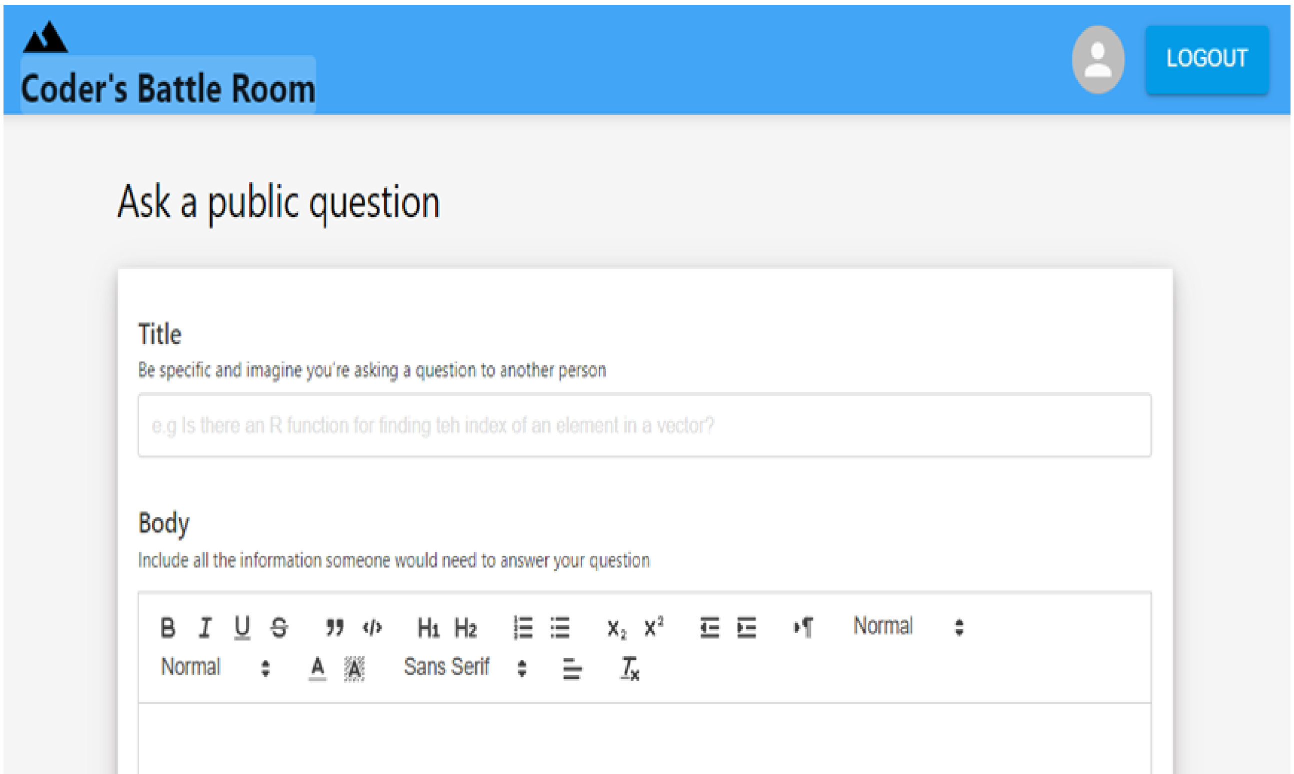

Fig. 2 Question Post Page

The above image shows the Question posting page of the platform, where users can ask questions and seek help from the community. Users can enter a clear title summarizing their question, provide details in the body section, and add relevant tags to categorize their query.

This setup ensures that questions are easily understandable and discoverable by others with similar interests or expertise. Overall, the Question posting page aims to simplify the process of seeking assistance and foster meaningful interactions among users within the platform.

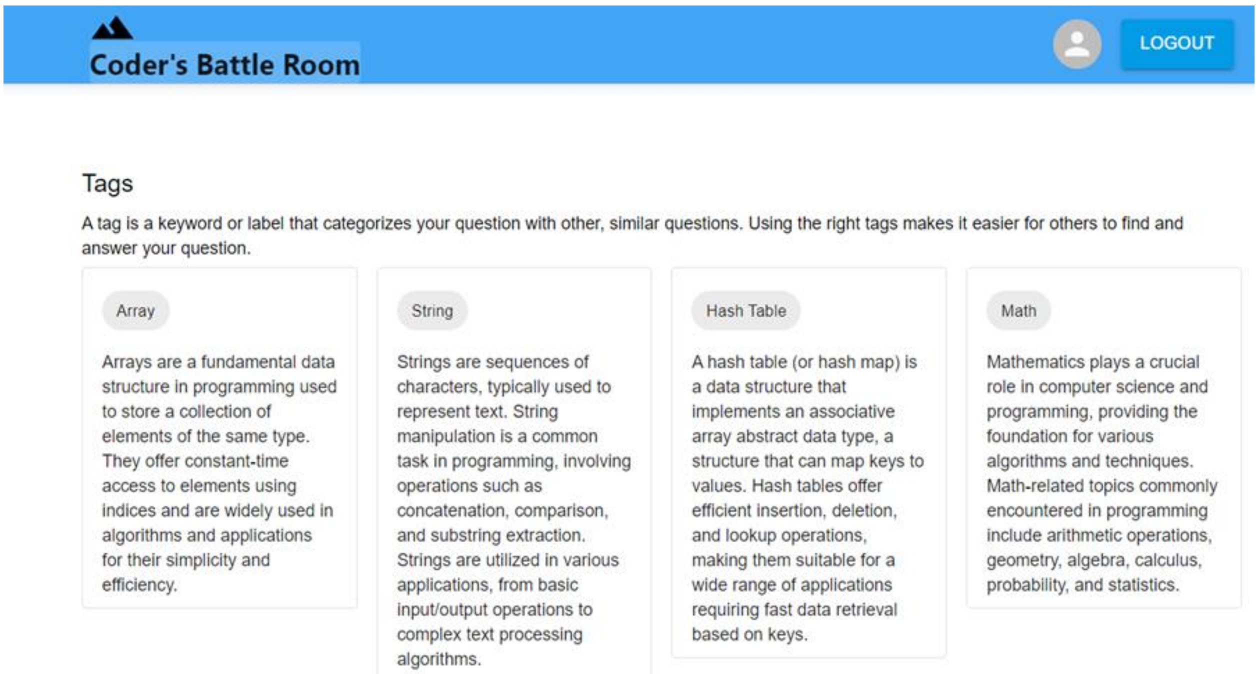

Fig. 3 Tags Information Page

The above tags page of the platform displays a list of tags used to categorize questions and topics discussed within the community. Users can explore various tags to find questions related to specific subjects or topics of interest.

By organizing questions into distinct categories, the tags page helps users navigate the platform efficiently and discover relevant content aligned with their interests and expertise. Overall, the tags page serves as a valuable resource for users to explore, engage, and contribute to discussions within the platform.

The articles page of the platform showcases a collection of diverse articles written by users on various topics related to data structures, algorithms, and competitive programming.

Users can explore these articles to gain insights, learn new concepts, and stay updated on the latest trends and developments in the field. Each article offers a unique perspective or tutorial, covering a range of subjects such as algorithm optimization techniques, programming language tips, and problem-solving strategies.

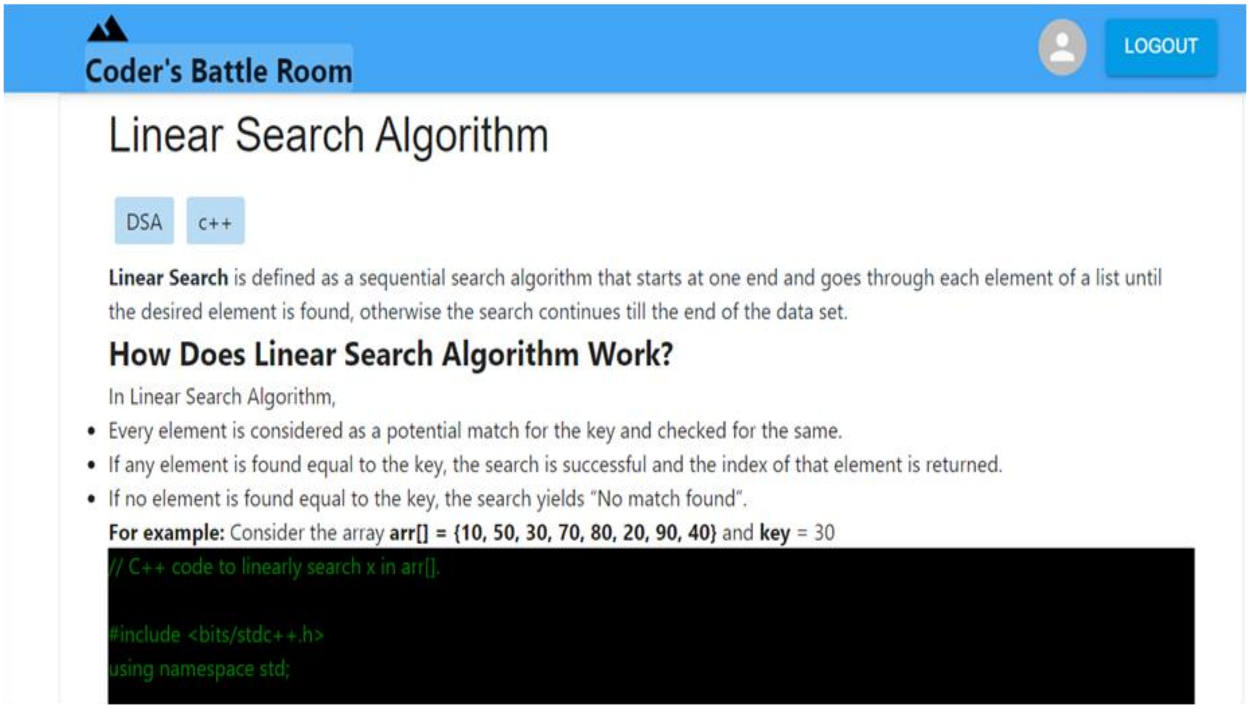

Fig. 4 Articles Page

By providing a platform for users to share their expertise and experiences through articles, the platform fosters knowledge sharing and collaboration among members of the programming community. Overall, the articles page serves as a valuable repository of educational resources and insights, enriching the learning experience for users and contributing to the collective knowledge base of the platform.

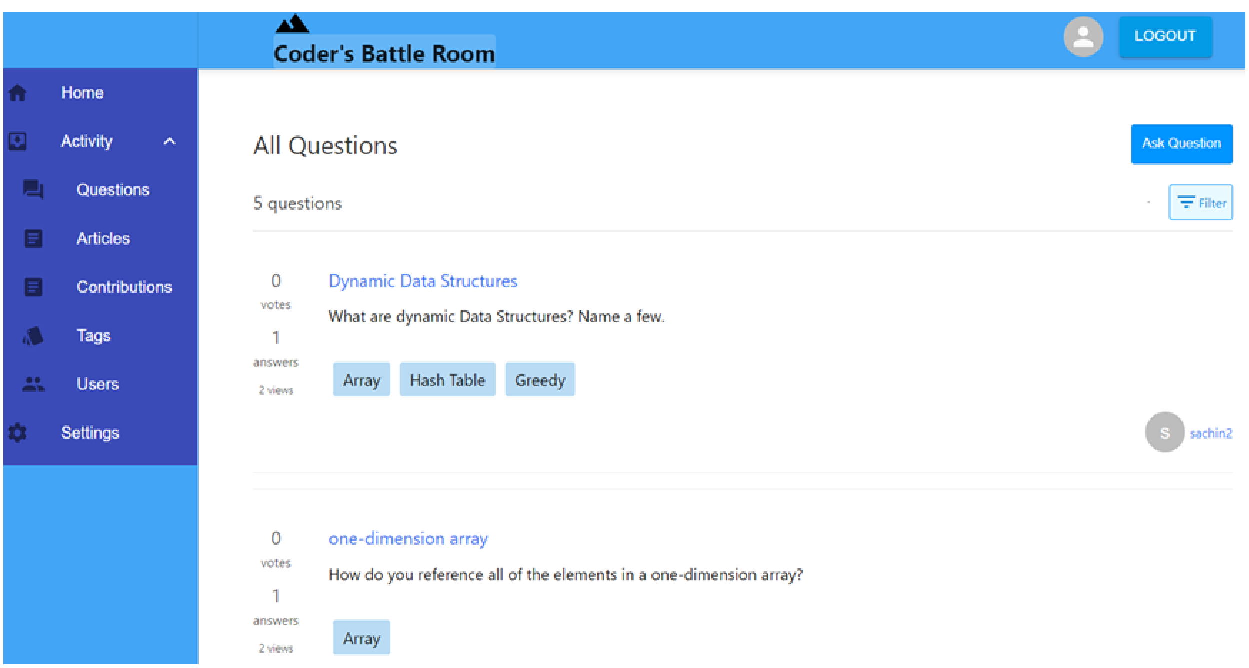

Fig. 5 Home Page

The home page of the platform serves as the central hub where users can explore a curated selection of questions posted by the community. Alongside these questions, users have access to convenient navigation options, including settings, user contributions, and articles. The user contributions section provides users with insights into their own contributions, including questions asked, answers provided, and articles published, fostering a sense of ownership and engagement within the community. Additionally, the articles section offers users a gateway to explore a wealth of educational resources and insights shared by fellow users, enriching their learning journey. Overall, the home page serves as a welcoming and informative entry point to the platform, empowering users to engage, contribute, and learn within the vibrant programming community.

4.2 Equations

Cloud computing involves various mathematical concepts and equations, primarily related to the areas of resource allocation, scalability, and performance optimization. Here are some mathematical equations and concepts commonly used in cloud computing:

• Elasticity refers to the ability of a cloud system to adapt to workload changes. The elasticity equation can be represented as:

$$\textbf{Elasticity} = \textbf{Change in Time}/\textbf{Change in Quantity} \quad \textbf{(1)}$$

• Scalability measures the ability of a system to handle an increasing amount of load or demand. It can be expressed as:

$$\textbf{Scalability} = \textbf{Change in Resources}/\textbf{Change in Performance} \quad \textbf{(2)}$$

• Resource utilization is the ratio of the actual usage of resources to the total available resources. It can be calculated using the formula:

Resource Utilization = (Total Available Resources/Actual Resource Usage) X 100% **(3)**

• Cloud computing involves costs, and optimizing these costs is crucial. The cost optimization equation can be represented as:

Cost Optimization=Cost/Performance **(4)**

• Service Level Agreement compliance is essential to ensure that the cloud service meets the agreed-upon performance metrics. It can be calculated as:

$$\textbf{SLA Compliance} = (\textbf{Total Number of Requests / Number of SLA Compliant Requests}) \times \textbf{100}\% \quad \textbf{(5)}$$

• Response time is a critical performance metric. It can be calculated using Little's Law:

Response Time = Request Arrival Rate / Number of Requests **(6)**

• Availability is a key metric for assessing the reliability of a cloud service. It is frequently represented as a percentage and determined by the following calculation:

$$Availability = (Total\ Time\ /\ Total\ Uptime) \times 100\% \tag{7}$$

• Throughput measures the amount of data transferred successfully over the network in a given period. It can be calculated as:

$$Throughput = Total\ Time\ /\ Total\ Data\ Transferred \tag{8}$$

4.3 Figures

AWS Lambda utilizes different cloud services available within the AWS ecosystem. Lambda functions can be triggered by various activities, including image uploads, in-app actions, website interactions, data from connected devices, or custom requests. Amazon Web Services has emphasized AWS Lambda's goal of eliminating the need for provisioning or managing virtual servers and its ability to automatically scale across multiple geographic regions (Availability Zones).

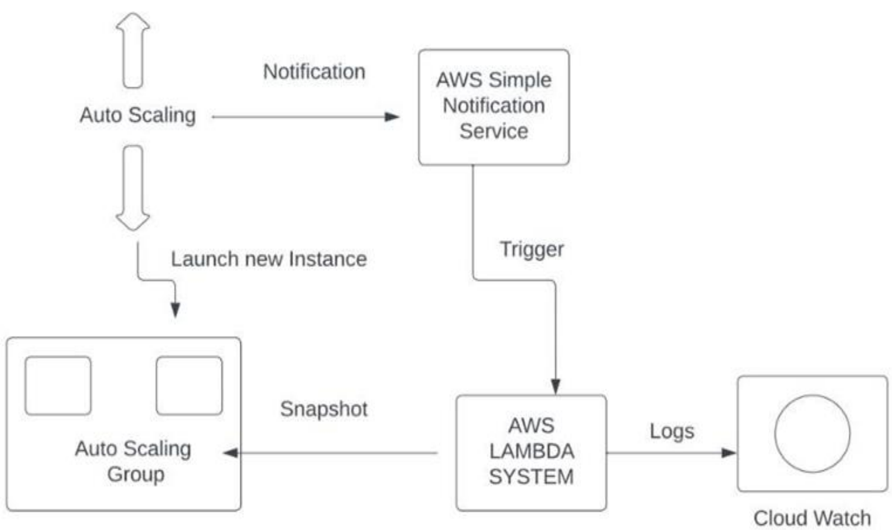

Fig. 6 AWS Lambda Demonstration

Amazon Web Services has highlighted numerous applications for AWS Lambda, including tasks such as real-time file processing, stream processing, Extract, Transform, Load (ETL) operations, and the creation of serverless backends for IoT, mobile, and web applications are supported.

Notable users of AWS Lambda include renowned companies like Netflix, SPS Commerce, Earth Network, Vidroll, Localytics, Seattle Times, Zillow, Bustle, and Major League Baseball Advanced Media.

These companies leverage serverless computing for diverse purposes including transcoding, monitoring, disaster recovery, compliance, data processing, sensor data analysis, real-time ad bidding, stream analytics, image resizing, mobile metrics tracking, and backend operations for mobile and web applications [8].

In the realm of serverless computing, cloud service providers take on the responsibility of managing the data center, servers, and runtime environment. Unlike other cloud models, serverless computing shifts more of the management and maintenance burden onto the cloud service provider, thereby relieving developers of these complexities [3].

Key characteristics of serverless computing include autoscaling, where functions scale automatically based on demand, allowing applications to handle a high volume of requests efficiently. Additionally, developers benefit from fast and independent deployment, as they can deploy or redeploy individual serverless functions without impacting the entire system. This level of independence in deployment surpasses that of microservices. Furthermore, serverless computing eliminates infrastructural costs associated with traditional cloud services, as companies only pay for the resources used rather than maintaining generously large machines to ensure availability [1][2][3].

Fig. 7(a) Contrast between Serverless Cloud Computing and Traditional Server-Based Approach

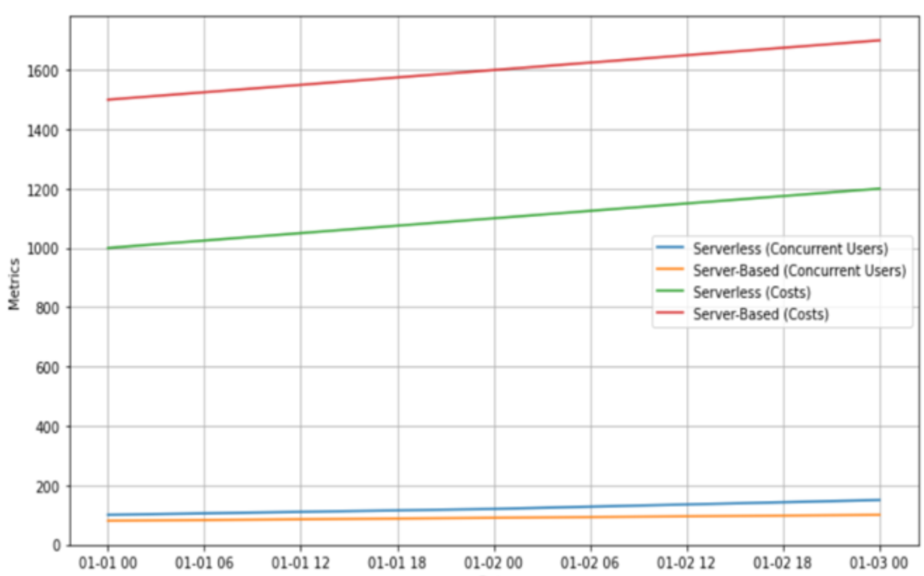

Fig. 7(b) Contrast between Serverless Cloud Computing and Traditional Server-Based Approach

Serverless cloud computing and server-based approaches offer distinct advantages and considerations across several factors. Regarding concurrent users, serverless computing typically excels due to its automatic scaling capabilities, efficiently managing fluctuating workloads without manual intervention. This scalability also translates into cost-effectiveness, as serverless models often operate on a pay-as-you-go basis, minimizing expenses during periods of low traffic compared to maintaining dedicated servers. However, while serverless computing reduces upfront infrastructure costs, it may lead to higher operational costs depending on usage patterns [1]. On the other hand, server-based approaches may involve greater management overhead, requiring manual provisioning, monitoring, and maintenance of servers. In contrast, serverless platforms abstract much of this management burden, allowing developers to prioritize the application's logic. Ultimately, the decision between serverless and server-based approaches depends on the specific requirements, workload characteristics, and budget considerations of each project [13][25].

5 State-of-the-Art Comparison

In this section, we compare key cloud computing models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Serverless Computing (FaaS)—to understand their suitability for dynamic applications like competitive coding platforms.

- **Performance & Scalability:** IaaS requires manual scaling but offers full control. PaaS has built-in scaling, but may struggle with high variability. Serverless (FaaS) provides automatic scaling, making it ideal for fluctuating workloads.
- **Cost Efficiency:** IaaS can be costly due to the need to maintain idle resources. PaaS reduces management costs but may still be expensive with fluctuating demand. Serverless is cost-effective for variable workloads, charging only for actual usage.
- **Flexibility & Ease of Use:** IaaS offers maximum flexibility but at the cost of complexity. PaaS simplifies management but limits customization. Serverless is the easiest to use, with no infrastructure management required, though it may be less flexible for long-running processes.
- **Security & Compliance:** IaaS provides strong security but requires user management. PaaS includes built-in security features but offers less control. Serverless abstracts security, which is managed by the provider, potentially complicating compliance.

Summary: Serverless computing emerges as the most suitable for dynamic, real-time applications due to its scalability, cost efficiency, and ease of use, though the choice ultimately depends on specific application requirements.

6 Conclusion

In this thorough examination, we have explored the utilization of cloud computing to enhance the speed, scalability, and reliability of websites. The integration of cloud computing into the competitive coding arena through a specialized problem discussion platform is not just a novel concept; it represents a strategic advancement that enhances the coding experience. This integration empowers a global community of coding enthusiasts to collaborate, learn, and excel, thereby fostering progress and excellence in the field. Looking ahead, the ongoing development of this platform holds the promise of fostering collaboration and innovation in the continuously expanding domain of competitive coding.

Throughout our exploration, we have embarked on a journey to uncover the transformative potential of a cloud-powered problem discussion platform within the realm of competitive coding. We commenced by recognizing the dynamic nature of coding competitions, which draw participants from diverse regions, highlighting the crucial necessity for a centralized platform dedicated to collaborative problem-solving. As we bring this discussion to a close, we reflect on the key insights gained and underscore the significance of this innovative approach.

Acknowledgements

We would want to convey our heartfelt gratitude to Dr. S. V. Shinkar our mentor, for his valuable advice and assistance in completing our research work. We would also like to thank all the other supporting personnel who assisted us by supplying the equipment that was essential and vital, without which we would not have been able to perform efficiently on this research work. We would also like to thank the University for accepting our research work in our desired field of expertise. We'd also like to thank our friends and parents for their support and encouragement as we worked on this assignment.

References

[1] "Serverless Architecture - A Revolution in Cloud Computing," by R. A. P. Rajan ,2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 2018, pp. 88-93, doi: 10.1109/ICoAC44903.2018.8939081.

[2] N. Mahmoudi and H. Khazaei, "Performance Modeling of Metric-Based Serverless Computing Platforms," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1899-1910, 1 April-June 2023, doi: 10.1109/TCC.2022.3169619.

[3] D. Taibi, K. Wawruch and J. Spillner , "Serverless Computing-Where Are We Now, and Where Are We Heading?" in IEEE Software, vol. 38, no. 1, pp. 25-31, Jan.-Feb. 2021, doi: 10.1109/MS.2020.3028708.

[4] S. L. Vadlamani and O. Baysal, "Studying Software Developer Expertise and Contributions in Stack Overflow and GitHub," 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), Adelaide, SA, Australia, 2020, pp. 312-323, doi: 10.1109/ICSME46990.2020.00038.

[5] G. Juve et al., "Scientific workflow applications on Amazon EC2," 2009 5th IEEE International Conference on E-Science Workshops, Oxford, UK, 2009, pp. 59-66, doi: 10.1109/ESCIW.2009.5408002.

[6] Y. Xia, Z. Mi and M. Wu, "A Survey on Serverless Computing and Its Implications for JointCloud Computing," 2020 IEEE International Conference on Joint Cloud Computing, Oxford, UK, 2020, pp. 94-101, doi: 10.1109/JCC49151.2020.00023.

[7] W. O'Meara and R. G. Lennon, "Serverless Computing Security: Protecting Application Logic," 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland, 2020, pp. 1-5, doi: 10.1109/ISSC49989.2020.9180214.

[8] T. Lynn, P. Rosati, A. Lejeune and V. Emeakaroha, "A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms," 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Hong Kong, China, 2017, pp. 162-169, doi: 10.1109/CloudCom.2017.15.

[9] G. McGrath, J. Short, S. Ennis, B. Judson and P. Brenner, "Cloud Event Programming Paradigms: Applications and Analysis," 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2016, pp. 400-406, doi: 10.1109/CLOUD.2016.0060.

[10] S. S. Shah, T. S. Bavaskar, S. S. Ukhale, R. A. Patil and A. S. Kalyankar, "Answer Ranking in Community Question Answer (QA) System and Questions Recommendation," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697396.

[11] "Exploring Constructive Learning Activity in Online Programming Discussion Forums," by I. H. Hsiao, 2015 IEEE 15th International Conference on Advanced Learning Technologies, Hualien, Taiwan, 2015, pp. 223-227, doi: 10.1109/ICALT.2015.59.

[12] J. Zhao and Y. Jiang, "Categories of questions in an online discussion forum: An analysis," 2010 5th International Conference on Computer Science & Education, Hefei, China, 2010, pp. 428-431, doi: 10.1109/ICCSE.2010.5593591.

[13] D. Kelly, F. Glavin and E. Barrett, "Serverless Computing: Behind the Scenes of Major Platforms," 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), Beijing, China, 2020, pp. 304-312, doi: 10.1109/CLOUD49709.2020.00050.

[14] S. Neela, Y. Neyyala, V. Pendem, K. Peryala and V. V. Kumar, "Cloud Computing Based Learning Web Application Through Amazon Web Services," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 472-475, doi: 10.1109/ICACCS51430.2021.9441974.

[15] H. Khatter and B. M. Kalra, "A new approach to blog information searching and curating," 2012 CSI Sixth International Conference on Software Engineering (CONSEG), Indore, India, 2012, pp. 1-6, doi: 10.1109/CONSEG.2012.6349518.

[16] Sajay K R and S. S. Babu, "A study of cloud computing environments for High Performance applications," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, India, 2016, pp. 353-359, doi: 10.1109/SAPIENCE.2016.7684127.

[17] X. Du, Z. He, H. Li and X. Zhu, "A study on evaluation of online discussion forums in cloud learning environment," 2016 International Conference on Progress in Informatics and Computing (PIC), Shanghai, China, 2016, pp. 711-715, doi: 10.1109/PIC.2016.7949591.

[18] H. Zhang, S. Wang, T. -H. Chen, Y. Zou and A. E. Hassan, "An Empirical Study of Obsolete Answers on Stack Overflow," in IEEE Transactions on Software Engineering, vol. 47, no. 4, pp. 850-862, 1 April 2021, doi: 10.1109/TSE.2019.2906315.

[19] Y. Li, S. Wang and T. N. Nguyen, "An Empirical Study on the Characteristics of QuestionAnswering Process on Developer Forums," 2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Seoul, Korea (South), 2020, pp. 318-319.

[20] "A Scheme of Answer Selection In Community Question Answering Using Machine Learning Techniques" by P. Kulkarni and M. Wakchaure, 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 879-883, doi: 10.1109/ICCS45141.2019.9065834.

[21] E. Pinho, L. B. Silva and C. Costa, "A cloud service integration platform for web applications," 2014 International Conference on High Performance Computing & Simulation (HPCS), Bologna, Italy, 2014, pp. 366-373, doi: 10.1109/HPCSim.2014.6903709.

[22] W. Lloyd, M. Vu, B. Zhang, O. David and G. Leavesley, "Improving Application Migration to Serverless Computing Platforms: Latency Mitigation with Keep-Alive Workloads," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, Switzerland, 2018, pp. 195-200, doi: 10.1109/UCC-Companion.2018.00056.

[23] Y. Tang and J. Yang, "Lambdata: Optimizing Serverless Computing by Making Data Intents Explicit," 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), Beijing, China, 2020, pp. 294-303, doi: 10.1109/CLOUD49709.2020.00049.

[24] S. Tripathi and T. Saini, "Predicting tags for the stack overflow questions using different classifiers," 2018 4th International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 2018, pp. 1-5, doi: 10.1109/RAIT.2018.8389059.

[25] B. Balis and M. Malawski, "Serverless Computing for Scientific Applications," in IEEE Internet Computing, vol. 26, no. 4, pp. 53-58, 1 July-Aug. 2022, doi: 10.1109/MIC.2022.3168810.

[26] J. Zhu, M. Gordon, J. Warner, J. White, R. Zanelatto and P. J. Guo, "Towards a domain specific visual discussion forum for learning computer programming: An empirical study of a popular MOOC forum," 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Atlanta, GA, USA, 2015, pp. 101-109, doi: 10.1109/VLHCC.2015.7357204.

[27] D. Filgueira Mendonca and L. Baresi, "Toward a Serverless Platform for Edge Computing," 2019 IEEE International Conference on Fog Computing (ICFC), Prague, Czech Republic, 2019, pp. 110, doi: 10.1109/ICFC.2019.00008.

[28] L. Ponzanelli, A. Mocci, A. Bacchelli and M. Lanza, "Understanding and Classifying the Quality of Technical Forum Questions," 2014 14th International Conference on Quality Software, Allen, TX, USA, 2014, pp. 343-352, doi: 10.1109/QSIC.2014.27.

[29] H. K. Dam, M. Choetkiertikul, D. Avery, T. Tran and A. Ghose, "Who Will Answer My Questions on Stack Overflow?" 2015 24th Australasian Software Engineering Conference, Adelaide, SA, Australia, 2015, pp. 155-164, doi: 10.1109/ASWEC.2015.28.

[30] C. Sutton and M. Allamanis, "Why, when, and what: Analyzing Stack Overflow questions by topic, type, and code," 2013 10th Working Conference on Mining Software Repositories (MSR), San Francisco, CA, USA, 2013, pp. 53-56, doi: 10.1109/MSR.2013.6624004.

[31] M. Sewak and S. Singh, "Winning in the Era of Serverless Computing and Function as a Service," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 2018, pp. 1-5, doi: 10.1109/I2CT.2018.8529465.

[32] M. Štula and H. Božiković, "Website design — Past, present and future," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2018, pp. 1476-1481, doi: 10.23919/MIPRO.2018.8400266.

[33] M. Štula and H. Božiković, "Web design — Past, present and future," 2018 41st International Conventions on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2018, pp. 1476-1481, doi: 10.23919/MIPRO.2018.8400266.

[34] M. H. L. B. BRaz, S. W. M. Siqueira and R. N. Melo, "Website technology for education and training," 14th International Workshop on Database and Expert Systems Applications, 2003. Proceedings., Prague, Czech Republic, 2003, pp. 337-341, doi: 10.1109/DEXA.2003.1232045.

[35] V. Okanovic, "Web application development with component frameworks," 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2014, pp. 889-892, doi: 10.1109/MIPRO.2014.6859693.