International Journal of Innovative Technology and Interdisciplinary Sciences
https://journals.tultech.eu/index.php/ijitis
ISSN: 2613-7305
Volume 7, Issue 2
DOI: https://doi.org/10.15157/ijitis.2024.7.2.68-79
Received: 13.07.2024; Revised: 25.08.2024; Accepted: 11.09.2024

Technical Paper

Continuous Deployment in Action: Developing a Cloud-Based Image Matching Game

Balamuralikrishna Thati[1], Katragadda Megha Shyam[2], S Sindhura[3], P Dedeepya[4]*, N Sampreet Chowdary[5]

[1]Department of CSE, Dhanekula, Institute of Engineering & Technology, Ganguru, Vijayawada, India
[2]Department of CSE, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, India
[3]CSE, NRI Institute of Technology, Agiripalli, India
[4]Computer Science & Engineering, PVP Siddhartha Institute of Technology, Vijayawada, India
[5] Department of CSE, P.V.P. Siddhartha Institute of Technology, Vijayawada, India
***dedeepya@pvpsiddhartha.ac.in**

Abstract

This project aims to develop an interactive image-matching card game leveraging HTML, JavaScript, and CSS, with a focus on deploying and hosting the game on a cloud computing platform. The game will be designed to enhance users' cognitive skills and entertainment experience through challenging memory exercises and engaging visuals. The development process involves creating a user-friendly interface using HTML for the structure, JavaScript for the game logic and interactivity, and CSS for styling and visual enhancements. Furthermore, the project will explore the integration of cloud computing technologies for hosting and deploying the game. This includes utilizing cloud storage solutions for storing game assets and user data, as well as deploying the game application on a cloud server for accessibility and scalability. By leveraging cloud computing infrastructure, the game will benefit from improved reliability, scalability, and accessibility, allowing users to access the game seamlessly from anywhere with an internet connection. Moreover, cloud-based deployment will facilitate easier updates and maintenance of the game, ensuring a smooth and uninterrupted gaming experience for players. Overall, the development of this image-matching card game demonstrates the potential of cloud computing in enhancing the accessibility, scalability, and performance of web-based applications, while also providing an entertaining and educational experience for users.

Keywords: Interactive Image Matching Game; Cloud Computing Deployment; Cognitive Skill Enhancement; Web-based Application; Seamless Accessibility

INTRODUCTION

In the realm of web-based applications, the fusion of interactive entertainment and cognitive enhancement has emerged as a promising avenue for engaging users in meaningful experiences. Leveraging the ubiquitous nature of the internet and the power of cloud computing, our project sets out to develop an innovative image-matching card game. Rooted in HTML, JavaScript, and CSS, this game aims not only to entertain but also to sharpen users' cognitive skills through challenging memory exercises and captivating visuals.

The objective of this paper is to delineate the development process and technological framework employed in creating this interactive game. Our approach encompasses the

utilization of HTML for structural integrity, JavaScript for dynamic game logic, and CSS for aesthetic enhancements, culminating in a user-friendly interface that seamlessly integrates entertainment with cognitive stimulation. Moreover, we explore the integration of cloud computing technologies to enhance the accessibility, scalability, and performance of our web-based application. By harnessing cloud storage solutions for housing game assets and user data, and deploying the game application on a cloud server, we aim to transcend the limitations of traditional hosting methods. This strategic adoption of cloud computing infrastructure not only ensures reliability across devices and network conditions but also facilitates seamless updates and maintenance through a continuous deployment pipeline.

The significance of this endeavor lies in its demonstration of how cloud computing can revolutionize the landscape of web-based applications. By affording improved reliability, scalability, and accessibility, our game exemplifies the potential of cloud-based deployment to redefine user experiences in the digital realm.

We explore the development process used to create our picture-matching card game in the next parts, explaining each aspect of interface design, game logic implementation, and cloud integration. We hope to shed light on the advantageous combination of technology and entertainment through this exposition, as well as implications for web apps going forward.

LITERATURE REVIEW

Continuous Deployment in Action

In Srithar S et al.'s research [1], to improve application development speed and quality, the paper addressed the switch from traditional software development concepts to DevOps. It highlights the effectiveness of cloud-based deployment by emphasizing the usage of Jenkins and Azure Cloud DevOps for continuous integration and deployment. The study highlights the advantages of Azure DevOps in expediting the deployment process by comparing different deployment tools and approaches. It refers to relevant publications on cloud application integration, software delivery metrics, and agile procedures. The study highlights how Azure DevOps is more efficient than more conventional methods in enhancing deployment time and performance metrics using a practical implementation of a.NET project in cloud environments.

Jessica D´ıaz et al. [2] discussed major concepts associated with DevOps principles, infrastructure automation, and deployment methodologies in IoT and edge computing settings. The challenges of implementing DevOps in embedded systems, DevOps in software development, DevOps transformation for multi-cloud IoT applications, and deployment and orchestration approaches for IoT are all covered. It also covers the fog computing and edge computing paradigms. Together, these investigations add to our understanding of the dynamic environment surrounding infrastructure management and continuous deployment, offering perspectives that reinforce the research presented in the study on the use of GitOps in IoT edge computing.

Neelam Singh et al [3] by using Jenkins for continuous integration, Ansible for

configuration management and deployment, and Kubernetes for application hosting presented an automated CI/CD pipeline. An overview of Jenkins architecture, Ansible's push-based configuration management methodology, and CI/CD methods are given. The implementation walks through setting up a Kubernetes cluster with Kops, configuring the servers of Ansible and Jenkins, connecting them, and setting up a task in Jenkins to push code from GitHub to Ansible, which uses Docker to deploy to the Kubernetes cluster.

The suggested architecture's execution environments, hosting choices, and plugins are contrasted with those of existing CI/CD technologies, such as GitLab, TeamCity, Bamboo, and CircleCI. Jenkins outperforms competing solutions in terms of build preparation and deployment speed, according to performance tests.

Sriniketan Mysari et al. [4] in their research paper examines how popular and widely used these tools are by comparing the amount of Stack Overflow questions for different integration tools like Jenkins, Bamboo, and TeamCity and the amount of GitHub forks for deployment tools like Ansible, Chef, Puppet, and Salt Stack. It also covers the approach of utilizing Ansible for continuous deployment and Jenkins for continuous integration with pipeline automation. Jenkins initiates the deployment process automatically following a successful integration. The authors stress the benefits of using Ansible and Jenkins, including their open-source nature, ease of configuration, independence from platforms, and ability to save time. However, they also stress the significance of configuring WinRM for remote connections and generating inventory and playbook files in Ansible for deployment automation.

Robert Botez et al. [5] presented an automated pipeline for cloud-based application deployment. from the Technical University of Cluj-Napoca in Roma- nia. It focuses on DevOps methodologies, guaranteeing scalability and minimal downtime through the use of Ansible for continuous deployment and Jenkins for continuous integration. The research emphasizes how to simplify software delivery by integrating six technologies, such as Docker and Kubernetes. Through the combination of these technologies, the paper highlights automation and dependability in cloud computing settings, namely in AWS, and presents effective deployment strategies.

Ralph Holz et al.[6] examined the topic of protecting software deployment pipelines from subversion attempts . Three methods of subverting a pipeline are covered by the authors: deploying an incorrect image, getting around pipeline checks, and permitting unauthorized access to production environments. By identifying trustworthy and untrustworthy components, modeling their interactions, and breaking down untrustworthy components into smaller trustworthy and un-trustworthy sections, they suggest an engineering approach to harden pipelines.

Jenkins, Chef, Docker, GitHub, and AWS are used by the authors to apply this method to a pipeline, which results in a re-architected pipeline with limited network connectivity and access privileges for each component. This method lessens the attack surface and limits the impact of compromised components, but it does not secure the pipeline.

Indika Perera et al. [7] studied several important topics in their research on the

automation of the pipeline for continuous integration and delivery (CI/CD). The authors talk about the concepts of agile software development and how they affect the creation of software products that are driven by the market. They look at the difficulties in making the switch from agile development to continuous deployment. The advantages and difficulties of the ideas of continuous delivery and integration are discussed. There are references to case studies of continuous deployment at businesses like Oando and Facebook. The utilization of microservices in mission-critical systems, setting metrics for CI/CD pipelines, and getting started with CI in software development are also covered in the paper. The adoption of continuous delivery is examined, and approaches, tools, difficulties, and practices in continuous integration, delivery, and deployment are summarized. Finally, Facebook's development and deployment procedures are emphasized.

Ganesh Semalty et al. [8] demonstrated the increase classification accuracy on unbalanced datasets that arise from system changes in a continuous deployment situation, On a ticket dataset, it applies natural language processing techniques such as TF- IDF for feature extraction and Linear SVC for classification. The research contrasts algorithmic-level options, such as altering the classifier algorithm, and data-level solutions, such as oversampling minority classes or undersampling majority classes, to handle the unbalanced dataset problem. It generates synthetic samples for minority classes using the Synthetic Minority Oversampling Technique, which raises the model's recall and F1 score. To guarantee that every fold is representative of every class, the study further uses stratified k-fold cross-validation.

Germano Veiga et al [9] by examining 150 ROS repositories, it explores how CI/CD procedures are currently used in robotic applications. It finds that 82.7% of repositories use some kind of CI, however only 43.3% have unit tests and 30.7% have other forms of tests, with very few utilizing code quality (6%) or coverage (4.7%) metrics. Despite 50.7% of repositories possessing the infrastructure needed to possibly benefit from simulation-based testing, the research observes that none of them apply simulation tests in their continuous integration pipelines. The next suggestion is to improve continuous integration pipelines (CI pipelines) by using robot application bundles created by AWS RoboMaker, uploading them to S3, and starting the simulations using AWS Lambda and Step Functions. This method's benefits over real-world testing in terms of cost and efficiency are demonstrated in an industrial use case inside the H2020 FASTEN project.

Enrique Moguel et al. [10] examined the deployment of quantum services in their research by using DevOps techniques to overcome the difficulties associated with developing quantum software and by identifying the optimal quantum computer that is accessible at runtime.

It covers how to develop quantum services using OpenAPI, how to expand the OpenAPI Code Generator to accommodate quantum web services, and how to utilize GitHub Actions to automate the Continuous Deployment process for deploying quantum services. By automating the creation and deployment of quantum services, the study seeks to standardize the development of quantum services, increase accessibility to quantum computing, demonstrate a promising success rate in the automatic generation and

deployment of quantum services, and highlight the potential to streamline the processes involved in circuit generation and deployment.

Hamed Fawareh et al [11] to strengthen the connection between the development and operation teams, discussed the shift from continuous integration to continuous delivery in DevOps. It talks about the advantages of continuous delivery and integration, emphasizing the value of automation, speedy delivery, and enhanced cooperation between development and operations. The research explores the benefits of DevOps in- including accelerated development cycles, elevated client con- tenement, and dependable deployment methodologies. It also discusses the tactics and difficulties involved in putting DevOps approaches into practice, highlighting the contribution that continuous integration and delivery provide to improving software development processes and cutting lead times [12-14].

Selin Aydin et al [15] highlighted the importance of artifact-centric perspectives in business processes, which provide simpler maintenance and change without IT experience. The research focuses on the automated design of continuous delivery pipelines using architecture models. Models and DSLs for delivery processes have been presented by a number of systems and initiatives, including Jenkins; nevertheless, these models usually take an activity-centric approach.

The advantages and challenges of implementing continuous delivery (CD) and continuous integration (CI) approaches, particularly the growing complexity of delivery models and systems, have been the subject of research. Research has also highlighted the necessity of extra assistance and tools for developers creating and overseeing delivery models and procedures, as these individuals frequently lack the necessary skills and expertise. As a result, the paper's method simplifies the necessary knowledge to the architectural details of the software project, making it more understandable for developers with different degrees of process expertise [16].

Mitesh Soni et al [17] discussed the challenges the insurance industry faces in bringing innovative ideas to market quickly and consistently. These challenges include the need for improved responsiveness to the ever-changing demands of the market, a quicker time to market for new initiatives and services, and support for creative customer interaction methods.

To increase performance and quality assurance, it presents DevOps culture as an emerging technique that expands on agile methodology to quickly construct apps and deploy them across environments in an automated manner [18]. In contrast to traditional application development and release management processes, the paper emphasizes the advantages of Continuous Integration (CI) and Continuous Delivery (CD), which enable the continuous delivery of high-quality artifacts to clients together with continually integrated feedback [19]. The paper aims to develop a proof of concept for developing an efficient framework for continuous integration, continuous testing, and continuous delivery. This framework will use the build pipeline concept to automate various tasks such as source code compilation, code analysis, test execution, packaging, infrastructure provisioning, deployment, and notifications [20].

METHODOLOGY

Using HTML, JavaScript, and CSS, the project intends to create an interactive picture-matching card game with an emphasis on improving players' cognitive abilities through difficult memory tests and captivating graphics. During the development process, a user-friendly interface is created utilizing CSS for style, JavaScript for game logic, and HTML for structure. Furthermore, the game program is deployed on a cloud server for scalability and accessibility, and cloud storage solutions are utilized for game assets and user data[21]. These integrations of cloud computing technologies extend to hosting and deployment.

Developing a user-friendly interface, putting complex game logic into practice, guaranteeing dependability across devices and network circumstances, and enabling smooth updates via a continuous deployment pipeline are among the challenges.

Automated Testing: provides Consistent Quality Assurance Immediate Feedback, Reduced Manual Effort.

Continuous Integration (CI): provides Detection of Issues, Streamlined Collaboration.

Code modifications are immediately saved in an S3 bucket by connecting GitHub to AWS Code Pipeline, which simplifies execution and lets users play the game without interruption from any location. All things considered, the project shows how cloud computing may enhance web-based applications' efficiency, scalability, and accessibility while giving users a fun and instructive gaming experience.

Conceptualization and Game Design

The project began with the Conceptualization and Game Design phase, during which the main idea and principles of the picture-matching card game were carefully developed. Inspired by popular memory games and cognitive psychology research [22], the project team set out on a voyage of conceptual refinement and creative innovation. The project's first stage involves designing the picture-matching card game as shown in Figure 1 to improve players' entertainment value and cognitive abilities.

To find important characteristics and design components that might support memory workouts and engagement, brainstorming sessions, and a literature study were carried out [23]. A wide range of visually engaging cards with different degrees of difficulty was created as part of the game design process to provide players with a hard yet entertaining experience.

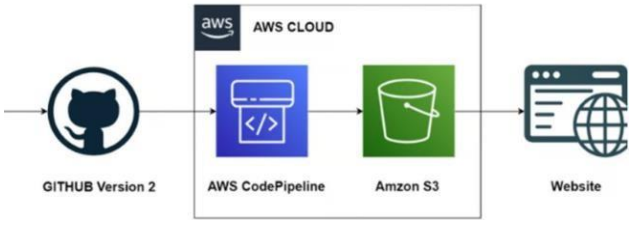

Figure 1. Workflow Diagram.

Development Framework Selection

The phase of Development Framework Selection was an essential decision-making process that aimed to choose the best technologies and instruments to accomplish the project's goals efficiently and effectively. During this phase, different development frameworks, libraries, and platforms were assessed for simplicity of use, scalability, compliance with project requirements, and community support [24].

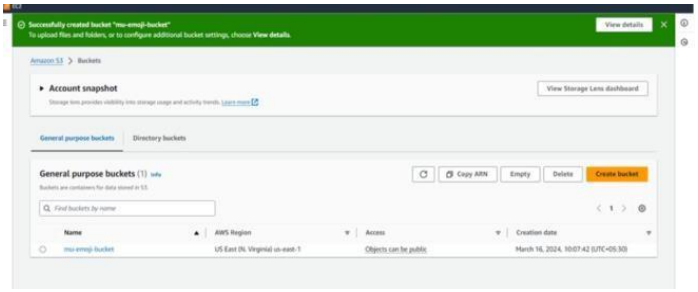

Figure 2. Creation of the Bucket

Important factors were the requirement for a flexible styling mechanism to improve visual aesthetics, a versatile programming language to execute intricate game logic, and a strong frontend framework to develop a dynamic and interactive user interface. The development framework was carefully chosen to guarantee that it was compatible with the project objectives and technical requirements after the conception phase. The main technologies that were selected were HTML, JavaScript, and CSS because of their enormous community resources, cross-platform compatibility, and adaptability. These languages served as the basis for integrating game logic, improving visual aesthetics, and building a solid and interactive user interface.

User Interface Design

The picture-matching card game's visual identity and interactive experience were greatly influenced by the User Interface Design phase, which had the main goal of developing an approachable, entertaining, and user-friendly interface that could be used by players of all device types and skill levels.

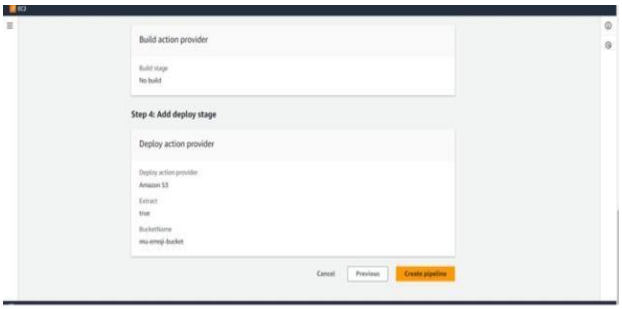

Figure 3. Continuous deployment pipeline

A crucial part of the development process was creating a user interface, which aimed to blend usefulness and aesthetics. Tools like wireframing and prototyping were used to see how the game interface would look and function. The creation of a responsive and user-

friendly interface that accommodates users with different device choices and skill levels was prioritized.

Game Logic Implementation

The primary development effort was concentrated on converting the conceptual idea of the picture-matching card game into working code during the Game Logic Implementation phase. JavaScript was used to convert the conceptual design into functional code to build game logic. This involved outline outlines the gameplay's guidelines, including the level progression, scoring system, and card matching techniques. Sophisticated algorithms were created to manage the real-time card flipping, matching, and shuffling, guaranteeing smooth gaming and compliance with the desired cognitive difficulties.

1. Visual Enhancements with CSS

The game interface's visual attractiveness was improved, and an immersive gameplay experience was produced by utilizing CSS. To develop animations, transitions, card layouts, and other visual aspects, custom stylesheets were made. Maintaining uniformity in design aesthetics across various screen sizes and resolutions was a priority to maximize the user experience on a variety of devices.

2. Integration of Cloud Computing Technologies

Using cloud computing's capabilities to improve the picture-matching card game's performance, scalability, and accessibility was made possible in large part by the Integration of Cloud Technologies phase.

To facilitate game hosting, data storage, deployment automation, and scalability management, this phase entailed integrating numerous cloud services, platforms, and infrastructure elements. Cloud computing technologies were incorporated into the gaming application development process to enable hosting and deployment. Game assets, such as pictures and configuration data, were stored using cloud storage services like Amazon S3. To provide scalability and accessibility for players globally, the game application was also installed on a cloud server using services like AWS EC2 Platform.

3. Continuous Deployment Pipeline

An essential part of the development process, the Continuous Deployment Pipeline (CDP) is made to automate the process of managing infrastructure modifications, releasing updates, and guaranteeing that the picture-matching card game will always receive new features.

To accelerate iteration and release cycles and optimize the deployment process, this phase integrates cloud services, deployment tools, automated testing frameworks, and version control systems.

The establishment of a continuous deployment pipeline aimed to optimize the gaming application's maintenance and update delivery processes. Code updates may be seamlessly deployed to the cloud server thanks to the automation of the deployment procedure provided by integration tools like GitHub and AWS Code-Pipeline. This strategy reduced interruptions and downtime, giving gamers a seamless and continuous gaming experience.

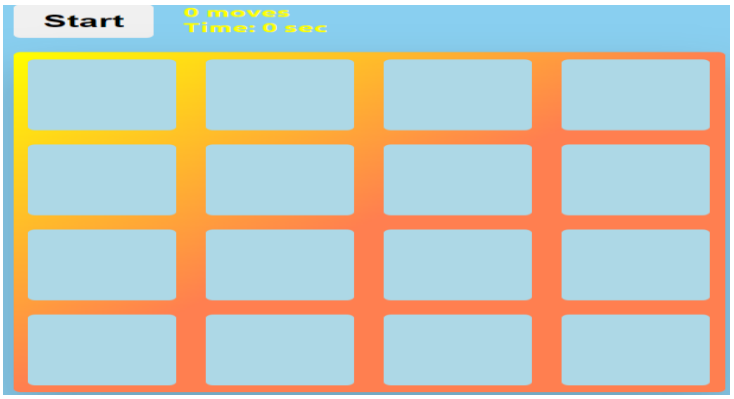

Figure 4. Emoji Matching.

RESULTS AND DISCUSSION

Presentation of Pre-test Scores of Experimental versus Control Groups

As evidenced by several metrics and assessments, the interactive picture matching card game developed with cloud computing technology produced notable improvements in accessibility, scalability, and performance.

A. Scalability and Performance

The gaming application was able to scale flexibly in response to demand because of the cloud architecture, which guaranteed excellent performance even during moments of high usage.

Scenarios from load testing proved that the system could manage more traffic without sacrificing usability or responsiveness. The scalability and efficiency of the cloud development were demonstrated by performance indicators, such as response times, throughput rates, and resource utilization, which regularly met or surpassed preset standards.

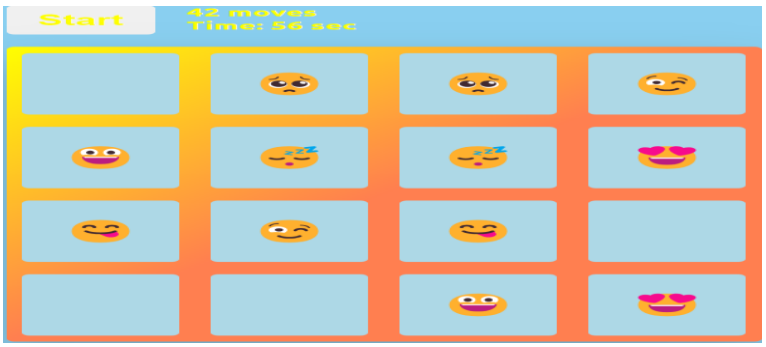

Figure 5. Matching the emojis

B. Accessibility and Availability

Cloud-based deployment facilitated seamless accessibility and availability of the game application, enabling users to access the game from anywhere with an internet connection. Content Delivery Networks (CDNs) optimized the delivery of static assets, reducing

latency and improving load times for players worldwide. High availability architectures, such as multi-region redundancy and failover mechanisms, ensured continuous availability of the game application, minimizing downtime and disruptions for players.

C. Deployment Efficiency

The implementation of a continuous deployment pipeline (CDP) streamlined the deployment process, enabling rapid and automated deployment of updates and enhancements to the game application. Integration with version control systems and automated testing frameworks facilitated efficient code management and quality assurance, while deployment automation tools ensured consistent and error-free deployments. The deployment pipeline improved development efficiency, shortened release cycles, and minimized the risk of deployment errors or inconsistencies.

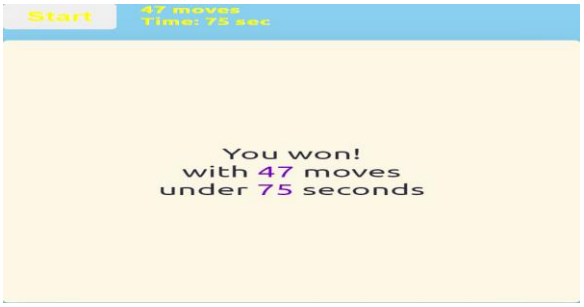

Figure 6. Moves after completing the matching

CONCLUSION

The integration of cloud computing technologies has enabled the development of an innovative and scalable interactive image-matching card game, showcasing the transformative potential of cloud-based applications. By leveraging cloud infrastructure and deployment automation, the game offers seamless accessibility and availability to players worldwide, ensuring uninterrupted gaming experiences and minimal downtime. The implementation of a continuous deployment pipeline streamlines the development process, facilitating rapid iteration and deployment of updates, ultimately leading to high levels of user engagement and satisfaction.

By utilizing cloud services, the game can be accessed by players from diverse geographical locations, breaking down barriers and creating a truly global gaming community. Cloud providers offer robust infrastructure with high availability and disaster recovery options, ensuring that the game remains operational even in the face of hardware failures or other disruptions. Furthermore, cloud services often come with advanced security features and compliance certifications, ensuring that user data is protected and regulatory requirements are met, thereby building trust with the gaming community.

CONFLICT OF INTERESTS

The authors declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

1. Srithar, S., et al. ”Cost-Effective Integration and Deployment of Enterprise Application Using Azure Cloud Devops.” 2022 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2022.
2. Lo´pez-Viana, Ramo´n, Jessica D´ıaz, and Jorge E. Pe´rez. ”Continuous Deployment in IoT Edge Computing: A GitOps implementation.” 2022 17th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2022.
3. Singh, Neelam, Aman Singh, and Vandana Rawat. ”Deploying Jenkins, Ansible and Kubernetes to Automate Continuous Integration and Con- tinuous Deployment Pipeline.” 2022 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). IEEE, 2022.
4. Mysari, Sriniketan, and Vaibhav Bejgam. ”Continuous Integration and Continuous Deployment Pipeline Automation Using Jenkins Ansible.” 2020 International conference on emerging trends in information tech- nology and engineering (IC-ETITE). IEEE, 2020.
5. Cepuc, Artur, et al. ”Implementation of a continuous integration and deployment pipeline for containerized applications in amazon web services using jenkins, ansible and kubernetes.” 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2020.
6. Madhuri, A., S. Phani Praveen, D. Lokesh Sai Kumar, S. Sindhura, and Sai Srinivas Vellela. "Challenges and issues of data analytics in emerging scenarios for big data, cloud and image mining." Annals of the Romanian Society for Cell Biology (2021): 412-423.
7. Vellela, Sai Srinivas, R. Balamanigandan, and S. Phani Praveen. "Strategic Survey on Security and Privacy Methods of Cloud Computing Environment." Journal of Next Generation Technology 2, no. 1 (2022).
8. Praveen, S. Phani, K. Thirupathi Rao, and B. Janakiramaiah. "Effective allocation of resources and task scheduling in cloud environment using social group optimization." Arabian Journal for Science and Engineering 43 (2018): 4265-4272.
9. C. D. Kothapalli, G. Navya, U. Jaladhi, S. R. Sulthana, D. L. S. Kumar and S. P. Praveen, "Predicting Buy and Sell Signals for Stocks using Bollinger Bands and MACD with the Help of Machine Learning," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 333-340, doi: 10.1109/ICSCSS57650.2023.10169500.
10. Praveen, S. Phani, Hesam Ghasempoor, Negar Shahabi, and Fatemeh Izanloo. "A Hybrid Gravitational Emulation Local Search-Based Algorithm for Task Scheduling in Cloud Computing." Mathematical Problems in Engineering 2023, no. 1 (2023): 6516482.
11. Praveen, S. Phani, and K. Thirupathi Rao. "An effective multi-faceted cost model for auto-scaling of servers in cloud." In Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 1, pp. 591-601. Springer Singapore, 2019.
12. Praveen, S. Phani, S. Sindhura, A. Madhuri, and Dimitrios A. Karras. "A novel effective framework for medical images secure storage using advanced cipher text algorithm in cloud computing." In 2021 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1-4. IEEE, 2021.
13. Sherubha, P., S. P. Sasirekha, A. Dinesh Kumar Anguraj, J. Vakula Rani, Raju Anitha, S. Phani Praveen, and R. Hariharan Krishnan. "An Efficient Unsupervised Learning Approach for Detecting Anomaly in Cloud." Comput. Syst. Sci. Eng. 45, no. 1 (2023): 149-166.
14. Rao, K. Thirupathi. "Client-awareness resource allotment and job scheduling in heterogeneous cloud by using social group optimization." International Journal of Natural Computing Research (IJNCR) 7, no. 1 (2018): 15-31.

15. Praveen, S. Phani, Balamuralikrishna Thati, Ch Anuradha, S. Sindhura, M. Mohammed Altaee, and Abdul Jalil. "A novel approach for enhance fusion based healthcare system in cloud computing." Journal of Intelligent Systems and Internet of Things 9, no. 1 (2023): 84-96.
16. Basha, Pathan Hussain, G. Sai Chatanya Kumar, and N. Raghavendra Sai. "SIGNATURE TRANSFER IN SECURED WAY FOR DISTRIBUTED INTRUSION DETECTION SYSTEM BY USING BLOCKCHAIN TECHNOLOGY."
17. Praveen, S. Phani, and K. Thirupathi Rao. "An Algorithm for Rank Computing Resource Provisioning in Cloud Computing." International Journal of Computer Science and Information Security (IJCSIS) 14, no. 9 (2016).
18. Arachchi, S. A. I. B. S., and Indika Perera. ”Continuous integration and continuous delivery pipeline automation for agile software project management.” 2018 Moratuwa Engineering Research Conference (MER- Con). IEEE, 2020.
19. Dwivedi, Yogendra Swaroop, Ganesh Semalty, and Amit Moondra. ”Predictive Technique To Improve Classification On Continuous System Deployment.” 2021 IEEE International Conference on Electronics, Com- puting and Communication Technologies (CONECCT). IEEE, 2021.
20. Gmeiner, Johannes, Rudolf Ramler, and Julian Haslinger. ”Automated testing in the continuous delivery pipeline: A case study of an online company.” 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 2015.
21. P. Dedeepya, D. Karishma, S. G. Manuri, T. Raghuvaran, V. Shariff and S. Sindhura, "Enhancing Cyber Bullying Detection Using Convolutional Neural Network," 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2023, pp. 1260-1267, doi: 10.1109/ICOSEC58147.2023.10276007.
22. P. Dedeepya, P. Chiranjeevi, V. Narasimha, V. Shariff, J. Ranjith and J. V. N. Ramesh, "Image Recognition and Similarity Retrieval with Convolutional Neural Networks," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 709-716, doi: 10.1109/ICACRS58579.2023.10404664.
23. P. Dedeepya, M. Yarrarapu, P. P. Kumar, S. K. Kaushik, P. N. Raghavendra and P. Chandu, "Fake News Detection on Social Media Through a Hybrid SVM-KNN Approach Leveraging Social Capital Variables," 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024, pp. 1168-1175, doi: 10.1109/ICAAIC60222.2024.10575681.
24. D. Swapna, U. K. Sri, V. S. N. Himaja, T. N. Varshita, V. Gayatri and S. P. Praveen, "Crypto Logistic Network: Food Supply Chain and Micro Investment using Blockchain," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 908-915, doi: 10.1109/ICACRS58579.2023.10404449.