
Research Article**Enhancing Efficiency and Scalability: DevOps Implementation for Educational Chat Systems****N. Meghana^{1*}**, **K. Sri Lakshmi²**, **M. Naga Lakshmi Teja Sree³**, **K. Srujana⁴**, **N. Ashok⁵**^{1,2,3,4,5}Dept. of Information Technology, Vasireddy Venkatadri Institute of Technology, Guntur, India*Corresponding Author: namburumeghana2020@gmail.com,**Received:** 02/Feb/2024; **Accepted:** 04/Mar/2024; **Published:** 31/Mar/2024. **DOI:** <https://doi.org/10.26438/ijcse/v12i3.1924>

Abstract: This study presents a thorough technique for the implementation of a Telegram chatbot created especially for educational institutions to expedite the tracking of attendance and academic results. The chatbot is constructed using Python's Telebot library and DevOps concepts. It is then carefully deployed on an AWS EC2 instance to ensure smooth interaction with a variety of educational systems and to meet the specific needs of different academic institutions. Jenkins makes continuous integration easier by automating the deployment and build processes, which increases scalability and efficiency. The chatbot's customization possibilities enable customized integration with current educational infrastructures, guaranteeing flexibility and adaptability. Additionally, the project's strong design allows for future growth and the addition of sophisticated features like learning management system integration. The chatbot seeks to maintain ongoing relevance and efficacy in promoting student participation and optimizing academic administration procedures by continuing to be flexible in response to new technological developments. This study demonstrates how chatbot technology may revolutionize educational services and administration, supporting the further development of academic institutions in the digital era. This study highlights the revolutionary potential of chatbot technology in transforming student services and academic administration by connecting technological innovation with educational demands.**Keywords:** Academic results, AWS Cloud, Chatbot, Continuous Integration, DevOps, Jenkins, Telegram.

1. Introduction

The use of cutting-edge technologies has become more and more necessary in today's educational environment to improve academic processes' accessibility and efficiency. Chatbot technology is one such innovation that has grown in popularity recently. Artificial intelligence-driven chatbots present a viable way to handle several facets of academic administration, such as tracking attendance and answering questions about results. The goal of this study is to investigate how chatbot technology can revolutionize the way that educational services and administration are provided and managed. The project's goal is to create and implement a Telegram chatbot that is especially suited for academic institutions by utilizing DevOps principles and Python's Telebot module. By enabling smooth communication between students and academic systems, the chatbot will improve the quality of the academic experience overall and offer individualized support. This study demonstrates how important it is to connect technological innovation with educational standards, opening the door for chatbot adoption to transform student services and academic administration.

Furthermore, the necessity for effective and scalable solutions has grown significantly as educational institutions negotiate the difficulties of the digital age. In order to ensure scalability and accessibility in the cloud, this project deploys the chatbot on an AWS EC2 machine. The project improves productivity and reliability by streamlining the build and deployment processes with Jenkins for continuous integration. Furthermore, the chatbot's built-in customization features allow for a smooth connection with a variety of educational systems, meeting the particular needs of different academic institutions. This research aims to show how chatbot technology can revolutionize academic administration and student services through careful design and strategic implementation, ultimately contributing to the ongoing evolution of educational institutions in the digital era.

2. Problem Statement

The process of deploying chatbots intended for educational settings in several academic institutions is frequently disorganized, laborious, and prone to mistakes. Delays, mishandled implementations, and challenges in expanding the system to suit the demands of the organization might result from this ineffective deployment procedure. For these

reasons, it is imperative to expedite the deployment process of academic chatbots in order to boost scalability, minimize errors, and increase efficiency. Institutions can guarantee the effective deployment and utilization of chatbots to offer students accurate and fast academic support by tackling these obstacles.

3. Related Work

The literature review in [1] examined chatbots while delving into historical development and technical breakthroughs. It examined implementation technologies, including pattern matching and machine learning, suggested classification schemes, and offered insights into the shortcomings of earlier stages. In addition, it covered architectural plans and talked about the dangers of using chatbots.

Similarly, in [2], the research looked at how society is affected by contemporary technology on a wide scale, particularly with the emergence of chatbots as virtual assistants. It demonstrated how adaptable chatbots can be for everything from helping people with everyday tasks to offering amusement and advice on business strategies. The study gave an overview of chatbot varieties and their uses, and it recommended a classification scheme based on usability and market trends.

A comprehensive review was conducted to examine the integration of chatbots in educational settings, building upon the findings from previous studies [3]. The study highlighted the possibility of chatbots providing individualized services to staff members and students at institutions, improving efficiency and accessibility in the education sector. It outlined the benefits, challenges, and future directions for study, clarifying the use of chatbot innovations in educational settings.

Our project intends to expedite the implementation of a Telegram chatbot customized for academic results and attendance checks, taking inspiration from these research publications.

In a similar vein, [4] emphasizes the importance of DevOps approaches in bridging the gap between development and operations teams as a solution to contemporary software development difficulties. It describes the fundamentals of DevOps and how important it is to improve software quality, deployment rates, and communication. Furthermore, [5] explores the convergence of cloud computing and DevOps, emphasizing the advantages of incorporating DevOps methodologies into cloud development and testing as well as their mutually beneficial interaction. The report emphasizes how crucial it is to use cloud technologies and automate DevOps procedures in order to deploy software quickly and reliably. Building on these discoveries, our project intends to use DevOps techniques to ensure efficiency, scalability, and ongoing development in academic support services by streamlining the implementation of our educational chatbot.

As per the past research's defined ideas, our project gives priority to implementing DevOps approaches in order to guarantee a successful deployment procedure. We use Jenkins for continuous integration and AWS Cloud for deployment because we understand how important DevOps is to improving the scalability and dependability of our systems. By providing a user-friendly interface for gaining access to academic information, our initiative prioritizes the user experience while also enhancing usability and accessibility. Our focus is on scalability, dependability, and 24/7 availability in order to satisfy the changing needs of students and educational institutions. Our dedication to scalability, affordability, and dependable performance is demonstrated by our choice to deploy on the AWS Cloud, and the use of DevOps techniques reduces the possibility of human error and improves system performance as a whole.

4. Experimental Method/Procedure/Design

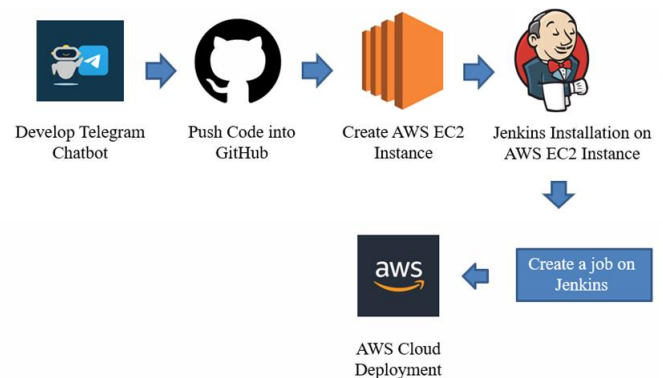


Figure 1. System Architecture

A. Telebot development in Python:

The Python Telebot library was used to create the Telegram chatbot for verifying attendance and academic outcomes. Writing Python code was required to implement the chatbot's features, which included responding to user commands, obtaining data, and managing commands.

B. Uploading the Code to GitHub:

To provide simple access to the source, the code was uploaded to a GitHub repository.

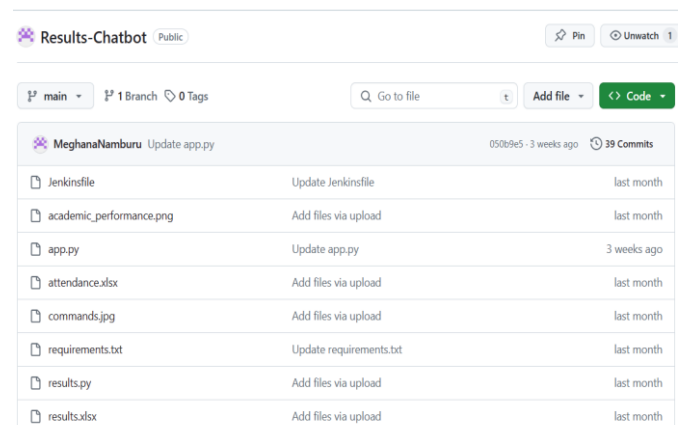


Figure 2. Project Repository

C. Configuring an AWS EC2 Instance:

An Elastic Compute Cloud (EC2) instance was set up to act as the chatbot's deployment environment. Scalable solutions like AWS EC2 enable flexible resource allocation according to demand.

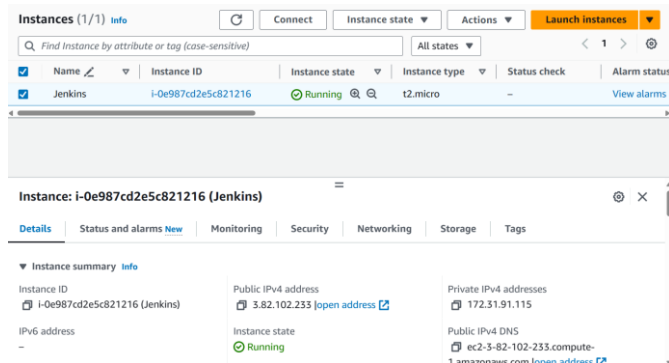


Figure 3. AWS EC2 Instance

D. Installing Jenkins on an AWS EC2 Instance:

The AWS EC2 instance was set up and configured to host Jenkins, a well-known continuous integration tool. Jenkins enables software development to be more efficient and reliable by facilitating automated build, testing, and deployment processes.

E. Setting Up and Configuring a Jenkins Job:

To automate the chatbot's build and deployment, a Jenkins job was set up. This meant creating a freestyle project within Jenkins and specifying the necessary build steps, and setting up triggers to enable automated execution.

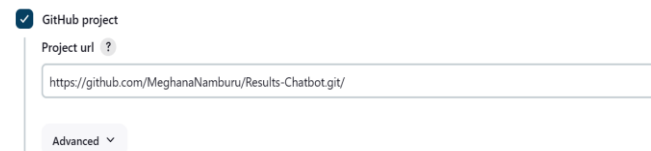


Figure 4. GitHub Project Integration

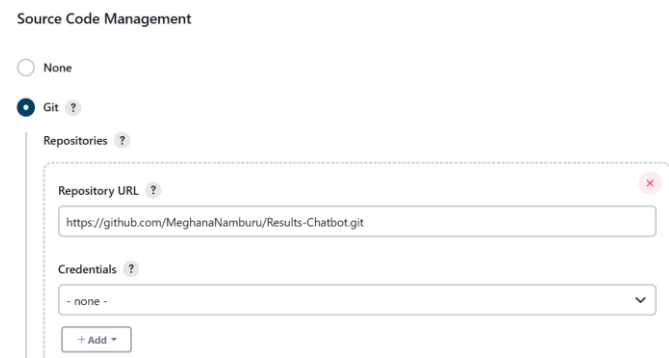


Figure 5. Source Code Management with Git



Figure 6. Branches to build

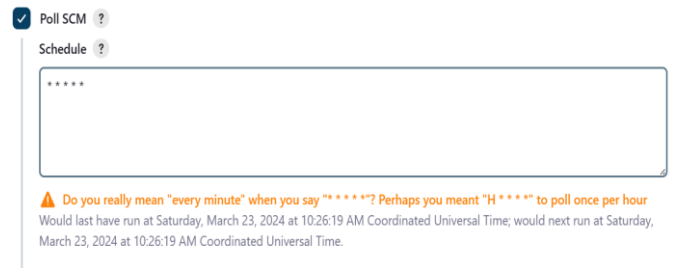


Figure 7. Polling Schedule

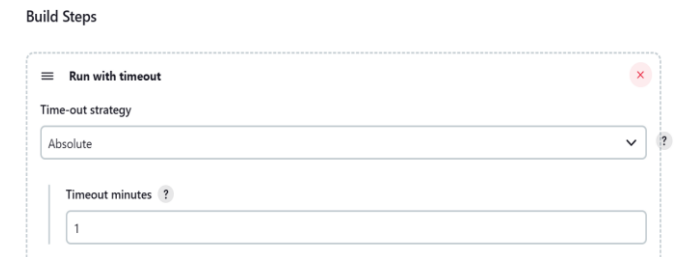


Figure 8. Build Steps – Timeout Settings

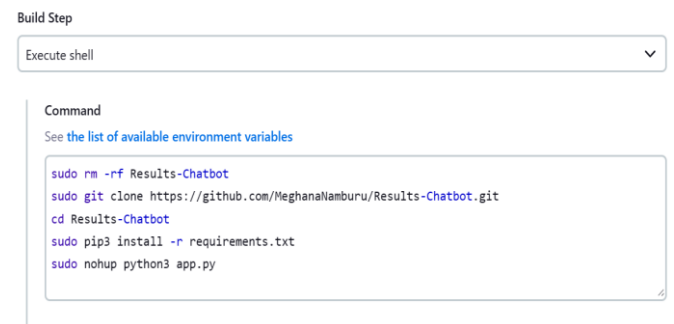


Figure 9. Build Steps – Execution Commands

The screenshots show the painstaking configuration processes that Jenkins undertook in order to coordinate the Telegram chatbot's deployment. Details about the Git repository are described, starting with the repository URL and going through the source code management configuration and branch specification. The integration between Jenkins and the specified GitHub repository is then established using the GitHub Project setup, which makes cooperation easy. Simultaneously, the Poll SCM configuration arranges for regular updates and builds by periodically monitoring the repository for changes. Moving on to the Build Steps setup, specific commands are outlined to carry out important tasks such as cloning the repository, installing dependencies, and starting the program. Every command and setting is carefully designed to optimize the deployment procedure, guaranteeing effectiveness, dependability, and smooth integration with the Jenkins environment.

F. Deploying the Chatbot on the AWS Cloud:

Taking advantage of the scalability and accessibility provided by AWS services, the chatbot was set up on the AWS Cloud infrastructure. In order for the chatbot program to function properly in the cloud, it had to be deployed to an AWS EC2 instance and configured.

```

Console Output
requirements.txt (line 1) (2.4.7)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from
matplotlib-->-r requirements.txt (line 1)) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages (from pandas-->-r
requirements.txt (line 2)) (2022.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas-->-r
requirements.txt (line 2)) (2024.1)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from pyTelegramBotAPI-->-r
requirements.txt (line 3)) (2.25.1)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.10/dist-packages (from openpyxl-->-r
requirements.txt (line 4)) (1.1.0)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7-
matplotlib-->-r requirements.txt (line 1)) (1.16.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with
the system package manager. It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
+ sudo nohup python3 app.py
    
```

Figure 10. Deploying the Chatbot

Overall, the process included installing Jenkins for continuous integration, building and configuring a Jenkins job, installing Python and Telebot for the development of the Telegram chatbot, and deploying the chatbot on the AWS Cloud infrastructure. This strategy made sure that the chatbot was developed, tested, and deployed quickly and effectively, showcasing how contemporary technology may be used to improve student accessibility and convenience while accessing academic content.

5. Results and Discussion

This section presents a set of screenshots that demonstrate the operational capabilities and efficacy of the deployed Telegram chatbot. With the help of these screenshots, users can see how well the chatbot performs in actual situations and how well it can answer questions from users and find academic resources. The functions of the chatbot are emphasized in these screenshots, providing information about how to use and operate it. This visual documentation offers solid proof of the chatbot's effectiveness in improving accessibility and expediting student academic support services, as well as serving to authenticate the chatbot's successful implementation.

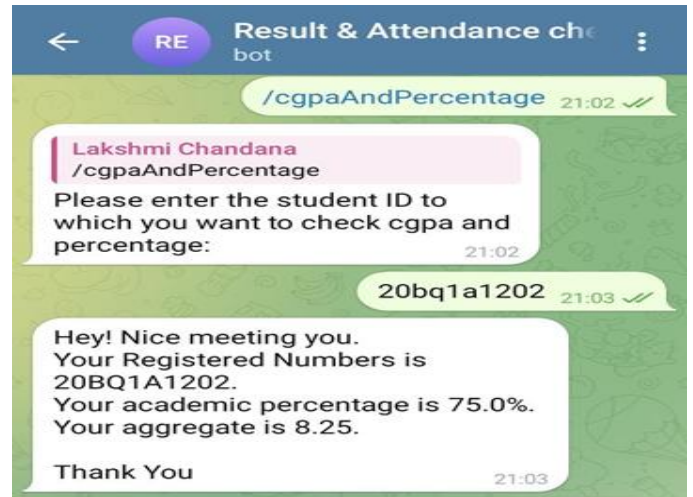


Figure 12. Check CGPA and Percentage

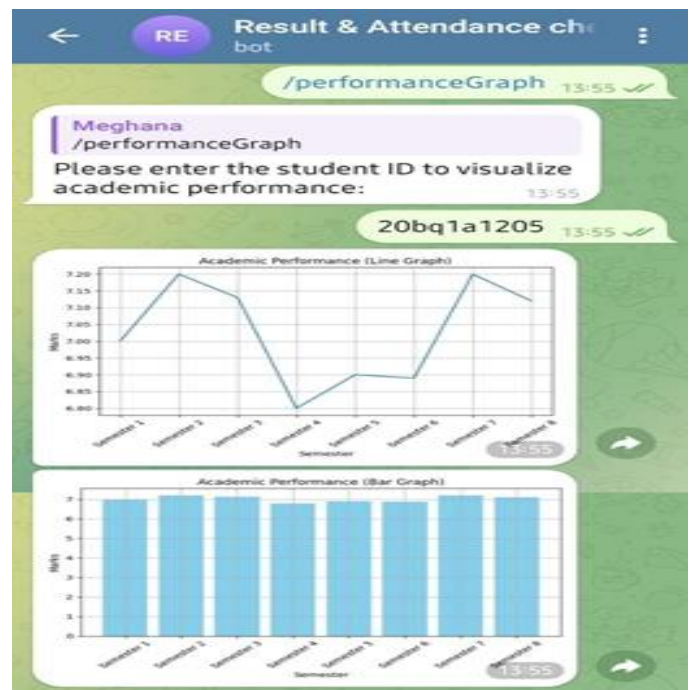


Figure 13. Visualize Academic Performance

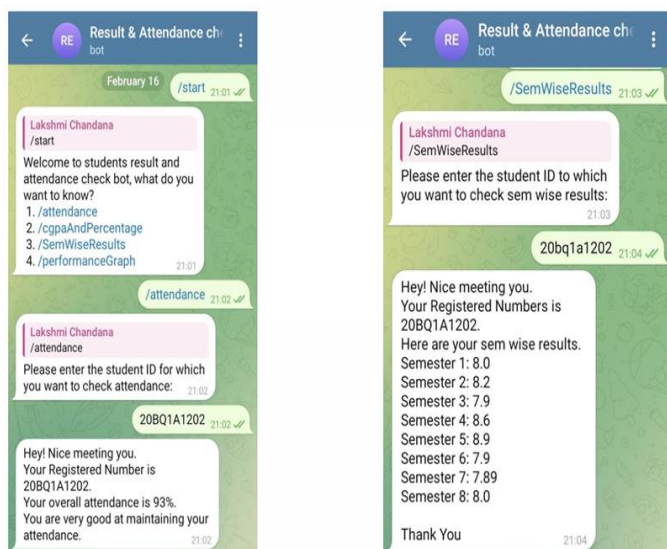


Figure 11. Check attendance and semester-wise results

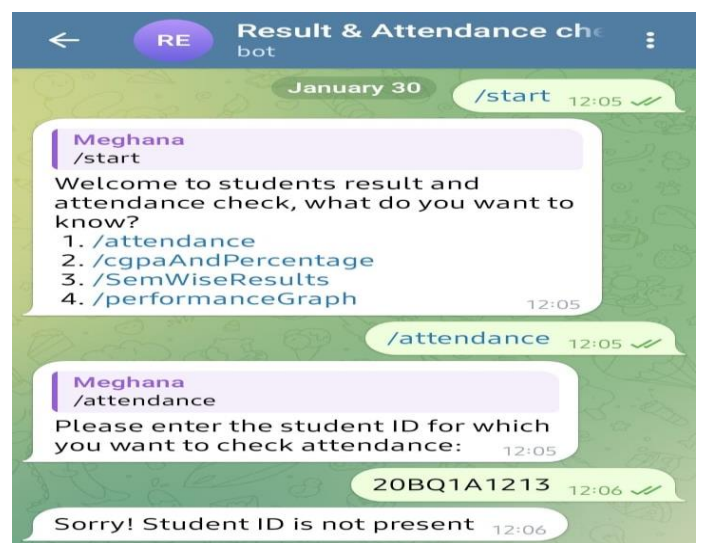


Figure 14. If Student Id is not present

The functionality of the chatbot is demonstrated in the screenshots, which show how it interacts with people to retrieve data in response to orders. The chatbot asks to retrieve data and asks for the student's roll number upon user input. The chatbot displays the necessary information to the user in a smooth manner if the roll number entered matches an entry in the dataset. On the other hand, in the event that the roll number that was supplied is not included in the dataset, the chatbot will instantly inform the user of this, guaranteeing open communication about the availability of data. This strong interface highlights how well the chatbot performs data retrieval tasks, improves user experience, and makes sure that messages about dataset coverage are understood.

```

Console Output
requirements.txt (line 1) (2.4.7)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from
matplotlib-->-r requirements.txt (line 1) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages (from pandas-->-r
requirements.txt (line 2) (2022.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas-->-r
requirements.txt (line 2) (2024.1)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from pyTelegramBotAPI-->-r
requirements.txt (line 3) (2.25.1)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.10/dist-packages (from openpyxl-->-r
requirements.txt (line 4) (1.1.0)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7-
matplotlib-->-r requirements.txt (line 1) (1.16.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with
the system package manager. It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
+ sudo nohup python3 app.py

```

Figure 15. Project successfully deployed on AWS cloud

Updates and alterations to the project are streamlined by the smooth interaction between the Jenkins task and the GitHub repository. The revised project is executed when the Jenkins job is restarted following modifications to the project's contents in the GitHub repository. The deployment process is made simpler by this effective approach, which guarantees that the most recent version of the project is always delivered without difficulty or delay.

6. Conclusion and Future Scope

In summary, the use of the Telegram chatbot for educational purposes is a noteworthy accomplishment in using contemporary technology to improve accessibility for students. The project successfully streamlined the deployment process, ensuring efficiency and reliability through the methodical application of DevOps concepts. The project has the potential to transform academic support services, as demonstrated by its successful deployment of a functional chatbot.

Future developments might include adding natural language processing to improve chatbot interactions, extending its capability to include more academic services, and further streamlining deployment procedures for even more efficiency and scalability.

Data Availability

We had retrieved data from a self-generated Excel sheet, where the sheet contains the roll numbers of the students and their corresponding results and attendance. The GitHub source repository for the project has the Excel sheets with the student roll numbers, academic results, and attendance data on them. The data is available in the repository for researchers who are interested in using it. (GitHub repository: <https://github.com/MeghanaNamburu/Results-Chatbot.git>)

Conflict of Interest

There isn't any conflict of interest between us.

Funding Source

None

Authors' Contributions

In this project, each author made significant contributions to different aspects. N. Meghana led the research efforts, diving into existing literature and framing the project's direction. K. Sri Lakshmi and M. Naga Lakshmi Teja Sree were responsible for implementing the chatbot, bringing the project to life with their software development skills. K. Srujana focused on planning for the future, researching ways to enhance the chatbot's capabilities. Together, their collaboration ensured a well-rounded and successful project outcome.

Acknowledgements

Our guide N Ashok played a pivotal role in guiding us through the intricacies of the project, offering invaluable insights and support that were instrumental in the successful completion of our endeavor.

References

- [1] Eleni Adamopoulou, Lefteris Moussiades "Machine Learning With Applications, Chatbots: History, Technology, And Applications", Science Direct, Vol.2, 2020.
- [2] Aishwarya Gupta, Divya Hathwar, Anupama Vijayakumar "Introduction To AI Chatbots", International Journal of Engineering Research & Technology (IJERT), Vol.9, Issue.7, pp.255-258, 2020.
- [3] Chinedu Wilfred Okonkwo, Abejide Ade-Ibijola "Computers and Education: Artificial Intelligence, Chatbots applications in education: A systematic review", Science Direct, Vol.2, 2021.
- [4] Aadil Hasan "A Review Paper on DevOps Methodology", International Journal of Creative Research Thoughts (IJCRT), Vol.8, Issue.6, pp.2583-2589, 2020.
- [5] Dhaya Sindhu Battin "Devops, A New Approach To Cloud Development & Testing", Journal of Emerging Technologies and Innovative Research (JETIR), Vol.7, Issue.8, pp.982-985, 2020.
- [6] Arpita S.K, Amrathesh, Dr. Govinda Raju M "A review on Continuous Integration, Delivery and Deployment using Jenkins", Journal of University of Shanghai for Science and Technology, Vol.23, Issue.6, pp.919-922, 2021.
- [7] Pooja D Pandit "A Case Study of Amazon Web Services", ResearchGate, 2021.
- [8] Rahul Saini, Rachna Behl "An Introduction to AWS – EC2 (Elastic Compute Cloud)", International Conference on Research in Management & Technovation, Vol.24, pp.99-102, 2020.
- [9] Eleni Rütz, Martin, Fachhochschule Wedel, Wedel "DEVOPS: A SYSTEMATIC LITERATURE REVIEW", ResearchGate, 2019.

- [10] Laiby Thomas, Subramanya Bhat “A Comprehensive Overview of Telegram Services - A Case Study”, International Journal of Case Studies in Business, IT, and Education (IJCSBE), Vol.6, No.1, pp.288-301, 2022.
- [11] Pallavi Deshwal, Poonam Ghuli “DevOps: Concept, Technology and Tools”, International Journal of Computer Sciences and Engineering, Vol.8, Issue.6, pp.73-78, 2020.
- [12] Mahadevi S. Namose, Shobha D. Patil “Standard DevOps Pipeline”, International Journal of Computer Sciences and Engineering, Vol.7, Issue.3, pp.641-646, 2019.

AUTHORS PROFILE

N. Meghana is a student at the Vasireddy Venkatadri Institute of Technology in Nambur, Guntur District, Andhra Pradesh, where she is pursuing an IV B. tech in information technology.

K. Sri Lakshmi is a student at the Vasireddy Venkatadri Institute of Technology in Nambur, Guntur District, Andhra Pradesh, where she is pursuing an IV B. tech in information technology.

M. Naga Lakshmi Teja Sree is a student at the Vasireddy Venkatadri Institute of Technology in Nambur, Guntur District, Andhra Pradesh, where she is pursuing an IV B. tech in information technology.

K. Srujana is a student at the Vasireddy Venkatadri Institute of Technology in Nambur, Guntur District, Andhra Pradesh, where she is pursuing an IV B. tech in information technology.

N. Ashok working as Associate Professor in the stream of Information Technology at Vasireddy Venkatadri Institute of Technology, Nambur, Guntur District, Andhra Pradesh.
