



# **OPSIFY: An Intelligent DevOps Chatbot for Streamlined Operations**

**D.Aswani<sup>1</sup>, N Teena<sup>2</sup>, K Pradeep<sup>3</sup>, N Santhosh Nayak<sup>4</sup>**

Assistant Professor, Department of CSE, ACE Engineering College, Hyderabad, Telangana, India<sup>1</sup>

IV B. Tech. Students, Department of CSE, ACE Engineering College, Hyderabad, Telangana, India<sup>2,3,4</sup>

**ABSTRACT:** Opsify is an intelligent DevOps chatbot that is specifically designed to revolutionize the way in which complex operational activities are handled by developers and system administrators. Utilizing highly sophisticated AI technologies like Retrieval-Augmented Generation (RAG), Opsify is capable of generating accurate, contextual, and domain-specific answers that are retrieved from highly reliable DevOps knowledge bases. Unlike other traditional chatbots, Opsify is also capable of understanding even gibberish inputs or unstructured data, thanks to its highly intelligent ability to understand user intent. Developed using a highly reliable FastAPI backend with WebSocket capabilities, Opsify is also containerized using Docker, making it highly scalable in any operational environment. Apart from answering complex DevOps-related questions, Opsify is also capable of automating routine DevOps activities like system monitoring, CI/CD troubleshooting, and infrastructure configuration. Developed specifically for today's highly demanding DevOps environments, Opsify is not only capable of answering complex DevOps-related questions but also provides highly reliable assistance in enhancing system stability. With its highly reliable integration with AI technologies, Opsify is highly distinguished from other traditional chatbots that were developed using highly unreliable AI technologies.

**KEYWORDS:** Large Language Models, Retrieval-Augmented Generation, Docker Containerization, CI/CD Pipelines, Jenkins Automation, Natural language processing.

## **I. INTRODUCTION**

Opsify is a smart chatbot for DevOps that utilizes Retrieval-Augmented Generation (RAG) for delivering accurate and context-specific responses based on curated and live data. It can effectively handle noisy inputs and perform tasks such as troubleshooting and system monitoring, enhancing user productivity. However, RAG-based models are complex in terms of effectively managing dynamic data and system reliability. The existing LLMOps primarily targets models but does not consider RAG-based data management. The RAGOps framework addresses this issue by supporting data lifecycle management, observability, and continuous monitoring. The Opsify and RAGOps duo offers a reliable and adaptive AI virtual assistant for dynamic environments.

## **II. LITERATURE SURVEY**

According to recent research, Artificial Intelligence is used in DevOps to make work faster and easier through AI systems.

The research paper Transforming DevOps with Intelligent Automation (2024) explains how AI can automate work and prevent system crashes, but it does not use advanced techniques like RAG and cannot handle ambiguous user input. Retrieval-Augmented Generation for Knowledge-Intensive NLP (2020) uses AI and data for more accurate results but may not be suitable for real-time DevOps.

RAGOps: Operating and Managing Retrieval-Augmented Generation Pipelines (2025) shows optimal RAG pipelines but is difficult to use and does not address real-world DevOps issues like ambiguous input.

These studies improve work quality, but none integrate all techniques or support real-time user interaction as the proposed Opsify system does.

### III. METHODOLOGY

The proposed system, Opsify, is an intelligent chatbot that uses the Retrieval-Augmented Generation (RAG) method to ensure that accurate responses are provided. The chatbot is developed in such a manner that it provides accurate responses to user queries pertaining to DevOps activities. DevOps activities include CI/CD issues, system monitoring, infrastructure issues, etc. However, the user query is not always in a structured format. Therefore, first, the user query is preprocessed using NLP techniques. This is done in order to understand the user query.

Once the user query is preprocessed, accurate responses are provided using the RAG method. This is done using the context provided by the knowledge base developed using the Opsify system pertaining to DevOps activities. Using the context, accurate responses are provided to the user query. This is helpful in resolving user problems, identifying user problems, and providing solutions pertaining to DevOps activities.

The Opsify system is developed using FastAPI with WebSocket. This is helpful in providing real-time communication with the user. Moreover, Docker is used in order to containerize the system. This is helpful in deploying the system. Finally, the response is displayed using the chatbot interface.

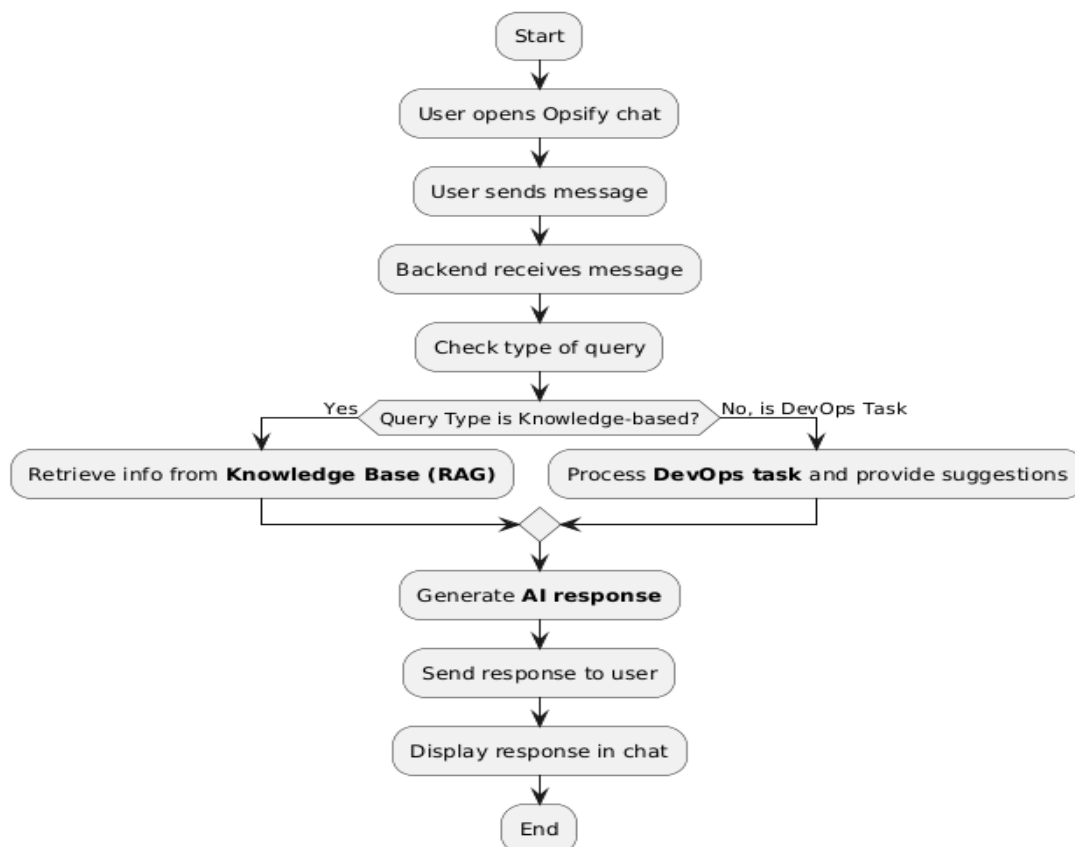


Fig 3.1 Opsify Methodology

### IV. SYSTEM ARCHITECTURE

The system architecture of the Opsify system is designed in such a way that it provides a smooth and intelligent DevOps chatbot experience. The system architecture of the Opsify system is such that the user can interact with the system by sending user queries to the system.

The user query is then sent to the FastAPI backend. The FastAPI backend is the central hub that handles all the user queries and coordinates with the other components of the system to process the user query in an efficient way. The system is designed in such a way that it provides fast and accurate results for complex DevOps-related user queries. The system uses state-of-the-art technologies that make the workflow seamless and user-friendly. Once the user query is received by the system, it is sent to the RAG module.

The RAG module sends the user query to the Vector Database (such as FAISS or Chroma). Once the user query is sent to the Vector Database, the data is fetched. The user query is then sent to the AI engine (such as the OpenAI GPT engine). Once the user query is sent to the AI engine, the user query is answered in an accurate way. Once the user query is answered in an accurate way, it is sent to the response formatter.

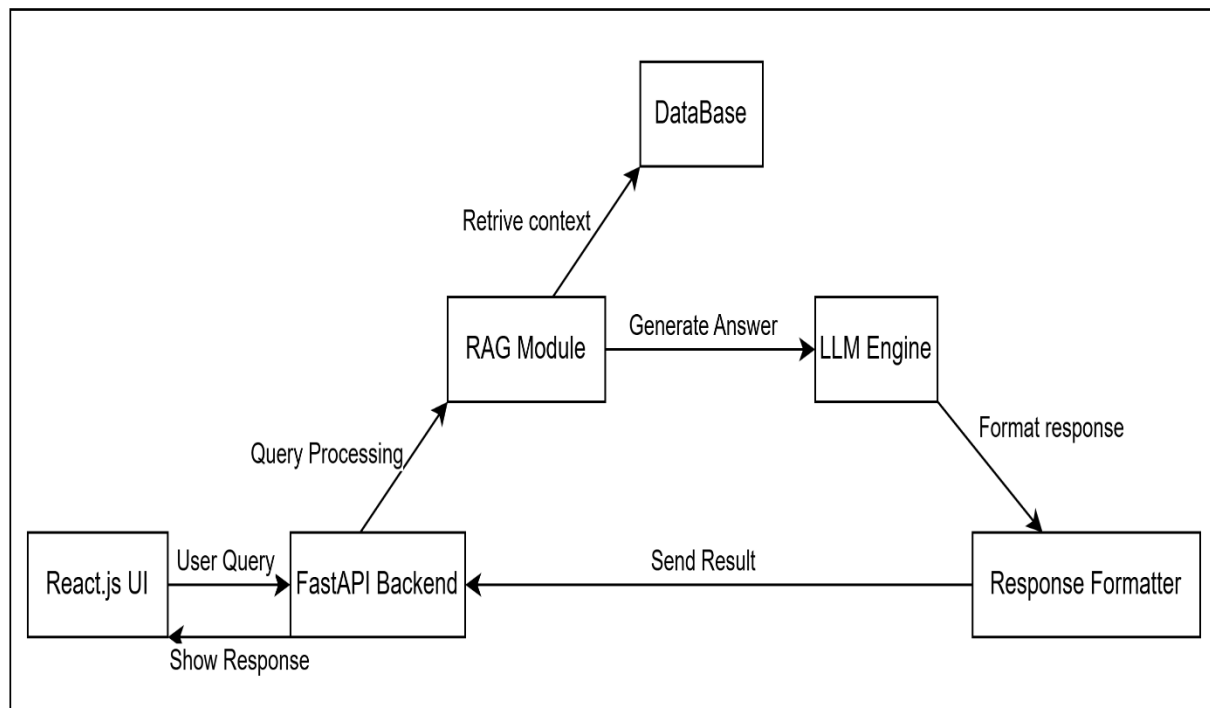


Fig 4.1 System Architecture

## V. RESULTS

Opsify allows a user to converse with the system to seek assistance for any task. When a user types a query, the answer appears on the screen.

The system reads and understands the query through smart language processing. It then presents a useful and understandable answer based on the user's need.

The answer appears in a simple way so that anyone can understand it. The user can also save or download the result as PDF.

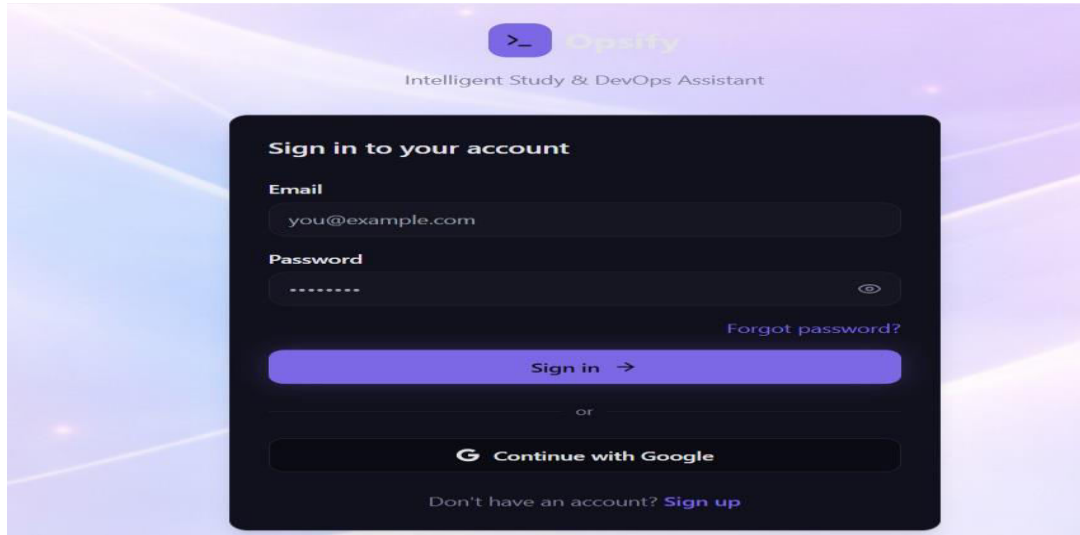


Fig 5.1 Opsify Login Page

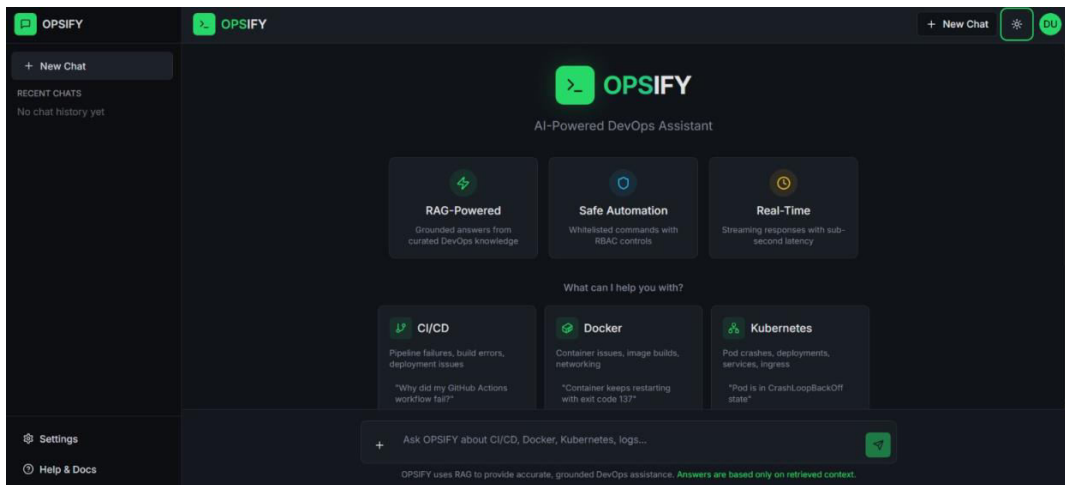


Fig: 5.2 User Interface Page

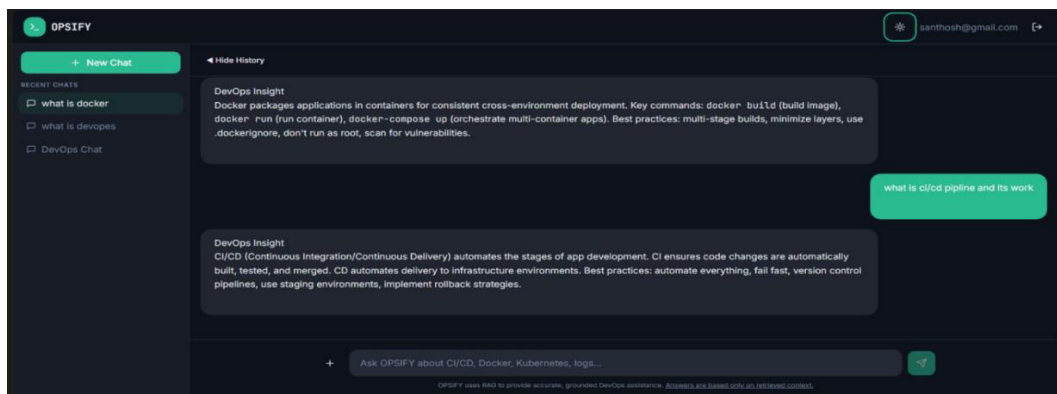


Fig 5.3 Query output page

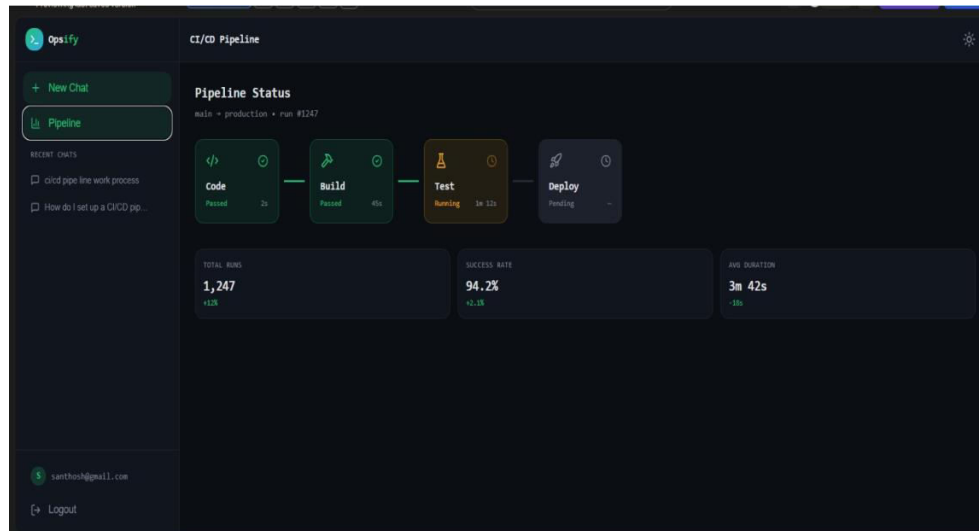


Fig 5.4 Opsify CI/CD Pipeline Page

## VI. CONCLUSION

Opsify is an intelligent chatbot for DevOps operations, which can simplify complex operations using Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) to offer accurate, context-aware responses to noisy queries. It is beyond a simple chatbot, as it can integrate with CI/CD pipelines, monitoring tools, and infrastructure scripts to automate build, debug, optimize, and predict operations. With real-time interaction through WebSockets, deployment through Docker, and scalable design, Opsify reduces manual efforts, avoids risks, and increases productivity for the team, making it a next-generation chatbot designed to improve modern DevOps operations.

## VII. FUTURE SCOPE

However, it is possible to make Opsify better in the future by employing other sophisticated AI models that will provide quick and accurate solutions for DevOps. Another possibility is to make it capable of supporting other languages, making it easy for users from different parts of the world to utilize it. Another improvement that can be made to Opsify is to integrate it with other popular DevOps tools like CI/CD tools, monitoring tools, and communication tools. Another improvement that can be made is to include smart recommendations that will be helpful in solving problems and improving system performance.

## REFERENCES

- [1] Xu, X., Weytjens, H., Zhang, D., Lu, Q., Weber, I., and Zhu, L. (2025). RAGOps: Operating and Managing Retrieval-Augmented Generation Pipelines. arXiv / SSRN. Link: <https://arxiv.org/abs/2506.03401>
- [2] Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., and Heinrich, E. (2025). Retrieval-Augmented Generation for Knowledge-Intensive NLP. arXiv. Link: <https://arxiv.org/abs/2005.11401>
- [3] Baqar, M., Naqvi, S., and Khanda, R. (2025). AI-Augmented CI/CD Pipelines: From Code Commit to Production with Autonomous Decisions. arXiv. Link: <https://arxiv.org/abs/2508.11867>
- [4] Brown, A., Roman, M., and Devereux, B. (2025). A Systematic Literature Review of Retrieval-Augmented Generation: Techniques, Metrics, and Challenges. arXiv. Link: <https://arxiv.org/abs/2508.06401>
- [5] Kodithyala, S. K. (2025). AI-Enhanced Platform Engineering: Revolutionizing CI/CD Pipelines Through Intelligent Automation. Link: <https://ijcesen.com/index.php/ijcesen/article/view/4248>
- [6] "Transforming DevOps with Intelligent Automation," Industry Report (2024) Link: <http://wjarr.com/content/transforming-devops-artificial-intelligence-deep-dive-intelligent-automation-predictive>.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details