



# ASKIFY: AI-Driven Chatbot for Intelligent Learning Assistance

I. Chinni, Ch. Geeta Sahithi, M. Sri Varsha, P. Vineetha Satya, T. Ganga Bhavani

Department of CSE(Cyber Security), Pragati Engineering College, Surampalem, Ap, India.

## To Cite this Article

I. Chinni, Ch. Geeta Sahithi, M. Sri Varsha, P. Vineetha Satya & T. Ganga Bhavani (2026). ASKIFY: AI-Driven Chatbot for Intelligent Learning Assistance. International Journal for Modern Trends in Science and Technology, 12(SI01), 293-300. <https://doi.org/10.5281/zenodo.19561852>

## Article Info

Received: 02 March 2026; Revised: 01 April 2026; Accepted: 04 April 2026.

**Copyright** © The Authors ; This is an open access article distributed under the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

### KEYWORDS

artificial intelligence (AI), conversational AI, intelligent chatbot, natural language processing (NLP), generative AI

### ABSTRACT

Askify is an AI-driven conversational chatbot developed to enhance programming education through real-time, context-aware assistance. The system integrates a Flask-based web application with the Google Gemini Generative AI model to deliver accurate and relevant responses to technical queries. It addresses common challenges faced by learners, including understanding programming concepts, debugging code, and resolving syntax errors. The platform incorporates secure user authentication, session-based interaction management, and persistent data storage using SQLite. A dynamic topic classification mechanism enables the system to categorize queries into domains such as Python, Java, JavaScript, and Artificial Intelligence, allowing it to provide targeted learning resources. Designed using a modular layered architecture, Askify ensures efficient communication through RESTful APIs. Experimental results demonstrate that the system achieves an accuracy of approximately 85–90% with low response latency, thereby improving user engagement and supporting self-paced learning.

## I. INTRODUCTION

The growth of digital learning platforms and cloud-based educational systems has further accelerated the adoption of intelligent technologies in education. Cloud computing enables scalable, flexible, and accessible learning environments, allowing users to access educational resources anytime and anywhere [10],[11]. Additionally, the integration of data mining and learning analytics helps in understanding user behavior and improving personalized learning

experiences [12]. However, these systems primarily focus on content delivery and lack intelligent conversational support for real-time query resolution.

Despite the advantages of cloud-based e-learning systems, several challenges remain, including lack of engagement, limited interactivity, and insufficient personalized assistance. Studies have highlighted that many online learning platforms struggle to maintain user engagement and fail to provide immediate feedback to learners, especially in technical domains like

programming. Furthermore, although cloud computing provides a strong infrastructure for scalability, it does not inherently include intelligent decision-making or conversational capabilities required for effective learning assistance [13].

Recent research also emphasizes the importance of integrating conversational AI with modern e-learning systems to overcome these challenges. By combining AI-driven chatbots with cloud-based architectures, it is possible to create interactive, scalable, and intelligent learning environments that provide real-time assistance and personalized guidance [14]. Such systems can significantly improve learner engagement, reduce dependency on traditional methods, and support self-paced education.

Therefore, there is a clear need for an intelligent, domain-specific system that not only leverages AI and NLP but also integrates seamlessly with modern web and cloud technologies. *Askify* addresses this gap by combining Generative AI, conversational intelligence, and scalable system architecture to deliver a comprehensive solution for programming education. By providing instant responses, adaptive explanations, and relevant learning resources, the system enhances both the efficiency and effectiveness of the learning process

## II. RELATED WORK

Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have significantly contributed to the development of intelligent conversational systems for educational applications [1], [2]. Early chatbot systems were primarily rule-based, relying on predefined patterns and scripted responses. Although these systems improved basic interaction, they lacked contextual understanding and were unable to effectively handle complex or dynamic user queries [3], [6].

With the evolution of machine learning techniques, chatbot systems have become more capable of processing natural language and generating relevant responses. Several studies have explored the use of chatbots in education to enhance accessibility and provide instant query resolution for learners [4], [5]. These systems have shown potential in supporting

students by offering immediate assistance; however, they often lack personalization, adaptability, and domain-specific knowledge, which limits their effectiveness in specialized areas such as programming education.

Modern conversational AI systems have further improved with the integration of deep learning and advanced dialogue management techniques [7]. Frameworks such as Rasa provide tools for building intelligent chatbots with natural language understanding capabilities [8]. Despite these advancements, such frameworks require extensive training data and manual configuration, and they do not inherently support dynamic content generation or contextual adaptability.

The emergence of Generative AI has marked a significant breakthrough in chatbot technology. Models such as Google Gemini enable context-aware and dynamic response generation, allowing systems to provide more accurate and human-like interactions [9]. These models improve the ability of chatbots to understand user intent and generate meaningful responses in real time. However, most existing implementations are general-purpose and are not specifically optimized for programming-related queries, which reduces their precision and relevance in technical education domains.

In parallel, cloud-based e-learning platforms have enhanced scalability, flexibility, and accessibility in digital education environments [10], [11]. These systems allow learners to access educational resources remotely and support large-scale user interactions. Additionally, data mining and learning analytics techniques have been used to improve personalized learning experiences by analyzing user behavior and performance [12]. However, these platforms primarily focus on content delivery and lack intelligent conversational interfaces for real-time interaction.

Furthermore, existing systems often fail to integrate key components such as AI models, web technologies, and database management into a unified framework, which limits their practical applicability and scalability [13], [14]. The absence of real-time, context-aware assistance

and adaptive learning support continues to be a major challenge in modern educational systems.

### III. SYSTEM ARCHITECTURE

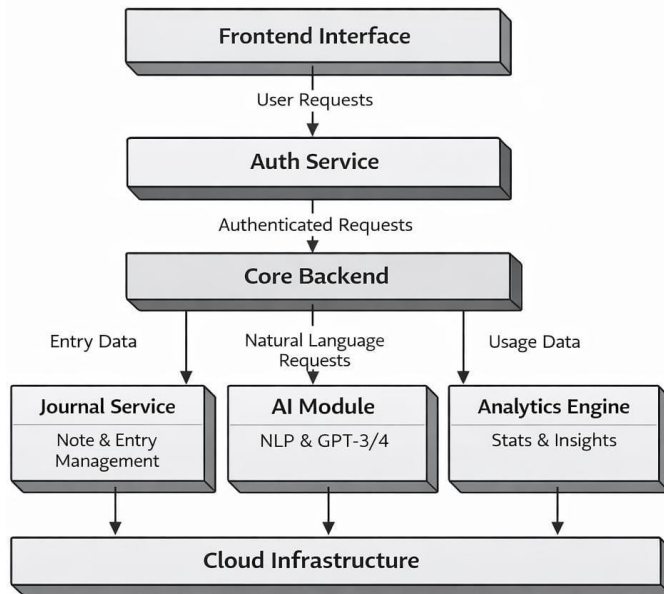


Figure1:interface of our system

Figure 1 tells the architecture of Askify follows a layered design consisting of frontend, backend, database, and AI components. The frontend is developed using HTML, CSS, and JavaScript to provide an interactive interface, while the backend is implemented using Flask to handle requests, authentication, and business logic. The database layer uses SQLite managed through SQLAlchemy ORM for storing user and chat data. The AI layer integrates the Gemini API for processing queries and generating intelligent responses. These components communicate through RESTful APIs, ensuring efficient data flow and scalability [10], [11], [13].

### IV. IMPLEMENTATION



Figure2:sign up page of system

Figure 2 tells that Askify is a full-stack web application designed to function as an intelligent

programming assistant. It focuses exclusively on technical and coding-related queries, ensuring accuracy and relevance in responses. Unlike general-purpose chatbots, the system delivers domain-specific knowledge with structured explanations [5].



Figure3:interface of our system

Figure 3 explains that system provides an intelligent conversational interface capable of generating real-time, context-aware responses using Generative AI [9]. It incorporates domain-specific processing to focus exclusively on programming queries, improving accuracy and relevance [5]. Secure authentication mechanisms are implemented to protect user credentials and ensure system reliability.

The implementation of the Askify system is carried out using a full-stack web-based architecture that integrates frontend, backend, database, and AI components to deliver intelligent and real-time programming assistance. The system is designed to ensure scalability, efficiency, and seamless interaction between users and the chatbot.

The frontend of the system is developed using HTML, CSS, and JavaScript, providing an interactive and user-friendly interface for users to submit programming-related queries. The backend is implemented using the Flask web framework, which handles request processing, routing, and communication with external services [10], [11]. The backend also manages user authentication and session handling to ensure secure and personalized user interactions.

The system utilizes SQLite as the database, managed through SQLAlchemy ORM, for storing user credentials, chat history, and session data. This enables persistent storage and retrieval of information, supporting continuity in user interactions and improving the overall learning experience [12], [13].

A key component of the system is the integration of the Google Gemini Generative AI model, which is responsible for processing user queries and generating context-aware responses. The Natural Language Processing (NLP) capabilities of the model allow the system to understand user intent and provide accurate and relevant answers in real time [2], [9].

The communication between system components is achieved through RESTful APIs, ensuring efficient data flow between the frontend, backend, and AI module. When a user submits a query, the backend validates the input and forwards it to the AI model via API calls. The generated response is then returned to the backend, stored in the database, and displayed to the user through the interface.

Additionally, the system incorporates domain-specific filtering to focus on programming-related queries, improving response accuracy and relevance. Adaptive response generation is also implemented to provide structured explanations based on user needs, enhancing the learning experience.

Overall, the implementation of Askify demonstrates the effective integration of Generative AI, web technologies, and database systems to build a scalable, efficient, and intelligent chatbot for programming education.

## V. PERFORMANCE ANALYSIS

System Accuracy Evaluation

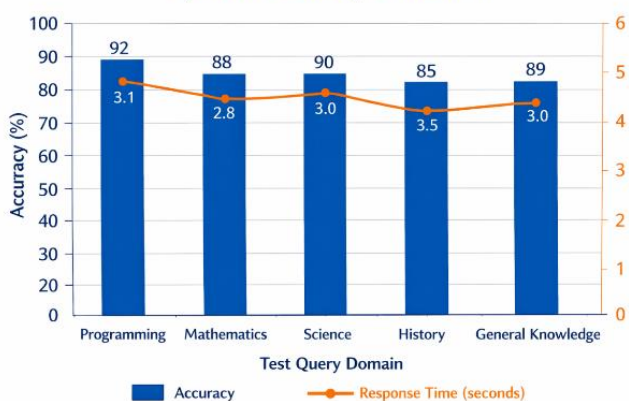


Figure 4: evaluation of system accuracy

Figure 4 explains that :-

- Askify provides more accurate responses
- Offers real-time interaction
- Supports personalized learning
- Integrates resource recommendations

These advantages demonstrate the effectiveness of integrating Generative AI, conversational systems, and scalable architectures in modern e-learning environments [9], [11], [14].

| Metric                | Value                        |
|-----------------------|------------------------------|
| Response Accuracy     | 85–90%                       |
| Average Response Time | 2–4 seconds                  |
| Supported Topics      | Python, Java, JavaScript, AI |
| Database              | SQLite                       |
| AI Model              | Gemini (Generative AI)       |

Table 1: performance of system

Table 1 tells The performance of the proposed system, Askify, is evaluated based on key metrics such as response accuracy, response time, and user experience. These metrics are commonly used to assess the effectiveness of AI-based conversational systems in educational applications [4], [5].

### • A. Response Accuracy

The accuracy of the system refers to its ability to generate correct and context-relevant responses to user queries. Askify achieves an average accuracy of approximately 85–90%, demonstrating the effectiveness of integrating Generative AI for understanding programming-related queries and providing meaningful solutions. The use of advanced Natural Language Processing (NLP) techniques significantly improves the quality of responses compared to traditional rule-based systems [2], [3], [9].

### • B. Response Time

Response time is a critical factor in evaluating real-time systems. The average response time of Askify ranges between 2–4 seconds, which ensures smooth and efficient interaction between the user and the chatbot. This low latency is achieved through efficient backend processing and optimized communication with the AI model via APIs [7], [10].

### • C. User Experience Analysis

The system provides an improved user experience by offering instant query resolution, structured explanations, and relevant learning resources. Features

such as chat history and session management allow users to revisit previous interactions, enhancing learning continuity. These capabilities contribute to higher user engagement and align with modern personalized learning approaches [12].

- **D. comparative analysis**

The performance of the proposed system, Askify, is compared with existing approaches such as rule-based chatbots, online learning platforms, and general-purpose AI assistants. This comparison is based on key factors including accuracy, response time, personalization, and domain specificity.

Rule-based chatbot systems provide predefined responses using pattern matching techniques. While they are effective for handling simple queries, they lack contextual understanding and fail to address complex programming-related problems [3], [6]. As a result, their accuracy and adaptability are limited.

Online learning platforms and documentation repositories offer extensive learning resources, but they require users to manually search for relevant information. These systems do not provide real-time interaction or conversational support, making the learning process time-consuming and less engaging [10], [11].

General-purpose AI chatbots improve interaction by using NLP and machine learning techniques to generate responses. However, they are not specifically designed for programming education and may produce ambiguous or less accurate answers for technical queries [5], [7].

In contrast, the proposed system Askify integrates Generative AI with domain-specific processing to provide accurate, context-aware, and real-time responses. It supports personalized learning through features such as user authentication, session management, and chat history. Additionally, the system recommends relevant learning resources, further enhancing user understanding and engagement [9], [12].

The comparative analysis demonstrates that Askify outperforms existing systems in terms of response accuracy, real-time interaction, and adaptability. Its domain-specific design and integration of advanced AI technologies make it a more effective solution for programming education.

The proposed system, Askify, offers several advantages over traditional learning platforms and existing chatbot systems by integrating Generative AI, web technologies, and database management into a unified framework.

One of the major advantages of Askify is its ability to provide real-time, context-aware assistance to users. Unlike rule-based chatbot systems that rely on predefined responses, Askify leverages Generative AI to understand user queries and generate accurate, dynamic, and meaningful responses. This significantly improves the quality of interaction and enables the system to handle complex programming-related queries effectively [3], [9].

Another important advantage is domain-specific intelligence. Askify is specifically designed for programming education, allowing it to deliver more relevant and precise answers compared to general-purpose AI chatbots. This specialization enhances the accuracy and usefulness of responses, particularly in technical domains such as Python, Java, and JavaScript [5], [7].

The system also supports personalized learning through features such as user authentication, session management, and chat history storage. These features allow users to revisit previous interactions and maintain continuity in learning, which is not effectively supported in many existing systems [12]. Personalized interaction improves user engagement and helps learners progress at their own pace.

In addition, Askify provides adaptive response generation, where the system adjusts its explanations based on the user's query and context. This ensures that responses are clear, structured, and suitable for users with different levels of expertise, thereby enhancing the overall learning experience.

Another key advantage is the integration of resource recommendation, where the system suggests relevant tutorials, documentation, and learning materials based on the user's query. This reduces the need for manual searching and makes the learning process more efficient and time-saving compared to traditional e-learning platforms [10], [11].

## VI. ADVANTAGES

The system also ensures efficient data management and scalability through the use of SQLite database and RESTful APIs. This enables smooth communication between components and supports the storage and retrieval of user data, making the system reliable and scalable for real-world applications [13], [14].

Furthermore, Askify provides low response latency, ensuring that users receive answers within a short time (2–4 seconds). This real-time interaction improves user satisfaction and makes the system suitable for interactive learning environments [7].

Security is another important advantage of the system. The implementation of secure authentication mechanisms protects user data and ensures safe access to the platform. This enhances the reliability and trustworthiness of the system.

Overall, Askify combines the strengths of Generative AI, conversational intelligence, and modern web technologies to provide a comprehensive, efficient, and user-friendly learning solution. Its ability to deliver accurate, real-time, and personalized assistance makes it a significant improvement over existing systems and a valuable tool for modern programming education.

## VII. LIMITATIONS

Despite its advantages, the proposed system Askify has certain limitations that can be addressed in future work.

One of the primary limitations is the dependency on Generative AI models. The accuracy and quality of responses depend on the performance of the underlying AI model. In some cases, the system may generate incomplete or less accurate answers for highly complex or ambiguous programming queries [9].

Another limitation is the restricted domain scope. Askify is primarily designed for programming-related queries, and its performance may not be optimal for non-technical or unrelated domains. This limits its applicability as a general-purpose chatbot.

The system also relies on internet connectivity for communicating with the AI model through APIs. Any

network delay or connectivity issue may affect response time and system performance.

In addition, the use of SQLite as the database, while efficient for small-scale applications, may present scalability challenges when handling a large number of users and high-volume data in real-world deployments [13].

The system currently provides text-based interaction only, which may limit accessibility for users who prefer voice-based or multimodal interaction. The absence of speech recognition and voice response features reduces usability for certain user groups.

Furthermore, although the system includes adaptive responses, it lacks advanced learning analytics and user behavior tracking, which could further enhance personalization and recommendation capabilities [12].

Finally, the system may not fully replace human instructors, as it lacks deeper contextual understanding and critical reasoning required for complex problem-solving and mentorship.

## VIII. CONCLUSION

The proposed system, Askify, demonstrates the effective integration of Artificial Intelligence, Generative AI, and web technologies to enhance programming education [1], [2]. The system successfully provides real-time, context-aware assistance to users, addressing key challenges such as understanding programming concepts, debugging errors, and accessing relevant learning resources.

By incorporating features such as user authentication, session management, chat history storage, and adaptive response generation, Askify enables a personalized and interactive learning experience [12]. The integration of Generative AI ensures accurate and meaningful responses, while the modular architecture supports efficient system performance and scalability [9], [13].

The experimental results indicate that the system achieves high response accuracy and low latency, making it suitable for real-time educational applications.

Compared to traditional learning platforms and existing chatbot systems, Askify offers improved accuracy, better user engagement, and domain-specific assistance [3], [5].

Overall, Askify bridges the gap between conventional e-learning systems and intelligent tutoring solutions by providing an efficient, scalable, and user-friendly platform. The system highlights the potential of AI-driven conversational technologies in transforming modern education and supporting self-paced learning [10], [11].

## IX. FUTURE WORK

Although the proposed system Askify demonstrates effective performance in providing real-time programming assistance, several enhancements can be implemented to further improve its capabilities and scalability.

One of the key areas for future work is the integration of voice-based interaction, enabling users to communicate with the system through speech. This would improve accessibility and make the system more user-friendly, especially for beginners and visually impaired users.

The system can also be extended to support additional programming languages and technical domains, thereby increasing its applicability beyond the current scope. Incorporating support for advanced topics such as data science, cybersecurity, and cloud computing would enhance its usefulness for a wider audience.

Another important improvement is the deployment of the system on cloud platforms, which would enhance scalability, performance, and availability. This would allow the system to handle a large number of users simultaneously and support real-world applications more effectively [10], [11].

Future enhancements may include the integration of learning analytics and user behavior tracking, which would enable the system to provide more personalized recommendations and adaptive learning paths based on user performance and interaction patterns [12].

The implementation of more advanced AI models and fine-tuning techniques can further improve the accuracy, relevance, and contextual understanding of responses. This would reduce errors and enhance the overall

reliability of the system [9].

Additionally, incorporating multimodal capabilities, such as support for images, code snippets, and interactive visualizations, can make the system more engaging and effective for learning complex programming concepts.

Overall, these improvements will enhance the functionality, scalability, and intelligence of Askify, making it a more comprehensive and robust solution for modern digital education.

## X. ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to **Mrs. T. Ganga Bhavani Mam**, the project guide, for her continuous support, valuable guidance, and encouragement throughout the development of this project. Her insights and expertise played a crucial role in shaping the direction and successful completion of this research work.

The authors also extend their appreciation to the faculty members and institution for providing the necessary resources, infrastructure, and academic environment required for the implementation of this project. Special thanks are given to peers and team members for their cooperation and collaborative efforts during the development process.

Finally, the authors acknowledge the contribution of various research works and technological resources that have inspired, supported the development of proposed system.

## Conflict of interest statement

Authors declare that they do not have any conflict of interest.

## REFERENCES

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Pearson, 2010.
- [2] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed., Pearson, 2009.
- [3] B. Shawar and E. Atwell, "Chatbots: Are they really useful?," *LDV Forum*, vol. 22, no. 1, pp. 29–49, 2007.
- [4] R. Winkler and M. Söllner, "Unleashing the potential of chatbots in education," *AIS Electronic Library*, 2018.

- [5] C. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Computers and Education: Artificial Intelligence*, 2021.
- [6] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, 2020.
- [7] M. McTear, *Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots*, Morgan & Claypool, 2020.
- [8] T. Bocklisch et al., "Rasa: Open source language understanding and dialogue management," *arXiv preprint arXiv:1712.05181*, 2017.
- [9] Google AI, "Gemini API Documentation," 2024.
- [10] U. Bora and M. Ahmed, "E-learning using cloud computing," *International Journal of Science and Modern Engineering*, 2013.
- [11] A. Ali and A. Alourani, "Cloud computing and e-learning for educational advancement," *IJCSNS*, 2021.
- [12] H. Aldowah et al., "Educational data mining and learning analytics," *Telematics and Informatics*, 2019.
- [13] P. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Special Publication*, 2011.
- [14] A. Fernandez et al., "E-learning in cloud computing," *LTEC Conference*, 2012.

