OPEN ACCESS JOURNALS
Indiana Journal of Multidisciplinary Research
Abbreviated Tittle- Ind J Multi Res
ISSN *(Online)*- 2583-3820
Journal Homepage Link- https://indianapublications.com/Journals/IJMR
DOI: https://doi.org/10.5281/zenodo.21559399

INDIANA PUBLICATIONS
PRODUCTIVE AND QUALITY RESEARCH

Research Article

Volume-06|Issue-04|2026

Rule Based Approach for Conversational AI Chatbot using Log pattern Detection

Tamil Iniyan KS[1], Shashikala HK[2], Rakesh M[3], Adeebulla Khan[4], Umar Ahmed Khan[5], Rahul Das[6]

[1,3,4,5,6]B. Tech 4th Year Department of Computer Science and Engineering Faculty of Engineering and Technology Jain (Deemed-to-be) University Bangalore, India.
[2]Professor Department of Computer Science and Engineering Faculty of Engineering and Technology Jain (Deemed-to-be) University Bangalore, India

Article History
Received: 05.05.2026
Accepted: 22.06.2026
Published: 25.07.2026

Citation
Tamil, I. K. S., Shashikala, H. K., Khan, A., Khan, U. A. & Das, R. (2026). Rule Based Approach for Conversational AI Chatbot using Log pattern Detection. *Indiana Journal of Multidisciplinary Research, 6*(4), 69-74.

Abstract: This paper serves as a focused review of one of the methodologies introduced in the larger project titled "Conversational AI Chatbot using Log Pattern Detection", which presents a smart, proactive system capable of identifying and addressing issues in advance. The approach discussed here centers on the rule-based model, specifically applied when errors have already taken place. To illustrate this method, a movie ticket booking platform has been developed with a Dialog flow- integrated chatbot. This chatbot handles user concerns such as booking confirmation issues, unavailable seats, or payment disruptions by relying on a set of predefined rules and conditional logic. By automating responses to frequent problems, the system offers immediate assistance and minimizes the need for human intervention. The platform is supported by a MongoDB backend and a user-friendly interface, ensuring seamless interaction. The paper highlights the relevance of rule-based chatbots in addressing known issues effectively within a specific domain.

Keywords: Rule-based chatbot, Dialogflow, error recovery, predefined logic, user assistance, movie ticket booking

INTRODUCTION

With the growing reliance on digital services, conversational interfaces have become integral in offering prompt user support and enhancing system interaction. [1] Among the various types of AI-enabled assistants, rule-based chatbots stand out for their clarity, dependability, and structured handling of repetitive user issues. This paper delves into the practical application of a rule- based chatbot within a movie ticket booking website, illustrating one of the core strategies explored in the broader research "Conversational AI Chatbot Using Log Pattern Detection."

In contrast to intelligent systems that anticipate and correct issues before they manifest, rule-based systems operate in a reactive manner, offering assistance after a problem has occurred.[2] These systems rely on predefined logic and decision trees that enable the chatbot to follow a step-by-step troubleshooting path. [3] They are especially effective in domains where most queries fall into well- defined categories and don't require dynamic decision-making.

As part of this project, [4] a Dialogflow-powered chatbot has been integrated into a movie booking platform to assist users with common problems such as confirming bookings, handling failed payments, addressing seat availability, and answering frequently asked questions. It uses basic if-else logic to interpret user inputs and respond with relevant information. [5] This rule- based interaction model ensures uniformity in responses and allows for immediate assistance, reducing dependency on customer support agents for straightforward issues.

The website includes multiple user-oriented modules, such as secure authentication, real-time movie listings, theatre selection interfaces, and a flexible payment system [6]. These components are backed by a MongoDB database, ensuring seamless data access and synchronization. [7] The chatbot serves an initial support layer, helping users navigate common obstacles efficiently and improving the overall experience.

This paper focuses specifically on the rule-based chatbot mechanism [8] as a practical and accessible solution for handling post-error conditions. It demonstrates how predefined rules and logic trees can streamline user support in platforms where issues are predictable and repetitive, ultimately contributing to improved system efficiency and user satisfaction [9].

LITERATURE SUREVY

[10] The paper "Conversational AI Chatbot Using Log Pattern Detection" by Tamil Iniyan and colleagues presents a smart chatbot designed to improve

***Corresponding Author:** Shashikala HK

user support and system reliability by analyzing system logs. Built using AWS tools like Lex, Lambda, Connect, SageMaker, and Personalize, the chatbot actively monitors user interactions and log data to catch issues before they escalate. It combines rule-based responses, self- healing functions, and human agent support to deliver smooth and personalized conversations. By spotting unusual patterns in logs, the system can alert support staff or take automated corrective actions—helping reduce system downtime and keep users happy. The use of machine learning, particularly LSTM models, adds predictive power to detect and resolve problems early. Overall, this work highlights how blending AI-driven chatbots with log analysis can create smarter, more responsive support systems.

[11] This study by Riandika, Susanto, and Respatria focuses on building a rule-based chatbot using the Dialogflow platform to streamline how the public accesses information on a government website. Specifically designed for the Semarang City Health Office, the chatbot answers frequently asked questions about food licensing services. By using predefined intents and sample questions, it delivers accurate, consistent responses—helping reduce the burden on human staff and speeding up the overall process. The team used black-box testing to ensure the chatbot worked as expected and met user needs. Overall, the project aims to make public services more efficient, accessible, and user-friendly through a structured, rules-based conversational system.

[12] In their paper, Pham and Cu present a smart scheduling system for service call centers that uses a rule- based approach to analyze large volumes of log data. The system combines big data techniques with expert- defined rules to create flexible work schedules tailored to both staff experience and customer demand. By automatically reviewing past service logs, it helps reduce staff overload, streamline operations, and boost customer satisfaction. The model was tested in real-world settings, including public services related to forestry, showing how effective and practical this combination of data analysis and rule-based decision-making can be in managing service teams more efficiently.

[13] Qi and colleagues introduce LogGPT, a novel framework that uses ChatGPT to detect anomalies in system logs by treating them like natural language. The system works by first cleaning and formatting raw log data, then crafting prompts that ChatGPT can understand, and finally interpreting its responses to identify unusual patterns. Tested on well-known datasets like BGL and Spirit, LogGPT holds its own against deep learning models like DeepLog and LogAnomaly. What sets it apart is its ability to not only detect problems effectively but also explain them in a more understandable way—making it a valuable tool for system monitoring where clarity and insight matter just as much as accuracy.

[14] In his master's thesis, "KINO: An Approach for Rule- Based Chatbot," T. D'Avila introduces a practical method for creating chatbots that follow structured, rule-based dialogue. The KINO approach is designed to be modular and easy to customize, allowing developers to build conversations that are both organized and flexible. By using a clear, readable syntax for defining rules and linking them to user intents, KINO helps simplify the process of designing chatbot behavior. D'Avila emphasizes the importance of transparency and maintainability in chatbot systems, showing that rule-based methods like KINO are especially useful in settings where consistent, easy-to- understand responses are crucial—such as customer support or internal company tools.

[15] Elsabeth Solomon and Surafel Luleseged Tilahun, in their paper, offer a detailed review of how rule-based chatbots are designed, exploring key methods like pattern matching, decision trees, finite state machines, and AIML. They emphasize that these systems are easy to build, reliable, and require minimal computing power— making them ideal for structured settings like healthcare and customer service. At the same time, the authors acknowledge some drawbacks, such as poor scalability, lack of learning capabilities, and difficulty handling the variety of human language. Still, the paper argues that rule-based chatbots are a solid choice for situations that demand predictable, controlled interactions, especially where accuracy and safety are essential.

PROPOSED METHODOLOGY

The development of the movie booking platform with a rule- based chatbot was executed through a phased, modular approach combining front-end interface design, robust back-end integration, and conversational AI implementation. The methodology emphasizes how traditional web technologies and rule-based AI can work together to offer a smooth and responsive user experience, especially in handling post-error user queries efficiently. A. Frontend Development Technologies Used: HTML, CSS, JavaScript Development Strategy: The frontend of the website was built to offer an intuitive, visually engaging, and seamless user experience. Using standard web development technologies, the team focused on developing interactive pages tailored to the core functionalities of a movie booking system.

Frontend Development: Technologies Used: HTML, CSS, JavaScript Development Strategy: The frontend of the website was built to offer an intuitive, visually engaging, and seamless user experience. Using standard web development technologies, the team focused on developing interactive pages tailored to the core functionalities of a movie booking system. The design

***Corresponding Author:** Shashikala HK

was made fully responsive using CSS media queries and flexible layout techniques to ensure compatibility across desktops, tablets, and smartphones. Attention was given to accessibility and minimal loading time to optimize user satisfaction. B. Backend Development and Database Integration Technologies Used: MongoDB (Database), Node.js (Backend server).

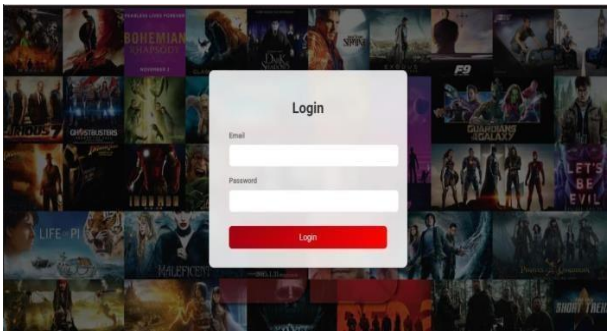

Figure 1: Login Page

Login Page: Figure 1 shows the login page of the website. Designed for secure authentication, this page includes input validation and smooth transitions for new user registration and existing user login.

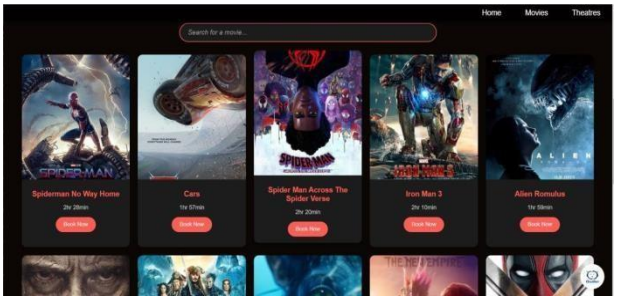

Figure 2: Home Page

Home Page: Figure 2 shows the home page of the website. Features a dynamic layout showcasing now-showing and upcoming movies, complete with filters based on genre, language, and popularity.

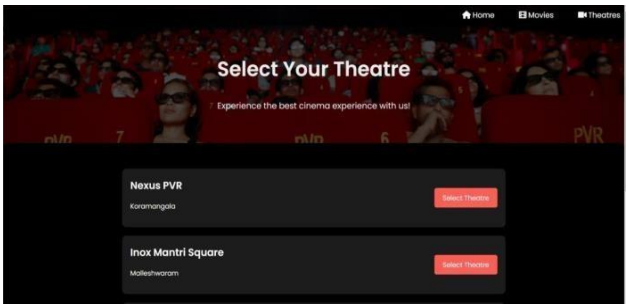

Figure 3: Theatre Page

- Theatre Page: Figure 3 shows the theatre page of the website. Allows users to browse through theatres, check available seating, and view price categories. Real-time updates are reflected using JavaScript for a dynamic experience.
- Payment Page: Implements a clean and responsive layout that supports multiple payment methods and ensures secure transactions.

The design was made fully responsive using CSS media queries and flexible layout techniques to ensure compatibility across desktops, tablets, and smartphones. Attention was given to accessibility and minimal loading time to optimize user satisfaction.

Backend Development and Database Integration
Technologies Used: MongoDB (Database), Node.js (Backend server).

Backend Logic and Data Management: The backend architecture plays a crucial role in synchronizing user actions with persistent data storage. Node.js was used as the server environment to handle requests from the frontend and process them asynchronously

- Database Connectivity: MongoDB served as the NoSQL database, efficiently storing movie information, user credentials, bookings, and theatre details
- API Development: RESTful APIs were created to allow communication between the frontend and the database. These APIs supported operations like fetching movie listings, updating seat availability, and recording transaction details.
- Validation and Security: Backend scripts included logic for validating user input, managing sessions securely, and preventing invalid or duplicate entries. All data exchanges were designed to be real-time, enabling live updates on booking status, payment confirmation, and seat selection, thus improving system reliability and user trust.

The Rule Based Approach:
Platform Used: Dialogflow (by Google) Purpose and Design Approach

To extend the website's functionality and assist users during error occurrences, a rule- based chatbot was integrated using Dialogflow. This chatbot operates on a fixed set of predefined conditions using if-else logic to handle routine queries and troubleshoot minor issues.

- Conversational Logic: Unlike machine learning-based models, this chatbot follows a rule-based flow that maps user queries to specific intent-response pairs. For example, if a user mentions "I didn't receive booking confirmation," the chatbot replies with guidance or re-verification steps based on preset logic.
- Coverage of Common Queries: The chatbot addresses concerns such as:
- Payment issues (e.g., "What do I do if my transaction failed?")
- Booking-related queries (e.g., "How can I change my selected seats?")
- General FAQs and assistance with navigating the website.

***Corresponding Author:** Shashikala HK

- Small Talk and Engagement: Beyond functional queries, small talk capabilities (like greetings or jokes) were added to make interactions more natural and friendly.
- Continuous Updates: As more user interactions were recorded, the decision tree of the chatbot was expanded to include new response paths, making the system more robust over time.

The chatbot acts as the first point of contact for the user after an error or confusion arises, guiding them based on pre-coded decision trees without requiring human intervention for every minor issue.

RESULTS AND DISCUSSION

User Interface and Frontend Experience

The movie booking system's user interface was developed with a strong focus on delivering an intuitive and visually engaging experience. Utilizing standard web technologies such as HTML, CSS, and JavaScript, the interface was structured to be fully responsive and interactive across a variety of devices and screen sizes.

1. Login Interface: The login module features a minimalist layout, prioritizing ease of use and clarity. Form fields include client-side validation to prevent incomplete or erroneous submissions. Additional functionality such as "Forgot Password" and a "Register" option ensures accessibility for both returning and new users.
2. Main Dashboard (Home Page): The landing page presents a curated selection of movies, each displayed within visually appealing, adaptive card components. These cards are enhanced with hover animations that reveal supplementary details, while a set of filtering options allows users to browse films based on genre, language, and release date, significantly improving content discoverability.
3. Theatre Selection Page: Theatres are displayed in a vertical card format, providing relevant information including name, geographic location, and real-time seat availability. An interactive "Select Theatre" button provides immediate visual feedback by altering its color and text once activated, thereby improving user interaction and choice confirmation.
4. Payment Interface: This section consolidates the booking information, presenting a detailed summary of the selected theatre, movie, and total cost. Multiple miniature images representing various payment options—such as credit cards, UPI, and wallets—are available, allowing users to choose their preferred method. These options are designed to change color and label text dynamically upon selection, adding a layer of user engagement and confirmation.
5. User Testing and Performance: Cross-platform testing validated that the frontend design maintained consistent aesthetics and performance. Page transitions were smooth, animations were responsive, and load times remained minimal. Feedback from usability testing sessions highlighted the intuitive nature of navigation, low learning curve, and overall user satisfaction.

Backend Integration and Data Management

The backend infrastructure of the movie booking platform was designed using MongoDB, a NoSQL database known for its high scalability and flexibility in handling semi-structured data. Its document-oriented nature made it particularly suitable for dynamic, data-intensive web applications.

Database Architecture: The backend database was organized into multiple collections, including Movie data, Theatres, Bookings, and Users. This modular schema enabled efficient management of various data entities such as movie metadata, theatre information, user profiles, and booking transactions.

Data Handling and Workflow: User interactions such as login authentication, account registration, and booking actions were seamlessly managed through server-side scripts interfacing with MongoDB. These interactions supported real-time updates, particularly for seat availability, and ensured that data persisted across sessions to enhance user continuity.

API Integration: A suite of RESTful APIs facilitated communication between the frontend and backend layers. These APIs were responsible for tasks such as:

- Serving movie listings dynamically to the homepage.
- Providing up-to-date seat availability for selected theatres.
- Recording booking confirmations and tracking payment details.

This architecture enabled modular and reusable components, streamlining both development and maintenance.

Security and Input Validation: To protect user data, server- side validation mechanisms were implemented alongside frontend checks, ensuring robust input sanitization.

Sensitive credentials were securely stored using cryptographic hashing algorithms. This layered approach to validation and security minimized vulnerabilities and reinforced data integrity throughout the system. By leveraging MongoDB's capabilities, the backend supported high-speed data access, seamless scalability, and real-time data synchronization. These features played a crucial role in ensuring the responsiveness and robustness of the overall application.

***Corresponding Author:** Shashikala HK

Chatbot Integration and Automated User Support

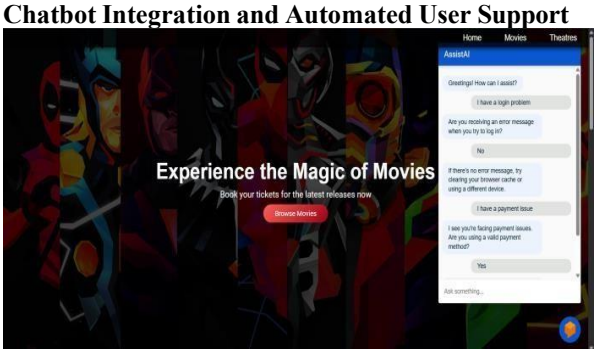

Figure 4: Home Page With Chatbot Integration

Figure 4 shows the integration of the chatbot in the website. To streamline user support and reduce manual intervention, a rule-based chatbot was integrated into the system using Google Dialogflow. This chatbot was strategically embedded across key pages of the application to provide instant assistance for common user concerns.

Intent Design and Interaction Flow The chatbot's structure was built around predefined intents that targeted frequently encountered scenarios. These included:

- Assistance with login-related queries using structured Yes/No prompts
- Guidance for movie booking and searching
- Troubleshooting for payment-related issues through nested sub-intents

Support for users unable to access the movie list

Each user query was mapped to a corresponding intent, and logical follow-up prompts were employed to maintain a coherent conversation flow. Rather than relying on machine learning models, the chatbot utilized conditional logic and branching to simulate interactive, two-way dialogues effectively.

Functionality and User Experience The chatbot was accessible throughout the application, including the login, theatre selection, and payment pages. Its key capabilities included:

- Immediate response to common issues like incorrect login details, booking errors, and payment failures
- Clear, step-by-step resolution instructions for straightforward problems
- Consistent response behavior due to its lightweight, deterministic design
- Minimal user learning curve owing to its intuitive interface.

CONCLUSION

The development of the movie booking system successfully combined a user-friendly web interface with a robust backend infrastructure, delivering a smooth and efficient experience for end users. The application effectively managed real-time data interactions, including user authentication, movie information, theatre availability, and booking transactions, through seamless communication with a MongoDB database.

A notable advancement in this system was the integration of a rule-driven chatbot designed using Google Dialogflow. This virtual assistant served as an intelligent support agent, enabling users to receive prompt help with routine tasks such as signing in, exploring movie options, and resolving payment-related queries. By implementing structured intents and logical follow-up flows, the chatbot provided timely and relevant responses, thereby improving the overall usability of the application.

The incorporation of the chatbot greatly enriched the platform's interactivity and accessibility. It ensured consistent support and added a conversational dimension to the user interface, demonstrating how automated dialogue systems can enhance digital platforms. This successful integration highlights the growing importance of intelligent assistants in improving user engagement and streamlining customer service processes in web applications.

REFERENCES

1. Dogan, O., & Gurcan, O. F. (2024). Enhancing e-business communication with a hybrid rule-based and extractive-based chatbot. *Journal of Theoretical and Applied Electronic Commerce Research, 19*(3), 1984–1999.
2. Maher, S., Kayte, S., & Nimbhore, S. (2020). Chatbots & its techniques using AI: A review. *International Journal for Research in Applied Science and Engineering Technology, 8*(12), 503–508.
3. Alias, S., Sainin, M. S., Fun, T. S., & Daut, N. (2019). Identification of conversational intent pattern using pattern-growth technique for academic chatbot. In *International Conference on Multi-disciplinary Trends in Artificial Intelligence* (pp. 263–270). Springer.
4. Nitschke, R., Wang, Y., Chen, C., Pyarelal, A., & Sharp, R. (2022). Rule-based event extraction for artificial social intelligence. In *Proceedings of the First Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning* (pp. 71–84).
5. D'Ávila, T. C. (2018). *KINO: An approach for rule-based chatbot development, monitoring and evaluation*.
6. Silkhi, H., Bakkas, B., & Housni, K. (2024). Comparative analysis of rule-based chatbot development tools for education orientation: A RAD approach. In *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security* (pp. 1–7).
7. Thorat, S. A., & Jadhav, V. (2020). A review on implementation issues of rule-based chatbot systems. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*.
8. Hamooni, H., Debnath, B., Xu, J., Zhang, H., Jiang,

***Corresponding Author:** Shashikala HK

G., & Mueen, A. (2016). LogMine: Fast pattern recognition for log analytics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 1573–1582). ACM. https://doi.org/10.1145/2983323.2983358

9. Friedman, D., Panigrahi, A., & Chen, D. (2024). *Representing rule-based chatbots with transformers* (arXiv Preprint arXiv:2407.10949).
10. Tamil Iniyan, K. S., Rakesh, M., Khan, U. A., Shashikala, H. K., Khan, A., & Das, R. (2024). Conversational AI chatbot using log pattern detection. In *2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)* (pp. 1–6). IEEE.
11. Riandika, M. A. N., Susanto, A., & Respatria, N. M. (2024). Chatbot features on websites using Dialogflow framework with rule-based method. *Jurnal Teknik Informatika (JUTIF), 5*(5), 1393–1403.
12. Pham, H. V., & Cu, L. K. (2020). Intelligent rule-based support model using log files in big data for optimized service call center schedule. In *Intelligent Computing in Engineering: Select Proceedings of RICE 2019* (pp. 931–942). Springer.
13. Qi, J., Huang, S., Luan, Z., Yang, S., Fung, C., Yang, H., Qian, D., Shang, J., Xiao, Z., & Wu, Z. (2023). LogGPT: Exploring ChatGPT for log-based anomaly detection. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)* (pp. 273–280). IEEE.
14. Solomon, E., & Tilahun, S. L. (2024). Rule-based chatbot design methods: A review. *Journal of Computational Science and Data Analytics, 1*(1), 75–84.
15. Saini, A., Verma, A., Arora, A., & Gupta, C. (2017). Linguistic rule-based ontology-driven chatbot system. In *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2017* (Vol. 2, pp. 47–57). Springer.

***Corresponding Author:** Shashikala HK