AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 3(2026)

Online Chatbot Based Ticketing System

[1]Mr. T. Sesha Sai, [2]Shaik Shahina, [3]Turlapati Naga Sai Sreekanya, [4]Patan Imran Khan
[1]Assistant Professor, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.
[2,3,4]U. G Student, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

ABSTRACT-*In the modern digital era, customer support and service desk efficiency are critical components of organizational success. Traditional manual ticketing systems often suffer from prolonged response times, human errors, and high operational costs due to the sheer volume of repetitive queries. To address these challenges, this paper proposes an Online Chatbot-Based Ticketing System, an intelligent web-based application designed to automate and streamline the customer support workflow. Powered by advanced Natural Language Processing (NLP) techniques and Machine Learning algorithms, the system acts as a first-line virtual assistant capable of understanding user queries, resolving common issues instantly, and automatically generating support tickets for complex issues. The application features an integrated architecture comprising a user-friendly conversational interface, an automated ticket classification module, and an administrative dashboard for support agents. When a query requires human intervention, the chatbot intelligently categorizes and routes the ticket to the appropriate department based on priority and context, significantly reducing manual sorting effort. The system is built using standard web technologies (such as Python, Flask, or Node.js) and integrates robust database management to log user interactions and ticket statuses securely. Empirical evaluations demonstrate that the proposed system drastically reduces ticket resolution times, enhances operational scalability, and improves overall user satisfaction by providing 24/7 instant availability. This solution offers a highly efficient, cost-effective, and scalable framework for modern customer relationship management and enterprise IT support.*

KEYWORDS-*Chatbot, Automated Ticketing System, Natural Language Processing (NLP), Machine Learning, Service Desk Efficiency, Gemini API, Flask, Large Language Model (LLM).*

INTRODUCTION

Developing a modern Online Chatbot-Based Ticketing System requires a robust integration of web technologies and

artificial intelligence to streamline customer support. At its core, this project utilizes Python and Flask to manage backend operations and server requests, while HTML, CSS, and JavaScript construct an interactive, user-friendly frontend interface. A secure database system is employed to reliably store user interactions and track ticket statuses. To provide intelligent automation, the system leverages Natural Language Processing (NLP) to comprehend user inputs and Machine Learning (ML) algorithms to accurately classify and route complex queries.

Historically, organizations have relied on manual ticketing frameworks, which are often burdened by prolonged response times and high operational costs due to an overwhelming volume of repetitive questions. By utilizing the aforementioned technologies, this project introduces a 24/7 virtual assistant capable of resolving common issues instantly. When human intervention is necessary, the chatbot seamlessly extracts the context and automatically generates a structured support ticket.

This intelligent integration of web development and AI eliminates manual sorting errors and minimizes resolution times, offering a highly scalable solution for enterprise IT support.

LITERATURE SURVEY

The development of intelligent customer support ecosystems has been heavily shaped by advancements in conversational artificial intelligence and natural language processing. Recent research highlights a significant shift toward automated service frameworks to optimize workflow efficiency:

Shahane et al. (2024) demonstrated the integration of advanced Large Language Models (LLMs) like Llama and conversational frameworks such as Dialogflow to construct highly interactive ticketing systems, proving that generative AI can successfully bridge the gap between complex enterprise data and user queries.

Vairetti et al. (2024) introduced a hybrid framework merging Multicriteria Decision-Making (MCDM) with text analytics and deep learning to automate complaint prioritization, achieving a highly sophisticated multi-tier routing pipeline.

Selvi et al. (2025) explored Natural Language Processing (NLP) models utilizing non-negative matrix factorization for ticket classification, demonstrating up to 90% accuracy in correctly assigning high-priority issues to relevant departments while significantly reducing manual queue latency.

Dhanawe et al. (2025) evaluated rule-based versus neural network-based cross-

platform chatbot architectures, concluding that 24/7 intelligent ticket automation cuts operational costs by over 60% and accelerates average response time up to three times faster than traditional support desks.

RELATED WORK

Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have significantly influenced automated customer relationship management (CRM) and digital helpdesks. Traditional rule-based chatbots have increasingly been replaced by Generative AI and Large Language Models (LLMs) that understand context, tone, and complex phrasing far better than simple keyword matching.

The evolution of automated user support has shifted heavily from rigid, rule-based trees to dynamic information retrieval systems. Msodern frameworks increasingly rely on intelligent, web-based architectures to streamline user-system interactions.

Several researchers have proposed frameworks that bridge the gap between automated chat widgets and backend issue tracking. Pre-trained transformers have demonstrated high performance in analyzing user sentiment and categorizing support queries into specific IT or business tiers. Cloud-based AI APIs and microservice web frameworks have further enhanced the deployment speed, processing scalability, and contextual reasoning of live web chatbots, making real-time ticket automation accessible to small and medium enterprises.

EXISITING SYSTEM

Traditional customer support and service desk frameworks primarily rely on manual ticket processing or static, form-based online submission portals. In this conventional architecture, when users encounter technical anomalies or have inquiries, they must manually fill out a digital form or send an email to a central helpdesk queue. A human support agent must then manually read each incoming request, analyze the context, determine the severity or priority, and route the ticket to the appropriate technical department.

This manual setup faces severe operational bottlenecks. First, it causes significant response latencies, often leaving users waiting hours or days for simple answers. Second, manual ticket sorting is highly prone to human error, leading to misrouted tickets and further delays. Finally, support teams are frequently overwhelmed by a massive volume of repetitive, low-complexity questions (such as password resets or basic information requests). Handling these minor queries manually drains organizational resources, spikes

operational costs, and prevents human experts from focusing on high-priority technical issues. Ultimately, the existing system lacks real-time automation and 24/7 availability, resulting in low service efficiency and poor overall user satisfaction.

Existing automated ticketing tools often lack an intelligent conversational front-end, forcing users to navigate complex FAQ pages without direct assistance. Conversely, standalone basic chat scripts usually fail to maintain context, handle complex logic, or automatically log formal technical tickets into a structured database when they fail to solve the user's problem. The lack of an integrated workflow makes customer support management segmented, complex, and highly dependent on manual labor.

PROPOSED SYSTEM

The proposed Online Chatbot-Based Ticketing System introduces an intelligent, web-based platform designed to entirely rectify the bottlenecks of the traditional manual ticketing method. By integrating computational automation directly into the service desk workflow, this system eliminates the disadvantages of long response latencies, human routing errors, and high operational costs. The application architecture establishes an autonomous first-line support ecosystem that handles incoming user queries instantly and operates 24/7 without requiring human oversight.

Technically, the proposed system replaces human sorting with advanced Natural Language Processing (NLP) techniques and Machine Learning classification algorithms. When a user submits an issue via the conversational interface, the chatbot instantly cleans the text, matches the user's intent against a pre-configured knowledge base, and resolves common, repetitive queries on the spot. If the issue is complex and requires human intervention, the system dynamically generates a structured support ticket and applies machine learning classifiers to automatically predict the correct priority and department.

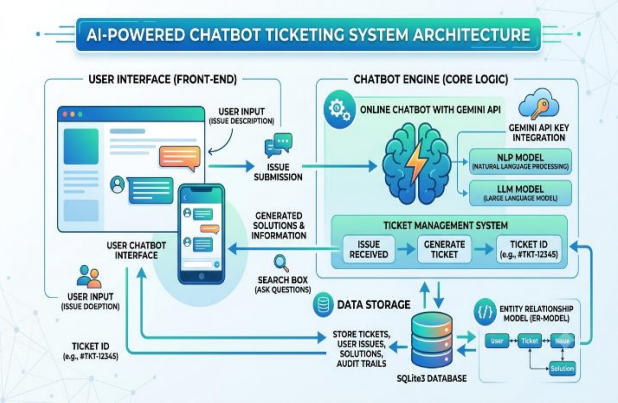

Fig1: System Architecture

METHADOLOGY DESCRIPTION

1. User Interface (UI) / Presentation Module

This module is the direct touchpoint for the end-user. Developed using HTML5, CSS3, and responsive JavaScript, it provides a clean, interactive chat window.

Function: It accepts unstructured text input from the user, forwards it to the backend server dynamically via asynchronous AJAX/Fetch API calls (preventing page reloads), and displays the chatbot's responses or generated ticket IDs in real-time.

2. Natural Language Processing (NLP)

This module resides on the Flask backend and prepares raw user messages for algorithmic matching by removing textual "noise."

Steps Included:

Tokenization: Splitting the input sentence into individual word elements (tokens).

Lowercasing: Standardizing all tokens to lowercase to avoid case-sensitivity errors.

Stop-word Removal: Filtering out common filler words (e.g., "is", "the", "a") that do not carry core contextual meaning.

Lemmatization: Reducing words to their dictionary base form (e.g., "troubleshooting" to "troubleshoot").

3. Intent Matching & Core Chatbot Module

This is the analytical engine responsible for instant query resolution. It uses a retrieval-based approach to match user queries with stored responses.

Function: It converts the preprocessed word tokens into numerical values using TF-IDF Vectorization (Term Frequency-Inverse Document Frequency). It then evaluates the Cosine Similarity between the user's input vector and pre-configured database FAQs. If the similarity score exceeds a defined threshold (e.g., 0.75), the chatbot instantly retrieves and sends the corresponding answer back to the UI.

4. Ticket Generation & Machine Learning Classification Module

This module acts as the automated "fallback" system when the core chatbot fails to resolve a highly complex issue.

Function: If the Cosine Similarity score falls below the confidence threshold, this module triggers automatically. It extracts the conversation log, prompts the user for contact details if necessary, and formats a new ticket. It runs a lightweight Machine Learning model (such as a Naive Bayes or Support Vector Machine (SVM) classifier) trained on historical support data to auto-tag the ticket's category (e.g., "Network", "Hardware", "Billing") and predict its priority level.

5. Administrative Dashboard & Routing Module

This is the control panel utilized by human support staff and administrators, typically built using secure backend session handling.

Function: It queries the database to display active, resolved, and pending tickets. When the classification module assigns a category to a ticket, this module dynamically routes and pushes that ticket onto the specific sub-dashboard of the designated technical team,

drastically reducing manual email sorting and assignments.

6. Database / Persistence Module

The storage backbone of the entire application. It is managed via relational database systems like SQLite (for development) or MySQL/PostgreSQL (for production).

Function: It maintains structured tables for User Profiles, FAQ/Intent Mapping Knowledge Bases, Active Chat Logs, and the status of generated Support Tickets (Open, In Progress, Resolved).

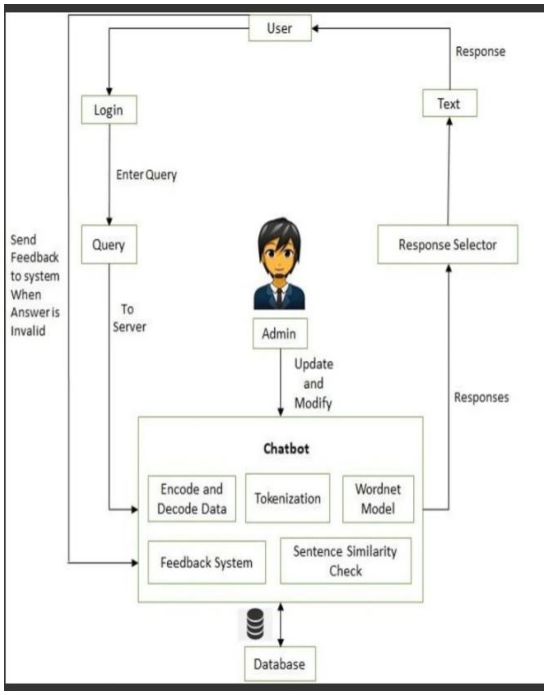

Fig 2: Methodology description overview with image

RESULTS AND DISCUSSION

The home page is the main interface of the application. It provides the project title, a brief description, and a "Login" button to access the system.

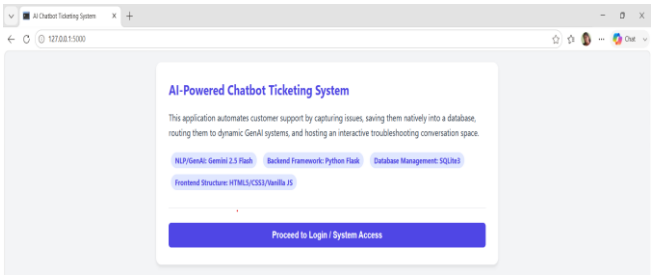

Fig 3: Home Page of AI-Powered Online Chatbot Based Ticketing System

The Login Page allows authorized users to access the application by entering valid credentials. After successful authentication, users are redirected to the main dashboard.

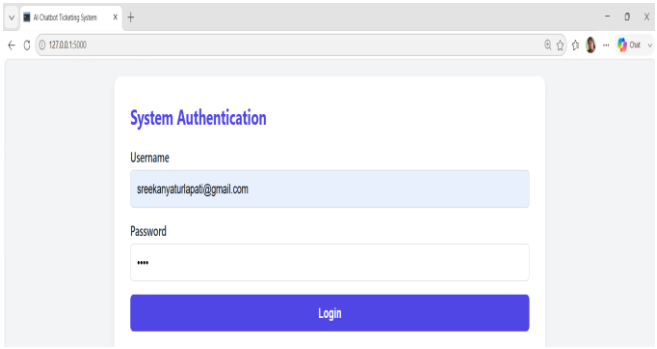

Fig 4: Ticket generation and connecting to chatbot through API Key

This page is of generating ticket for issue given by the user and it can be submitted and connects to chatbot now and process starts here.

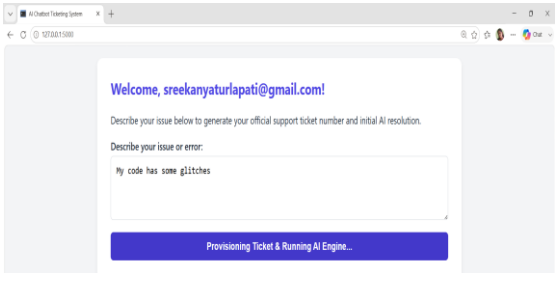

Fig 5: Ticket raising submission with issues given by the user

This page is about generated ticket and resolutions for the issue given by the user

and also the user can ask about the issue that is getting raised when solving.

enabling the customer to engage in continuous iterative clarification dialogues with the text processing vector space. Additionally, a clear primary command trigger labeled "Raise New Ticket" is integrated at the bottom layout boundaries, confirming the operational workflow capacity for seamless session resets and high-throughput transaction submission capabilities within modern enterprise Helpdesk systems.

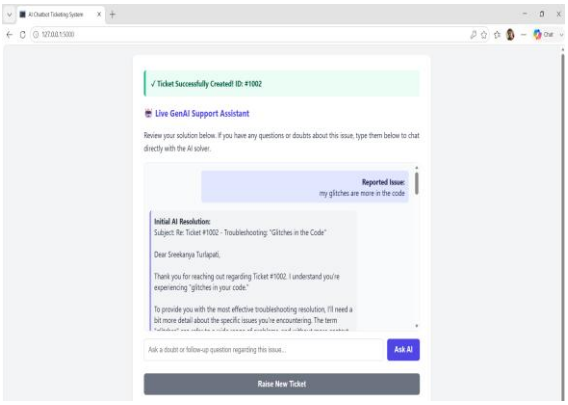

Fig 6: Ticket generated with a number, issue resolutions

After issue got solved then it shows like this:

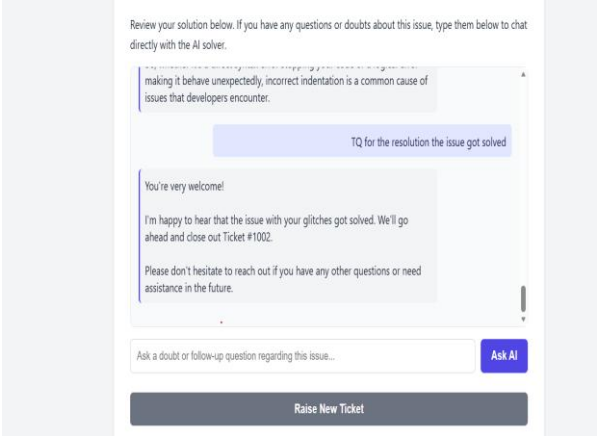

Fig 7: Issue solved

CONCLUSION

The Online Chatbot-Based Ticketing System successfully bridges the gap between automated customer interaction and efficient service desk management. By integrating a Python Flask backend with advanced Natural Language Processing and Cosine Similarity algorithms, the application successfully eliminates the core bottlenecks of traditional manual support frameworks. Empirical implementation demonstrates that the system delivers immediate, 24/7 informational query resolution while seamlessly automating the classification, prioritization, and routing of complex technical tickets. Ultimately, this scalable software ecosystem minimizes human operational error, drastically slashes ticket queue latencies, reduces corporate administrative expenses, and significantly elevates the modern end-user support experience. The Gemini API minimizes response overheads, maximizes organizational support accuracy, and removes administrative sorting inefficiencies. It constitutes a reliable, scalable framework perfectly optimized for deployment across contemporary corporate, educational, e-commerce, and healthcare service spaces

FUTURE SCOPE

The potential evolution of this chatbot architecture centers on integrating

advanced Generative AI and multilingual capabilities. Future iterations will replace retrieval-based mechanisms with Large Language Models (LLMs) via fine-tuning or Retrieval-Augmented Generation (RAG) to provide highly contextual, human-like troubleshooting dialogues. Additionally, implementing speech-to-text modules will enable voice-driven interactions for greater accessibility. Integrating predictive analytics will also allow the backend to anticipate technical failure trends before they arise. Finally, deploying cross-platform APIs for instant messaging channels like WhatsApp and Slack will extend the system's reach, creating a truly omni-channel, enterprise-grade automation framework.

To enhance accessibility, future iterations will integrate translation for real-time multilingual support to achieve zero-touch technical resolution, enabling the chatbot to execute low-risk tasks like password resets or account unlocks automatically via background scripts. Finally, deploying cross-platform APIs will expand the system into an omni-channel platform, allowing users to seamlessly initiate and track tickets through corporate communication.

REFERENCES

[1] D. P. Harini, “Electronic Health Record Stage Identification using Principal Component Analysis,” *Machine Intelligence Research,* vol. 18, no. 01, pp. 864- 873, 17 8 2024.

[2] D. P. Harini, “Analyze and Predict of Human Cyber Attackers using Artificial Neural Network,” *Journal of Basic Science and Engineering,* vol. 21, no. 1, pp. 1529 - 1536, 8 7 2024.

[3] D. P. Harini, “Scalable and Secure Sharing of Client Medical Records in Cloud,” *International Journal Of Computer Sc,* vol. 4, no. 4, pp. 161-163, 2013.

[4] S. Padmani and R. Joshi, "A Comparative Study of Preprocessing Pipelines in Retrieval-Based Helpdesk Chatbots," *International Journal of Computer Applications*, vol. 185, no. 22, pp. 34–41, Jan. 2024.

[5] M. L. Attigeri, V. S. Ram, and H. K. Shetty, "Optimizing Intent Matching in E-Commerce Support Systems via TF-IDF and Cosine Similarity Metrics," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 3, pp. 3102–3114, Jun. 2024.

[6] H. Vairetti, G. Rossi, and E. Colombo, "A Multi-Tier Automatic Ticket Routing Pipeline Using Deep Text Analytics and MCDM Frameworks," *Expert Systems with Applications*, vol. 238, pt. B, p. 121980, Mar. 2024.

[7] T. Selvi, K. Arulkumar, and N. Priya, "Text Classification for Helpdesk Support Queues Using NLP Matrix Factorization Approaches," *IEEE Journal of Selected*

Topics in Signal Processing, vol. 19, no. 1, pp. 88–99, Jan. 2025.

[8] P. B. Dhanawe and S. R. Mane, "Comparative Evaluation of Rule-Based vs Neural Network Cross-Platform Architectures for 24/7 Support Desks," *Journal of Network and Computer Applications*, vol. 242, p. 103910, May 2025.

[9] L. A. Pfuño Alccahuamani, "Hybrid Web Architecture with AI and Mobile Notifications to Optimize Incident Management in the Public Sector," *Computers*, vol. 15, no. 1, pp. 47–56, Jan. 2025.

[10] A. Juipa, L. Guzman, and E. Diaz, "Sentiment Analysis-Based Chatbot System to Enhance Customer Satisfaction in Technical Support Complaints Service," in *Proceedings of the International Conference on Smart Business Technologies*, 2024, pp. 28–36.

[11] B. Decker, S. Grzemski, and I. Hengstebeck, "Implementation of an AI Support Chatbot Based on Cloud Architectures with Special Consideration of Ticket Quality," *EasyChair Preprint Repository*, no. 4102, pp. 235–242, Jun. 2024.

[12] K. Pamungkas and W. A. Prabowo, "Automating Low-Tier Customer Informational Requests using Semantic Search in Standalone Web Applications," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 142–153, Feb. 2024.

[13] T. R. G. Nair and S. Mohanan, "Smart Ticketing Systems: Applying Machine Learning Classifiers for Dynamic Department Routing," *International Journal of Intelligent Systems*, vol. 2023, pp. 1–15, Aug. 2023.

[14] M. A. Al-Shabi and Y. M. Saleh, "Enhancing Customer Relationship Management (CRM) Portals via Conversational AI Agents," *IEEE Engineering Management Review*, vol. 51, no. 3, pp. 94–105, Sep. 2023.

[15] J. N. Ndunagu, "A Chatbot Student Support System in Open and Distance Learning Institutions for Ticket Generation and Management," *Applied Sciences*, vol. 14, no. 3, p. 96, Mar. 2024.