FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 10, October, 2025, pp 368 – 375
DOI: https://doi.org/10.33003/fjs-2025-0910-3969

SCALABLE AND REAL TIME EMBEDDABLE RETRIEVAL-AUGMENTED GENERATION (RAG) ANALYTICS SYSTEM FOR CUSTOMER SERVICE

[1]Adigun Taiwo, *[2]Eweoya Ibukun, [3]Sodiq Kazeem, [4]Ajayi Oluwabukola F., [4]Ayankoya Folasade, [4]Akande Oyebola and [5]Adetunji Olusogo

[1] Department of Software Engineering, University of Lay Adventist of Kigali, Kigali, Rwanda.
[2]Department of Software Engineering, Babcock University, Ogun State, Nigeria.
[3] Department of Computer Engineering, Yaba College of Technology, Lagos State, Nigeria.
[4]Department of Computer Science, Babcock University, Ogun State, Nigeria.
[5] Department of Computer Engineering, Olabisi Onabanjo University, Ogun State, Nigeria.

Correspondent Author's E-mail: eweoyai@babcock.edu.ng

ABSTRACT

Customer service has transitioned from traditional face-to-face and phone-based interactions to digital platforms that emphasize speed, scalability, and personalization. Despite these advances, AI-driven tools like chatbots often face challenges in contextual understanding, handling multi-step queries, and enabling smooth escalation, which can lead to dissatisfaction. This research develops a scalable, real-time embeddable Retrieval-Augmented Generation (RAG) analytics system that integrates AI efficiency with human adaptability. The system architecture employs FastAPI, Celery, and Centrifugo for backend processing, ReactJS with Vite for the frontend, and PostgreSQL for secure data handling. It incorporates OpenAI's GPT-3.5-turbo API for natural language processing and NovuHQ for real-time notifications, ensuring context-aware responses and timely human intervention. An iterative development model guided the design, enabling incremental refinements through continuous feedback from customers, agents, and administrators. Key features include iframe embedding, direct web links, reusable components, real-time chat, Google OAuth authentication, session tracking, analytics, and escalation pathways. Testing confirmed that the system effectively handles routine queries while seamlessly escalating complex cases to human agents. Evaluation results highlight improved scalability, reduced response time, and preserved personalization. Its embeddable design supports adoption across diverse sectors, including SMEs and educational institutions. Future extensions will explore multilingual capabilities, sentiment-driven escalation, and CRM integration for holistic customer relationship management.

Keywords: Adaptive customer service, RAG, Chatbot, Real-time systems, Human escalation, AI-driven analytics

INTRODUCTION

Customer support has historically been regarded as a demanding and stressful profession, often serving as the frontline of customer satisfaction. Traditionally, service delivery relied heavily on face-to-face interactions and telephone calls, which provided a human touch but also imposed significant limitations in terms of scalability and efficiency (Dixon, et al., 2010). With the rapid advancement of technology, customer service has undergone major transformations, particularly through the rise of digital platforms such as chatbots, virtual assistants, and social media support channels (Maheshwaram, 2024). These tools allow businesses to connect with customers in more efficient and scalable ways than previously possible Kumar et al.,2024)

Despite these advancements, customer service requirements vary significantly across industries, with each demanding different levels of quality, type, and extent of support. In today's highly competitive environment, customer SERVICE is no longer a supplementary business function but a core determinant of customer satisfaction and loyalty (Rane, 2023). However, traditional systems often fall short, struggling with scalability, consistency, and timely responses. This frequently results in poor customer experiences and higher operational costs for businesses (Tratta, n.d).

Artificial Intelligence (AI) has emerged as a potential solution, offering capabilities such as automated responses and real-time interaction (Coursera,2025).AI-driven systems, particularly chatbots, excel at handling routine tasks quickly and at scale. Yet, they also face notable limitations, such as insufficient contextual understanding, inability to manage multi-step or complex queries, and risks of generating inaccurate or inappropriate responses. This can lead to customer frustration and ultimately damage business relationships (Adamopoulou & Moussiades, 2020).

To address these challenges, researchers and practitioners increasingly advocate for hybrid service models that integrate AI with human escalation and analytics (Mayer et al.,2024). In such systems, AI handles repetitive, structured queries, while complex or sensitive issues are escalated to human agents who can apply critical thinking and empathy (Arcega-Punzalan, 2025).This synergy ensures that businesses can balance efficiency, accuracy, and personalization, offering both scalability and high-quality customer engagement (Rafalski,2025).

At the same time, evolving customer expectations have heightened the pressure on businesses to provide fast, reliable, and personalized services. Long wait times on phone calls, generic chatbot responses, and inadequate handling of complex issues remain common problems that erode trust and satisfaction. Customers today expect seamless service across multiple digital channels, but chatbots frequently fail to account for unique contexts, creating friction instead of resolution.

This study responds to these issues by proposing an AI-enhanced customer service framework that combines automated efficiency with human adaptability. The main contributions include:

i. The development of an AI-augmented chatbot with human escalation to improve efficiency and customer satisfaction.

ii. The system's performance is effective in meeting diverse industry needs.

MATERIALS AND METHODS

System Requirements and Development Tools

The system involves the following entities customers, customer service agents (human interceptors), and the system administrator. The research utilized the following tools; FastAPI, Celery, Centrifuge as backend, ReactJS, Vite as frontend, PostgreSQL as database. Also, OpenAI (version: gpt3.5-turbo) API for natural language processing and NovuHQ for push alerts and notifications were used as API.A Dedicated server with minimum of 4GB RAM to handle the expected user load and database operations and end users and agents should be able to access the system from any internet-connected device.

Iterative Process Model

This research adopted the iterative process model (figure 1), because it allows for incremental development, making it easier to add and refine complex features like human interception. Continuous feedback from end users, customer service agents, and administrators can be incorporated into each iteration, improving the chatbot's effectiveness and user experience. This approach also reduces risks by enabling thorough testing of individual components, like natural language understanding and escalation capabilities, before advancing to the next stage. Furthermore, the adaptability of the iterative model allows for adjustments to requirements as they evolve, ensuring that the final product aligns closely with user needs and performs well in real-world scenarios.

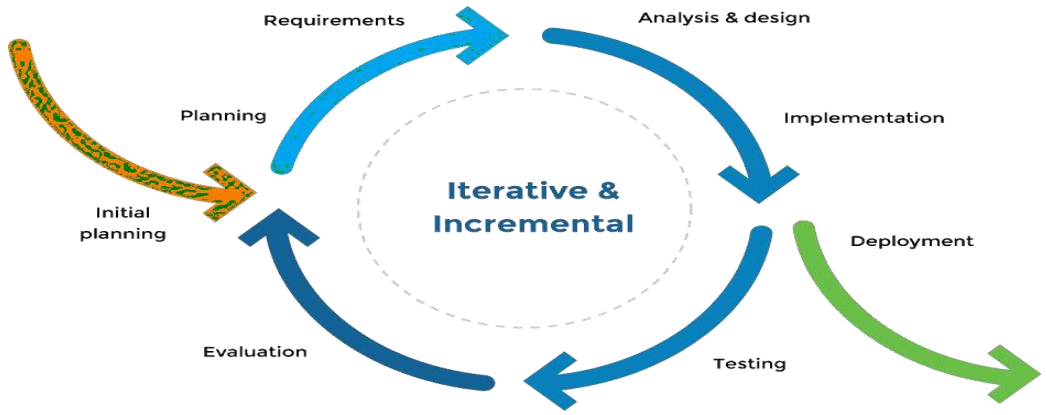

Figure 1: Iterative Development Software Process Model

System Architecture

The system architecture enables scalable, secure, and efficient customer interactions with human interception for complex queries. The intuitive frontend allows users to chat, view histories, and access customer service features. It connects to a FastAPI backend that processes queries, manages sessions, and escalates unresolved issues to human agents. Celery manages background tasks like chatbot processing and chat logging, while a PostgreSQL database securely stores user data and chat histories. While Python supports multithreading, its Global Interpreter Lock (GIL) can limit true parallel execution of CPU-bound tasks. This constraint makes handling a large number of concurrent real-time connections challenging using Python alone. Centrifugo was introduced to handle real time service alone allowing python to focus on business logic and resource allocation. User authentication is handled via hashed passwords (using Argon2) and Google OAuth. Integrating OpenAI's language model, the chatbot provides context-aware responses and flags complex queries for human escalation, ensuring efficient operations and customer support.

System Design

This section outlines the design of the chatbot system, focusing on how various components will interact to provide an efficient and user-friendly experience (Figure 2). The system is designed to facilitate customer interactions through a web-based chatbot that leverages OpenAI's API for context-aware responses and allows for human agent intervention when necessary.

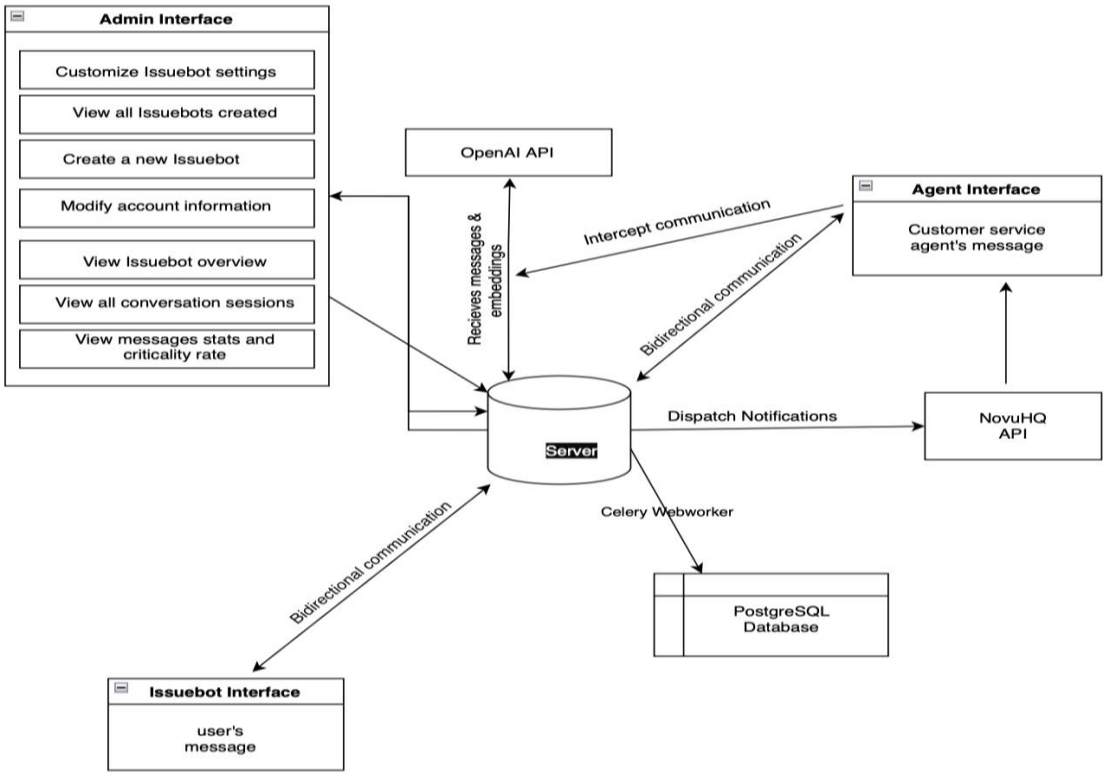

Figure 2: Application's Component Interaction

The customer use cases are *start chat:* this allows the customer to initiate a chat session; *send message:* this permits the customer to send messages within the chat; *view current history:* this allows the customer to view their chat history; *request human agent:* this lets the customer request to escalate a chat to a human agent.

Customer service agent use cases are *login:* this allows the agent to log in to the application; *signup:* this enables the agent to create a new account; *respond as human agent:* this allows the agent to respond to customers who requested human assistance; *receive notifications:* the agent is notified when they are needed to assist a customer.

The system administrator use cases are *login:* this allows the system administrator to log in to the application; *signup:* this enables the system administrator to create a new account; *view analytics:* this allows the administrator to view usage and performance analytics; *manage all functions*: the admin can access other functionalities related to user management (not explicitly shown but implied by the connections to various use cases).

System Information Flow

Information exchange occurs between the entities of the system, and the entities within the system are: *Admin:* Configures and monitors the system by customizing settings, managing IssueBots, updating account details, and viewing statistics; *IssueBot User:* Sends messages to the system and receives automated responses from the IssueBot; *System:* Central controller that handles messages, stores data, interacts with external APIs (OpenAI & NovuHQ), and facilitates communication between users, admins, and agents; *PostgreSQL Database:* Stores all messages and relevant data for persistence and reporting; *OpenAI API:* Processes user messages by generating embeddings, allowing the system to better understand and respond to messages; *NovuHQ API:* Sends real-time notifications to admins or agents when important events (like escalations) occur; *Agent:* Intercepts user messages if needed and responds directly via the Agent Interface when human intervention is required. The figure 3 shows the flow of information within the system.

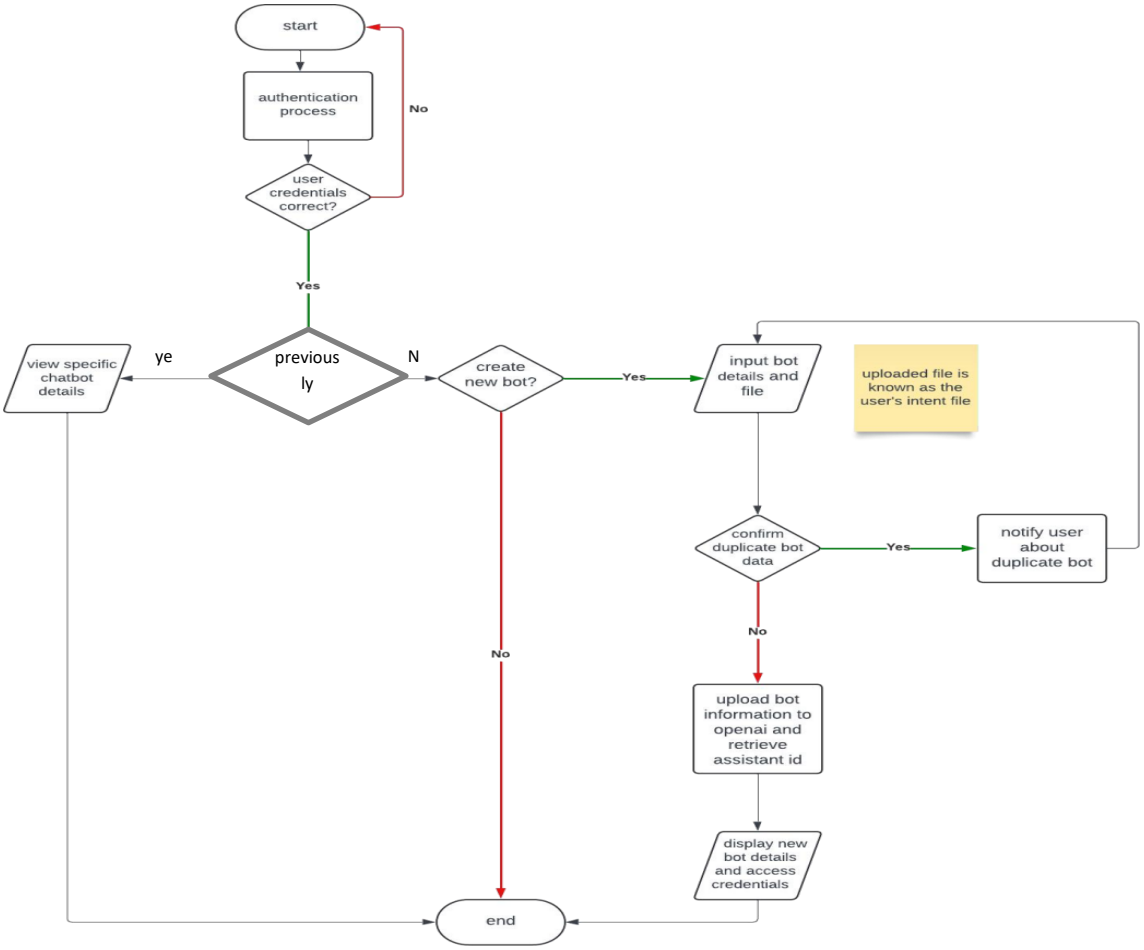

Figure 3: Flow Diagram for the System

RESULTS AND DISCUSSION

The following are the outputs of the implementation of the system:

Landing Page

This page i.e. Figure 4 contains an overview of what to expect from the system, it provides an overview of the services the system provides, and it also provides call-to-action buttons for login and registration.

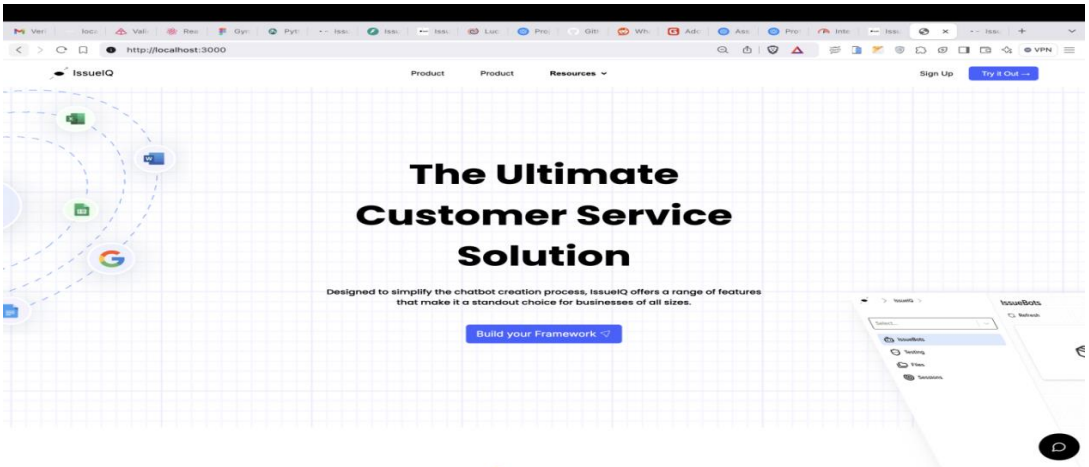

Figure 4: Showing the Landing Page

Authentication Pages

The figure 5 contains set of pages which provide companies an opportunity to register or login an account in the system, asking for very little information and then securely logging them into the system upon completion. It also gives them the ability to authenticate using google authentication since it is the most method of onboarding the world today.

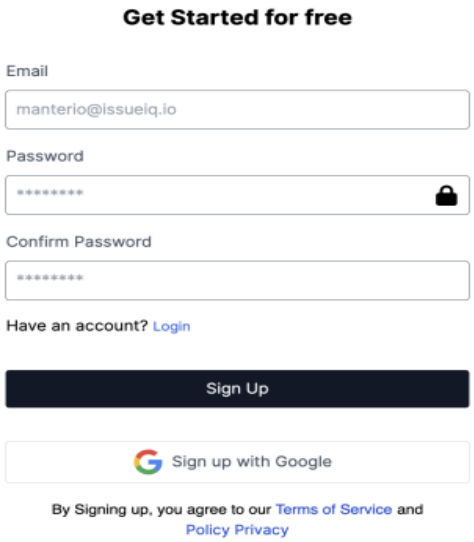

Figure 5: Showing the Signup Page

Issuebots Page

The figure 6 provides information about the list of issuebots that has been created on that particular account. It captures information like the issuebot name and date of creation.

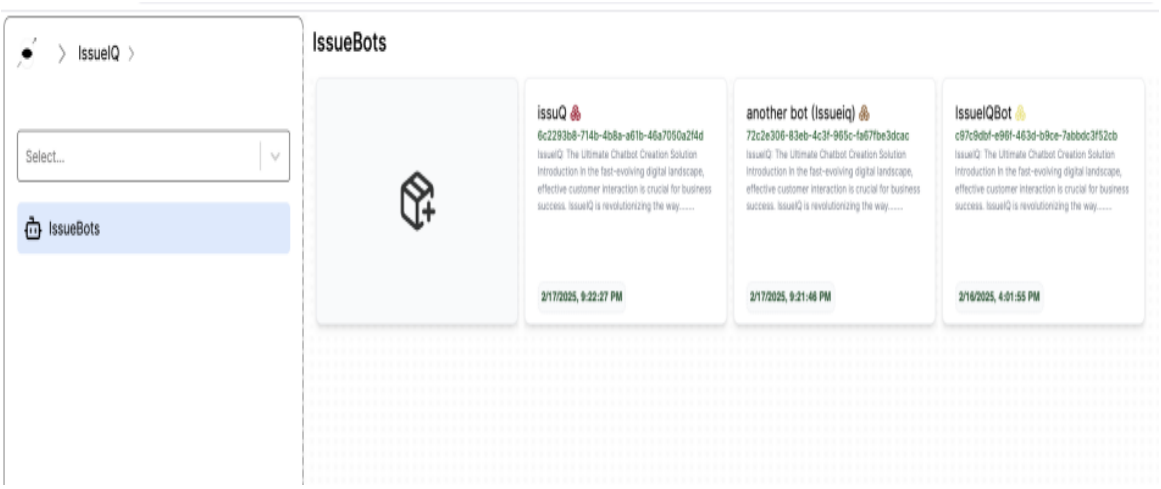

Figure 6: Showing the Issuebots Page

Integrations Page

The figure 7 presents integration page which contain detailed information about how the chatbot of every issuebot can be integrated into every platform to suit user needs. The issuebot chatbot can be integrated via 3 main methods.

i. ***Embed using an Iframe*:** This method allows users to embed the chatbot within a webpage using an <iframe> tag, seamlessly integrating it into their website.

ii. ***Sharing the generic web url*:** Users can share a direct link to the chatbot's hosted web interface, making it accessible without embedding.

iii. ***Embedding using the web component script*:** This method uses a custom JavaScript snippet to embed the chatbot as a reusable web component, offering more flexibility in integration. It supports 'HTML' and 'JSX/TSX'.

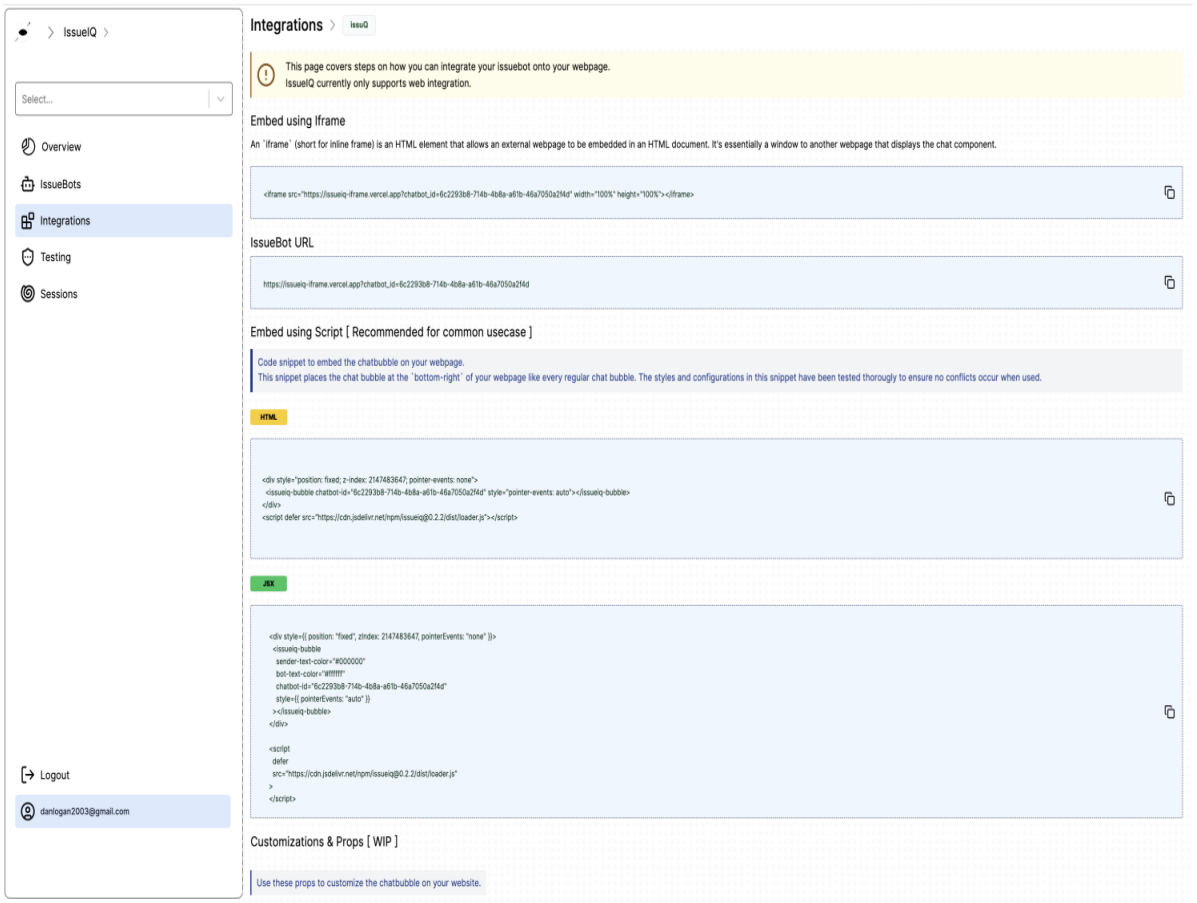

Figure 7: Showing the Integrations Page

Testing Page

The Testing Page (Figure 8) allows users to interact with their chatbot in a controlled environment before deployment. Users can simulate conversations, verify responses, and fine-tune chatbot behavior to ensure accuracy and reliability. This feature helps improve chatbot performance before real-world use.

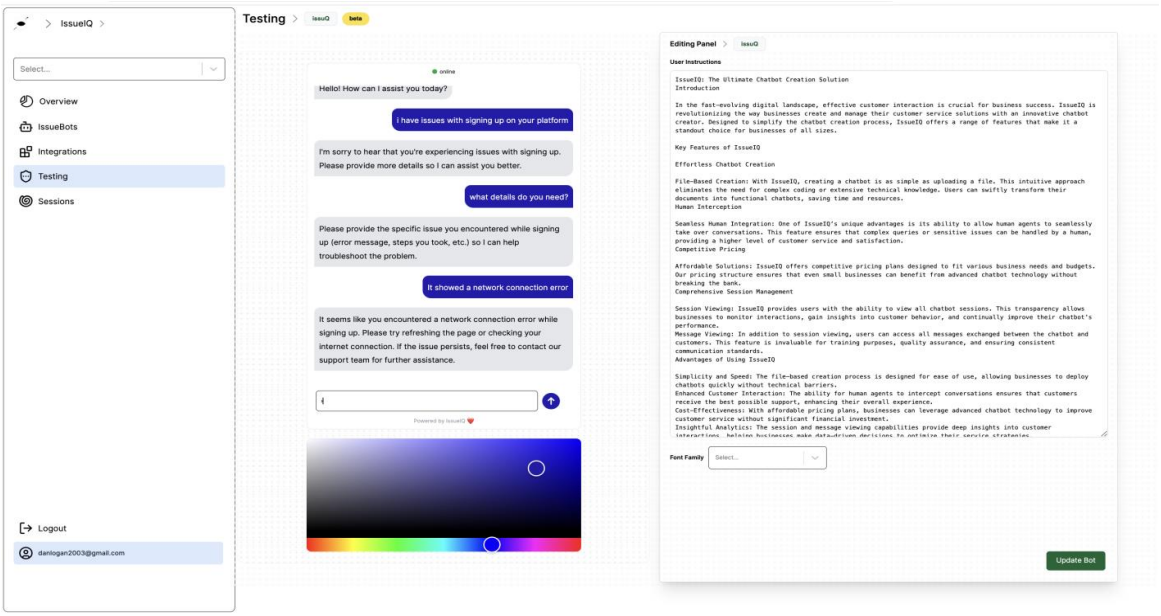

Figure 8: Showing the Testing Page

Sessions Page

The Sessions Page (Figure 9) displays a list of all active and past chatbot interactions. Users can monitor live conversations, review chat history, and analyze session details. This page helps users track customer interactions and improve chatbot responses based on real usage data.

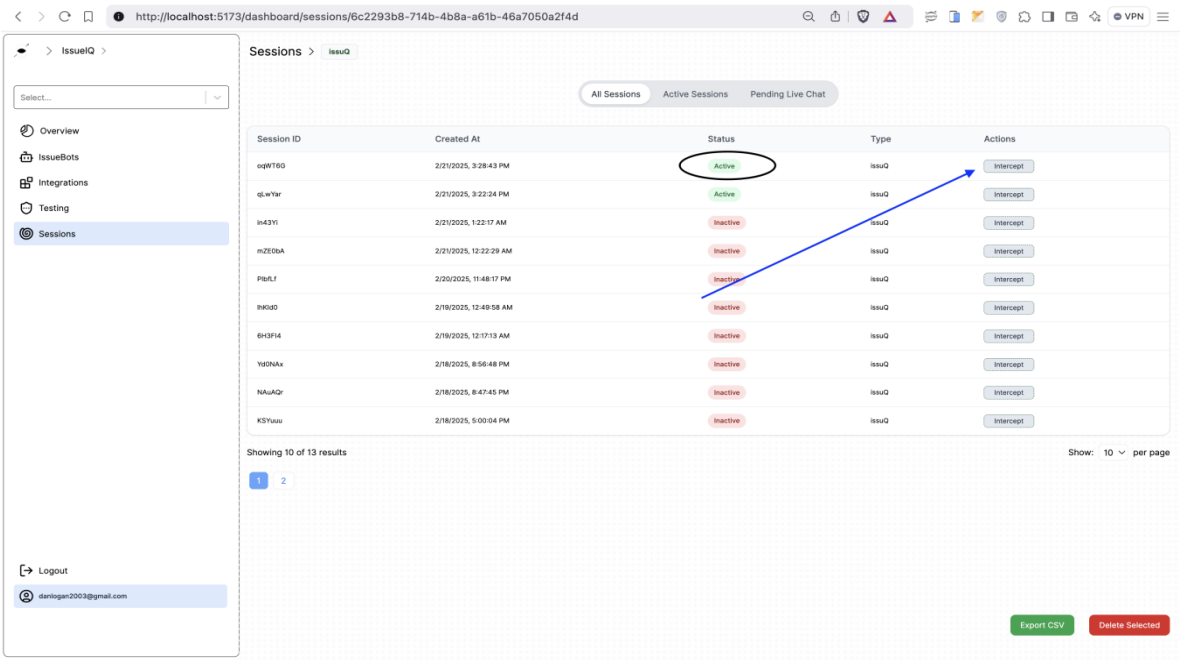

Figure 9: Session Page

Performance Evaluation

The overview page provides a high-level summary of the chatbot's performance and activity (Figure 10). Users can see key metrics such as total interactions, active sessions, and user engagement trends. This page helps users quickly assess how their chatbot is performing. The first graph described the rate at which the issuebot chatbot is being created that shows the total interactions on the system at different points. The second graph indicates the number of active sessions initiated at different points, while the third graph indicates the number of messages received on the chatbot describing the user engagements trends.

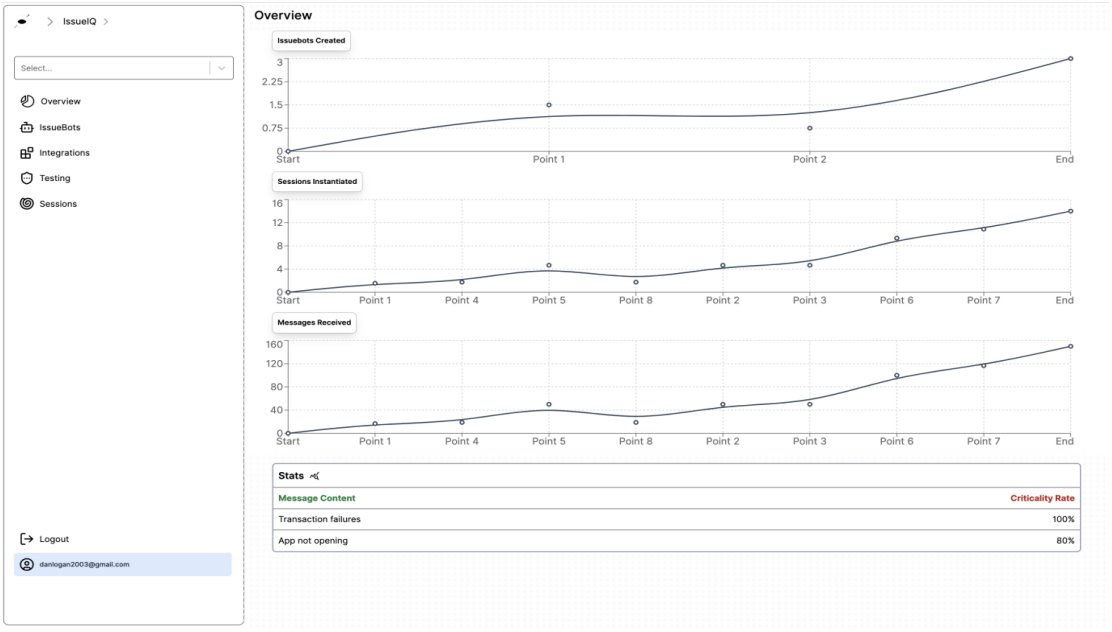

Figure 10: Evaluation Outputs

CONCLUSION

This research successfully achieved its goal of building a modern, intelligent customer service chatbot platform with human-in-the-loop capabilities. The system balances automation and personalization, allowing chatbots to handle simple inquiries while ensuring seamless escalation to human agents for more complex issues. The embeddable nature of the chatbot makes it highly adaptable to different industries, enabling companies to embed customer service directly into their existing platforms. The system shows that businesses can improve their customer service efficiency by actively monitoring chatbot performance data to continually improve their intent files and escalation processes. Also, companies can pilot the chatbot in controlled environments to capture real-world feedback and refine system performance. Future research should consider multilingual support for Chatbots, incorporate real-time sentiment analysis to detect user frustration and trigger

escalation earlier and explore direct integration with popular CRM platforms like HubSpot, Zoho, or Salesforce, to give businesses a 360-degree view of customer interactions.

ACKNOWLEDGEMENT

We want to appreciate the valuable inputs from the following undergraduate students of Software Engineering Department, Babcock University, Ilishan-Remo, Ogun state, Nigeria;

i. Ikotun Collins.
ii. Olawuyi Abd-Basit.
iii. Briggs Golda.

They contributed immensely during the process of development and testing of the system.

REFERENCES

Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications, 2*, 100006. https://doi.org/10.1016/j.mlwa.2020.100006.

Arcega-Punzalan, C.(2025). AI Customer Service – Enhancing Experiences Without Losing the Human Connection Retrieved 23 June 2025 from https://www.amworldgroup.com/blog/ai-customer-service

Coursera (2025). What Is Artificial Intelligence? Definition, Uses, and Types Retrieved 22 July 2025 from https://www.coursera.org/articles/what-is-artificial-intelligence.

Dixon, M., Freeman, K., & Toman, N. (2010). Stop trying to delight your customers. Harvard Business Review. https://hbr.org/2010/07/stop-trying-to-delight-your-customers.

Kumar, V., Ashraf, A. Nadeem, W. (2024). AI-powered marketing: What, where, and how? International Journal of Information Management, Volume 77, 2024, 102783, ISSN 0268-4012, https://doi.org/10.1016/j.ijinfomgt.2024.102783

Rafalski, K. (2025). Why AI Actually Makes Customer Experience More Human? Retrieved 23 June 2025 from https://www.netguru.com/blog/ai-in-customer-experience.

Rane,N.L Achari,A.and Choudhary, S.P.(2023). Enhancing Customer Loyalty Through Quality Of Service:Effective Strategies To Improve Customer Satisfaction,Experience, Relationship, And Engagement. *International Research Journal of Modernization in Engineering Technology and Science.* Volume:05/Issue:05/May-2023.

Tratta (n.d). AI's Role in Enhancing Customer Communications in Financial Services Retrieved 21 August 2025 from https://www.tratta.io/blog/ai-enhancing-customer-communications-financial-services.

Mayer, V., Schüll, M., Aktürk, O., Guggenberger,T.(2024). Designing Human-AI Hybrids: Challenges and Good Practices from a Multiple CaseStudy. Proceedings of Forty-Fifth International Conference on Information Systems Bangkok, Thailand.

Maheshwaram,V.(2024). The Evolution of Customer Service in the Digital Era. *OSR Journal of Business and Management* . e-ISSN:2278-487X, p-ISSN: 2319-7668. Volume 26, Issue 10. Ser. 12 (October. 2024), PP 01-05 https://doi.org/10.9790/487X-2610120105.