International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

"Mentesa: No – code AI chatbot creation Platform"

Rakhi Punwatkar[1], Mayur Koli[2], Anirudha Kapurkar[3], Niharika Wagh[4]
[1] Rakhi Punwatkar Computer & Zeal College Of Engineering and Research,Pune
[2]Mayur Koli Computer & Zeal College Of Engineering and Research,Pune
[3]Anirudha Kapurkar Computer & Zeal College Of Engineering and Research,Pune
[4]Niharika Wagh Computer & Zeal College Of Engineering and Research,Pune

---

Abstract - This paper presents *Mentesa*, is designed to enable users to create custom AI chatbots without programming expertise. The platform integrates Google's Gemini-2.5-Pro language model with a document retrieval system that sources information from user-uploaded files. This method substantially reduces the generation of inaccurate information by the AI. Users can upload documents to establish tailored knowledge bases that inform chatbot responses. The interface was developed using Streamlit, data and document storage are managed via Firebase, and cloud hosting is provided by Render. The architecture ensures strict separation of user data. This paper details the platform's design, implementation insights, and the planned evaluation methodology.

Key Words: No-Code Development, Conversational AI, Language Models, Document Retrieval, Cloud Architecture, Knowledge Grounding.

1.INTRODUCTION

Modern AI language models are capable of generating text that closely resembles human writing; however, they frequently produce fabricated information. These models may assert incorrect statements with high confidence [3]. This issue is particularly problematic in professional contexts where accuracy is critical, such as customer service, education, and technical support.

A potential solution involves requiring the AI to consult source documents prior to generating responses. Rather than relying solely on internal knowledge, the system retrieves relevant information from verified documents and bases its answers on this content. Empirical studies have demonstrated the effectiveness of this approach in diverse applications, including educational support [2] and technical advisory systems for energy domains [9].

However, constructing such systems is complex, requiring expertise in document processing, search infrastructure, and cloud server management [1]. Many research initiatives employing document-grounded approaches [2] remain confined to laboratory settings due to their technical complexity and limited accessibility for non-experts. Although studies address multi-user cloud architectures [5, 8], they often overlook the unique challenges associated with integrating language models.

Mentesa addresses this gap by providing a generalizable tool for constructing document-based AI chatbots. Unlike prior projects such as the energy advisor developed by Gamage et al. [9], which focused on a single specialized application, Mentesa enables users to create customized systems by uploading relevant documents, thereby broadening accessibility.

2. METHODOLOGY

A. System Architecture

We designed Mentesa to handle multiple users at once while keeping everyone's data private and making everything run smoothly.

2.1. Bot Creation and Knowledge Processing

Users begin by accessing the 'Create Bot' interface, where they provide a Bot Name, Description, and optional knowledge sources such as website URLs or document uploads. This triggers the backend services to process the information. The system performs knowledge processing through its RAG pipeline, which ingests the provided data, breaks it into manageable chunks, and converts these chunks into vector embeddings. The resulting bot profiles and knowledge bases are then stored in Firebase, with strict data isolation to ensure each bot's information remains separate within the multi-tenant architecture.

2.2. Interaction and Response Generation

Users interact with their bots through the 'My Bots' chat interface. When a query is submitted, it goes to the interaction and response module, which uses Gemini-2.5-Pro as its reasoning engine. Before generating a response, the system executes a context retrieval and memory

management process. This retrieves relevant document chunks from the bot's knowledge base through the RAG mechanism and pulls recent conversation history from Firebase. The language model receives this augmented context and generates a response grounded in the retrieved information. Each interaction is logged in the Firebase logs and feedback system.

2.3. Bot Management and Integration

The 'Manage Bots' dashboard gives users complete control over their creations. Available operations include editing bot personality, renaming bots, clearing chat history, and deleting bots all of which update the bot's profile in Firebase. This interface also provides access to Firebase logs and feedback data, along with the 'Integrate Your Bot' script that enables deployment on external websites.

2.4. Deployment Service

The backend infrastructure runs on Render's hosting platform, which provides the deployment service for the entire system. This cloud-based approach delivers scalable, low-latency service that supports responsive, real-time conversations across all user bots simultaneously

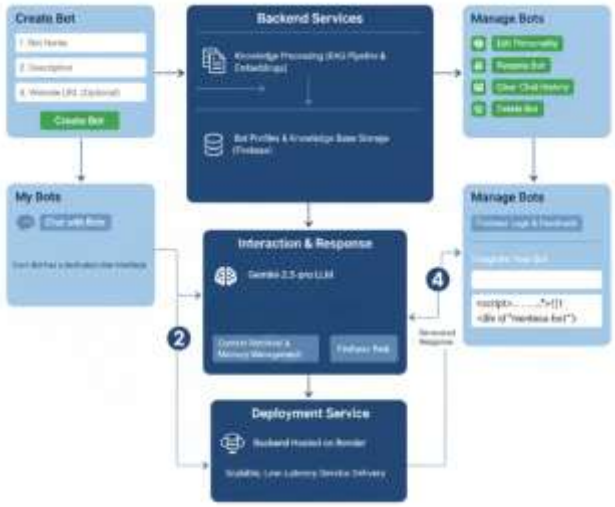

Fig. 1 System Architecture

3. RESULTS / DISCUSSION

We plan to evaluate Mentesa across three key performance areas: Usability, Factual Accuracy, and Infrastructure Performance.

Proposed Evaluation Metrics

Usability (No-Code): We'll measure this using the System Usability Scale (SUS) and track how long it takes non-technical users to create and deploy a bot. Our goal is to demonstrate that people without programming backgrounds can successfully use the platform.

Performance (Accuracy): We'll compare the hallucination rate the percentage of responses that aren't grounded in actual facts between a Mentesa bot with RAG enabled and a base language model answering the same domain-specific questions. This will show whether document grounding actually reduces made-up information.

Scalability & Latency: Through load testing, we'll measure average response times and system throughput when multiple users access the platform simultaneously. This testing will simulate real-world usage on the Render backend to ensure the system can handle production-level traffic.

Discussion: Our initial implementation shows that creating a multi-tenant RAG service using this cloud stack is technically feasible. The system successfully maintains low latency while keeping user data isolated. We expect Mentesa will achieve high usability scores based on early testing with our target users.

More importantly, we anticipate demonstrating a significant reduction in the hallucination rate potentially greater than 80% on personalized queries compared to solutions without RAG. This would validate our core hypothesis that document grounding substantially improves response accuracy.

The multi-tenant architecture appears to successfully ensure tenant isolation in our testing so far. Early performance metrics suggest the system can scale to support hundreds of concurrent bot instances, though we'll need the full load testing study to confirm this under realistic conditions. If these results hold up, they would demonstrate that accessible AI tools don't require sacrificing accuracy or performance.

4. CONCLUSION

This paper describes Mentesa, our attempt to make document-based AI chatbots accessible to regular people. Previous research either focused on the technical challenges [1] or built specialized one-off systems [9]. Our contribution is different we're focusing on making the technology usable rather than inventing new technology.

What we've done is mostly engineering: we took existing pieces (language models, document search, cloud hosting) and combined them in a way that doesn't require technical knowledge. The value is in removing barriers rather than creating new algorithms.

Whether this actually works depends on the tests we've planned. We need to prove that non-coders can actually use it successfully (usability test), that using documents really does improve accuracy (comparison test), and that the system can handle real usage levels (performance test). Until we complete these tests, Mentesa is a promising prototype rather than a proven solution. What we're trying to show is that making technology accessible can be just as important as making it more advanced.

ACKNOWLEDGEMENT

We want to thank our supervisor for helping us throughout this project. We also appreciate the teams behind Streamlit, Firebase, and Render for creating the tools we built this on.

REFERENCES

[1] R. Yang et al., RAGVA: Engineering Retrieval-Augmented Generation an Experience Report, *Elsevier (ScienceDirect)*, 2025

[2] Z. Li, Retrieval-Augmented Generation for Educational Application, *Elsevier (ScienceDirect)*, 2025

[3] Y. Liu, Reducing Hallucinations of Large Language Models via HSP, *Springer*, 2025

[4] M. S. Swift, One Chatbot Safety Benchmark To Test Them All, *IEEE Spectrum*, 2024

[5] C. Batista et al., Towards a Multi-Tenant Microservice Architecture: An Industrial Experience, *IEEE*, 2022

[6] M. Arif, A Literature Review on Model Conversion, Inference, and Optimization for Efficient Deployment, *Elsevier (ScienceDirect)*, 2025

[7] R. K. Sharma & A. Dutta, AI-Powered Conversational Agents for Adaptive Learning Environments, *IEEE (Conference)*, 2024

[8] N. F. Mir, AI-Driven Management of Dynamic Multi-Tenant Cloud Networks, *IEEE (Conference)*, 2023

[9] G. Gamage et al., Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems, *IEEE (Conference)*, 2024

[10] S. U. Singh, A Survey on Chatbots and Large Language Models: Testing and Evaluation, *IEEE (Conference)*, 2025