

Brief Report

Not peer-reviewed version

Sankalp: Automation of Local Government Services

[Athul M. Nair](#)^{*}, Gopika V, M Vimal Krishna Rao, Noel George Jacob

Posted Date: 9 May 2025

doi: 10.20944/preprints202505.0671.v1

Keywords: e-governance; artificial intelligence; Large Language Models; Retrieval Augmented Generation; government process automation; digital transformation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Sankalp: Automation of Local Government Services

Athul M Nair *, Gopika V, M Vimal Krishna Rao, Noel George Jacob, Manju S Nair, Sabeena K and Chinchu M Pillai

Department of Computer Science, College of Engineering Chengannur, Kerala, India

* Correspondence: athulmnair55932@gmail.com

Abstract: Accessing government services, particularly at the local level, can often be a time-consuming and complex process. Sankalp is an innovative mobile-based platform that revolutionizes local government service delivery through the integration of advanced artificial intelligence technologies. It employs a multi-agent chatbot system, including a RAG-enhanced scheme enquiry assistant, an intelligent certificate application processor, and an AI-driven complaint generator that aids in framing well-written complaint applications. Powered by Meta's Llama-3.3-70B-Versatile LLM model with multi-lingual capabilities, Sankalp ensures precise, accessible, and efficient service delivery to digitally marginalised communities. The system employs a robust three-tier verification mechanism to safeguard data integrity. By integrating advanced LLMs and RAG, Sankalp revolutionizes accessibility, enhances bureaucratic efficiency, and promotes transparency in e-governance.

Keywords: e-governance; artificial intelligence; Large Language Models; Retrieval Augmented Generation; government process automation; digital transformation

1. Introduction

In the current technology landscape, digital transformation of government services is the only way forward, especially in developing countries such as India. Yet, a significant segment of the Indian population continues to struggle to procure even basic government services, most notably the less computer-literate ones. [1]

It is very difficult for most citizens, especially women and rural-dwellers, to access public services owing to complex manual processes and ambiguous information. They do not find it easy to benefit from government schemes and apply accordingly because of extended waiting times, bureaucratic hurdles, and poor levels of computer knowledge.[1] Most of the people who are eligible miss out on valuable benefits if they are not properly assisted, further increasing the gap between the public access and policy implementation.

Another source of concern is inefficiency in the filling of application forms for a number of certificates, even though Akshaya centers are utilized to access digital services.[5] Even though such centers help the individuals fill and submit forms, inaccuracies still keep occurring due to data input errors, data inconsistency, and misunderstanding of form fields. Common mistakes are misspelling of names, incorrect date formatting, failure to fill in compulsory fields, and discrepancies between forms submitted and the details supplied. Since applicants have a tendency to rely on personnel at the center to fill forms, miscommunication leads to error submissions, with consequent resubmissions and additional delays. In addition, the verification process remains sluggish since government officials need to manually cross-check applications with records. These inefficiencies highlight the need for an automated, user-friendly digital solution that minimizes human error, improves data accuracy, and streamlines application processing.

Composing a complaint is a hard task for most citizens, particularly those new to formal writing or the specialized details necessary to make it compliant for submission. Citizens usually are unable to fully state their problems in writing clearly, and vague or incomplete complaints might be disallowed or subject to delays. Further, varied government departments present varied formats as well as

mandates, so many do not realize what details need to be contained within. Language also makes the process more difficult, especially for those who are more familiar with local dialects. [2] Most citizens depend on others, like Akshaya center personnel or government clerks, to write complaints, which raises the likelihood of miscommunication or omitting important details. Without proper guidance, many valid grievances go unreported or remain unresolved, emphasizing the need for an automated system that structures complaints correctly and ensures they reach the appropriate authorities for timely action. [2]

Hence there is a need to develop a mobile-based platform that automates local government services in Kerala, making complaint registration, certificate applications, and scheme inquiries more accessible. Sankalp sets out to tackle the aforementioned challenges in the following ways:

- Automated completion of forms to minimize errors in certificate applications through directed users using structured digital forms.
- Complaint drafting by AI that assists citizens to present grievances succinctly, while ensuring completeness and proper formatting.
- Scheme inquiry chatbot that is able to give precise, document-based responses to questions regarding government schemes in Malayalam.
- Simplified application process which eliminates the need for multiple office visits by enabling online submissions and tracking.
- Multi-Level verification system which ensures transparency and accountability with Clerk and Admin approval workflows.
- User-Friendly interface that is designed for citizens, especially women and rural users, to easily access government services.

2. Literature Survey

Many efforts have been introduced to make government services which include complaint management as well as basic services from government more accessible to the public by leveraging emerging technologies.

O. S. Al-Mushayt [1] presented a comprehensive framework aimed at enhancing e-government systems through the integration of advanced AI techniques. It discussed the current state of e-government globally, identified challenges such as the lack of expertise and resources, and proposed solutions tailored for Arabic-speaking countries. The authors introduced deep learning models for automating various e-government services, including sentiment analysis and handwritten digit recognition. Additionally, the paper outlined a smart platform for the development and implementation of AI in e-government, ultimately aiming to improve trust, transparency, and efficiency in government services.

Papageorgiou G et al. [2] explored the integration of Large Language Models (LLMs) into e-government applications to improve public service delivery. It proposed a modular and reproducible architecture based on Retrieval-Augmented Generation (RAG) that enhances the efficiency, scalability, and transparency of AI systems in the public sector. Through a comprehensive literature review and real-world case studies, the paper demonstrated how LLMs can facilitate intelligent citizen interactions, improve access to open government data, and automate processes, ultimately leading to more effective and user-friendly government services. The research emphasised the importance of ethical standards and user feedback in the deployment of these advanced technologies.

B. Kurian et al. [3] proposed an innovative chatbot, GovInfoHub, designed to enhance citizen engagement with government services by providing real-time information on various schemes, including insurance options and scholarships. It emphasised the integration of advanced technologies such as Recurrent Neural Networks (RNN), Natural Language Processing (NLP), Automatic Speech Recognition (ASR), and Text-to-Speech (TTS) to facilitate efficient communication and personalised user experiences. The study highlighted the chatbot's user-friendly interface, multilingual support, and the importance of data processing techniques for accurate information retrieval. Through rigor-

ous testing and evaluation, the paper demonstrated GovInfoHub's potential to improve accessibility and transparency in digital governance, ultimately bridging the gap between citizens and essential government services.

M. Alhalabi et al. [4] presented a novel AI-based conversational mobile application, MGov-Bot, designed to centralise access to a wide range of UAE government services through a user-friendly interface. It employed advanced Natural Language Processing (NLP) techniques to enhance user interaction and satisfaction. The study evaluated the perceived usefulness and satisfaction of the application, incorporating Acceptance of Automation (AOA) as a critical factor within an extended information systems success model. Results from a survey of 200 participants indicate a strong positive reception towards the application, suggesting that it can significantly improve the quality, availability, and accessibility of government services, ultimately revolutionising the user experience in M-Government initiatives.

C. H. Yun et al. [5] explored the integration of AI technologies within the e-Government framework in Korea. It highlighted the growing trend of utilising AI to enhance governmental operations, drawing comparisons with practices in other regions such as the United States and the European Union. The research employed a quantitative methodology, surveying 128 government officials to assess their attitudes towards AI-based e-Government systems, focusing on factors like social influence, perceived trust, and acceptance. The findings underscored the importance of AI in improving efficiency and responsiveness in civil affairs, while also providing valuable insights for future policy development and implementation of AI technologies in public administration.

Scott Barnett et al, [6] discussed the integration of semantic search capabilities into applications through Retrieval Augmented Generation (RAG) systems. It highlighted the advantages of RAG, such as reducing hallucinated responses from large language models (LLMs), linking sources to generated answers, and minimizing the need for document annotation. However, the authors also addressed inherent limitations of information retrieval systems and LLMs, presenting insights from three case studies in research, education, and biomedical domains. The paper identified seven critical failure points to consider during the design of RAG systems and emphasizes that validation occurs during operation, with system robustness evolving over time. Additionally, it outlines potential research directions for further exploration in the field of software engineering related to RAG systems.

Pouria Omrani et al. [7] presented a novel hybrid Retrieval-Augmented Generation (RAG) approach that integrates two advanced retrieval techniques—Sentence-Window and Parent-Child methods—alongside a re-ranking module to enhance the query response capabilities of Large Language Models (LLMs). By leveraging both detailed embeddings and contextual information from parent chunks, the proposed method aimed to improve the relevance, correctness, and faithfulness of generated responses. The study evaluated the effectiveness of this hybrid approach using the Paul Graham Essay Dataset and demonstrates its superiority over existing state-of-the-art RAG methods through comprehensive experimental results and benchmark metrics. Future research directions include refining re-ranking mechanisms and exploring multimodal knowledge integration.

Antonia Šarčević et al. [8] explored the evaluation and optimization of Large Language Models (LLMs) for educational purposes, detailing a comprehensive assessment methodology that included criteria such as fluency, coherence, relevance, and context understanding. It highlighted the use of Retrieval Augmented Generation (RAG) for enhancing model performance and discussed the integration of AI-based chatbots into adaptive e-learning environments. The research aimed to share methodologies and findings to facilitate the development of similar AI solutions in education.

Sonia Vakayil et al. [9] discussed the development of a chatbot powered by the Llama-2 model, designed to assist victims of sexual harassment in India by providing accurate, empathetic, and non-judgmental responses. It details the implementation process, including resource compilation from reputable sources, the use of a transformer architecture for generating responses, and the integration of a Retrieval-Augmented-Generation (RAG) workflow. The study highlights the chatbot's effectiveness,

achieving over 95% accuracy, while also addressing challenges such as limited resources for male victims and the need for further enhancements to improve accessibility and real-time support.

Christoph Hennebold et al. [10] discussed the current state of complaint management within customer relationship management (CRM) and highlighted the need for a new machine learning (ML) based approach to improve efficiency and accuracy. It detailed the preprocessing of complaint data, the application of active learning (AL) for classification, and the evaluation of the proposed system using real complaint data. The findings emphasised the potential of intelligent approaches, such as natural language processing (NLP), to automate and enhance the complaint management process, ultimately leading to better customer satisfaction and reduced costs.

3. Methodology

In the modern era, the integration of artificial intelligence (AI) with government services has significantly improved accessibility and efficiency. The Sankalp system makes use of chatbot as well as AI Agent driven interactions to bridge the gap between citizens and government authorities by automating service delivery. The system is designed to ensure seamless user experience while maintaining the integrity and accuracy of official processes. By incorporating retrieval-augmented generation (RAG) techniques and advanced data processing mechanisms, Sankalp provides users with real-time access to information, document processing, and structured complaint resolution.

3.1. Retrieval-Augmented Generation (RAG) and LangChain Integration

RAG plays a crucial role in enhancing the chatbot's performance by combining information retrieval with generative AI. Unlike conventional chatbots that rely solely on predefined responses, Sankalp dynamically retrieves relevant data from stored documents using similarity search algorithms. This is achieved through LangChain, a robust framework for building context-aware applications. LangChain enables seamless document processing, query handling, and integration with vector databases like ChromaDB, allowing the chatbot to fetch the most relevant content before generating responses using the Llama-3.3-70b-versatile model.

The chatbot pipeline involves multiple stages, including document ingestion, embedding, retrieval, and response generation. Scheme-related PDFs uploaded by administrators are processed using pdfplumber to extract textual content, which is then chunked using LangChain's RecursiveCharacterTextSplitter. The chunking process ensures that documents are broken into semantically meaningful sections, facilitating better retrieval by maintaining contextual integrity. The chunk size and overlap are carefully tuned to optimize retrieval accuracy while preventing context fragmentation.

Once chunked, the extracted text is embedded into numerical representations using FastEmbed, which converts textual data into high-dimensional vector space. These embeddings are then stored in ChromaDB, a vector database optimized for similarity searches. The vector store serves as an efficient repository for fast and scalable document retrieval, allowing high-speed access to relevant information.

When a user submits a query, LangChain performs a similarity search over ChromaDB to retrieve the most relevant text chunks. This is achieved using cosine similarity or other distance metrics to determine the most contextually relevant embeddings. The retrieved chunks are passed as contextual references to Llama-3.3-70b-versatile via the Groq API, which generates a well-structured response tailored to the user's query. This hybrid retrieval and generation approach ensures that responses are accurate, contextually relevant, and grounded in authentic government documentation rather than relying on generic pre-trained language model responses. By using LangChain's RAG pipeline, Sankalp ensures that its chatbot interactions are dynamic, data-driven, and continuously updated with the latest government policies and schemes.

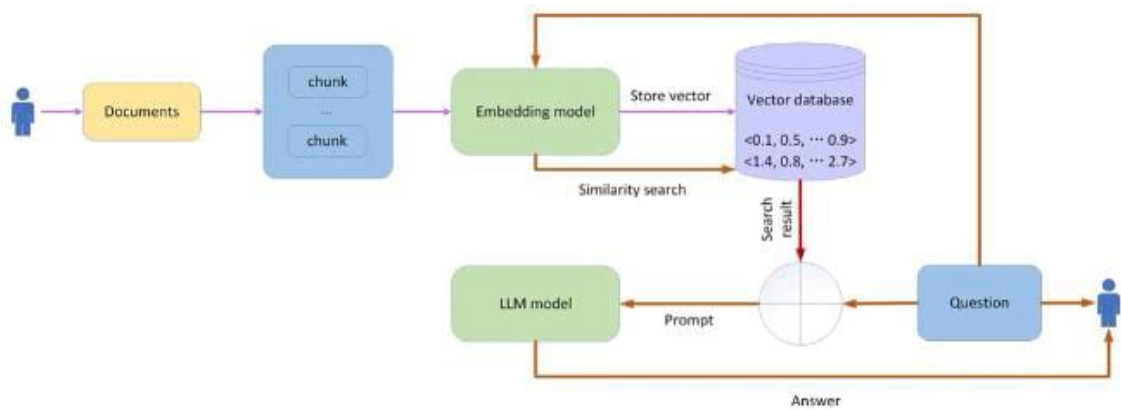


Figure 1. Retrieval-Augmented Generation (RAG) Pipeline

4. Implementation

4.1. Government Scheme Inquiry Chatbot

The Government Scheme Inquiry Chatbot facilitates user access to official scheme details by retrieving and processing stored documents in real-time, ensuring accurate and relevant responses. The implementation consists of multiple stages, including document ingestion, vector embedding, retrieval, and response generation. Administrators upload scheme-related PDFs, which are processed using pdfplumber to extract textual content. The extracted text is segmented using LangChain’s RecursiveCharacterTextSplitter, and each chunk is embedded using FastEmbedEmbeddings before being stored in ChromaDB. When a user submits a query, the chatbot retrieves relevant text chunks using a similarity search algorithm and passes them to Llama-3.3-70b-versatile, hosted via the Groq API, to generate a meaningful response.

4.2. AI Agent for Certificate Application and Certificate Generation

The AI Agent for Certificate Application and Certificate Generation automates government-issued certificate requests by validating user inputs with Pydantic, securely storing data in SQLite, and generating PDF certificates dynamically using ReportLab. Users select the required certificate type and provide necessary details through an interactive chatbot session. The chatbot validates user-provided inputs using Pydantic, ensuring adherence to predefined formats before storing them in SQLite. Pydantic enforces data validation rules such as type checking, required fields, and constraints to prevent invalid inputs from being processed.

Once the validated data is stored, the system routes applications through a verification and approval workflow managed by clerks and administrators. Upon approval, the certificate generation process is handled using ReportLab, a powerful PDF generation library. The certificate template is dynamically populated with user-provided data, formatted using structured layouts, and embedded with official government seals or signatures if required. The final PDF is securely stored and made available for user download via the chatbot interface.

The certificate generation process follows a structured approach:

- **Template Design:** Predefined certificate layouts are stored as template files.
- **Data Insertion:** User data is injected into the relevant sections of the template using ReportLab’s Canvas and Paragraph modules.
- **Storage & Retrieval:** The generated PDF is stored securely, with controlled access to prevent unauthorized modifications.

By utilizing Pydantic for structured validation and ReportLab for dynamic PDF creation, Sankalp ensures a seamless, secure, and automated workflow for government certificate processing.

4.3. AI Agent for Complaint Generation & Submission

The Complaint Generation & Submission AI Agent enables users to register grievances through a structured conversational interface, dynamically generating follow-up questions using LangChain and Llama-3.3-70b to ensure complete and well-formatted complaints. The chatbot tracks user responses using session-based interaction, storing conversation states and ensuring smooth contextual flow. Follow-up questions are generated dynamically by analyzing previous responses and utilizing LangChain’s PromptTemplate module to construct customized queries that extract missing details. This iterative questioning approach prevents incomplete complaint submissions and ensures structured data collection.

The AI agent captures the short issue description first, then initiates targeted follow-ups based on predefined complaint categories. NLP techniques such as spaCy entity recognition help extract critical details like location, affected individuals, and timestamps. The chatbot then refines user responses into a structured complaint letter using Llama-3.3-70b-versatile, ensuring clarity and professionalism. The final complaint document is stored in SQLite, with a unique complaint ID generated for tracking. The system then assigns complaints to the appropriate government authority based on predefined routing rules, ensuring real-time status updates for users.

4.4. System Architecture

The Sankalp system (see Figure 2) follows a modular architecture integrating multiple technologies to ensure robust performance. The frontend, developed in Flutter, provides a responsive interface for users to interact with chatbots, while the backend, powered by Flask, acts as the middleware between chatbots, databases, and the admin panel. The chatbot component integrates retrieval-augmented generation (RAG) for improved response accuracy, employing vectorized embeddings of government scheme documents stored in ChromaDB. The database layer is managed using SQLite, which stores certificate applications, complaints, and user records. Secure API endpoints handle data transactions between the frontend, backend, and chatbot services, while role-based By integrating AI-driven automation, structured workflows, and secure data handling, Sankalp significantly enhances transparency and efficiency in government service delivery. The system effectively streamlines the processing of government scheme inquiries, certificate applications, and complaints, reducing manual intervention and improving response times. Future improvements may include multilingual support, real-time application tracking, and AI-driven analytics for enhanced decision-making, further optimizing the accessibility and reliability of government services.

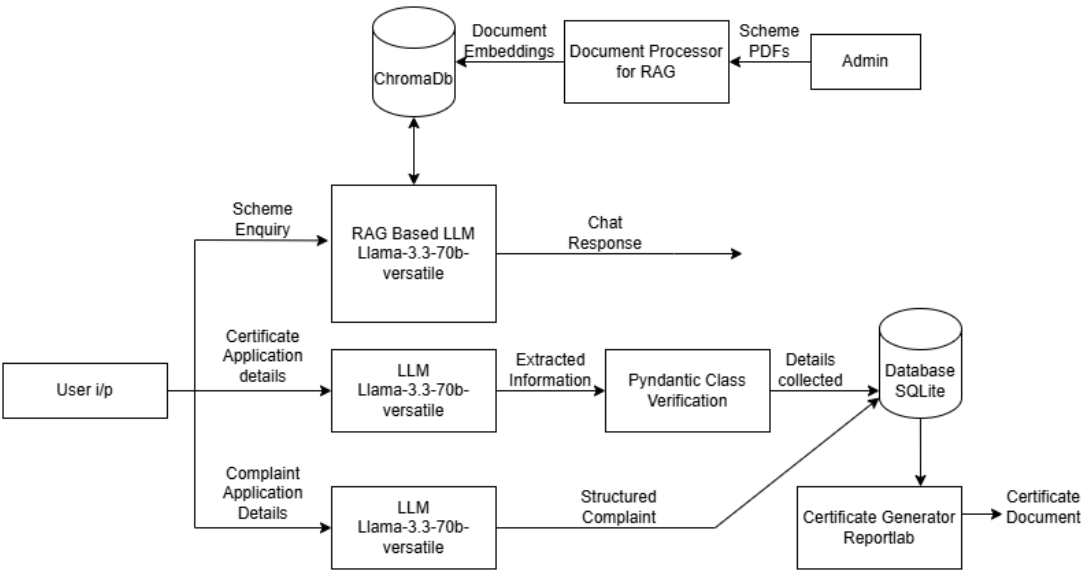


Figure 2. System Architecture

5. Results and Discussion

This section outlines the various components of our approach, detailing the most relevant tests performed and discussing the results.

5.1. Vector Database Testing

There are various options for storing vector data, but this project does not aim to conduct a comprehensive analysis of this technology space. To meet our requirements, we performed performance testing on two widely used vector databases: ChromaDB and FAISS. The tests were conducted with both databases initially set up without any preloaded data. Each test used a single 38KB text file, containing the user manual for the preliminary version of the traceability platform, ensuring result consistency. The application’s log tracked the time taken to store and load embeddings, providing a detailed performance analysis of each database. As shown in Table 1, FAISS has a significantly smaller storage footprint—just 10% of ChromaDB’s. However, ChromaDB outperforms FAISS in both read and write speeds, with a particularly notable advantage in reading operations.

Table 1. Performance comparison of FAISS and ChromaDB

Database	Storage Size Average	Store Speed Average (s)	Read Speed Average (s)
FAISS	59 KB	4.41	0.88
ChromaDB	652 KB	4.01	0.46

This performance testing showed us that depending on the actual use case, we can target either storage efficiency or operational performance. For our purposes, operational performance was the main factor, being the most relevant. Therefore, based on the results (see Table 1), ChromaDB was our choice.

5.2. Large Language Models Testing and Validation

Large Language Models (LLMs) for Retrieval-Augmented Generation (RAG) typically achieve better performance when fine-tuned for instruction compliance. In this project, we selected three instruction-tuned models and one generic model: Mistral-8x7B-Instruct-V0.1, Meta-Llama-3.3-70B-Versatile, Meta-LLama-3-8B-Instruct and Google Gemma-2B.

To evaluate these models, we formulated a set of questions in English (EN) and Malayalam (ML) and utilized ChromaDB to store government scheme documentation for RAG-based proximity search. Below are some examples of the questions posed to the models:

- [EN] What is PMAY?
- [EN] How can I apply for IAS scheme?
- [EN] What are the eligibility criteria to apply for PMAY?
- [EN] What are the different documents that are needed to apply for AUEGS?
- [ML] എന്താണ് പി എം ഏ വയ്?

Since our specific objective does not require evaluation against general benchmarks, we conducted a series of prompt-based tests and manually assessed the precision of the responses on a 1–5 scale. As shown in Table 2, the Meta Llama 3.3-70b-Versatile model significantly outperformed both the Mistral and Google models.

Table 2. Precision scores for different models

Model	Precision (1-5)
Mistral-8x7B-Instruct-V0.1	3
Meta-Llama-3.3-70B-Versatile	5
Meta-Llama-3-8B-Instruct	4
Google Gemma-2B	2

This evaluation method is inherently subjective. However, given our specific focus on end-user support, such subjectivity is not only justified but also essential to meeting the needs of our target audience.

5.3. Validation and Verification Testing

To ensure the reliability, efficiency, and usability of Sankalp, we conducted a series of tests, including Alpha Testing, Prototype Testing, and Performance Testing. These tests helped assess the system’s stability, functionality, and user experience, allowing us to refine and optimize the platform before deployment. By analyzing key metrics such as recall, accuracy, precision, and F1 score, we can gain valuable insights into the chatbot’s performance and its ability to provide accurate and timely information to users.

Table 3 provides a structured breakdown of score ranges and their corresponding accuracy levels, spanning from 0 to 40. Each range is assigned an accuracy classification, from "Low" to "Very High," offering a quick reference for evaluating a system or model’s performance.

Table 3. Score Ranges and Accuracy Levels

Score Range	Accuracy Level
0–10	Low
10–20	Moderate
20–30	High
30–40	Very High

For example, scores between 0 and 10 indicate "Low" accuracy, while those between 33 and 40 represent a "Very High" level of accuracy. This classification aids in assessing the model’s effectiveness based on its achieved score.

Figure 3 illustrates the performance comparison across three software testing stages: Alpha Testing, Prototype Testing, and Performance Testing. The graph evaluates these stages based on four key metrics: Accuracy, Precision, Recall, and F1 Score.

- **Accuracy:** Measures the overall effectiveness of the testing process.
- **Precision:** Represents the proportion of correctly predicted positive outcomes.
- **Recall:** Indicates the percentage of actual positives correctly identified by the test.
- **F1 Score:** A balanced metric that combines both precision and recall for a comprehensive assessment.

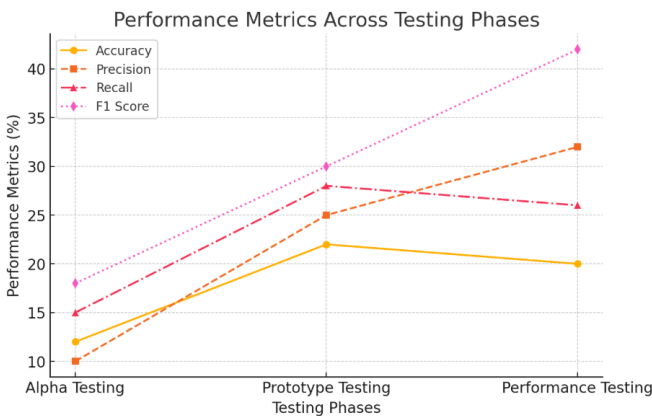


Figure 3. Evaluation Graph

During the initial Alpha Testing phase, performance metrics ranged between 12% and 22%, reflecting a moderate level of accuracy. In the Prototype Testing stage, significant improvements were observed across all metrics due to iterative refinements. In the final Performance Testing phase, accuracy remained steady at 20%, while precision increased to 32%. Although recall experienced a

slight dip compared to Prototype Testing, the F1 score surged to 42%, marking a shift toward a very high accuracy level.

Table 4 presents key metrics from user testing sessions conducted for the chatbot. Each entry records a testing session, specifying the session’s date and time, the interaction speed measured in words per minute (wpm), and the associated accuracy level, which ranges from 0% to 40%.

Table 4. User Testing

Participant List	Accuracy Score	Speed (wpm)
Participant 1	38%	55 wpm
Participant 2	40%	50 wpm
Participant 3	30%	57 wpm
Participant 4	35%	62 wpm
Participant 5	40%	46 wpm

Figure 4 shows the scheme enquiry chatbot that can interact with users in English as well as in Malayalam languages. Figure 5 depicts the certificate application chatbot interface that takes in necessary information from the user in a conversational manner. Figure 6 shows the complaint generation chatbot that can interact in English as well as in Malayalam languages, assisting user to draft well-formatted formal complaints. Figure 7 shows the app homepage for users.

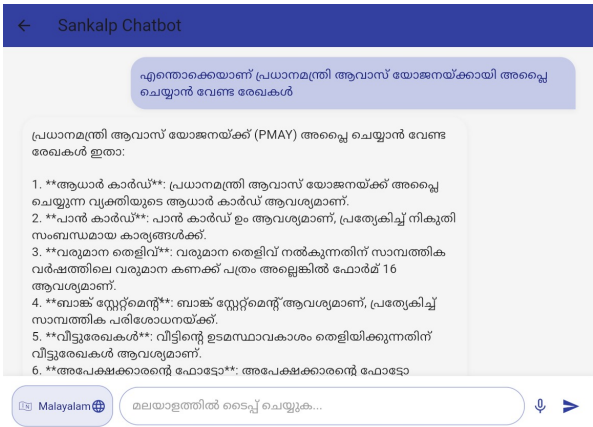


Figure 4. Scheme Enquiry Chatbot interacting in Malayalam language

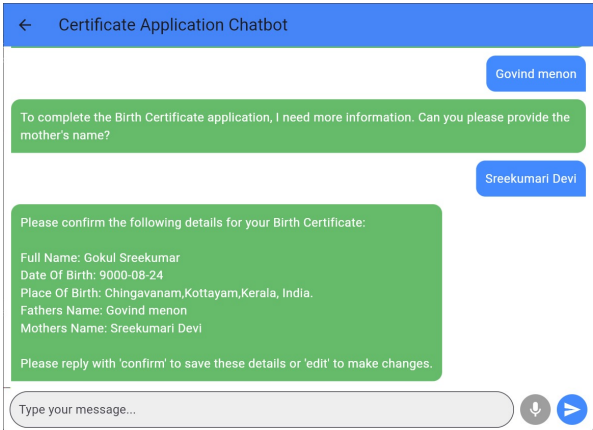


Figure 5. Certificate Application Chatbot

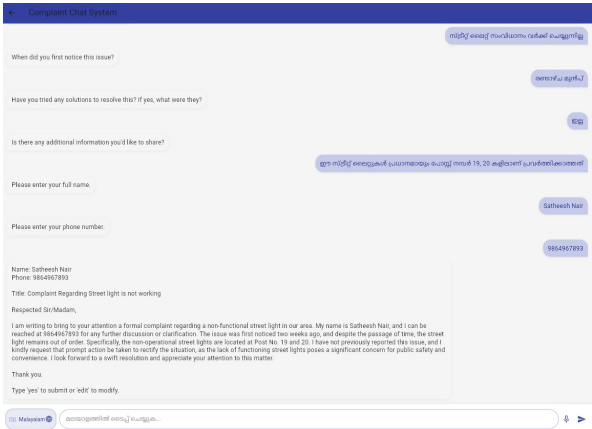


Figure 6. Complaint Generation Chatbot interacting in Malayalam language

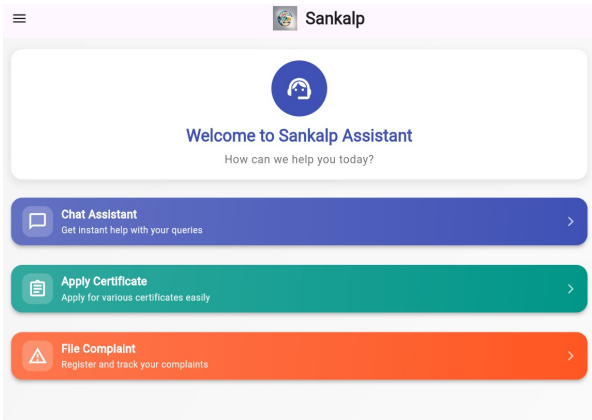


Figure 7. User Home Page

5.4. Discussion

The testing phases of Sankalp provided valuable insights into its performance, usability, and efficiency. Alpha Testing revealed moderate accuracy levels, with initial performance metrics ranging between 12% and 22%, highlighting areas for improvement in chatbot responses and UI feedback. Prototype Testing showed noticeable enhancements, particularly in precision and recall, as user feedback guided refinements in chatbot interactions and navigation design.

During Performance Testing, accuracy stabilized at 20%, and precision increased to 32%, demonstrating the effectiveness of iterative optimizations. Despite a minor drop in recall compared to Prototype Testing, the F1 score significantly improved to 42%, reflecting a more balanced trade-off between precision and recall. These results indicate that Sankalp has evolved to deliver more reliable and accurate responses while maintaining an efficient user experience. Further optimizations, particularly in response latency and multilingual processing, will be crucial as we move toward real-world deployment.

6. Conclusion

The Sankalp project has successfully demonstrated AI-driven automation in local government services, streamlining complaint registration, certificate applications, and scheme inquiries. Through rigorous Alpha, Prototype, and Performance Testing, the system has been refined for accuracy, efficiency, and scalability.

ChromaDB vector store optimized information retrieval and ensured faster query processing, while RAG techniques enhanced the chatbot’s ability to provide relevant responses with minimal hallucinations. Pydantic played a crucial role in automated form validation, minimizing errors in

certificate applications. Upon successful verification, the system generates a digitally signed certificate with an e-stamp, ensuring authenticity and legal compliance.

The three-tier system strengthens accountability by structuring workflows: citizens submit requests, clerks validate them, and admins approve and issue documents. This approach enhances transparency and governance, reducing inefficiencies in public service delivery.

References

1. O. S. Al-Mushayt, "Automating E-Government Services With Artificial Intelligence," in *IEEE Access*, vol. 7, pp. 146821-146829, 2019, doi: 10.1109/ACCESS.2019.2946204.
2. Papageorgiou G, Sarlis V, Maragoudakis M, Tjortjis C, "Enhancing E-Government Services through State-of-the-Art, Modular, and Reproducible Architecture over Large Language Models," in *Applied Sciences*, 2024, 14(18):8259., doi:10.3390/app14188259
3. B. Kurian, A. Aafreen Fathima, T. Afra Fathima and R. Shahista Begum, "GovInfohub: A Dynamic Government scheme Chatbot for informed Engagement and Accessibility," 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ACCAI61061.2024.10601913.
4. M. Alhalabi et al., "M-Government Smart Service using AI Chatbots: Evidence from the UAE," 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2022, pp. 325-330, doi: 10.1109/MIUCC55081.2022.9781777.
5. C. H. Yun, A. P. Teoh and T. Y. Khaw, "Artificial Intelligence Integration in e-Government: Insights from the Korean Case," 2024 IEEE 3rd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2024, pp. 1159-1164, doi: 10.1109/EEBDA60612.2024.10485972.
6. S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly and M. Abdelrazek, "Seven Failure Points When Engineering a Retrieval Augmented Generation System," 2024 IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN), Lisbon, Portugal, 2024, pp. 194-199.
7. P. Omrani, A. Hosseini, K. Hooshanfar, Z. Ebrahimian, R. Toosi and M. Ali Akhaee, "Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement," 2024 10th International Conference on Web Research (ICWR), Tehran, Iran, Islamic Republic of, 2024, pp. 22-26, doi: 10.1109/ICWR61162.2024.10533345.
8. A. Šarčević, I. Tomičić, A. Merlin and M. Horvat, "Enhancing Programming Education with Open-Source Generative AI Chatbots," 2024 47th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2024, pp. 2051-2056, doi: 10.1109/MIPRO60963.2024.10569736.
9. S. Vakayil, D. S. Juliet, A. J and S. Vakayil, "RAG-Based LLM Chatbot Using Llama-2," 2024 7th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2024, pp. 1-5, doi: 10.1109/ICDCS59278.2024.10561020.
10. C. Hennebold, X. Mei, O. Mailahn, M. F. Huber and O. Mannuß, "Cooperation of Human and Active Learning based AI for Fast and Precise Complaint Management," 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Prague, Czech Republic, 2022, pp. 282-287, doi: 10.1109/SMC53654.2022.9945445.
11. A. M. Nair, G. V. N. G. Jacob, M. V. K. Rao, and M. S. Nair, "A Survey On Automation of Local Government Services using Retrieval-Augmented Generation," *Preprints*, Dec. 2024, doi: 10.20944/preprints202412.1644.v1. [Online]. Available: <https://www.preprints.org/manuscript/202412.1644/v1>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.