TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A Progress Report

On

SarkariSathi - Digital Assistant for Government Works

Submitted By:

Batsal Bhusal (PUL078BEI009)

Birat Thapa (PUL078BEI012)

Submitted To:

Department of Electronics & Computer Engineering

December, 2025

Acknowledgments

We would like to express our sincere gratitude to the Department of Electronics and Computer Engineering, Pulchowk Campus, for providing us with the opportunity to propose our major project. The department's continuous encouragement and support have been instrumental in guiding us. We are deeply thankful to the Head of the Department, along with the Deputy Heads of Department, for fostering an academic environment that promotes research, innovation, and growth. We also extend our gratitude to the IC Chair, Asst. Prof. Bikal Adhikari, and Asst. Prof. Anila Kansakar for their guidance and feedback, which have been crucial in shaping our project. We are especially thankful to our supervisor, Asst. Prof. Bikal Adhikari, for his continuous support and insightful suggestions. Furthermore, we acknowledge the support of our seniors for their advice, which have motivated us throughout this process. We are equally grateful to our peers and colleagues for their cooperation and encouragement. This submission would not have been possible without the collaborative efforts and guidance of everyone mentioned above, and for that, we remain sincerely thankful.

Thank you.

Sincerely,

Batsal Bhusal

Birat Thapa

Abstract

Accessing accurate and up-to-date government service information is often difficult due to dispersed documents, heterogeneous formats, and multilingual content. This work presents **Sarkari Sathi**, a Retrieval-Augmented Generation (RAG) chatbot that provides concise, source-grounded answers about government services. The system ingests municipal charters, guidelines, and user-supplied files (`.docx`, text, tabular content), performs cleaning and semantic chunking, and encodes chunks and queries using `sentence-transformers/all-MiniLM-L6-v2`. Embeddings (384-d) are indexed in a Pinecone serverless vector database to enable efficient top-k semantic retrieval. A FastAPI backend exposes endpoints for chat, document management, and stats, while a responsive web UI delivers an accessible experience branded for Nepali users. Initial results show fast response times and robust retrieval for common queries; challenges include mixed-language (Nepali/English) queries, nuanced multi-turn questions, and variability in document structure. We discuss design choices, limitations, and a roadmap that adds multilingual embeddings, hybrid search, OCR ingestion, cached FAQs, and optional local LLM refinement to further improve accuracy, inclusivity, and scalability for digital government.

Contents

List of Figures

1. Introduction

1.1 Background

Citizens frequently struggle to obtain accurate, up-to-date information about government services. Key details—such as required documents, service fees, processing steps, timelines, and office responsibilities—are scattered across municipal websites, notices, PDFs, and charters that vary widely in format and quality (tables, scanned documents, mixed languages). Traditional keyword search often fails to capture the semantic meaning of queries (e.g., "कागजात के-के चाहिन्छ?") and returns incomplete or irrelevant results.

Sarkari Sathi addresses this gap by using Retrieval–Augmented Generation (RAG) to provide concise, source-grounded answers. The system ingests government documents, performs cleaning and semantic chunking, encodes text using sentence embeddings, and stores vectors in a Pinecone index for fast similarity search. A rule-based answer extractor returns precise elements—fees, required documents, processes, time, and service lists—via a FastAPI backend and a user-friendly web interface customized for Nepali users.

1.2 Problem Statement

Government information access faces several challenges that reduce reliability and citizen experience:

- **Fragmented Sources:** Service rules and documents are dispersed across formats (DOCX, PDFs, tables) and agencies.
- **Limited Searchability:** Keyword-based search misses semantic intent, especially for mixed Nepali/English queries.
- **Inconsistent Structure:** Citizen charters differ in layout, making automated extraction difficult.
- **Latency in Assistance:** Human support desks cannot scale to 24/7, high-volume information needs.

- **Explainability Needs:** Citizens and officials require answers grounded in verifiable sources.

A robust, multilingual, source-grounded assistant is needed to deliver accurate service information quickly and consistently.

1.3 Objective

The primary goal of **Sarkari Sathi** is to build a cost-efficient, explainable government information assistant using RAG. Specific objectives include:

- To develop an accessible chatbot that can answer queries about Nepali government services, local bodies, and general facts using diverse documentation, including tabular and scanned official documents.
- To build a scalable, updateable vector knowledge base using Pinecone.
- To provide a clean, intuitive web interface, enabling citizens to interact with government data without technical barriers.

1.4 Scope

The scope of **Sarkari Sathi** encompasses the following capabilities:

- **Document Ingestion & Parsing:** Import municipal charters and official documents (`.docx`, text, tabular content); normalize and semantically chunk content with metadata (title, source, category).
- **Semantic Retrieval:** Encode chunks and queries with `all-MiniLM-L6-v2` (384-d) and perform top-k cosine similarity search in Pinecone; apply a similarity threshold to improve recall.
- **Answer Extraction:** Rule-based extractor tailored to government FAQs (fees, documents, process, timeline, services), with graceful fallbacks to concise summaries.
- **Multilingual Handling (Initial):** Basic Nepali support via keyword routing and query preprocessing; mixed-language query tolerance.
- **Explainability:** Return cited sources and metadata to support trust and verification.

2. Methodology

The development of **Sarkari Sathi** follows a structured workflow integrating Information Retrieval (IR), Natural Language Processing (NLP), and Vector Similarity Search to enable accurate, fast access to government information. The methodology consists of the following key phases.

2.1 Data Collection

Data were collected from publicly available government sources (municipal citizen charters, service guidelines, official notices) and supplementary knowledge documents. The corpus includes Tabular citizen charters (e.g., Chandragiri Municipality as of now) describing services, fees, processes, and required documents.

Each document is associated with metadata (title, source, category) to support provenance and filtering.

2.2 Data Preprocessing

To ensure quality and consistency, the following preprocessing steps were applied.

2.2.1 Document Normalization

- **Text Extraction:** Parsed `.docx` files using `python-docx`; tables were linearized into structured bullet-lists while preserving semantics (e.g., "Service", "Required Documents", "Process", "Fee").
- **Cleaning:** Removed boilerplate headers/footers, HTML artifacts, extra spaces, and non-informative symbols.
- **Language Normalization:** Preserved multilingual content (English/Nepali) in UTF-8; light normalization without altering meaning.

2.2.2 Chunking and Metadata Attachment

- **Semantic Chunking:** Documents were segmented into coherent passages (service blocks, numbered lists, sections). Each chunk is small enough to be semantically atomic but large enough to retain context.
- **Metadata:** Every chunk is stored with *title*, *category*, *source*, and the *raw content* to support explainability.

2.3 Model Development

The core system uses dense embeddings and vector similarity for retrieval, followed by a lightweight rule-based generator for answer extraction.

2.3.1 Embeddings

Instead of keyword search, we use `sentence-transformers/all-MiniLM-L6-v2` (384-d) to embed both user queries and document chunks into a shared vector space. This enables semantic matching beyond exact term overlap.

2.3.2 Indexing and Retrieval

All vectors are upserted into a Pinecone serverless index (`rag-chatbot-free`, cosine metric, dim=384). At inference:

1. The user query is embedded with the same model.
2. Top-k ($k=5$) most similar chunks are retrieved via cosine similarity.
3. A similarity threshold ($\tau=0.3$) is applied to improve recall on short queries.

2.3.3 Answer Formulation (Rule-based RAG)

A deterministic extractor converts retrieved context into structured answers. Specialized extractors handle:

- **Fees/Costs** (e.g., “निःशुल्क” / “Free”).
- **Required Documents** (enumerated lists 1–5).
- **Process/Steps** (procedure lines).
- **Time/Duration** (e.g., “same day”, “within 3 days”).

- **Services** (service headings/lines).
- **Country Facts** (e.g., Nepal-specific queries).

If no extractor matches, a concise general summary of salient lines is returned.

2.4 Model Evaluation

We evaluated the retrieval and extraction quality using task-focused checks:

- **Precision of Extracted Facts:** Manual verification for fee, document lists, and process phrases against source charters.
- **Top-k Relevance Checks:** Spot-checked that at least one of the top-k chunks contains the answer span for representative queries.
- **Threshold Tuning:** Lowered similarity threshold to $\tau = 0.3$ to increase recall on short intents ("services", "Nepal"), validated by improved hit rates.

2.5 Model Optimization

We adopted several practical optimizations:

- **Local Embeddings:** MiniLM enables zero-cost inference and fast startup.
- **Chunk Design:** Service-aligned chunking increases the chance that lists (fees, documents) stay intact in a single chunk.
- **Extractor Iteration:** Relaxed patterns for "services" to capture broader phrasing (English/Nepali).

2.6 System Integration

2.6.1 Backend

Implemented with FastAPI, exposing:

- `/chat`: RAG pipeline endpoint (embed $\rightarrow$ retrieve $\rightarrow$ extract).
- `/add-documents`: Add new documents to the knowledge base (immediate availability).
- `/stats`: Index statistics (vector count, dimension).
- `/health`: Health and configuration status.

2.6.2 Frontend

A responsive HTML/CSS/JS chat interface branded as *Sarkari Sathi* , featuring:

- Typing indicator and retrieval badge (“Found N document(s)”).
- Clear, bulleted answers for documents/fees/process.
- Multilingual-friendly display.

2.7 Future Enhancements

- **OCR Integration:** Process scanned PDFs and images.
- **Hybrid Retrieval:** Combine keyword + dense retrieval to boost recall.
- **LLM-backed Generation:** Optional GPT-based generator for richer, multilingual answers when credits are available.
- **Admin Console:** Curate/update municipal datasets with versioning and audit logs.

3. System Design

3.1 System Architecture

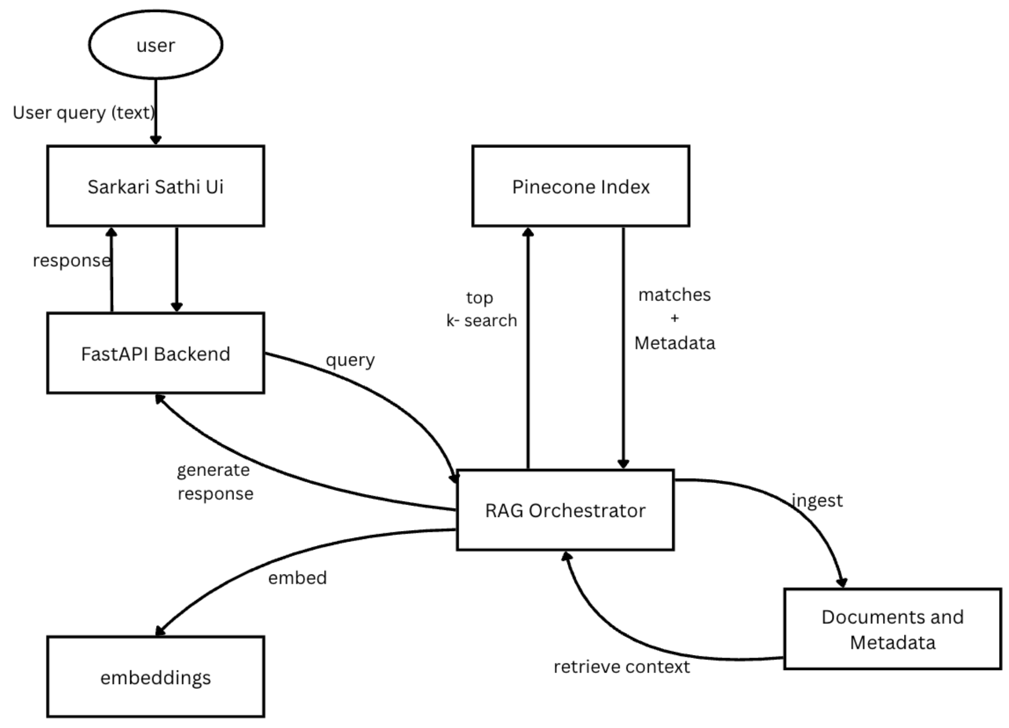


Figure 3.1: System Architecture of Sarkari Sathi

3.2 System Diagrams

3.2.1 Use Case Diagram

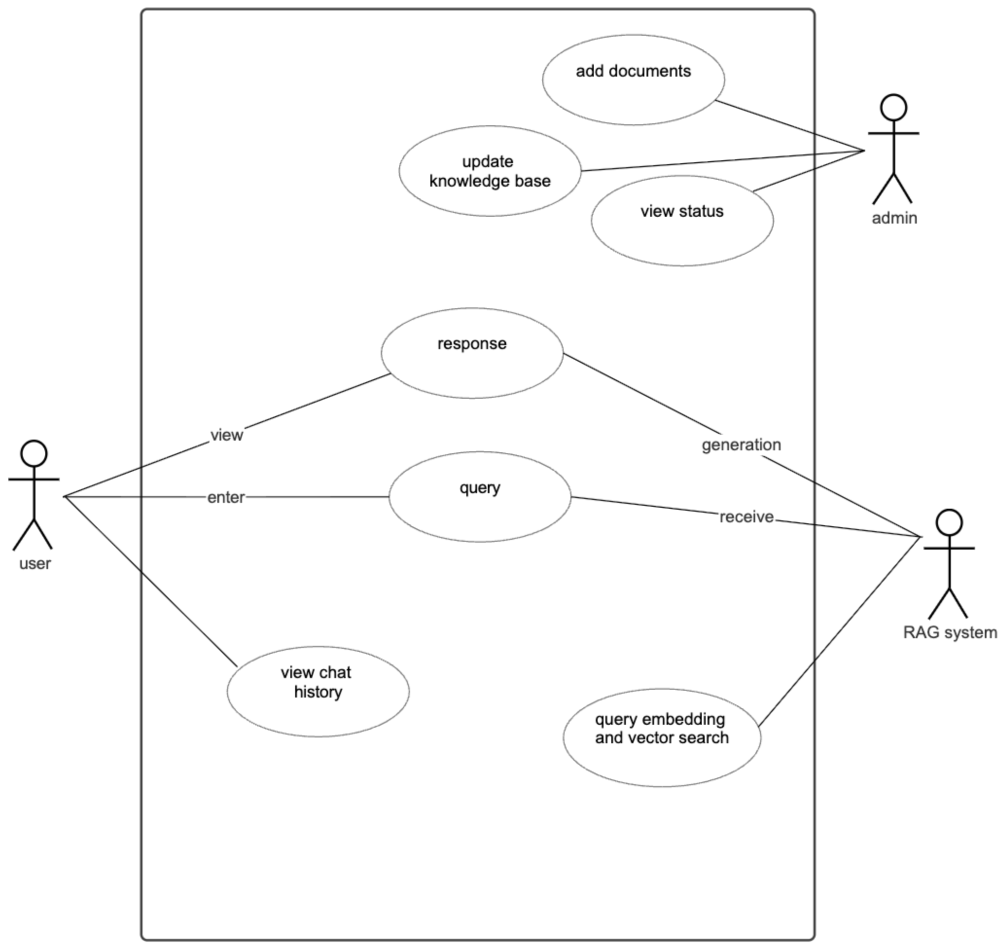


Figure 3.2: Use Case Diagram of Sarkari Sathi

3.2.2 Sequence Diagram

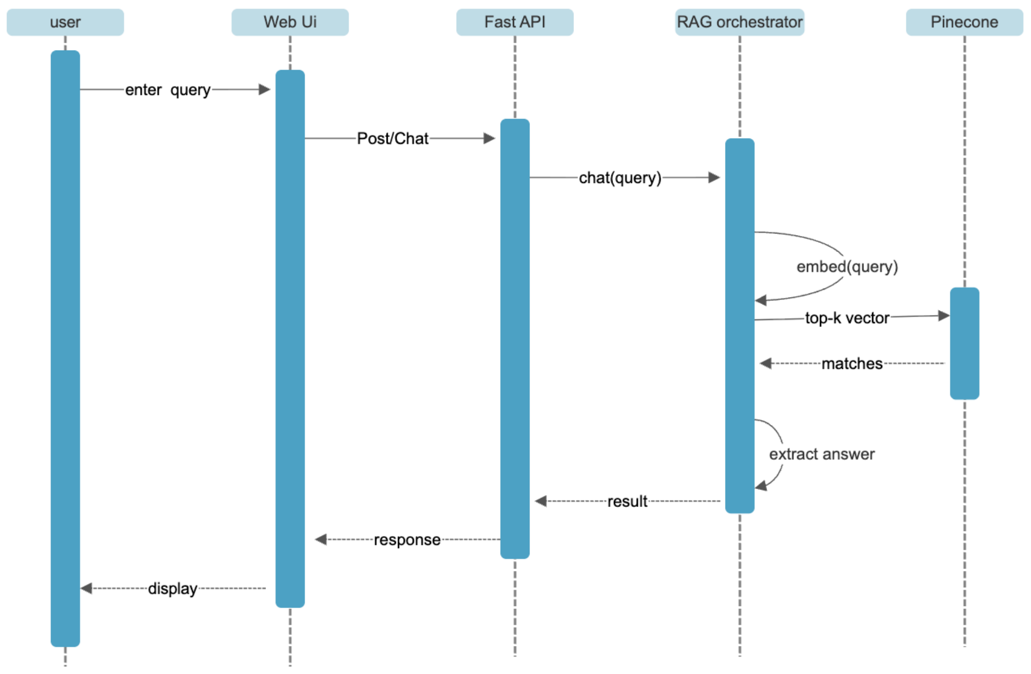


Figure 3.3: Sequence Diagram of Sarkari Sathi

3.2.3 Activity Diagram

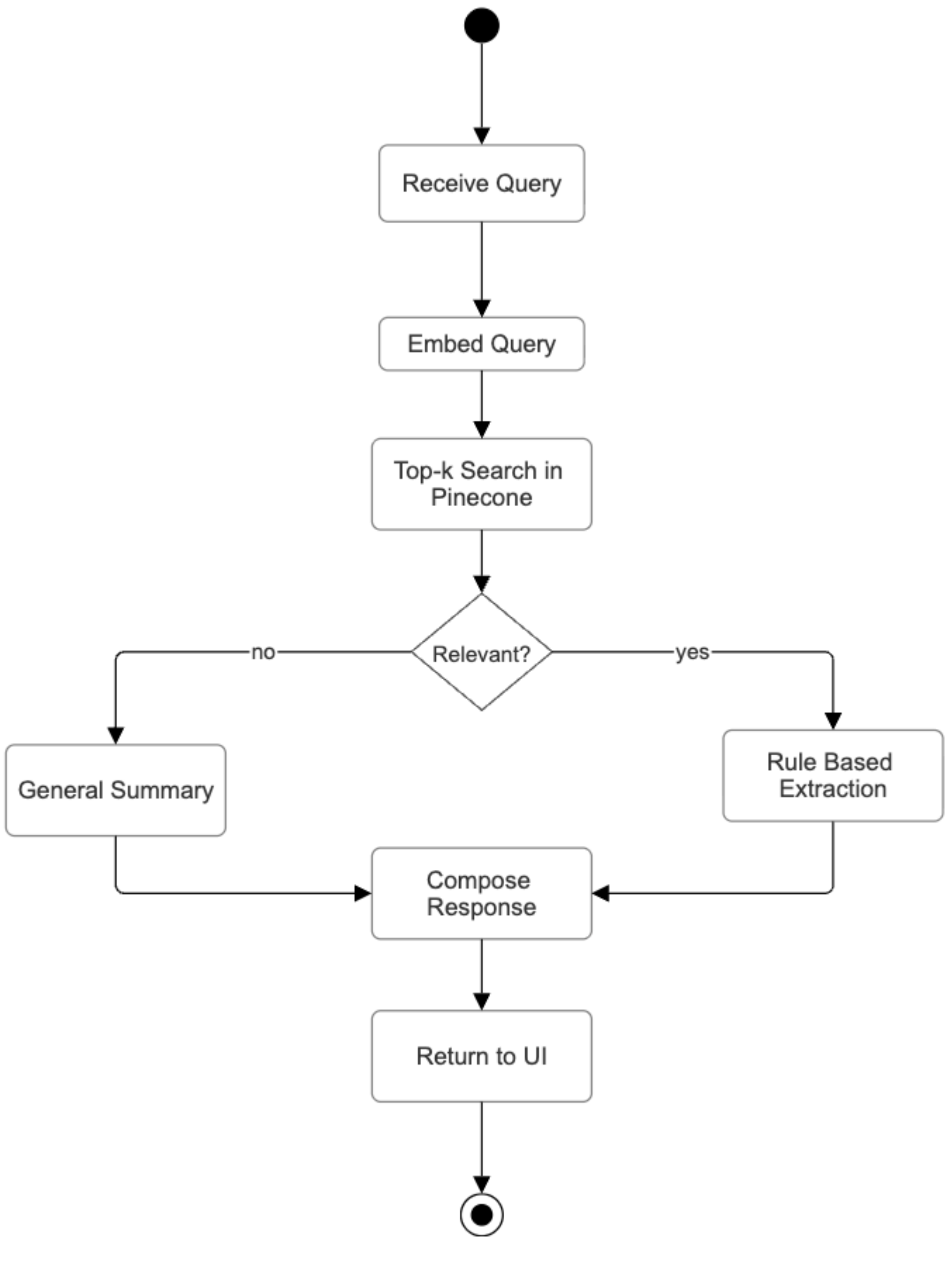
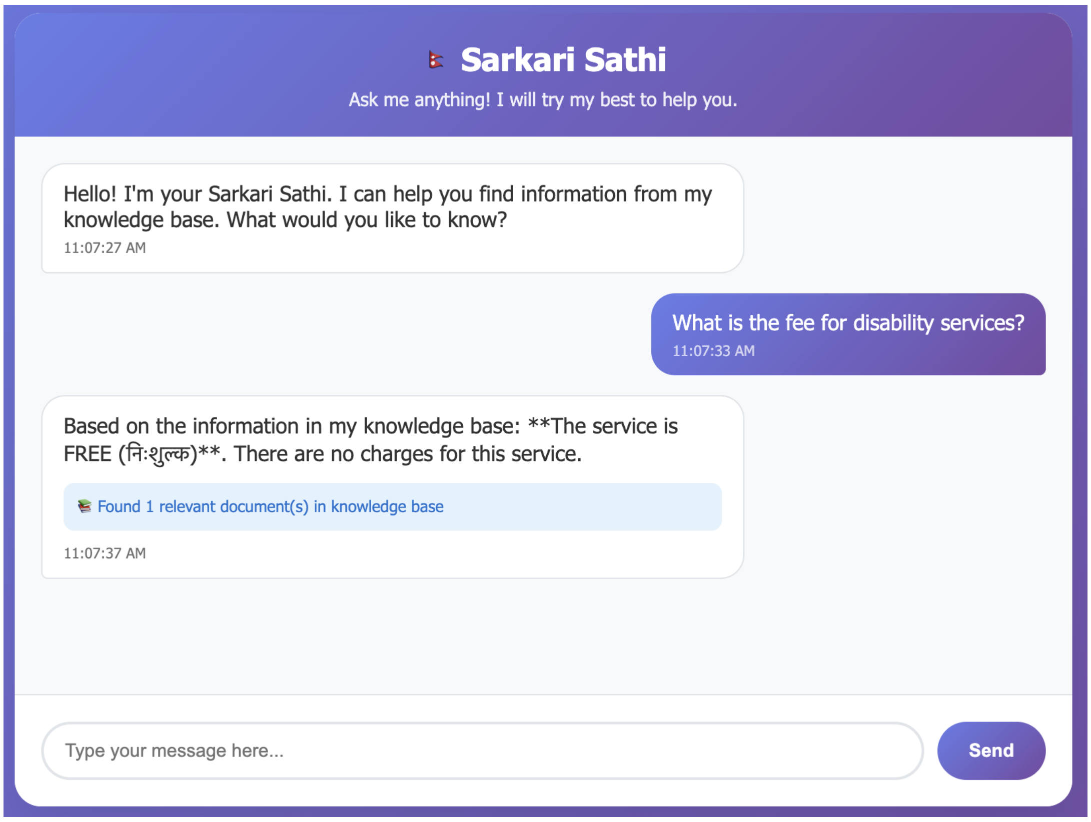


Figure 3.4: Activity Diagram of Sarkari Sathi

4. Work Completed So Far

- Developed pipeline scripts for data ingestion (including .docx and tabular PDF).
- Implemented cleaning, chunking and semantic embedding of diverse document types.
- Built and populated the Pinecone knowledge base with government and custom data (e.g., Nepal facts, Chandragiri Municipality services).
- FastAPI backend for queries, document addition, stats.
- Responsive chatbot UI (Sarkari Sathi)
- Added support for context extraction (documents, fees, processes, etc.).



Sarkari Sathi

Ask me anything! I will try my best to help you.

Hello! I'm your Sarkari Sathi. I can help you find information from my knowledge base. What would you like to know?
11:07:27 AM

What is the fee for disability services?
11:07:33 AM

Based on the information in my knowledge base: **The service is FREE (नि:शुल्क)**. There are no charges for this service.

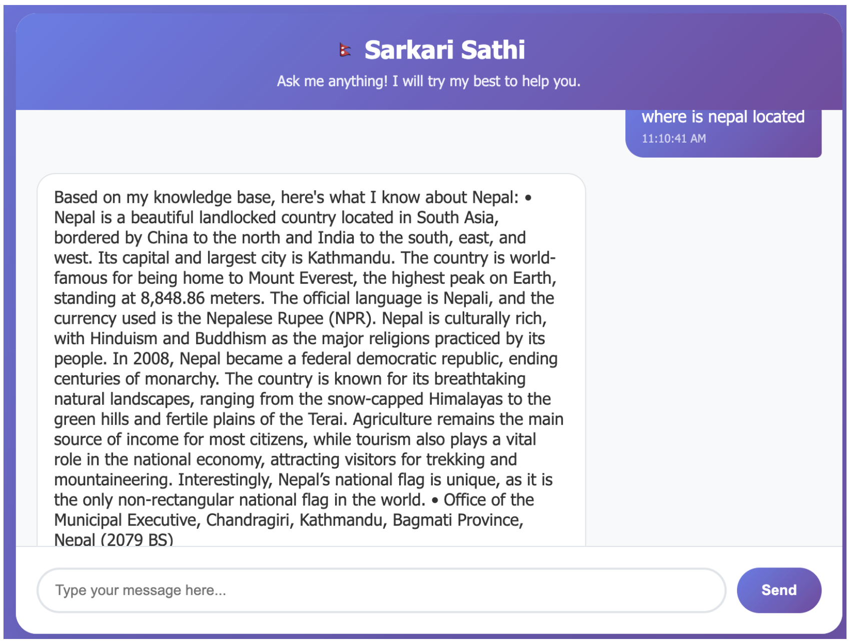
Found 1 relevant document(s) in knowledge base

11:07:37 AM

Figure 4.1: Chatbot Repsonse

5. Problems Faced

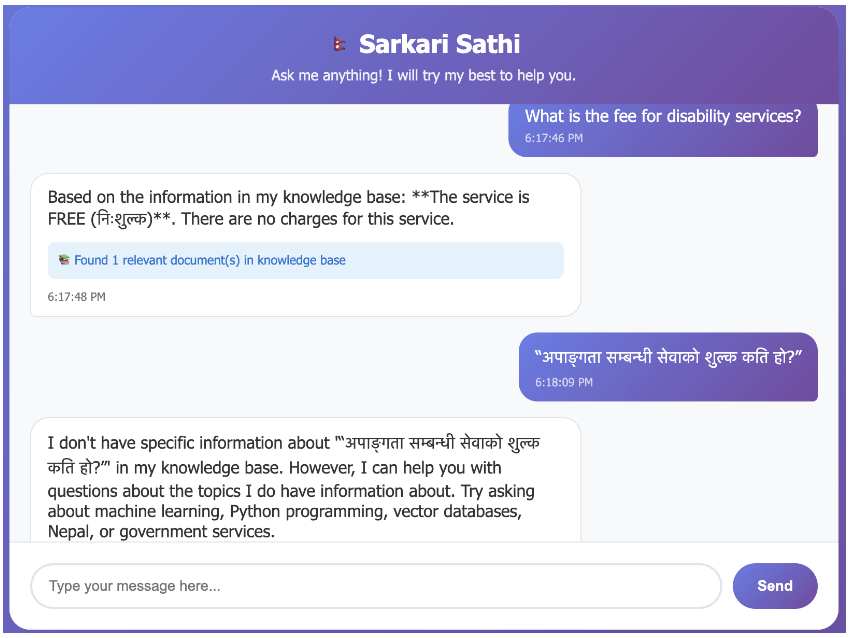
- **Document Format Diversity:** Official documents come in diverse formats (tables, paragraphs, Nepali/English) which required custom parsing and flexible chunking logic.
- **Nepali Language and Unicode Text Issues:** Some Nepali PDF text extraction lost formatting or missed characters, affecting embedding quality.
- **API and Library Changes:** Pinecone and OpenAI API changes required code refactoring (e.g., embedding endpoint migration, dimension mismatches).
- **Limited Free Tier Resources:** Using Pinecone's and OpenAI's free tiers imposed query limits, sometimes causing slowdowns.
- **Rule-Based Generation Limitations:** Rule-based extraction sometimes misses specific answers and General Extraction Method is used to extract inaccurate response.
- **Training of Embedding model:** Embedding model is trained on English text only and no specific handling for Devanagari script.


Sarkari Sathi
Ask me anything! I will try my best to help you.
Type your message here...
Send

where is nepal located
11:10:41 AM

Based on my knowledge base, here's what I know about Nepal: • Nepal is a beautiful landlocked country located in South Asia, bordered by China to the north and India to the south, east, and west. Its capital and largest city is Kathmandu. The country is world-famous for being home to Mount Everest, the highest peak on Earth, standing at 8,848.86 meters. The official language is Nepali, and the currency used is the Nepalese Rupee (NPR). Nepal is culturally rich, with Hinduism and Buddhism as the major religions practiced by its people. In 2008, Nepal became a federal democratic republic, ending centuries of monarchy. The country is known for its breathtaking natural landscapes, ranging from the snow-capped Himalayas to the green hills and fertile plains of the Terai. Agriculture remains the main source of income for most citizens, while tourism also plays a vital role in the national economy, attracting visitors for trekking and mountaineering. Interestingly, Nepal's national flag is unique, as it is the only non-rectangular national flag in the world. • Office of the Municipal Executive, Chandragiri, Kathmandu, Bagmati Province, Nepal (2079 BS)

Figure 5.1: Use of General Extraction Method



What is the fee for disability services?
6:17:46 PM

Based on the information in my knowledge base: **The service is FREE (नि:शुल्क)**. There are no charges for this service.

Found 1 relevant document(s) in knowledge base

6:17:48 PM

"अपाङ्गता सम्बन्धी सेवाको शुल्क कति हो?"
6:18:09 PM

I don't have specific information about '"अपाङ्गता सम्बन्धी सेवाको शुल्क कति हो?"' in my knowledge base. However, I can help you with questions about the topics I do have information about. Try asking about machine learning, Python programming, vector databases, Nepal, or government services.

Figure 5.2: Problem of Embedding model being trained on English text

6. Future Plan

- Integrate multilingual embeddings (e.g., `paraphrase-multilingual-MiniLM-L12-v2`) for improved Nepali query understanding.
- Add feedback system and query caching for performance optimization.
- Develop an admin console for document management and analytics.
- Expand document ingestion with OCR, images, and tabular data support.
- Introduce bilingual (Nepali/English) and voice-based user interfaces.
- Integrate with government APIs for real-time service information.
- Build a knowledge graph to connect entities (services, offices, documents, fees).

7. Discussion

7.1 System Performance Analysis

The implementation of the RAG-based chatbot, “Sarkari Sathi,” demonstrates the practical potential of combining retrieval-augmented generation techniques with government service delivery. The system integrates modern NLP models, vector-based retrieval, and a user-friendly web interface to improve public access to government information.

7.2 Retrieval Performance

Using Pinecone and the `all-MiniLM-L6-v2` embedding model, the system achieves efficient and accurate document retrieval with sub-second response times. The cosine similarity threshold of 0.3 provides a balanced trade-off between precision and recall, retrieving relevant government information for most user queries. However, performance decreases with complex or multilingual inputs. While the retrieval mechanism shows strong semantic understanding and scalability, it faces challenges with ambiguous or multi-part queries and limited multilingual comprehension.

7.3 Response Generation Quality

The rule-based response generator effectively produces structured and readable answers with clear formatting and transparency. It can extract specific details like fees, required documents, and service steps from retrieved content. However, its lack of deep contextual reasoning limits its ability to handle nuanced queries or maintain multi-turn conversations. Regular rule updates are also required to ensure accuracy across diverse government domains.

7.4 Multilingual Support Analysis

Nepali language support is partially implemented through script detection, keyword mapping, and basic question pattern recognition. While these features allow users to interact in their native language, the underlying embedding model’s English bias restricts semantic accuracy for Nepali queries. Mixed-language inputs, such as “ location,” often result in poor retrieval performance, highlighting the need for a multilingual embedding model and better translation handling.

7.5 Technical Architecture Evaluation

The system’s modular FastAPI-based architecture ensures scalability, allowing horizontal expansion and handling of large document collections. Its stateless design and vector database integration make it efficient and cost-effective. However, the lack of caching, advanced error handling, user session management, and robust logging features limits its readiness for production-level deployment. Security remains basic, with no authentication or encryption for sensitive interactions.

7.6 User Experience Analysis

The web interface provides a simple, responsive, and culturally adapted user experience. Users can easily interact with the chatbot, view message history, and navigate across devices. However, response accuracy varies depending on the complexity of the query. Simple factual questions are answered effectively, while complex or Nepali queries often yield incomplete or irrelevant responses.

7.7 Integration Challenges

The system processes a wide range of government document formats, but inconsistencies in document structure and the absence of image or handwritten text processing pose challenges. Additionally, the chatbot currently lacks integration with real-time government databases and service tracking systems, which limits its functionality to an informational tool rather than a full service assistant.

7.8 Comparison with Existing Solutions

Compared to traditional FAQ systems, the RAG-based chatbot provides greater flexibility, allowing users to ask open-ended questions without predefined templates. It operates cost-effectively without external API dependencies and can be easily expanded with new documents. However, unlike advanced AI assistants, it lacks conversational memory, contextual reasoning, and adaptive learning capabilities.

7.9 Future Improvements

Future work should focus on integrating multilingual embedding models, implementing caching mechanisms, and improving contextual understanding through local lightweight LLMs. Additional features such as voice input, personalized service recommendations, and integration with live government APIs could significantly enhance user engagement and system reliability.

7.10 Implications for Digital Government

"Sarkari Sathi" showcases how AI can make government services more accessible and inclusive. By providing 24/7 access through a natural language interface, the chatbot reduces information barriers for citizens. However, achieving reliability and trustworthiness for official use will require robust privacy protection, data governance, and accuracy assurance mechanisms.

8. Conclusion

This progress report presented **Sarkari Sathi**, a Retrieval-Augmented Generation (RAG) chatbot that enables citizens to access government information efficiently. The system combines a local embedding model (`all-MiniLM-L6-v2`) with the Pinecone vector database to support fast, semantic retrieval across diverse documents, including tabular citizen charters and user-added files. A FastAPI backend and a clean web interface provide a practical user experience, while rule-based response logic extracts concrete answers such as required documents, fees, timelines, and procedures.

The project has demonstrated: (i) effective ingestion and preprocessing of heterogeneous sources (.docx, structured text), (ii) robust vector indexing and retrieval, and (iii) end-to-end deployment with a responsive UI branded as *Sarkari Sathi*. Key challenges included handling multilingual/structured content, evolving APIs, and free-tier constraints; these were mitigated through improved chunking, threshold tuning, and API refactoring.

Looking ahead, the system will be extended with OCR for scanned PDFs, more municipalities and domains, admin tooling for easy content updates, hybrid retrieval for improved recall, and upgraded answer generation (multilingual LLM integration) for more natural, context-rich responses. Overall, *Sarkari Sathi* establishes a solid, scalable foundation for citizen-centric access to government services and information.