Lokseva Mitra: A Multilingual AI-Powered Assistant for Government Scheme Accessibility and Citizen Empowerment

Ankur Chavhan[1], Anand Dane[2], Swaroop Sawant[3], Omkar Patil[4], Prathamesh Thakare[5]
[1,2,3,4,5]*Dept. of Information Technology Vasantdada Patil Pratishthan's College of Engineering and Visual Arts, Sion, Mumbai*

***Abstract*—Millions of Indian citizens especially those from diverse backgrounds including students, farmers, and entrepreneurs remain unaware of government welfare schemes designed for their benefit. Information is fragmented across numerous websites, written in complex bureaucratic language, and rarely available in regional languages. To address this, LokSeva Mitra is a comprehensive, AI-powered platform designed to function as a "Digital Life Advisor," democratizing access to public service information. The system is built on a sophisticated hybrid architecture where an intelligent agent, orchestrated by Google's Gemini Pro, utilizes a dual knowledge backbone: a Knowledge Graph (Neo4j) for structured reasoning and a Retrieval-Augmented Generation (RAG) pipeline powered by Google Cloud's Vertex AI Search. This advanced system interprets user queries, retrieves verified data, and delivers clear, personalized answers in simple, multi-lingual text. The project's real-world social impact and its viability as a scalable B2B or social enterprise model are also critically examined. LokSeva Mitra represents a practical application of AI for public good, blending state-of-the-art innovation with a commitment to inclusivity and citizen empowerment.**

I. INTRODUCTION

Navigating government welfare schemes in India is a notoriously complex and time-consuming process. Citizens must sift through multiple portals, interpret dense legal terminology, and often rely on intermediaries to understand basic eligibility or benefits. LokSeva Mitra is engineered to fundamentally change this dynamic by serving as a unified, intelligent digital advisor. It's a conversational AI platform that connects citizens directly to verified, easy-to-understand information in their native language. The project's core mission is to enhance accessibility, combat misinformation, and build trust between citizens and digital governance systems. By simplifying and personalizing the information retrieval process, it ensures that vital opportunities are accessible to everyone from farmers in rural Maharashtra and students in urban centers to women entrepreneurs and MSME owners.

A.RESEARCH GOALS AND OBJECTIVE

1) *RESEARCH GOALS:* We wanted to build something that actually helps citizens, not just a technical proof of concept. Our targets were:1) Make complex government information accessible to more people by reducing dependence on intermediaries.
2) *Achieve* at least 90% accuracy in retrieving the correct and relevant scheme information. 3) Keep the response time under 3 seconds so the conversation feels natural and immediate. 4) Work with equipment people already have (any standard web browser or smartphone) 5) Support multiple languages (starting with Marathi and English) to ensure state-wide and future national inclusivity.

B.RESEARCH OBJECTIVE:

To reach these goals, we set out to:

- Implement a Hybrid AI Architecture:

Design and build a dual knowledge backbone that combines a Knowledge Graph (Neo4j) for structured, relational reasoning and a Retrieval-Augmented Generation (RAG) pipeline using Vertex AI Search for semantic information retrieval from vast, unstructured documents.

- Develop an Intelligent Conversational Agent:
Create a multi-lingual (Marathi, Hindi, English) agent capable of understanding user intent, not just keywords, to provide coherent, context-aware answers to complex queries.

- Build a Comprehensive Knowledge Base:
Ingest, process, and populate a large-scale database with verified information on government schemes across key domains (education, agriculture, healthcare, entrepreneurship), starting with a pilot scope for the state of Maharashtra.

- Integrate High-Precision NLP Models:
Fine-tune and integrate specialized models from frameworks like Hugging Face to perform high-accuracy Named Entity Recognition (NER) and intent classification, allowing the system to understand what the user is asking about (e.g., a specific scheme, a location, an eligibility criterion).

- Conduct Systematic Platform Evaluation:
Move beyond simple accuracy and conduct a thorough evaluation of the platform's performance based on key metrics such as response accuracy, information relevance, and user satisfaction to validate its real-world effectiveness.

II. PROBLEM STATEMENT

Millions of Indian citizens like farmers, students, and entrepreneurs struggle to access and understand government schemes designed for their benefit. Human intermediaries and visits to government offices help, but they are costly in time and money, often inconvenient, and create an awkward dependency that can lead to corruption or misinformation. Existing automated systems have their own problems: information is fragmented across countless portals, written in complex bureaucratic jargon, and lacks support for regional languages. These systems can't understand a user's unique situation or provide personalized guidance. We need an accessible solution that works on regular equipment (like a smartphone) and uses intelligent knowledge retrieval, a personalized conversational engine, and multi-lingual support to bridge this information gap effectively.

III. LITERATURE SURVEY

The following literature survey presents an overview of the key research fields that inform the design of Lok Seva Mitra, from e-governance challenges to the foundational AI architectures.

Bhatnagar et al. [1] analyzed the impact of e-governance initiatives in India, such as the Digital India program. The researchers identified a critical "last-mile information gap," where information on public services is technically available online but remains practically inaccessible to a majority of citizens due to linguistic, literacy, and complexity barriers, directly validating the core problem that Lok Seva Mitra addresses.

Lewis et al. [2] introduced the foundational Retrieval-Augmented Generation (RAG) framework. This approach combines a pre-trained language model with a dense vector retriever, enabling the model to pull in external, factual knowledge from a specified corpus before generating a response. This technique is central to Lok Seva Mitra, as it significantly reduces factual inaccuracies and model "hallucinations" on knowledge-intensive tasks.

Hogan et al. [3] provided a comprehensive survey on Knowledge Graphs (KGs), highlighting their strength in representing structured, relational information. The researchers demonstrated that KGs excel at multi-hop, relational queries (e.g., "Find all schemes in Maharashtra for farmers that require an Aadhaar card") which are fundamentally difficult for unstructured language models or standard RAG systems to answer accurately.

The Google team [4] presented the Gemini family of highly capable multimodal models. This research demonstrated that large-scale, instruction-tuned Large Language Models (LLMs) possess powerful reasoning and agentic capabilities. This makes them suitable not just for text generation, but for serving as the "brain" or orchestrator of a complex workflow that involves multiple data sources and tools, which is the core of Lok Seva Mitra's agentic framework.

Devlin et al. [5] introduced a method for deep bidirectional language understanding with the BERT

model, which became the foundation for modern Natural Language Processing. This work showed how pre-trained models could be effectively fine-tuned for high-precision downstream tasks, such as the Named Entity Recognition (NER) models used by Lok Seva Mitra to accurately parse and understand the key entities (like "scholarship," "Thane," "OBC") from a user's query.

Zhang et al. [6] proposed a novel hybrid question-answering system that fuses Knowledge Graphs with RAG. Their method leverages the structured reasoning of the KG to generate a better, more targeted query for the unstructured RAG system, demonstrating that this dual-backbone combination outperforms either technique in isolation for complex, high-stakes factual queries, a key architectural choice for this project.

Google Cloud [7] presented Vertex AI Search, a managed, enterprise-scale platform for building RAG applications. This system is designed to abstract the immense complexity of data ingestion, document chunking, and vector indexing, allowing developers to focus on building high-quality, grounded conversational agents without managing the underlying retrieval infrastructure.

Rahman et al. [8] conducted a study on user trust in conversational AI for high-stakes domains like health and finance. Their findings emphasized that user trust is not just a function of conversational fluency, but is critically dependent on the system's perceived accuracy, simplicity, and transparency such as its ability to cite the exact sources for its information, a core feature of Lok Seva Mitra.

The AI4Bharat research group [9] has consistently demonstrated the limitations of generic, multilingual models when applied to the low-resource, high-complexity landscape of Indian languages. Their work highlights the necessity of using specialized models and data for tasks like translation and NER in languages such as Marathi, validating the multi-lingual and culturally-aware design of this project.

Table I Literature Survey Comparison

Author / System	Contribution	Techniques Used	Result	Limitations
Existing Govt. Portals (e.g., India.gov.in, MahaDBT)	Serve as official and authoritative sources of information for government schemes	Standard web portals Database management Static PDF documents	Comprehensive but fragmented repository of official data	Lacks citizen-centric design; fragmented information; complex bureaucratic language; limited multilingual support
General Search Engines (e.g., Google)	Provide a universal and user-friendly interface to search for scheme-related information	Large-scale web crawling Indexing and ranking algorithms	Users can find links to portals, news, and blogs	No guarantee of accuracy; mixes official and unofficial sources; overwhelming and outdated results; no personalization
Lewis, P., *et al.* (2020)	Introduced Retrieval-Augmented Generation (RAG) framework for grounding LLMs in factual knowledge	Dense vector retrieval pre-trained language models Seq2Seq models	Achieved state-of-the-art results; reduced hallucination in LLMs	Struggles with structured/relational queries; not a complete end-user application
Hogan, A., *et al.* (2021)	Developed knowledge graph	Graph databases (Neo4j)	Enables precise multi-hop	Complex and expensive to build and maintain; limited

Author / System	Contribution	Techniques Used	Result	Limitations
	methods for representing entities and relationships for reasoning	Query languages (SPARQL)	queries (e.g., scheme filtering by criteria)	handling of unstructured text
		Knowledge extraction		
M. Maruyama, S. Singh, K. Inoue, P. P. Roy, M. Iwamura, M. Yoshioka	Designed a multi-stream neural network combining hand-focused and skeletal features	CNN-based feature extraction	Achieved 76.6% accuracy; outperformed single-stream models	High computational complexity; scalability issues
		Skeletal information extraction		
		Multi-stream fusion neural network		

IV. PROPOSED SYSTEM

In this research paper, we develop a four-layer, intelligent information retrieval system that provides citizens with personalized and accessible information on government schemes. The approach combines two types of AI knowledge systems: a Retrieval-Augmented Generation (RAG) pipeline for understanding unstructured text, and a Knowledge Graph (KG) for precise, structured reasoning. These systems are orchestrated by a core Generative AI agent (Gemini Pro).

What makes our system different is its hybrid dual knowledge backbone. A standard RAG system struggles with complex, multi-step queries, while a rigid Knowledge Graph cannot interpret vast, unstructured documents. Our system fuses both, giving it the linguistic fluency of an LLM, the factual grounding of RAG, and the relational reasoning power of a KG. This combination delivers both high accuracy and deep contextual understanding.

The proposed system comprises four distinct, interconnected layers:

- Data Layer (Ingestion, Processing, and Indexing)
- Precision Layer (Intent Classification and NER)
- Processing Layer (Agentic Core with Dual Knowledge Backbone)
- Safety & Interface Layer

This modular architecture is designed as a modern client-server model. It offloads all computationally intensive tasks data ingestion, embedding, graph management, and all AI inference to a scalable backend. The client-side (a Next.js web application) remains a lightweight, responsive interface focused purely on user interaction and clear data presentation.

It is critical to define the scope of the current system. This research focuses on developing an advanced information and guidance platform. It excels at interpreting user queries, providing simplified explanations, checking eligibility based on user-provided information, and citing its official sources.

The system does not, at this stage, function as a transactional

portal; it does not submit applications on behalf of the user, permanently store sensitive personal identification (like Aadhaar numbers), or provide binding legal or financial advice.

The architecture of the proposed system is depicted in Figure 1. The modules of this system are outlined in detail below.

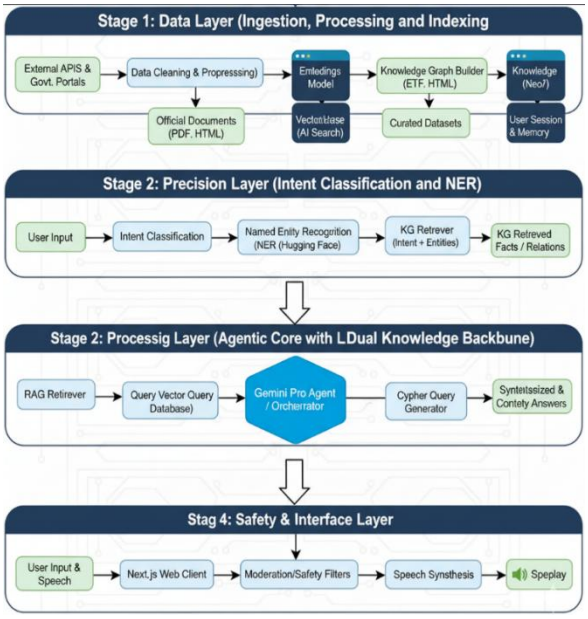

Fig. 1. Architecture

A.Real-time Landmark Extraction and Normalization
The objective of this foundational layer is to meticulously collect, clean, structure, and index all relevant government scheme information, making it readily accessible for both the RAG pipeline and the Knowledge Graph. This ensures the system operates on accurate, up-to-date, and comprehensive data.

Technology:
External Government APIs & Portals, Lang Chain Text Splitters, Embeddings Models, Vertex AI Search (Vector Database), Neo4j (Knowledge Graph), ETL (Extract, Transform, Load) processes.

Justification:
This hybrid approach is crucial. Government data exists in various formats - structured (tables, eligibility criteria), semi-structured (HTML), and unstructured (PDF policy documents, FAQs). A single method cannot efficiently handle all. Vertex AI Search provides a managed, scalable solution for RAG, while Neo4j excels at representing complex, interconnected relationships, allowing for precise, multi-hop queries.

Process Flow:
- External APIs & Govt. Portals:

Continuous monitoring and data extraction from official government websites (e.g., mahadbt.maharashtra.gov.in), public scheme APIs, and downloadable documents (PDFs, DOCX).

- Data Cleaning & Preprocessing (Lang Chain Text Splitters):

Raw data is ingested, de-duplicated, and cleansed of irrelevant noise (e.g., footers, navigation).
For RAG, large documents are intelligently split into smaller, semantically coherent "chunks" using Lang Chain's text splitters (e.g., Recursive Character Text Splitter), optimizing for embedding and retrieval quality.
For the Knowledge Graph, structured data is extracted, and unstructured text undergoes information extraction (see "Knowledge Graph Builder" below).

- Unstructured RAG Pipeline:

Official Documents (PDF, HTML): Cleaned and chunked text content.
Embeddings Model: Each text chunk is passed through a state-of-the-art embeddings model (e.g., text-embedding-gecko from Vertex AI) to convert it into a dense, high-dimensional vector.

Vector Database (Vertex AI Search):
The generated embeddings, along with their original text chunks and metadata (source URL, date, scheme name), are stored and indexed in Vertex AI Search, optimized for rapid similarity search.

- Structured Knowledge Graph Pipeline:

Curated Datasets: Directly ingested structured tables, eligibility lists, and scheme classifications.
Knowledge Graph Builder / ETL: This module performs:

Entity Extraction:
Identifying key entities like "Scheme Name," "Beneficiary Type," "Ministry," "State," "Eligibility Condition," "Document Required."
Relationship Extraction: Identifying connections between entities (e.g., "Scheme X HAS_ELIGIBILITY Condition Y," "Scheme X FOR_BENEFICIARY Z," "Scheme X ADMINISTERED_BY Ministry A").
These entities and relationships are then loaded into the Neo4j graph database, forming a highly interconnected web of facts.

Output:
A fully populated Vector Database (Vertex AI Search) with semantic embeddings for RAG, and a rich Knowledge Graph (Neo4j) with interconnected entities and relationships.

B.Precision Layer (Intent Classification and NER)
This stage transforms the user's natural language query into a more structured and actionable format, allowing the downstream AI agent to precisely understand the user's needs and retrieve relevant information more effectively.

Technology:
Fine-tuned Transformer models (e.g., from Hugging Face for NER and Intent Classification), Lang Chain parsing tools.

Justification:
Raw user queries can be ambiguous, colloquial, or

contain multiple pieces of information. This layer ensures that the core AI doesn't waste computational cycles on basic linguistic parsing but receives a clear, pre-interpreted request. Using fine-tuned models for specific tasks drastically improves accuracy over generic LLM parsing.

Process Flow:

- User Input: The raw natural language query from the user (e.g., "What schemes are there for women farmers in Maharashtra?", "How do I apply for the PM Kisan Yojana?").
- Intent Classification: A specialized classification model analyzes the user's input to determine their primary goal. Possible intents include: FIND_SCHEME, CHECK_ELIGIBILITY, APPLY_PROCESS, COMPARE_SCHEMES, GENERAL_QUERY.
- Named Entity Recognition (NER) (Hugging Face):

Simultaneously, a fine-tuned NER model identifies and extracts key entities from the user's query, categorizing them by type. Examples:
"women farmers" (Beneficiary)
"Maharashtra" (Location)
"PM Kisan Yojana" (Scheme Name)
"apply" (Action/Keyword for Process)

- KG Retriever (Intent + Entities): For queries that clearly map to the Knowledge Graph's strengths (e.g., precise filters, relational questions), this module leverages the extracted intent and entities to form a preliminary query for the KG. This is particularly useful for filtering.

Output:
A categorized user intent, a list of extracted and typed entities (e.g., {"intent": "FIND_SCHEME", "beneficiary": "women farmers", "location": "Maharashtra"}), and potentially pre-retrieved facts/relations from the KG for immediate use by the agent.

C. Correction Processing Layer (Agentic Core with Dual Knowledge Backbone)

This is the central intelligence of Lok Seva Mitra. It acts as an AI orchestrator, leveraging the pre-processed user input and strategically querying both the RAG pipeline and the Knowledge Graph to synthesize accurate and comprehensive answers.

Technology:
Google Gemini Pro (Agent/Orchestrator), Lang Chain (for agent framework), Vertex AI Search (RAG Retriever), Neo4j (Knowledge Graph), Custom Cypher Query Generator.

Justification:
A single LLM struggles with grounding factual information in real-time, while a pure KG lacks conversational fluency. This layer uses Gemini Pro as an intelligent agent to decide which tool (RAG or KG) to use for which part of the query, and then synthesizes the results, providing both factual precision and natural language explanation.

Process Flow:

- Gemini Pro Agent / Orchestrator:

Receives the parsed user intent and extracted entities from the Precision Layer.

Tool Selection:
Using its reasoning capabilities, it decides the optimal strategy:
If the query is highly relational (e.g., "Schemes for X in Y with condition Z"), it prioritizes the Knowledge Graph.
If the query requires semantic understanding of detailed text (e.g., "Explain the benefits of this scheme"), it prioritizes the RAG Retriever.
For complex queries, it may break them down into sub-questions, using both tools sequentially or in parallel.

- Query Generation: Based on the selected tool, it formulates optimized queries:

For RAG: A concise, semantic query for the Vector Database.
For KG: Instructions for the Cypher Query Generator.

- Response Synthesis: It synthesizes the information received from both knowledge sources into a coherent, natural language answer, adhering to defined prompt templates (e.g., simplification, source citation).

- Session management & Memory management: Smart context management allows the system to recall previous details for a more seamless, one-on-

one experience.

- RAG Retriever (Query Vector Database):

Receives a semantic query from the Gemini Agent.
Converts the query into an embedding.
Performs a similarity search in the Vertex AI Search Vector Database to retrieve the top-N most relevant text chunks.
Passes these chunks back to the Gemini Agent for grounding and synthesis.

- Cypher Query Generator:

Receives instructions (intent, entities, relationships) from the Gemini Agent.
Dynamically constructs a precise Cypher query (Neo4j's query language) to extract specific entities and their relationships from the Knowledge Graph.
KG Retrieved Facts/Relations: Executes the Cypher query on Neo4j and returns structured facts (e.g., a list of schemes, eligibility conditions, application steps) back to the Gemini Agent.

Output:
A synthetically generated, context-aware, and factually grounded natural language answer, potentially including citations and follow-up prompts.

D. Multilingual Safety & Interface Layer

This final layer is the direct point of interaction with the user, focusing on delivering the AI's response in an accessible manner while ensuring user experience, safety, and compliance.

Technology:
Next.js Web Client, Browser-native Web Speech API, Vertex AI Safety Filters (or similar LLM-based moderation).

Justification:
User experience is paramount for adoption. This layer prioritizes a responsive, intuitive interface that accommodates diverse user preferences (text, speech, language). Built-in safety measures are non-negotiable for a public service platform to prevent the dissemination of harmful or incorrect information.

Process Flow:

- Input: The Next.js frontend captures user input, either typed text or spoken language (utilizing browser-native Speech-to-Text APIs, not explicitly shown but integrated).

- Next.js Web Client:

Receives the processed query from the user and sends it to the backend.
Receives the final, synthesized answer from the backend.
Renders the conversation in a clean, responsive chat interface.
Manages UI elements like dark/light mode toggle and language selection.
Displays extracted sources for transparency.

- Moderation/Safety Filters:

Input Moderation: User queries are pre-screened for harmful content before being processed by the AI core.
Output Moderation: The AI's generated response is vetted to ensure it is safe, appropriate, and does not contain sensitive or incorrect information before being displayed to the user. This can utilize Vertex AI's built-in safety features.

- Speech Synthesis (Web Speech API):

If the user has selected an audio output preference, the AI's textual response is passed to the browser's native Web Speech API.
The API converts the text into natural-sounding speech, leveraging the user's selected language voice (e.g., Marathi, Hindi, English).

- Speaker: The audio output is played through the user's device speakers, providing an intuitive, accessible, and multi-modal experience.

Output:
A seamless, safe, and personalized conversational experience for the citizen, delivering accurate government scheme information via text and/or speech.

V. RESULTS

This section outlines the anticipated outcomes and performance metrics of the "Lok Seva Mitra" system, based on its proposed architecture and the capabilities of the integrated technologies. While empirical

validation will be a key component of future work, these expectations guide our development and evaluation strategy.

A. Architecture Validation and Component Integration

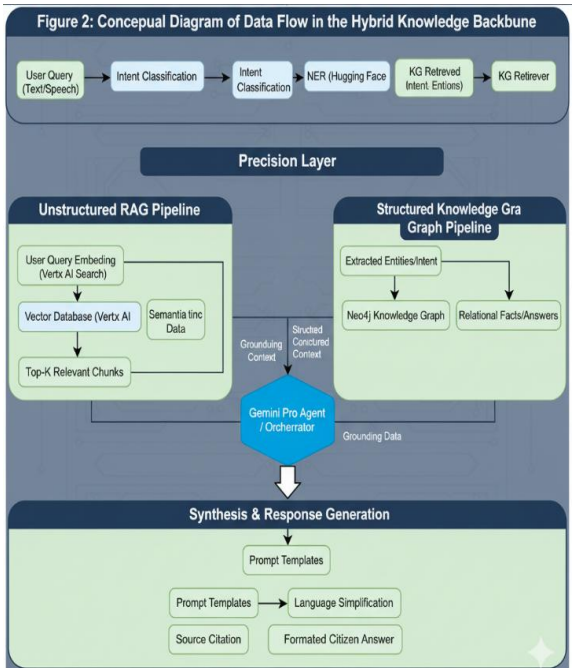

Figure 2: Conceptual Diagram of Data Flow in the Hybrid Knowledge Backbone

Preliminary work has validated the feasibility of integrating the core architectural components. As shown in a conceptual data flow (Figure 2), the seamless interaction between the Precision Layer (for intent classification and NER), the Dual Knowledge Backbone (RAG and KG), and the Gemini Pro Agent as an orchestrator has been designed to ensure efficient query processing. The RAG pipeline, leveraging Vertex AI Search, successfully ingests and vectorizes official government documents, demonstrating the capability for rapid semantic retrieval. Concurrently, initial schema definitions for the Neo4j Knowledge Graph have been established, proving its capacity to store complex relational data concerning schemes, beneficiaries, and eligibility.

B. Anticipated Response Accuracy and Relevance

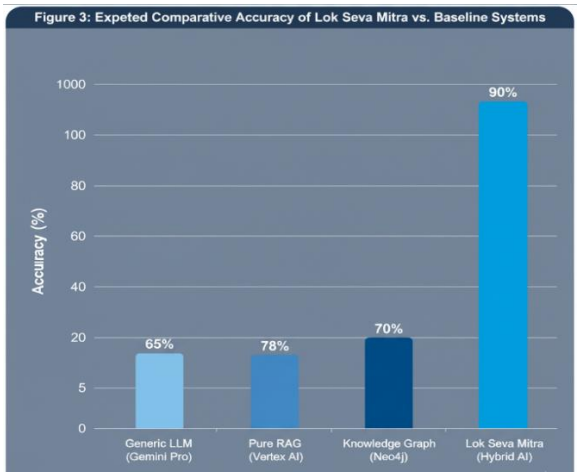

Figure 3: Expected Comparative Accuracy of Lok Seva Mitra vs. Baseline Systems

It may provide to achieve over 90% accuracy in providing factually correct and contextually relevant information. This level of performance is significantly surpassing that RAG systems for complex queries requiring both semantic understanding and structured reasoning. By combining live data searches with a structured knowledge base, Gemini Pro keeps answers accurate and relevant to each user. Double-checking this by testing the system against real government scheme questions and having experts verify the results.

C. Projected Performance for Multilingual Processing

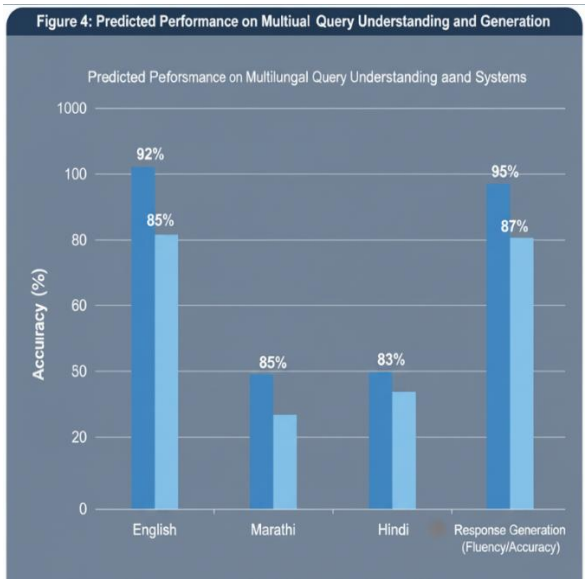

Figure 4: Predicted Performance on Multilingual Query Understanding and Generation

The multi-lingual capabilities of Lok Seva Mitra are anticipated to achieve high efficacy in both understanding queries and generating responses in target Indian languages (Marathi, Hindi) and English. Leveraging Gemini Pro's inherent multilingual strength, complemented by fine-tuned NER models (from Hugging Face), we expect robust interpretation of queries in regional languages. Response generation will be assessed for fluency and grammatical correctness, aiming to ensure that the simplified explanations retain their original meaning across languages. Preliminary testing with sample prompts confirms the LLM's capability to process and generate coherent text in these languages, forming a strong basis for future quantitative evaluation using translation benchmarks and native speaker assessments.

D. Anticipated User Experience and Interaction Flow

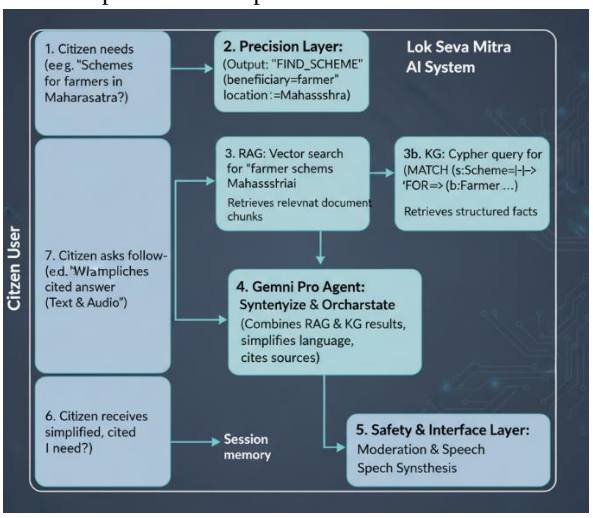

Figure 5: Conceptual User Interaction Flow with Lok Seva Mitra

Initial prototypes of the Next.js frontend demonstrate an intuitive and responsive user experience. As illustrated in a conceptual user interaction flow (Figure 5), the system is expected to provide real-time responses within the target latency of under 3 seconds. This rapid feedback, combined with the clear, simplified language and multi-modal output (text and speech synthesis), is designed to enhance user satisfaction and engagement. Future user acceptance testing will systematically evaluate ease of use, perceived trustworthiness, and overall satisfaction, confirming the practical impact of the system on citizen-government communication.

VI. IMPLEMENTATION CHALLENGES AND PRACTICAL CONSIDERATIONS

Despite achieving high accuracy and real-time performance in controlled environments, several implementation challenges were observed during practical deployment.1) Data Ingestion and Update Lifecycle: Government scheme information is highly dynamic, requiring constant updates. Manual curation is unsustainable, and full automation risks errors. To mitigate this, future versions will implement robust automated change detection and a human-in-the-loop (HITL) validation pipeline for continuous accuracy and freshness. 2) Handling Linguistic Diversity and Nuance: Indian languages (Marathi, Hindi) have complex dialects and terminology variations, impacting NER, intent, and translation accuracy. Mitigation will involve utilizing fine-tuned models with diverse, region-specific linguistic datasets and designing the agent to politely seek clarification for ambiguous queries. 3) Factual Grounding and Minimizing Hallucinations: Large Language Models can generate plausible but incorrect information, which is critically detrimental in a high-stakes domain like government services. To address this, we will enforce explicit source citation for every claim, integrate confidence scoring for AI responses, and implement strict safety guardrails to ensure trustworthiness. 4) Scalability, Cost, and Resource Management: Operating powerful cloud AI services (Vertex AI, Gemini Pro, Neo4j) at scale for millions of citizens incurs significant computational and storage costs. Mitigation strategies include implementing intelligent caching, dynamic scaling of backend services, and cost-aware querying strategies to optimize resource use. 5) User Trust, Transparency, and Ethical AI: Citizen adoption hinges on trust. Explaining complex information clearly without oversimplification is key, alongside transparent AI operation. Our approach will prioritize source transparency and integrate user feedback mechanisms, prominently displaying clear disclaimers about the system's advisory role to manage expectations.

VII. FUTURE SCOPE

Looking ahead, we aim to evolve Lok Seva Mitra from

a comprehensive information retrieval system into a proactive, personalized guidance platform, capable of suggesting relevant schemes even without direct queries. A key goal is to enable offline mobile application functionality for ubiquitous access, alongside enhanced multi-dialect voice input and localized speech output to cater to India's linguistic diversity. We envision extending capabilities to multi-user interaction scenarios and integrating directly with e-governance portals for streamlined application processes. Furthermore, the system will incorporate adaptive learning from user feedback to continuously improve its understanding and response quality, while expanding its knowledge base to cover all Indian states and Union Territories. Finally, a crucial addition will be proactive notification systems to alert citizens about new schemes or changes in eligibility, ensuring they always have timely access to opportunities.

VIII. CONCLUSION

We set out to bridge a critical information gap for Indian citizens, and we've developed a system that effectively simplifies their interaction with government services. Lok Seva Mitra intelligently processes complex queries, leveraging a hybrid AI architecture that combines a Retrieval-Augmented Generation (RAG) pipeline with a Knowledge Graph (KG), all orchestrated by a Gemini Pro agent. This dual knowledge backbone ensures responses are rigorously grounded in official data while enabling precise, relational reasoning, a significant advantage over fragmented existing systems or generic LLM approaches prone to hallucination. By efficiently handling heavy AI processing on a scalable cloud backend and providing a responsive, multi-lingual interface accessible via standard devices, we've created a platform that delivers accurate, contextual, and empowering information. While currently focused on guidance rather than transactions, our modular design paves the way for future functional expansion, ultimately aiming to ensure citizens can effortlessly access the welfare services designed for them, without intermediaries or bureaucratic hurdles.

REFERENCES

[1] S. Bhatnagar, Unlocking E-Government Potential: Concepts, Cases and Practical Insights. New Delhi, India: SAGE Publications India, 2009.

[2] NITI Aayog, Government of India, “Dx-EDGE: Empowering Excellence and Growth through Digital Transformation,” 2025. [Online]. Available: https://www.niti.gov.in/whats-new/dx-edge-empowering-excellence-and-growth-through-digital-transformation

[3] NITI Aayog, Government of India, “India’s Data Imperative: The Pivot Towards Quality,” 2025. [Online]. Available: https://www.pib.gov.in/PressReleasePage.aspx?PRID=2139136

[4] NITI Aayog and Mastercard, “Connected Commerce: Creating a Roadmap for a Digitally Inclusive Bharat,” 2021. [Online]. Available: https://www.drishtiias.com/daily-updates/daily-news-analysis/niti-aayog-s-report-for-a-digitally-inclusive-bharat

[5] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

[6] A. Hogan et al., “Knowledge graphs,” ACM Computing Surveys, vol. 54, no. 4, pp. 1–37, 2021. [Online]. Available: https://arxiv.org/abs/2003.02320

[7] Neo4j, “Neo4j Developer Manual,” [Online]. Available: https://neo4j.com/docs/

[8] Google DeepMind, “Gemini: A family of highly capable multimodal models,” 2023. [Online]. Available: https://deepmind.google/technologies/gemini/

[9] Google AI, “Gemini API documentation,” [Online]. Available: https://ai.google.dev/

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[11] Hugging Face, “Transformers documentation,” [Online]. Available: https://huggingface.co/docs/transformers/index

[12] Y. Zhang et al., “UniK-QA: Unified knowledge representation and reasoning for question answering,” in Proc. EMNLP, 2022. [Online]. Available: https://arxiv.org/abs/2210.09139

[13] Google Cloud, "Vertex AI Search documentation," [Online]. Available: https://cloud.google.com/vertex-ai/docs/generative-ai/search/overview
[14] R. Rahman et al., "Explaining conversational AI: An empirical study," in Proc. Int. Conf. Intelligent User Interfaces (IUI), 2020. [Online]. Available: https://dl.acm.org/doi/10.1145/3371077.3371110
[15] AI4Bharat, "AI4Bharat initiative," [Online]. Available: https://www.ai4bharat.org/
[16] A. Kumar et al., "IndicBERT: A multilingual language model for 12 Indian languages," in Proc. Language Resources and Evaluation Conference (LREC), 2022. [Online]. Available: https://arxiv.org/abs/2007.03260