Multi-Lingual Legal Document Assistant

Dr. K. Baalaji, Pandalaneni Mani Prakash, Paloju Venu Gopal Sai, Parkepalli Aditya Sai Ram, Paramkusham Varun
Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research Chennai, India

Abstract— **Legal documents are difficult for many readers because contractual language, statutory references, exception-heavy clauses, and procedural terminology demand careful interpretation. This work presents a multilingual legal document assistant that transforms uploaded files into a document-grounded question-answering workspace. The system follows a retrieval-augmented generation pipeline in which legal files are parsed, normalized, segmented into semantically coherent chunks, embedded, and indexed in a vector database for evidence retrieval. At query time, the assistant retrieves the most relevant clauses, assembles a bounded context, and generates an explanation that stays tied to the uploaded material through citation-aware answer construction. The architecture also supports bilingual interaction and voice-first access through speech playback and phone-call integration, making the platform usable for readers who prefer Tamil or spoken responses over conventional text-heavy interfaces. The design combines a web dashboard, authentication layer, document service, RAG orchestrator, chat API, language adapter, and audio service with external components such as Pinecone, file storage, Gemini 2.5 Flash, ElevenLabs, and Twilio. The resulting framework improves clause discoverability, traceability, and accessibility while preserving a modular cloud deployment model suitable for future extension.**

Index Terms - **retrieval-augmented generation, legal document analysis, multilingual question answering, vector database, citation grounding, voice interface, telephony integration, cloud deployment**

I. INTRODUCTION

Legal documents shape commercial transactions, employment arrangements, tenancy, compliance obligations, and civil procedures, yet many readers encounter them only when an urgent decision or dispute appears. The difficulty is not limited to unfamiliar vocabulary. Legal meaning is often distributed across definitions, operative clauses, exceptions, annexures, and timelines that must be interpreted together rather than in isolation. As a result, ordinary users frequently depend on slow manual review or informal online explanations that may omit the exact language of the file they need to understand.

Traditional document search is a weak fit for this problem. Keyword matching can locate a word such as termination, indemnity, or arbitration, but it often fails when the same meaning is expressed through alternate wording, conditional phrasing, or cross-references between distant sections. A user may retrieve a sentence containing the right term while still missing the clause that controls liability, notice duration, or penalty conditions. In legal workflows, shallow retrieval therefore creates false confidence rather than true understanding.

Large language models make interaction easier because they can restate complex passages in plain language, yet unconstrained generation introduces a different risk. A fluent answer that is not anchored to the uploaded file may sound persuasive while still being incomplete, overgeneralized, or inconsistent with the document. In a legal setting, even a small unsupported inference can mislead the reader about duties, deadlines, or dispute pathways. The central design requirement is therefore not just readability but evidence-grounded explanation.

Retrieval-augmented generation offers a practical bridge between semantic search and controlled legal assistance. In a RAG pipeline, document segments are embedded into vector representations, retrieved through similarity search, and supplied to the generator as bounded evidence. This shifts answer formation away from generic memory and toward the passages that actually appear in the user workspace. For legal questions, the value of this design lies in clause-level relevance, citation traceability, and the ability to simplify language without detaching from the source material.

Accessibility is equally important. Many legal support interfaces are English-dominant and assume that the user is comfortable with long-form text interaction. The project considered in this paper addresses that

limitation by supporting bilingual explanation and voice-first use. A reader can upload a file, ask a question in plain language, request the answer in Tamil or English, and receive either text or speech output. The same reasoning pipeline can also be exposed through a phone-call interface for users who are more comfortable speaking than typing.

The uploaded architecture diagrams describe this idea at two levels. The first shows a layered flow from user interface to document processing, vector storage, retrieval, and language generation over secure cloud infrastructure. The second expands the design into concrete services such as a web dashboard, voice playback, phone-call interface, authentication, document handling, prompt construction, citation formatting, language adaptation, and audio delivery. Building from those artifacts, this paper formalizes the project as a cloud-native multilingual legal assistant for document-grounded explanation.

The remainder of the paper is organized as follows. Section II reviews relevant work on legal retrieval, legal summarization, RAG reliability, privacy, and multilingual conversational systems. Section III explains the proposed framework and maps the uploaded project architecture into implementation modules. Section IV discusses functional results and comparative strengths. Section V concludes the paper and outlines future enhancement directions.

II. RELATED WORK

Recent work in legal language technology has moved from static information retrieval toward retrieval-grounded generation. Systems such as LexDrafter and other domain-tuned legal assistants show that document-linked generation improves precision when the task requires drafting support, summarization, or terminology-sensitive interpretation. This shift is especially relevant for legal assistance because correctness depends not only on language fluency but also on whether the answer can be traced back to a supporting passage.

A second research direction focuses on retrieval quality itself. Legal RAG studies emphasize that relevance failures often come from chunk boundaries, ranking strategy, and context ordering rather than from the generator alone. Long legal files contain definitions, operative provisions, schedules, and exceptions that may be separated by many pages. Reliable systems therefore need clause-aware segmentation, metadata-rich indexing, and retrieval logic that surfaces the governing passage rather than merely the nearest lexical match.

Researchers have also investigated orchestration strategies that sit between retrieval and generation. Multi-stage pipelines and multi-agent filtering architectures improve answer quality by screening retrieved passages before they are handed to the model. In the legal domain, this matters because a single user question may require the system to identify the right clause, resolve a cross-reference, extract an obligation, and then translate the result into plain language. Naive single-step prompting often performs poorly on such layered reasoning tasks.

Safety and privacy studies add another important constraint. Legal documents routinely contain personal information, commercial terms, and dispute-sensitive material. Work on RAG safety shows that retrieval does not automatically eliminate harmful or misleading outputs if the context assembly process is weak. Similarly, privacy research demonstrates that poorly designed retrieval systems can expose more context than necessary. These findings support document-scoped workspaces, minimal-context prompting, retention controls, and explicit refusal behavior when the system lacks supporting evidence.

Multilingual and voice interfaces form the final foundation for the present project. Prior work on conversational legal support, multilingual information access, and low-latency speech systems indicates that legal assistance becomes more usable when explanation can move across languages and channels without losing grounding. In the present design, these ideas are combined with citation-aware generation so that accessibility improvements do not come at the cost of evidence control.

III. PROPOSED SYSTEM

The proposed system converts a user-uploaded legal file into an interactive evidence-grounded workspace. After authentication, the document is stored in cloud-backed file storage, parsed into machine-readable text, segmented into legal chunks, and indexed as embeddings for semantic retrieval. When the user submits a query, the system retrieves the most relevant chunks, assembles a bounded prompt, and generates an explanation that remains tied to the file through citation

formatting. This architecture is designed to reduce unsupported responses while still presenting the answer in plain language.

The layered block diagram attached to the project highlights four major planes: the user interface, the application layer, the data layer, and secure cloud infrastructure. Within that flow, document processing feeds a vector database, the RAG engine retrieves evidence, and the language model produces the final answer in Tamil or English. The more detailed service-level architecture complements this view by exposing the presentation layer, application modules, supporting microservices, and third-party integrations used in the prototype.

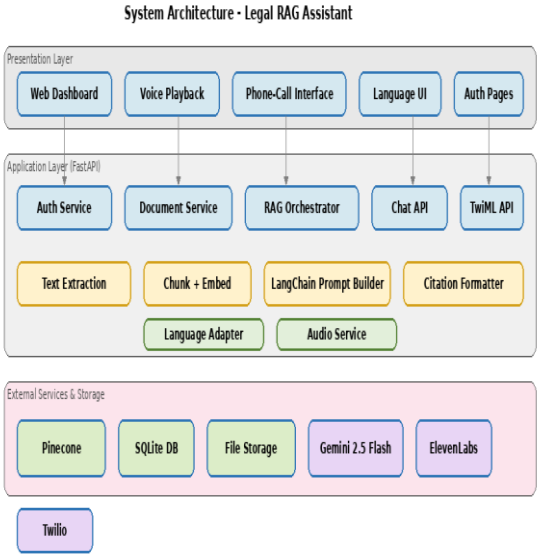

Figure 1. Layered architecture of the multilingual legal RAG assistant

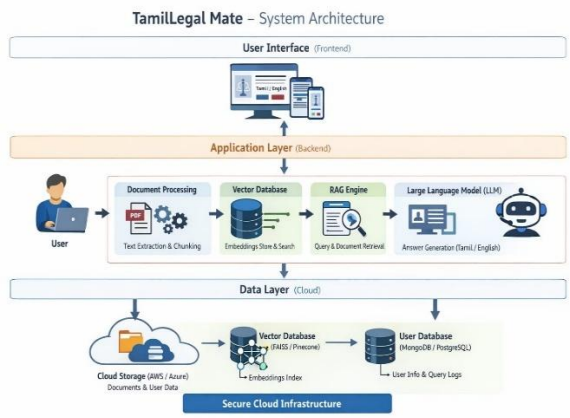

Fig. 2. Detailed service architecture with Pinecone, Gemini 2.5 Flash, ElevenLabs, SQLite DB, file storage, and Twilio integration

A. User Authentication and Document Intake

The interaction begins with authentication pages that create a scoped workspace for each user and document set. File validation ensures that supported uploads such as PDF or Word files are accepted, while metadata such as document identifier, upload time, and processing status are attached to the workspace. This scoping matters in legal settings because answer relevance and privacy both depend on keeping retrieval bounded to the intended document collection.

A web dashboard gives the user a single entry point for upload, indexing status, question entry, citation viewing, language selection, and response history. Instead of forcing the user to move between separate applications for storage, retrieval, translation, and speech playback, the design keeps the workflow inside one interface.

B. Parsing, Chunking, and Embedding Pipeline

The document service performs text extraction and normalization after upload. Depending on source quality, the parser can handle digitally generated files directly and fall back to OCR-style processing for low-quality scans. The cleaned text is then segmented into semantically coherent chunks with limited overlap so that clause continuity is preserved without making retrieval excessively broad.

Each chunk is embedded and written to the vector index together with page information, chunk identifiers, and source metadata. The uploaded high-level architecture suggests Pinecone or FAISS as the retrieval layer, while the detailed service view explicitly positions Pinecone as the main vector backend for the prototype. These embeddings create the searchable knowledge space that later supports similarity-based clause retrieval.

C. Retrieval and Context Assembly

When a user asks a question, the query is embedded and matched against the indexed chunks. The RAG orchestrator ranks results, filters them by workspace or document identifier, and assembles a context package for the generator. Context assembly is not treated as a trivial step: the order of retrieved chunks, the amount of evidence passed forward, and the diversity of returned passages all influence whether the answer will be reliable.

The design shown in the service-level diagram includes a prompt-building component and a citation formatter. This is important because legal assistance requires more than simply selecting nearby text. The system must preserve clause identifiers, page spans,

and passage order so that the final response can explain the issue in plain language while still pointing the user back to the exact supporting evidence.

D. Citation-Aware Answer Generation

The language model layer is responsible for answer generation, but it operates under explicit constraints. The prompt instructs the model to explain the retrieved clauses, highlight duties and limitations, avoid speculation, and acknowledge when the uploaded document does not support the requested claim. This bounded behavior is essential in legal contexts, where the absence of evidence must be surfaced rather than hidden behind fluent wording.

The project diagrams identify Gemini 2.5 Flash as the fast generation engine and a citation formatter as a companion module. Together, they support an answer pattern in which the explanation remains readable yet traceable. Instead of producing generic legal advice, the system behaves as a guided reading layer over the uploaded file.

E. Multilingual, Speech, and Phone-Call Delivery

A language adapter converts the answer into the user-selected language without changing the evidentiary basis. In the present project, Tamil and English are the main target languages. This makes the platform useful for readers who may understand the document better when it is restated in a familiar regional language rather than in dense legal English.

The same answer can be routed to voice playback through an audio service and text-to-speech layer. The detailed system architecture links this flow to ElevenLabs for speech synthesis and Twilio through a TwiML API for phone-call access. As a result, the assistant can function as both a browser-based legal explainer and a call-driven information interface, widening access for users who prefer speaking and listening over reading long screens of text.

F. Data Layer and Deployment Considerations

The attached diagrams show a secure cloud backbone beneath the application logic. Cloud storage holds uploaded documents and user data, the vector database stores embeddings, and a metadata store records workspace state, query logs, and operational status. The high-level architecture mentions MongoDB or PostgreSQL for user data, while the detailed prototype view includes SQLite DB for lightweight metadata management. This difference can be interpreted as a prototype-to-production path rather than a design conflict.

Because legal documents may contain sensitive information, the deployment model should preserve document-level isolation, encrypted storage, and minimal-context prompting. A modular service boundary also allows future replacement of the embedding model, the vector backend, or the speech stack without redesigning the full system. This makes the framework practical for iterative improvement as the project matures.

IV. RESULTS AND DISCUSSION

The present project was assessed primarily through functional behavior rather than through a large benchmark corpus. The most important questions were whether the assistant could retrieve clause-level evidence from uploaded documents, answer in a readable form, preserve source traceability, and support multilingual plus voice-first interaction. In those respects, the architecture provides a stronger first-pass document understanding experience than plain keyword search or unconstrained chat interaction.

A practical strength of the system is workflow compression. A user can upload a file, wait for indexing to finish, ask a question in natural language, inspect the cited passage, request the same answer in Tamil, and listen to it through speech playback or a phone-call channel. This sequence mirrors the project outputs that the final journal version can illustrate through screenshots of upload, retrieval, citation-linked answer generation, and bilingual response delivery.

The design also improves trust because it explicitly separates document-grounded explanation from unsupported generalization. When retrieval produces sufficient evidence, the assistant can explain a clause, summarize its effect, or convert it into a short checklist of obligations. When the evidence is insufficient, the same bounded design can request clarification or state that the document does not answer the question directly. This is preferable to producing a confident but weakly grounded narrative.

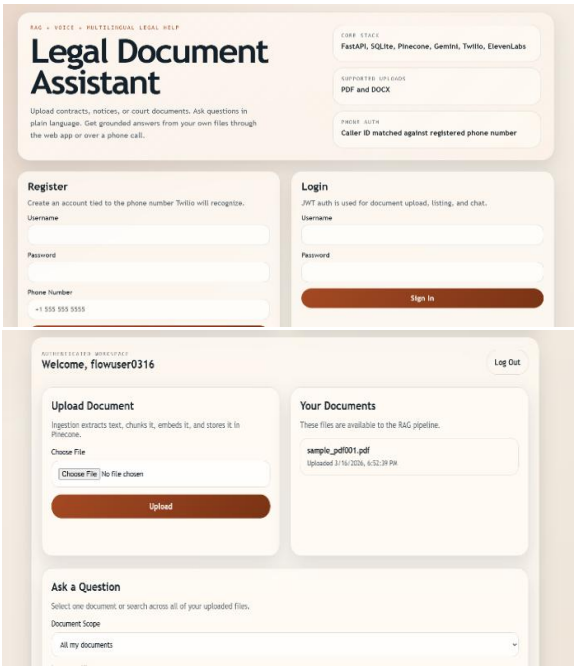

Fig. 3. Output screenshot for document upload and indexing

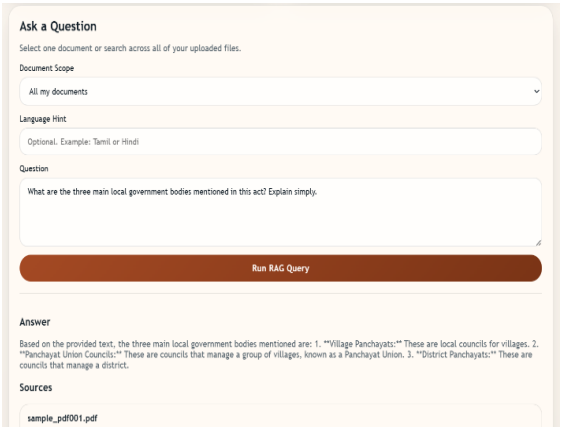

Fig. 4. Output screenshot for citation-grounded question answering

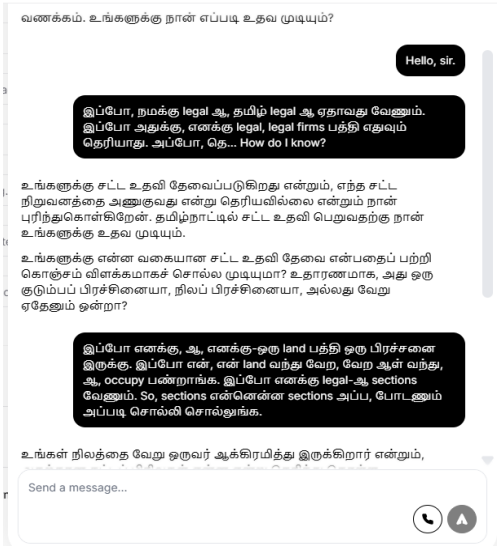

Fig. 5. Output screenshot for multilingual or voice-based response delivery

TABLE I: QUALITATIVE COMPARISON OF LEGAL ASSISTANCE APPROACHES

Method	Semantic Retrieval	Grounding Control	Citations	Multilingual	Voice / Phone	Suitability
Keyword-based legal search	Low	Moderate	Low	Low	None	Limited
Generic LLM without retrieval	Moderate	Low	None	Moderate	Optional	Risk-prone
Manual legal review	High	High	High	Reviewer dependent	Human dependent	Accurate but slow
Text chatbot with retrieval	Moderate to High	Moderate	Moderate	Limited	None	Useful but less accessible
Proposed multilingual legal RAG assistant	High	High	High	High	High	Comprehensive

The comparison indicates that the proposed assistant combines the strengths of semantic retrieval, evidence control, citation traceability, multilingual explanation, and voice interaction in a single pipeline. Keyword search remains useful for exact lookup, and manual review remains necessary when formal legal judgment is required, but neither alternative provides the same balance of accessibility and scalable first-pass interpretation.

The screenshot placeholders prepared for the final journal version highlight document upload and indexing, citation-grounded answer output, and multilingual or speech-based response delivery.

V. CONCLUSION

This paper presented a multilingual legal document assistant that combines document parsing, vector indexing, retrieval-augmented generation, citation-aware answering, and speech-enabled delivery inside a cloud-ready architecture. By grounding responses in uploaded files, the framework reduces the risk associated with unsupported explanation and turns static legal documents into an interactive workspace for clause discovery and plain-language understanding.

The system is particularly valuable because it treats accessibility as a first-class design goal rather than as an optional interface layer. Bilingual response adaptation, voice playback, and phone-call access extend legal explanation to users who may not benefit from conventional English-only text interfaces. At the

same time, the use of scoped workspaces, metadata-rich retrieval, and citation formatting helps preserve traceability and user trust.

Future work includes better clause-level ranking, dialect-aware speech, finer citation granularity, and document-type specialization for contracts and notices.

REFERENCES

[1] A. Chouhan and M. Gertz, "LexDrafter: terminology drafting for legislative documents using retrieval augmented generation," arXiv, 2024.

[2] M. Dhore, A. Vimal, A. Agrawal, R. Bajaj, and R. Barde, "BetterCall: AI-based legal assistant," in Proc. 5th Int. Conf. Image Processing and Capsule Networks, 2024, pp. 248-256.

[3] M. S. Rahman et al., "Optimizing legal text summarization through dynamic retrieval-augmented generation and domain-specific adaptation," Symmetry, vol. 17, no. 5, p. 633, 2025.

[4] M. Reuter, T. Lingenberg, R. Liepina, and F. Lagioia, "Towards reliable retrieval in RAG systems for large legal datasets," in Proc. Natural Legal Language Processing Workshop, 2025, pp. 17-30.

[5] S. M. W. Rahman, S. Kim, H. Choi, D. S. Bhatti, and H. N. Lee, "Legal Query RAG," IEEE Access, 2025.

[6] B. An, S. Zhang, and M. Dredze, "RAG LLMs are not safer: a safety analysis of retrieval-augmented generation for large language models," arXiv, 2025.

[7] R. C. Barron, M. E. Eren, O. M. Serafimova, C. Matuszek, and B. S. Alexandrov, "Bridging legal knowledge and AI: retrieval-augmented generation with vector stores, knowledge graphs, and hierarchical factorization," arXiv, 2025.

[8] "BharatLex: custom AI chatbot for legal query-driven using RAG and optimized LLM fusion," Atlantis Press, 2025.

[9] C. Y. Chang et al., "MAIN-RAG: multi-agent filtering retrieval-augmented generation," arXiv, 2024.

[10] "DEAL: high-efficacy privacy attack on retrieval-augmented generation systems via LLM optimizer," OpenReview, 2025.

[11] "Enhancing the precision and interpretability of retrieval-augmented generation in legal technology: a survey," IEEE Access, 2025.

[12] "BNS Mitra: RAG-optimized LLM-based AI-powered legal virtual assistant," IEEE Xplore, 2025.

[13] "A reusable prompting framework for applying large language models to legal tasks," IEEE Journals and Magazine, 2026.

[14] "Multilingual low-latency emergency VoIP system using LLM for speech reconstruction," IEEE Xplore, 2025.

[15] "Exploring LLM applications in law: a literature review on current legal NLP approaches," IEEE Xplore, 2025.