WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI Smart Legal Assistant System

Adithi M D[1], Anushree M[2], Archana R[3], Chameli Nayaha J[4], Mrs. JENIFER R [5]

Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India
Corresponding author: Mrs. JENIFER R | rjenifercse@siet.ac.in

DOI : https://doi.org/10.55248/gengpi.07.1026.34081

ABSTRACT :

Access to legal information and routine case-related services is often fragmented across documents, professionals, and court workflows. This paper presents the design and implementation scope of AI Smart Legal Assistant, a web-based platform that combines client–lawyer service management with document-aware legal question answering. The system integrates role-based authentication, user profiles, case records, hearing and appointment management, notifications, and controlled document access. Its artificial intelligence layer uses PDF text extraction and optical character recognition, legal text cleaning and chunking, embedding generation, semantic and hybrid retrieval, and retrieval-augmented generation (RAG). Retrieved legal provisions are used to support act and section identification, citation checks, and explanations produced by a locally served Ollama llama3.2:3b model. FastAPI REST services connect the application layers, while PostgreSQL supports persistent records. The paper describes the modular architecture, intended processing workflow, security boundaries, and a reproducible evaluation framework. The supplied feature inventory establishes the system scope; numerical accuracy, latency, and usability results are not claimed without experimental evidence.

Keywords: Legal technology; Indian law; retrieval-augmented generation; OCR; hybrid retrieval; FastAPI; PostgreSQL; access control.

1. Introduction

Legal service workflows involve multiple actors and information sources, including clients, lawyers, case records, uploaded documents, court hearing schedules, and statutory provisions. A digital legal assistant can reduce fragmentation by providing a common interface for operational services and evidence-supported information retrieval. However, legal text is sensitive to jurisdiction, statutory numbering, amendments, and document context. A conversational response that sounds plausible is not necessarily legally accurate.

The proposed AI Smart Legal Assistant addresses two complementary requirements: transactional service workflows and legal knowledge assistance. Its service layer supports user registration, role-specific dashboards, case and appointment management, hearing tracking, and notifications. Its knowledge layer processes legal documents and retrieves relevant statutory passages before generating explanations. The objective is decision support and information access, not the replacement of qualified legal advice.

2. Related Work

Retrieval-augmented generation combines information retrieval with language generation to provide external context at answer time (Lewis et al., 2020). Dense passage retrieval uses learned representations to rank passages against questions (Karpukhin et al., 2020). Sentence-BERT provides sentence embeddings suited to semantic similarity (Reimers and Gurevych, 2019). These techniques motivate embedding-based legal search, but statutory retrieval additionally requires exact act and section matching, source provenance, and careful treatment of legal updates.

OCR extends retrieval to scanned documents, although extraction errors may propagate into downstream indexing. In a legal setting, document-level permissions and grounded citations must therefore be enforced throughout the pipeline rather than added only at the chat interface.

3. System Requirements and Functional Scope

The platform supports three primary functional groups. Identity and service management includes registration and login, role-based authentication and authorization, client and lawyer dashboards, profile management, case management, lawyer search, appointment booking, court hearing management, and notifications. Document management includes uploads, ownership validation, access control, extraction, OCR, analysis, and document-based question answering. AI-assisted legal research includes legal knowledge base creation, cleaning, chunking, embeddings, RAG, semantic and hybrid search, act/section identification, citation validation, and natural-language explanations.

Implementation and integration work covers Ollama llama3.2:3b, FastAPI and REST APIs, PostgreSQL connectivity, frontend–backend integration, input validation, error handling, Swagger UI API testing, retrieval accuracy testing, navigation improvements, and Git-based team integration.

4. Proposed Architecture

The architecture is organized into five layers: (i) a web presentation layer for client and lawyer dashboards; (ii) an identity and authorization layer that applies role and ownership checks; (iii) a FastAPI service layer exposing business and AI endpoints; (iv) a PostgreSQL persistence layer for accounts, cases, appointments, hearings, notifications, and document metadata; and (v) a legal intelligence layer responsible for extraction, indexing, retrieval, evidence assembly, and local language-model inference. The principal components are summarized in Fig. 1.

A request for case information follows the authenticated application path and is filtered by the requester's role and record ownership. A legal question follows a separate evidence-oriented path: query interpretation, retrieval from authorized sources, optional act/section constraints, source validation, and response generation. These paths share identity controls but have different reliability requirements.

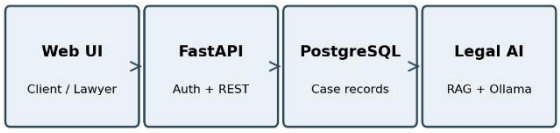


Fig. 1 - Conceptual architecture of the AI Smart Legal Assistant.

5. Legal Document Processing and RAG

5.1. Extraction and preprocessing

Uploaded PDF files are processed using direct text extraction where possible and OCR for image-based pages. The extracted text is cleaned and segmented into chunks while preserving document identity, page location, act name, section number, and other available metadata. Chunk boundaries should avoid separating a statutory heading from its provision, exceptions, or explanations. The processing stages are illustrated in Fig. 2.

5.2. Indexing and retrieval

Embeddings represent chunks for semantic similarity search. Hybrid retrieval combines semantic similarity with lexical and metadata signals so that exact references such as "BNSS Section 173" are not displaced by broadly related passages. Candidate results are ranked with emphasis on the requested act, section, and supporting context. Document permissions should be checked before evidence is exposed to the user or language model.

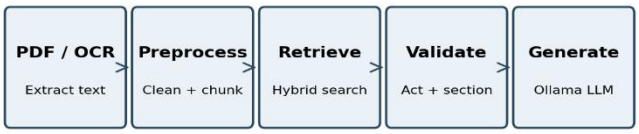


Fig. 2 - Illustrative legal document processing and grounded answer workflow.

5.3. Grounded response generation

The retrieval layer supplies selected passages to Ollama llama3.2:3b. The response prompt should instruct the model to distinguish retrieved facts from interpretation, identify cited acts and sections, and decline unsupported conclusions. Citation validation checks that cited sections occur in the retrieved evidence. A valid-looking citation alone does not establish that the legal interpretation is correct or that the provision is currently in force.

6. Application Services and Security

Role-based access control separates client and lawyer operations and limits document and case access to authorized users. Document ownership validation should be performed server-side on each read or mutation request. API input validation, controlled error responses, and authenticated routes reduce unintended data exposure. PostgreSQL provides persistence for structured records; uploaded document storage and retention rules require explicit deployment configuration. Sensitive legal files should be protected in transit and at rest, and access should be logged where appropriate.

7. Implementation and Integration

The project feature inventory identifies a FastAPI backend with REST endpoints, PostgreSQL connectivity, an Ollama-hosted local language model, and a frontend integrated with service and AI APIs. Swagger UI is used for endpoint inspection and testing. GitHub branches and team integration support parallel development across frontend, backend, AI/RAG, and application features. These statements describe reported implementation components; this manuscript does not independently verify deployment readiness or complete end-to-end operation.

8. Evaluation Methodology

A journal-grade evaluation should use a fixed test collection containing questions from each indexed Indian Act, exact act–section lookups, ambiguous legal questions, multi-page provisions, scanned PDFs, document ownership scenarios, and adversarial cross-role access attempts. Ground-truth references should be independently checked against authoritative statute texts and their effective dates.

Retrieval can be assessed with Recall@k, Precision@k, mean reciprocal rank (MRR), and exact section-match accuracy. Generated answers should be reviewed for citation correctness, evidence support, completeness, and unsupported claims. OCR quality can be evaluated using character and word error rates. Service-level testing should cover API status codes, authorization denials, upload limits, response latency, and error handling. Usability testing can evaluate completion of appointment booking and case-tracking tasks.

9. Discussion and Limitations

Combining case services with document-grounded legal assistance creates a single workflow for users, but the integrated design introduces trade-offs. Exact legal references can be missed by purely semantic ranking; OCR errors may corrupt sections; statutory amendments can invalidate otherwise well-formed answers; and a locally hosted model may generate assertions that exceed retrieved evidence. The architecture should therefore prioritize metadata-aware retrieval, authoritative-source updates, explicit uncertainty, and human review for consequential legal decisions.

No experimental dataset, benchmark scores, independent user study, or production security audit was supplied for this paper. Accordingly, the present work is a system-design and implementation report rather than a validated performance comparison. Results must be inserted only after testing has been completed and recorded.

10. Conclusion and Future Work

AI Smart Legal Assistant brings together role-based legal service management and retrieval-grounded statutory assistance. Its reported modules cover identity, cases, lawyers, hearings, documents, OCR, hybrid retrieval, RAG, citation checks, and locally served explanation generation. Future work includes verified benchmark evaluation across indexed Acts, amendment-aware knowledge-base refresh, calibrated abstention on weak evidence, security assessment, and user studies with legal professionals and clients. These steps are necessary to establish reliability beyond functional integration.

REFERENCES :

1. Bhattacharya, P., et al. (2019). Identification of Rhetorical Roles of Sentences in Indian Legal Judgments. JURIX 2019.
2. Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. NeurIPS 33, 1877–1901.
3. Chalkidis, I., et al. (2022). LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. ACL, 4310–4330.
4. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of Law School. Findings of EMNLP.
5. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT, 4171–4186.
6. Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2024). RAGAs: Automated Evaluation of Retrieval Augmented Generation. EACL System Demonstrations.
7. Government of India. (1954). Special Marriage Act, 1954 (Act 43 of 1954), as amended.
8. Government of India. (1955). Hindu Marriage Act, 1955 (Act 25 of 1955), as amended.
9. Government of India. (2023). Bharatiya Nagarik Suraksha Sanhita, 2023 (Act 46 of 2023). Gazette of India.
10. Government of India. (2023). Bharatiya Nyaya Sanhita, 2023 (Act 45 of 2023). Gazette of India.
11. Government of India. (2023). Bharatiya Sakshya Adhiniyam, 2023 (Act 47 of 2023). Gazette of India.
12. Guha, N., et al. (2023). LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. NeurIPS Datasets and Benchmarks.
13. Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M.-W. (2020). REALM: Retrieval-Augmented Language Model Pre-Training. ICML, 3929–3938.
14. Henderson, P., et al. (2022). Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset. NeurIPS Datasets and Benchmarks.

15. Izacard, G., and Grave, E. (2021). Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. EACL, 874–880.

16. Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547.

17. Kapoor, A., et al. (2022). HLDC: Hindi Legal Documents Corpus. Findings of ACL 2022.

18. Karpukhin, V., et al. (2020). Dense Passage Retrieval for Open-Domain Question Answering. EMNLP, 6769–6781.

19. Khattab, O., and Zaharia, M. (2020). ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR, 39–48.

20. Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 33, 9459–9474.

21. Malkov, Y. A., and Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. IEEE TPAMI, 42(4), 824–836.

22. Nogueira, R., and Cho, K. (2019). Passage Re-ranking with BERT. arXiv:1901.04085.

23. Rashkin, H., Reitter, D., Tomar, G. S., and Das, D. (2023). Measuring Attribution in Natural Language Generation Models. Computational Linguistics, 49(4), 777–840.

24. Reimers, N., and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP-IJCNLP, 3982–3992.

25. Robertson, S., and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval, 3(4), 333–389.

26. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). Role-Based Access Control Models. IEEE Computer, 29(2), 38–47.

27. Smith, R. (2007). An Overview of the Tesseract OCR Engine. ICDAR, 629–633.

28. Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS 30.

29. Xu, Y., et al. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. KDD, 1192–1200.

30. Xu, Y., et al. (2021). LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. ACL-IJCNLP, 2579–2591.