iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85774

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

LawGPT: AI-Driven Legal Evidence Analyzer and RAG Intelligence Platform

Rahul R[1], Lohith Swamy S M[2], Prajwal V[3], Chetan T[4], Prof. Puneeth S P[5]
[1, 2, 3, 4]*Department ofInformation Science and Engineering, Bapuji Institute ofEngineering and Technology, Davanagere, Karnataka, India*
[6]*Assistant Professor, Department ofInformation Science and Engineering, Bapuji Institute ofEngineering and Technology, Davanagere, Karnataka, India*

Abstract: ***Digital legal research tools can help legal practitioners organize case documentation, analyze evidentiary filings, and access statutory resources for judicial case preparation. However, document ingestion, precedent retrieval, and adversarial argument formulation are frequently presented across disconnected tools. This paper describes LawGPT, an interactive AI-assisted web application that combines case workspace management, multi-page PDF evidence extraction, Retrieval-Augmented Generation (RAG) legal analysis, adversarial argument synthesis, precedent search, and automated case brief generation. Users can initialize case records, upload evidentiary writ petitions or FIR reports, explore applicable statutory provisions, and review structured analytical reports. The system uses a React web interface, a Node.js and Express orchestration backend with a MongoDB database, and a Python FastAPI microservice integrated with ChromaDB vector storage, as implemented in the project repository. Functional test cases are reported for user registration, JWT login, case creation, PDF document ingestion, AI entity and timeline extraction, adversarial argument generation, precedent retrieval, and PDF report export. These checks indicate that the tested features operated as expected in the reported project environment. They do not establish judicial validity, binding legal accuracy, or courtroom outcome guarantees. LawGPT is intended as a legal research and case preparation support application and does not replace qualified legal counsel or judicial adjudication.***

Keywords: Legal intelligence; Retrieval-Augmented Generation (RAG); Evidence analysis; Adversarial arguments; ChromaDB; FastAPI

I. INTRODUCTION

Legal evidence analysis and statutory research are foundational components of judicial proceedings and litigation preparation. Digital legal technologies increasingly provide mechanisms to organize case records and search statutory repositories.[1,2] However, conventional platforms frequently address keyword search or isolated document storage separately, offer limited contextual grounding in uploaded case files, or provide minimal automated support for constructing balanced prosecution and defense arguments.[3,4]

This project presents LawGPT, an integrated web application designed to unify multi-page evidence extraction, vector-based legal retrieval, and structured brief generation within a single workspace. The system incorporates case lifecycle management, chronological timeline extraction from uploaded petitions, statutory penal mapping, adversarial argument synthesis, precedent lookup, contextual RAG chat, and downloadable PDF report compilation. The aim of the project is to provide an accessible interface for analyzing textual case evidence and exploring grounded legal strategies.

II. MATERIAL AND METHODS

A. Study Design

This project used a software engineering and end-to-end functional testing approach to produce LawGPT, an interactive web application for AI-driven legal evidence analysis and case preparation support. The work concerns the microservice architecture and functional behavior ofa software platform; it was not a prospective empirical study oflive courtroom trials.

B. Study Location and Duration

The project was carried out in the Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India. The project synopsis identifies the academic development period as 2025–2026.

©IJRASET: All Rights are Reserved

C. Study Population and Sample Size

Human litigants or study participants were not recruited for the software evaluation described in the project report. Accordingly, a participant sample size and clinical sample-size calculation are not applicable. The reported evaluation concerns functional test cases and verification against judicial filings (such as High Court writ petitions) rather than measurements collected from a human population.

D. System Scope and Selection Criteria

The application scope included role-based user registration and JWT login, case file initialization, multi-page PDF document upload and OCR extraction, entity and chronological timeline extraction, statutory mapping, adversarial argument generation for petitioner and respondent roles, semantic precedent search, interactive case chat, and PDF report export. For this software evaluation, the documented scope was limited to the listed application modules and functional test cases. Formal judicial rulings, live court e-filing, and binding verdict prediction were outside the reported evaluation.

E. Procedure Methodology

The application workflow begins when a user registers or logs in via JWT authentication and opens the workspace dashboard. The user can create a case record and upload evidentiary documents such as writ petitions, FIR reports, or witness statements. Backend services process these files through text cleaning and chunking pipelines, converting text segments into vector embeddings stored in ChromaDB.[9,15] When analysis is requested, the Python FastAPI microservice retrieves relevant statutory sections and seeded judicial precedents to generate an executive summary, chronological timeline, extracted entities (persons, organizations,
statutes, places), adversarial arguments, and an evidence admissibility health score.[14] Case records, extracted pages, analyses, chat logs, and hearing schedules are stored in MongoDB and can be exported as a privileged PDF case brief.[16]

The interface provides dedicated modules for the dashboard, case list, overview, document manager, AI analysis tab, adversarial arguments tab, judgments research hub, RAG chat, hearing scheduler, and report generator. The implementation uses React, Vite, and Tailwind CSS for the frontend, Node.js and Express with MongoDB for core orchestration, and Python FastAPI with Pydantic and ChromaDB for AI retrieval.

F. Functional Testing & Statistical Analysis

Functional verification covered registration, login, case creation, PDF ingestion, AI factual analysis, adversarial argument generation, precedent vector search, and PDF report compilation. No statistical analysis of human participant data was reported; evaluation is presented through deterministic functional test outcomes across the eight verified system workflows.

III. RESULT

The application screens documented in the repository cover the login view, case analytics dashboard, new case form, case workspace overview, document upload tab, AI legal analysis tab, judgments search tab, adversarial arguments tab, RAG chat tab, and PDF report generator.

Together, these interfaces demonstrate the end-to-end workflow available through LawGPT.

In end-to-end verification (Case #1456, *Mandya District Kurubara Association* writ petition, 62 pages), the document pipeline extracted page text and indexed embeddings without error. The AI analysis module identified chronological events (such as the December 28, 2025 AGM and February 21, 2026 term expiry), extracted legal entities, and mapped Sections 12, 13, and 27A of the
Karnataka Societies Registration Act, 1960 along with Articles 226 and 227.[10] The arguments module synthesized petitioner grounds citing Supreme Court precedents[11,12,13] alongside respondent rebuttals and computed an 85% Evidence Health Score. All eight functional test cases (Table 1) passed (100% within the verified test suite).

A. Functional Test Results

Test ID	Feature tested	Reported expected behavior	Status
TC-1	User registration & RBAC	User account is registered and role is assigned successfully	Pass
TC-2	JWT login authentication	Login succeeds with valid credentials and issues bearer token	Pass
TC-3	Case initialization	New case workspace and party metadata are stored in MongoDB	Pass
TC-4	PDF document ingestion	Multi-page PDF is uploaded, chunked, and indexed in ChromaDB	Pass
TC-5	AI case & entity analysis	Executive summary, timeline, entities, and statutes are displayed	Pass

Snapshots



IV. DISCUSSION

LawGPT brings several legal research, document parsing, and case preparation workflows together in one unified workspace, making multi-document case analysis more convenient to manage. The case workspace links PDF evidence ingestion, chronological event extraction, statutory penal mapping, precedent lookup, adversarial argument generation, and automated brief compilation. This arrangement gives the platform a broad legal-support role: advocates, junior lawyers, and law students can upload case petitions and inspect structured factual matrices and statutory references without navigating across disconnected tools. Research in Legal Judgment Prediction (LJP) and natural language processing in judicial domains provides a relevant context for this kind of automated document comprehension and decision-support system.[1,2,3]

The functional results indicate that the microservice pipelines documented in the project repository responded as expected in the reported test set. JWT registration and role-based access control secure user workspaces; PDF upload and chunking services extract text and index vector embeddings; AI analysis routes extract dates, entities, and statutory sections; and argument and report services synthesize prosecution/petitioner grounds, defense rebuttals, and downloadable PDF briefs. All eight reported functional test cases passed. These outcomes are evidence of feature-level operation across the React, Node.js, FastAPI, MongoDB, and ChromaDB architecture under tested conditions. They do not measure long-term enterprise scalability, zero-hallucination guarantees across all statutory domains, or whether users achieve favorable outcomes in contested litigation.

The integration of vector-based Retrieval-Augmented Generation (RAG) with adversarial argument synthesis is consistent with the wider principle that legal AI systems must ground generative outputs in verified statutory and precedent repositories.[2,9] Prior domain datasets and studies have examined Indian Supreme Court judgments, bail outcome prediction, and legal document segmentation in relation to machine learning models.[5,6,7,8] These publications provide background for the computational legal tasks represented in LawGPT; they do not validate this prototype or its specific analytical heuristics as an official judicial citator. The present project should therefore be interpreted as a software engineering implementation that organizes and accelerates access to case evidence and legal arguments, rather than as a certified legal adjudication engine.

Several practical considerations govern real-world legal usage. Extracted timelines, entity counts, and admissibility scores depend on the legibility and completeness of uploaded PDF documents, and the platform's outputs can only be as reliable as the underlying OCR text extraction, seeded judgment corpus, and LLM prompting rules.

The Evidence Health Score and adversarial arguments displayed in the workspace should be clearly distinguished from formal judicial admissibility rulings, and generated case reports should be treated as preliminary research briefs rather than final court pleadings. The application incorporates confidentiality notices and role-based access restrictions, and practitioners must verify all statutory provisions and case citations against official court reporters prior to filing.

The system also has technical and evaluation limitations. Precedent retrieval currently relies on the locally seeded ChromaDB judgment collection, and the project report does not document a formal usability trial with practicing advocates, an independent cryptographic security audit, or quantitative retrieval precision/recall benchmarks across regional Indian courts. Privileged legal records require strict encryption-at-rest, audit trails, and compliance with data protection regulations. Future evaluation should document these safeguards and measure citation accuracy with legal domain experts before drawing conclusions about professional adoption.

Further development could prioritize connecting the ChromaDB vector store with live national judicial repositories (such as eCourts and Supreme Court judgment APIs), improving OCR accuracy for scanned vernacular pleadings, and expanding bilingual localization (building upon the English and Kannada interface translation files included in the frontend repository). Argument generation rules and evidence scoring metrics should be evaluated against expert-annotated case briefs before being described as legally comprehensive. A future study could define quantitative retrieval benchmarks (such as Mean Reciprocal Rank and citation precision) in advance alongside structured usability evaluations with law students and legal aid clinics.

V. CONCLUSION

LawGPT is an integrated AI-assisted web application for analyzing legal case evidence, retrieving statutory precedents, and generating structured adversarial briefs. It combines secure case workspace management, multi-page PDF document ingestion, chronological event and entity extraction, statutory penal mapping, petitioner and respondent argument synthesis, semantic precedent search, contextual RAG chat, and downloadable PDF report compilation. The project repository and verification suite document the decoupled microservice workflow across React, Node.js, MongoDB, FastAPI, and ChromaDB, and record eight functional test cases as passed, indicating that the tested modules performed their stated functions in the reported environment.

The results support the technical feasibility of bringing multi-tier microservice orchestration and Retrieval-Augmented Generation together within a unified legal intelligence platform and provide a foundation for further refinement. They do not establish judicial validity, zero-error statutory citation, sustained practitioner adoption, or guaranteed courtroom outcomes. LawGPT should therefore be understood as a legal research, academic education, and case-preparation prototype, not as a substitute for qualified legal representation or formal judicial reasoning.

Future work should evaluate usability and citation accuracy with practicing lawyers and law students, expand the vector database with comprehensive national case law, validate evidence admissibility heuristics, and audit enterprise privacy and encryption safeguards. Larger evaluations with legal domain specialists could then examine case preparation efficiency and retrieval precision using standardized benchmarks. With these steps, the application can be advanced as a responsible, accessible platform for legal evidence analysis while keeping its capabilities and limitations transparent.

REFERENCES

[1] N. Aletras, D. Tsarapatsanis, D. Preotiuc-Pietro, and V. Lampos, "Predicting Judicial Decisions of the European Court of Human Rights: A Natural Language Processing Perspective," PeerJComputer Science, vol. 2, p. e93, 2016. (https://peerj.com/articles/cs-93/)

[2] J. Cui, X. Shen, F. Nie, Z. Wang, and Y. Chen, "A Survey on Legal Judgment Prediction: Datasets, Metrics, Models and Challenges," arXiv preprint arXiv:2204.04859, 2022. (https://arxiv.org/abs/2204.04859)

[3] M. Medvedeva, M. Vols, and M. Wieling, "Using Machine Learning to Predict Decisions of the European Court of Human Rights," Artificial Intelligence and Law, vol. 28, no. 2, pp. 237–266, 2020. (https://link.springer.com/article/10.1007/s10506-019-09255-y)

[4] C. O'Sullivan and J. Beel, "Predicting the Outcome of Judicial Decisions Made by the European Court of Human Rights," in Proc. 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science (AICS), CEUR Workshop Proceedings, vol. 2563, 2019. (https://ceur-ws.org/Vol-2563/aics_26.pdf)

[5] NyayaAnumana, "Indian Legal Judgment Prediction Dataset and Benchmark," arXiv preprint arXiv:2412.08385, 2024.

[6] (https://arxiv.org/abs/2412.08385)

[7] Indian Bail Judgments Dataset, "Indian Bail Judgments Dataset for Legal NLP Research," arXiv preprint arXiv:2507.02506, 2025.

[8] (https://arxiv.org/pdf/2507.02506)

[9] IndicLegalQA, "Indian Supreme Court Judgments Dataset," Kaggle Datasets, 2024. (https://www.kaggle.com/datasets/vangap/indian-supreme-court-judgments)

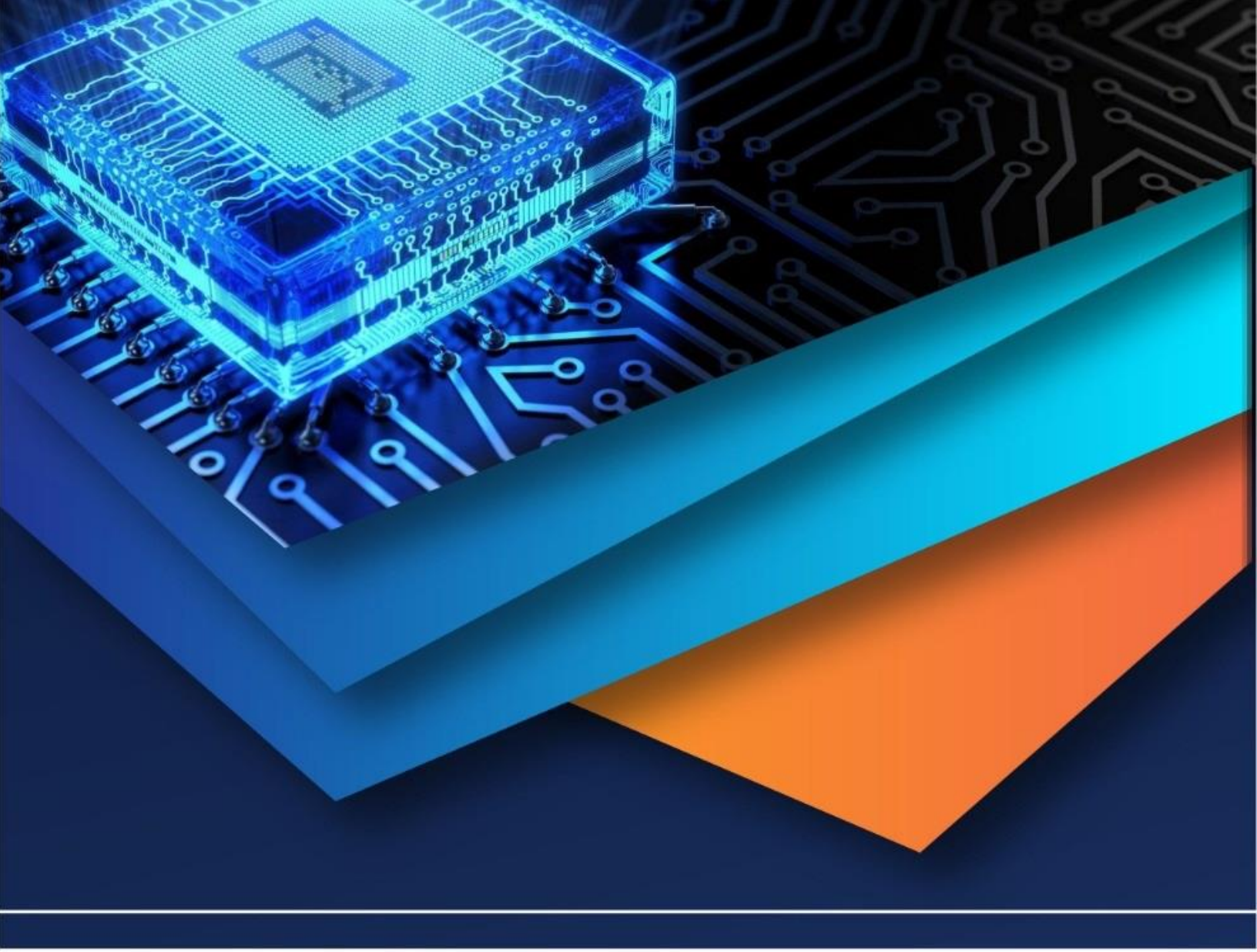
[10] LegalSeg, "Indian Legal Judgments Dataset for NLP Applications," arXiv preprint arXiv:2502.05836, 2025. (https://arxiv.org/abs/2502.05836)

[11] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

©IJRASET: All Rights are Reserved

[12] Government of Karnataka, "The Karnataka Societies Registration Act, 1960 (Karnataka Act No. 17 of 1960), Sections 12, 13, and 27A," Department of Parliamentary Affairs and Legislation, 1960.
[13] Supreme Court of India, A.K. Kraipak & Ors. v. Union of India & Ors., AIR 1970 SC 150, 1969.
[14] Supreme Court of India, Whirlpool Corporation v. Registrar of Trade Marks, Mumbai & Ors., (1998) 8 SCC 1, 1998.
[15] Supreme Court of India, State of U.P. v. Sudhir Kumar Singh & Ors., (2021) 19 SCC 706, 2020.
[16] FastAPI, "FastAPI Framework: High Performance REST APIs with Python and Pydantic," FastAPI Documentation, 2026.
[17] Chroma, "ChromaDB: Open-Source Embedding and Vector Database Documentation," Chroma Documentation, 2026.
[18] MongoDB, "MongoDB Document Database Architecture and Mongoose ODM Manual," MongoDB Documentation, 2026.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)