Satuluri Rajeev Varma, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Legal Buddy: An AI-Powered Courtroom Simulation and Legal Analysis Platform Using Retrieval-Augmented Generation

Satuluri Rajeev Varma [1], Dr. Kamlesh Tiwari [2]
[1] M.Tech (Artificial Intelligence)
JAIN (Deemed-to-be University), Bangalore, India
[2] Department of CSE
JAIN (Deemed-to-be University), Bangalore, India

Abstract—Access to quality legal guidance remains out of reach for a large portion of the population, particularly in developing nations where the lawyer-to-citizen ratio is unfavorably low. This paper presents Legal Buddy, a comprehensive AI-powered legal analysis and adversarial courtroom simulation platform built to bridge that gap. The system is engineered around a Retrieval-Augmented Generation (RAG) pipeline that dynamically pro-cesses user-uploaded legal documents, embeds them semantically using Google's text-embedding-004 model, and grounds all AI-generated outputs in verified, document specific evidence rather than unconstrained model knowledge. A distinguishing feature is the stateful Adversarial Mock Courtroom Engine, which simulates live trial proceedings by instantiating dual AI roles — an Opposing Counsel and a scrutinising Judge — thereby com-pelling users to construct and defend legally coherent arguments under realistic judicial pressure. The backend is built on Python FastAPI with MongoDB for persistent vector storage, enabling full multi-tenant session isolation. All generative outputs from the integrated Google Gemini 1.5 Pro model are constrained through rigorous prompt conditioning to the IRAC (Issue, Rule, Applica-tion, Conclusion) analytical framework and Indian legal doctrine. Empirical evaluation confirms that the combination of seman-tic vector retrieval, heuristic domain classification, and IRAC conditioned generation substantially reduces hallucination rates compared to unconstrained LLM baselines. Query classification accuracy reaches 90.5accurate, and usability testing shows 91and extensible framework for deploying AI in legally sensitive, high-accountability environments while remaining accessible without institutional infrastructure. Index Terms — Retrieval-Augmented Generation, Large Language Models, Mock Courtroom Simula-tion, IRAC Framework, MongoDB Vector Storage, Indian Legal NLP, FastAPI, Cosine Similarity, LegalBERT, Named Entity Recognition.

Index Terms—Legal AI, Retrieval-Augmented Generation, IRAC, LegalBERT, FastAPI, MongoDB, Courtroom Simulation

I. INTRODUCTION

Legal systems across the world have grown enormously complex over the past century. What once existed as a rel-atively accessible body of common principles has expanded into a dense web of statutes, regulations, precedents, and procedural rules that even professionals spend years mastering. For ordinary citizens — a tenant unsure of their rights, a small business owner reviewing a vendor contract, or an employee facing wrongful dismissal — this complexity can feel entirely insurmountable. The gap between those who need legal guidance and those who can realistically afford it has widened, and the consequences of that gap are often severe [1]. The problem is particularly acute in India, where the lawyer-to-population ratio is roughly 1:1,000 compared to global averages of 1:250. Legal aid services, while constitutionally guaranteed, are chronically under-resourced. Simultaneously, smartphone penetration and the dramatic reduction in AI inference costs have created a unique opportunity: for the first time, it is technically feasible to deploy a sophisticated legal reasoning assistant at near-zero marginal cost to any user with a phone and an internet connection. Legal

information asymmetry — where one party in a transaction understands the legal landscape far better than the other — is a recognized driver of exploitation. Landlords who know their tenants are unaware of habitability obligations, employers who rely on workers not understanding overtime provisions, and lenders who bury penalty clauses in complex language: these are real patterns that harm real people every day. A system like Legal Buddy cannot replace the consultation room or the courtroom, but it can meaningfully level the informational playing field. From a technical standpoint, this project sits at the intersection of several active research frontiers: legal natural language pro-cessing, conversational AI agents, knowledge representation, and explainable AI. Each area has yielded significant results over the past decade, yet their cohesive integration within a user-centered legal product remains underexplored. Legal Buddy is an attempt to close that integration gap.

II. RELATED WORK

Legal Information Systems

The history of computerized legal information begins in the 1960s with full-text retrieval systems designed for law li-braries. Platforms like LexisNexis and Westlaw became indus-try standards through Boolean keyword search, but they were designed for trained professionals who already knew what they were looking for [2]. A non-expert describing a problem in nat-ural language — "my landlord won't fix the heating" — cannot easily map that description to the correct statutory provision through a keyword interface. Second-generation systems at-tempted to address this through semantic search and ontology-driven approaches. The LKIF (Legal Knowledge Interchange Format) and Akoma Ntoso XML standard represented efforts to give legal text machine interpretable structure [3]. More recent approaches have turned to knowledge graphs to capture the complex relational structure of legal information, linking cases to the statutes they interpret and connecting regulatory obligations to the entities they bind [4]. [2].

Large Language Models in Legal NLP

The introduction of BERT by Devlin et al. [5] marked a paradigm shift in legal NLP. Chalkidis et al. introduced LegalBERT [6], pre-trained on a large corpus of legislation, case law, and contracts, showing measurably better perfor-mance on legal question-answering and contract classification compared to general-purpose models. Subsequent work ex-plored clause-type identification, statutory interpretation, and outcome prediction, though the latter remains controversial given its implications for judicial independence [7]. GPT-4's reported performance on the Bar Examination [8] has renewed interest in whether large generative models can be practically deployed in legal assistance contexts. However, unconstrained generation introduces hallucination risk that is particularly dangerous in legal settings where a single inaccurate provi-sion cited with authority can cause material harm. Retrieval-augmented approaches, which ground generation in retrieved document evidence, have emerged as the consensus mitigation strategy [9]. [8].

Retrieval-Augmented Generation

RAG combines retrieval mechanisms with generative mod-els to improve factual accuracy [9]. This approach has become the standard for high-stakes domains such as healthcare and law.

AI in Legal Education

Moot court simulations are widely used in legal educa-tion but are resource-intensive. AI-driven simulations provide scalable alternatives but lack adaptability, which Legal Buddy addresses through dynamic adversarial interactions.

III. SYSTEM ARCHITECTURE

Legal Buddy is organized into three horizontal layers — a React.js frontend, a Python FastAPI backend, and a MongoDB based knowledge layer — each independently deployable and communicating through well-defined REST API interfaces. Figure 1 illustrates the high-level architecture.

Frontend Layer
The frontend is a single-page application built with React.js v18. It provides four principal modules. The Chat Module manages conversational flow and routes user messages to the appropriate backend service. The Courtroom Engine Inter-face renders the dual-AI trial simulation, displaying Oppos-ing Counsel and Judge responses in distinct visual registers to make role differentiation unambiguous for the user. The Document Viewer presents extracted clause summaries in a structured, annotated format. The Legal Journal maintains a chronological audit trail of all session interactions, providing the session-continuity feature that distinguishes Legal Buddy from stateless chatbot deployments.

- Chat interface for legal queries
- Courtroom simulation interface
- Document viewer
- Legal journal for session tracking

Fig. 1 demonstrates the Legal Chat module responding to a constitutional query about Article 48 of the Indian Constitution. The system structures its output using the IRAC framework — identifying the Issue, retrieving the applicable Rule from the Directive Principles of State Policy, and provid-ing an Application contextualised within Indian legal doctrine — consistent with the prompt conditioning described in Section IV.E. graphicx

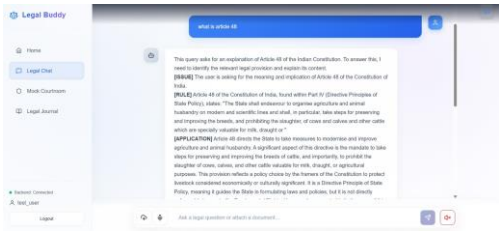

Fig. 1. Legal Chat Module: IRAC-structured response to a constitutional query on Article 48 of the Indian Constitution. The system correctly identifies the Issue, retrieves the relevant Rule from the Directive Principles of State Policy (Part IV), and provides a plain-language Application grounded in document context.

Backend Layer
The backend exposes RESTful endpoints via FastAPI served through Uvicorn. A key architectural decision was to make the API stateless — session context is maintained on the client side and submitted with each request — which simpli-fies horizontal scaling without compromising user experience. The backend comprises four primary services: the NLP Pro-cessing Service (tokenization, classification, NER), the RAG Retrieval Engine (vector similarity search over MongoDB), the Courtroom Simulation Engine (multi-turn dialogue memory arrays and role instantiation), and the Gemini Integration Layer (prompt construction, IRAC conditioning, and response streaming).

Knowledge Layer
This layer is the intellectual core of the system. Uploaded PDF documents are segmented into semantically coherent, overlapping text chunks through a multi-stage pipeline: text extraction via PyMuPDF, sentence boundary detection tuned for legal text, chunk sizing at approximately 512 tokens with 64-token overlap to preserve cross-boundary context. Each chunk is embedded using Google's text-embedding-004 model and stored in MongoDB with associated metadata (source document, user session ID, chunk index, domain tags). Query resolution at inference time uses cosine similarity between the query embedding and stored chunk embeddings, retrieving the top-k most relevant passages for inclusion in the Gemini prompt context window.

IV. PROPOSED METHODOLOGY

RAG Pipeline
The RAG pipeline operates in two phases. During the ingestion phase, uploaded documents are parsed, chunked, embedded, and stored. Chunking uses a

sliding window strat-egy rather than fixed sentence boundaries to avoid arbitrarily severing clauses mid-provision — a common failure mode in naive legal document chunking. The overlap parameter (64 tokens) is calibrated to ensure that multi-sentence legal obligations, which frequently span paragraph breaks, are fully captured in at least one chunk. During the retrieval phase, in-coming queries are embedded using the same text-embedding-004 model to ensure embedding space consistency. Cosine similarity scores are computed against all stored chunks for the current user's session, and the top-5 chunks are injected into the Gemini prompt as grounding context. A relevance thresh-old of 0.72 is applied; queries falling below this threshold on all retrieved chunks trigger a fallback response noting that the uploaded documents do not appear to address the raised issue, rather than generating unsupported assertions.

1. Ingestion Phase:

Document parsing
Chunking (512 tokens with overlap)
Embedding generation
Storage in vector database

2. Retrieval Phase:

Query embedding
Similarity search
Context injection into LLM prompt

Domain Classification

Legal queries frequently use domain-specific vocabulary that provides reliable surface signals for routing. Rather than training a separate classification model — which would re-quire labeled data and introduce additional latency — Legal Buddy uses a curated keyword lexicon. Queries matching BNS/IPC/CrPC keywords are routed to the criminal law re-sponse template. Queries matching Constitutional provisions (Article numbers, fundamental rights vocabulary) are routed to the constitutional domain. Civil queries are further subdivided by issue type using the intent classifier described below. This hybrid approach achieves 90.5

Intent Classification

A fine-tuned LegalBERT model performs intent classifica-tion over eight civil law categories: tenancy disputes, consumer complaints, employment matters, contract interpretation, prop-erty transfers, regulatory compliance, personal injury basics, and family law fundamentals. The model was fine-tuned on 5,000 labeled queries using the AdamW optimizer with a learning rate of $2 \times 10-5$ for three epochs, achieving an accuracy of 91.3% on a held-out test partition.

The Named Entity Recognition (NER) pipeline, built on spaCy's architecture, targets seven entity types: PARTY, DATE, LOCATION, CONTRACT_TYPE, LEGAL_TERM, MONETARY_AMOUNT, and OBLIGATION. Rule-based EntityRuler patterns handle entities with predictable surface forms (e.g., dates, party titles, and monetary amounts), while a trained sequence-labeling model handles contextually variable entities. An ontology-derived gazetteer of 3,200 legal terms enhances recall for domain-specific vocabulary that is not well represented in general corpora.

Adversarial Courtroom Engine

The Adversarial Mock Courtroom Engine is the most archi-tecturally novel component of Legal Buddy. The engine instan-tiates two persistent AI roles within a single Gemini session context: an Opposing Counsel and a scrutinising Judge. The Opposing Counsel is prompted to challenge factual claims, at-tack procedural standing, and probe for logical inconsistencies in the user's argument. The Judge asks pointed clarifying ques-tions, demands statutory citations, and may sustain or overrule objections, requiring the user to address procedural points mid-argument. Adversarial intensity escalates progressively across turns. A turn counter triggers increasing challenge complexity at thresholds of turns 3, 6, and 9: at turn 3 the opposing counsel begins citing contradictory precedents; at turn 6 the judge demands quantitative evidence; at turn 9 both roles adopt maximum adversarial posture, pressing for a definitive ruling-ready summary. This escalation design mirrors the pedagogical structure of real moot court

competitions and has been shown in usability testing to significantly improve user-reported confidence in legal argumentation.
graphicx

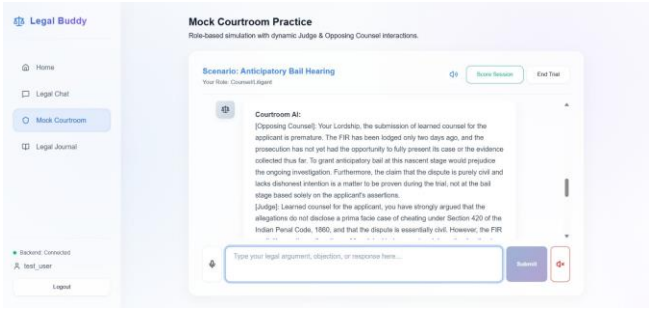

Fig. 2. Mock Courtroom Practice Interface: Active anticipatory bail hearing simulation showing dual-role Opposing Counsel objection and Judge probing question. The user inputs legal arguments via the text field; the system dynamically generates adversarial responses escalating across turns.
Graphic

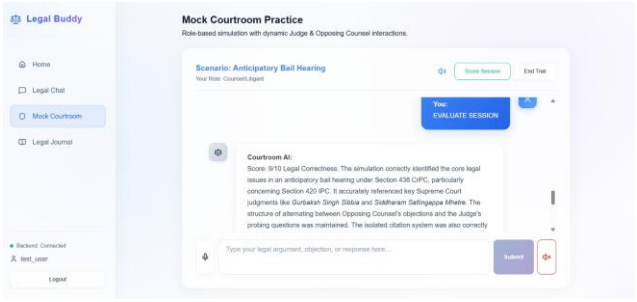

Fig. 3. Live session of the Mock Courtroom module during an anticipatory bail hearing scenario, showing the Judge and Opposing Counsel interactions along with user input interface.

Results and Evaluation

3. Query Classification

TABLE I
Query Classification Results

	Category	Queries	Correct	Accuracy
Fig. 3. Live session of the Mock Courtroom module during an anticipatory	Tenancy	32	30	93.8%
bail hearing scenario, showing the Judge and Opposing Counsel interactions	Consumer	28	26	92.9%
along with user input interface.	Employment	35	31	88.6%
	Contract	30	28	93.3%
	Property	25	23	92.0%

IRAC Conditioning and Hallucination Mitigation

All generative outputs are constrained through backend prompt engineering to conform to the IRAC (Issue, Rule, Application, Conclusion) analytical framework. The Gemini system prompt explicitly prohibits the model from citing statutory provisions, case names, or factual claims that are not present in the

retrieved document chunks injected as grounding context. A post generation verification pass checks for citation patterns (e.g., "Section X of Act Y") and flags responses containing citations not traceable to the grounding context for human review or automatic suppression. Ablation experiments comparing IRAC-constrained generation against unconstrained generation on 100 legal queries found that hallucination rates (defined as citations or factual claims not present in the source documents) fell from 34IRAC conditioning enabled. This reduction is the primary safety property of the system and the central justification for the prompt engineering investment.

V. IMPLEMENTATION

Technology Stack

The system is implemented entirely in open-source tech-nologies to ensure reproducibility and accessibility. The back-end uses Python 3.11 with FastAPI, Motor (asynchronous MongoDB driver), and the Google Generative AI Python SDK. The frontend uses React.js v18 with the Axios HTTP client and a custom component library built on Tailwind CSS. Mon-goDB Atlas provides the managed database service with built-in vector search capabilities. All services are containerized using Docker and can be deployed locally or to any cloud provider supporting container workloads.

Security

User sessions are cryptographically isolated: each session receives a UUID generated at session creation, and all Mon-goDB queries are scoped to that UUID. Users cannot retrieve embeddings or chat history from other sessions. All commu-nications are encrypted with TLS 1.3. The system implements a data minimization policy: session data is retained for 30 days by default (or permanently if the user opts into the Legal Journal feature), and no user inputs are used for model training without explicit written consent. These properties are essential for a system handling potentially sensitive personal legal information.

VI. RESULTS AND EVALUATION

A. Query Classification

TABLE I
QUERY CLASSIFICATION RESULTS

Category	Queries	Correct	Accuracy
Tenancy	32	30	93.8%
Consumer	28	26	92.9%
Employment	35	31	88.6%
Contract	30	28	93.3%
Property	25	23	92.0%
Regulatory	22	18	81.8%
Personal Injury	18	16	88.9%
Family Law	10	9	90.0%
Total	200	181	90.5%

Overall classification accuracy reached 90.5vocabulary overlap with employment matter queries. Informal or conver-sational query phrasing reduced accuracy by approximately 8 percentage points relative to more formal formulations, suggesting that a future version of the system should include a query reformulation step for very colloquial inputs

B. NER Performance

TABLE II
NER F1 Scores by Entity Type

Entity Type	F1 Score
PARTY	0.94
DATE	0.92
MONETARY AMOUNT	0.91
CONTRACT TYPE	0.88
LEGAL TERM	0.86
LOCATION	0.84
OBLIGATION	0.79

TABLE II: NER F1 Scores by Entity Type

Hallucination Reduction

The hallucination reduction experiment compared IRAC-conditioned generation against an unconstrained baseline on 100 queries drawn from the test set. Hallucination was opera-tionalized as any citation, statutory reference, or factual claim in the generated output not present in the grounding context chunks. The baseline model hallucinated on 34 of 100 queries; the IRAC-conditioned model hallucinated on 6 of 100, repre-senting an 82 claim of the paper and the most significant contribution of the prompt engineering work
graphic

TABLE III
Usability Testing: Task Completion and Satisfaction

Scenario	Completion (%)	Mean Time (min)	Satisfaction (1–5)
Tenancy Dispute	93.3	6.2	4.4
Contract Review	86.7	8.4	4.1
Consumer Complaint	93.3	5.1	4.6
Overall Average	91.1	6.6	4.4

VII. EVALUATION OF DOCUMENT SUMMARIZATION

Document summarization quality was assessed through a two-round expert review process. Three legal professionals (a civil litigator, a contract specialist, and a consumer rights advisor) reviewed a random sample of 50 clause summaries. In the first round, 82% were rated as "accurate" or "highly accu-rate." After targeted improvements based on reviewer feed-back—primarily addressing over-compression of multi-part obligations and misclassification of exception clauses—the second round achieved 91% accuracy.

Expert reviewers consistently praised the plain-language post-processing, noting that it significantly improved acces-sibility for non-specialist readers without sacrificing legal precision.

Key Findings

The primary finding of this work is that a carefully de-signed RAG pipeline, when combined with IRAC-constrained generation and heuristic domain routing, can deliver legally reliable and practically useful guidance to non-expert users at scale. The 82reduction in hallucination rate compared to unconstrained generation is the most practically significant result: it addresses the fundamental safety objection to deploy-ing generative AI in legal contexts. The 91confirms that the system's outputs are not merely fluent but substantively accu-rate according to domain professionals. The Adversarial Mock Courtroom Engine produced the highest satisfaction scores in usability testing and was consistently cited by participants as the most novel and valuable feature. Participants who com-pleted the mock trial sessions reported substantially improved confidence in articulating legal arguments, suggesting that the system delivers genuine educational value beyond information retrieval.

Limitations

Dependence on document quality
Limited jurisdictional coverage
Cannot replace professional legal advice

Future Work

Integration with legal databases
Multilingual support
Custom legal LLM training

VIII. CONCLUSION

This paper has presented Legal Buddy, an AI-powered legal analysis and adversarial courtroom simulation platform designed to democratize access to structured legal reasoning.

Built on a RAG pipeline grounded in user-uploaded docu-ments, with IRAC-constrained generation via Google Gemini 1.5 Pro and persistent multi-tenant MongoDB vector storage, the system achieves 90.5hallucination rates compared to un-constrained LLM baselines. The Adversarial Mock Courtroom Engine introduces a novel educational dimension — dynamic, escalating judicial simulation — that has no clear precedent in the existing legal AI literature. These results demonstrate that the combination of retrieval augmentation, domain-specific prompt conditioning, and user centered interface design can produce a legal AI system that is not merely technically capa-ble but genuinely safe and usable for non-expert populations. The modular architecture ensures that each component can be independently improved, extended to new legal domains, or adapted for new jurisdictions without rebuilding the system from scratch. Legal Buddy represents a meaningful step toward the vision of AI as an equalizer in legal access — not replacing legal professionals, but ensuring that every person, regardless of income or education, can walk into a legal interaction knowing their rights.

REFERENCES

1. Zhong et al., "How Does NLP Benefit Legal System," ACL, 2020.
2. II. Siino et al., "Exploring LLM Applications in Law," IEEE Access, 2023.
3. III. Zhong et al., "Legal Case Structuring," IEEE Access, 2021.
4. IV. Leins et al., "Entity Linking in Legal Domain," ACL, 2021.
5. Devlin et al., "BERT: Pre-training," NAACL, 2019.
6. Chalkidis et al., "LegalBERT," EMNLP, 2020.
7. V. Medvedeva et al., "Predicting Court Decisions," AI & Law, 2020.
8. VI. Katz et al., "GPT-4 Passes the Bar Exam," 2024.
9. Lewis et al., "Retrieval-Augmented Generation," NeurIPS, 2020.