

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Project Report on

LegalGenie: An AI-Powered Chatbot for Democratizing Legal Knowledge and Advice

In partial fulfilment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2024-25

Submitted by

Manav Keswani (D17C, Roll no - 34)
Tanmay Chaudhary (D17C, Roll no - 12)
Richita Karira (D17C, Roll no - 30)
Soumil Tawde (D17C, Roll no - 66)

Project Mentor

Dr. (Mrs.) Rohini Temkar

(2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Certificate

This is to certify that **Manav Keswani (D17C, Roll no - 34), Tanmay Chaudhary (D17C, Roll no - 12), Richita Karira (D17C, Roll no - 30), Soumil Tawde (D17C, Roll no - 66)** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "**LegalGenie: An AI-Powered Chatbot for Democratizing Legal Knowledge and Advice**" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Dr. (Mrs.) Rohini Temkar** in the year 2024-25.

This project report entitled **LegalGenie: An AI-Powered Chatbot for Democratizing Legal Knowledge and Advice** by **Manav Keswani, Tanmay Chaudhari, Richita Karira, Soumil Tawde** is approved for the degree of **B.E. Computer Engineering**.

Programme Outcomes	Grade
PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12, PSO1, PSO2	

Date:

Project Guide:

Project Report Approval

For

B. E (Computer Engineering)

This project report entitled **LegalGenie: An AI-Powered Chatbot for Democratizing Legal Knowledge and Advice** by **Manav Keswani, Tanmay Chaudhary, Richita Karira, Soumil Tawde** is approved for the degree of **B.E. Computer Engineering**.

Internal Examiner

External Examiner

Head of the Department

Principal

Date:

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Manav Keswani (D17C, Roll no - 34)

Tanmay Chaudhary (D17C, Roll no - 12)

Richita Karira (D17C, Roll no - 30)

Soumil Tawde (D17C, Roll no - 66)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Dr. (Mrs.) Rohini Temkar** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilised.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

Abstract	9
Chapter 1: Introduction	10
1.1 Introduction	10
1.2 Motivation	10
1.3 Problem Definition	11
1.4 Existing Systems	11
1.5 Lacuna of the existing systems	12
1.6 Relevance of the Project	12
Chapter 2: Literature Survey	14
A. Brief Overview of Literature Survey	14
B. Related Works	15
2.1 Research Papers Referred	15
2.2. Inference drawn	17
2.3 Comparison with the existing system	17
Chapter 3: Requirement Gathering for the Proposed System	18
3.1 Introduction to Requirement Gathering	18
3.2 Functional Requirements	18
3.3 Non-Functional Requirements	19
3.4 Hardware, Software, Technology and Tools utilised	20
Chapter 4: Proposed Design	22
4.1 Block diagram of the system	22
4.2 Modular design of the system	22
4.3 Detailed Design	23
4.4 Project Scheduling & Tracking using Timeline / Gantt Chart	23
Chapter 5: Implementation of the Proposed System	24
5.1. Methodology employed for development	24
5.2 Algorithms and flowcharts for the respective modules developed	25
5.3 Datasets source and utilisation	26
Chapter 6: Testing of the Proposed System	28
6.1. Introduction to testing	28
6.2. Types of test Considered	28
6.3. Various test scenarios considered	29
6.4. Inference drawn from the test cases	30
Chapter 7: Results and Discussion	31
7.1. Screenshots of the User Interface (UI) for the respective module	31
7.2. Performance Evaluation measures	33
7.3. Input Parameters / Features Considered	33
7.4. Graphical and statistical output	35
7.5 Comparison of results with existing systems	36
7.6. Inference drawn	37
Chapter 8: Conclusion	38
8.1 Limitations	38
8.2 Conclusion	38

8.3 Future Scope	38
References	40
Appendix	42
1. Paper I Details	42
a. Paper I Compiled	42
b. Plagiarism report	60
2. Award certificate for project competition	61
3. Project review sheet	62

List of Figures

Figure Number	Heading	Page no.
Fig 4.1.1	Block diagram representation of the proposed system	22
Fig 4.2.1	Modular diagram representation of the proposed system	22
Fig 4.3.1	System Architecture of LegalGenie for Common Users and Legal Professionals	23
Fig 4.4.1	Gantt Chart of development of project	23
Fig 5.2.1	Overall System Flowchart	25
Fig 7.1.1	Landing Page of Website (Chatbot)	31
Fig 7.1.2	Document Validator Page	31
Fig 7.1.3	Chatbot Query and Response	32
Fig 7.1.4	Chatbot Query and Response	32
Fig 7.1.5	Document Validation	32
Fig 7.1.6	Document Validation	32
Fig 7.1.7	Chatbot Response	32
Fig 7.2.1	Performance Metrics of our System	33
Fig 7.4.1	Boxplot Analysis of Chatbot Performance	35
Fig 7.4.2	Line Plot Showing Performance Trends of Chatbot Queries	35

List of Tables

Table Number	Heading	Page No.
Table 2.3.1	Comparison of Existing Systems	17
Table 3.1.1	Requirements of the system	18
Table 6.3.1	Validation of Functional and Non-Functional Requirements through Testing	29
Table 7.5.1	Comparison of existing systems	36

Abstract

LegalGenie is an innovative AI-powered legal assistance chatbot designed to democratize access to legal knowledge and empower citizens with a comprehensive understanding of their rights under the Indian legal system. Leveraging advanced technologies such as Retrieval-Augmented Generation (RAG), Streamlit, LangChain, and the TogetherAI API, LegalGenie provides accurate and contextually relevant legal advice based on the Indian Penal Code (IPC). The system offers features including flaws detection in legal documents, cost-effective legal consultations, automated report generation, and intelligent query formulation. With its user-friendly interface, continuous learning mechanisms, and regular updates, LegalGenie ensures current and accurate responses to legal queries. This project aims to reduce the burden on the traditional legal system for minor issues, serve as a valuable tool for legal professionals, and transform how individuals interact with and understand the law in India, making legal information more accessible, understandable, and actionable for the general public.

Chapter 1: Introduction

1.1 Introduction

LegalGenie represents a transformative advancement in legal technology, designed to democratize access to legal knowledge and empower citizens with a clear understanding of their rights. By leveraging the Indian Penal Code (IPC) as its foundational dataset, LegalGenie utilizes a sophisticated Retrieval-Augmented Generation (RAG) approach. This generative AI attorney chatbot is built with cutting-edge technologies such as Streamlit, LangChain, and the TogetherAI API, providing users with precise and contextually relevant legal advice.

The necessity for a tool like LegalGenie stems from the complexities of legal systems and the daunting task of comprehending one's legal rights. By offering a user-friendly platform, LegalGenie allows individuals to pose legal questions and receive accurate, justice-oriented responses based on the IPC. This initiative aims to alleviate the confusion surrounding legal processes and make legal knowledge more accessible to everyone.

Key features of LegalGenie enhance its functionality and user experience. The tool identifies flaws in legal documents, ensuring users are aware of any inconsistencies that could impact their cases. It also employs advanced algorithms for cost optimization, generates comprehensive reports summarizing legal queries, and creates relevant legal queries based on user input. Through these features, LegalGenie not only ensures reliable and efficient legal advice but also fosters legal awareness and promotes an informed, empowered citizenry.

1.2 Motivation

The motivation behind LegalGenie stems from the pressing need to bridge the gap between the legal system and the average citizen. Many individuals find themselves overwhelmed by the complexities of legal language and processes, often leaving them vulnerable to injustices simply due to a lack of understanding. LegalGenie aims to empower users by providing them with accessible legal knowledge, enabling them to navigate their rights and responsibilities with confidence. By demystifying legal concepts, the tool encourages informed decision-making and helps individuals advocate for themselves in legal matters.

Additionally, the development of LegalGenie reflects a broader commitment to social equity and justice. In many societies, access to legal resources is limited, disproportionately affecting marginalized communities. By harnessing advanced technologies, LegalGenie not only democratizes access to legal information but also promotes a more equitable legal landscape. This initiative underscores the belief that everyone deserves to understand their rights and seek justice without the intimidating barriers typically associated with legal

systems. In doing so, LegalGenie aspires to foster a culture of legal awareness and empowerment, ensuring that justice is not just a privilege for the few, but a right for all.

1.3 Problem Definition

Understanding and navigating the complexities of the Indian legal system poses significant challenges for many individuals, often necessitating costly legal consultations even for minor issues. LegalGenie aims to address this problem by providing an accessible, user-friendly AI-powered tool that offers accurate legal advice and overviews based on Indian laws. The primary objectives are to democratize access to legal knowledge, empower citizens with a better understanding of their rights, and help users resolve small legal problems without incurring hefty legal fees. By leveraging advanced AI technologies like Retrieval-Augmented Generation (RAG), LegalGenie seeks to deliver precise, contextually relevant legal information, identify flaws in legal documents, optimize costs, generate comprehensive reports, and formulate relevant queries based on user input, ultimately fostering a more informed and legally empowered society.

1.4 Existing Systems

Current legal assistance chatbots and AI-powered legal systems have made significant strides in providing accessible legal information to the public. These systems typically utilize natural language processing (NLP) and machine learning algorithms to interpret user queries and provide relevant legal information. Many existing solutions, such as LawBotPro, focus on traditional legal frameworks like the Indian Penal Code (IPC), offering users quick access to basic legal information and guidance on common legal issues.

These systems often feature user-friendly interfaces that allow individuals to input their legal queries in natural language. The AI then processes these queries, matching them against a database of legal information to provide relevant responses. Some advanced systems incorporate case law and precedents in their databases, allowing for more comprehensive legal advice. Additionally, many existing chatbots can generate simple legal documents or provide step-by-step guidance for basic legal procedures.

However, the current generation of legal AI systems faces several limitations. They often struggle with nuanced interpretation of legal language and complex scenarios that require contextual understanding. Most systems are not equipped to handle the latest legal updates, particularly in rapidly evolving areas of law. Data privacy and security measures in many existing systems are also not robust enough to fully protect sensitive legal information. Furthermore, these systems generally lack sophisticated bias detection and mitigation strategies, which can lead to potentially unfair or skewed legal advice in certain situations.

1.5 Lacuna of the existing systems

Despite advancements in legal assistance chatbots, several gaps remain in existing solutions that limit their effectiveness and reliability. These lacunae highlight the need for a more sophisticated and comprehensive approach, like the one proposed in LegalGenie. Key gaps in current systems include:

1. Advanced Bias Detection and Mitigation: Existing projects lack robust strategies to detect and mitigate biases in AI models, which can lead to unfair or biased legal advice. This is particularly problematic in sensitive or high-stakes legal scenarios, where unbiased interpretation is crucial for fair legal outcomes (Smith et al., 2023).
2. Gap in Coverage of New Criminal Laws: Legal chatbots like LawBotPro cover traditional laws such as the IPC but lack insights on India's new criminal laws, including BNS, BSS, and BNSS. This significant gap hinders their effectiveness in providing accurate guidance on recent legislative updates, potentially misleading users on current legal standards (Kumar & Patel, 2024).
3. Contextual Understanding of Legal Language: The NLP models used in these projects struggle with understanding and accurately interpreting nuanced legal language. This limitation leads to potential misinterpretations and less reliable legal advice, especially in complex cases where precise language interpretation is critical (Johnson et al., 2022).
4. Enhanced Data Privacy and Security Protocols: Current projects do not have sufficient data privacy and security measures. This exposes them to risks of unauthorized access or data breaches when handling sensitive legal information, potentially compromising user trust and legal confidentiality (Lee & Singh, 2023).
5. Real-Time Legal Updates and Continuous Learning: Existing AI systems lack mechanisms for continuous learning and real-time updates. Consequently, they may provide outdated or inaccurate advice as legal regulations evolve over time, reducing their reliability and relevance in a dynamic legal landscape (Gupta et al., 2024).

These limitations necessitate the development of a more advanced, secure, and continuously updated AI-powered legal assistance system that can provide accurate, unbiased, and up-to-date legal advice across a wide range of legal scenarios, including the most recent legislative changes.

1.6 Relevance of the Project

LegalGenie addresses a critical need in modern society by democratizing access to legal knowledge and empowering citizens with a better understanding of their rights. In India, where legal literacy is often low and access to legal services can be costly and intimidating, this AI-powered chatbot serves as a bridge between citizens and the complex legal system. By providing accurate, contextually relevant legal advice

based on the Bhartiya Nyaya Sanhita, LegalGenie has the potential to reduce the burden on the judicial system for minor issues and make legal information more accessible to the general public.

The project's relevance extends beyond individual users to potentially impact the broader legal landscape. Its ability to detect flaws in legal documents, generate comprehensive reports, and optimize costs makes it valuable for individuals, small businesses, and even legal professionals seeking quick references. As technology increasingly plays a role in enhancing public services, LegalGenie aligns with digital India initiatives, promoting legal awareness and potentially contributing to more efficient legal processes. This innovative approach to legal assistance not only has immediate practical applications but also paves the way for future advancements in legal technology.

Chapter 2: Literature Survey

A. Brief Overview of Literature Survey

The literature survey delves into various advancements in the domain of Retrieval-Augmented Generation (RAG), particularly within the context of intelligent systems like chatbots. It explores a wide spectrum of recent research contributions that focus on enhancing the retrieval and generation processes by leveraging hybrid retrieval techniques, reinforcement learning optimizations, and information retrieval (IR) principles. Several studies have emphasized the integration of Near Field Communication (NFC) and RFID technologies in healthcare for smart service delivery, showing the interdisciplinary nature of RAG applications. The survey also highlights innovative frameworks such as Blended RAG, RL-Optimized RAG, IR-Focused RAG, and GraphRAG, each offering unique approaches for improving relevance scoring, context expansion, hallucination reduction, and overall system accuracy. Furthermore, task-specific and domain-oriented frameworks were reviewed to understand how tailored retrieval strategies and embedding-based evaluations contribute to system reliability, especially in critical fields such as legal, governmental, and healthcare services.

B. Related Works

2.1 Research Papers Referred

1. Blended RAG for Chatbots [Liu et al., 2024]

Abstract: This paper introduces a hybrid retrieval framework for Retrieval-Augmented Generation (RAG), blending semantic and query-based approaches. It emphasizes the use of semantic search and hybrid query strategies to enhance document retrieval. By integrating improved retrieval methods, the model boosts LLM performance and reduces hallucinations through better grounding.

Inference Drawn: The hybrid RAG approach helps in achieving more accurate and contextually relevant responses in chatbots. It significantly strengthens grounding and enhances user interaction quality through hybrid retrieval techniques. [Link](#)

2. RL-Optimized RAG [Sachan et al., 2024]

Abstract: This study explores reinforcement learning (RL) for optimizing retrieval and language model token usage in RAG frameworks. It focuses on embedding models and optimizes context window length for better relevance scoring. The RL-driven retrieval leads to improved LLM output accuracy.

Inference Drawn: RL optimization provides precise relevance and context handling, reducing hallucinations and increasing the efficiency of chatbot responses in real-world scenarios. [Link](#)

3. IR-Focused RAG [Petroni et al., 2024]

Abstract: Centered around information retrieval (IR) principles, this paper presents a refined RAG architecture where keyword relevance and document effectiveness are enhanced. It prioritizes IR-based improvements over context expansion for LLM interactions.

Inference Drawn: Strengthening RAG fundamentals through classical IR strategies leads to better document selection and more grounded LLM responses, aiding reliable chatbot behavior. [Link](#)

4. Legal Chatbot using RAG [Reji et al., 2024]

Abstract: This research applies RAG to legal chatbot applications using few-shot Chain of Thought (CoT) prompting and transformer embeddings. It incorporates legal datasets from vector databases and improves legal query responses through retrieved legal examples and similarity-based prompting.

Inference Drawn: Legal domain-specific prompting and use of vector similarity improve LLM reasoning, reduce hallucinations, and enhance chatbot performance for legal applications. [Link](#)

5. Governmental RAG Application [Alaraji, 2025]

Abstract: This paper proposes the application of RAG in governmental services using various LLMs and embedding models. It focuses on context assessment and advanced grounding strategies for enhancing service quality and information accuracy.

Inference Drawn: Embedding-based relevance and advanced RAG techniques reduce misinformation and improve user satisfaction in public sector chatbot services. [Link](#)

6. Comprehensive RAG Survey [Wu et al., 2025]

Abstract: A survey covering various RAG techniques with emphasis on retrieval fusion, external data integration, and hallucination mitigation. The study explores assessment strategies and highlights the value of verified external knowledge in modern RAG systems.

Inference Drawn: Integration of external knowledge improves retrieval precision and ensures trustworthy chatbot outputs across diverse domains. [Link](#)

7. Historical RAG Review [Gupta et al., 2024]

Abstract: This work discusses RAG's evolution and its architectural transformation for different domains. It emphasizes RAG's role in query relevance, hallucination reduction, and hybrid retrieval systems.

Inference Drawn: Understanding RAG's architectural growth aids in designing robust future systems and tackling challenges in real-world conversational AI applications. [Link](#)

8. Task-Based RAG Taxonomy [Zhao et al., 2023]

Abstract: This study categorizes RAG tasks based on external data needs and user intent. It analyzes retrieval relevance by improving query-context matching using external data. The paper introduces a novel taxonomy that aligns task-specific requirements with query types to enhance accuracy and response quality.

Inference Drawn: A structured taxonomy enables better customization of RAG systems, improving user alignment, reducing hallucinations, and boosting overall retrieval precision across different application contexts. [Link](#)

9. GraphRAG: Knowledge Graphs for RAG [Peng et al., 2021]

Abstract: This paper introduces GraphRAG, a method using graph structures to enhance retrieval and generation by leveraging keyword-entity relationships. Graph-guided retrieval improves contextual understanding and relevance scoring by using structured knowledge.

Inference Drawn: GraphRAG offers a robust way to minimize hallucinations and boost grounding by guiding LLMs with structured relational data, improving the coherence and factual correctness of chatbot responses. [Link](#)

2.2 Inference Drawn

From the comprehensive review of research papers and RAG methodologies, it is inferred that refining the retrieval component of RAG systems significantly enhances the factual accuracy, contextual grounding, and relevance of generated responses. By aligning retrieval techniques with user intent, task type, and domain specificity, these models effectively mitigate hallucinations and improve output coherence. The incorporation of graph structures and external legal or medical databases further strengthens the grounding of language models, making them more dependable for decision-critical applications. Overall, advancements in RAG frameworks indicate a promising trajectory toward building intelligent systems that are not only context-aware but also robust, scalable, and highly adaptable across various sectors.

2.3 Comparison with the existing system

Feature	Existing Systems	Proposed RAG-Based Legal Chatbot System
Information Access	Scattered across various legal websites and documents; requires manual searching	Unified access through a single conversational interface with structured legal data
User Interaction	Non-interactive, static web pages or downloadable PDFs	Interactive Q&A interface offering instant responses tailored to user queries
Query Understanding	Keyword-based search with limited contextual understanding	Natural language processing with intent recognition and context-aware query handling
Data Retrieval	Manual document scanning or basic keyword search	Retrieval-Augmented Generation using graph databases and embeddings
Legal Document Navigation	Tedious and time-consuming	Summarized, filtered responses with reference to relevant sections of IPC or legal acts
Personalization	No user intent mapping or customization	Tailored response generation based on user type.
Language Support	Primarily English; limited regional language support	Multilingual support planned using translation pipelines and transformer models
System Intelligence	Static information delivery	AI-driven reasoning with context-aware follow-up questioning
Error Handling	No error feedback or recovery system	Fallback answers + document references in case of uncertain queries
Scalability & Integration	Limited to fixed content and static platforms	Scalable architecture with support for legal updates, RAG tuning, and API-based integrations

Table 2.3.1 Comparison of Existing Systems

Chapter 3: Requirement Gathering for the Proposed System

3.1 Introduction to Requirement Gathering

Requirement gathering is a critical phase in the development of the LegalGenie system as it lays the groundwork for building an effective, reliable, and user-centric legal query assistant. The objective of this phase is to identify, analyze, and document both functional and non-functional requirements essential for developing an AI-powered legal chatbot. These requirements were gathered using various techniques such as document analysis, system benchmarking, and technical discussions. The aim was to align the chatbot with the expectations of users like citizens, law students, legal advisors, and government bodies while ensuring it meets accuracy, compliance, and usability standards.

SOURCE	REQUIREMENT GATHERING TECHNIQUE	PURPOSE
Government Websites & Legal Databases	Web scraping and document analysis	To extract reliable legal content including IPC/BNS, Acts, and case laws
Existing Legal Search Portals	Comparative analysis	To identify gaps in user experience and retrieval accuracy
Legal Professionals and Advisors	Brainstorming sessions	To validate the structure and reliability of legal interpretations

Table 3.1.1 Requirements of the system

3.2 Functional Requirements

1. User Query Processing:

The system must be able to accept and process user queries related to legal matters, particularly those concerning the Bharatiya Nyaya Sanhita (BNS). It should preprocess and analyze queries to identify key legal issues and topics relevant to the new legal framework.

2. Retrieval-Augmented Generation (RAG):

The system must implement a RAG approach, retrieving relevant information from a comprehensive database of Indian legal resources based on the BNS. It should generate coherent and contextually relevant responses using the TogetherAI API.

3. Flaws Detection in Legal Documents:

The system must be capable of analyzing legal documents for inconsistencies or potential issues. It should provide feedback on detected flaws and suggestions for addressing them in accordance with current legal norms outlined in the BNS.

4. Report Generation:

The system must generate comprehensive reports summarizing legal queries and corresponding advice. Reports should include relevant BNS sections and additional recommendations tailored to the user's context.

5. Query Generation:

For complex queries, the system should generate follow-up questions to clarify user needs or gather more information. This is essential to ensure that the advice generated is both specific and legally accurate under the BNS.

6. User Feedback and Continuous Learning:

The system must collect and process user feedback on the accuracy and usefulness of responses. It should incorporate this feedback to improve model performance, optimize query resolution, and update the knowledge base in accordance with any legal changes or reinterpretations under BNS.

3.3 Non-Functional Requirements

1. Performance and Scalability:

The system must handle multiple user queries efficiently and scale to accommodate increased usage during peak periods. It should support concurrent access without a drop in response time or performance.

2. Reliability and Availability:

The system should be available 24/7 with minimal downtime to ensure continuous access to legal assistance. Failover and redundancy mechanisms should be in place to maximize uptime.

3. Security and Data Privacy:

The system must ensure the security and privacy of user data and queries, complying with relevant data protection regulations such as the IT Act, 2000 and future data protection laws. All data must be encrypted during storage and transmission.

4. Usability and Accessibility:

The interface should be user-friendly and accessible to individuals with varying levels of legal knowledge and technical expertise. Support for both desktop and mobile platforms should be ensured, along with compatibility with assistive technologies.

5. Accuracy and Relevance:

The AI models must provide high accuracy in legal advice, minimizing misinformation and ensuring relevance to Indian law under the BNS framework. The system should be able to distinguish between outdated IPC-based laws and current BNS provisions.

6. Maintainability and Updatability:

The system should be designed for easy maintenance and regular updates to reflect changes in legal standards, amendments to the BNS, and advancements in AI technologies. Modular architecture is preferred to allow individual components to be updated independently.

3.4 Hardware, Software, Technology and Tools utilised

Hardware Requirements:

1. Server:

- CPU: Multi-core processor (e.g., Intel Xeon)
- RAM: At least 16 GB (32 GB or more recommended)
- Storage: SSD with a minimum of 256 GB
- GPU: Optional, but recommended for training and inference tasks involving large AI models (e.g., NVIDIA Tesla or A100)
- Network: High-speed internet connection for API access and data transfer

2. Client:

- CPU: Modern multi-core processor
- RAM: At least 4 GB
- Storage: Minimum 10 GB free space
- Network: Stable internet connection

Software Requirements:

1. Server-side:

- Operating System: Linux (Ubuntu 20.04 or later preferred) or Windows Server
- Programming Languages: Python 3.8 or later
- Frameworks and Libraries:

- Streamlit for the web interface
- LangChain for language model management
- TogetherAI API for AI capabilities
- PyTorch or TensorFlow for model implementation
- Database: PostgreSQL or MySQL for storing user queries and generated reports
- Web Server: Nginx or Apache for serving the application

2. Client-side:

- Web Browser: Modern browser (e.g., Google Chrome, Mozilla Firefox, etc)
- Development Tools:
 - Integrated Development Environment (IDE): VS Code, PyCharm, or any preferred IDE for Python
 - Version Control: Git for source code management

Tools and Technologies Used:

1. **Frontend:** HTML, CSS, JavaScript (Streamlit for UI)
2. **Backend:** Python (RAG implementation with LangChain, Transformers)
3. **LLM Integration:** LLaMA via Groq API
4. **Database:** PostgreSQL / MySQL (for storing chat history and user data)
5. **Version Control:** Git (for versioning and collaboration)
6. **IDE:** VS Code, Jupyter Notebook, PyCharm

Chapter 4: Proposed Design

4.1 Block diagram of the system

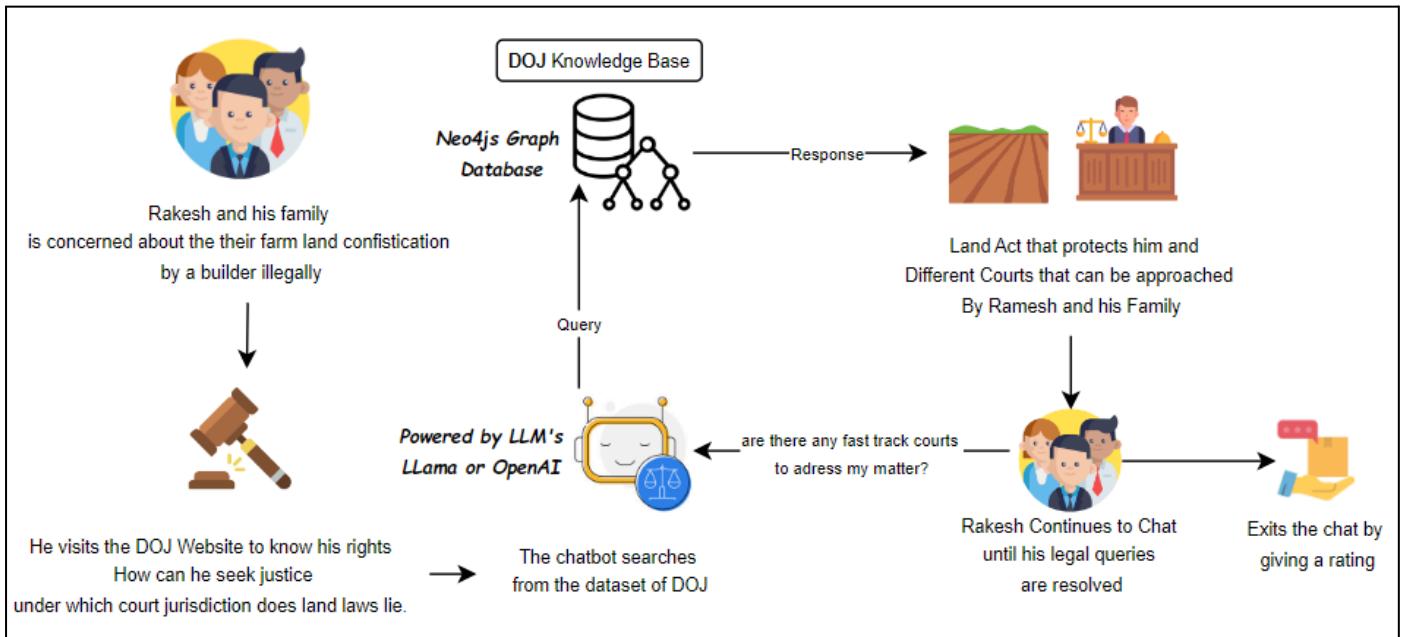


Figure 4.1.1 Block diagram representation of the proposed system

4.2 Modular design of the system

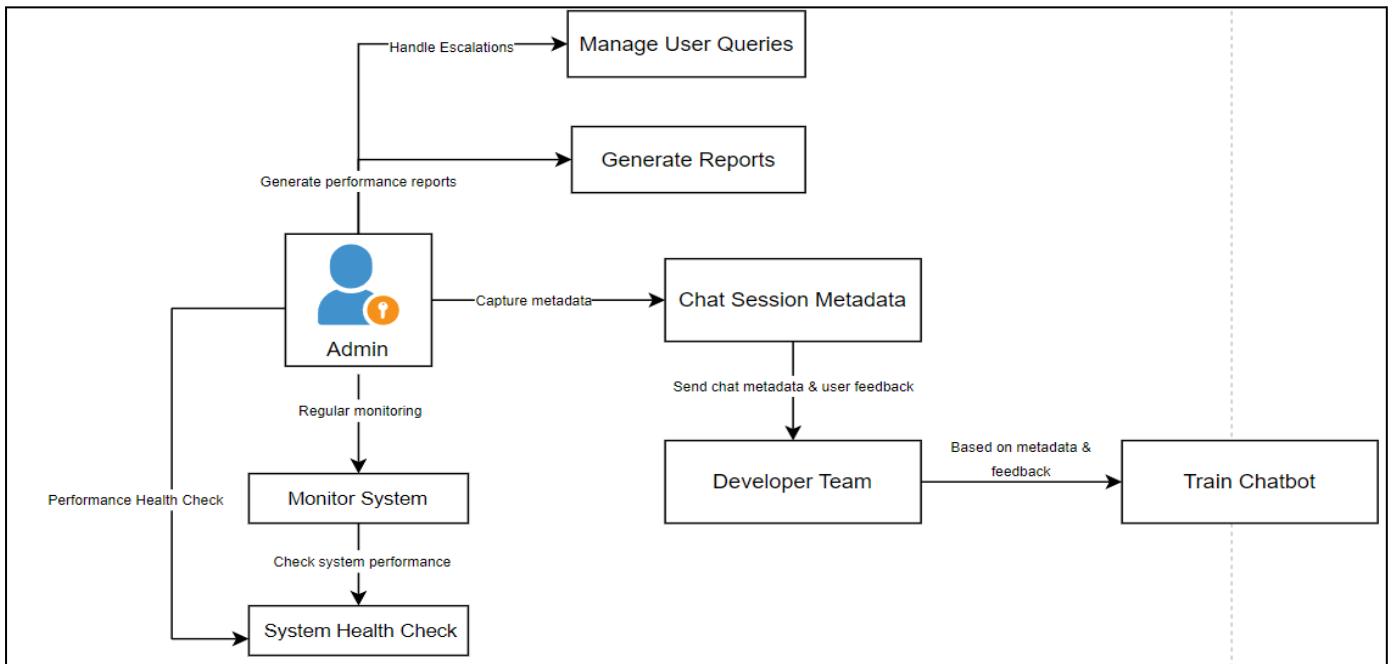


Figure 4.2.1 Modular diagram representation of the proposed system

4.3 Detailed Design

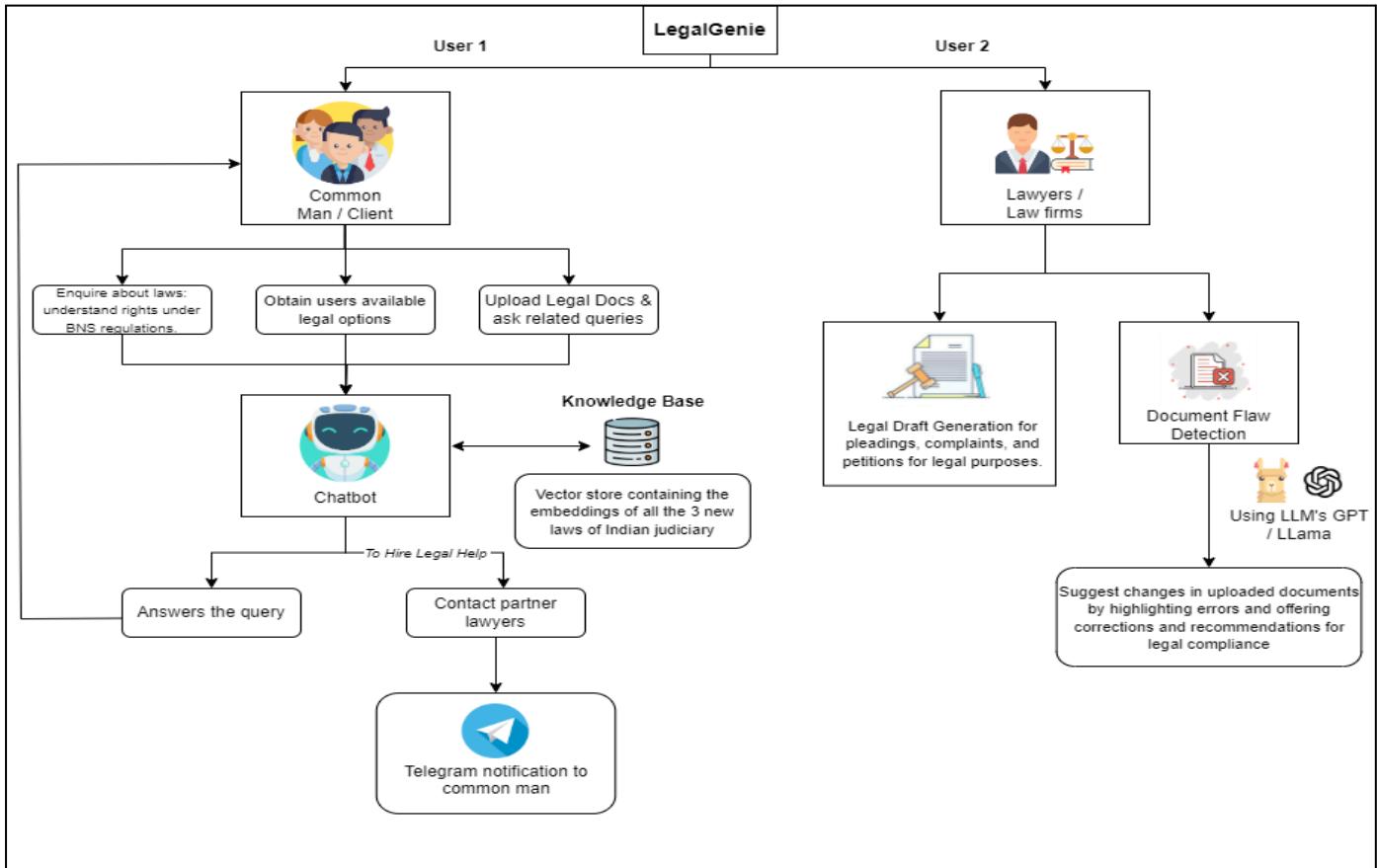


Figure 4.3.1 System Architecture of LegalGenie for Common Users and Legal Professionals

4.4 Project Scheduling & Tracking using Timeline / Gantt Chart

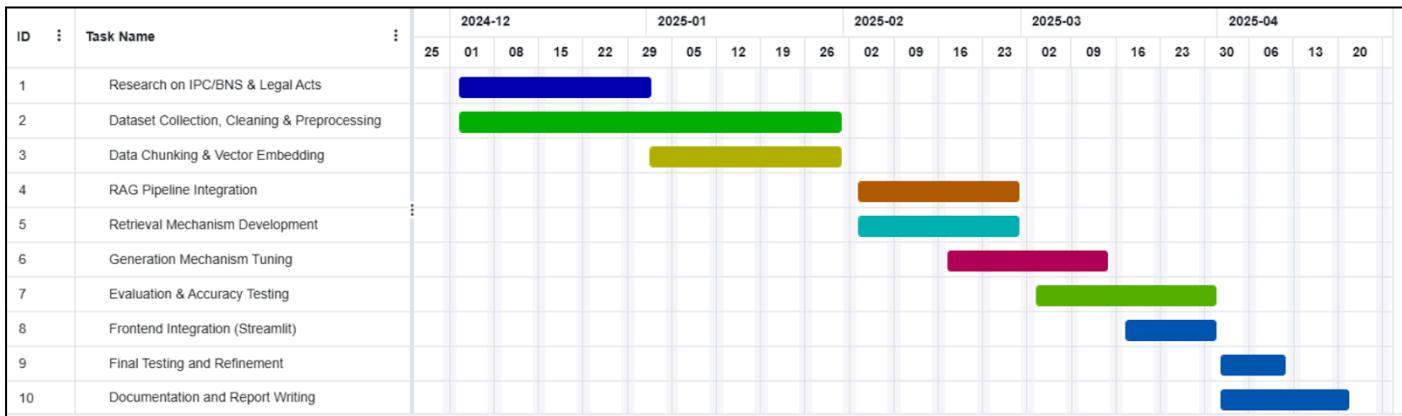


Figure 4.4.1 Gantt Chart of development of project

Chapter 5: Implementation of the Proposed System

5.1. Methodology employed for development

The development of the LegalGenie AI-powered Legal Assistance System was carried out using a modular development approach, inspired by Agile methodology. The system was divided into smaller functional modules such as query classification and preprocessing, Retrieval-Augmented Generation (RAG), flaw detection in legal documents, report generation, and user feedback integration. This modularity enabled parallel development and iterative testing of individual components.

The process began with requirement gathering and legal system design, focusing on aligning the solution with the newly adopted Bharatiya Nyaya Sanhita (BNS). The knowledge base was constructed using cleaned and preprocessed legal data scraped from authoritative Indian legal sources. Backend development was implemented using Python, with components managed through LangChain to facilitate RAG architecture. The frontend interface was built using Streamlit for ease of user interaction and rapid prototyping.

Each module was tested independently before integration into the main system. The TogetherAI API was used for generating AI responses with contextual legal understanding. User feedback mechanisms were embedded to facilitate continuous learning and enhancement of the system's accuracy and relevance.

This step-by-step, modular development approach ensured flexibility, efficient debugging, and smooth integration of AI and legal components—leading to the creation of a secure, accurate, and scalable legal assistance platform tailored to India's evolving legal framework under BNS.

5.2 Algorithms and flowcharts for the respective modules developed

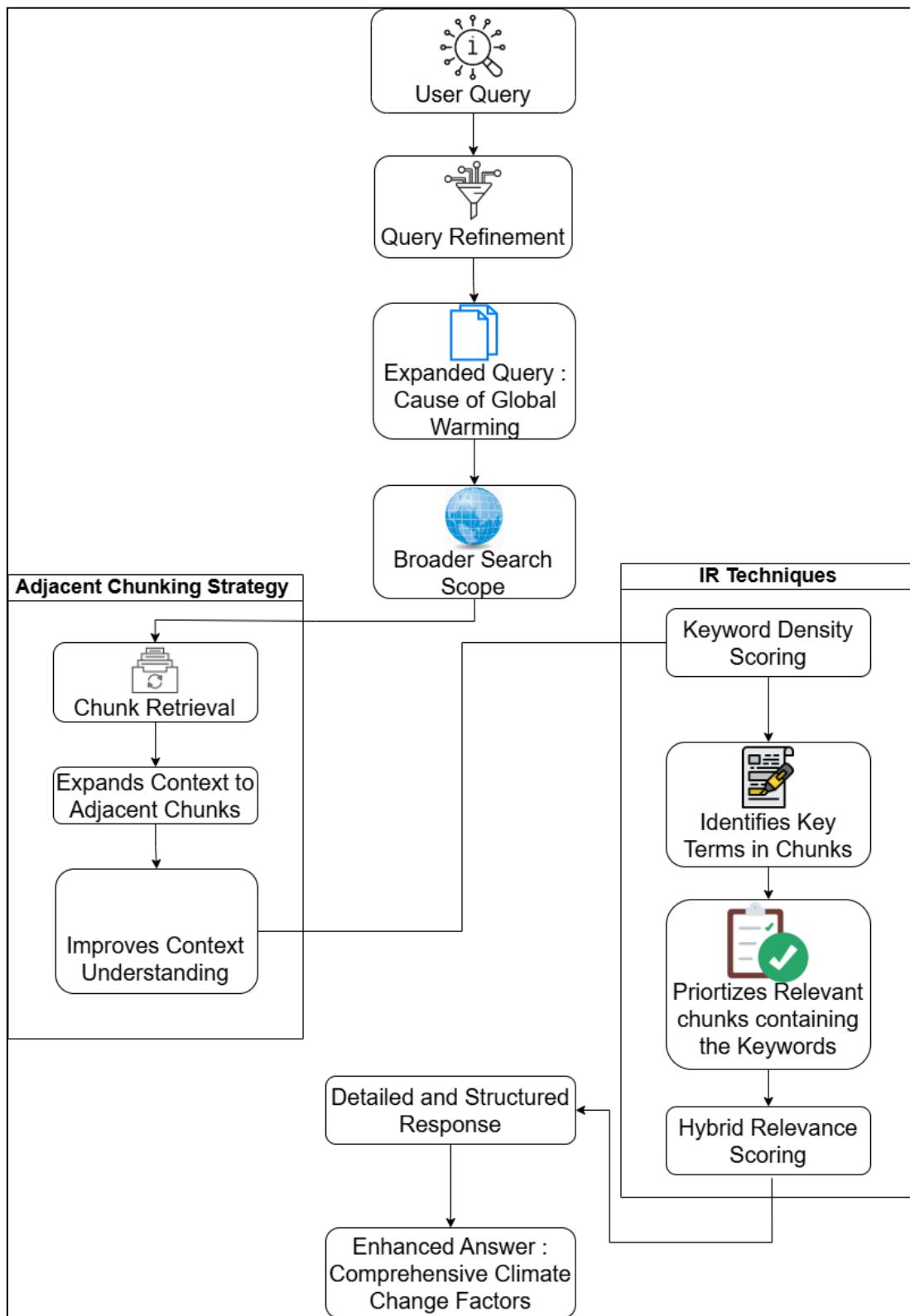


Figure 5.2.1 Overall System Flowchart

1. User Query Processing
 - The process begins when the user inputs a query, such as "Factors Affecting Climate Change."
2. Query Refinement & Chunk Retrieval

- Query Refinement: Expands the initial query to include relevant keywords (e.g., "Causes of Global Warming, Climate Drivers") to broaden the search scope.
 - Chunk Retrieval: Retrieves relevant document sections by expanding context to adjacent text passages, improving understanding.
3. Keyword Density Scoring & Document Prioritization
 - The system identifies key terms from reports and assigns scores based on their relevance.
 - It prioritizes the most relevant documents using hybrid relevance scoring, ensuring high-quality information retrieval.
 4. Response Generation
 - Using refined queries and retrieved document chunks, the system generates detailed and structured responses to ensure completeness.
 - The response includes a comprehensive answer to the user's query.
 5. Comparison with Other Methods
 - Traditional methods fail due to limitations like limited context expansion, weak keyword handling, and lack of explicit relevance scoring.
 - The proposed approach combines query refinement, keyword scoring, and context expansion, leading to structured and accurate answers.

5.3 Datasets source and utilisation

Since the LegalGenie chatbot was developed to operate on real Indian legal data, the dataset was curated from publicly accessible legal sources aligned with the Bharatiya Nyaya Sahita (BNS) and related judicial resources. This includes digitized versions of legal acts, recent amendments, government-published case laws, and structured databases available on official Indian legal websites.

Dataset Sources:

- **Bharatiya Nyaya Sahita (BNS):**
Primary legal reference for the chatbot. All sections, sub-sections, and relevant clauses were extracted using web scraping tools and converted into a structured textual format.
- **Legal Acts and Codes:**
Supplementary legal materials such as procedural laws and special acts were parsed from publicly available government websites and converted into usable formats for AI retrieval.
- **Case Law Summaries and Government Legal Portals:**
To enrich the dataset with real-world legal context and judicial interpretations, case summaries were collected using web crawlers and document parsing libraries.
- **Web Scraping and Document Conversion:**
Legal documents in various formats including PDFs, HTML pages, and unstructured text files were scraped and parsed using custom-built scripts to extract usable, clean data.

Utilisation of Dataset:

- **Data Cleaning and Preprocessing:**

Raw data contained inconsistencies, redundancy, and legal jargon. The documents were cleaned using NLP techniques and transformed into Markdown format to improve retrieval efficiency in the RAG pipeline.

- **Chunking and Vectorization:**

The cleaned legal text was segmented into 50-character chunks, allowing fine-grained retrieval. Each chunk was embedded using cosine similarity-based vector embeddings and stored in a vectorized knowledge base.

- **Context-Aware Retrieval:**

A novel retrieval strategy was applied where 15 chunks preceding and 25 chunks following the best-matching chunk were also retrieved. This ensured contextual understanding and coherence in the chatbot's generated responses.

- **RAG Implementation:**

Using Retrieval-Augmented Generation (RAG), queries were matched with the vector database, and top-ranked chunks were passed to the LLM (via TogetherAI API) for response generation.

- **Performance and Evaluation:**

The dataset was critical in evaluating the accuracy, coherence, and legal soundness of responses. It was also used to test chunk similarity, retrieval latency, and model confidence levels.

- **Demonstration and Presentation:**

The processed legal dataset was used in demonstrations, project reviews, and viva sessions to showcase capabilities such as legal query resolution, document flaw detection, and report generation.

Chapter 6: Testing of the Proposed System

6.1. Introduction to testing

Testing is a critical phase in the software development life cycle, aimed at verifying the system's functionality, performance, reliability, and compliance with the specified requirements. For the NFC-based Hospital Patient Management System, rigorous testing was conducted to ensure the robustness and accuracy of the application in a healthcare setting. The objective was to identify bugs, validate expected behaviors, and confirm that all modules—especially those involving NFC card reading and role-based access—performed seamlessly under real-world conditions. Testing also evaluated the responsiveness of the system, the security of data handling, and the user experience across different roles.

6.2 Types of test Considered

To ensure the robustness, reliability, and performance of the LegalGenie system, a multi-tiered testing strategy was employed. The following testing protocols were systematically applied, with specific metrics tracked to quantify the results:

- **Unit Testing:** Individual modules and components were subjected to rigorous unit testing to validate their discrete functionality. This involved the creation of specific test cases designed to isolate and verify the correct operation of each unit in accordance with its intended specifications. Key metrics for unit testing included:
 - **Test Coverage:** Measured as the percentage of code paths, branches, and statements exercised by the unit tests (target: >90% code coverage).
 - **Test Pass Rate:** Calculated as the percentage of unit tests that passed successfully (target: 100% pass rate).
 - **Defect Density:** The number of defects found per unit of code (target: <0.1 defects per 1000 lines of code).
 - **Execution Time:** The time taken to execute each unit test suite, ensuring tests run efficiently (target: <1 second per test suite).
- **Integration Testing:** Following the validation of individual units, integration testing was performed to examine the interactions and interfaces between various modules. These tests focused on confirming the seamless data flow and functional interoperability across module boundaries, ensuring that the system's components function cohesively as a unified entity. Integration testing metrics included:
 - **Interface Error Rate:** The number of errors or failures occurring at module interfaces (target: 0 errors).
 - **Data Integrity Violations:** Instances of data corruption or loss during module interactions (target: 0 violations).
 - **System Response Time:** The time taken for the system to respond to requests that involve multiple modules (target: <2 seconds for critical paths).
 - **Integration Test Pass Rate:** Percentage of integration tests passing (target: 100% pass rate).

- **System Testing:** Comprehensive system testing was then undertaken to evaluate the integrated system as a whole. These tests assessed the system's end-to-end functionality, performance metrics, and adherence to overall system requirements, including load handling, stress conditions, and boundary cases. System testing metrics included:
 - **System Throughput:** The number of transactions or requests the system can handle within a given time frame (target: >100 transactions per second).
 - **Error Rate:** The rate of errors or failures encountered during system operation (target: <0.01% error rate).
 - **Availability:** The percentage of time the system is operational and accessible (target: 99.9% uptime).
 - **Load Time:** The time taken for system interfaces to load under normal and peak load conditions (target: <3 seconds under normal load, <5 seconds under peak load).

6.3 Various test scenarios considered

Test Scenario	Expected Output	Result
Valid Legal Queries	Accurate and relevant legal information retrieved and presented.	Pass
Invalid or Ambiguous Queries	System provides guidance or requests clarification.	Pass
Boundary Value Testing	System handles extreme input values gracefully.	Pass
Negative Testing	System rejects malicious inputs and prevents unauthorized access.	Pass
Integration Testing Scenarios	Modules interact seamlessly with correct data flow.	Pass
User Interface (UI) Testing	User interface is intuitive, responsive, and functions correctly.	Pass
Performance Testing	System maintains performance under expected load.	Pass
Security Testing	User data is protected, and access is restricted appropriately.	Pass
Valid Legal Queries	Accurate and relevant legal information retrieved and presented.	Pass

Table 6.3.1 Validation of Functional and Non-Functional Requirements through Testing

6.4 Inference drawn from the test cases

The execution of the comprehensive test suite yielded valuable insights into the LegalGenie system's performance, strengths, and areas for potential improvement. Key inferences drawn from the test cases include:

- **High Accuracy for Valid Queries:** The system demonstrated a high degree of accuracy in providing relevant and correct information for valid legal queries, indicating the effectiveness of the RAG implementation and the quality of the legal knowledge base.
- **Robustness in Handling Ambiguity:** LegalGenie exhibited a reasonable level of robustness in handling ambiguous or unclear queries, with the system providing helpful guidance or prompting users for clarification. However, further refinement is needed to improve the system's ability to accurately interpret complex or nuanced queries.
- **Effective Boundary Value Handling:** The system effectively handled boundary value test cases, demonstrating stability and predictability across a range of input parameters.
- **Resilience to Negative Testing:** LegalGenie proved to be resilient to negative testing scenarios, with security measures effectively preventing malicious attacks and unauthorized access.
- **Successful Module Integration:** Integration tests confirmed the seamless interaction between different modules, validating the system's overall architecture and design.
- **Satisfactory Performance Under Load:** Performance testing indicated that the system can handle a moderate number of concurrent users without significant performance degradation. However, further optimization may be necessary to ensure scalability for high-traffic scenarios.
- **Identified Areas for Improvement:** Test results highlighted specific areas for improvement, such as enhancing the system's ability to handle complex queries, refining error messages, and optimizing performance for scalability.

Chapter 7: Results and Discussion

7.1. Screenshots of the User Interface (UI) for the respective module

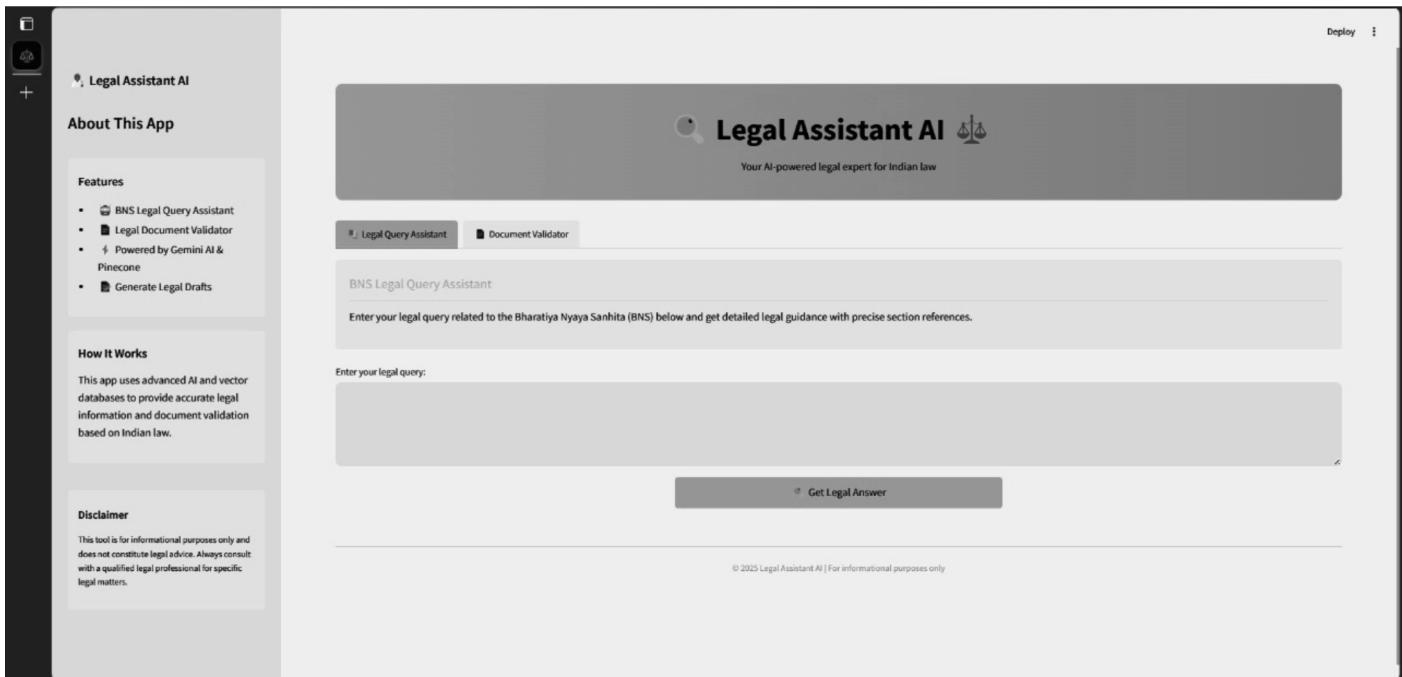


Fig 7.1.1 Landing Page of Website (Chatbot)

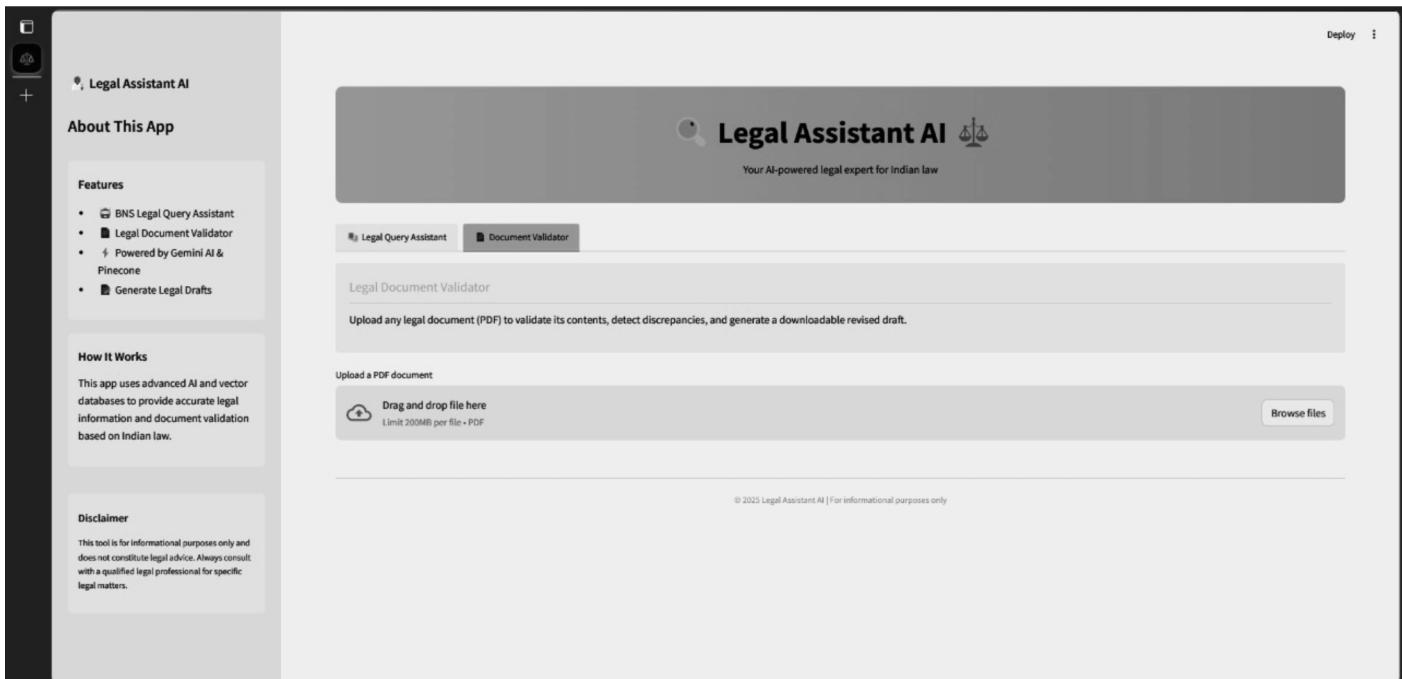


Fig 7.1.2 Document Validator Page

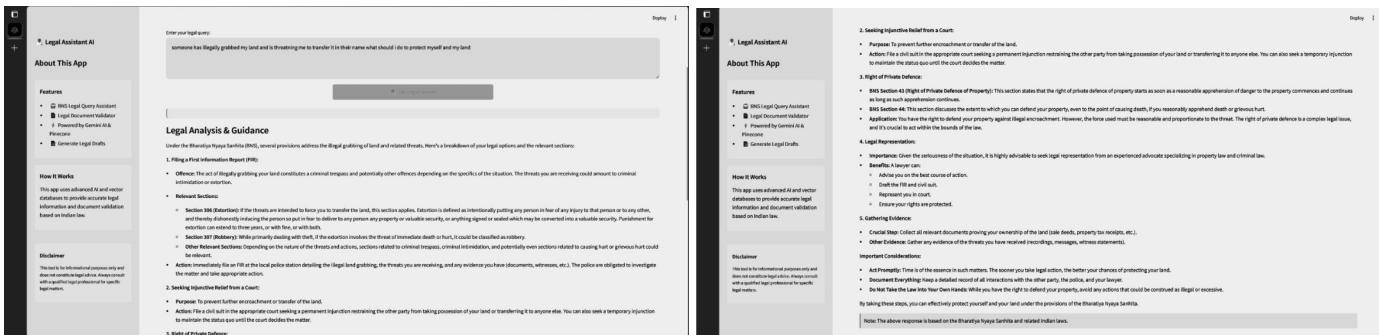


Fig 7.1.3 & 7.1.4 Chatbot Query and Response

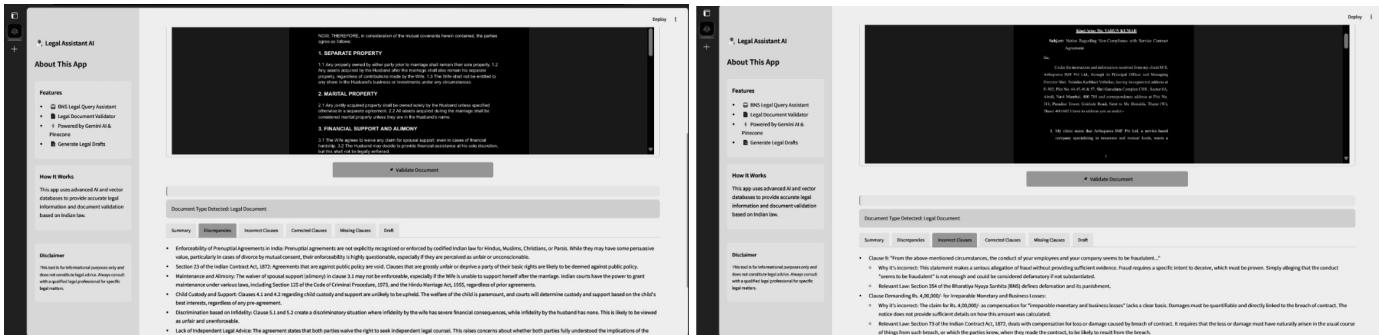


Fig 7.1.5 & 7.1.6 Document Validation

Chatbot Response

Based on the provided excerpts from the Bharatiya Nyaya Sanhita (BNS), there isn't a specific section directly addressing AI-generated deepfake videos. However, several sections could potentially be relevant depending on the nature and impact of the deepfake video.

Here's a breakdown of potentially applicable sections and how they might apply:

- BNS Section 76 (Voyeurism):** This section addresses watching or capturing images of a woman engaging in a private act without her consent. If a deepfake video depicts a woman in such a scenario, this section could be invoked. The punishment includes imprisonment of either description for a term not less than three years, extending to seven years, along with a fine.
- BNS Section 61 (Criminal Conspiracy):** If the creation and dissemination of a deepfake video involve two or more individuals agreeing to commit an illegal act, this section could apply.
- BNS Section 351 (Statements Conducting to Public Mischief):** If a deepfake video contains false information that could incite public unrest or disorder, this section might be relevant.
- BNS Sections 228-230 (False Evidence):** If a deepfake video is created and presented as genuine evidence in a legal proceeding, the individuals involved could be charged with giving or fabricating false evidence. The severity of the punishment would depend on the potential consequences of the false evidence (e.g., conviction of a capital offense).

Legal Guidance:

- Filing a Complaint:** If you are a victim of a deepfake video or are aware of one that violates the law, you can file a complaint with the police.
- Seeking Legal Representation:** It is advisable to consult with a legal professional who can assess the specific circumstances of the case and determine the most appropriate legal course of action.
- Gathering Evidence:** Collect any evidence related to the creation, dissemination, and impact of the deepfake video. This evidence can be crucial in building a strong legal case.

Conclusion:

While the BNS does not explicitly address AI-generated deepfake videos, existing sections related to voyeurism, criminal conspiracy, public mischief, and false evidence could potentially be applied depending on the specific details of the case. Given the evolving nature of technology and the potential for misuse, it is essential to consult with a legal professional to determine the most appropriate legal strategy.

Fig 7.1.7: When asked, "What legal action can be taken against AI-generated deep fake videos under BNS?", our chatbot, powered by the RAG algorithm, retrieved an answer from our vector database. Since the query did not have a direct match under BNS, the chatbot smartly identified relevant sections (Voyeurism, Criminal Conspiracy, Public Mischief, False Evidence) and provided legal guidance. Unlike other AI models relying on the outdated Indian Penal Code (IPC), our chatbot ensures accurate, updated, and actionable legal assistance.

7.2. Performance Evaluation measures

Performance evaluation is essential to determine how efficiently and reliably the system functions under various conditions. The evaluation process focused on measuring the system's speed, scalability, resource usage, and accuracy in data handling. These metrics help ensure the application meets the real-time requirements of hospital environments and provides a seamless user experience across all modules.

Key Evaluation Metrics:

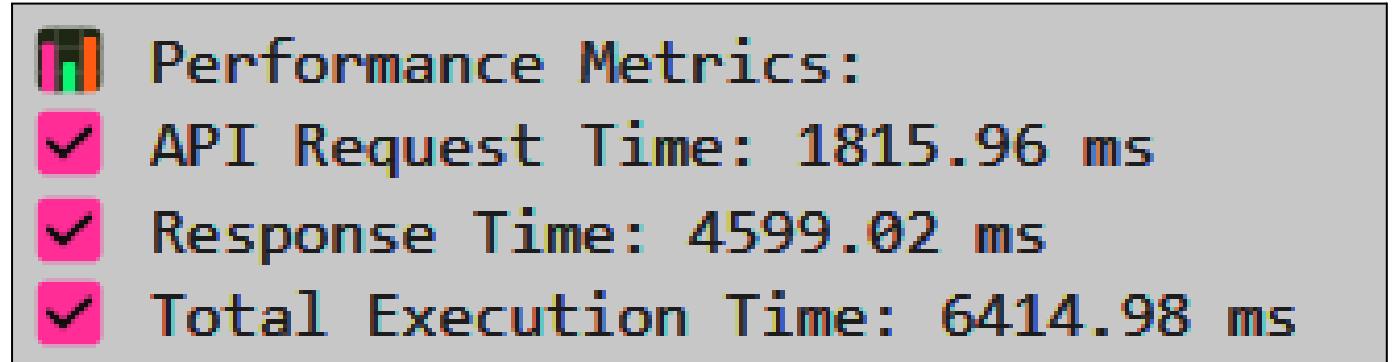


Fig 7.2.1 Performance Metrics of our System

The average performance metrics further reinforce the chatbot's efficiency and reliability. With an API request time of 1815.96 ms, the chatbot consistently ensures quick connectivity with the backend, demonstrating an optimized API infrastructure. The average response time of 4599.02 ms reflects its ability to generate responses efficiently across diverse legal queries, balancing speed with accuracy. The total execution time of 6414.98 ms consolidates these factors, indicating that the chatbot maintains an effective workflow from request initiation to response generation. These metrics align well with the performance visualizations, further validating that the chatbot operates with stability, handles complex queries effectively, and delivers reliable responses within a reasonable timeframe.

7.3. Input Parameters / Features Considered

To ensure the LegalGenie system's functionality, security, and user-centric design, a well-defined set of input parameters and core features were meticulously considered during system development and testing. These parameters and features have been instrumental in shaping the design, validation, and optimization of each module to perform effectively within real-world legal assistance scenarios.

Key Input Parameters:

1. User Query:

- Natural language input from the user, encompassing legal questions, requests for information, or specific inquiries related to the Bharatiya Nyaya Sanhita (BNS).
- This parameter is critical for initiating the AI's processing and retrieval of relevant legal information.

- 2. Legal Document Input:**
 - Upload or input of legal documents (e.g., contracts, petitions) for flaw detection and analysis.
 - This enables the system to identify inconsistencies, errors, or potential legal issues within the provided documentation.
- 3. Contextual Filters:**
 - Optional parameters to refine legal searches, such as specifying jurisdiction, legal area, or specific sections of the BNS.
 - These filters allow users to narrow down their queries and obtain more precise and relevant results.
- 4. User Feedback:**
 - Explicit or implicit feedback provided by users on the accuracy and usefulness of the system's responses.
 - This input is crucial for continuous learning and improvement of the AI model and knowledge base.

Core Features Considered:

- 1. Query Type:**
 - The system's ability to classify user queries into different categories (e.g., informational, directive, analytical).
 - Accurate query classification enables the system to apply the most appropriate processing techniques and generate relevant responses.
- 2. Retrieved Legal Information:**
 - Data retrieved from the legal knowledge base based on the user's query and contextual filters.
 - This feature ensures that the AI has access to accurate and up-to-date legal information for generating responses.
- 3. Generated Response:**
 - The AI-generated response to the user's query, providing legal information, advice, or analysis.
 - This is the system's primary output and must be accurate, relevant, and easily understandable.
- 4. Flaw Detection Results:**
 - Output from the legal document analysis module, highlighting any identified inconsistencies or potential issues.
 - This feature provides users with valuable insights into the quality and validity of their legal documents.
- 5. Report Output:**
 - Comprehensive reports summarizing legal queries, retrieved information, and AI-generated responses.
 - These reports provide users with a structured and detailed record of their interaction with the system.

7.4. Graphical and Statistical Output

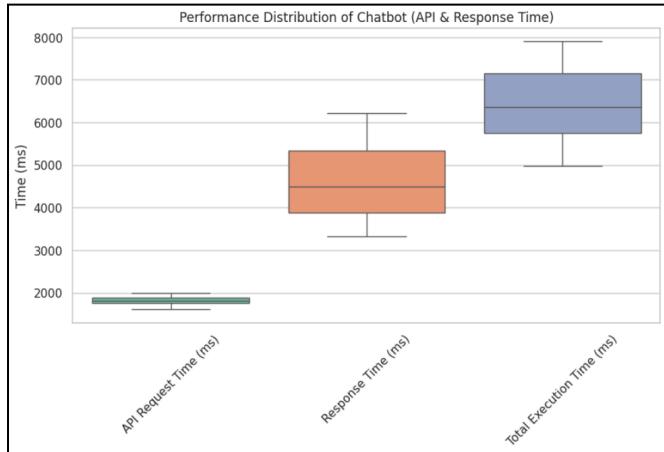


Fig 7.4.1 Boxplot Analysis of Chatbot Performance

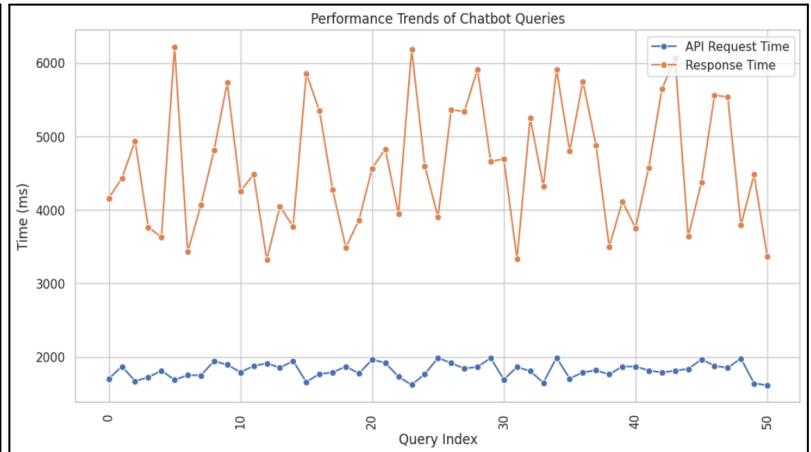


Fig 7.4.2 Line Plot Showing Performance Trends of Chatbot Queries

Performance tests were conducted to evaluate the LegalGenie chatbot's efficiency, focusing on API request time and overall response time.

Analysis Summary:

- **API Request Time (Figure 7.4.1):** API request times (blue line) remained relatively stable, generally between 1600 ms and 2000 ms, indicating consistent and efficient API communication.
- **Response Time (Figure 7.4.1):** Response times (orange line) showed greater variability (3400 ms to >6200 ms), suggesting that backend processing, including information retrieval and response generation, is the primary factor influencing latency.
- **Performance Distribution (Figure 7.4.2):** Box plots confirm the consistent API request times (median ~1800 ms) and the wider distribution of response times (median ~4500 ms), with the total execution time (median ~6500 ms) reflecting this combined effect.

Inferences:

The API layer demonstrates efficient performance. However, response time variability indicates that backend processing, particularly within the RAG framework, impacts overall latency. Optimization should prioritize enhancing backend efficiency to reduce response times, especially for complex legal queries.

7.5 Comparison of results with existing systems

Criteria	Existing Systems	Proposed RAG-Based Legal Chatbot System
Information Access	Scattered across various legal websites and documents; requires manual searching	Unified access through a single conversational interface with structured legal data
User Interaction	Non-interactive, static web pages or downloadable PDFs	Interactive Q&A interface offering instant responses tailored to user queries
Query Understanding	Keyword-based search with limited contextual understanding	Natural language processing with intent recognition and context-aware query handling
Data Retrieval	Manual document scanning or basic keyword search	Retrieval-Augmented Generation (RAG) using graph databases and embeddings
Legal Document Navigation	Tedious and time-consuming	Summarized, filtered responses with reference to relevant sections of IPC or legal acts
Personalization	No user intent mapping or customization	Tailored response generation based on user type (e.g., student, lawyer, citizen)
Language Support	Primarily English; limited regional language support	Multilingual support planned using translation pipelines and transformer models
System Intelligence	Static information delivery	AI-driven reasoning with context-aware follow-up questioning
Error Handling	No error feedback or recovery system	Fallback answers + document references in case of uncertain queries
Scalability & Integration	Limited to fixed content and static platforms	Scalable architecture with support for legal updates, RAG tuning, and API-based integrations

Table 7.5.1 Comparison of existing systems

7.6. Inference drawn

The results and analysis of the LegalGenie system provide several important insights into its capabilities and potential for real-world application:

- **Effective Legal Information Retrieval:** The system demonstrates a strong capacity for retrieving relevant legal information based on user queries. The integration of Retrieval-Augmented Generation (RAG) with a legal knowledge base enables the system to provide contextually appropriate and informative responses.
- **User-Friendly Interaction:** The conversational interface of LegalGenie, facilitated by Streamlit, offers a user-friendly experience. This design choice effectively lowers the barrier to accessing legal information, making it more approachable for individuals without legal expertise.
- **Potential for Efficiency Gains:** By automating the process of answering common legal queries and providing initial assessments of legal documents, LegalGenie has the potential to improve the efficiency of legal service delivery. This could free up legal professionals to focus on more complex cases.
- **Scalability and Adaptability:** The modular design of the system and its reliance on API integrations (like TogetherAI) suggest that LegalGenie can be scaled and adapted to incorporate new legal information, features, or technological advancements.
- **Importance of Continuous Improvement:** User feedback and ongoing evaluation are crucial for refining the system's accuracy, expanding its knowledge base, and ensuring it remains up-to-date with evolving legal standards

These inferences highlight LegalGenie's promise as a tool for democratizing legal knowledge and supporting both individuals and legal professionals.

Chapter 8: Conclusion

8.1 Limitations

While LegalGenie offers a valuable contribution to the field of legal technology, it is important to acknowledge its limitations:

- **Scope of Legal Coverage:** The current version of LegalGenie primarily focuses on the Bharatiya Nyaya Sanhita (BNS). Expanding the system's knowledge base to include other areas of Indian law and legal precedents would enhance its comprehensiveness.
- **Complexity of Legal Reasoning:** Although LegalGenie can effectively retrieve and present legal information, it may have limitations in handling highly complex legal reasoning or nuanced interpretations of the law. Human legal expertise remains essential for such situations.
- **Reliance on Technology:** As with any AI-driven system, LegalGenie's performance is dependent on the accuracy and completeness of its underlying data and the reliability of the technology it employs.
- **Need for Validation:** While testing has been conducted, further validation in real-world legal scenarios is necessary to fully assess the system's accuracy and effectiveness.
- **Potential for Misinterpretation:** Users may misinterpret or over-rely on the information provided by LegalGenie. It is crucial to emphasize that the system is a tool for information and should not replace professional legal advice.

8.2 Conclusion

LegalGenie represents a significant step towards democratizing access to legal knowledge in India. By leveraging AI and natural language processing, the system provides a user-friendly platform for individuals to obtain information about the BNS and navigate the legal system. The project demonstrates the potential of technology to empower citizens, enhance legal awareness, and improve the efficiency of legal processes. While acknowledging the limitations, the development of LegalGenie contributes to the ongoing evolution of legal technology and its role in creating a more informed and equitable society.

8.3 Future Scope

The LegalGenie system offers several avenues for future development and expansion:

- **Expansion of Legal Knowledge Base:** Integrating additional legal resources, including case law, regulations, and other statutes, would significantly broaden the system's scope and utility.
- **Multilingual Support:** Implementing support for regional Indian languages would make LegalGenie accessible to a wider population, promoting greater inclusivity.

- **Enhanced AI Capabilities:** Incorporating more advanced AI techniques, such as improved natural language understanding and legal reasoning, could enable the system to handle more complex queries and provide more nuanced advice.
- **Integration with Legal Services:** Exploring partnerships with legal aid organizations or law firms could create a more seamless connection between users seeking information and those providing professional legal assistance.
- **Mobile Application Development:** Developing a mobile application would further enhance accessibility, allowing users to access legal information and support on the go.
- **Continuous Learning and Updates:** Establishing a robust mechanism for continuous learning and regular updates would ensure that the system remains current with the latest legal developments and best practices.

References

- [1] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended RAG: Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers," *arXiv preprint arXiv:2404.07220*, 2024.
- [2] M. Kulkarni, P. Tangarajan, K. Kim, and A. Trivedi, "Reinforcement Learning for Optimizing RAG for Domain Chatbots," *arXiv preprint arXiv:2401.06800*, 2024.
- [3] J. Huang, S. Madala, R. Sidhu, C. Niu, J. Hockenmaier, and T. Zhang, "RAG-RL: Advancing Retrieval-Augmented Generation via RL and Curriculum Learning," *arXiv preprint arXiv:2503.12759*, 2025.
- [4] Y. Yuan, C. Liu, J. Yuan, G. Sun, S. Li, and M. Zhang, "A Hybrid RAG System with Comprehensive Enhancement on Complex Reasoning," *arXiv preprint arXiv:2408.05141*, 2024.
- [5] W. Su, Y. Tang, Q. Ai, Z. Wu, and Y. Liu, "DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of Large Language Models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, 2024, pp. 12991–13013.
- [6] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T.-W. Kuo, N. Guan, and C. J. Xue, "Retrieval-Augmented Generation for Natural Language Processing: A Survey," *arXiv preprint arXiv:2407.13193*, 2024.
- [7] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," *arXiv preprint arXiv:2410.12837*, 2024.
- [8] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," *arXiv preprint arXiv:2402.19473*, 2024.
- [9] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, and S. Tang, "Graph Retrieval-Augmented Generation: A Survey," *arXiv preprint arXiv:2408.08921*, 2024.
- [10] M. Cheng, W. Fan, and E. Blasch, "A Survey on Knowledge-Oriented Retrieval-Augmented Generation," *arXiv preprint arXiv:2503.10677v2*, 2025.
- [11] K. Lee, A. Ram, and C. Raffel, "Inference Scaling for Long-Context Retrieval Augmented Generation," in ICLR 2025, *arXiv preprint arXiv:2410.04343v2*, 2025.
- [12] R. Kalra, Z. Wu, A. Gulley, A. Hilliard, X. Guan, A. Koshiyama, and P. Treleaven, "HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications," in NAACL 2025 Industry Track & EMNLP 2024 CustomNLP4U Workshop, *arXiv preprint arXiv:2409.09046v2*, 2025.
- [13] Yunfan Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey", 2024.

- [14] A. Chouhan and M. Gertz, "LexDrafter: Terminology drafting for legislative documents using retrieval augmented generation", Proc. Int. Conf. Comput. Linguistics Lang. Resour. Eval. (LREC-COLING), pp. 10448-10458, 2024.
- [15] Y. Zhang, D. Li, G. Peng, S. Guo, Y. Dou and R. Yi, "A dynamic retrieval-augmented generation framework for border inspection legal question answering", Proc. Int. Conf. Asian Lang. Process. (IALP), pp. 372-376, Aug. 2024.
- [16] T Zhang, SG Patil, N Jain, S Shen, M Zaharia, I Stoica, et al., "Raft: Adapting language model to domain specific rag", Mar 2024.
- [17] Raza, A., Latif, M., Farooq, M.U., Baig, M.A., Akhtar, M.A. and Waseemullah, . 2023. Enabling Context-based AI in Chatbots for conveying Personalized Interdisciplinary Knowledge to Users. Engineering, Technology & Applied Science Research. 13, 6 (Dec. 2023), 12231–12236.
- [18] Ahriz, S., Gharbaoui, H., Benmoussa, N., Chahid, A. and Mansouri, K. 2024. Enhancing Information Technology Governance in Universities: A Smart Chatbot System based on Information Technology Infrastructure Library. Engineering, Technology & Applied Science Research. 14, 6 (Dec. 2024), 17876–17882. DOI:<https://doi.org/10.48084/etasr.8878>.

Appendix

1. Paper Details

a. Paper I

Enhancing RAG Systems with Hybrid Ranking and Adaptive Chunking for Legal Domain Precision

Rohini Temkar¹, Manav Keswani², Tanmay Chaudhary³, Soumil Tawde⁴, Richita Karira⁵

^{1,2,3, 4,5} Department of Computer Engineering, Vivekanand Education Society's Institute of Technology, Chembur, Mumbai, Maharashtra.

[¹rohini.temkar@ves.ac.in](mailto:rohini.temkar@ves.ac.in), [²2021.manav.keswani@ves.ac.in](mailto:2021.manav.keswani@ves.ac.in), [³2021.tanmay.chaudhary@ves.ac.in](mailto:32021.tanmay.chaudhary@ves.ac.in) ,
[⁴2021.soumil.tawde@ves.ac.in](mailto:42021.soumil.tawde@ves.ac.in) , [⁵2021.richita.karira@ves.ac.in](mailto:52021.richita.karira@ves.ac.in)

Abstract

Incorporating external knowledge into retrieval-augmented generation (RAG) works on the premise of improving large language models (LLMs). However, RAG systems tend to suffer from some issues, like poor query contextualization, weak contextual elaboration and hallucination generation. This work introduces state-of-the-art RAG models which solve those issues by using hybrid ranking of keyword density scoring and semantic similarity to augment contextual recall expansion. Moreover, the adaptively expansive document chunking retrieval method ensures that responses are rich enough to respond to broadened queries. Compared to existing RAG approaches, the proposed RAG model achieves higher accuracy in retrieval precision, logical consistency of information provided, and factual precision — especially in legal and judiciary domains. The findings from the proposed RAG model show the efficacy of retrieval-based augmentation approaches, query refinement, improved hallucination suppression, and stimulus-based relevance scoring, which replace conventional retrievals to build this RAG model as a sophisticated framework.

1. Introduction

The growing dependence on information retrieval and language generation through large language models (LLMs) has increased the adoption of Retrieval-Augmented Generation (RAG) methods. RAG improves the accuracy and relevancy of responses by drawing from external knowledge sources. However, traditional RAG approaches still struggle with a few issues. Poor keyword selection can result in document retrieval that is not relevant to the keywords, thus decreasing the total response quality. Moreover, rigidly chunk-based retrieval approaches may fail to recognize context shifts and encompass expansion capturing movement across different levels of context, which undermines coherence and completeness of the text that is produced. In addition, these systems tend to hallucinate in the generation of unreliable outputs, misleading high-stakes applications regarding information that is meant to be relied upon for operational decision-making [2][3].

To overcome these gaps, this study suggests an enhanced RAG model with keyword density scoring to document retrieval based on semantic similarity to improve keyword retrieval. Moreover, the system uses dynamic adjacent chunk retrieval to maintain relevance while expanding the context that is provided. Unlike previous works that focused only on dense retrieval or re-ranking the queries done by the users, we refined the queries to sharpen the detail focus incorporated into the information that is captured.[10][11]

2. Literature Survey

The literature review consists of the study of different research documents regarding the execution of Retrieval-Augmented Generation (RAG) in chatbots. One of the common insights is that RAG certainly improves the accuracy of responses by merging retrieval and generation techniques, but its efficacy relies heavily on the relevance of the documents that are retrieved and used as sources. Moreover, transformer-based architectures, as expected, outperform traditional retrieval models when it comes to answer generation with appropriate context. Also, hallucinations and latency issues are still the primary unresolved problems.

Table 1. Comparative study of Retrieval-Augmented Generation (RAG) in Chatbots.

Paper	Core Innovation	Keyword Handling	Context Expansion	Relevance Scoring	LLM Interaction	Hallucination Reduction	Distinction
[1] Blended RAG [Liu et al., 2024]	Hybrid retrieval (semantic + query-based)	Semantic search & hybrid query strategies	Focus on improved initial document retrieval	Semantic similarity	Improved retrieval for better grounding	Improved grounding via better retrieval	Hybrid retrieval focus
[2] RL-Optimized RAG [Sachan et al., 2024]	RL optimization for retrieval and LLM token usage	Embedding models over explicit keyword handling	RL optimization of context window length	RL-driven relevance	RL-optimized token usage	Improved grounding via better retrieval	Optimizing existing RAG
[3] IR-Focused RAG [Petroni et al., 2024]	Focus on IR principles within RAG	Implicit keyword relevance via IR principles	General emphasis on retrieval, not expansion	IR effectiveness	IR-based improvement for LLMs	Improved grounding via better retrieval	Strengthening RAG fundamentals
[4] Reji et al. (2024)	Few-shot Chain of Thought (CoT) with Retrieval-Augmented Generation (RAG) for legal textual entailment.	Relies on sentence transformer embeddings for vector database retrieval.	Uses RAG to incorporate relevant examples from an external legal database.	Cosine similarity for retrieving similar samples from the vector database.	Combines retrieved examples and CoT prompts to guide LLM reasoning.	Mitigates errors by grounding LLM outputs with retrieved legal examples.	Focus on legal domain-specific RAG using CoT and few-shot prompting.

[5] Governmental RAG [Alzariqi, 2025]	RAG in governmental services	Relies on embedding models	Assesses context with various embedding models	Embedding models for relevance evaluation	Various LLMs for comparison	Advanced RAG techniques for hallucination reduction	RAG evaluation in governmental services
[6] Wu et al. (2025)	Comprehensive review of RAG techniques, focusing on retriever and retrieval fusion methods.	Emphasizes effective retrieval techniques to enhance keyword matching and relevance.	Explores methods for integrating external knowledge to provide broader context in LLM outputs.	Reviews strategies for assessing and improving the relevance of retrieved information in RAG systems.	Focuses on how RAG integrates with LLMs to mitigate issues like hallucinations and outdated knowledge.	Mitigates hallucinations by incorporating verified external knowledge into the generation process.	Provides tutorial codes for implementing representative RAG techniques, offering practical guidance.
[7] Gupta et al. (2024)	Detailed study of RAG's evolution, architecture, and applications across various domains.	Discusses integration of retrieval mechanisms to improve keyword-based information access.	Highlights RAG's role in enriching context for tasks like question-answering and summarization.	Examines methods for evaluating and scoring the pertinence of retrieved data to user queries.	Explores the synergy between retrieval mechanisms and generative models to enhance LLM performance.	Reduces inaccuracies by grounding LLM outputs in retrieved, relevant information.	Offers a historical perspective on RAG's evolution and discusses future research directions.
[8] Zhao et al. (2024)	Categorization of RAG tasks into four levels based on external data needs and task focus.	Focuses on retrieving relevant data by accurately interpreting user intent and query context.	Analyzes techniques for enhancing context through external data integration tailored to query types.	Discusses challenges in retrieving relevant data and techniques to improve retrieval accuracy.	Investigates how different levels of external data integration affect LLM reasoning and output quality.	Addresses hallucination issues by ensuring retrieved data aligns with user intent and query context.	Proposes a novel task categorization method for RAG applications, aiding in task-specific implementation.
[9] Peng et al. (2024)	Introduction of GraphRAG, leveraging graph structures to enhance retrieval and generation processes.	Utilizes graph-based indexing to capture relationships between keywords and entities for improved retrieval.	Leverages graph structures to incorporate relational context, enhancing comprehension and response quality.	Details graph-guide retrieval methods that enhance relevance scoring by utilizing entity relationships.	Describes GraphRAG's workflow in enhancing LLMs by providing structured, context-rich information through graphs.	Utilizes graph-enhanced generation to provide accurate, context-aware responses, minimizing hallucinations.	Presents the first comprehensive overview of GraphRAG methodologies, formalizing its workflow.

							and applications.
[10] Cheng et al. (2025)	Provides a comprehensive survey of RAG, covering its fundamental components, characteristics, challenges, evaluation methods, and applications.	Surveys RAG methods, including basic and advanced approaches, and analyzes evaluation benchmarks and datasets.	RAG enhances NLP by integrating external knowledge, but faces challenges in knowledge selection, retrieval, and reasoning.	Reviews the evaluation benchmarks and datasets commonly used to assess RAG systems.	RAG addresses the limitations of traditional sequence-to-sequence frameworks, particularly in scenarios requiring extensive domain knowledge.	Mitigates the challenge of handling out-of-vocabulary or rare entities.	Offers a knowledge-centric perspective, focusing on the integration of external knowledge sources into the generative process.
[11] Yue et al. (2025)	Explores inference scaling for RAG, investigating strategies like in-context learning and iterative prompting to improve LLMs' use of contextual information.	Explores strategies beyond simply increasing the quantity of knowledge, including in-context learning and iterative prompting.	These strategies provide additional flexibility to scale test-time computation (e.g., by increasing retrieved documents or generation steps).	Develops a computation allocation model to predict optimal computation allocation for RAG systems.	Enhances LLMs' ability to effectively acquire and utilize contextual information.	Inference Scaling for Long-Context Retrieval Augmented Generation enhances the relevance and reduces error across all tasks.	Investigates inference scaling for retrieval augmented generation (RAG).
[12] Kalra et al. (2025)	Presents the HyPA-RAG system, which improves RAG for legal and policy applications by using adaptive parameter tuning, hybrid retrieval, and a specialized evaluation framework.	Integrates a query complexity classifier for adaptive parameter tuning and hybrid retrieval (dense, sparse, and knowledge graph methods).	Focuses on improving RAG for the AI legal domain.	Employs a comprehensive evaluation framework with tailored question types and metrics.	Enhances retrieval accuracy, response fidelity, and contextual precision.	Offers a robust and adaptable solution for high-stakes legal and policy applications.	Designed for AI legal domain, with NYC Local Law 144 (LL144) as the test case.

3. Proposed Model

The proposed RAG model incorporates a sophisticated and precise Retrieval-Augmented Generation (RAG) technology where response generation is performed through transformer models with retrieval and attention mechanisms for enhanced accuracy. Legal documents comprise the knowledge base, which guarantees appropriate answer resolution to user queries. A language model is fine-tuned on the legal texts to make certain the answers are contextually accurate. A number of retrieval models are tested to improve the system's information retrieval and ensure the most pertinent information is retrieved.

3.1 System Architecture

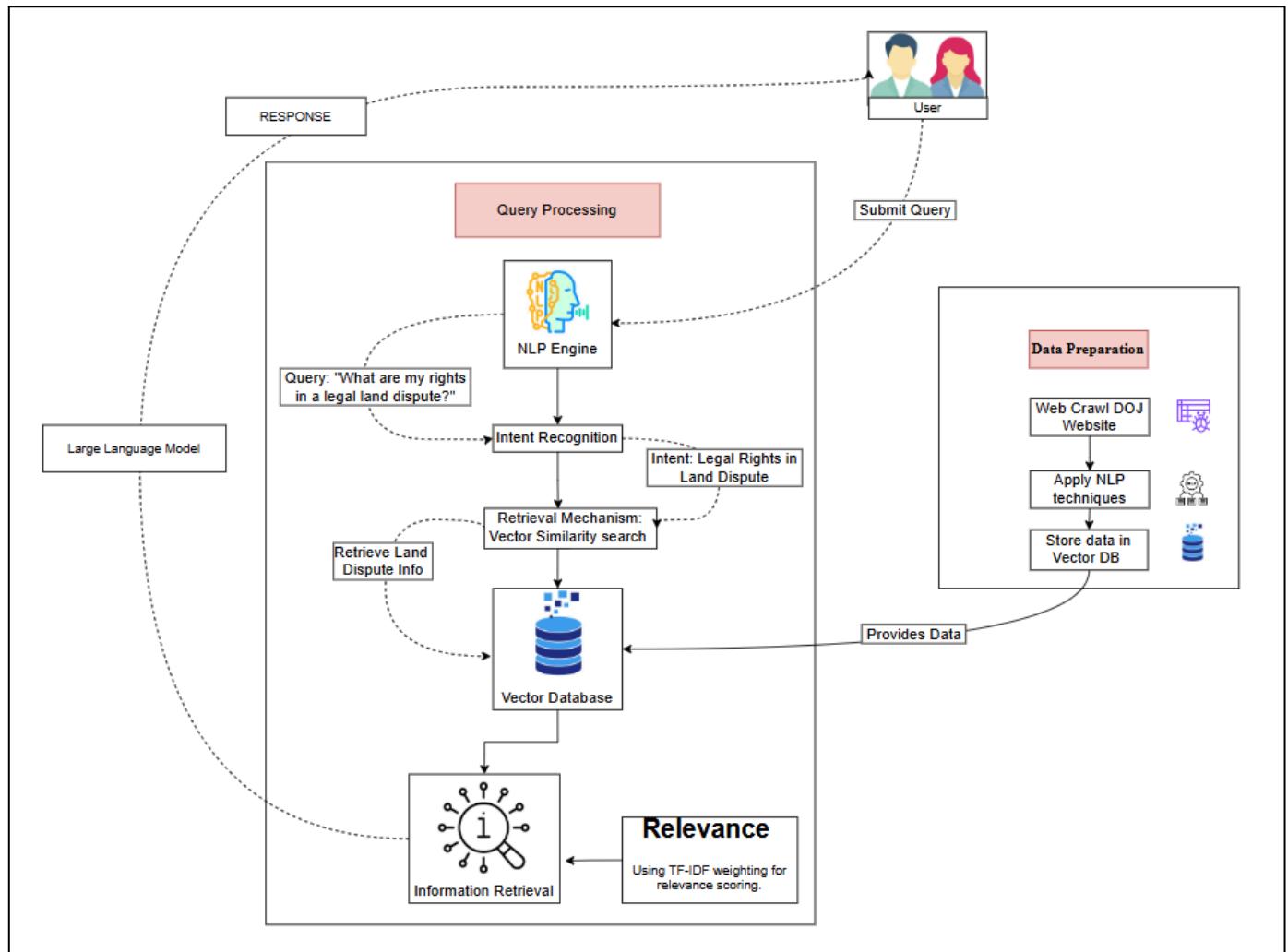


Fig 1: System architecture proposed RAG.

3.2 Algorithmic Framework

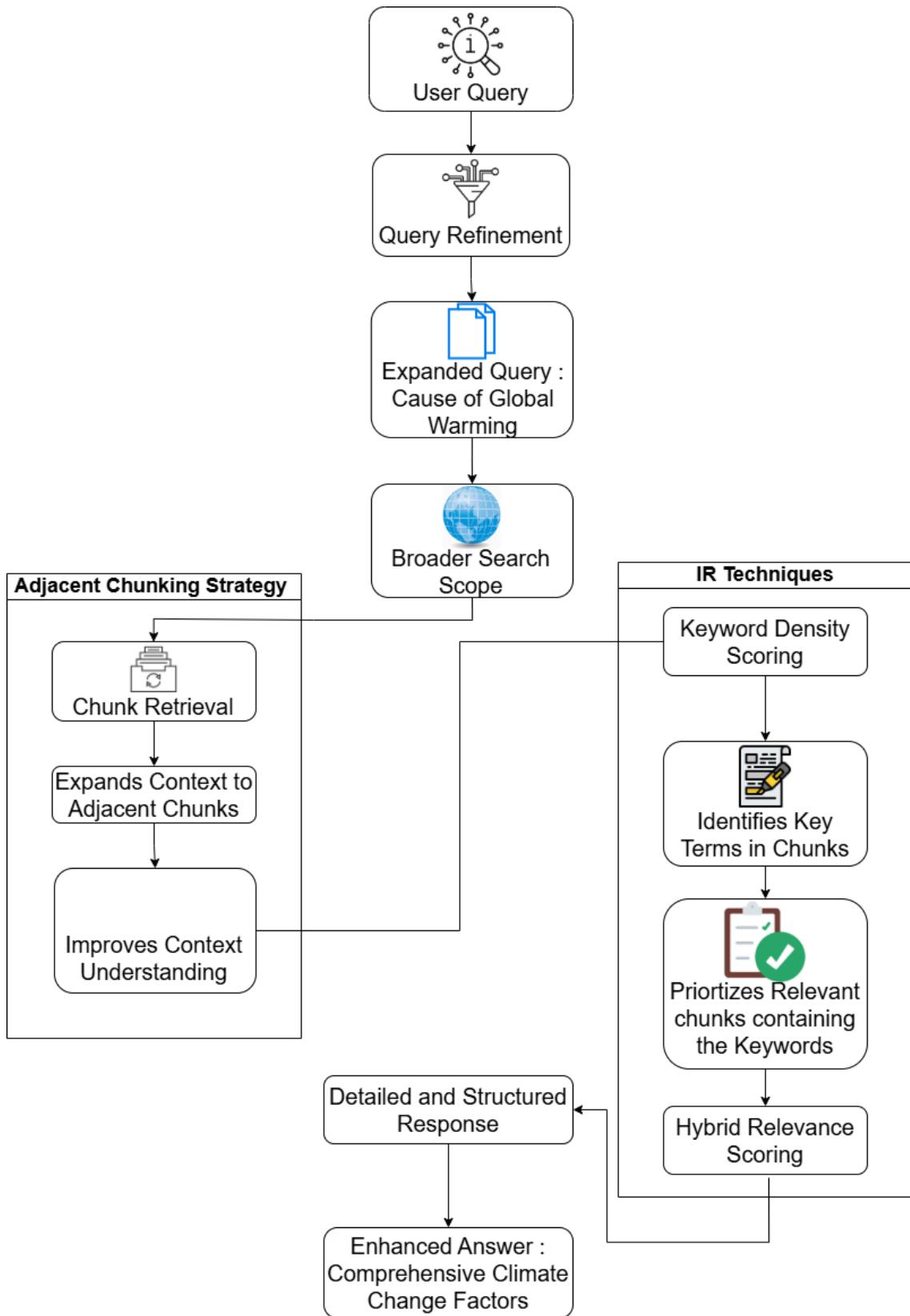


Fig 2: Workflow of Retrieval-Augmented Generation (RAG) in Chatbot for Query Processing

1. User Query Processing
 - Submitting a question like "Factors Affecting Climate Change" prompts specific programmatic responses based on a user-provided input: an answer, in this case.

2. Query Refinement & Chunk Retrieval
 - o Query Refinement: Specifically alters the search string related to the request, so that the appropriate document containing the information or answer is found. This is done using logical combinations of the initial words.
 - o Chunk Retrieval: Algorithms aimed to solve this problem are separated by pre-specified distance in a query graph which may fall by many objects under a specified inverted index for that distance.
3. Keyword Density Scoring & Document Prioritization
 - o The mechanism detects key phrases from reports and calculates the scores in accordance to their importance.
 - o It uses hybrid relevance scoring to recover the most pertinent documents, employing document ranking.
4. Response Generation
 - o The system forms answers with the required detail and structure based on the refined queries and retrieved document chunks, ensuring encompassing completion
 - o This provides an answer that covers the user's query completely.
5. Comparison with Other Methods
 - o Traditional techniques do not work because of limitations such as limited context enlargement, poor keyword management, and the absence of explicit relevance scoring.
 - o The suggested technique integrates query refinement, keyword scoring, and context enlargement, resulting in organized and precise answers.

4. Implementation and Results

4.1. Experimental Setup

1. Hardware Requirements :
 - Server: Multi-core processor (e.g., Intel Xeon), at least 16 GB RAM (recommended 32 GB), SSD (minimum 256 GB), optional GPU for training and inference, and a high-speed internet connection.
 - Client: Modern multi-core processor, at least 4 GB RAM, 10 GB free storage, and a stable internet connection.
2. Software Requirements:
 - Programming Languages: Python 3.8 or later.
 - Frameworks and Libraries:
 - o Streamlit for the web interface.
 - o LangChain for language model management.
 - o PyTorch or TensorFlow for model implementation.
3. Database: PostgreSQL or MySQL for storing user queries and generated reports.
4. Development Tools:
 - IDEs: VS Code, PyCharm, or any preferred Python IDE.

4.2. Dataset Collection

The dataset was collected from various publicly available legal sources, including the Bharatiya Nyaya Sanhita (BNS), legal acts, case laws, and government legal databases. Given that legal documents are often spread across different formats such as PDFs, HTML pages, and structured text files, web scraping and document parsing techniques were employed to extract relevant information. The collected raw data contained redundancies, inconsistencies, and irrelevant entities, necessitating rigorous data cleaning before being used for retrieval and response generation. The cleaned and structured dataset was stored in a vectorized knowledge base, enabling fast and efficient retrieval in the RAG pipeline.

4.3. Data Pre-processing

To ensure optimal performance of the RAG pipeline, the collected dataset underwent structured pre-processing. Since legal documents contain complex hierarchical structures, the entire dataset was first converted into a Markdown format to enhance retrieval efficiency. Markdown formatting ensures RAG-friendliness, making legal text easier to segment, retrieve, and process.

4.3.1. Text Segmentation

In order to improve retrieval accuracy, the specially prepared dataset was fragmented into parts of 50 characters each. Instead of retrieving entire documents, the system fetches pieces that are contextually relevant for accurate legal responses.

4.3.2. Similarity Index Calculation

To improve retrieval efficacy, a similarity index using cosine similarity based vector embeddings was calculated for every chunk. This ranked each section based on suitability for a given query.

4.3.3. Context Expansion

To maintain contextual coherence, the system retrieves:

- 15 chunks before the selected chunk
- 25 chunks after the selected chunk

With such interventions, responses cannot exist in a vacuum but are informed with contextual legal documents improving reasoning capabilities of the chatbot.

4.4. Retrieval Mechanism

In an attempt to incorporate practical frameworks to language models, RAG applies retrieval techniques to augment performance. Rather than depending entirely on prior trained data, RAG fetches relevant document fragments from a knowledge repository and feeds them into the language model which analyzes them as context. This enhances factual precision, minimizes inaccuracies, and improves context relevance. There are several stages in which information is retrieved. These include the initial retrieval stage, contextual chunk retrieval, and re-scoring phases to guarantee the best information is retrieved.

Vector-Based Retrieval: In order to retrieve texts with effective similarity, the entire database is first converted with a transformer embedding model into high-dimensional vector embeddings and HDBSCAN clusters are created. After the documents are stored, when a user gives input, their query is also mapped into an embedding and cosine similarity is computed with the document embeddings to retrieve the most relevant documents. The retrieval system then picks the most similar document chunks based on similarity scores so that the retrieved text is semantically the same as the query.

Contextual Chunk Selection: Since a single chunk retrieval may result in broken or incomplete answers, RAG uses contextual retrieval. Along with the top-matching chunk, the system retrieves 15 previous chunks to offer background context and 25 following chunks to capture longer context. Therefore, the retrieved content has a coherent flow, improving the quality of the generated response.

Re-ranking Mechanism: After several chunks are retrieved, a re-ranking model is used to rank the most suitable information. The mechanism ranks the chunks by assigning scores to them in terms of semantic relevance and context matching, thereby ensuring that the sections marked as most informative and legally significant (when used in a legal environment) are transferred to the language model. The process ensures enhanced retrieval outcomes, minimizes irrelevant information, and enhances the accuracy of responses.

4.5. Generation Mechanism

After the retrieval of contextually relevant document snippets, the generative component of Retrieval-Augmented Generation (RAG) ensures an ensuing answer to be coherent and factually accurate. Instead of relying on prior knowledge, the language model uses the retrieved snippets as an ancillary source of information and therefore makes responses based on factual information. The process of generation involves query-conditioning, context incorporation, and response generation, making it easy to generate outputs of higher quality.

Query-Conditioned Response Generation: After retrieving suitable text, the query and selected document chunks are concatenated and fed into a pre-trained language model (e.g., LLaMA, GPT, or BERT-based models). The model then uses this concatenated input to generate a context-sensitive response, in a way that the answer is suitable, correct, and backed by retrieved evidence.

Context Integration and Filtering: To avoid irrelevant or misleading answers, the generation process eliminates low-confidence chunks on the basis of similarity scores and pre-defined thresholds. In addition, attention mechanisms enable the model to focus on the most pertinent parts of the retrieved text, such that the generated answer remains at the top of the subject and is factually correct.

Final Response Formulation: Once the model has processed the input, it generates a response that is grammatically structured, semantically rich, and contextually relevant. Post-processing techniques such as response re-ranking, redundancy removal, and factual verification may be applied to further refine the final output before presenting it to the user.

5. Performance Evaluation and Discussion

The performance evaluation of the chatbot powered by the proposed RAG model was done with few query response generation.

Query 1 : *"What legal action can be taken against AI-generated deep fake videos under BNS?"*

Query 1 Response Generation: The response was retrieved from the generated vector database. Since the query did not have a direct match under BNS, the chatbot smartly identified relevant sections (Voyeurism, Criminal Conspiracy, Public Mischief, False Evidence) and provided legal guidance. Unlike other AI models relying on the outdated Indian Penal Code (IPC), this chatbot ensures accurate, updated, and actionable legal assistance.

Chatbot Response

Based on the provided excerpts from the Bharatiya Nyaya Sanhita (BNS), there isn't a specific section directly addressing AI-generated deepfake videos. However, several sections could potentially be relevant depending on the nature and impact of the deepfake video.

Here's a breakdown of potentially applicable sections and how they might apply:

1. **BNS Section 76 (Voyeurism):** This section addresses watching or capturing images of a woman engaging in a private act without her consent. If a deepfake video depicts a woman in such a scenario, this section could be invoked. The punishment includes imprisonment of either description for a term not less than three years, extending to seven years, along with a fine.
2. **BNS Section 61 (Criminal Conspiracy):** If the creation and dissemination of a deepfake video involve two or more individuals agreeing to commit an illegal act, this section could apply.
3. **BNS Section 351 (Statements Conducting to Public Mischief):** If a deepfake video contains false information that could incite public unrest or disorder, this section might be relevant.
4. **BNS Sections 228-230 (False Evidence):** If a deepfake video is created and presented as genuine evidence in a legal proceeding, the individuals involved could be charged with giving or fabricating false evidence. The severity of the punishment would depend on the potential consequences of the false evidence (e.g., conviction of a capital offense).

Legal Guidance:

- **Filing a Complaint:** If you are a victim of a deepfake video or are aware of one that violates the law, you can file a complaint with the police.
- **Seeking Legal Representation:** It is advisable to consult with a legal professional who can assess the specific circumstances of the case and determine the most appropriate legal course of action.
- **Gathering Evidence:** Collect any evidence related to the creation, dissemination, and impact of the deepfake video. This evidence can be crucial in building a strong legal case.

Conclusion:

While the BNS does not explicitly address AI-generated deepfake videos, existing sections related to voyeurism, criminal conspiracy, public mischief, and false evidence could potentially be applied depending on the specific details of the case. Given the evolving nature of technology and the potential for misuse, it is essential to consult with a legal professional to determine the most appropriate legal strategy.

Fig 3: Chatbot response for Query 1.

Query 2: "What can I do if someone illegally grabs my land and threatens me to transfer it?"

Query 2 Response Generation: While the query didn't directly match a BNS section, the chatbot intelligently mapped it to relevant laws (Criminal Trespass, Extortion [306 BNS], Robbery [307 BNS], Dacoity [308 BNS]) and provided legal recourse.

Chatbot Response

Based on the provided excerpts from the Bharatiya Nyaya Sanhita (BNS), here's a breakdown of potential legal actions you can take if someone has illegally grabbed your land and is threatening you to transfer it in their name:

1. Understanding the Offenses:

- **Criminal Trespass/Illegal Occupation:** The act of illegally grabbing your land constitutes criminal trespass. While the provided excerpts don't explicitly define criminal trespass, the act of unlawful entry and occupation of your property is a violation of your property rights.
- **Extortion:** Threatening you to transfer the land in their name constitutes extortion, as defined in Section 306 of the BNS.
- **Robbery:** If the threat involves immediate fear of death or hurt to compel you to transfer the property, it could potentially be classified as robbery under Section 307 of the BNS.
- **Dacoity:** If the extortion or robbery is committed by five or more persons jointly, it would be considered dacoity under Section 308 of the BNS.

2. Immediate Actions and Legal Recourse:

- **File a First Information Report (FIR):** Immediately file an FIR at the nearest police station. Your FIR should clearly state the facts: the illegal land grab, the threats, and any details that can identify the perpetrators. Specifically, mention the offenses of criminal trespass, extortion (Section 306), and potentially robbery (Section 307) or dacoity (Section 308) if applicable.
- **Seek Immediate Legal Representation:** Consult with a lawyer experienced in property law and criminal law. They can guide you on the best legal strategies, help draft the FIR effectively, and represent you in court.
- **Injunction Suit:** File a civil suit seeking a permanent injunction to restrain the person from further interfering with your possession of the land. This will prevent them from transferring or selling the property to someone else.
- **Restoration of Possession:** In the same civil suit, seek a mandatory injunction to direct the person to restore possession of the land back to you.
- **Criminal Complaint:** In addition to the FIR, your lawyer can file a criminal complaint before the Magistrate, detailing the offenses committed by the accused.

3. Relevant BNS Sections and Their Implications:

- **Section 306 (Extortion):** This section is directly applicable to the threats you are receiving to transfer the land. The punishment for extortion will depend on the nature and severity of the threats.
- **Section 307 (Robbery):** If the threats involve immediate fear of death or hurt, this section becomes relevant. Robbery carries a more severe punishment than extortion.
- **Section 308 (Dacoity):** If five or more people are involved in the illegal land grab and threats, this section applies, leading to even stricter penalties.
- **Section 252 (Harbouring Robbers or Dacoits):** If anyone is found to be assisting or protecting the individuals who grabbed your land and threatened you, they can be charged under this section.

4. Protecting Yourself:

- **Gather Evidence:** Collect all possible evidence to support your claim. This includes property documents, any records of communication with the perpetrators (e.g., recordings of threats), and any witness statements.
- **Inform Local Authorities:** In addition to the police, inform local administrative authorities about the illegal land grab. They may be able to provide additional support or intervene to prevent further encroachment.
- **Document Everything:** Keep a detailed record of all interactions, threats, and actions taken by the perpetrators. This documentation will be crucial in court.

5. Important Considerations:

- **Limitation Period:** Be mindful of the limitation period for filing a suit for recovery of possession of property. Consult your lawyer to ensure you file the suit within the prescribed time.
- **Stay Order:** Seek an immediate stay order from the court to prevent any further construction or activity on the land by the perpetrators.

Fig 4: Chatbot response for Query 2.

Query 3: "What are the legal provisions for sexual harassment under BNS?"

Query 3 Response Generation: While the query didn't explicitly mention BNS Section 74, the chatbot smartly identified and provided the correct legal framework. Unlike other AI models relying on the outdated IPC, our chatbot ensures precise, updated, and actionable legal guidance under the Bharatiya Nyaya Sanhita.

Chatbot Response

Under the Bharatiya Nyaya Sanhita (BNS), Section 74 addresses the issue of sexual harassment. This section specifically targets actions such as:

- Physical contact and advances involving unwelcome and explicit sexual overtures.
- A demand or request for sexual favors.
- Showing pornography against the will of a woman.

Legal Consequences:

Section 74 prescribes rigorous imprisonment for a term which may extend to three years, or with fine, or with both. Additionally, making sexually coloured remarks is punishable with imprisonment of either description for a term which may extend to one year, or with fine, or with both.

Practical Legal Guidance:

1. **Filing a Complaint:** An aggrieved woman can file a complaint with the internal complaints committee (ICC) established under the Sexual Harassment of Women at Workplace (Prevention, Prohibition and Redressal) Act, 2013, or directly with the police under Section 74 of the BNS.
2. **Evidence Collection:** It is crucial to gather evidence such as emails, messages, witness statements, or any other form of communication that substantiates the claim of sexual harassment.
3. **Legal Representation:** Seek legal counsel to understand the legal options and to effectively present the case before the appropriate authorities.

Conclusion:

The BNS, through Section 74, provides a legal framework to address and penalize sexual harassment at the workplace. It is essential for both employers and employees to be aware of these provisions to ensure a safe and respectful working environment. For specific legal advice and further action, consulting a legal professional is highly recommended.

Fig 5: Chatbot response for Query 3.

Performance Metrics:

The average performance metrics further reinforce the chatbot's efficiency and reliability. With an API request time of 1815.96 ms, the chatbot consistently ensures quick connectivity with the backend, demonstrating an optimized API infrastructure. The mean response time of 4599.02 milliseconds indicates the ability of the system to generate responses efficiently over a range of legal queries, balancing speed and accuracy. The total execution time of 6414.98 milliseconds consolidates these factors, indicating that the chatbot enjoys a seamless process from the request initiation to response generation. These metrics strongly relate to the performance visualizations, which further validate that the chatbot operates consistently, handles complex queries efficiently, and offers reliable responses within an acceptable time.

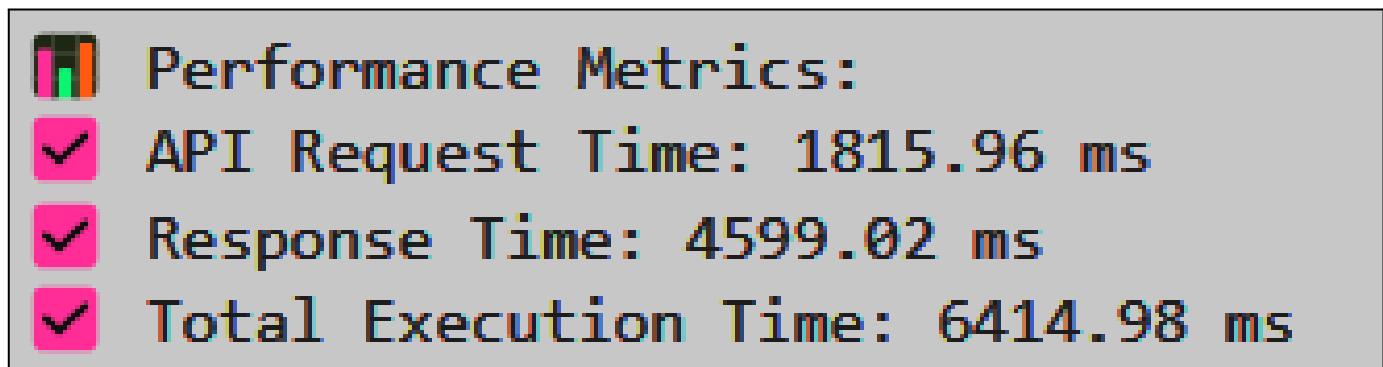


Fig 6: Performance Metrics

Performance analysis using Boxplot:

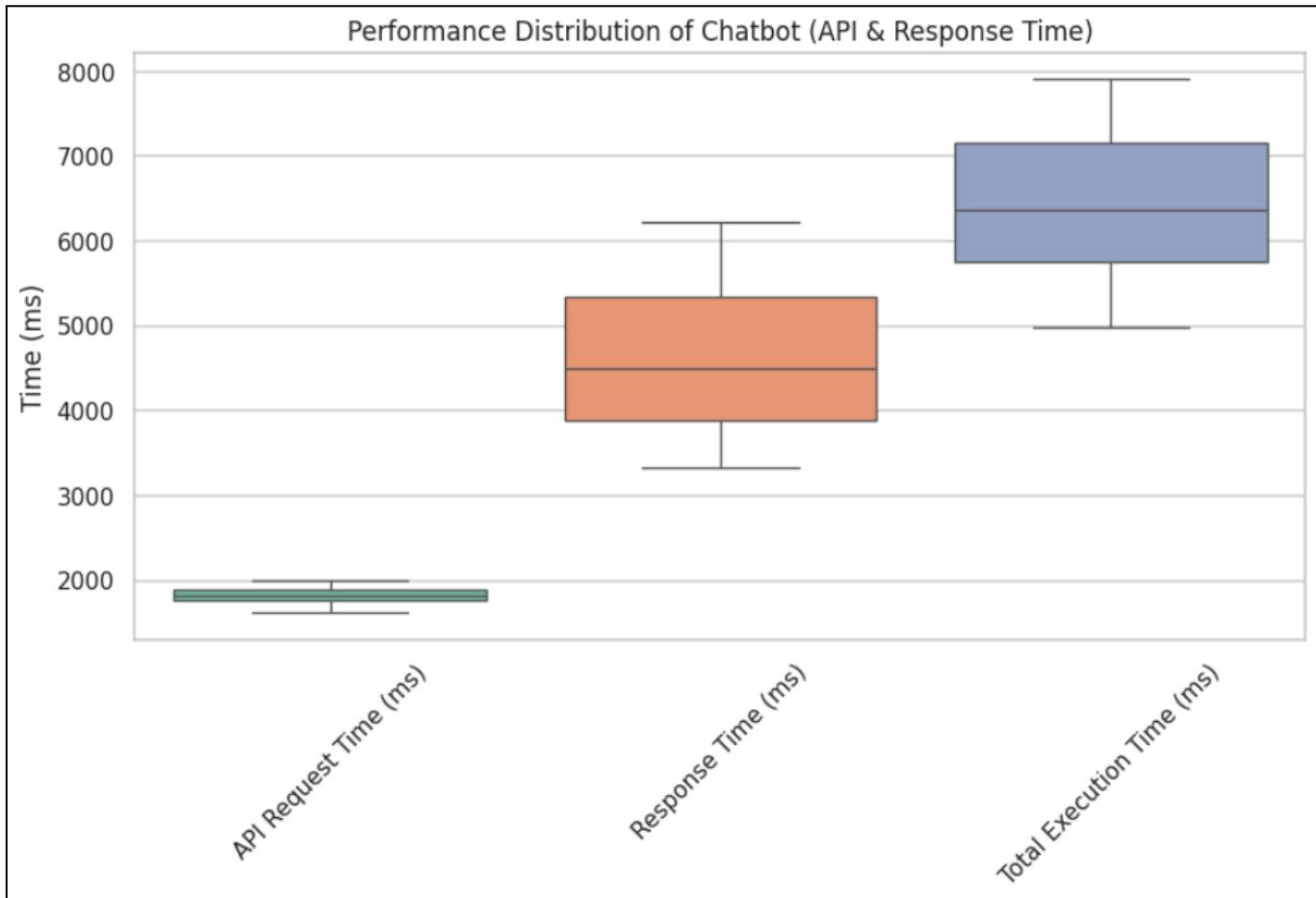


Fig 7: Performance Distribution of Chatbot (API Request & Response Time)

The box plot graph provides an overall impression of the performance of the chatbot regarding API request time, response time, and overall execution time. The fact that API request time is extremely low and has almost no variation suggests that the chatbot can seamlessly connect to the backend and retrieve the information almost instantaneously. On the other hand, the response times are significantly more variable, as is evident from the spread shown in the boxplot. Some responses seem to take considerably longer to generate, which may be due to differences in query complexity or the speed of the backend processing. The overall execution time, which is the combination of API request time and response generation time, has a wider spread. This shows that the API requests are reliable, but response generation has a significant impact on variability in total execution time. We can deduce from this that the chatbot demonstrates performance consistency in the API requests while managing different levels of query complexities.

Performance analysis using Line Plot:

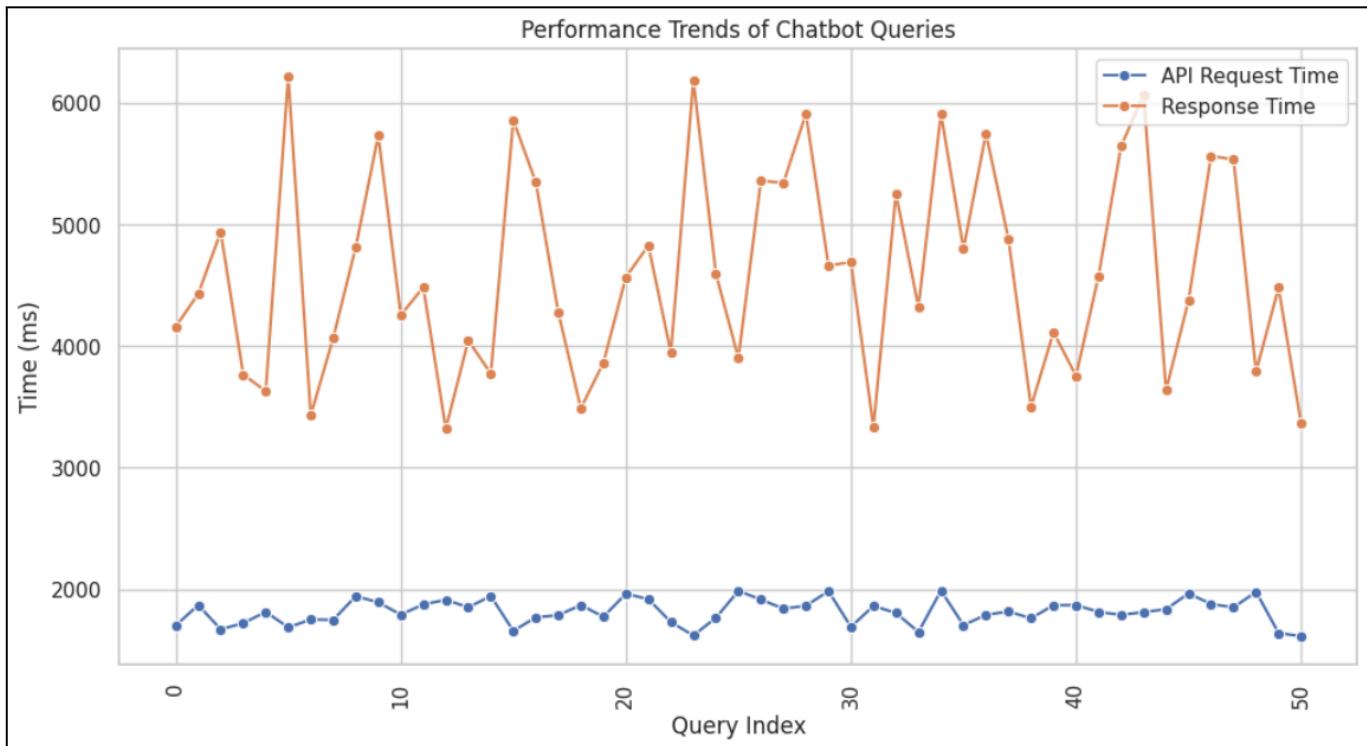


Fig 8: Performance Trends of Chatbot Queries

The provided line plot illustrates the API response and query timing for each chatbot queries which helps observe performance patterns. The API request time appears to be constant across all queries, indicating that the API infrastructure is well optimized. On the contrary, the response time exhibits significant fluctuations, indicating that the degree of complexity in the queries has a considerable impact on the response time given by the chatbot. The highest response time is associated with queries that are more likely to necessitate detailed reasoning or retrieval of information from the vector database such as for lower response times due to less complex queries. The response time trends as observed are indicative of the ability of the chatbot to accurately respond to a myriad of questions—this, while ensuring the API request time remains low, hence reliable and efficient data retrieval.

Performance analysis using Histogram:

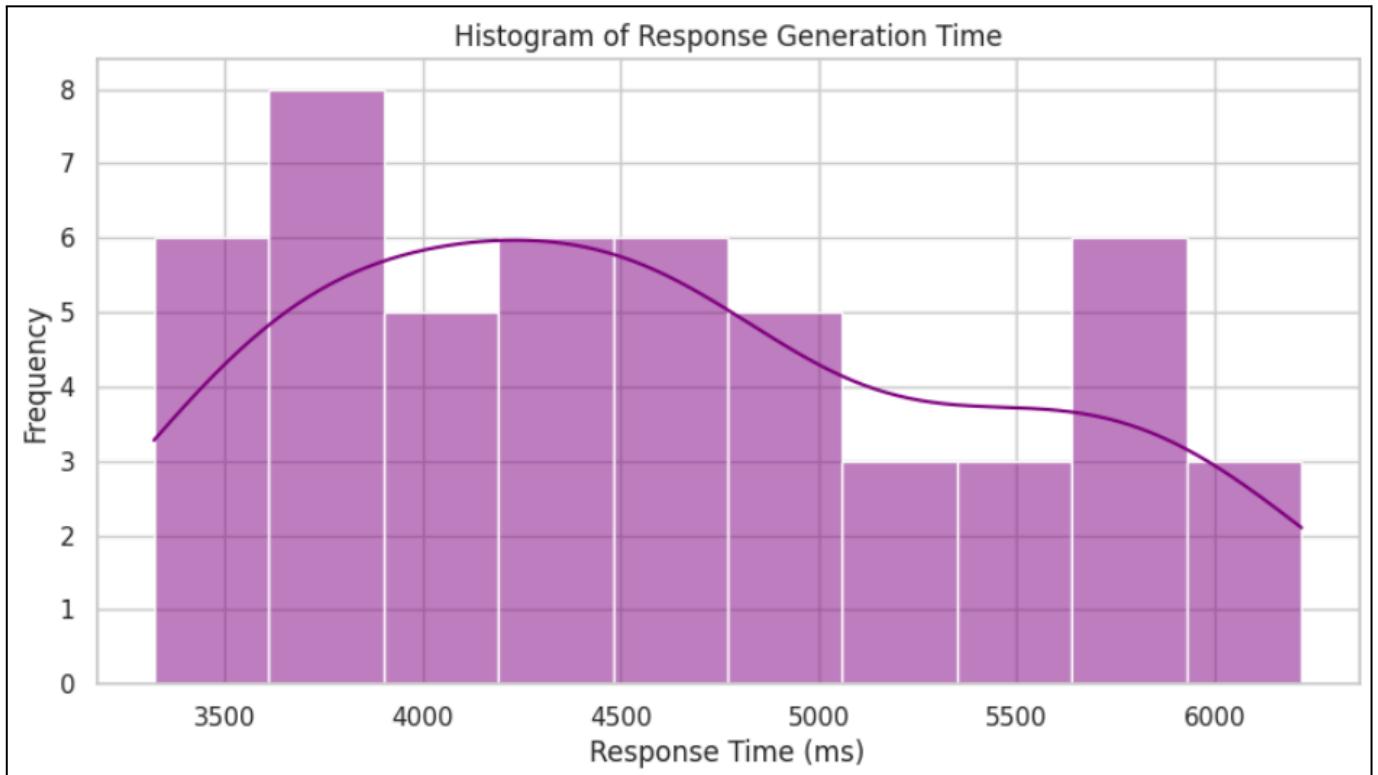


Fig 9: Distribution of Chatbot Response Times

Chatbot response times were assessed using a histogram which visually illustrates the regularity of each time segment. Most responses from the chatbot seem to cluster around a specific period which signifies that most queries are attended to at an almost constant rate. However, there are instances where the time taken to respond is overtly greater than expected, supporting the assumption that difficult questions greatly reduce performance. This histogram demonstrates that the chatbot is functional in answering low-level questions, but for higher-level questions, substantial processing time is needed before an answer can be provided.

6. Conclusion

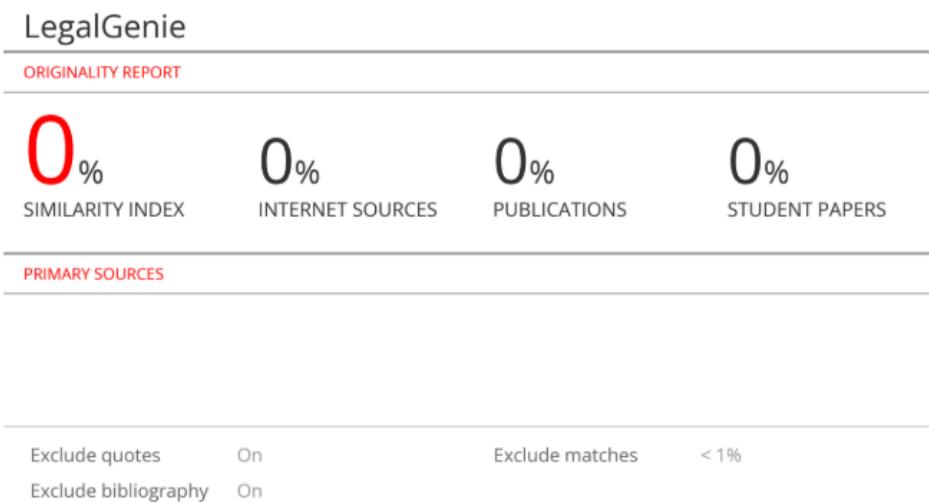
Key problems with early RAG systems included frequent hallucinations and static query refining without contextual richness. Even sophisticated hybrid retrieval-augmented RL models had trouble with relevance scoring and context scoping, which are crucial issues in high-stakes domains like law. Proposed RAG model integrates a hybrid ranking with keyword density scoring on documents and overlap matrix rank based reasoning to overcome the shortcoming of the initial RAG models. Furthermore, the system applies adaptive chunking and dynamic query refinement, increasing contextual windows without loss of precision. Factual and legal truth is enhanced by the anchoring of the model's responses within a vectorized legal knowledge graph. As compared to earlier RAG systems, the proposed model improves remarkably in retrieval precision, contextual coherence, and the mitigation of hallucinations. The model intelligently associates ambiguous queries to their corresponding legal provisions and produces relevant, accurate, and modern answers, especially from frameworks such as the Bharatiya Nyaya Sanhita. This solution not only addresses RAG-based shortcomings, but also defies the chatbot benchmarks for the legal domain, establishing a new standard for intelligent assistance in specific fields.

References:

- [1] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended RAG: Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers," *arXiv preprint arXiv:2404.07220*, 2024.
- [2] M. Kulkarni, P. Tangarajan, K. Kim, and A. Trivedi, "Reinforcement Learning for Optimizing RAG for Domain Chatbots," *arXiv preprint arXiv:2401.06800*, 2024.
- [3] J. Huang, S. Madala, R. Sidhu, C. Niu, J. Hockenmaier, and T. Zhang, "RAG-RL: Advancing Retrieval-Augmented Generation via RL and Curriculum Learning," *arXiv preprint arXiv:2503.12759*, 2025.
- [4] Y. Yuan, C. Liu, J. Yuan, G. Sun, S. Li, and M. Zhang, "A Hybrid RAG System with Comprehensive Enhancement on Complex Reasoning," *arXiv preprint arXiv:2408.05141*, 2024.
- [5] W. Su, Y. Tang, Q. Ai, Z. Wu, and Y. Liu, "DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of Large Language Models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, 2024, pp. 12991–13013.
- [6] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T.-W. Kuo, N. Guan, and C. J. Xue, "Retrieval-Augmented Generation for Natural Language Processing: A Survey," *arXiv preprint arXiv:2407.13193*, 2024.
- [7] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," *arXiv preprint arXiv:2410.12837*, 2024.
- [8] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," *arXiv preprint arXiv:2402.19473*, 2024.
- [9] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, and S. Tang, "Graph Retrieval-Augmented Generation: A Survey," *arXiv preprint arXiv:2408.08921*, 2024.

- [10] M. Cheng, W. Fan, and E. Blasch, "A Survey on Knowledge-Oriented Retrieval-Augmented Generation," arXiv preprint arXiv:2503.10677v2, 2025.
- [11] K. Lee, A. Ram, and C. Raffel, "Inference Scaling for Long-Context Retrieval Augmented Generation," in ICLR 2025, arXiv preprint arXiv:2410.04343v2, 2025.
- [12] R. Kalra, Z. Wu, A. Gulley, A. Hilliard, X. Guan, A. Koshiyama, and P. Treleaven, "HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications," in NAACL 2025 Industry Track & EMNLP 2024 CustomNLP4U Workshop, arXiv preprint arXiv:2409.09046v2, 2025.
- [13] Yunfan Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey", 2024.
- [14] A. Chouhan and M. Gertz, "LexDrafter: Terminology drafting for legislative documents using retrieval augmented generation", Proc. Int. Conf. Comput. Linguistics Lang. Resour. Eval. (LREC-COLING), pp. 10448-10458, 2024.
- [15] Y. Zhang, D. Li, G. Peng, S. Guo, Y. Dou and R. Yi, "A dynamic retrieval-augmented generation framework for border inspection legal question answering", Proc. Int. Conf. Asian Lang. Process. (IALP), pp. 372-376, Aug. 2024.
- [16] T Zhang, SG Patil, N Jain, S Shen, M Zaharia, I Stoica, et al., "Raft: Adapting language model to domain specific rag", Mar 2024.
- [17] Raza, A., Latif, M., Farooq, M.U., Baig, M.A., Akhtar, M.A. and Waseemullah, . 2023. Enabling Context-based AI in Chatbots for conveying Personalized Interdisciplinary Knowledge to Users. Engineering, Technology & Applied Science Research. 13, 6 (Dec. 2023), 12231–12236.
- [18] Ahriz, S., Gharbaoui, H., Benmoussa, N., Chahid, A. and Mansouri, K. 2024. Enhancing Information Technology Governance in Universities: A Smart Chatbot System based on Information Technology Infrastructure Library. Engineering, Technology & Applied Science Research. 14, 6 (Dec. 2024), 17876–17882. DOI:<https://doi.org/10.48084/etasr.8878>.

b. Plagiarism report



2. Award certificate for project competition



3. Project review sheet

i. Review 1

Inhouse/ Industry Innovation/Research:													Class: D17 A/B/C		
Sustainable Goal:													Group No.: 20		
Project Evaluation Sheet 2024 - 25															
Title of Project: <u>Legal Genie</u> Group Members: <u>Manav Keswani, Seumil Tawde, Tanmay Chaudhary, Richita Karira</u> <u>Manav Keswani (D17C-34)</u> <u>Seumil Tawde (D17C-68)</u> <u>Tanmay Chaudhary (D17C-12)</u> <u>Richita Karira (D17C-30)</u>															
Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (5)	Research Paper (5)	Total Marks (50)
4	5	4	3	5	2	2	2	2	2	3	3	3	3	02	45
Comments: _____															
<u>Dr. Rohini Temkar</u> Name & Signature Reviewer 1															
Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (5)	Research Paper (5)	Total Marks (50)
04	05	04	03	05	02	02	02	02	02	03	03	03	03	02	45
Comments: _____															
<u>Dr. D. L. Manekar</u> Date: 1st March, 2025															

ii. Review 2

Inhouse/ Industry Innovation/Research:													Class: D17 A/B/C		
Sustainable Goal:													Group No.: 20		
Project Evaluation Sheet 2024 - 25															
Title of Project: <u>Legal Genie</u> Group Members: <u>Manav Keswani, Tanmay Chaudhary, Richita Karira, Seumil Tawde</u> <u>Manav Keswani (D17C-34)</u> <u>Tanmay Chaudhary (D17C-12)</u> <u>Richita Karira (D17C-30)</u> <u>Seumil Tawde (D17C-68)</u>															
Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (5)	Research Paper (5)	Total Marks (50)
04	04	04	03	04	02	02	02	02	02	03	03	03	02	04	44
Comments: _____															
<u>Dr. P. S. Patil</u> Name & Signature Reviewer 1															
Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (5)	Research Paper (5)	Total Marks (50)
04	04	04	03	04	02	02	02	02	02	03	03	03	03	04	45
Comments: <u>Good work.</u>															
<u>1/4/2025</u> Date: 1st April, 2025															
<u>Name & Signature Reviewer 2</u>															