Legal Aid Provider Portal

SHARANYA[1], SOPNIL PATIL[2], SURAJ T H[3], VINOD M C[4], VAHID MULLA[5]

[1]Professor, Dept. of Computer Science Engineering,
Rajiv Gandhi Institute of Technology, Bengaluru, India
[2,3,4,5] Dept. of Computer Science Engineering,
Rajiv Gandhi Institute of Technology, Bengaluru, India

Abstract

Accessing and understanding legal information can be difficult for ordinary citizens due to complex legal terminology and lengthy legal documents. Artificial Intelligence (AI) and Natural Language Processing (NLP) can help make legal information easier to access through conversational systems [1], [4]. However, general-purpose AI models may generate inaccurate or unsupported legal information [2].

This paper proposes Legal Aid Provider, an AI-based system designed to help users understand their legal rights and possible lawful next steps under Indian law. The proposed system uses a structured legal knowledge base and Retrieval-Augmented Generation (RAG) to retrieve relevant legal provisions before generating responses [8]. Sentence Transformer models are planned for semantic representation and retrieval [3], while LLaMA 3 and Ollama are considered for local response generation [6], [7]. The system also considers multilingual interaction to improve accessibility for Indian users [5]. The objective is to provide simple, source-oriented legal information while encouraging users to seek professional legal assistance when required.

Keywords— Legal Aid, Artificial Intelligence, Retrieval-Augmented Generation, RAG, Legal Information Retrieval, Semantic Search, FAISS, Sentence Transformers, Large Language Model, Indian Constitution, Bharatiya Nyaya Sanhita, Bharatiya Nagarik Suraksha Sanhita.

I. INTRODUCTION

Understanding Indian laws can be challenging for ordinary citizens because legal documents often contain complex terminology, lengthy provisions, and detailed legal procedures. As a result, people may find it difficult to identify their legal rights or determine the appropriate lawful steps to take in a particular situation. Artificial Intelligence (AI) and Natural Language Processing (NLP) can help address this problem by allowing users to interact with legal information using natural language [1]. Previous research has explored automated legal advice systems and AI-based chatbots for resolving legal queries in the Indian context, demonstrating the potential of conversational AI in improving access to legal information [1], [4]. However, generative AI systems may sometimes produce inaccurate or unsupported information, which is a significant concern when they are used for legal applications [2].

To reduce this limitation, the proposed **Legal Aid Provider** uses a Retrieval-Augmented Generation (RAG) approach, in which relevant legal information is retrieved from a structured knowledge base before the response is generated [8]. Sentence Transformer models can be used to convert user queries and legal provisions into vector representations for semantic retrieval [3]. The proposed system also considers LLaMA 3 for language generation and Ollama for running the model locally [6], [7]. By combining legal document retrieval with AI-based response generation, the system aims to provide users with simple and understandable legal information, relevant legal provisions, and possible lawful next steps. Multilingual interaction is also considered to improve accessibility for users across India [5]. The system is intended to assist users in understanding legal information and does not replace qualified advocates or formal legal services.

II. LITERATURE SURVEY

Kulkarni et al. proposed an **automated legal advice system for Indian law using Natural Language Processing (NLP)**. Their work demonstrated how NLP techniques can be used to process legal queries and provide relevant legal information to users, showing the potential of AI in improving access to legal assistance [1]. However, such systems need to accurately connect user queries with the appropriate legal provisions to provide reliable information.

Zurek et al. evaluated **ChatGPT as a legal advisor** and examined the potential of generative AI in legal applications. Their study highlighted the ability of conversational AI to understand and respond to legal questions, while also pointing out concerns regarding the reliability and accuracy of AI-generated legal responses [2]. These limitations indicate the need for legal systems that use trusted external sources along with language models.

Bhambhoria et al. studied the **transferability of pretrained models for legal NLP tasks**. Their research showed the usefulness of pretrained language models for handling specialized legal language and tasks [3]. This supports the use of embedding-based models, such as Sentence Transformers, for representing legal documents and user queries.

Kaur et al. developed an **AI chatbot for legal query resolution in the Indian judicial system**, demonstrating how conversational interfaces can help users interact with legal information in a simple manner [4]. However, improving the connection between chatbot responses and specific legal sources remains important for reliable legal assistance.

Jain et al. investigated **multilingual NLP for Indian languages**, highlighting the importance of multilingual capabilities for AI systems intended for India's diverse population [5]. Supporting multiple languages can make legal information more accessible to users who may not be comfortable using English.

Meta AI Research introduced **LLaMA 3**, an open family of large language models suitable for various natural-language applications [6]. Ollama provides a platform for running large language models locally, which can support applications requiring local model execution [7].

Lewis et al. introduced **Retrieval-Augmented Generation (RAG)**, which combines information retrieval with language generation by providing retrieved external information as context to the language model [8]. This approach is particularly useful for legal applications because relevant legal provisions can be retrieved before generating an answer. Chroma DB provides an open-source vector database that can be used to store and retrieve embeddings for semantic search [9].

Overall, the reviewed studies demonstrate the potential of **NLP, AI chatbots, pretrained models, multilingual processing, LLMs, and**

RAG for improving access to legal information. Based on these works, the proposed **Legal Aid Provider** combines a structured legal knowledge base, semantic retrieval, and LLM-based response generation to provide users with simple, accessible, and source-oriented legal information.

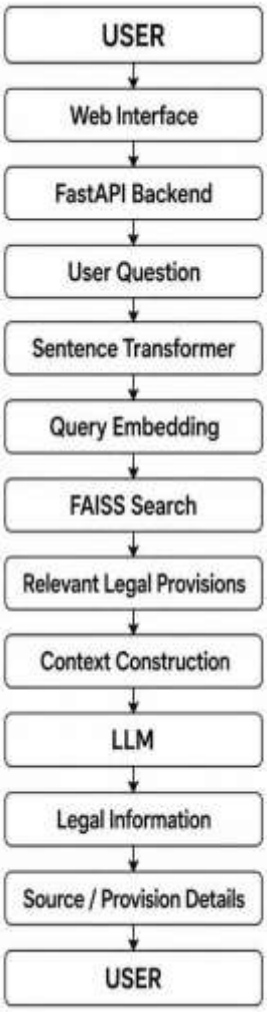
III. PROPOSED SYSTEM

The proposed Legal Aid Provider is an AI-based legal information and guidance system designed to help ordinary citizens understand Indian laws and their legal rights. The system allows users to describe their legal problems in natural language through a web-based interface. The user's query is processed by the backend and relevant legal provisions are retrieved from a structured legal knowledge base.

The system uses Sentence Transformer models to convert the user's query and legal documents into vector representations, enabling semantic similarity-based retrieval [3]. A vector database such as ChromaDB can be used to store and retrieve these embeddings efficiently [9]. The retrieved legal provisions are then provided as context to the Large Language Model (LLM) using the Retrieval-Augmented Generation (RAG) approach [8]. This helps the system generate responses based on relevant legal information rather than relying only on the model's internal knowledge.

For response generation, the proposed system can use LLaMA 3 with Ollama for local execution [6], [7]. The backend is developed using FastAPI, which provides communication between the web interface, retrieval components, and language model [12]. The system is designed to explain relevant legal provisions in simple language, identify applicable laws where possible, and suggest appropriate lawful next steps**.**

The complete proposed workflow is:



IV. METHODOLOGY

The methodology of the proposed Legal Aid Provider consists of multiple stages, beginning with legal document preparation and continuing through retrieval and response generation.

1. Legal Document Collection

The first stage involves collecting the legal documents required for the knowledge base. The current implementation uses the Constitution of India, BNS, and BNSS.

These documents have different structures, so they cannot simply be combined as plain text. Their internal organization must first be identified.

2. Text Extraction

The legal documents are processed using Python-based PDF extraction techniques. Text is extracted from the source documents and prepared for structural analysis.

The extracted text may contain page numbers, headings, footnotes, amendment information, and other document elements. Therefore, processing is required before the content can be used for semantic retrieval.

3. Pattern Identification

The extracted documents are analyzed to identify repeated structural patterns.

For example, the Constitution contains: Preamble, Parts, Chapters, Articles, Schedules, Appendices.

BNS contains: Chapters, Sections, Section Titles, Legal Text

BNSS contains: Chapters, Sections, Schedules, Legal Text

Identifying these patterns allows the system to preserve the relationship between a legal provision and its surrounding structure.

4. Dataset Structuring

Each identified legal provision is converted into a structured record. The standardized record contains information such as: ID, Law Code, Law Name, Provision Type, Provision Number, Provision Title, Chapter Number, Chapter Title, Legal Text, Source Document, Source Pages. This metadata is important for retrieval, filtering, source identification, and response generation.

5. Dataset Unification

The individual Constitution, BNS, and BNSS datasets are normalized and combined into: unified_legal_dataset.jsonl

6. Dataset Validation

Before the data is used for embeddings, a validation program checks the structural integrity of the dataset.

The validation checks:

- Total record count.
- Legal source.
- Provision type.
- Text availability.
- Required metadata.
- Duplicate IDs.
- Valid record objects.

The final validation result is:

Total records: 1,783, Missing text: 0, Metadata issues:0, Duplicate IDs: 0, Invalid record objects: 0

The provision distribution is:

Sections: 934, Articles: 584, Schedules: 240, Appendices: 24, Preamble: 1

This confirms that the current dataset has passed the defined structural validation stage.

7. Embedding Generation

The next stage will convert each legal record into an embedding using a Sentence Transformer model.

For example:

Legal Provision

↓

Sentence Transformer

↓

Numerical Embedding

↓

Vector

The user query will also be converted into a vector using the same model.

8. Vector Indexing

The generated legal embeddings will be stored in a FAISS index. FAISS will allow the system to perform similarity searches between a query vector and the legal-document vectors.

9. Query Retrieval

When the user asks a question, the system will generate a query embedding and search the FAISS index. The system will retrieve the most relevant legal records based on semantic similarity.

10. Context Construction

The retrieved legal records will be combined into a context containing the relevant legal text and metadata. This context will be provided to the LLM.

11. Response Generation

The LLM will receive: User Question & Retrieved Legal Context and generate a response based on the available legal information.

12. Source Presentation

The system will identify the relevant law and provision whenever possible.

For example:

Law: Constitution of India

Provision: Article 14

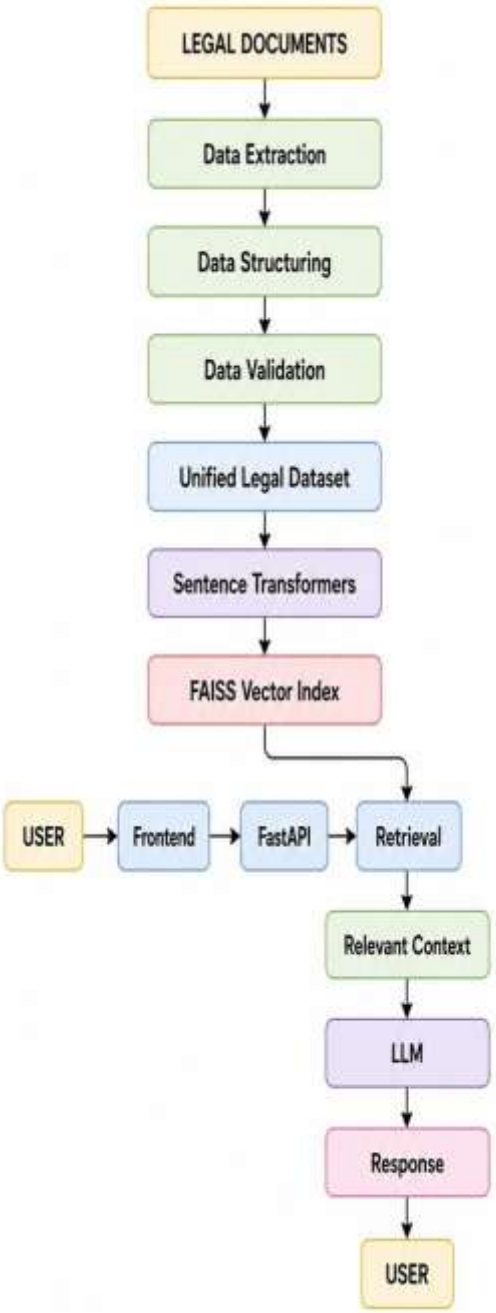
Title: ...

This allows users to understand where the information originated.

V. SOFTWARE IMPLEMENTATION

The software implementation of the proposed Legal Aid Provider is designed as a modular web application. The backend will use **Python and FastAPI** to provide APIs between the frontend, retrieval system, database, and language model. The legal-data processing pipeline has already been implemented to extract, structure, normalize, and validate the legal datasets. The current legal knowledge base contains 1,783 validated records from the Constitution, BNS, and BNSS. The next software stage will implement the embedding and vector-search pipeline. Sentence Transformers will generate embeddings from the legal records. FAISS will then create the vector index required for similarity-based retrieval. The backend will receive user questions and pass them to the retrieval system. Retrieved legal provisions will be supplied to the LLM. The frontend will provide a simple conversational interface through which users can submit questions and view generated responses. The application will also include authentication and administrative functionality where required. The existing backend architecture can be extended to manage users, consultations, administrative operations, and system-related data.

The overall implementation can be represented as:



VI. RESULTS AND DISCUSSION

The current implementation successfully demonstrates the preparation of a structured legal knowledge base for the proposed AI-based legal information system. The Constitution of India, BNS, and BNSS have been processed and combined into a unified dataset containing 1,783 records. The Constitution contributes 771 records, BNS contributes 364 records, and BNSS contributes 648 records.

The final provision distribution is:

Provision Type	Number
Sections	934
Articles	584
Schedules	240
Appendices	24
Preamble	1
Total	**1,783**

During development, the dataset-processing pipeline was refined to correctly distinguish different legal structures. This was necessary because schedules and other non-section content were initially being classified incorrectly. A dedicated validation system was subsequently used to verify the unified dataset.

The final validation result was:

Missing text: 0

Metadata issues: 0

Duplicate IDs: 0

Invalid records: 0

This demonstrates that the current knowledge base satisfies the defined structural requirements for proceeding to embedding generation. The next implementation stage will generate semantic embeddings and create the FAISS vector index. Retrieval experiments will then be performed using representative legal questions. The complete RAG system will subsequently be evaluated using retrieval relevance, response quality, citation correctness, response consistency, and response time.

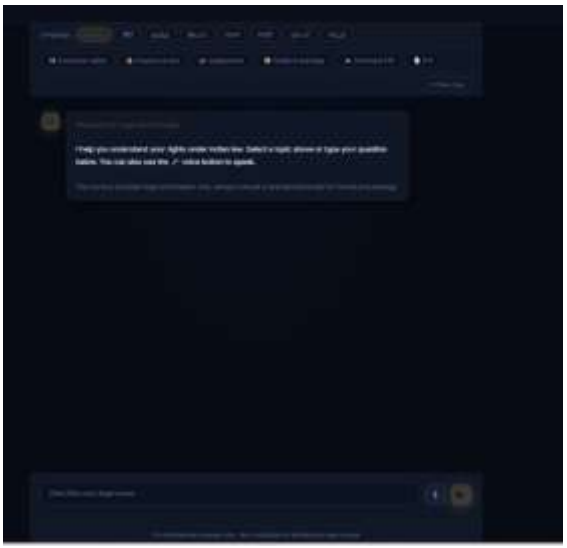
It is important to note that the current validation confirms the **technical structure and metadata integrity** of the dataset. It does not constitute independent legal certification of every extracted sentence or guarantee that every document represents the most recent legal amendment. Source verification and legal-content review remain necessary for production deployment.

UI/UX:

Landing page



Chat Page



VII. CONCLUSION

The Legal Aid Provider presents an AI-based approach for improving access to understandable Indian legal information. The proposed system combines structured legal data, semantic embeddings, FAISS-based retrieval, Retrieval-Augmented Generation, and Large Language Models into a single framework. The current implementation has successfully processed the Constitution of India, BNS, and BNSS and created a unified dataset containing 1,783 records. The dataset includes 584 Articles, 934 Sections, 240 Schedules, 24 Appendices, and the Preamble. A dedicated validation pipeline was developed to verify text availability, metadata completeness, provision classification, duplicate identifiers, and record validity. The final validation produced zero missing text records, zero metadata issues, zero duplicate IDs, and zero invalid record objects. This provides a structured foundation for the next stage of development. The proposed RAG pipeline will use Sentence Transformers to generate semantic embeddings and FAISS to retrieve relevant legal provisions. The retrieved information will then be supplied to an LLM to generate understandable legal information and guidance.

The complete system will integrate the RAG pipeline with a FastAPI backend, web interface, database, authentication, administration, and source-based response presentation. The system will maintain a strict focus on lawful information and will not be designed to promote or facilitate illegal activities. With further implementation, testing, source verification, and evaluation, the proposed Legal Aid Provider can provide a useful platform for improving access to Indian legal information while maintaining appropriate technical and legal limitations.

VIII. REFERENCES

[1] Kulkarni, P., et al. (2022). Automated Legal Advice System for Indian Law Using NLP. International Journal of Computer Applications, 184(15).

[2] Zurek, T., et al. (2023). ChatGPT as a Legal Advisor: An Evaluation. Journal of AI and Law, 11(2).

[3] Bhambhoria, R., et al. (2021). Transferability in Pretrained Models for Legal Tasks. Natural Legal Language Processing Workshop 2021.

[4] Kaur, A., et al. (2023). AI Chatbot for Legal Query Resolution in Indian Judicial System. IJACSA, 14(3).

[5] Jain, D., et al. (2022). Multilingual NLP for Indian Languages. EMNLP 2022.

[6] Meta AI Research. (2024). LLaMA 3: Open Foundation and Fine-Tuned Chat Models. https://ai.meta.com/blog/meta-llama-3

[7] Ollama. (2023). Run Large Language Models Locally. https://ollama.com

[8] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 2020.

[9] Chroma. (2023). ChromaDB: The Open-Source Embedding Database. https://www.trychroma.com

[10] Government of India. (2005). The Right to Information Act, 2005. https://indiacode.nic.in

[11] National Legal Services Authority. (2022). Legal Aid Services in India: Access and Awareness Study. https://nalsa.gov.in

[12] Tiangolo, S. (2023). FastAPI Documentation. https://fastapi.tiangolo.com