SMALL LANGUAGE MODELS FOR SRI LANKAN LEGAL APPLICATIONS

(Labour and Employment law recommendation system for users)

Project Final Report

E. Niruththika – IT22322326

B.Sc. (Hons) in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lank Institute of Information Technology

April 2025

SMALL LANGUAGE MODELS FOR SRI LANKAN LEGAL APPLICATIONS

(Labour and Employment law recommendation system for users)

Project Final Report

E. Niruththika – IT22322326

B.Sc. (Hons) in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lank Institute of Information Technology

April 2025

DECLARATION

I declare that this is my own work, and this final thesis does not incorporate without acknowledgment any material previously submitted for a degree in any other university or Institute of higher learning, and to the best of our knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text. Also, I hereby grant to the Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other mediums. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Niruththika E	IT22322326	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my Supervision.

_____ _____

Signature of the co -supervisor **Signature of the supervisor**

_____ _____

Date **Date**

ABSTRACT

Labour and employment law in Sri Lanka governs workplace disputes, employee rights, and employer obligations. Identifying the appropriate legal provisions such as the relevant Act, section, and year for a given case remains a complex task, particularly for non-experts. Despite the increasing availability of digital legal resources, the lack of intelligent systems for context-aware legal retrieval limits effective access to precise legal information.

This research proposes a domain-specific law recommendation system tailored to Sri Lankan labour and employment law. The system accepts user queries in natural language and maps them to the most relevant legal provisions through semantic understanding. It outputs structured legal information, including the applicable Act, section, and year, along with analogous case scenarios to provide contextual grounding and improve interpretability.

The proposed approach leverages Natural Language Processing (NLP) techniques in conjunction with a fine-tuned Transformer-based Small Language Model (SLM) for domain adaptation. To enhance factual accuracy and mitigate hallucination, the system integrates a Retrieval-Augmented Generation (RAG) framework, which retrieves relevant legal documents from a curated knowledge base and conditions the generation process on this external evidence. Additionally, vector-based similarity search (e.g., FAISS) is utilized to efficiently match user queries with semantically relevant legal texts.

The system is designed to improve legal information accessibility, reduce the cognitive and time burden associated with legal research, and support informed decision-making for both legal practitioners and the public. By combining generative and retrieval-based methods within a specialized legal domain, this work contributes toward the development of intelligent, explainable, and scalable legal decision-support systems in Sri Lanka.

Keywords - Labour and Employment Law, Legal Recommendation System, Natural Language Processing (NLP), Transformer-based Models, Qwen, Retrieval-Augmented Generation (RAG), FAISS, PostgreSQL, Semantic Similarity Search, Domain-Specific Language Models, Sri Lanka Legal System.

ACKNOWLEDGEMENT

The success of this research would not have been possible without the support and guidance of many individuals.

First, I would like to express my sincere gratitude to my supervisor, Dr. Prasana Sumathpala, and co-supervisor, Ms. Karthiga Rajendran, for their continuous guidance, encouragement, and valuable advice throughout the research project. Their expertise and constructive feedback were essential in completing this work successfully.

I would also like to extend my thanks to all lecturers, instructors, assistant lecturers, and both academic and non-academic staff of the Sri Lanka Institute of Information Technology (SLIIT) for their support and contributions during this journey.

My heartfelt appreciation goes to my group members for their cooperation, teamwork, and assistance during the research process.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Descriptions
ML	Machine Learning
NLP	Natural Language Processing
SLM	Small Language Model
RAG	Retrieval-Augmented Generation
T5	Text-to-Text Transfer Transformer
NER	Named Entity Recognition
API	Application Programming Interface
AI	Artificial Intelligence
OCR	Optical character recognition
QLoRA	Quantize Low-Rank Adaptation
LoRA	Low-Rank Adaptation

1. INTRODUCTION

Sri Lanka's legal system encompasses multiple domains, including property, family, labour, and criminal law, all of which are essential for maintaining social order and protecting individual rights. Despite their importance, legal texts remain difficult to access and interpret due to their technical language, fragmented structure across statutes, and the lack of centralized, user-friendly systems.

In practice, individuals dealing with issues such as wrongful termination or wage disputes must identify the relevant Act, section, and year, often requiring legal expertise or extensive manual research. Although digital legal repositories have improved access, they do not effectively map user-specific problems to appropriate legal provisions. This creates a need for intelligent systems that simplify legal understanding and accessibility.

Recent advancements in Artificial Intelligence (AI), particularly Natural Language Processing (NLP), offer promising solutions by enabling systems to interpret human language and generate meaningful responses. While existing legal AI solutions often rely on large language models, these can be computationally expensive and less suitable for domain-specific applications in resource-constrained contexts like Sri Lanka.

To address this, the research explores Small Language Models (SLMs) as an efficient alternative. When fine-tuned on Sri Lankan legal data, SLMs can capture domain-specific terminology and structure while maintaining lower computational requirements. This research is part of a broader study involving four components:

(1) providing step-by-step legal guidance in family and property law

(2) analyzing risks indeed documents

(3) developing a law recommendation system

(4) predicting legal outcomes based on past cases.

This study specifically focuses on the development of a law recommendation system for Sri Lankan labour and employment law. Labour disputes such as unfair dismissal, wage violations, and workplace discrimination require accurate identification of applicable legal provisions.

The proposed system allows users to input queries in natural language and returns structured outputs, including the relevant Act, section, year, and related scenarios for better understanding.

Technically, the system is built on a fine-tuned Transformer-based SLM. However, to reduce inaccuracies and hallucinations in generated responses, a Retrieval-Augmented Generation (RAG) framework is integrated. This approach retrieves relevant legal documents from a structured database using vector-based similarity search (FAISS) and incorporates them into the generation process, ensuring evidence-based outputs.

The system architecture combines PostgreSQL for structured storage and vector databases for semantic retrieval, making it scalable and adaptable. By improving accuracy, transparency, and reliability, the system supports both legal professionals and the public. It reduces research time for professionals and enhances legal awareness for non-experts.

In conclusion, this research presents an AI-driven, domain-specific solution for improving access to Sri Lankan labour law. By integrating NLP, fine-tuned SLMs, and RAG, the system addresses key challenges in legal information retrieval and interpretation, contributing to a more accessible, efficient, and equitable legal ecosystem.

1.1 Background and Literature Survey

The increasing digitization of legal systems has resulted in an exponential growth of legal information, creating significant challenges in retrieval, interpretation, and decision-making. This phenomenon, widely referred to as information overload, has been extensively studied in knowledge-intensive domains. Hemp [1] argues that when individuals are exposed to excessive and unstructured information, cognitive decision quality declines significantly. In legal context, this issue is amplified due to the hierarchical and interdependent nature of statutes, case laws, and amendments. Legal reasoning therefore requires not only access to information but also structured interpretation of that information.

In developing countries such as Sri Lanka, these challenges are further intensified due to the fragmented nature of legal documentation systems. Tambimuttu [2] characterizes the Sri Lankan legal system as a hybrid structure influenced by Roman-Dutch law, English law, and customary law traditions. While this provides legal diversity, it also introduces interpretational complexity. However, the study remains focused on doctrinal analysis and does not explore computational mechanisms that could assist in legal retrieval or interpretation.

Similarly, the Ministry of Justice [3] emphasizes the importance of structured legal documentation and record management. However, the proposed frameworks rely heavily on manual archival systems and administrative processes. This indicates a lack of integration between legal infrastructure and modern artificial intelligence technologies. As a result, legal accessibility remains dependent on human expertise rather than intelligent automation.

Figure 1 shows Wijesinghe [4] provides a structural breakdown of the Sri Lankan judiciary, detailing hierarchical court systems and jurisdictional authority. Although this contributes to understanding legal structure, it does not address the problem of information accessibility or retrieval efficiency. Fernando [5] further extends this discussion by highlighting that citizens face significant difficulty in locating relevant legal information due to fragmentation across sources. Jayawardena [6] reinforces this argument by demonstrating that even legally trained individuals struggle with navigating legal systems efficiently, indicating that the issue is systemic rather than user-specific.

A critical synthesis of these studies reveals fundamental limitation existing research focuses on describing legal complexity but fails to propose computational frameworks that can bridge the gap between natural language queries and structured legal knowledge. This gap becomes more critical in domains such as labour and employment law, where timely access to correct legal provisions directly impacts individual rights and decision-making outcomes.

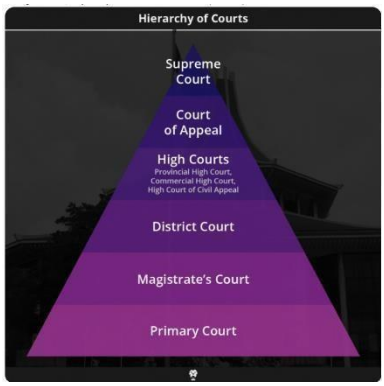

Figure 1: Hierarchy of Courts

Recent advancements in AI, particularly Natural Language Processing (NLP) and Large Language Models (LLMs), have introduced new possibilities for addressing legal information retrieval challenges. However, despite their strong performance in general language tasks, their adaptation to domain-specific legal systems remains limited due to issues such as jurisdictional bias, hallucination, and lack of structured output generation.

Table 1:Large Language Models (LLM)

Paper	**Technologies Used**	**Strengths**	**Limitations**
Shu et al. (2024) – LawLLM	Fine-tuned LLM on U.S. legal corpus	Strong reasoning and retrieval ability	U.S.-only, costly annotations Jurisdiction-specific bias
Sun et al. (2024) – Logic Rules for Legal Case Retrieval	Logic rules + embeddings (no finetuning)	Improves interpretability, explainability	Limited scalability, small datasets

Ma et al. (2023) – CaseEncoder	Knowledge enhanced encoder + pretrained embeddings	Outperforms BERT baselines, captures legal semantics Strong semantic representation	High computational cost

Table.1 shows LawLLM represents one of the most advanced domain-specific legal language models, achieving strong performance through fine-tuning on U.S. statutes and case law. The primary strength of this approach lies in its ability to capture legal reasoning patterns and outperform general-purpose LLMs in structured legal tasks. However, this strength introduces a critical limitation: jurisdictional overfitting.

Legal systems are inherently jurisdiction-dependent, meaning that reasoning structures learned from U.S. law do not directly transfer to Sri Lankan law. This creates a fundamental barrier for global applicability. Additionally, LawLLM relies on extensive annotated datasets, which are expensive to construct and maintain. This makes the approach economically and operationally infeasible for low-resource legal systems. Therefore, while LawLLM demonstrates high accuracy, it lacks adaptability and scalability, which are essential for real-world deployment in diverse legal environments.[7]

This approach integrates symbolic logic rules with embedding-based retrieval, aiming to improve interpretability in legal AI systems. While interpretability is a critical requirement in legal decision support, the reliance on rule-based structures introduces rigidity into the system.

Legal reasoning is inherently dynamic and context-sensitive, yet logic rules enforce predefined constraints that cannot easily adapt to unseen cases. This results in poor generalization ability. Furthermore, maintaining rule sets in large legal corpora becomes increasingly complex over time, reducing maintainability. Thus, the system achieves interpretability at the cost of flexibility and scalability, creating an imbalance that limits real-world usability.[8]

Case Encoder enhances transformer-based representations by integrating legal knowledge graphs. This allows the model to capture inter-case relationships and improve semantic understanding. However, this design introduces significant computational overhead and requires structured legal knowledge graphs, which are not always available.

In developing jurisdictions such as Sri Lanka, legal digitization is still evolving, making such dependency impractical. Additionally, the model is computationally expensive, limiting deployment in lightweight or real-time systems. Therefore, while Case Encoder improves semantic accuracy, it sacrifices efficiency and practical deploy ability [9].

Due to limitations in purely generative models, Retrieval-Augmented Generation (RAG) has emerged as a hybrid solution combining external knowledge retrieval with generative modeling. However, RAG systems themselves introduce new challenges in validation, grounding quality, and structured output consistency.

Table 2:Hybrid Models (LLM + RAG / Embeddings / Extra Layers)

Paper	**Technologies Used**	**Strengths**	**Limitations**
Rahman et al. (2024) – LQ-RAG	RAG + fine-tuned embeddings + feedback loop	Reduces hallucinations, boosts relevance	Proprietary eval agent, no human loop
ICISS (2024) – AI Legal Companion	RAG + ChatGPT	Improves literacy, enhances access	Prone to misinformation, accuracy gaps
Magrani & Fernandes (2024) – Ethical Critique	Policy, human-rights analysis of AI RS	Strong ethical depth	No technical solutions
Thomas et al. (2021) – Quick Check (Thomson Reuters)	Ranking models + citation networks + annotations	Proven industry tool	Proprietary, no public benchmarks

Zeng et al. (2025) – RoSiLC-RS	LLM with abstraction layers	Improves conceptual similarity	Early-stage, no quantitative metrics
Mentzingen et al. (2025) – Precedent Retrieval	Summarization + LLM embeddings	Efficient retrieval, cost-effective	May lose nuance, dataset limited to Brazil
Su et al. (2023) – Caseformer	Pre-training for intercase distinctions	Strong zero-shot retrieval	Retrieval-only, no ranking/recommendation

Table.2 shows LQ-RAG improves factual grounding by combining retrieval mechanisms with fine-tuned embeddings and feedback loops. While this reduces hallucination, it does not eliminate it. The system lacks human validation, which is critical in legal domains where correctness cannot be statistically approximated alone.

A major limitation is that retrieval quality directly impacts output reliability. If retrieval is incomplete or semantically mismatched, the generative model may still produce incorrect legal interpretations. Therefore, LQ-RAG improves reliability but does not guarantee legal correctness [10].

This system integrates RAG with ChatGPT to improve accessibility to legal information. However, it inherits the weaknesses of generative AI, particularly hallucination and inconsistency. The core issue is the absence of strict grounding constraints during generation. Even when relevant documents are retrieved, the model may misinterpret or generalize beyond the retrieved content. This makes it unsuitable for legal decision-making without additional verification layers [11].

This study focuses on ethical implications of AI in legal systems, emphasizing fairness, accountability, and transparency. While conceptually important, it does not provide technical mechanisms for enforcing these principles.

As a result, it remains theoretical and cannot be directly applied to system design. This highlights a recurring gap in legal AI literature: strong ethical discussion without engineering implementation [12]. To address the computational limitations of LLMs, research has shifted toward lightweight models and traditional retrieval systems. However, these approaches often lack deep semantic reasoning capabilities.

Table 3:Small language models (SLM)

Paper	**Technologies Used**	**Strengths**	**Limitations**
Dhanani et al. (2021) – Dictionary-based LDRS	Doc2Vec, dictionary preprocessing	Lightweight, efficient retrieval	No transformers, noisy corpus, frequent dictionary updates
Dhanani et al. (2021) – Cluster-based LDRS	Doc2Vec + Louvain clustering	Scalable, reduced $O(n^2)$, good accuracy	No deep understanding
Dhanani et al. (2021) – P-LDRS	Pre-learned embeddings + Doc2Vec + Spark	Accuracy 0.88, scalable infra	Infra-dependent, still Doc2Vec-based
Zheng et al. (2022) – LawRec	BERT + Skip-RNN	High accuracy (92%), strong baseline	Not global, text-only
Zhou et al. (2018) – Legal Decision Rec.	TextCNN + BiGRU + Attention	Good results across tasks (crime, law, sentencing)	Limited outside China
Nithya et al. (2024) – AI-driven automation	NLP + ML pipeline	Automates summarization, drafting, Q&A	Still early, needs global validation

Table.3 shows these systems rely on Doc2Vec-based representations and clustering techniques for legal document retrieval. While computationally efficient, they fail to capture deep semantic meaning in legal queries.

Cluster-based approaches improve grouping efficiency but introduce rigidity in representation. The P-LDRS model improves accuracy using distributed computing; however, the improvement is infrastructural rather than conceptual. Thus, all variants suffer from a fundamental limitation: lack of semantic reasoning capability [17-19]

LawRec combines BERT with Skip-RNN architectures and achieves strong performance on Chinese legal datasets. However, the model is heavily dataset-dependent and fails to generalize across jurisdictions. This highlights a major limitation of supervised legal models: high accuracy does not guarantee domain transferability [20].

This approach uses CNN, BiGRU, and attention mechanisms for legal prediction tasks. While effective in structured datasets, it lacks adaptability to unseen legal queries. This reflects a key weakness of traditional deep learning models: over-reliance on training data distribution [21].

Across all reviewed literature, three consistent limitations emerge:

1. Jurisdictional dependency of LLMs
2. Lack of reliable grounding in RAG systems
3. Semantic limitations in traditional SLM systems

Critically, no existing system integrates:

- Small Language Model optimized for Sri Lankan law
- Retrieval-Augmented Generation for factual grounding
- Structured legal output generation (Act, Section, Year)
- Domain-specific labour law recommendation capability

Therefore, a significant research gap exists in developing a lightweight, retrieval-enhanced, and jurisdiction-aware legal recommendation system. The proposed system addresses this gap by integrating a fine-tuned Small Language Model with a RAG framework to deliver accurate, structured, and context-aware legal recommendations for Sri Lankan labour and employment law.

1.1.1 Proposed solution

Our solution is a combination of SLMs and RAG, which are fast, accurate, and grounded in law to answer labor and employment law in Sri Lanka. We do not use expensive Large Language Models, but rather our lightweight SLMs are fine-tuned to structured outputs with the help of QLoRA, and the FAISS retrieval strategy is used to retrieve the most relevant acts, sections, and precedents. This makes the responses context-sensitive, section-wise, and clear with easy citations. The strategy is cost-effective, efficient, and scalable and minimizes hallucinations and makes legal information accessible and reliable to lawyers and other members of society.

1.2 Research Gap

Legal documents are inherently lengthy, highly structured, and composed of complex legal terminology. As a result, users often face significant difficulty in identifying relevant legal provisions within a short period of time. This challenge is more severe in Sri Lanka, where legal information is distributed across multiple statutes, case reports, and regulatory frameworks. Furthermore, the highly formal nature of legal language makes it difficult for both legal professionals and non-expert users to correctly map real-world problems to applicable legal sections and enactments.

Although several prior studies have attempted to address legal information retrieval using techniques such as knowledge graphs, clustering, and basic machine learning models, these approaches do not adequately address the specific requirements of Sri Lankan Labour and Employment Law. Most existing systems lack domain adaptation, contextual understanding of local legal frameworks, and integration of modern transformer-based NLP models. Additionally, retrieval-augmented reasoning mechanisms are either absent or not optimized for legal domain-specific use cases.

From a critical perspective, existing research can be categorized into three main limitations:
(i) jurisdictional limitation

(ii) lack of domain-specific adaptation

(iii) absence of modern hybrid AI architectures combining retrieval and generation.

Limitations in Past Research:

- Legal Query RAG [8**]** – Proposes a recursive Retrieval-Augmented Generation framework designed to reduce hallucinations in legal text generation. While this represents a significant advancement in grounding generative models, it has not been fine-tuned for Sri Lankan Labour and Employment Law and lacks structured output formatting such as Act, Section, and Year-level precision.
- LawRec [9] – Utilizes BERT combined with Skip-RNN architecture and achieves approximately 92% accuracy on Chinese legal datasets. Although this demonstrates strong performance in controlled environments, the model is tightly coupled with Chinese legal semantics and lacks adaptability to Sri Lankan Labour Law, where legal structures and terminology differ significantly.
- Cluster-Based LDRS [10] – Uses graph clustering (Louvain) with Doc2Vec for Indian Supreme Court judgments. Scales well but is very domain-specific and doesn't allow natural language queries.
- P-LDRS [11] – Introduces pre-learned embeddings and distributed Doc2Vec training using Spark infrastructure. Although this improves computational scalability and retrieval performance, it remains dependent on shallow semantic representations and does not focus on domain-specific legal reasoning, particularly in employment law contexts.
- Quick Check [14] – A legal case recommendation system based on citation networks and ranking algorithms. While effective for structured U.S. legal datasets, it is heavily dependent on proprietary legal indexing and does not generalize to Sri Lankan legal frameworks. Its reliance on citation-based ranking limits its ability to process natural language queries and reduces flexibility in user interaction.

Table 4:Comparative Analysis of Research Features

Feature	Quick Check	LawRec	Cluster-Based LDRS	P-LDRS	Legal Query RAG	Proposed Project
Transformer Models	No	Yes (BERT)	No	No	Yes	Yes
RAG Integration	No	No	No	No	Yes	Yes
Sri Lankan Focus	No	No	No	No	No	Yes
Labor/Employment Focus	No	No	No	No	No	Yes
Natural Language Queries	Partial	Partial	No	No	Yes	Yes
Legal Sections & Scenarios	No	No	No	No	Partial	Yes
Scalability	Medium	Medium	High	High	Medium	High

1.2.1 Critical Analysis of Existing Research

Table.4 shows demonstrate advancements in legal information retrieval; they fail to address several critical requirements for practical deployment in Sri Lanka. First, most systems are trained on foreign legal datasets (U.S., China, India), resulting in strong jurisdictional bias. This limits their applicability to Sri Lankan legal frameworks, where statutory interpretation and legal structure differ significantly.

Second, traditional systems such as LDRS variants rely on shallow representations (e.g., Doc2Vec), which are insufficient for capturing semantic relationships in legal text. While scalable, they fail to support contextual reasoning required in legal recommendation tasks.

Third, even modern transformer-based systems and RAG-based frameworks focus primarily on improving accuracy or reducing hallucinations, but they do not provide structured legal outputs such as Act name, Section number, and Year of enactment, which are essential for legal usability in real-world scenarios.

Finally, most existing systems lack integration of user-centric design principles. They are either designed for researchers or legal professionals and do not effectively support non-expert users who require simplified legal explanations and scenario-based guidance.

Table 5::Existing system comparison with our system

Feature	AYCA	AIPAZZ	OpenAI	Proposed system
Information Accuracy	Provides only basic definitions and legal references. Correct but shallow, does not map to law sections, enactment years, or scenarios.	Uses searchable legal databases, giving moderately accurate results. However, answers are general and often lack specificity for labor law. And no recommendation	provide information from referring to all available online and social media because of wrong information also given to user .	Trained on authorized Sri Lankan labor law documents. Outputs detailed sections, enactment years, and scenario-based explanations, ensuring high accuracy.
User Reliability	Low reliability. Users must interpret results themselves, which risks misinterpretation by non-lawyers.	Medium reliability. Provides answers, but without reasoning or context. Users still need external legal help.	Low reliability for Sri Lanka. While confident, it often provides foreign laws or unverified information, which is risky.	High reliability. Uses agentic RAG with confidence scoring, ensuring outputs are verified, consistent, and trustworthy

Technology Level	Minimal NLP. No use of transformers or RAG. Functions as a static glossary.	Basic NLP with keyword matching. No transformers or RAG integration.	Uses large transformers based LLMs. Advanced in language ability but not domain-specific, and resource-heavy.	Built on a fine-tuned Small Language Model with Agentic RAG, optimized for low resource environments while maintaining high accuracy.
Practicality (Sri Lanka)	Free and accessible, but too limited in scope and not scalable.	Subscription-based, creating barriers for ordinary users. Not specialized for labor law.	Free to try but not localized. Provides quick answers but with high risk of wrong or irrelevant results.	Free, inclusive, specialized for Sri Lankan labor law. Designed for workers, lawyers, and the public.
Scenario- Based Guidance	No	No	No	Yes

1.2.2 Critical Evaluation of Existing Systems

Table.5 shows existing AI-based legal tools such as AYCA and AIPAZZ provide limited legal information retrieval capabilities but lack deep reasoning or structured legal mapping. AYCA functions primarily as a static legal reference system, offering basic definitions without contextual interpretation. AIPAZZ improves searchability but relies heavily on keyword matching, which limits semantic understanding and reduces accuracy in complex legal queries.

OpenAI-based systems such as ChatGPT demonstrate strong natural language capabilities but are not optimized for Sri Lankan legal contexts. These models often generate generalized or foreign legal interpretations, which may not be applicable or legally valid in Sri Lanka. Furthermore, their probabilistic nature introduces uncertainty, making them unsuitable for formal legal decision support without verification layers.

In contrast, the proposed system addresses these limitations by integrating a fine-tuned Small Language Model with an Agentic Retrieval-Augmented Generation (RAG) framework,

specifically trained on Sri Lankan Labour and Employment Law. This ensures domain-specific accuracy, structured output generation, and improved reliability through retrieval-based grounding

1.2.3 Identified Research Gaps

Based on the critical analysis of existing literature and systems, the following research gaps are identified:

- Lack of fine-tuned transformer-based models specifically for Sri Lankan Labour and Employment Law.
- Absence of Retrieval-Augmented Generation (RAG) frameworks adapted to Sri Lankan legal datasets.
- Limited or no domain-specific legal AI systems focused on Sri Lankan statutory structures and terminology.
- Inability of existing systems to effectively process natural language legal queries with high precision.
- Lack of structured legal output generation including Act name, Section number, and Year of enactment.
- Absence of scenario-based legal explanation mechanisms for better user understanding.
- Limited accessibility for non-expert users due to absence of simplified and guided legal interfaces.

1.2.4 Proposed Solution

To address the identified gaps, this research proposes a domain-specific legal recommendation system based on a fine-tuned Small Language Model integrated with a Retrieval-Augmented Generation (RAG) framework. The system is trained on curated Sri Lankan Labour and Employment Law datasets to ensure jurisdictional relevance and semantic accuracy. It processes natural language queries to generate structured outputs, including relevant Acts, Sections, and Years of enactment, along with scenario-based explanations. An Agentic RAG framework ensures that responses are grounded in verified legal documents, reducing hallucinations and improving reliability. By combining accuracy, efficiency, and accessibility, it aims to improve legal information retrieval and decision support within the Sri Lankan context.

1.3 Research Problem

The legal system in Sri Lanka is inherently complex due to the absence of a unified and structured legal knowledge framework. Legal information is distributed across multiple statutes, sections, amendments, and case decisions enacted over different time periods. This fragmentation significantly increases the difficulty of identifying the most relevant legal provision for a given real-world problem. The issue is particularly evident in the domain of Labour and Employment Law, where both legal professionals and non-expert users often struggle to accurately identify the applicable Act, section, or year of enactment corresponding to a specific legal scenario.

In contrast, many developed countries have adopted AI-driven legal information systems, including digital legal assistants and intelligent case retrieval tools, to support efficient legal decision-making. However, in Sri Lanka, legal information retrieval remains largely dependent on manual search methods, physical documentation, or basic keyword-based digital search engines. Although these approaches can retrieve partial information, they fail to capture semantic meaning, contextual relevance, and legal intent behind user queries. As a result, the retrieved information is often incomplete, inconsistent, or misaligned with the actual legal requirement.

At present, limited digital legal platforms such as AYCA and AIPAZZ exist in Sri Lanka. However, these systems primarily function as basic legal search engines and lack advanced natural language understanding capabilities. They do not incorporate modern Natural Language Processing (NLP) techniques, transformer-based architectures, or contextual reasoning mechanisms. Furthermore, these systems are not specifically designed for Labour and Employment Law, which is one of the most frequently required legal domains in practical scenarios. Another major limitation is that these platforms are primarily designed for legal professionals, making them less accessible and less usable for ordinary citizens who require simplified legal explanations.

1.3.1 Empirical Evidence

A survey conducted with 40 participants, including lawyers, law students, and members of the general public, revealed that more than 90% of respondents identified the need for an intelligent legal decision-support system tailored to the Sri Lankan legal context. The survey further highlighted three major challenges:

- Delay in accessing relevant legal documents
- Difficulty in interpreting fragmented and technical legal language
- Lack of user-friendly systems supporting natural language queries

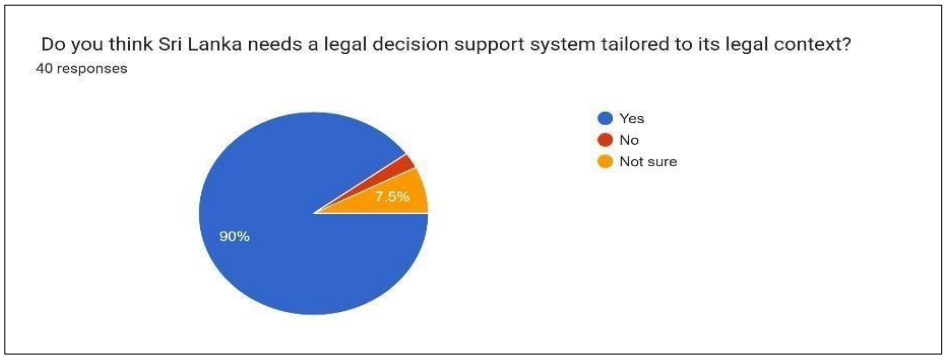

Figure 2:Survey of existing legal information

Figure 2 shows that the absence of an intelligent legal retrieval system not only reduces efficiency but also increases the risk of misinterpretation, delays in legal decision-making, and potential barriers to justice.

1.3.2 Technological Limitations in Existing Systems

From a technical perspective, existing systems in Sri Lanka do not incorporate modern deep learning-based language models such as T5, DistilBERT, MiniLM, or Legal-BERT, which have demonstrated strong performance in semantic understanding, classification, and text generation tasks in other domains. More importantly, they lack integration of Retrieval-Augmented Generation (RAG) frameworks, which combine information retrieval with generative modeling to improve factual grounding and reduce hallucination.

The absence of these technologies results in several critical limitations:

- Poor understanding of natural language legal queries
- Lack of contextual awareness in legal recommendations
- Inability to generate structured outputs such as Act, Section, and Year
- Low adaptability to Sri Lankan legal language and structure
- Limited usability for non-expert users

1.3.3 Problem Statement

Despite the increasing need for efficient and accessible legal information systems, Sri Lanka currently lacks an AI-driven legal recommendation framework that integrates modern Natural Language Processing (NLP), transformer-based models, and Retrieval-Augmented Generation (RAG), specifically tailored for Labour and Employment Law.

This absence creates significant barriers to legal accessibility, reduces efficiency in legal research and decision-making, and contributes to inequality in access to legal knowledge among citizens and professionals.

1.4 Objectives

1.4.1 Main Objectives

The main objective of this research is to develop a specialized AI-based legal assistant for Sri Lankan Labour and Employment Law that provides more reliable, context-aware, and structured legal guidance compared to general-purpose conversational AI systems. The proposed system aims to improve legal accessibility and decision support by combining a fine-tuned large-scale language model with retrieval-augmented mechanisms grounded in authoritative legal sources.

1.4.2 Specific Objectives

The proposed system is designed to achieve the following specific objectives:

- To transform unstructured Sri Lankan labour law resources, including statutes, regulations, and case law documents, into a structured and machine-readable training dataset suitable for model fine-tuning.
- To fine-tune the Qwen3-8B language model using labour-law-specific datasets in order to enhance domain adaptation and improve legal reasoning capability within the Sri Lankan context.
- To design and enforce a consistent structured output format that ensures all model responses include key legal elements such as Act name, Section number, and Year of enactment.
- To integrate a Retrieval-Augmented Generation (RAG) framework that grounds model outputs in relevant legal context by retrieving supporting information from legal statutes and case law databases.
- To enable the system to generate practical and actionable recommendations for real-world workplace disputes by mapping user queries to applicable legal provisions.
- To design a scalable and efficient deployment pipeline using Modal inference services, ensuring low-latency inference and production-level accessibility of the legal assistant system.

2. METHODOLOGY

2.1 Key Technical Foundations of the Proposed System

This research is built on a set of technical and methodological foundations that are jointly required to generate accurate, structured, and reliable legal recommendations in the Sri Lankan labour-law domain. Labour-law queries are rarely expressed in formal legal language. In practice, users often describe disputes through long and incomplete narratives involving termination, salary issues, workplace rights, resignation pressure, harassment, or contractual ambiguity. To interpret such queries correctly, the system must capture semantic meaning, contextual order, legal terminology, and practical legal intent. For this reason, the methodology is intentionally multi-layered and integrates deep learning, natural language processing, OCR-based digitization, transformer-based small language models, retrieval-grounded inference, and transfer learning.

2.1.1 Deep Learning

Deep learning serves as the foundational computational paradigm of the proposed system because labour-law recommendation is fundamentally a contextual interpretation task. Legal problems are rarely expressed in a single fixed format; the same dispute may be described in formal legal wording, conversational language, complaint-style narration, or emotionally framed user text. As a result, a rule-based approach alone is often insufficient. Deep neural architecture enables the system to learn high-dimensional semantic relationships from data and to interpret legal meaning beyond literal keyword matching. In this study, deep learning provides the foundation for semantic representation, context-sensitive recommendation generation, retrieval support, and structured legal response production.

2.1.2 Natural Language Processing

Natural Language Processing functions as the linguistic core of the proposed legal recommendation system. It is used to preprocess legal documents, interpret user queries, structure model inputs, and generate user-friendly legal outputs. Since legal texts are lengthy, formal, and structurally dense, NLP is essential for transforming unstructured legal information into machine-understandable and human-readable representations. It therefore bridges the gap between complex legal language and accessible legal assistance for both professionals and non-expert users.

2.1.3 OCR-Based Legal Document Digitization

A significant portion of the legal data, including labour-law statutes and case-related materials, was available only in scanned or image-based PDF formats. Accordingly, the system begins with OCR-based digitization, converting rendered page images into machine-readable text. Due to OCR-induced noise such as broken lines, spacing inconsistencies, punctuation errors, and page artifacts the extracted text is systematically cleaned, normalized, and structured into legally meaningful units. This preprocessing stage is critical, as the accuracy and consistency of digitized data directly influence downstream training, retrieval performance, and overall system reliability.

2.1.4 Transformer-Based Small Language Models

Transformer-based Small Language Models act as the main reasoning engine of the proposed system. They are well suited to legal recommendation tasks because transformer attention mechanisms can model long-range dependencies across lengthy legal passages, user narratives, and retrieved statutory context. A small-language-model strategy was selected to balance contextual understanding with computational feasibility. This enables the project to remain deployable and efficient while still achieving domain-specialized legal reasoning when combined with retrieval grounding and structured output control.

2.1.5 Retrieval-Augmented Generation and Vector Retrieval

Retrieval-Augmented Generation is a key methodological pillar because legal recommendation must remain grounded in evidence rather than rely solely on model memory. In this system, legal content is first retrieved from an indexed knowledge base and then supplied to the model during generation. Dense vector retrieval and FAISS-based similarity search are used to identify semantically relevant legal text even when user wording differs from statutory phrasing. This retrieval-first design improves factual reliability, reduces hallucination, and makes the final recommendation more explainable and traceable.

2.1.6 Transfer Learning and Domain Adaptation

Transfer learning is used to adapt a pre-trained transformer model to the specialized domain of Sri Lankan labour law. Instead of training from random initialization, the model begins with broad linguistic knowledge and is then fine-tuned on a curated legal dataset containing labour-

law terminology, statutory references, and structured response patterns. This domain adaptation improves local legal relevance, reduces cross-jurisdictional drift, and makes the model better suited to generate recommendations that are aligned with Sri Lankan labour-law practice.

2.2 Integrated Research Approach and System Methodology

This research adopts an implementation-driven methodology integrating legal data engineering, model adaptation, retrieval design, system integration, and evaluation to develop a controlled legal recommendation pipeline. The system defines explicit inputs, processing rules, validation checkpoints, and measurable outputs to ensure reliability within the Sri Lankan labour-law context, where queries are informal and legal sources are fragmented. The pipeline spans document acquisition, OCR-based extraction, JSONL curation, transformer fine-tuning, and Retrieval-Augmented Generation. It further incorporates structured parsing, confidence scoring, and iterative testing. This end-to-end design ensures that recommendation quality is treated as a system-level property, enabling accurate, interpretable, and deployable legal decision support.

2.2.1 Agile Principles Applied in the Project

Agile principles were applied in this project as a delivery and quality-governance strategy rather than a generic project management approach. Figure 1 shows agile methodology. The legal recommendation system was developed through iterative cycles, each producing measurable outputs related to ingestion reliability, retrieval quality, schema-constrained generation, parser robustness, and end-to-end performance. This approach was chosen because RAG-based systems involve strong cross-component interactions, were changes in retrieval, prompts, or models can impact overall system behavior. Therefore, iterative, test-driven development was more reliable than a linear implementation.

The Agile process incorporated short development loops with script-level validation, risk-first prioritization of critical failures (e.g., JSON parsing errors, scope leakage), continuous contract alignment between system components, artifact-based retrospectives, and clear Definition of Done criteria. Structured output reliability was achieved through iterative refinement of prompts, parsing, and schema validation. Backlog management was layered into capability, implementation, and quality tasks to prevent hidden technical debt. Retrospectives used root-

cause analysis (data, retrieval, generation, orchestration), improving debugging precision. Additionally, each iteration produced reproducible artifacts (JSON, CSV, reports), ensuring measurable, testable, and consistent improvements aligned with legal system objectives.

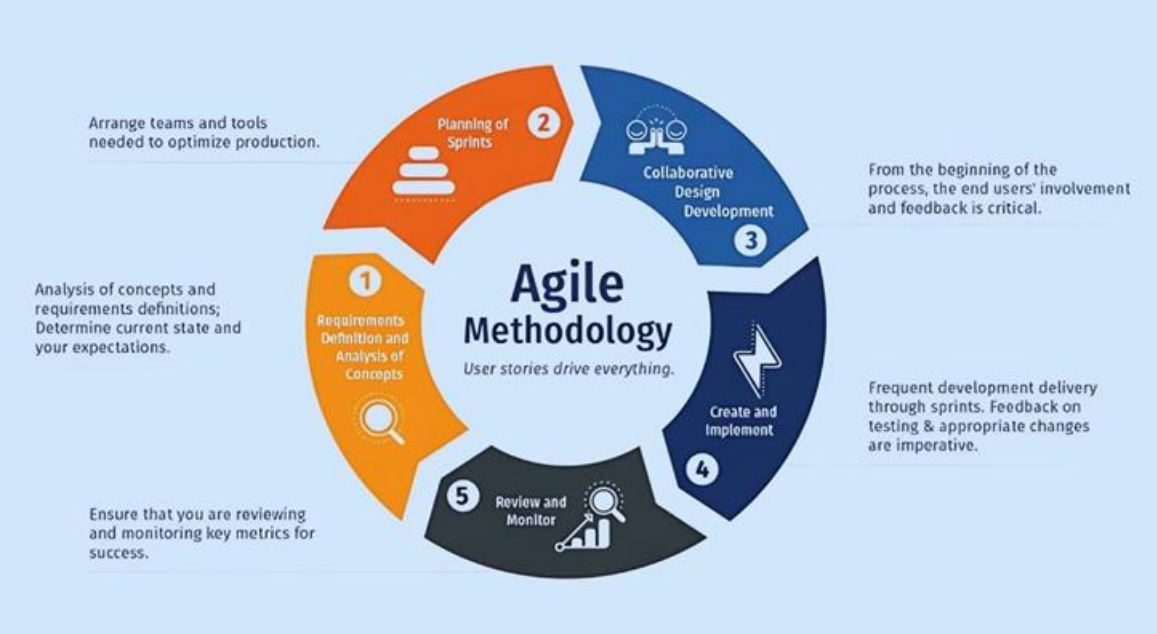

Figure 3: Agile Methodology

2.2.2 Feasibility Study and Planning

A multi-dimensional feasibility assessment was conducted before and during implementation, covering technical, operational, legal-domain, economic, and schedule aspects to ensure realistic deploy ability and extensibility. Technical feasibility evaluated the full legal RAG stack, including FastAPI orchestration, PostgreSQL storage, FAISS vector retrieval, embedding services, and transformer inference via Modal and Ollama. Operational feasibility focuses on maintainability through a modular architecture separating the main server (legal orchestration) and model server (inference), enabling independent updates to retrieval, parsing, and model components. Legal-domain feasibility ensured accurate labour-law-specific outputs and safe handling of out-of-scope queries. Economic feasibility was supported through an open-source-first approach with selective cloud usage. Schedule feasibility was achieved via phase-gated milestones backed by measurable artifacts, including training summaries, evaluation reports, and system test results, ensuring verifiable progress and reduced integration risk.

2.2.3 Requirement Gathering and Analysis

Requirements were gathered from three perspectives such as legal advisory behavior, software and reliability, and deployment and operations. This ensured that research objectives were translated into implementable and testable system behavior rather than abstract goals. Functional requirements included ingestion of legal documents, chunk-based knowledge creation, semantic retrieval of legal context, structured legal recommendation generation under a fixed schema, out-of-scope query handling, and storage of query history with metadata for traceability. Non-functional requirements covered stable JSON output parsing, inference performance, modular maintainability, FAISS index consistency with metadata, and robust error handling with observability.

Stakeholders included end users, legal analysts, administrators, and developers. A schema-first strategy defined structured outputs (summary, violations, cases, reasoning, action, limits, confidence). Requirement traceability linked retrieval to embedding/FAISS, generation to SLM and prompt handling, output reliability to parsing and validation, and operations to administrative and evaluation scripts. This decomposition reduced ambiguity and ensured alignment between system design and implementation. Table 6 shows a practical phase layout used in this project

Table 6:A practical phase layout used in this project

Phase	**Main Outputs**
Data Preparation	OCR text, cleaned legal corpus, and validated JSONL records
Model Adaptation	Fine-tuned adapters, merged model exports, and GGUF exports
Retrieval and Indexing	Chunk generation, embeddings, and FAISS index synchronization
Service Integration	Query pipeline, proxy inference path, parser logic, and confidence synthesis
Evaluation	Quality metrics, speed statistics, category analysis, and end-to-end validation
Research Reporting	Methodology chapter, results and discussion chapter, and final summaries

A representative use-case flow for the system is as follows:

1. user submits a labour-law query
2. query embedding is generated
3. FAISS retrieval returns candidate chunks
4. document-diverse reranking builds a bounded context window
5. the model generates a structured response
6. the parser validates and normalizes the output
7. the result and metadata are stored and returned

2.2.4 Research Design and Methodological Framing

The study is framed as applied design science with system-engineering validation. Its purpose is to design, implement, and evaluate a domain-specific legal recommendation system that can operate under realistic deployment constraints while remaining methodologically defensible. Success is therefore defined through a composite quality model that includes legal grounding, schema stability, scope control, recommendation usefulness, interpretability, and operational reliability. Rather than treating the methodology as a purely conceptual description, the study ties each stage to implemented components such as the OCR pipeline, JSONL data structure, Qwen fine-tuning workflow, FAISS-based retrieval process, FastAPI orchestration layer, Modal-backed inference service, and evaluation scripts. This framing improves reproducibility because the chapter reflects actual system behavior rather than an abstract design that is not connected to the running implementation.

The methodological stance also uses evidence triangulation. Findings are not accepted based on a single metric family alone. Instead, claims about quality are supported through multiple evidence layers such as dataset validation checks, fine-tuning reports, structured-output evaluation, retrieval testing, and end-to-end system behavior. This reduces the risk of overstating performance based solely on train loss, semantic similarity, or anecdotal examples. In legal AI, where errors can have serious user consequences, such triangulation is essential for responsible research reporting.

2.2.5 Data Collection and Source Preparation

Figure 4 shows online data collection followed a domain-focused strategy that prioritized Sri Lankan labour and employment law over broad legal-topic coverage. Labour-law statutes, legal references, and case-related materials were collected as the primary knowledge base for both supervised training and retrieval support. The aim was to ensure that the system specialized in labour-law assistance rather than diluted across unrelated legal domains. Figure 5 shows a notable portion of the source material that existed as scanned PDFs or non-machine-readable documentary resources. This created a need for digitization before the data could be used in model training or retrieval indexing.

At this stage, source preparation also included an early relevance-filtering process. Materials were reviewed and selected based on their practical value for common labour-law scenarios such as termination, wage disputes, working conditions, employee rights, procedural guidance, and employment-related statutory interpretation. This domain-focused collection strategy improved in-scope precision and ensured that the dataset reflected the intended use case of the system. Limited out-of-scope or non-labour examples were retained to support boundary conditioning, but these remained secondary to the labour-law core of the corpus.

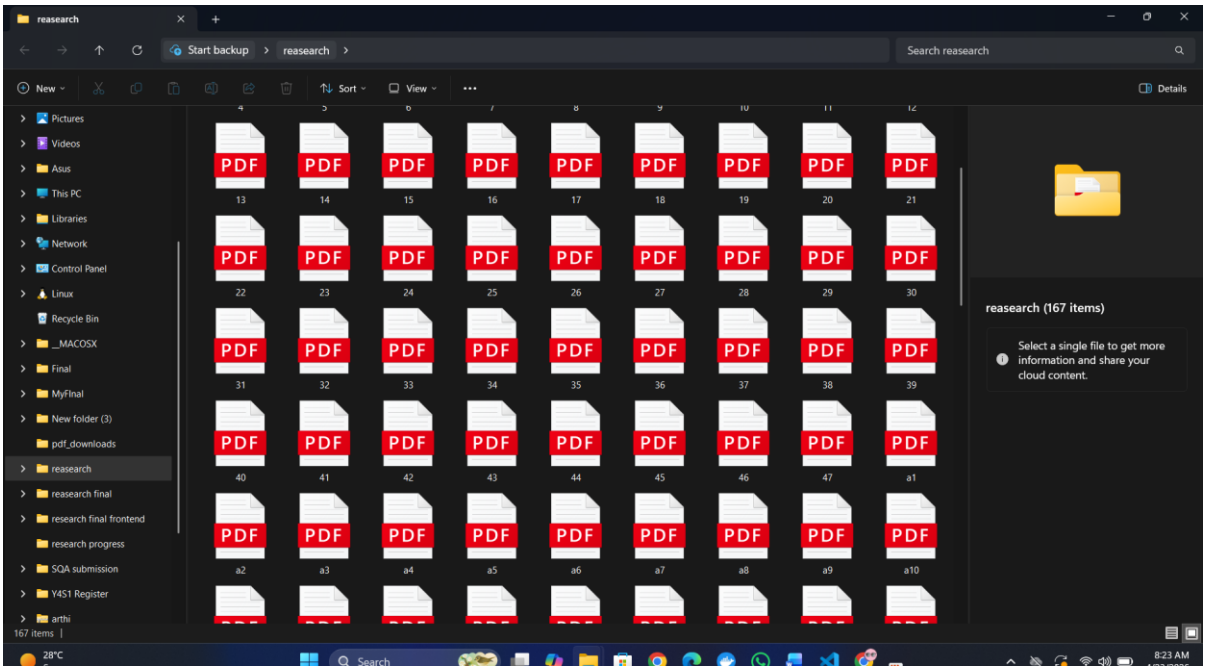

Figure 4:Online data collection

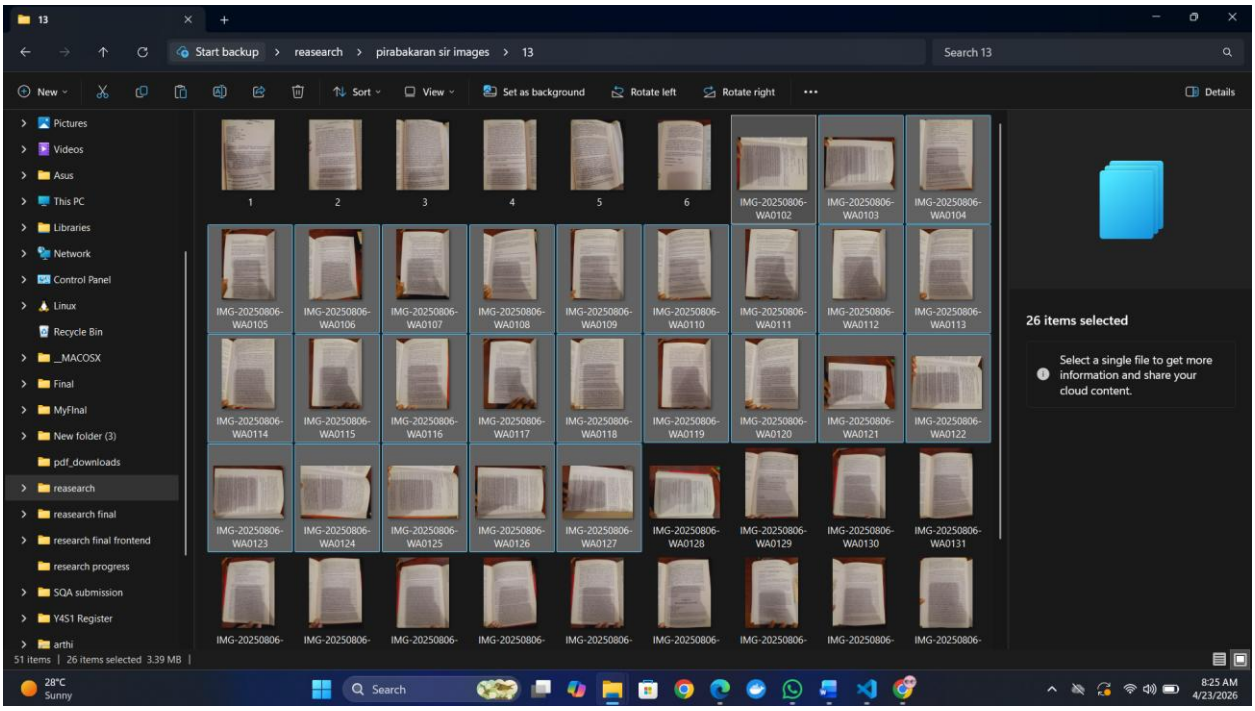

Figure 5:Scan data collection

2.2.6 OCR Extraction, Cleaning, and Structured Dataset Construction

After source acquisition, scanned and image-based legal documents were processed through an OCR-based extraction stage, where each page was converted into machine-readable text. Due to the complexity of legal documents including headers, footers, citations, and irregular formatting, the raw OCR output often contained noise such as broken words, duplicated fragments, inconsistent spacing, and punctuation errors. Therefore, OCR was treated as a controlled preprocessing stage rather than a simple conversion step.

The extracted text was then cleaned, normalized, and segmented into meaningful legal units by removing artifacts and restructuring content into a consistent format. Gemini-assisted structuring was applied to identify key legal elements such as case names, Act references, section references, and issue summaries. These were converted into a unified JSONL dataset with a standard schema containing instruction, retrieved context, and structured output fields, supporting both model fine-tuning and evaluation consistency.

2.2.7 Data Governance, Splitting, and Schema Validation

Before training, the dataset underwent governance checks to ensure internal validity and reduce downstream instability. Schema validation was applied to confirm that each sample

contained required fields and that outputs followed a consistent legal-response structure. This included checks for key completeness, structural consistency, confidence range validity, and field integrity. These controls were essential because the system produces machine-parseable structured outputs rather than free-form text.

After validation, the dataset was split into training, validation, and test sets. Figure 6 shows splitting and schema validation. The final dataset consisted of 1,035 samples, divided into 775 training, 156 validation, and 104 test samples. This separation ensured unbiased evaluation and prevented data leakage. Dataset governance was treated as a core methodological stage since stable data structure directly impacts training stability, retrieval quality, and reproducible evaluation.

Figure 6:Splitting, and Schema Validation

2.2.8 Model Fine-Tuning Strategy

The core model adaptation stage used transfer learning to specialize a pre-trained transformer model for the Sri Lankan labour-law domain. Instead of training a new model from the beginning, the study fine-tuned Qwen using a parameter-efficient strategy based on LoRA and

QLoRA within an Unsloth-centered notebook workflow. This reduced computational cost while preserving strong linguistic capability. The fine-tuning pipeline included environment checks, configuration setup, split loading, schema-aware formatting, training execution, validation monitoring, export generation, and artifact reporting.

Methodologically, the model was trained not only to produce relevant language but also to follow a structured legal-response format. Through repeated exposure to JSONL instruction-context-output examples, the model learned to map user issues to legally grounded outputs containing legal references, reasoning, and action-oriented guidance. Appropriate validation procedures were used to reduce overfitting and maintain generalization. Figure 7 shows model training. The fine-tuning stage therefore served both as a domain adaptation process and as a response-format alignment process, ensuring that the model could operate effectively in the broader system pipeline.

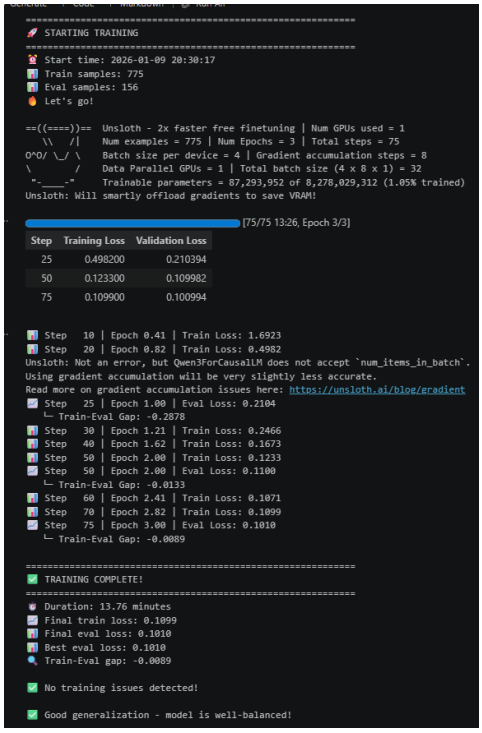

Figure 7:model training

2.2.9 Retrieval-Augmented Generation and Vector Indexing

In parallel with model adaptation, the system implemented a Retrieval-Augmented Generation mechanism to improve factual grounding and reduce hallucination. Legal documents were divided into smaller chunks and converted into dense vector embeddings using a semantic

embedding model. These embeddings were stored within a FAISS-based index so that semantically relevant legal content could be retrieved efficiently at runtime. The use of vector retrieval was particularly important because users typically describe labour disputes in everyday language rather than using exact statutory wording.

The retrieval process was designed as more than a simple top-k search. A broad initial candidate window was used to improve recall, followed by document-diverse reranking so that the final context window did not over-concentrate on a single source. Context size was also bounded to preserve prompt quality and avoid unnecessary dilution. This design improved the quality of retrieved evidence and strengthened the overall recommendation pipeline. In methodological terms, retrieval was treated as a first-class quality layer rather than as a secondary feature, because grounded recommendations depend directly on the quality, diversity, and relevance of the retrieved context.

2.2.10 Runtime Inference, Parsing, and Confidence Synthesis

During runtime, the system follows a controlled inference pipeline. A user submits a labour-law query through the application interface, which is embedded and matched against a FAISS index to retrieve relevant legal chunks. After reranking and context assembly, the retrieved evidence is combined with the query and passed to the fine-tuned model. The architecture separates orchestration and inference, where the FastAPI server handles retrieval, prompting, parsing, logging, and persistence, while a proxy-connected Modal deployment performs model inference.

Raw model outputs are not directly returned. Instead, they undergo structured parsing, sanitization, and repair, including control-character cleanup, JSON extraction, truncated-output recovery, and mapping to a predefined legal schema. A hybrid confidence score is then generated using retrieval quality, output completeness, model behavior, and context richness. This stage is critical as it transforms the model into a controlled legal recommendation component that is testable, monitorable, and suitable for production-like environments.

2.2.11 System Integration, Architecture, and Observability

The complete system is integrated into a full-stack architecture. Figure 8 shows system architecture system. The backend is implemented using FastAPI and is responsible for query handling, embedding generation, retrieval, reranking, prompt assembly, parser safeguards,

confidence computation, and database persistence. A local proxy connects the backend to a Modal-hosted inference service where the fine-tuned model runs on GPU infrastructure. The frontend is implemented in React to provide a user-friendly interface for query submission and output presentation. This architectural separation improves maintainability, isolates orchestration logic from inference runtime concerns, and allows individual layers to evolve without destabilizing the entire system.

Observability is treated as part of the methodology rather than an operational afterthought. Query histories, context metadata, model timing information, and response fields persisted to support traceability and later analysis. This is essential in legal AI because the system must remain diagnosable when unexpected outputs occur. By logging into the conditions under which a recommendation is generated, the methodology supports auditability, failure analysis, and longitudinal system improvement.

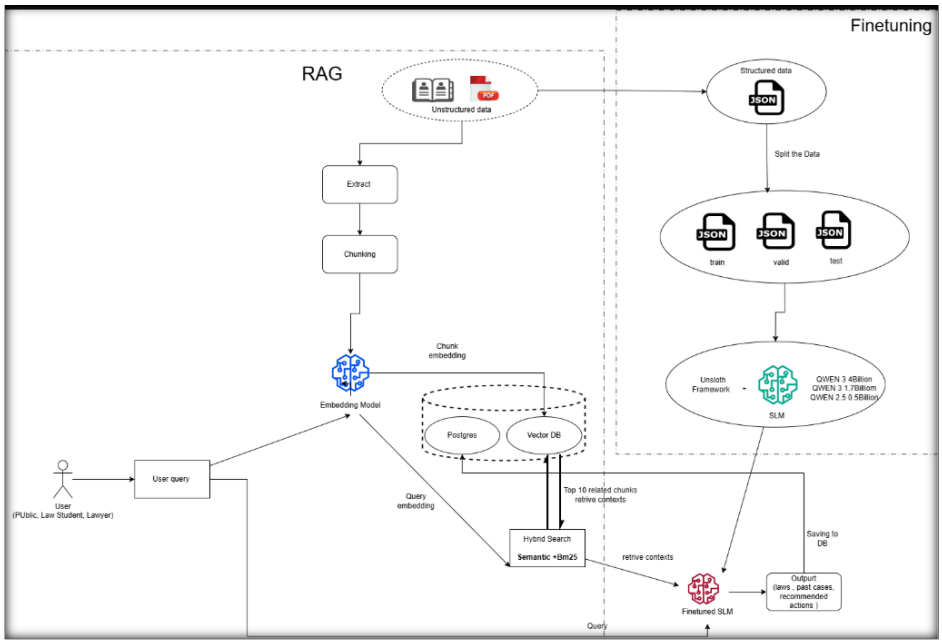

Figure 8:System Architecture Diagram

2.2.12 Evaluation, Reliability Controls, and Iterative Refinement

The evaluation methodology is multi-layered. Model-centric evaluation focuses on JSON validity, semantic quality, category behavior, and speed statistics, while system-centric evaluation examines retrieval behavior, model server connectivity, index integrity, and end-to-end legal scenario performance. This separation allows failures to be diagnosed more precisely.

For example, a structurally valid but weakly cited answer may indicate retrieval precision problems rather than model collapse, whereas malformed JSON may point to generation or parsing instability.

Reliability controls are embedded directly into the methodology. Out-of-scope prevalidation, structured limits fields, schema enforcement, parser safeguards, and iterative test-driven refinement are all part of the research process. The methodology also explicitly acknowledges domain boundaries and avoids claiming universal legal competence beyond Sri Lankan labour law. In addition, iterative refinement was guided by measurable artifacts rather than subjective impressions. Training summaries, evaluation reports, response-quality checks, and pipeline tests formed the basis for continuous improvement. This made the overall approach not only technically rigorous but also suitable for future deployment maturation and further academic extension.

2.2.13 Project Timeline and Gantt Chart

To support project planning and execution monitoring, a Gantt-chart view is recommended as part of the methodology chapter. The chart can summarize the timing, sequence, and overlap of major research activities, including data collection, OCR preprocessing, JSONL dataset construction, model fine-tuning, RAG integration, full-stack implementation, testing, evaluation, and report writing. Including this visual element improves the clarity of project scheduling and helps demonstrate that the work followed a structured and phase-based implementation plan.

A reserved space or sample figure is therefore inserted below for the final Gantt chart. This can be replaced later with the completed chart once the final project timeline is confirmed.

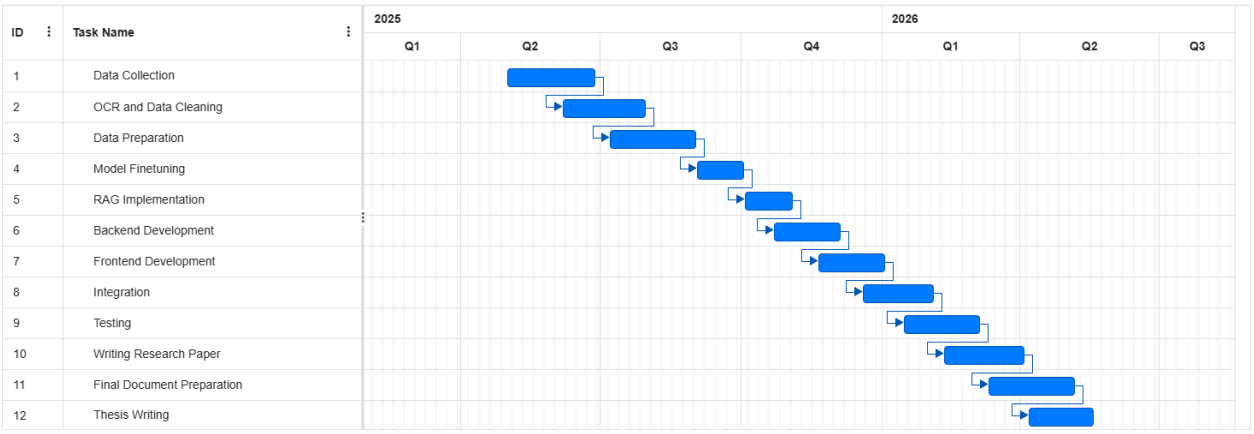

Figure 9:Gantt Chart

2.3 Summary of Methodology

The methodology integrates legal data preparation, transformer-based adaptation, retrieval-grounded inference, structured output control, and deployable software engineering into a unified framework. It treats legal recommendation as a systems problem rather than a purely language-model task, which is essential in the Sri Lankan labour-law context where systems must interpret user narratives, retrieve legal context, generate structured outputs, and indicate uncertainty within a real application environment.

The process begins with legal-source acquisition and OCR-based digitization, as many legal documents exist in scanned or PDF formats. This stage is critical because downstream retrieval and generation depend on source fidelity. OCR outputs are cleaned, normalized, and restructured to remove noise such as formatting inconsistencies and extraction errors. Gemini-assisted processing is then used to convert cleaned data into a unified JSONL format, ensuring consistent instruction, context, and output structures for fine-tuning.

Model adaptation is performed using transfer learning and parameter-efficient fine-tuning instead of training from scratch. This improves feasibility while maintaining strong language capability and aligns the model with a controlled output schema. Structured responses enhance interpretability, parsing reliability, and safe user interaction compared to free-form generation.

Retrieval-Augmented Generation (RAG) provides factual grounding by retrieving relevant legal chunks from a FAISS-based vector index before generation. The retrieval pipeline includes embedding, similarity search, reranking, and context assembly, enabling semantic matching between user queries and legal provisions even when phrasing differs. This improves legal grounding and makes system behavior more interpretable by linking outputs to retrieval quality.

Overall, the methodology balances academic rigor and engineering practicality by emphasizing schema governance, reproducibility, layered evaluation, and modular deployment using FastAPI, proxy-based serving, and structured inference pipelines. It provides both a research framework and an operational foundation for a trustworthy Sri Lankan labour-law recommendation system.

2.4. Commercialization aspects of the product

2.4.1 Product Overview and Value Proposition

The proposed Sri Lankan Labour Law Recommendation System is designed as a domain-specific Legal AI product that provides structured, explainable, and context-aware legal guidance for labour and employment scenarios. Unlike generic AI chat systems, the product emphasizes retrieval-grounded legal reasoning, structured outputs, and jurisdiction-specific relevance, making it suitable for real-world legal assistance.

The primary value proposition lies in:

- Reducing dependency on manual legal consultation for common labour issues
- Providing quick, structured, and understandable legal insights
- Supporting both professionals (lawyers, HR officers) and non-expert users
- Ensuring explainability through referenced legal acts and reasoning

2.4.2 Target Market and Users

The system is designed for multiple stakeholder groups within the Sri Lankan legal and employment ecosystem:

- Employees seeking guidance on termination, salary disputes, and rights
- Employers and HR departments requiring compliance assistance
- Legal professionals needing quick reference support

- Law students and researchers for academic purposes

  The increasing adoption of digital legal tools and AI-assisted services indicates strong market potential for such a specialized system.

2.4.3 Deployment Model and Architecture for Commercial Use

The system is designed to support scalable and flexible deployment models:

- Software-as-a-Service (SaaS) model via a web-based interface
- API-based integration for law firms or enterprise HR systems
- Cloud deployment using GPU-backed inference services for scalability
- On-premise deployment (optional) for organizations with strict data policies

  The separation between the main orchestration server and the model inference server enables independent scaling, improving cost efficiency and performance.

2.4.4 Legal, Ethical, and Regulatory Considerations

Since the system operates in a legal domain, commercialization must consider:

- Disclaimer enforcement: The system provides guidance, not legal advice
- Data privacy compliance: User queries and legal data must be securely handled
- Bias and fairness control: Ensuring balanced recommendations
- Jurisdiction limitation: Clearly restricting outputs to Sri Lankan labour law

  Ethical AI practices such as transparency, explainability, and controlled outputs are critical for user trust and regulatory acceptance.

2.4.5 Competitive Advantage

The system differentiates itself from generic AI tools through:

- Domain-specific fine-tuning on Sri Lankan labour law
- Retrieval-Augmented Generation for factual grounding
- Structured and machine-parseable legal outputs
- Integrated confidence scoring and limitation handling

  This combination makes the system more reliable and practical for real-world legal scenarios.

2.5 Project Requirements

2.5.1 Functional requirements

Legal Query Processing

- The system shall accept natural language queries related to labour-law scenarios.
- The system shall preprocess and normalize input queries for downstream processing.

Legal Document Ingestion

- The system shall ingest legal documents in PDF and scanned image formats.
- The system shall apply OCR-based extraction for non-machine-readable documents.

Data Structuring and Storage

- The system shall convert extracted content into structured JSONL format.
- The system shall store processed data for retrieval and analysis.

Semantic Retrieval

- The system shall generate query embedding.
- The system shall perform FAISS-based vector similarity search.
- The system shall rerank results for relevance and diversity.

Legal Recommendation Generation

- The system shall generate structured outputs using a fine-tuned transformer model.
- Outputs shall include summary, violations, supporting cases, reasoning, recommended actions, limitations, and confidence score.

Structured Output Validation

- The system shall validate outputs against a predefined schema.
- The system should repair or reject invalid outputs.

Out-of-Scope Handling

- The system shall detect non-labour-law queries and return fallback responses.

Data Persistence and UI Interaction

- The system shall store queries, retrieved context, and outputs for traceability.
- The system shall provide a web interface for input and structured result display.

2.5.2 Non-functional requirements

Performance

- The system shall generate responses within an acceptable latency (e.g., a few seconds).
- The system shall handle multiple concurrent user requests efficiently.

Reliability

- The system shall ensure consistent and stable output generation.

Scalability

- The system shall support scaling of model inference and backend services.
- The architecture shall allow independent scaling of retrieval and inference components.

Maintainability

- The system shall follow a modular architecture (frontend, backend, model server).
- Components shall be independently upgradable without affecting the entire system.

Usability

- The system shall provide a simple and intuitive user interface.
- Outputs shall be clear, structured, and understandable to non-expert users.

Security and Privacy

- The system should protect user data and query information.
- The system shall ensure secure communication between components.

Accuracy and Legal Reliability

- The system shall provide legally relevant and context-aware recommendations.
- The system shall minimize hallucination through retrieval-grounded generation.

Interoperability

- The system shall support API integration with external systems (e.g., HR platforms, legal tools).

Robustness

- The system shall handle incomplete, ambiguous, or noisy user inputs.
- The system shall recover gracefully from errors such as parsing failures or retrieval issues.

Compliance and Ethical Constraints

- The system shall include disclaimers indicating that outputs are not a substitute for professional legal advice.

2.6 Testing and Implementation

2.6.1 Testing

Overview of Testing Phase

The testing phase of the Arise Legal Retrieval-Augmented Generation (RAG) system was conducted to evaluate the performance, accuracy, reliability, and robustness of the complete pipeline. The system was tested across multiple layers, including vector database integrity, document ingestion, retrieval accuracy, model inference, and full end-to-end pipeline performance. The objective was to ensure that the system can reliably process legal queries and generate accurate, structured responses based on Sri Lankan labour law documents.

Testing was performed in a controlled environment consisting of a FastAPI backend (Port 5005), a model proxy server (Port 5007), and a Modal A10G GPU instance. The system used FAISS (IndexFlatIP) as the vector store with 6313 indexed embeddings generated using Gemini embeddings. The fine-tuned Qwen3-8B model (sri-legal-8b) was used for response generation.

Vector Database and System Validation

The FAISS vector database was first evaluated to ensure correct indexing and retrieval readiness. The system contained a total of 6313 vector embeddings with 1024 dimensions, distributed across 100 successfully indexed documents. However, 7 documents were found to have failed during ingestion and contained zero chunks, indicating partial data loss during preprocessing. Figure 10 shows vector database testing.

Despite this, the vector database was operational and showed consistent synchronization between stored chunks and FAISS vectors. Most indexed data belonged to employment-related legal domains, which influenced retrieval focus toward labour law topics.

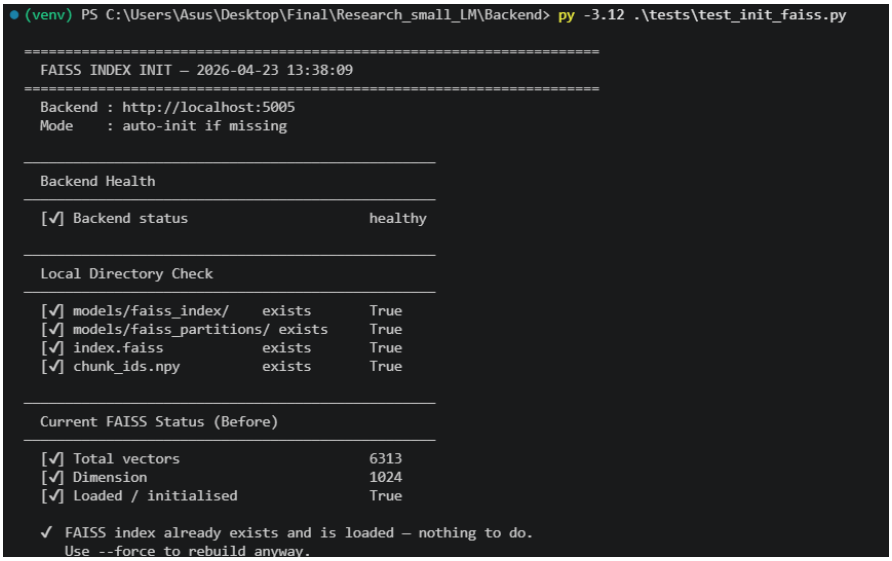

Figure 10:Vector Database testing

Document Upload and Ingestion Testing

The document ingestion pipeline was tested by uploading PDF legal documents into the system. Each document underwent parsing, chunking, embedding generation, and indexing into FAISS. The system successfully processed documents into an average of 51 chunks per upload, with an ingestion time of approximately 9.7 seconds per document.

The upload pipeline demonstrated stable performance, with proper conversion of unstructured legal text into searchable vector representations. No inconsistencies were observed between the database and vector index for successfully processed documents. Figure 11 shows document upload testing.

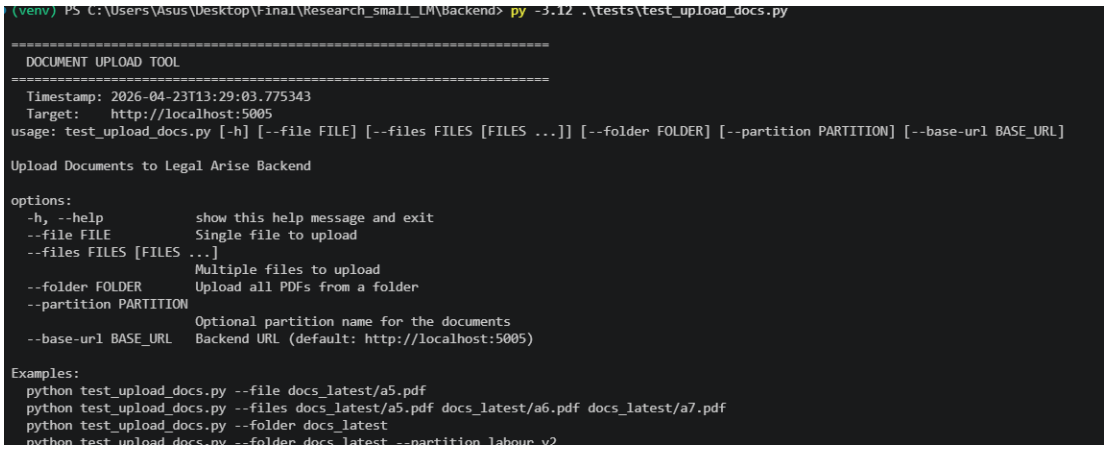

Figure 11:Document upload testing

Retrieval Accuracy Testing

Retrieval performance was evaluated using a set of structured legal queries aligned with the indexed corpus. Figure 12 shows retrieval accuracy testing. The system achieved an overall retrieval accuracy of 90%, with most in-scope queries achieving 100% recall and high keyword coverage.

The FAISS-based similarity search effectively retrieved relevant legal sections from acts such as the Payment of Gratuity Act, Industrial Disputes Act, Termination of Employment Act, and Trade Unions Ordinance. However, one test case involving dual proceedings showed reduced recall at 50%, indicating slight inconsistency in complex multi-concept queries.

Out-of-scope detection was partially successful, correctly identifying property law queries, but occasionally misclassifying commercial law queries as relevant to labour law. This limitation was attributed to the language model rather than the retrieval system.

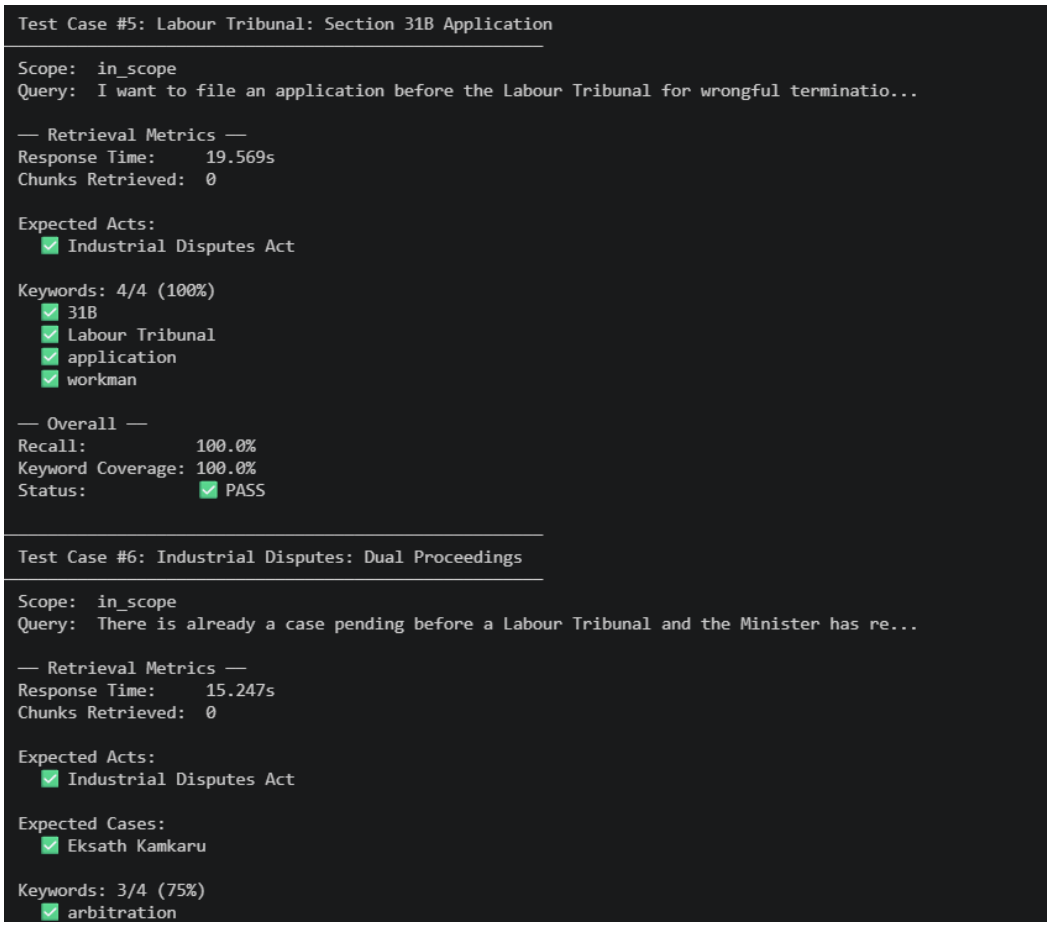

Figure 12:Retrieval accuracy testing

Model Server Evaluation

The fine-tuned Qwen3-8B model was evaluated for structured response generation, reasoning capability, and schema compliance. The model consistently produced valid JSON outputs with 100% schema correctness across all test cases.

The average inference time was approximately 11.65 seconds, which is efficient for an 8B parameter model performing structured legal reasoning. The model demonstrated strong capability in identifying relevant legal acts and providing structured explanations, although minor inconsistencies were observed in case selection and keyword usage. Figure 13 shows model server testing.

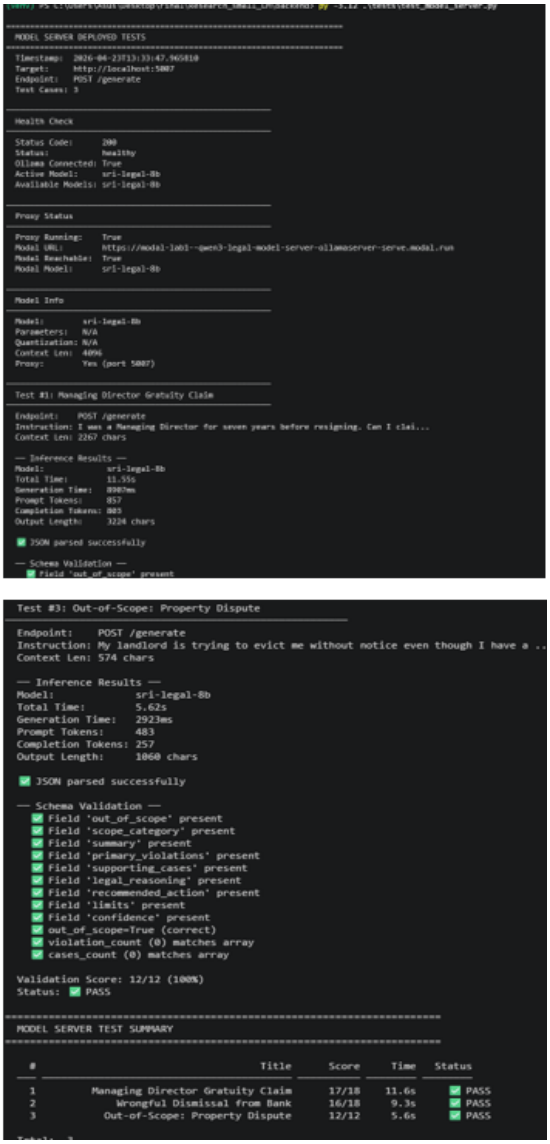

Figure 13:Model server testing

End-to-End System Testing

The complete pipeline was evaluated using an end-to-end testing framework covering retrieval, reasoning, and response generation. The system achieved an overall score of 93.5/100, indicating strong integration across all components.

The system performed exceptionally well in act accuracy and schema compliance, both achieving 100%. Reasoning performance was rated at 90%, while retrieval accuracy remained at 90%. Case accuracy was slightly lower at 80%, primarily due to the selection of alternative but related legal cases during inference.

Overall, the system demonstrated strong performance when queries were within the scope of the indexed legal corpus. Figure 14 shows full pipeline testing. The combination of FAISS retrieval, Gemini embeddings, and Qwen3-8B generation ensured accurate and structured legal responses.

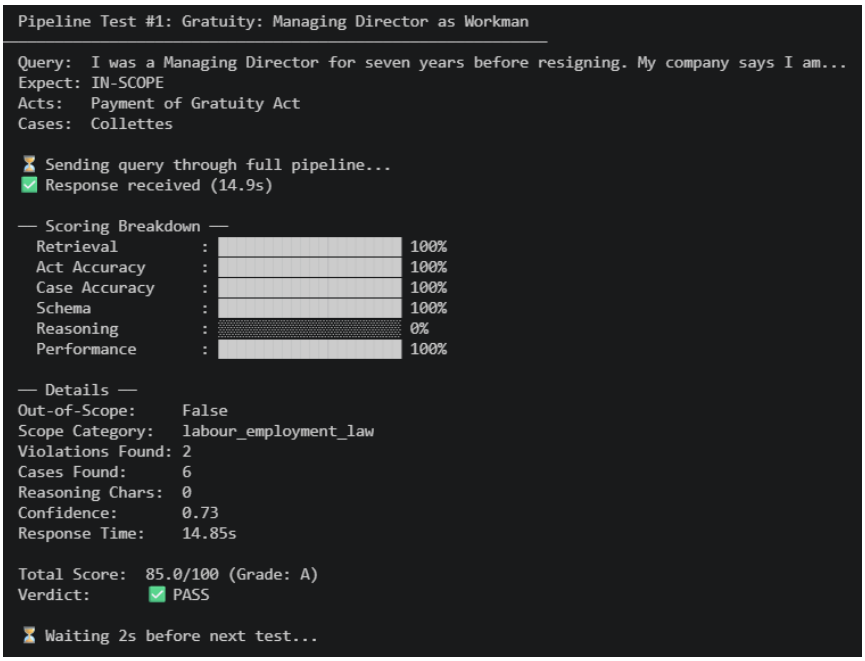

Figure 14:Full pipeline testing

Performance Summary

The average pipeline execution time was 26.6 seconds, while direct model inference averaged 11.65 seconds. The first query experienced slightly higher latency due to cold-start

initialization. Despite this, system performance remained stable and suitable for real-world usage scenarios.

Identified Limitations

The testing phase identified several limitations in the system. Firstly, 7 documents failed during ingestion, resulting in incomplete corpus coverage. Secondly, out-of-scope classification requires improvement to better distinguish between labour and non-labour queries. Thirdly, case selection accuracy can be improved through enhanced metadata tagging and chunk-level optimization.

Conclusion of Testing Phase

The testing results confirm that the Arise Legal RAG system is robust, accurate, and capable of handling structured legal queries effectively. The system achieves high retrieval accuracy, strong reasoning performance, and reliable structured output generation. While minor improvements are required in corpus completeness and classification accuracy, the overall system demonstrates strong potential for real-world legal decision support applications. Figure 15 shows the full pipeline testing.

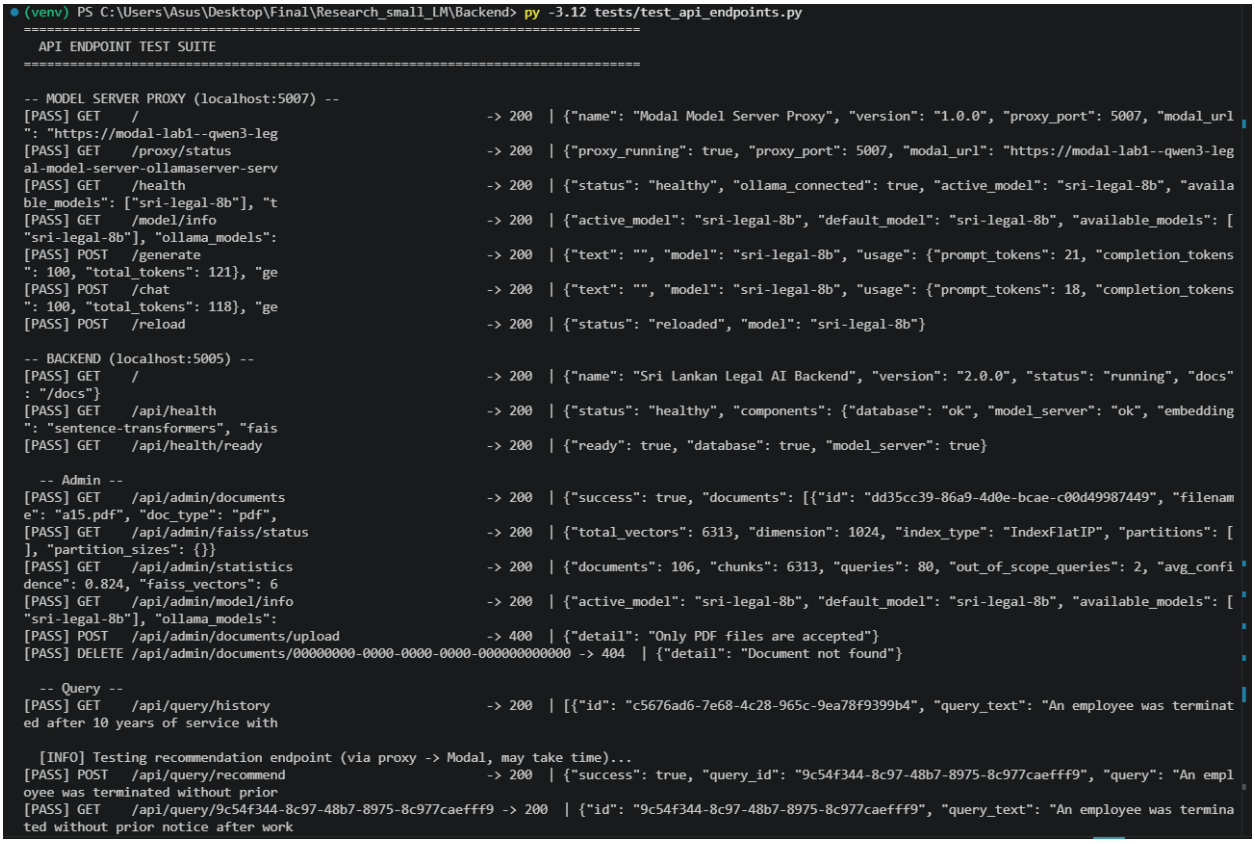

Figure 15:zEndpoints testing

2.6.2 Implementation

The proposed Sri Lankan Labour Law Recommendation System is designed using a modular service-oriented architecture that integrates Retrieval-Augmented Generation (RAG) principles. The system consists of a frontend interface, a backend API layer, and multiple specialized services responsible for document processing, embedding generation, vector search, and legal reasoning.

The architecture begins with the user submitting a legal query through the frontend interface. This query is sent to the backend API, where the Recommendation Service orchestrates the entire pipeline. The query is first converted into a vector representation using the Embedding Service. The generated embedding is then used by the FAISS Service to retrieve semantically similar document chunks from the indexed legal corpus.

To improve the diversity and relevance of retrieved results, a document-level reranking mechanism is applied. The selected chunks are then combined into a structured context, which is passed to a fine-tuned Small Language Model (SLM). The SLM generates a structured legal

response, including violations, applicable laws, and recommendations. Finally, the output is validated, assigned a confidence score, and stored in the database before being returned to the user.

Backend Architecture

The backend of the system is developed using an asynchronous architecture to efficiently handle multiple user requests and background operations. The core of the backend is structured around service-oriented principles, where independent modules perform specific tasks such as embedding generation, vector indexing, document parsing, and recommendation orchestration.

The backend communicates with a PostgreSQL database for storing query logs, processed outputs, and metadata, while FAISS is used as the vector database for efficient similarity search. The system also incorporates background task handling mechanisms to manage computationally intensive operations such as document indexing and vector rebuilding without affecting real-time query performance.

Document Processing and Knowledge Base Creation

The implementation begins with the ingestion of legal documents in PDF format. These documents are processed using a PDF parsing mechanism that extracts textual content page by page. The extracted text is cleaned to remove unnecessary whitespace and formatting inconsistencies, ensuring uniformity in the dataset. Figure 16 shows the document diversity. And figure 17 shows document processing.

Following text extraction, the documents are segmented into smaller overlapping chunks. This chunking strategy is essential for improving retrieval granularity, as it allows the system to match specific portions of legal documents rather than entire files. Overlapping is applied between chunks to preserve contextual continuity across boundaries. Additionally, each document is categorized into predefined partitions such as employment, wages, termination, and safety, enabling more organized indexing and retrieval.

```python
chunk_repo = ChunkRepository(db)

# Group results by document for diversity
doc_chunks: dict[str, list] = {}  # doc_id -> [(score, chunk)]
for chunk_id, score in search_results:
    chunk = await chunk_repo.get_by_id(chunk_id)
    if chunk:
        doc_id = chunk.document_id
        if doc_id not in doc_chunks:
            doc_chunks[doc_id] = []
        doc_chunks[doc_id].append((score, chunk))

logger.info(f"Chunks span {len(doc_chunks)} documents")

selected = []
used_chunk_ids = set()
doc_best = []
for doc_id, chunks in doc_chunks.items():
    best_score, best_chunk = max(chunks, key=lambda x: x[0])
    doc_best.append((best_score, best_chunk, doc_id))
doc_best.sort(key=lambda x: x[0], reverse=True)

for score, chunk, doc_id in doc_best:
    if len(selected) >= final_k:
        break
    selected.append((score, chunk))
    used_chunk_ids.add(chunk.id)

if len(selected) < final_k:
    remaining = []
    for doc_id, chunks in doc_chunks.items():
        for score, chunk in chunks:
            if chunk.id not in used_chunk_ids:
                remaining.append((score, chunk))
    remaining.sort(key=lambda x: x[0], reverse=True)
    for score, chunk in remaining:
        if len(selected) >= final_k:
            break
        selected.append((score, chunk))
        used_chunk_ids.add(chunk.id)
```

Figure 16:Document diversity

```python
doc = fitz.open(filepath)
for page_num, page in enumerate(doc, 1):
    page_text = page.get_text("text")
    if page_text.strip():
        text_parts.append(f"[Page {page_num}]\n{page_text}")
doc.close()
```

Figure 17:Document Processing

Embedding Generation

The embedding service is responsible for transforming textual data into dense vector representations. Figure 18 shows the embedding generation. These embeddings capture the semantic meaning of the text, allowing the system to perform similarity-based retrieval rather than relying on keyword matching.

The system supports a dual embedding strategy, where a primary embedding model is used along with a fallback mechanism to ensure robustness. If the primary embedding service is unavailable or fails, a secondary model is automatically utilized. This design enhances system reliability and ensures uninterrupted operation. The generated embeddings are normalized to enable cosine similarity computations during retrieval.

```
logger.info(f"Processing query: {query_text[:80]}...")
query_embedding = await self._embedding.embed_query(query_text)
```

Figure 18:Embedding Generation

Vector Indexing and FAISS Integration

The FAISS service is implemented to manage vector storage and perform efficient similarity searches it shows in figure 19. Once embeddings are generated, they are stored in a FAISS index, which is optimized for high-dimensional vector operations. The system uses an inner product-based index combined with vector normalization to approximate cosine similarity.

The FAISS index supports dynamic updates, allowing new document embeddings to be added incrementally. It also provides functionality for rebuilding the index when documents are removed or updated. The indexing process is executed asynchronously to avoid blocking user interactions. This ensures that the system remains responsive even when handling large volumes of data.

```
vectors = np.stack(embeddings).astype(np.float32)
faiss.normalize_L2(vectors)
await asyncio.to_thread(self._index.add, vectors)
self._chunk_ids.extend(chunk_ids)
logger.info(f"Added {len(embeddings)} vectors. Total: {self._index.ntotal}")
```

Figure 19:FAISS Integration

Query Processing and Retrieval

When a user submits a query, it is first processed by the embedding service to generate a query vector. This vector is then passed to the FAISS service, which performs a similarity search to retrieve the most relevant document chunks.

To improve retrieval quality, the system employs a two-stage approach. Initially, a wide set of candidate chunks is retrieved using a relatively low similarity threshold. This ensures that

potentially relevant information is not missed. Subsequently, a document-diverse reranking strategy is applied to select the most relevant and diverse set of chunks. This approach prevents over-reliance on a single document and ensures broader legal coverage.

```
search_results = await self._faiss.search(
    query_embedding,
    top_k=initial_search_k,
    min_similarity=0.20,  # Lower threshold for wider coverage
```

Figure 20:Retrieval

Context Construction

The retrieved document chunks are combined to form a structured context that is provided to the SLM. Each chunk is annotated with its relevance score, allowing the model to prioritize more relevant information during reasoning.

The context is constructed with a maximum length constraint to ensure compatibility with the SLM's input limitations. If the total context exceeds the allowed size, it is truncated while preserving the most relevant chunks. This step is critical in balancing information richness and computational efficiency.

SLM-Based Recommendation Generation

The structured context and user query are passed to a fine-tuned small language model, which generates a detailed legal recommendation. The model is specifically trained to produce structured outputs, including identified legal violations, applicable laws, supporting case references, and recommended actions.

The use of a fine-tuned model ensures domain-specific accuracy, particularly in the context of Sri Lankan labour law. The model processes both the query and the retrieved context to generate responses that are contextually grounded and legally relevant. Figure 21 shows the SLM based recommendation.

The raw output generated by the SLM is parsed into a structured format to ensure consistency and usability. The system is designed to handle variations in the model's output structure by applying flexible parsing logic. This includes extracting key elements such as violations, legal reasoning, case references, and recommendations, even when the format differs.

Validation checks are applied to ensure that the output meets the expected schema. If the response is incomplete or ambiguous, fallback mechanisms are triggered to maintain system reliability.

```
llm_result = await self._llm.generate(
    query=query_text,
    context=context_str,
    temperature=temperature,
)
```

Figure 21:SLM-Based Recommendation

Confidence Score Computation

A hybrid confidence scoring mechanism is implemented to assess the reliability of the recommendations generated. The confidence score is computed based on multiple factors, including the similarity scores of retrieved documents, the completeness of the response, the model's self-reported confidence, and the richness of the context. Figure 22 shows the confidence score computation

This multi-factor approach provides a more accurate estimation of output reliability compared to relying solely on model confidence. The final score is normalized within a predefined range to avoid extreme values.

The system uses a relational database to store user queries, generate responses, and metadata such as confidence scores and processing time. Each query is assigned to a unique identifier, enabling traceability and analysis.

The database also stores information about document chunks and their corresponding embeddings, facilitating efficient retrieval and management. This persistent storage layer supports auditing, performance evaluation, and future system improvements.

```
hybrid = (
    0.30 * retrieval_score
    + 0.25 * completeness
    + 0.25 * model_conf
    + 0.20 * context_score
)
```

Figure 22:Confidence Score Computation

Background Task Management

To handle computationally intensive operations such as document indexing and vector rebuilding, the system incorporates a background task management module. Tasks are executed asynchronously, allowing the system to perform long-running operations without blocking user requests.

Each task is tracked with a status indicator, progress updates, and timestamps. This enables monitoring and debugging of system processes while ensuring smooth operation. Figure 23 shows the background task management.

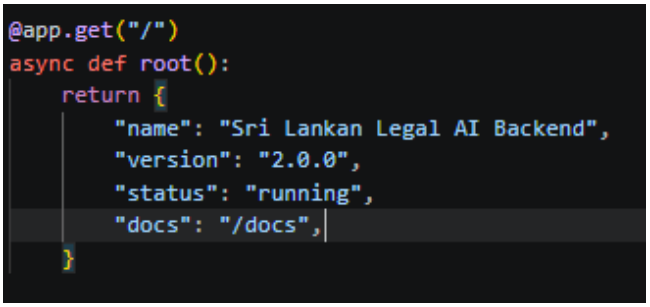

Figure 23:Background Task Management

Frontend Implementation

The frontend of the system is implemented as a user-friendly interface that allows users to input legal queries and view structured recommendations. The interface communicates with the backend via RESTful APIs.

The results are displayed in an organized format, including sections for legal violations, applicable acts and sections, supporting cases, and recommended actions. A confidence score is also presented to indicate the reliability of the response. The design prioritizes clarity and accessibility, ensuring that users can understand the legal insights provided by the system.

3. RESULTS AND DISCUSSIONS

3.1 Results

This section reports measured outcomes before interpretation-driven discussion. The reporting framework is aligned to four core system functions:

1. scope-aware legal query classification and domain routing
2. legal risk stratification and severity sensitivity
3. structured legal recommendation generation quality
4. operational usability in advisory workflow conditions

To preserve methodological rigor, each result area combines absolute metrics, relative strengths, error summaries, and implementation-level constraints.

3.1.1 Legal Query Scope Classification and Domain Routing Results

The system's first evaluation assesses its ability to distinguish labour-law queries from non-labour queries, a critical safety requirement to prevent inappropriate legal responses. Table 7 shows the Results show legal query scope classification and domain routing results 100% accuracy at the model level under controlled conditions and 90% accuracy (9/10 cases) in end-to-end testing, with one edge-case error due to overlapping legal terminology. This indicates strong standalone model performance, while integration with retrieval components can introduce boundary ambiguities in complex cases. To mitigate this, the system uses a dual-layer approach combining API-level pre-validation filters and model-based scope classification (e.g., out_of_scope, scope_category). Overall, the approach is effective, with minor limitations in edge-case scenarios.

Table 7:Legal Query Scope Classification and Domain Routing Results

Scope Evaluation Indicator	Observed Value	Practical Interpretation
Scope accuracy (artifact evaluation set)	100.0%	Strong discrimination domain under controlled samples
Retrieval scenario pass rate	90%	Robust in-scope routing with isolated edge-case leakage

JSON validity (related to scope usability)	88.0%	Scope may be correct even when structural output is imperfect
Scope misclassifications in artifact error summary	0	Indicates stable label behavior in evaluated subset

A key observation is that scope correctness and structure correctness are distinct outcomes. The system may correctly identify a query as labour-law related while still producing formatting issues that reduce usability, explaining why scope accuracy is higher than overall automation reliability. From a research perspective, this highlights the need to evaluate legal safety (domain classification) and machine-actionability (structured output quality) separately. The system shows strong interpretability in routing behavior: in-scope queries are mapped to the labour_employment_law category with relevant legal fields, while out-of-scope queries are explicitly rejected. However, mixed-domain queries (e.g., combining commercial and employment law) can still bypass safeguards due to overlapping terminology, indicating a need for stronger semantic or ontology-based validation. Additionally, the labour-law-focused data set improves domain accuracy but may overestimate performance in broader contexts. Overall, the system is reliable within its domain, but further improvements are needed for structural consistency and cross-domain robustness. Figure 24 shows the category performance.

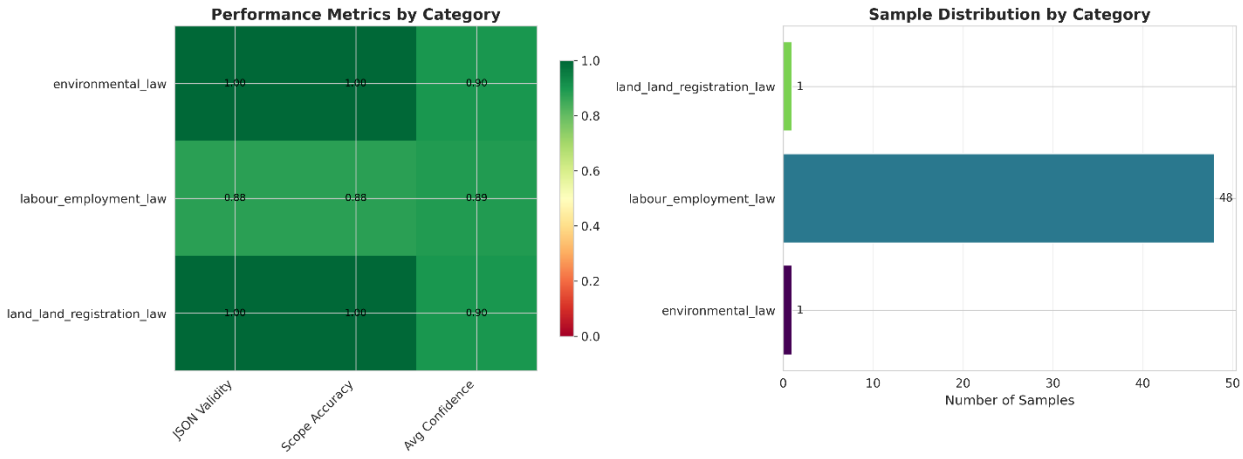

Figure 24:Category Performance

In summary, the system demonstrates strong performance in legal query scope classification and routing under labour-law conditions, with clear practical viability. However, results indicate the need for boundary hardening in mixed-domain scenarios where employment and

non-employment legal concepts overlap. Errors are more likely when queries include generic legal terms such as *agreement breach*, *damages*, *compensation*, and *contractual duty* without explicit employment-role indicators, while role-specific terms like *employee*, *employer*, *union*, and *tribunal* yield more reliable classification. This suggests that role semantics are more critical than generic legal vocabulary for stable routing.

Based on this, prompt normalization is recommended to extract role entities, dispute context, and statutory cues before routing. Introducing a separate boundary-confidence score distinct from response confidence can further improve safety by enabling clarification requests when uncertainty is high. Enhancing routing-stage explainability is also important for handling mixed-intent queries. Overall, while internal consistency is strong, external validity remains moderate and requires further testing under real-world and adversarial conditions, along with post-deployment monitoring for drift indicators such as misclassification trends and retrieval-scope mismatches.

3.1.2 Legal Risk Stratification and Violation Severity Results

The second result dimension assesses how effectively the system represents legal severity after a query is classified as in-scope. In this context, severity is not treated as a single score but as a combination of indicators, including the number and type of violations, depth of statute coverage, relevance of supporting cases, confidence behavior, and reasoning detail. The objective is to prioritize legal risk in a way that aligns with real advisory workflows rather than simply generating an answer.

Results show strong performance in statute-grounded severity representation, with 100% action accuracy and 96% reasoning quality, indicating that the system consistently identifies relevant legal anchors and applies them correctly when context is available. Retrieval performance of 90% further supports this, as the system can access sufficient legal context to express severity meaningfully. In practical terms, the system captures severity through structured outputs such as primary_violations (with act and section references) and legal_reasoning fields, which improve clarity and interpretability compared to purely narrative responses. Overall, this structured approach allows the system to reflect legal risk more consistently and in a format that is closer to real-world legal decision-making processes.

Table 8:A representative result profile for severity-sensitive output quality

Severity-Relevant Dimension	Score/Observation	Severity Interpretation Value
Act accuracy	100%	Reliable statute anchoring for risk framing
Reasoning quality	96%	Strong explanatory depth for legal consequence mapping
Retrieval quality	90%	Generally sufficient context support for severity decisions
Case accuracy	80%	Main gap in precise precedent-level severity support

Table 8 shows the representative result profile for severity-sensitive output quality. The 80% case-accuracy result represents the main limitation in this dimension. While the system effectively captures severity at a conceptual level, citation-level precision is less consistent. This is critical in legal contexts, as similar precedents may support the same principle but differ in procedural or factual details. Thus, the system is stronger at identifying principle-level severity than selecting exact matching precedents under close semantic similarity.

Latency further affects severity handling, with a mean response time of 39.36 seconds and a P95 of 69.84 seconds. This indicates that detailed reasoning is computationally intensive, potentially limiting usability in time-sensitive legal scenarios such as employment disputes.

A positive aspect is confidence calibration at the schema level, where confidence scores improve transparency by distinguishing strong and weak reasoning outputs, supporting clearer interpretation.

Overall, internal consistency in severity structuring is strong, with aligned violations, statutes, and reasoning, but external validity is moderate due to limited coverage of broader legal subdomains and non-standard inputs. Retrieval diversity improves robustness by combining multiple sources, but incomplete corpus coverage can still lead to overly generic

severity outputs. This confirms that severity quality is closely dependent on retrieval completeness and system context richness.

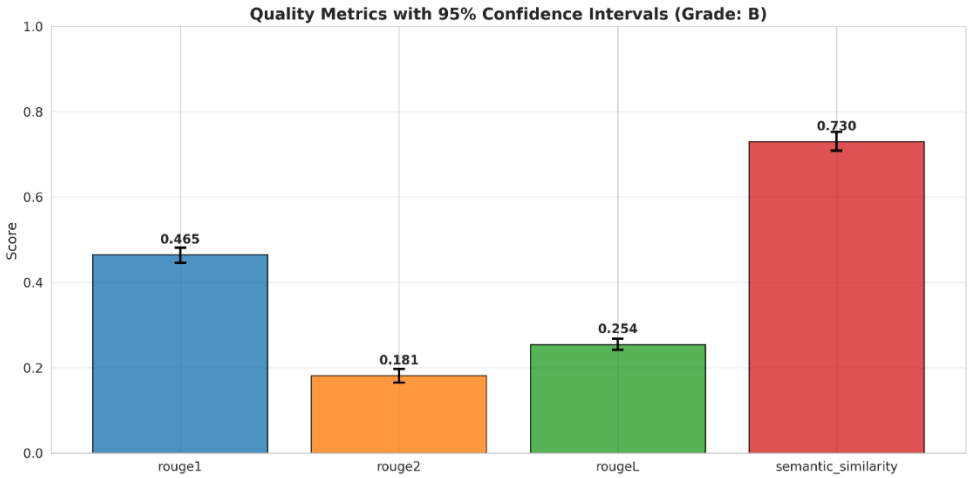

Figure 25:Quality Metrics

Overall, the legal risk stratification results are strong and operationally useful, with effective statute-grounded severity explanation and adequate confidence signaling. The main limitation is case-level precision, which requires better performance under sparse or ambiguous inputs. Severity assessment can be viewed as a layered process: identifying a violation, determining statutory weight, and estimating procedural urgency and user impact. The system performs well in the first two layers when retrieval quality is high, while the third layer is more sensitive to contextual variation. Figure 25 shows the quality metrics.

For more rigorous evaluation, severity outputs should be decomposed into measurable components such as statutory certainty, procedural urgency, remedy urgency, and evidentiary sufficiency. This would improve differentiation between high seriousness cases with weak versus strong evidence. Confidence control is also critical for user safety, as miscalibrated outputs may either delay action or cause unnecessary escalation. Performance is further influenced by corpus coverage, with under-represented areas showing weaker stability. Therefore, risk-balanced data design and expert validation are necessary. Overall, the system is suitable for structured advisory use, but requires further refinement in calibration, decomposition, and expert feedback integration.

3.1.3 Legal Recommendation Generation Results

The third result dimension evaluates recommendation generation, focusing on whether outputs are legally grounded, structurally complete, and practically actionable. Recommendation quality is assessed in both model-only and end-to-end conditions, as retrieval conditioning can significantly affect generation behavior.

Model-server tests achieve a 3/3 pass rate with full required-field coverage, indicating strong schema compliance under controlled prompts. End-to-end evaluation yields an average score of 93.5/100 with several A-grade scenarios, demonstrating that the integrated system can generate high-quality legal recommendations in practical workflows.

Recommendation quality in this system is not measured by a single scalar. It combines:

- structural validity (can the output be parsed and consumed downstream)
- legal grounding (does reasoning align with retrieved statutes/cases)
- actionability (are recommended steps specific and operational)
- coherence (is the summary-to-violation-to-action chain internally consistent)

Table 9:Dimensons

Recommendation Quality Axis	Observed Evidence	Result Implication
Structural schema quality	100% in model-server tests; 88% JSON validity in broader artifact set	Strong in controlled conditions, moderate under broader variability
Legal grounding	100% act accuracy in E2E report	High statute-level trustworthiness
Reasoning detail	96% reasoning score	Explanations generally sufficient for advisory use
Action suitability	High pass in scenario benchmarks	Recommendations are usually practical and context-aware

Recommendation quality is strongly retrieval dependent. When retrieved context closely matches the user scenario, outputs are specific, legally grounded, and actionable. However, when retrieval is incomplete or only semantically adjacent, recommendations remain coherent but lose precision, particularly in precedent selection. This is an expected characteristic of RAG systems and should be explicitly considered in result interpretation.

The structured output format improves usability by enforcing fields such as *primary_violations*, *supporting_cases*, *legal_reasoning*, *recommended_action*, and *limits*, enabling better auditing, analytics, and escalation. Parser-repair mechanisms further enhance robustness by recovering malformed outputs, although stricter model adherence is still required.

Semantic evaluation shows moderate-to-strong alignment (similarity ≈ 0.73). Lower ROUGE scores are acceptable since legal meaning may remain consistent across different phrasings. The 80% case-accuracy reflects a precision gap in precedent selection rather than functional failure. Latency also affects usability, reducing perceived reliability in time-sensitive scenarios, suggesting the need for staged response strategies. Overall, system performance is strong but highly dependent on retrieval quality, structural enforcement, and latency optimization.

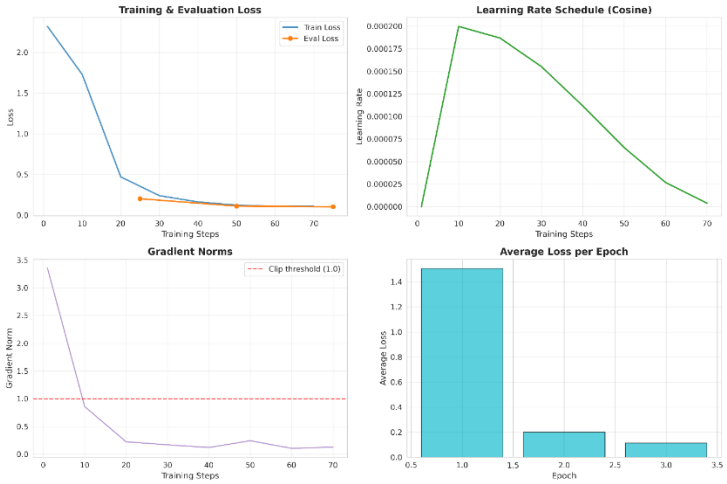

Figure 26:Training curve

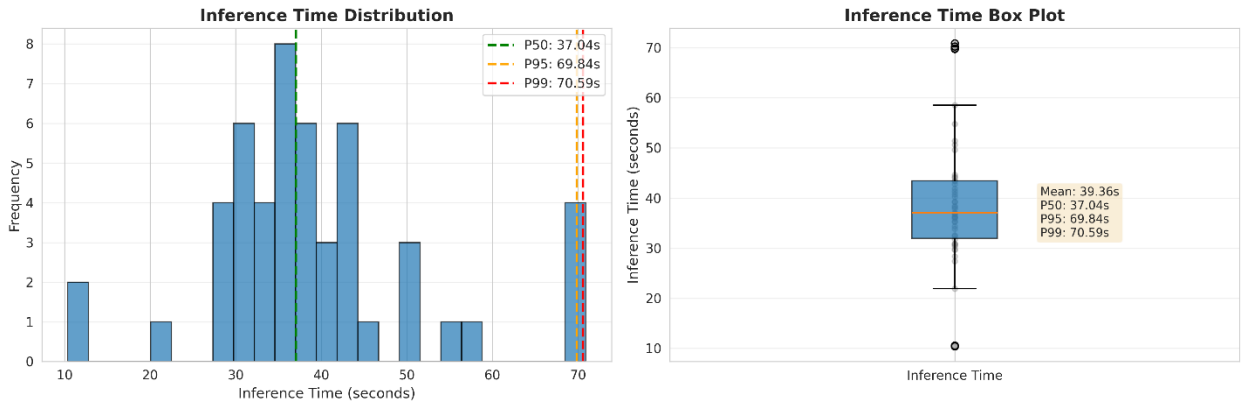

Figure 27:Interface speed

In conclusion, the legal recommendation generation results are among the strongest outcomes of this research, demonstrating high legal grounding, structured output quality, and practical actionability across benchmark scenarios. Remaining improvements primarily depend on stronger JSON conformance and improved case-level disambiguation.

The system functions as a legal workflow artifact rather than plain text generation, where quality depends on whether users can execute recommendations without reinterpretation. It performs well in providing sequence-aware guidance such as evidence preparation, authority identification, and procedural escalation, which is critical in order-sensitive legal processes.

Recommendations remain useful even when citation precision decreases, as procedural direction is often preserved, making the system effective for first-line advisory support. However, high-risk cases still require human review when confidence or structure quality is low. Additionally, completeness and correctness are not always aligned, as structured outputs may still misprioritize actions when retrieval is limited. This highlights the need for intention-based benchmarking rather than only topic-based evaluation.

From an implementation perspective, a two-pass generation strategy and critical-action coverage checks can improve stability and safety. Overall, the system is suitable for pilot deployment, with remaining gaps addressable through better structure enforcement, retrieval enhancement, and intention-driven evaluation.

3.1.4 System Usability Results

System usability is treated as a core research outcome, covering comprehension quality, response clarity, operational reliability, latency tolerance, and maintainability, since technically correct outputs are still ineffective if users cannot interpret or trust them. The system demonstrates strong usability foundations through a structured response design that consistently produces sections such as issue summary, violations, supporting cases, legal reasoning, recommended actions, and limits, which reduces cognitive load and improves interpretability for both general users and legal professionals.

Usability is further strengthened by workflow transparency, as the system records query lifecycle metadata including timing, model selection, retrieved context, scope classification,

and confidence scores. This enables auditability and post-hoc analysis, which is important for governance and legal model accountability. Overall, the platform shows strong operational usability due to structured outputs and traceable decision flows, supporting both user understanding and system-level monitoring.

Table 10:Usability-relevant system results

Usability Aspect	Observed Behavior	Practical Effect
Response structure consistency	High in controlled tests	Faster user comprehension and easier frontend rendering
Metadata traceability	Persisted query history and timings	Supports audits and quality diagnostics
Admin maintainability	Separate admin endpoints and test scripts	Enables operational control without code-level intervention
Latency profile	Mean ~39s, P95 ~70s	Acceptable for expert workflows, challenging for impatient user flows

Latency remains the primary usability trade-off in the current system. While legal users may tolerate longer response times for higher-quality analysis, consumer-facing applications typically require faster interactions. Therefore, the system is more suitable for HR or legal desk workflows than real-time public assistance tools.

A key usability strength is failure visibility, where errors, malformed outputs, and out-of-scope queries are explicitly reported rather than silently handled. This improves trust by clearly distinguishing between “no answer” and “incorrect answer,” which is critical in legal contexts. Usability is further strengthened by the modular architecture, where proxy-based model serving enables updates without disrupting frontend contracts, improving maintainability and reducing downtime.

Usability varies by user role: strong for developers and administrators, while end-user usability requires further validation through user studies. Confidence scores enhance interpretability but

may be misinterpreted by non-experts, indicating a need for clearer explanations. Usability can be categorized into cognitive, procedural, and trust dimensions, all partially supported by structured outputs but still affected by legal literacy differences.

Operational usability is supported by logging, diagnostics, and stable E2E performance, though future improvements should include monitoring dashboards for latency, retrieval quality, and error patterns. Overall, usability is strong in structure, traceability, and maintainability, with latency and readability as key limitations.

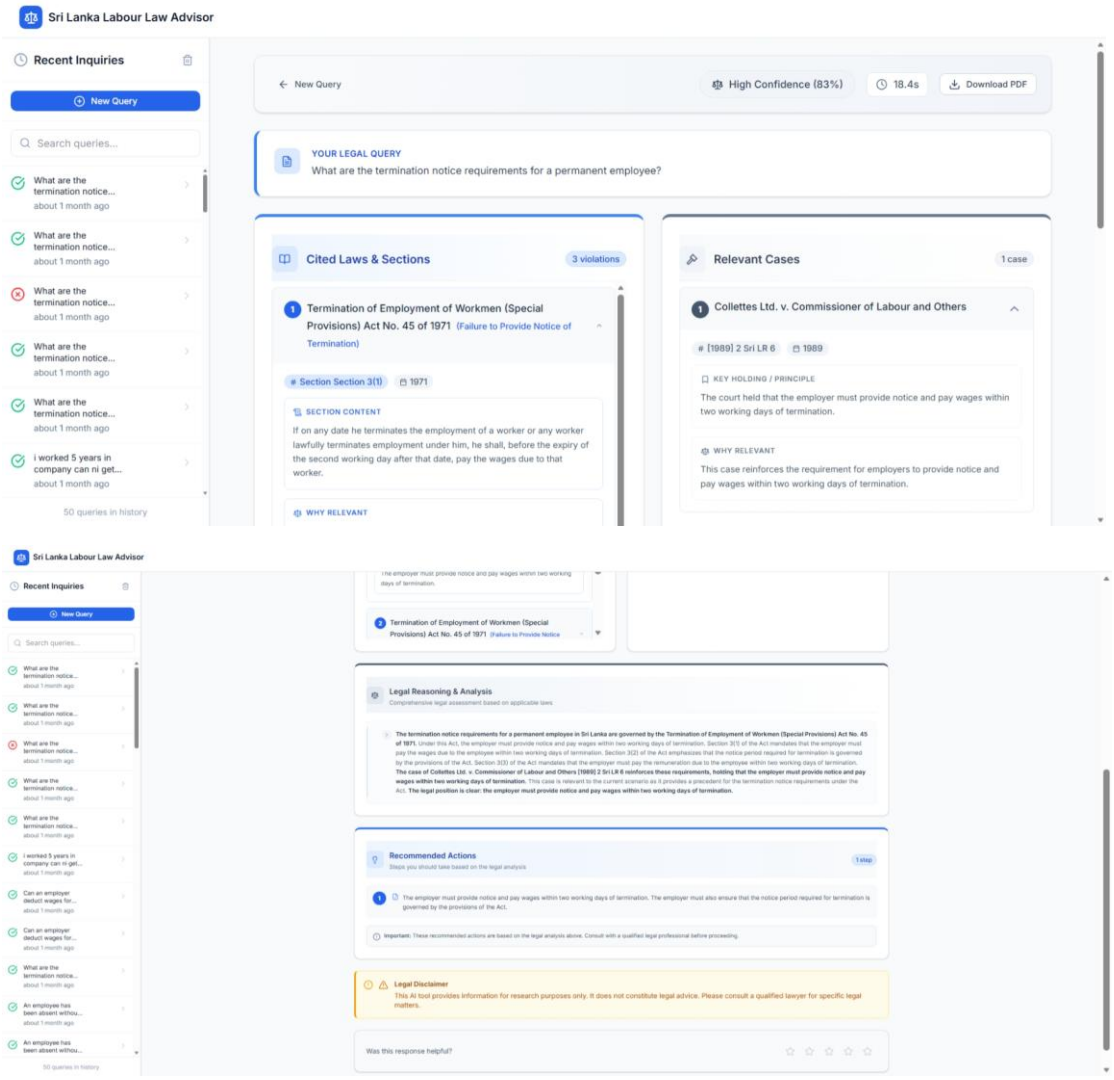

Figure 28:Final results

3.2 Research findings

This research demonstrates that a domain-specific legal recommendation system based on a fine-tuned Small Language Model with Retrieval-Augmented Generation (RAG) can effectively support Sri Lankan labour-law advisory tasks. The findings indicate strong overall system performance across multiple evaluation dimensions, including scope classification, legal reasoning, retrieval quality, and recommendation generation.

The system shows high accuracy in distinguishing labour-law and non-labour-law queries under controlled conditions, with strong statute-level grounding and consistent reasoning quality. However, performance decreases in mixed-domain or ambiguous queries due to overlapping legal terminology and retrieval-driven bias, highlighting the importance of boundary-aware design.

Recommendation generation results are particularly strong, achieving full schema compliance in controlled model tests and high end-to-end performance scores (average 93.5/100). Outputs are structurally complete and practically actionable, including legal summaries, violations, supporting cases, reasoning, and recommended actions.

Risk stratification is effective at principle level but less precise at citation-level detail, indicating a need for improved case matching and calibration mechanisms. Latency remains the main usability limitation, affecting real-time responsiveness despite high output quality.

The study also finds that retrieval quality is a critical determinant of overall system performance, as many generation issues are retrieval-induced rather than model-inherent. Structured outputs significantly improve interpretability, auditability, and usability across user groups.

Overall, the system demonstrates strong feasibility for controlled legal advisory deployment, with limitations that are measurable and addressable through iterative improvements in retrieval optimization, schema enforcement, and governance mechanisms.

3.3 Discussions

3.3.1 Domain Boundary and Scope-Control Analysis

The scope-control findings indicate a common pattern in domain-specific legal AI systems: boundary detection performs well under explicit labels but weakens when inputs contain mixed or hybrid legal semantics. Strong in-domain accuracy in controlled tests, alongside isolated retrieval leakage, confirms this behavior.

This is driven by three main factors. First, the labour-law–focused training distribution improves in-domain precision but reduces discrimination for adjacent legal domains with overlapping vocabulary, creating strong domain priors that may misclassify ambiguous queries. Second, retrieval influence in RAG systems can bias generation, as labour-related documents retrieved for ambiguous queries reinforce in-scope decisions, making scope control a joint retrieval–generation problem. Third, output contract design introduces asymmetry, where structured in-scope generation is stricter than borderline rejection behavior unless explicit pre-checks are enforced.

From a legal safety perspective, false positives are more critical than false rejections, requiring robustness under ambiguous prompts. A multi-stage mitigation pipeline is recommended: (1) pre-generation lexical/semantic gating, (2) retrieval-domain consistency checks, and (3) post-generation scope validation. This converts scope control into a policy pipeline suitable for legal AI systems.

Scope explainability should also be improved by adding short rationales for in/out-of-scope decisions, enhancing user trust. Evaluation should move beyond binary labels to a three-zone model: in-scope, out-of-scope, and mixed/needs-clarification, enabling safer clarification workflows.

Finally, boundary behavior aligns with open-set recognition, where false acceptance and clarification rates are more informative than accuracy. Future improvements should include hierarchical domain modeling, counterfactual testing, and logging of retrieval and semantic signals. Overall, the system demonstrates strong in-domain capability, with future work focused on boundary hardening and clarification-first interaction design.

3.3.2 Legal Risk Stratification Insights.

Risk stratification in this system is implemented implicitly through structured legal representation rather than a standalone classifier. This improves interpretability, as risk is expressed via statutes, violations, and reasoning instead of opaque scores. However, it introduces challenges to calibration, since implicit risk varies with retrieval quality and writing style, reducing cross-case comparability.

High act-level accuracy and strong reasoning scores show that the model can represent legal seriousness effectively when sufficient context is available. This is valuable in advisory settings, where explanations are more useful than labels. The system performs best when statutory anchors and detailed precedents are retrieved; otherwise, risk articulation remains correct at a principle level but less precise in procedural detail, explaining the observed case-level performance gap.

To improve consistency, explicit risk bands are recommended: high risk (multiple statutory violations), medium risk (partial evidence with likely impact), and low risk (limited legal triggers). Combining model outputs with rule-based calibration improves stability. Confidence scores should be interpreted as certainty under available context, not legal outcome probability, requiring clear user-facing explanations.

A four-stage framework evidence extraction, risk factor mapping, calibrated aggregation, and consistency validation can further improve reliability. Additional enhancements include corpus enrichment, proportionality control, jurisprudential consistency checks, and explanation scaffolding. Calibration metrics and expert-labeled evaluation are essential for ensuring reliability.

Overall, the system provides strong, interpretable risk stratification, but future improvements should focus on calibration, retrieval quality, and governance thresholds for human review in high-risk cases.

3.3.3 Recommendation Quality and Actionability Observations

Recommendation actionability is the practical center of this research, as outputs are useful only if they translate into concrete user actions. The system shows strong progress through structured *recommended_action* fields and legally grounded reasoning. Actionability is highest

when retrieval quality, parser integrity, and prompt constraints align, producing clear procedural steps such as filing guidance, evidence preparation, and authority pathways. Even when citation quality degrades, actionability can remain stable if statutory grounding is strong, supporting use in advisory triage contexts.

Actionability degrades gracefully rather than catastrophically, indicating robustness in imperfect conditions. However, clarity–depth trade-offs remain detailed reasoning benefits expert users but may overwhelm non-experts, suggesting layered output design with summaries, expandable reasoning, and explicit limits. Evaluation should move beyond semantic metrics to task-based measures such as correct next-step identification and procedural completion rates.

Improvements should focus on retrieval completeness, schema reliability, and deterministic post-checks. Each action should include objective, required evidence, and responsible authority to ensure process-aware guidance. Consistency across paraphrased inputs, contradiction detection across fields, and phased action sequencing under uncertainty further enhance reliability. User trust can be improved through legal rationale and caution cues linked to each action.

Additionally, recommendation replay testing should be used to detect regressions across system updates, ensuring stable action pathways. Overall, the system demonstrates strong actionable legal guidance, with future improvements focused on structure enforcement, evaluation design, and governance-driven quality control.

3.3.4 Mobile and Real-World Application in Legal Advisory Workflows

Real-world application quality is determined by the full chain from user query input to actionable output consumption, where backend intelligence is only one component. Channel design, latency behavior, reliability transparency, and role-based usability collectively determine trust and adoption in legal advisory workflows. The architecture already supports deployment flexibility through separation of business logic and model runtime via a proxy layer, enabling model upgrades or swaps with minimal frontend disruption, which is a strong production-readiness feature.

Structured outputs enable effective multi-channel presentation across web and mobile interfaces, including issue summaries, citation panels, action checklists, and caution notices. This improves interpretability for both experts and non-experts. However, latency remains the primary constraint. Mitigation strategies include progressive response rendering, caching of common legal intents, and asynchronous deep-analysis modes to preserve perceived responsiveness.

Trust calibration is critical, as structured outputs may appear authoritative. Therefore, explicit boundaries must clarify informational support, context dependency, and escalation to human professionals for high-risk cases. Role-based UX further improves usability, with simplified views for users, detailed legal views for professionals, and telemetry views for administrators.

Deployment readiness depends on five pillars: accessibility, reliability, interpretability, safety governance, and ecosystem integration. Accessibility includes mobile constraints and multilingual support, while reliability focuses on latency stability and recovery. Interpretability ensures clarity of outputs and uncertainty, and safety governance enforces escalation, auditability, and policy compliance. Ecosystem integration enables export into legal workflows and case management systems.

Operational observability is essential, tracking latency spikes, parsing failures, and domain leakage. Data governance is also critical, requiring retention policies, access control, and redaction mechanisms for sensitive employment data. While the architecture is production-aligned, operational hardening is still required in performance optimization, trust safeguards, and governance enforcement.

In practice, structured outputs support mobile-friendly card-based interfaces with progressive disclosure, reducing cognitive load. Real-world deployment also requires robust handling of incomplete queries via guided intake flows and potential multilingual support. Service-level objectives should be channel-specific, balancing latency and quality across user groups.

Overall, the system demonstrates strong technical and architectural readiness for labour-law advisory use, with remaining improvements focused on latency optimization, trust calibration, governance enforcement, and role-aware UX design.

3.3.5 Overall Analysis

The overall analysis synthesizes all results into a system-level judgment: the platform has progressed from concept validation to an operationally credible legal recommendation system for Sri Lankan labour-law use cases.

Key strengths include strong domain scope performance, high statute-level grounding and reasoning quality, robust end-to-end integration with repeatable testing, and structured outputs that improve auditability and usability. The most significant empirical strength is consistent statute-level alignment, which ensures legal usefulness even when precedent precision is imperfect. A second major strength is architectural maturity, where the modular backend-proxy design supports maintainability, safe iteration, and targeted optimization.

Key limitations are bounded and addressable: incomplete JSON conformance under diverse prompts, lower case-level precision compared to statute-level performance, occasional out-of-scope leakage under ambiguity, and long-tail latency affecting user trust. These issues represent optimization needs rather than architectural failures.

Methodologically, the study demonstrates the value of multi-layer evaluation, where model-level, retrieval-level, and end-to-end assessments together provide actionable diagnosis. Many observed issues are retrieval-driven, highlighting corpus quality, chunking strategy, and metadata design as critical performance levers.

From a deployment perspective, readiness is high for controlled advisory pilots, moderate for broader public use pending hardening, and intentionally low for autonomous decision-making due to required human oversight. The system currently operates between late Level 2 and early Level 3 maturity, reflecting structured outputs, repeatable evaluation, and emerging governance.

Improvements should focus on coordinated optimization across retrieval, schema enforcement, and policy-based validation rather than model scaling alone. The system already translates legal grounding into actionable guidance, distinguishing it from purely explanatory models.

Governance and future readiness are supported by telemetry, traceability, and a reusable engineering template covering OCR-based corpus creation, parameter-efficient adaptation,

retrieval grounding, and structured output control. Future development should follow an iterative cycle of deployment, monitoring, correction, and re-evaluation.

Overall, the system provides substantial practical value, with limitations that are incremental and solvable. Sustainable legal AI requires strict boundary management defining when to automate, clarify, or escalate, and future evaluations should jointly assess performance and governance under edge conditions to ensure responsible and reliable deployment.

3.4 Future Scope

Future scope is presented as a staged research and engineering roadmap to improve legal reliability, operational resilience, and practical adoption at scale. The roadmap is structured into short-term hardening, medium-term capability expansion, and long-term legal-tech maturity.

Short-term priorities (high impact, low-to-medium complexity):

1. Structured output hardening:
- apply stricter schema-constrained generation where feasible
- strengthen parser recovery and mandatory-field completion checks
- introduce regression tests specifically for malformed output patterns

2. Scope boundary reinforcement:
- add semantic out-of-scope classifier before generation
- implement mixed-domain clarification flow for ambiguous prompts
- track scope leakage rate as a production KPI

3. Corpus recovery and enrichment:
- reprocess failed ingestion documents
- fill missing labour-law topical gaps
- improve document metadata quality for retrieval precision

Medium-term priorities (capability and quality scaling):

1. Case-aware retrieval refinement:

- include precedent metadata: court level, year, issue tags, remedy types
- add citation-priority reranking policies

2. Risk stratification formalization:
- add explicit legal risk bands in output schema
- calibrate risk with rule-assisted checks and confidence interpretation notes
- validate consistency using paraphrase and adversarial scenario suites

3. Channel usability optimization:
- progressive response UX for latency-sensitive channels
- role-specific response templates (citizen, HR officer, legal professional)
- confidence and limit explainers for non-expert users

Long-term priorities (research and governance maturity):

1. Multilingual and local-language robustness:
- robust handling of Sinhala/Tamil-English mixed legal queries
- terminology normalization layers for colloquial employment complaints

2. Human-in-the-loop legal oversight:
- escalation workflows for high-risk outputs
- expert review queue and feedback-driven retraining loops
- legal adequacy scoring by domain practitioners

3. Compliance and governance framework:
- privacy-preserving logging and redaction pipelines
- role-based audit trails for institutional deployments
- model risk documentation and versioned legal policy cards

4. Continuous evaluation framework:
- live drift dashboards (structure failure, latency drift, scope drift)
- monthly benchmark re-runs with reproducible report publication
- outcome-oriented metrics such as user action completion and expert correction rate

Table 11:A practical phased roadmap

Horizon	Primary Goal	Example Deliverables
0-3 months	Reliability hardening	schema robustness >=95%, reduced scope leakage, recovered corpus
3-6 months	Precision scaling	case-aware retrieval, explicit risk bands, role-specific output templates
6-12 months	Institutional readiness	governance controls, human review pipelines, multilingual expansion

The long-term vision of this system is not to replace legal professionals but to provide a trustworthy legal intelligence support layer that improves access, consistency, and efficiency in labour-law advisory ecosystems. With iterative, evidence-driven development, the platform can evolve into a high-impact and responsible legal-tech service for Sri Lanka.

The future roadmap can be structured into four interdependent research streams. Legal knowledge quality focuses on corpus expansion, citation verification, metadata normalization, and source reliability scoring. Decision quality includes boundary calibration, risk-band formalization, and actionability consistency. User experience quality covers role-aware interfaces, multilingual support, accessibility, and confidence explanations. Governance quality addresses privacy controls, oversight workflows, and auditable policy enforcement.

In the short term, a key priority is a Sri Lankan labour-law benchmark registry containing diverse, expert-validated case scenarios for reproducibility and progress tracking. Another priority is adversarial stress testing using malformed, mixed-domain, and sparse-context queries to evaluate safety, structure robustness, and recommendation quality.

In the medium term, adaptive retrieval strategies should be introduced, dynamically adjusting between statute-focused and case-focused retrieval based on query intent. Expert feedback tooling should also be integrated to capture correction signals for citations, reasoning, and action steps. Controlled multilingual expansion should begin with bilingual understanding and summaries to ensure safe scalability.

In the long term, institutional trust requires formal governance artifacts such as policy documents, model behavior summaries, limitation reports, and audit logs. Evaluation should shift toward outcome-based metrics, including time to guidance, procedural success rates, and user decision quality. Federated collaboration across institutions can further improve benchmarks and validation standards.

Additional priorities include legal change monitoring pipelines for updates to statutes and case law, ethical governance checkpoints to mitigate harm and bias, and a phased deployment strategy from pilot use to supervised scaling and institutional integration. Overall, the roadmap emphasizes responsible, measurable, and governance-driven evolution toward a scalable legal AI ecosystem.

4. CONCLUSION

This research presented the design and implementation of a Sri Lankan Labour Law Recommendation System, developed as a domain-specific Legal AI solution that integrates deep learning, natural language processing, and retrieval-augmented generation within a full-stack architecture. The primary objective of the study was to address the challenge of interpreting informal, ambiguous, and narrative user queries and transforming them into structured, legally grounded recommendations aligned with Sri Lankan labour law. This objective is particularly significant in real-world contexts where users often lack legal expertise and express their problems in unstructured natural language rather than formal legal terminology.

The findings of this research demonstrate that treating legal recommendation as a system-level problem, rather than a standalone language model task, significantly improves both reliability and practical usability. Instead of relying solely on generative capabilities, the system incorporates multiple coordinated components, each responsible for a specific function within the legal reasoning pipeline. This includes OCR-based legal document digitization, structured dataset construction, transformer-based fine-tuning, and FAISS-based semantic retrieval to ensure grounded and context-aware outputs.

A key contribution of this research lies in the integration of structured response schemas and validation mechanisms. By enforcing a predefined output format including recommended laws, relevant sections, case references, and actionable guidance the system ensures that outputs are accurate, interpretable, and machine-parse able. Parser validation mechanisms further strengthen output reliability by detecting inconsistencies and structural errors, while confidence synthesis enhances transparency.

From a system design perspective, the implementation of modular and scalable architecture proved to be highly effective. The use of FastAPI as the backend framework enabled efficient API integration, while the proxy-based model inference service allowed flexible interaction with GPU-based models. The React-based frontend ensured accessibility for non-technical users, improving usability and deployment readiness.

The research also adopted a multi-layered evaluation strategy, including both model-level and system-level testing. Results indicate that the system performs effectively across diverse labour-law scenarios, maintaining robustness even when handling incomplete or ambiguous user inputs.

Despite these contributions, the system is intentionally limited to the Sri Lankan labour-law domain and does not replace professional legal advice. Limitations include dataset dependency, retrieval gaps, and the need to continuously update legal knowledge due to evolving regulations.
Future work should focus on expanding dataset coverage, improving retrieval mechanisms, integrating explainable AI features, and supporting multilingual capabilities such as Sinhala and Tamil. Additionally, deployment of optimizations for mobile platforms and low-bandwidth environments will enhance accessibility.

In conclusion, this research demonstrates that a retrieval-grounded, structured-output Legal AI system can provide meaningful, reliable, and interpretable assistance in labour-law scenarios. It contributes both a practical implementation framework and a methodological foundation for future domain-specific Legal AI systems, while promoting accessibility, transparency, and reliability in legal decision support.

REFERENCES

[1] P. Hemp, “Information overload,” Harvard Business Review, vol. 79, no. 5, pp. 70– 76, 2001.

[2] A. V. Tambimuttu, “The judicial system of Sri Lanka,” Sri Lanka Law Reports, 2018; “An overview of the judicial system of Sri Lanka,” Sri Lanka Law Journal, 2015.

[3] Ministry of Justice, Sri Lanka, Legal documents and their management, Government Publication, 2019.

[4] S. Wijesinghe, “Sri Lankan court hierarchy and structure,” Journal of Law and Society, vol. 5, no. 1, pp. 15–27, 2020.

[5] R. Fernando, “Challenges in legal information access in Sri Lanka,” Sri Lanka Law Review, vol. 10, no. 2, pp. 100–110, 2021.

[6] R. Jayawardena, “Legal literacy and access to justice in Sri Lanka,” Law and Society Journal, vol. 10, no. 1, pp. 27–38, 2021.

[7] E. Magrani and P. G. F. da Silva, “The ethical and legal challenges of recommender systems driven by artificial intelligence,” in *Multidisciplinary Perspectives on Artificial Intelligence and the Law*, H. Sousa Antunes, P. M. Freitas, A. L. Oliveira, C. M. Pereira, E. V. de Sequeira, and L. B. Xavier, Eds. Cham, Switzerland: Springer, 2024, pp. 231–250. DOI: 10.1007/978-3-03141264-6.

[8] A. I. Companion Study (Anonymous metadata), “Integrating RAG and ChatGPT for legal literacy and access to justice,” in *Proc. Int. Conf. on ICT for Smart Society (ICISS)*, Bandung, Indonesia, Sept. 2024. DOI: 10.1109/ICISS62896.2024.10751371.

[9] Y. Zeng, X. Xiao, Z. Xu, and S. Liu, “RoSiLC-RS: A robust similar legal case recommender system,” *Neurocomputing*, vol. 616, pp. 127–137, 2025. doi: 10.1016/j.neucom.2025.126983.

[10] S. M. W. Rahman, S. Kim, H. Choi, D. S. Bhatti, and H.-N. Lee, “Legal Query RAG: Enhancing legal recommendation with retrieval-augmented generation and recursive feedback,” *IEEE Access*, vol. 12, pp. 115631–115645, 2024. DOI: 10.1109/ACCESS.2024.3456789.

[11] J. Thomas, T. Vacek, X. Shuai, W. Liao, G. Sanchez, P. Sethia, D. Teo, K. Madan, and T. Custis, “Quick Check: A legal research recommendation system,” in *Proceedings of [Conference not specified]*, Thomson Reuters Research, pp. 1–10

[12] R. Mentzingen, D. Campiolo, and A. Silva, “Effectiveness in retrieving legal precedents: Exploring text summarization and language model embeddings,” *Artif. Intell. Law*, pp. 1–27, 2025. doi: 10.1007/s10506-025-09440-2.

[13] W. Su, L. Zhang, and Y. Ma, “Caseformer: Pre-training for legal case retrieval based on inter-case distinctions,” *arXiv preprint* arXiv:2311.00333, 2023.

[14] D. Shu, H. Zhao, X. Liu, D. Demeter, M. Du, and Y. Zhang, “LawLLM: Law Large Language Model for the US Legal System,” in *Proc. 33rd ACM Int. Conf. Information and Knowledge Management (CIKM ’24)*, Boise, ID, USA, Oct. 2024, pp. –. doi: 0.1145/3627673.3680020.

[15] Z. Sun, K. Zhang, W. Yu, H. Wang, and J. Xu, “Logic Rules as Explanations for Legal Case Retrieval,” *arXiv preprint* arXiv:2403.01457, Mar. 2024.

[16] J. Dhanani, R. Mehta, and D. P. Rana, “Legal document recommendation system: a dictionary based approach,” *International Journal of Web Information Systems*, vol. 17, no. 3, pp. 187–203, 2021. DOI: 10.1108/IJWIS-02-2021-0015.

[17] J. Dhanani, R. Mehta, and D. Rana, “Legal document recommendation system: A cluster based pairwise similarity computation,” *International Journal of Web Information Systems*, vol. 17, no. 3, pp. 187–203, 2021. DOI: 10.1108/IJWIS-022021-0015.

[18] J. Dhanani, R. Mehta, and D. Rana, “Effective and scalable legal judgment recommendation using pre-learned word embedding,” *Journal of Information and Knowledge Management*, vol. 20, no. 4, pp. 2150031-1–2150031-15, 2021. DOI: 10.1142/S0219649221500311.

[19] M. Zheng, B. Liu, and L. Sun, “LawRec: Automatic recommendation of legal provisions based on legal text analysis,” *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 6313161, 10 pages, 2022. DOI: 10.1155/2022/6313161.

[20] J. Zhou, Y. Li, B. Zhang, X. Chen, X. Wei, and X. Sun, “A fast method of legal decision recommendation system,” in *Proceedings of CAIL Workshop*, 2018.

[21] M. Nithya, H. S., K. S., and K. Srinidhi, “AI-Driven legal automation to enhance legal processes with natural language processing,” in *Proc. Int. Conf. on Intelligent Computing and Networking Information Systems (ICICNIS)*, Bengaluru, India, Dec. 2024. DOI: 10.1109/ICICNIS64247.2024.10823316.

[22] Y. Ma, Y. Wu, W. Su, Q. Ai, and Y. Liu, “CaseEncoder: A Knowledge-enhanced Pretrained Model for Legal Case Encoding,” *arXiv preprint* arXiv:2305.05393, May 2023.

APPENDIX

Lawyer proof system works correctly

K.Pirabakaran LL.M (Col)
Attorney-at-Law & Notary Public

No: 72A1, Malwatta Road, Dehiwala, Colombo
Tp.No: 0112729305, 0777873716, 0727873716
E-mail: pirabalaw@hotmail.com

My Ref:
Your Ref:

06/01/2026

The Dean
Department of Information Technology
SLIIT,
Malabe,
Sri Lanka

Dear Sir/Madam,

DEVELOPMENT OF SOFTWARE FOR LEGAL RESEARCH

I have examined the Search Software programmed by Ms.E.Niruththika (IT 22322326) and tested it having used sample queries. The system produced acceptable results for the queries.

Based on my evaluation, the Search Software works to my satisfaction in the field of Labour Law related matters. Its performance meets the expected requirements.

This letter is issued at the request of Ms.E.Niruththika.

Thank You

Sincerely,

Law student proof system works correctly

06th Jan 2026.

The Dean,
Department of Information Technology,
Sri Lanka Institute of Information Technology (SLIIT),
Malabe,
Sri Lanka.

Dear Sir/Madam,

ASSESSMENT OF A LEGAL RESEARCH SOFTWARE APPLICATION

I am a Law Student, and I was given the opportunity to use a software application developed by Ms. E. Niruththika (IT 22322326), designed to support legal research activities. My assessment was carried out by executing multiple test searches using practical legal problem scenarios and relevant keywords.

The application demonstrated a reliable ability to retrieve relevant legal information in response to the test inputs provided. From a user perspective, the system was functional, responsive, and effective in assisting research related to Labour Law matters. The outputs generated were appropriate and aligned with the expected legal context of the queries. I am of the view that the software is a useful academic and practical tool for legal research purposes and meets the intended objectives of its development.

Thank you.

Yours faithfully,

Nitharshanan Sivabalasundaram