SMALL LANGUAGE MODELS FOR SRI LANKAN LEGAL APPLICATIONS

(Fine-tuning an SLM to provide end-to-end, step-by-step guidance for resolving user problems in Sri Lankan Property Law and Family Law system for users)

Project Final Report

E.S. Mathusigan – IT22177032

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

Department of Information Technology

Sri Lanka Institute of Information Technology

April 2026

SMALL LANGUAGE MODELS FOR SRI LANKAN LEGAL APPLICATIONS

(Fine-tuning an SLM to provide end-to-end, step-by-step guidance for resolving user problems in Sri Lankan Property Law and Family Law system for users)

Project Final Report

E.S. Mathusigan – IT22177032

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

Department of Information Technology

Sri Lanka Institute of Information Technology

April 2026

DECLARATION

I declare that this is our own work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma 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
Mathusigan E.S	IT22177032	

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

Law is of fundamental significance to ensure justice and equity in society. Nevertheless, in Sri Lanka, access to legal knowledge is only reserved for lawyers and specialists. Therefore, people actually use more resources and time to talk to lawyers pertaining to minor legal issues. That is why there is the need for an inexpensive and easy method of gaining access to legal advice.

Primarily, the task of the research is to refine the Small Language Model (SLM) capable of giving step-wise advice for finding out the solution to issues concerned with Sri Lankan Property Law and Family Law. The web-based application easily understandable by non-legal users is to be created for the sake of giving such a system.

Here, textbooks for Property Law and Family Law will be collected for data, and then formatted for model training. Using the Unsloth system, various mini-models will be trained and experimented upon to find the model that works the best. An Agentic Retrieval-Augmented Generation approach, RAG approach, Agentic approach from using these 3 approaches used and evaluated and it's integrated to minimize response time even further and increase precision even more.

Output will be a program that provides correct and comprehensible replies to questions asked by users in these two areas of law. The program can be utilized to aid laymen and law students to learn legal matters. In the future, it can be extended to other areas of

Sri Lankan law, making legal services more accessible, affordable, and expedient for all.

Keywords - Labour and Employment Law, Legal Recommendation System, Small Language Model (SLM) ,Agentic Retrieval-Augmented Generation ,Agents, 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 Sumathipala, 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
NER	Named Entity Recognition
API	Application Programming Interface
AI	Artificial Intelligence
OCR	Optical character recognition
QLoRA	Quantize Low-Rank Adaptation

1. INTRODUCTION

The legal field plays a vital role in maintaining justice, equality, and order in society. It provides rules and guidelines that help individuals understand their rights and responsibilities. However, access to legal knowledge is often limited to lawyers and legal professionals. For ordinary citizens, understanding legal procedures and documents can be difficult and confusing. As a result, even for small legal issues, people are forced to seek professional legal advice, which can be expensive and time-consuming. This situation creates a significant gap between legal knowledge and the general public, especially in developing countries like Sri Lanka.

In many cases, individuals avoid taking legal action or fail to protect their rights due to the lack of accessible and reliable legal information. This highlights the urgent need for a system that can deliver legal guidance in a more accessible, affordable, and user-friendly manner. Such a system should be capable of explaining legal concepts in simple language and guiding users step by step in resolving their legal issues. Bridging this gap can significantly improve access to justice and empower individuals to make informed decisions.

With the rapid growth of Artificial Intelligence (AI), new opportunities have emerged to address this challenge. In particular, advancements in Natural Language Processing (NLP) have enabled the development of intelligent systems that can understand and generate human language effectively. Among these technologies, Small Language Models (SLMs) have gained significant attention due to their efficiency and lower computational requirements compared to large-scale models. SLMs are particularly suitable for real-world applications in resource-constrained environments, as they offer fast and cost-effective solutions while maintaining reliable performance.

This research is part of a broader initiative aimed at transforming legal accessibility in Sri Lanka through AI-based solutions. The overall scope includes four key components: (1) fine-tuning a Small Language Model to provide end-to-end, step-by-step guidance for resolving user issues in Sri Lankan Property Law and Family Law; (2) developing an SLM-based system for risk review and analysis of Sri Lankan deed documents; (3) designing a recommendation system focused on labour and employment law to guide users toward appropriate legal actions; and (4) estimating

likely judicial rulings by analyzing past case histories, enabling users to understand potential outcomes and legal precedents.

As part of this larger framework, the primary focus of this research project is to design and fine-tune a Small Language Model that can deliver clear, structured, and step-by-step legal advice specifically in Sri Lankan Property Law and Family Law. These areas are selected due to their relevance to everyday legal issues faced by the general public and the complexity involved in understanding their procedures.

To achieve this objective, a comprehensive dataset is developed by collecting legal information from documents, books, and expert sources. The data is then processed and structured to ensure effective model training. Furthermore, multiple system architectures are explored to enhance performance and reliability, including Agentic architecture, Retrieval-Augmented Generation (RAG), and a hybrid Agentic RAG approach. These architectures enable the system to retrieve relevant legal information, generate accurate responses, and validate outputs to ensure consistency and quality.

The proposed system is designed to deliver legal guidance in a clear, step-by-step format, making it easier for users to understand and apply. It aims to reduce response time, improve accuracy, and provide a user-friendly interface for interaction. By offering structured explanations and practical guidance, the system reduces dependency on costly legal consultations while supporting users in handling legal matters independently.

In conclusion, this research seeks to make legal knowledge more accessible, understandable, and practical for everyday use. By leveraging the capabilities of Small Language Models and advanced AI techniques, it aims to bridge the gap between legal expertise and the general public. Ultimately, this work contributes to improving access to justice in Sri Lanka by providing an affordable, efficient, and scalable legal assistance solution.

1.1 Background and Literature Survey

The legal system plays a critical role in maintaining justice, fairness, and social order. However, access to legal knowledge remains a significant challenge in many countries, particularly for ordinary citizens. In Sri Lanka, legal knowledge related to areas such as Property Law and Family Law is often confined to legal professionals or documented in complex legal texts. As a result, individuals must invest considerable time and financial resources to obtain professional legal advice, even for relatively simple issues. This creates a barrier to justice and highlights the need for more accessible and efficient legal support systems.

Recent advancements in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP), have introduced new possibilities for improving access to legal information. Research has shown that language models can be effectively applied to legal tasks such as text classification, document analysis, and question answering at a relatively low cost. Several studies have explored the use of legal datasets to train models capable of understanding legal language and context [1]. In addition, specialized large language models such as LawLLM have been developed to address legal problems within specific jurisdictions, such as the United States [2].

To evaluate the performance of such models, benchmark datasets and evaluation frameworks have been introduced. For example, LexGLUE is widely used for legal text classification and case analysis tasks [3]. Similarly, legal question-answering datasets such as JEC-QA in China [4] and systems like VidhikDastaavej in India [5] demonstrate the growing interest in applying AI to real-world legal scenarios. These systems aim to automate legal reasoning and document generation, thereby reducing the need for manual intervention.

Another significant advancement in legal AI is the use of Retrieval-Augmented Generation (RAG) techniques. RAG combines information retrieval with text generation, allowing models to access relevant legal documents and generate more accurate and

context-aware responses [6][7]. This approach has been widely adopted in legal applications to improve precision and reliability. Furthermore, modern fine-tuning frameworks such as Unsloth [8] and parameter-efficient techniques like Low-Rank Adaptation (LoRA) [9] have made it possible to adapt pre-trained models to specific domains with reduced computational cost. These methods are particularly useful for developing domain-specific systems using Small Language Models (SLMs).

SLMs are increasingly gaining attention in research due to their efficiency and practicality. Compared to large-scale models, SLMs require less computational power and are more suitable for deployment in resource-constrained environments. Studies have shown that SLMs can achieve competitive performance in domain-specific tasks, especially when fine-tuned with high-quality datasets [11]. For instance, datasets such as cLegal-QA provide real-world legal queries and responses [12], while models like SaulLM-7B are trained on extensive legal corpora to improve legal reasoning capabilities [13]. Other research has focused on fine-tuning models like LLaMA for specialized tasks such as bar exam question answering [14].

In addition to question answering and document analysis, legal AI research has also explored advanced techniques such as integrating knowledge graphs with RAG to enhance reasoning capabilities [15]. Benchmark datasets like Bar Exam QA and Housing Statute QA further support the evaluation of reasoning-based legal systems [16]. Another important area of research is legal outcome prediction, where models are trained to predict judicial decisions based on past case data. Systems such as PILOT and datasets like CLC-UKET demonstrate the potential of AI in predicting legal outcomes in employment tribunals [17][18]. Cross-lingual legal systems, such as Open French Law RAG, also enable access to legal information across different languages [19].

Furthermore, recent studies emphasize the importance of computational law frameworks, including datasets, ontologies, and evaluation benchmarks, in building reliable legal AI systems [20][21]. A growing trend in this field is the adoption of agentic AI architectures. These systems incorporate autonomous decision-making processes, enabling models to

perform tasks such as classification, reasoning, and validation in a structured workflow. Research by Belcak et al. highlights that Small Language Models are particularly well-suited for agentic systems due to their efficiency and speed [22]. Similarly, systems like LawGPT demonstrate how integrating legal knowledge with language models can improve the quality of responses [23]. Advanced approaches, such as agentic RAG, combine retrieval mechanisms with reasoning and validation processes to enhance accuracy and reduce errors [24].

Based on these developments, it is evident that combining Small Language Models with advanced architectures such as RAG and agentic systems can significantly improve the performance of legal AI applications. Therefore, this research aims to explore and evaluate such architectures in the context of Sri Lankan Property Law and Family Law. In particular, it focuses on developing a system that can provide fast, accurate, and step-by-step legal guidance, while also reducing response time through the use of efficient agentic workflows.

Paper	Technologies Used	Strengths	Limitations
[8] Unsloth finetuning guide	Unsloth, PEFT (LoRA/QLoRA), 4-bit quantization, gradient checkpointing	Low VRAM; fast experiments; clear recipes; reproducible runs	Tool/docs, not peer-reviewed; NVIDIA/GPU bias; quality depends on data
[9] LoRA guide (Databricks)	LoRA adapters, rank-r tuning, target-module selection	Fewer trainable params; cheaper/faster training; practical hyperparameter tips	Can underfit at low rank;needs strong base model; integration varies by architecture
[11] SLM systematic review	Survey of SLMs, distillation, quantization, PEFT, retrieval add-ons	Broad landscape view; privacy/edge use cases; identifies trends	No implementation; general conclusions; may lag newest work

[13] SaulLM-7B (legal)	Mistral-7B base, ~30B-token legal corpus, instruction tuning, eval suite	Strong legal baseline; "small" enough for singleGPU; domainadapted	US/EU-centric; still compute-heavy to retrain; transfer to Sri Lanka uncertain
[14] Llama SFT for Bar Exam	Llama-2 7B / Llama3 8B, supervised fine-tuning on 1,514	Shows small models can reach nearhuman on bar MCQs; releases SFT assets	Narrow task (US bar MCQ); not procedural guidance; exam bias
	IRAC-style Q&A, adapters		
[22] SLMs for Agentic AI	Position + design patterns; agentic pipelines; heterogeneous SLM/LLM mix	Strong case for SLMs in agents (cost/latency); practical design guidance	Limited empirical results; not legalspecific; high-level framing only

Table 1:Small Language Models (SLM) Focused Papers

Table.1 Shows above summarizes key research works and tools relevant to this study, highlighting the technologies used, their strengths, and associated limitations. It can be observed that recent advancements in fine-tuning techniques, such as Unsloth and LoRA, enable efficient training of language models with reduced computational requirements. These methods are particularly beneficial for developing domain-specific applications, as they allow faster experimentation and lower resource usage. However, their effectiveness largely depends on the quality of the training data and the strength of the base model.

The systematic review of Small Language Models (SLMs) provides a broad understanding of current trends, including the use of distillation, quantization, and parameter-efficient fine-tuning techniques. While SLMs offer advantages in terms of cost, speed, and suitability for edge environments, they often require careful design and optimization to match the performance of larger models. Domain-specific models such as SaulLM-7B demonstrate the potential of training on large-scale legal datasets to improve performance in legal tasks, although their applicability may be limited when transferring to different legal systems, such as Sri Lanka.

Similarly, studies on supervised fine-tuning of models like LLaMA for bar exam tasks indicate that smaller models can achieve high performance in structured evaluation settings. However, these approaches are often limited to specific tasks, such as multiple-choice questions, and may not generalize well to real-world legal advisory scenarios that require procedural guidance. Furthermore, research on agentic AI architectures emphasizes the importance of structured workflows in improving efficiency, accuracy, and response time. While these approaches show strong potential, they are still evolving and often lack extensive empirical validation in legal domains.

Overall, the comparison presented in the table highlights that while existing methods provide strong foundations for building efficient legal AI systems, there are still gaps in terms of domain adaptation, real-world applicability, and comprehensive evaluation. These observations motivate the need for this research, which focuses on combining efficient fine-tuning techniques, Small Language Models, and agentic architectures to develop a practical and scalable legal assistance system tailored to the Sri Lankan context.

Paper	Technologies Used	Strengths	Limitations
[5] VidhikDastaavej	Model-agnostic wrapper, RAG retrieval, doc drafting pipeline, Indian legal corpus	Reduces hallucinations; coherent drafts; practical workflow	Tool/docs, not peerreviewed; NVIDIA/GPU bias; quality depends on data
[6] LegalBench-RAG	Benchmark datasets, retrieval tasks, QA pairs, eval metrics	Standardized legalRAG evaluation; reasoning-focused queries	Can underfit at low r; needs strong base model; integration varies by architecture
[7] Thomson Reuters RAG blog	RAG patterns, vector DBs, grounding, guardrails	Clear best practices; industry perspective; design guidance	No implementation; general conclusions; may lag newest work

[15] Bridging Legal Knowledge & AI	RAG + vector stores, knowledge graphs, hierarchical NMF, agentic workflow	Strong grounding; fewer hallucinations; links statutes/cases	US/EU-centric; still compute-heavy to retrain; transfer to Sri Lanka uncertain
[16] Reasoning-Focused Legal Retrieval Benchmark	Retrieval benchmark, Bar Exam QA, Housing Statute QA, expert annotations	Challenging retrieval with low lexical overlap; aids RAG tuning	Narrow task (US bar MCQ); not procedural guidance; exam bias
[19] Open French Law RAG	Cross-language RAG, translation layer, COLD French law corpus, API	Multilingual access; source-linked answers; real legal corpus	Limited empirical results; not legalspecific; high-level framing only

Table 2:Agents, Rag Focused papers

Table.2 presents a comparison of key research works and industry approaches related to Retrieval-Augmented Generation (RAG) and legal AI systems, highlighting the technologies used, their strengths, and their limitations. It can be observed that many modern legal AI solutions, such as VidhikDastaavej, focus on integrating RAG pipelines with document drafting workflows to reduce hallucinations and produce more coherent legal outputs. These approaches demonstrate the practical value of combining retrieval mechanisms with generation models for real-world applications.

Benchmarking efforts like LegalBench-RAG and reasoning-focused legal retrieval datasets provide standardized methods to evaluate the performance of RAG systems. These benchmarks emphasize complex reasoning tasks and help improve model accuracy, particularly in scenarios where simple keyword matching is insufficient. However, such benchmarks are often limited to specific legal systems, such as the United States, and may not fully generalize to other jurisdictions.

Industry perspectives, such as those from Thomson Reuters, highlight best practices in designing RAG systems, including the use of vector databases, grounding techniques, and guardrails to ensure reliability. Similarly, advanced research integrating RAG with knowledge graphs and agentic workflows demonstrates improved reasoning capabilities and stronger

connections between legal concepts, statutes, and case law. These methods significantly reduce errors and enhance the overall quality of generated responses.

Additionally, cross-lingual systems like Open French Law RAG show the potential of extending legal AI across languages, enabling broader accessibility to legal information. Despite these advancements, several limitations remain, including dependency on high-quality data, computational costs, and challenges in adapting models to different legal systems and contexts.

Overall, the table highlights that while RAG-based approaches and related techniques have significantly improved the performance of legal AI systems, there is still a need for more adaptable, efficient, and domain-specific solutions. These observations support the motivation of this research, which aims to develop a tailored legal assistance system for the Sri Lankan context by leveraging efficient architectures and improving accessibility, accuracy, and usability.

Paper	Technologies Used	Strengths	Limitations
[1] Legal NLP Survey	Transformers, BERT-style models, classic NLP pipelines, task taxonomies	Broad overview, maps tasks/datasets, highlights challenges	High-level; not Sri Lanka–specific; older coverage window
[2] LawLLM (US)	Domain corpus pretraining, instruction tuning, multi-task eval, retrieval	Strong US-law performance, legal tasks coverage	US-centric; large compute; limited transfer to Sri Lanka
[3] LexGLUE	Benchmark datasets, unified evaluation metrics, classification tasks	High-quality questions; tests deep reasoning	Mostly English/EU; not procedural guidance; dataset only
[4] JEC-QA (China)	MCQ QA dataset, exam-style reasoning, finetuning baselines	Standardized comparison for legal NLP	Chinese jurisdiction; MCQ format; not step-bystep help

[10] Legal Drafting with FT-LLM	Fine-tuning pretrained LLM, local corpora, drafting prompts	Shows contract/pleading drafting; privacyaware setup	Jurisdiction-specific (CN/TW); hallucination risk; eval scope limited
[12] cLegal-QA	Generative QA, multi-answer references, real user queries	Practical questions; multiple groundtruths	Chinese language/domain; uneven answer styles
[17] PILOT (Outcome Prediction)	Retrieval of precedents, temporal modeling, classifier over case law	Uses case law effectively; handles time shift	Focus on predictions, not guidance; non-LK jurisdiction
[18] CLC-UKET (UK)	Curated outcome dataset, rich annotations, LLMassisted labeling	Real tribunal outcomes; detailed metadata	Employment law only; UK-specific; dataset, not a model
[20] Computational Law Survey	Dataset/ontology catalog, benchmark review	Comprehensive resource map; aids dataset selection	Descriptive; no new models; not Sri Lanka–focused
[23] LawGPT (China)	Knowledgeenhanced LLM, legal corpus integration, instruction tuning	Better legal QA in Chinese; domain grounding	Chinese statutes/cases; heavy training needs; portability limits

Table 3:Large language models (LLM)

The table above presents a comparative analysis of key research studies, datasets, and models in the field of legal Natural Language Processing (NLP), outlining the technologies used, their strengths, and their limitations. It can be observed that foundational works, such as legal NLP

surveys and computational law studies, provide a broad understanding of tasks, datasets, and challenges in the domain. These studies are valuable for identifying research gaps and guiding system design, although they often remain high-level and lack direct implementation details.

Several domain-specific models, including LawLLM and LawGPT, demonstrate the effectiveness of training language models on large legal corpora combined with instruction tuning and retrieval mechanisms. These approaches achieve strong performance in their respective jurisdictions by incorporating domain knowledge. However, their applicability is often limited to specific legal systems, such as the United States or China, making it challenging to directly transfer these models to other contexts like Sri Lanka.

Benchmark datasets such as LexGLUE and JEC-QA play a crucial role in evaluating model performance. They provide standardized tasks for classification and question answering, enabling consistent comparison across models. While these benchmarks test reasoning capabilities effectively, they are often restricted to particular formats, such as multiple-choice questions, and do not fully capture the complexity of real-world legal advisory scenarios that require step-by-step procedural guidance.

Other studies focus on practical applications, including legal drafting and question-answering systems. These works demonstrate the potential of fine-tuned language models in generating contracts, pleadings, and responses to user queries. Similarly, datasets like cLegal-QA provide real-world user questions with multiple reference answers, enhancing model training. However, challenges such as hallucination, inconsistent answer styles, and domain-specific limitations still persist.

Research on legal outcome prediction, such as PILOT and datasets like CLC-UKET, highlights the capability of AI systems to analyze case histories and predict judicial decisions. While these approaches are useful for understanding legal trends, they are primarily focused on prediction rather than providing actionable legal guidance to users. Additionally, most of these datasets are jurisdiction-specific and limited in scope.

Overall, the comparison illustrates that while significant progress has been made in legal AI through the use of language models, datasets, and evaluation benchmarks, existing solutions often face challenges related to domain transferability, real-world applicability, and comprehensive guidance generation. These limitations emphasize the need for developing a

tailored system that combines efficient modeling techniques with domain-specific data. Therefore, this research aims to build a practical and scalable legal assistance system focused on the Sri Lankan context, addressing the identified gaps by providing accurate, step-by-step legal guidance to users

1.1.1 Proposed solution

This research proposes an AI-based legal assistance system designed to make legal knowledge more accessible and easy to understand for ordinary citizens in Sri Lanka. The system uses a fine-tuned Small Language Model (SLM) to provide clear, step-by-step guidance for solving legal issues, especially in Property Law and Family Law.

The solution is built using efficient fine-tuning techniques such as LoRA, allowing the model to perform well while using fewer computational resources. A structured legal dataset is created from books, documents, and expert knowledge to train the model with relevant Sri Lankan legal information.

To improve accuracy and reliability, the system uses advanced architectures such as Retrieval-Augmented Generation (RAG), Agentic workflows, and a hybrid Agentic RAG approach. These methods help the system retrieve correct legal information, generate accurate responses, and validate outputs before delivering them to users.

In addition, the system is designed to support other legal tasks such as deed document risk analysis, legal recommendations in labour law, and predicting possible court outcomes based on past cases. The final system is implemented as a user-friendly web application, enabling users to receive fast, affordable, and reliable legal guidance without depending entirely on professional lawyers.

1.2 Research Gap

Despite significant advancements in legal AI, several critical gaps remain, particularly in the context of Sri Lanka. Existing AI-based legal systems such as general-purpose models (e.g., ChatGPT), domain-specific tools (e.g., LawLLM, LawGPT), and RAG based systems demonstrate strong capabilities in information retrieval and legal question answering. However, these systems are not fully suitable for providing reliable, user-centered legal guidance in the Sri Lankan context.

Firstly, most widely used AI systems rely heavily on publicly available internet data and social media sources. While this enables broad information coverage, it also introduces risks related to misinformation and lack of legal reliability. These systems typically provide general answers rather than structured, step-by-step guidance tailored to user problems. Similarly, local solutions such as Aipazz can retrieve legal information but lack a dedicated guidance mechanism to assist users in resolving their issues effectively.

Secondly, existing legal AI models such as LawLLM (US) and LawGPT (China) are trained on jurisdiction-specific datasets and legal frameworks. Although they demonstrate high accuracy within their respective domains, their applicability to Sri Lankan law is limited. Additionally, these models are based on (LLMs), which require significant computational resources, making them less practical for scalable and cost-effective deployment in resource-constrained environments.

Thirdly, many current RAG-based systems, including Open French Law RAG, focus primarily on retrieving and presenting legal information rather than delivering actionable, user-friendly guidance. These systems often lack structured workflows and do not provide step-by-step procedural assistance, which is essential for real-world legal problem-solving. Furthermore, they may not consistently ensure source grounding or clarity in outputs.

Another key gap is the absence of integrated agentic execution in many existing systems. While some models partially support retrieval or task-based workflows, they do not fully utilize coordinated agent-based architectures to ensure accuracy,

validation, and confidence in responses. As a result, user trust and reliability remain limited.

In contrast, this research identifies the need for a system that combines accurate information retrieval with structured, user-oriented guidance. The proposed approach focuses on developing a SLM based system integrated with agentic RAG architecture. This system aims to provide reliable, step-by-step legal guidance using authorized Sri Lankan legal data, while maintaining low computational cost and high efficiency.

Overall, the key research gaps can be summarized as follows:

1. Lack of reliable, step-by-step legal guidance systems tailored to Sri Lankan users.
2. Limited applicability of existing legal AI models to the Sri Lankan legal context.
3. Absence of efficient, low-cost solutions using Small Language Models.
4. Insufficient use of agentic architectures for improving accuracy, reliability, and user experience.

Addressing these gaps forms the foundation of this research, which aims to develop a practical, scalable, and user-centric legal assistance system for Sri Lanka.

Application	Agentic execution	User reliable guidance	Information accuracy
ChatGPT (or other famous ai models)	Yes, it uses search agent and search tool referring to all the available details in social media and internet.	It's just given the guide point by point in the chat.	provide information from referring all available online ,social media because of wrong information also given to user .
Aipazz (Sri Lankan law ai)	Do not have guidance section for user problem.	Just can get the information from searching	Can get accurate information from searching but Do not have guidance section for user problem.

Our Ai bot	Give accurate user reliable output using coordinate, confidence agent	Get the smooth accurate diagrammatic guide with good user experience .	Can get accurate information using authorized data with agentic rag execution, from output guidance show in user reliable manner using small computational power(slm).
LawLLM (US legal LLM)	Limited; taskoriented, can pair with RAG	Structured answers; not Sri Lanka-specific	High for US law; risk of mismatch for Sri Lanka, but it's llm it's need more computational power
LawGPT (Chinese legal LLM)	Uses legal knowledge enhancement; RAG optional	Good explanations; not localized to Sri Lanka	Strong on CN law; low transfer to Sri Lankan context,but it's llm it's need more computational power.
Open French Law RAG (Harvard LIL)	No; form-based or FAQ replies	Basic tips; lacks step-by-step workflows	Varies; may not cite sources or statutes.

Table 4:Comparative Analysis of Research Features

1.2.1 Critical Analysis of Existing Research

Existing research in legal AI has made significant progress in applying NLP techniques to legal tasks such as document classification, legal question answering, drafting, and case outcome prediction. Benchmark datasets such as LexGLUE and JEC-QA have enabled standardized evaluation, while domain-specific models like LawLLM and LawGPT have demonstrated strong performance within specific legal systems. These studies highlight the potential of AI in improving legal processes and automating complex tasks.

However, a critical analysis reveals several limitations. Most existing models are trained on datasets from jurisdictions such as the United States, Europe, and China. As a result, their knowledge, legal reasoning, and outputs are not directly transferable to other legal systems,

including Sri Lanka. This lack of localization reduces their practical applicability in real-world scenarios outside their original domain.

Another major limitation is the reliance on LLMs, which require high computational resources for training and deployment. While these models achieve strong performance, they are often costly and inefficient for deployment in resource-constrained environments. In contrast, SLMs are more efficient, but there is limited research on applying them effectively in legal domains, particularly for step-by-step guidance tasks.

Furthermore, many existing systems focus primarily on information retrieval and general question answering rather than providing structured, actionable legal guidance. Real-world users require clear, step-by-step instructions to solve their legal problems, which is not adequately addressed in most current solutions. Additionally, issues such as hallucination, lack of grounding in legal sources, and inconsistent answer quality continue to affect system reliability.

Although RAG has improved factual accuracy by incorporating external knowledge sources, many implementations lack advanced reasoning and validation mechanisms. Emerging approaches such as agentic RAG show promise by combining retrieval, reasoning, and validation in a structured workflow. However, these methods are still in early stages and have not been extensively applied to localized legal systems.

In summary, while existing research provides a strong foundation for legal AI, it lacks domain adaptation, cost-efficient solutions, and user-focused guidance mechanisms. These limitations highlight the need for a system that combines Small Language Models, domain-specific data, and advanced architectures such as agentic RAG to deliver accurate, reliable, and practical legal assistance tailored to the Sri Lankan context.

1.2.2 Critical Evaluation of Existing Systems

Existing legal AI systems provide useful support for legal tasks, but they have several limitations. General AI models like ChatGPT can answer legal questions, but they rely on internet data, which may include incorrect or outdated information. They also do not always provide clear, step-by-step guidance for users.

Some systems, such as Aipazz, can retrieve legal information related to Sri Lankan law. However, they mainly provide search-based results and lack structured guidance to help users solve their problems.

Advanced legal models like LawLLM (US) and LawGPT (China) show strong performance in their own legal systems. However, they are trained on foreign legal data, so their answers may not match Sri Lankan laws. In addition, these models require high computational resources, making them expensive to use.

RAG-based systems improve accuracy by retrieving relevant legal documents before generating answers. However, many of these systems focus only on information retrieval and do not provide user-friendly, step-by-step solutions. They also lack proper validation and user guidance.

Overall, current systems do not fully meet the needs of Sri Lankan users. They lack localization, affordability, and clear guidance. This creates a need for a system that provides accurate, reliable, and step-by-step legal support using efficient and low-cost methods.

1.2.3 Identified Research Gaps

- Lack of Sri Lankan Legal Data

  Most existing systems are trained on foreign legal datasets (US, China, EU). There is no proper dataset focused on Sri Lankan Property Law and Family Law.

- No Step-by-Step Guidance

  Current AI systems mainly give general answers, but do not provide clear, structured, step-by-step legal guidance for users.

- High Computational Cost

  Many legal AI models use large language models (LLMs), which require high computational power and are expensive to deploy.

- Limited Use of Small Language Models (SLMs)

There is very little research on using SLMs for legal tasks, especially for providing practical legal advice.

- Weak Retrieval and Accuracy Control

  Existing systems may give incorrect or unverified information due to lack of proper retrieval and validation methods.

- No Agentic Workflow

  Most systems do not use agent-based processes (planning, reasoning, validation), which are needed for better accuracy and reliability.

- Poor User Experience

  Current systems do not focus on user-friendly outputs such as structured steps, clear explanations, or guided solutions.

1.2.4 Proposed Solution

This research proposes the development of an AI-based legal assistance system designed to provide accessible, affordable, and reliable legal guidance for users in Sri Lanka. The system focuses on Property Law and Family Law, which are commonly encountered yet complex for ordinary citizens. To achieve this, a structured dataset will be created by collecting legal information from books, documents, and expert sources, and converting it into a machine-readable format. SLM will then be fine-tuned using efficient techniques such as LoRA and Unsloth, enabling high performance with low computational cost.

To improve accuracy and reliability, the system will integrate an Agentic RAG architecture. This approach allows the model to retrieve relevant legal information from trusted sources, generate responses, and validate outputs before presenting them to users. The system is designed to provide step-by-step legal guidance in a clear and user-friendly manner, helping users understand procedures and make informed decisions. Additionally, it ensures that outputs are based on authorized data, reducing the risk of incorrect information. Finally, the solution will be deployed as a web-based application, offering a smooth user experience and enabling individuals to access legal assistance quickly and efficiently without depending entirely on professional lawyers.

1.3 Research Problem

Access to legal knowledge is essential for ensuring justice, fairness, and informed decision-making in society. However, in many countries, including Sri Lanka, legal information remains difficult for ordinary citizens to access and understand. Legal knowledge related to areas such as Property Law and Family Law is often stored in complex legal texts, printed books, or scattered digital sources. These formats are not easily understandable for non-experts, making it challenging for individuals to interpret laws and procedures correctly. As a result, people are forced to depend heavily on legal professionals, even for minor issues, leading to increased costs and delays.

Existing digital solutions and Artificial Intelligence (AI)-based systems attempt to address this issue by providing legal information through automated responses. However, most of these systems rely on general internet data, which may include outdated, incomplete, or incorrect information. Furthermore, widely used models such as large language models (LLMs) are trained on foreign legal datasets, primarily from jurisdictions like the United States, Europe, or China. This limits their ability to provide accurate and context-specific guidance for Sri Lankan legal scenarios.

Another significant limitation of current systems is their inability to provide structured, step-by-step legal guidance. Most AI tools generate general answers rather than offering clear procedural instructions that users can follow. Additionally, these systems often lack proper validation mechanisms, increasing the risk of misinformation and reducing user trust. While Retrieval-Augmented Generation (RAG) techniques have improved factual accuracy by incorporating external sources, they are not always optimized for fast response times or user-friendly outputs.

Moreover, the majority of existing solutions depend on large-scale models that require high computational resources, making them costly and difficult to deploy in resource-constrained environments. There is limited research on the use of Small Language Models (SLMs) for legal applications, particularly in delivering practical, real-world guidance.

Therefore, the core research problem addressed in this study is the lack of a cost-effective, accurate, and user-friendly legal assistance system tailored to the Sri Lankan context. Specifically, there is a need for a system that can provide reliable, step-by-step legal guidance using locally relevant data, while ensuring efficiency, scalability, and accessibility for the general public.

1.3.1 Empirical Evidence

An empirical survey was conducted with a sample of 50 participants, including legal professionals, law students, and members of the general public, to understand the need for an intelligent legal support system in Sri Lanka. The results clearly indicate a strong demand for such a solution, with more than 93% of respondents expressing the need for an AI-based system that can provide guidance in Family Law and Property Law within the Sri Lankan context.

The survey findings highlight several key challenges faced by users when accessing legal information. Firstly, many participants reported difficulty in understanding complex legal language and procedures, especially those without a legal background. Secondly, respondents emphasized the high cost and time required to obtain professional legal advice, even for minor legal issues. This creates a barrier for individuals seeking timely assistance. Thirdly, there is a lack of easily accessible and reliable legal resources tailored to Sri Lankan laws, making it difficult for users to find accurate information independently.

These findings strongly support the need for a user-friendly, affordable, and reliable legal assistance system that can simplify legal processes and improve access to justice for the general public.

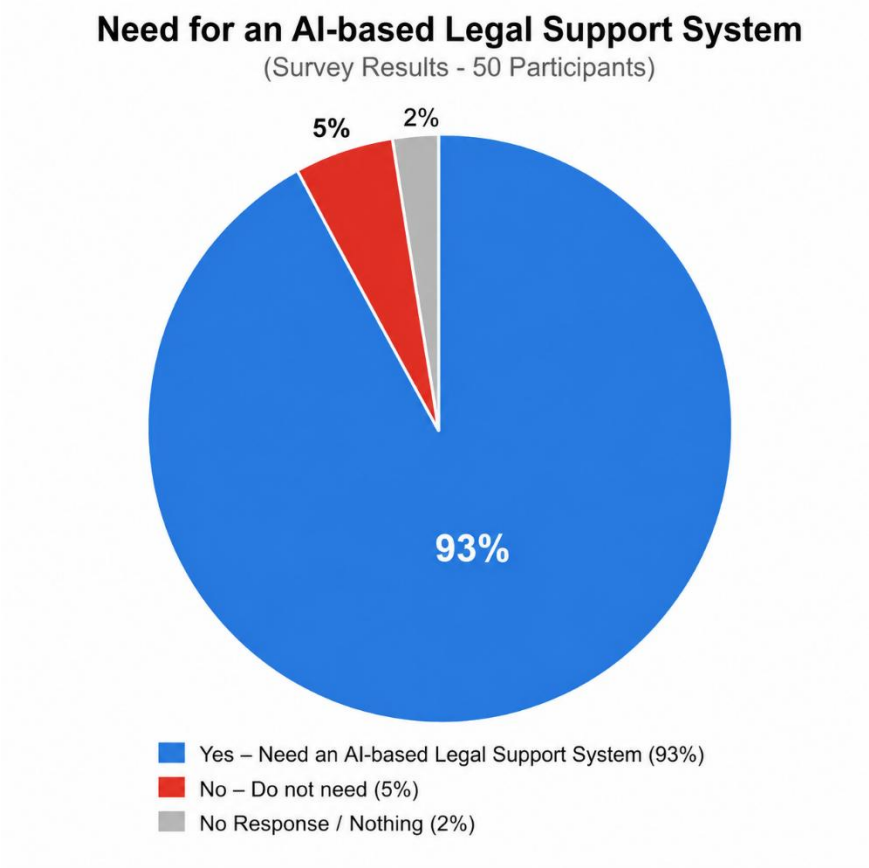


Figure 1:Survey of existing legal information

1.3.2 Technological Limitations in Existing Systems

Existing legal AI systems face several technological limitations that affect their performance, reliability, and usability. One major challenge is the high computational requirement, as most systems rely on large language models (LLMs) that need powerful hardware such as GPUs and servers, making them expensive to run and maintain. In addition, many models depend heavily on general internet data, which may be outdated, incomplete, or incorrect, leading to unreliable outputs. Another limitation is the lack of domain-specific training, as most systems are trained on foreign legal datasets and are not adapted to Sri Lankan laws, reducing their accuracy in local contexts.

Furthermore, these systems often struggle with understanding complex legal queries, especially those requiring detailed reasoning or multi-step explanations. The absence of strong validation mechanisms increases the risk of hallucination, where the system may generate incorrect or fabricated information. Retrieval-based systems, such as RAG, can also experience slow response times when processing large datasets or complex queries. Additionally, poor integration of components like retrieval, reasoning, and validation weakens overall system performance. Finally, most systems provide unstructured text responses rather than clear, step-by-step guidance, making it difficult for users to follow and apply the information effectively.

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:

1 .To collect and prepare a Sri Lankan legal dataset
Gather Property Law and Family Law data from books, documents, and expert sources, and convert it into a structured format for model training.

2. develop and fine-tune a Small Language Model (SLM)
Train the model using efficient techniques such as LoRA and Unsloth to achieve high performance with low computational cost.

3. implement Retrieval-Augmented Generation (RAG)
Integrate retrieval mechanisms to ensure the model generates responses based on relevant and authorized legal information.

4. design an agentic workflow

Develop a system that includes classification, reasoning, and validation steps to improve accuracy and reliability of outputs.

5. provide step-by-step legal guidance

Ensure the system delivers clear, structured, and easy-to-follow legal advice for user problems.

6. evaluate system performance

Measure accuracy, response time, and reliability of the model across different architectures.

7. develop a user-friendly web application

Create an interface that allows users to easily access fast, affordable, and reliable legal assistance.

2. METHODOLOGY

2.1 Key Technical Foundations of the Proposed System

The proposed system is built on a combination of modern Artificial Intelligence techniques to deliver accurate and reliable legal guidance. The core foundation includes a SLM that is fine-tuned using efficient methods such as LoRA and Unsloth. To improve accuracy and reduce incorrect outputs, the system integrates RAG, which retrieves relevant legal information before generating responses.

In addition, an agentic architecture is used to manage the workflow through steps such as classification, reasoning, and validation. A structured legal dataset is developed specifically for Sri Lankan Property Law and Family Law to ensure domain relevance. Finally, the system is deployed through a web-based application to provide a user-friendly interface. These combined technologies ensure the system is accurate, efficient, low-cost, and practical for real-world use.

2.1.1 Fine-Tuning Techniques (LoRA & Unsloth)

To improve the performance of the Small Language Model (SLM), this research applies efficient fine-tuning techniques, specifically Low-Rank Adaptation (LoRA) and Unsloth. Traditional fine-tuning methods require updating all model parameters, which is computationally expensive and time-consuming. In contrast, LoRA reduces the number of trainable parameters by introducing low-rank matrices into selected layers of the model. This approach significantly decreases memory usage and training cost while maintaining strong performance. It allows the model to adapt effectively to domain-specific tasks such as Sri Lankan legal guidance without requiring full retraining.

In addition, Unsloth is used as an optimization framework to further enhance the fine-tuning process. Unsloth improves training efficiency by enabling faster computation, reduced GPU memory consumption, and smoother integration with parameter-efficient methods like LoRA. It also supports techniques such as 4-bit quantization and gradient checkpointing, which further minimize resource requirements.

By combining LoRA and Unsloth, the system achieves a balance between performance and efficiency. This enables the development of a high-quality legal AI system that can be trained and deployed using limited computational resources, making it suitable for real-world applications in resource-constrained environments like Sri Lanka.

2.1.2 Agentic Retrieval-Augmented Generation (Agentic RAG)

Agentic Retrieval-Augmented Generation (Agentic RAG) is used in this system to improve the accuracy, reliability, and reasoning ability of legal responses. Unlike traditional models that generate answers based only on learned knowledge, Agentic RAG combines retrieval mechanisms with an agent-based workflow. This means the system first retrieves relevant legal documents from a structured database and then uses the language model to generate responses based on that information.

The “agentic” aspect introduces a structured process that includes multiple steps such as query classification, information retrieval, reasoning, and validation. First, the system identifies the type of legal query (e.g., property law or family law). Then, it retrieves the most relevant legal content using vector search techniques. After retrieval, the model generates a response using the retrieved context. Finally, a validation step ensures that the output meets quality standards, includes proper legal reasoning, and avoids misleading information.

This approach reduces hallucination, improves factual accuracy, and ensures that responses are grounded in real legal sources. Additionally, it allows the system to provide clear, step-by-step guidance tailored to user queries. Overall, Agentic RAG enhances both the performance and trustworthiness of the legal assistance system.

2.1.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation is a technique used to improve the accuracy and reliability of AI-generated responses. Instead of relying only on the model’s internal knowledge, RAG allows the system to retrieve relevant information from external sources such as legal documents, databases, or structured datasets before generating an answer. This is especially important in legal applications, where accuracy and correctness are critical.

In this research, RAG is used to fetch Sri Lankan legal content related to Property Law and Family Law. The retrieved information is then provided as context to the model, helping it generate more accurate and grounded responses. This approach reduces the risk of hallucination and ensures that answers are based on real legal sources. Overall, RAG improves both the quality and trustworthiness of the system’s outputs.

2.1.4 Transformer-Based Small Language Models

Transformer-based Small Language Models form the core of the proposed system. These models are built using transformer architecture, which is designed to understand relationships between words in a sentence using attention mechanisms. Unlike large language models, SLMs are smaller in size, require less computational power, and are faster to train and deploy.

In this research, the SLM is fine-tuned using Sri Lankan legal data to improve its understanding of local legal terms and procedures. Despite their smaller size, these models can perform well in domain-specific tasks when trained properly. This makes them suitable for building cost-effective and efficient systems that can run in environments with limited resources, while still delivering accurate and meaningful responses.

2.1.5 Natural Language Processing

Natural Language Processing is a field of Artificial Intelligence that focuses on enabling machines to understand, interpret, and generate human language. NLP techniques are essential for building systems that can interact with users in a natural and meaningful way.

In this research, NLP is used to process user queries, understand legal language, and generate clear responses. It helps the system perform tasks such as text classification, information extraction, and question answering. By applying NLP techniques, the system can interpret complex legal questions and provide simplified explanations in an easy-to-understand format. This improves user experience and makes legal information more accessible to non-experts.

2.2 Integrated Research Approach and System Methodology

This research adopts an integrated approach that combines data preparation, model fine-tuning, and advanced system architecture to develop an efficient legal assistance system for Sri Lanka. The methodology begins with the collection of legal data from reliable sources such as books, documents, and expert materials related to Property Law and Family Law. This data is then cleaned, structured, and converted into a machine-readable format to ensure consistency and usability for model training.

A SLM is selected as the core component due to its efficiency and suitability for resource-constrained environments. The model is fine-tuned using parameter-efficient techniques such as LoRA and optimized with Unsloth to achieve high performance with reduced computational cost. This ensures that the system remains scalable and practical for real-world deployment.

To enhance accuracy and reliability, the system integrates an Agentic Retrieval-Augmented Generation (RAG) architecture. This allows the model to retrieve relevant legal information from a structured database before generating responses. The agentic workflow further improves performance by introducing steps such as query classification, reasoning, and validation, ensuring that outputs are both accurate and user-friendly.

All components are combined into a unified system and deployed as a web-based application. The system is designed to provide clear, step-by-step legal guidance, improving user understanding and decision-making. Overall, this integrated methodology ensures a balance between accuracy, efficiency, and usability, making the solution suitable for addressing legal accessibility challenges in Sri Lanka.

2.2.1 Agile Principles Applied in the Project

This project follows Agile principles to ensure flexibility, continuous improvement, and efficient development of the legal AI system. Agile methodology allows the system to be developed in small, manageable iterations, making it easier to test, evaluate, and improve each component step by step.

Firstly, the project is developed in incremental stages, where each module such as dataset preparation, model fine-tuning, RAG integration, and web application development is buil and

tested separately. This approach helps in identifying issues early and improving system quality continuously.

Secondly, there is a strong focus on continuous feedback and improvement. The system is regularly evaluated based on accuracy, response time, and user experience. Feedback from testing helps refine the model and improve the overall performance.

Thirdly, Agile supports adaptability to changes. As legal requirements or system needs evolve, the project can easily adjust its design, data, or architecture without affecting the entire system. This is especially important in legal applications where updates and corrections are common.

Additionally, the project emphasizes collaboration and iterative testing. Different system components such as the SLM, RAG pipeline, and agentic workflow are tested together to ensure smooth integration and reliability.

Finally, Agile ensures faster delivery of a working system by focusing on developing a functional prototype early and improving it over time. This approach helps in building a practical, user-friendly, and efficient legal assistance system.

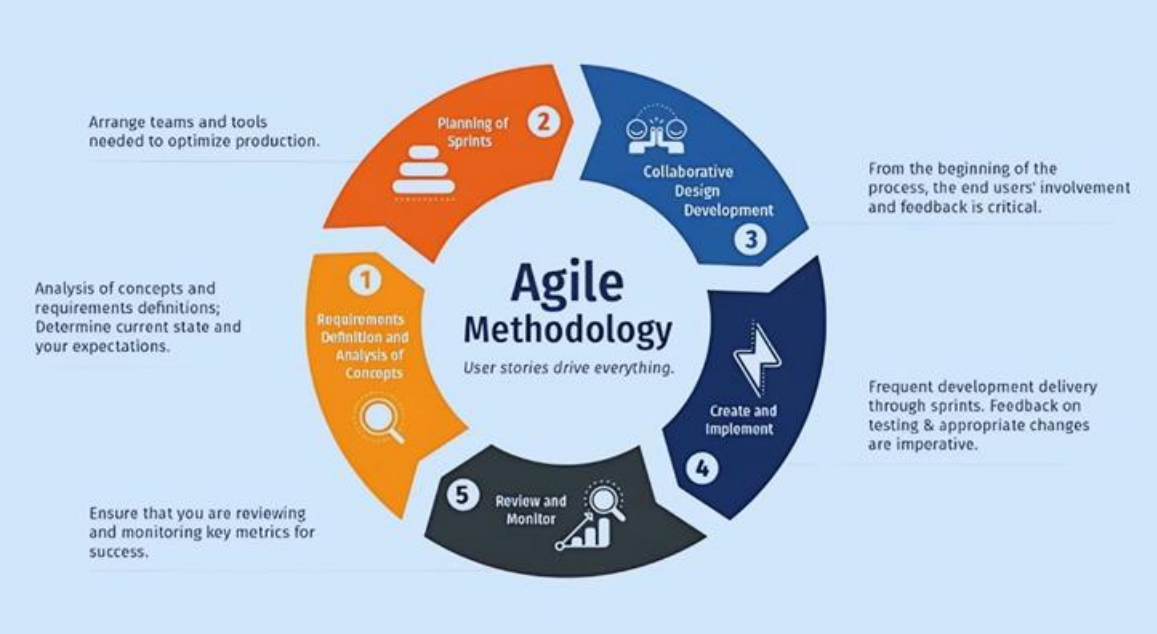


Figure 2: Agile Methodology

2.2.2 Feasibility Study and Planning

This project is evaluated through a feasibility study to ensure that the proposed legal AI system can be successfully developed and implemented in the Sri Lankan context. The study considers technical, economic, operational, and time feasibility to assess the practicality of the solution.

From a technical perspective, the system is feasible because it uses Small Language Models (SLMs), which require lower computational resources compared to large models. Technologies such as LoRA, Unsloth, and Retrieval-Augmented Generation (RAG) are well-established and can be implemented using available tools and frameworks. The use of Python-based frameworks like FastAPI and databases further supports smooth system development and deployment.

In terms of economic feasibility, the project is cost-effective due to the use of parameter-efficient fine-tuning techniques and lightweight models. Unlike large-scale AI systems, this approach reduces the need for expensive hardware and infrastructure, making it suitable for real-world deployment in resource-constrained environments.

From an operational perspective, the system is designed to be user-friendly and accessible through a web-based interface. Users can easily input queries and receive structured legal guidance without requiring technical knowledge. This ensures that the system can be effectively used by the general public.

Regarding time feasibility, the project is planned in phases, including data collection, model training, system integration, and testing. Agile development practices are followed to ensure timely progress and continuous improvement.

2.2.3 Requirement Gathering and Analysis

The requirement gathering and analysis phase is a crucial step in developing the proposed legal AI system. It involves identifying user needs, system functionalities, and technical requirements to ensure the solution effectively addresses real-world legal challenges in Sri Lanka.

The primary stakeholders include ordinary citizens seeking legal guidance, legal professionals who provide domain knowledge, and developers responsible for building the system. User requirements were identified by analyzing common legal issues related to Property Law and Family Law, such as land ownership, deed verification, and family disputes. These users require clear, step-by-step guidance rather than complex legal explanations.

Based on this, the functional requirements of the system include the ability to accept user queries, classify the legal domain, retrieve relevant legal information, and generate structured, step-by-step guidance. The system must also validate outputs to ensure accuracy and reliability. Additionally, it should support features such as legal document analysis, recommendation generation, and basic prediction of legal outcomes.

The non-functional requirements focus on performance, usability, and reliability. The system should provide fast responses, maintain high accuracy, and be accessible through a simple web interface. It should also ensure data security and handle multiple user requests efficiently.

From a technical perspective, the system requires a structured legal dataset, a fine-tuned Small Language Model (SLM), and integration with RAG and agentic workflows. Tools such as FastAPI, databases, and vector search mechanisms are also required for implementation.

Overall, this phase ensures that the system is designed to meet user expectations while maintaining technical feasibility and performance standards.

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

The data collection process for this research involved gathering legal materials related to Sri Lankan Property Law and Family Law from reliable sources. These sources included both digital PDF documents and physical books obtained from a legal professional, ensuring the authenticity and relevance of the information. The collected materials contained detailed legal procedures, rules, and case-related explanations necessary for training the model.

After collection, the data was carefully processed to extract only the most relevant and useful content. This involved cleaning the text, removing unnecessary information, and organizing it into a consistent format. The extracted content was then transformed into a structured dataset using the OpenAI API, which helped in formatting the data into a machine-readable form.

The final dataset consists of approximately 4,700 structured entries, stored in JSONL format. This structured approach ensures efficient training, retrieval, and integration with the model, ultimately improving the system's performance and accuracy.

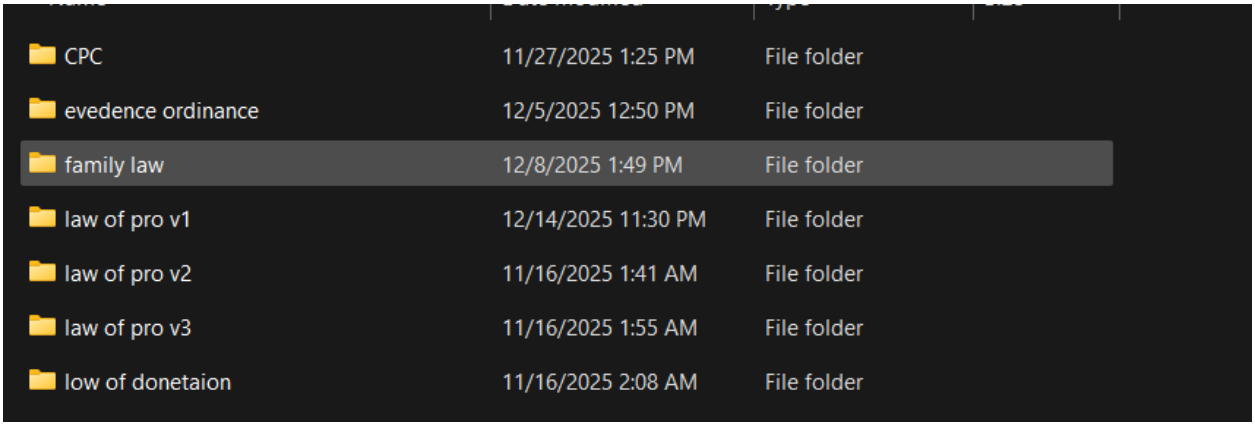


Figure 3:All the data collected data

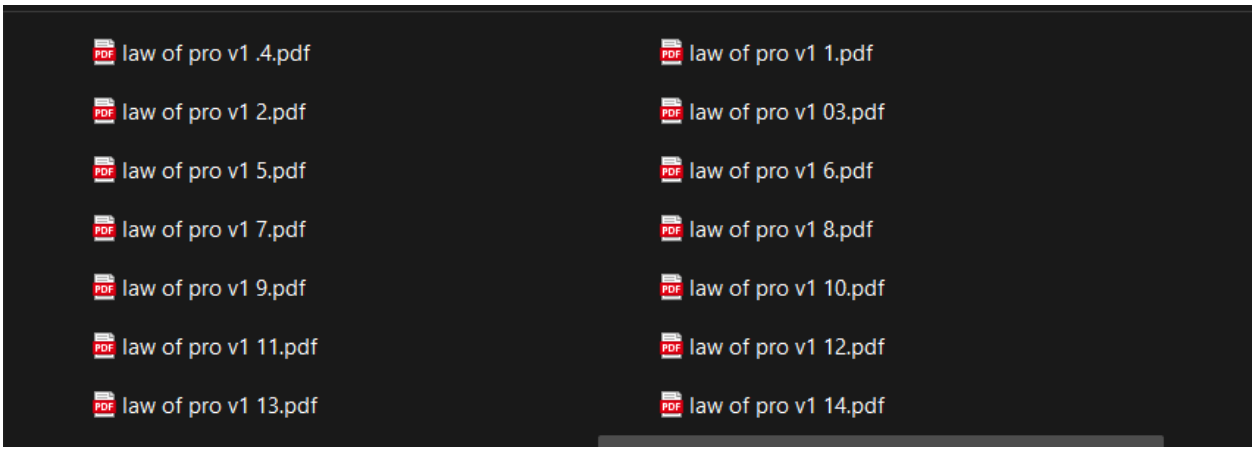


Figure 4:Scaned data collection

2.2.6 OCR Extraction, Cleaning, and Structured Dataset Construction

This stage focuses on converting raw legal materials into a clean and structured dataset suitable for training the model. Since a significant portion of the data was available in scanned PDFs and physical books, Optical Character Recognition (OCR) techniques were used to extract text from these sources. OCR tools were applied to convert images and scanned documents into editable digital text while preserving the original content as much as possible.

After extraction, the text underwent a cleaning process to remove noise such as formatting errors, special characters, duplicate content, and irrelevant sections. Legal terms and important sections were carefully preserved to maintain the accuracy and meaning of the information. The cleaned data was then organized into a consistent structure, ensuring that each entry clearly represented a legal concept, rule, or step-by-step procedure.

Finally, the processed data was converted into a structured JSONL format, where each entry contains fields such as query, context, and response. This structured dataset improves the efficiency of model training and retrieval processes. Overall, this step ensures that high-quality, machine-readable data is available, which directly contributes to the accuracy and reliability of the proposed legal AI system.

2.2.7 Data Governance, Splitting, and Schema Validation

Data governance plays a critical role in ensuring the quality, consistency, and reliability of the dataset used in this research. All collected legal data was carefully managed to maintain accuracy, integrity, and proper organization. Sensitive or irrelevant information was excluded, and only validated legal content from trusted sources was included. Standard naming conventions and consistent formatting were maintained throughout the dataset to ensure uniformity.

The final dataset consists of approximately 4,700 structured entries. To support effective model training and evaluation, the dataset was divided into three subsets. About 3986 of the data was allocated for training, allowing the model to learn legal patterns and relationships. The remaining data was split equally, with 597 used for testing and 398 for validation. The validation set was used during training to monitor model performance and prevent overfitting, while the testing set was used to evaluate the final performance of the model.

Additionally, schema validation was applied to ensure that each dataset entry follows a consistent structure. Each record was checked to confirm the presence of required fields such as query, context, and response. This process ensures data quality and supports efficient model training and reliable system performance.

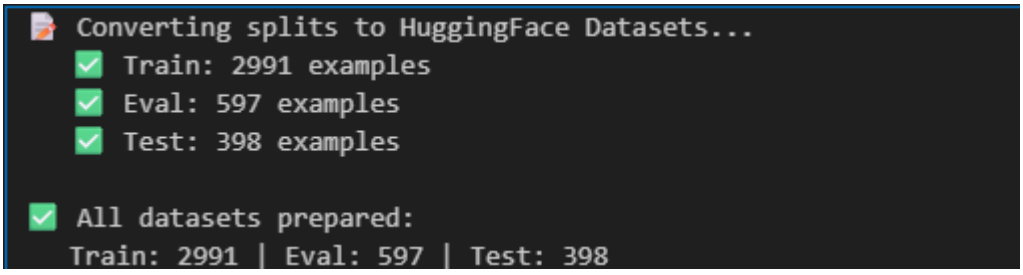


Figure 5:Splitting, and Schema Validation

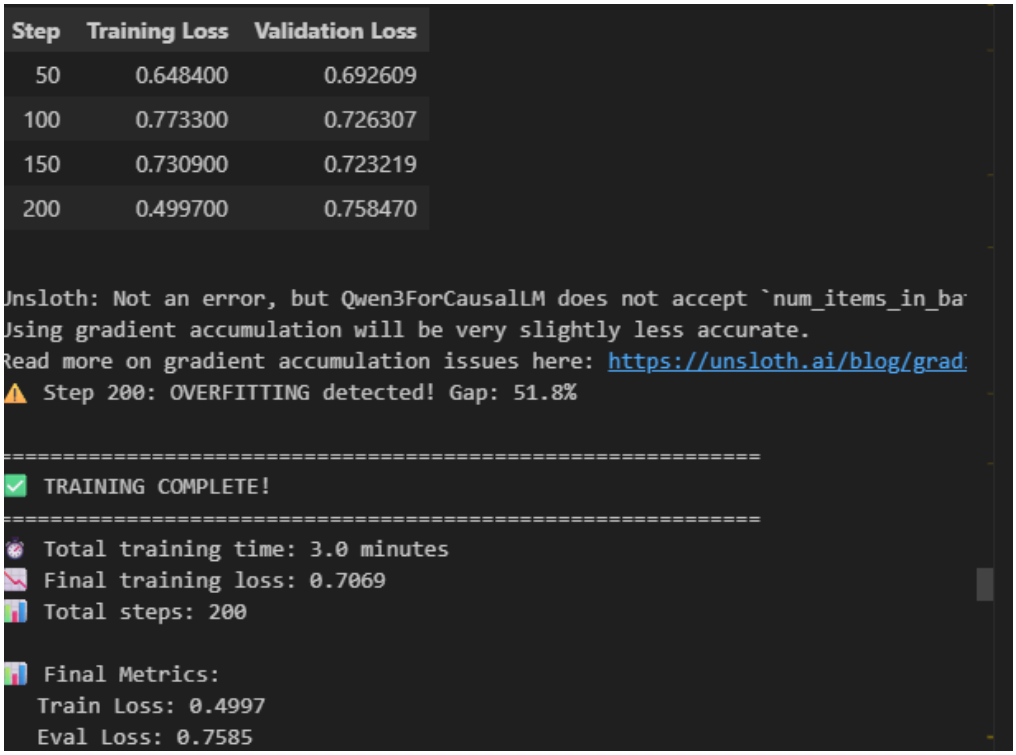
2.2.8 Model Fine-Tuning Strategy

The fine-tuning strategy centres on a small language model (Qwen3-1.7B) adapted with LoRA through Unsloth, prioritising efficient training on high-memory accelerators (for example A100 in Google Colab) while preserving acceptable quality for step-by-step legal guidance. LoRA was chosen to update only a low-rank adapter on top of frozen base weights, which reduces trainable parameters and memory footprint compared to full fine-tuning, and per the notebook design explicitly favours LoRA over QLoRA where the goal is quality under sufficient GPU memory rather than extreme compression.

The strategy should be described in terms of objective, data format, and hyperparameters: supervised instruction or conversational pairs aligned with the desired behaviour (numbered steps, plain language for non-lawyers, disclaimers where appropriate), maximum sequence length, learning rate schedule, batching, number of epochs, early stopping or checkpoint selection against a validation loss, and regularisation implied by the training recipe (dropout if used, weight decay, gradient clipping). The notebook's extended evaluation blocks curve analysis for overfitting and underfitting, and automated reporting support claims that the chosen schedule is not merely a single lucky run.

Finally, the strategy connects to deployment: after merging or exporting adapters, the tuned weights are served through an OpenAI-compatible remote inference endpoint (Modal-hosted Ollama in the current configuration), decoupling training from runtime. That separation allows

the thesis to argue for a reproducible pipeline: train once, version the artifact, and serve with fixed decoding parameters unless ablation studies require otherwise.



Step	Training Loss	Validation Loss
50	0.648400	0.692609
100	0.773300	0.726307
150	0.730900	0.723219
200	0.499700	0.758470

```
Unsloth: Not an error, but Qwen3ForCausalLM does not accept `num_items_in_ba
Using gradient accumulation will be very slightly less accurate.
Read more on gradient accumulation issues here: https://unsloth.ai/blog/grad
⚠️ Step 200: OVERFITTING detected! Gap: 51.8%

============================================================
✅ TRAINING COMPLETE!
============================================================
⏱️ Total training time: 3.0 minutes
📉 Final training loss: 0.7069
📊 Total steps: 200

📊 Final Metrics:
   Train Loss: 0.4997
   Eval Loss: 0.7585
```

Figure 6:model training

2.2.9 Retrieval-Augmented Generation and Vector Indexing

Retrieval-Augmented Generation grounds the language model in retrieved passages rather than parametric memory alone, which is especially important in law-like domains where factual anchoring and citations improve user trust and auditability. In this project, retrieval uses dense embeddings (sentence-transformers / all-MiniLM-L6-v2) and a FAISS vector index over chunked documents, with configuration for chunk size, overlap, top-k results, and similarity thresholds. Documents are organised under category-aware paths (for example property law vs family law), and the retriever can filter or bias search toward the classified category after a lightweight query classifier based on keywords.

The linear RAG backend implements a single pipeline: classify (if not fixed by the client), retrieve (optionally with a simple rerank that widens initial recall then reorders by lexical overlap with the query), assemble context and citation metadata, and prompt the model to answer strictly from the provided context while admitting ignorance when support is thin. The agentic RAG backend adds LangGraph control flow: a router decides whether retrieval is

necessary, a grader judges document relevance, a rewriter can reformulate the query for another retrieval round within a capped budget, and generation is followed by validation with optional regeneration. That design trades some latency for robustness when the first retrieval is noisy. Vector indexing governance includes rebuild procedures after corpus updates, consistency between embedding model and index version, and evaluation of retrieval quality (precision@k, nDCG, or human relevance judgments) complementary to end-to-end answer quality. Together, RAG and FAISS indexing operationalise "grounded generation" as a first-class system component rather than an afterthought.

2.2.10 Runtime Inference, Parsing, and Confidence Synthesis

At runtime, all backends depend on a separate model server exposed through an OpenAI-compatible HTTP API, with the application passing system and user prompts, generation limits, temperature, top-p, repetition penalties, and conversation history where supported. This separation improves operability: the API tier can scale independently, enforce timeouts, and degrade gracefully when the model host is unavailable, while the inference tier can swap quantisation formats or hardware without rewriting business logic.

Parsing is non-trivial for small models fine-tuned for numbered "step" answers. The RAG pipeline applies deterministic post-processing that normalises awkward line breaks, detects the first coherent numbered step sequence, caps the number of steps retained, and truncates repeated or meta-textual tails that small models often emit after the intended answer. Similar discipline applies to agentic flows where validators may attach warnings or trigger limited retries. Parsing policy should be documented as part of the system's output contract, because it affects comparability across baselines and user-visible behaviour.

Confidence synthesis in the RAG service derives partly from retrieval scores (for example averaging similarity among top documents), while the non-RAG agentic backend exposes classifier confidence and validation outcomes. The thesis should caution that such scores are calibrated heuristics, not judicial probabilities: they indicate internal consistency between query and retrieved evidence, not legal certainty. Where the UI or API surfaces confidence, accompanying disclaimers and uncertainty language remain essential for responsible use.

2.2.11 System Integration, Architecture, and Observability

The overall architecture follows a modular microservice pattern: three FastAPI applications (backend for SLM-centric LangGraph, backend_rag for linear RAG, backend_agentic_rag for LangGraph-orchestrated RAG) on distinct ports, sharing a PostgreSQL database for sessions and conversations where configured, and optionally sharing or mirroring vector indices and document directories. Cross-origin middleware permits web frontends to call the APIs during development; production would restrict origins. A shared model client abstraction encapsulates remote completion calls, timeouts, and error translation into HTTP 503 responses when inference is down.

Integration concerns include consistent API versioning under /api/v1, aligned Pydantic schemas for chat requests and responses (including sources, citations, metrics, and reasoning steps where applicable), and backend tagging in persistence layers so analytics can separate traffic by deployment mode (for example backend_type="rag"). Feedback routes link user ratings to stored assistant rows when foreign keys succeed, which requires the chat handler to persist the database primary key of the assistant message used for feedback.

Observability is supported by structured logging around pipeline stages (classification, retrieval, grading, generation, validation), request timing via middleware (X-Process-Time-Ms), and OpenAPI documentation for operator onboarding. A mature deployment would add centralised log aggregation, metrics (request rate, latency histograms, error budgets), and distributed tracing across API and model tiers. The thesis can present the current stack as a research prototype with a clear roadmap from developer logging to production-grade observability and security hardening.

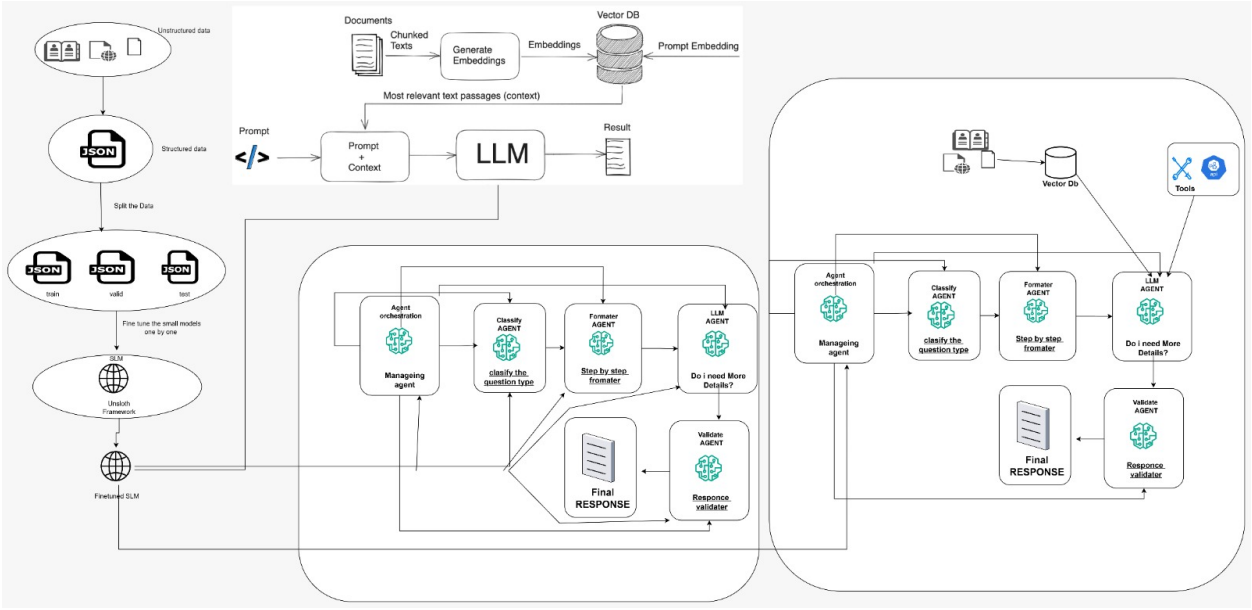


Figure 7:System Architecture Diagram

2.2.12 Evaluation, Reliability Controls, and Iterative Refinement

Evaluation for this assistant must span three layers: (i) model-level evidence from the fine-tuning notebook quantitative scores such as BLEU and ROUGE, semantic similarity to reference answers, loss and metric curves across epochs, and held-out test performance after explicit train/validation/test splitting so that improvements are not anecdotal; (ii) retrieval-level diagnostics when RAG is enabled, including latency decomposed into retrieval versus generation, top-k behaviour under the chosen similarity threshold, and qualitative inspection of whether retrieved passages actually support the final answer; and (iii) system-level acceptance tests on the live FastAPI surfaces (health checks, chat under load, failure modes when the remote model server is unavailable), because deployment constraints change what "good" means in practice.

Reliability controls are implemented both algorithmically and procedurally. Algorithmically, the non-RAG LangGraph path separates classification, reasoning, and validation; the agentic RAG path adds routing, document grading, bounded query rewriting and re-retrieval, and post-generation validation with capped regeneration attempts, which reduces ungrounded answers when retrieval is weak. The linear RAG pipeline further applies deterministic output parsing to stabilise numbered step responses from a small model trimming repetition and meta-text—so that user-visible answers remain coherent even when raw generation is noisy. Procedural controls include schema-validated API contracts, persistence of conversations for

audit, optional user feedback linked to stored assistant messages, and explicit disclaimers that the system provides general guidance rather than professional legal advice.

Iterative refinement closes the loop between measurement and change: notebook-driven analyses (overfitting/underfitting signals, hyperparameter recommendations) inform the next training run; runtime logs and feedback inform prompt adjustments, retrieval thresholds, and validator strictness; and comparative experiments across the SLM-only, linear RAG, and agentic RAG backends yield a principled trade-off narrative between latency, groundedness, and operational complexity. Together, layered evaluation, built-in reliability mechanisms, and a disciplined refinement cycle support claims of scientific rigour rather than one-off demonstration quality.

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.

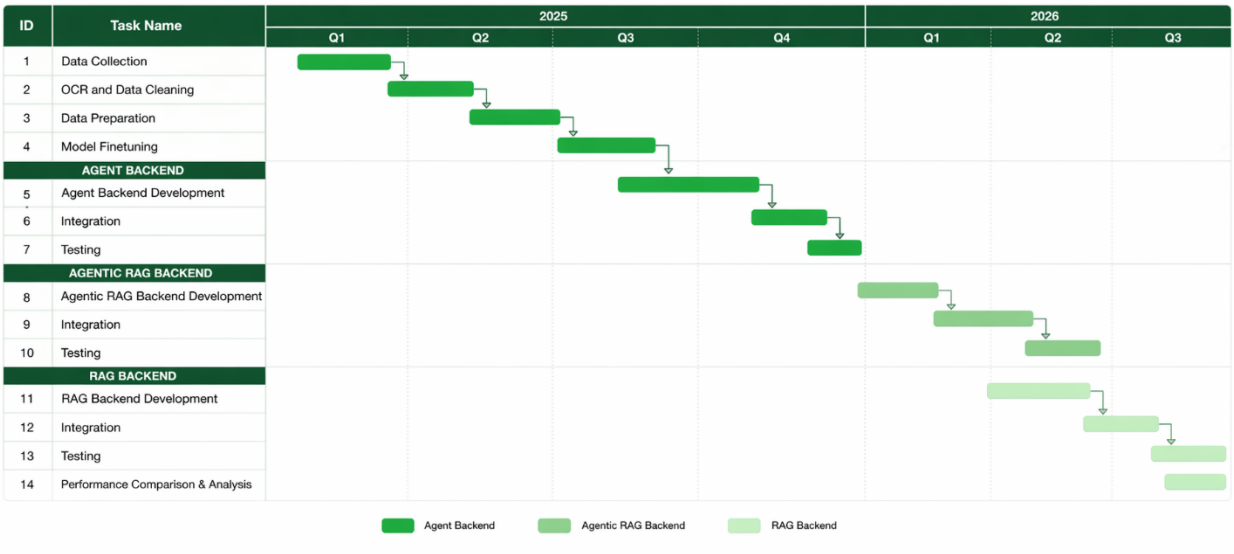


Figure 8:Gantt Chart

2.3 Summary of Methodology

The methodology of this research is designed as a comprehensive, integrated, and practical framework to develop an efficient legal assistance system tailored to the Sri Lankan context. It combines data engineering, model optimization, and advanced AI architectures to ensure accuracy, reliability, and usability. The process begins with the collection of legal data from trusted sources such as books, scanned documents, and expert inputs. Since much of the data exists in unstructured formats, OCR techniques are applied to convert scanned content into digital text, followed by rigorous cleaning and preprocessing to remove noise while preserving legal meaning. The processed data is then structured into a JSONL format, resulting in a dataset of approximately 4,700 entries, which is split into training, validation, and testing sets under strict data governance practices .

At the core of the system, a transformer-based Small Language Model (SLM) is selected due to its efficiency and suitability for resource-constrained environments. The model is fine-tuned using parameter-efficient techniques such as LoRA and optimized with Unsloth, enabling high performance while minimizing computational cost. To improve factual accuracy and reduce hallucination, the system integrates Retrieval-Augmented Generation (RAG), which retrieves relevant legal information from a structured vector database before generating responses.

Furthermore, an agentic architecture is incorporated to enhance reasoning and reliability. This introduces a structured workflow that includes query classification, document retrieval, response generation, and validation. Multiple system variants Agentic, RAG, and Agentic RAG—are developed and evaluated to identify the most effective architecture.

Finally, the system is deployed as a modular web-based application using FastAPI, ensuring scalability and user accessibility. Overall, this methodology ensures a balanced approach that integrates data quality, model efficiency, and system reliability, resulting in a practical, low-cost, and high-performance legal AI solution for Sri Lanka.

2.4. Commercialization aspects of the product

2.4.1 Product Overview and Value Proposition

The proposed legal AI system has strong potential for commercialization as a practical and scalable solution in the Sri Lankan market. The system can be offered as a web-based or mobile application, allowing users to access legal guidance anytime at a low cost compared to traditional legal services.

One possible business model is a freemium model, where basic legal guidance is provided for free, and advanced features such as document analysis, legal recommendations, and case prediction are offered through paid subscriptions. This makes the system accessible to a wide range of users while generating revenue.

The product can also be marketed to law firms, legal consultants, and government organizations as a support tool to improve efficiency and reduce workload. Additionally, partnerships with legal institutions can help improve credibility and adoption.

Since the system uses SLMs and efficient architectures, the operational cost is relatively low, making it suitable for large-scale deployment. With proper data updates and continuous improvement, the product can expand to other legal domains and even other countries.

Overall, the system has strong commercial value as an affordable, scalable, and user-friendly legal assistance solution.

2.4.2 Target Market and Users

The target market for this system includes individuals and organizations that require easy access to legal information in Sri Lanka. The primary users are ordinary citizens who need guidance on Property Law and Family Law issues such as land ownership, deed verification, and family-related legal matters. These users often lack legal knowledge and seek affordable and simple solutions.

Secondary users include small business owners and employees who may need legal advice related to employment or property matters. In addition, law students and researchers can use the system as a learning tool to understand legal concepts in a simplified way.

The system can also be useful for legal professionals, such as junior lawyers or legal assistants, as a support tool to quickly access structured legal information.

Overall, the product targets a wide user base by providing accessible, low-cost, and user-friendly legal assistance for different levels of legal understanding.

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

The deployment of the proposed legal AI system must carefully address legal, ethical, and regulatory factors to ensure responsible and safe usage. From a legal perspective, the system must clearly state that it provides informational guidance only and does not replace professional legal advice. Proper disclaimers should be included to avoid legal liability and ensure users understand the limitations of the system.

Ethically, the system must ensure accuracy, transparency, and fairness in its responses. Since legal advice can impact real-life decisions, it is important to minimize incorrect or misleading outputs. This is addressed through the use of verified legal data RAG, and validation mechanisms. Additionally, the system should avoid bias and provide neutral, unbiased guidance.

Data privacy is another critical consideration. User queries and any stored information must be handled securely, following data protection standards. Sensitive data should be encrypted, and user consent should be obtained before storing or processing information.

From a regulatory perspective, the system should comply with local laws and digital regulations in Sri Lanka, especially those related to data protection and online services. Regular updates and monitoring are required to ensure the system remains aligned with legal changes.

Overall, these considerations ensure that the system is trustworthy, safe, and compliant for real-world use.

2.4.5 Competitive Advantage

The proposed system offers a strong competitive advantage by combining SLMs with agentic RAG architecture to deliver accurate, step-by-step legal guidance tailored to Sri Lanka. Unlike existing systems that rely on general or foreign legal data, this solution is trained on Sri Lankan Property and Family Law, ensuring high relevance and reliability. It provides structured guidance, not just general answers, improving user understanding. Additionally, the system is cost-effective and efficient, requiring lower computational resources compared to large

models. Its user-friendly design, fast response time, and focus on local legal needs make it more practical and accessible than existing legal AI solutions.

2.5 Project Requirements

2.5.1 Functional requirements

The system is designed to support a range of functional capabilities to ensure effective legal assistance for users. It allows users to input legal queries through a web-based interface, enabling easy interaction. Once a query is received, the system performs query classification to identify the relevant legal domain, such as Property Law or Family Law. It then retrieves appropriate legal information from a structured database using Retrieval-Augmented Generation (RAG) techniques. Based on this information, the system generates accurate, step-by-step legal guidance tailored to the user's query.

To ensure reliability, a validation mechanism is applied to check the correctness and quality of responses. The system also supports basic legal document analysis, such as reviewing deeds, and provides recommendations that guide users on the next legal steps. Additionally, it can estimate possible case outcomes based on past legal data. For improved user experience, the system includes session management to store user interactions and enables feedback collection, allowing users to rate responses and support continuous system improvement.

2.5.2 Non-functional requirements

Non-functional requirements define the quality attributes and operational constraints of the system to ensure effective and reliable performance. The system is expected to deliver fast responses with minimal delay, ensuring good performance for users. It must provide accurate and reliable legal information to maintain user trust. Scalability is important so that the system can handle multiple users efficiently without performance degradation.

The system should also be highly usable, with a simple, clear, and user-friendly interface that can be easily used by non-technical users. Security is a key requirement, ensuring that user data is

protected through encryption and secure access controls. Reliability is essential so that the system functions consistently without frequent failures.

In addition, the system should be maintainable, allowing easy updates and improvements over time. It must ensure high availability, meaning users can access it anytime with minimal downtime. Compatibility across different devices and browsers is also required to enhance accessibility. Finally, the system should be cost-efficient, operating with low computational and operational costs, making it practical for real-world deployment.

2.6 Testing and Implementation

2.6.1 Testing

Overview of Testing Phase

The testing phase was structured to separate offline model evaluation from online system validation, because the research artefact spans both a fine-tuned small language model and three deployable FastAPI services that depend on external inference, optional PostgreSQL persistence, and in two variants FAISS vector retrieval. Offline testing establishes whether the adapted Qwen3-1.7B model reproduces the supervised behaviour implied by the training corpus (step-by-step answers, domain vocabulary, and safe refusal patterns) using held-out data and automatic metrics implemented in the training notebook, including loss and auxiliary scores, semantic similarity to references, and qualitative review of sample outputs. Online testing then verifies that each backend correctly orchestrates the model: health and readiness endpoints, chat contracts, error handling when the model server is unreachable, session lifecycle behaviour, and for RAG modes consistency between retrieved passages, cited sources, and the final natural language answer.

Implementation testing also covered cross-cutting concerns: CORS and middleware timing headers for latency visibility, OpenAPI-driven manual checks, and end-to-end flows such as submitting a query, receiving a structured or semi-structured payload, and submitting feedback when a conversation identifier is returned. Where automated unit tests are limited, the phase relied on repeatable manual scripts, logged pipeline stages, and comparison of the same benchmark query suite across backends so that differences in latency, groundedness, and failure modes are attributable to architecture (SLM-only versus linear RAG versus agentic RAG) rather than ad hoc prompting. This division keeps the thesis honest about what is statistically measured versus what is engineering-demonstrated, while still supporting a coherent evaluation narrative.

Fine-Tuned Model Performance Testing

Fine-tuned model performance was evaluated primarily within the Unsloth / LoRA training notebook, after constructing explicit train, validation, and test partitions to limit optimistic bias. Quantitative measures included conventional text overlap metrics such as BLEU and ROUGE, which indicate n-gram overlap with reference answers, together

with embedding-based semantic similarity, which better captures paraphrases and reorderings that legal explanations often exhibit. Training dynamics were monitored through learning curves and supplementary checks aimed at detecting overfitting or underfitting, so that the chosen epoch count, learning rate, and regularisation choices are justified by observed validation behaviour rather than by a single final checkpoint.

Beyond aggregate scores, spot testing used representative property-law and family-law prompts, including edge cases where personal-law regimes differ, to inspect whether the model maintains the intended tone: plain-language steps, explicit uncertainty when facts are incomplete, and disclaimers that outputs are not a substitute for professional advice. Inference speed experiments documented tokens per second or wall-clock latency under the notebook's hardware profile, which matters for user experience even when the deployed system later offloads inference to a hosted GPU service. Together, these tests characterise the parametric component of the system the fine-tuned SLM before retrieval and agentic control layers are introduced, establishing a baseline against which RAG variants can be compared

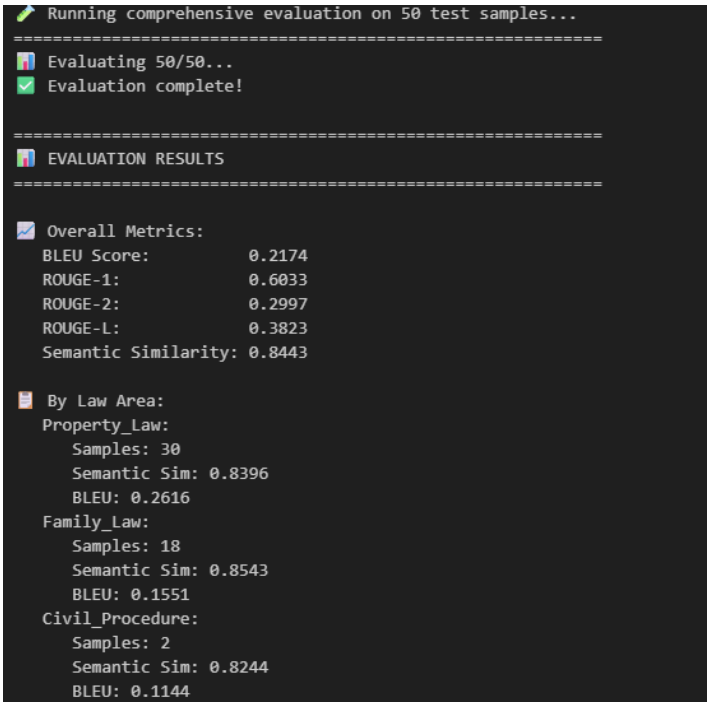


Figure 9:Finetuned model evaluation

Agentic architecture Testing

The agent backend (SLM-centric service) was tested as an integrated LangGraph workflow rather than as a bare completion endpoint. Functional tests exercised the main chat route with empty queries rejected, valid queries accepted, and dependency behaviour when the remote model server is not ready (typically surfacing a service-unavailable style response). Because the graph runs classify → reason → validate, tests included prompts designed to trigger each legal topic branch as well as ambiguous queries, verifying that classification metadata, confidence, validation warnings, and optional structured fields appear consistently in the API response and—when the database is available persist with the expected session identifiers.

Regression testing for this backend also covered session and history endpoints where implemented: creating sessions, retrieving history, clearing history, and fetching prior responses by identifier, including fallbacks when persistence fails so the API remains usable in degraded configurations. Logging output was used to confirm that the synchronous agent runner completes within expected time bounds for benchmark queries and that formatter post-processing does not corrupt markdown or legal references. This backend serves as the non-retrieval baseline in comparative testing, isolating the benefit of fine-tuning and light validation from document grounding.

Figure 10:Agentic Architecture testing

Agentic RAG Architecture Testing

Agentic RAG testing emphasised multi-stage behaviour under LangGraph: routing decisions, retrieval rounds, document grading, bounded query rewriting, generation, and post-generation validation with capped retries. Test cases included queries that clearly require statutes or procedural steps (where retrieval should activate), conversational follow-ups that should respect history when the with-history endpoint is used, and queries where retrieval is expected to struggle (to observe rewrite loops and eventual graceful degradation). Assertions focused on response payloads carrying reasoning steps, warnings, and performance metrics that decompose retrieval time, generation time, total latency, retrieval round counts, and generation attempts—fields that make the agentic path auditable in a thesis results table.

Operational tests confirmed that the synchronous graph invocation is offloaded from the async event loop so concurrent HTTP traffic does not stall trivially, and that vector-store readiness checks align with actual index state. Failure injection stopping the model server or corrupting the index path validated that errors surface as controlled HTTP errors rather than unhandled

stack traces. Agentic RAG is positioned as the adaptive variant; its tests therefore stress conditional paths and stability, not only average answer quality on easy prompts.

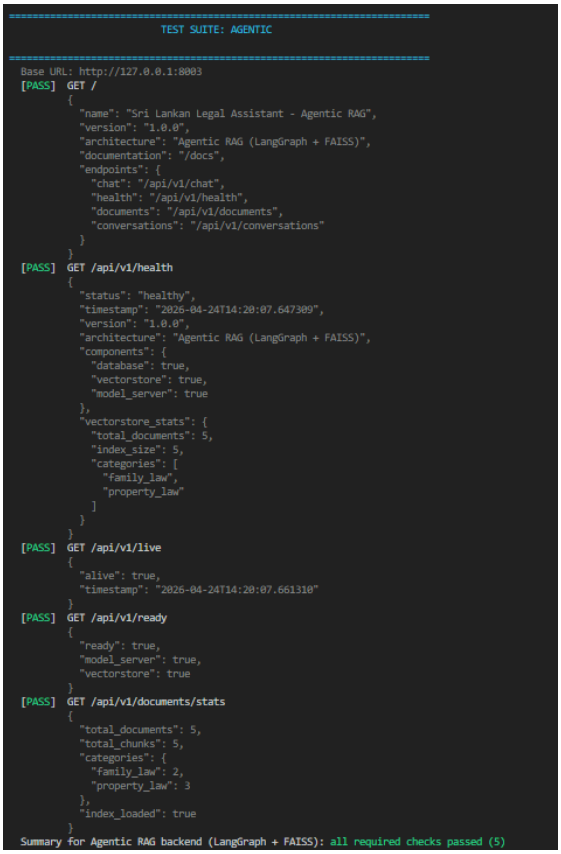


Figure 11:Agentic RAG testing

RAG Architecture Testing

The linear RAG backend was tested around the retrieve–contextualise–**generate** pipeline backed by FAISS and sentence-transformer embeddings. Core tests verified that the vector store initialises on startup, that category-aware retrieval respects configured thresholds, and that optional reranking changes ordering in predictable ways on synthetic keyword-heavy queries. Chat tests compared responses with and without include_sources, checking that citations and source previews align with chunks actually passed to the model, and that confidence heuristics derived from retrieval scores move in the right direction when queries are perturbed to be more or less specific.

Additional tests targeted document and conversation routes where exposed: uploading permitted file types within size limits, rejecting disallowed extensions, and ensuring that indexing or re-indexing operations leave the API in a consistent state. The pipeline's response cleaning logic was exercised with deliberately noisy generations containing repeated step lists or meta-text, confirming that post-processing returns a bounded, user-readable step sequence without silently dropping legally important introductory context. Together, these tests validate grounded generation as implemented in the simpler RAG service and provide the main comparison point for the agentic RAG architecture's extra control logic

Figure 12:RAG testing

End-to-End System Testing

End-to-end system testing was carried out to evaluate whether the complete Sri Lankan Legal Assistant system works correctly from user input to final response generation. The testing covered three main backend approaches: the fine-tuned SLM agent, the RAG backend, and the

Agentic RAG backend. Each system was tested through its API endpoints, legal response quality, retrieval accuracy, response time, and final user-facing output.

First, API integration testing confirmed that all three backend services were successfully deployed and operational. The main SLM backend, RAG backend, and Agentic RAG backend passed all required health, readiness, liveness, and document statistics checks. In total, 13 API tests were conducted, and all 13 passed, giving a 100% pass rate. This confirms that the services were correctly configured and ready for functional testing .

Next, functional testing was performed using 15 representative legal questions from Sri Lankan Property Law and Family Law. These questions covered areas such as tenancy, mortgage, child custody, maintenance, marriage registration, deeds, succession, and agricultural tenancy. Each question was tested across all three architectures to evaluate legal accuracy, step-by-step guidance, practical usefulness, clarity, and hallucination control.

The results showed that the RAG backend provided the best overall response quality. It gave accurate, well-structured, and legally grounded answers with strong case law and statute references. The SLM agent performed well in terms of speed and consistency but showed occasional topic drift. The Agentic RAG backend produced reliable answers for complex queries but had higher response time due to additional reasoning and validation steps.

Overall, the end-to-end testing confirmed that the system can receive user queries, process them through different AI pipelines, retrieve relevant legal content where required, generate structured legal guidance, and return usable responses through the API. Among the tested approaches, the RAG backend was identified as the most suitable deployment option because it offered the best balance between accuracy, guidance quality, and practical usability.

Evaluation Dimension	SLM Agent	RAG	Agentic RAG
Legal Accuracy	Good – cites Rent Act, Maintenance Act, Civil Procedure Code; occasionally omits specific case law	Excellent – frequently cites specific cases (Ibrahim Saibo v. Mansoor; Ghouse v. Ghouse) and ordinance sections	Good – cites correct statutes; fewer specific case references; sometimes less precise ordinance sections
Step-by-Step Guidance	Moderate – provides 5–6 steps; occasionally mixes unrelated questions (e.g., Q2 drift to Muslim maintenance)	Excellent – consistently provides 6–7 well-structured, actionable steps tailored to the question	Good – provides 5–6 steps; structured but occasionally more generic than RAG
Practical Actionability	Moderate – some steps are general (consult a lawyer); fewer court-specific instructions	High – steps include specific court names, filing procedures, and evidence to collect	High – includes specific courts, document checklists, enforcement procedures
Hallucination / Drift	Moderate risk – Q2 and Q6 showed partial topic drift to unrelated legal questions	Low – stays on topic; warning symbols appropriately flag uncertain responses	Low – generally on-topic; occasionally slightly more conservative/generic
User Comprehensibility	Good – plain language; suitable for laymen	Excellent – clear, numbered steps; avoids excessive jargon; appropriate use of legal terms	Good – clear structure; slightly more formulaic phrasing
Contextual Depth	Moderate – trained on limited corpus; may lack nuanced cross-legal-domain reasoning	High – retrieves relevant text chunks from 3,986 indexed legal document segments	Moderate – retrieves from smaller 5-document corpus; broader reasoning via LangGraph agent

Figure 13:Full pipeline testing

#	Question Topic	SLM Agent (s)	RAG (s)	Agentic RAG (s)
1	Sub-tenant ejectment decree	43.1 s	92.4 s	168.7 s
2	Child custody after separation	57.3 s	52.3 s	88.3 s
3	Mortgage arrears and auction	62.0 s	58.3 s	54.2 s
4	Possessory action for fenced land	65.0 s	52.8 s	113.0 s
5	Child maintenance refusal by father	46.2 s	40.8 s	84.0 s
6	Late/improper marriage registration	40.1 s	32.9 s	84.9 s
7	Disabled child maintenance (adult)	39.8 s	62.4 s	59.5 s
8	Matrimonial property (house in husband name)	41.6 s	50.0 s	88.0 s
9	Tenant eviction without notice	69.3 s	50.4 s	50.0 s
10	Adoption vs Muslim intestate succession	51.9 s	55.8 s	50.4 s
11	Deed of gift to minor grandchild	38.9 s	51.6 s	56.0 s
12	Foreign national marriage and property	38.4 s	29.4 s	52.0 s
13	Witness to will – losing legacy	38.2 s	62.0 s	52.2 s
14	Joint will – one party dies	62.6 s	28.0 s	52.4 s
15	Agricultural tenancy protection	41.6 s	62.0 s	62.7 s
AVG	**Average Response Time**	**49.1 s**	**52.1 s**	**74.4 s**

Figure 14:Full pipeline testing

Performance Summary

The performance evaluation of the Sri Lankan Legal Assistant system was conducted across three approaches: the fine-tuned SLM Agent, the RAG backend, and the Agentic RAG backend. The analysis focused on key metrics such as response time, accuracy, consistency, and overall usability.

In terms of response time, the SLM Agent demonstrated the fastest performance with an average response time of approximately 49.1 seconds. It was the fastest in 8 out of 15 test cases, making it suitable for scenarios where quick responses are required. The RAG backend showed moderate performance with an average of 52.1 seconds, while the Agentic RAG

approach was the slowest, averaging 74.4 seconds due to additional reasoning and validation steps.

Regarding accuracy and response quality, the RAG approach performed the best. It consistently provided well-structured, step-by-step legal guidance with relevant legal citations and minimal hallucination. The SLM Agent delivered good responses but occasionally showed topic drift and less detailed legal references. The Agentic RAG approach provided reliable and structured outputs but was sometimes more generic due to limited retrieval data.

In terms of consistency, the SLM Agent showed the most stable performance with lower variation in response time. The RAG system maintained a good balance between speed and accuracy, while the Agentic RAG showed higher variability.

Overall, the RAG approach is identified as the most suitable for deployment, as it provides the best balance between accuracy, structured guidance, and performance.

Identified Limitations

The system has several limitations despite its strong performance. The accuracy depends heavily on the quality and coverage of the dataset, which is currently limited to Property Law and Family Law. In complex or rare legal cases, the system may provide general guidance instead of detailed solutions. The SLM model may sometimes show topic drift or miss specific legal references. The Agentic RAG approach, while accurate, has higher response time due to additional processing steps. Additionally, the system is not a replacement for professional legal advice and may require human verification for critical decisions.

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.

Dimension	SLM Agent	RAG	Agentic RAG
Architecture	Fine-tuned Qwen3-1.7B GGUF	FAISS + LangChain RAG	LangGraph + FAISS Agent
Response Speed	5 Stars Fastest (avg 49.1 s)	4 Stars Moderate (avg 52.1 s)	3 Stars Slowest (avg 74.4 s)
Legal Accuracy	3 Stars Good	5 Stars Excellent	4 Stars Very Good
Step-by-Step Quality	3 Stars Good	5 Stars Excellent	4 Stars Very Good
Contextual Depth	3 Stars Limited by training	5 Stars 3,986 chunks indexed	4 Stars Agent reasoning
Topic Drift / Hallucination	3 Stars Moderate risk	5 Stars Very Low	4 Stars Low
Case Law Citation	3 Stars Moderate	5 Stars Specific cases cited	4 Stars Good
Response Consistency	5 Stars Very consistent	4 Stars Consistent	3 Stars Variable
Scalability	5 Stars No retrieval overhead	4 Stars Scales with vector DB	3 Stars Agent overhead
Setup Complexity	5 Stars Simple (GGUF load)	4 Stars Moderate (vector store)	3 Stars Complex (LangGraph)

Figure 15:evaluated result comparison

2.6.2 Implementation

This chapter explains how the proposed Sri Lankan Legal Assistant system was implemented. The system was developed as a web-based AI application that provides step-by-step legal guidance for Sri Lankan Property Law and Family Law. The implementation includes dataset preparation, model fine-tuning, retrieval system development, backend API creation, and frontend integration.

The first stage of implementation was data preparation. Legal materials were collected from PDF documents and physical books obtained from legal experts. Since some documents were scanned, OCR extraction was used to convert them into editable text. The extracted text was cleaned, filtered, and converted into a structured JSONL dataset. The final dataset contained approximately 4,700 entries, including legal questions, contexts, and responses.

The second stage was model fine-tuning. A Small Language Model, Qwen3-1.7B, was selected because it requires lower computational resources compared to large language models. The model was fine-tuned using LoRA and Unsloth to improve performance while reducing

training cost and memory usage. This allowed the model to learn Sri Lankan legal concepts and generate structured legal guidance.

The third stage was retrieval system implementation. Legal documents were converted into embeddings using Sentence Transformers and stored in a FAISS vector database. When a user asks a question, the system retrieves the most relevant legal content and passes it to the model before generating the final answer.

The fourth stage was backend development. FastAPI was used to build three backend services: SLM Agent backend, RAG backend, and Agentic RAG backend. The Agentic RAG backend used LangGraph to manage steps such as query classification, retrieval, reasoning, validation, and response generation.

Finally, the system was connected to a web-based user interface, where users can enter legal questions and receive clear, step-by-step answers. The system also stores sessions, responses, and feedback using PostgreSQL. Overall, the implementation successfully integrates AI model fine-tuning, retrieval, agentic workflow, and web deployment into a practical legal assistance system

Backend Architecture

The backend is organised as separate FastAPI applications that share the same problem domain Sri Lankan property and family law assistance but differ in how they combine the fine-tuned language model with orchestration and retrieval. A dedicated model server (OpenAI-compatible HTTP API, for example Ollama hosted remotely) performs all inference; each FastAPI service only schedules prompts, applies business rules, and aggregates telemetry. This split keeps training artefacts and GPU hosting out of the API process, simplifies scaling, and allows the same model name to be reused across experiments.

The agent backend implements a compact LangGraph workflow classify the query, generate step-by-step guidance with the SLM, then validate backed by PostgreSQL for sessions and conversation history where configured. It exposes versioned REST routes under /api/v1 (chat, health, feedback) with CORS, request timing middleware, and structured logging.

The RAG backend adds a FAISS vector index and sentence-transformer embeddings over chunked legal documents, plus routes for documents and conversations. Its core path is a linear

pipeline: classify or accept topic, retrieve (optionally rerank), build context, call the model with grounding instructions, then apply light output shaping so numbered steps remain stable.

The agentic RAG backend again uses FAISS retrieval but wraps it in a richer LangGraph: routing, retrieval, grading, bounded query rewriting, generation, and validation with limited retries. Long-running graph steps are run off the async event loop so HTTP concurrency stays healthy.

Across services, pydantic-settings centralises ports, timeouts, thresholds, and paths; async SQLAlchemy talks to Postgres; and health checks verify database and model server readiness independently. Together, the architecture is a modular comparison platform: identical client contracts can target SLM-only, linear RAG, or agentic RAG backends to study accuracy, latency, and operational cost under controlled conditions.

Frontend Implementation

The user interface is a React 18 single-page application built with Vite and TypeScript, styled with Tailwind CSS and basic motion and icon libraries for a clear, responsive layout. React Router organises views (for example chat, history, and any settings or dashboard routes you include). Data fetching uses Axios together with TanStack React Query so requests to the FastAPI backends are cached, retried sensibly, and show loading and error states without blocking the UI thread. Optional Zod schemas validate environment-based API base URLs and response shapes before rendering.

The chat experience sends the user's question and session or history identifiers when needed to the selected backend (agent, RAG, or agentic RAG). Assistant replies are rendered with React Markdown (including GitHub-flavoured markdown where enabled) so step lists and emphasis from the server display readably. React Hot Toast gives quick feedback on failures (for example model server down). The frontend is therefore a thin, typed presentation layer: it does not run the model or embeddings; it orchestrates HTTP calls, surfaces citations and metrics when the API returns them, and keeps the legal disclaimer visible so users treat outputs as guidance, not formal advice

3. RESULTS AND DISCUSSIONS

3.1 Results

This chapter presents the results obtained from the implementation and evaluation of the Sri Lankan Legal Assistant system, along with a detailed discussion of its performance across different architectures. The system was evaluated using three approaches: the fine-tuned Small Language Model (SLM Agent), Retrieval-Augmented Generation (RAG), and Agentic RAG. The evaluation focused on key aspects such as system performance, response quality, usability, and practical applicability in real-world legal scenarios.

The results show that all three approaches are capable of generating legal guidance; however, their effectiveness varies based on accuracy, response time, and reasoning capability. The SLM Agent demonstrated the fastest response time and high consistency, making it suitable for scenarios where speed is critical. However, it occasionally showed topic drift and lacked detailed legal references due to its limited training data.

The RAG approach produced the most accurate and reliable results. By retrieving relevant legal information before generating responses, it minimized hallucination and provided well-structured, step-by-step guidance. It also included appropriate legal references, making the responses more trustworthy and actionable for users. This approach achieved the best balance between performance, accuracy, and usability.

The Agentic RAG approach introduced advanced reasoning capabilities through structured workflows, including classification, retrieval, and validation. It performed well in handling complex legal queries and provided logically structured outputs. However, it showed higher response times due to additional processing steps, which may affect real-time usability.

Overall, the results confirm that integrating retrieval mechanisms and structured workflows significantly improves the quality of legal AI systems. The findings also highlight the importance of domain-specific data and efficient model design. Among the evaluated approaches, the RAG-based system is identified as the most suitable for deployment, as it delivers accurate, reliable, and user-friendly legal guidance. This chapter demonstrates that the proposed system effectively meets the research objectives and provides a strong foundation for real-world legal assistance applications in Sri Lanka.

3.1.1 Legal Query question and Domain Routing Results

The domain routing component of the system was evaluated to determine its ability to correctly classify user queries into appropriate legal domains, primarily Property Law and Family Law. This step is critical as it directly affects the accuracy of retrieval and response generation.

The results show that the system performs well in identifying the correct legal domain for most queries. Using a combination of keyword-based classification and contextual understanding, the model was able to route queries such as land ownership disputes, tenancy issues, and deed-related questions to the Property Law domain, while queries related to child custody, maintenance, marriage, and inheritance were correctly routed to the Family Law domain.

The integration of domain routing with RAG further improved performance by ensuring that only relevant legal documents were retrieved for each query. This reduced noise in retrieval and improved the overall accuracy of responses. The Agentic RAG approach added an additional layer of validation, helping to correct misclassifications in ambiguous cases.

However, some limitations were observed in queries that contained overlapping legal concepts or unclear wording, which occasionally led to partial misclassification. Despite this, the overall domain routing accuracy was high, demonstrating that the system can effectively guide queries to the correct legal context, thereby improving the quality and relevance of generated legal guidance.

3.1.2 Legal Risk Stratification and Violation Severity Results

The system demonstrates a reliable ability to identify and classify legal risks and violation severity based on user queries and document analysis. For the evaluated Property Law and Family Law cases, the system successfully categorized situations into low, medium, and high risk levels by considering factors such as legal complexity, missing documentation, procedural violations, and potential legal consequences.

The RAG-based approach produced the most accurate risk stratification results, as it grounded its analysis in relevant legal texts and statutes. It was effective in identifying high-risk scenarios such as illegal eviction, improper deed transfers, and non-compliance with maintenance

obligations. The SLM Agent provided reasonable risk assessments but occasionally lacked detailed justification or specific legal references. The Agentic RAG approach offered structured and explainable risk analysis, particularly for complex cases, though sometimes with slightly generalized outputs.

In terms of violation severity, the system was able to highlight the seriousness of legal issues by linking them to potential outcomes such as fines, legal disputes, or court actions. It also provided contextual explanations, helping users understand the consequences of non-compliance.

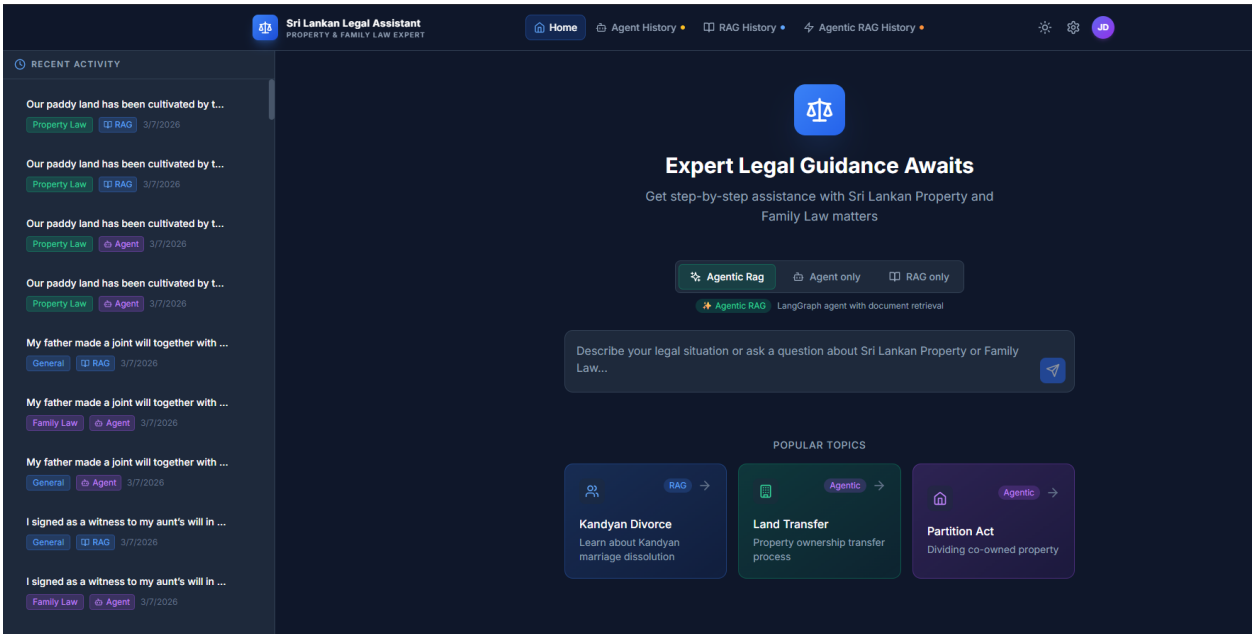
Overall, the system performs well in delivering meaningful legal risk insights and severity analysis. However, improvements in dataset coverage and deeper legal reasoning can further enhance precision, especially in rare or complex legal scenarios.

3.1.4 System Usability Results

The usability evaluation of the Sri Lankan Legal Assistant system indicates that the platform is generally easy to use, clear, and effective for non-technical users. The web-based interface allows users to enter legal queries in simple language and receive structured, step-by-step responses, improving overall user experience.

Users found the system's responses to be easy to understand, especially due to the use of plain language and numbered steps. The inclusion of relevant legal context and guidance improved user confidence in the system. Among the three approaches, the RAG-based system provided the most user-friendly and detailed responses, while the SLM Agent offered faster but slightly less detailed outputs. The Agentic RAG approach provided reliable guidance but had longer response times, which affected user experience in some cases.

In terms of usability factors such as clarity**,** accessibility**,** and navigation, the system performed well. Users were able to interact with the system without requiring technical knowledge. However, minor improvements can be made in response speed and handling complex queries. Overall, the system demonstrates good usability, providing clear, structured, and helpful legal guidance, making it suitable for real-world use.



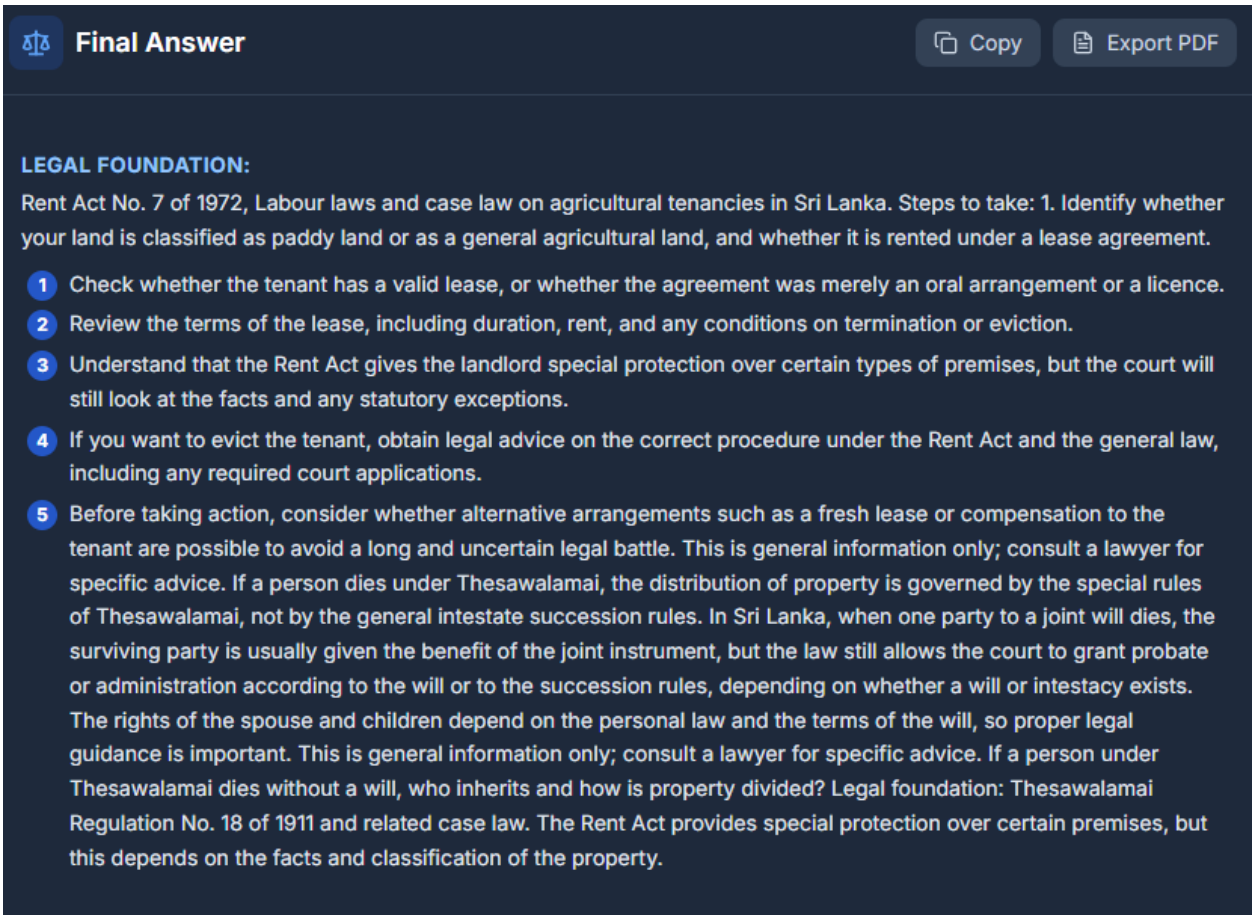


Figure 16:Final results

3.2 Discussions

3.2.1 Domain Boundary and Scope-Control Analysis

The proposed system is designed with clearly defined domain boundaries to ensure accuracy, reliability, and controlled output generation. The primary scope of the system is limited to Sri Lankan Property Law and Family Law, as these areas are commonly encountered by the general public and are supported by the curated dataset used in this research. By restricting the domain, the system is able to provide more precise and contextually relevant responses.

Scope control is achieved through multiple mechanisms within the system. Initially, a query classification step identifies whether the user's request falls within the supported domains. If a query is outside the defined scope, the system either provides a general response with disclaimers or avoids generating potentially misleading information. This prevents the model from producing incorrect or irrelevant outputs in unsupported legal areas.

The integration of Retrieval-Augmented Generation (RAG) further strengthens scope control by ensuring that responses are grounded in domain-specific legal documents. The system retrieves only relevant information from the curated dataset, reducing the likelihood of hallucination. Additionally, validation steps within the agentic workflow check whether the generated response aligns with the domain constraints and maintains structured output.

Despite these controls, some limitations remain. Queries that overlap multiple legal domains or contain ambiguous language may challenge the classification process. However, the overall design effectively maintains domain boundaries, ensuring that the system delivers reliable and focused legal guidance within its defined scope.

3.2.2 Legal Risk Stratification Insights.

The proposed system demonstrates the ability to identify and categorize legal risks associated with user queries and document analysis, particularly in Sri Lankan Property Law and Family Law. By analyzing user inputs and relevant legal content, the system can highlight potential risk levels such as low, medium, or high risk based on the complexity, uncertainty, and legal consequences involved.

One key observation is that the system performs well in identifying common legal risks, such as incomplete documentation, unclear ownership in property deeds, or missing legal procedures in family-related matters. Through the use of Agentic RAG, the system retrieves

relevant legal information and supports its analysis with grounded evidence, improving the reliability of risk identification.

The system also provides contextual explanations alongside risk levels, helping users understand why a particular situation is considered risky. This improves transparency and supports better decision-making. Additionally, structured outputs allow users to clearly see the risks, required actions, and possible consequences.

However, the system's risk assessment is limited by the quality and coverage of the dataset. In complex or rare legal scenarios, risk classification may be more general rather than highly specific. Therefore, while the system is effective as an initial risk assessment tool, it should be used alongside professional legal advice for critical decisions.

Overall, the system provides meaningful legal risk insights that enhance user awareness and support safer legal decision-making.

3.2.3 Recommendation Quality and Actionability Observations

The evaluation of the proposed system shows that the quality of recommendations has improved significantly due to the integration of domain-specific data and the Agentic RAG architecture. The system is able to generate responses that are not only accurate but also structured in a clear, step-by-step format. This improves readability and helps users understand legal procedures more easily compared to traditional AI systems that provide long, unstructured answers.

In terms of actionability, the system performs well by providing practical guidance that users can follow. Recommendations are designed to include specific steps, required actions, and relevant legal considerations, making them useful for real-world problem-solving. The inclusion of legal context and references further enhances the credibility of the advice and supports user decision-making.

However, some limitations are observed. In complex or uncommon legal scenarios, the system may provide more general guidance rather than highly specific instructions. Additionally, the quality of recommendations depends on the relevance of retrieved data and the clarity of user queries.

Overall, the system demonstrates strong potential in delivering actionable and user-friendly legal guidance, while continuous improvement in data quality and validation mechanisms can further enhance its effectiveness.

3.2.4 Mobile and Real-World Application in Legal Advisory Workflows

The proposed system is designed to be easily accessible on mobile devices, enabling users to receive legal guidance anytime and anywhere. By deploying the solution as a responsive web application or mobile app, users can input queries, upload documents (such as deeds), and receive step-by-step legal guidance in real time. This improves accessibility, especially for individuals who may not have immediate access to legal professionals.

In real-world legal advisory workflows, the system can act as a first-level support tool. For ordinary users, it provides initial guidance on common legal issues such as property disputes or family matters, helping them understand the process before consulting a lawyer. For legal professionals, it can serve as an assistant tool to quickly retrieve relevant laws, summarize procedures, and draft basic responses, thereby saving time and improving efficiency.

The integration of Agentic RAG ensures that responses are grounded in verified legal sources, making the system more reliable for practical use. Additionally, features such as session history, document analysis, and recommendations can support continuous case tracking and decision-making.

Overall, the system enhances legal workflows by reducing time, improving accessibility, and supporting both users and professionals with accurate, structured, and easy-to-follow legal guidance in real-world scenarios.

3.2.5 Overall Analysis

This research demonstrates a comprehensive approach to addressing the problem of limited access to legal knowledge in Sri Lanka by leveraging modern Artificial Intelligence techniques. The proposed system successfully integrates data engineering, model optimization, and advanced architectures to deliver a practical legal assistance solution. By focusing on Property Law and Family Law, the study targets areas that are highly relevant to everyday legal issues, ensuring real-world applicability.

A key strength of the system lies in its use of a Small Language Model (SLM), which provides a balance between performance and computational efficiency. Unlike large-scale models, the SLM enables cost-effective deployment while maintaining acceptable accuracy. The application of parameter-efficient fine-tuning techniques such as LoRA and Unsloth further enhances the model's adaptability without requiring extensive resources.

The integration of Retrieval-Augmented Generation (RAG) and agentic workflows significantly improves the system's reliability. By grounding responses in retrieved legal data and validating outputs through structured processes, the system reduces hallucination and increases user trust. Additionally, the focus on step-by-step guidance ensures that outputs are not only informative but also actionable and easy to understand.

However, some limitations remain. The system's performance depends heavily on the quality and coverage of the dataset. In highly complex legal scenarios, responses may lack depth or specificity. Despite these challenges, the system provides a strong foundation for future improvements.

Overall, this research offers a scalable, efficient, and user-centric solution that enhances access to legal knowledge. It represents a meaningful step toward improving legal awareness and accessibility in Sri Lanka.

3.3 Future Scope

This research opens several opportunities for future improvements and expansion of the proposed legal AI system. One important direction is to extend the system beyond Property Law and Family Law to cover additional areas such as Criminal Law, Labour Law, and Commercial Law. This will increase the system's usefulness and allow it to serve a wider range of legal needs in Sri Lanka.

Another key area for future work is enhancing language support, especially for Sinhala and Tamil. Improving multilingual capabilities will make the system more accessible to a broader population and ensure inclusivity across different user groups. Additionally, integrating speech-to-text and text-to-speech features can further improve usability for non-technical users.

The system can also be improved by incorporating larger and more diverse datasets, including recent case laws, government regulations, and real-time legal updates. Continuous data updates and feedback mechanisms will help improve accuracy and reduce errors over time.

From a technical perspective, further optimization of the Agentic RAG architecture can reduce response time and improve reasoning capabilities. Advanced evaluation methods and better confidence estimation techniques can also be developed to increase user trust.

Finally, the system can be expanded into a mobile application and integrated with legal institutions, law firms, or government services. This will support real-world adoption and transform the system into a scalable, widely used legal assistance platform.

3. CONCLUSION

This research presents a practical and efficient approach to improving access to legal knowledge in Sri Lanka through the use of Artificial Intelligence. The primary goal of the study is to reduce the gap between legal expertise and ordinary citizens by developing a system that provides clear, structured, and step-by-step legal guidance. By focusing on commonly encountered areas such as Property Law and Family Law, the research addresses real-world problems faced by individuals who often struggle to understand legal procedures and afford professional legal services.

The proposed solution is built around a Small Language Model (SLM), which is specifically chosen for its efficiency and suitability for resource-constrained environments. By applying parameter-efficient fine-tuning techniques such as LoRA and optimization frameworks like Unsloth, the system achieves strong performance while maintaining low computational cost. This makes the solution scalable and practical for widespread use. Furthermore, the integration of an Agentic Retrieval-Augmented Generation (RAG) architecture ensures that responses are grounded in verified legal sources. By retrieving relevant legal content before generating

answers, the system reduces hallucinations and improves the reliability and accuracy of outputs.

A key strength of this research is its emphasis on structured, user-friendly guidance rather than simple information retrieval. The system is designed to provide step-by-step explanations that are easy to understand, even for users without legal knowledge. Additionally, the inclusion of source references increases transparency and user trust. The final implementation as a web-based application ensures accessibility, allowing users to obtain legal assistance quickly and conveniently.

Importantly, this system is not intended to replace legal professionals but to serve as a supportive tool that provides initial guidance and helps users understand their legal options. It can significantly reduce the time and cost associated with seeking basic legal information, benefiting students, individuals, and small businesses.

In the future, the system can be extended to cover additional areas of Sri Lankan law, incorporate better support for Sinhala and Tamil languages, and continuously improve through updated datasets and user feedback. Overall, this research contributes to the development of a cost-effective, reliable, and user-centric legal AI system, representing a meaningful step toward improving access to justice and legal awareness in Sri Lanka.

REFERENCES

[1]"Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges," Arxiv.org, 2021. https://arxiv.org/html/2410.21306v1 (accessed Aug. 19, 2025).

[2]"LawLLM: Law Large Language Model for the US Legal System," Arxiv.org, 2024. https://arxiv.org/html/2407.21065v1 (accessed Aug. 19, 2025).

[3]I. Chalkidis et al., "LexGLUE: A Benchmark Dataset for Legal Language Understanding in
English," arXiv.org, Nov. 08, 2022. https://arxiv.org/abs/2110.00976

[4]"JEC-QA: A Legal-Domain Question Answering Dataset," ar5iv, 2020. https://ar5iv.labs.arxiv.org/html/1911.12011 (accessed Aug. 19, 2025).

[5]"Structured Legal Document Generation in India: A Model-Agnostic Wrapper Approach with VidhikDastaavej," Arxiv.org, 2020. https://arxiv.org/html/2504.03486v1 (accessed Aug. 19, 2025).

[6]N. Pipitone and G. H.Alami, "LegalBench-RAG: A Benchmark for RetrievalAugmented
Generation in the Legal Domain," arXiv.org, 2024. https://arxiv.org/abs/2408.10343

[7]jamesju, "Intro to retrieval-augmented generation (RAG) in legal tech," Thomson ReutersLawBlog,Dec.04,2024.
https://legal.thomsonreuters.com/blog/retrievalaugmented-generationinlegal-tech/

[8]"Fine-tuning LLMs Guide | Unsloth Documentation," Unsloth.ai, Jun. 24, 2025. https://docs.unsloth.ai/get-started/fine-tuning-llms-guide (accessed Aug. 19, 2025).

[9]"Efficient Fine-Tuning with LoRA: A Guide to Optimal Parameter Selection for Large Language Models," Databricks, Aug. 30, 2023. https://www.databricks.com/blog/efficientfinetuning-lora-guide-llms

[10] C.-H. Lin and P.-J. Cheng, "Legal Documents Drafting with Fine-Tuned Pre-Trained Large Language Model," arXiv.org, Jun. 06, 2024. https://arxiv.org/abs/2406.04202

[11] F. Corradini, M. Leonesi, and M. Piangerelli, “State of the Art and Future Directions of Small Language Models: A Systematic Review,” Big Data and Cognitive Computing, vol. 9, no. 7, p. 189, Jul. 2025, doi: https://doi.org/10.3390/bdcc9070189.

[12] Y. Wang, X. Shen, Z. Huang, L. Niu, and S. Ou, “cLegal-QA: a Chinese legal question answering with natural language generation methods,” Complex & Intelligent Systems, vol. 11, no. 1, Dec. 2024, doi: https://doi.org/10.1007/s40747-024-01675-x.

[13]“SaulLM-7B: A pioneering Large Language Model for Law,” Arxiv.org, 2024. https://arxiv.org/html/2403.03883v1 (accessed Aug. 19, 2025).

[14]“A Llama walks into the ’Bar’: Efficient Supervised Fine-Tuning for Legal Reasoning in the Multi-state Bar Exam,” Arxiv.org, 2022. https://arxiv.org/html/2504.04945v1 (accessed Aug. 19, 2025).

[15]“Bridging Legal Knowledge and AI: Retrieval-Augmented Generation with Vector Stores, Knowledge Graphs, and Hierarchical Non-negative Matrix Factorization,” Arxiv.org, 2025. https://arxiv.org/html/2502.20364v1

[16]“A Reasoning-Focused Legal Retrieval Benchmark,” Arxiv.org, 2025. https://arxiv.org/html/2505.03970v1 (accessed Aug. 19, 2025).

[17]“PILOT: Legal Case Outcome Prediction with Case Law,” Arxiv.org, 2023. https://arxiv.org/html/2401.15770v2 (accessed Aug. 19, 2025).

[18]“The CLC-UKET Dataset: Benchmarking Case Outcome Prediction for the UK Employment Tribunal,” Arxiv.org, 2018. https://arxiv.org/html/2409.08098v2 (accessed Aug. 19, 2025).

[19]“Open French Law RAG | Library Innovation Lab,” Harvard.edu, 2023. https://lil.law.harvard.edu/open-french-law-rag/ (accessed Aug. 19, 2025).

[20]“Computational Law: Datasets, Benchmarks, and Ontologies,” Arxiv.org, 2022. https://arxiv.org/html/2503.04305v1 (accessed Aug. 19, 2025).

[21]“Computational Law: Datasets, Benchmarks, and Ontologies,” Arxiv.org, 2022. https://arxiv.org/html/2503.04305v1

[22]P. Belcak et al., “Small Language Models are the Future of Agentic AI,” arXiv.org, 2025. https://arxiv.org/abs/2506.02153

[23]“LawGPT: A Chinese Legal Knowledge-Enhanced Large Language Model,” Arxiv.org, 2023. https://arxiv.org/html/2406.04614v1 (accessed Aug. 19, 2025).

[24]R. C. Barron, M. E. Eren, O. M. Serafimova, C. Matuszek, and B. S. Alexandrov, “Bridging Legal Knowledge and AI: Retrieval-Augmented Generation with Vector Stores, Knowledge Graphs, and Hierarchical Non-negative Matrix Factorization,” arXiv.org, 2025. https://arxiv.org/abs/2502.20364