SMALL LANGUAGE MODELS FOR SRI LANKAN LEGAL APPLICATIONS

25-26J-240

Project Proposal Report

Mathusigan senthan – IT22177032
Thuvaraga Anantharajah -IT22030412
Abiramy Thiyakesan - IT22049322
Niruththika Erambanathan- IT22322326

Supervisor Name : DR.Prasanna Sumathipala

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

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2025

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.

25-26J-240

Project Proposal Report

Mathusigan senthan – IT22177032

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

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2025

DECLARATION

We declare that this is our own work, and this proposal does not incorporate without acknowledgement 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 acknowledgement is made in the text.

Name	Student ID	Signature
Mathusigan.E.s	IT22177032	

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

Name of supervisor: DR.Prasanna Sumathipala

Signature of the supervisor 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 (RAG) approach will be 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

Small Language Model (SLM) , Artificial Intelligence (AI) , Retrieval-Augmented Generation (RAG)

Table of Contents

DECLARATION....3
ABSTRACT....i
LIST OF FIGURES....iii
LIST OF TABLES....iii
LIST OF ABBREVIATIONS....iv
1.INTRODUCTION....1
1.1 Background and Literature Review....2
1.2.Research Gap....7
1.3.Research Problem....11
2. OBJECTIVES....12
2.1 Main Objectives....12
2.2 Specific Objectives....12
3.METHODOLOGY....13
3.1.Data Collection and Preparation....13
3.2. Model Selection and fine tuning....14
3.3.System Architecture and Workflow Layers....15
3.4.Evaluation and Comparison....16
3.5 .Tools and Technologies....17
3.6.Implementation&Deployment....18
4. PROJECT REQUIREMENTS....19
4.1 Functional Requirements....19
4.2 Non-Functional Requirements....19
4.3. Expected Test Cases....20
4.4.Feasibility Study....21
5. SYSTEM DIAGRAM AND GANTT CHART....22
5.1 Technical Daigram....24

5.2 Gannt chart 26
6. CONCLUSION 27
7. BUDGET AND BUDGET JUSTIFICATION 28
REFERENCES 29

LIST OF FIGURES

Figure 1:System Diagram for Relevance and Abuse Detection 22
Figure 2:Technical diagram 24
Figure 3:Gannt Chart 26

LIST OF TABLES

Table 1:List of Abbreviations iv
Table 2:Slm Focused Researches 4
Table 3:RAG & AGENTS FOCUS PAPERS 5
Table 4:Llm Focused Papers 6
Table 5:Local Evaluation Methods 10

LIST OF ABBREVIATIONS

Key Words	Meaning
SLM	Small Language model
API	Application Programming Interface
NLP	Natural Language Processing
AI	Artificial Intelligence

Table 1:List of Abbreviations

1.INTRODUCTION

Legal field holds paramount significance to deliver justice, equity, and direction to human beings all over the world. Nevertheless, legal knowledge is generally limited to lawyers and legal experts. Therefore, citizens dealing with even slight legal matters are left with no option other than taking professional advice, for which they have to pay heavily and spend ample time. To fill this gap, the world is in urgent need of a system that can deliver legal knowledge more accessibly, economically, and credibly to the masses.

Owing to the latest advances in Artificial Intelligence (AI), it is now possible to design such systems that can provide legal assistance in consumer-friendly formats. Dev teams can build cost-effective solutions that provide correct, stepwise legal advice through the use of Small Language Models (SLMs). It not only reduces costs, it also facilitates practical implementation in resource-scarce settings.

Its scope is limited to the legal field of Sri Lanka, although it has considered four principal areas:

1.Fine-tuning of SLM to give end-to-end stepwise advice to solve user issues in Sri Lankan Property Law and Family Law ,

2 Fine-tuning an SLM for risk reviewing and analyzing in Sri Lankan deed documents

3. Designing a recommending system for users, taking particular focus on labour and employment law.

4.Estimate likely judicial rulings by reviewing case histories to allow users to see likely results and precedents.

As part of that larger agenda, my research project is to design an SLM that can give stepwise, end-to-end advice in Sri Lankan Property Law and Family Law. While doing so, it seeks to demystify legal procedures and make them easily understandable to ordinary

citizens, reducing reliance upon expensive legal advice while fostering access and equity to all of justice.

1.1 Background and Literature Review

Law is valuable for maintaining justice and fairness in society, yet many nations experience issues with common people having poor access to legal knowledge. In Sri Lanka, much knowledge of property law and family law is primarily in books or with lawyers, so individuals need to spend money and time to receive advice. With advances in Artificial Intelligence (AI), new research demonstrates that language models can assist legal tasks at little cost. Many works describe applying NLP and legal datasets to train models for law comprehension [1], and others even constructed legalspecific LLMs like LawLLM for the legal system of the United States [2]. Performance of models is tested with measures like LexGLUE for legal text classification and case examination [3]. Other works concern legal question answering, for instance JEC-QA for China [4] or Indian works for legal document generation with VidhikDastaavej [5]. Retrieval-Augmented Generation (RAG) techniques also gained fame, where legal document retrieval is combined with generation to elevate precision [6][7]. For model fine-tuning, practical frameworks like Unsloth [8] and lightweight techniques like LoRA [9] are now ready, and they enable smaller models to achieve excellence at domain-specific tasks. Moreover, some works experimented on legal document drafting by fine-tuning pre-trained models [10] and also systematic reviews indicate how Small Language Models (SLMs) become increasingly valuable due to the fact that they require less compute and deliver good results in the vast majority of cases [11]. For legal datasets, Chinese cLegal-QA offers real user queries and replies [12] while SaulLM-7B is large model constructed solely for law with billions of tokens [13]. More targeted research fine-tuned Llama models for the United States. Bar Exam with supervised learning [14]. Other newer works also integrate RAG with knowledge graphs for enhanced legal reasoning [15], or build reasoning-oriented retrieval benchmarks like Bar Exam QA and Housing Statute QA [16]. Case prediction of outcomes is another research area, with models such as PILOT utilizing former case law to predict judgment [17] and the CLC-UKET dataset offering outcome prediction for UK Employment Tribunal [18]. Other projects also permit cross-language access like the Open French Law RAG, to access legal texts written in another language [19]. Lastly, computational law surveys emphasize the role of datasets, ontologies, and benchmarks for legal AI system training [20][21].More current works also emphasize agentic execution in AI systems. Belcak et al. describe how Small Language Models are particularly optimal for agentic AI as they are less expensive, quicker, yet powerful enough for non-general tasks [22]. In similar works, like LawGPT, legalknowledge augmented models can respond to law-related questions more efficiently [23]. Moreover, Barron et al. propose agentic RAG techniques integrated with vector stores as well as

knowledge graphs for enhanced legal reasoning and fewer legal text processing errors [24]. From all these results, it is clearly evident that agentic architecture can indeed enhance precision as well as expedite processing time that is quite beneficial for legal advisory systems. In this project, I also hope to evaluate time-reducing agentic architecture to enable users to receive quick and accurate answers for Sri Lankan Property Law and Family Law.

SLM FOCUSED RESERCHS

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 2:Slm Focused Researches

RAG & AGENTS FOCUS PAPERS

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 3:RAG & AGENTS FOCUS PAPERS

LLM FOCUSED PAPERS

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 4:Llm Focused Papers

1.2.Research Gap

Access to legal information is inconsistent between countries. In many countries—Sri Lanka among them—laws reside in books, subscription sites, or scans rather than clean, machine-readable text. Legal NLP research at the global level demonstrates robust results for legal Q&A, classification, retrieval, drafting, and judgment prediction tasks [1][2][3][4][10][11]. However, much of it is limited to the U.S., Europe, and China. LexGLUE [3] dataset, for instance, JEC-QA [4] dataset, and cLegal-QA dataset [12] represent foreign laws, legal style, and precedents. Likewise, legal models such as LawLLM [2], SaulLM-7B [13] model, and barexam–tuned models [14] were never trained from Sri Lankan Property or Family Law. A transparent jurisdictional gap exists: no open, Sri Lanka-oriented corpus and test set that reflect local doctrine, procedures, and deed practice.

A second gap is concerning data access and structure. Authoritative sources of Sri Lankan property and family law are typically printed books, scanned PDFs, or dispersed web pages. These are cumbersome to index and noisy for ML. Previous work in computational law argues for curated datasets, ontologies, and top-notch annotations [20][21]. Sri Lanka lacks to date a common, digitized corpus with citation metadata, section delineations, and provenance. Without such foundation, tasks such as retrieval, grounding, and stepwise guidance are brittle and difficult to reproduce.

Third is resources and cost. Most prominent legal models are extensive and costly to maintain. That's unsustainable for a public-facing service in Sri Lanka. Recent research concludes that Small Language Models (SLMs) become economical for agentic use by virtue of their reduced cost and velocity, but they are successful only for concentrated tasks [22]. Unsloth-like tooling and parameter-efficient techniques such as LoRA reduce compute and cost and boost domain adaptation [8][9]. Nevertheless, despite searching doggedly, we could find no research directly that applies SLMs with PEFT to Sri Lankan needs. The compute-aware gap remains: no demonstrated accurate and cheap pattern for Sri Lankan property and family law.

Fourth is retrieval and speed. RAG is common to anchor answers in source law and minimize hallucinations [6][7][15]. More recent concepts shift to agentic RAG, in which agents plan, reflect, select tools, and hone evidence for superior factuality and reasoning [16][24]. Special-domain projects such as Open French Law RAG demonstrate how codified law can be searchable and valuable [19]. Nevertheless, we have not yet witnessed

an agentic RAG pipe customized for Sri Lankan law, optimized for quality and fast response. In use cases, users often require quick, step-wise advice for day-to-day tasks (e.g., steps for deeds, filing protocols), rather than extended essays. Executing latency-aware, time-reducing plans—selecting shallow versus deep retrieval, varying chunk sizes, caching frequent queries, and pruning tool invocations—has not been experimented upon here.

Local evaluation and safety is a fifth gap. Available benchmarks (e.g., LexGLUE; reasoncentric QA such as Bar Exam or Housing Statute QA) quantify progress elsewhere [3][16], but they do not fill in for Sri Lankan statute, form, and real-user questions. Legalknowledge–augmented models (e.g., LawGPT) demonstrate higher grounding [23], yet Sri Lanka does not possess comparable metrics: grounded cite rate, statute coverage, and bilingual/multilingual accuracy over Sinhala, Tamil, and English. Since local test suites and guardrails do not exist, it is not possible to quantify reliability or to evaluate SLM+RAG versus lawyer-authored advice.

Indeed, they are the following:

1. Jurisdiction and data gap: No openly accessible, digitized, adequately annotated Sri Lankan corpus for Property and Family Law.

2.Methodology gap: No application of SLMs with Unsloth/LoRA on Sri Lankan materials to deliver low-cost, step-by-step guidance [8][9][11][22].

3.System gap: Absence of time-aware, agentic RAG design for Sri Lankan legal workflows with local evaluation mechanisms and latency targets [15][16][19][24].

This research is original in that it combines three strands for Sri Lanka: (i) developing a local, high-quality corpus; (ii) applying compute-effective fine-tuning over SLMs; and (iii) creating an agentic, latency-sensitive RAG system. Instructed by legal NLP precedents [1][2][3][4][10][11][13][14][15][16][19][20][21][22][23][24] of prior research, legal and social LLM studies, and agentic RAG research, it seeks to yield novel datasets, practical architecture, and local evaluation procedures of direct utility to property and family law users in.

Solution with my research

The solution that this research gives is a step-by-step plan to make a smart but low-cost legal help system for Sri Lanka.

First, all the law books and documents on Sri Lankan Property Law and Family Law will be scanned and converted into clean digital text. The text will be stored in a structured JSON format so that it can be easily used for training.

After preparing the data, five different small language models (encoder–decoder models) will be fine-tuned using this dataset. Each model will be trained and tested to check which one gives the most clear and correct answers for local legal problems.
From the trained models, the one that performs the best (fast, accurate, and reliable) will be chosen as the main model for the system.

Then, different agentic Retrieval-Augmented Generation (RAG) architectures will be tested. This means the system will not only generate answers but also search and retrieve exact law texts before replying. The best RAG structure that reduces time and gives the most reliable guidance will be selected.

Finally, the chosen small language model and the best agentic RAG setup will be combined into one user-friendly web application. This app will provide step-by-step legal guidance using only trusted Sri Lankan law sources, making the answers both accurate and easy to understand.

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 5:Local Evaluation Methods

1.3.Research Problem

Access to legal information is the foundation of the provision of justice and fairness in every society.

But in Sri Lanka, ordinary citizens typically find it difficult to access legal advice for minor cases due to legal information being mostly locked up in textbooks or only reachable by lawyers. It acts as an access barrier as one has to bear extra costs and time to reach legal experts, thus leading to procrastination and unequal access to justice. From elsewhere, research has shown that Artificial Intelligence (AI) technologies, specifically language models, can provide cheap legal services through answering questions, case prediction, and retrieval-based systems [1][2][3]. Services such as LawLLM in the United States of America or LawGPT in China already existed

developed to facilitate citizens to access law easily. While such developments took place at the global level, no such open digital system is available for property law and family law at Sri Lanka. The data is mostly in book form, PDFs, or spread over web sources, it is not easy for ordinary users to inquire and access.

Popular AI technologies such as ChatGPT sometimes respond incorrectly due to their reliance upon sources of social media and other websites that sometimes publish wrong information. Without an appropriate AI-powered system, citizens are left to rely solely upon the old-fashioned legal consultations. Without an organized AI-powered tool, citizens are again relying solely upon old-fashioned consultations, making it more expensive and discriminatory. If that issue is not addressed, access to justice will be limited and mostly restricted to poor people or rural populations.

Also of concern is the fact that almost all legal AI models that are out today are rather large and resource-hungry, hence not practical for the limited-resource environment of Sri Lanka. While newer works hint at the promise of Small Language Models (SLMs) and lightweight fine-tuning methods such as LoRA or Unsloth [22], none has yet been tried and tested for dealing with Sri Lankan legal matters. Further, little work is available for agentic Retrieval-Augmented Generation (RAG) methods to reduce the response time and increase its reliability for deployment locally [23][24]. Thus, the research question is the non-existence of an economical, Sri Lanka–focussed AI system that has integrated small language models fine-tuned to property and family law, as well as agentic RAG, to deliver step-wise legal advice. It will not only increase access, it will also increase fairness in the legal world of the nation.

2. OBJECTIVES

2.1 Main Objectives.

The main objective of this research is to design and fine-tune a Small Language Model (SLM) that can provide reliable, step-by-step guidance in Sri Lankan Property Law and Family Law. The study aims to make legal knowledge more accessible, affordable, and practical for ordinary citizens who often struggle to find timely legal help. By leveraging efficient fine-tuning methods such as Unsloth and LoRA, and combining them with an agentic Retrieval-Augmented Generation (RAG) architecture, the project seeks to reduce response time, improve accuracy, and deliver a userfriendly web application that bridges the gap between people and legal resources in Sri Lanka.

2.2 Specific Objectives

1. To collect and digitize legal data from Sri Lankan Property Law and Family Law, including textbooks, case records, and online legal resources.
2. To fine-tune multiple small language models using efficient training methods and evaluate their performance.
3. To implement an agentic RAG framework that enhances accuracy and reduces processing time for legal question answering.
4. To build a user-friendly web application that delivers step-by-step legal guidance in local contexts.
5. To compare the performance of the developed SLM-based system with existing largemodel and traditional approaches in terms of accuracy, cost, and usability.
6. To explore the scalability of this approach for other areas of Sri Lankan law.

3.METHODOLOGY

The proposed system will be developed in several stages to ensure both accuracy and practicality. First, legal data related to Sri Lankan Property Law and Family Law will be collected from textbooks, scanned documents, official websites, and digital resources. This data will then be cleaned, structured, and converted into a suitable format for training. Next, small language models will be fine-tuned using the Unsloth framework, since it provides efficient methods for domainspecific training with limited resources. To improve accuracy and reduce response time, an Agentic Retrieval-Augmented Generation (RAG) architecture will be integrated, where retrieval modules fetch the most relevant law texts and agentbased reasoning produces clear step-by-step answers. The final system will be deployed as a user-friendly web application that allows individuals to type their queries and instantly receive legal guidance. Testing and evaluation will be carried out by comparing system responses with expert-verified answers to measure reliability, speed, and usability. The whole approach is designed to make legal help more accessible, affordable, and scalable, with the possibility of expanding into other areas of Sri Lankan law in the future

3.1.Data Collection and Preparation

The purpose of this work is to build a clean and trustworthy Sri Lankan Property Law and Family Law corpus. To achieve this, physical law books will be scanned page by page, and Optical Character Recognition (OCR) will be applied to extract the text. In addition, content from official law websites, gazettes, court summaries, and PDF law guides will be downloaded and converted into text. Along with the extracted text, citation metadata such as the title, author, year, page or section, URL, or publisher will be preserved.

Once collected, the data will undergo cleaning and structuring. OCR errors will be corrected, unnecessary headers and footers will be removed, and punctuation and spelling will be normalised. The content will be organised into clear sections, such as Act, Part, Section, and Schedule, each assigned a unique ID. Language tagging will be applied to distinguish Sinhala, Tamil, and English materials, with consistent transliteration created

where required. The text will then be split into smaller chunks of 300–600 tokens with overlaps to support effective retrieval.

The final structured data will be stored in JSONL format with fields including {id, text, source, section, language, date}. Licences and permissions will be tracked carefully, and any restricted material will be excluded from the corpus.

3.2. Model Selection and fine tuning

The goal is to choose efficient Small Language Models (SLMs) and adapt them to local law. Models will be selected with a size of seven billion parameters or fewer, a permissive licence, low VRAM requirements, a strong multilingual tokenizer, and acceptable baseline latency on the target CPU/GPU hardware.

For training, Unsloth will be used with parameter-efficient fine-tuning methods such as LoRA to reduce compute cost. Supervised datasets will be constructed to include question–answer pairs, step-by-step legal procedures, and deed-review checklists; these will be split into training, validation, and test sets following an 80/10/10 ratio, with fixed random seeds for reproducibility. Hyperparameters—including learning rate, batch size, number of epochs, and LoRA rank—will be tracked closely, and early stopping will be applied where appropriate.

Model quality will be checked with automatic metrics such as accuracy and exact match for procedural steps, as well as citation-match rate; in addition, a small blind set will undergo human review by law students or mentors to assess clarity and legal soundness.

For agentic Retrieval-Augmented Generation (RAG) integration, the objective is to improve factual accuracy while reducing response time. The pipeline will use a retriever that performs vector search over the chunked corpus with BM25 as a fallback, a re-ranker to reorder top passages for better relevance, and a planner agent that classifies the user query as simple or complex to decide between shallow or deep retrieval. Grounded generation will require the SLM to answer only from the retrieved passages while inserting

citations, and an optional self-check or reflection step will perform a quick consistency pass for high-risk questions.

Latency will be controlled by caching frequent queries and embeddings, dynamically adjusting top-k and context length based on query type, and allowing early exit when a confidence threshold is met. Safety measures will include displaying a disclaimer, blocking out-of-scope topics, and surfacing sources for transparency.

3.3.System Architecture and Workflow Layers

1. Web UI (Client Layer)

   Simple page where users type questions and see numbered, step-by-step answers with citations. Mobile-friendly; supports Sinhala/Tamil/English text entry.

2. API Gateway & Auth

   Single entry point for the app (/ask, /sources, /feedback),Handles HTTPS, ratelimits, user/session tokens, and basic analytics.

3. Orchestrator / Planner (Agent Layer)

   Classifies the query (simple vs. complex),Decides shallow or deep retrieval, tools to call, and confidence thresholds to stop early.

4. Retrieval Layer

   Vector search over embeddings + BM25 keyword fallback, Pulls top relevant chunks from Sri Lankan property/family law corpus.

5. Re-ranking Layer

   Reorders retrieved passages for best match to the question, Reduces context length before generation (faster, cheaper).

6. Generation Layer (SLM)

   Fine-tuned Small Language Model produces the answer grounded only in retrieved text,Inserts inline citations and outputs a clear procedure/checklist.

7. Post-processing & Safety

   Formats steps (1-2-3…), highlights key documents/forms,Applies guardrails: disclaimers, out-of-scope handling, and sensitive-query rules.

8. Knowledge & Data Stores

   Document store (clean text with metadata: section, source, page),Vector DB (embeddings). Cache for frequent queries and popular passages.

9. Monitoring & Logging

   Stores anonymous feedback for continuous improvement.

10. DevOps & Deployment

User Workflow

User asks a question → planner decides retrieval depth → retrieve & re-rank → generate grounded answer with references → return a clear, numbered procedure.

3.4.Evaluation and Comparison

The evaluation strategy for this project is designed to measure both the technical performance and the practical utility of the proposed system. Benchmarking is carried out against several baselines, including a search-only retrieval system, a generic large language model (LLM) without access to local data, and, where computationally feasible, larger opensource models. To ensure relevance, the test sets are curated from Sri Lankan legal scenarios such as property transfers, deed verification, marriage and divorce procedures, and child custody cases.

A multi-dimensional evaluation framework is employed. Accuracy is assessed by comparing the correctness of generated steps against gold-standard references. Faithfulness is measured by calculating the proportion of statements that are directly supported by retrieved and cited passages. Latency is recorded in terms of both average response time and the 95th percentile (p95) to capture outlier delays. In addition, subjective measures such as user-rated helpfulness

and clarity are collected using Likert scales, and system usability is evaluated through the System Usability Scale (SUS). Cost-effectiveness is also considered by estimating per-query compute expenses and comparing them against alternative systems.

To deepen understanding of model behavior, ablation studies are performed. These include evaluating performance with and without a re-ranking component, comparing shallow versus deep retrieval strategies, testing different LoRA rank configurations, and contrasting quantised with full-precision model deployments. Validation of results is not limited to automatic metrics alone; expert legal reviewers assess a sample of outputs, and any disagreements between reviewers are carefully tracked. This process provides feedback for refining prompts, data pipelines, and model outputs.

3.5 .Tools and Technologies

Tools

- OCR Tools: Tesseract OCR, Google Vision API (for text extraction from scanned pages).
- Data Management: PostgreSQL/MySQL for database storage.
- Development Tools: Visual Studio Code, Jupyter Notebook, GitHub for version control.
- Testing Tools: Postman (API testing), Selenium (UI testing).

Technologies

- Backend: Python Fast api (to build APIs and integrate AI models).
- Frontend: React.js with Tailwind/MUI for building a user-friendly interface.
- AI Frameworks: Unsloth, Hugging Face Transformers, lang graph, LangChain for RAG pipeline.
- Deployment: AWS EC2 or Azure cloud for hosting backend and model. • Security: JWT authentication, SSL encryption for safe communication

3.6.Implementation&Deployment

Deployment of the system emphasizes cost-efficiency, maintainability, and scalability. Services such as the API layer, retrieval-augmented generation (RAG) pipeline, and vector database are containerised for portability and ease of deployment. The models are quantised, for example to 4-bit precision, in order to reduce memory consumption and operational cost. Hosting is planned on modest GPU or CPU instances to strike a balance between affordability and performance.

Operational robustness is ensured through modern DevOps practices. Continuous integration and deployment (CI/CD) pipelines allow safe updates, with blue-green or canary release strategies minimizing disruption during new deployments. Observability is supported by dashboards monitoring latency, error rates, and cache hit ratios, enabling proactive identification of bottlenecks or failures.

The architecture is designed for growth. New laws can be incorporated as separate indices, with built-in support for Sinhala and Tamil alongside English. The corpus will be refreshed and re-indexed periodically, and scheduled evaluation runs will be performed to monitor performance drift. Governance principles guide deployment, including adherence to content licences, transparent citation of sources, a defined takedown process, and a clear data retention policy with anonymisation of logs.

This methodology aligns well with the objectives of the project. It ensures clarity and focus by mapping each phase to specific goals with well-defined inputs and outputs. Reliability is reinforced through versioned datasets, fixed random seeds, and consistent evaluation metrics. Validity is supported by grounding outputs in retrieved and cited legal sources, while transparency is guaranteed through the use of citations, logs, and documented decision processes. Finally, adaptability is preserved by employing modular services, caching strategies, and planner agents that allow dynamic tuning between accuracy and speed depending on use-case requirements

4. PROJECT REQUIREMENTS

4.1 Functional Requirements

1. Family Law, Property Law in a simple text box.
2. Legal Data Retrieval – The system must search and retrieve relevant sections from the digitized Sri Lankan law dataset (books, PDFs, websites).
3. SLM-Based Response Generation – A fine-tuned Small Language Model (SLM) should generate step-by-step answers based on user queries.
4. Agentic RAG Integration – The system must combine retrieval with reasoning agents to provide accurate and time-efficient outputs.
5. Step-by-Step Guidance – Responses must be clear, actionable, and structured to guide users like a lawyer consultation.
6. User-Friendly Web Interface – The system must provide a clean and simple UI so that ordinary people can access legal help easily.
7. Multi-Device Support – The web application must be usable on desktops, tablets, and mobile devices.
8. Evaluation Module – The system should log responses and allow testing of accuracy, latency, and usability for research purposes.

4.2 Non-Functional Requirements

These define the quality attributes of the system:

1. Performance – Responses should be generated within 10 seconds for typical queries.
2. Scalability – The architecture must support future extensions (e.g., adding Labor Law, Criminal Law).
3. Usability – Interface must be simple enough for non-technical users (e.g., ordinary citizens).
4. Security – Sensitive legal queries must be handled securely with encrypted communication (HTTPS).

5. Maintainability – The system should be modular, allowing easy updates to the dataset or model.
6. Reliability – The system should ensure consistent and accurate outputs even under multiple user requests.
7. Cost Efficiency – Use of small language models (SLMs) and Unsloth framework to reduce computation cost compared to large models. 8. reduce computation cost compared to large models.

4.3. Expected Test Cases

Some possible test cases to validate the system:

1. Query Understanding

   Input: “What are the requirements to transfer a property deed in Sri Lanka?”

   Expected Output: A step-by-step explanation citing property law rules.

2. Family Law Query

   Input: “How can I file for divorce in Sri Lanka?”

   Expected Output: Clear procedure including required documents, court process, and legal conditions.

3. Accuracy of Retrieval o

   Input: “Explain the inheritance rights of children under Sri Lankan law.”

   Expected Output: Retrieval of correct law sections + simplified explanation.

4. System Response Time

   Test: Measure average response time for 100 user queries.

   Expected Result: 90% of responses delivered in under 5 seconds.

5. Multi-Device Usability

Test: Open the system on desktop, tablet, and mobile.

Expected Result: UI adapts and works without issues on all devices.

6. Error Handling

Input: “Tell me about space law in Sri Lanka.”

Expected Output: System politely responds that data is not available for that legal area.

7. Comparison Test

Test: Compare system answers with lawyer-provided answers for 50 sample queries.

Expected Result: At least 80% alignment in correctness and completeness.

4.4.Feasibility Study

The feasibility study checks if the project is realistic. It looks at:

Technical Feasibility: Available tools like OCR (Optical Character Recognition), NLP models, and Unsloth fine-tuning make the project technically possible.

Economic Feasibility: The system reduces the cost of accessing legal advice compared to hiring lawyers for small issues, making it cost-effective.

Operational Feasibility: Law students, researchers, and the public can easily use the system through a web application.

Time Feasibility: With the right team and tools, the system can be developed within the planned timeline.

5. SYSTEM DIAGRAM AND GANTT CHART

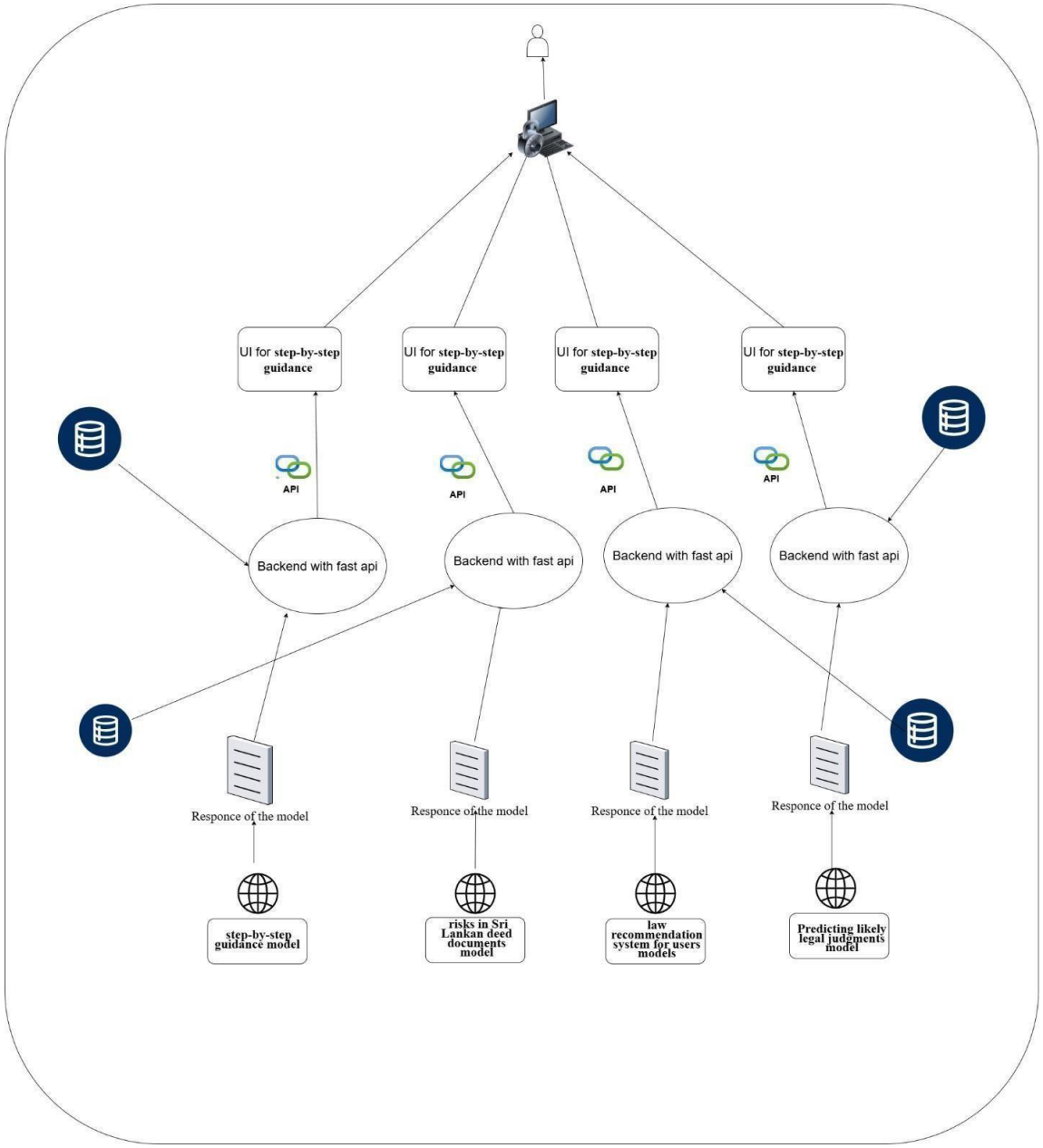

Figure 1:System Diagram for Relevance and Abuse Detection

This diagram illustrates the microservice architecture used in our application. Each of the four team members is responsible for developing an independent service, and they have the flexibility to use any framework, database, or dependencies of their choice.
Despite this diversity in the technology stack, all services are seamlessly integrated into a unified, user-friendly web interface.
The frontend UI provides step-by-step guidance to the user, ensuring clarity and usability.
Each UI module communicates with its respective backend service built using FastAPI, which exposes APIs for smooth interaction.
The backend services are connected to databases as needed and handle specialized tasks through different models:
1.Step-by-step guidance model ,2.Risk detection in Sri Lankan deed documents,
3.Law recommendation system for users,4.Predicting likely legal judgments
Each model generates a response that is passed back through the backend, processed, and displayed to the user in the web interface.
This architecture ensures:
Scalability, since each microservice can be developed, deployed, and scaled independently.
Flexibility, as developers are not restricted to a single stack or dependency set.
Integration, with all services contributing to one cohesive application that prioritizes the user experience.

5.1 Technical Daigram

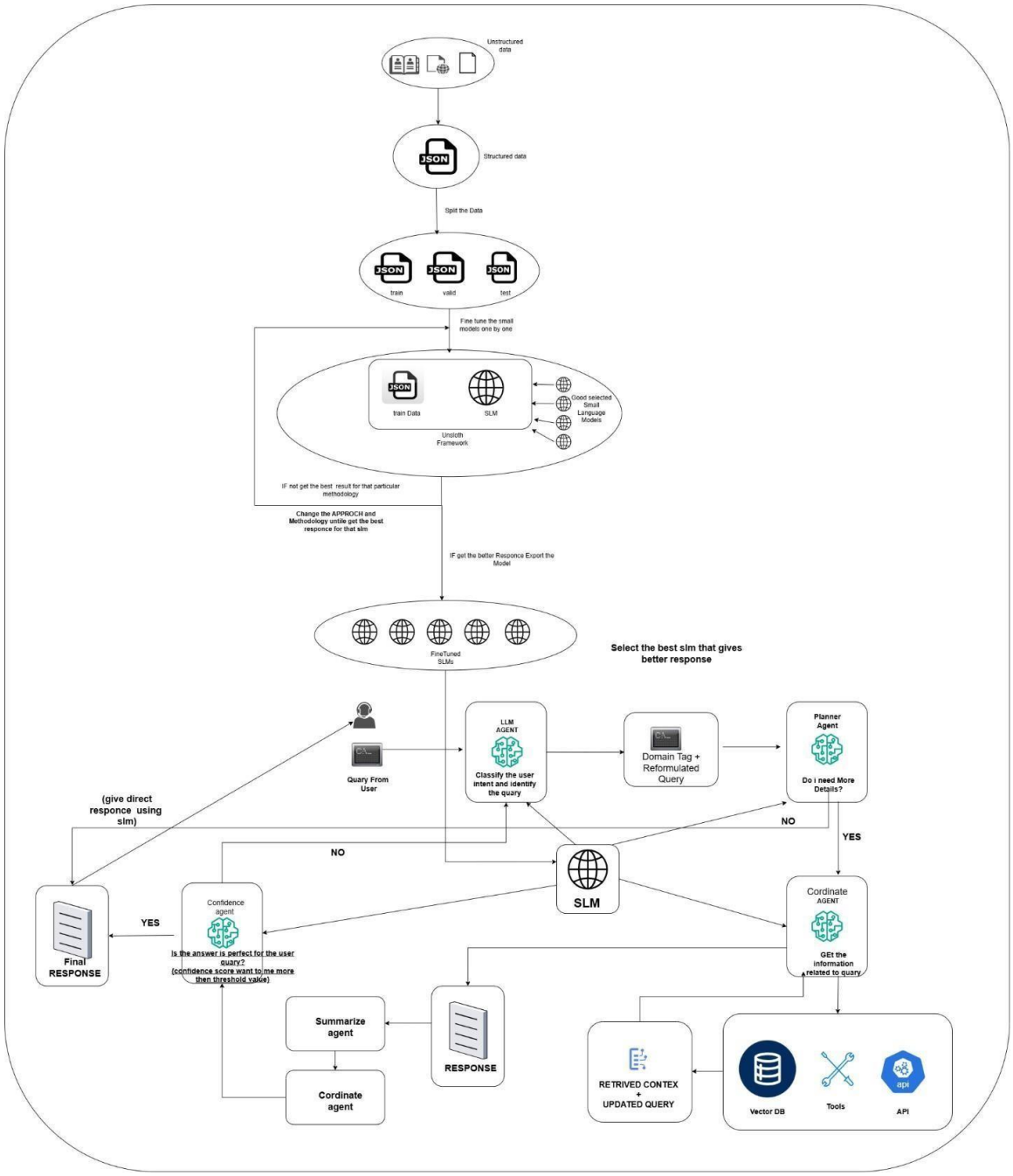

Figure 2:Technical diagram

structure the data → fine-tune 5 small models → select the best → test different RAG agents → deploy the best agentic architecture as a web app.

1. **Data Structuring & Preprocessing**

   The process begins with collecting **unstructured data**.

   The data is then converted into a **structured JSON format**, which allows for consistent formatting and easier ingestion during model fine-tuning.

2. **Fine-Tuning with Small Language Models (SLMs)**

   Using the prepared JSON data, I will fine-tune **five different small language models** (encoder–decoder architectures).

   Each model will be evaluated, and the **best-performing model** (in terms of accuracy, relevance, and response quality) will be selected for deployment.

3. **Agentic RAG Architecture**

   After fine-tuning, I will experiment with **different agentic RAG architectures**.

   These architectures combine **retrieval mechanisms, LLM reasoning, and external tools** (e.g., vector databases, APIs, utilities) to improve the reliability and factual grounding of responses.

   The aim is to determine **which RAG configuration yields the best performance** for our use cases.

4. **Query Handling & Response Generation**

   A user query enters the system and is first refined by the LLM (query rewriting).

   If external knowledge is needed, the system determines **which sources to retrieve from** (vector DB, tools, APIs).

   Retrieved context is merged with the updated query and passed back to the LLM for reasoning.

   The LLM then decides whether the response is sufficient as-is or requires refinement.

   Finally, the system outputs the **best possible response** to the user.

5. **Iterative Optimization**

   If responses are suboptimal, the system iteratively adjusts **fine-tuned models, prompt templates, and retrieval strategies** until the most effective configuration is achieved.

5.2 Gannt chart

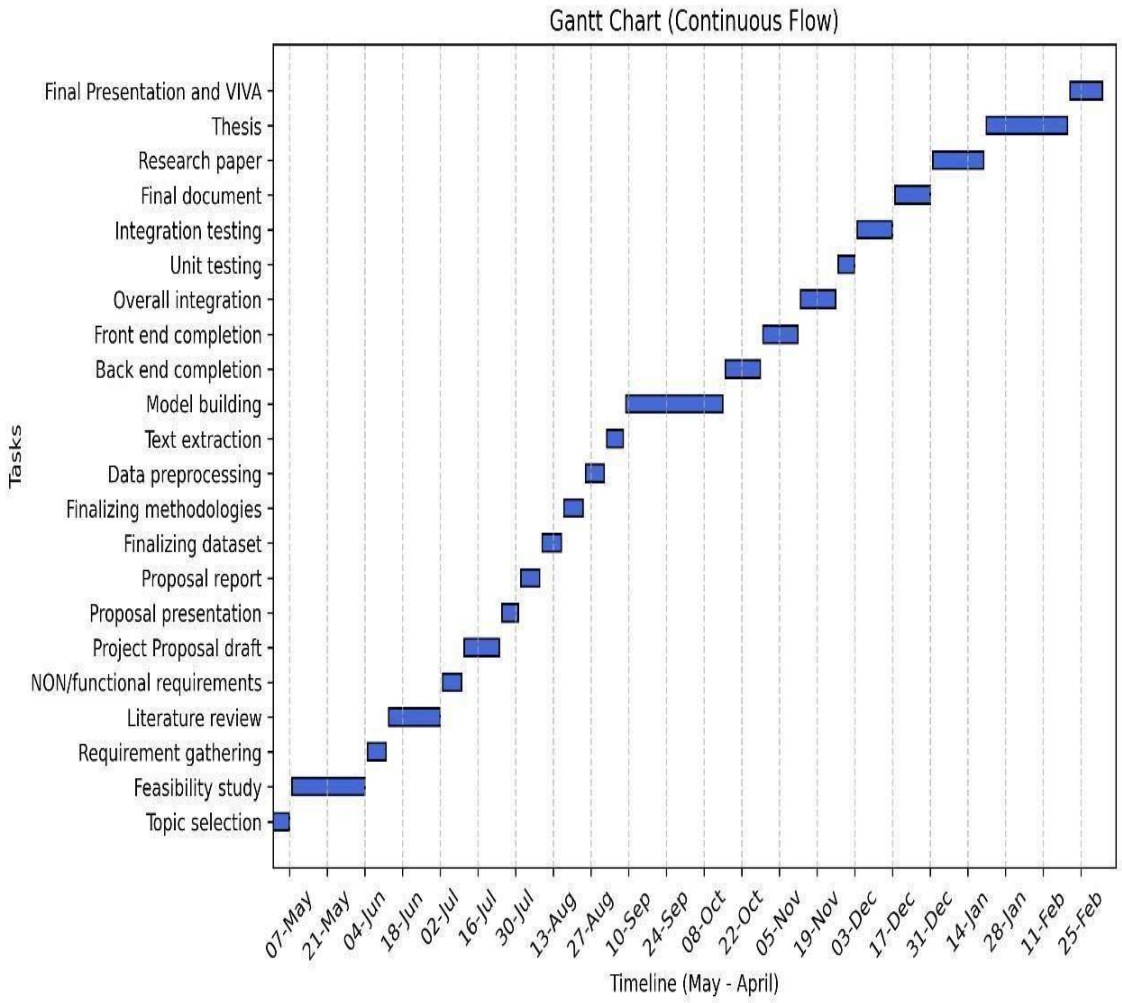

Figure 3:Gannt Chart

6. CONCLUSION

This research plans to make legal help easier and cheaper for people in Sri Lanka. We will build a small language model that gives clear, step-by-step guidance for Property Law and Family Law. The model will be trained with trusted local books and documents, and it will use an agentic RAG method to find the right law text before answering. This helps reduce mistakes and gives faster, more reliable replies. The final system will be a simple web app that anyone can use.

By using efficient fine-tuning (Unsloth/LoRA) and a small model, the solution will be affordable to run, even with limited computers. It will show answers with sources, so users can trust the information. This can save time and money for students and the public who need help with common legal issues.

The system is not a replacement for lawyers, but it can give first guidance and help people understand their options. In the future, we can add more areas of Sri Lankan law, support Sinhala and Tamil better, and keep improving accuracy with new data and feedback. This work is a practical step toward fairer access to legal knowledge in Sri Lanka.

7. BUDGET AND BUDGET JUSTIFICATION

The Smart Legal Guidance For Sri Lankan Property Law , Family Law for budget is provided in LKR and includes key expenditures for web hosting, development tools, internet, cloud services, stationery, and incidental charges, assuring a sustainable implementation within Sri Lanka's court framework.

Category	Cost (LKR)	Justification
Web Hosting	5,000	Covers basic hosting for the judicial dashboard, estimated using local providers such as srilankaonline.lk.
Development Tool Cost	10,000	Includes licenses for programming tools such as Google Colaba, Visual Studio Code and PostgreSQL.
Cloud Services (GCP)	15,000	Estimated GCP consumption (e.g., e2-micro instances @ ~LKR 2.4/hour, 500 hours/month for 9 months) and storage requirements.
Stationary Cost	10,000	Covers paper ,Data extraction from Scanning printing, and office supplies for documentation and meetings.
Miscellaneous	3,000	Allows for unexpected costs such as travel and small equipment.
Total	**43,000**	

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