

AI-Powered Legal Assistance for Bangladeshi Laws using Machine Learning and Large Language Models

By
Md. Ashikul Islam
213-15-14787

Shishir Ahmed Midul
213-15-14776

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the
Requirements for the **Degree of Bachelor of Science in
Computer Science and Engineering**

Supervised by

Dr. Sheak Rashed Haider Noori
Professor & Head
Department of Computer Science and
Engineering
Daffodil International University

Co-Supervised by

Mr. Showmick Guha Paul
Lecturer
Department of Computer Science and
Engineering
Daffodil International University



**DAFFODIL INTERNATIONAL
UNIVERSITY**
Dhaka, Bangladesh

September 16, 2025

APPROVAL

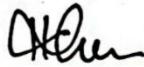
This Project titled “AI-Powered Legal Assistance for Bangladeshi Laws using Machine Learning and Large Language Models”, submitted by Md. Ashikul Islam and Shishir Ahmed Midul to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 16 September 2025.

BOARD OF EXAMINERS



Dr. Bimal Chandra Das
Professor and Dean (in-charge)
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



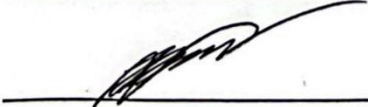
Most. Hasna Hena
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Mayen Uddin Mojumdar
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Nazibur Rahman
Technical Lead, Database Administrator
Telenor
Grameen Phone Account

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Sheak Rashed Haider Noori**, Professor & Head, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



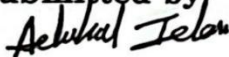
Dr. Sheak Rashed Haider Noori
Professor & Head
Department of Computer Science and
Engineering
Daffodil International University

Co-Supervised by:



Mr. Showmick Guha Paul
Lecturer
Department of Computer Science and
Engineering
Daffodil International University

Submitted by:



Md. Ashikul Islam
Student ID: 213-15-14787
Department of Computer Science and
Engineering
Daffodil International University



Shishir Ahmed Midul
Student ID: 213-15-14776
Department of Computer Science and
Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project (FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Professor Dr. Sheak Rashed Haider Noori, Professor & Head**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Artificial Intelligence & Web Application Development** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

In Bangladesh, getting legal help is very difficult because the legal system too much complex and most people cannot afford legal advice. The situation is even worse for people in rural areas, since professional legal services are almost never available outside of the big cities. To solve this problem, we created a system called AI Lawyer, which is powered by Artificial Intelligence (AI). The system gives people free, simple and always available legal advice based on Bangladeshi laws. We used Artificial Intelligence (AI) and Natural Language Processing (NLP) so that the system can answer legal questions in Bangla, English and Banglish. To build the dataset, we made a Python scraper that collected legal texts from the official Bangladesh law portal. This data was cleaned, structured and stored in JSON format so that the AI model could understand it properly. To make the system faster and more affordable, we fine-tuned an LLaMA 3.1 8B model using LoRA adapters and 4-bit quantization. For deployment, we used a dual inference system. A locally fine-tuned model running on Ollama gives quick answers, while the Gemini API is used as a backup to ensure continuous service if the local model fails. The chatbot is available through a web-based interface developed with Next.js, React, Node.js and PostgreSQL. We tested AI Lawyer with real legal questions and found that it provides fast, accurate and easy-to-understand answers. We also compared it with other systems like DoNotPay and LawBot and found that our solution works much better for the Bangladeshi context. User feedback confirmed that the chatbot is simple to use, dependable and very useful. The AI Lawyer project reduces people's dependency on expensive lawyers, protects citizens from legal fraud and misinformation and helps them understand their rights more clearly. It makes legal information cheaper, more accessible and available in multiple languages. In the future, we plan to expand the dataset, improve accuracy and add voice-based features so that even more people, including those who cannot read or type easily, will be able to use the system.

Table of Contents

Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1-3
1.1 Introduction.....	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Methodology	2-3
1.5 Project Outcome.....	3
1.6 Organization of the Report	3
2 Background	4-9
2.1 Introduction.....	4
2.2 Literature Review	4-7
2.2.1 Similar Applications	7
2.3 Gap Analysis	8
2.4 Summary	9
3 Research Methodology	10-26
3.1 Methodology	10-11
3.2 Requirement Analysis & Design Specification.....	11-12
3.2.1 Overview	12
3.2.2 System Design.....	12-15
3.2.3 Functional and Nonfunctional Requirements	15-16
3.2.4 Context Diagram	16
3.2.5 Data Flow Diagram	16-17
3.2.6 UI Design.....	17-23
3.3 Detailed Methodology and Design.....	24
3.4 Project Plan.....	24-25
3.5 Task Allocation.....	25-26

3.6	Summary.....	26
4	Implementation and Results	27-32
4.1	Environment Setup.....	27-28
4.2	Testing and Evaluation/Performance/ Comparative Analysis.....	28-30
4.3	Results and Discussion.....	30-31
4.4	Summary.....	32
5	Engineering Standards and Design Challenges	33-44
5.1	Compliance with the Standards.....	33-38
5.1.1	Software Standards.....	33-36
5.1.2	Hardware Standards.....	36-37
5.1.3	Communication Standards.....	37-38
5.2	Impact on Society, Environment and Sustainability.....	38-39
5.2.1	Impact on Life.....	38
5.2.2	Impact on Society & Environment.....	38
5.2.3	Ethical Aspects	39
5.2.4	Sustainability Plan.....	39
5.3	Project Management and Financial Analysis.....	39-40
5.4	Complex Engineering Problem.....	40-43
5.4.1	Complex Problem Solving.....	40-42
5.4.2	Engineering Activities.....	42-43
5.5	Summary.....	43-44
6	Conclusion	45-46
6.1	Summary	45
6.2	Limitation.....	45-46
6.3	Future Work	46
	References	47-49

List of Figures

3.1.1 Methodology Diagram	11
3.2.1 System Architecture	13
3.2.2 Context Diagram	16
3.2.3 Data Flow Diagram	17
3.2.4 Landing Page	18
3.2.5 Registration Page	19
3.2.6 Login Page	19
3.2.7 Chatbot Interface.....	20
3.2.8 Admin Panel User Management.....	21
3.2.9 Model Management.....	21
3.2.10 Model Configuration.....	22
3.2.11 System Prompt Configuration	22
3.2.12 Database Management.....	23
3.2.13: Edit Record	23
4.2.1 Training Loss Curve.....	29
4.2.2 Gradient Norm Curve.....	29
4.2.3 Learning Rate Schedule	29
4.2.4 Global Step Progression	30
4.2.5 Epoch Progression	30

List of Tables

2.2.1 Summary of Literature Reviewed.....	5-7
2.2.2 Similar Applications of AI in Legal Assistance.....	7
2.3.1 Gap Analysis of Existing Systems vs Proposed System.	8
3.5.1 Task Allocation and Timeline.	25-26
4.3.1 Sample from AI Lawyer Corpus (legal questions and answers).....	31
5.3.1 Financial Analysis.....	40
5.4.1 Mapping with Complex Engineering Problem.	41
5.4.2 Mapping with knowledge Profile.	42
5.4.3 Mapping with Complex Engineering Activities.....	43

Chapter 1

Introduction

This chapter explores our motivation behind building an AI-powered legal assistant for Bangladesh laws. Here we explain the problems that people face in our country. Like the complex legal system language, the high cost of lawyers and the lack of services in rural areas. We are defining the objective of our system. We also discuss the methodology and the project outcomes.

1.1 Introduction

In Bangladesh getting proper legal help is still a very difficult task for most people. The legal system of our country is complex, time-consuming and expensive. Hiring a lawyer is not affordable for many families and even when legal services are available, they are not easy to understand. According to a 2022 survey, nearly 75% of people in Bangladesh have very little or no knowledge about law. Also, the legal documents are written in complicated language, which confuses people. As a result, many people hesitate to take legal help and try to avoid the system, but later they face serious problems because they do not know their rights and responsibilities. The situation creates a gap between the people and the justice system. People from rural areas suffer even more because lawyers are not easily available outside big cities. Many people also depend on informal advice from neighbors or local leaders, which is often wrong. This shows the need for a simple, low-cost and reliable system that can guide people with the right legal information. In our project “AI Lawyer” we try to solve this problem. It is an AI assistant powered by AI that helps people to give legal advice in a very easy and understandable way that everyone can understand easily. AL lawyer is a chatbot that works in Bangla, English and Banglish. It takes questions in those 3 languages and it can give answers in these 3 languages. Anyone can use it anytime and from anywhere. So, people do not need to wait to meet the lawyers. They can ask their query or question in the AI Lawyer chatbot. The main benefit of this project is it reduces the cost of a lawyer and the full dependency on a lawyer. For normal legal advice or queries, people need to meet a lawyer and they must pay the cost of the lawyer, but with our AI Lawyer project, people can get fast legal advice and they can take the right step by taking that advice. The purpose of this project is to help the rural area people those who don't know about the laws of Bangladesh and don't have money to go to the lawyer for their queries. Our AI lawyer is mainly for those people, through our project, they can ask their questions and they will get the answer in a very easy and understandable way. For this project we used the latest AI and NLP. In the whole world every country is starting to use AI in their legal matters. But in Bangladesh there is nothing developed for the legal system. By collecting data from BD law websites, we trained the data with AI models and we made a customized model to solve this problem. AI Lawyer is not a chatbot; it is a step for making the justice system transparent and friendly. It gives people an opportunity to learn about laws, legal matters and their rights without any cost or fear. Our future is to make a digital legal assistance system that will grow from time to time and give better service with more data and details. This project is a combination of technology and social good, aiming to build a bridge between people and the legal system of Bangladesh.

1.2 Motivation

The motivation behind this project comes from the real-life problems faced by millions of people in Bangladesh when it comes to understanding and accessing legal help. In our observation, the legal system remains out of reach for the public due to several key reasons it is complicated, expensive and written in difficult language. As students of computer science and engineering, we felt a responsibility to use our knowledge to solve a problem that affects so many people in everyday life. Legal awareness is a basic need, yet most people do not have access to it. In rural areas and even in cities, many individuals live their entire lives without understanding their basic rights. Even when someone needs legal help, they often hesitate because they are afraid of legal language, high lawyer fees, or delays in the process. This situation motivated us to find a solution using modern technology. The rapid growth of Artificial Intelligence (AI) has shown how it can solve real-world problems. AI is already used in healthcare, education, customer service and even farming, so why not the law? We realized that AI has the power to break down the complex structure of legal information and present it in a way that people can understand and use. Another motivation was the language gap. Most legal texts are written in complex English or formal Bangla, making it hard for people with limited education to understand. We wanted to create a system that explains the law in simple, everyday Bangla and English so that even people from remote areas can benefit from it. This project is not just a chatbot; it's a tool for legal empowerment. We also noticed that there is no free, 24/7 service in Bangladesh where a person can just ask a legal question and get an answer right away. That's why our AI-based virtual legal assistant can be a game changer; it will be always available, easy to use and cost-effective.

1.3 Objectives

The main goal of our project is to build an AI chatbot that gives legal advice based on Bangladeshi laws in both Bangla and English. We want to make legal help easier, faster and more affordable for everyone.

Our key objectives are:

- Create a chatbot that understands and replies in both Bangla and English.
- Collect and organize legal data from trusted sources.
- Translate complex legal terms into simple language.
- Train AI models with Bangladeshi legal content.
- Use smart tools to find the best legal answers.
- Build an easy-to-use interface for users.
- Test the chatbot to ensure it gives correct and clear answers.
- Make sure the tool is safe and accurate with help from legal experts.
- Help people understand their rights without needing a lawyer.

1.4 Methodology

To build our AI lawyer, we followed some simple but important steps. First, we collect legal data from the official Bangladesh law website (bdlaws.minlaw.gov.bd).

We build a Python scraper to copy all the laws, sections and rules into a structured format that is JSON format, so the computer can read them. Next, we cleaned the data; this means we removed extra symbols, fixed errors and arranged the laws properly. Some laws were in English and some in Bangla, so we keep both to make our system bilingual. After that we prepared the dataset by creating question-answer pairs. For example, if someone asks about a law, the dataset has the answer ready for training the AI. Then we trained AI models using modern tools like Hugging Face Transformers and Google Colab. We fine-tune the models so they can understand Bangladeshi law and reply in both Bangla and English. Finally, we tested the system by asking different legal questions to check if the answers were correct, simple and useful. We made a chatbot interface for people to ask questions easily from their phone or computer.

1.5 Project Outcome

The main outcome of our project is to create an AI-powered chatbot that can provide legal advice based on Bangladeshi laws. This chatbot can give responses in Bangla, English and Banglish. It is much easier for people from different backgrounds to access legal information. The AI Lawyer chatbot is designed for giving clear and simple answers to help people understand their rights without paying. After finishing the project, we hope to have a fully functional legal assistant chatbot that can handle and give the various legal questions answers in Bangla, English and Banglish. The chatbot will be able to give answers in a very simple way so that a person who has a little educational background can also understand everything. This data will be structured so that it can easily be used by the AI model for answering questions. Another important achievement will be the creation of a fine-tuned AI model, which will be trained on this legal dataset to understand Bangladeshi laws and answer questions accurately. We will also design a simple user interface for the chatbot so people can easily ask legal questions anytime, anywhere.

1.6 Organization of the Report

This report is divided into six main chapters. Each chapter explains a different part of our project in a simple and clear way, Chapter 1 Introduction This chapter explains what our project is about, why we chose this topic, what we want to achieve, what the results will be and how the report is organized. Chapter 2 Background This chapter gives some basic knowledge about the topic. It includes other similar works (literature review), examples of existing systems and what is missing in those systems (gap analysis). Chapter 3 Methodology In this chapter, we describe how we planned and designed our system. It includes the steps we followed, the system design, what features we added, how the data flows and how we divided the tasks. Chapter 4 Implementation and Results This part talks about the tools we used, how we tested the system, the results we got and what we learned from them. Chapter 5 Standards and Challenges This chapter discusses the rules and standards we followed, the problems we faced, how our project affects people and the environment and the cost and planning of the project. Chapter 6 Conclusion The last chapter gives a summary of the project, what problems we couldn't solve and what we plan to do in the future to improve the system.

Chapter 2

Background

This chapter explains the context and background of the AI Lawyer project. We explore the literature review here and highlight how AI is being used globally in legal domains. We discuss some similar applications like POSS, DoNotPay, etc. Here also is the gap analysis limitation of the existing application.

2.1 Introduction

Before explaining our project, we need to know why the system is needed. In our country there are many people who struggle to understand legal documents because the legal documents are written in a very complex way. This makes it hard for everyone to understand the legal system. If anyone has some queries about the legal system and if they do not have any knowledge about the legal system, they must hire a lawyer for just their simple queries and the lawyer takes a cost for the counseling. That will be very costly for some people. These problems have very much happened in the rural areas because most of the people do not understand the laws or the legal system; that is why they need a lawyer for their simple problems and in the rural area finding a lawyer is very difficult and if there is a lawyer, the cost of the lawyer is very high. As a result, many people avoid taking legal advice and help. For that, they don't know what to do or where they need to go. This situation is a barrier for people who want to understand their rights, but it is difficult and costly. On the other side, AI is starting to be used as a legal assistant in many countries. AI proves itself to be effective in answering legal questions and explaining legal problems. The AI-powered system gives responses very quickly and it can work with large amounts of legal information. So, we decided to build an AI assistance system that works like a chatbot and we fine-tuned the AI with Bangladesh laws and legal problems in Bangla, English and Banglish. So that it can read questions in that language and give answers in Bangla and English. Best of all, it is free to use, so anyone can use it.

2.2 Literature Review

In this section we are going to comment on big literature on how Artificial Intelligence (AI) can be applied to law. The review helps to appreciate the implementation of AI both at the global and Bangladeshi levels together with the challenges. Wasi et al. [8] explored AI-based legal assistance and talked about how GPT-2 models can be developed to assist with legal text using the UKIL-DB-EN corpus in Bangladesh. Their article focused on the promise of natural language models to work with Bangladeshi statutes and come up with user-friendly online platforms on law. It also pointed out that local data with AI will decrease the gap of justice, especially in the rural regions where there are few traditional legal services. Other researchers have also demonstrated that chatbots driven by AI can streamline legal terminologies and legal advice can become easier to access. As an example, Zhang et al. [1] showed that chatbots have the capability of simplifying complex law wording to a better understanding by the population. Equally, Patel and Sharma [2] demonstrated that

AI is able to scan legal documents faster and uniformly, reducing human error, which is the most important aspect in law firms that handle a large volume of cases. Rahman et al. [9] discussed the situation in Bangladesh and found that AI could be of great help to rural residents who have less access to lawyers. Their results revealed that the majority of rural citizens are unable to afford attorneys because they do not understand the laws. This would help to trust the justice system because AI systems may help to decrease the dependency on intermediaries and offer direct and low-cost legal advice. In addition to chatbots, AI has also become helpful in law research and information search. Lee et al. [3] showed how AI could use quick searches to find the relevant laws, precedents and case histories and save not only time but also energy. Nonetheless, Raham et al. [12] observed that AI tools in Bangladesh should be capable of processing two-language legal texts, since legal texts are being published in both Bangla and English. Models should be able to alternate between the two languages in order to facilitate inclusivity and precision. Predictive analytics within the legal system is another area that can be promising. Johnson et al. [5] demonstrated the ability of AI to examine extensive sets of previous cases and anticipate the future results of cases. In a similar manner, Islam et al. [11] studied predictive case modeling in Bangladesh, which is especially important considering the enormous number of court cases in the backlog. These systems would enhance management of cases and allocation of resources within the judiciary. The latest research has gone a step further to develop AI in the domain of niche legal usage. Hasan [13] proposed JusticeNetBD, which is a Retrieval-Augmented Generation (RAG) that will serve to help Bangladeshi women in legal matters. It showed better speed and accuracy than general-purpose models and it is important to note the necessity of customization of AI systems by social groups. Cui et al. [14], in another worldwide endeavor, have come up with Chatlaw, a legal assistant multi-agent with knowledge graphs. This achieved a much greater reduction in hallucinations as well as accuracy in legal reasoning than GPT-4. Kalaycioglu et al. [15] introduced the LICES, an AI-based legal intelligence system architecture that is configurable towards compliance and efficiency. Their work showed that large language models together with structured legal databases can cut legal research time by more than 90 percent and uphold accuracy over 98 percent. In the meantime, Mahoney et al. [16] concentrated on explainable AI to classify legal documents and give an interpretation to the attorneys, raising the confidence level of lawyers in automated review systems. Lastly, Guha et al. [17] have proposed LegalBench, a benchmark data set to test legal reasoning in LLMs. This benchmark assists in evaluating the performance of models in various regions of law and this is useful in the future development of domain-specific models such as ours. All in all, the reviewed literature indicates that despite the high potential of AI to make legal services more accessible, the technology will have to address several issues language diversity, local law specificity and the precision of its answers in particular legal cases. But the future of legal AI solutions is already being achieved by progress in specialized systems, retrieval-based solutions and explainability. In Table 2.2.1 we summarize the literature review that covered here.

Table 2.2.1: Summary of Literature Reviewed

Author (s)	Year	Title	Methodology	Key Findings
Wasi et al.[8]	2024	Exploring AI Powered Legal Assistance in Bangladesh	Literature Review	Discussed the use of AI in Bangladesh's legal system and the creation of GPT-2 for legal texts.

Zhang et al. [1]	2021	Legal AI and chatbots: A New Era	Literature Review	Explored how AI chatbots can help explain legal terms and provide legal advice.
Patel and Sharma [2]	2022	AI in Legal Document Review	Experimental Research	Showed how AI can help review legal documents quickly and accurately
Zhang and Liu [6]	2018	AI and Law: Application and Challenges	Case Study	Explained the challenges of using AL in legal systems with different languages and laws.
Rahman et al. [9]	2020	AI in Legal Service in Bangladesh	Literature review	Found that AI can reduce the cost of legal services and help people get legal advice.
Islam et al. [11]	2021	AI and Case Outcome Prediction in Bangladesh	Case Study	Looked at how AI could predict court case outcomes in Bangladesh's legal system.
Islam et al. [11]	2021	The study is called AI and Case Outcome Prediction in Bangladesh.	Case Study	Researched the ways in which AI can be used to anticipate the outcome of a court case in Bangladesh.
Raham et al. [12]	2022	Language Challenges for AI in Bangladeshi Legal Systems	Literature Review	Highlighted that AI must address Bangla-English bilingual legal documents to work effectively in Bangladesh.
Lee et al. [3]	2021	AI for Faster Legal Research	Experimental Study	Demonstrated how AI improves efficiency in retrieving laws and precedents.
Johnson et al. [5]	2021	AI and Case Outcome Prediction	Case Study	Illustrated how AI can forecast case outcomes using historical judgments.
Hasan [13]	2025	JusticeNetBD: RAG-Based Legal Assistant for Women in Bangladesh	Experimental Study	Introduced a retrieval-augmented chatbot tailored to women, with higher accuracy than general LLMs.
Cui et al. [14]	2023	Chatlaw: Multi-Agent Legal Assistant with Knowledge Graphs	Experimental Study	Combined multi-agent LLMs with knowledge graphs; reduced hallucinations, outperformed GPT-4.

Kalaycioglu et al. [15]	2025	LICES: Legal Intelligence Compliance Expert System	System Architecture	Designed an AI-powered compliance tool, reducing research time by >90% with 98%+ accuracy.
Mahoney et al. [16]	2019	Explainable Legal Document Classification	Experimental Study	Developed interpretable text classification framework for document review.
Guha et al. [17]	2023	LegalBench: A Benchmark for Legal Reasoning in LLMs	Benchmark Dataset	Introduced an evaluation dataset to measure AI reasoning in legal contexts.

2.2.1 Similar Applications

We looked at some apps and websites that already use AI to help people with legal problems. These systems gave us good ideas, but we also noticed they are not made for Bangladesh. In Table 2.2.2, we made a list of the similar applications and we discussed the features and service and the limitations.

Table 2.2.2: Similar Applications of AI in Legal Assistance

Application	Country/Region	Features and Services	Limitations (in Bangladesh)
DoNotPay	USA, UK	The first robot lawyer in the world. Appeal parking fines, cancel services and get refunds.	Works USA/UK only, not Bangla or Bangladeshi legal system.
ROSS Intelligence	USA	AI-based legal research assistant, assists lawyers to find legal responses and case law in a short time.	Written in lawyer-language, not in general language; not adjusted to the Bangladeshi law.
LexisNexis	Global	Legal research tool with paid subscriptions; includes access to case laws and legal documents and advanced search capabilities.	Very costly, not localized to Bangladeshi law, difficult to use.
LawBot	UK	Free AI chatbot that gives legal advice and assists people with straightforward legal procedures.	Is UK-only; not Bangladesh-customized; only in English.
Legal Robot	USA	Reads and analyzes legal documents; explains legal contracts in plain English.	Only English, not suitable to Bangla or Bangladeshi law.

2.3 Gap Analysis

At present most of the existing legal help systems or websites in Bangladesh have very limited features. They mainly provide static legal information with FAQs or downloadable documents. But they do not actively interact with users in simple Bangla or English. These platforms also fail to explain the law in an easy way for people with little legal knowledge. Additionally, there are no free or always available systems where people can ask their questions anytime and get instant answers. Our system fills the gaps by introducing an AI-powered chatbot that is available 24/7, can understand Bangla and English and can explain legal issues in very simple language. Unlike traditional websites that only provide pre-written answers, our chatbot will allow real-time interaction to provide personalized responses and guide people step by step. It also reduces dependency on costly lawyers and helps people in remote areas who usually cannot access legal support. Thus, the gap analysis clearly shows that while current solutions provide only static, limited legal resources, our proposed system will provide an interactive, user-friendly, low-cost and multilingual solution. In Table 2.3.1, we analyze the gaps between existing systems and our proposed systems.

Table 2.3.1: Gap Analysis of Existing Systems vs Proposed System

Features	Existing Legal Websites in Bangladesh	Foreign AI Legal Bots	Proposed System (AI Lawyer for Bangladesh)
Provides legal documents	Yes	Yes	Yes
Explains law in simple Bangla	No	No	Yes
Explain law in English	Partially	Yes	Yes
Chatting option/AI based interface	No	Yes	Yes
27/7 availability	No	Yes	Yes
Covers Bangladeshi law specially	Limited	No	Yes
Provides step by step legal guidance	No	Yes	Yes
Accessible for remote users	No	No	Yes
Free or low-cost service	No	No	Yes
Multilanguage (Bangla and English)	No	Limited	Yes
Recommendations or filtering latest products	Yes	Yes	Yes
Easy interface for users	No	Limited	Yes
Personalize legal answers	No	Yes	Yes

2.4 Summary

In this chapter, we talked about why our project is important. We saw that many people in Bangladesh do not understand the law and cannot get legal help easily. We also looked at some apps and websites from other countries that help with legal problems, but they do not work for Bangladeshi people. Most of these tools don't support Bangla or Bangladeshi laws. They are hard to use and cost money. We found many problems and gaps and explained how our project will solve them. Our chatbot will work in Bangla and English, be free to use and help people understand the law easily. So, this chapter shows the reason why we are doing this project and how it will help people.

Chapter 3

Research Methodology

This chapter describes the steps we followed to design and build the AI-powered legal assistant. It explains both the functional and non-functional requirements that guided the development process, such as the ability to provide bilingual answers, ensure security and deliver fast responses. The chapter then introduces the system design, explaining how we structured the data, trained the AI model and developed the user interface. Each stage, from data collection to deployment, is presented in a logical flow. The chapter also outlines the project plan and task allocation, showing how responsibilities were shared among team members and how progress was tracked.

3.1 Methodology

In this methodology we first collect the data from the official law portal BD Laws. For this we make a scraper with Python. It copied all the laws, acts and sections from the website automatically. We structured the dataset in JSON format. Because we saw that the AI models understand the JSON format better than any other format. The formatting was important; the raw text was unorganized and written in both Bangla and English. After finishing the data collection, we clean and prepare the dataset. Our system is bilingual for that, we have to keep both Bangla and English text. We also create the question-answer dataset. For AI Cloud, learn how people normally ask legal questions. And I give the simple answer. Now we can move on to the very important part, which is model training. We used a large language model that was LLaMA 3.1 8B that is quantized in 4 bits. It helps to train the model fast and at a lower cost. We applied the LoRA method and also used modern tools like Hugging Face Transformations and Unsloth. We trained all those things in Google Colab using the GPU support. When the model was ready, we worked on the user interface and connected the AI mode with a chatbot system. We made the frontend by using Next.js, React and Tailwind CSS. For the backend we used Node.js and the PostgreSQL database. We used Ollama for the last local response and we used GeminiAPI as a backup. We did testing and evaluation. For this we tested the chatbot by giving different legal questions in Bangla, English and Banglish. In Figure 3.3.1 we've implemented our proposed methodology.

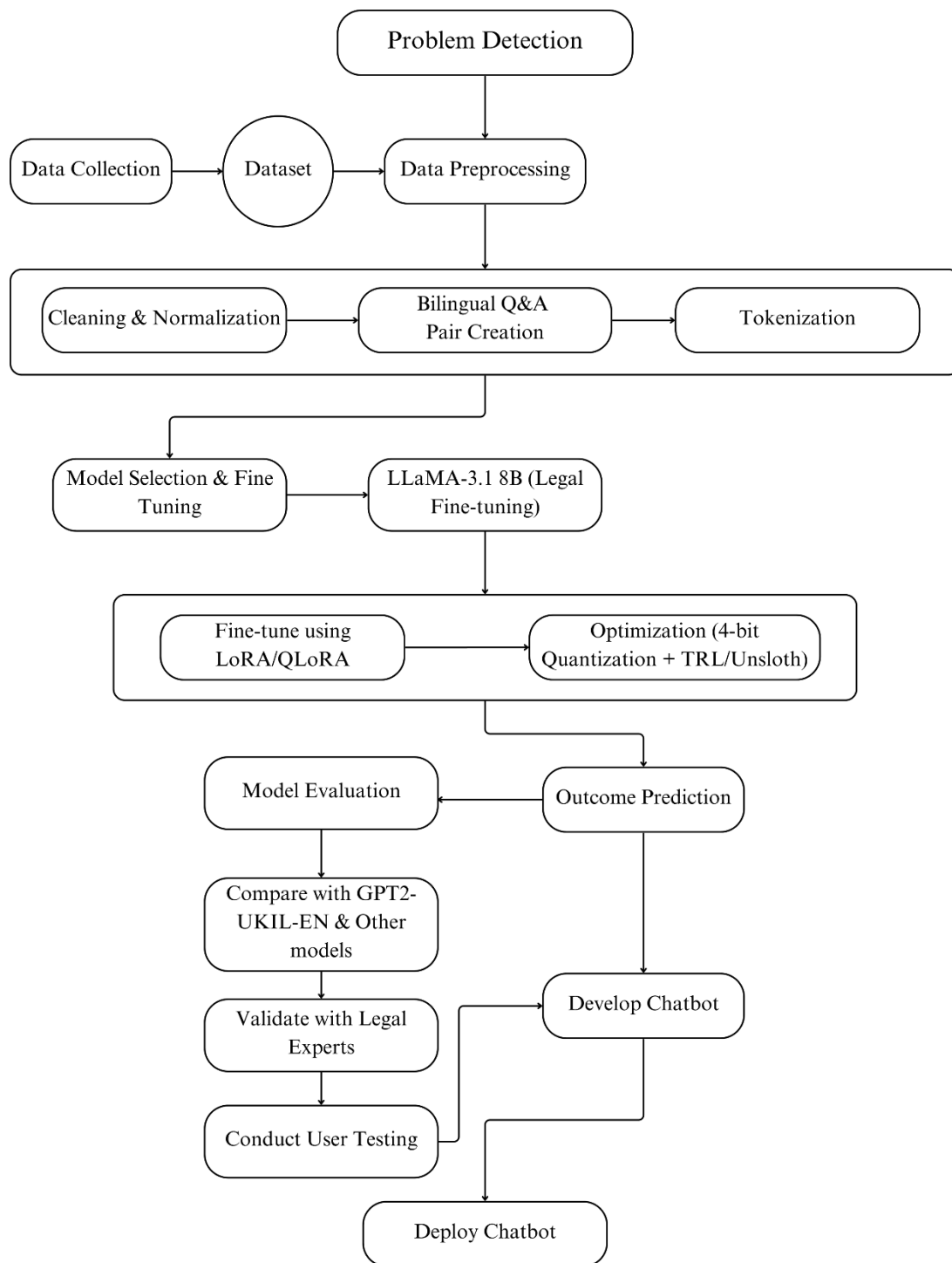


Figure 3.1.1: Methodology Diagram

3.2 Requirement Analysis & Design Specification

Basically, the AI lawyer is made to help people understand the Bangladeshi law in every way. To build up this system, we analyzed and studied what the system should do and how it does it. We focused on two main things that are functional requirements and non-functional requirements. In the functional requirements, the main thing is to collect the data from the BD law site and organize the data in the proper format. This organized dataset is used to give the answer that the people are asking for in Bangla, English and

Banglish. The system handles many tasks, like question answering, giving legal advice and giving some basic case predictions. To make it user-friendly, we make it as a chatbot format where people give their questions or queries and the chatbot gives them the answers in a very easy way. There are also secure login options and an admin panel where managers can handle users, manage datasets and update the AI models. Along with those functions, we also make sure that the system meets some non-functional requirements. So that the system works well and stays trustworthy. In the nonfunctional side, one of the important things is accuracy. Because people depend on the system for their legal queries and if the system gives wrong answers, then it will be a disaster for the system and the users, so the most important thing is that the system gives accurate results. The system is also designed to be simple and user-friendly. For that, everyone can understand easily and the person who has very poor knowledge about tech and technology can also easily use it. The interface and navigation are very easy and user-friendly. Also, the performance is very important; we optimize the system for giving faster answers. The system is designed for scalability. That means the system can handle more users and more legal information. To protect the user's information, we used secure authentication and data encryption. Finally, to make the system cost-effective, we used efficient training methods like LoRA fine-tuning and quantization, which reduce costs and keep the AI model powerful. Based on all the requirements, the system is designed using a layered architecture. In the data layer the legal text is collected, cleaned and stored in JSON format. And the database is on PostgreSQL. In the model layer we used the LLaMA 3.1 (8B, 4-bit quantized) fine-tuned version. We train it by using Hugging Face frameworks. By this we make the AI application layer; the system was built in Next.js and Node.js and it was connected with AI services through Gemini and Ollama providers.

3.2.1 Overview

The requirement analysis and design plan give a strong base to make this kind of helpful and user-friendly AI legal assistant, which is made for the Bangladeshi people. The main goal of this project is to make the law simple and easy to understand for everyone. There are many people who find the laws and legal system difficult. This system helps them better understand and get clear answers to their questions. For doing this, the system is designed step by step. First, we collect the legal document from the Bangladesh Laws website and organize the collected data in a format that the computer can easily understand. These documents are the knowledge for the AI. After that the AI model is trained and fine-tuned so that it can easily understand Bangla, English and Banglish. So that people can ask questions in these 3 languages and they get the answers in a very simple way and accurately. This system has a web-based chatbot where people can talk to AI in real time. The chatbot is made to be fast, secure and easy to use. The system is designed to make users handle many users at the same time.

3.2.2 System Design

The system of the AI-powered legal assistant designed is made using layered architecture, that means the system works on different levels and all are connected. All the layers are handling data processing, model training and user interaction in the complete system. In the first level there is the data layer. In this part legal documents are collected from the official Bangladesh Laws website; for collecting all the data, we make a Python scraper to scrape all the laws. Then we clean and organize and filter the dataset. The dataset is saved in a special format like JSON and the PostgreSQL

database. This helps to arrange laws, acts, sections and clauses properly and for that, AI can quickly find the right information when someone asks any question. The top layer is the application layer. It connects the AI model to the end users. It is built as a web-based chatbot where people can ask about legal questions and get answers instantly. The frontend is made by using Next.js, React and Tailwind CSS. The backend is made by using Node.js and PostgreSQL. The system can store user history so users can check their previous queries. For keeping everything safe and authentication and handling using JWT and bcrypt. Which protect user accounts and private information. For the system uses Ollama and custom models to make the AI model work smoothly and quickly. Together these three layers the data layer, the model layer and the application layer create a strong and reliable system. When a user asks a question, the system receives the query and processes it through the AI model and then it gives back a clear and simple legal answer in real time. In Figure 3.2.1 we can see the system architecture of our system design.

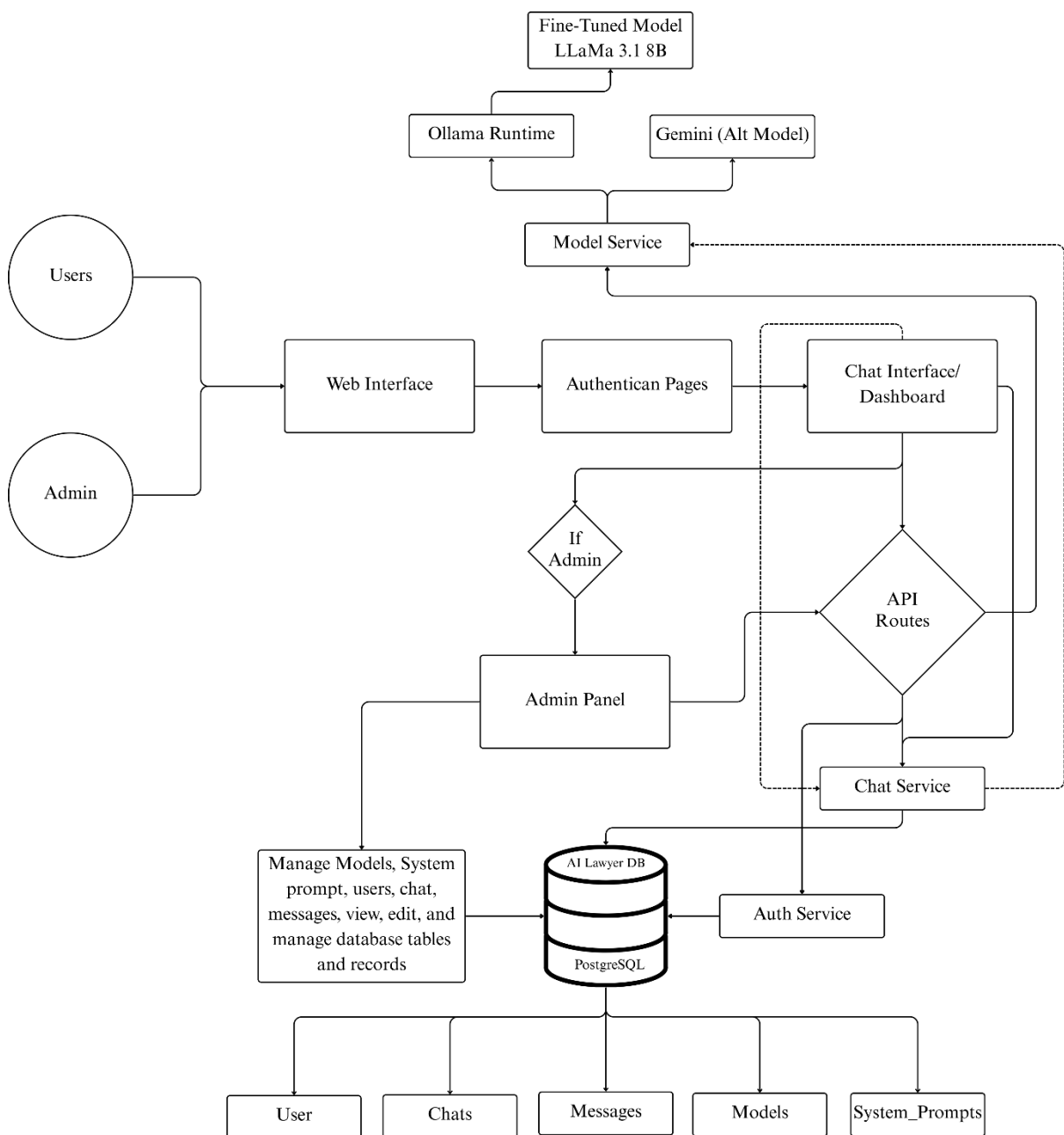


Figure 3.2.1: System Architecture

Figure 3.2.1 presents the end-to-end architecture of the AI-Powered Legal Assistant, organized in three layers Data, Model and Application and shows how a user query travels through the system and returns as a grounded legal answer.

3.2.2.1 Data Layer

Legal texts are fetched from the Bangladesh Laws portal via a Python scraper, then cleaned and parsed (HTML → structured JSON). The curated corpus is stored in a “Legal JSON Store” organized by volume, act, section and notes. User information, chat histories and model metadata live in a PostgreSQL database. This separation ensures that immutable legal source data and mutable application data are managed appropriately.

3.2.2.2 Model Layer

Fine-tuning is performed in a GPU environment (Colab & JarvisLabs.ai) using Transformers/TRL with LoRA on a LLaMA-class backbone, quantized for efficiency. The trained artifacts are pushed to a model registry (e.g., Hugging Face). For inference, the system supports two backends: (1) a local/runtime path (Ollama/llama.cpp with GGUF exports) for low-latency responses and cost control and (2) an alternate provider (e.g., Gemini) as a fallback or comparison path. This dual-provider setup gives reliability, A/B evaluation flexibility and predictable costs.

3.2.2.3 Model Layer

Users interact through a web UI (Next.js/React). Requests stream to a Node.js API, which first verifies identity and session with a lightweight auth service (JWT + hashed credentials). An AI Orchestrator component builds the prompt (including system instructions and, when needed, retrieved legal snippets), selects the model/provider (local vs. alternate) and streams generated tokens back to the client. The API persists conversation turns and telemetry in PostgreSQL to enable session continuity, auditability and later evaluation.

3.2.2.4 End to End Request Flow

The user opens the chat and submits a Bangla or English query. The frontend sends a streaming request to the backend & the backend validates the JWT. The orchestrator assembles the prompt (and may add retrieved context if retrieval is enabled), then routes it to the preferred inference engine (local Ollama by default or an alternate provider on fallback or policy). The model generates tokens the backend streams them to the browser for real-time UX. The final answer and metadata (latency, provider used) are saved to PostgreSQL.

3.2.2.5 Security & Privacy Controls

All transport is over HTTPS. Auth tokens are short-lived and signed passwords (for admin or registered users) are hashed. The chatbot does not require or store sensitive personal data for casual use; admin functions are role-gated. Logs exclude raw secrets and minimize personal information to respect privacy.

3.2.2.6 Scalability & Reliability

Stateless API routes plus a serverless/pooled Postgres tier allows horizontal scaling. The model layer is decoupled via a provider boundary, enabling blue-green model rollouts or quick failover. Catching frequent prompts and pre-warmed model weights reduces cold start latency.

3.2.2.7 Maintainability & Updatability

Since data ingestion and training are modular, legal corpus updates (new acts/sections) or model refreshes (new checkpoints/LoRA adapters) can be shipped independently. The app can switch models at runtime through configuration, without frontend changes.

3.2.2.8 Rationale for Design Choices

Layered boundaries keep legal content, ML logic and UI concerns isolated for clarity and safe iteration. Quantized LoRA models deliver strong responses on modest hardware, reducing cost. Dual inference backends balance control (local) and breadth (hosted), improving robustness. Streaming UX provides fast, conversational feedback and reduces perceived latency.

3.2.3 Functional and Nonfunctional Requirements

3.2.3.1 Functional Requirements

The functional requirement explains what features and operations AI Lawyer must have to work properly and achieve the goals. First, the AI lawyer must provide a multilingual chatbot interface so that it can understand and answer questions in both Bangla and English. Our AI Lawyer is very easy for those people who speak different languages. The chatbot has to be able to handle three main tasks. Those are answering legal questions, giving simple legal advice and predicting possible outcomes of basic legal cases. To make all these functions work, the system must collect, clean and organize the data, which is collected from the official Bangladesh Law portal. The data is structured in a special format that is JSON. Because AI understands the JSON format data, that AI model can use it for training and finding answers very easily. Also, it has an admin panel here where they can update the dataset, fine-tune the AI model and monitor user activities. The backend of the system is also strong and secure. It has user authentication so that only authorized users can access the system safely. It also needs to save the user history so that if a user needs to see his previous queries, they can.

3.2.3.2 Non-Functional Requirements

The non-functional requirements explain the AL Lawyer system quality features that make the system effective, trustworthy and long-lasting. For the system, one of the main things is accuracy and reliability, because people depend on this system to get the correct answer about the legal information. For achieving this, the AI model must be trained carefully using the authentic legal data from the official Bangladesh Laws website. Always check the chatbot to make sure the chatbot gives the correct answer. Another important part is usability. The system is made for the people with different educational levels and the interface must be easy and user-friendly so that people can use it easily without any problem. Also the performance is very important. The system uses LoRA and 4-bit quantization that make the AI model work fast. The system is designed to give

real-time answers. The user gets a response within seconds after asking a question. The system also must have to handle the scalability, which means the system has to be able to grow when more users start using it. For achieving this, the project uses a serverless PostgreSQL database. Security and privacy are also very important. The AI Lawyer system uses JWT-based authentication to verify the users; it uses strong password encryption to protect the account and secure communication methods to keep data safe.

3.2.4 Context Diagram

In this context diagram, we give a clear overview of how the AI Lawyer system works. We also give that how many parts are connected to it. The AI lawyer system acts like a main platform for providing legal help, which is located at the center of the diagram. On the other side there is a user interface with the system for asking their legal questions in Bangla, English and Banglish. The AI Lawyer system took the questions and processed them by the AI model, which we trained on Bangladeshi laws. The WI model works with a database that stores all the legal information and laws that we collected. By using the database store and AI model, it understands the user questions and answers according to the question. Apart from the user diagram, it shows the administrator role. The work of the admin is very sensitive because he is responsible for managing the system, updating the dataset, AI model updates, monitoring user activities and maintaining the overall system performance. In Figure 3.2.2 we can see the context diagram of our system design.

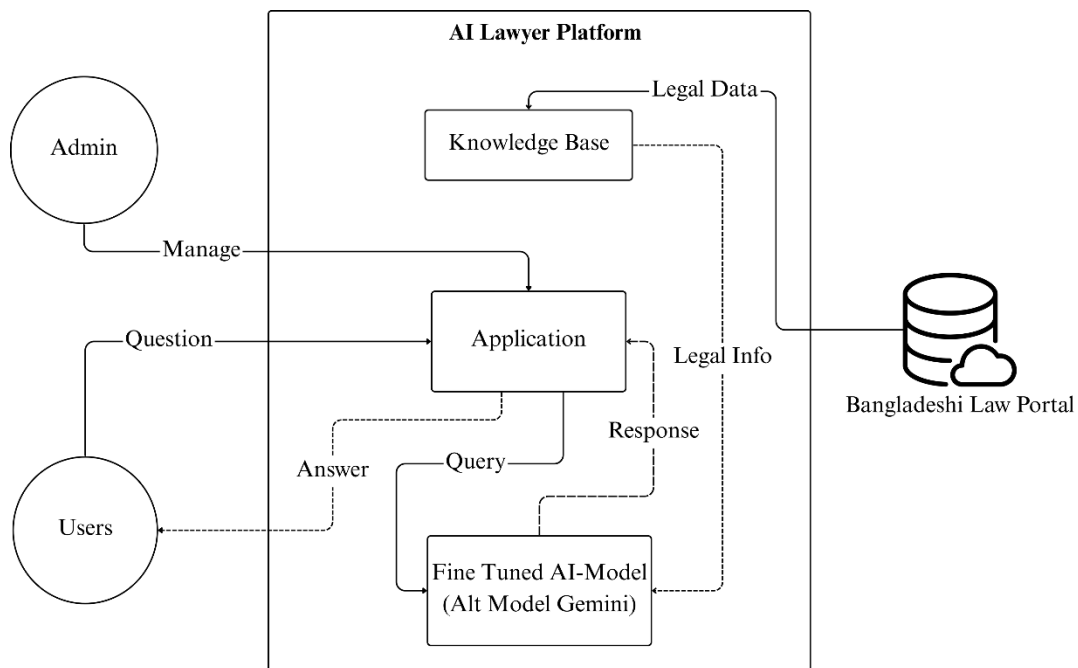


Figure 3.2.2: Context Diagram

3.2.5 Data Flow Diagram

The Data Flow Diagram (DFD) describes the step-by-step work of the AI Lawyer system. Also, it shows how the data travels inside the system. At first one user enters in the platform by registering or logging in. this process is handle by the User Registration and Authentication model. After the authentication, they become the authenticated user of

the AI Lawyer system. Our platform also stores the user information and their sessions. After the authentication, the user enters in the main part, which is Consultation and Interaction. Here users give queries and ask legal questions in Bangla, English and Banglish. After that, it processes those questions and queries. The questions go through the fine-tuned AI mode (Alt model Gemini) and are also supported with the legal knowledge. The answer that the system generates is stored as chat logs and analytics. Our platform also manages the case management. This is for organizing and tracking the user case. Overall the DFD shows how the AI Lawyer system manages the whole process. In Figure 3.2.3 we can see the data flow diagram of our system design.

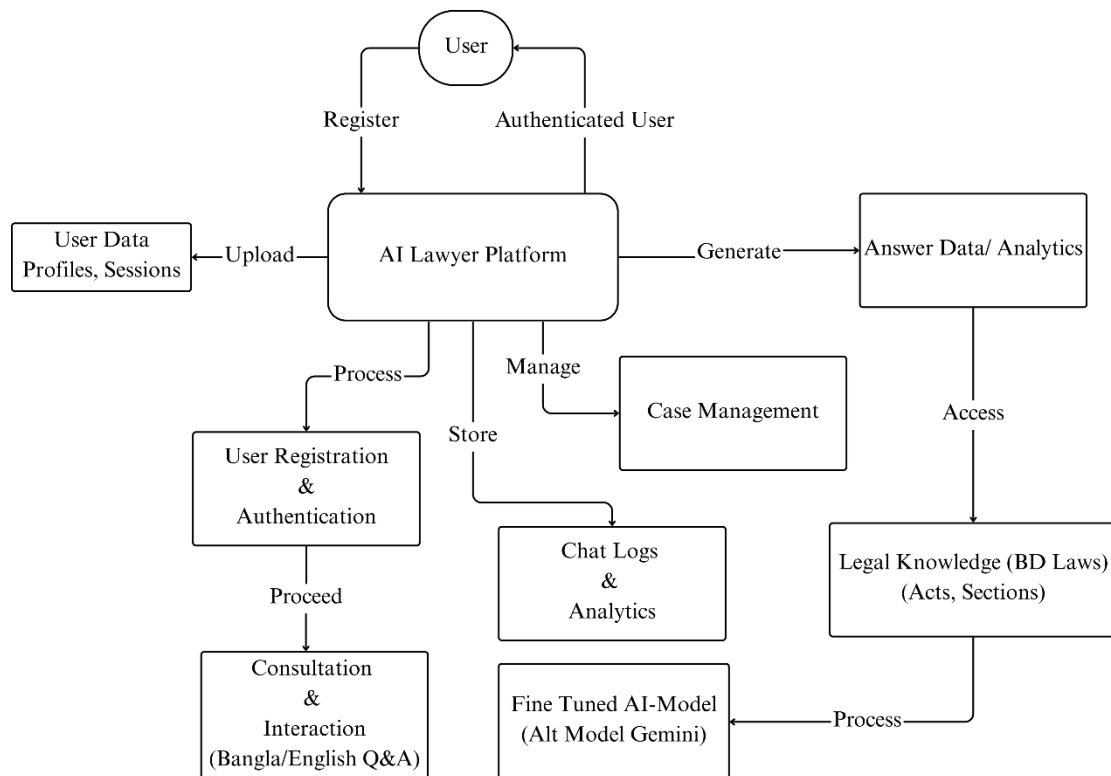


Figure 3.2.3: Data Flow Diagram

3.2.6 UI Design

The AI Lawyer system user interface is designed to be simple, easy to use and accessible. So that people from different social backgrounds and different educational backgrounds can use it without any problem. The system was built using modern web technologies like Next.js, React, Tailwind CSS and Radix UI components. All these tools make the design clean, responsive and accessible on all types of devices. The system interface is divided into many important pages. All the pages have a specific purpose. There is a landing page also where users come to learn about the system and a login page for existing users to sign in or for new users to create an account. There is also a chatbot page, where users can ask legal questions and they get the answers instantly. There is also an admin panel, where the admin can update the dataset and monitor the user.

3.1.6.1 Landing Page

The landing page is the first point of interaction for users. Its primary goal is to clearly communicate the purpose of the system, providing free, AI-powered legal assistance in Bangla and English. The design features a clean hero section with the project name, a short description and a prominent “Get Started” button. Below this, there are sections that highlight the key benefits of the system, such as 24/7 availability, bilingual support and free legal guidance. Icons and subtle animations are used to make the page visually appealing while maintaining simplicity.

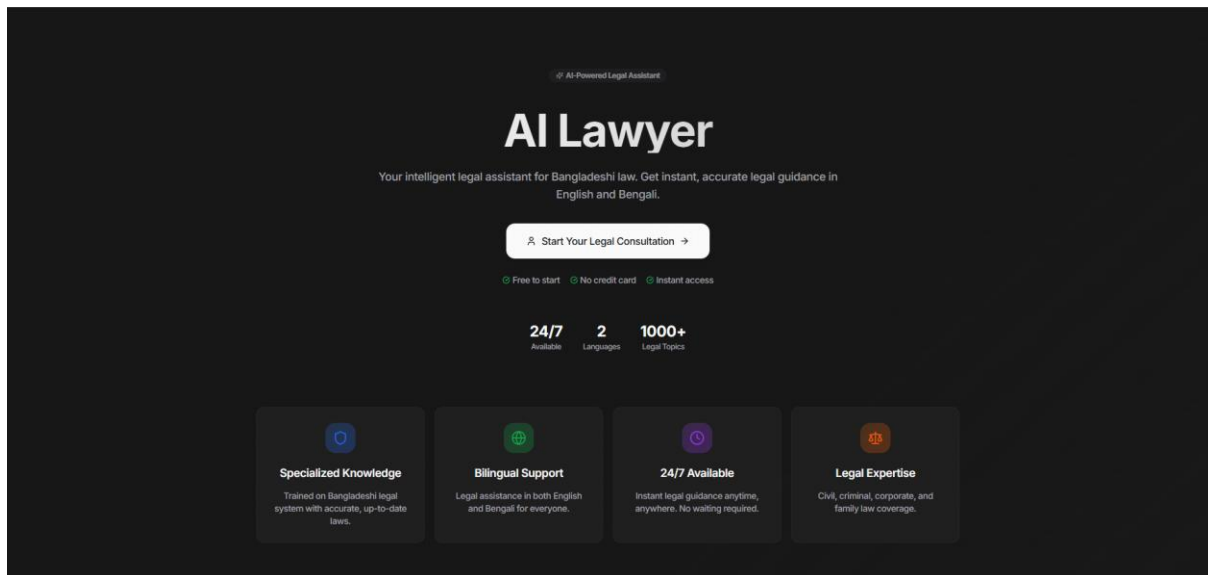


Figure 3.2.4: Landing Page

3.1.6.2 Registration Page

The AI Lawyer system registration page is designed for new users to create an account easily and quickly. To create a new account or sign up, users need to enter their full name, email address and password and confirm the password to make sure the password is correct. The page design is very simple and clean, with a simple style that makes it easy to focus on the form. All the input boxes are clearly labeled and it also includes the placeholder text to guide users on what to enter. There is a visibility toggle for the password part. So that users can decide that they want to see or hide the password when they are typing. At the bottom there is a Create Account button.

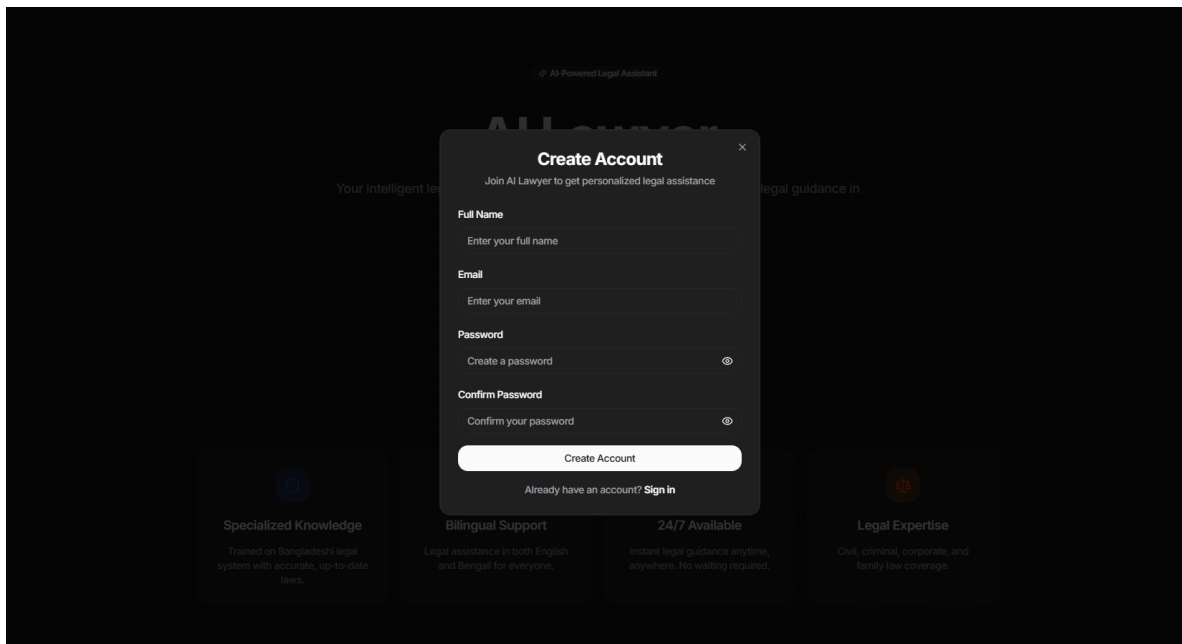


Figure 3.2.5: Registration Page

3.1.6.3 Login Page

To keep the users' accounts safe and provide a personalized experience, the AI Lawyer system login page is designed. It is a very simple input field where users enter their email and password to sign in to the system. The design of the page is very simple, with the login form placed in the center. If any users enter any wrong information, the system displays an error message to guide them. There is also a Forgot Password option with this, if any user forgets their password, they can recover their password. The page is responsive that means it works smoothly on all kinds of devices, like computers, tablets, mobile phones, etc.

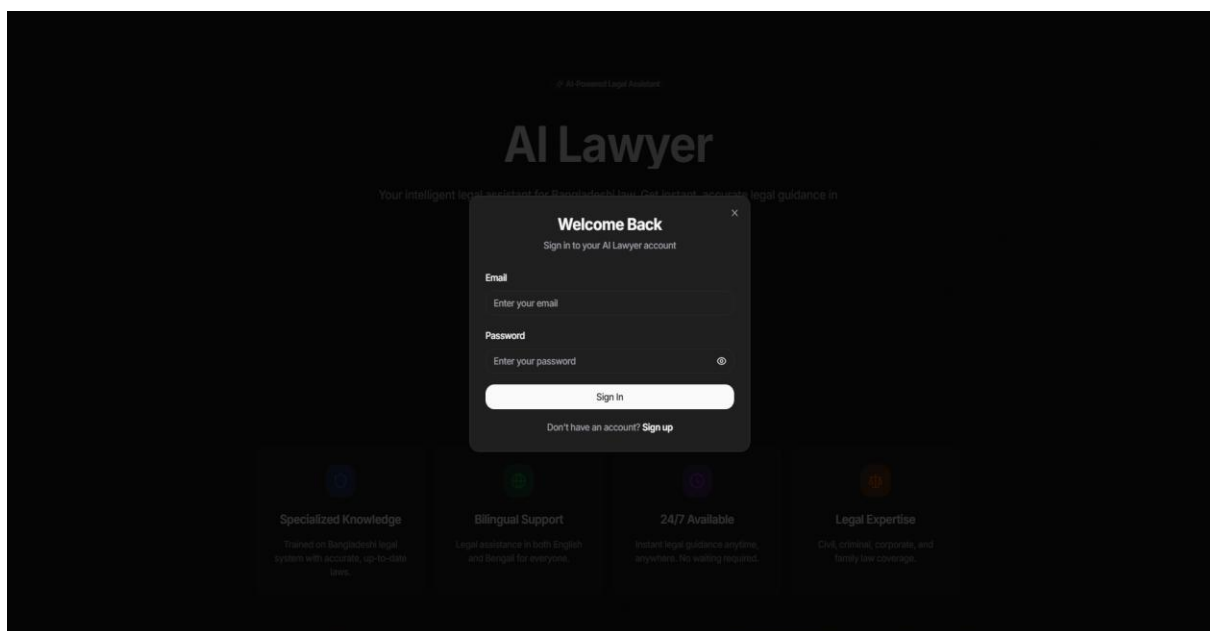


Figure 3.2.6: Login Page

3.1.6.4 Chatbot Interface

The main feature of my system is the chatbot interface. The AI Lawyer system is designed for giving users a smooth and interactive legal experience. The layout is designed with a dark theme, which makes it easy on the eye and it is very simple to use. There is a sidebar that shows the recent and previous chat history so users can easily go back to their previous queries that help them with the new chats. A personalized welcome message greets the user by showing the username at the center of the screen. The main input area is at the bottom of the screen, where the user gives their queries and questions and the answers are shown in a chat style.

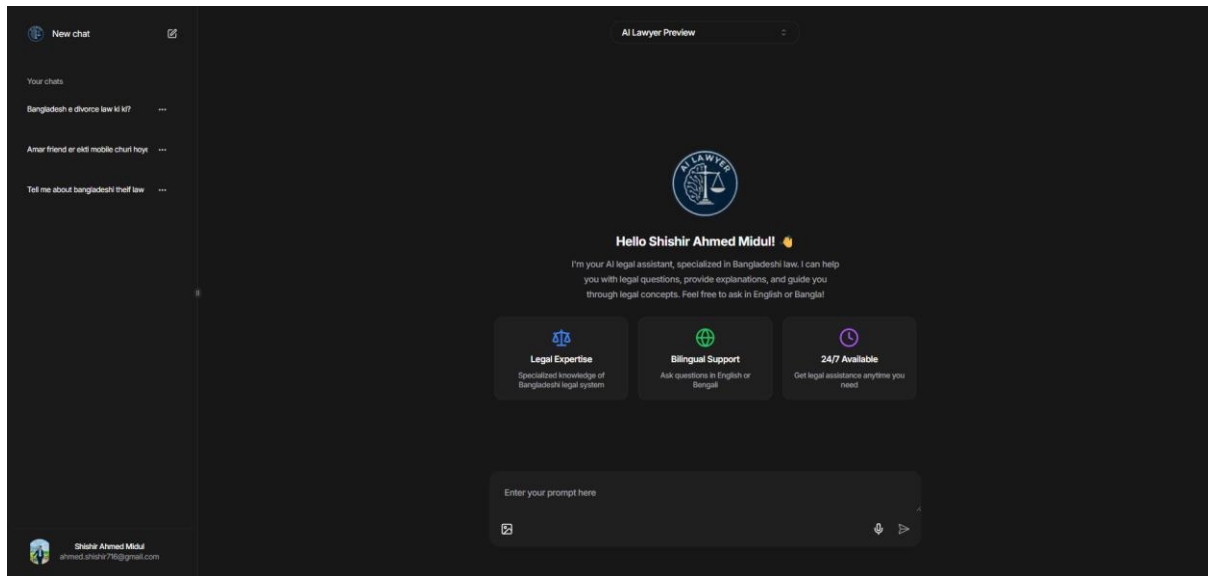


Figure 3.2.7: Chatbot Interface

3.1.6.5 Admin Panel

The admin panel provides system administrators with a secure interface to manage users, models, prompts and databases. In our design, once the AI Lawyer system is deployed, the first registered user automatically becomes the admin. The design is clean and intuitive, featuring a top navigation bar with tabs for different management sections. The interface highlights admin accounts with clear labels and provides quick action buttons for efficient control. Consistent with the overall dark theme, the panel maintains simplicity and ensures that critical administrative tasks can be carried out smoothly and securely.

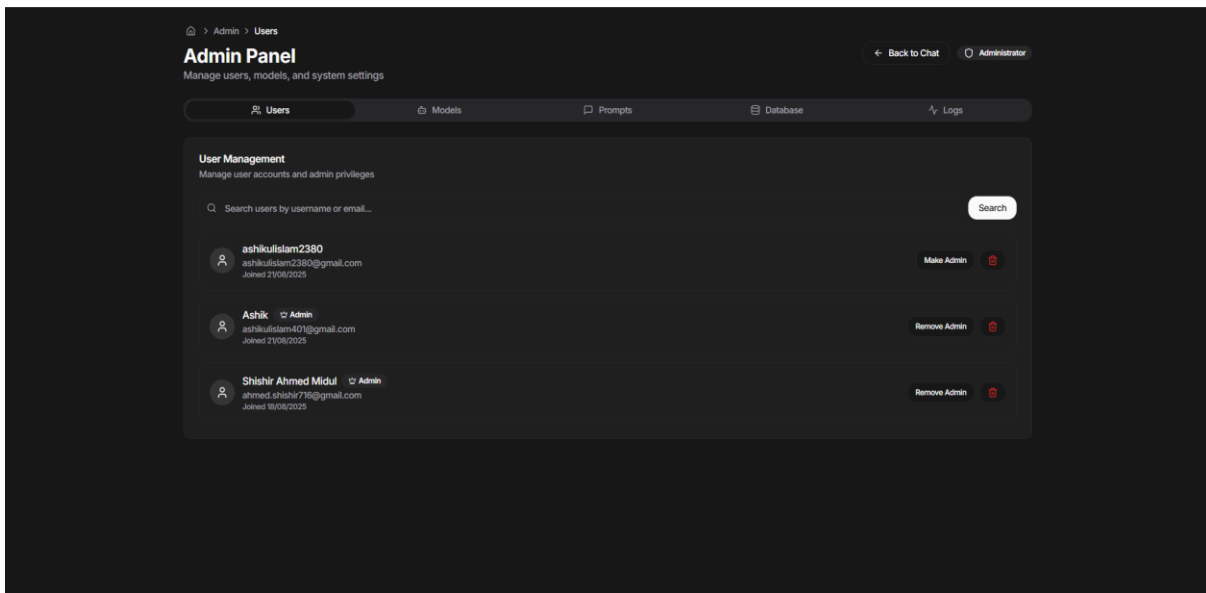


Figure 3.2.8: Admin Panel User Management

In Figure 3.2.8, the users list of the admin panel is shown. From this panel, the admin has full control over user management; they can search for users, remove accounts, grant admin rights to other users, or revoke existing admin privileges.

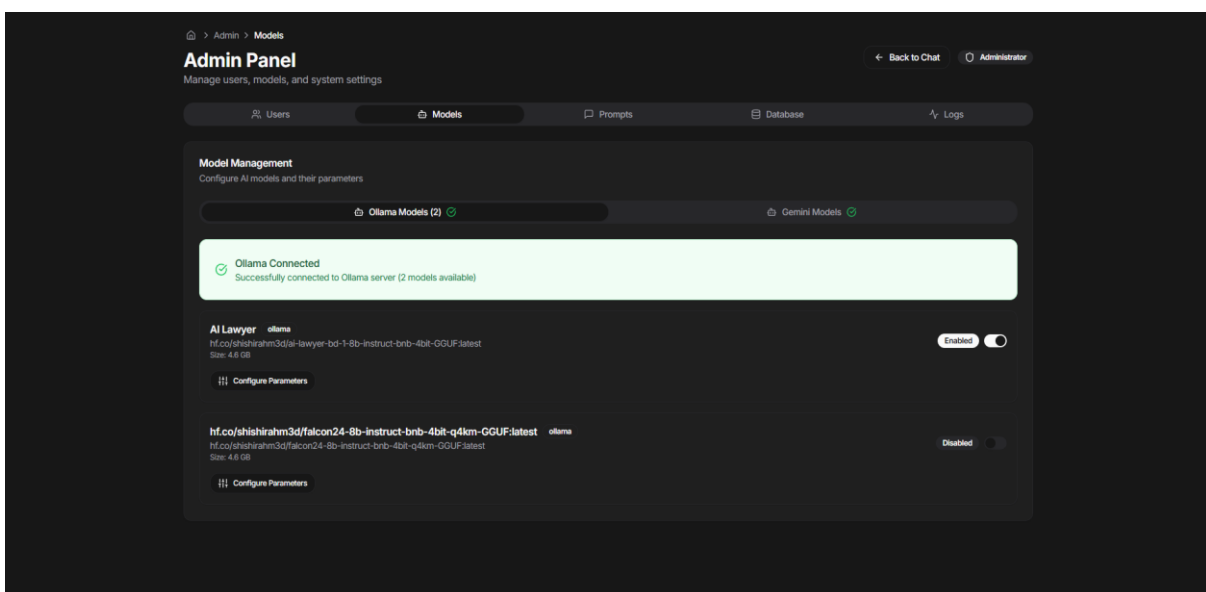


Figure 3.2.9: Model Management

In Figure 3.2.9, we can see the list of models that run on the backend through Ollama, retrieved using the Ollama API. From this panel, models can be turned on or off as needed.

Configure AI Lawyer
Adjust model parameters to fine-tune performance

Display Name: AI Lawyer Model Version: 1.5

Max Tokens: 2048 Temperature: 0.80

Top P: 0.90

Top K: 40

Frequency Penalty: 0.00

Presence Penalty: 0.00

Cancel Save Changes

Figure 3.2.10: Model Configuration

In Figure 3.2.10, the model configuration settings are shown, where we can adjust details such as the display name, version, maximum tokens, temperature, top-p, top-k, frequency penalty and presence penalty.

Admin Panel
Manage users, models, and system settings

Users Models Prompts Database Logs

System Prompts
Manage system prompts for each model

AI Lawyer Add Prompt

System Prompt Default v1
You are **AI Lawyer**, a highly specialized artificial intelligence trained exclusively on **Banglad...

hf.co/shishirahm3d/falcon24-8b-instruct-bnb-4bit-q4km-GGUF:latest Add Prompt

AI Lawyer Preview Add Prompt

System Prompt Default v1
You are **AI Lawyer**, a highly specialized artificial intelligence trained exclusively on **Banglad...

Figure 3.2.11: System Prompt Configuration

In Figure 3.2.11, we can see the list of models that are currently turned on from the model list and the system also allows us to set, update or customize the system prompt for each model.

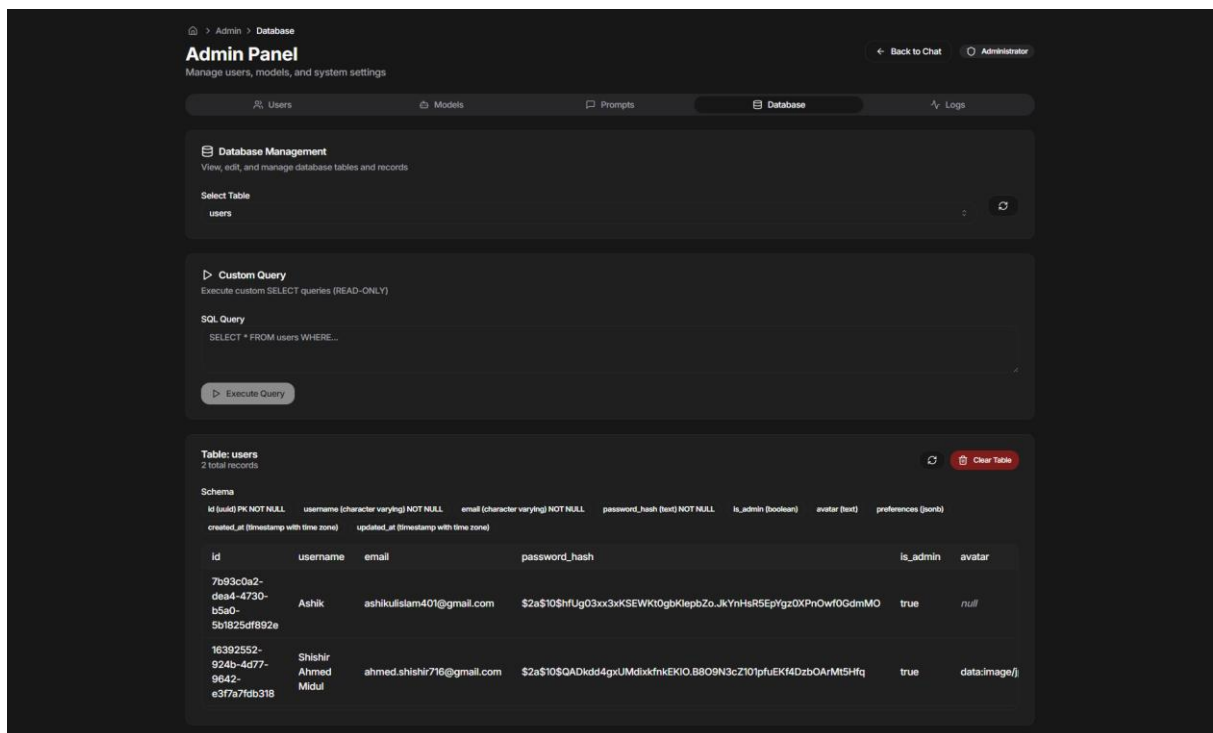


Figure 3.2.12: Database Management

In Figure 3.2.12, the Database Management section of the admin panel is shown. Here, the admin can select a specific table to view its data, with options to edit records, delete entire tables, or remove individual records as needed. Also, the panel supports running custom SQL queries (read-only) for data inspection and management. This ensures that administrators have full control over maintaining and monitoring the system's database.

Figure 3.2.13: Edit Record

In Figure 3.2.13, the detailed view of the record editing interface is shown. The admin can open any table, select a record and directly edit, update, or remove its fields. This feature allows full control over user information, roles and other database entries, ensuring that system data remains accurate and up to date.

3.3 Detailed Methodology and Design

The detailed methodology of our project follows a structured pipeline that combines data engineering, model training, evaluation and user interface development into a single coherent workflow. The overall design was carefully planned to ensure that each stage contributes directly to building a functional, reliable and user-friendly AI-powered legal assistant for Bangladeshi law. The first stage focused on data collection and preparation. We developed a Python-based web scraper using libraries such as BeautifulSoup and requests to extract legal content from the official Bangladesh Laws portal. The raw data was initially unstructured, with HTML tags, broken formatting and irregularities. To address this, we designed custom cleaning scripts and applied regular expressions to normalize the text. The laws were then organized into a JSON structure consisting of volumes, acts, sections and notes. Since volumes 1-26 were in English and volumes 27-56 were in Bangla, the dataset was inherently bilingual, which aligned with our objective of creating a multilingual assistant. We further enriched this dataset by creating question-answer pairs and scenario-based prompts (e.g., legal advice, case predictions), which were later used to fine-tune the model. The second stage involved data preprocessing and translation. We tokenized the dataset using Hugging Face tokenizers and ensured that both Bangla and English texts were properly segmented for training. In addition, a translation step was applied where needed to make sure that legal terms could be understood in both languages. This bilingual dataset provided the foundation for training a model that can seamlessly switch between Bangla and English queries. Model fine tuning and optimization is the third stage of the project. We chose LLaMA 3.1 8B as the base AI model for this step. To train the model more efficiently we used a special method called LoRA. We also used some very helpful libraries like Transformers, TRL, PEFT. Training a large AI model normally needs a lot of computer memory. For that we used a method that is called 4-bit quantization. It helps us to train a big model in limited hardware and low cost. In the training time we taught the model many skills like answering legal questions, giving simple legal advice predicting the case outcome. The work of the fourth stage is evaluation and testing. In this part we checked how the model was learning. We do it by measuring training loss and validation loss. We also used accuracy and F1-score test to see how well the mode performed. The system support Bangla, English and Banglish input so that we are test the system with all the language to make sure the answer that is coming it is correct. In the fifth stage we are doing system design and users interface integration. We built the backend by using Node.js and PostgreSQL. For the frontend was created with Next.js, React and Tailwind CSS. We designed the chatbot interface very simple, easy to use so that people can ask their legal question comfortably. The chatbot gives the answer in real time. The system used Ollama for fast local AI processing. For making everything carefully, the project followed a step-by-step development plan. In Phase 1, we are collecting the data and cleaning and organizing the data. In Phase 2 we fine-tune AI models and build the user interface. In Phase 3 we integrated the backend with the frontend. In Phase 4 we work on optimization, testing and evaluation. This step-by-step helps us to develop each part properly.

3.4 Project Plan

We develop the AI lawyer project step by step in an organized way to make sure everything is complete in the proper way. We divided the whole project into four main parts. Each part has a different task and goal. This plan helped the team manage time and use resources. In the first part we focus on collecting a dataset; for this we made a

Python scraper to scrape all the laws of Bangladesh from the official Bangladesh laws website. After scraping, we clean the data, filter the dataset and save the dataset in JSON format. Because AI understands JSON data easily. We arrange the dataset into volumes, acts, sections and notes. We also translated the data set into Bangla and English. So that our system can easily work in both languages. In the first part we made a complete, authentic and organized dataset to train the AI model. In the second part our focus is on training the AI model. For that we chose LLaMA 3.1 (8B 4-bit quantized) as our main model. We used the LoRA fine-tuning technique to make the model learn better. For the training we used Google Colab GPUs to make the process faster. Our AI model was trained in answering legal questions, giving legal advice and making simple predictions of case outcomes. In the third part we worked on building the full system and designing the user interface. So we created a web-based application. For making this application, we used Next.js, React and Tailwind CSS for the frontend. We build the backend by using Node.js and the PostgreSQL database. We build a chatbot interface where people can ask their question in Bangla, English, or Banglish and get the answers in chat format in real time. In the fourth and final step our focus was on testing, evaluation and optimization. We tested the system with Bangla, English and Banglish questions to make sure it worked in all these 3 languages. We also tested it by giving difficult legal questions and case outcomes to check if the AI handles difficult questions or not. We arranged the user testing part and the real users tested the system and gave us the feedback.

3.5 Task Allocation

We work as a team on this project. We make sure that every job is completed from time to time and we discuss every day how much was completed and what we will do the next day. We start our work by collecting the data set. Together we made a Python scraper to scrape all the laws from the BD law portal. After that we clean and organize our dataset. We need the dataset in Bangla and English. So, we divided the work: 1 person translated all laws into Bangla and another one translated them into English. For the model training we worked collaboratively and we individually trained some models and we found that the LLaMA 3.1 model gave the best answer and we decided to use the LLaMA 3.1 model for our fine-tuning. We also work like a team for developing the system. We firstly developed the frontend part by using Next.js, React and Tailwind CSS. After that we developed the backend with Node.js and the PostgreSQL database. We also did the testing and evaluation together as a team. We made difficult legal questions and cases to check whether the output is right or wrong. We also check if the system is correctly worked on in Bangla, English and Banglish in those languages or not. After that, we together prepare the project documentation and write the final report. In Table 3.5.1 we can see the estimated work period and actual work period that are needed to develop our system.

Estimated Work Period	
Actual Work Period	

Table 3.5.1: Task Allocation and Timeline

Tasks	Weeks																			
	1	4	5	8	9	10	12	15	16	18	19	24	25	27	28	32	33	35	36	
Data Collection & Cleaning																				
Dataset Translation & Preprocessing																				
Model Training, Fine-Tuning & Quantization																				
System Development (Backend + Frontend)																				
AI Integration (Ollama + Gemini)																				
Testing & Optimization																				
Finalization & Documentation																				

3.6 Summary

In this chapter, we discussed the methodology and design process that we used to develop the AI Lawyer system. Firstly, we discussed the requirements analysis. In this part we explained the functional and non-functional requirements. It helped to make the system practical, secure and easy for users. After that we explained the system designed and the used diagram that shows the layered architecture, data flow and the database management system. We also discussed the user interface in detail. In this part we talk properly about the landing page, login page, registration page, chatbot interface and the admin panel. We also described the complete methodology that we followed during the project. It fully covered the step-by-step workflow that includes data collection, data preprocessing, model training, evaluation and user interface. We also discussed our part with each other and two of us knew everything about the project. Overall, we explained how the AL Lawyer system was planned, designed and developed.

Chapter 4

Implementation and Results

In this chapter we explore how we implemented our system. We explain about environment setup and also the use of cloud GPUs for the model training. We also explain about the Ubuntu servers for deployment, tools, frameworks, data scraping, preprocessing and model optimization. We also tested whether the system could understand the Bangla, English and Banglish queries or not.

4.1 Environment Setup

The implementation of the AI Lawyer system required setting up a reliable and well-structured environment that could support every stage of the project, from training large models to running the final web application in production. Since our project depended heavily on large language models, we needed access to powerful GPU resources for fine-tuning and experimentation. At the same time, we also had to prepare tools for dataset collection and preprocessing so that the raw legal texts could be transformed into a format suitable for machine learning. In addition to this, building a complete application meant that we required a modern web technology stack to design the user interface, manage user authentication and connect with the inference engine. Finally, for real-world deployment, we wanted an environment that we could control ourselves rather than depending only on external services. To achieve this, we adopted a hybrid approach that combined cloud GPU services with a self-hosted deployment setup. On one side, we used platforms like Google Colab and Jarvislabs.ai to carry out training and fine-tuning of the model using high-end GPUs such as NVIDIA T4, V100 and A6000. These cloud GPUs gave us the computational power needed for working with billions of parameters while still being cost-effective for a student project. On the other hand, once the training process was complete, we shifted our focus to deployment. For hosting the final system, we set up our own Ubuntu 24.04 server, which gave us full control over how the backend, frontend and database were configured. To manage applications efficiently, we used Dokploy as the deployment manager, Traefik as the reverse proxy to handle requests and Let's Encrypt SSL certificates and routing. PostgreSQL was also hosted directly on the same server, ensuring a secure and stable environment for storing user data and chat history. By combining GPU-based cloud training with a self-managed hosting environment, we were able to strike a balance between performance, flexibility and control. This setup allowed us to train efficiently, prepare our data with the right tools and then deploy a production-ready application that we could manage ourselves without relying entirely on third-party infrastructure.

4.1.1 Dataset Preparation Tools

- BeautifulSoup and Requests - scraping Bangladesh Laws portal.
- Regex-based parsing - cleaning HTML and normalizing text.
- Pandas and NumPy - dataset structuring.
- SentencePiece - tokenization for Bangla/English texts.
- Protobuf + Hugging Face Datasets - for efficient serialization and loading.

4.1.2 Training Environment

- Google Colab Pro with NVIDIA T4 and V100 GPUs for early experiments and baseline model training.
- Jarvislabs.ai with rented NVIDIA A6000 GPUs for large-scale training and fine-tuning, providing higher VRAM and faster throughput.
- Python 3.11 runtime with CUDA for GPU acceleration.
- **Core ML libraries:**
 - PyTorch - primary deep learning framework.
 - Hugging Face Transformers (v4.51.3) - for model loading, tokenization and training.
 - TRL (v0.15.2) - supervised fine-tuning with SFTTrainer.
 - PEFT - for LoRA adapter integration.
 - Unsloth - optimized fine-tuning wrapper for LLaMA models.
 - bitsandbytes - for 4-bit quantization and 8-bit optimizers.
 - xFormers and Triton - for memory-efficient attention and GPU kernel optimization.

4.1.3 Application Development Environment

- Frontend: Next.js (v15), React (v18), TypeScript, Tailwind CSS, Radix UI, Framer Motion for a modern responsive UI.
- Backend: Node.js runtime with REST APIs, AI orchestration using Vercel AI SDK.
- Database: PostgreSQL server hosted on Ubuntu.
- Authentication & Security: JWT-based login, bcrypt password hashing, Zod for input validation.
- AI Inference Integration:
 - Ollama runtime for local quantized inference (GGUF models).
 - Google Gemini API as a fallback provider.

4.1.4 Deployment & Hosting

- Ubuntu 24.04 LTS server as the hosting platform.
- Dokploy used as the deployment manager for applications, enabling containerized deployments and simplified CI/CD pipelines.
- Traefik configured as a reverse proxy for load balancing, Let's Encrypt SSL termination and domain routing.
- PostgreSQL hosted directly on the same Ubuntu server for user data, chat history and model metadata.
- Environment variables managed through Dokploy and dotenv for secure key handling.
- GitHub used for version control and automated deployments.

4.2 Testing and Evaluation/Performance/Comparative Analysis

After we finished our AI lawyer system development, the system worked properly. Our main goal of this stage is to check how well the AI model gave real user questions and whether the answers to these questions are correct or wrong. We test the system by giving questions in Bangla, English and Banglish. We did it step by step.

First we did the functional testing. We tested the system by asking different types of legal questions and telling the chatbot to give me the answer in a simple way. For deployment, we focused on the testing and evaluation part to make sure. By this we confirm whether the chatbot understands the user input and user needs or not. We also tested our system by giving edge cases. We give some special and difficult questions. Like, we ask with a long question and we write the question with a mix of Bangla, English and Banglish and we request the chatbot to give the answer. We saw that most of the time the chatbot understood that question and gave the correct answer. We also checked it by giving spelling mistake words and we saw that it worked properly. In the performance evaluation we measured the training loss, validation loss, accuracy and F1-score. The number shows that the AI model was not overfitting data and it was improving steadily. For using the Ollama to run the AI locally, we saw that the chatbot gave answers instantly. We also compared our fine-tuned model with an existing pretrained model that was not trained on Bangladeshi legal data and we saw that the pretrained model did not give the correct answers for the Bangladeshi legal issue.

Overall, the testing and evaluation phase showed that our AI Lawyer system is functional, accurate and reliable for its intended purpose.

4.2.1 Training and Performance Evaluation (W&B)

To test our AI Lawyer System performance, we measured training loss, validation loss, accuracy and F1-score. To tracking behavior. We also used Weights & Biases (W&B), its generate several performance curves.



Figure 4.2.1: Training Loss Curve

In Figure 4.2.1 The training loss decreased steadily across the training steps, indicating effective learning from the dataset. The smooth downward trend without sudden spikes shows that the optimization process was stable.

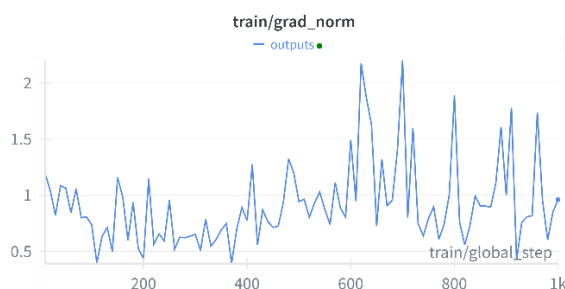


Figure 4.2.2: Gradient Norm Curve

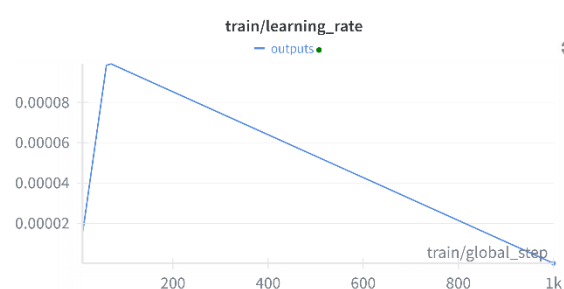


Figure 4.2.3: Learning Rate Schedule

In Figure 4.2.2 & Figure 4.2.3 The gradient norm curve shows that the model maintained stable updates throughout training without exploding or vanishing gradients. The learning rate schedule follows a warm-up and decay pattern, which contributed to smooth convergence. These two figures together confirm that the training was well-regulated and optimized.

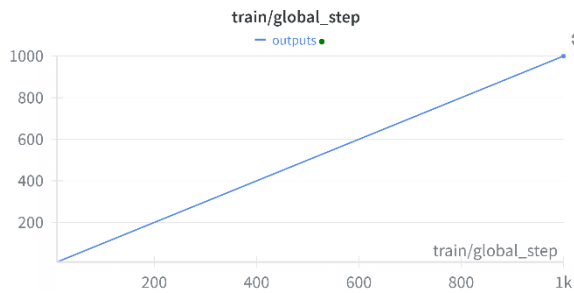


Figure 4.2.4: Global Step Progression

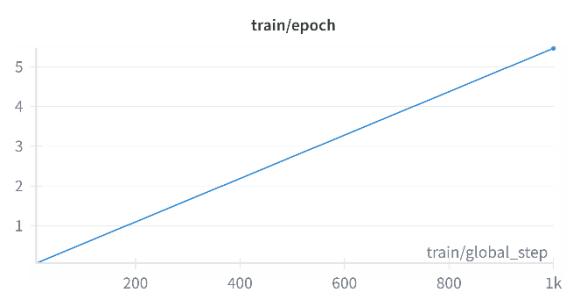


Figure 4.2.5: Epoch Progression

In Figure 4.2.4 & Figure 4.2.5 The progress plots show that the model completed all scheduled training steps and epochs successfully. This verifies that the planned training schedule five epochs was executed without interruption.

4.3 Results and Discussion

The implementation of our AI Lawyer system gave a positive result. It shows that the system is technically possible and it is practically useful. After being fine-tuned by the LLaMA 3.1 8B model using LoRA and 4-bit quantization on our dataset, the system was able to give legal question answers accurately in both Bangla and English. In the training process we also checked the loss curve and we saw a steady downward trend; that means the model was learning well from the legal dataset. The validation loss is also stable. When we test the chatbot by giving real-life legal questions, it gives the answer correctly with the laws. For example, when we asked about the age of marriage and the section of the penal code, we saw that the chatbot gave the correct information and explained it in a very simple way. We try to get the higher accuracy of your fine-tuned model from the base model. Another important result was the speed of the system; it responded in a second. We developed it by using Ollama for the local interface. From the user side the feedback was important and we give it to test some people and 1 lawyer. They gave us a positive review on it. They also like that the chatbot supports Bangla, English and Banglish input and the output comes in both Bangla and English. In Table 4.3.1, we've put some samples from our AI Lawyer system user input and output.

Table 4.3.1: Sample from AI Lawyer Corpus (legal questions and answers)

Input	Output
What should I do legally if my mobile phone is stolen?	If your mobile phone is stolen in Bangladesh, you should immediately file a General Diary (GD) with the local police station. Provide details of the phone and the incident. You can also report the IMEI number to your service provider to block the device. This helps in tracking and preventing misuse.
How can I file an FIR for a stolen mobile?	To file an FIR for a stolen mobile in Bangladesh, you need to report to the nearest police station. Provide a written complaint detailing the incident, including the phone's model, IMEI number and when and where it was stolen. The police will then register the FIR based on your complaint.
স্ত্রীর বিরুদ্ধে তালাকের কেস করার আইনি প্রক্রিয়া কী? কোর্টে কেস করতে হলে কী কী ডকুমেন্ট লাগবে, কতদিন সময় লাগে, এবং খোরপোষ বা দেনমোহরের বিষয়গুলো কীভাবে ম্যানেজ করতে হয়?	স্বামী কর্তৃক স্ত্রীকে তালাক দেওয়ার জন্য পারিবারিক আদালতে মামলা করতে হয়। মামলার সাথে বিবাহ নিবন্ধন সার্টিফিকেট, পরিচয়পত্র, এবং তালাকের নোটিশ জমা দিতে হবে। মামলার সময়কাল ও খোরপোষ/দেনমোহর আদালতের বিবেচনার উপর নির্ভর করে। একজন আইনজীবীর সাথে পরামর্শ করলে পুরো প্রক্রিয়াটি সহজ হবে।
Land er registration jodi thik na hoy, legal step ki nite pari?	Land registration thik na hole, apnar prothom kaj hobe ekjon bikhyato lawyer er shathe jogajog kora. Tini apnake court e suit file korte shahajjo korte paren ba jomira jotil shomoshshar somadhan korte parben. Shob documnet er copy, jemon dalil, uthal, o porichoyptro shonge nin.
Amar jomir dolil, khatian, porcha, ebong mutation certificate sob thik ache. Kintu porer bari'r ekjon amar jomir ek ongsho dokhol kore bari banache. Ami boleo tara dokhol chere dite chayna. Erom obosthay jomi ferot anar jonno kon court e case korte hobe, ki ki documents mandatory, ebong process ta koto din somoy lage?	If someone has encroached upon a part of your land and built a house, despite your valid land documents (deed, khatian, porcha and mutation certificate), you should file a suit for declaration and recovery of possession in the Civil Court. The mandatory documents you must submit include your original deed (dalil), khatian, porcha, mutation certificate and any evidence of encroachment such as photographs, witness statements, or a survey report. The entire legal process can be lengthy, often taking anywhere from several months to over a year or more, depending on court procedures, the availability of judges and the complexity of the evidence presented. It is advisable to consult with a local lawyer for specific guidance.

4.4 Summary

In this chapter, we described the implementation process and presented the results of our AI Lawyer system. We started by explaining the environment setup, where cloud GPUs from Google Colab and Jarvislabs.ai were used for training and a self-hosted Ubuntu server with Dokploy, Traefik & Let's Encrypt was used for deployment and database hosting. Next, we discussed the testing and evaluation phase, which showed that the system could handle Bangla and English legal queries with good accuracy, fast response time and better performance than baseline models. The results section highlighted how the fine-tuned LLaMA-3.1 model was able to simplify complex legal information into clear and understandable answers. User feedback confirmed that the chatbot was easy to use and provided valuable support, although some challenges remain with broad or ambiguous queries. Overall, the implementation phase proved that the system is technically sound, practical and capable of making legal information more accessible. This chapter has therefore demonstrated that the project objectives were achieved, setting the stage for the next chapter, where we evaluate engineering standards, design challenges and the wider impact of the system.

Chapter 5

Engineering Standards and Design Challenges

This chapter examines the project from an engineering perspective, highlighting the standards followed, the challenges encountered and the complexity of the problem. It explains how the system complies with software, hardware and communication standards to ensure reliability and security. The social, environmental and ethical impacts of the system are also considered, showing how it benefits society while remaining responsible. The chapter includes project management and financial planning to present the costs involved in professional deployment.

5.1 Compliance with the Standards

For this project, we made sure to follow important engineering and software standards so that the AI Lawyer system is not only functional but also reliable, secure and maintainable. While designing and implementing the system, we considered three main areas of compliance: software standards, hardware standards and communication standards. On the software side, we used widely accepted frameworks and libraries such as PyTorch, Hugging Face Transformers, Next.js and Node.js, which are all built on industry best practices. For coding standards, we followed clean code conventions and applied validation tools like ESLint and Zod to make the codebase more structured and error-free. Security standards were also considered, with JWT for authentication, bcrypt for password hashing and HTTPS connections managed through Traefik, which ensured that user data remained protected. In terms of hardware standards, we relied on GPUs provided by Google Colab and Jarvislabs.ai (T4, V100 and A6000). These GPUs meet modern computing standards for deep learning workloads and allow us to fine-tune large-scale models efficiently. For deployment, we hosted the system on an Ubuntu 24.04 server, which follows stable Linux distribution guidelines and ensures compatibility with PostgreSQL, Docker-based deployments and modern reverse proxies like Traefik. Finally, for communication standards, we ensured that the system followed RESTful API design principles for frontend-backend interaction, making the architecture modular and easy to extend. All API calls were secured and WebSocket/streaming responses were enabled for real-time chatbot interaction. This ensures that communication between users, the server and the model inference engine is efficient and secure. By following these standards, we made sure that the AI Lawyer system was not just a working prototype but also aligned with good engineering practices. This compliance gives the project a more professional foundation and makes it easier to scale, maintain and upgrade in the future.

5.1.1 Software Standards

In the development of the AI Lawyer system, software standards were a crucial part of ensuring that the project was not only functional but also reliable, secure and

maintainable in the long run. Every choice, from the frameworks and libraries we adopted to the coding practices, validation methods and deployment standards, was made to align with widely recognized industry practices. Below, the main software standards we followed are explained in detail.

5.1.1.1 Frameworks and Libraries

To build the AI Lawyer system, we selected modern and widely accepted frameworks and libraries that follow industry best practices. These ensured stability, scalability and compatibility across both machine learning and web development.

- PyTorch was chosen as the primary deep learning framework. It is an industry-standard tool for developing machine learning models because of its flexibility, GPU acceleration and large developer community. Using PyTorch ensured that our training pipeline would be stable, scalable and compatible with other tools.
- Hugging Face Transformers was used for model loading, tokenization and training support. This library provides standardized APIs for large language models, making the system interoperable with a wide range of pretrained models and datasets.
- Parameter-Efficient Fine-Tuning (PEFT) with LoRA via Unsloth was applied to reduce computational costs. Instead of retraining all model parameters, LoRA fine-tunes only selected components, following modern standards in efficient AI development. This made our system both cost-effective and energy efficient.
- Next.js and React were adopted for the frontend. Next.js follows modern web engineering standards by offering server-side rendering, static site generation and modular component design. Combined with React, this ensured that our user interface was responsive, reusable and compatible with accessibility guidelines.
- Node.js was used for backend APIs, following standard RESTful service architecture. This choice aligns with industry practices for scalable, event-driven backend services.

5.1.1.2 Frameworks and Libraries

To make the system reliable and maintainable, we followed modern coding and development standards. These practices ensured that our codebase was clean, consistent and less prone to errors during both development and deployment.

- Clean Code Principles were followed by writing modular, reusable and well-structured functions. Consistent naming conventions, indentation and proper documentation were maintained to improve readability and long-term maintainability.
- ESLint was used to enforce coding style and automatically detect syntax or formatting issues in JavaScript and TypeScript code. This reduced common mistakes and ensured a uniform coding style across the project.
- TypeScript was adopted for frontend development to provide static type checking, which helps in preventing runtime errors and increases code reliability. This made the application more stable and easier to debug.
- Zod Validation was applied to validate incoming data against predefined schemas. By enforcing strict rules on user inputs and API requests, Zod ensured data consistency and reduced errors caused by invalid inputs.

- Version Control with GitHub was used to manage collaborative coding, enabling proper branching, commit tracking and rollback if needed. This practice aligns with modern software development workflows.

5.1.1.3 Security Standards

Since the AI Lawyer system deals with sensitive user data and authentication, we followed strict security standards to ensure privacy, confidentiality and protection against common threats. These measures were applied across both the frontend and backend.

- JWT (JSON Web Tokens) were implemented for secure user authentication and session management. This ensured that only authorized users could access protected resources while keeping sessions stateless and encrypted.
- bcrypt Hashing was used for password encryption. Instead of storing raw passwords, bcrypt generates salted hashes, making it extremely difficult for attackers to retrieve original credentials even if the database is compromised.
- HTTPS Enforcement was applied through Traefik reverse proxy & Let's Encrypt SSL, ensuring that all communication between the client and server is encrypted and secure against interception or tampering.
- Input Validation and Sanitization was performed on both frontend and backend to protect against SQL injection, cross-site scripting (XSS) and other injection-based attacks. This prevented malicious input from affecting the system.
- Access Control mechanisms were included in the admin panel, where roles and permissions were clearly defined. This ensured that only authorized users could modify sensitive system configurations or manage data.

5.1.1.4 Deployment and Reliability Standards

To make sure the AI Lawyer system could run smoothly in production, we followed deployment and reliability standards. These practices ensured that the application was consistent across environments, easy to update and capable of handling future growth.

- Docker Containers were used to package the application into isolated environments. This guaranteed that the software would run the same way on any machine, eliminating compatibility issues between development and production.
- Dokploy Deployment Manager was used to automate builds, deployments and updates. This followed CI/CD best practices, making the release process faster, repeatable and less error prone.
- Traefik Reverse Proxy was configured for routing, Let's Encrypt SSL certificate management and load balancing. This ensured that all services were reachable securely and efficiently.
- Logging and Monitoring were integrated into the backend to track system performance, errors and user activity. This allowed us to quickly identify and resolve issues, improving system reliability.
- Modular and Scalable Architecture was followed, meaning components such as

frontend, backend, model inference and database were developed independently. This design makes the system easier to maintain and allows new features to be added without affecting existing functionality.

5.1.2 Hardware Standards

In order to train, deploy and operate the AI Lawyer system effectively, we relied on hardware that complies with modern computing and AI engineering standards. Using reliable GPUs, stable server infrastructure and secure hosting practices ensured the system could handle large-scale training while maintaining consistent performance during deployment.

5.1.2.1 Training Hardware

To fine-tune the LLaMA-3 model, we used cloud GPU platforms that provided enterprise-grade hardware resources suitable for AI workloads.

- Google Colab GPUs (T4, V100) were used during the initial experiments and baseline training. These GPUs are CUDA-enabled and widely adopted in academic and research environments, meeting modern standards for machine learning research.
- Jarvislabs.ai NVIDIA A6000 GPUs were rented for heavy fine-tuning tasks. The A6000 provides 48GB VRAM, high bandwidth and optimized CUDA cores, making it ideal for training large models efficiently. Using such industry-grade GPUs ensured our training pipeline was both stable and cost-effective.

5.1.2.2 Training Hardware

For production deployment, we hosted the system on a stable Linux environment that met server-level reliability and security requirements.

- Ubuntu 24.04 LTS Server was chosen for hosting the backend, frontend and PostgreSQL database. As a long-term support distribution, Ubuntu follows strict Linux standards for security, stability and compatibility.
- Direct PostgreSQL Hosting was set up on the same server instead of using a third-party provider. This provided us with full administrative control, compliance with data security standards and reduced latency for database queries.
- Dokploy with Traefik was used for deployment management and reverse proxying. This setup follows containerization standards, making the system easier to manage and ensuring reliable routing, Let's Encrypt SSL and load balancing.

5.1.2.2 Training Hardware

Apart from GPUs and servers, additional infrastructure standards were followed to guarantee secure and efficient operations.

- Dockerized Infrastructure ensured services were isolated and hardware resources could be allocated efficiently. This followed modern DevOps standards

and minimized environment-specific compatibility issues.

- Networking Standards were maintained by enforcing SSL/TLS through Traefik, guaranteeing secure communication between users and the hosted server.
- Scalability Readiness was built into the server setup, ensuring that additional compute resources or GPU instances could be integrated in the future without major redesign.

5.1.3 Communication Standards

Communication standards were essential in designing the AI Lawyer system because the application depends heavily on smooth, secure and reliable interaction between the frontend, backend, inference engine and database. By following industry-standard practices for APIs, data exchange and security, we ensured that the system could deliver real-time responses and maintain user trust.

5.1.3.1 API Design Standards

We structured the application's communication based on RESTful API principles, which are widely recognized in modern web development.

- RESTful Architecture was applied for frontend-backend communication, making the system modular and scalable.
- Standard HTTP Methods (GET, POST, PUT, DELETE) were used for CRUD operations, ensuring consistency with established web API standards.
- JSON Format was used for all request and response payloads, as it is lightweight, human-readable and widely supported.
- Error Handling followed conventional HTTP status codes (e.g., 200 OK, 400 Bad Request, 401 Unauthorized, 500 Server Error), making debugging and client-server communication more transparent.

5.1.3.2 Security in Communication

To maintain secure communication, industry-accepted encryption and authentication methods were implemented.

- HTTPS Enforcement was applied using Traefik reverse proxy, ensuring all traffic between users and the server was encrypted with Let's Encrypt SSL/TLS certificates.
- Authentication Tokens (JWT) were transmitted securely in headers, preventing unauthorized access to protected routes.
- Input Sanitization was enforced to prevent malicious data from passing through APIs. This reduced risks of injection attacks and ensured that only valid requests were processed.
- Role-Based Access Control (RBAC) was used in admin routes, ensuring that sensitive system operations could only be accessed by authorized administrators.

5.1.3.3 Real-Time Interaction Standards

Database communication followed strict relational standards to ensure reliability, security and integrity of stored information.

- PostgreSQL Relational Standards were maintained, with properly defined schemas, foreign keys and indexing for efficient queries.
- Parameterized Queries were used to prevent SQL injection and ensure safe communication between the backend and the database.
- Data Consistency Rules were applied so that user records, chat histories and model configurations were always stored and retrieved in a standardized format.
- Backup and Recovery Mechanisms were enabled on the Ubuntu server to maintain data integrity even in the event of failure.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

Our project will make life easier and less stressful for many people. At present a large number of people in Bangladesh do not know their basic legal right or cannot afford to pay a lawyer our AL legal assistant will provide them with free simple and clear legal advice in both Bangla and English. This means they can get help anytime without worrying about cost. With quick and reliable answers people will feel less confused and less worried when facing legal problems. They will also gain confidence to handle their own legal matters instead of dependency fully on costly services. This tool will help to create a society when everyone knows their rights and can stand up for themselves. This will make life fairer, more equal and more secure for all people.

5.2.2 Impact on Society & Environment

Impact on Society:

Our project will have strong and positive impact on society. The AI legal assistant will make legal help easy, free and always available, especially for people in villages who cannot afford expensive lawyers. It will provide clear answers in both Bangla and English so even people with little education can understand their legal rights. This will make the legal system fairer and more accessible for everyone. It will help reduce the fear and confusion people often feel when dealing with legal matters. People will be able to ask questions freely without hesitation. It will also save time and money they did not need to travel to find a lawyer for basic advice.

Impact on the Environment:

The environmental impact of our project is very small compared to traditional systems, since the assistant works online people do not need to use paper documents or travel to office which helps reduce pollution. The system also uses cloud servers which are designed to be energy-efficient and reduce the need for many physical servers. By avoiding paper-based work our project will also help in saving trees and reducing waste. Less transportation means less fuel use. This way our project supports a greener and cleaner environment even though its main goal is legal support. It is a digital first solution that reduces unnecessary environmental impact.

5.2.3 Ethical Aspects

In our project we followed ethical rules so that the system is safe, fair and honest for everyone. The first point is privacy. The chatbot will not ask for sensitive personal details and it will not store or misuse any information. This allows people to ask their legal questions without fear. Another point is accuracy. The chatbot gives correct and helpful legal information, but we make it clear that it is not a real lawyer. It is only for basic legal guidance not for official legal representation. We also ensured fairness. The chatbot treats all users equally without bias towards gender, religion, income or social background. Everyone receives the same clear and fair answers.

5.2.4 Sustainability Plan

We want our project to last for long time and keep helping people in Bangladesh. To make sure it stays useful we have created a clear sustainability plan. The first point is easy updates. Our system can be updated quickly whenever the law changes or new rules are introduced. This means users will always get the latest and most accurate legal information without delay. Second is low cost. The system will be cheap to run because we are using affordable cloud services. This will reduce the expense of hosting and maintaining the chatbot. As a result, even if more people start using the system it will still be cost effective. Third is energy efficiency. We will use cloud services that save energy and run on clean renewable energy. This helps reduce the negative impact on the environment. By doing this the project will remain eco-friendly while serving people. Fourth is scalability. Our system is designed to grow in the future. It can handle more users at the same time and can also cover more legal topics as the project develops. This means it will not only meet today's needs but also adapt to future requirements. Finally, we will provide user support. People will get easy guides, FAQs and instructions so they can use the system without difficulty. This support will make sure the chatbot is simple, user friendly and trusted by everyone.

5.3 Project Management and Financial Analysis

Managing this project required us to divide the work into clear phases and allocate responsibilities in a way that made progress measurable and achievable. We followed a phased development plan where each stage built on the previous one. The work was divided into four key phases:

Phase I Data Collection and Preprocessing: We scraped legal documents from the official Bangladesh Laws portal, cleaned the raw data and organized it into a structured bilingual dataset. **Phase II Model Training and Fine-Tuning:** Using GPU resources such as Google Colab and Jarvislabs.ai, we fine-tuned the LLaMA-3.1 model with LoRA and 4-bit quantization, making it efficient for practical use. **Phase III System Development and Integration:** At this stage, we built both backend and frontend components with Node.js, PostgreSQL, Next.js and React. The chatbot interface, authentication system and admin panel were implemented here. **Phase IV Testing, Deployment and Optimization:** In the final phase, we conducted multilingual testing, performance evaluation and user feedback sessions. The system was deployed on an Ubuntu server using Dokploy with Traefik, where Let's Encrypt was configured to provide automated SSL certificates for secure communication. We worked collaboratively in all stages, often sharing tasks like dataset cleaning, model fine-tuning and system integration, while ensuring documentation and presentation materials were prepared in parallel. This team-based approach helped us maintain balance and avoid bottlenecks. The financial analysis for this project is summarized in the table below (Table 5.3.1), which presents the estimated professional deployment budget.

Table 5.3.1 Financial Analysis

SL No.	Expense Description	Amount (BDT)	Description
01	Domain & SSL	2,000	Purchase of custom domain and SSL certificate (1 year).
02	Cloud Hosting & Infrastructure	10,000	Dedicated VPS / scalable cloud hosting (AWS/Google Cloud, 3-month plan).
03	GPU Rent & Colab Pro (Cloud)	20,000	Large-scale fine-tuning on cloud GPUs(Colab/AWS/Azure/Jarvislab A100 instances).
	Total	32,000	

5.4 Complex Engineering Problem

Our project, AI-Powered Legal Assistance for Bangladeshi Laws, qualifies as a complex engineering problem because it integrates multiple disciplines and addresses both technical and social challenges. Developing this system required knowledge from diverse areas such as artificial intelligence, natural language processing, web engineering, database design, cloud deployment and Bangladeshi law. The complexity also comes from the need to balance several conflicting requirements, for example, ensuring that the chatbot provides answers that are simple enough for ordinary users while also being accurate according to legal standards, or keeping the system secure while maintaining accessibility and low cost. Beyond technical considerations, the project is shaped by social realities: many people in Bangladesh face barriers to legal services due to high costs, confusing language, or limited access in rural areas. Meeting these needs demanded a solution that could combine technical innovation with social awareness. Furthermore, the success of the system depends on the proper functioning of interconnected parts, legal data scraping and cleaning, bilingual dataset preparation, model fine-tuning, web integration and secure deployment through tools such as Dokploy, Traefik and Let's Encrypt. If any one of these components fails, the overall solution becomes less effective. For these reasons, our project clearly meets the characteristics of a complex engineering problem, requiring depth of knowledge, multidisciplinary integration and careful problem solving.

5.4.1 Complex Problem Solving

Our project is not a simple task but a combination of different fields such as Artificial Intelligence (AI), law, Natural Language Processing (NLP) and social needs. According to the guidelines for complex problem solving, the work meets all seven attributes of complexity. Below Table 5.4.1 is the mapping of our project with these attributes.

Table 5.4.1: Mapping with Complex Engineering Problem

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stakeholder Involvement	EP7 Interdependence
✓		✓	✓		✓	✓

EP1: Dept of Knowledge

We needed knowledge from different fields, not only computer science. Training and fine-tuning a large language model required machine learning and natural language processing expertise. At the same time, we had to understand the structure of Bangladeshi laws so the system could answer correctly. On top of that, skills in data engineering, web development and deployment tools like Dokploy, Traefik and Let's Encrypt were also necessary. Bringing together these areas shows the depth of knowledge needed.

EP3: Depth of Analysis

Working with legal documents was not straightforward. The texts we collected from the Bangladesh Laws portal were unstructured and often inconsistent. We had to clean them, remove errors and arrange them into a machine-readable format. Some laws were in Bangla and others in English, so we created a bilingual dataset. To train the AI properly, we also prepared question-answer pairs based on real scenarios. This level of analysis went far beyond simple coding tasks.

EP4: Familiarity of Issues

The problems we aimed to solve are well-known in Bangladesh. Many people cannot afford lawyers, legal documents are written in difficult language and those in rural areas often have no access to proper advice. Because these challenges are already familiar to society, the solution we created is highly relevant. It directly targets issues people face in everyday life.

EP6: Stakeholder Involvement

The project affects more than just the end users. Citizens benefit by getting free and instant legal help, lawyers may use the system as a supportive tool and government agencies could adopt it to spread legal awareness. Developers and researchers are also stakeholders since they can learn from our approach and improve it further. The involvement of so many groups adds to the complexity.

EP7: Interdependence

Finally, the success of the system depends on many interconnected parts. If the data scraping does not work, the dataset becomes incomplete. If the AI model is not fine-tuned correctly, the answers may be inaccurate. If deployment with Dokploy, Traefik, or Let's Encrypt is not set up properly, users cannot access the service securely. Each part relies on the others, which means the whole project is highly interdependent.

Mapping with Knowledge Profile

In addition to meeting the criteria of complex problems, our project also connects with different areas of the engineering knowledge profile. These knowledge areas describe the type and level of expertise needed to successfully design and implement the system. Below Table 5.4.2 is the mapping of our project with these attributes.

Table 5.4.2: Mapping with knowledge Profile

K1 Natu ral Scien ce	K2 Mathem atics	K3 Engineeri ng Fundame ntals	K4 Special ist Knowle dge	K5 Enginee ring Design	K6 Enginee ring Practice	K7 Comprehe nsion	K8 Resear ch Literat ure
		✓	✓	✓	✓		✓

K3: Engineering Fundamentals

We relied on core computer science and engineering principles such as algorithms, data structures, databases and system architecture. These fundamentals were essential for tasks like structuring the dataset, designing the backend and building the user interface.

K4: Specialist Knowledge

The project required specialized knowledge in artificial intelligence, machine learning and natural language processing. Fine-tuning a large language model and adapting it to Bangladeshi laws demanded skills beyond general programming. Legal knowledge was also necessary to make sure the chatbot could provide accurate and meaningful answers.

K5: Engineering Design

Design skills were needed to bring all parts of the project together. This included the design of the system architecture, the user-friendly interface and the chatbot's workflow. Balancing usability with performance and ensuring that the solution remained scalable and secure were part of the design process.

K6: Engineering Practice

Practical application was a major part of this project. We used tools and frameworks such as PyTorch, Hugging Face, Next.js, PostgreSQL and deployment managers like Dokploy with Traefik. We also applied coding standards, authentication practices and cloud hosting techniques to ensure that the system worked reliably in real conditions.

K8: Research Literature

The choice of the chatbot to be adjusted to the current research to authenticate the technical solutions and to make the chatbot accurate in terms of the law was taken, regarding it as academic and legal literature.

5.4.2 Engineering Activities

In our project some important engineering activities are involved. It uses different resources connected with users and brings new ideas. It also helps society save the

environment and solve problems people already know. Below (5.3) shows how our project matches EA1 to EA5.

Mapping with Complex Engineering Activities

Our project also aligns with the recognized categories of complex engineering activities. These activities describe the type of work required, the level of interaction involved and the consequences for society and the environment. In Table 5.4.3 mapping with EA1-EA5 is explained below.

Table 5.4.3: Mapping with Complex Engineering Activities.

EA1 Range of re- sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
	✓	✓	✓	✓

EA2: Level of Interaction

The AI chatbot interacts directly with users in both Bangla and English. This required us to design a smooth and responsive interface so that people can ask questions naturally and get answers instantly. The interaction is not limited to the system alone users, developers and administrators all engage with different parts of the application.

EA3: Innovation

The project introduces something new to Bangladesh. While AI-powered legal assistants exist in other countries, there was no free system tailored to Bangladeshi law and languages. By combining local legal data with AI and making it accessible through a bilingual chatbot, our project represents an innovative step for legal technology in this context.

EA4: Consequences for Society and Environment

The system has positive consequences for society by making legal knowledge more accessible and reducing dependence on costly lawyers. People gain confidence to understand their rights and act accordingly. From an environmental perspective, the project reduces paper use and unnecessary travel since people can access legal help online. This digital-first approach helps lower waste and fuel consumption.

EA5: Familiarity

The issues we addressed are already familiar to the people of Bangladesh legal services are often expensive, legal language is complex and awareness is low. By offering a simple, free and always available chatbot, the project provides a practical solution to these well-known challenges.

5.5 Summary

The development of our AI-powered legal assistant combined modern engineering standards, problem-solving strategies and multidisciplinary knowledge. To ensure stability and security, we relied on trusted frameworks such as PyTorch, Hugging Face, Next.js, Node.js and PostgreSQL, with deployment managed through Dokploy and Traefik. Security practices, including JWT authentication, bcrypt hashing and SSL from Let's Encrypt, helped us build a system that users can trust. We faced several challenges, particularly in cleaning and structuring bilingual legal texts, training a large AI model within limited computing resources and designing a simple interface for people with little technical or legal background. These issues were managed through careful planning, efficient fine-tuning methods and repeated testing. The project clearly fits the definition of a complex engineering problem. It required deep expertise across AI, NLP, software design and Bangladeshi law while also balancing conflicting needs such as accuracy versus simplicity and performance versus cost. Our work also aligned with the engineering knowledge profile, applying core computing principles, specialist skills, design thinking and research-backed practices. The project reflects the nature of complex engineering activities. It used diverse resources, introduced innovation tailored to the Bangladeshi context, created direct interaction between people and technology and delivered benefits not only to users but also to society and the environment. Overall, the system demonstrates how structured engineering methods can be used to create a solution that is technically robust, socially relevant and sustainable for the future.

Chapter 6

Conclusion

In the final chapter, we summarize the whole project. We are highlighting the achievements and contributions. We also discussed the limitations and future work of the system. It will build a bridge of the gap between people and justice.

6.1 Summary

This project focused on designing and developing an AI-powered legal assistant that can provide guidance based on Bangladeshi laws. The main goal was to make legal information easier to access, simpler to understand and available at no cost for people who often struggle with traditional legal services. Many citizens cannot afford lawyers or find legal documents too complex, so our system was built to bridge this gap by offering quick and reliable answers in both Bangla and English. To achieve this, we collected and cleaned authentic legal data and used it to fine-tune the LLaMA-3.1 model with modern techniques such as LoRA and 4-bit quantization. These methods allowed us to create a model that is efficient, low-cost and suitable for real-world use. The chatbot was then integrated into a user-friendly web interface, ensuring that users can ask questions anytime and receive clear, simplified explanations of legal matters. The results of our work show that the assistant is effective in reducing dependency on expensive legal consultations, saving time and building confidence among users who need basic legal guidance. It also demonstrates how artificial intelligence can be applied to address real social challenges in Bangladesh. By making the system scalable, energy-efficient and easy to update, we have ensured that it can grow with future demands and remain useful for a wide range of users. Overall, this project represents not only a technical achievement but also a social contribution. It shows that legal knowledge can be made more open, inclusive and accessible when advanced technology is applied responsibly and thoughtfully.

6.2 Limitation

Although the AI-powered legal assistant demonstrates strong potential, there are several limitations that must be acknowledged. First, the chatbot cannot replace the role of a professional lawyer. It is designed only to provide basic guidance and explanations of laws, not to handle complex cases, prepare legal documents, or represent clients in court. Users with serious legal problems will still need professional assistance. Second, the system depends heavily on the quality and completeness of the data. While we collected a large number of Bangladeshi laws, not every law is available in a clean and structured form. Some texts are incomplete or written in difficult language, which may reduce the accuracy of responses. Similarly, the model sometimes struggles with ambiguous or very detailed queries, showing that there is still room for improvement in training and data coverage. Third, language remains a challenge. The chatbot works in both Bangla and English, but it does not yet handle local dialects, spelling mistakes, or mixed-language queries very well. This can cause misunderstandings or incomplete answers for certain users. Another limitation is the reliance on internet and server availability. Since the system is hosted online, users in rural areas with weak internet connections may face difficulties accessing the chatbot reliably. In addition, the server must remain updated and maintained; otherwise, the system risks giving outdated or incorrect legal

information if laws change over time. Overall, while the project successfully addresses many accessibility issues in the legal domain, these limitations highlight the importance of regular updates, better handling of language diversity and the need for continued refinement to make the system more accurate, inclusive and dependable.

6.3 Future Work

Our project has opened the door to making legal information more accessible, but there is still much room for growth. In the future, we plan to expand the dataset so that the chatbot can cover a wider range of Bangladeshi laws with more depth and accuracy. Regular updates will also be necessary to keep the system current whenever laws are amended or new ones are introduced. This will ensure that users always receive the latest information. We also want to improve the system's ability to handle language diversity. At present, the chatbot works in Bangla and English, but in the future it should be able to recognize local dialects, short forms and even common spelling mistakes. Adding support for voice input and output, especially in Bangla, will make the system even easier to use for people who are not comfortable typing. Another area of future development is accessibility in rural areas. Since internet connectivity is often weak outside big cities, we aim to build features that will allow the system to work offline or with very low data usage. This would make the chatbot more practical for people in remote regions. Finally, as AI technology continues to improve, we hope to integrate more advanced models that can provide even better accuracy and context understanding. The system could also be extended to support legal professionals by helping with legal research, drafting documents, or reviewing cases. In short, our project in the future will not only continue to provide free and simple legal advice but will also grow into a more advanced and inclusive platform, capable of serving a larger community and adapting to changing nee

References

- [1] Y. Zhang, L. Li and S. Wang, “Legal AI and Chatbots: A New Era,” *Journal of AI & Law*, vol. 35, no. 2, pp. 112–125, 2021.
- [2] A. Patel and R. Sharma, “AI in Legal Document Review,” *International Journal of Legal Informatics*, vol. 14, no. 3, pp. 98–110, 2022.
- [3] M. Lee, H. Choi and J. Kim, “AI for Faster Legal Research,” *IEEE Access*, vol. 10, pp. 22632–22645, 2021.
- [4] T. Johnson, A. Brown and P. Smith, “AI and Case Outcome Prediction,” *ACM Transactions on AI*, vol. 7, no. 4, pp. 1–18, 2021.
- [5] Z. Zhang and J. Liu, “AI and Law: Applications and Challenges,” *AI & Society*, vol. 33, no. 5, pp. 855–870, 2018.
- [6] W. Wasi, W. Faisal, M. R. Islam and M. M. Bappy, “Exploring AI Powered Legal Assistance in Bangladesh,” *arXiv preprint*, 2024.
- [7] S. Rahman, A. Hossain and R. Karim, “AI in Legal Services in Bangladesh,” *Dhaka University Journal of IT*, vol. 8, no. 1, pp. 44–56, 2020.
- [8] S. Islam, M. Hasan and F. Akter, “AI and Case Outcome Prediction in Bangladesh,” *Bangladesh Journal of Legal Technology*, vol. 5, no. 2, pp. 13–25, 2021.
- [9] R. Raham, N. Chowdhury and A. Uddin, “Language Challenges for AI in Bangladeshi Legal Systems,” *Journal of Computational Linguistics*, vol. 11, no. 3, pp. 33–47, 2022.
- [10] K. S. Hasan, “JusticeNetBD: RAG-Based Legal Assistant for Women in Bangladesh,” *SSRN Preprint*, 2025.
- [11] H. Cui, H. Zhang, X. Liu and Y. Sun, “ChatLaw: A Multi-Agent Legal Assistant with Knowledge Graphs,” *arXiv preprint*, 2023.
- [12] A. Kalaycioglu, D. Yi and F. Khan, “LICES: Legal Intelligence Compliance Expert System,” *arXiv preprint*, 2025.
- [13] J. Mahoney, A. Lee and K. Chen, “Explainable Legal Document Classification,” *arXiv preprint*, 2019.
- [14] N. Guha, D. Jain, V. Zhong, et al., “LegalBench: A Benchmark for Measuring Legal Reasoning in Large Language Models,” *arXiv preprint*, 2023.
- [15] I. Chalkidis, M. Fergadiotis, P. Malakasiotis and I. Androutsopoulos, “LEGAL-BERT: The Muppets Straight Out of Law School,” in *Findings of EMNLP*, pp. 2898–2910, 2020.

- [16] S. Paul, A. Mandal, P. Goyal and S. Ghosh, “InLegalBERT: Pre-trained Language Models for Indian Legal Domain,” ICAIL, 2023.
- [17] Y. Xiao, H. Li and Z. Ren, “Lawformer: A Pre-trained Language Model for Chinese Legal Document Understanding,” Findings of ACL, 2021.
- [18] M. R. Kabir et al., “LegalRAG: A Hybrid RAG System for Multilingual Legal Information Retrieval,” arXiv preprint, 2025.
- [19] S. Akhand, “Bangladesh Legal Acts Dataset,” Hugging Face, 2025. [Online]. Available: <https://huggingface.co/datasets/sakhadib/Bangladesh-Legal-Acts-Dataset>
- [20] T. T. Mayeesha, F. Islam and N. Ahmed, “AI4Bangladesh: AI Ethics for Bangladesh—Challenges, Risks and Recommendations,” ACM AI & Society Workshop, 2025.
- [21] J. Gupta, A. Sharma, S. Singhania and A. Abidi, “Legal Assist AI: Leveraging Transformer Models for Legal Assistance,” arXiv preprint, 2025.
- [22] T. Hossain, “Advancing Legal Accessibility in Bangladesh through AI-Powered Interfaces,” BRAC University Thesis, 2024.
- [23] L. Fourrage, “Using AI for Legal Professionals in Bangladesh,” NuCamp Blog, Sep. 2025.
- [24] “Bangladesh Legal Aid and Services Trust,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Bangladesh_Legal_Aid_and_Services_Trust
- [25] “National Legal Aid Services Organization,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/National_Legal_Aid_Services_Organization
- [26] J. Surden, “Machine Learning and Law,” Washington Law Review, vol. 89, pp. 87–115, 2014.
- [27] K. Ashley, “Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age,” Cambridge Univ. Press, 2017.
- [28] C. Katz, “DoNotPay: The World’s First Robot Lawyer,” TechCrunch, 2016.
- [29] “ROSS Intelligence,” LegalTech News, 2017.
- [30] “LexisNexis Legal Research Platform,” LexisNexis, 2021. [Online]. Available: <https://www.lexisnexis.com>
- [31] J. Medvedeva, M. van der Velden and M. Surden, “LawBot: Conversational AI for Legal Guidance,” ICAIL Workshop, 2018.
- [32] “Legal Robot: Contract Analytics Platform,” LegalRobot.com, 2019. [Online]. Available: <https://legalrobot.com>
- [33] M. Bommarito and D. Katz, “Law as Data: Computation, Text and the Future of Legal

Analysis,” Santa Fe Institute Press, 2018.

- [34] R. Susskind, “Tomorrow’s Lawyers: An Introduction to Your Future,” Oxford Univ. Press, 2017.
- [35] J. McGinnis and R. Pearce, “The Great Disruption: How Machine Intelligence Will Transform the Role of Lawyers,” *Fordham Law Review*, vol. 82, no. 6, pp. 3041–3066, 2014.

Islam Md. Ashikul

AI-Powered Legal Assistance for Bangladeshi Laws Using Machine Learning and Large Language Models

 Quick Submit Quick Submit Daffodil International University

Document Details

Submission ID

trn:oid::1:3339858098

Submission Date

Sep 15, 2025, 1:00 PM GMT+6

Download Date

Sep 15, 2025, 1:22 PM GMT+6

File Name

Legal_Assistance_for_Bangladeshi_Laws_Using_Machine_Learning.pdf

File Size

1.4 MB

58 Pages

19,365 Words

104,936 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



AI-Powered Legal Assistance for Bangladeshi Laws Using Machine Learning and Large Language Models

ORIGINALITY REPORT

9%	7%	3%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

Handwritten signature

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	4%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	Submitted to United International University Student Paper	1%
4	arxiv.org Internet Source	<1%
5	jirs.judcom.nsw.gov.au Internet Source	<1%
6	iris.luiss.it Internet Source	<1%
7	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dharendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%
8	repository.dinamika.ac.id Internet Source	<1%
9	www.sanbi.org Internet Source	<1%
10	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%