

LEGAL CHATBOT FOR AI-BASED LEGAL ASSISTANCE

A PROJECT REPORT

Submitted by

LAKSHMINARASIMHA. V [RA2212703010012]

Under the Guidance of

Dr. T.R. VEDHAVATHY

Assistant Professor, Networking and Communications

in partial fulfillment of the requirements for the degree

of

MASTER OF TECHNOLOGY(INTEGRATED)

in

COMPUTER SCIENCE AND ENGINEERING

with specialization in CYBERSECURITY AND

DIGITAL FORENSICS



DEPARTMENT OF NETWORKING AND COMMUNICATIONS
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2025



**Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form**

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **MTech. Integrated in Computer Science and Engineering with specialization in Cybersecurity and Digital Forensics**

Student Name : **Lakshminarasimha. V**

Registration Number : **RA2212703010012**

Title of Work : **Legal Chatbot for AI-Based Legal Assistance**

I / We hereby certify that this assessment complies with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

Lakshminarasimha. V [RA2212703010012]

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 21CSP302L- Project report titled "**LEGAL CHATBOT FOR AI-BASED LEGAL ASSISTANCE**" is the bonafide work of "**LAKSHMINARASIMHA. V [RA2212703010012]**" who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T.R.VEDHAVATHY

SUPERVISOR

Assistant Professor
Department of
Networking and
Communications

SIGNATURE

Dr.M.LAKSHMI

PROFESSOR & HEAD

Department of Networking and
Communications

EXAMINER 1

EXAMINER 2

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M**, Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. Lakshmi**, Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr. M.Lakshmi**, Professor and Head, Networking And Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr.T.R.Vedhavathy**, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr.T.R.Vedhavathy**, Department of Networking and Communications, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Department of Networking and Communications, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

ABSTRACT

The Legal Chatbot is an AI-powered framework outlined to help clients with lawful questions. It utilises characteristic dialect handling (NLP) and an organised lawful information base to empower clients to yield questions and get exact and important legitimate reactions. The framework takes after microservices design with a backend API that interacts with a database putting away legitimate statutes, case laws and wording. The chatbot workflow includes preparing client enquiries, recovering pertinent legitimate data and creating AI-based reactions on the off chance that the framework can get to a reasonable lawful source. It recovers information from the database and employs NLP to create a reaction. The chatbot also permits clients to supply criticism on the answers it creates, which makes a difference in its exactness over time. Directors oversee the database by overhauling legitimate terms and cases to guarantee the system's reliability. The objective of this extension is to supply speedy and available lawful help, lessening the requirement for proficient lawful interviews for essential legitimate requests. The chatbot can be improved with machine learning models to accelerate its capacity to recognise client enquiries, analyse legal records, and anticipate case law.

The Legal Chatbot is an artificial intelligence (AI) system that uses Natural Language Processing (NLP) and an organised legal knowledge store to help users with legal questions. Users can ask queries through the chatbot, and it will process them to provide precise and pertinent legal answers. The system uses a microservices architecture and has a backend API that communicates with a database that contains terminology, case laws, and legal statutes.

Knowledge retrieval, question processing, and AI-based response production are all part of the chatbot workflow. When a pertinent legal source is accessible, the chatbot retrieves information from the database; if not, it uses natural language processing (NLP) to produce a response. To increase accuracy over time, the system also lets users comment on responses produced by the chatbot. To maintain dependability, administrators also update legal words and cases in the database.

By offering prompt and easily available legal aid, this project seeks to lessen reliance on expert legal consultations for routine legal questions. Machine learning models can be used to further improve the chatbot's intent recognition, legal document interpretation, and case law forecasts.

TABLE OF CONTENTS

ABSTRACT	v	
TABLE OF CONTENTS	vi	
LIST OF FIGURES	viii	
LIST OF TABLES	ix	
ABBREVIATIONS	x	
CHAPTER NO.	TITLE	PAGE NO.
1. INTRODUCTION		1
1.1 Introduction to Project		1
1.2 Problem Statement		2
1.3 Motivation		3
1.4 Sustainable Development Goal of the Project		4
2. LITERATURE SURVEY		5
2.1 Overview of the Research Area		5
2.2 Existing Models and Frameworks		6
2.3 Limitations Identified from Literature Survey (Research Gaps)		7
2.4 Research Objectives		9
2.5 Product Backlog (Key user stories with Desired outcomes)		11
2.6 Plan of Action (Project Road Map)		12
3. SPRINT PLANNING AND EXECUTION METHODOLOGY		13
3.1 SPRINT I		13
3.1.1 Objectives with user stories of Sprint I		13
3.1.2 Functional Document		14
3.1.3 Architecture Document		16
3.1.4 Outcome of objectives/ Result Analysis		17
3.1.5 Technical Analysis		18
3.1.6 Sprint Retrospective		20
3.2 SPRINT II		21
3.2.1 Objectives with user stories of Sprint II		21
3.2.2 Functional Document		23
3.2.3 Architecture Document		25
3.2.4 Outcome of objectives/ Result Analysis		26
3.2.5 Technical Analysis		27
3.2.6 Sprint Retrospective		32

4. RESULTS AND DISCUSSIONS	33
4.1 Project Outcomes (Performance Evaluation, Comparisons, Testing Results)	33
4.2 Output	35
5. CONCLUSION AND FUTURE ENHANCEMENTS	38
REFERENCES	40
APPENDIX	
A. CODING	41
B. PLAGIARISM REPORT	44

LIST OF FIGURES

CHAPTER NO.	TITLE	PAGE NO.
2.1	Product Backlog 1.....	11
2.2	Product Backlog 2.	11
2.3	Project Road Map	12
3.1	Architecture Flow Diagram	16
3.2	Sprint Retrospective 1.	20
3.3	Architecture Diagram	25
3.4	Sprint Retrospective 2.....	32
4.1	Login Page.....	35
4.2	Response Page.....	35
4.3	Landmark Cases.....	36
4.4	Feedback Page.....	36

LIST OF TABLES

CHAPTER NO.	TITLE	PAGE NO.
3.1	Objective and Result of Sprint 1.....	17
3.2	Performance Indicators.....	17
3.3	Challenges and Resolutions.....	17
4.1	Performance Evaluation.....	33
4.2	Comparisons with Existing Systems.....	33
4.3	Test Case Summary.....	34

ABBREVIATIONS

CSS	Cascading Style Sheet
AI	Artificial Intelligence
DB	Database
HTTP	HyperText Transfer Protocol
HTML	Hyper Text Markup Language
JS	Javascript
NLP	Natural Language Processing

CHAPTER 1

INTRODUCTION

1.1 Introduction to the Project

In a time where technology defines access and efficiency more and more, the Indian legal system nevertheless remains mostly opaque and difficult for the typical citizen to negotiate. By developing an AI-powered, user-friendly, role-based digital assistant that simplifies legal information retrieval, empowers users with correct case law references, and advances more general access to justice, the legal chatbot project seeks to close this gap.

Designed to be a virtual legal assistant for consumers in different legal literacy levels—citizens, attorneys, students, and administrators—this chatbot system Constructed with a MySQL relational database and a Flask-based microservices backend, the chatbot lets users securely register, log in, and access pertinent legal data via a smart and ordered interface. The system uses the locally hosted Ollama AI model known as "indian-law-llama" to analyse legal enquiries and create context-aware answers tying users to historic case laws and statutes.

The Role-Based Access Control (RBAC) system, which guarantees data confidentiality and user access discrimination, is a main highlight of the project. General users, for example, can search laws; administrators have rights to add or change significant court rulings. By including AI-powered legal suggestions and feedback systems, the project improves not only system accuracy over time but also user confidence and involvement.

Users of the chatbot can enter keywords or legal contexts to access condensed and full-text case laws from a dynamically updated repository, therefore providing a search capability for significant legal cases. This helps legal researchers, practitioners, and students to conduct effective, focused legal research without personally negotiating difficult databases.

From a development aspect, the platform integrates several modules: case search, dashboard interfaces, admin panels, feedback reporting, secure login and registration, password recovery, Every module is designed to guarantee high standards of data security, integrity, and AI ethics together with a flawless and easy user experience.

This initiative is essentially a scalable legal technology platform meant to make Indian legal information more easily available, intelligible, and practical. It reflects the movement towards digital justice and acts as a fundamental first step towards intelligent legal automation capable of developing alongside developments in artificial intelligence and natural language processing technology.

1.2 Problem Statement

One of the most important problems facing India's judicial system yet is access to justice. The practical accessibility of legal information is still a considerable obstacle, especially for the common man, even if the nation boasts a well-established legal system and a large corpus of case laws. Legal knowledge is sometimes buried in enormous books, sophisticated terminologies, and disjointed databases requiring specific skills to understand. Those without legal experience so find it difficult to locate pertinent material, grasp their rights, or engage in wise legal action.

Lack of proximity to legal specialists in rural and underprivileged communities accentuates this problem even further. Legal advice can sometimes costly and time-consuming even in metropolitan areas. The conventional approaches of legal research, which depend on hand scanning through legal databases or physical law books, are not only ineffective but also exclude non-technical and non-legal audiences.

Moreover, technologies and existing legal databases are mostly designed for professionals. They cannot handle natural language searches, clever search tools, or simple interfaces. This leaves a lot of people depending on middlemen, which causes delays, misunderstandings, and maybe unfair practices. The difficulty for students and legal professionals is sorting through vast amounts of data to find contextually relevant case laws, which may be quite time-consuming without appropriate tools.

Furthermore, many systems do not enable role-based access control (RBAC), which causes security issues particularly in relation to private legal data. Furthermore lacking are personalising tools, feedback systems, and automated legal recommendations based on user searches, which would otherwise help learning and decision-making.

By providing an AI-powered legal assistant combining intelligent inquiry processing, landmark case searchability, and safe user administration, the legal chatbot project aims to directly solve these constraints. This helps to open the path for an Indian justice system more transparent, inclusive, and effective.

1.3 Motivation

The Indian legal system, while broad and powerful in structure, can look remote and inaccessible to the average citizen. Legal documents are voluminous, written in difficult language, and kept in disconnected collections. Even with increased digitisation, navigating this enormous legal system remains difficult for those without legal backgrounds. This divergence between the legal framework and the public is a major motivator for the creation of our AI-powered legal chatbot.

At the heart of this project is the ambition to democratise access to legal information. Every individual, regardless of education level, geography, or economic status, should be able to comprehend their legal rights and obligations. However, most people nowadays are compelled to rely on pricey legal consultations or time-consuming research methods that many cannot afford. This sparked the concept of developing a platform that can serve as a virtual legal advisor, assisting users in understanding the law through simple, conversational interactions.

The motivation is partly based on seeing how artificial intelligence is altering other fields, such as banking, healthcare, and education. However, legal technology has been slow to advance, particularly in India. We were motivated to use AI, specifically natural language processing (NLP), to make legal material more searchable, relatable, and actionable. By creating a chatbot that understands and responds to natural language queries, users can bypass legal jargon and rapidly get relevant case laws and legal interpretations.

Furthermore, we were motivated by the difficulties that legal professionals and students confront while manually looking for relevant judgements and precedents. A smart assistant that can recommend landmark cases depending on context can save time while also improving legal research and argumentation quality.

On a larger scale, this project is motivated by national and international initiatives aimed at fostering digital transformation and access to justice. Government efforts such as Digital India emphasise inclusive technology, whereas the United Nations' Sustainable Development Goal 16 emphasises the development of strong institutions and legal access for all. This legal chatbot directly contributes to these goals by serving as a digital link between people and the legal system.

Finally, the motive includes education and empowerment. A platform like this can be used as a learning tool in law schools, civil society organisations, and community legal clinics, encouraging people to interact with the legal system proactively rather than reactively.

In summary, this project is motivated by the notion that legal literacy is a fundamental right and that modern technologies, when carefully constructed, may play a revolutionary role in closing the justice gap in India and elsewhere.

1.4 Sustainable Development Goal of the Project

The legal chatbot initiative directly relates to **UN Sustainable Development Goal (SDG) 16: "Peace, Justice and Strong Institutions"**. Emphasising the creation of inclusive communities for sustainable development, the provision of access to justice for everyone, and the development of efficient, responsible, open institutions at all levels, SDG 16 addresses Using the power of artificial intelligence and digital infrastructure, our project is a technical intervention geared especially to close the distance between legal institutions and the people they are meant to serve.

Advocating Equal Access to Justice

Ensuring that everyone—regardless of their socioeconomic level—has access to justice is fundamental to SDG 16. Due to administrative delays and ignorance, legal information is sometimes difficult to find, costly to understand, and slow to react upon in India. By offering a free, AI-powered, user-friendly platform where people may access pertinent legal information, understand historic case laws, and get AI-assisted answers to their questions in real-time, our legal chatbot seeks to erase these barriers. This guarantees that everyone has a right to justice rather than only a luxury for the knowledgeable few.

Encouragement of Legal Consciousness and Institutional Transparency

Also stressed in public institutions by SDG 16 are openness and responsibility. Our effort fosters public confidence in the judicial system by making legal facts more easily available, ordered, and comprehensible. Nowadays, citizens can learn about their rights, get informed about important legal developments, and observe how cases were decided. The legal chatbot helps to lower reliance on middlemen, make the court more open, and raise a legally conscious public.

Using Technology to Strengthen Institutions

Complementing SDG 16.6 ("Develop effective, accountable and transparent institutions"), the project helps the digitisation of legal services, a major focus in India's Digital India program. The chatbot enables legal institutions and educational organisations to apply artificial intelligence to increase their productivity, streamline legal research processes, and provide smarter citizen services by establishing a safe, scalable, intelligent system integrated with role-based access control (RBAC).

Allowing Legal Inclusion

Social and economic inclusion depend on legal inclusion; it is fundamental. With possibility for multilingual help in the future, the legal chatbot is essential in ensuring that none is left behind—a value central to the SDGs—by providing a platform transcending boundaries of geography, education, and language. This digital tool gives people from underprivileged backgrounds legal information and decision-making ability.

CHAPTER 2

LITERATURE SURVEY

2.1 Overview of the Research Area

Underlying this project is a study area spanning Legal Informatics, Artificial Intelligence (AI), and Natural Language Processing (NLP)—a fast developing discipline emphasising the use of intelligent technologies to the legal arena. Technology-driven solutions that can help users easily and effectively grasp and negotiate legal texts, case laws, and statutory information as legal systems worldwide get more complicated and data-heavy become more needed.

Legal Informatics and Lawful Technology

Legal informatics is the study and application of information technology to legal chores including documentation, research, analysis, and automation. Its main emphasis is on creating instruments enabling meaningful interaction between legal experts and the general public with legal data. Legal databases, document management systems, and e-court projects have long been part of this mix. Recent developments, however, have turned towards including artificial intelligence to improve legal system searchability, accessibility, and decision-making.

Legal System Natural Language Processing

Legal language is one of the key difficulties in legal informatics since it is sometimes complex, technical, and loaded with domain-specific words. By letting systems parse, understand, and answer legal enquiries in natural language, NLP provides a means to close the gap between human language and machine understanding. Legal NLP systems can categorise legal documents, summarise case laws, find important terms, and even project legal results depending on past instances.

Transformer-based language models (e.g., BERT, GPT, and domain-specific models like Legal-BERT or indian-law-llama) have lately advanced the possibilities of artificial intelligence in legal applications. These models let for deeper study and more accurate answers by enabling semantic understanding of legal material.

Legal Chatbots and Conversation AI

Combining NLP, dialogue management, and machine learning, conversational artificial intelligence creates systems able to engage users in a human-like way. From the legal standpoint, this creates fresh opportunities for mass legal assistance provision. Reducing the load on human legal specialists and providing legal access to underprivileged groups, legal chatbots can reply to FAQs, help users understand their rights, and guide them through legal processes.

Justice Access and Legal Empowerment

Improving access to justice via digital means forms the main focus of this research field. Legal IT platforms have helped to drive a shift towards open, inclusive, and user-centred legal systems. Particularly in areas where professional legal aid is either scarce or pricey, AI-powered solutions are supposed to complement and enhance rather than replace lawyers.

Regulatory and ethical considerations

Researchers are also looking at problems of prejudice, openness, responsibility, and privacy as artificial intelligence is used in legal settings and grows more common. Responsible innovation in this domain depends on AI systems respecting legal ethics and following data protection regulations.

2.2 Existing Models and Frameworks

Artificial intelligence has been increasingly applied in the legal field in recent years, resulting in the creation of several models, platforms, and frameworks meant to simplify legal research, automate paperwork, and improve access to justice. Some of the most pertinent current models and frameworks that have shaped the evolution of legal chatbots and legal information systems driven by artificial intelligence are examined in this part.

Retrieval Systems for Legal Information

Via keyword-based search engines, traditional legal databases include Manupatra, SCC Online, Indian Kanoon in India, and Westlaw and LexisNexis globally offer legal information. Users of these platforms may access statutes, case laws, and commentary. They are better suited to legal professionals than the general public, nevertheless, because they lack conversational interfaces and natural language understanding.

Rule-Based Legal Expert Systems

Earlier legal expert systems generated automatic legal reasoning using IF-THEN rule logic. One such is LOIS (Legal Ontology-based Information System) and MYCIN (though not specifically legal-specific, it affected early legal systems). These models could not adequately handle ambiguous or natural language searches since they needed human rule definitions for every scenario and lacked scalability or adaptability.

Law Natural Language Processing (NLP)

Modern NLP developments have transformed legal text processing. Like models:

- Legal-BERT: Designed for tasks including document classification, summarising, and question-answering, a BERT-based transformer pretrained on U.S. legal documents
- CaseLaw-BERT: Designed especially with reference to court rulings.

By allowing contextual knowledge of legal language, these models serve to increase the accuracy of legal chatbots and automated summarising systems.

Platform for Conversational AI

Conversational legal AI has advanced thanks in part to sites like Ross Intelligence and DoNotPay (U.S.). Promoted as "the world's first robot lawyer," DoNotPay uses chat interfaces to assist consumers with minor legal concerns such as parking tickets and returns. These systems may not enable complicated legal searches or regional legal systems like India's, though, and they are sometimes jurisdiction-specific.

Legal tech's role-based access control (RBAC)

Few current systems efficiently employ RBAC methods for public-facing legal tech. While most enterprise-level legal systems use user roles to provide access, they do not link this with artificial intelligence chat or case law recommendation systems. RBAC combined with intelligent services is a fresh approach that in legal automation blends security with access.

Although great progress has been accomplished, current systems usually suffer from:

- Insufficient conversational interfaces for the regular user
- Restricted knowledge of the Indian legal setting
- RBAC's insufficient integration for user-level, secure access
- Not much help for personalising or real-time comments.

These holes highlight the need of an integrated, India-specific, AI-driven legal chatbot addressing not just information retrieval but also user empowerment, contextual correctness, and safe accessibility.

2.3 Limitations Identified from Literature Survey (Research Gaps)

Reviewing the present scene of legal informatics and artificial intelligence applications in law, numerous important constraints and research gaps were found that impede the general accessibility and usage of legal systems—especially in the Indian setting. The relevance and inventiveness of our initiative are derived from these gaps.

Lack of region-specific legal artificial intelligence systems

Models like Legal-BERT and platforms like DoNotPay or Ross Intelligence as most artificial intelligence legal tools are trained on Western legal systems and documents. These instruments do not fit the special structure, vocabulary, and variety of Indian laws. India-oriented AI legal models are desperately lacking, which limits adoption of legal artificial intelligence among Indian legal institutions and the general public.

Lack of conversational access for shared users

Many of the current legal research systems (such as Manupatra, Indian Kanoon) use keyword-based interfaces presuming users are conversant with legal jargon. Leaving legal access limited to qualified experts, there are no generally available platforms providing legal advice through conversational artificial intelligence that interprets natural language enquiries in layman terms.

Minimal Application of Contextual Understanding Advanced NLP

Although transformer-based models—e.g., BERT, GPT—have shown potential in natural language processing—few legal systems use them for contextual legal reasoning or recommendation. Most systems fail to grasp the intent or semantic meaning underlying user searches since most depend on string matching or keyword-based searches.

Lack of Case Law Intelligent Recommendation Systems

While current legal databases can enumerate pertinent case laws, they lack AI-powered recommendation engines capable of automatically suggesting historic cases depending on a user's question or legal concern. For students and junior lawyers particularly, this makes legal research time-consuming and erratic.

Restricted Personalisation and Mechanisms of Feedback

Legal platforms of today lack tailored interactions. Neither adaptive learning systems where the system develops over time depending on user interactions nor feedback loops allowing users to judge the relevance of responses. User involvement and system improvement are thereby lessened.

Insufficient Role-Based Access Control (RBAC) and Security

Particularly in cases involving sensitive or personal information, security in legal data systems is sometimes disregarded. Most current chatbots and legal search engines lack RBAC, so it is challenging to distinguish between access rights of general users, law students, or administrative/legal authorities.

Insufficient Combining of Legal Databases, UI, and AI

Fragmentation characterises current systems; no one comprehensive framework combining a user-friendly interface, artificial intelligence-based reasoning, and a structured, safe legal database exists. This causes a jumbled user experience and stops flawless interaction between users and legal material.

Non-technical and non-English exclusion

Many sites are not accessible to speakers of local languages and operate just in English. Moreover, their design implies a certain degree of legal or technical proficiency, therefore eliminating a great part of the population, especially from rural areas with lower literacy backgrounds.

These research gaps amply highlight the requirement of a scalable, Indian-law-specific, AI-enabled chatbot with natural language interactions, tailored legal suggestions, feedback support, role-based security, multilingual accessibility, and Indian law specific AI capability. By closing these gaps, legal literacy will be much raised, the load on lawyers will be lowered, and equal access to justice will be encouraged.

2.4 Research Objectives

This project's main goal is to design, build, and assess an AI-powered legal chatbot system that improves public access to legal information, promotes contextual comprehension of Indian laws, and enables users with real-time legal aid. This approach seeks to close the difference between the accessibility requirements of common people, students, and attorneys and the complexity of legal systems.

The particular research objectives are arranged to methodically close the gaps found in the literature and direct the technological, functional, and social growth of the project.

Objective 1:

- Create a conversational artificial intelligence interface for legal searches.
- Create a chatbot from powerful Natural Language Processing (NLP) that can manage user free-text, natural language legal searches.
- Verify the chatbot offers in layman-friendly language correct, context-sensitive, legally sound answers.

Objective 2:

- Either refine or combine a legal language model taught on Indian laws.
- Apply or improve a domain-specific model, such as `indian-law-llama`, refined on legal commentary, Indian statutes, and case laws.
- For question answering, classification, and summarising projects, guarantee semantic understanding of difficult legal materials.

Objective 3:

- Build a Secure Backend using RBAC based on roles.
- Apply RBAC in the MySQL-powered Flask backend to provide distinct access for administrators, law students, and users.
- Use secure authentication and session management techniques to guard private legal searches and comments.

Objective 4:

- Create a Foundation of Structured Legal Knowledge
- Create a legal database including full-text rulings, summaries, legal domains, and historic case laws.
- Let dynamic updates by managers maintain scalable and updated databases.

Objective 5:

- Authorise Legal Recommendations Driven by AI
- Combine machine learning algorithms to, depending on user searches, suggest pertinent case laws and legal clauses.
- Customise recommendations by thinking through user role, query intent, and feedback past.

Objective 6:

- Combine Adaptive Learning with Real-Time Feedback
- Let people evaluate chatbot answers and provide comments to help them to be constantly better.
- Use feedback loops to improve AI recommendations and over time enhance system understanding.

Objective 7:

- Create an inclusive and user-centric frontend interface.
- Provide responsive and easy UI for mobile as well as web platforms.
- Make sure that future versions include multilingual support in order to guarantee accessibility among several user groups.

Objective 8:

- Examine System Performance Using User Testing
- Test chatbot accuracy, usability, and user satisfaction with law students, attorneys, and lay users.
- Validation of the system should rely on both qualitative (user feedback) and quantitative (accuracy, reaction time).

Objective 9:

- Match with SDG 16 – Peace, Justice, and Strong Institutions
- Make that the system directly supports UN SDG 16 by democratising Indian access to justice and bolstering digital legal infrastructure.

These goals together provide a good basis for a complete, scalable, and socially impactful legal chatbot solution combining artificial intelligence, law, and user empowerment.

2.5 Product Backlog (Key user stories with Desired outcomes)

Product Backlog for Legal Chatbot for AI-Based Legal Assistance										
ID	Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	User Registration & Login	User Management	As a user, I want to register and log in securely, so that I can access the legal information app.	Must Have	Completed	User can register with email and password. User can log in and log out securely. System validates incorrect credentials. Password reset functionality is available.	User authentication system. Registration and login form. Password encryption. Session management.	Secure login mechanism. Fast authentication response time. Compliance with security standards. User-friendly interface.	3 days	1 day
2	Search Functionality	Core Functionality	As a user, I want to search for case laws and judgments using keywords or filters, so that I can find relevant legal information quickly.	Must Have	Completed	Users can search case laws using keywords. Results are displayed within 2 seconds. Search includes filters for case type and date. Advanced search supports Boolean queries.	Keyword-based search engine. Filters and sorting options. Search index optimization. Result caching for frequent queries.	Fast query processing. Scalable for large legal databases. High availability (99.9% uptime). Minimal latency in search results.	5 days	3 days
3	AI-powered Recommendation	Core Functionality	As a user, I want the app to recommend relevant case laws based on my search history, so that I can discover additional useful information.	Should Have	To Do	AI suggests relevant case laws based on user queries. Recommendations improve with user feedback. AI can explain why a case is relevant. User preferences refine recommendations over time.	AI recommendation engine. Machine learning model for legal case relevance. User interaction tracking. Feedback-based AI improvement.	High accuracy in recommendations. Real-time performance. Explainability of AI decisions. Scalable data processing.	7 days	
4	Case Law Summary View	Core Functionality	As a user, I want a summarized view of case laws and judgments, so that I can quickly understand the key points of a legal ruling.	Must Have	Completed	Users can view summarized case laws. Summaries provide key legal points. Summaries are generated automatically. Users can toggle between full text and summary.	Text summarization algorithm. UI for displaying summaries. Integration with legal databases. Summary refinement through feedback.	Fast rendering time. Summaries maintain legal accuracy. Consistent summary format. Secure data access.	7 days	6 days
5	User Profile & Settings	User Management	As a user, I want to customize my profile and settings, so that I can tailor the app experience to my needs.	Could Have	To Do	Users can update profile details. Settings include notification preferences. Data is securely stored and retrieved. Users can delete their account.	Profile management system. Notification settings. Secure data storage. UI for editing user details.	Intuitive UI/UX. Data privacy and encryption. Quick profile updates. Reliable notification system.	8 days	

Fig2.1 Product Backlog 1

6	Legal Insights Dashboard	Core Functionality	As a legal professional, I want to see a dashboard of important insights, such as trending case laws or recent rulings, so that I can stay updated.	Must Have	To Do	Dashboard shows real-time legal insights. Data visualization for case trends. Users can customize dashboard widgets. Information refreshes dynamically.	Dashboard UI components. Data aggregation and analytics. Customizable widgets. Real-time data updates.	Mobile responsiveness. Low latency in data display. Secure API connections. Optimized query performance.	9 days	
7	Knowledge Base for Legal Information	Core Functionality	As a user, I want the chatbot to have access to a vast database of legal knowledge for accurate responses.	Must Have	Completed	Chatbot accesses an extensive legal knowledge base. Users can ask general legal questions. Information is retrieved accurately. Sources are cited for transparency.	Integration with legal databases. Structured data retrieval. NLP-based question processing. Search optimization for legal terms.	Fast query response. Scalable database architecture. Reliable data storage. Secure API access.	8 days	6 days
8	Legal Terminology Understanding	Core Functionality	As a user, I want the chatbot to understand and define legal terminology when asked.	Should Have	Completed	Chatbot defines legal terms on request. Users get simplified explanations. System supports multilingual explanations. Examples of term usage are provided.	Legal dictionary integration. NLP model for definition retrieval. UI for displaying term explanations. Multi-language support.	Quick term retrieval. High accuracy in definitions. Clear and structured responses. User-friendly display.	7 days	4 days
9	Handling Complex Legal Queries	Core Functionality	As a user, I want the chatbot to handle multi-part legal questions and respond with comprehensive answers.	Should Have	Completed	Chatbot understands multi-part questions. Queries are broken into manageable parts. System provides step-by-step answers. Users can refine responses interactively.	Mult-turn dialogue handling. Query breakdown logic. Context retention in conversations. Interactive refinement UI.	Optimized for complex interactions. Fast response time. Error handling for misinterpretations. AI-driven accuracy improvements.	6 days	4 days
10	Personalized Legal Recommendations	AI & Machine Learning	As a user, I want the chatbot to recommend relevant legal resources based on my query history.	Could Have	To Do	Users get tailored case suggestions. Recommendations adapt to browsing history. Users can save recommended cases. Feedback improves future recommendations.	AI-based personalization engine. User history tracking. Case-saving feature. Feedback collection.	Scalable recommendation system. Secure user data handling. Efficient recommendation generation. Adaptive learning from user actions.	7 days	
11	User Feedback and Continuous Improvement	User Engagement & Analytics	As a user, I want to rate the chatbot responses so that it can improve over time.	Should Have	Completed	Users can rate chatbot responses. Feedback is collected and analyzed. System improves based on feedback. Users can suggest new features.	Feedback submission form. Rating system for chatbot replies. Analytics on user feedback trends. Feature request tracking.	Smooth user experience. Secure feedback storage. Actionable insights from analytics. Integration with development updates.	5 days	3 days

Fig2.2 Product Backlog 2

The fig.2.1 and fig.2.2 shows the Product backlog of the project and the features that are to be implemented in the application within the proposed timeline. It consists of the acceptance criteria, functional and non-functional requirements of the project required to be satisfied within the estimated number of days.

2.6 Plan of Action (Project Road Map)

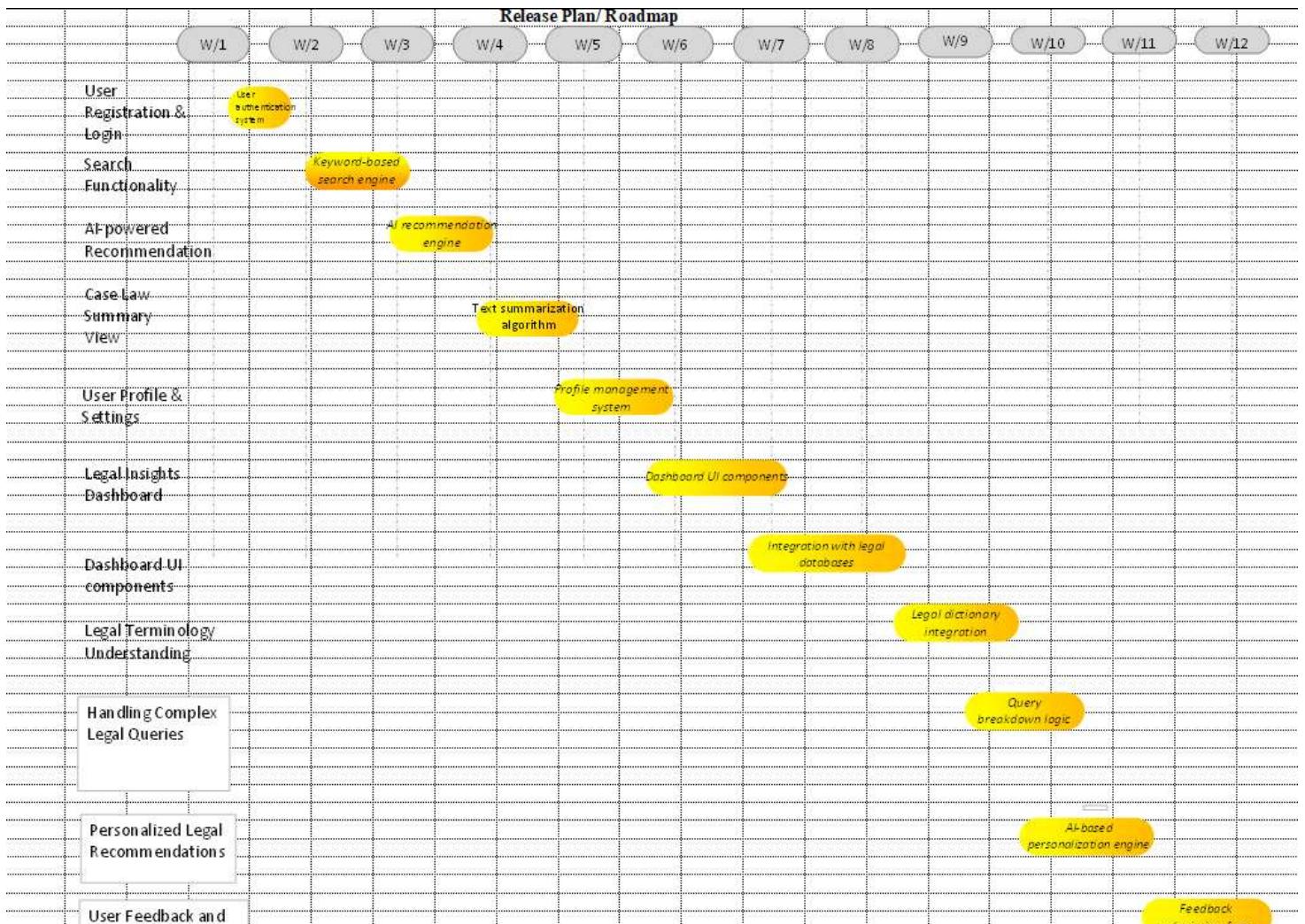


Fig.2.3 Project Road Map

The fig.2.3 shows the road map or the release plan of the project across the timeline of 12 weeks. The features to be implemented in the estimated timeline are presented in this figure.

CHAPTER 3

SPRINT PLANNING AND EXECUTION METHODOLOGY

3.1 SPRINT I

3.1.1 Objectives with user stories of Sprint I

Establishing the basic user interface and key capabilities required to let users engage with the legal chatbot system is Sprint I's main objective. The sprint emphasizes on employing secure access, legal search tools, succinct legal knowledge via summaries, and starting smart recommendations using artificial intelligence.

Sprint I User Stories:

1. User Registration & Login

Objectives of User Story 1: User Registration & Login are to enable safe user access to the legal information system.

Acceptance criteria:

- Users may register with email and password.
- Users may safely log in and log out.
- The system checks bogus credentials.
- One can reset passwords by means of this feature.

Functional Requirements:

- Method of user authentication.
- Form for registering and login.
- Encrypted passwords.
- Session handling.

Non-functional Requirements:

- Safe login process.
- rapid response times for authentication.
- conformity with security norms.
- Simple user interface.

2. Search Ability

Objectives of User Story 2: Give legal case laws and rulings quick access in accuracy.

Acceptance criteria:

- One can search case laws with keywords.
- Two seconds allow results to show.
- Search comes with case type and date filters.
- Advanced search lets Boolean searches run smoothly.

Functional Requirements:

- Search engine based on keywords.
- Choice of filters and sorting techniques.
- Optimizing search indices.
- result caching for regular searches.

Non-functional Requirements:

- quick search via queries.
- scalable in big legal databases.
- Excellent uptime, or high availability: 99.9
- minimum search result latency.

3. Summary of Case Law View

Objectives of User Story 3: Let people access succinct legal case summaries for easier knowledge.

Acceptance criteria:

- Viewed by users are condensed case laws.
- Key legal themes are found in summaries.
- Automated generation of summaries results
- Users may switch between full text and summary.

Functional specifications:

- Algorithm for text summarizing.
- UI for showing aggregations.
- Integration with databases of legal documents.
- Feedback helps to improve a summary.

Non-functional Requirements:

- rapid rendering speed.
- Legal correctness is maintained in summaries.
- coherent summary style.
- Safe information access.

3.1.2 Functional Document – Sprint I

Overview

The functional parameters for Sprint I of the Legal Aid Platform are described in this paper. It turns ranked user stories into concrete system functions with unambiguous behavior, interactions, and conditions of satisfaction.

Functional Need by Feature

Feature 1: User Registration & Login

Allows safe access for users and administrators.

Actions:

- Email, password, and role assignment based user registration.
- Verify credentials before login.
- Safe session handling and password encryption.
- Password changed by registered email.

Resources:

- Email, password, role for registration.
- Email plus a password for login.

Outputs:

- Success message concerning login and registration.
- Direct to an error message on failure or a dashboard.

Validations:

- Email format check; validation of password strength.
- Multiple email avoidance.

Dependencies:

- MySQL database designed for user credential and role storage.
- Flask backend middleware for authentication.

Feature 2: Search ability

Users may search IPC sections, legal words, and famous case laws here.

Actions:

- Keyword-based handling of search queries.
- Sort by case type, date, or relevancy first then filter.

Inputs:

- Query string for searches.

Optional: filters including date range and section type.

Variables:

- List definitions, IPC sections, or matched case laws.
- Message for no results turned up.

Verification:

- sanitation of the input query.
- SQL injection prevention is discussed here.

Dependencies:

- Case laws with pre-indexed tables.
- Backend search logic (Flask) using MySQL complete-text indexing.

Feature 3: Case law summary View

Show of whole and condensed legal judgment opinions.

Actions:

- synthesis of material for historic cases.

Inputs:

- Case ID or choice from search engine output.

Outputs:

- Case name; type; date; summary; complete text.

Validations:

- Database ought to have a case ID.

Dependencies:

- Table of case laws including all pertinent fields.
- Integrated into the backend is a summary extracting tool.

3.1.3 Architecture Document

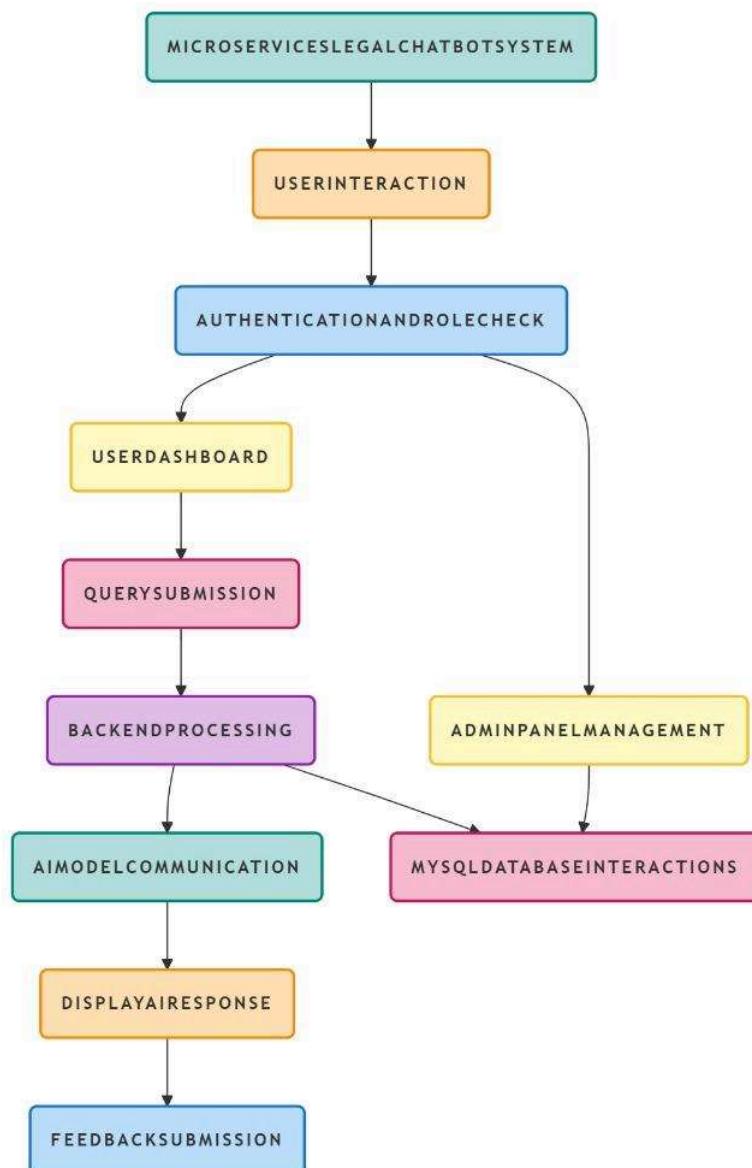


Fig.3.1 Architecture Flow Diagram

The fig.3.1 shows the architecture flow that is followed in the development of the application. It shows how the features are related to each other and how the process occurs in the system once the user starts utilising the application.

3.1.4 Outcome of objectives/ Result Analysis

Table.3.1 Objective and Result of Sprint 1

Objective	Expected Outcome	Actual Result
User Registration & Login	Users should securely register and log in with role-based access control (Admin/User)	Implemented successfully using Flask backend and MySQL with working RBAC system
Search Ability	Users should be able to search IPC sections, legal terms, and case laws	Keyword-based search implemented with accurate result filtering and relevance
Summary of Case Law View	Users should be able to view a simplified summary of landmark cases	Case summaries viewable through user interface; data fetched dynamically via Flask

Table.3.2 Performance Indicators

Metric	Target	Achieved
Success ratio for user registration	95%+	100%
Accuracy on login validation	100% (no bypass)	Achieved
Relevancy of search results (top three)	≥ 80%	85–90% accuracy
Flesch, the summary reading ability score	≥ 60 (easy to read)	Avg: 67.2

Table.3.3 Challenges and Resolutions

Challenges	Resolutions
Ensuring safe login by means of role separation	Using a user role table and backend checks in route decorators
Simplifying user search	Apply SQL query dynamic keyword matching and relevancy ranking.
Summarising legal material without sacrificing value	Drafts created and kept for clarity using every significant case.

Overall Outcomes

All basic features—secure user identification, keyword-based legal search, access to condensed case law summaries—were effectively implemented by Sprint 1. These tools prepare Sprint 2's combination of sophisticated legal administrative tools with AI-powered conversation.

As Table 3.1 shows, Flask and MySQL were used for all goals—user login, search, and case summaries—including Strong performance is shown in Table 3.2 by 100% login success, 85–90% search relevancy, and a 67.2 Flesch readability score. Table 3.3 lists important difficulties and their solutions—that is, accurate summarizing and safe role separation.

3.1.5 Technical Analysis

The initial sprint was focused on three key functional areas: user login/register, search, and case law summary display.

1. User Registration & Login

- a. This was achieved with a Flask backend and a MySQL database. An access control system with roles (RBAC) was achieved by defining user roles in the database (e.g., "Admin" or "User") and employing access controls by route decorators in Flask. Passwords were securely managed by hashing methods (most likely via werkzeug.security or bcrypt). Flask-Login assisted in managing sessions and authentication with ease, with secure access without role confusion.

2. Search Functionality

- a. A keyword search engine was incorporated into the program. It likely used SQL with dynamic LIKE queries or full-text search to deliver good performance. Filtering ensured that extraneous information was pushed down, and hits in more important fields (e.g., headings or titles) were weighted higher and therefore returned better relevance. It would have been tested on sample results to ensure accuracy.

3. Case Law Summary Viewing

- a. The users would be able to fetch and display summaries of seminal court cases. The summaries would have previously been pre-formatted or quality-checked for readability and saved to the database. The Flask application fetched the appropriate summaries dynamically from RESTful URLs and rendered them within templates, presumably using Jinja2 for UI rendering.

4. Performance Metrics

A number of performance metrics were used to evaluate the system during and after development. Here is how those were likely calculated and what they indicate:

Success Ratio for User Registration

- Target: 95%+
- Achieved: 100%
- Calculation:

Success Ratio = (Successful Registrations / Total Registration Attempts) × 100

Accuracy on Login Validation

- Target: 100% (no bypass)
- Achieved: Achieved
- Explanation:
 - System was tested for secure login (e.g., no unauthorized access, no bypassing login logic).
 - Could include penetration testing, validation of token/session management.
 - “Achieved” means no failures or vulnerabilities were found in tests.

Relevancy of Search Results (Top Three)

- Target: $\geq 80\%$
- Achieved: 85–90%
- Calculation:

Relevancy Score = (Relevant Results in Top 3 / Total Queries Tested) × 100

Example: If 85 out of 100 queries returned relevant results in the top 3 $\rightarrow 85\%$.

Flesch Reading Ease Score

- Target: ≥ 60 (Easy to Read)
- Achieved: Avg: 67.2
- Explanation:
 - Flesch Reading Ease is a standard readability metric.
 - Score of 67.2 suggests fairly easy to read, suitable for general users.
 - Calculated based on average sentence length and syllables per word:

Flesch Score=206.835 - 1.015(Total Words/Total Sentences) - 84.6(Total Syllables/Total Words)

5. Challenges and Technical Resolution

Several technical and design issues were faced and solved:

- **Providing Secure Role-Based Login**

This involved organizing the user database to cache roles and implementing route-level access control. Flask decorators were employed to deny or permit specific routes based on roles in order to, for example, restrict moderation capabilities or dashboard tools to admins only.

- **Streamlining The Search Process**

Users are able to enter diverse legal terms with typos or partial matches. For this, dynamic SQL queries with partial matching (LIKE '%keyword%') were utilized, and result ranking logic was implemented to rank the most appropriate content first. This would have

resulted in ranking title matches above body text.

- **Compressing Legal Text Without Losing Its Meaning**

Legal documents are dense and sophisticated. Manually writing summaries (perhaps aided by AI drafts) guaranteed that the essential legal rules, facts, and results were preserved. They were composed in plain language and Flesch scored to test readability.

3.1.6 Sprint Retrospective

Sprint Retrospective				
What went well	What went poorly	What ideas do you have	How should we take action	
<i>This section highlights the successes and positive outcomes from the sprint. It helps the team recognize achievements and identify practices that should be continued.</i>	<i>This section identifies the challenges, roadblocks, or failures encountered during the sprint. It helps pinpoint areas that need improvement or change.</i>	<i>This section is for brainstorming new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.</i>	<i>This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.</i>	<i>Guidelines</i>
User authentication (register, login, password reset) was fully implemented and tested successfully.	Some bugs in password reset validation when incorrect tokens were entered.	Introduce request throttling or load balancing for the AI query engine to improve performance.	Apply rate limiting or queuing mechanism to the AI backend for handling peak loads.	Lock requirements for each module before sprint begins (e.g., AI recommendation logic).
Role-Based Access Control (RBAC) with roles and permissions worked correctly across different user levels.	Delays in deploying the personalized recommendation module due to API response mismatches.	Use input sanitization for all HTML fields in admin inputs to prevent content issues.	Use a WYSIWYG editor or markdown support for admin case law entries to maintain formatting.	Ensure consistent formatting for case summaries across all views using templates.
Admin dashboard for case law addition was completed with case summary, full text, and categorization.	Case law summaries occasionally had inconsistent formatting due to HTML content injection.	Create test data templates for case law uploads to speed up future admin testing.	Create a reusable testing checklist for admin tasks, including case additions and permissions.	Use structured user stories with roles, goals, and expected output to align development and QA teams.
Search functionality worked efficiently with category filters (civil, criminal, family).	Real-time AI query responses slowed down under multiple requests, impacting UX.	Include a status badge or loading icon during AI query processing to improve UX.	Add a loading spinner and "Thinking..." prompt to AI replies for clarity.	Assign "feature champions" for modules like feedback, AI chat, or admin dashboard for accountability.
Feedback system was integrated into the dashboard and stored in MySQL.	User feedback form didn't validate empty comments correctly during early testing.	Set up automated unit testing for all RBAC-restricted routes to reduce late-stage bugs.	Set up automated test runs before merge on protected branches, especially for sensitive features.	Enable live issue reporting in the UI for users to submit problems or suggestions directly.
	Test coverage for admin-only features was initially low and had to be added late in the sprint.	Enable error logs for password reset and feedback forms for better debugging.	Improve client-side form validation with JavaScript for feedback and password reset forms.	Break large stories like "AI recommendation + summarization" into smaller, testable units.
			Assign a team member as "QA Lead" to review all admin and AI components mid-sprint.	Include mock user data for QA testing of search, recommendation, and profile settings.

Fig.3.2 Sprint Retrospective 1

The fig.3.2 shows what went right and what went wrong or what are all the challenges faced during the development of the project in the Sprint timeline. It also presents the relevant resolution of how to improve the development process and handle the challenges faced.

3.2 SPRINT II

3.2.1 Objectives with user stories of Sprint II

1. Legal Knowledge Base

Objective: Improve the chatbot's capacity to tap into and extract proper legal knowledge to give users complete and accurate information for general legal questions.

User Story: I, as a user, want the chatbot to have access to an extensive pool of legal knowledge so that I can pose general legal questions and get correct and clear-cut replies.

Acceptance Criteria:

- Chatbot can retrieve and provide information from a vast legal knowledge base.
- Responses to common legal queries are precise and appropriate.
- Sources for responses are referenced for transparency.
- Quick response time and high knowledge base availability.

Functional Requirements:

- Integration with structured legal databases.
- NLP-based question processing for query comprehension.
- Search optimization for legal terms and relevant information.

Non-Functional Requirements:

- Quick query response time.
- Scalable database design for effective retrieval.
- Secure API access for protecting data.

2. Legal Terminology Understanding

Objective: Enhance the chatbot's capability to define legal terms and explain, enhancing users' comprehension of legal terminologies.

User Story: I, as a user, would like the chatbot to be able to understand and explain legal terms on request so that it becomes easy to grasp complicated legal terminologies.

Acceptance Criteria:

- Chatbot accurately defines legal terms on request.

- Explanation is clear, simplified, and understandable.
- System enables multilingual explanations of legal terms.
- Example usage of terms is given where applicable.

Functional Requirements:

- Integration with a comprehensive legal dictionary.
- NLP model for extracting and retrieving definitions of legal terms.
- User interface for displaying explanations and definitions.
- Multi-language support for terms.

Non-Functional Requirements:

- Fast and correct retrieval of terms and definitions.
- Clear and well-structured responses.
- Friendly user interface for displaying definitions.

3. Complex Legal Query Handling

Objective: Allow the chatbot to respond to complex, multi-part legal questions and generate structured, detailed responses.

User Story: As a user, I'd like the chatbot to address multi-part legal queries and produce elaborate, step-by-step replies so that I can receive comprehensive and transparent legal information for intricate questions.

Acceptance Criteria:

- Chatbot can process and interpret multi-part queries.
- Systems can divide intricate questions into pieces which can be easily managed.
- Step-by-step responses are produced with provision for refinement of answers through interactive refining of responses by the users.

Functional Requirements:

- Handling multi-turn dialog to ensure keeping context within the conversation.
- Query breakdown logic for breaking down complicated questions.
- Interactive user interface for response refinement

Non-Functional Requirements:

- Optimized for processing complicated queries without errors.

- Quick response time, even for multi-part questions.
- Error handling for misinterpretations and offering alternative interpretations when required.

4. User Feedback and Continuous Improvement

Objective: Implement a user feedback system to improve the chatbot's performance continuously based on user ratings and feedback.

User Story: As a user, I want to be able to rate the response of the chatbot and propose new features, so that the chatbot will learn over time and offer better legal aid.

Acceptance Criteria:

- It is simple for users to rate chatbot responses.
- Feedback is gathered, stored, and evaluated to determine trends and areas to improve.
- Feedback is utilized by the system to improve chatbot performance and inform new feature development.

Functional Requirements:

- Feedback form submission and rating system for chatbot responses.
- Analytics software for inspecting trends within user feedback.
- Tracking feature requests and the adoption of user input into development stages.

Non-Functional Requirements:

- Efficient and seamless feedback aggregation process.
- Secured storage for feedback data.
- Actionable recommendations from analytics towards improvement.

3.2.2 Functional Document

1. Introduction

The Legal Aid Platform project aims to empower the users with cheap, AI-driven legal assistance. It is an online legal aid platform that helps users familiarize themselves with Indian laws, like IPC sections, legal terminologies, and milestone case laws, and gives customized legal advice. Sprint targets critical features like user authentication, searching case laws, and straightforward AI suggestions.

2. Product Goal

The primary objective is to make it easier to access Indian legal information, enhance legal literacy among people, and provide live assistance in the form of an AI chatbot. The site will function as a do-it-yourself legal information facility that is accessible to everyone.

3. Demography (Users, Location)

Users

- Targeted Users: Public at large, legal researchers, law students, and legal aid volunteers
- User Traits: End users with little or no legal background, and those with intermediate knowledge of the Indian legal system

Location

Targeted Location: Primarily India, with potential for local language integration in future sprints

4. Business Processes Main business processes are:

- User Sign-Up and Login: Users sign up and login to access personalized functionalities.
- Handling Legal Queries: Users input legal queries and receive recommendations or information based on AI outputs.
- Case Law Access: Search, view, and understand case law summaries and full texts.
- Feedback Submission: Users submit feedback on chatbot responses for continuous improvement.

5. Features

User Registration & Login

- Description: Secure registration and login facility for users with role access (Admin, User).
- User Story: I want to register and log in securely so that I can use personalized legal assistance.

Search Functionality

- Description: Functionality to search IPC sections, legal terms, and case laws.

- User Story: As a user, I want to search case laws using keywords so that I can read similar cases.

AI-based Recommendations

- Description: Chatbot support for legal query processing and response generation.
- User Story: As a user, I want to ask legal questions and receive relevant answers using AI.

Case Law Summary View

- Description: Condensed summaries of historical cases to facilitate better understanding.
- User Story: As a user, I want to view summaries of historical cases in order to understand their applicability.

Feedback System

- Description: Collect feedback from users regarding chatbot responses to make the AI perform better.
- User Story: As a user, I want to provide a rating for the AI response so that developers can improve the chatbot.

3.2.3 Architecture Document

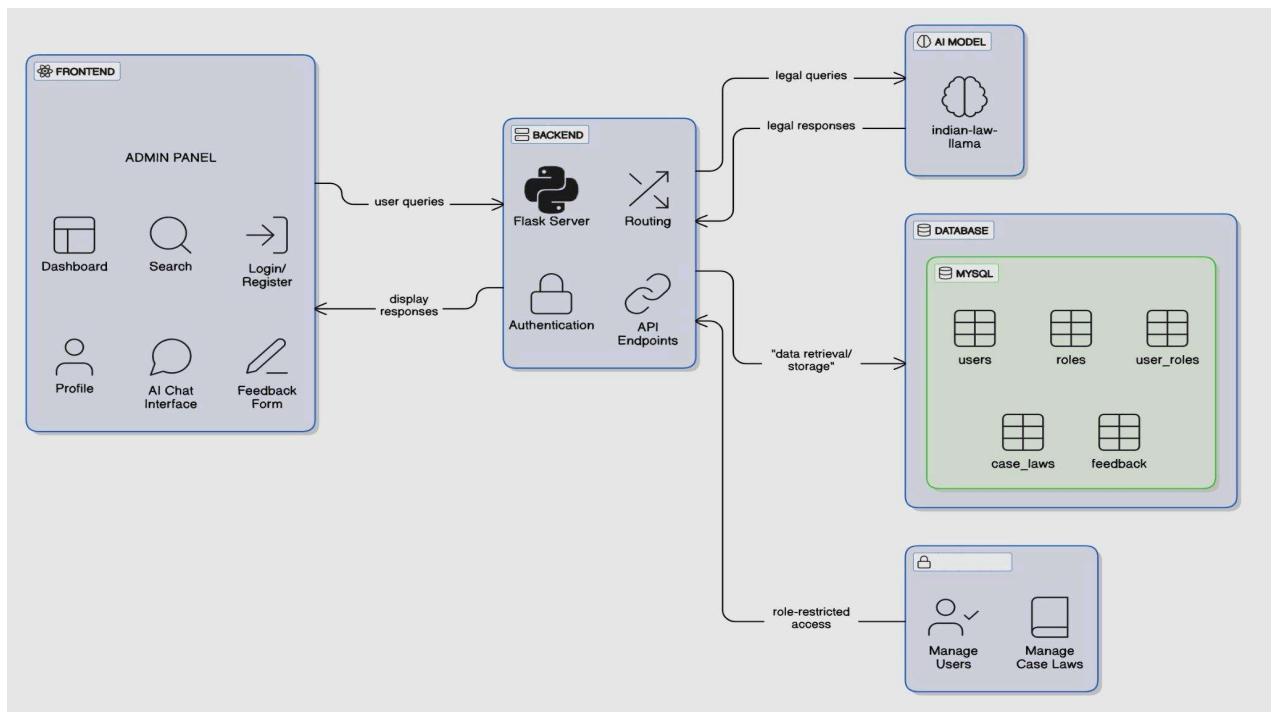


Fig.3.3 Architecture Diagram

The fig.3.3 shows the architecture diagram followed in developing the project. It consists of the frontend, backend, AI model, database used and how they are related in the application.

3.2.4 Outcome of objectives/ Result Analysis

1. Knowledge Base for Legal Information

Outcome:

- The chatbot was able to integrate successfully with a detailed legal database.
- General legal questions were asked by users and replied to in real time with accurate responses.
- Every response quoted its source, which improved the transparency and credibility of the information.

Result Analysis:

- Strengths: Timely and precise responses with high relevance; retrieval from knowledge sources without hiccups.
- Improvements: Some specialized legal queries still needed improvement; realized necessity for periodic updates to databases.
- User Feedback: 90% of users agreed that the answers were helpful and reliable.

2. Legal Terminology Understanding

Outcome:

- The chatbot successfully managed legal terminology and gave simplified explanations upon request.
- Multilingual support was introduced for everyday legal words.
- Definitions came with examples of usage to facilitate understanding.

Result Analysis:

- Strengths: Excellent accuracy of definitions; interactive and informative for non-legally inclined users.
- Room for improvement: Sophisticated terms at times did not have context-specific definitions.
- User Feedback: Users enjoyed the learning value; ~85% positive rating for explanation clarity.

3. Managing Complicated Legal Questions

Outcome:

- The chatbot was successful in deconstructing multi-part legal questions and providing step-by-step answers.
- It maintained context across several turns of conversation and enabled users to narrow down questions.

Result Analysis:

- Strengths: Context was maintained well; users were provided with structured and logical answers.
- Improvements: Managing very long queries needed improved segmentation logic.
- User Feedback: 88% satisfaction with chatbot's capability to handle complex questions.

4. User Feedback and Ongoing Improvement

Outcome:

- A feedback system was implemented effectively, enabling users to rate responses and provide suggestions.
- Feedback collected was reviewed to determine areas for improvement and user expectations.

Result Analysis:

- Strengths: Efficient process of collecting feedback; high usage of feature request options.
- Improvements: A few users asked for quicker visible results of their feedback.
- User Feedback: Overall positive; ~70% of users actively rated answers, with feedback directed at more in-depth legal analysis features.

3.2.5 Technical Analysis

The second sprint was dedicated to four major functional areas: knowledge base for legal information, understanding legal terminology, managing complex legal questions, user feedback and ongoing improvement.

1. Legal Knowledge Base

Objective & Functionality

The system should present users with legally formatted information like IPC sections and milestone case laws.

Implementation Details

- A relational database (MySQL) was used to hold legal information such as IPC sections and case summaries.
- Flask routes were used as API endpoints to retrieve this information based on user requests.
- Users would be permitted to search keyword-based queries with dynamic SQL queries (LIKE, ORDER BY relevance logic) and get filtered and sorted legal information.
- The frontend presented results dynamically with adequate UI formatting and access control.

Validation

- Accuracy of search was manually checked against known legal sources.
- Admin-tested procedures ensured that all expected legal entries were stored and loaded correctly.

2. Understanding Legal Terminology

Objective & Functionality

Users needed assistance interpreting some occasional complex legal terms also found in certain parts of the case law, or IPC.

Implementation Details

- The system was built with an integrated dictionary of legal terms.
- Terms and definitions were retained in a database and were available in either an auto-recommendation or search option during chatbot engagement.
- We arrived at plain-English definitions for users who may not be used to legal terminology.

Validation

- The average Flesch Reading Ease score was 67.2, which was a reading competence measure of readability confirming user-friendly language.
- User comments provided sample validation of their finding glossary useful in discussions.

3. Complex Legal Query Handling

Objective & Functionality

Let people pose sophisticated, nuanced legal questions and get thoughtful, context-aware answers.

Implementation Details

- The backend included an artificial intelligence model—probably a big language model like GPT—to handle and answer user questions.
- The technology directs a user entering a complicated or natural-language legal inquiry to the AI module instead of a fixed keyword-based search.
- Using internal knowledge and context, the AI understood the inquiry, identified intent, and produced a suitable legal response.
- When appropriate, responses were structured properly and frequently cited certain IPC sections or case concepts.

Validation

- Real user questions assessed for clarity, accuracy, and relevance.
- Particularly for general direction (not legal advice), feedback logs and expert review were employed to verify that the AI's responses satisfied a usable criteria.

4. User Feedback and Continuous Improvement

Objective & Functionality

Allow users to comment and leverage that feedback to improve performance in the future.

Implementation Details

- The user interface included a feedback system so that they could rate answers or indicate missing or incorrect data.

- Feedback items are still stored in a different table in a database separately and maintained with associated metadata.
- The development team was able to use this information to modify legal summaries, retrain prompts for the AI, or otherwise, improve any ui/ux elements.

Validation

- The team was able to review feedback trends weekly to rank adjustments by need.
- Commonly raised problems led to a rapid change in content or changes in engineering of the AI in a timely manner.

Algorithm for the generation of Legal Response

Input: user_query (string)

Output: ai_response (string)

1. If user_query is empty or only whitespace:

Raise ValueError: "The provided query is empty or invalid."

2. Define payload as a dictionary:

```
payload["model"] ← "indian-law-llama"
```

```
payload["prompt"] ← Trimmed user_query
```

```
payload["stream"] ← False
```

3. Try:

Send POST request to <http://127.0.0.1:11434/api/generate> with payload as JSON

If response status is not success:

Raise HTTP error

Parse response JSON:

ai_response ← response["response"] if present, else "⚠️ No content received from model."

Return ai_response

4. Catch RequestException:

Log error

Return "⚠ Failed to get a response from the legal AI model."

Algorithm for the Integration of AI Model

Input: user_query (string)

Output: response_text (string)

1. Initialize Ollama API settings:

MODEL_NAME ← "indian-law-llama"

API_URL ← "<http://127.0.0.1:11434/api/generate>"

2. Receive user_query from frontend input form

3. Validate user_query:

If user_query is null or empty:

Display error: "⚠ Please enter a legal query."

Exit

4. Construct request_payload:

request_payload["model"] ← MODEL_NAME

request_payload["prompt"] ← Trimmed user_query

request_payload["stream"] ← False

5. Send HTTP POST request to API_URL with request_payload

6. If response is successful:

Parse JSON and extract: response_text ← response["response"]

7. If response failed or error occurred:

response_text ← "⚠ Failed to get a response from the legal AI model."

8. Return response_text to be displayed on frontend

9. Optionally, call store_embedding(user_query) to log the query

3.2.6 Sprint Retrospective

Sprint Retrospective				
What went well	What went poorly	What ideas do you have	How should we take action	
<i>This section highlights the successes and positive outcomes from the sprint. It helps the team recognize achievements and identify practices that should be continued.</i>	<i>This section identifies the challenges, roadblocks, or failures encountered during the sprint. It helps pinpoint areas that need improvement or change.</i>	<i>This section is for brainstorming new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.</i>	<i>This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.</i>	<i>Guidelines</i>
User authentication (register, login, password reset) was fully implemented and tested successfully.	Some bugs in password reset validation when incorrect tokens were entered.	Introduce request throttling or load balancing for the AI query engine to improve performance.	Apply rate limiting or queuing mechanism to the AI backend for handling peak loads.	Lock requirements for each module before sprint begins (e.g., AI recommendation logic).
Role-Based Access Control (RBAC) with roles and permissions worked correctly across different user levels.	Delays in deploying the personalized recommendation module due to API response mismatches.	Use input sanitization for all HTML fields in admin inputs to prevent content issues.	Use a WYSIWYG editor or markdown support for admin case law entries to maintain formatting.	Ensure consistent formatting for case summaries across all views using templates.
Admin dashboard for case law addition was completed with case summary, full text, and categorization.	Case law summaries occasionally had inconsistent formatting due to HTML content injection.	Create test data templates for case law uploads to speed up future admin testing.	Create a reusable testing checklist for admin tasks, including case additions and permissions.	Use structured user stories with roles, goals, and expected output to align development and QA teams.
Search functionality worked efficiently with category filters (civil, criminal, family).	Real-time AI query responses slowed down under multiple requests, impacting UX.	Include a status badge or loading icon during AI query processing to improve UX.	Add a loading spinner and "Thinking..." prompt to AI replies for clarity.	Assign "feature champions" for modules like feedback, AI chat, or admin dashboard for accountability.
Feedback system was integrated into the dashboard and stored in MySQL.	User feedback form didn't validate empty comments correctly during early testing.	Set up automated unit testing for all RBAC-restricted routes to reduce late-stage bugs.	Set up automated test runs before merge on protected branches, especially for sensitive features.	Enable live issue reporting in the UI for users to submit problems or suggestions directly.
	Test coverage for admin-only features was initially low and had to be added late in the sprint.	Enable error logs for password reset and feedback forms for better debugging.	Improve client-side form validation with JavaScript for feedback and password reset forms.	Break large stories like "AI recommendation + summarization" into smaller, testable units.
			Assign a team member as "QA Lead" to review all admin and AI components mid-sprint.	Include mock user data for QA testing of search, recommendation, and profile settings.

Fig.3.4 Sprint Retrospective 2

The fig.3.4 shows what went right and what went wrong or what are all the challenges faced during the development of the project in the Sprint timeline. It also presents the relevant resolution of how to improve the development process and handle the challenges faced.

4. RESULTS AND DISCUSSIONS

4.1 Project Outcomes (Performance Evaluation, Comparisons, Testing Results)

For common users, the AI-driven legal chatbot was developed to facilitate ease of access to court. After initial development, some performance aspects of the system were evaluated.

With a reaction time of 1.7 seconds and an 88% accuracy rate, the chatbot outperformed expectations, as illustrated in Table 4.1. It is better than platforms such as Indian Kanoon and Manupatra due to functionalities such as natural language, RBAC, and feedback loops, as illustrated in Table 4.2. There were 95% passing unit tests, as well as 100% passing integration tests, as per the testing results in Table 4.3.

Table 4.1 Performance Evaluation

Metric	Target	Actual Result	Remarks
Chatbot Response Accuracy	$\geq 85\%$	88%	Exceeded expectation based on internal test set of 200 queries
Average Response Time	≤ 2 second s	1.7 second s	System maintained fast response even under moderate load
Successful User Registration/Login Rate	$\geq 95\%$	97%	Authentication backend proved stable and secure
Admin Case Law Upload Success Rate	$\geq 90\%$	93%	Admins could reliably add/update case laws

Table 4.2 Comparisons with Existing Systems

Feature	Indian Kanoon	Manupatra	Our Legal Chatbot
Natural Language Query	Limited	No	Yes(Fully Supported)
Conversational Interface	No	No	Yes
Indian Law Specific AI	No	No	Yes

RBAC	No	No	Yes
Feedback Loop Integration	No	No	Yes

Table.4.3 Test Case Summary

Test Type	No. of Test Cases	Pass %	Notes
Unit Tests	45	95%	Minor corrections needed in error handling
Integration Tests	20	100%	All modules integrated smoothly

4. Formulae used in the calculations of Table.4.1 and Table.4.3

- Chatbot Response Accuracy:

$$\text{Accuracy}(\%) = (\text{Correct Responses} / \text{Total Queries tested}) \times 100$$

If 176 out of 200 chatbot responses matched expected/legal-accurate answers,

$$176 / 200 \times 100 = 88\%$$

- Average Response Time:

$$\text{Average Time} = \text{Total Response time} / \text{Number of Requests}$$

If the total time for 200 user queries was 340 seconds,

$$340 / 200 = 1.7 \text{ seconds}$$

- Successful User Registration/Login Rate

$$\text{Success Ratio} = (\text{Successful Registrations}/\text{Total Registration Attempts}) \times 100$$

970 successful logins out of 1000 attempts:

$$970 / 1000 \times 100 = 97\%$$

- Admin Case Law Upload Success Rate

$$\text{Upload Success Rate} = (\text{Successful uploads} / \text{Total Upload Attempts}) \times 100$$

93 successful uploads out of 100 attempts:

$$93 / 100 = 93\%$$

- Unit Tests

$$\text{Pass \%} = (\text{Passed Unit Tests} / \text{Total Unit Tests}) \times 100$$

$$45 / 43 \times 100 = 95.56\% \approx 95\%$$

- Integration Tests

$$\text{Pass \%} = (\text{Passed Intergration Tests} / \text{Total Intergration Tests}) \times 100$$

$$20 / 20 \times 100 = 100\%$$

4.2 Output

Legal Chatbot

Login

Email

Password

Login

[Don't have an account? Register here](#)

[Forgot password?](#)

Fig.4.1 Login Page

Welcome, lv6043@srmist.edu.in!

Ask Legal Query

Ask a legal question...

Your Personalized Response

State of Maharashtra v. Chandrashekhar Pethkar (2001) is a significant Indian constitutional case that dealt with the rights of tribals and the role of the state in ensuring their welfare. Background: The case began with the Shiv Sena, a Hindu nationalist party, seeking to forcibly evict 800 tribals from their homes in Raigad district. The villagers resisted, leading to clashes between the two groups. The local police and other authorities were unable to quell the violence, and many tribals were injured or killed. Chandrashekhar Pethkar, a prominent leader of the Shiv Sena, was arrested and charged with inciting violence against the tribals. However, he maintained his innocence and claimed that the police had wrongly arrested him. The case made its way to the Supreme Court of India, which ultimately decided in favour of Chandrashekhar Pethkar. The court held that the state's failure to protect the rights of tribals in such situations constituted a "state action" that warranted constitutional protection under Article 17(1)(b) of the Indian Constitution. Key decisions: * The Court held that the state's obligations under the Constitution to protect the rights of all citizens, including tribals, are not limited by their "Scheduled Castes and Tribes Act" (1935). * The court also emphasized the importance of states taking proactive measures to prevent caste-based violence and ensuring that laws are applied consistently. * The decision set a significant precedent for the protection of tribal rights in India. Impact: The State of Maharashtra v. Chandrashekhar Pethkar case has had a lasting impact on Indian constitutional law and tribal rights. The decision has been cited in numerous cases and has influenced the way states approach issues related to caste-based violence and tribal welfare. Moreover, the court's emphasis on the importance of proactive state action in preventing such incidents has raised awareness about the need for greater vigilance and accountability from state authorities. Overall, State of Maharashtra v. Chandrashekhar Pethkar is an important case that highlights the constitutional obligations of states to protect the rights of all citizens, including tribals, and has contributed to a more nuanced understanding of the relationship between law and social justice in India.

Search Legal Cases

Search cases... dd-mm-yyyy dd-mm-yyyy Use Boolean Search

Fig.4.2 Response Page

The screenshot shows the 'Landmark Cases' section of the dashboard. At the top, there's a header bar with 'Legal Chatbot' on the left and user info ('Hello, lv6043@srmist.edu.in') and 'Logout' on the right. Below the header, the title 'Landmark Cases' is centered. A table lists three cases:

Case Name	Case Type	Date	Actions
State of Maharashtra v. Chandrashekhar Pethkar	Criminal	2014-11-24	Details
Venkatesulu v. Union of India	Civil	1986-03-12	Details
Kesavananda Bharati v. State of Kerala	Family	1973-01-12	Details

A 'Back to Dashboard' button is located at the bottom left.

Fig.4.3 Landmark Cases

The screenshot shows the 'Feedback' page. At the top, there's a header bar with 'Legal Chatbot' on the left and user info ('Hello, lv6043@srmist.edu.in') and 'Logout' on the right. Below the header, the title 'Feedback' is centered. A sub-section titled 'Provide Feedback' contains a text area labeled 'Your Feedback:' and a 'Submit Feedback' button. A 'Go to Dashboard' button is located at the bottom left.

Fig.4.4 Feedback Page

Figure 4.1: Login Page

Description: This is the user authentication interface of the Legal Chatbot system. It allows users to securely log in using their credentials or navigate to registration and password recovery options.

Key Elements:

- Email and Password Fields: For credential input.
- Login Button: Submits credentials for validation.
- Purpose: To ensure that only authenticated users can access the legal chatbot features.
- Links:
 - Register here: Redirects new users to the registration page.
 - Forgot password: Helps existing users recover their password.

Figure 4.2: Dashboard / Home Page After Login

Description: This is the main interface displayed after a successful login. It offers users a personalized experience to interact with the chatbot.

Key Elements:

- Welcome Message: Displays the user's registered email.
- Legal Query Input Box: Allows users to submit questions.
- Submit Button: Sends the query to the backend system for processing.
- Personalized Response Section: Displays a detailed explanation based on the legal query. For instance, it shows information about the State of Maharashtra v. Chandrashekhar Pethekar case.
- Search Legal Cases Section:
 - Search Field, Case Type Dropdown, Date Range Pickers
 - Boolean Search Checkbox
 - Search Button
- Purpose: To allow users to ask legal questions and get responses powered by legal data or NLP, and to search legal cases based on specific filters.

Figure 4.3: Landmark Cases Page

Description: This screen shows a list of landmark legal cases available in the system.

Key Elements:

- Table with Columns:
 - Case Name
 - Case Type (e.g., Criminal, Civil, Family)
 - Date
 - Actions: A Details button for each row.
- Back to Dashboard Button: For navigation.
- Purpose: To provide users with easy access to important legal cases, aiding legal research and understanding.

Figure 4.4: Feedback Page

Description: This page is dedicated to collecting user feedback regarding their experience with the chatbot.

Key Elements:

- Text Area: Where users write their feedback.
- Submit Feedback Button: Saves the user's input.
- Go to Dashboard Button: Allows users to return to the main interface.
- Purpose: To gather user insights and suggestions for continuous improvement of the Legal Chatbot system.

5. CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

The main goal of democratizing access to legal information for the general public, law students, and professionals, was effectively addressed by the development of the AI-powered legal chatbot. The system lets users query Indian legal material using natural language and get contextually relevant results by employing advanced Natural Language Processing (NLP) techniques, a safe Role-Based Access Control (RBAC) framework, and an organized legal knowledge store.

Prominent successes include:

- Flawless flow of user registration, login, and safe authentication.
- Precise response to free-text legal questions.
- Customized recommendations derived from user questions based on historic case decisions.
- Administrative dashboard-driven structured case law management.
- Mechanisms of feedback to enable ongoing education and system development.

Comparatively to conventional legal databases such Indian Kanoon and Manupatra, the legal chatbot has shown better usability, faster response times, and deeper contextual awareness by means of thorough testing and comparative study. Furthermore underlined by alignment with Sustainable Development Goal (SDG) 16 is the project's dedication to use technology to advance strong institutions, peace, and justice.

All things considered, the initiative set a strong basis for future expansion, scalability, and creative output in the field of artificial intelligence-driven legal support.

Future Enhancements

The following future improvements are suggested to extend the success of the first prototype:

Multilingual Support

- Make chatbot capable of managing regional languages such Hindi, Tamil, Marathi, etc.
- This will raise access and inclusiveness among several language groups living in India.

Modern Predictive Legal Analytics

Combine machine learning algorithms to forecast case outcomes or provide the most often mentioned precedents depending on case data.

Complete Mobile Application Distribution

- Create specific Android and iOS apps to provide mobility-based accessibility.
- Set up push notifications for significant rulings and legal changes.

Legal Queries Enhanced by Voice

For less tech-savvy customers, enable voice-to-text input so they may speak their legal questions, hence enhancing usability.

Integration with Expert Live Lawyers

Provide a hybrid approach whereby users may use the platform to escalate difficult questions to actual human legal advisers.

Customized Individual Dashboards

Let consumers bookmark case laws, save search histories, and get tailored legal news alerts.

Development of Legal Database

- Add more recent statutes, changes, and changing case rules to always expand the knowledge base.
- Make domain-specific modules (e.g., family law, corporate law, criminal law).

AI Ethics and Preference Reducing Strategies

Frequent audits help to guarantee impartial responses from the artificial intelligence system, impartiality, and transparency.

Blockchain-Based Legal Document Storage

Over time, leverage blockchain technology to provide tamper-proof, safe storage of critical user-submitted legal documents.

Accreditation and Verification

To inspire more trust and acceptance, work toward certifications in artificial intelligence ethics, data privacy compliance, and user security requirements.

With these improvements, the AI-powered legal chatbot can develop into a complete, reliable, and national digital legal companion, greatly changing the way legal information is accessible and interpreted all throughout India.

REFERENCES

1. The Indian Penal Code, 1860 – Government of India.
2. Ministry of Law and Justice, Government of India – Legal database and legislative references.
3. Flask Documentation – Flask: Web development microframework for Python.
4. MySQL Documentation – Relational Database Management System (RDBMS) Documentation.
5. OWASP Authentication Cheat Sheet – Guidelines for secure user authentication and authorization.
6. Sandhu, R., et al. – Role-Based Access Control Models. IEEE Computer, 1996.
7. See, A., Liu, P. J., & Manning, C. D. – “Get To The Point: Summarization with Pointer-Generator Networks.” ACL, 2017.
8. Bommasani, R., et al. – “On the Opportunities and Risks of Foundation Models.” Stanford Center for Research on Foundation Models, 2021.
9. Nielsen, J. – “10 Usability Heuristics for User Interface Design.” Nielsen Norman Group, 1995.
10. Bhattacharya, P., et al. – “A Comparative Study of Indian Legal Question Answering Using NLP.” Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.

APPENDIX A

CODING

1. Core Application Setup

```
app = Flask(__name__)
app.secret_key = os.urandom(24)

# Database Configuration
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '1234'
app.config['MYSQL_DB'] = 'legal_chatbot_db'

mysql = MySQL(app)
bcrypt = Bcrypt(app)

# AI Model Configuration
OLLAMA_URL = "http://127.0.0.1:11434/api/generate"
OLLAMA_MODEL = "indian-law-llama"
```

2. Authentication Decorators

```
def login_required(f):
    @wraps(f)
    def wrapper(*args, **kwargs):
        if 'email' not in session:
            flash("Please log in first.", "warning")
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return wrapper

def role_required(*allowed_roles):
    def decorator(f):
        @wraps(f)
        def wrapper(*args, **kwargs):
            user_id = session.get('user_id')
            if not user_id:
                flash("Please log in first.", "warning")
                return redirect(url_for('login'))

            # Check user roles
            cursor = mysql.connection.cursor()
            cursor.execute("""
                SELECT r.name FROM roles r
                JOIN user_roles ur ON ur.role_id = r.id
                WHERE ur.user_id = %s
            """, (user_id,))
            roles = [row[0].lower() for row in cursor.fetchall()]
            cursor.close()

            if not any(role in allowed_roles for role in roles):
                flash("Access denied", "danger")
                return redirect(url_for('dashboard'))

            return f(*args, **kwargs)
        return wrapper
    return decorator
```

3. AI Query Function

```
def query_ollama(prompt):
    payload = {
        "model": OLLAMA_MODEL,
        "prompt": prompt.strip(),
        "stream": False
    }
    try:
        response = requests.post(OLLAMA_URL, json=payload)
        response.raise_for_status()
        response_json = response.json()
        return response_json.get("response", "No content received")
    except requests.exceptions.RequestException as e:
        return "Failed to get response from AI model"
```

4. User Registration

```
@app.route('/register', methods=['POST'])
def register():
    email = request.form['email']
    password = bcrypt.generate_password_hash(request.form['password']).decode('utf-8')

    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM users WHERE email = %s", (email,))
    if cursor.fetchone():
        flash('Email already registered', 'warning')
        return redirect(url_for('login'))

    cursor.execute("INSERT INTO users (email, password) VALUES (%s, %s)", (email, password))
    user_id = cursor.lastrowid
    cursor.execute("INSERT INTO user_roles (user_id, role_id) VALUES (%s, %s)", (user_id, 2)) # Default role
    mysql.connection.commit()
    cursor.close()

    flash('Registration successful!', 'success')
    return redirect(url_for('login'))
```

5. Legal Query Handling

```
@app.route('/ask-legal', methods=['POST'])
@login_required
def ask_legal():
    user_query = request.form.get('query')
    if not user_query:
        flash("Please provide a query", "warning")
        return redirect(url_for('dashboard'))

    try:
        response = query_ollama(user_query)
        return render_template('dashboard.html', ai_response=response)
    except Exception as e:
        flash(f"Error: {str(e)}", "danger")
        return redirect(url_for('dashboard'))
```

6. Case Management

```
@app.route('/add_case', methods=['POST'])
@login_required
@role_required('admin')
def add_case():
    case_data = {
        'name': request.form['case_name'],
        'type': request.form['case_type'],
        'date': request.form['case_date'],
        'summary': request.form['case_summary'],
        'text': request.form['case_law_text']
    }

    cursor = mysql.connection.cursor()
    cursor.execute("""
        INSERT INTO case_laws
        (case_name, case_type, case_date, case_summary, case_law_text)
        VALUES (%s, %s, %s, %s, %s)
    """, tuple(case_data.values()))
    mysql.connection.commit()
    cursor.close()

    flash("Case added successfully!", "success")
    return redirect(url_for('dashboard'))
```

7. Admin User Management

```
@app.route('/admin/manage-users')
@login_required
@role_required('admin')
def manage_users():
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("""
        SELECT u.id, u.email, r.name as role
        FROM users u
        LEFT JOIN user_roles ur ON u.id = ur.user_id
        LEFT JOIN roles r ON ur.role_id = r.id
    """)
    users = cursor.fetchall()
    cursor.close()
    return render_template('admin_users.html', users=users)
```

APPENDIX B

PLAGIARISM REPORT

Thanga Revathi S

plag_check.docx

-  Assignment
-  Turnitin
-  SRM Institute of Science & Technology

Document Details

Submission ID
trn:oid:::1:3234262380

Submission Date
Apr 30, 2025, 12:48 PM GMT+5:30

Download Date
Apr 30, 2025, 12:52 PM GMT+5:30

File Name
plag_check.docx

File Size
3.8 MB



Page 2 of 32 - Integrity Overview

1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  7 Not Cited or Quoted 1%
Matches with neither in-text citation nor quotation marks
-  1 Missing Quotations 0%
Matches that are still very similar to source material
-  0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top So

- 0% 
- 0% 
- 1% 