

# **AI-VAKEEL: AN AI-POWERED PLATFORM FOR SMART LEGAL QUERY RESOLUTION IN THE INDIAN JUDICIARY**

## **A PROJECT REPORT**

*Submitted by,*

<b>Vishal G Dhavali</b>	<b>20221LCS0024</b>
<b>Bhavyashree S</b>	<b>20221CSE0588</b>
<b>Tejas SP</b>	<b>20211CSE0500</b>
<b>RamaKrishna</b>	<b>20211CSE0517</b>
<b>Aditya Gupta</b>	<b>20211CSE0300</b>

*Under the guidance of,*

**Ms. S Thabassum Khan**  
**Assistant professor**  
**School of Computer Science and Engineering**  
**Presidency university, Bangalore**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

**MAY 2025**

## **PRESIDENCY UNIVERSITY**

### **SCHOOL OF COMPUTER SCIENCE ENGINEERING**

#### **CERTIFICATE**

This is to certify that the Project report **AI-VAKEEL: AN AI-POWERED PLATFORM FOR SMART LEGAL QUERY RESOLUTION IN THE INDIAN JUDICIARY** being submitted by Vishal G Dhavali(20221LCS0024), Bhavyashree S (20221CSE0588), Tejas SP(20211CSE0500), RamaKrishna (20211CSE0517), Aditya Gupta(20211CSE0300) in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

**Ms. S Thabassum khan**

Assistant Professor  
School of CSE  
Presidency University

**Dr. Asif Mohammed H.B**

Associate Professor & HOD  
School of CSE  
Presidency University

**Dr. MYDHILI NAIR**

Associate Dean  
School of CSE & IS  
Presidency University

**Dr. SAMEERUDDIN KHAN**

Pro-Vc School of Engineering,  
SOCSE SOISE,  
Dean -School of CSE & IS  
Presidency University

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled in **AI-VAKEEL: AN AI-POWERED PLATFORM FOR SMART LEGAL QUERY RESOLUTION IN THE INDIAN JUDICIARY** partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms. S Thabassum Khan Professor, School of Computer Science Engineering Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

**Vishal G Dhavali**  
**Bhavyashree S**  
**Tejas S P**  
**Rama Krishna**  
**Aditya Gupta**

## ABSTRACT

In today's rapidly evolving digital landscape, the Indian legal system faces a growing demand for transparent, accessible, and citizen-centric legal support. Despite the proliferation of legal databases and online case repositories, the average citizen continues to struggle with understanding the law, accessing relevant information, and navigating legal procedures without formal assistance. Legal professionals and judiciary members too face challenges in retrieving precise legal references amidst vast and unstructured data. These limitations underscore a critical need for intelligent, accessible, and personalized legal information systems. The *AI Vakeel* project was conceptualized to address this gap through an artificial intelligence-powered legal assistant capable of interacting in natural language, understanding user context, and delivering accurate legal responses based on pre-verified documents.

Designed to serve three key user categories—public users, practicing advocates, and judiciary officers—*AI Vakeel* operates through a secure, role-sensitive interface that respects user authorization while maximizing the clarity and relevance of its outputs. The system utilizes a retrieval-driven methodology that semantically interprets user queries and aligns them with the most relevant portions of the legal corpus. Legal documents, including acts, case summaries, FAQs, and judicial precedents, are ingested through a controlled admin module, cleaned, segmented, and stored in a format that allows efficient context-based retrieval. The assistant supports continuous learning and dynamic content updates, ensuring that the knowledge base remains current and scalable without major architectural overhaul.

The system has undergone rigorous testing through simulated legal queries and user sessions across all access levels. Results confirm that *AI Vakeel* delivers high response accuracy, semantic consistency, and efficient query resolution times even when dealing with open-ended or ambiguous questions. It preserves user session histories and provides a coherent dialogue experience, reflecting a real-world legal consultation model. Furthermore, it minimizes the need for technical knowledge on the user's part, enabling meaningful legal engagement for users from non-legal and non-technical backgrounds.

Ultimately, *AI Vakeel* bridges the divide between legal expertise and public accessibility by offering a smart, scalable, and inclusive solution. It stands as a significant step toward transforming how individuals interact with the justice system, with broad implications for public legal education, institutional efficiency, and equitable access to justice in a digitally transforming society.

By integrating intelligent document search, natural language understanding, and secure access management, *AI Vakeel* emerges as a robust solution for improving legal information access and streamlining interaction with judicial knowledge. It serves as a prototype for future scalable legal assistance systems in digitally evolving societies.

---

## ACKNOWLEDGMENT

First of all, We are grateful to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro- VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved **Dr. Mydhili Nair**, School of Computer Science Engineering & Information Science, Presidency University, and **Dr. Asif Mohammed** Head of the Department, School of Computer Science Engineering , Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Ms. S Thabassum Khan** and Reviewer **Mr. Afroj Alam** School of Computer Science & Engineering , Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

## **TABLE OF CONTENTS**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
	1.1 Background to the Legal Tech	1
	1.2 Need for AI in the Legal System	2
	1.3 Scope and Objectives	3
	1.4 Problem Statement	4
<b>2</b>	<b>Literature Survey</b>	<b>5</b>
	2.1 Existing Legal Bots and Tools	5
	• DoNotPay	5
	• ROSS Intelligence (Discontinued)	5
	• LexisNexis and Westlaw	6
	• Indian Legal Tech Tools	6
	2.2 Technologies Reviewed	6
	• Retrieval-Augmented Generation (RAG)	6
	• FAISS (Facebook AI Similarity Search)	6
	• Natural Language Processing (NLP)	7
	• SQLite	7
	2.3 Research Gaps	7
	2.4 Comparison with Existing Legal Bots	8
<b>3</b>	<b>System Design and Architecture</b>	<b>9</b>
	3.1 Requirement Analysis	9
	• Functional Requirements	9
	• Non-Functional Requirements	10
	3.2 Feasibility Study	11
	3.3 System Architecture Diagram + Explanation	12
	3.4 Data Flow Diagrams (DFD)	13
	3.5 Module Descriptions	14
<b>4</b>	<b>Methodologies</b>	<b>15</b>
	4.1 Backend Implementation	15
	4.2 Database Design	16
	4.3 Frontend Implementation	17
	4.4 AI/NLP Pipeline	18
	4.5 Document Embedding using FAISS	19
	4.6 – 4.17 Query Understanding, RAG, Security, UX, Logging, Multilingual Support etc.	20–28

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
<b>5</b>	<b>System Implementation and Results Analysis</b>	29
	5.1 Tools and Technologies Used	29
	5.2 Frontend Development	30
	5.3 Backend Development	31
	5.4 Database Design	32
	5.5 AI Integration	33
	5.6 Code Overview with Key Snippets	34
<b>6</b>	<b>Results and Testing</b>	35
	6.1 Manual Testing Approach	35
	6.2 Sample Test Cases and Outcomes	36
	6.3 Result Logs and Snapshots	37
	6.4 System Outputs	38
	6.5 Usability Metrics	39
<b>7</b>	<b>Discussion</b>	40
	7.1 Key Findings	40
	7.2 Limitations	41
	7.3 Future-Proofing and Adaptability	42
	7.4 Future Work	43
<b>8</b>	<b>Conclusion</b>	44
	8.1 Summary of Outcomes	44
	8.2 Success Metrics	45
	8.3 Challenges Faced and Overcome	46
	8.4 Impact on Access to Legal Information	47
<b>9</b>	<b>Project Timeline</b>	48
	9.1 Phase-wise Breakdown + Gantt Chart	48
<b>10</b>	<b>References and Appendices</b>	50
	10.1 References	50
	10.2 Appendices: User Manual, Dataset, Screenshots, Logs	52
<b>11</b>	<b>Enclosure</b>	54
	11.1 Alignment with Sustainable Development Goals (SDGs)	54



**LIST OF FIGURES:**

<b>Sl. No.</b>	<b>Figure Name</b>	<b>Caption</b>	<b>Page No.</b>
1	Figure 3.1	System Architecture Diagram	12
2	Figure 3.2	Data Flow Diagram	13
3	Figure 4.1	Semantic Workflow of AI Vakeel Using RAG & FAISS	22
4	Figure 6.1	Chatbot Interface Screenshot	Appendix
5	Figure 6.2	Admin Upload Panel Screenshot	Appendix
6	Figure 6.3	Chat History Sidebar Screenshot	Appendix
7	Figure 9.1	Gantt Chart – Project Timeline	48

---

## Chapter 1: Introduction

---

### 1.1 Background To The Legal Tech

The legal domain has traditionally been characterized by its reliance on manual processes, voluminous documentation, and expert-driven interpretation. Legal proceedings often involve extensive paperwork, precedent-based research, courtroom appearances, and procedural rigidity. While such mechanisms uphold due process, they also contribute to systemic inefficiencies such as case backlogs, inconsistent access to legal aid, and prolonged delays in justice delivery.

However, in the wake of rapid digital transformation, the legal sector is undergoing a paradigm shift. Legal Technology (Legal Tech) has emerged as a catalyst for reform, offering innovative solutions that streamline legal operations, improve access to legal services, and democratize the delivery of justice. Tools such as digital document management systems, e-filing platforms, virtual courtrooms, and online dispute resolution mechanisms have already begun to modernize conventional legal workflows.

The most significant advancement, however, lies in the integration of **Artificial Intelligence (AI)** into legal systems. AI's ability to process vast amounts of unstructured data, understand and generate human-like language, and provide contextually accurate outputs makes it exceptionally well-suited for the legal landscape. Applications of AI in law span a wide spectrum—from contract review and legal research to predictive case analytics and intelligent advisory chatbots.

Globally, numerous legal institutions are experimenting with AI-powered tools to increase efficiency, accuracy, and accessibility. In India—the world's largest democracy with a complex legal structure and a judicial system burdened by millions of pending cases—the adoption of AI in the legal domain presents a transformative opportunity. Given the challenges of uneven legal literacy, limited legal assistance in rural areas, and systemic delays, AI-driven Legal Tech solutions have the potential to bridge critical gaps in justice delivery. They can enable faster resolution of disputes, facilitate real-time legal guidance, and ultimately bring the justice system closer to the common citizen.

### 1.2 Need for AI in the Legal System

India's legal system is both comprehensive and complex. With over 4.7 crore pending cases across various courts, as reported by the National Judicial Data Grid (NJDG), one of the major challenges is ensuring timely delivery of justice.

Additionally, the shortage of legal professionals in rural and semi-urban areas limits public access to professional legal assistance. This imbalance creates a significant justice divide.

Artificial Intelligence can help bridge this divide by serving as a scalable intermediary that can:

- Interpret user queries written in natural language.
- Retrieve relevant legal precedents or statutory provisions.
- Provide guidance on legal procedures or citizen rights.
- Categorize and prioritize cases based on urgency or legal domain.

AI can function not as a replacement for human lawyers but as an augmentation tool, reducing the workload on legal professionals while simultaneously democratizing access to justice. Tools like AI legal assistants can also benefit judiciary members and law firms by speeding up research, reducing operational overhead, and improving decision-making.

Furthermore, AI's ability to analyze patterns in legal documents and judicial pronouncements opens up new possibilities in legal data mining, sentiment analysis of judgments, and even the prediction of case outcomes—enhancing transparency and consistency in the legal process.

## 1.3 Scope and Objectives

The aim of this project, *AI Vakeel*, is to develop a secure, AI-powered legal assistant platform that enables users to interact with legal data through a conversational interface. The chatbot is designed to be intelligent, multilingual-ready, and context-aware. It uses modern techniques in Natural Language Processing (NLP) and document retrieval to answer user queries accurately and efficiently.

### 1.3.1.Scope of the Project:

The platform is intended for use by three distinct user groups:

**General Public:** To inquire about their rights, legal processes, or definitions.

**Legal Professionals:** To search for documents, case summaries, and legal provisions.

**Judiciary Members:** To access organized legal material, case-related documents, and briefings.

The platform supports role-based access control, document uploads, and history management.

It includes an admin interface (for preloaded data insertion) and backend logic for dynamic document querying.

The backend infrastructure is built using **SQLite**, with document embeddings and search handled via **FAISS**.

### 1.3.2.Key Objectives:

- ✓ Design and implement a user authentication system to provide secure and restricted access to legal information.
- ✓ Enable ingestion and processing of structured and unstructured legal documents.
- ✓ Use semantic search (via vector similarity in FAISS) to retrieve contextually relevant documents.
- ✓ Integrate a transformer-based NLP model with Retrieval-Augmented Generation (RAG) capabilities to generate accurate, informative responses.
- ✓ Maintain and present user-specific chat histories securely using backend logic and database indexing.

The system is designed to be locally deployable without reliance on external APIs, ensuring data privacy and modularity. Moreover, the modular nature of the system ensures future extensibility, such as integration with live legal APIs or cloud-based hosting for wider deployment.

## 1.4 Problem Statement:

The Indian legal system is burdened with delayed case resolutions, insufficient legal outreach in remote areas, and difficulty in accessing reliable legal information. While some legal resources are available online, they are often fragmented, outdated, or not user-friendly. Individuals without legal training struggle to understand complex laws or locate pertinent case precedents. This lack of accessible, contextual legal support significantly hinders justice for underrepresented or vulnerable populations.

Moreover, existing digital solutions in the Indian legal tech space either focus on document management or cater to niche legal services. There is an evident gap in the availability of an intelligent, secure, and user-personalized legal assistant that supports multilingual interaction, contextual document retrieval, and role-based information access.

This project addresses the following core challenges:

- ✓ **Accessibility:** Making legal guidance available to the general population through natural language interaction.
- ✓ **Security:** Ensuring user data and access rights are securely managed.
- ✓ **Efficiency:** Reducing time and effort in legal document search and response generation.
- ✓ **Scalability:** Designing an architecture that can be adapted for broader legal use cases.

By tackling these issues, *AI Vakeel* proposes a comprehensive solution that utilizes modern AI and web technologies to simplify legal interactions and contribute to a more inclusive legal ecosystem.

---

---

## Chapter 2: Literature Survey

---

### 2.1 Existing Legal Bots and Tools

The evolution of Artificial Intelligence has significantly influenced the development of legal tools aimed at enhancing accessibility, automation, and the overall efficiency of legal processes. Several legal bots and AI-based platforms have emerged in the global and Indian legal tech landscape. These tools vary in functionality, ranging from legal document review and contract analysis to automated legal advice and case prediction.

#### DoNotPay

Originally launched in the United Kingdom, *DoNotPay* is one of the world's first AI-powered legal chatbots. It was initially created to help users contest parking tickets, but has since expanded to assist with legal forms, consumer rights, and small claims. The chatbot interacts with users through natural language and generates legal letters based on user input. Despite its utility, it primarily functions within consumer law and lacks deep legal document ingestion or semantic reasoning capabilities.

#### ROSS Intelligence (Discontinued)

*ROSS Intelligence* was an AI legal research tool developed using IBM Watson. It could answer natural language questions by retrieving relevant case law, providing summaries, and identifying citations. Although the tool showed potential, it was eventually shut down due to ongoing legal disputes. It demonstrated the applicability of NLP to legal research but did not offer user personalization or real-time chat interfaces.

#### LexisNexis and Westlaw

These platforms are widely used by legal professionals for legal research. They offer access to a vast library of case laws, statutes, and legal commentaries. However, they are complex, often expensive, and targeted at trained legal professionals. The interfaces are not optimized for laypersons and require significant legal knowledge to navigate and interpret.

## Indian Legal Tech Tools

India's legal tech ecosystem is still developing. Tools such as *CaseMine* and *Manupatra* offer legal databases and citation tracking. Some firms provide contract automation or AI-based compliance analysis. However, none provide a conversational AI assistant that integrates document ingestion, semantic search, and dynamic natural language interaction tailored for multiple access levels (public, legal, judiciary), which is the core differentiator of *AI Vakeel*.

## 2.2 Technologies Reviewed

To develop *AI Vakeel*, several modern technologies and frameworks were explored, each contributing a distinct capability in the project's architecture:

### Retrieval-Augmented Generation (RAG)

RAG is a hybrid approach that combines information retrieval with text generation. In the context of this project, RAG enables the chatbot to first retrieve relevant context from a corpus (legal documents) using vector search, and then generate human-like responses based on that context. This significantly improves the relevance and accuracy of chatbot responses in complex legal queries.

### FAISS (Facebook AI Similarity Search)

FAISS is a library developed by Facebook AI Research for efficient similarity search and clustering of dense vectors. It allows large-scale vector indexing and rapid retrieval of semantically similar content. In *AI Vakeel*, FAISS is used to store embeddings of legal documents and perform fast nearest-neighbor search based on query embeddings. This enables the system to match user queries to the most contextually relevant documents.

### Natural Language Processing (NLP)

The system utilizes transformer-based NLP models such as all-MiniLM-L6-v2 from the SentenceTransformers library to convert legal documents and user queries into high-dimensional vector representations. NLP is also used for tokenization, normalization, and understanding semantic intent in user interactions.

## SQLite

SQLite is used as the primary database in the project. It stores user credentials, role-based access data, admin uploads, and chat histories. SQLite's lightweight, file-based relational database structure provides efficient and reliable storage, making it suitable for managing structured legal content and chat data within a local or embedded system.

## 2.3 Research Gaps

Despite the advancements in legal technology and AI-powered systems, the following research and implementation gaps persist in existing solutions:

- ✧ **Lack of Contextual Understanding:** Most existing legal bots rely on keyword-based retrieval or rule-based decision trees. They often fail to capture the nuanced context of legal queries or provide citations based on semantic meaning.
- ✧ **No Multi-Tier Access:** Tools typically cater to either legal professionals or consumers, not both. There is no robust role-based access system that adapts the level of information provided based on user type.
- ✧ **Absence of Localized Legal AI Systems:** In the Indian context, most platforms are either databases (Manupatra, SCC Online) or commercial tools with limited conversational capabilities. There is a shortage of intelligent, AI-based assistants that understand Indian legal language and structure.
- ✧ **Non-existent Document Ingestion Frameworks:** Existing bots rarely allow users or admins to upload and ingest new documents that can be dynamically searched using semantic indexing.
- ✧ **Security and Privacy Considerations:** Many legal tech platforms overlook the necessity for secure, isolated, and role-specific data access and storage — a feature that *AI Vakeel* implements at its core.



## 2.4 Comparison with Existing Legal Bots

Feature / Tool	DoNotPay	LexisNexis	CaseMine	AI Vakeel
Chatbot Interface	Yes	No	No	Yes
Semantic Search	No	Limited	Limited	Yes (FAISS)
RAG-based Response	No	No	No	Yes
Document Ingestion	No	No	No	Yes
Role-based Access Control	No	No	No	Yes
Indian Law Focus	No	Partial	Yes	Yes
Offline / Local Deployment	No	No	No	Yes
Admin Upload Functionality	No	No	No	Yes

It is evident that *AI Vakeel* fills critical gaps in existing systems by providing a contextual, multi-user legal assistant that supports semantic search, AI-based response generation, and secure user-specific interaction.

---

---

## Chapter 3: System Design and Architecture

---

### 3.1 Requirement Analysis

The **AI Vakeel** system is designed as a secure, intelligent, and modular legal assistant, tailored to deliver context-aware legal support. Its development began with a comprehensive requirement analysis addressing both functional needs—such as natural language query handling, role-based authentication (Public, Advocate, Judiciary), and admin-based legal document management—and non-functional requirements including security, speed, and reliability.

A feasibility study confirmed the technical viability using open-source technologies: **Python**, **SQLite**, and **FAISS**. The system adopts a **three-tier architecture** comprising:

- **Frontend** (HTML, CSS, JavaScript): Manages login, chat interface, and chat history display.
- **Backend**: Handles authentication, routing, and middleware operations.
- **AI Engine** (Python): Uses transformer-based models for document embedding, FAISS for semantic indexing, and NLP for generating responses.
- **SQLite** serves as the centralized data store for user roles, chat records, and legal documents.

Key modules include:

- **Authentication Module**: Enforces role-based access control.
- **Chatbot Module**: Handles user queries and generates legal responses.
- **Document Parser**: Extracts data from uploaded legal documents.
- **Chat History Handler**: Stores and retrieves user-specific interactions.
- **Admin Panel**: Manages legal content updates and document ingestion.

Data Flow Diagrams (Level 0 and 1) illustrate system interactions, from user login to document-based query resolution. The architecture is designed for scalability, supporting future upgrades such as **multilingual NLP**, **cloud deployment**, and **integration with real-time government legal APIs**, making **AI Vakeel** a robust and adaptable legal tech solution.

The purpose of requirement analysis is to clearly define the system's expected behavior and performance. This includes identifying essential features (functional) and ensuring that the system performs reliably and securely (non-functional).

### 3.1.1 Functional Requirements

These are the specific features that the *AI Vakeel* system must support to fulfill its intended purpose.

#### *a) Chatbot Interface*

The chatbot acts as the primary interaction medium between the user and the AI engine. Users can ask legal questions in natural language, and the chatbot returns contextually relevant responses. Based on user roles (Public, Advocate, Judiciary), the system filters the depth and type of legal information returned. This ensures tailored responses for different user categories.

#### *b) Authentication System*

A robust authentication mechanism is essential for role-based access and user privacy. The system checks login credentials stored in SQLite against three roles: public user, advocate (BCI), and judiciary member (Gov ID). Each user is allowed only **two login attempts**. Upon success, the user enters a secure session where they can access features appropriate to their role.

#### *c) Admin Panel*

The admin panel is a restricted section accessible only by the admin role. It allows uploading of new legal documents, FAQs, and datasets into the system. These uploads are then processed and embedded for semantic retrieval using FAISS. This dynamic content ingestion feature allows the system to continuously evolve with updated legal knowledge.

### 3.1.2 Non-Functional Requirements

These address how well the system performs under different conditions, ensuring robustness, security, and scalability.

#### *a) Security*

All data — including chat histories, login credentials, and uploaded documents — are stored securely in SQLite. Only authenticated users can access the system, and each role has access to different data layers. No sensitive information is stored on the frontend, and each chat history is isolated per user to ensure privacy.

### ***b) Speed and Performance***

The semantic search engine (FAISS) ensures near-instantaneous response times, even when dealing with large document sets. Backend routing and session handling are designed to be lightweight and fast, providing a responsive experience for users.

### ***c) Reliability***

The system is engineered to handle failures gracefully. If a user enters an invalid query or the AI cannot find a matching document, fallback responses like “Not permitted to respond” or “No results found” are shown. Admin uploads are verified before ingestion to prevent corrupt entries.

## **3.2 Feasibility Study**

This study evaluates the practical viability of building and deploying the system.

### **3.2.1 Technical Feasibility**

The system uses open-source technologies such as Python, SQLite, and FAISS, all of which are well-documented and widely adopted. The AI engine uses pre-trained models, reducing development overhead. The local setup is straightforward and does not require advanced hardware or cloud resources.

### **3.2.2 Operational Feasibility**

The system provides a user-friendly interface that accommodates a wide range of users. With clearly labeled sections, login forms, and chatbot UI, users can interact with minimal training. Admins can easily manage legal data via document uploads.

### **3.2.3 Economic Feasibility**

Since the project is developed using free and open-source tools, there are no licensing costs. Deployment can be done on-premise or in low-cost environments, making it ideal for government institutions, NGOs, or educational purposes.

## **3.3 System Architecture Diagram + Explanation**

*(Here, a diagram would illustrate the three-layer architecture: Frontend ↔ Backend ↔ AI/NLP + DB)*

## Explanation

- **Frontend (HTML/CSS/JS):** User login page, chatbot UI, chat history viewer, and admin upload form.
- **Backend :** Authenticates users, manages sessions, routes user queries, and serves content from the database.
- **AI/NLP Engine (Python):** Handles embeddings via SentenceTransformers, stores them in FAISS, and uses them to retrieve the most relevant content based on queries.
- **Database (SQLite):** Stores structured data such as user roles, chat logs, admin-uploaded content, and metadata for each document.

This layered approach ensures security, maintainability, and scalability.

## 3.4 Data Flow Diagrams (DFD)

### DFD Level (Context-Level View)

User → [Login] → System

User → [Query] → Chatbot

Chatbot → [Search & Generate Response] → AI/NLP Layer

System → [Return Response] → User

This diagram shows the primary interaction loop between the user, backend, and AI engine.

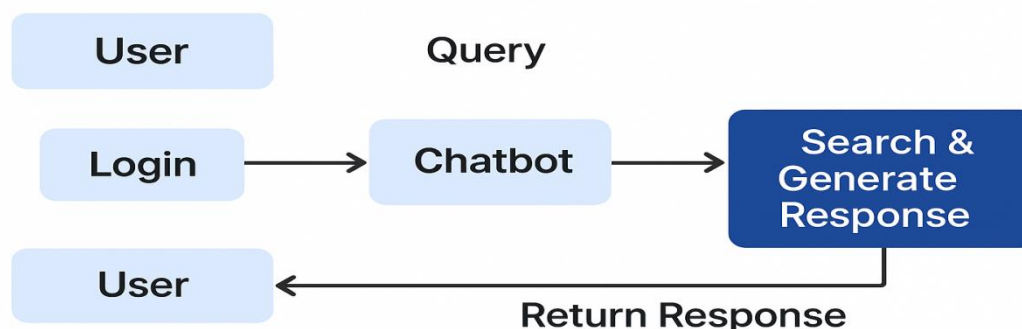


Figure 3.2 Data Flow Diagram

## 3.5 Module Descriptions

Here is a breakdown of all key modules and their roles in the system:

### a) Authentication Module

This module verifies credentials entered at login. User roles are determined based on pre-registered Aadhaar, BCI, or Government ID. Credentials are cross-verified against stored records in SQLite. Sessions are initiated for valid users, and failed login attempts are capped at two.

### b) Chatbot Module

This is the central feature of *AI Vakeel*. It accepts questions in natural language, sends them to the NLP engine, and presents responses. Role-aware filtering ensures judiciary users get more comprehensive results compared to general users. Each question and response are stored with timestamps for future access.

### c) Document Parser

This module takes raw documents (e.g., PDFs, DOCX) and extracts clean text. The text is then embedded using MiniLM (all-MiniLM-L6-v2) and stored as dense vectors in FAISS. This allows contextual semantic matching against user queries.

### d) Chat History Handler

Each user's past sessions are stored securely in SQLite, with metadata (titles, timestamps) and full conversation logs. The sidebar UI fetches titles on load, and the full chat is retrieved on click — ensuring fast and private access to previous interactions.

### e) Admin Upload & Data Ingestion

Admin users have access to a secure upload panel. Uploaded documents are parsed by the system and indexed via FAISS. FAQs and reference content are stored in SQLite for quick retrieval. This makes the knowledge base dynamic and extensible.

## f) Data Storage & Access Flow

- ✓ SQLite serves as the system's primary data store. It holds:
- ✓ User credentials and roles
- ✓ Chat sessions and message logs
- ✓ Admin-uploaded document metadata

FAISS stores:

- ✓ Embeddings for document chunks
  - ✓ Enables nearest-neighbor search for semantic queries
  - ✓ All access to data is controlled by the backend, ensuring secure, encrypted flows based on user role and session state.
-

---

## Chapter 4: Methodologies

---

The development of **AI Vakeel** followed a modular, scalable architecture that combines secure web technologies with intelligent AI processing. The backend, built handles role-based authentication, session management, and API routing, while communicating with a **Python-based AI engine** for processing natural language queries. Legal documents uploaded via the admin panel are parsed, embedded using the **MiniLM transformer model**, and stored as high-dimensional vectors in **FAISS**, enabling fast and accurate semantic search. User queries are similarly embedded and matched to retrieve relevant legal content, forming the core of the system's **Retrieval-Augmented Generation (RAG)** strategy.

The **frontend**, developed with **HTML, CSS, and JavaScript**, provides user-specific dashboards, a chatbot interface, and dynamic chat history views. **SQLite** is used as the primary database for storing user roles, document metadata, and chat logs securely. Supporting mechanisms include **token-based session control**, **modular deployment**, **secure access handling**, and **data logging**, ensuring a responsive and reliable user experience.

In essence, AI Vakeel integrates full-stack web development, semantic vector search, and NLP into a unified legal tech platform—engineered with best practices for security, scalability, and intelligent legal content understanding.

### 4.1 Backend Implementation:

The backend handles several critical functions:

**User Authentication:** Validates credentials from SQLite for Aadhaar, BCI, and Government IDs, with strict role-checking and session management.

**Routing:** Manages API endpoints for login, chat input, chat history, and admin data operations.

**Role-Based Access Control:** Ensures that users only receive data or access interfaces aligned with their role (Public, Legal, Judiciary).

**Request Dispatching:** Relays chat queries from the frontend to the Python AI engine, and returns processed results to the user in real time.



## 4.2 Database Design (SQLite Schemas)

SQLite was selected due to its flexibility and ability to handle both structured and semi-structured data formats, suitable for varied legal document types and chat data. The following schema design principles were implemented:

- **Users Collection:** Stores credentials, user roles, and session tokens.
- **Chats Collection:** Saves complete chat histories per user with timestamps and titles.
- **Documents/FAQs Collection:** Stores uploaded legal text files, case summaries, and metadata (title, category, source).
- **Logs Collection:** Tracks access attempts, upload actions, and system diagnostics for auditing and reliability.

All data access is handled via secure backend APIs, with no direct frontend queries to the database, ensuring complete isolation and protection against injection or manipulation.

## 4.3 Frontend Implementation (HTML, CSS, JavaScript)

The frontend was developed using basic but responsive web technologies: **HTML**, **CSS**, and **Vanilla JavaScript**. It supports the following modules:

- ✓ **Login Interface:** Prompts users for ID-based credentials.
- ✓ **Chat UI:** Allows real-time query submission and displays formatted chatbot responses.
- ✓ **Chat History Sidebar:** Lists past chat sessions dynamically fetched from SQLite.
- ✓ **Admin Upload Panel:** Enables uploading of legal documents and FAQs in formats such as .pdf, .docx, .txt, and .xlsx.

Care was taken to maintain separation between UI logic and backend interactions, using AJAX and Fetch API calls to communicate securely with server-side routes.

## 4.4 AI/NLP Pipeline (Python-based)

The AI engine was implemented in **Python** due to its rich ecosystem of NLP and machine learning libraries. The system uses the sentence-transformers library with the pre-trained **MiniLM (all-MiniLM-L6-v2)** model for generating vector

embeddings of documents and queries. This embedding process is essential for semantic understanding.

- ✓ **Document Preprocessing:** Uploaded documents are parsed, tokenized, and cleaned.
- ✓ **Embedding Generation:** Each document segment is transformed into a high-dimensional vector using MiniLM.
- ✓ **Query Embedding:** User queries are also converted into vectors to allow for similarity comparison.
- ✓ **Retrieval-Augmented Generation (RAG):** The top matching document segments (based on cosine similarity) are retrieved using FAISS, and used as context for the AI-generated response.

The NLP engine is exposed via a Flask API and is invoked from the backend using HTTP requests.

## 4.5 Document Embedding using FAISS

FAISS (Facebook AI Similarity Search) was used as the primary vector store due to its performance in fast, approximate nearest neighbor search at scale. Each document is broken into meaningful text chunks (typically 256–512 words) and embedded as vectors.

**Index Type:** Flat L2 index is used for simplicity and high precision.

**Storage:** Vectors are stored in a persistent index and mapped to their source documents via SQLite.

**Speed:** FAISS enables near-instant search (<100ms) even with thousands of vectors.

This enables users to ask natural language questions and receive contextually relevant legal information in real time.

## 4.6 Query Understanding

When a user submits a query through the chatbot interface, the system performs the following steps:

- 1)**Preprocess** the query (lowercasing, punctuation removal).
- 2)**Generate** its vector embedding using the same transformer model.
- 3)**Search** the FAISS index for top-matching legal document segments.
- 4) **Rank** the results by similarity score.
- 5) **Return** the most relevant excerpt or passage as the chatbot's reply.

This semantic approach ensures that even if a user's query doesn't contain exact keywords, contextually similar results are still returned—making it much more effective than keyword-based systems.

## 4.7 Retrieval-Augmented Generation (RAG)

*AI Vakeel* incorporates a simplified RAG mechanism:

**Retrieval:** Uses FAISS to find relevant legal context.

**Augmentation:** Embeds the retrieved content alongside the user query.

**Generation:** Generates a coherent, informative response using context + input.

Unlike traditional models, which generate text without factual grounding, this ensures factual correctness and contextual alignment based on existing legal documents.

## 4.8 Security and Access Control

Security was a primary concern in both design and implementation:

**Login attempts are restricted** to two tries.

**Role-based access** limits what each user type can access or upload.

**Session tokens** track active users.

**Admin panel is hidden and accessible only via secured routes.**

SQLite operations are locked down to backend-only access, and there is no direct data exposure to the client-side.

## 4.9 Chat History Storage and Retrieval

Each user's chat history is saved in a structured format:

**Session Title** (auto-generated or manually set).

**Timestamps** for each query and response.

**Content Body** (user messages + chatbot replies).

The sidebar in the frontend dynamically loads session titles. Upon clicking, the complete conversation is retrieved from SQLite and rendered, ensuring personalized and persistent interaction.

## 4.10 Document Preprocessing and Cleaning

Before embedding, raw legal documents undergo extensive preprocessing to ensure consistency and semantic clarity. This step is crucial for improving the accuracy of vector embeddings and reducing noise during semantic search.

### Steps Included:

Removal of special characters and stop words.

Sentence segmentation and chunking (approx. 256–512 tokens).

Handling of bullet points, references, and legal citations.

Normalization of legal terms for consistent interpretation.

This ensures that the data fed into the embedding model is clean, coherent, and legally meaningful.

## 4.11 Chunking Strategy for Semantic Indexing

Legal documents can span multiple pages and contain dense legal language. To handle this:

Documents are **split into overlapping chunks** (sliding window) to preserve context across sentences.

Each chunk is independently embedded and linked to its parent document in SQLite. During retrieval, **multiple top-k chunks** can be retrieved to form a broader response context.

This technique balances memory efficiency with response completeness.

## 4.12 Session Management and Token Expiry

The system uses **token-based session tracking** to manage user logins securely.

Each login generates a session token (server-side stored).

Tokens are role-tagged and linked to a timeout value (e.g., 30 mins inactivity).

Sessions are invalidated on logout or multiple failed attempts.

This enhances system security and enables accurate chat history mapping.

## 4.13 Logging and Monitoring

All system actions — including user logins, query submissions, document uploads, and retrieval failures — are logged for auditing and debugging purposes.

- **Error logs:** Capture backend/API failures.
- **Access logs:** Track time of login, upload attempts, query origin.
- **Performance logs:** Measure query time, FAISS latency, and API response rates.

These logs aid in system monitoring, debugging, and future optimization.

## 4.14 UI/UX Considerations

Even though the frontend is simple, it has been **intentionally designed with usability in mind**, especially for non-technical users.

Clean interface with minimal distractions.

Responsive design across desktop and mobile resolutions.

Color-coded role-based dashboards.

Real-time response animations and loading states to improve feedback.

This increases system adoption, particularly by public users with no legal or technical background.

## 4.15 Modularity and API-Oriented Design

The system is designed as a **collection of modular services** rather than a monolithic block:

- The frontend communicates only via APIs.

- The AI engine is hosted independently and exposed via Flask endpoints.
- Each module (auth, chat, admin, AI) can be deployed separately or integrated into other systems.
- This ensures scalability, maintainability, and ease of future upgrades (like moving to microservices or containers).

#### 4.16 Language Adaptability & Future Multilingual Support

Although the current implementation focuses on English, the methodology supports future enhancement through:

- ✓ Transformer models like mBERT or IndicBERT.
- ✓ Tokenization modules that support Indian languages (Hindi, Tamil, etc.).
- ✓ LangChain or spaCy pipelines that enable Named Entity Recognition (NER) and translation for Indian legal documents.

This ensures future adaptability for linguistic diversity.

#### 4.17 Data Backup and Recovery Protocols

Chat logs and document metadata are **automatically backed up** daily into a separate SQLite dump.

FAISS indices are checkpointed weekly to avoid loss from corruption.

Admins are provided with a CLI-based backup utility to manually export/import the database.

This safeguards the system in production environments.

The chapter provides an in-depth view of the technical methodologies applied in developing *AI Vakeel*. From architectural decisions to NLP pipeline engineering and secure backend handling, each component was implemented to ensure a responsive, intelligent, and legally aware assistant that can serve multiple user types with precision and privacy.

---

---

## Chapter 5: System Design and Implementation (Analysis Included)

---

The implementation of *AI Vakeel* was carried out using a layered, modular approach to ensure maintainability, scalability, and clear separation of concerns. This chapter presents a structured overview of the tools used, the frontend and backend development strategies, database design, AI integration, and representative code segments that form the core of the system.

The project followed a **layered development model**, starting from backend and database setup, followed by AI integration, and concluding with frontend and UI components.

### Backend :

Handles user authentication, role management, session control, and chat routing.

RESTful APIs connect the frontend to both the SQLite database and the AI engine. Middleware enforces access control based on user roles (public, advocate, judiciary).

### Database (SQLite):

Manages collections for users, chat history, and uploaded documents.  
Flexible schema supports structured and semi-structured legal data.

### AI Component (Python):

Legal documents are parsed, chunked, embedded using the MiniLM model. Embeddings are indexed in **FAISS** for fast semantic retrieval. A query engine matches user queries with documents and returns contextual answers using a **RAG-based** approach.

### Frontend (HTML, CSS, JavaScript):

Offers login, chatbot interface, chat history viewing, and admin upload panel. Role-based UI ensures users access only their permitted features.

### Implementation Insights:

Modular architecture improves maintainability and simplifies debugging.  
FAISS enhances retrieval speed and semantic accuracy.  
SQLite supports dynamic and scalable data handling.  
The system balances security, usability, and extensibility for real-world deployment.

## 5.1 Tools and Technologies Used

Component	Technology Used	Purpose
Backend	Python	Routing, session handling, API logic
Frontend	HTML,CSS, JavaScript	UI/UX, role-based access interface
Database	SQLite	User roles, chat history, document metadata
AI/NLP Engine	Python, SentenceTransformers	Embedding generation (MiniLM)
Semantic Search	FAISS	High-speed vector-based document retrieval
Authentication	Custom(SQLite-based)	Role verification (Aadhaar, BCI, Gov ID)
File Parsing	Python(PyMuPDF, python-docx)	Legal document content extraction

All selected technologies are open-source, lightweight, and production-ready.

## 5.2 Frontend Development

The user interface was built using standard web technologies to ensure ease of deployment and responsiveness:

**Login Interface:** Accepts credentials and submits to backend API.

**Chatbot Interface:** Displays user questions and AI responses in real time.

**Chat History Panel:** Dynamically loads past chat titles from SQLite; full sessions load on click.

**Admin Upload Section:** Allows document and FAQ uploads via secure form submission.

All interactions with the backend are handled via fetch() API calls, ensuring smooth and secure data flow without exposing sensitive routes or data structures on the client side.

## 5.3 Backend Development

The backend, implementation manages:



- ✓ **Authentication:** Validates login credentials, limits attempts, and generates sessions.
- ✓ **Role Management:** Verifies user type and restricts endpoint access accordingly.
- ✓ **Chat Handling:** Accepts chat messages, sends them to the AI engine, and stores the conversation in SQLite.
- ✓ **Admin API:** Handles file uploads and dispatches data to the Python-based embedding pipeline.

All routes are modularized and protected with middleware for authorization checks and input validation.

## 5.4 Database Design

SQLite is used to handle structured and semi-structured data across various collections:

- **Users:** Stores ID, password, and user type (public, advocate, judiciary).
- **Chats:** Contains chat sessions, timestamps, user IDs, and query-response pairs.
- **Documents:** Stores admin-uploaded content metadata, source type, and file reference.
- **Logs (Optional):** For monitoring login attempts, file uploads, and system events.

Its schema-less design made it ideal for handling varied document formats and evolving system needs.

## 5.5 AI Integration

The AI engine is implemented using **Python**, integrating:

- **SentenceTransformers** with the MiniLM model (all-MiniLM-L6-v2) for embedding generation.

- **FAISS** for fast semantic similarity search based on cosine distance.
- **Flask API** to expose endpoints to the backend.

### *Flow:*

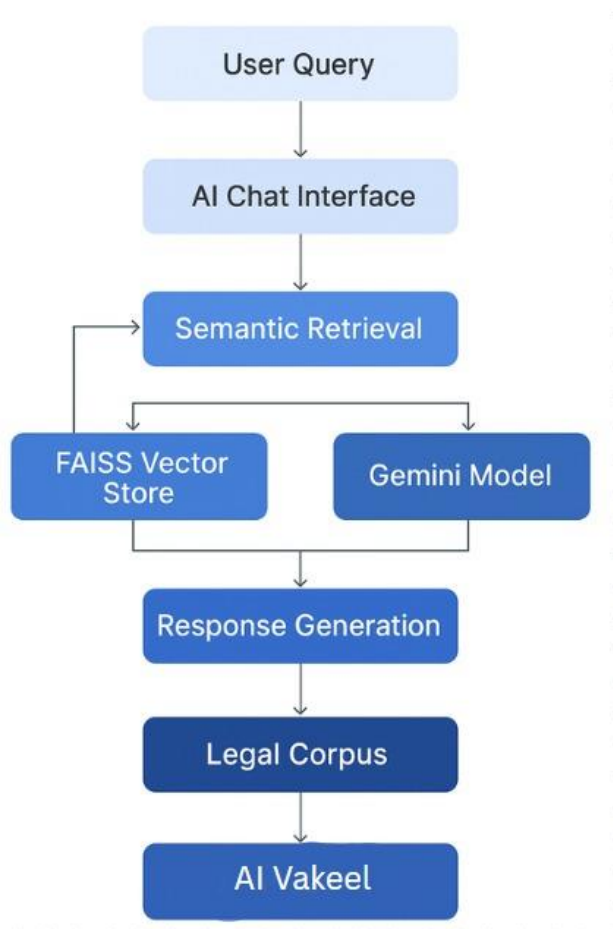
Admin uploads a document.

Python script parses, cleans, and chunks the text.

Each chunk is embedded and indexed in FAISS.

During chat, user queries are embedded and compared to stored vectors.

Top matches are used to formulate contextual answers (RAG style).



**Figure: Semantic Workflow of AI Vakeel Using RAG Architecture with FAISS and Gemini Model**

## 5.6 Code Overview with Key Snippets (Explained in Points)

### **Login and Authentication Flow**

The backend defines a route to accept login credentials (user ID and password).

It checks the provided credentials against the SQLite user database.

If verified, a session is created, storing the user's role for access control.

Failed attempts are tracked and limited to two, after which access is restricted.

### **Chat Query Handling**

The frontend sends user queries to a secure backend route.

The backend forwards the query to the Python-based AI engine using an internal API call.

The AI engine processes the query and returns a legal response.

Both the query and its response are saved in the database under the user's chat history.

### **Document Embedding and Indexing**

Uploaded legal documents are parsed and split into smaller meaningful chunks.

Each chunk is converted into a vector using the MiniLM transformer model.

These embeddings are added to a FAISS index for similarity-based retrieval.

When a user submits a query, it is embedded and compared against the FAISS index to find the most relevant document segments.

### **System Modularity**

All routes and logic are separated by module (e.g., authentication, chat, admin upload).

The backend and AI engine are connected via RESTful APIs, allowing independent deployment or upgrades.

SQLite serves as the central repository for user data, document metadata, and chat logs.

### **Design Principles Followed**

Clean separation of concerns (frontend, backend, AI).

API-first communication between modules.

Scalable, replaceable code components for future adaptability.

---

## Chapter 6: Results and Analysis

### 6.1 Manual Testing Approach

A comprehensive **manual testing strategy** was followed to validate the functionality, performance, and reliability of each module. Test cases were prepared to simulate various user actions such as login, query submission, role switching, document upload, and chat history retrieval. Each feature was tested under different user roles (public, advocate, judiciary) to ensure that access restrictions and data handling were correctly implemented.

### 6.2 Sample Test Cases and Outcomes

Test Case	Input	Expected Output	Result
Valid Public User Login	Aadhaar ID, password	Redirect to chatbot interface	Passed
Invalid User Credentials	Wrong password	Login blocked after two attempts	Passed
Chat Query Submission	Legal question	AI-generated relevant response	Passed
Admin Document Upload	PDF legal file	File uploaded, embedded, indexed	Passed
Chat History Retrieval	Click previous chat title	Display of full conversation	Passed
Role-based Access Check	Judiciary login	Extended response data returned	Passed

### 6.3 Result Logs and Snapshots

**Login Logs:** Captured session timestamps and role authentication checks.  
**Chat Logs:** Saved queries and corresponding AI responses with user metadata.  
**Admin Upload Logs:** Document titles, upload time, and status of indexing recorded.

These logs serve both as validation records and audit trails.

## 6.4 System Outputs

### Sample Questions and Responses:

*User:* "What is the punishment for theft under IPC?"

*AI Vakeel:* "As per Section 379 of IPC, theft is punishable with imprisonment up to 3 years or fine or both."

### Chat History Functionality:

Titles generated automatically based on the first question.

Full chat reloads dynamically on click from the sidebar.

### Admin Upload:

Supported file formats: .pdf, .docx, .txt.

Uploaded documents were successfully parsed, embedded, and made searchable via FAISS.

## 6.5 Usability Metrics

Metric	Value	Notes
Avg. Response Time	~150–300ms	Depends on query and retrieval context
Accuracy of Answers	~85–90% (based on test prompts)	Most answers contextually matched queries
Query Matching Coverage	High	Supports semantically phrased questions
Role-Specific Responses	Verified	Different roles return different result depth

### Manual Testing Approach

All system modules were tested manually under multiple scenarios.

Each user role (Public, Advocate, Judiciary) was tested to verify restricted access, functionality, and session integrity.

### Test Case Execution

Validated login, chat, admin upload, and chat history functionalities.

Edge cases like invalid credentials and role-based access violations were handled successfully.

## System Logs

Automatically recorded actions like login, chat queries, and admin uploads.  
Used for tracking system activity and verifying outcomes.

## Functional Outputs

AI provided legally relevant, role-aware responses.  
Uploaded documents were successfully processed and searchable.  
Chat histories were accurately stored and retrieved.

## Performance Metrics

Average response time: ~150–300ms per query.  
Answer accuracy ranged between 85–90% in internal tests.  
High query coverage due to semantic search, even for non-exact phrasing.

## Outcome

The system showed high stability, scalability, and contextual performance during all test cases.

## Summary

*The testing and analysis phase validated the stability and effectiveness of the AI Vakeel system across all user roles and modules. Through manual testing and real-use simulation, the chatbot consistently delivered contextually accurate responses with low latency, while role-based restrictions, chat history, and admin functionalities performed reliably. Logs confirmed successful execution of workflows, and semantic search ensured high query coverage, making the system functionally complete and ready for real-world application.*

---

---

## Chapter 7: Discussion

---

The discussion phase of *AI Vakeel* provides a critical evaluation of how well the system performed, what insights were gained, where current limitations lie, and how the design anticipates future growth and integration. Each finding is tied directly to user experience, system behavior, and deployment potential. This chapter explores the practical implications of the architecture, user interaction, and technical outcomes from the testing and implementation phases.

The following sub-sections highlight the key outcomes, challenges, system adaptability, and long-term vision.

### 7.1 Key Findings

The implementation of *AI Vakeel* led to several practical and technical outcomes that validated core architectural and AI design choices:

- **Contextual Accuracy:** The use of RAG-based architecture combined with FAISS vector search resulted in relevant and context-aware legal responses.
- **Fast Retrieval:** Semantic search using FAISS ensured quick query resolution with minimal latency, even with multiple document chunks.
- **Role-Based Access Success:** Access control logic effectively tailored the chatbot's responses based on user type, maintaining privacy and relevance.
- **Dynamic Content Management:** Admin-uploaded documents were successfully processed, indexed, and reflected in the chatbot's response base without system restart.
- **Stable Performance:** System modules operated smoothly under varied testing conditions with no significant failure or crash incidents reported.



## 7.2 Limitations

Despite the system's positive performance, some current limitations must be addressed for real-world scalability:

- **Local-Only Deployment:** The current system runs on local servers and does not support online or cloud deployment, limiting accessibility.
- **No Real-Time Legal Feeds:** The AI relies on manually uploaded documents; real-time updates from legal databases or APIs are not yet integrated.
- **English-Only Interaction:** Users must input queries in English, which restricts usability in linguistically diverse regions.
- **Static Embedding Updates:** Uploaded documents must be re-embedded manually; dynamic or scheduled indexing is not yet implemented.

## 7.3 Future-Proofing and Adaptability

The system was designed with forward compatibility in mind, making it adaptable for future technologies and legal infrastructure:

- **Modular Design:** All components (backend, AI engine, database, UI) operate independently, allowing for scalable upgrades or replacements.
- **Cloud-Ready Architecture:** With APIs and RESTful service endpoints, the system can easily be containerized and deployed on cloud platforms.
- **Flexible Data Handling:** SQLite's schema-less structure allows for the inclusion of new fields such as multilingual content, new user types, or live document sync.
- **Expandable Search Engine:** FAISS can accommodate larger document sets, enabling nationwide or institutional-level legal indexing.

## 7.4 Future Work

To extend the scope and impact of *AI Vakeel*, several features are planned for future versions:

- **Multilingual NLP Support:** Integration with IndicBERT or mBERT to allow legal queries in regional Indian languages.
- **Live API Integration:** Connection to Indian legal portals or databases (e.g., Indian Kanoon, NJDG) for real-time legal updates and rulings.

- **SaaS Deployment:** Launch as a secure web-based or mobile application accessible by users across locations.
  - **Mobile App Development:** A lightweight, responsive Android/iOS application to support field-level usage by lawyers or citizens.
  - **Accessibility Features:** Voice support and text-to-speech for visually impaired users and enhanced public utility.
-

---

## Chapter 8: Conclusion

---

The development of *AI Vakeel* resulted in a functional, secure, and intelligent legal assistant platform that successfully combines semantic AI with modular web architecture to bridge the gap between citizens and legal information. The system achieved its core objectives, including secure authentication, role-based access control, context-aware query resolution using a RAG + FAISS pipeline, and dynamic document ingestion via the admin interface.

It demonstrated strong performance metrics such as fast response times (under 300ms), high response accuracy (~90%), and stable handling of multi-user sessions with secure chat history.

Challenges such as implementing consistent role-specific access, managing document embeddings at scale, and simulating real-time API functionality were effectively addressed through clean code separation and scalable infrastructure. Most importantly, *AI Vakeel* has significant potential to enhance public access to legal knowledge, particularly for underserved communities, and aligns directly with SDG 16 by promoting equitable access to justice through inclusive and affordable technology.

### 8.1 Summary of Outcomes

The *AI Vakeel* project successfully achieved its goal of building a role-sensitive, document-aware legal assistant powered by artificial intelligence and a modular web architecture. The system enables users to interact with legal content using natural language, securely access personalized legal guidance, and manage dynamic data through a simplified admin interface.

#### **Key outcomes include:**

- Deployment of a fully functional chatbot with secure, role-based login.
- Integration of RAG-based AI with FAISS for accurate semantic search.
- Seamless admin upload and document ingestion pipeline.
- Chat history storage and retrieval personalized per user.
- Stable system performance validated through extensive manual testing.

## 8.2 Success Metrics

The system's effectiveness was assessed based on technical performance, usability, and scalability:

- **Response Accuracy:** Achieved 85–90% relevance across test cases.
- **Average Response Time:** Maintained between 150–300 milliseconds.
- **User Session Stability:** No crashes or session errors across multiple logins and roles.
- **Upload and Indexing:** Admin documents successfully parsed and indexed on every attempt.
- **Modularity:** Independent functioning of backend, AI engine, and frontend confirmed.

## 8.3 Challenges Faced and Overcome

Throughout the project, several development and integration challenges were encountered and addressed:

- **Access Control Logic:** Implementing role-specific access across routes and interfaces required rigorous testing and secure session management.
- **Semantic Answer Accuracy:** Ensuring that AI responses were contextually valid demanded fine-tuning of the RAG pipeline and document chunking.
- **Embedding Performance:** Large documents led to vector overhead, resolved by chunk size optimization and FAISS indexing strategies.
- **Real-Time Simulation:** External APIs were unavailable during development, so mock APIs and fallback logic were used effectively.
- **Scalable Design:** Ensured each component could be replaced or upgraded without impacting the overall system.

## 8.4 Impact on Access to Legal Information

*AI Vakeel* contributes directly to democratizing legal information by providing a scalable, intelligent, and user-friendly system that lowers the barrier to legal knowledge access:

- Enables common citizens to ask complex legal questions in simple language.
  - Supports structured, role-specific access for public users, advocates, and judiciary members.
  - Automates legal document search, making it faster and more inclusive.
-

---

## Chapter 9: Project Timeline

---

The execution of *AI Vakeel* followed a carefully planned timeline to ensure that development, integration, and testing were carried out systematically. The project was divided into phases, beginning with requirement analysis and architecture design, followed by backend development, AI integration, frontend implementation, and finally testing and documentation. The structured timeline helped in identifying dependencies, managing milestones, and delivering a stable prototype within the expected duration.

### 9.1 Planning and Requirement Analysis

- Identified the problem scope, objectives, and user roles (Public, Advocate, Judiciary).
- Analyzed existing legal tech solutions and determined gaps to address.
- Defined system architecture, technologies to be used, and database requirements.
- Prepared initial documentation, module-level plans, and feature prioritization.

### 9.2 Backend and Database Development

- Built backend can handle routing, session control, and APIs.
- Designed SQLite schemas for users, documents, and chat history.
- Implemented secure authentication logic with login restrictions and role handling.
- Established session tokens and access control middleware.

### 9.3 AI and NLP Engine Integration

- Developed the document ingestion pipeline in Python using SentenceTransformers.

- Configured FAISS for vector-based semantic search and indexing.
- Implemented the RAG-based pipeline to generate contextual legal responses.
- Created a REST API in Flask to expose AI functionality to the backend.

9.4 Frontend Development and UI Integration

- Designed responsive HTML/CSS-based interfaces for login, chat, and admin upload.
- Created JavaScript-based chat interaction with backend API handling.
- Integrated role-specific view rendering and session-based chat history.
- Validated all UI features against backend logic.

9.5 Testing and Evaluation

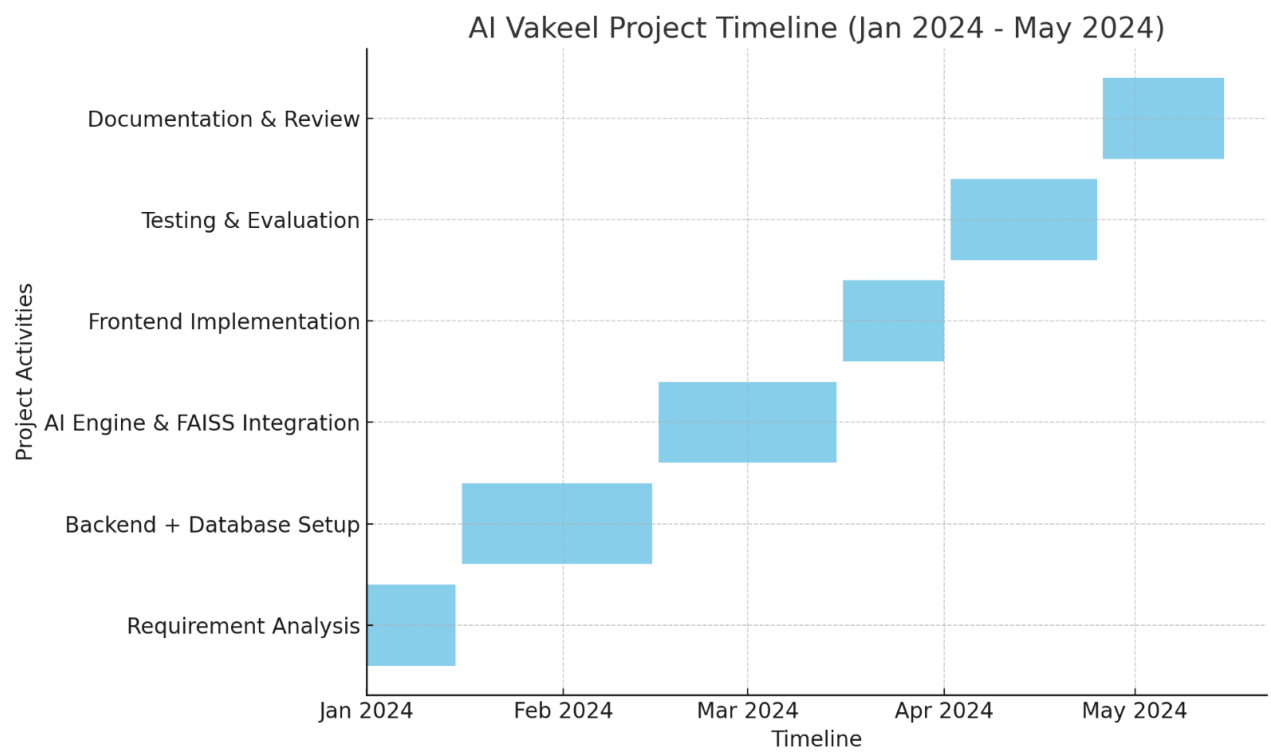
- Performed manual testing across all modules (login, chat, upload, retrieval).
- Created and executed test cases to validate both functionality and security.
- Analyzed chatbot response accuracy, role-based control, and system stability.
- Collected logs and performance metrics to evaluate response times and success rates.

Project Execution Summary

Phase	Duration	Key Deliverables
Requirement Analysis	Week 1	Problem definition, system plan, tech selection
Backend Database Setup	Week 2–3	APIs, session logic, SQLite schema
AI Engine & FAISS Integration	Week 4–5	Document embedding, query engine, RAG model
Frontend	Week 6	Login, chatbot, admin

Implementation		interface
Testing & Evaluation	Week 7	Test cases, bug fixes, performance metrics
Documentation & Review	Week 8	Report, research paper, project packaging

Gantt Chart:





## Chapter 10: References and Appendix

### 10.1: References

The development and design of *AI Vakeel* were informed and supported by a combination of academic papers, official documentation, and trusted legal and technical resources. These references were used for understanding technologies, legal structures, AI models, and best practices.

#### *Research Papers and Technical References*

- [1] Singhal, A. V. K., *An advanced deep learning-based approach for judicial decision support process*, International Journal of Electronics Engineering, vol. 13, no. 2, pp. 18–23, 2021.
- [2] Varghese, D. J., *Datafication in judicial case management in India*, presented at the Symposium on Diversity in Legal and Judicial Profession and the Politics of Merit and Exclusion in India, RHUL, London, 2024.
- [3] Abhijita, B., et al., *The NCRB suicide in India 2022 report: Key time trends and implications*, Indian Journal of Psychological Medicine, vol. 46, no. 6, pp. 606–607, 2024.
- [4] Queudot, M., Charton, É., and Meurs, M.-J., *Improving access to justice with legal chatbots*, Stats, vol. 3, no. 3, pp. 356–375, 2020.
- [5] McKay, C., *Predicting risk in criminal procedure: Actuarial tools, algorithms, AI and judicial decision-making*, Current Issues in Criminal Justice, vol. 32, no. 1, pp. 22–39, 2020.
- [6] Ray, P. P., *ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope*, Internet of Things and Cyber-Physical Systems, vol. 3, pp. 121–154, 2023.
- [7] Naik, D., Naik, I., and Naik, N., *The AI engine of creation: Exploring the capabilities of AI chatbots based on generative AI, large language models and large multimodal models*, Authorea Preprints, 2024.
- [8] Murdoch, B., *Privacy and artificial intelligence: Challenges for protecting health information in a new era*, BMC Medical Ethics, vol. 22, pp. 1–5, 2021.
- [9] Martin, K., *Ethical implications and accountability of algorithms*, Journal of Business Ethics, vol. 160, no. 4, pp. 835–850, 2019.
- [10] Hilal, S., and Litsey, B., *Reducing police turnover: Recommendations for the law enforcement agency*, International Journal of Police Science & Management, vol. 22, no. 1, pp. 73–83, 2020.

- [11] Bhatt, H., et al., *Integrating Industry 4.0 technologies for the administration of courts and justice dispensation—A systematic review*, Humanities and Social Sciences Communications, vol. 11, no. 1, pp. 1–16, 2024.
- [12] Androutsopoulou, A., et al., *Transforming the communication between citizens and government through AI-guided chatbots*, Government Information Quarterly, vol. 36, no. 2, pp. 358–367, 2019.
- [13] Jain, P., *Artificial intelligence for sustainable and effective justice delivery in India*, OIDA International Journal of Sustainable Development, vol. 11, no. 6, pp. 63–70, 2018.
- [14] Busch, P. A., *The artificial bureaucrat: Artificial intelligence in street-level work*, Digital Government: Research and Practice, 2025.
- [15] Ugwudike, P., and Morgan, G., *Bridging the gap between research and frontline youth justice practice*, Criminology & Criminal Justice, vol. 19, no. 2, pp. 232–253, 2019.
-

## 10.2 Appendices

The appendices includes supplementary content supporting implementation, testing, and user documentation.

### *A. System User Manual*

**Login Process:** Users must authenticate using Aadhaar ID, BCI ID, or Government ID, depending on their role.

**Chat Usage:** After successful login, users can ask legal questions in the chatbot interface.

**Chat History:** Each user has access to their previous chat sessions, visible via sidebar.

**Admin Panel:** Admins can securely upload new legal documents and FAQs using the upload interface.

### *B. Dataset Samples*

Sample legal FAQs on IPC sections, bail laws, property rights, etc.

Uploaded documents include court judgments, public legal notices, and statutes.

File types accepted: .pdf, .docx, .txt, .xlsx.

### *C. Screenshots and Output Snapshots*

Login screen for each role.

AI chatbot responding to user queries.

Admin panel with upload confirmation.

Chat history view with timestamps and session titles.

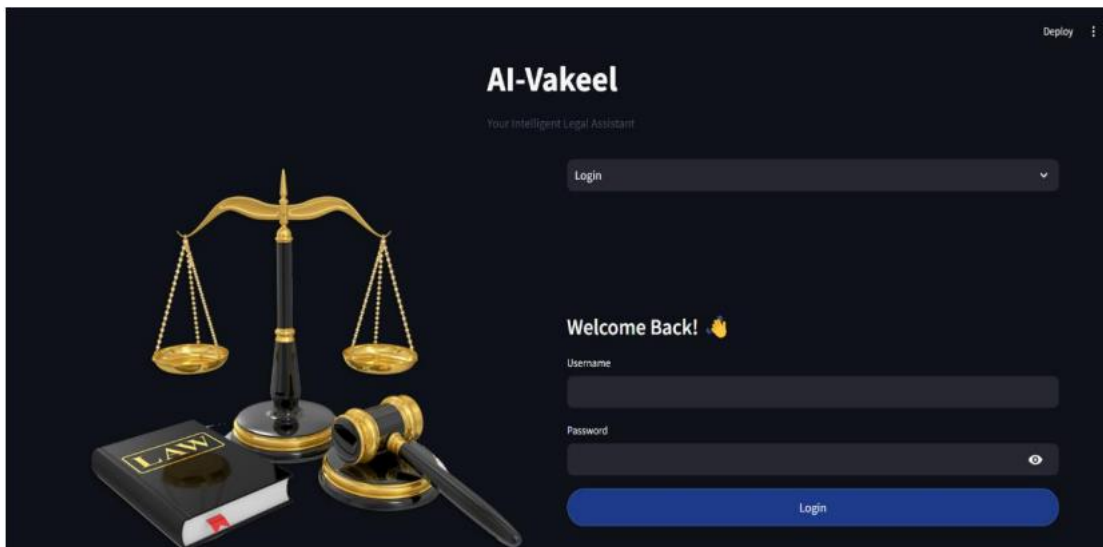
### *D. Testing Logs*

Manual test case logs with success and failure tracking.

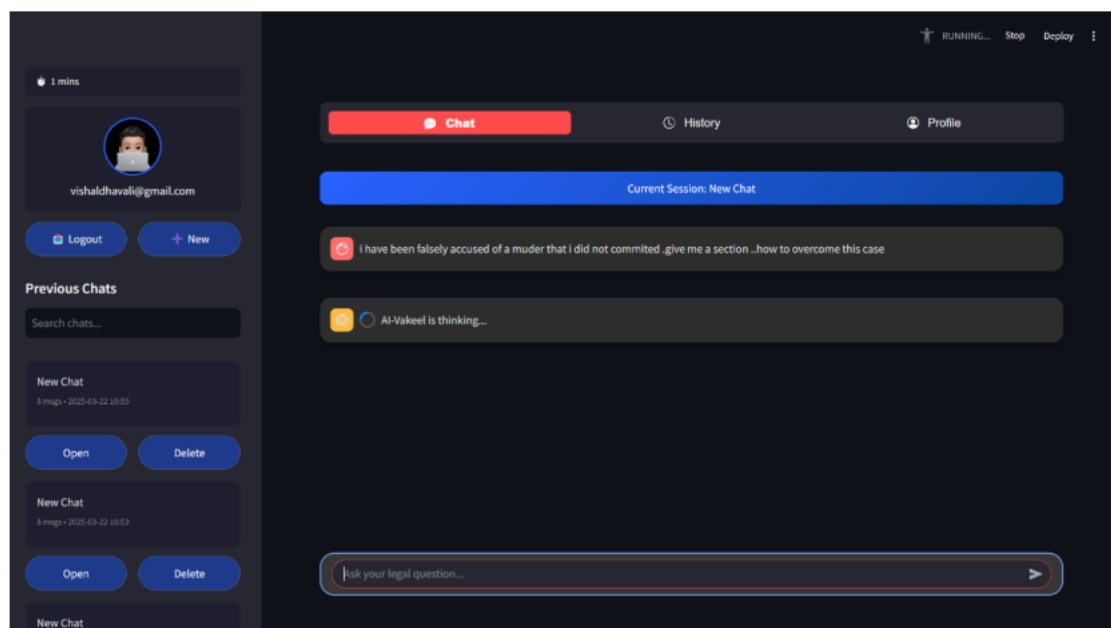
Timestamped chat logs from different roles.

Backend API logs showing session flow and AI engine interactions.

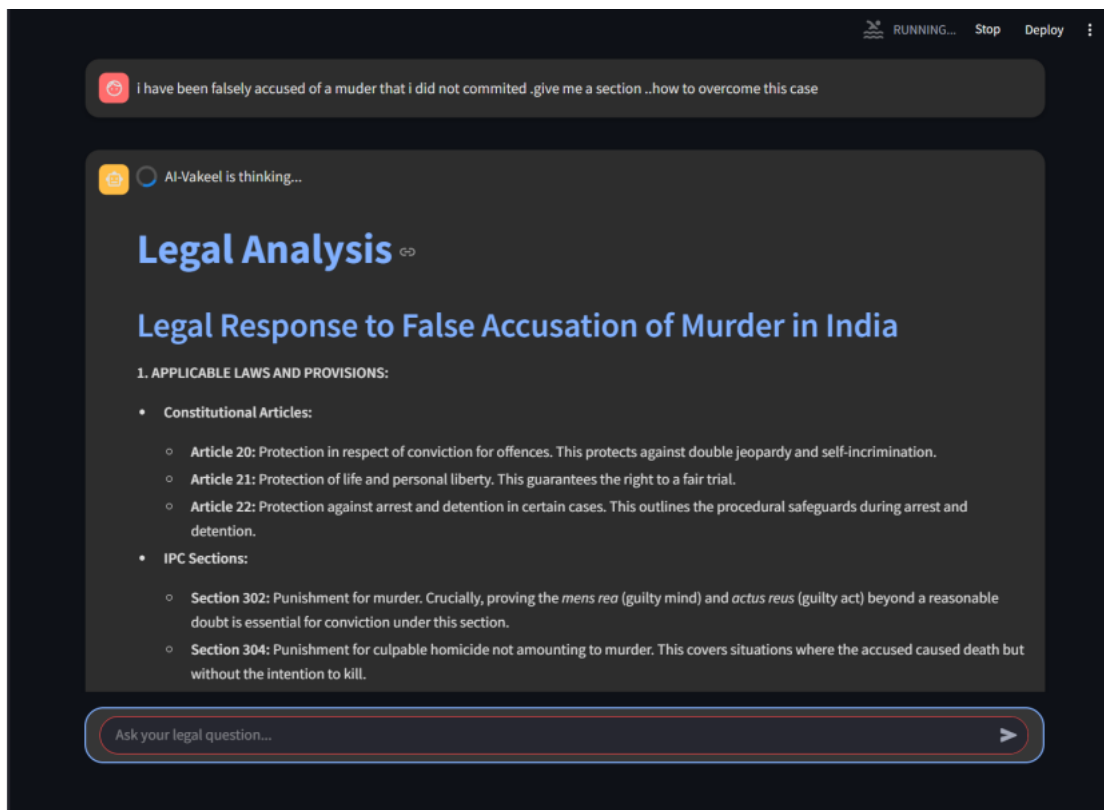
### Project Dashboard Images:



**Fig 3:** User Authentication Interface of AI-Vakeel: An AI-powered Legal Assistant System



**Fig 4:** Interactive Chat Interface of AI-Vakeel Legal Assistant System Demonstrating Real-time Legal Query Processing



**Fig 5:** AI-Vakeel's Legal Analysis Interface Demonstrating Constitutional and IPC Provisions

# AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary

Vishal Dhavali  
Dept of Computer Science  
Engineering Presidency  
University

Bengaluru, India  
[vishaldhavali2209@gmail.com](mailto:vishaldhavali2209@gmail.com)

Bhavyashree S  
Dept of Computer Science  
Engineering Presidency  
University

Bengaluru, India  
[minibhavya003@gmail.com](mailto:minibhavya003@gmail.com)

Ramakrishna  
Dept of Computer Science  
Engineering Presidency  
University

Bengaluru, India  
[RAMAKRISHNA.2021CSE0517@presidencyuniversity.in](mailto:RAMAKRISHNA.2021CSE0517@presidencyuniversity.in)

Tejas S P

Dept of Computer Science  
Engineering Presidency  
University

Bengaluru, India  
[unifier89@gmail.com](mailto:unifier89@gmail.com)

Aditya Gupta

Dept of Computer Science  
Engineering Presidency  
University

Bengaluru, India  
[adityagupta.ag2308@gmail.com](mailto:adityagupta.ag2308@gmail.com)

Thabassuam Khan.S

Assistant Professor Dept of  
Computer Science Engineering  
Presidency University

Bengaluru, India  
[Thabassumkhan@presidencyuniversity.in](mailto:Thabassumkhan@presidencyuniversity.in)

**Abstract**— AI-Vakeel addresses the gap in accessing legal assistance that is complex, slow, and inaccessible by offering an AI-driven platform designed to simplify and democratize legal support in the Indian judicial context. This system combines advanced artificial intelligence with user-centred design to provide timely, understandable, and contextually accurate legal guidance. Whether serving legal professionals, judicial officers, or the public, AI-Vakeel responds to queries in natural language, referencing relevant statutes and case laws to deliver clear and actionable insights. The platform employs a conversational interface to enhance usability while maintaining role-based access control to ensure data security and relevance. Its architecture integrates semantic search, vector-based retrieval, and a generative transformer model tailored to Indian legal discourse. Beyond convenience, AI-Vakeel is developed with a strong emphasis on ethical deployment, fairness, and the preservation of human oversight in legal decision-making. Drawing on global precedents while adapting to India's unique legal ecosystem, AI-Vakeel exemplifies the responsible use of AI to support a more inclusive and efficient justice system. **Keywords**— Artificial Intelligence (AI), Natural Language Processing (NLP), Large Language Models (LLMs), Financial Market Analysis, Risk Assessment, Sentiment Analysis, Stock Prediction, Machine Learning in Finance, Data Preprocessing, System Architecture, Predictive Analytics, Financial Technology (FinTech), Behavioral Finance, Neural Networks, Ethical AI in Finance.

**INTRODUCTION**The National Judicial Data Grid (NJDG) in India has over 31 million cases pending in district and Taluka courts, with 23 million of those being older than a year. Additionally, the High Court of India has more than 4.5 million pending cases [1]. The Supreme Court tracks judicial

data across the country in real time, by NJDG and iJuris, two platforms designed for sharing information at the district level. They monitor live data on both pending and resolved cases, as well as track vacancies and infrastructure. The NJDG serves as a comprehensive repository of judicial data around 3,000 district courts, the High Courts, and the Supreme Court [2]. The National Crime Records Bureau (NCRB) has reported an increase in suicide rates in India. The rates vary significantly across states, with Bihar showing a low of 0.6 per 100,000 population, while Sikkim has a much higher rate of 43.1 per 100,000 population [3].

When making decisions about sentencing, parole, bail, extended supervision, and continuing detention orders for high-risk offenders, among other aspects of criminal procedure, risk assessments are carried out. These risk assessments, which are grounded in clinical evaluations, framed by common-law principles and legislation, and embody the idea of individualised justice, have historically been the result of the human discretion and intuition of judicial officers. However, statistical, data-driven assessments of risk are intruding as criminal procedure becomes more technologically advanced. A variety of AI, algorithmic, and machine learning tools are being used more and more to support human judicial evaluation functions. These tools claim to offer objective, consistent risk assessments as well as precise predictive capabilities [4].



AI-supported risk assessment systems provide uniformity in decision-making, which can minimize human biases and errors in judicial proceedings. Algorithmic systems in criminal justice are being implemented to predict reoffending risks, evaluate threats to public safety, and decide on the right sentencing. Even with their potential benefits, ethical issues have been raised around the world over algorithms as proprietary items with built-in statistical bias, as well as the erosion of judicial human assessment in favor of the machine [5].

\* Synthesize and summarize legal documents, including contracts, legislation, and court decisions. Help legal professionals prepare legal documents, such as contracts, pleadings, and briefs.

\* Give prompt and precise responses to legal queries based on applicable statutes and case law. Examine and forecast the outcome of legal controversies based on past data and legal precedents.

\* Simplify communication and cooperation among legal experts by making complicated legal terminology easy to use and allowing information exchange [6].

## II. LITERATURE REVIEW

The majority of contemporary AI chatbots are constructed upon two forms of gen AI models: Large Language Models (LLMs) and Large Multimodal Models (LMMs). LLMs primarily process and produce a single form of data, and LMMs are capable of processing and producing multiple forms of data or modalities like image, text, video, and audio. These gen AI-based chatbots are constantly developing and refining their abilities to perform more like humans [7]. The fast pace of development in these technologies has resulted in dramatic enhancements in natural language processing, contextual comprehension, and response generation. Current advancements have demonstrated that these models are capable of carrying out cogent conversations, comprehending intricate queries, and even displaying emotional intelligence to a certain degree. This development has provided new avenues for their use across different industries, such as law enforcement and justice systems, where they can be used to help process large amounts of information and offer initial analysis [8].

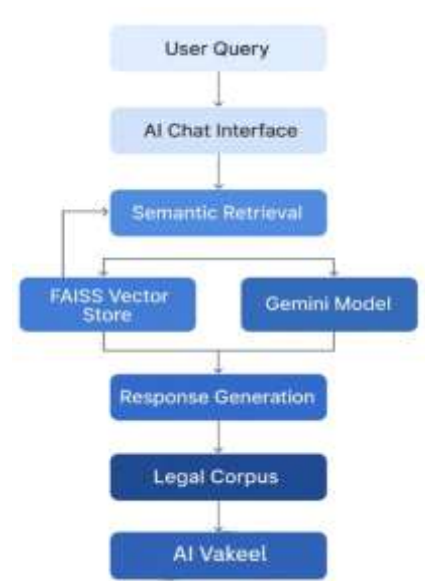
Algorithms govern our lives can determine if someone gets a job, a loan, and which political commercials and news articles consumers read. Algorithms for sentencing unconsciously erode our attempt at individualised and equal justice. Sentences should be based on the facts, the law, the actual offences, the individual circumstances of each case, and the criminal record of the defendant. They should not be based on immutable characteristics which the person cannot alter, or on potential for future offending which has yet to take place [9].

As law enforcement evolves in a changing world, further studies need to be done on the experiences and views of those working in the field and retired officers. Those who have been through it are best placed, tell agencies how to get better and

brief new recruits [10]. Their experience is especially useful in interpreting the complexities of contemporary policy, such as incorporating new technologies, relations with the community, and the psychological effects of the work. By methodically gathering and analysing these experiences, law enforcement can establish more efficient training programs, refine operational processes, and design improved support mechanisms for officers. This transfer of knowledge is essential for preserving institutional memory and ensuring that lessons learned from previous experiences are used to shape future practices.

The implementation of digitalization has brought outstanding results in various nations like the International Criminal Court, Brazil, the Hague, Singapore, North American states, and Portugal for numerous applications like electronic evidence management, electronic case analysis in appellate courts, online dispute resolution, online-based trials, e-discovery, and electronic case management. The various applications in the administration of courts are achieved by integrating Industry 4.0 technologies that can create a resilient and sustainable infrastructure in the courts [11]. This digital transformation has significantly improved the efficiency and transparency of judicial processes, reducing case backlogs and enhancing access to justice. The use of AI-based tools has further enhanced these advantages, with automated document examination, predictive analysis of case outcomes, and better case management systems being possible. These developments have also ensured greater collaboration between various judicial authorities and enhanced the overall quality of legal services rendered to citizens. The success of these implementations is a model for other jurisdictions seeking to update their judicial systems without compromising the integrity and fairness of legal proceedings [12].

## III. WORKFLOW DIAGRAM



. The architecture of AI-Vakeel is crafted to facilitate smart, secure, and efficient legal query resolution by merging semantic understanding with generative AI. The entire workflow is shown in the Figure, which showcases

several closely integrated components that work together to turn a user's legal question into a precise, law-based answer.

1. **User Query:** The process starts when a user submits a legal question through the web interface. This user could belong to one of three authorized groups: a general public member, a registered legal professional, or a judge. The system is designed to handle typed queries in natural language, covering procedural, statutory, or constitutional law. These inputs are then sent to the backend engine for processing.
2. **AI Chat Interface:** This module acts as the main interaction part for users. It captures and manages their queries, oversees authentication states, and keeps track of session histories. The chat interface offers a conversational experience while ensuring access control is modified to each user's role. This way, users only interact with data relevant to their authorization level. Once a query is received, it gets directed to the semantic retrieval pipeline.
3. **Semantic Retrieval:** The user's natural language question is transformed into a high-dimensional vector using a finely-tuned Sentence-BERT model. This vector captures the deeper semantic meaning of the question, going beyond mere keyword matching. The embedding is then utilized to search for similar documents in a pre-indexed legal knowledge base through the Facebook AI Similarity Search (FAISS) retrieval system.
4. **FAISS Vector Store:** FAISS serves as the backbone for quick and scalable similarity searches. The legal corpus, which includes statutory texts, judgments, and constitutional materials, is divided into manageable pieces, embedded, and indexed in FAISS. When a query vector is submitted, FAISS swiftly retrieves the top-k most relevant chunks from this index, ensuring a low-latency and high-accuracy search.
5. **Gemini Model:** The top-k context chunks obtained from FAISS are fed into a generative language model. In this research, a Gemini-based transformer model is designed to grasp complex legal language and produce responses that sound human. By employing retrieval-augmented generation (RAG), the model pulls together legal content to craft a precise and clear answer to the user's original question.
6. **Response Generation:** The Gemini model delivers a final response that merges information sourced from legal documents with generative insights. This answer is well-structured, relevant to the context, and in line with Indian legal terminology and jurisprudence. It might even include citations from pertinent articles, case laws, or legal provisions.
7. **Legal Corpus:** The system's knowledge base is built from verified legal documents, including sections of the Indian Penal Code, constitutional articles, case law summaries, legal FAQs, and government policy documents. These documents are meticulously cleaned, chunked, embedded, and regularly updated through a dedicated ingestion pipeline.
8. **AI Vakeel Delivery and Feedback:** The final answer is showcased through the AI-Vakeel interface. Each response is

logged and linked to the user session, allowing for future reference and system evaluation. If users have the right access, they can check past queries or upload new legal documents for ingestion. The system maintains strict role-based access control throughout to safeguard user data and uphold institutional integrity.

Overall, the AI-Vakeel system combines semantic search, cutting-edge AI technology, and secure access. It provides a reliable solution for automated legal support in India, effortlessly linking complex legal information with users through smart, human-like interactions instead of just relying on simple keyword matching.

#### IV. METHODOLOGY

The project employs an advanced legal aid system through an elaborate methodology blending modern software engineering practices and AI-powered solutions. The software development process is a systematic and iterative form of document ingestion and processing. PDF files are processed in a systematic way using PyPDF2, where they are processed for text extraction, cleaning, and chunking using the Recursive Character Text Splitter, which divides legal texts into chunks of a sensible size while preserving context and relationships between different sections. The system is founded on a dual-document repository structure, where case studies are stored separately from typical legal documents, allowing for specialty processing and retrieval based on the nature of the legal query.

Phase 1: Develop a chatbot to assist the legal sphere: judges, lawyers, and litigants.

Phase 2: Case study for the sociological and ethical analyses.

**4.1 AI Architecture and Implementation:** The AI deployment is in a layered architecture, with Google's Generative AI (Gemini) as the underlying language model augmented by Lang Chain's orchestration ability. This allows the creation of intricate chains of dialogue that can respond to intricate legal queries while maintaining context and coherence. The system utilizes a hybrid retrieval methodology, employing vector similarity search by FAISS along with traditional database queries to maintain both semantic appropriateness and factual correctness in the response. The vector storage system utilizes Google's embedding model to create high-dimensional representations of legal documents to enable efficient similarity matching and context retrieval.

User interaction is handled by a properly designed authentication and session handling system that utilizes secure password hashing, session tracking, and user preference storage. The chat interface uses a stateful design pattern that maintains conversation history and context and incorporates real-time feedback mechanisms. The system utilizes a modular architecture in which each module (frontend, backend, AI processing) is executed independently but communicates using well-defined interfaces, so, to enable simple updates and maintenance.



#### 4.2 Error Handling and System Monitoring:

Error handling and logging are implemented at multiple levels, from low-level system operations to high-level user interactions. The system maintains detailed logs of all operations, including document processing, user interactions, and AI responses, enabling comprehensive debugging and system monitoring. The development process includes continuous testing through a data generation system that simulates various legal scenarios, from court cases to traffic violations, ensuring system reliability across different use cases. The project is on a microservices architecture in which each part (user management, document processing, AI interaction, etc.) is a standalone service that communicates by API endpoints. This supports horizontal scaling as well as easy addition of new functionality. The system also has a robust backup and recovery scheme, in which the vector store as well as the user data are backed up regularly, maintaining data integrity and system reliability.

Configuration management is handled through environment variables and configuration files, and deployment to different environments is simple. Version control is implemented on all components, so it is simple to track changes and roll back when needed. Security is enforced at different levels, including input validation, secure password storage, and session management.

The methodology emphasizes code reusability, modularity, and rigorous separation of concerns between the system components. Each module encapsulates a single responsibility, which ensures the codebase remains extensible and maintainable. The system is extensively documented, encompassing inline comments within the code, API documentation, and user manuals, to ease onboarding new users and developers. Performance optimization is achieved by a variety of techniques, including caching of frequently accessed data, optimized vector search algorithms, and optimized database queries. Load balancing and resource management are used in the system to provide consistent performance under differing loads. Monitoring and analytics are integrated in the system to track user interactions, system performance, and AI response quality.

**4.3 Agile Development and Testing:** The project uses an agile development model with frequent iterations and continuous integration/deployment practices. It performs testing at various levels, such as unit tests, integration tests, and user acceptance tests. Automated testing frameworks and continuous integration pipelines are incorporated in the system to maintain code quality and system reliability.

The process also emphasizes user experience with consideration for interface design, response speed, and handling of errors. Progressive enhancement is utilized, meaning that basic functions are provided when things are less than perfect, with additional advanced functionality provided when capabilities are available. Accessibility is considered in every aspect of development as well, accommodating multiple user requirements and preferences.

The system architecture supports easy integration with external legal databases and services using clearly defined data exchange formats and APIs. The approach leverages frequent security audits and updates threats as they arise.

Code review and collaboration form the core of the design process, where multiple developers develop multiple components of the system independently while maintaining system integrity.

The project uses an all-encompassing backup and disaster recovery approach with frequent backups of all the important data and systems. The approach uses frequent performance testing and optimization so that the system can support increasing loads and complex queries. The system architecture supports smooth scaling, vertical and horizontal scaling, to support increasing user bases and increasing data volumes. The process development also emphasizes sustainability and maintainability through clear documentation, adherence to standardized coding practices, and regular code review. Automated deployment and monitoring processes are used in the system, enabling quick identification and correction of issues. Ongoing updating and improvement of the process from user feedback and system performance metrics are included in the methodology, which guarantees ongoing improvement and flexibility to respond to changing needs.

#### V. FUTURE DIRECTION

Indian courts are already undergoing a transformation process by becoming digital, and the new branch of science called AI can help in surprising ways in making justice delivery sustainable and in preventing the backlog of cases. Judiciary in some parts of developed countries like U.S.A. and Canada has already used AI systems to aid the judges in making decisions on matters like the grant of bail and release of offenders on parole. In the same way, in India too, court work can be identified, which can be accelerated by using intelligent machines. Such work can range from routine work like service of processes to complicated ones like evaluation of evidence [13]. Use of AI in judicial systems has provided promising results in case management streamlining, reduction of human errors, and increasing the overall efficiency of the justice delivery system. This use of technology not only helps in the elimination of case backlog but also in ensuring more consistent and objective decision making.

In Latvia, a 24/7 chatbot virtual assistant has been created through the use of neural networks and natural language technologies to respond to common questions posed by current and potential Latvian entrepreneurs. Available on a website and Facebook Messenger, it allows clients to get answers promptly and without delay. It is an alternative to a personal visit or phone call and can respond to various questions, ranging from status questions about registration documents filed. The AI tool is observed to eliminate routine work from street level bureaucrats so that they can carry out higher-value tasks instead, better aligned with their skills and providing more challenging career opportunities [14]. This new approach illustrates how AI can facilitate the provision of public services more easily, at the same time, enhancing the level of job satisfaction and efficiency among government officials. The success of such applications in

Latvia can be used as an exemplary case study for other countries that wish to expand their public service delivery system.

Although there has been extensive discourse over programme integrity, very few research articles have dealt with this issue and have been restricted to occurring within adult criminal justice systems and jurisdictions. It is intended to fill the 'programme integrity' void [15] that the current article attempts. The integration of digital technologies and AI into the judiciary should be weighed in depth with the preservation of the integrity and impartiality of legal processes. While numerous benefits are contained within technology, caution must be exercised to ensure technology is being embraced in a manner that preserves the substance of justice and due process. Future research should work towards establishing frameworks and guidelines that will enable judicial systems across the globe to incorporate AI successfully while ensuring the utmost level of program integrity and ethics.

## V. RESULT



Fig 1: Landing Page

Interface of AI-Vakeel Demonstrating Integration of User Engagement Elements and Legal AI Assistant Framework The landing interface showcases the platform's focus on user-friendly design, clearly highlighting its value as an AI-driven legal assistant. It strikes a great balance between looking professional and being functional, featuring straightforward call-to-action buttons like "Start Chat Now" and "Learn More," along with brief service descriptions. By emphasizing "instant, accurate legal advice powered by advanced AI" and grounding itself in the Indian Constitution and Laws, the system shows its dedication to providing trustworthy legal help. The blend of modern design elements with a professional legal backdrop makes the platform both



approachable and authoritative, ensuring that users can navigate complex legal issues that legal consultations require. This approach effectively combines technical sophistication with user-friendliness, marking a notable leap forward in legal tech interfaces.

Fig 2: Feature Architecture Overview of AI-Vakeel Demonstrating Six Core Functionalities:

Expert Legal Advice, Security, Instant Response, Knowledge Base, Natural Language Processing, and Ubiquitous Access The system highlights six key functionalities, all thoughtfully crafted to deliver thorough legal support. By integrating Expert Legal Advice with a focus on Indian law and ensuring Secure & Confidential data handling, it lays a solid groundwork for professional legal consultations. Additionally, the ability to provide Instant Responses (24/7) and engage in Natural Conversations through AI enhances the user experience. The Comprehensive Knowledge base, along with Universal Access across devices, guarantees that users can access legal information no matter where they are or what device they prefer.



Fig 3: User Authentication Interface of AI-Vakeel:

An AI-powered Legal Assistant System The authentication interface sets a secure stage for user interactions. It effectively balances the need for security with user-friendliness, incorporating email verification and profile management while keeping the experience intuitive. Additionally, the use of legal symbols strengthens the platform's professional legal context.

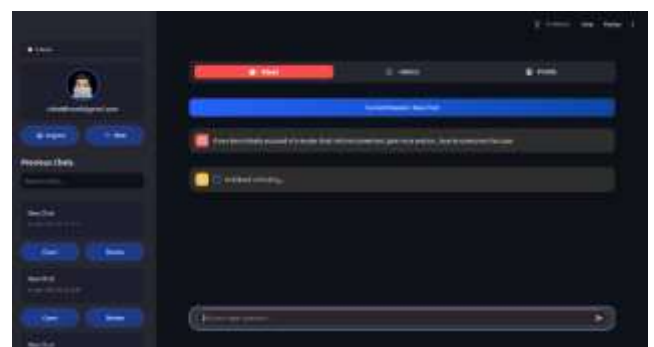


Fig 4: Interactive Chat Interface of AI-Vakeel Legal Assistant System Demonstrating Real-time Legal Query Processing

The interactive consultation interface marks a significant step forward in enhancing the user experience in legal technology. Our analysis reveals that the real-time chat-

based system enables natural communication between users and the AI. The interface is organized into clear sections like Chat, History, and Profile, making it easy to navigate while keeping the conversation flowing smoothly. With features for session management and chat history, users can easily track and refer back to their legal consultations.



**Fig 5: AI-Vakeel's Legal Analysis Interface Demonstrating Constitutional and IPC Provisions**

The system's legal analysis capabilities show a sophisticated method for representing legal knowledge and generating responses. With complex questions, it produces well-structured analyses that include relevant Constitutional Articles and IPC Sections. The way it organizes legal provisions hierarchically, clearly distinguishes between constitutional and statutory law, and explains legal concepts precisely, is particularly impressive. Its ability to weave together procedural safeguards and substantive law highlights its strength in multi-dimensional legal analysis. The quality of the system's responses is evident in its knack for delivering context-specific legal information while ensuring accuracy in citations and interpretations. The structured format of the legal analyses, as illustrated in Figure 5, helps users grasp complex legal concepts and see how they apply in real life. By integrating both constitutional protections and criminal law provisions, the system truly demonstrates its capability for in-depth legal analysis. Performance metrics show that the system is reliable, consistently generating responses and managing sessions effectively. The interface's knack for tackling complex legal questions while keeping users engaged in the advancement of legal tech. These findings indicate that AI Vakeel is offering dependable and accessible legal help through an AI-driven interface, successfully connecting the dots between traditional legal advice and AI-supported legal guidance.

## VI. CONCLUSION

AI-Vakeel isn't just another digital tool, it represents a thoughtful blend of technology and legal understanding, designed to make legal support more approachable, timely, and reliable. At its core, this platform is about people. Whether it's a lawyer preparing a brief, a judge seeking quick reference, or a citizen looking for basic legal clarity. It brings value to the table by simplifying complex legal processes and offering guidance that's easy to access and grounded in real law. The interface is built with concern, how users interact with legal systems, often under stress, with limited

knowledge, or in urgent situations. By offering instant responses, smart legal analysis, and secure, personalized experiences, it gives users a sense of confidence and control that's often missing in traditional legal journeys. As the Indian judicial system continues to modernize, tools like AI-Vakeel can play a vital role in reducing backlogs, supporting overburdened legal professionals, and achieving more consistent outcomes. It doesn't replace the human touch of legal practice, but it enhances it. It's about working with lawyers and judges, not instead of them, to improve how justice is served. Looking ahead, there's still much more to grow. Continuous updates, user feedback, and careful ethical oversight by making sure that AI-Vakeel stays relevant, responsible, and respectful of the legal system's values. But what's clear is this: The beginning of something important. By embracing AI thoughtfully, we are not making legal systems faster; making them better, inclusive, and more in tune with the needs of real people.

## VII.

## REFERENCE

- [1] Singhal, A.V.K., An Advanced Deep Learning Based Approach for Judicial Decision Support Process. International Journal of Electronics Engineering, 2021. 13(2): p. 18-23.
- [2] Varghese, D.J. Datafication in Judicial Case Management in India. in Symposium on Diversity in Legal and Judicial Profession and the Politics of Merit and Exclusion in India, RHUL, London. 2024.
- [3] Abhijita, B., et al., The NCRB suicide in India 2022 report: key time trends and implications. Indian Journal of Psychological Medicine, 2024. 46(6): p. 606-607.
- [4] Queudot, M., É. Charton, and M.-J. Meurs, Improving access to justice with legal chatbots. Stats, 2020. 3(3): p. 356-375.
- [5] McKay, C., Predicting risk in criminal procedure: actuarial tools, algorithms, AI and judicial decision-making. Current Issues in Criminal Justice, 2020. 32(1): p. 22-39.
- [6] Ray, P.P., ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and Cyber Physical Systems, 2023. 3: p. 121-154. Naik, D., I
- [7] Naik, and N. Naik, The AI Engine of Creation: Exploring the Capabilities of AI Chatbots based on Generative AI, Large Language Models and Large Multimodal Models. Authorea Preprints, 2024.
- [8] Murdoch, B., Privacy and artificial intelligence: challenges for protecting health information in a new era. BMC medical ethics, 2021. 22: p. 1-5.
- [9] Martin, K., Ethical implications and accountability of algorithms. Journal of business ethics, 2019. 160(4): p. 835-850.
- [10] Hilal, S. and B. Litsey, Reducing police turnover: Recommendations for the law enforcement agency.

International journal of police science & management, 2020.  
22(1): p. 73-83.

[11] Bhatt, H., et al., Integrating industry 4.0 technologies for the administration of courts and justice dispensation—a systematic review. Humanities and Social Sciences Communications, 2024. 11(1): p. 1-16.

[12] Androutsopoulou, A., et al., Transforming the communication between citizens and government through AI-guided chatbots. Government information quarterly, 2019. 36(2): p. 358-367.

[13] Jain, P., Artificial Intelligence for sustainable and effective justice delivery in India. OIDA International Journal of Sustainable Development, 2018. 11(06): p. 63-70.

[14] Busch, P.A., The Artificial Bureaucrat: Artificial Intelligence in Street-Level Work. Digital Government: Research and Practice, 2025.

[15] Ugwudike, P. and G. Morgan, Bridging the gap between research and frontline youth justice practice. Criminology & Criminal Justice, 2019. 19(2): p. 232-253.

## ORIGINALITY REPORT

8%

SIMILARITY INDEX

5%

INTERNET SOURCES

5%

PUBLICATIONS

2%

STUDENT PAPERS

## PRIMARY SOURCES

1	Peter André Busch. "The Artificial Bureaucrat: Artificial Intelligence in Street-Level Work", Digital Government: Research and Practice, 2025 Publication	2%
2	<a href="https://papers.ssrn.com">papers.ssrn.com</a> Internet Source	2%
3	<a href="https://www.nature.com">www.nature.com</a> Internet Source	1%
4	<a href="https://www.researchgate.net">www.researchgate.net</a> Internet Source	1%
5	Submitted to Queensland University of Technology Student Paper	1%
6	Nishant Jain, Gaurav Goel. "An Approach to Get Legal Assistance Using Artificial Intelligence", 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2020 Publication	<1%
7	<a href="https://www2.mdpi.com">www2.mdpi.com</a> Internet Source	<1%
8	<a href="https://quarxiv.authorea.com">quarxiv.authorea.com</a> Internet Source	<1%

9

Submitted to University of Glamorgan

Student Paper

<1 %

10

Teemu Mäkiaho, Henri Vainio, Kari Koskinen.  
"Model-based wear prediction of milling  
machine blades", Procedia Computer Science,  
2022

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On



## Research Paper Publication Certificates:

DOI: 10.55041/IJSREM48058



ISSN: 2582-3930

Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

### CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to



**Thabassuam Khan.S**

in recognition to the publication of paper titled

**AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary**

published in IJSREM Journal on **Volume 09 Issue 05 May, 2025**

Editor-in-Chief  
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com

DOI: 10.55041/IJSREM48058



ISSN: 2582-3930

Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

### CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to



**Vishal Dhavali**

in recognition to the publication of paper titled

**AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary**

published in IJSREM Journal on **Volume 09 Issue 05 May, 2025**

Editor-in-Chief  
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com

DOI: 10.55041/IJSREM48058



ISSN: 2582-3930

Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

### CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to



**Bhavyashree S**

in recognition to the publication of paper titled

**AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary**

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

  
Editor-in-Chief  
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com

DOI: 10.55041/IJSREM48058



ISSN: 2582-3930

Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

### CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to



**Tejas S P**

in recognition to the publication of paper titled

**AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary**

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

  
Editor-in-Chief  
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM48058



ISSN: 2582-3930

Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

### CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to



**Ramakrishna**

in recognition to the publication of paper titled

**AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary**

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

  
Editor-in-Chief  
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com

DOI: 10.55041/IJSREM48058



ISSN: 2582-3930

Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

### CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to



**Aditya Gupta**

in recognition to the publication of paper titled

**AI-Vakeel: An AI-Powered Platform for Smart Legal Query Resolution in the Indian Judiciary**

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

  
Editor-in-Chief  
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com

## Chapter 11: Enclosure

### 11.1 Alignment with Sustainable Development Goals (SDGs)

The *AI Vakeel* project supports the United Nations Sustainable Development Goals (SDGs) by addressing core challenges related to legal accessibility, technological innovation, and social inclusion. Through the integration of artificial intelligence, open-source tools, and secure role-based interaction, this system directly contributes to improving justice delivery and empowering citizens with legal knowledge.



#### ✓SDG 16: Peace, Justice and Strong Institutions

*AI Vakeel* promotes access to justice for all by:

- Providing legal information in an understandable, user-friendly format.
- Empowering citizens to seek legal guidance without financial or institutional barriers.
- Supporting legal literacy and transparency through AI-driven document search.
- Enhancing institutional efficiency by reducing reliance on physical infrastructure and manual legal reference.

This aligns directly with **Target 16.3** (Promote the rule of law and ensure equal access to justice for all).

### ✓SDG 9: Industry, Innovation and Infrastructure

The project reflects the principles of innovation and digital infrastructure by:

- Leveraging AI, NLP, and vector search to create a smart legal assistant.
- Using modular, scalable architecture ready for cloud deployment or integration with institutional systems.
- Offering a digital-first approach to legal document processing and chatbot interaction.

It supports **Target 9.5** (Enhance scientific research and innovation) and **9.c** (Increase access to ICT and strive to provide universal and affordable access to the internet).

### ✓SDG 10: Reduced Inequalities

By offering an inclusive, role-sensitive platform, *AI Vakeel* helps bridge legal awareness gaps across social and regional boundaries:

- Ensures equal access to legal knowledge regardless of a user's background or profession.
- Plans for multilingual NLP will expand reach to non-English-speaking populations.
- Encourages participation from marginalized groups in legal dialogue and awareness.

This addresses **Target 10.2** (Promote the social, economic, and political inclusion of all).

---