



LegalBot: An AI-Based Legal Chatbot System for Accessible Legal Guidance

Bhoomi Singh¹, Rishabh Pandey², Bhoomi Gupta³, Prarthita Lade⁴, Mr. Kashif Shaikh⁵

Frontend Developer, Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Mumbai, India¹

AI Engineer, Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Mumbai, India²

Backend Developer, Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Mumbai, India³

Database Manager, Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Mumbai, India⁴

Guide, Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Mumbai, India⁵

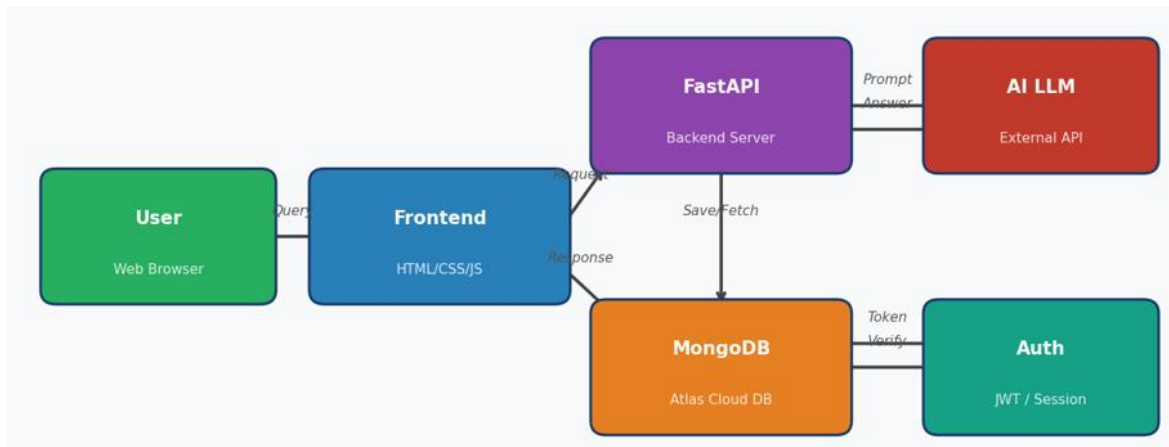


Fig. 1. LegalBot Data Flow Diagram

Abstract: Access to legal information remains a significant challenge for a large portion of the general public due to complex legal terminology, high consultation costs, and limited availability of legal professionals. This paper presents LegalBot, an AI-powered legal chatbot system designed to bridge this gap by providing real-time, conversational legal guidance to users. Built on a modern full-stack architecture comprising Python, FastAPI, and MongoDB, LegalBot integrates an external AI language model to generate context-aware responses to user queries. The system supports user authentication, persistent chat history, and a clean web-based interface. This paper describes the system architecture, development methodology, implementation details, and evaluation results. A comparative analysis with existing legal chatbot systems is also provided. Future enhancements including multilingual support, voice assistance, and mobile application deployment are discussed.

Keywords: Legal chatbot, Artificial intelligence, FastAPI, MongoDB, Natural language processing, Legal assistance, Full-stack system

I. INTRODUCTION

The legal domain has long been characterized by complexity and inaccessibility. Legal language, shaped by centuries of precedent and statute, is often incomprehensible to the ordinary citizen. This creates a significant barrier for individuals seeking to understand their rights, navigate legal procedures, or respond to legal disputes. In India alone, a shortage of legal professionals combined with a rapidly growing population has led to a substantial backlog of unresolved legal cases and widespread legal illiteracy [1].

The rapid advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies has opened new avenues for addressing these challenges. AI-powered chatbots have already demonstrated their effectiveness in



domains such as healthcare, customer service, and education. However, their penetration into the legal domain remains limited, and existing solutions often cater to specific legal niches or geographic jurisdictions.

This paper presents LegalBot, a conversational AI system developed as an academic mini-project to address the need for accessible legal guidance. LegalBot enables users to interact with an AI assistant through a clean web interface, ask legal questions in plain language, and receive relevant, understandable responses. The system also stores conversation history, supports user authentication, and is built on scalable cloud-compatible technologies.

The key contributions of this work are:

- Design and implementation of a full-stack AI-based legal chatbot using Python, FastAPI, and MongoDB.
- Integration of an external large language model API for generating intelligent, context-aware legal responses.
- A secure user authentication system with persistent chat history management.
- A comparative evaluation of LegalBot against existing legal chatbot systems in the literature.

The remainder of this paper is organized as follows. Section II reviews related work in the area of legal chatbots. Section III describes the system architecture and design. Section IV presents the development methodology and tools used. Section V discusses implementation details and results. Section VI presents a comparative analysis. Section VII outlines future directions, and Section VIII concludes the paper.

II. RELATED WORK

The development of AI-based legal assistance systems has gained momentum over the past decade. Several notable systems have been proposed and deployed, each addressing specific aspects of the legal domain.

LawGPT 1.0, introduced by Nguyen Ha Thanh [2], is a virtual legal assistant fine-tuned on GPT-3 for the legal domain. It is capable of answering legal questions and generating legal documents in English. However, LawGPT 1.0 does not incorporate Reinforcement Learning from Human Feedback (RLHF), limiting its adaptability. Additionally, it does not provide a user authentication system or persistent conversation storage, and it currently supports only the English language.

Avyanna, proposed by Sil et al. [1], is an Android-based legal chatbot specifically designed to address women-related legal issues under the Indian Penal Code. Avyanna provides emergency service buttons for police, ambulance, and fire services, and supports IPC section lookups. While it is highly relevant to the Indian context, it is limited to women-related legal matters and does not offer a general-purpose legal consultation interface.

DoNotPay, developed by Joshua Browder [3], is one of the earliest and most widely cited legal chatbots, originally designed to appeal parking tickets in the United Kingdom. While it has expanded significantly, it remains geographically and functionally restricted. Ross Intelligence, powered by IBM Watson [4], targets legal professionals for bankruptcy research, and LawGeex [5] focuses on automated contract review for businesses.

A systematic review of these systems reveals several common limitations: lack of general-purpose applicability, absence of user authentication, no persistent chat history, restricted language support, and reliance on static rule-based approaches. LegalBot is designed to address these gaps through a modern, scalable, and extensible full-stack architecture.

III. SYSTEM ARCHITECTURE

LegalBot follows a layered, modular architecture designed for scalability and maintainability. The system is composed of four primary components: a user interface layer, an API processing layer, an AI processing layer, and a database layer.

- User Interface: The frontend allows users to interact with the system.
- API Layer: The backend processes requests and responses.
- AI Processing: The external API generates intelligent responses.
- Database: MongoDB stores user and chat data.
- Response Output: The system sends results back to the user.

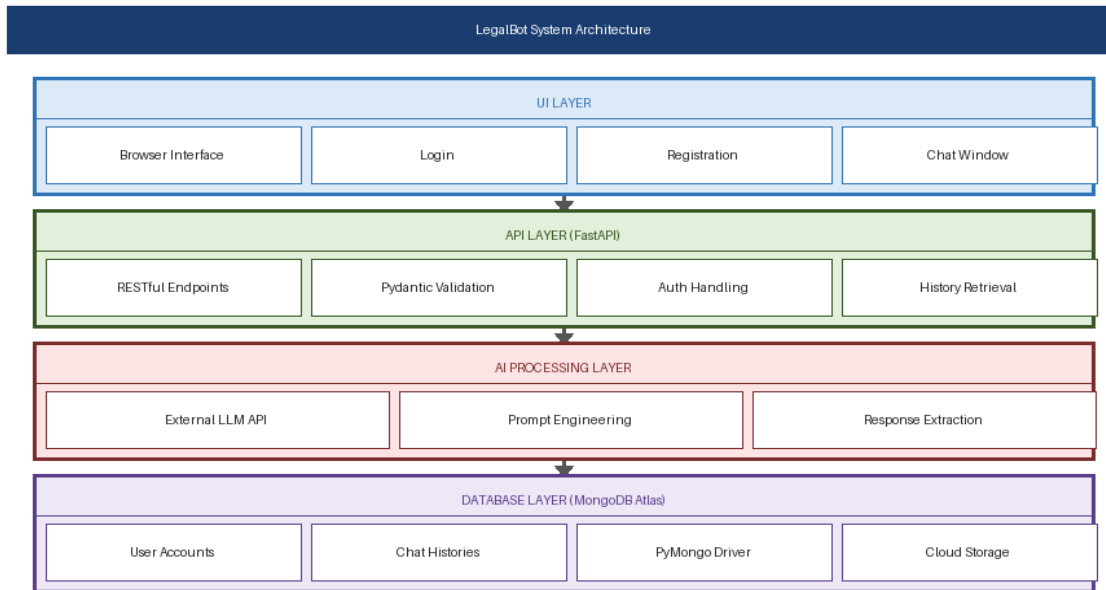


Fig. 2.

LegalBot System Architecture Diagram

A. User Interface Layer

The frontend provides a browser-based interface through which users register, log in, and interact with the chatbot. The interface is designed to be minimal and intuitive, allowing users without technical knowledge to access legal information easily. The home page serves as a navigation hub, and the chatbot interface displays the conversation in a threaded format.

B. API Layer

The backend is implemented using FastAPI, a high-performance Python web framework. FastAPI handles all client requests including user registration, authentication, chat message submission, and history retrieval. RESTful API endpoints are defined for each major system function, and request validation is enforced using Pydantic models.

C. AI Processing Layer

LegalBot integrates an external large language model (LLM) API to generate responses. When a user submits a query, the FastAPI backend forwards the message to the AI API along with a system prompt that constrains the model to respond as a legal assistant. The response is then extracted from the API output and returned to the frontend. This design allows the AI component to be swapped or upgraded without modifying the rest of the system.

D. Database Layer

MongoDB Atlas, a cloud-hosted NoSQL database, is used to persist user accounts and conversation histories. Each user record contains authentication credentials, while chat documents store message threads keyed by user ID. MongoDB was chosen for its schema flexibility, scalability, and compatibility with Python-based backends through the PyMongo driver.

IV. DEVELOPMENT METHODOLOGY

The LegalBot project was developed following the Agile software development methodology. Agile was selected for its suitability to academic team projects, where requirements may evolve and parallel development across components is necessary.

The project was organized into short iterative development cycles (sprints), each targeting specific functional deliverables. The team of four members was structured around functional roles to ensure parallel progress: AI Engineer (Rishabh Pandey) for integration of the AI API, prompt engineering, and response handling; Frontend Developer (Bhoomi Singh) for design and implementation of the user interface; Backend Developer (Bhoomi Gupta) for FastAPI server setup, endpoint development, and API integration; and Database Manager (Prarthita Lade) for MongoDB schema design, connection management, and data storage logic.



The development tools employed included Visual Studio Code as the primary code editor, GitHub for version control and collaborative development, Postman for API testing and validation, MongoDB Atlas for cloud database management, and Chrome DevTools for frontend debugging and performance inspection.

V. IMPLEMENTATION AND RESULTS

A. Technologies Used

The system is implemented using the following technology stack: Python as the primary programming language for backend development; FastAPI, a modern high-performance web framework for building RESTful APIs with native support for asynchronous processing and automatic documentation generation; MongoDB, a NoSQL database to store user profiles and chat histories; an external AI Language Model API invoked to generate intelligent responses to user queries; and HTML/CSS/JavaScript as standard web technologies for the frontend user interface.

B. Key System Features

The implemented system provides the following functional capabilities:

1. **User Registration and Authentication:** New users can create accounts using an email address and password. Returning users authenticate through a secure login process. Session management ensures that only authenticated users can access the chatbot.
2. **Real-Time Chat Interface:** Users submit legal queries through a chat window and receive AI-generated responses in real time. The interface maintains a running thread of the conversation.
3. **Persistent Chat History:** All conversations are saved to MongoDB and can be retrieved in future sessions, allowing users to review prior interactions and track ongoing legal matters.
4. **Responsive Frontend:** The interface is accessible via standard web browsers on both desktop and mobile devices.

C. Results and Evaluation

The system was tested for functional correctness across all major components. User authentication flows functioned correctly, with secure credential handling. The AI component successfully generated relevant and coherent responses to a range of legal queries, including questions about tenant rights, employment law, and contract basics. Conversation records were accurately persisted in MongoDB and retrieved correctly on subsequent sessions. The interface rendered correctly across tested browsers, with no significant layout or usability issues observed. API response times were within acceptable limits for an interactive chatbot application, with no critical performance bottlenecks identified during testing.

D. System Interface Screenshots

The following figures illustrate the user interface and working of the LegalBot system, including user registration, authentication, homepage interface, and chatbot interaction.

“The system is branded as LawSage in the user interface.”

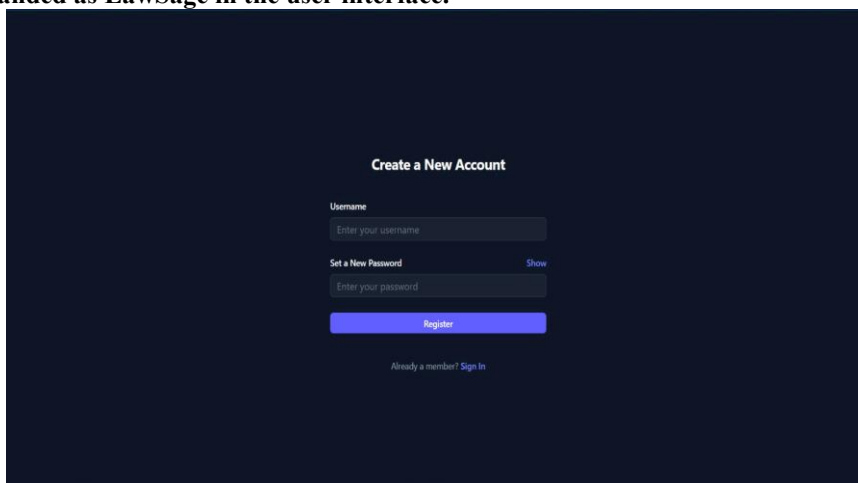


Fig. 3. User Registration Interface

Description (Fig. 3):

The registration interface allows new users to create an account by entering a username and password. The system ensures secure user onboarding with a simple and intuitive design.

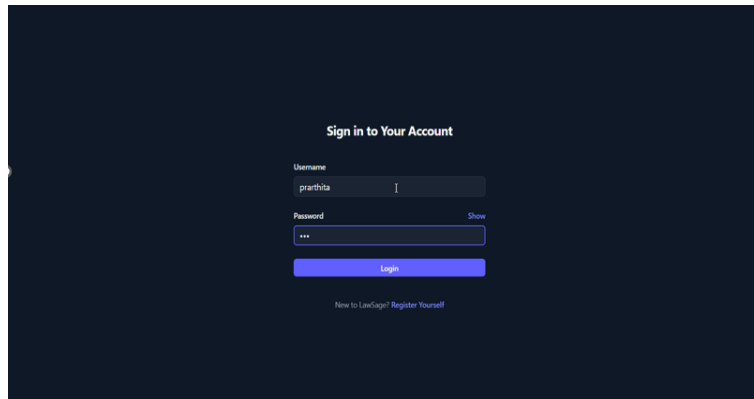


Fig. 4. User Login Interface

Description (Fig. 4):

The login interface enables existing users to authenticate using their credentials. It ensures secure access to the LegalBot system and redirects authenticated users to the chatbot interface.



Fig. 5. LegalBot Homepage Interface

Description (Fig. 5):

The homepage presents an overview of the LegalBot system, highlighting its purpose and features. It provides options for language selection and allows users to initiate chatbot interaction seamlessly.

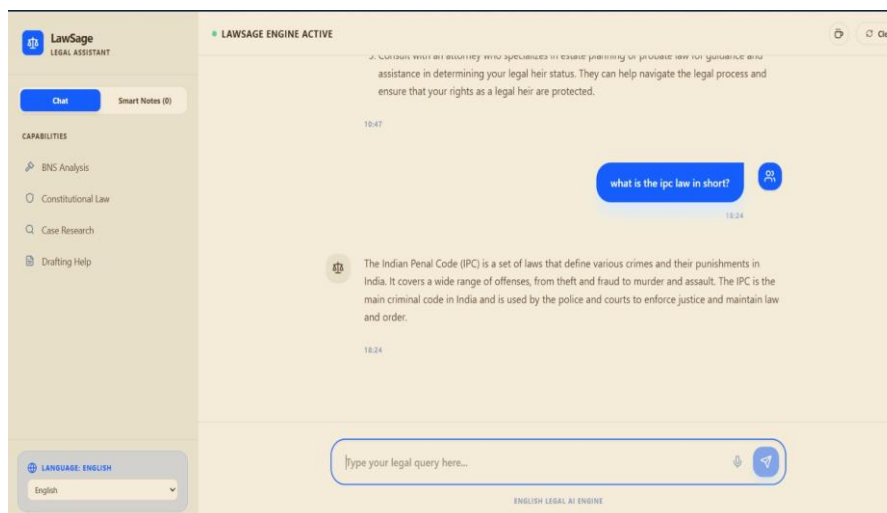


Fig. 6. LegalBot Chat Interaction Interface

**Description (Fig. 6):**

The chatbot interface allows users to input legal queries and receive AI-generated responses in real time. The system maintains a conversational flow and displays responses clearly, enhancing user experience and accessibility.

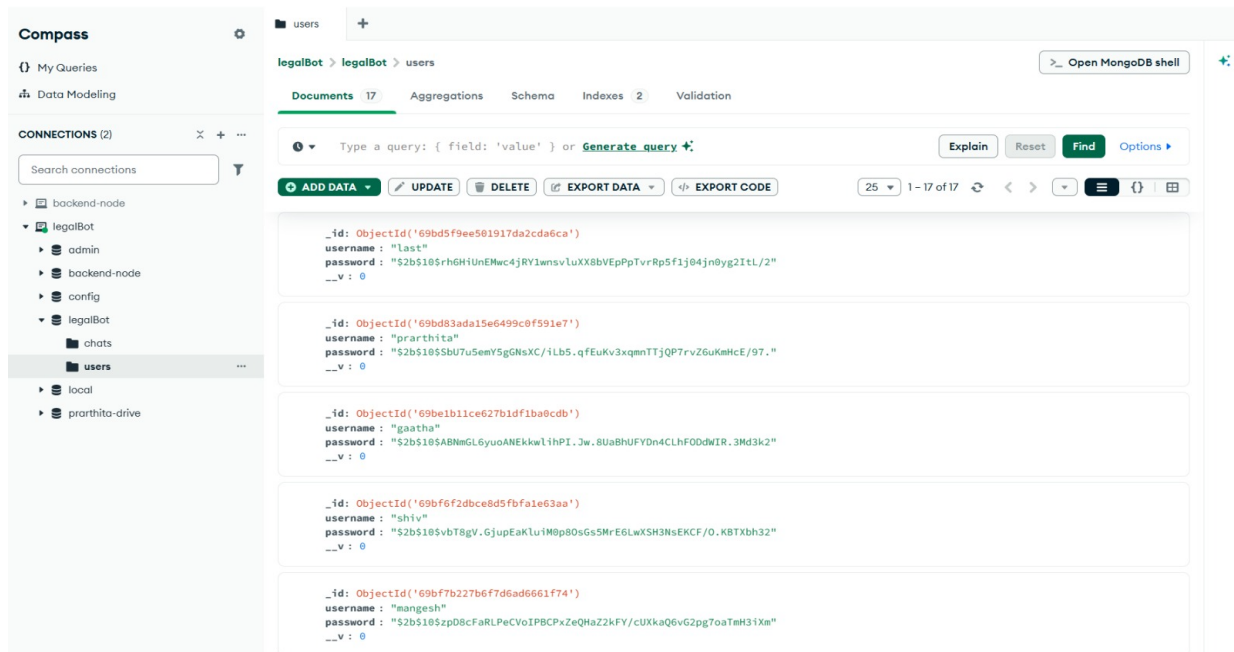


Fig. 7. MongoDB Database and Collections View

Description (Fig. 7):

The MongoDB Compass interface illustrates the database schema used in the LegalBot system. It includes collections for user authentication and chat history storage, enabling persistent data management and efficient query handling within the backend.

VI. COMPARATIVE ANALYSIS

Table I presents a comparative analysis of LegalBot against two related systems discussed in the literature: LawGPT 1.0 and Avyanna. The comparison evaluates each system across key dimensions including technology base, target users, authentication, chat storage, and additional features.

TABLE I
Comparative Analysis of LegalBot, LawGPT 1.0, and Avyanna

Feature	LegalBot (Our System)	LawGPT 1.0	Avyanna (India)
Base Technology	FastAPI + Claude AI API	GPT-3 Fine-tuned	Dialogflow NLP
Target Users	General public	Legal professionals	Women (India)
Authentication	Secure login system	Not specified	Firestore login
Chat History Storage	MongoDB (cloud)	Not specified	Not specified
Multilingual Support	Planned (future scope)	English only	English only
Emergency Services	Not included	Not included	Police, Ambulance, Fire
Mobile App	Planned (future scope)	Not available	Android app
Deployment	Cloud-ready	Not specified	Firebase



As shown in Table I, LegalBot occupies a distinct position among existing systems. Unlike LawGPT 1.0, which requires fine-tuning a large language model and is restricted to English, LegalBot leverages an API-based integration that is both easier to maintain and more adaptable to future model updates. Unlike Aavyanna, which is specifically designed for women-related legal issues in India via an Android application, LegalBot provides a general-purpose legal assistant accessible via a web browser to a broader user base. The inclusion of persistent chat history and a full-stack architecture distinguish LegalBot as a more complete end-to-end system for everyday users.

VII. FUTURE SCOPE

While the current implementation of LegalBot demonstrates a functional proof of concept, several enhancements are planned to extend its capabilities and reach:

- Voice Assistant Integration: Adding speech recognition and text-to-speech capabilities will make LegalBot accessible to users with limited literacy or disabilities.
- Multilingual Support: Extending the system to support regional Indian languages such as Hindi, Marathi, and Bengali will significantly broaden its user base.
- Improved AI Model: Fine-tuning or selecting a domain-specific legal language model will improve the accuracy and reliability of responses to specialized legal queries.
- Mobile Application: A dedicated Android and iOS application will improve accessibility and user engagement.
- Legal Document Upload and Analysis: Allowing users to upload legal documents (e.g., contracts, notices) for AI-assisted analysis will greatly enhance system utility.
- Cloud Deployment: Full deployment on a cloud platform such as AWS or Azure will ensure high availability and scalability.

VIII. CONCLUSION

This paper has presented LegalBot, an AI-based legal chatbot system designed to make legal information more accessible and understandable to the general public. The system integrates a FastAPI backend, a MongoDB cloud database, and an external AI language model API to deliver a responsive, full-stack conversational legal assistant. Users can register and authenticate securely, interact with the chatbot to receive real-time legal guidance, and revisit their conversation history at any time.

A comparative analysis with LawGPT 1.0 and Aavyanna confirms that LegalBot addresses key gaps in existing systems, particularly regarding general-purpose applicability, full-stack integration, and persistent data management. The Agile development methodology employed ensured continuous progress and early issue detection throughout the project lifecycle.

LegalBot represents a promising step toward democratizing access to legal information through AI technology. With the planned enhancements in multilingual support, voice interaction, and mobile deployment, the system has the potential to serve a diverse and underserved user population at scale.

ACKNOWLEDGMENT

The authors would like to thank Mr. Kashif Shaikh for his guidance and support throughout the development of LegalBot. This work was carried out as part of the academic curriculum at Thakur Shyamnarayan Engineering College, Mumbai, India.

REFERENCES

- [1]. R. Sil, A. Nag, S. Parveen, and B. K. Nath, "Intelligent Response System for Indian Law: An Approach to Women Security," *Journal of Network and Innovative Computing*, vol. 11, pp. 047–057, 2023.
- [2]. N. H. Thanh, "A Brief Report on LawGPT 1.0: A Virtual Legal Assistant Based on GPT-3," arXiv preprint arXiv:2302.05729, 2023.
- [3]. J. Browder, "DoNotPay: The World's First Robot Lawyer," 2016. [Online]. Available: <https://donotpay.com>
- [4]. E. Janoski-Haehlen and S. Starnes, "The Ghost in the Machine: Artificial Intelligence in Law Schools," *Duquesne Law Review*, vol. 58, no. 1, pp. 3–30, 2020.
- [5]. M. Queudot, E. Charton, and M.-J. Meurs, "Improving Access to Justice with Legal Chatbots," *Stats*, vol. 3, no. 3, pp. 356–375, 2020.



- [6]. A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems, pp. 5998–6008, 2017.
- [7]. K. Ramesh, S. Ravishankaran, A. Joshi, and K. Chandrasekaran, "A Survey of Design Techniques for Conversational Agents," Communications in Computer and Information Science, pp. 336–350, 2017.
- [8]. R. Dale, "Law and Word Order: NLP in Legal Tech," Natural Language Engineering, vol. 25, no. 1, pp. 211–217, 2019.