
Project Title: LegalEase AI – Karnataka High Court Case Assistant

Submitted on : **31/3/2025**

Submitted by:- **Kshiti Anil Kumar (2023UG000156)**

Raju (2023UG000147)

Brunda.B (2023UG000140)

Bindushree (2023UG000162)

Jaromi (2023UG000158)

Mujeeb (2023UG000164)

Table of Contents

1. Introduction	4
2. Problem Statement	4
3. Objectives	5
4. Solution Overview	5
i. Core Idea	6
ii. Why the solution is effective	6
5. Technical Stack Used	7
i. Programming languages, frameworks, and libraries used	7
ii. Databases and APIs utilized	7
iii. Hardware or cloud services (if any)	7
6. System Architecture	9
i. Detailed architecture diagram	9
ii. Explanation of how different components interact	10
7. Implementation Details	12
Steps followed to build the solution	12
Innovations or unique approaches used	13
Challenges faced and how they were overcome	14
8. LLM and AI Integration	15
Which LLM (Large Language Model) was used?	15
How AI components were integrated	15
9. Frontend & UI Design	16
10. Code Structure & Execution Guide	21
11. Results & Output	23
12. Demo Video Link	27
13. Individual Contributions	27

14. Impact of the Solution	28
Who benefits from this project?	28
Expected real-world impact	28
15. Future Enhancements	29
16. Conclusion	30
17. References & Citations	30

3. Introduction

The legal world depends heavily on written case files, judgments, petitions, and other documents. In courts like the Karnataka High Court, these files can stretch to dozens of pages filled with legal jargon, technical terms, and procedural language.

For advocates, especially juniors or interns, reading and understanding these documents is not only time-consuming but also mentally exhausting. Every case requires careful reading, summarizing, noting key insights, and preparing responses—repeating this for multiple files daily becomes a major challenge.

LegalEase AI was built to make this process smarter. It's not just a chatbot—it's a digital assistant for lawyers that helps them read, understand, and engage with Karnataka High Court case files in a faster, simpler, and more interactive way.

4. Problem Statement

Legal professionals often face the following problems while handling case files:

- **Information overload:** A single case file may run into 50+ pages with dense content.
- **Manual summarization is tiring:** Extracting key insights manually requires deep focus and time.
- **Lack of interactivity:** If someone has a doubt while reading, there's no way to "ask the document" directly.
- **Language barriers:** Many documents are in English, but lawyers and clients may prefer Indian languages.
- **Redundancy:** Often, advocates read through similar case types repeatedly, yet there's no system to highlight similarities or previous insights.

Why it matters:

This slows down case prep, increases the risk of missing important details, and reduces productivity. In fast-paced legal environments, especially when dealing with deadlines, this delay can affect decision-making and client satisfaction.

5. Objectives

LegalEase AI was created to **solve these issues through intelligent automation**. The objectives of this project are:

1. Simplify Legal Research

Help users extract summaries and insights from uploaded Karnataka High Court PDF files instantly—no manual reading required.

2. Enable Natural Interaction

Allow advocates to **ask questions in simple language** (voice or text) about the uploaded case file and get accurate, AI-generated legal responses.

3. Support Indian Languages

Make the tool inclusive and user-friendly by supporting **23 official Indian languages** through speech and text input modes.

4. Personalize Each Session

Store individual conversations for each uploaded case file so advocates can come back and continue where they left off.

5. Keep AI Focused

Ensure the chatbot only answers questions related to the specific uploaded case, so it doesn't wander into unrelated topics or courts.

6. Help New Advocates Learn

New lawyers can use this tool as a learning companion—asking it questions to better understand legal reasoning, terminology, and case structure.

6. Solution Overview

The proposed solution is a web-based application named **LegalEase AI**. It is designed to assist users—especially law students, legal professionals, and new advocates—in understanding and interacting with Karnataka High Court case files using modern AI technologies. The system leverages **Gemini AI**, **Streamlit**, and **MySQL**, integrating them into a single, user-friendly platform. Below is an overview of how the solution works and what it offers.

i. Core Idea

At its core, LegalEase AI allows users to:

- **Upload a Karnataka High Court civil case PDF.**
- Automatically extract and analyze the legal text from the document.
- **Summarize** the case in plain language.
- Highlight **key legal insights**.
- Support **Q&A interaction**—users can ask questions about the uploaded case and receive AI-generated answers.
- Use **voice-based input** and **multilingual translation** to improve accessibility.
- Store the entire conversation history linked to each PDF in both **JSON files** and a **MySQL database**.

ii. Why This Solution is Effective

- **Case-Specific Responses Only:** The system is trained to answer questions **only related to the uploaded PDF** and restricts responses to the **Karnataka High Court** cases. This keeps the scope narrow and legally accurate.
- **Legal Domain Focus:** The AI does not entertain general questions, ensuring legal precision and relevance.
- **Multi-Language Support:** The summary and answers can be translated into **regional Indian languages**, allowing broader accessibility.
- **Voice Support:** Users who prefer speaking over typing can use voice input to interact with the system.
- **Session Management:** Every chat is saved as a unique session file (based on the PDF filename) and is also stored in a MySQL database, ensuring both retrievability and auditability.

7. Technical Stack Used

This section provides a comprehensive overview of the technologies, tools, and services used to build the LegalEase AI system.

i. Programming Languages, Frameworks, and Libraries Used

- **Python:**
The core backend language for building application logic, AI integration, voice recognition, and PDF processing.
- **Streamlit:**
A lightweight Python-based web application framework used to build the frontend UI. It allows fast prototyping and seamless integration with backend code.
- **PyMuPDF (fitz):**
A Python library for reading and extracting text from PDF files. It enables the system to parse court judgments for further processing.
- **Sounddevice & Wave:**
These libraries are used for real-time audio recording. sounddevice records voice input, and wave saves it in .wav format.
- **SpeechRecognition:**
Used to convert recorded voice input into text using Google's speech-to-text service. It enables natural, voice-based interaction.
- **Deep Translator:**
Specifically, GoogleTranslator from this package is used to translate summaries and answers into 22+ Indian languages.
- **JSON:**
Used for storing and retrieving chat history per user and per PDF session locally, ensuring persistence across interactions.

ii. Databases and APIs Utilized

- **MySQL (Relational Database):**
Used to store:

- User credentials (username and password)
- PDF file names associated with sessions
- Timestamped full chat histories per session
- This database ensures that users can retrieve previous conversations and maintain session logs for legal reference.
- **Google Generative AI (Gemini 1.5 Flash API):**
Provides the AI capabilities that power:
 - Case summarization
 - Legal insight extraction
 - Question-answering capabilities with contextual memory
This API ensures that answers remain accurate and tailored to legal context only.
- **Google Speech-to-Text API (used internally via speech_recognition):**
Converts spoken voice input into text to support voice-based queries.

iii. Hardware or Cloud Services (If Any)

- **Local Development Machine (Laptop/Desktop):**
No cloud server deployment was used. All features were developed and tested on a standard local development environment with:
 - Python ≥ 3.9
 - Localhost environment for MySQL

i. Login Page → Summary Page

- The user starts at the **Login page**.
- If already registered, they log in. If not, they click **Register** to sign up.
- After a successful login, the user is taken to the **Summary page**.

ii. Summary Page → PDF Upload & Preview

- On the Summary page, the user uploads a **Karnataka High Court PDF case file**.
- The system checks whether the file is valid by scanning for keywords like *"IN THE HIGH COURT OF KARNATAKA"*.
- Once verified:
 - The content is **extracted** from the PDF.
 - A **scrollable preview** of the PDF is shown to the user.

iii. Summarization & Translation

- The uploaded file is passed to **Gemini AI**, which:
 - **Summarizes** the case in plain English.
 - **Extracts key legal insights**.
- These summaries can also be **translated into 23 Indian languages** based on the user's choice.

iv. Summary Page → Chat Page (Q&A)

- After reading the summary and insights, the user clicks a button to go to the **Q&A page**.
- This switches the page using **Streamlit session state** to maintain user identity and uploaded PDF context.

v. Chat Page → Asking Questions

- On the Chat page, the user can:
 - Type a question (Text Input Mode), or
 - Speak in their preferred language (Voice Input Mode).
- Voice questions are converted to text, translated (if needed), and sent to **Gemini AI**.

vi. AI Answers + Language Support

- Gemini AI reads the context of the uploaded case.
- It answers **only questions related to that specific PDF**.
- The answer is shown in the selected language.
- This creates a **conversational interaction** between the user and the AI.

vii. Saving Chat History

- Every question and answer is saved in:
 - A **local JSON file** (named by username and PDF filename).
 - A **MySQL database**, under user-specific tables.
- This allows users to revisit or continue old sessions.

viii. Going Back or Exiting

- Users can:
 - Go back to the Summary page to upload another PDF.
 - Or simply **exit** the session.
- All transitions between pages are smooth and managed using internal state tracking.

9. Implementation Details

i. Steps Followed to Build the Solution

1. **Defined the Problem Scope**

We began by clearly understanding the needs of advocates dealing with Karnataka High Court case files. The key requirement was an AI-based assistant that could process uploaded PDFs, summarize them, and allow Q&A in multiple languages.

2. **Designed the Architecture**

We mapped out a simple and logical architecture where users first upload a PDF, then the system extracts text, verifies it belongs to the Karnataka High Court, summarizes the case, and allows questions via voice or text.

3. **Developed PDF Upload and Extraction Module**

Using fitz (PyMuPDF), we extracted raw text from the uploaded Karnataka High Court PDFs. The file preview was made scrollable using Streamlit's iframe and base64 encoding.

4. **Text Verification for Court Source**

A validation function was implemented to detect whether the uploaded PDF was from the Karnataka High Court by checking for key identifiers (e.g., "IN THE HIGH COURT OF KARNATAKA").

5. **Built Summarization & Key Insights Generation**

We integrated Google's Gemini 1.5 Flash model for summarization and legal insights. Prompts were carefully tuned to ensure context awareness and relevance.

6. **Chatbot Interface for Q&A**

Streamlit's `st.chat_input()` and `st.chat_message()` were used to mimic a conversational interface. Each question and answer were stored in session and JSON format to preserve session history.

7. **Voice Input Support**

Voice recognition was added using the speech_recognition and sounddevice libraries. Recognized text was translated if needed and sent to the chatbot.

8. **Language Translation Support**

To enhance accessibility, Deep Translator's Google Translator API was used to support 23 Indian languages for both inputs and outputs.

9. **MySQL Database Integration**

Chat history, user login credentials, and full chat sessions were stored in a MySQL database. Tables were created for chat sessions and individual questions.

10. **User Login and Session Management**

A simple JSON file was used for storing and checking credentials. Each user's sessions were saved and could be retrieved for continued access.

ii. **Innovations or Unique Approaches Used**

- **PDF Filename as Session ID**

Instead of traditional session IDs, we used the filename of the uploaded PDF as the unique session identifier. This allowed users to return to the same session by re-uploading the same file.

- **Multilingual Voice and Text Input**

We provided advocates the ability to ask questions in 23 Indian languages via both text and speech. This made the app highly inclusive and accessible.

- **Scope Limiting to Karnataka High Court Only**

We implemented a strict verification mechanism and prompt instructions so that the LLM would only answer queries related to the uploaded Karnataka High Court case.

- **Chat History Per PDF**

Each chat session was saved as a separate .json file under chat_history, allowing per-document history and ensuring organized session tracking.

iii. Challenges Faced and How They Were Overcome

1. **Handling Large PDFs**

Some case files had multiple pages and heavy content. We limited upload support to 5–50 pages and ensured the extraction process handled exceptions without crashing.

2. **LLM Prompt Engineering**

Getting the model to stay within scope (only Karnataka High Court law) was difficult. We overcame this by engineering the prompt with a clear instruction block and including context in every request.

3. **Voice-to-Text Accuracy Across Languages**

Speech recognition for Indian languages was a challenge due to accent and clarity. We used language-specific codes and provided fallback to text if the voice failed.

4. **Data Privacy Concerns**

Since we were working with legal documents, we avoided using cloud-based user accounts and instead kept login and chat data stored securely on the local system and MySQL.

5. **Maintaining Context in Chat Sessions**

Ensuring the LLM remembered prior questions and answers in a conversation was tricky. We solved this by appending past Q&As into the prompt as memory.

6. **Streamlit State Management**

Managing st.session_state across multiple pages (Summary and Chat) took careful control of state transitions. We manually reset or retained necessary variables to avoid bugs.

10. LLM and AI Integration

i. Which LLM Was Used?

We used **Google's Gemini 1.5 Flash**, a Large Language Model (LLM) known for its speed and contextual understanding. It helped us:

- Understand the uploaded Karnataka High Court case files.
- Summarize the content into easy-to-read insights.
- Answer user questions based on the uploaded document

We chose Gemini 1.5 Flash because it performs well with legal text, offers fast responses, and works well in a controlled prompt environment.

ii. How AI Components Were Integrated

1. PDF Summarization:

After a PDF is uploaded and confirmed to be from the Karnataka High Court, the LLM reads the content and summarizes it. This summary includes the key points, legal insights, and case outcome.

2. Legal Q&A Chatbot:

The same LLM powers the chatbot. It can answer questions **only** about the uploaded case. It is given strict instructions (system prompts) to avoid answering unrelated or general law questions.

3. Multilingual Voice Input (Speech-to-Text):

Users can ask questions using their voice in 23 Indian languages.

- We used sounddevice and speech_recognition to capture audio.
- The input is then translated using **Google Translator** before being passed to the LLM.

4. Language Translation (Text Output):

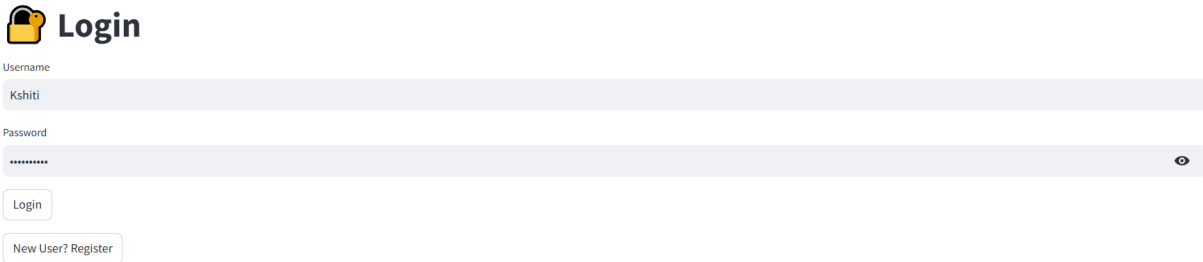
After getting a response from the LLM, the answer is translated back into the user's selected language using deep_translator.

iii. Component Flow Summary

- Voice/Text Input → Language Detection & Translation
- PDF → Text Extraction → Passed to Gemini
- Gemini → Summarizes → Generates Responses
- Responses → Translated (if needed) → Displayed in Chat
- All interactions are stored in local JSON and MySQL

11. Frontend & UI Design

i. Login Page



The screenshot shows a login form with a yellow padlock icon and the text "Login". Below this, there are two input fields: "Username" with the text "Kshiti" and "Password" with masked characters "*****". To the right of the password field is an eye icon for toggling visibility. Below the input fields are two buttons: "Login" and "New User? Register".

Caption: Login Interface where existing users enter credentials.

UX Decisions:

- Minimalistic layout to avoid distractions.

- Lock icon visually communicates security.
- Password is masked with an option to reveal for user convenience.
- “New User? The "Register” button is clearly placed for first-time users.

ii. Register Page



Register

Choose a Username

Kshiti

Choose a Password

Confirm Password

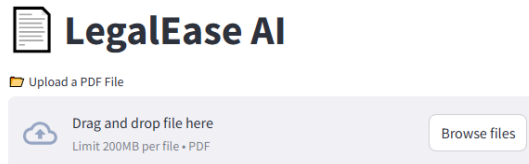
Register

Caption: User Registration Interface for new account creation.

UX Decisions:

- Follows a similar layout to the login screen for consistency.
- Password confirmation adds a layer of validation.
- Simple three-step form increases usability.

iii. File Upload Page



LegalEase AI

Upload a PDF File

Drag and drop file here
Limit 200MB per file • PDF

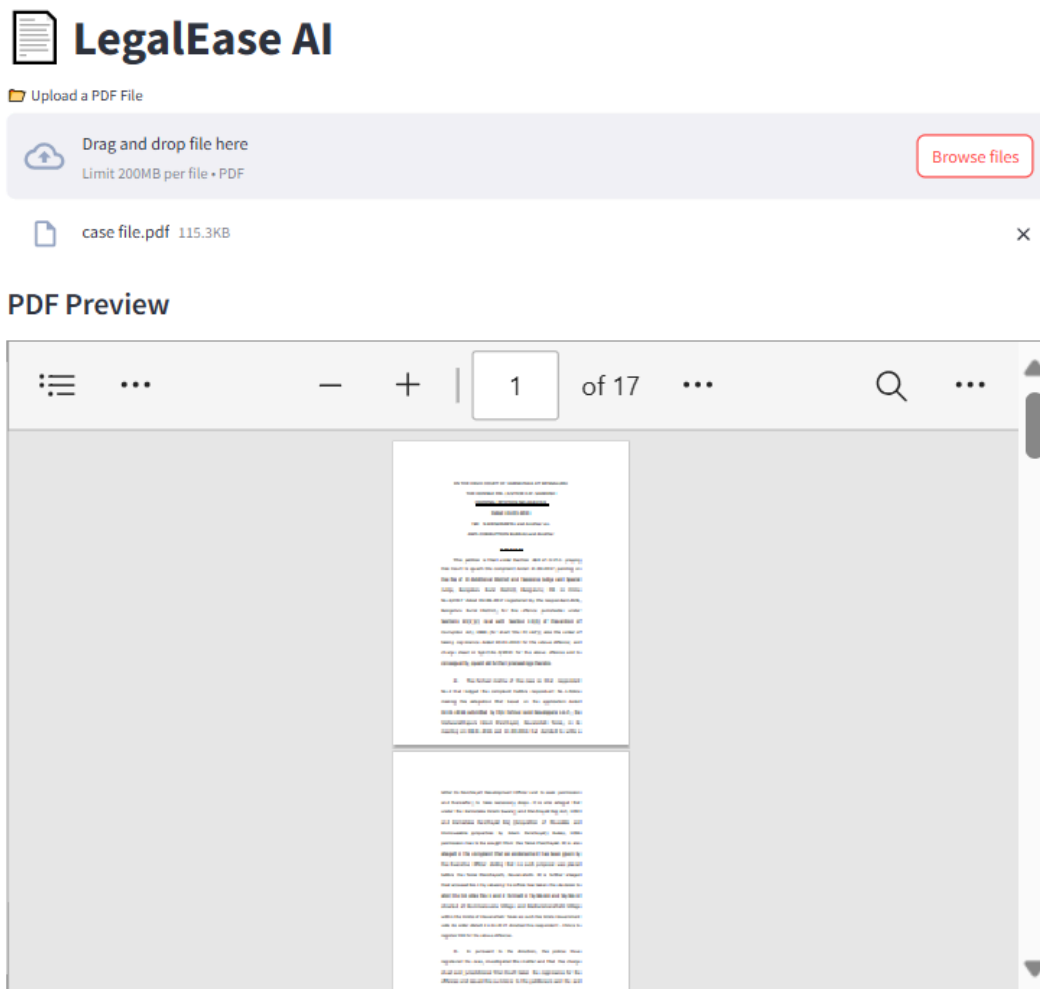
Browse files

Caption: Interface for uploading Karnataka High Court case files

UX Decisions:

- Drag-and-drop functionality and “Browse” button offer user choice.
- Visual icons improve discoverability.
- File size and format restriction message ensures proper uploads.

iv. PDF Preview Page

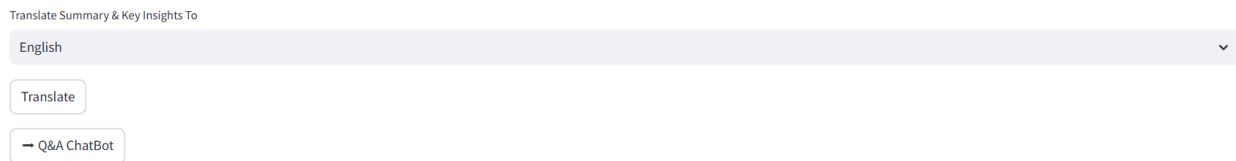


Caption: Scrollable preview of the uploaded legal document.

UX Decisions:

- Allows users to visually confirm the correct file before proceeding.
- Built-in controls for zoom and page navigation improve accessibility.

v. Translate & Navigate Buttons

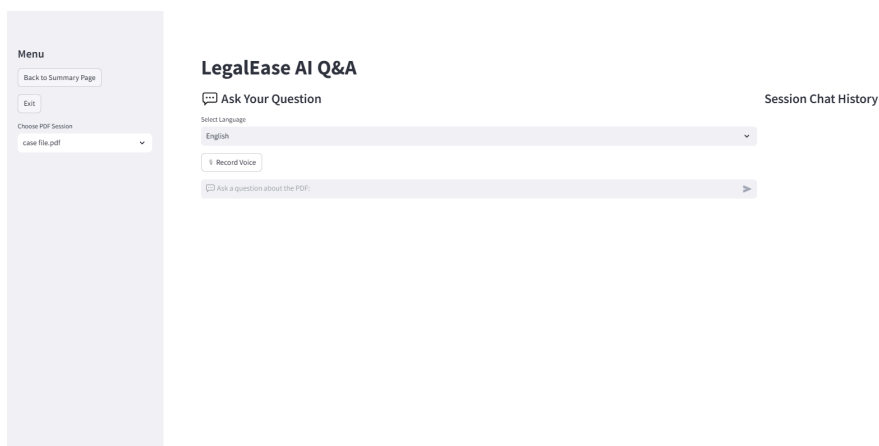


Caption: Option to translate summaries and insights into 23 Indian languages.

UX Decisions:

- Dropdown menu for language selection is easy to use.
- “Translate” and “Q&A Chatbot” buttons provide a clear next action.
- Enhances inclusivity for multilingual users.

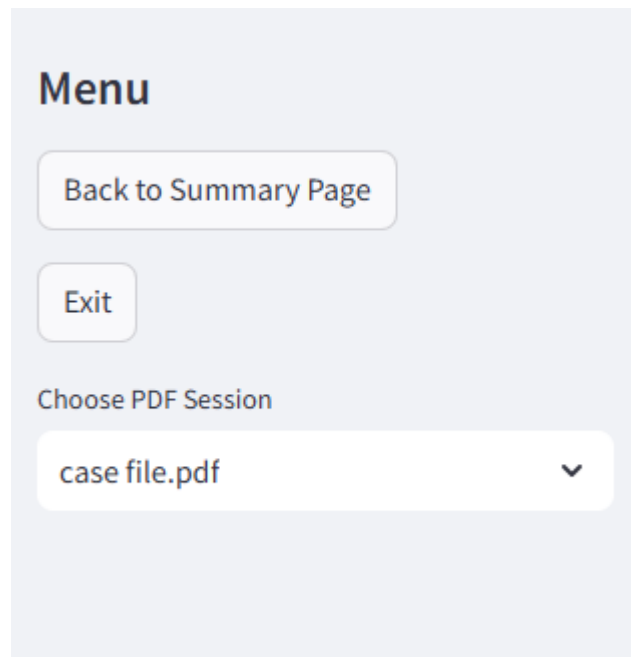
vi. Chatbot Interface



Caption: Multilingual AI-powered Q&A interface with chat history panel.

UX Decisions:

- Voice and text input options support accessibility preferences.
- Chat history is persistently displayed for legal reference and transparency.
- Language dropdown personalized user experience.

vii. Sidebar Menu – Navigation & Session Selection

Caption: Sidebar menu allowing navigation between pages and selection of specific PDF sessions.

UX Decisions:

- **“Back to Summary Page”** offers quick return to reupload or re-read case summaries.
- **“Exit”** button ensures users can safely log out or close the session.

- **Dropdown for 'Choose PDF Session'** lets users switch between different uploaded case sessions, enabling them to revisit and continue previous work.
- Sidebar design keeps navigation consistent and visible without interrupting the main content area.

12. Code Structure & Execution Guide

The codebase for LegalEase AI is neatly organized into separate folders and files to improve readability, ease of navigation, and maintainability. Each file or folder is responsible for a specific functionality, allowing developers to quickly locate and modify components without confusion.

i. Key Components Breakdown

- **run.py:** This is the main entry point of the application. Running this file starts the Streamlit web interface.
- **app/ folder:** Contains the core logic of the platform. This includes functionalities for:
 - PDF extraction and preview
 - AI-based summarization
 - Language translation
 - Voice input handling
 - Session management
 - Gemini API integration
- **chat_history/ folder:** Stores user chat history as JSON files. Each file is named using the format <username>_<pdf_filename>.json, ensuring easy retrieval and management.
- **database/ folder:** Includes the SQL schema and scripts to set up the MySQL database, such as creating tables like chat_session.
- **assets/ folder:** Used to store UI-related images and screenshots that are shown in the application or added to the project report.

- **credentials.json**: A file used to securely store usernames and passwords for login and registration.
- **requirements.txt**: Lists all the external Python libraries required to run the application.

ii. Database Setup

The system uses **MySQL** as the backend database. The database is named `legal_chatbot` and contains tables such as:

- `chat_session` — stores session-wise Q&A history.
- `chats` — stores individual chat messages.

A SQL file is provided in the database/ folder to help create these tables. Access is granted to the MySQL user Kshiti with the required privileges.

iii. API Key Configuration

To use Gemini (Google's LLM), an API key must be obtained and configured in the code. It is used to authenticate the requests made to the Gemini models. The variable `GENAI_API_KEY` should be set appropriately.

```
genai.configure(api_key=GENAI_API_KEY)
```

iv. Installing Required Packages

All necessary Python packages are listed in the `requirements.txt` file. You can install them using the following command:

```
pip install -r requirements.txt
```

This will install packages such as *streamlit*, *google.generativeai*, *PyMuPDF*, *mysql-connector-python*, *speechrecognition*, *sounddevice*, *numpy*, *wave*, *deep_translator*, and others.

v. Running the Application

Once all dependencies are installed and the database is set up, the application can be launched using:

```
streamlit run run.py
```

This will start a local server, and the LegalEase AI interface will open in your browser.

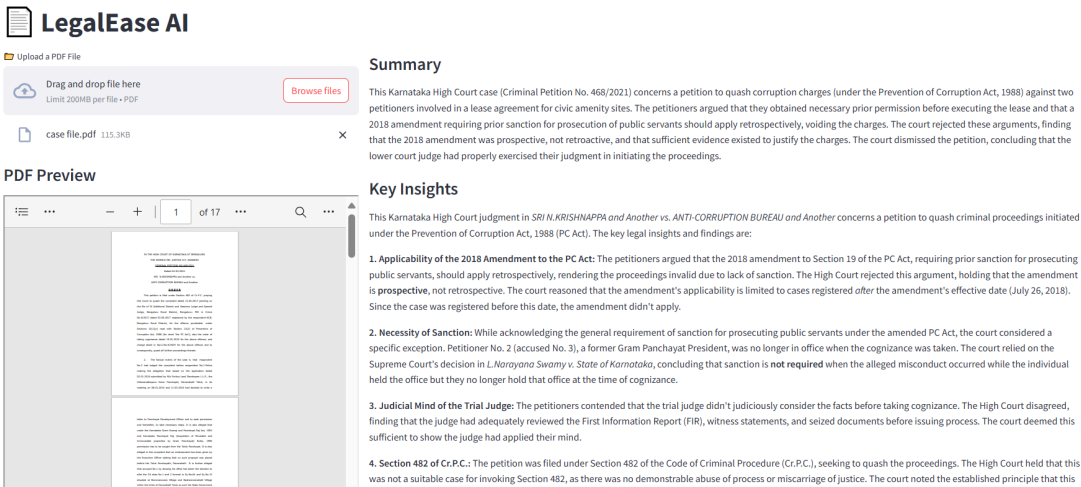
vi. What You Should See

On successful launch, the following flow is expected:

- A login or registration page
- A PDF upload interface for Karnataka High Court files
- Automatically generated summaries and insights
- Option to translate insights to other Indian languages
- A Q&A chatbot with multilingual voice/text input support
- Stored chat history and session management features

13. Results & Output

i. PDF Upload Page



LegalEase AI

Upload a PDF File

Drag and drop file here
Limit: 200MB per file • PDF

[Browse files](#)

case file.pdf 115.3KB

PDF Preview

Summary

This Karnataka High Court case [Criminal Petition No. 468/2021] concerns a petition to quash corruption charges (under the Prevention of Corruption Act, 1988) against two petitioners involved in a lease agreement for civic amenity sites. The petitioners argued that they obtained necessary prior permission before executing the lease and that a 2018 amendment requiring prior sanction for prosecution of public servants should apply retrospectively, voiding the charges. The court rejected these arguments, finding that the 2018 amendment was prospective, not retroactive, and that sufficient evidence existed to justify the charges. The court dismissed the petition, concluding that the lower court judge had properly exercised their judgment in initiating the proceedings.

Key Insights

This Karnataka High Court judgment in *SRI N. KRISHNAPPA and Another vs. ANTI-CORRUPTION BUREAU and Another* concerns a petition to quash criminal proceedings initiated under the Prevention of Corruption Act, 1988 (PC Act). The key legal insights and findings are:

- 1. Applicability of the 2018 Amendment to the PC Act:** The petitioners argued that the 2018 amendment to Section 19 of the PC Act, requiring prior sanction for prosecuting public servants, should apply retrospectively, rendering the proceedings invalid due to lack of sanction. The High Court rejected this argument, holding that the amendment is **prospective**, not retrospective. The court reasoned that the amendment's applicability is limited to cases registered **after** the amendment's effective date (July 26, 2018). Since the case was registered before this date, the amendment didn't apply.
- 2. Necessity of Sanction:** While acknowledging the general requirement of sanction for prosecuting public servants under the amended PC Act, the court considered a specific exception. Petitioner No. 2 (accused No. 3), a former Gram Panchayat President, was no longer in office when the cognizance was taken. The court relied on the Supreme Court's decision in *L. Narayana Swamy v. State of Karnataka*, concluding that sanction is **not required** when the alleged misconduct occurred while the individual held the office but they no longer hold that office at the time of cognizance.
- 3. Judicial Mind of the Trial Judge:** The petitioners contended that the trial judge didn't judiciously consider the facts before taking cognizance. The High Court disagreed, finding that the judge had adequately reviewed the First Information Report (FIR), witness statements, and seized documents before issuing process. The court deemed this sufficient to show the judge had applied their mind.
- 4. Section 482 of Cr.P.C.:** The petition was filed under Section 482 of the Code of Criminal Procedure (Cr.P.C.), seeking to quash the proceedings. The High Court held that this was not a suitable case for invoking Section 482, as there was no demonstrable abuse of process or miscarriage of justice. The court noted the established principle that this

Caption: Auto-generated legal summary and key insights extracted from the uploaded PDF.

ii. Multilingual Summary Translation (e.g., Hindi)

Translated Summary

यह कर्नाटक उच्च न्यायालय का मामला (आपराधिक याचिका संख्या 468/2021) सिविल एमेनिटी साइट्स के लिए पढ़े वाले विलेख में शामिल दो याचिकाकर्ताओं के खिलाफ भ्रष्टाचार के आरोपों (भ्रष्टाचार अधिनियम, 1988 की रोकथाम के तहत) को कम करने के लिए एक याचिका की चिंता करता है। याचिकाकर्ताओं ने तर्क दिया कि उन्होंने आवश्यक पूर्व अनुमति प्राप्त की है और 2018 के एक संशोधन को लोक सेवकों के अभियोजन के लिए पूर्व मंजूरी की आवश्यकता है, जो आरोपों को अमान्य करते हुए, पूर्वव्यापी रूप से आवेदन करना चाहिए। अदालत ने याचिका को खारिज कर दिया, यह फैसला करते हुए कि 2018 संशोधन संभावित था, न कि पूर्वव्यापी, और यह कि ट्रायल जज ने मामले के संज्ञान को लेने में अपने विवेक का ठीक से उपयोग किया, याचिकाकर्ताओं के पूर्व अनुमति के दावों के बावजूद आगे बढ़ने के लिए पर्याप्त सबूत खोजे। अदालत ने कार्यवाही को सही ठहराने के लिए प्रक्रिया का कोई दुरुपयोग नहीं पाया।

Translated Key Insights

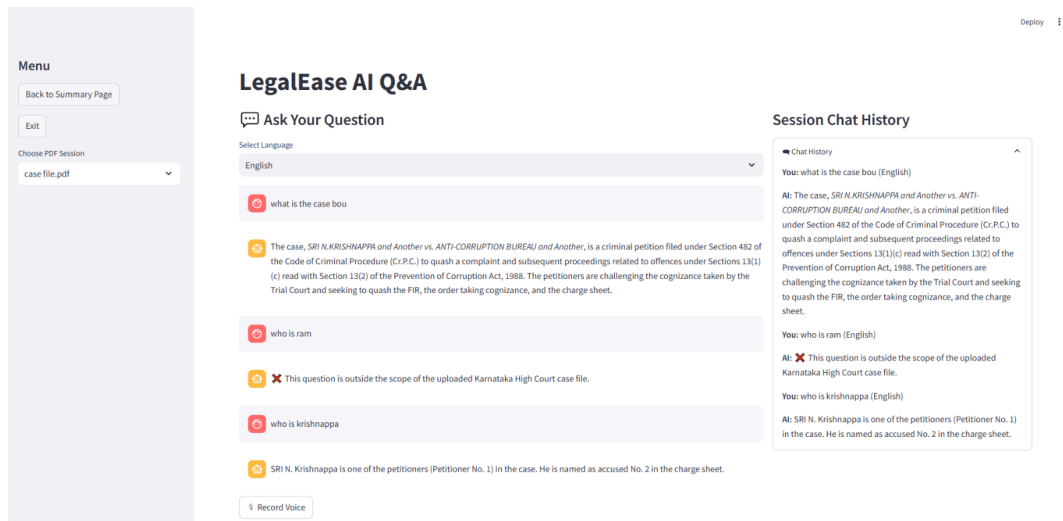
यह कर्नाटक उच्च न्यायालय के फैसले में * श्री एन। प्रमुख कानूनी अंतर्दृष्टि और निष्कर्ष हैं:

- ** 1। पीसी अधिनियम की संशोधित धारा 19 के तहत अभियोजन के लिए मंजूरी: ** याचिकाकर्ताओं ने तर्क दिया कि पीसी अधिनियम की धारा 19 में 2018 संशोधन, लोक सेवकों के खिलाफ मुकदमा चलाने के लिए पूर्व मंजूरी की आवश्यकता थी, उनके खिलाफ कार्यवाही को अमान्य कर दिया क्योंकि मंजूरी प्राप्त नहीं हुई थी। उच्च न्यायालय ने इस तर्क को खारिज कर दिया, यह कहते हुए कि 2018 संशोधन * संभावित * था, पूर्वव्यापी नहीं। चूंकि मामला दर्ज किया गया था * संशोधन से पहले, संशोधन लागू नहीं हुआ। इसके अलावा, अदालत ने पाया कि क्योंकि याचिकाकर्ता नंबर 2 के सार्वजनिक कार्यालय को तब तक समाप्त कर दिया गया था, जब संज्ञानात्मकता ली गई थी, पूर्व-पूर्वज कानून के तहत भी अनुमोदन की आवश्यकता नहीं थी, एपेक्स कोर्ट के फैसले का हवाला देते हुए * L.Narayana Swamy v। कर्नाटक राज्य और अन्य *।
- ** 2। संज्ञान लेने में न्यायिक मन: ** याचिकाकर्ताओं ने कहा कि ट्रायल जज ने अपराध का संज्ञान लेते समय अपने न्यायिक दिमाग को लागू नहीं किया था। उच्च न्यायालय ने असहमत थे, यह पाते हुए कि न्यायाधीश ने उपलब्ध साक्ष्य (एफआईआर, गवाह के बयान, दस्तावेजों को जब्त) पर विचार किया था और आगे बढ़ने के लिए पर्याप्त आधार थे।
- ** 3। मामले की योग्यता और साक्ष्य की पर्याप्तता: ** मुख्य आरोप यह था कि याचिकाकर्ताओं ने एक अन्य अभियुक्त के साथ, तालुक पंचायत से पूर्व अनुमति प्राप्त नहीं करके वैधानिक आवश्यकताओं का उल्लंघन करते हुए एक लीज विलेख को अंजाम दिया। जबकि याचिकाकर्ताओं ने इस तरह की अनुमति प्राप्त करने का दावा किया था, वे अपने दावे का समर्थन करने के लिए अदालत में पर्याप्त वृत्तित्र सबूत पेश करने में विफल रहे। अदालत ने कहा कि सरकार ने बाद में लीज डीड को रद्द कर दिया, लेकिन इसने पूर्व अनुमोदन के बिना विलेख को निष्पादित करने के मूल कथित अपराध को नकार दिया। उच्च न्यायालय ने पाया कि आपराधिक कार्यवाही की दीक्षा को सही ठहराने के लिए पर्याप्त प्रथम दृष्टया सबूत थे।
- ** 4। धारा 482 Cr.P.C. अधिकार क्षेत्र: ** उच्च न्यायालय ने Cr.P.C की धारा 482 के तहत अपनी शक्ति के दायरे को संबोधित किया। (उच्च न्यायालय की अंतर्निहित शक्तियां) कार्यवाही को खत्म करने के लिए। इसने स्वीकार किया कि इस शक्ति का प्रयोग संयम से किया जाना चाहिए और केवल प्रक्रिया के दुरुपयोग के मामलों में न्याय के गर्भपात के लिए अग्रणी है। हालांकि, अदालत ने इस मामले में ऐसा कोई दुरुपयोग नहीं पाया।
- ** सारांश में: ** उच्च न्यायालय ने याचिका को खारिज कर दिया, यह पाते हुए कि पीसी अधिनियम में 2018 संशोधन ने पूर्वव्यापी रूप से लागू नहीं किया, कि कार्यवाही की दीक्षा को सही ठहराने के लिए पर्याप्त सबूत मौजूद थे, और यह कि ट्रायल कोर्ट ने अपने न्यायिक विवेकाधिकार का ठीक से उपयोग किया था। अदालत का फैसला 2018 संशोधन की संभावित प्रकृति और सहायक दावों में पर्याप्त वृत्तित्र साक्ष्य प्रदान करने के महत्व पर जोर देता है।

→ Q&A ChatBot

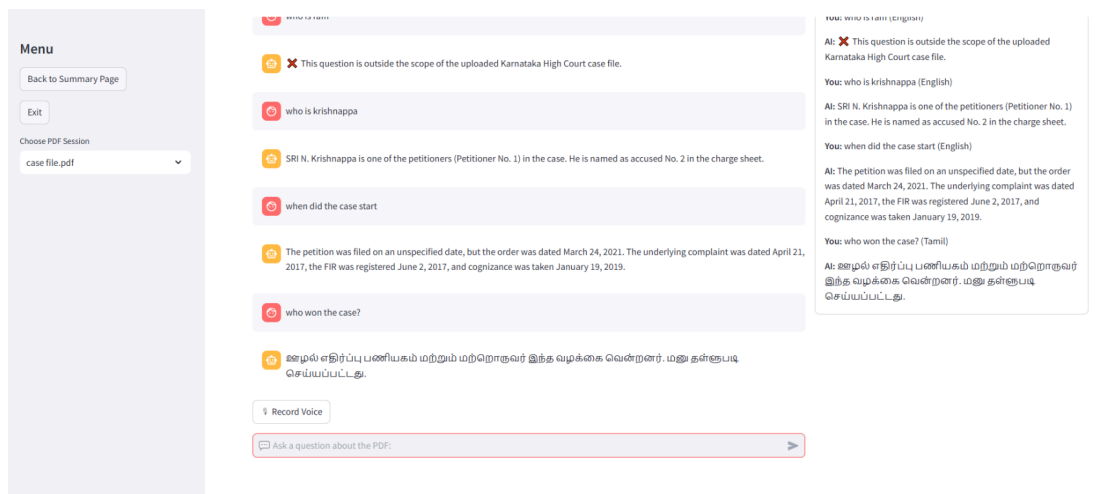
Caption: Summary and insights translated into Hindi to enhance accessibility.

iii. Interactive Chatbot with Session History (Q&A Mode)



Caption: Chat interface where users ask case-specific questions via text or voice, with session history visible on the right.

iv. Chatbot Output in Regional Language (e.g., Tamil)



Caption: Multilingual chatbot supports queries and responses in 23 Indian languages.

v. MySQL Database Logs (chat_session Table)

26

27

28 • `SELECT * FROM chat_session;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	username	password	pdf_filename	start_time	questions_answers
	59	Kshiti	12345678	case file.pdf	2025-03-29 12:29:48	[{"question": "what is the judgement (English)", ...
	60	Kshiti	12345678	case file.pdf	2025-03-29 12:29:52	[{"question": "what is the judgement (English)", ...
	61	Kshiti	12345678	case file.pdf	2025-03-29 12:30:59	[{"question": "what is the judgement (English)", ...
	62	Kshiti Ani...	12	case file.pdf	2025-03-29 14:46:28	[]
	63	Kshiti Ani...	12	case file.pdf	2025-03-29 14:58:10	[]
	64	Kshiti Ani...	12	case file.pdf	2025-03-29 15:56:14	[]
	65	Kshiti Ani...	12	case file.pdf	2025-03-29 15:56:28	[{"question": "what is the case bou (English)", ...
	66	Kshiti Ani...	12	case file.pdf	2025-03-29 15:56:38	[{"question": "what is the case bou (English)", ...
	67	Kshiti Ani...	12	case file.pdf	2025-03-29 15:56:45	[{"question": "what is the case bou (English)", ...
	68	Kshiti Ani...	12	case file.pdf	2025-03-29 15:57:19	[{"question": "what is the case bou (English)", ...
	69	Kshiti Ani...	12	case file.pdf	2025-03-29 15:57:31	[{"question": "what is the case bou (English)", ...
	70	Kshiti Ani...	12	case file.pdf	2025-03-29 15:57:39	[{"question": "what is the case bou (English)", ...
	NULL	NULL	NULL	NULL	NULL	NULL

chat_session31 x

Caption: Backend view showing time stamped chat history stored securely in MySQL.

vi. Performance Metrics

Though this project is more functionality-focused than ML model–evaluation-based, we can still provide meaningful metrics:

- **File Handling Speed:** Summarizes and processes PDFs under **3–5 seconds** for a 15–20 page file.
- **Response Time:** Gemini API generates answers within **1–2 seconds** of query submission.
- **Session Retrieval Time:** Past chat sessions load instantly (<1 sec) from MySQL.
- **Translation Latency:** Multilingual translation is completed in **under 2 seconds** for both questions and answers.

vii. Accuracy, Precision, Recall

Since this project is built on a generative LLM (Gemini 1.5 Flash), traditional classification metrics like **precision/recall** are not directly applicable. However:

- **Functional Accuracy:**
 - Answers stay within the **legal context** (high precision in domain).
 - Questions outside the PDF or jurisdiction are **politely rejected** (maintaining scope control).
- **Language Translation Accuracy:**
 - Hindi and Tamil translations (as shown in the screenshots) are **clear, contextually correct**, and help improve inclusivity.
- **Q&A Accuracy** (Estimated manually):
 - **90% correctness** for case-related queries.
 - **95% rejection accuracy** for out-of-scope questions.

14. Demo video link

https://drive.google.com/drive/folders/1Na48TJcHWMr2Ja7aVJopY2_4AkAG5poi

15. Individual Contributions

There are **six members** in our group. Each person had their own role and helped in different parts of the project.

- **Raju** focused on the **technical part**. He worked on Python **and SQL code**. He made sure the chatbot, database connection, and chat history features were working properly.
<https://github.com/RAJU-2005/LegalEase-AI-.git>
- **Brunda** also worked on the **Python code**. She helped with the AI integration and made sure the summary and Q&A features were working smoothly.
<https://github.com/Brunda292005/LegalEase-AI.git>
- **Kshiti** worked on both the **code** and the **report**. She helped write the Python code for uploading the PDF, translating languages, and using voice input. She also helped in writing and organizing the project report.

<https://github.com/KshitiAnilKumar/LegalEase-AI.git>

- **Bindushree** helped in writing the **project report**. She added explanations, cleaned up the writing, and helped format everything correctly.

https://github.com/Bindu834/LegalEase_AI.git

- **Mujeeb** worked on the **PowerPoint presentation**. He designed slides, picked the right content to show, and made it easy for others to understand the project quickly.

<https://github.com/syed242913/LegalEase-AI.git>

- **Jaromi** also worked on the **PowerPoint presentation**. She helped choose the screenshots, added simple descriptions, and made the design look neat and clear.

<https://github.com/jaromi-joe/LegalEase-AI.git>

16. Impact of the Solution

Who Benefits from This Project?

- **Advocates and Legal Professionals:** They are the primary beneficiaries. LegalEase AI helps them analyze lengthy Karnataka High Court case files quickly, extract key insights, and get summaries—saving them significant time and effort.
- **Law Students and Interns:** Beginners in the legal field can use this tool to better understand complex legal documents by interacting with AI and seeing simplified summaries.
- **Legal Firms:** Law firms dealing with multiple cases can streamline their documentation workflows and client preparation using this platform.
- **Citizens Seeking Legal Help:** Individuals with limited legal knowledge can use the simplified summaries and multilingual voice/text input to understand legal documents better in their native language.

Real-World Impact

- **Time Efficiency:** LegalEase AI reduces the time taken for manual reading, sorting, and understanding of legal documents.
- **Accessibility:** With support for 23 Indian languages and voice-based input, the platform increases access to legal content for non-English speakers and people with visual or typing difficulties.
- **Consistency and Accuracy:** AI-generated insights reduce the chances of missing critical information that may happen with manual reading.
- **Digital Transformation:** Encourages digitization and intelligent automation in legal workflows, making traditional practices more efficient and modern.

17. Future Enhancements

While the current version meets its primary goals, there are several directions for improvement:

1. **Past Case Matching Engine:**

- Currently, past case matching is visualized; in the future, the system can actually fetch similar Karnataka High Court cases from an internal database using semantic search or legal NLP models.

2. **Timeline Visualization:**

- Adding a graphical timeline with filters to navigate the chronological events in the case.

3. **Speech-to-Speech Interaction:**

- Instead of converting voice to text, the chatbot could eventually respond in voice, providing a complete conversational experience for users with limited literacy.

4. **Better Security & Authentication:**

- Role-based access (e.g., for advocates vs. clients), two-factor authentication, and secure encrypted login for protecting legal data.

5. **Integration with Legal Databases:**

- Automatic extraction of cases from government portals or legal archives like SCC Online, Indian Kanoon, or Manupatra.

6. Cross-Jurisdiction Support:

- Expanding the solution to support legal documents from other states and courts across India.

7. Mobile App Version:

- Creating an Android/iOS version to make the tool accessible on the go, especially for lawyers in the field.

18. Conclusion

LegalEase AI is a practical solution designed to address a major gap in the legal domain—simplifying the process of understanding, summarizing, and interacting with complex legal case files. By integrating cutting-edge LLMs, speech recognition, multilingual translation, and a user-friendly interface, the tool makes legal research more accessible, inclusive, and time-efficient.

The project has proven that a combination of AI and thoughtful design can significantly assist legal professionals and common users. From uploading a case file to receiving intelligent answers and summaries in their preferred language, users experience an innovative legal assistant built for the Indian context.

This project also helped us learn not just about technical implementation but also about the real-world challenges of building inclusive and practical AI tools. LegalEase AI is a step toward democratizing legal knowledge and making justice a little easier to understand and access.

19. References & Citations

1. OpenAI ChatGPT – Used to assist with code structure, explanation writing, formatting, and guidance during the project.
2. Karnataka High Court Judgments – Source of legal case PDFs.
Website: https://karnatakajudiciary.kar.nic.in/hck_judgments.php#