Smart Legal Form Builder

- AI-Enhanced Document Creation

Seoyeon Kim
College of Social Science Hanyang University
Dept.of Tourism
Seoul, Korea
soen0814@hanyang.ac.kr

Taeyeon Kim
College of Engineering Hanyang University
Dept.of Information Systems
Seoul, Korea
tangboll5075@gmail.com

Suhyeon Yu
College of Engineering Hanyang University
Dept.of Information Systems
Seoul, Korea
sh265826@naver.com

Soohan Jeong
College of Engineering Hanyang University
Dept.of Information Systems
Seoul, Korea
soohan9909@gmail.com

Jiwon Han
College of Business Hanyang University
Dept.of Business
Seoul, Korea
jiwon8623@hanyang.ac.kr

***Abstract*— The Smart Legal Form Builder leverages advanced AI provides users with analyzed results using advanced AI technologies including Retrieval-Augmented Generation (RAG) and the GPT-4 model. It helps users resolve legal issues efficiently by addressing barriers such as limited legal knowledge, complex terminology, and high costs associated with traditional legal services. This paper describes the methodology and technologies used to build the platform, including dataset preparation, model fine-tuning, and system implementation. The dataset curated from multiple legal sources ensures that the AI model generates accurate and contextually relevant responses to legal cases such as sexual harassment, online defamation, second-hand fraud, and forcible sexual assault. The RAG framework combines retrieved legal precedents with AI-generated responses to enhance the reliability of the model, enabling it to predict settlement amounts, sentencing, and actionable legal advice.**

The application's functionality is not limited to document creation. Features such as real-time input validation, seamless integration of user interaction and AI-based automation ensure a smooth and accessible experience. Built using Flutter, the platform offers cross-platform compatibility, allowing Android users to access a user-friendly interface. The generated documents are processed using Syncfusion Flutter PDF, allowing users to preview, adjust, download and submit professional-quality PDFs directly from the app. It also features a user-friendly main menu and completion confirmation system, enhancing the overall user experience and driving continued engagement.

Smart Legal Form Filler demonstrates the transformative potential of AI in legal technology, reducing reliance on expensive professional help and making legal services more accessible. Future improvements include expanding the scope of cases, enhancing multilingual support, and seamless filing of complaints through integration with the official legal system. Through continuous innovation, the platform aims to democratize access to justice and bridge the gap between individuals and the legal system.

Keywords - Legal technology, AI-based legal tools, Search-based generation (RAG), litigation automation, legal document generation, GPT-4, cross-platform applications, sexual harassment, online defamation, online transaction fraud, sexual assault, access to justice, cost-effective legal solutions, AI legal assistance, multilingual legal assistance, legal case law search

I. INTRODUCTION

1. Motive

For many, the law is difficult and inaccessible. Everyone has had to file a complaint at some point in their lives, but for individuals who lack legal expertise or resources, even this basic task can be difficult. This problem disproportionately affects marginalized communities, amplifying social inequality and widening the gap between those who have access to professional legal assistance and those who do not.

However, advances in AI and automation are helping to solve this problem through the development of apps. Smart Legal Form Filler leverages these technologies to streamline the legal process and help individuals write legal documents professionally and independently. The app specializes in filing complaints for four major categories: sexual harassment, online defamation, second-hand trading fraud, and forcible sexual assault. This not only addresses the immediate barriers, but also makes filing more inclusive and accessible.

2. Problem Statement

A. Lack of public understanding of legal terms and documents.

According to the 2019 National Legal Awareness Survey, 76.3% of respondents reported difficulty understanding legal terms, and 78.4% reported difficulty understanding legal documents. This lack of understanding is a major reason why individuals have difficulty participating in the legal system. Misunderstanding of legal terms leads to errors in writing documents, creating additional obstacles to achieving legal resolution.

B. Lack of interest and familiarity with legal information

According to the 2021 National Legal Awareness Survey, 79.9% of respondents reported that they had never searched for legal information online. In addition, only 23.4% of

respondents discussed legal topics with acquaintances to better understand the law, indicating a lack of interest in legal information. This suggests that although legal tools exist, most individuals do not know how to use them effectively.

C. Costs and Barriers to Access to Legal Professionals

Legal services are often overpriced, especially for tasks such as filing a complaint. Many individuals face time and financial burdens when seeking professional help. Low-income individuals cannot even afford legal consultations. Despite the availability of legal knowledge online, many still rely on expensive legal professionals due to the complexity of the complaint.

These statistics and challenges highlight the need to streamline the legal document writing process and provide tools that enable individuals with limited legal knowledge to write their own legal documents. They also highlight the urgent need to bridge the gap between professional services and everyday users. Smart Legal Form Writer aims to address these systemic issues by automating the writing of legal documents while ensuring accuracy, accessibility, and compliance with legal standards, while significantly reducing costs and efforts.

3. Vision and Goals

The core of Smart Legal Form Maker is to provide an accessible, intuitive and reliable solution. The app reduces the burden of filing a complaint and allows users to create the necessary legal documents in an instant. The goal is to make it easy for anyone, not just professionals, to write a complaint.

The platform eliminates the reliance on legal professionals and allows users to independently draft complaints. It automates complex processes and provides step-by-step questions and guidance. The platform aims to simplify the complaint drafting process for four main categories: By providing templates and guidance for these cases, the smart legal form builder helps users complete their complaints without the hassle of complex legal procedures. In doing so, the app reduces the burden of legal jargon, simplifies procedural requirements, and allows users to fully express their claims and demands.

At the end of the process, the smart legal form filler imagines a fairer legal environment.

- Users save significant amounts of time and money on legal documents.
- The legal system becomes more accessible to marginalized groups.
- Users experience the power and confidence to independently address their legal needs.
- It evolves into an extensible framework that can be expanded into other complaint areas.

The success of this feature will be measured not only by the efficiency and accuracy of the complaints generated, but also by the transformative impact it has in reducing the gap in access to justice and creating a more just society.

II. DATASETS

This section describes the structure and generation method of the dataset used to train the Retrieval-Based Generation (RAG) system. The dataset contains legal cases in four categories: sexual harassment, online defamation, online transaction fraud, and sexual assault. Each case contains key information such as the incident description, legal agreement, sentence, and summary of the verdict.

1. Dataset Structure:

The dataset used to train the RAG system is structured in JSON format, with each case record containing detailed information about a legal case. Each record contains the following fields:

- id: Unique identifier for each event
- category: The legal category of the incident (e.g., "sexual harassment")
- text: A brief summary or key points of the legal case.
- settlement: The amount of monetary settlement paid in the case (if any)
- sentence: a sentence imposed by a court, usually in months
- verdict_summary: A brief summary of the court's decision.
- keywords: A list of keywords that describe important aspects of the incident (e.g., crime type, location, and other relevant details).

Each incident in the dataset contains the information necessary for the system to identify patterns, find similar incidents, and generate legal insights in response to user queries.

2. How to create a dataset:

The dataset is generated through the following steps:

1) Data Collection: Legal cases are collected from public legal sources such as court records and news articles and are categorized into four main types: sexual harassment, online defamation, second-hand transaction fraud, and forcible assault.

2) Preprocessing: Raw data is cleaned and formatted to ensure a consistent and concise description of each event. Data is tokenized and indexed for easy retrieval by the RAG system.

3) Annotation: Each case is annotated with relevant information such as category, settlement amount, sentence, and summary of judgment. Keywords are added to increase searchability.

4) Classification: The dataset is divided into four categories based on the type of legal issue to enable efficient querying and retrieval of similar cases.

5) Data Storage and Indexing: Datasets are stored in JSON format and indexed using FAISS for fast retrieval by the RAG system.

3. Leveraging the RAG dataset

1) Dataset Overview: Legal/Regulatory Text Analysis Data

- Source: AI Hub Dataset 71723
- Explanation:
  - This dataset contains over 60,000 annotated event data for AI training, labeled with important features such as event summaries, Q&A sets, and term keywords.
  - It represents the actual distribution of legal case types, ensuring balanced representation across categories, and includes at least 2,000 cases per category.
- Purpose:
  - Designed to support academic and industrial research in the fields of AI and legal natural language processing.
  - It helps improve the performance of legal case summarization, prediction, and related natural language understanding tasks.
  -

2) Dataset Overview: Legal/Regulatory Judgment and Contract Analysis Data

- Source: AI Hub Dataset 580
- Explanation:
  - Contains processed data including facts, arguments and analysis of favorable/unfavorable contractual clauses extracted from over 10,000 legal decisions.
  - Labelling and tagging of illegal or beneficial provisions are included for consumer impact assessment.
- Purpose:
  - Facilitates accessibility to legal decisions and contracts for non-specialists.
  - Tailored to improve AI performance in understanding legal text structure and identifying important elements.

4 Filter and customize your dataset

- Initial Approach: Initially, the system was trained using all civil and criminal law data. However, the lack of examples related to the selected categories led to poor performance of the RAG system.
- Optimization: We created a dataset of about 1,000 legal cases categorized into categories. This ensured better relevance and improved the performance of the RAG system, especially in categories such as online abuse and fraud.

III. METHODOLOGY

This section describes the methodology for building a GPT-4-based retrieval-based generation (RAG) system to answer legal questions. The system uses AI (GPT-4) and document retrieval (via LangChain and FAISS) to provide accurate answers based on real legal documents. The solution is implemented as a web API using Flask. Below, we provide a detailed description of the main components involved in building and deploying the system.

1. Models and Preferences:

The legal question-answering system is built around Flask and uses OpenAI's GPT-4 model to generate answers. The system incorporates a retrieval-based generation (RAG) approach, where GPT-4 searches a set of legal documents for relevant information before generating the answers.

Key technologies used:

- Flask: A lightweight Python web framework that handles HTTP requests and provides an API.
- OpenAI GPT-4: A large language model (LLM) that generates human-like responses to legal queries.
- LangChain: A powerful framework for building applications that integrate LLM with external data sources such as legal documents. Used to build RAG pipelines.
- Facebook AI Similarity Search (FAISS): A fast similarity search library used to store and retrieve document embeddings in high-dimensional spaces.
- OpenAI Embeddings: Converts legal documents into vector representations used for similarity search and retrieval.

Frontend and backend integration:

- Frontend (Flutter): The platform's frontend is built using Flutter and provides a user-friendly interface for Android users. The frontend collects user input and provides dynamic questionnaires to collect case-specific information. It also supports PDF generation, allowing users to preview, adjust, download, and submit legal documents.
- Backend (ExpressJS and Flask): The backend system handles user requests, processes input collected via ExpressJS and Flask APIs and generates legal documents via AI models. The backend ensures high availability, scalability, and security, and integrates with databases to manage user data and case-specific responses.

Preferences:

- Python 3.8+: Your system requires Python 3.8 or later.
- Environment variables: The OpenAI API key is stored in the .env file and loaded to securely access GPT-4 using the dotenv library.
- LangChain and FAISS: These libraries handle text embeddings for related document retrieval and fast lookups.

2. Preparing the dataset

For the RAG system to provide accurate and contextual answers, a dataset consisting of legal documents is required. These documents contain various legal case information, including case description, settlement amount, and sentence. The dataset is stored in JSON format, and each document contains important metadata.

Dataset Structure:

- JSON Files: The system loads legal case data stored in multiple JSON files. The files are structured to include case information (case description), category (legal category of the case), settlement amount (settlement amount), and sentence (sentence or penalty period).
- Text and Metadata: The dataset consists of:
    - Text: Describes the details of a legal case.
    - Metadata: Includes key attributes such as case category, settlement amount, and sentence.

Data Partitioning:

- The data is organized into different legal categories (online insults, second-hand fraud, sexual harassment, and forced molestation). This segmentation helps the model respond effectively to specific legal issues.

3. Data Preprocessing and Tokenization

Data preprocessing ensures that legal documents are prepared in a structured and clean format that can be integrated with GPT-4 and RAG systems.

The following steps are implemented:

Text preprocessing:

- Clean up legal text by removing unnecessary characters, duplicate formats, and irrelevant data.

Maintain consistency in data structure and format so that the LangChain framework and GPT-4 can process it smoothly.

Tokenization and Embedding:

- Embedding Generation: Generate high-dimensional vector embeddings for each legal document using LangChain's OpenAI Embeddings.
- Vectorization: Tokenizing documents into smaller units to enable efficient searching and retrieval within the system.
- FAISS integration: Using FAISS, we create a vector store of these embeddings to enable similarity search to quickly identify relevant documents during user queries.

4. Legal Document Generation Workflow

The initial goal was to autonomously generate complete legal documents using GPT-4. However, the generated results did not meet the required quality for legal compliance and consistency.

To address these limitations, we adopted a hybrid approach:

- Dynamic Templates: We created pre-designed templates for each legal category (e.g., sexual harassment, fraud).
- Dynamic data insertion: Dynamically populates user-supplied data (e.g., plaintiff name, incident date) into the appropriate sections of the template.
- Static sections: Parts of a legal document that do not require user input are predefined and automatically filled in for consistency.

This method ensures the reliability of predefined legal templates and overcomes the limitations of not using GPT-4.

5. RAG System Implementation

RAG combines document retrieval and generative AI to ensure that generated responses are based on factual data:

1) Search: Search for relevant legal documents in a pre-built vector repository using FAISS.

2) Augmentation: Provides context to GPT-4 by providing retrieved data to generate a response.

Implementation Workflow

- Step 1: User input is processed through the frontend and sent to the backend for analysis.
- Step 2: FAISS retrieves the most relevant legal documents based on the user's query.
- Step 3: Format the retrieved document and pass it to GPT-4 as context.
- Step 4: GPT-4 generates accurate legal advice or draft documents using the retrieved data.

System Output

The generated response contains:

- Predicted settlement amount: Based on similar cases.
- Expected sentence: Calculated using the retrieved data.
- Legal Insights: Personalized advice from AI legal advisors.
- Case law reference: Includes retrieved documents used in the response.

6. Prepare and customize your dataset

We used two main datasets to train the system:

1) AI Hub Dataset 71723: Contains over 60,000 annotated legal cases labeled with summaries, Q&A pairs, and keywords. It has balanced representation with at least 2,000 cases in each legal category.

2) AI Hub Dataset 580: Contains over 10,000 legal decisions and contract documents, with labeled clauses and annotations for consumer benefits or violations.

Customization to fit your target category

We filtered the dataset to focus on relevant civil and criminal law categories:

- Secondhand trading fraud
- Sexual harassment
- Online Defamation
- Forced rape

To address the data imbalance problem and improve the accuracy of the RAG system, we created a customized dataset consisting of 1,000 cases.

Filter and save data:

- Filtering: Select only relevant cases to fit the project goals.
- Storage: All processed data is stored in JSON format and indexed using FAISS for fast retrieval during queries.

IV. EVALUATION & ANALYSIS

The Smart Legal Form Filler is currently in training, and performance evaluation and analysis will be performed after the model is fully trained and validated. However, based on the current development progress, the following evaluation and analysis directions have been set:

1. Evaluation Criteria

The main metrics for evaluating model performance are accuracy, precision, recall, and F1 score. These metrics are essential for judging how well the model generates complaint-generating texts that meet legal standards.

- Accuracy: Measures how well the generated complaints match predefined templates and meet legal requirements.
- Precision: Evaluates how accurately the model generates legally relevant information from complaint text.
- Recall: Ensure that all required elements, such as complainant details and incident description, are included in the generated complaint text.
- F1 score: Provides an overall performance indicator that combines both aspects, providing a balance between precision and recall.

These metrics help us evaluate how effectively the model generates legally compliant and relevant complaints.

2. Initial Results and Analysis

The KOGPT2 model is accurately generating basic details such as plaintiff and defendant information based on initial results. However, more complex sections such as case description and legal outcome require further refinement. There is a possibility of inaccuracies in describing case details or legal outcome, which will be addressed in future iterations of model training.

Additionally, data imbalance was observed in categories such as sexual harassment and sexual assault, where training data was limited. This imbalance may reduce model accuracy for certain legal cases, which will be a major focus for improvement.

3. Improvement Strategy

To improve model performance, we will implement the following improvement strategies:

- Data Augmentation: Increase the training data for different event categories to improve the model's ability to generalize across different scenarios.
- Hyperparameter tuning: Improve model performance by optimizing key hyperparameters such as learning rate, batch size, and number of training epochs.
- Fine-tuning: The model is further fine-tuned to fit different incident types so that it effectively meets the specific legal requirements of each category.
- Using these strategies will allow model performance to improve over time, resulting in more accurate and legally compliant complaint generation.

4. Initial task

Initial implementations of the RAG system suffered from imbalances in the case law dataset. These limitations affected the accuracy of case retrieval and legal document generation.

5. Optimizing Data Sets

- To address the imbalance, we created a custom dataset containing approximately 1,000 customized legal cases.
- We significantly improved the performance of the RAG system for four categories provided by the application

6. Advantages of the RAG System

- Reliability: Ensures factual and reliable output using pre-searched legal case data.
- Scalability: RAG's architecture can be extended to additional case types with minimal overhead.

V. PLATFORM ARCHITECTURE AND COMPONENTS

This section describes the architecture of the Smart Legal Form Filler platform and provides a detailed description of its key components. The platform is designed with front-end and back-end technologies and an AI-based engine that automates the creation of legal documents. Users can create legally compliant documents in various categories such as second-hand fraud, online abuse, sexual harassment, and forced sexual assault. The following sections provide a detailed description of its components, including the front-end, back-end, AI models, and other important modules.

1. Frontend Storage Description

- Repository: Smart Legal Form Filler/Frontend

- Function & Role: The front-end store is responsible for providing the user interface and experience. This store uses Flutter to create cross-platform applications that work on Android and iOS devices.

- Files:

- lib/:
    - main.dart: The entry point of a Flutter application, initializes the app and sets up navigation.
    - Screens/:
        - category_selection_screen.dart: Allows the user to select a category of legal case (e.g., sexual harassment, fraud).
        - completion_screen.dart: Display the final screen after document creation is complete, and implement a button to return to the main page.
        - document_form.dart: This is the screen where the user can enter details about a legal case.
        - loading_screen.dart: A transition screen that shows a loading indicator while processing user data.
        - main_screen.dart: This is the main menu screen that allows you to navigate to various functions of the app.
        - pdf_result_screen.dart: Displays the final document, with download options and a View Incident Report button.
        - splash_screen.dart: This is the splash screen that is initially displayed when the app is launched.
        - analyzing_criminal.dart: Linked with the case report button. Linked with the GPT API, it generates advice on the case, and uses RAG technology to display similar precedents to the user's case, the expected settlement amount, the expected sentence of the offender, and three similar precedents.
    - utils/: Contains utility files for data validation, constants, and reusable components.

2. Backend_AI Storage Description

- Repository: Smart-Legal-Form-Builder/backend AI

- Function & Role: The Backend_AI repository supports legal document generation using AI, specifically the GPT-4 model. It provides the logic to process user input and generate legal documents based on the questions answered by the user.

- Files:

- main.py: The main server file that implements Flask routes that handle user queries and interact with the RAG (Request-Based Build) system.
- RAG_dataset/:
    - ForcedAssault.json: Contains legal case data related to forced assault incidents.
    - sexualharassment.json: Contains data about sexual harassment.
    - OnlineSwearing.json: Focuses on online abuse and defamation cases.
    - Used Goods Transaction Fraud.json: Deals with cases related to used goods transaction fraud.
- README.md: Provides basic documentation for backend setup and functionality.

3. AI Repository Description

- Repository: Smart-Legal-Form-Builder/AI

- Function & Role: The AI Repository serves to train and fine-tune AI models used to generate legal documents. The repository contains scripts for data preparation, model training, and legal response generation.

- Files:

- RAG_dataset/:
    - ForcedAssault.json: Contains legal case data related to forced assault incidents.
    - sexualharassment.json: Contains data about sexual harassment.
    - OnlineSwearing.json: Focuses on online abuse and defamation cases.

- Used Goods Transaction Fraud.json: Deals with cases related to used goods transaction fraud.

- complaint_rag.py: A script that handles the creation of legal documents using RAG.
- complaint_rag2.py: A replacement or updated script to improve documentation generation.

4. AI_for_dataset Repository Description

- Repository: Smart-Legal-Form-Builder/AI for dataset

- Function & Role: The AI_for_dataset repository is responsible for creating and preparing datasets used for AI model training. It performs the work of organizing, structuring, and augmenting legal data to make it suitable for AI training.

- Files:

- .ipynb_checkpoints/:
  - generator_ngram_creativity-checkpoint.ipynb: A Jupyter Notebook for generating and augmenting the datasets.
- kogpt2/:
  - init.py: Initializes the kogpt2 module.
  - data.py: Handles data loading and preprocessing tasks.
  - mxnet_kogpt2.py: Implements KoGPT2 using MXNet.
  - pytorch_kogpt2.py: Implements KoGPT2 using PyTorch.
  - utils.py: Provides utility functions for model training and data processing.
- samples/: Contains sample data sets and model output for reference.
- LICENSE: Specifies the license conditions of the repository.
- README.md: Contains documentation on preparing the dataset and training the AI.
- dataset.csv: A structured data set used for training AI models.
- generator.py: A script that augments existing data sets or generates synthetic data.
- main.py: A central script that runs the data preprocessing and augmentation workflow.

5. Summary of the Updated Structure

- Frontend:
  - Focuses on UI/UX for cross-platform applications.
  - Includes several screens and utility components for seamless interaction.
- Backend_AI:
  - Manages the server logic and API routes.
  - Processes user data and interacts with the RAG system.
- AI:
  - Implements the RAG system with GPT-4 for legal document generation.
  - Includes case-specific data sets and test scripts.
- AI_for_dataset:
  - Prepares and augments data sets AI model training.
  - Supports KoGPT2 implementation for localized legal data processing.

The Smart Legal Form Filler platform leverages multiple repositories and files to provide a seamless experience for users who want to create legal documents. The front-end repository builds the user interface, while the back-end_AI repository integrates AI models to process user input and create legal documents. The AI repository fine-tunes and trains AI models using legal data, while the AI_for_dataset repository prepares and organizes the datasets needed to train the models. By combining these technologies and processes, users can easily create legal documents without prior legal knowledge.

VI. Related Work (e.g., existing studies)

1. Existing Tools and Platforms

A. Rocket Lawyer

Rocket Lawyer is a multi-purpose platform that provides users with a variety of tools to create, manage, and store legal documents. It offers a variety of pre-written templates for common legal needs, including contracts, leases, and wills. The platform streamlines document creation through guided questionnaires, allowing users to customize templates with case-specific details. Rocket Lawyer also connects users with licensed attorneys who provide expert legal advice and document reviews when needed. This combination of automation and human assistance is particularly useful for individuals and small businesses seeking comprehensive legal assistance.

B. LegalZoom

LegalZoom is one of the most well-known platforms for creating custom legal documents online. It offers an extensive library of templates and a guided process for users to fill in relevant details and create custom documents for personal and business use. In addition to document creation, LegalZoom's services include trademark registration, business formation assistance, and providing expert legal advice. Although it does not specialize in complaints handling, its user-friendly

interface and wide range of services make it a great solution for individuals and businesses looking for an efficient and accessible legal tool.

C. LawDepot

LawDepot focuses on enabling users to create legal documents quickly and efficiently through a user-friendly platform. It offers an extensive library of customizable templates that cover a wide range of legal needs, including contracts, agreements, and estate planning documents. By answering a series of guided questions, users can create documents tailored to their specific situation. LawDepot is especially useful for individuals looking for a cost-effective solution to handle everyday legal issues without professional assistance. The focus on customization and simplicity is consistent with the growing demand for accessible and simple legal tools.

2. Libraries and Frameworks

A. Flutter

Flutter is used to develop cross-platform applications. The framework allows developers to create responsive and interactive user interfaces using a single code base. Flutter's widget-based architecture simplifies the implementation of key features such as step-by-step questionnaire modules, real-time document previews, and PDF generation capabilities. The platform provides a consistent and seamless experience for users. This reduces development time and maintains high performance.

B. Axios

Axios is used to manage real-time communication between the frontend and the backend via HTTP requests and API calls. It plays a key role in sending user input submitted in the questionnaire to the backend for processing by the AI model. Once the questionnaire is created, Axios returns it to the frontend for the user to preview and download. This integration ensures fast, reliable, and lag-free interactions.

3. Artificial Intelligence and Natural Language Processing Resources

A. Hugging Face Transformers

Hugging Face Transformers played a key role in the development of the smart legal form filler, providing a flexible framework for testing a variety of NLP tasks. Early prototypes leveraged the framework's capabilities to explore text summarization, dynamic question answering, and contextual understanding, which were reflected in the design of the platform's step-by-step survey interface. The framework enabled us to evaluate the adaptability of different models to legal datasets, providing useful insights that shaped the fine-tuning strategy that was later implemented with the GPT-4 API. While primarily used for research and experimentation, Hugging Face Transformers was essential in improving AI-based workflows and establishing a solid foundation for user-centric solutions.

4. Relevant Studies and Documentation

A. "AI in Legal Tech: Opportunities and Challenges"

The study highlighted the transformative potential of AI in legal technology, focusing on automating document creation and streamlining complex legal processes. This enabled the platform to meet the nuanced needs of users using dynamic question flows and context-aware text generation.

B. National Legal Awareness Surveys (2019, 2021)

The survey uncovered widespread public concerns about legal terminology and accessibility, suggesting the difficulty of writing legal documents and the need for a complaint writing app. The key findings of difficulty understanding legal documents and limited online interaction with legal resources led to consideration of a streamlined process and question-based complaint writing functionality.

C. Blogs and Developer Resources

Practical resources from platforms like OpenAI and GitHub have helped us implement scalable and user-friendly systems. These guides have helped us integrate AI with front-end and back-end systems to ensure optimal performance and a smooth user experience.

VII. CONCLUSION: DISCUSSION

Smart Legal Form Filler offers an innovative approach to overcoming the barriers of the existing legal system. It emphasizes user-friendliness and utilizes AI technology and intuitive design to enable users to independently create accurate and legally compliant legal documents. The app simplifies the process of suing wrongdoers, reducing reliance on expensive legal services and reducing distance from complex and difficult laws.

Key Achievements:

- Improved accessibility: The platform simplifies complex legal tasks, making legal documents accessible to non-experts.
- Cost and time efficiency: Users can create professional-quality documents in minutes, significantly reducing reliance on expensive legal services.
- Technological Innovation: By leveraging cutting-edge artificial intelligence and integrating scalable backend solutions, the platform sets a new standard for legal technology.

Challenges and Future Directions:

Although the smart legal form filler has achieved its initial goals, several challenges still remain:

A. Case Type Extension

- Expand services by developing templates for additional case types, such as real estate disputes, labor disputes, consumer complaints, and public service-related issues, in addition to sexual harassment, online defamation, second-hand transaction fraud, and forced molestation.
- Expand the scope of the platform to address a wider range of legal issues by comprehensively supporting diverse user needs.

B. Cooperation with legal professionals

- Incorporates an optional review feature that allows users to connect with legal experts to review and improve draft documents.
- Work with your attorneys to ensure that the platform's output is accurate and up-to-date by ensuring that the templates reflect the latest legal precedents and regulatory changes.

By addressing these challenges, Smart Legal Form Filler will strengthen its position as a versatile and user-centric legal solution and expand its capabilities to better serve its global users.

In conclusion, Smart Legal Form Filler is not only a tool for automating legal documents, but also an innovative solution that creates a more inclusive legal ecosystem. Through continuous development and improvement, it will democratize access to law and help the legal system serve everyone equally.

VIII. REFERENCES

[1] National Legal Awareness Survey, “Survey on public awareness of legal terminology and access to legal resources,” 2019 & 2021. [Data set].

[2] OpenAI, “GPT-4 API documentation,” 2024. [Online]. Available: https://platform.openai.com/docs

[3] Flutter Team, “Flutter documentation,” 2024. [Online]. Available: https://flutter.dev

[4] Axios Project, “Axios documentation,” 2024. [Online]. Available: https://axios-http.com

[5] Syncfusion, “Syncfusion Flutter PDF documentation,” 2024. [Online]. Available: https://www.syncfusion.com/flutter-widgets/flutter-pdf-viewer

[6] Hugging Face, “Transformers documentation,” 2024. [Online]. Available: https://huggingface.co/docs/transformers

[7] Brookings Institution, “Embracing AI in legal practice: Opportunities and challenges,” May 6, 2024. [Online]. Available: https://www.brookings.edu/articles/embracing-ai-in-legal-practice-opportunities-and-challenges-the-techtank-podcast/

[8] AI Hub, "AI 학습용 법률/규정 텍스트 분석 데이터," 2024. [Online]. Available: https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=71723

[9] AI Hub, "AI 학습용 판결문 및 약관 텍스트 데이터," 2024. [Online]. Available: https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=580

[10] Meta AI, "FAISS Documentation: Scalable similarity search for AI applications," 2024. [Online]. Available: https://github.com/facebookresearch/faiss

[11] LangChain, "LangChain documentation: Building applications with LLMs," 2024. [Online]. Available: https://docs.langchain.com/

[12] Flask, "Flask documentation: Web application development," 2024. [Online]. Available: https://flask.palletsprojects.com/

[13] TensorFlow Team, "TensorFlow documentation," 2024. [Online]. Available: https://www.tensorflow.org

[14] PyTorch Team, "PyTorch documentation," 2024. [Online]. Available: https://pytorch.org