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 technologies, including Retrieval-Augmented Generation (RAG) and the GPT-4 model, to automate the creation of legally compliant complaint forms. By addressing barriers such as limited legal knowledge, complex terminology, and high costs associated with traditional legal services, this platform empowers users to efficiently navigate legal challenges.**

This study outlines the methodologies and technologies employed in building 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 for legal cases such as secondhand fraud, online abuse, sexual harassment, and assault. The RAG framework enhances the model's reliability by combining retrieved legal precedents with AI-generated responses, enabling predictions of settlement amounts, sentences, and actionable legal advice.

The functionality of the application extends beyond document creation. Features such as real-time input validation, seamless integration of user interaction and AI-based automation ensure a smooth and accessible experience. Developed using Flutter, the platform offers cross-platform compatibility, allowing Android and iOS users to access the same responsive and user-friendly interface. The generated documents are processed via Syncfusion Flutter PDF, allowing users to preview, adjust, download, and submit professional-quality PDFs directly from the app. The app also features a user-friendly main menu and completion confirmation system, designed to enhance the overall user experience and encourage ongoing engagement.

The Smart Legal Form Builder exemplifies the transformative potential of AI in legal technology, reducing reliance on expensive professional assistance and making legal services more accessible. Future improvements include expanding case coverage, enhancing multilingual capabilities, and integrating with official legal systems for seamless complaint filing. Through continuous innovation, this platform aims to democratize access to justice and bridge the gap between individuals and the legal system.

Keywords— Legal Technology, AI-Powered Legal Tools, Retrieval-Augmented Generation (RAG), Complaint Automation, Legal Document Generation, GPT-4, Cross-Platform Application, Secondhand Fraud, Online Abuse, Sexual Harassment, Assault and Injury, Access to Justice, Cost-Effective Legal Solutions, AI Legal Assistant, Multilingual Legal Support, Legal Precedent Retrieval

I. INTRODUCTION

1. Motivation

For many, the law is difficult and inaccessible. While everyone has had to draft a complaint at some point in their lives, this basic task is often difficult for individuals who lack the legal expertise or resources to do so. This problem disproportionately affects marginalized communities, amplifying social inequality and widening the gap between those who can and cannot access professional legal assistance.

However, advances in AI and automation are helping to solve this problem through the development of apps. Smart Legal Form Builder leverages these technologies to streamline the legal process and help individuals draft legal documents professionally and independently. The app specializes in creating complaints for four key categories: secondhand transaction fraud, online abuse, sexual harassment, assault and injury. This not only addresses immediate barriers, but also makes filing a complaint 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 having difficulty understanding legal terms, and 78.4% found legal documents difficult to understand. This lack of understanding is a major cause of individuals' difficulties in participating in the legal system. Misinterpretation of legal terms often leads to errors in document preparation, which creates further obstacles to achieving legal resolution.

B. Limited Interest and Familiarity with Legal Information

According to the 2021 National Legal Awareness Survey, 79.9% of respondents reported never having searched for legal information online. Additionally, only 23.4% discussed legal topics with acquaintances to better understand the law, indicating a lack of interest in legal information. This suggests that even though legal tools exist, most individuals do not know how to use them effectively.

C. High Costs and Barriers to Accessing Legal Professionals

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

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

3. Vision and Goals

The core of Smart Legal Form Builder is to provide an accessible, intuitive and reliable solution. The app allows users to reduce the burden of filing a complaint and instantly create the necessary legal documents. The goal is to make filing a complaint easy and convenient for anyone, rather than just a professional.

The platform aims to eliminate the dependency on legal professionals and allow users to independently draft complaints. It automates complex processes and provides step-by-step questions and guidance. The platform aims to simplify the process of drafting complaints for four major categories: secondhand transaction fraud, online abuse, sexual harassment, assault and injury. By providing templates and guidance for these cases, Smart Legal Form Builder helps users complete 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 articulate their claims and demands.

At the end of the process, Smart Legal Form Builder envisions a more equitable 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 develops into a scalable framework that can be extended to other areas of complaint.

The success of this feature will be measured not only by the efficiency and accuracy of the complaints it generates, but also by its transformative impact in closing the gap in legal access and creating a more just society.

II. DATASETS

This section describes the structure and generation method of the datasets used for training the Retrieval-Augmented Generation (RAG) system. The dataset contains legal cases from four categories: Sexual Harassment, Online Defamation, Fraud in Online Transactions, and Sexual Assault. Each case contains key information such as the incident description, legal settlement, sentence, and verdict summary.

1. Dataset Structure:

The dataset used for training the RAG system is structured in a JSON format, where each case record contains detailed information about a legal case. Each record includes the following fields:

- id: A unique identifier for each case
- category: The legal category of the case (e.g., "Sexual Harassment")
- text: A brief summary of the legal incident or the core content of the case
- settlement: The financial settlement amount awarded in the case (if any)
- sentence: The sentence imposed by the court, typically in months
- verdict_summary: A concise summary of the verdict handed down by the court
- keywords: A list of keywords that describe important aspects of the case (e.g., type of crime, location, and other relevant details)

Each case in the dataset includes the necessary information to allow the system to identify patterns, retrieve similar cases, and generate legal insights for user queries.

2. Dataset Generation Method:

The dataset is generated through the following steps:

1) Data Collection: Legal cases are gathered from public legal sources, including court records and news articles, and categorized into four main types: Sexual Harassment, Online Defamation, Fraud in Online Transactions, and Sexual Assault.

2) Preprocessing: The raw data is cleaned and formatted, ensuring consistent and concise descriptions of each case. The data is then tokenized and indexed for easy retrieval by the RAG system.

3) Annotation: Each case is annotated with relevant information such as category, settlement, sentence, and

verdict summary. Keywords are added to enhance searchability.

4) Categorization: The dataset is divided into four categories based on the type of legal issue, allowing for efficient querying and retrieval of similar cases.

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

3. RAG Dataset Utilization

1) Dataset Overview: Legal/Regulatory Text Analysis Data

- **Source**: AI Hub Dataset 71723
- **Description**:
  - This dataset includes over 60,000 annotated case data for AI training, with labels assigned to critical features like case summaries, Q&A sets, and terminology keywords.
  - It represents real-world distributions of legal case types to ensure balanced representation across categories, with at least 2,000 cases per category.
- **Purpose**:
  - Designed to support academic and industrial research in AI and legal natural language processing.
  - Helps improve performance in 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**
- **Description**:
  - Contains processed data from over 10,000 legal judgments, including extracted facts, claims, and analysis of advantageous/disadvantageous contract clauses.
  - Includes labeling and tagging of unlawful or beneficial clauses to assess consumer impact.
- **Purpose**:
  - Facilitates accessibility to legal judgments and contracts, especially for non-experts.
  - Tailored to enhance AI performance in understanding legal text structures and identifying critical elements.

4. Dataset Filtering and Customization

- Initial Approach: Initially, all civil and criminal law-related data were used to train the system. However, the lack of relevant examples in the selected categories (e.g., fraud, online defamation) led to decreased RAG system performance.
- Optimization: Approximately 1,000 tailored legal cases were generated by filtering and customizing the datasets. This ensured better relevance and improved RAG system performance, particularly for categories like online abuse and fraud.

III. METHODOLOGY

This section details the methodology implemented to build the Smart Legal Form Builder platform, which utilizes Retrieval-Augmented Generation (RAG) and GPT-4 technologies for legal document creation. The methodology encompasses data preparation, system implementation, and integration of AI pipelines to provide accurate and legally compliant complaint forms. Below is an overview of the technical process and components involved in building the platform.

1. AI Pipeline Overview

The RAG framework is central to the platform, combining retrieval-based approaches with GPT-4's generative capabilities. The following steps illustrate the AI pipeline:

- FAISS Storage Construction: Legal datasets in JSON format are embedded into vector representations and stored in FAISS (Facebook AI Similarity Search). This allows efficient retrieval of case-specific information during user queries.
- Retriever Configuration: A retrieval engine identifies and ranks the most relevant documents based on their similarity to user-provided inputs. The top three documents are selected for augmentation.
- Prompt Template Design: Customized templates are created for each legal category, integrating references like case numbers and extracted documents. These prompts ensure that GPT-4 generates responses that adhere to legal standards.
- RAG Chain Construction: Using LangChain, the system integrates FAISS retrieval and GPT-4 generation to form an optimized RAG chain. This enables accurate, context-aware outputs grounded in factual data

2. Dataset Preparation

To ensure the model generates contextually relevant and legally compliant responses, datasets from diverse legal sources were curated, processed, and augmented.

Dataset Sources:

- Legal/Regulatory Text Analysis Data:
  - 60,000+ annotated legal cases with summaries, Q&A pairs, and keyword tagging.

- Balanced distribution across categories, including civil and criminal law cases.

- Judgment and Contract Analysis Data:
  - 10,000+ cases extracted from contracts, judgment records, and rulings. Data includes unlawful clause tagging and consumer advantage/disadvantage analysis.
- GPT-Generated Complaint Templates:
  - AI-generated drafts for underrepresented categories like sexual harassment and assault.
  - Approximately 1,000 synthetic datasets were added to address data imbalance.

Data Segmentation:

- Data was categorized into key legal areas: fraud in secondhand transactions, online abuse, sexual harassment, and assault. Segmentation ensures targeted responses based on case types.

3. Data Preprocessing and Tokenization

To integrate datasets effectively into the RAG system, preprocessing and tokenization steps were performed:

Text Cleaning:

- Unnecessary characters, redundant formatting, and irrelevant data were removed to ensure uniformity across datasets.

Embedding and Vectorization:

- LangChain's OpenAIEmbeddings converted legal documents into high-dimensional vector representations.
- FAISS was used to index these vectors for fast and accurate retrieval.

Linguistic Pattern Analysis:

- Transformers libraries analyzed recurring sentence structures and linguistic patterns to identify commonalities across cases. This helped design standardized templates and prompts.

4. Legal Document Generation Workflow

Given the lack of fixed formats for complaints, a hybrid approach combining templates and dynamic data insertion was adopted:

- Template Design: Templates were created for each legal category, extracting repetitive phrases and legal language patterns from existing documents. Pre-defined sections such as case numbers, incident details, and legal claims were incorporated.
- Dynamic Data Insertion: User inputs (e.g., names, dates, locations) were dynamically populated into the templates, ensuring personalized and legally compliant outputs.
- Static Section Pre-Definition: Common elements of complaints, such as introductions and closing statements, were auto-filled to maintain consistency.

5. RAG System Implementation

The RAG system integrates retrieval and generation to ensure accurate and relevant responses. The workflow is as follows:

- Step 1: User-provided data is collected via the frontend and sent to the backend for processing.
- Step 2: FAISS retrieves the top three most relevant documents based on the input query.
- Step 3: Retrieved documents are provided as input to GPT-4, enhancing the model's ability to generate fact-based responses.
- Step 4: GPT-4 produces detailed outputs, including predicted settlement amounts, estimated sentences, and actionable legal advice.

6. Integration and Deployment

Frontend:

- Developed using Flutter, the frontend supports cross-platform functionality, providing a seamless interface for Android and iOS users. Features include real-time input validation, dynamic questionnaires, and PDF previews.

Backend:

- Implemented with Flask, the backend ensures efficient communication between the frontend and the AI modules. It handles user requests, integrates the RAG system, and generates documents in real time.

Environment Setup:

- Python 3.8+ was utilized, with dotenv for secure API key management. LangChain and FAISS libraries were essential for embedding and retrieval processes.

IV. Evaluation & Analysis

The Smart Legal Form Builder is currently undergoing training, and performance evaluation and analysis will be conducted once the model is fully trained and validated. However, based on the current development progress, the following evaluation and analysis directions have been established.

1. Evaluation Metrics

The key metrics for evaluating the model's performance include accuracy, precision, recall, and F1-score. These metrics are essential in determining how well the model generates text that aligns with legal standards for complaint generation.

- Accuracy: Measures how closely the generated complaints match the predefined templates and meet legal requirements.
- Precision: Evaluates the accuracy with which the model generates legally relevant information in the complaint text.
- Recall: Ensures that all necessary elements, such as complainant details and incident descriptions, are included in the generated complaint text.
- F1-score: Balances precision and recall for an overall performance assessment.
- Template Compliance: Assesses the alignment of generated text with predefined complaint templates based on recurring linguistic and structural patterns.

These evaluation metrics will guide the assessment of how effectively the model generates legally compliant and relevant complaints.

2. Initial Results and Analysis

Dataset Integration and Template Design:

- Data Sources: The complaint templates are built by integrating GPT-generated complaint samples and collected legal documents. These datasets were analyzed and merged to generate structured templates for each case type.
- Pattern Recognition: Using Transformers, frequently occurring phrases and sentence patterns were extracted and used to identify linguistic structures unique to legal complaints. This step enabled the automation of initial draft generation tailored to specific case categories.

Early Outcomes:

- Basic Complaint Sections: Sections like complainant and defendant information are generated accurately across all case categories.
- Complex Language Challenges: Incident descriptions and legal outcomes require further refinement to handle nuanced language and ensure contextual accuracy.
- Template-Driven Structure: The system effectively incorporates pre-defined complaint structures, with user data dynamically populated to create comprehensive drafts.

3. Improvement Strategy

To enhance the model's performance, the following improvement strategies will be implemented

- Data Augmentation: Increase the training data for various incident categories, particularly for underrepresented legal cases, to improve the model's ability to generalize across diverse scenarios.
- Hyperparameter Tuning: Optimize key hyperparameters such as learning rate, batch size, and the number of training epochs to enhance the model's performance.
- Fine-tuning: Further fine-tune the model to specialize in different incident types, ensuring that legal requirements specific to each category are met effectively.

By employing these strategies, the model's performance is expected to improve over time, leading to more accurate and legally compliant complaint generation.

4. Dataset Optimization

Dataset Creation Workflow:

1) GPT-Generated Complaint Samples:

- Over 1,000 legal complaint samples were generated using GPT models, covering categories such as fraud, harassment, and assault.
- The generated samples provided diverse linguistic variations and sentence structures.

2) Collected Case Data:

- Additional datasets were gathered from legal archives and pre-annotated to include metadata like case type and outcomes.
- This dataset was analyzed to identify recurring phrases, structural elements, and logical flow.

3) Merged Templates:

- The two datasets were combined to create highly optimized templates for specific case types, enhancing both coverage and accuracy.

Linguistic Analysis:

- Using Transformers libraries, recurring sentences and language patterns were extracted. These patterns were analyzed to:
  - Identify key components of legal complaints.
  - Establish consistency in tone and format.
  - Ensure compliance with legal standards.

5. RAG System Advantages

- Reliability: Utilizing pre-retrieved legal case data ensures factual and trustworthy outputs.
- Efficiency: Speeds up the document drafting process, enabling users to generate legal complaints in minutes.
- Scalability: RAG's architecture allows expansion into additional case types with minimal overhead.

V. PLATFORM ARCHITECTURE AND COMPONENTS

This section outlines the architecture of the Smart Legal Form Builder platform and provides a detailed explanation of its key components. The platform is designed with both front-end and back-end technologies, along with an AI-powered engine that automates legal document generation. It enables users to create legally compliant documents in various

categories such as secondhand transaction fraud, online abuse, sexual harassment, and assault. The following sections provide detailed descriptions of the components, including the frontend, backend, AI models, and other critical modules.

1. Frontend Repository Explanation

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

- Function & Role: The frontend repository is responsible for providing the user interface and experience. This repository utilizes Flutter to create a cross-platform application that works on both Android and iOS devices.

- Files:

- lib/:
  - main.dart: The entry point of the Flutter application, initializing the app and setting up navigation.
  - Screens/:
    - category_selection_screen.dart: Allows users to select the category of their legal case (e.g., harassment, fraud).
    - completion_screen.dart: Displays the final screen after completing the document generation, with options to download or share.
    - document_form.dart: A screen for users to input details about their legal case using dynamic forms.
    - loading_screen.dart: A transitional screen showing a loading indicator while processing user data.
    - main_screen.dart: The main menu screen providing navigation to various features of the app.
    - pdf_preview.dart: Provides a preview of the generated PDF document.
    - pdf_result_screen.dart: Displays the final document with options for download.
    - splash_screen.dart: An initial splash screen shown when the app is launched.
  - utils/: Contains utility files for data validation, constants, and reusable components.

2. Backend_AI Repository Explanation

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

- Function & Role: The Backend_AI repository powers the legal document generation using AI, particularly 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 implementing Flask routes to handle user queries and interact with the RAG (Retrieval-Augmented Generation) system.
- RAG_dataset/:
  - 강제추행.json: Contains legal case data related to sexual harassment cases.
  - 성희롱.json: Includes data on workplace harassment.
  - 온라인욕설.json: Focuses on online abuse and defamation cases.
  - 중고거래사기.json: Covers cases related to secondhand transaction fraud.
- README.md: Provides basic documentation about the backend setup and functionality.

3. AI Repository Explanation

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

- Function & Role: The AI repository is responsible for training and fine-tuning the AI models used to generate legal documents. This repository includes scripts for preparing data, training models, and generating legal responses.

- Files:

- RAG_dataset/:
  - 강제추행.json: Contains detailed case data for sexual harassment cases.
  - 성희롱.json: Includes workplace harassment cases.
  - 온라인욕설.json: Covers online abuse cases.
  - 중고거래사기.json: Contains data on fraud related to secondhand transactions.
- complaint_rag.py: Script for handling legal document generation using RAG.
- complaint_rag2.py: An alternate or updated script for refining document generation.

4. AI_for_dataset Repository Explanation

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

- Function & Role: The AI_for_dataset repository is responsible for creating and preparing datasets that are used for training the AI models. It handles the cleaning, structuring, and augmentation of legal data to make it suitable for AI training.

- Files:

- .ipynb_checkpoints/:

    - generator_ngram_creativity-checkpoint.ipynb: A Jupyter Notebook for generating and augmenting 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 handling.
- samples/: Contains sample datasets and model outputs for reference.
- LICENSE: Specifies the licensing terms for the repository.
- README.md: Contains documentation about dataset preparation and AI training.
- dataset.csv: A structured dataset used for training AI models.
- generator.py: Script for generating synthetic data or augmenting existing datasets.
- main.py: The central script for running data preprocessing and augmentation workflows.

5. Summary of the Updated Structure

- Frontend:
    - Focuses on UI/UX for cross-platform applications.
    - Contains multiple 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 datasets and scripts for testing.
- AI_for_dataset:
    - Prepares and augments datasets for training the AI models.
    - Supports KoGPT2 implementation for localized legal data processing.

The Smart Legal Form Builder platform leverages multiple repositories and files to provide a seamless experience for users seeking legal document creation. The Frontend repository builds the user interface, while the Backend_AI repository integrates the AI model to process user input and generate legal documents. The AI repository fine-tunes and trains the AI model using legal data, while the AI_for_dataset repository prepares and cleans the datasets necessary for model training. By combining these technologies and processes, the platform enables users to easily create legal documents, even without prior legal knowledge.

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

1. Existing Tools and Platforms

A. Rocket Lawyer

Rocket Lawyer is a versatile platform that provides users with tools to create, manage, and store legal documents. It offers a variety of pre-built templates for common legal needs, including contracts, leases, and wills. The platform streamlines document creation with guided questionnaires, allowing users to customize templates with case-specific details. Rocket Lawyer also connects users with licensed attorneys to provide on-demand legal advice and document reviews, giving users access to expertise when they need it. This combination of automation and human assistance makes Rocket Lawyer especially valuable to individuals and small businesses seeking comprehensive legal assistance.

B. LegalZoom

LegalZoom is one of the most well-known platforms for creating personalized legal documents online. It offers an extensive library of templates and provides users with a guided process to input relevant details and create custom documents for personal and business use. LegalZoom's services extend beyond document creation, including trademark registration, business formation assistance, and access to expert legal advice. Although not specialized in complaints handling, its user-friendly interface and wide range of offerings make it a great solution for individuals and businesses looking for efficient and accessible legal tools.

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, ensuring compatibility across Android and iOS devices. The framework leverages a single codebase, allowing developers to create responsive and interactive user interfaces. 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's cross-platform compatibility provides a consistent and seamless experience for users. This allows users to reduce development time and maintain high performance.

B. Axios

Axios is used to manage real-time communication between the front end and the back end via HTTP requests and API calls. It plays a critical role in processing user input submissions from the questionnaire to the back end for processing by the AI model. Once the questionnaire is generated, Axios returns it to the front-end for the user to preview and download. This streamlined integration ensures fast, reliable, and lag-free interactions.

C. PDF Generation with Flutter Libraries

PDF generation and export is managed using Flutter-compatible libraries such as [syncfusion_flutter_pdf], which converts AI-generated text into professional-quality PDF documents. These documents are formatted to include appropriate headers, footers, and jurisdictional compliance elements, ready for submission or sharing. The ability to create custom layouts ensures that documents meet both legal and professional expectations.

3. AI and NLP Resources

A. OpenAI GPT-4 API

The OpenAI GPT-4 API is the underlying technology behind Smart Legal Form Builder, which can automatically generate well-formed and context-sensitive complaint forms. To optimize performance, the model was fine-tuned using a curated dataset categorized into four legal case types: second-hand fraud, online abuse, sexual harassment, and assault. The fine-tuning process involved collecting and preprocessing legal documents to highlight the specific terminology, logical structure, and formatting required for each category. By training the model on this categorized dataset, it was able to adapt to the nuances of legal language and format standards, ensuring that the output matches the user-provided input and legal requirements. Validation and testing demonstrated the model's ability to efficiently generate professional-quality documents, reduce reliance on manual drafting, and improve accessibility to legal services.

B. Hugging Face Transformers

Hugging Face Transformers played a pivotal role during the development phase of Smart Legal Form Builder, offering a flexible framework for testing various natural language processing tasks. Early prototypes leveraged its capabilities to explore text summarization, dynamic question-answering, and contextual understanding, all of which informed the design of the platform's step-by-step questionnaire interface. The framework also enabled the evaluation of different models' adaptability to legal datasets, providing valuable insights that shaped the fine-tuning strategy later implemented with the GPT-4 API. Although primarily used for research and experimentation, Hugging Face Transformers proved essential in refining the AI-based workflows and establishing a robust foundation for the platform's user-centric solution.

4. Relevant Studies and Documentation

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

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

B. National Legal Awareness Surveys (2019, 2021)

This survey revealed widespread public concerns about legal terminology and accessibility, which in turn suggested the challenges of legal document creation and the need for a complaint writing app. The key findings of difficulty understanding legal documents and limited online engagement with legal resources led us to consider simplified processes and question-based complaint writing capabilities.

C. Blogs and Developer Resources

Practical resources from platforms such as OpenAI, React Native, and Cassandra helped us implement scalable and user-friendly systems. These guides helped streamline the integration of AI with front-end and back-end systems, ensuring optimal performance and seamless user experience.

VII. CONCLUSION: DISCUSSION

The Smart Legal Form Builder represents a significant innovation in legal technology by leveraging Retrieval-Augmented Generation (RAG) and GPT-4 to simplify and automate the creation of legally compliant complaint forms. Through the integration of dynamically designed templates, real-time user input, and AI-enhanced document generation, the platform has demonstrated its potential to bridge the gap between individuals and the complexities of the legal system.

Key Achievements:

- System Efficiency: By incorporating pre-built templates into the RAG framework, the system

achieves higher accuracy and contextual relevance when generating legal complaints. This approach mitigates the inherent limitations of GPT alone.

- Template Integration: The platform underscores the importance of curated templates, which not only improve the precision of AI-generated outputs but also ensure compliance with legal standards.
- AI and User Synergy: By combining user input with intelligent document generation, the platform provides a seamless and accessible experience, allowing users to create professional-grade legal documents without requiring expertise.

Challenges and Lessons Learned:

- Limitations of GPT in Legal Contexts: Initial attempts to directly generate legal complaints using GPT models highlighted the need for a structured approach. The inclusion of templates proved essential for maintaining accuracy and relevance
- Data Accuracy and Curation: Building a reliable dataset was a critical realization, as the accuracy of outputs depends significantly on the quality and structure of input data.

Future Directions:

To further enhance the platform's capabilities, several areas of improvement and expansion have been identified:

- Dataset Expansion and Optimization: Continuously enrich the dataset with a broader variety of legal cases to improve model performance across underrepresented categories such as sexual harassment and fraud.
- Enhanced RAG Implementation: Develop an advanced prediction system within the RAG framework to offer case-specific outcomes such as expected settlement amounts and sentencing guidelines.
- Collaboration with Legal Experts: Engage legal professionals to validate templates, improve system outputs, and ensure compliance with evolving legal standards.
- Platform Scalability: Expand the platform to cover additional legal document types and refine the user interface for greater accessibility.
- Regulatory Compliance: Ensure adherence to international and local legal regulations, incorporating updates to legal frameworks as needed.

Vision for the Future:

The Smart Legal Form Builder envisions a legal ecosystem where individuals and small businesses can independently and efficiently address their legal needs. By democratizing access to legal tools, the platform not only reduces the cost and effort associated with professional legal services but also empowers users to navigate legal challenges confidently.

Through continuous innovation and collaboration, this platform will evolve into a comprehensive legal support system, setting new standards for accessibility, efficiency, and accuracy in legal technology. The integration of future features such as multilingual support and seamless connections with legal institutions will further enhance its utility and societal impact.

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