

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Project Report on
CaseSense Analytics

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2023-24

Submitted by
Rahul Motwani - D17A - 44
Nimish Chidrawar - D17A - 11
Malhar Kajale - D17A - 28
Mohit Shahdadpuri - D17A - 62

Project Mentor
Dr. Sujata Khedkar

(2023-24)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering



Certificate

This is to certify that **Rahul Motwani (D17A, 44), Nimish Chidrawar (D17A, 11), Malhar Kajale (D17A, 28), Mohit Shahdadpuri (D17A, 62)** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on “**CaseSense Analytics**” as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Dr. Sujata Khedkar** in the year 2023-24.

This project report entitled **CaseSense Analytics** by **Rahul Motwani, Nimish Chidrawar, Malhar Kajale, Mohit Shahdadpuri** is approved for the degree of **B.E. Computer Engineering**.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide: **Dr. Sujata Khedkar**

Project Report Approval

For

B. E (Computer Engineering)

This project report entitled ***CaseSense Analytics*** by ***Rahul Motwani, Nimish Chidrawar, Malhar Kajale, Mohit Shahdadpuri*** is approved for the degree of ***B.E. Computer Engineering.***

Internal Examiner

External Examiner

Head of the Department

Principal

Date:
Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Rahul Motwani (44)

Nimish Chidrawar (11)

Malhar Kajale (28)

Mohit Shahdadpuri (62)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to **Dr. Sujata Khedkar** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr. (Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement at several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

Chapter No.	Title	Page No.
1	Introduction	11
1.1	Introduction to the project	11
1.2	Motivation for the project	11
1.3	Problem Definition	12
1.4	Existing Systems	12
1.5	Lacuna of the Existing Systems	12
1.6	Relevance of the Project	13
2	Literature Survey	14
A	Brief overview of Literature Survey	14
2.1	Research Papers a. Abstract of the research paper b. Inference drawn from the paper	14
2.3	Inference Drawn	22
2.4	Comparison with the Existing Systems	23
3.	Requirement Gathering for the proposed System	26
3.1	Introduction to Requirement Gathering	26
3.2	Functional Requirements	28
3.3	Non-Functional Requirements	28
3.4	Hardware, Software, Technology and tools utilised	29
3.5	Constraints	30
4.	Proposed Design	31
4.1	Block diagram of the system	31
4.2	Modular design of the system	32
4.3	Detailed Design (Flowchart)	33
4.4	Project Scheduling & Tracking using Timeline / Gantt Chart	35
5.	Implementation of the Proposed System	36

5.1	Methodology employed for development	36
5.2	Algorithms and flowcharts for the respective modules developed	36
5.3	Datasets source and utilisation	42
6.	Testing of the Proposed System	44
6.1	Introduction to testing	44
6.2	Types of tests Considered	44
6.3	Various test case scenarios considered	45
6.4	Inference drawn from the test cases	46
7.	Results and Discussions	48
7.1	Screenshots of User Interface (UI) for the respective module	48
7.2	Performance Evaluation measures	50
7.3	Input Parameters / Features considered	53
7.4	Inference drawn	53
8.	Conclusion	55
8.1	Limitations	55
8.2	Conclusion	55
8.3	Future Scope	55
	References	56
	Appendix	58
1	Paper I	58
a	Paper I	58
b	Plagiarism Report of Paper I	63
c	Project review sheet	64

List of Figures

Fig no.	Heading	Page no.
4.1	Block Diagram of the Proposed System	31
4.2	Modular Diagram	32
4.3.1	Level 0 Data Flow Diagram	33
4.3.2	Level 1 Data Flow Diagram	34
4.3.3	Level 2 Data Flow Diagram	34
4.4	Gantt Chart	35
5.1	Pseudocode for building history for solving general queries	37
5.2	Pseudocode for additional clause generation	38
5.3	Pseudocode for answering questions about the legal document	40
5.4	Pseudocode for generating questions for filling the template	41
6.1	Rent Agreement Template	45
6.2	Prompt for additional clause generation	45
6.3	Results produced by question generation module	46
6.4	Handling invalid JSON LLM responses	46
6.5	Text as sliding content	47
6.6	Enhanced prompt for clause generation	47
6.7	Questions generated by GPT 3.5	47

6.8	Output with valid JSON content due to applied constraints	47
7.1.1	Home Page	48
7.1.2	Legal Document Generation Form Page	48
7.1.3	Document Chat Page	49
7.1.4	Discover New Clauses Page	49
7.1.5	Help Desk Page	50
7.2.1	Assignment and Assumption Agreement Clause-1	51
7.2.2	Assignment and Assumption Agreement Clause-2	51
7.2.3	Non Disclosure Agreement Clause-1	51
7.2.4	Non Disclosure Agreement Clause-2	52
7.2.5	Arbitration Agreement Clause-1	52
7.2.6	Arbitration Agreement Clause-2	52

List of Tables:

Table no.	Heading	Page no.
6.1	Test case scenarios	46
6.2	Inferences drawn from test cases	47

Abstract

The project aims to solve the critical need for accessible legal resources and services in India by leveraging state of the art Natural Language Processing and Generative Pre-trained Transformers. A legal document completion interface is built for the user to upload their document templates and to generate simple to understand questions for document filling assistance. Furthermore, two chatbot interfaces are designed for solving user queries about the document and to explore new clauses and loopholes based on the uploaded document. In order to assure the users about the generated content, explainability is implemented in the form of similarity scores and by providing references to the documents from which the answer is derived by utilizing Retrieval Augmented Generation LLM pipelines. Various LLM models such as Mistral-7B, Mixtral-8x7B and GPT 3.5 have been explored to identify experts for the required tasks.

Chapter 1: Introduction

1.1. Introduction:

Due to limited access to legal resources, a gap exists in drafting legal documents in plain language in the legal landscape. Our solution aims to bridge this gap by introducing an AI-Powered Legal Documentation Assistant.

Over 80% of businesses struggle to access legal resources, mainly due to affordability. We employ state-of-the-art Natural Language Processing (NLP) and Machine Learning (ML) techniques to train our model. This model can convert convoluted legal jargon into plain language, making it accessible to anyone.

Our platform features an extensive library of legal document templates. Users can easily access these templates and customize them on-the-fly by providing specific inputs where required. This process streamlines document creation while ensuring legal accuracy.

Our user-friendly interface ensures that individuals with no legal background can navigate the platform effortlessly. For complex legal challenges, our platform provides users with the option to seek expert advice.

1.2. Motivation:

The project aims to not only empower lawyers, paralegals and legal teams with automation of tedious document preparation but also empowers small businesses and individuals who cannot afford legal resources. Our motivation is rooted in the belief that AI can improve access to justice and has the potential to generate accurate and reliable documentation.

Legal documents, such as contracts, agreements, and pleadings, often employ intricate language and legal jargon that can be daunting for those without a legal background. This language barrier poses a significant obstacle to accessing legal resources and understanding one's legal rights and obligations. Many individuals and small businesses, lacking the financial means to hire legal experts, find themselves at a disadvantage when dealing with legal matters.

This project's importance goes beyond technology since it upholds the values of fairness and inclusivity. We want to democratize access to legal information and equip people and small companies to confidently navigate the legal system by translating legalese into everyday English. Our project aims to build a connection between the legal community and the general public, promoting informed decision-making and minimizing inequalities in access to the justice system.

1.3. Problem Definition:

Legal documentation can be a complicated and time-consuming process, especially for individuals and small businesses who may not have access to legal resources. In addition, the language and jargon used in legal documents can be difficult for non-lawyers to understand, which can lead to errors and misunderstandings.

The objective of this project is to develop an AI-powered solution that can simplify legal documentation for individuals and small businesses in India, by automatically drafting legal documents in plain language and using easy-to-understand terms.

The proposed solution can greatly benefit individuals and small businesses in India, who often face challenges with legal documentation due to limited access to legal resources.

By simplifying legal documentation, this solution can potentially save time, reduce errors, and increase access to justice.

1.4. Existing Systems:

Documate: Documate is a platform that enables users to automate the creation of legal documents using templates and conditional logic. It allows legal professionals to streamline their document creation process by automating repetitive tasks. Users can create customized templates for various legal documents such as contracts, agreements, and legal letters. Documate offers features like document assembly, electronic signature integration, and document management.

HotDocs: HotDocs is a document automation software used primarily by law firms and legal departments to create complex legal documents. It uses a question-and-answer format to gather information from users and then generates customized documents based on predefined templates. HotDocs is known for its flexibility and scalability, allowing users to automate the creation of a wide range of legal documents, from simple contracts to intricate legal agreements. It integrates with various case management systems and document management platforms to streamline the document creation and management process.

1.5. Lacuna of the Existing System:

1. No existing software that provides a comprehensive solution to the problem of generating, sharing, and storing legal documents.
2. No AI based solution is available for customization of generated documents.
3. No explanation is available for the generated document in plain language.
4. Similarly, no system which explains an already existing document as input is available.

- No 24x7 assistant such as chatbot available for improved customer support.

1.6. Relevance of the Project:

The relevance of the project lies in addressing the significant challenges faced by individuals and small businesses in India when dealing with legal documentation. It aims to simplify the legal documentation process using AI-powered technology. Some major goals are as follows

- Mitigating Complexity:** Legal documentation is known for its complexity, which can be a barrier for individuals and small businesses. The project's goal of simplifying clauses in legal documents directly addresses this problem.
- Increased Understanding:** Making legal language more understandable by simplifying clauses can lead to better comprehension of legal agreements. This can help prevent misunderstandings and disputes, contributing to smoother legal transactions.
- Access to Justice:** The project aims to make legal documentation accessible to a wider audience, particularly those who lack access to legal resources. By simplifying legal documentation, the project contributes to improving access to justice for individuals and small businesses who might otherwise struggle with legal matters.
- Error Reduction and Time Efficiency:** Automating document creation can significantly reduce errors and enhance the reliability of legal documents. The solution can streamline the document creation process, saving time for individuals and small businesses who might otherwise spend significant hours navigating legal complexities.

Chapter 2 : Literature Survey

A. Brief Overview of Literature Survey

Our literature survey focused on research papers in the LLM-aided contract understanding and generation domain. We also reviewed some papers on RAG (Retrieval-Augment-Generation), AI in the legal domain and on contract drafting. We also reviewed papers on legal datasets such as CUAD (Contract Understanding Atticus Dataset) used in our project and papers on contract drafting using LEDGAR. Overall, the literature survey throws light on all the aspects pertaining to our project and related work in multiple domains.

B. Related Works

2.1 Research Papers Referred (Abstract and Inference Drawn in own words)

1. Lam, K. Y., Cheng, V. C., & Yeong, Z. K. (2023). Applying large language models for enhancing contract drafting. In Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workspace (LegalAIIA 2023).
 - a. **Abstract:** Experimental results utilizing the LEDGAR dataset demonstrate the utility of Large Language Models (LLMs) in contract drafting. Automated comparison results generated by LLMs serve as valuable hints or recommendations for users to revise and enhance clauses, thereby simplifying the task of contract drafting for legal professionals in an augmented intelligence framework.
 - b. **Inference drawn:** The findings underscore the practical application of LLMs in augmenting contract drafting processes, facilitating more efficient and informed decision-making by legal professionals. By providing automated comparisons and recommendations, LLMs enhance the drafting workflow, potentially reducing the time and effort required for contract creation while maintaining accuracy and quality. This research highlights the transformative potential of augmented intelligence approaches in legal document preparation, leveraging the capabilities of LLMs to streamline complex tasks in the legal domain.
2. Hendrycks, D., Burns, C., Chen, A., & Ball, S. (2021). Cuad: An expert-annotated nlp dataset for legal contract review. arXiv preprint arXiv:2103.06268.
 - a. **Abstract:** Many specialized domains, including law, lack deep learning advancements due to the high cost of expert annotators for large labeled datasets. Addressing this challenge, we introduce the Contract Understanding Atticus Dataset (CUAD) for legal contract review, featuring over 13,000 annotations from legal experts. The dataset tasks models with identifying

crucial contract segments for human review. While Transformer models show promising initial performance, our study highlights the impact of model design and dataset size on performance. Despite progress, significant enhancements are needed. CUAD serves as a valuable research benchmark for the broader NLP community, being one of the few large, specialized NLP datasets annotated by domain experts.

- b. **Inference drawn:** The introduction of CUAD fills a crucial gap in specialized NLP datasets, providing a challenging benchmark for legal contract understanding. The findings underscore the potential of Transformer models in legal tasks, while also indicating areas for improvement. As the only large dataset annotated by legal experts, CUAD has the potential to drive advancements in legal NLP research, facilitating the development of more accurate and efficient AI systems for contract review and analysis.
3. Martin, L., Whitehouse, N., Yiu, S., Catterson, L., & Perera, R. (2024). Better Call GPT, Comparing Large Language Models Against Lawyers. arXiv preprint arXiv:2401.16212.
 - a. **Abstract:** This paper pioneers a comparison between Large Language Models (LLMs) and traditional legal contract reviewers, evaluating accuracy, speed, and cost efficiency. Benchmarking LLMs against Senior Lawyers' ground truth, we find that LLMs can match or exceed human accuracy in identifying legal issues. LLMs complete reviews in seconds, significantly faster than human counterparts, and operate at a fraction of the cost, with a 99.97 percent reduction compared to traditional methods. These findings herald a transformative shift in legal practice, as LLMs promise to revolutionize accessibility and efficiency in legal services. Our research advocates for embracing LLMs to usher in a new era of legal workflows, challenging conventional practices.
 - b. **Inference drawn:** The groundbreaking comparison underscores the potential of LLMs to revolutionize legal contract review, offering superior accuracy, speed, and cost efficiency compared to traditional methods. These results herald a paradigm shift in the legal industry, emphasizing the need to embrace AI technologies to enhance accessibility and effectiveness in legal services. The era of LLM dominance in legal workflows is imminent, prompting a reevaluation of established practices and fostering innovation in legal service delivery.
4. Aggarwal, V., Garimella, A., Srinivasan, B. V., & Jain, R. (2021). CLAUSEREC: A Clause Recommendation Framework for AI-aided Contract Authoring. arXiv preprint arXiv:2110.15794.
 - a. **Abstract:** Contracts, integral to numerous business workflows, have seen limited natural language processing (NLP) research, particularly in generation. This paper introduces the task of clause recommendation to facilitate contract authoring. A two-staged pipeline is proposed:

first, predicting the relevance of a specific clause type for addition in a contract, and then recommending top clauses based on the contract context. BERT is pretrained on an existing clause library and utilized for prediction and recommendation. Various methods, including classification and similarity-based heuristics for relevance prediction, and generation-based approaches for recommendation, are experimented and evaluated across multiple clause types. Analyses of the results highlight the advantages and limitations of different methods in this research domain.

- b. **Inference drawn:** This paper addresses the scarcity of NLP research in contract processing and generation by introducing the task of clause recommendation. The proposed pipeline offers a systematic approach to aid and expedite contract authoring. Through experimentation and evaluation of different methods, the paper sheds light on the efficacy of various techniques in predicting clause relevance and recommending clauses based on context. The findings provide valuable insights into the potential applications and challenges of employing NLP in contract document workflows.
5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.
- a. **Abstract:** While large pre-trained language models excel in fine-tuning for downstream NLP tasks, their ability to access and manipulate knowledge remains limited, particularly on knowledge-intensive tasks where task-specific architectures outperform them. Addressing this gap, we propose a general-purpose fine-tuning recipe for retrieval-augmented generation (RAG) models, which combine pre-trained parametric and non-parametric memory for language generation. Our approach utilizes a pre-trained seq2seq model as parametric memory and a dense vector index of Wikipedia as non-parametric memory, accessed via a pre-trained neural retriever. We compare two RAG formulations and demonstrate their effectiveness on various knowledge-intensive NLP tasks, setting new state-of-the-art results in open domain question answering tasks. Additionally, for language generation tasks, we find that RAG models produce more specific, diverse, and factually accurate language compared to parametric-only seq2seq baselines.
 - b. **Inference drawn:** This paper introduces a novel fine-tuning approach for retrieval-augmented generation (RAG) models, which leverage both pre-trained parametric and non-parametric memory for language generation. By combining these two types of memory, RAG models achieve superior performance on knowledge-intensive NLP tasks, surpassing both parametric

seq2seq models and task-specific architectures. Furthermore, RAG models demonstrate enhanced language generation capabilities, generating more specific, diverse, and factual language compared to existing baselines. This research highlights the potential of integrating non-parametric memory into pre-trained language models to enhance their knowledge access and manipulation abilities for various NLP tasks.

6. Zödi, Zsolt. (2019). The limits of plain legal language: understanding the comprehensible style in law. *International Journal of Law in Context*. 2019. 246–262. 10.1017/S1744552319000260.
 - a. **Abstract:** The paper explores the difficulty of enhancing the clarity of legal texts, commonly regarded as a linguistic concern by advocates of plain-legal-language. It argues that despite attempts to simplify legal language, understanding extends beyond mere vocabulary and grammar, as supported by corpus linguistic studies.
 - b. **Inference drawn:** The complexity of legal language is attributed not only to linguistic intricacies but also to the systemic and interpretive nature of law, alongside the increasing significance of technical regulations, as highlighted by the proposed framework delineating three pragmatic scenarios where comprehension challenges arise.
7. Queudot M, Charton É, Meurs M-J. Improving Access to Justice with Legal Chatbots. *Stats*. 2020; 3(3):356-375. <https://doi.org/10.3390/stats3030023>
 - a. **Abstract:** Limited affordability of legal services renders marginalized individuals devoid of professional guidance, necessitating self-representation; consequently, two chatbots have been created—one tailored to immigration issues utilizing Government of Canada data, while the other educates bank employees about job-related legalities, employing diverse techniques and algorithms.
 - b. **Inference drawn:** Recognizing the financial barrier to legal assistance, the development of specialized chatbots signifies an innovative approach to democratizing access to legal information and representation, with the availability of the immigration chatbot for research purposes enhancing its potential impact on addressing societal inequalities.
8. Amato F, Fonisto M, Giacalone M, Sansone C. An Intelligent Conversational Agent for the Legal Domain. *Information*. 2023; 14(6):307. <https://doi.org/10.3390/info14060307>
 - a. **Abstract:** CREA2 is an AI-powered conversational agent specialized for the legal domain, offering users natural language interaction and comprehensive legal support, including guidance on legal procedures, terminology comprehension, document drafting, and dispute resolution

within the European Union.

- b. **Inference drawn:** The development of CREA2 reflects a significant advancement in leveraging AI to address multifaceted legal needs, potentially revolutionizing legal assistance by offering accessible, personalized, and efficient solutions for various legal challenges.
9. Hricik, D., Morgan, A.-L. S., & Williams, K. H. (2018). Ethics of Using Artificial Intelligence to Augment Drafting Legal Documents. *Texas A&M Journal of Property Law*, 4, 465. <https://doi.org/10.37419/JPL.V4.I5.3>
- a. **Abstract:** The article examines the profound influence of technology, particularly AI and automation, on the legal sector, emphasizing their potential to augment efficiency and reduce costs. However, it underscores the crucial responsibility of lawyers to ensure that the technology employed meets competency standards and ethical obligations, urging careful assessment of automated services' competence and adherence to ethical rules, such as confidentiality and conflicts of interest.
 - b. **Inference drawn:** As technology increasingly integrates into legal practice, this article underscores the necessity for lawyers to navigate the ethical implications, emphasizing the critical role of vigilance and assessment in safeguarding professional standards and client interests amidst technological advancements.
10. Zhongxiang Sun. (2023). A Short Survey of Viewing Large Language Models in Legal Aspect.
- a. **Abstract:** The survey investigates the impact of Large Language Models (LLMs) across diverse domains, with a specific focus on the legal field. It analyzes LLM applications, legal challenges, and available data resources, culminating in a discussion on future directions. The paper aims to offer a comprehensive overview of LLMs in law, emphasizing their potential advantages and hurdles.
 - b. **Inference drawn:** By examining the breadth of LLM applications and associated challenges in the legal domain, this survey underscores the need for a nuanced understanding of LLMs' role in shaping legal practices, while also highlighting opportunities for leveraging these models to address complex legal issues effectively in the future.
11. Hafner, Carole & Lauritsen, Marc. (2007). Extending the power of automated legal drafting technology.. 59-68.
- a. **Abstract:** The paper scrutinizes deficiencies in existing legal drafting software and presents a novel system architecture aiming to enhance expressiveness and user experience. Through a

practical scenario involving a customer service representative addressing a consumer complaint, it demonstrates the identified issues and the proposed architectural design.

- b. **Inference drawn:** By delineating the shortcomings of current legal drafting software and proposing an innovative architecture, this paper underscores the importance of addressing usability challenges in legal technology, ultimately aiming to streamline and improve the drafting process for legal professionals.

12. Marković, M., Gostojić, S. Legal document assembly system for introducing law students with legal drafting. Artif Intell Law (2022). <https://doi.org/10.1007/s10506-022-09339-2>

- a. **Abstract:** The paper presents a method for instructing law students in legal document drafting utilizing machine-readable legal knowledge, comprising enacted legislation, document templates, and assembly instructions. It introduces the LEDAS (LEgal Document Assembly System), an interactive platform guiding users through document assembly, elucidating the relationship between input data and document claims, and serving as an educational resource for honing drafting skills.
- b. **Inference drawn:** Through the introduction of LEDAS, this paper not only addresses the pedagogical challenges in teaching legal drafting but also demonstrates the potential of leveraging technology to provide interactive and hands-on learning experiences, enhancing students' proficiency in legal document composition.

13. Lankester, R. (2018). Implementing Document Automation: Benefits and Considerations for the Knowledge Professional. Legal Information Management, 18(2), 93-97.
doi:10.1017/S1472669618000208

- a. **Abstract:** Targeting knowledge professionals in law firms and legal departments, the article elucidates document automation's definition, benefits for knowledge management, and its potential to enhance internal knowledge development and reshape legal service delivery. Focused on project implementation, resource allocation, and document preparation, it emphasizes the importance of proper business engagement for successful adoption.
- b. **Inference drawn:** By emphasizing the strategic significance of document automation beyond mere efficiency gains, this article underscores its transformative potential in fostering internal knowledge growth, enhancing client collaboration, and optimizing legal service delivery, stressing the critical role of proactive business engagement in achieving these objectives.

14. Waltl, B., Landthaler, J., Scepankova, E., Matthes, F., Geiger, T. H. O. M. A. S., Stocker, C., & Schneider, C. (2017). Automated extraction of semantic information from german legal documents. In IRIS: Internationales Rechtsinformatik Symposium.
- Abstract:** The paper explores two real-world use cases conducted within an established data science environment: (1) the computer-supported extraction of dispute years in German cases, and (2) the extraction of legal definitions and determination of contexts for legal terminology in judgments.
 - Inference drawn:** Through these investigations, the paper illustrates the practical application of data science techniques in legal contexts, highlighting their potential to automate labor-intensive tasks such as information extraction and terminology analysis, thereby enhancing efficiency and accuracy in legal research and analysis processes.
15. I. T. Imam, Y. Arifat, K. S. Alam and S. A. Shahriyar, "DOC-BLOCK: A Blockchain Based Authentication System for Digital Documents," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 1262-1267, doi: 10.1109/ICICV50876.2021.9388428.
- Abstract:** This research paper tackles the issue of document forgery in the modern technology-driven era, stressing the growing necessity of authenticating crucial documents like banking records, government papers, transactions, and educational certificates. Recognizing the complexity and time-consuming nature of traditional verification methods, the paper introduces a decentralized web application leveraging Ethereum blockchain and P2P cloud storage for digital document authentication.
 - Inference drawn:** By proposing a decentralized solution using blockchain technology, the paper addresses the pressing need for secure and efficient document verification methods in today's digital landscape, offering potential enhancements in reliability, transparency, and accessibility compared to traditional approaches.
16. Schweighofer, E. (2010). Semantic Indexing of Legal Documents. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds) Semantic Processing of Legal Texts. Lecture Notes in Computer Science(), vol 6036. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12837-0_9
- Abstract:** Automated semantic indexing presents a promising solution to address the inadequate recall of legal information systems, leveraging the powerful tools offered by the semantic web for markup and ontological representation. However, its application in legal contexts remains limited due to issues such as improper knowledge structuring and the absence of automated knowledge acquisition. This paper outlines the current state of semantic indexing and proposes

the development of a dynamic electronic legal commentary.

- b. **Inference drawn:** By highlighting the potential of automated semantic indexing in enhancing legal information retrieval, this paper underscores the importance of addressing existing challenges such as knowledge structuring and acquisition to facilitate its widespread adoption in legal applications. The proposal for a dynamic electronic legal commentary suggests a practical avenue for leveraging semantic indexing to augment legal research and analysis processes.
17. Villata, S. (2020). Sentence embeddings and high-speed similarity search for fast computer assisted annotation of legal documents. In Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020 (Vol. 334, p. 164). IOS Press.
- a. **Abstract:** The research paper delves into the hurdles of machine learning-based systems aiding legal tasks, heavily reliant on human annotation of language in legal documents. Traditionally sequential, such annotations are laborious and costly. The study introduces a proof-of-concept for "laterally" annotating sentences, potentially mitigating these challenges.
 - b. **Inference drawn:** By proposing a lateral annotation method, the paper addresses the time-consuming nature of traditional sequential annotation processes, offering a potential avenue for improving efficiency and reducing costs in machine learning-based legal systems. This approach could potentially enhance the scalability and practicality of such systems, paving the way for more effective utilization in legal tasks.
18. Jobin, A., Ienca, M. & Vayena, E. The global landscape of AI ethics guidelines. *Nat Mach Intell* 1, 389–399 (2019). <https://doi.org/10.1038/s42256-019-0088-2>
- a. **Abstract:** In recent years, various organizations, encompassing private firms, research institutions, and public sectors, have released ethical principles and guidelines for artificial intelligence (AI). This study scrutinizes these documents to uncover a burgeoning global consensus on ethical AI, identifying five core principles: transparency, justice, non-maleficence, responsibility, and privacy. The findings emphasize the imperative of integrating ethical considerations and robust implementation strategies into guideline development endeavors.
 - b. **Inference drawn:** By highlighting the emergence of a global consensus on ethical AI principles, this study underscores the growing recognition of the significance of ethical frameworks in AI development and deployment. The identification of these core principles provides a foundational basis for guiding AI-related policies and practices, aiming to ensure

- responsible and equitable AI innovation and utilization across diverse sectors and stakeholders.
19. Ng, J., Haller, E., & Murray, A. (2022). The ethical chatbot: A viable solution to socio-legal issues. *Alternative Law Journal*, 47(4), 308-313.

- a. **Abstract:** Chatbots, renowned for their technical versatility, extend beyond legal guidance to address social disconnectedness issues. This article surveys diverse socio-legal chatbots, emphasizing their utilization by organizations offering legal services, particularly community legal services. It underscores chatbots' potential to generate social impact while upholding legal ethics and AI principles.
- b. **Inference drawn:** By recognizing chatbots' dual role in providing legal guidance and tackling social isolation, this article underscores their broader societal relevance beyond legal contexts. The emphasis on ethical adherence underscores the importance of integrating ethical considerations into the development and deployment of socio-legal chatbots, ensuring they contribute positively to both legal and social spheres while upholding ethical standards.

2.2 Inference drawn

The literature survey highlights a profound shift within the legal landscape driven by improvements in artificial intelligence (AI) and natural language processing (NLP). Large Language Models (LLMs) have emerged as powerful tools with transformative ability throughout diverse legal responsibilities, from contract drafting and overview to semantic indexing. These models offer excellent accuracy, pace, and cost efficiency, challenging traditional techniques and paving the way for augmented intelligence methods in legal workflows. The introduction of specialized datasets like CUAD and revolutionary frameworks like clause advice further catalyzes research and development in legal NLP, imparting valuable benchmarks and systematic methodologies for enhancing legal file methods.

Moreover, the moral dimensions of AI within the legal domain have garnered increasing interest, with worldwide efforts aimed at establishing moral suggestions for AI development and deployment. The convergence around center moral principles including transparency, justice, and responsibility underscores the importance of responsible AI practices to ensure equitable and accountable use of AI technologies. Overall, the literature survey indicates a pivotal moment in the legal domain , where AI-powered solutions offer unparalleled opportunities for performance, accessibility, and social effect, albeit with the vitality of ethical considerations at the leading edge of innovation and implementation.

2.3 Comparison with the existing systems

Some existing systems in place to deal with this problem of legal document generation are as follows

1. Contract Express: ([Link](#))

Thomson Reuters's Contract Express is a well-known platform for document automation and contract creation. Legal practitioners can use it to create templates with specified clauses and variables, which automates the drafting of contracts and other legal documents.

2. HotDocs: ([Link](#))

This document automation programme is used by a variety of sectors, including the legal one. By putting together predetermined clauses and templates based on user inputs, it enables the development of dynamic and interactive documents.

3. LegalZoom: ([Link](#))

LegalZoom is an online legal services platform that provides document generation services for individuals and small businesses. Users can create legal documents such as wills, contracts, and business formation documents using their templates.

4. PandaDoc: ([Link](#))

While not exclusively a legal document generation platform, PandaDoc offers document automation features that can be useful for legal professionals. It allows for the creation of customizable proposals, contracts, and other documents.

Existing systems for legal document generation, such as Contract Express, HotDocs, LegalZoom, and PandaDoc, offer valuable tools for legal practitioners and individuals. However, there are significant gaps in these systems compared to the proposed area of work.

1. Comprehensive Solution:

- Existing Systems: The mentioned systems focus primarily on document automation and template-based generation. While they are useful for specific tasks, they do not provide a comprehensive solution that covers the entire legal document lifecycle, including customization, explanation, and integration with expert advice.
- Proposed Area of Work: The proposed area of work aims to offer an all-encompassing solution that addresses not only document generation but also customization, plain language explanations, integration with existing legal resources, and access to expert advice. It fills the gaps left by existing systems by providing a holistic approach to legal document management.

2. AI-Based Customization:

- Existing Systems: Most existing systems rely on user-defined templates and inputs, requiring manual adjustments for customization. They lack AI-driven capabilities to dynamically adapt documents based on user requirements.

- Proposed Area of Work: The proposed system leverages AI to automatically customize generated documents, ensuring they align precisely with user needs. This feature streamlines the document creation process, saving time and enhancing accuracy.

3. Plain Language Explanations:

- Existing Systems: None of the existing systems offer plain language explanations for the generated documents. Users may struggle to understand legal jargon and implications.
- Proposed Area of Work: The proposed system bridges this gap by providing clear and concise plain language explanations for document content. This empowers users to comprehend the legal aspects fully, promoting transparency and understanding.

4. Document Explanation System:

- Existing Systems: Existing systems do not include features for explaining existing legal documents. Users may possess documents they do not fully understand, leaving them vulnerable to legal risks.
- Proposed Area of Work: The proposed system introduces a document explanation feature, allowing users to decipher the content of existing documents. This enhances legal literacy and enables informed decision-making.

5. 24/7 Customer Support:

- Existing Systems: Customer support in existing systems may not be available 24/7, leading to potential delays in addressing user inquiries and issues.
- Proposed Area of Work: The proposed system includes a 24/7 chatbot or virtual assistant, ensuring round-the-clock assistance. This enhances user support, provides timely responses, and improves the overall user experience.

To summarize, while existing systems offer valuable document automation features, our work goes beyond by providing a comprehensive, AI-driven solution that addresses customization, plain language explanations, document explanation, and 24/7 support. It aims to bridge the existing gaps in legal document management, making legal resources more accessible and user-friendly.

Chapter 3: Requirements

The extensive set of requirements that guide the project's development are outlined in this chapter. The design, development, and deployment phases of the project are heavily influenced by these criteria. The chapter starts out by classifying the needs into functional and non-functional groups, then looks at the necessary hardware and software. This section further clarifies the limitations that can affect the project's delivery and execution. When taken as a whole, these conditions and restrictions offer a precise road map for the project's successful conclusion and link it with the anticipated goals and results.

3.1 Proposed Model

The proposed model outlines the architecture and functionalities of an AI-powered legal document assistant, aimed at streamlining the generation, understanding, and validation of legal documents. This model explains key components and features aligned with the methodology.

1. Document Generation System

- Document Ingestion system: Take the document template input from the user in word/ docx/ jpg format which forms a starting template.
- Document Extension System: Develop a modular template completion/ extension system comprising suggestions of the legal clauses, variables based on the existing uploaded template.
- Custom Input Form Generation System: Design intuitive, web-based forms tailored for specific legal document types, ensuring user-friendly data entry for the template document.
- Document Assembly Logic: Implement intelligent logic to dynamically assemble documents based on user input.

2. Chatbot Development for Custom Clauses

- Conversational Interface: Create a user-centric chatbot interface with natural language understanding capabilities.
- Large Language Models: Utilize state-of-the-art Large Language Models (e.g. MixtralAI/ GPT 3.5) to interpret user intents and context for clause customization.
- Knowledge Base and ML: Incorporate a knowledge base of legal clauses and ML models for generating custom clauses, providing real-time assistance during document creation.

3. Legal Clause Explanation System

- Clause Explanation System: Enables users to ask for explanation of specific clauses through a user-friendly interface, for which the explanation will be derived from the context of the document combined with the existing knowledge base using RAG.
- Legal Knowledge Base: Build an extensive legal knowledge base comprising explanations, precedents, and legal insights.
- Caching System: Implement a caching mechanism to ensure swift retrieval of clause explanations, optimizing user experience.

4. Document Explanation for User-Uploaded Documents

- Document Upload Feature: Integrate a secure document upload feature, allowing users to submit legal documents not generated by the system.
- NLP and Clause Extraction: Employ advanced NLP techniques to parse and extract relevant clauses, terms, and context from uploaded documents.
- Intelligent and Reliable Explanation: Leverage the Legal Clause Explanation System to provide comprehensive explanations for clauses and terms found in uploaded documents while ensuring strict compliance with data privacy regulations, also showing the source of explanation to the user for greater reliability.
- Contextual Explanation System: The explanation of the document is given with the contextual understanding based on the previously asked questions.

5. Legal Support Integration

- Access to Legal Professionals: Seamlessly integrate access to qualified legal professionals or lawyers for consultations, offering expert guidance.
- Secure Communication: Establish secure and private communication channels, safeguarding sensitive legal discussions between users and legal experts.

6. Blockchain-Based Secure Storage

- Blockchain Implementation: Develop a blockchain-based system for the secure storage of document hashes, ensuring tamper-proof records.
- Hash Generation and Verification: Generate unique document hashes and timestamp them on the blockchain for authenticity and integrity verification.
- User-Friendly Validation: Implement a user-friendly validation mechanism, allowing users to easily verify the authenticity of their documents.

3.2 Functional Requirements

1. The system must guide users in creating their own legal documents easily.
2. The system must ingest user's template documents of different types of legal documents.
3. The system should build an input system to fill template documents dynamically.
4. The input system must be reliable and error-free.
5. The input system should provide a good user experience.
6. The system must provide a chatbot to add custom clauses.
7. The chatbot must utilize machine learning to generate custom clauses.
8. The system should suggest some clause additions with respect to the input document.
9. The system must generate accurate documents from user inputs.
10. The system must provide easy-to-understand explanations to legal clauses.
11. The system must serve requests quickly using caching.
12. The cache should be updated regularly based on database changes.
13. The system must allow users to sign up and log in.
14. The system must securely store and retrieve documents based on the authenticated user requests.
15. The system must store user documents logically separately in a database.
16. The web application created should be responsive to various screen sizes.
17. The web application should be mobile friendly.
18. The web application should provide global service.
19. The web application should handle network errors.
20. The web service must be RESTful.
21. The input document should be checked for vulnerabilities using sentence similarity with valid documents.

3.3. Non-Functional Requirements

1. The user interface should be intuitive, user-friendly, and accessible to individuals with varying levels of technical expertise.
2. The system should provide clear instructions and guidance to users throughout the document creation process.

3. The document generation process should be efficient, ensuring quick turnaround times for users.
4. The system should be capable of handling multiple concurrent users without significant degradation in performance.
5. The solution should be highly reliable, minimizing downtime and ensuring that users can access the platform whenever needed.
6. Automated document generation should be accurate and consistent, reducing the likelihood of errors.
7. The system should be scalable to accommodate a growing number of users and increasing data volumes.
8. It should handle a variety of legal document types and complexities as users' needs evolve.
9. The solution should allow users to customize generated documents according to their specific requirements.
10. Chatbot for generation of custom clauses should be used to insert those clauses at appropriate positions in the document.
11. The system should be designed to prioritize ethical considerations, including fairness, transparency, and accountability in document generation.
12. Safeguards should be in place to prevent bias in language and contents.
13. The solution should comply with relevant legal standards and regulations governing the creation and handling of legal documents and should be legally accurate.

3.4. Hardware, software, technology and tools used

Hardware Requirements

- 16 GB RAM
- 512 GB of storage
- i5 or i7, 6th generation or higher
- 12 GB GPU or higher

Software Requirements

- Update web browser
- FastAPI
- Next.JS v13.5.2
- React-to-pdf v1.0.1

Technology and Tools used

- In the project the web application frontend is built using Next.JS and the backend application programming interface (API) for the same web application is built using FastAPI.
- To provide 24x7 automated support to the user to solve their queries we allow the user to

interact with a chatbot. The chatbot is supported with an open source LLM model known as MistralAI-Instruct which was obtained from HuggingFace Hub.

- The backend API provides an interface to the document generation from user provided templates and the chat with documents features. It deals with accessing and utilizing Large Language models (LLMs) provided by OpenAI and Mixtral-8x7B (46.7B).

3.5. Constraints of working

1. It is important that the generated documents should be legally accurate. The variance of documents amongst countries and jurisdictions also poses a key constraint. To validate the output, the source of explanation is shown to the user where essential.
2. Building a Retrieval Augment Generation model for customization of documents while ensuring legal compliance and accuracy is a cumbersome task.
3. The generated response by the LLMs should be in JSON format to ensure the reliability of utilizing this response in the front end web application.
4. Creating such a system requires careful consideration of ethical issues, such as biases in document generation, handling sensitive information, and ensuring that users are aware of the system's limitations.

Chapter 4: Proposed Design

4.1. Block Diagram of the proposed system:

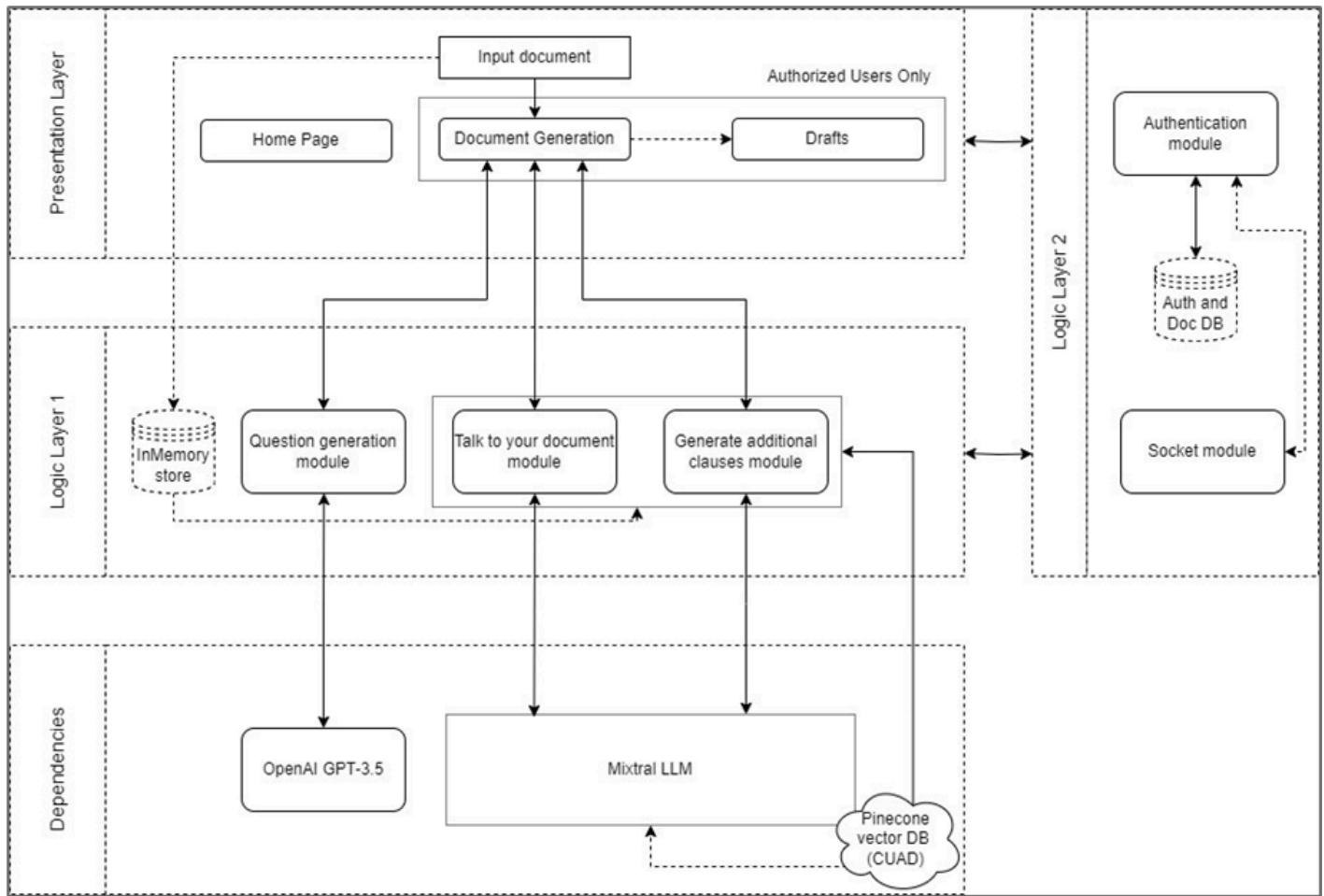


Fig 4.1: Block Diagram of the Proposed System

There are three layers in the software architecture, they are as follows:

1. **Presentation Layer:** This layer comprises the user interface, responsive and user-centric, which the user will interact with to perform the functional tasks like Generating a legal document, getting its explanation, adding additional clauses, etc. There are three key areas on the frontend, the Home Page, Document Generation page and the Helpdesk page.
2. **Logic Layer:** It covers the backend technologies and the logic incorporated by the layer to perform the functional tasks requested by the presentation layer.
 - a. **Logic Layer 1:** It achieves in the execution of the three significant areas of Question Generation, talking with the document and the additional clause generation for the system. The layer acts as a middle interface for ingesting the user input, pre-processing the same and transforming it in an appropriate format which is then sent to the dependencies for further processing. This layer uses the in-memory store to store the input and the output temporarily for the user operations till the user interaction is completed.

- b. Logic Layer 2: It does the critical and mandatory tasks of authentication and authorization of the users so that they can interact with their own account with the system and maintain a degree of personalization for themselves.
- 3. Dependencies: It consists of the state of the art third party resources which we outsource via their APIs to leverage them for conducting the functional requirements in a most possible accurate manner like Mixtral LLM which is a cutting-edge natural language processing model designed to enhance conversational experiences through its advanced understanding of context and GPT 3.5 that excels in understanding and generating human-like text across a wide range of tasks, leveraging massive amounts of data and advanced deep learning techniques for improved performance.

4.2. Modular diagram of the system:

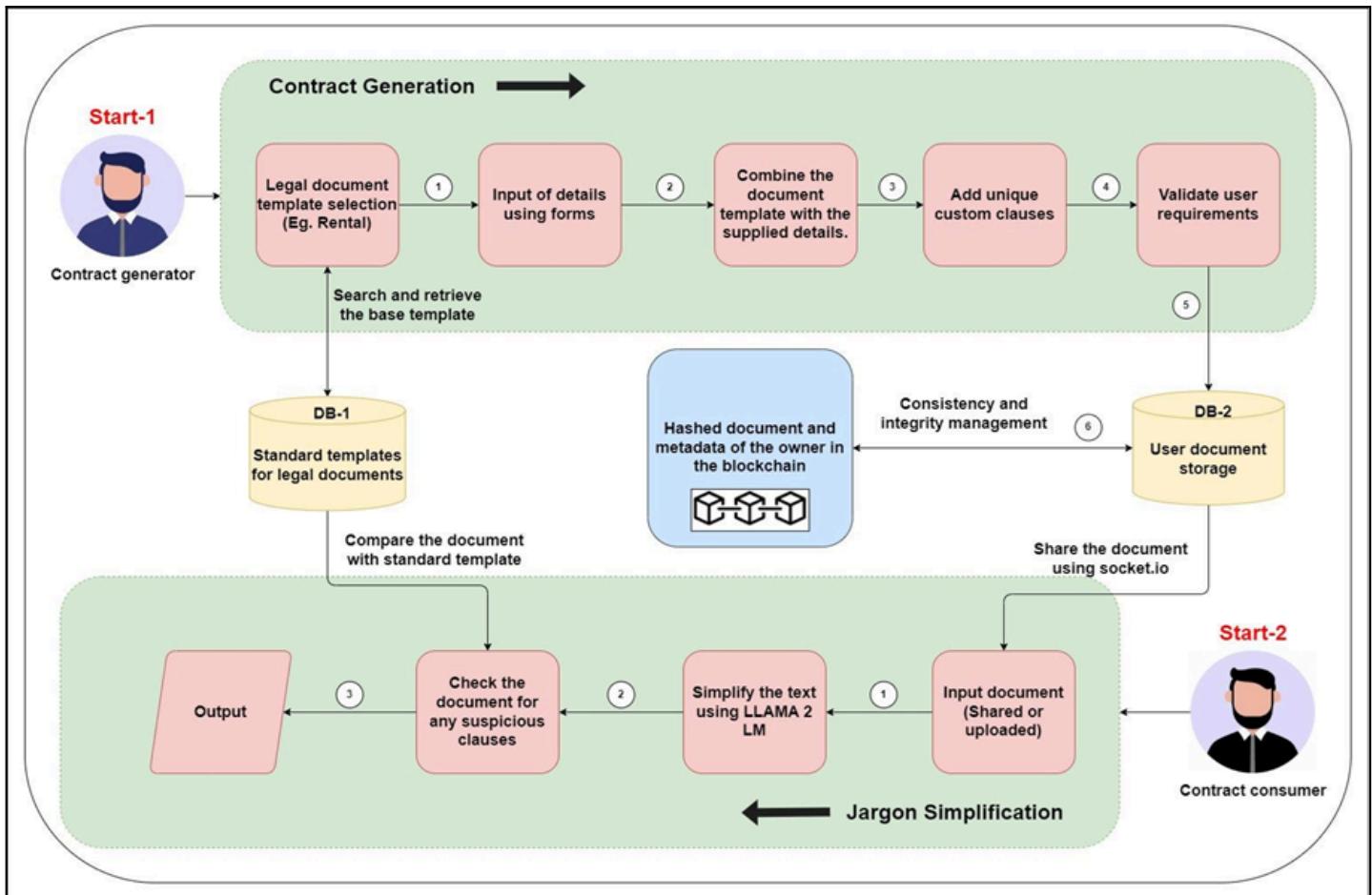


Fig 4.2: Modular Diagram

The diagram in Fig 4.2 refers to the modular diagram of AI-powered Legal Documentation Assistant whose main steps are explained below:

1. Suppose user 1 is the user which generates a contract. He or she will first select the type of the document.
2. Our system will fetch the base template of the document from database 1.
3. Then the user will be prompted to enter his or her information which will be combined to fill the

base template.

4. After the generation of the document the user will be allowed to add any custom clauses into it.
5. Then the document is checked whether it validates all the user requirements by showing the plaintext version to the user. If yes, it is stored into database 2. And the document hash and owner metadata is stored into the blockchain.
6. Also, the document can be shared to the document consumer directly via our platform. In that case, all owners metadata is stored along with document hash into the block.
7. The user 2 can receive the shared document or can even upload his or her document to get the plaintext version of any clauses or can also leverage our system to check for any non ethical clauses by comparing with base template and CUAD dataset.

4.3. Detailed Design:



Fig 4.3.1. Level 0 Data Flow Diagram

The diagram in Fig 4.3.1. represents the level 0 of the Data Flow Diagram. It shows the highest level implementation of the project. There are two external entities in the project viz. Contract Drafter and Contract Signatory. The Contract Drafter interacts with the system to generate the required legal documents by providing the inputs as per the selected document type. The Contract signatory is the recipient of the shared document or is the uploader a new document.

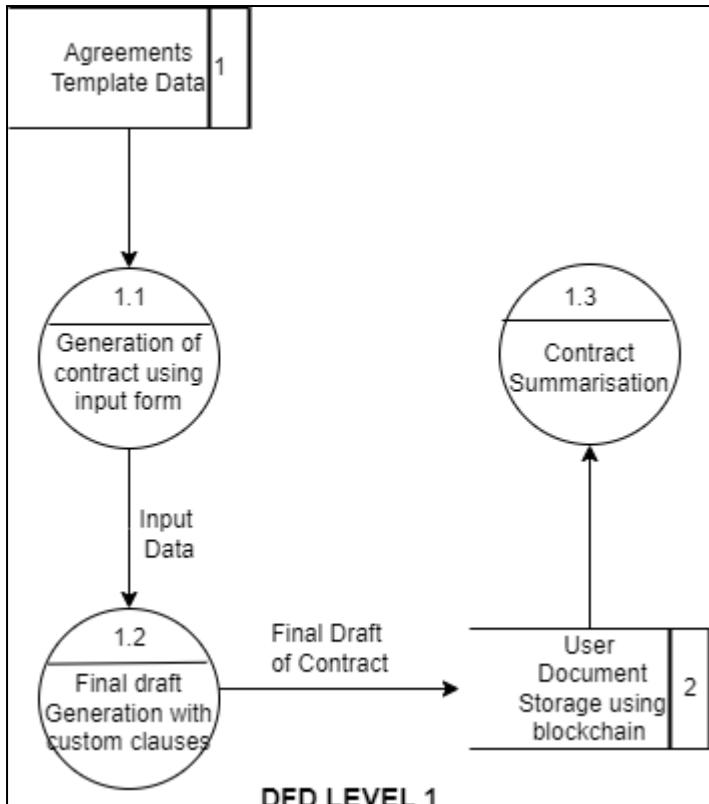


Fig 4.3.2 Level 1 Data Flow Diagram

The diagram in Fig 4.3.2 is the Level 1 Data Flow Diagram which demonstrates the processes in the Level 0 diagram in more detail. A storage unit is described for storing the legal document templates and the created legal documents using the project.

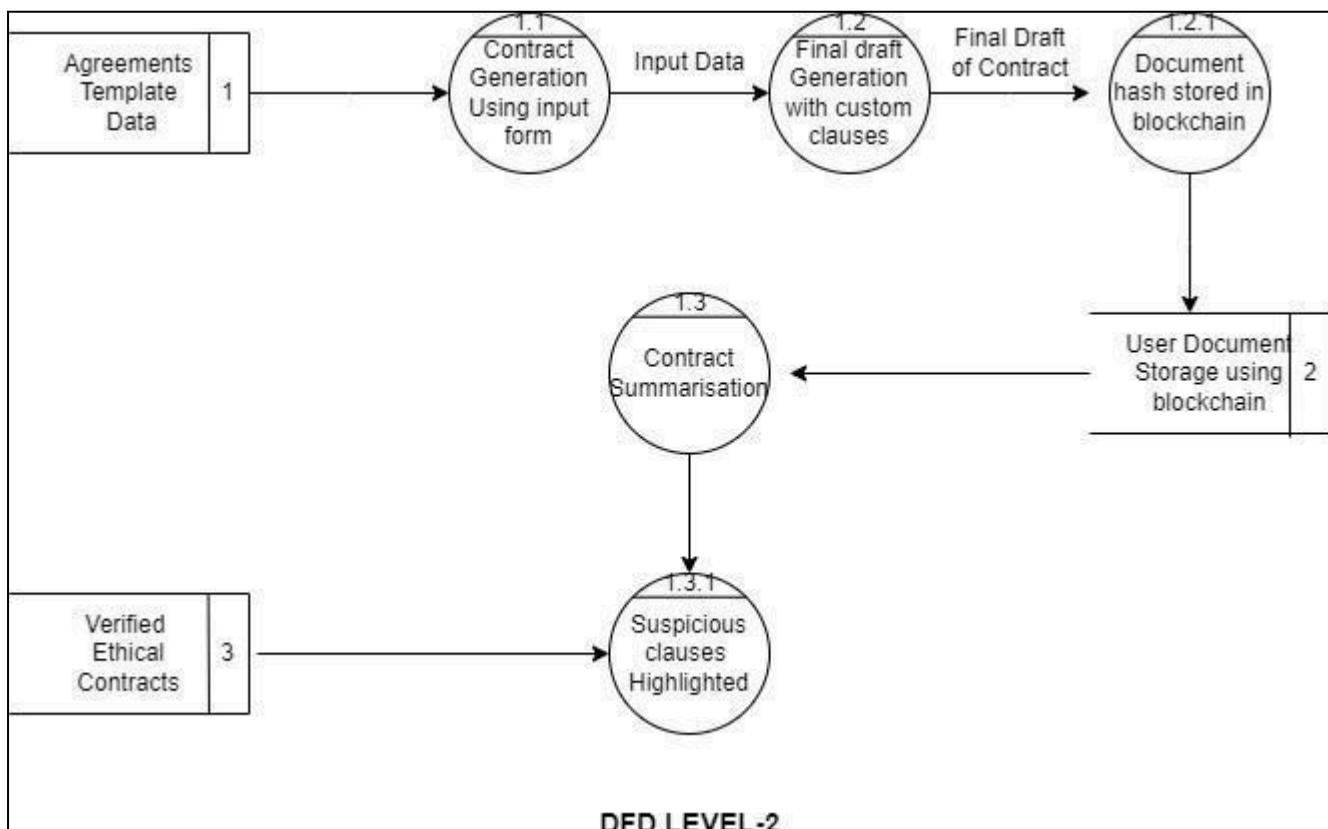


Fig 4.3.3. Level 2 Data Flow Diagram

The diagram in Fig 4.3.3 is a level 2 Data Flow Diagram which simplifies two of the processes in level 1 into child processes which gives a further more detailed and simplified data flow of AI powered legal Document Assistant. A data store with verified ethical contracts is described which helps in highlighting suspicious clauses. The level 2 DFD also included the process for storage of legal documents using blockchain.

4.4. Project Scheduling & Tracking using Time line / Gantt Chart:



Fig 4.4 : Gantt chart

The Gantt chart of our project shows we worked for the whole semester 7 to define the project objectives in which the scope and the requirements were finalized. This was backed by a solid research of existing solutions in the market in this area. Also, the designing and planning of the project was done in this period which gave smooth initial workflow for the project and allowed to start the work parallelly by all the team members. Also, a major part of the frontend was developed before semester 7 ended combined with a partial backend.

The next semester we worked on the major backend part where the Fast API was developed which connected the frontend with the Pine Cone Vector Database and the third party LLMs hosted by openAI like GPT 3.5 and Mixtral LLM. After the integration was completed. The testing iterations commenced.

Chapter 5: Implementation of the Proposed System

5.1 Methodology employed for development:

1. Creation of a document generation system which takes user document template input through custom forms and generates questions for filling the document.
2. Development of a chatbot which can handle generation of custom clauses based on user input.
3. Integrating and fine-tuning a LLM to solve general law and rights related user queries.
4. Building a document explanation system which takes a legal document as input from the user.
5. Providing legal support based on the complexity of the problem through lawyers.
6. Creating a blockchain based system for secure storage of contract hashes for easy validation of authenticity of contract.

5.2. Algorithms and Flowcharts for the respective modules developed

1. Algorithm description for solving general queries using MistralAI:

a. Pseudocode

```
const modelUrl = 'https://api-inference.huggingface.co/models/mistralai/Mistral-7B-Instruct-v0.1';

async function formatPrompt(message, history) {
    let prompt = "<s>";
    for (const [userPrompt, botResponse] of history) {
        prompt += `[INST] ${userPrompt} [/INST] ${botResponse}</s> `;
    }
    prompt += `[INST] ${message} [/INST]`;
    return prompt;
}

async function getHelpDeskResponse(prompt, history, temperature = 0.9, maxNewTokens = 256, topP = 0.95, repetitionPenalty = 1.0) {
    temperature = parseFloat(temperature);
    if (temperature < 1e-2) {
        temperature = 1e-2;
    }
    topP = parseFloat(topP);

    const generateParams = {
        temperature,
        max_new_tokens: maxNewTokens,
        top_p: topP,
        repetition_penalty: repetitionPenalty,
        do_sample: true,
        seed: 42,
    };

    const formattedPrompt = await formatPrompt(prompt, history);

    let output = "";
```

```

try {
  const response = await fetch(modelUrl, {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.NEXT_PUBLIC_HUGGINGFACE_API_KEY}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      inputs: formattedPrompt,
      parameters: generateParams
    })
  });

  if (!response.ok) {
    return {
      output: 'Sorry there was an error. Please try again after a few seconds',
      updatedHistory: history
    }
    // throw new Error(`HTTP error! Status: ${response.status}`);
  }

  const responseData = await response.json();
  output = responseData[0].generated_text.split('[/INST] ').pop()
}

```

```

history.push([prompt, output])

return {output, updatedHistory: history};
} catch (error) {
  console.error(error);
  return {
    output: 'Sorry there was a network error. Please check your connection and try again after a few seconds',
    updatedHistory: history
  }
}

```

Fig 5.1. Pseudocode for building history for solving general queries

- b. Steps
 - i. Define the Hugging Face model URL (modelUrl) that points to the model you want to use for text generation.
 - ii. Create a function (formatPrompt) to format the user's prompt along with conversation history.
 - iii. Create a function (generate) to generate text based on the given prompt, history, and optional generation parameters.
 - iv. Parse and validate the optional generation parameters, ensuring that temperature is within bounds.
 - v. Create the generateParams object with the specified parameters for text generation.
 - vi. Format the input prompt with history using the formatPrompt function.
 - vii. Initialize an empty string for the output text.
 - viii. Try to send a POST request to the Hugging Face API with the formatted prompt and generation parameters.
 - ix. Check if the response is successful, and if not, handle the error.
 - x. Parse the response JSON and extract the generated text if it exists.
 - xi. Return the generated text as the output.

2. Algorithm description of Additional Clause Generation:

a. Pseudocode

```
def get_additional_legal_clauses(documentType, legalClauses):

    query = f"Given a {documentType} which has the following clauses {legalClauses}.
    Return additional clauses that can be added to the legal agreement."

    json_response = query_pipeline.run(query=query, params={"Retriever" : {"top_k": 5}, "debug": True})

    for component in json_response['_debug']:
        print(component, json_response['_debug'][component]['exec_time_ms'])

    answers = json_response['answers']
    documents = json_response['documents']

    for ans in answers:
        answer = ans.answer
        break

    document_names = []

    for document in documents:
        document_names.append(document.id)

    return answer, document_names
```

Fig 5.2. Pseudocode for additional clause generation

b. Steps

- i. Define the input parameters: `documentType` (string) and `legalClauses` (list of strings).
- ii. Construct the query string by formatting the `documentType` and `legalClauses` into a natural language query.
- iii. Call the `query_pipeline.run` function with the constructed query and specified parameters for retrieval (top_k=5) and debugging (debug=True).
- iv. Print the execution time for each component involved in the query processing.
- v. Extract the list of answers and documents from the JSON response.
- vi. Retrieve the first answer from the list of answers.
- vii. Initialize an empty list to store the document names.
- viii. Iterate through the list of documents and append each document's ID to the document_names list.
- ix. Return the retrieved answer and the list of document_names.

3. Algorithm description for answering questions about the legal document :

a. Pseudocode

```

def compute_sim(reference_texts, generated_response):
    reference_text = ' '.join(reference_texts)

    docs=[reference_text, generated_response]

    # Create TFidfVectorizer
    tfidf= TfidfVectorizer()

    # Fit and transform the documents
    tfidf_vector = tfidf.fit_transform(docs)

    # Compute cosine similarity
    cosine_sim=cosine_similarity(tfidf_vector, tfidf_vector)

    # Print the cosine similarity
    return cosine_sim[0][1]

```

```

def chat_with_rag_pipeline(query, chat_pipeline, summary="", legal_clauses=""):

    json_response = chat_pipeline.run(query=query, params={"Retriever" : {"top_k": 5}, "debug": True}, meta={'summary': legal_clauses+summary})

    for component in json_response['_debug']:
        print(component, json_response['_debug'][component]['exec_time_ms'])

    answers = json_response['answers']
    documents = json_response['documents']

    for ans in answers:
        answer = ans.answer
        break

    document_names = []
    document_content = []

    for document in documents:
        document_names.append(document.id)
        document_content.append(document.content)

    document_content.append(summary)
    # print(documents)

    docs = [Document(f"{summary}"), Document(f"{answer}")]

```

```

async def chat_doc(summary: Annotated[str, Form()], query: Annotated[str, Form()]):
    start_time = time.time()

    document_store = await get_document_store()

    retriever = BM25Retriever(document_store=document_store, top_k=10)

    chat_pipeline_solo = Pipeline()
    chat_pipeline_solo.add_node(component=retriever, name="Retriever", inputs=["Query"])
    chat_pipeline_solo.add_node(component=prompt_node_chat, name="PromptNode", inputs=["Retriever"])

    answer, relevant_documents, updated_summary, similarity = chat_with_rag_pipeline(query, chat_pipeline_solo, summary)
    response_data = jsonable_encoder(json.dumps({"answer": answer, "relevant_documents": relevant_documents,
                                                "updated_summary": updated_summary, "similarity": similarity}))
    print("Time took to process the request and return response is {} sec".format(time.time() - start_time))
    res = Response(response_data)
    return res

```

Fig 5.3. Pseudocode for answering questions about the legal document

- b. Steps
 - i. Join the reference texts into a single string.
 - ii. Create a TF-IDF vectorizer and fit it to the reference text and generated response.
 - iii. Compute the cosine similarity between the TF-IDF vectors.
 - iv. Return the cosine similarity score.
 - v. Run the chat pipeline with the given query, retrieval parameters, and metadata (summary and legal clauses).
 - vi. Extract the answer and the retrieved documents from the pipeline response.
 - vii. Compute the similarity between the retrieved documents and the generated answer using `compute_sim`.
 - viii. Generate a new summary using a prompt template and the retrieved documents and answer.
 - ix. Get the document store asynchronously.
 - x. Create a retriever (BM25Retriever) with the document store and specify the number of top results to retrieve.
 - xi. Create a pipeline and add the retriever and a prompt node for generating the answer.
 - xii. Call `chat_with_rag_pipeline` to get the answer, relevant documents, updated summary, and similarity score.
 - xiii. Encode the response data (answer, relevant documents, summary, similarity) as JSON.
 - xiv. Return the encoded response data.

4. Algorithm description for generating questions for filling the template :

- a. Pseudocode

```

def replace_dots_with_mask(input_string):

    counter = 1
    result_string = ''
    start = 0
    for m in re.finditer(r'\.{3,}|_{3,}', input_string):
        end, newstart = m.span()
        result_string += input_string[start:end]
        rep = f" [MASK{counter}] "
        result_string += rep
        start = newstart
        counter += 1
    result_string += input_string[start:]

    return result_string

```

```

async def generate_ques():
    for idx, quesDocIndex in enumerate(quesDocsIndex):

        document = preprocessed_docs[quesDocIndex].content

        document = replace_dots_with_mask(document)

        document = document.replace("\u2026", "")

        result = prompt_node.prompt(prompt_template=prompt_template, document=document)

        result[0].answer = re.sub(r'(\s*)', r'\1', result[0].answer)
        print(result[0].answer)
        answerJson = json.loads(str(result[0].answer))
        removeItems = []
        for key, value in answerJson.items():
            if key not in document:
                removeItems.append(key)

        for key in removeItems:
            answerJson.pop(key)

        print('{"questions": ' + str(json.dumps(answerJson)) + ', "document": "' + document + '"}')

        document = clean_json_string(document)

        yield '{"questions": ' + str(json.dumps(answerJson)) + ', "document": "' + str(document.replace('\n', '\\n')) + '", "docIndex": ' + str(idx) + ', "totalDocs": ' + str(totalDocs) + '}'

```

Fig 5.4. Pseudocode for generating questions for filling the template

- b. Steps
 - i. Initialize a counter variable to 1.
 - ii. Initialize an empty string `result_string` to store the modified input string.
 - iii. Initialize a variable `start` to keep track of the starting index for slicing the input string.
 - iv. Use a regular expression to find all occurrences of three or more consecutive dots (...) or underscores (_) in the input string.

v. For each match found:

- Append the substring of the input string from `start` to the start of the match to `result_string`.
- Construct a mask string `rep` in the format `"[MASK{counter}]` and append it to `result_string`.
- Update `start` to the end of the match.
- Increment the `counter` variable.

vi. Append the remaining part of the input string (from `start` to the end) to `result_string`.

vii. Strip any leading or trailing whitespace from `result_string` and return it.

viii. In the `generate_ques` function, Iterate over the indices of the question documents (`quesDocsIndex`).

ix. For each index:

- Retrieve the content of the corresponding document from `preprocessed_docs`.
- Call the `replace_dots_with_mask` function on the document content to replace consecutive dots or underscores with a mask token.
- Remove the ellipsis character (`\u2026`) from the document content.
- Call the `prompt_node.prompt` function with the `prompt_template` and the modified document content to generate a result.
- Clean up the result by removing any trailing comma and space before the closing curly brace.
- Print the cleaned-up result.
- Load the result as a JSON object.
- Initialize a list `removeItems` to store keys that are not present in the document content.
- Iterate over the key-value pairs in the JSON object and append keys that are not present in the document content to `removeItems`.
- Remove the keys in `removeItems` from the JSON object. Print a JSON object containing the questions (the modified JSON object) and the document content.
- Clean the document content using the `clean_json_string` function. Yield a JSON object containing the questions, the cleaned document content, the document index, and the total number of documents.

5.3. Datasets source and utilization:

In this project, the **Contract Understanding Atticus Dataset (CUAD)** served as the context for training the RAG-based large language model (LLM). CUAD is a publicly available dataset specifically designed for legal contract review tasks. It comprises over **13,000** annotations across **510** commercial legal contracts, manually labeled by legal experts from The Atticus Project. These

annotations pinpoint **41 crucial legal clause types** frequently encountered in corporate transactions like mergers and acquisitions.

The rationale behind using CUAD stemmed from its suitability for providing context to the RAG encoder-decoder model. During training, the encoder component processed the contract text, rich with legal intricacies, which then informed the decoder's generation of relevant outputs. This approach empowered the LLM to develop an understanding of the legal domain and learn to identify significant clauses within contracts, crucial for tasks like contract review automation.

Chapter 6: Testing of the Proposed System

6.1. Introduction to Testing:

Software testing is the sequence of activities that happen during software testing. By employing a sane software testing life cycle, an organization ends up with a quality strategy more likely to produce better results. Why is this so important, though? It all boils down to customer satisfaction. Presenting a perfect product to the customer is the end goal of every organization.

Nothing puts off customers more than bug-filled user experience. So when enterprises realized this, they began to include testing as a mandatory part of the SDLC. Since then, testing has become an integral part of every organization.

Project Testing Phase means a group of activities designated for investigating and examining progress of a given project to provide stakeholders with information about actual levels of performance and quality of the project. It is an attempt to get an independent view of the project to allow stakeholders to evaluate and understand potential risks of project failure or mismatch. The purpose of the testing phase is to evaluate and test declared requirements, features, and expectations regarding the project prior to its delivery in order to ensure the project matches initial requirements stated in specification documents.

6.2. Types of tests considered:

A. User Interface (UI) Integration:

These tests focus on ensuring that the RAG model seamlessly fits into the application's user interface. We want to make sure that when you're using the application, the RAG model's retrieval, augmentation, and generation features work smoothly with all the different parts of the interface, like buttons, menus, and screens. These tests help us confirm that there are no hiccups or issues when you interact with the application's UI while using the RAG model.

B. LLM Generation Accuracy:

These tests focus on how accurately the model can create legal text or clauses when it's asked to. We want to see if the model can produce the right kind of legal language depending on what type of clause it's supposed to generate. To do this, we compare the legal clauses the model comes up with to real legal clauses. We use measures like ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (BiLingual Evaluation Understudy) to see how close the model's output is to the real thing.

C. Legal Content Similarity:

In these tests, we compare the content generated by the model with the legal Contract Understanding Atticus Dataset (CUAD) dataset we already have stored in the vector index. We want to see how similar the generated content is to the existing legal information. To do this, we use similarity metrics. If the similarity scores are high enough, it means that the generated content is relevant to the query we gave the model. This helps us gauge how well the model is performing in terms of generating content that matches what we already know.

6.3. Various test case scenarios considered

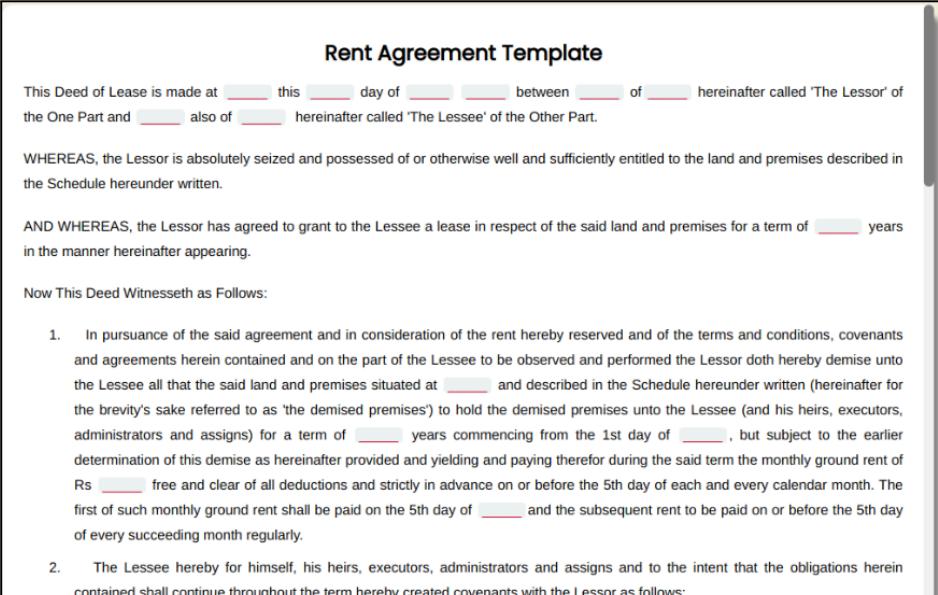
Case	Test
1. The user interface should have good user experience by providing easy to understand content navigation.	 <p>Rent Agreement Template</p> <p>This Deed of Lease is made at _____ this _____ day of _____ between _____ of _____ hereinafter called 'The Lessor' of the One Part and _____ also of _____ hereinafter called 'The Lessee' of the Other Part.</p> <p>WHEREAS, the Lessor is absolutely seized and possessed of or otherwise well and sufficiently entitled to the land and premises described in the Schedule hereunder written.</p> <p>AND WHEREAS, the Lessor has agreed to grant to the Lessee a lease in respect of the said land and premises for a term of _____ years in the manner hereinafter appearing.</p> <p>Now This Deed Witnesseth as Follows:</p> <ol style="list-style-type: none"> 1. In pursuance of the said agreement and in consideration of the rent hereby reserved and of the terms and conditions, covenants and agreements herein contained and on the part of the Lessee to be observed and performed the Lessor doth hereby demise unto the Lessee all that the said land and premises situated at _____ and described in the Schedule hereunder written (hereinafter referred to as 'the demised premises') to hold the demised premises unto the Lessee (and his heirs, executors, administrators and assigns) for a term of _____ years commencing from the 1st day of _____, but subject to the earlier determination of this demise as hereinafter provided and yielding and paying therefor during the said term the monthly ground rent of Rs _____ free and clear of all deductions and strictly in advance on or before the 5th day of each and every calendar month. The first of such monthly ground rent shall be paid on the 5th day of _____ and the subsequent rent to be paid on or before the 5th day of every succeeding month regularly. 2. The Lessee hereby for himself, his heirs, executors, administrators and assigns and to the intent that the obligations herein contained shall continue throughout the term hereby created covenants with the Lessor as follows:
2. The additional clause generation prompt for LLM should produce correct results.	<p>Answer the question truthfully based solely on the given documents. If the documents do not contain the answer to the question, say that answering is not possible given the available information. Extract legal clauses from the contexts and provide related answers.</p>

Fig 6.1. Rent Agreement Template

Fig 6.2. Prompt for additional clause generation

3. The question generation module should produce relevant questions to the contract.	<p>Where is this Deed of Lease made? On what day is this Deed of Lease made? Between whom is this Deed of Lease made? Who is referred to as 'The Lessor' in this Deed of Lease? Who is referred to as 'The Lessee' in this Deed of Lease? How many years is the lease for the said land and premises? What is the term of the lease for the said land and premises? For how long is the lease granted for the said land and premises? What is the duration of the lease for the said land and premises? How many years is the lease for the said land and premises granted for? What is the length of the lease for the said land and premises? For</p>
--	--

Fig 6.3. Results produced by question generation module

4. The output from the question generation module should be valid JSON LLM responses	<pre>{ "[MASK1]": "Where was the agreement of sale made?", "[MASK2]": "What is the specific day of the agreement?", "[MASK3]": "What is the specific month of the agreement?", "[MASK4]": "What is the specific year of the agreement?", "[MASK5]": "Who is the father of the vendor?", "[MASK6]": "Where does the vendor reside?", "[MASK7]": "Who" }</pre> 
--	---

Fig 6.4. Handling invalid JSON LLM responses

Table 6.1 : Test case scenarios

6.4. Inference drawn from the test cases

Case	Test
------	------

1. The user interface is designed to separate relevant text content into self-sufficient sliding content.

Fill the form to prepare lease document

Progress:
1 of 9

Where is this Deed of Lease made?

What is the date of this Deed of Lease?

What month is this Deed of Lease made?

Where is A located?

Where is B located?

➡

After filling the blanks you will be allowed to download the document

This draft of Deed of Lease (for a Term of Years) Rent Agreement was downloaded from LawRato.com Consult the bestDocumentation lawyer at [http://lawrato.com/documentation-lawyersDraft of Deed of Lease \(for a Term of Years\) Rent Agreement](http://lawrato.com/documentation-lawyersDraft of Deed of Lease (for a Term of Years) Rent Agreement)This Deed of Lease is made at _____ this _____ day of _____ between A of _____ hereinafter called 'The Lessor' of the One Part and B also of _____ hereinafter called 'The Lessee' of the Other Part.

Fig 6.5. Text as sliding content

2. The additional clause generation prompt for LLM is tuned to produce succinct and accurate responses.

Given a {documentType} which has the following clauses {legalClauses}. Return additional clauses that can be added to the legal agreement.

Fig 6.6. Enhanced prompt for clause generation

3. The question generation module has been provided GPT 3.5 access for better results than Mixtral-8x7B

"Where is this Deed of Lease made?"
 "What is the date of this Deed of Lease?"
 "What month is this Deed of Lease made?"
 "Where is A located?"
 "Where is B located?"

Fig 6.7. Questions generated by GPT 3.5

4. The output from the question generation module is passed through many constraints to validate JSON content

```
{
  '[MASK1]': 'Where is the land and premises situated?',
  '[MASK2]': 'How many years is the term of the lease?',
  '[MASK3]': 'What is the starting date of the lease?',
  '[MASK4]': 'What is the monthly ground rent in Rs?'
}
```

Fig 6.8. Output with valid JSON content due to applied constraints

Table 6.2 : Inferences drawn from test cases

Chapter 7: Results and Discussions

7.1 Screenshots of User Interface

Home Page : This is the homepage of our website which describes our project LegalEase for the users. Users could signup or login so that authenticated users could only login into the website with their proper credentials.

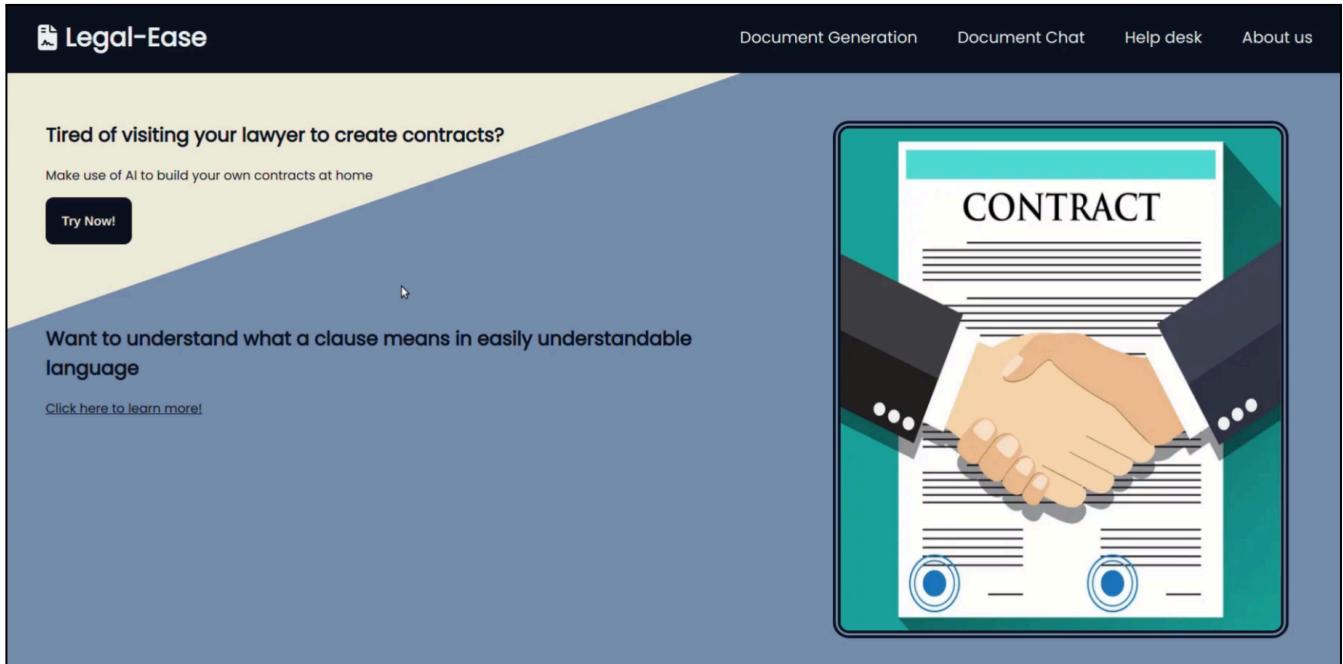


Fig 7.1.1: Home Page

Legal Document Generation Form Page : Users can answer the questions asked in the form and it will generate a legal agreement using the base template. As the users fill the answers in the form correspondingly the blank spaces in the document gets filled.

A screenshot of the Legal Document Generation Form Page. The top navigation bar is identical to the home page. The main form area has a light beige background. It contains two sections: one on the left for entering information and one on the right showing the generated legal document. The left section includes fields for 'Progress' (1 of 5), 'Who is referred to as the 'AGENCY' in this agreement?' (Mohit Industries), and 'Who is referred to as the 'CONTRACTOR' in this agreement?' (Rahul Motwani). An arrow points from the left section to the right section. The right section displays a template of a legal agreement with placeholders for 'Mohit Industries' and 'Rahul Motwani'.

Fig 7.1.2 : Legal Document Generation Form Page

Document Chat Page : Users can chat regarding the legal document generated. Here is a screenshot of a user asking for the summary of the legal document.

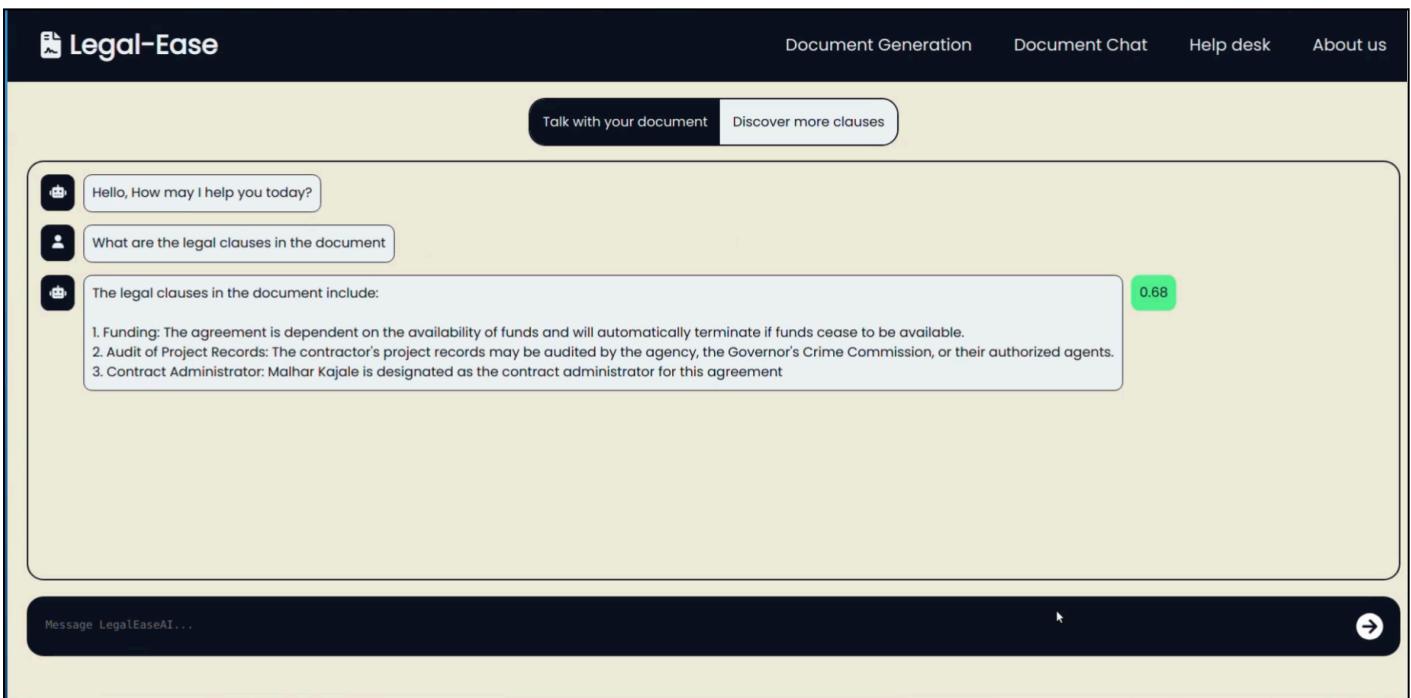


Fig 7.1.3 :Document Chat Page

Discover New Clauses : Users can discover new clauses which can be added in the legal document. Then the user could also chat with the chatbot regarding new clauses that could be added.

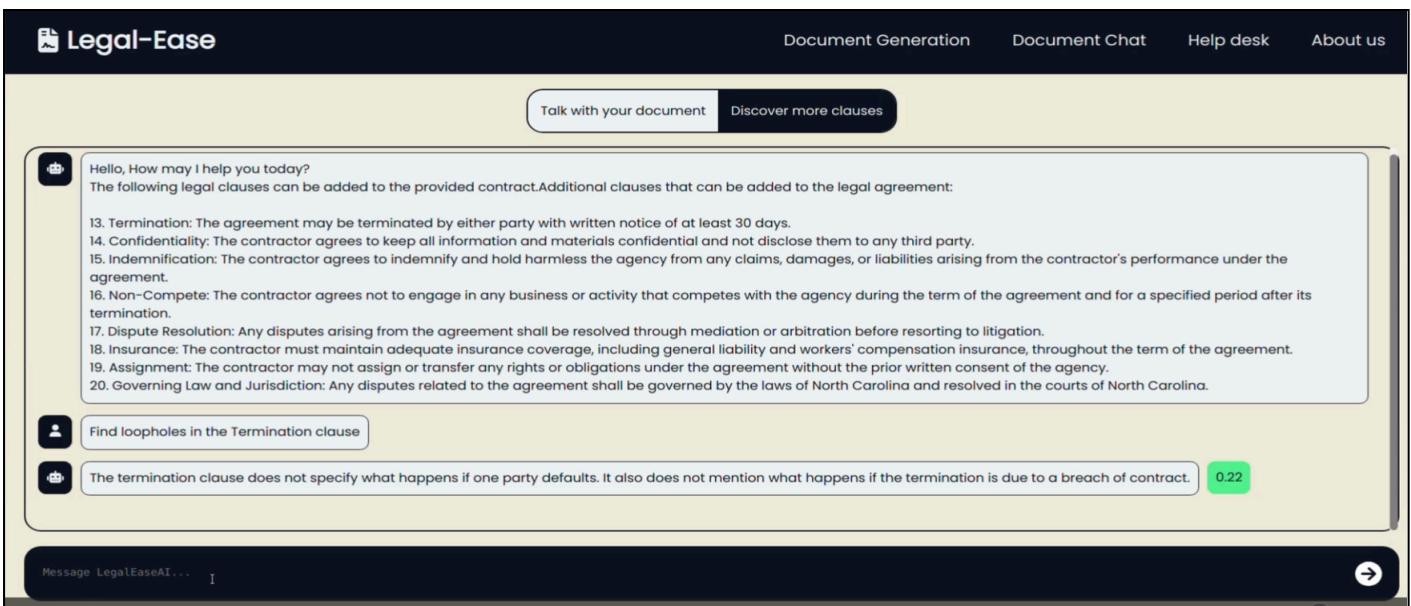


Fig 7.1.4 Discover New Clauses Page

Help Desk Page : A chatBot is also provided where users can ask any query regarding law. Here a query and its generated response has been shown in the screenshot.

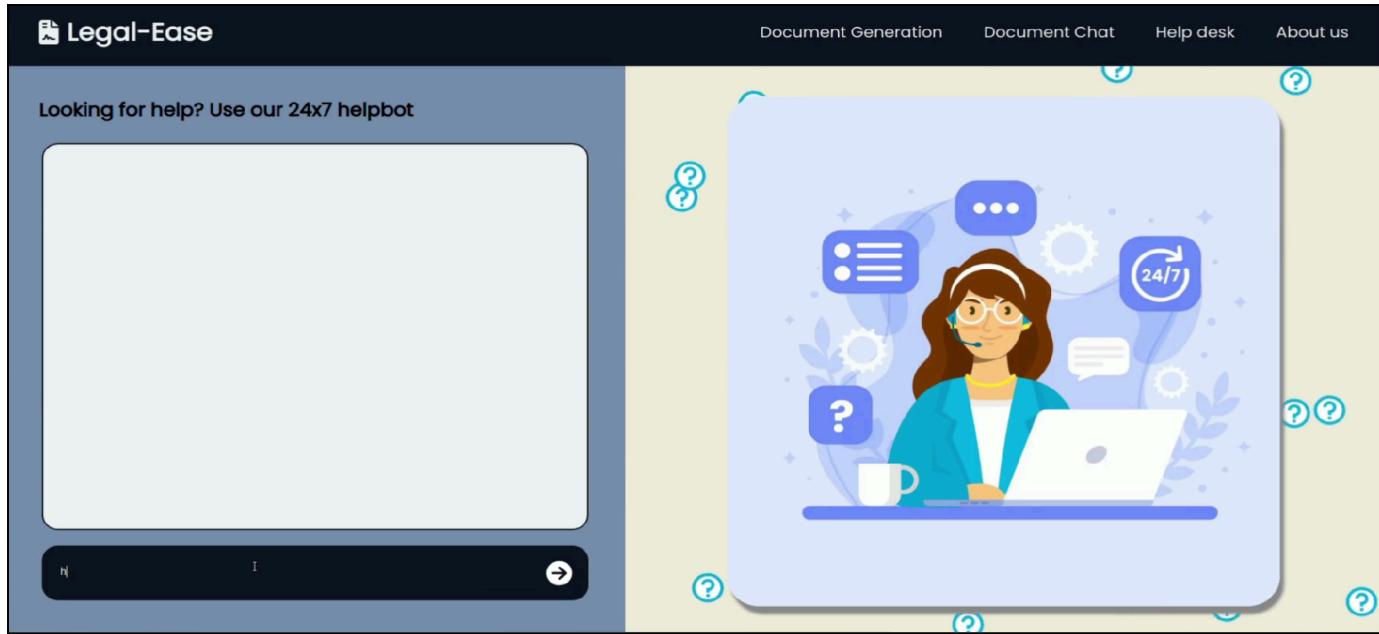


Fig 7.1.5 : Help Desk Page

7.2. Performance Evaluation measures

To guarantee the accuracy of our legal documents, we use a method called ROUGE metrics for evaluating customized clauses.

Steps involved in performance evaluation:

1. We start by obtaining sample legal contracts from the internet. Then, we remove specific clauses from these contracts. Afterward, we pass the modified document, which lacks certain clauses, to our system for clause generation. Each time, we instruct the system to generate the missing clause types. Following this, we compare the generated clauses with the original ones using ROUGE metrics. This process helps us assess how closely our system's generated clauses match the originals, ensuring precision and minimizing the risk of errors that could adversely affect users.
2. The figures below show the evaluation metrics rouge-1, rouge-2, rouge-3, precision, recall and F-1 scores of generated clauses with respect to the clauses referred from the internet.

1] Assignment and Assumption Agreement

Reference	Each Party shall indemnify and hold harmless other Party and Lessor against any and all loss, liability, damage or expenses which may be incurred by other Party and Lessor due to any claims of a third party in connection with the breach, default or non-performance of the SLA by a Party on or after the date of execution of this Agreement			
Generated	The Assignee agrees to indemnify and hold harmless the Assignor, its affiliates, and their respective officers, directors, and employees from any and all claims, losses, liabilities, damages, expenses, and costs (including attorneys' fees and court costs) arising out of or related to the Assignee's breach of this Agreement.			
Clause-type	Indemnification			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.3	0.149	0.275
	Precision	0.279	0.123	0.256
	F1-score	0.289	0.135	0.265

Fig 7.2.1 : Assignment and Assumption Agreement Clause-1

Reference	Parties agree that they may not amend this Agreement in any way that will materially affect the Lessor's rights or provisions of the SLA.			
Generated	This Agreement may not be amended or modified except in writing executed by both Parties.			
Clause-type	No Amendment			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.4	0.071	0.2
	Precision	0.273	0.043	0.136
	F1-score	0.324	0.054	0.162

Fig 7.2.2 : Assignment and Assumption Agreement Clause-2

2] Non Disclosure Agreement

Reference	The Company, in its discretion, may disclose the Confidential Information to its employees, officers, agents and third party consultants on a need to know basis in furtherance of the Project.			
Generated	The company may disclose Confidential Information to its employees, agents, and representatives who need to know the information for the purposes of this Agreement, provided that such individuals are bound by confidentiality obligations no less restrictive than those set forth herein. The company remains responsible for any breach of confidentiality by its representatives.			
Clause-type	Disclosure to Company Representatives			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.302	0.137	0.279
	Precision	0.5	0.241	0.462
	F1-score	0.377	0.175	0.348

Fig 7.2.3 : Non Disclosure Agreement Clause-1

Reference	The Corporation shall protect, indemnify, and hold the Company harmless from any loss suffered by the Company due to the authorized use of the Confidential Information by the Company.		
Generated	Each party agrees to indemnify and hold harmless the other party, its affiliates, directors, officers, and employees from any claims, liabilities, damages, or expenses, including reasonable attorneys' fees, arising out of their breach of this agreement.		
Clause-type	Indemnity		
Scores		Rouge-1	Rouge-2
	Recall	0.235	0.057
	Precision	0.364	0.08
	F1-score	0.286	0.067
Rouge-I	0.176	0.273	0.214

Fig 7.2.4 : Non Disclosure Agreement Clause-2

3] Arbitration Agreement

Reference	If any provision of this Agreement is or becomes illegal, invalid or unenforceable in any jurisdiction, the illegality, invalidity or unenforceability of that provision will not affect the validity of any other provision of this Agreement		
Generated	If any provision of this Agreement is held invalid or unenforceable, such provision shall be struck and the remaining provisions shall be enforced.		
Clause-type	Severability		
Scores		Rouge-1	Rouge-2
	Recall	0.5	0.333
	Precision	0.417	0.219
	F1-score	0.455	0.264
Rouge-I	0.5	0.417	0.455

Fig 7.2.5 : Arbitration Agreement Clause-1

Reference	No failure to enforce any of its right by a Party shall not constitute waiver of such right.		
Generated	Failure to enforce any provision of this Agreement shall not constitute a waiver of any other provision, and the waiver of any right or remedy on any occasion shall not be construed as a bar to or waiver of any right or remedy on any future occasion.		
Clause-type	Waiver		
Scores		Rouge-1	Rouge-2
	Recall	0.37	0.139
	Precision	0.625	0.294
	F1-score	0.465	0.189
Rouge-I	0.296	0.5	0.372

Fig 7.2.6 : Arbitration Agreement Clause-2

7.3 Input Parameters / Features considered

- Users should have their credentials to login to our website.
- After the login users can choose the legal document or upload a base legal document template and answer the questions in the form to get a legal document.
- With the help of talk to your document ChatBot users can ask queries regarding the generated legal document
- With the help of discover new clauses page a user can discover new legal clauses that can be added in the document. After discovering the new clauses he can also chat with the ChatBot regarding the queries related with the new clauses.
- All the drafts of the users can be accessed by using the drafts module in the Navbar.
- Users are also provided with a help desk on the website where they can ask any queries related to law.

7.4 Inference drawn

- The LegalEase system developed by us integrates and explores advanced technologies such as large language models to generate legal documents efficiently. By leveraging conversational interfaces where users have to simply answer the related queries asked by the system regarding the legal documents users can easily generate, understand, and customize legal documents. This can lead to faster and efficient legal document generation.
- The customized legal clauses generated by the system can be helpful for the users to add custom clauses in the legal document which are not present in the base template. The users could also ask various queries related to these custom clauses to our system to understand the clauses thoroughly.
- The generated custom clauses are also evaluated by the system and shown to the user so that the user does not make any mistake before putting the custom clause in the legal document.

- The project can also potentially reduce human error and bias that may exist in the manual document generation. Automated legal document generation can ensure that legal jargons which are difficult to understand do not come into the way of the users to get justice.

Chapter 8: Conclusion

8.1. Limitations:

Key challenges and limitations in our project are as follows:

1. Legal Accuracy and Compliance.
2. Staying in-sync with modifications in templates as per updated laws.
3. Handling and managing ethical issues and biases.

8.2. Conclusion:

1. Our project simplifies the process of creating and comprehending legal contracts.
2. Customized forms are available for various contract types, enabling users to easily draft personalized contracts.
3. Users can add customized clauses by describing them in our chatbot, leveraging advanced language models.
4. For individuals unfamiliar with legal terminology, our platform converts complex contracts into plain text for better understanding.
5. We aspire to transform contract creation and comprehension, promoting inclusivity and clarity in the realm of legal agreements.

8.3. Future Scope:

- a) Collaborating with top-notch legal professionals to enhance our application from their feedback.
- b) Fine tuning and deploying the LLMs for using them remotely.
- c) Experimenting with the output quality by trying out more prompt templates.
- d) Enhancing the system by adding multilingual contract generation and explanation capability.

References

1. Lam, K. Y., Cheng, V. C., & Yeong, Z. K. (2023). Applying large language models for enhancing contract drafting. In Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workspace (LegalAIIA 2023).
2. Hendrycks, D., Burns, C., Chen, A., & Ball, S. (2021). Cuad: An expert-annotated nlp dataset for legal contract review. arXiv preprint arXiv:2103.06268.
3. Martin, L., Whitehouse, N., Yiu, S., Catterson, L., & Perera, R. (2024). Better Call GPT, Comparing Large Language Models Against Lawyers. arXiv preprint arXiv:2401.16212.
4. Aggarwal, V., Garimella, A., Srinivasan, B. V., & Jain, R. (2021). CLAUSEREC: A Clause Recommendation Framework for AI-aided Contract Authoring. arXiv preprint arXiv:2110.15794.
5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.
6. Zödi, Zsolt. (2019). The limits of plain legal language: understanding the comprehensible style in law. International Journal of Law in Context. 2019. 246–262. 10.1017/S1744552319000260.
7. Queudot M, Charton É, Meurs M-J. Improving Access to Justice with Legal Chatbots. *Stats*. 2020; 3(3):356-375. <https://doi.org/10.3390/stats3030023>
8. Amato F, Fonisto M, Giacalone M, Sansone C. An Intelligent Conversational Agent for the Legal Domain. *Information*. 2023; 14(6):307. <https://doi.org/10.3390/info14060307>
9. Hricik, D., Morgan, A.-L. S., & Williams, K. H. (2018). Ethics of Using Artificial Intelligence to Augment Drafting Legal Documents. Texas A&M Journal of Property Law, 4, 465. <https://doi.org/10.37419/JPL.V4.I5.3>
10. Zhongxiang Sun. (2023). A Short Survey of Viewing Large Language Models in Legal Aspect.
11. Hafner, Carole & Lauritsen, Marc. (2007). Extending the power of automated legal drafting technology.. 59-68.
12. Marković, M., Gostojić, S. Legal document assembly system for introducing law students with legal drafting. Artif Intell Law (2022). <https://doi.org/10.1007/s10506-022-09339-2>
13. Lankester, R. (2018). Implementing Document Automation: Benefits and Considerations for the Knowledge Professional. Legal Information Management, 18(2), 93-97. doi:10.1017/S1472669618000208
14. Waltl, B., Landthaler, J., Scepankova, E., Matthes, F., Geiger, T. H. O. M. A. S., Stocker, C., & Schneider, C. (2017). Automated extraction of semantic information from german legal documents. In IRIS: Internationales Rechtsinformatik Symposium.

15. I. T. Imam, Y. Arafat, K. S. Alam and S. A. Shahriyar, "DOC-BLOCK: A Blockchain Based Authentication System for Digital Documents," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 1262-1267, doi: 10.1109/ICICV50876.2021.9388428.
16. Schweighofer, E. (2010). Semantic Indexing of Legal Documents. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds) Semantic Processing of Legal Texts. Lecture Notes in Computer Science(), vol 6036. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12837-0_9
17. Villata, S. (2020). Sentence embeddings and high-speed similarity search for fast computer assisted annotation of legal documents. In Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020 (Vol. 334, p. 164). IOS Press.
18. Jobin, A., Ienca, M. & Vayena, E. The global landscape of AI ethics guidelines. *Nat Mach Intell* 1, 389–399 (2019). <https://doi.org/10.1038/s42256-019-0088-2>
19. Ng, J., Haller, E., & Murray, A. (2022). The ethical chatbot: A viable solution to socio-legal issues. *Alternative Law Journal*, 47(4), 308-313.

Appendix

a. Paper Details :-

Exploring RAG For Contract Understanding and Clause Generation using LLMs

Dr. Sujata Khedkar
Dept. of Computer Engineering
VESIT
Mumbai, India
sujata.khedkar@ves.ac.in

Rahul Motwani
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.rahal.motwani@ves.ac.in

Malhar Kajale
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.malhar.kajale@ves.ac.in

Nimish Chidrawar
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.nimish.chidrawar@ves.ac.in

Mohit Shahdadpuri
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.mohit.shahdadpuri@ves.ac.in

Abstract—This paper explores the use of LLMs with RAG for generation and understanding of legal contracts. Retrieval Augment - Generation (RAG) is a technique which is used to provide context to the models so that the responses are precise to the domain of concern. The paper explores the use of RAG in the legal domain, especially contracts due to their widespread applications and need of manual effort in understanding and reviewing them. The paper outlines a solution that holds utility not only for users within the legal domain but also extends its applicability beyond. The solution spans from generating contract clauses using templates to providing a chatbot for users to grasp both generated and predefined clauses within the contract.

Index Terms—Retrieval-Augmented-Generation (RAG), Legal contract generation, legal contract understanding

While many researches explored use of LLMs and their efficacy, RAG in legal domain was not explored. We also use RAG for Question Answering to help parties involved in the contracts fully understand the clauses before signing the document.

B. Problem Statement

Contract drafting is a manual-intensive process for lawyers while understanding of those contracts is complicated and time-consuming for small businesses and individuals. The use of legal jargon makes it hard to interpret for a person with no legal background.

Mostly contracts follow a default template with some common clauses for every case with changes in the details of stakeholders. The information can be taken from the user using a question-answer format with automated question generation.

Any additional legal clauses can either be recommended by the system or be added based on a brief description by the user. Implement chatbots for users to interact with their documents. *C. Objectives*

- 1) Build a solution whose components should be incorporated with the required legal domain knowledge using LLMs with RAG. Since RAG based models are more efficient and scalable than fine-tuning, this approach was chosen.
- 2) Incorporate templates of various contract types and generate questions to fill the template.
- 3) Generate additional legal clauses for the document.
- 4) Allow the user to ask questions regarding a system generated or an external document. The chatbot should be capable of answering the questions in as simple language as possible.

II. OVERVIEW

A. Motivation

Along with requiring legal context, contract drafting also requires a solution to ensure modification of templates on the fly, an interactive chatbot to explain the contract and clauses and generate new clauses. The motivation was to develop a solution which addresses and fulfills all these gaps.

- 5) Evaluate the generated clauses for their accuracy and deduce the effectiveness of RAG for the problem statement

type and passing it in the prompt in the RAG pipeline which already holds the context to the contract. Lewis et al. evaluate RAG models on Question Answering tasks and find the results promising.

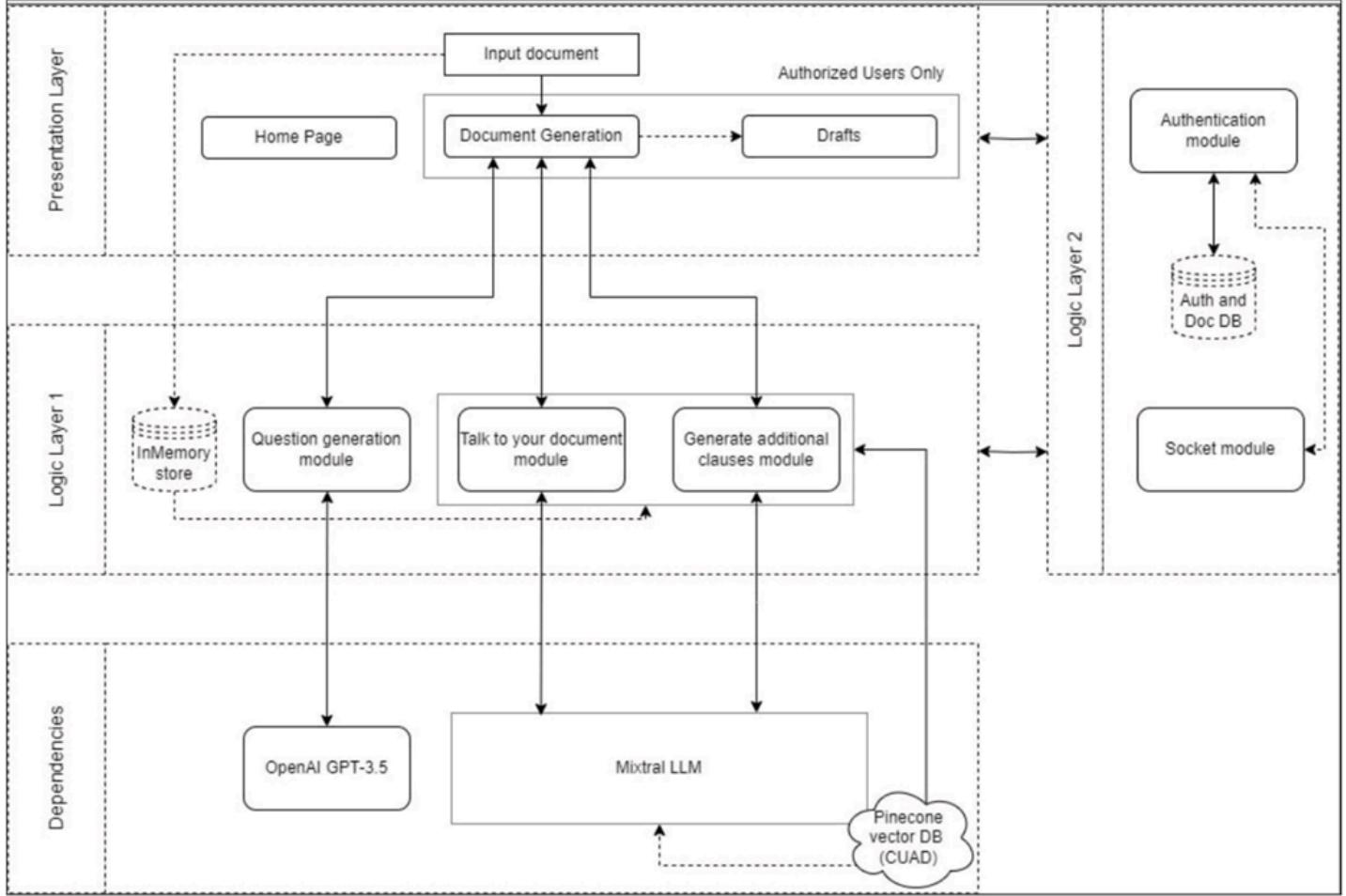


Fig. 1. Architecture

III. THEORY

A. Limitations and Research Gap

In [1], Kwok-Yan-Lam et al. has explored the usefulness of LLMs in contract drafting and proposed various prompt writing techniques. In [3], Martin et al. has compared performance of LLMs with large context windows to lawyers and concluded that LLMs perform better than Junior lawyers. As discussed by Hendrycks et al.[2] , small companies and individuals need a tool which can translate complex legal jargon in plain text for them while also ensuring maximum legal precision. Our project is an assistive tool for all the stakeholders involved in the process of contract drafting.

Aggarwal et al. [4] proposes CLAUSEREC framework for clause recommendation in contract drafting as a two-step process out of which the second step which utilizes the contract context to suggest clauses is of relevance to our work. However, we generate additional clauses by taking the clause

B. Project Contribution

Martin et al. [3] suggests using models with larger context windows over RAG. But increasing user base and popularity of LLMs are making it difficult to make such models cheaply accessible and therefore, we propose RAG as a cost-effective solution for small businesses, individuals and law firms.

The novelty of our project stems from exploring the use of RAG with open source LLMs for contract understanding and generation along with proposing a solution which can not only assist lawyers but also help small businesses and individuals in understanding the clauses in a contract.

C. Architecture

The architecture is divided into three key modules, each playing a vital role in the contract generation process. The Additional Clause Generation Module enriches contracts by providing supplementary legal clauses using advanced technologies. The Talk to Your Document Module facilitates interactive communication with generated documents via a

chat-bot interface. Lastly, the Question Generation Module automates the creation of pertinent questions that help filling the contracts based on user-provided contract templates. Together, these modules streamline document creation, interaction, and completion.

1) Additional Clause Generation Module

Before generating additional clauses, the Contract Understanding Atticus Dataset (CUAD) was entered into a Pinecone Vector Index. The vector index was used for retrieving similar contract clauses. After much trial and error, it was decided that a chunk length of 100 words with an overlap between chunks of 10 words is the most suitable for the task of retrieving legal clauses accurately. These chunks were converted into embeddings using a sentence-transformer model and finally stored in the vector database. Fig 2 represents the entire process succinctly.

When the user wants to generate additional legal clauses, they have to input the document template from which new clauses will be discovered. This uploaded document is preprocessed as shown in Fig 2 and chunked with a chunk length of 5 sentences and an overlap of 2 sentences. The chunked documents are then stored in a memory-based document store provided by haystack.

Using two custom prompts, one for identifying the type of legal document and another for identifying the clauses present in the document, relevant content was retrieved from the uploaded document. The retrieved content is passed along with the two custom prompts to the Mixtral-8x7B model to predict and generate the required information of type of document and legal clauses in the document. This process can be seen in Fig x.

The last sub-process as seen in Fig 2 for generating new relevant clauses is to pass the acquired type of legal document and legal clauses present in the document along with a custom prompt to the Mixtral-8x7B model. In this manner, modified Retrieval Augmented Generation is utilized to generate additional legal clauses for the provided input document.

2) Talk to Your Document Module

In this module a RAG framework is created for enabling the users to inquire and solve questions about the uploaded legal document. The user if first required to upload a legal document. This uploaded document is preprocessed as shown in Fig 2 and chunked with a chunk length of 5 sentences and an overlap of 2 sentences. The chunked documents are then stored in a memory-based document store provided by haystack.

When the user asks a question, the relevant chunks are fetched from the document store using a BM25 retriever. Afterwards the top 5 retrieved chunks along with the query is inserted into the instruction prompt and passed to the

Mixtral8x7B model to generate the answer. The generated answer is then summarized to maintain a historical summary of the questions asked. This summary is then passed to the instruction prompt for any subsequent questions by the user. This process is shown in diagrammatic form in Fig 2.

3) Question Generation Module

The Question Generation Module automatically produces relevant questions based on the user-provided contract template. This helps to streamline the document filling process. The template is preprocessed in which sentences with blank spaces are identified. The GPT-3.5 model developed by OpenAI is then fed these chosen sentences with blank spaces, and it produces questions intended to elicit responses from users that provide the missing data in the template. The entire process is represented by Fig 2. This expedites the data entry process and guarantees correctness when creating contracts.

While performing the task of question generation, it was identified that the Mixtral8x7B model is unable to reliably perform this task. Apart from this unreliability, it is also difficult to decompose the task of question generation into two mutually exclusive tasks that can be run serially. Hence, it was decided to try and use the GPT 3.5 model to perform this task successfully.

The following prompt was engineered to reliably produce questions that pertain to the blank spaces.

IV. RESULTS AND EVALUATION

Since there is requirement of precision in clause generation, a human-in-the-loop (cite) method is more secure. We aim to assist a knowledgeable professional in the task.

Figures 3 to 13 display the rouge-1, rouge-2 and rouge-l recall, precision and f1 scores of generated clauses in comparison with reference clause along with clause type. The methodology employed to evaluate the clause generation module is as follows.

- 1) We used sample legal contracts available on the Internet.
- 2) Specific clauses were deleted from the legal contract and the document was then passed to the clause generation module.
- 3) We generated each deleted clause by specifying each clause type separately to the RAG. Same document was passed to the clause generation module every time.
- 4) The generated and reference clauses were then compared using ROUGE metrics. The clauses are listed in the subsequent figures 3 to 13.
- 5) This procedure was performed for 3 different types of legal contracts - Assignment and Assumption Agreement (figs. 3 to 5), Non-Disclosure Agreement (figs. 6 to 8) and Arbitration Agreement (figs. 9 to 13).

V. CONCLUSION

This paper explores use of RAG in LLMs for legal contract drafting and understanding. We describe our solution in detail and demonstrate how providing legal context using CUAD v1

facilitates the generation of practical and applicable clauses. Our solution aims to facilitate contract comprehension for users and optimize the process of contract review for small businesses and law firms.

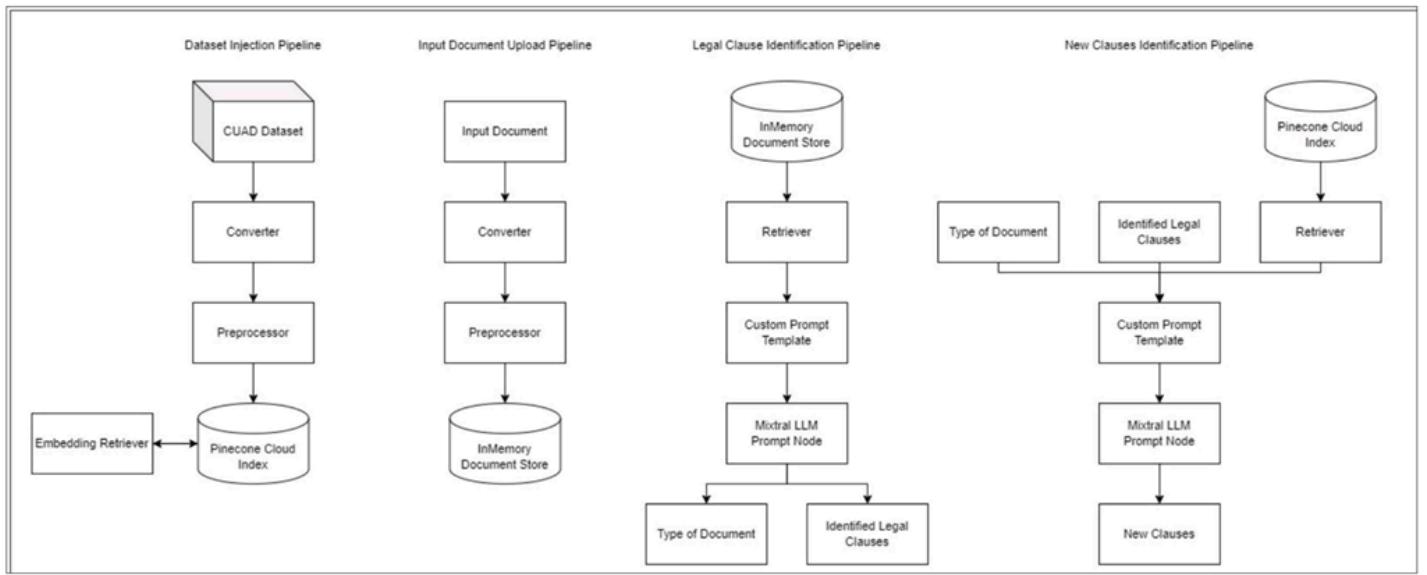


Fig. 2. LLM Pipeline Flow Diagram

Reference	Each Party shall indemnify and hold harmless other Party and Licensee against any and all loss, liability, damage or expenses which may be incurred by other Party and Licensee due to any claims of a third party in connection with the breach, default or non-performance of the SLA by a Party on or after the date of execution of this Agreement																
Generated	The Assignee agrees to indemnify and hold harmless the Assignor, its affiliates, and their respective officers, directors, and employees from any and all claims, losses, liabilities, damages, expenses, and costs (including attorneys' fees and court costs) arising out of or related to the Assignee's breach of this Agreement.																
Clause-type	Indemnification																
Scores	<table border="1"> <thead> <tr> <th></th> <th>Rouge-1</th> <th>Rouge-2</th> <th>Rouge-L</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>0.3</td> <td>0.149</td> <td>0.275</td> </tr> <tr> <td>Precision</td> <td>0.279</td> <td>0.123</td> <td>0.256</td> </tr> <tr> <td>F1-score</td> <td>0.289</td> <td>0.135</td> <td>0.265</td> </tr> </tbody> </table>		Rouge-1	Rouge-2	Rouge-L	Recall	0.3	0.149	0.275	Precision	0.279	0.123	0.256	F1-score	0.289	0.135	0.265
	Rouge-1	Rouge-2	Rouge-L														
Recall	0.3	0.149	0.275														
Precision	0.279	0.123	0.256														
F1-score	0.289	0.135	0.265														

Fig. 3. 1st generated-actual clause pair

Reference	The Company, in its discretion, may disclose the Confidential Information to its employees, officers, agents and third party consultants on a need to know basis in furtherance of the Project.																
Generated	The company may disclose Confidential Information to its employees, agents, and representatives who need to know the information for the purposes of this Agreement, provided that such individuals are bound by confidentiality obligations no less restrictive than those set forth herein. The company remains responsible for any breach of confidentiality by its representatives.																
Clause-type	Disclosure to Company Representatives																
Scores	<table border="1"> <thead> <tr> <th></th> <th>Rouge-1</th> <th>Rouge-2</th> <th>Rouge-L</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>0.302</td> <td>0.137</td> <td>0.279</td> </tr> <tr> <td>Precision</td> <td>0.5</td> <td>0.241</td> <td>0.462</td> </tr> <tr> <td>F1-score</td> <td>0.377</td> <td>0.175</td> <td>0.348</td> </tr> </tbody> </table>		Rouge-1	Rouge-2	Rouge-L	Recall	0.302	0.137	0.279	Precision	0.5	0.241	0.462	F1-score	0.377	0.175	0.348
	Rouge-1	Rouge-2	Rouge-L														
Recall	0.302	0.137	0.279														
Precision	0.5	0.241	0.462														
F1-score	0.377	0.175	0.348														

Fig. 6. 4th generated-actual clause pair

Reference	Parties agree that they may not amend this Agreement in any way that will materially affect the Licensee's rights or provisions of the SLA.																
Generated	This Agreement may not be amended or modified except in writing executed by both Parties.																
Clause-type	No Amendment																
Scores	<table border="1"> <thead> <tr> <th></th> <th>Rouge-1</th> <th>Rouge-2</th> <th>Rouge-L</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>0.4</td> <td>0.071</td> <td>0.2</td> </tr> <tr> <td>Precision</td> <td>0.273</td> <td>0.043</td> <td>0.136</td> </tr> <tr> <td>F1-score</td> <td>0.324</td> <td>0.054</td> <td>0.162</td> </tr> </tbody> </table>		Rouge-1	Rouge-2	Rouge-L	Recall	0.4	0.071	0.2	Precision	0.273	0.043	0.136	F1-score	0.324	0.054	0.162
	Rouge-1	Rouge-2	Rouge-L														
Recall	0.4	0.071	0.2														
Precision	0.273	0.043	0.136														
F1-score	0.324	0.054	0.162														

Fig. 4. 2nd generated-actual clause pair

Reference	No failure to enforce any of its right by a Party shall not constitute waiver of such right.																
Generated	Any waiver of any provision of this Agreement must be in writing and signed by the Party against whom the waiver is sought to be enforced. No waiver of any provision shall be deemed a further or continuing waiver of such provision or a waiver of any other provision.																
Clause-type	Waiver																
Scores	<table border="1"> <thead> <tr> <th></th> <th>Rouge-1</th> <th>Rouge-2</th> <th>Rouge-L</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>0.323</td> <td>0.048</td> <td>0.226</td> </tr> <tr> <td>Precision</td> <td>0.625</td> <td>0.118</td> <td>0.438</td> </tr> <tr> <td>F1-score</td> <td>0.426</td> <td>0.068</td> <td>0.298</td> </tr> </tbody> </table>		Rouge-1	Rouge-2	Rouge-L	Recall	0.323	0.048	0.226	Precision	0.625	0.118	0.438	F1-score	0.426	0.068	0.298
	Rouge-1	Rouge-2	Rouge-L														
Recall	0.323	0.048	0.226														
Precision	0.625	0.118	0.438														
F1-score	0.426	0.068	0.298														

Fig. 5. 3rd generated-actual clause pair

Reference	The Corporation shall protect, indemnify, and hold the Company harmless from any loss suffered by the Company due to the authorized use of the Confidential Information by the Company.																
Generated	Each party agrees to indemnify and hold harmless the other party, its affiliates, directors, officers, and employees from any claims, liabilities, damages, or expenses, including reasonable attorneys' fees, arising out of their breach of this agreement.																
Clause-type	Indemnity																
Scores	<table border="1"> <thead> <tr> <th></th> <th>Rouge-1</th> <th>Rouge-2</th> <th>Rouge-L</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>0.235</td> <td>0.057</td> <td>0.176</td> </tr> <tr> <td>Precision</td> <td>0.364</td> <td>0.08</td> <td>0.273</td> </tr> <tr> <td>F1-score</td> <td>0.286</td> <td>0.067</td> <td>0.214</td> </tr> </tbody> </table>		Rouge-1	Rouge-2	Rouge-L	Recall	0.235	0.057	0.176	Precision	0.364	0.08	0.273	F1-score	0.286	0.067	0.214
	Rouge-1	Rouge-2	Rouge-L														
Recall	0.235	0.057	0.176														
Precision	0.364	0.08	0.273														
F1-score	0.286	0.067	0.214														

Fig. 7. 5th generated-actual clause pair

Reference	To the extent absolutely necessary, the Company may disclose Confidential Information, if required by any judicial or governmental request, requirement or order; provided that the Company will provide with sufficient prior written notice of such request, requirement or order to the Corporation so that the Corporation shall contest such request, requirement or order to oppose the disclosure of Confidential Information. The Company shall cooperate with the Corporation in its attempts to oppose such disclosure.																
Generated	In the event that a Party is required by law to disclose Confidential Information to a governmental or regulatory authority, it shall promptly notify the other Party and use reasonable efforts to limit the disclosure to the extent required by law.																
Clause-type	Compelled Disclosure to Authorities																
Scores	<table border="1"> <thead> <tr> <th></th> <th>Rouge-1</th> <th>Rouge-2</th> <th>Rouge-L</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>0.433</td> <td>0.158</td> <td>0.267</td> </tr> <tr> <td>Precision</td> <td>0.289</td> <td>0.098</td> <td>0.178</td> </tr> <tr> <td>F1-score</td> <td>0.347</td> <td>0.121</td> <td>0.213</td> </tr> </tbody> </table>		Rouge-1	Rouge-2	Rouge-L	Recall	0.433	0.158	0.267	Precision	0.289	0.098	0.178	F1-score	0.347	0.121	0.213
	Rouge-1	Rouge-2	Rouge-L														
Recall	0.433	0.158	0.267														
Precision	0.289	0.098	0.178														
F1-score	0.347	0.121	0.213														

Fig. 8. 6th generated-actual clause pair

Reference	If any provision of this Agreement is or becomes illegal, invalid or unenforceable in any jurisdiction, the illegality, invalidity or unenforceability of that provision will not affect the validity of any other provision of this Agreement.			
Generated	If any provision of this Agreement is held invalid or unenforceable, such provision shall be struck and the remaining provisions shall be enforced.			
Clause-type	Severability			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.5	0.333	0.5
	Precision	0.417	0.219	0.417
	F1-score	0.455	0.264	0.455

Fig. 9. 7th generated-actual clause pair

Reference	No failure to enforce any of its right by a Party shall not constitute waiver of such right.			
Generated	Failure to enforce any provision of this Agreement shall not constitute a waiver of any other provision, and the waiver of any right or remedy on any occasion shall not be construed as a bar to or waiver of any right or remedy on any future occasion.			
Clause-type	Waiver			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.37	0.139	0.296
	Precision	0.625	0.294	0.5
	F1-score	0.465	0.189	0.372

Fig. 10. 8th generated-actual clause pair

Reference	This Agreement contains a binding arbitration provision that may be enforced by either Party.			
Generated	The Parties agree that any award rendered by the arbitrator shall be final and binding on the Parties and enforceable in any court of competent jurisdiction.			
Clause-type	Enforceability			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.182	0	0.091
	Precision	0.286	0	0.143
	F1-score	0.222	0	0.111

Fig. 11. 9th generated-actual clause pair

Reference	This Agreement shall be interpreted in favor of arbitration. No rule of construction shall be applied to undermine any presumption in favor of enforcement of this Agreement or in favor of arbitration.			
Generated	In this Arbitration Agreement, unless the context otherwise requires, "Parties" refers to the Employee and the Company, "Arbitration" refers to the process of resolving disputes through final and binding arbitration conducted under the Indian Arbitration and Conciliation Act, 1996, and "Arbitrator" refers to a mutually-agreeable neutral arbitrator who shall be a retired judge of Chennai High Court.			
Clause-type	Interpretation			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.14	0.019	0.07
	Precision	0.3	0.04	0.15
	F1-score	0.19	0.026	0.095

Fig. 12. 10th generated-actual clause pair

Reference	No amendment, supplement, modification or restatement of any provision of this Agreement shall be binding unless it is in writing and signed by each Party.			
Generated	Any amendment to this Agreement must be in writing and signed by both Parties			
Clause-type	Amendments			
Scores		Rouge-1	Rouge-2	Rouge-l
	Recall	0.571	0.385	0.571
	Precision	0.333	0.208	0.333
	F1-score	0.421	0.27	0.421

Fig. 13. 11th generated-actual clause pair

VI. REFERENCES

1. Lam, K. Y., Cheng, V. C., & Yeong, Z. K. (2023). Applying large language models for enhancing contract drafting. In Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workspace (LegalAIIA 2023).
2. Hendrycks, D., Burns, C., Chen, A., & Ball, S. (2021). Cuad: An expert-annotated nlp dataset for legal contract review. arXiv preprint arXiv:2103.06268.
3. Martin, L., Whitehouse, N., Yiu, S., Catterson, L., & Perera, R. (2024). Better Call GPT, Comparing Large Language Models Against Lawyers. arXiv preprint arXiv:2401.16212.
4. Aggarwal, V., Garimella, A., Srinivasan, B. V., & Jain, R. (2021). CLAUSEREC: A Clause Recommendation Framework for AI-aided Contract Authoring. arXiv preprint arXiv:2110.15794.
5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

b. Plagiarism Report of Paper I :-

Exploring RAG For Contract Understanding and Clause Generation using LLMs

1 Dr. Sujata Khedkar
Dept. of Computer Engineering
VESIT
Mumbai, India
sujata.khedkar@ves.ac.in

Rahul Motwani
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.rahul.motwani@ves.ac.in

Malhar Kajale
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.malhar.kajale@ves.ac.in

Nimish Chidrawar
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.nimish.chidrawar@ves.ac.in

Mohit Shahdadpuri
Dept. of Computer Engineering
VESIT
Mumbai, India
2020.mohit.shahdadpuri@ves.ac.in

Exploring RAG For Contract Understanding

ORIGINALITY REPORT

3%
SIMILARITY INDEX

0%
INTERNET SOURCES

3%
PUBLICATIONS

0%
STUDENT PAPERS

PRIMARY SOURCES

- 1 Juhi Chhatlani, Tejas Mahajan, Rushabh Rijhwani, Advait Bansode, Gresha Bhatia. "DermaGenics - Early Detection of Melanoma using YOLOv5 Deep Convolutional Neural Networks", 2022 IEEE Delhi Section Conference (DELCON), 2022 Publication 3%

c. Project Review Sheet :-

Inhouse/ Industry _Innovation/Research:

Class: D17 A/B/C

Sustainable Goal:

Project Evaluation Sheet 2023 - 24

Group No.: 9

Title of Project: Case Sense AnalyticsGroup Members: Rahul Motwani (44), Malhar Kajale (28), Nimish Chidrawar (11), Mohit Shahdadpuri (62)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	3	4	2	2	2	2	2	3	3	3	3	4	45

Comments: Integrate Image based doc and calculate similarity of created doc with std doc templateDr. Sujata Poddar
Name & Signature Reviewer 1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	3	4	2	2	2	2	2	3	3	3	3	4	45

Comments: class wise interpretation can be implemented

Date: 10th february, 2024

Jyoti

Name & Signature Reviewer 2

Inhouse/ Industry _Innovation/Research:

Class: D17 A/B/C

Sustainable Goal:

Group No.: 9

Title of Project: CaseSense AnalyticsGroup Members: Rahul Motwani (44), Nimish Chidrawar (11), Malhar Kajale (28), Mohit Shahdadpuri (62)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	5	3	5	2	2	2	2	2	3	3	3	3	4	46

Comments: good work

Name & Signature Reviewer 1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	3	5	2	2	2	2	2	3	3	3	3	4	45

Comments: -

Date: 9th March, 2024

Jyoti

Name & Signature Reviewer 2