A REPORT
ON

e-Portal for Case Management using Artificial Intelligence

Submitted by,

Ananya T V	20211ISE0013
Madan Gowda I M	20211ISE0010
Dhangar Rakesh	20221LIE0003

Under the guidance of,

Mr. Saptarsi Sanyal

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **e-Portal for Case Management using Artificial Intelligence** being submitted by Ananya T V, Madan Gowda I M, Dhangar Rakesh bearing roll number's 20211ISE0013, 20211ISE0010, 20221LIE0003 respectively in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Information Science and Engineering is a Bonafide work carried out under my supervision.

Mr. Saptarsi Sanyal
Assistant Professor
Presidency School of Computer
science and engineering
Presidency University

Dr. Pallavi R
Professor & HOD
Presidency School of Computer
science and engineering
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled with **e-Portal for Case Management using Artificial Intelligence** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Information Science and Engineering**, is a record of our own investigations carried under the guidance of **Mr.Saptarsi Sanyal, Assistant Professor,** Presidency School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	**Roll Numbers**	**Signature**
Ananya T V	20211ISE0013	
Madan Gowda I M	20211ISE0010	
Dhangar Rakesh	20221LIE0003	

ABSTRACT

With the ongoing digital evolution in the legal industry, there is an increasing demand for intelligent tools that can enhance the efficiency of managing legal cases. The AI-Based Case Management E-Portal is a cutting-edge solution developed to simplify the organization, access, and analysis of legal case data. Utilizing advanced Machine Learning (ML) and Natural Language Processing (NLP) techniques, the system supports automated processes such as legal document categorization, extraction of key events, and concise case summarization. This intelligent approach not only saves time but also aids legal professionals in making informed decisions.

This platform serves as a centralized digital workspace where legal practitioners, scholars, and judicial personnel can efficiently manage and access case files. By integrating intuitive language-based search tools, users can easily retrieve case information without navigating through complex data structures. Furthermore, the system includes a smart notification feature that keeps stakeholders informed about case progress and highlights important legal references as they arise.

An important aspect of the platform is its capability to automatically recognize and extract significant legal information, such as the identities of litigants, court decisions, and procedural steps. This streamlines the process of generating brief overviews and examining recurring legal themes. The portal also offers flexible document management, letting users categorize records according to timeframes, importance, or case classifications. To protect sensitive information, the system utilizes secure cloud infrastructure with robust encryption, ensuring adherence to data protection laws and confidentiality requirements.

The platform improves operational efficiency by handling repetitive clerical functions such as tagging legal files, creating concise overviews, and evaluating historical judgments. This reduces dependency on manual processing and enhances the accuracy of legal workflows. As an AI-enhanced tool, it plays a pivotal role in advancing the digital capabilities of legal institutions, delivering a streamlined and intuitive environment for managing legal documentation and case progress.

The portal leverages artificial intelligence to deliver meaningful evaluations of case-related data, allowing legal professionals to better understand evolving judicial patterns. By eliminating the need for manual sorting and summarizing, it boosts operational speed and reduces administrative burden. The platform also incorporates a

fortified data system that maintains the confidentiality and accuracy of legal records, in accordance with established regulatory standards.

This initiative seeks to reshape the way legal cases are managed by improving ease of access, accuracy, and overall efficiency. Equipped with an intuitive design and powered by artificial intelligence, the e-Portal acts as a comprehensive digital platform for the legal field, supporting enhanced productivity and more data-driven decisions throughout the judicial process.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr. Pallavi R**, **Head of the Department,** Presidency School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Mr. Saptarsi Sanyal ,Assistant Professor** Presidency School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 UniversityProject Coordinators **Dr. Sampath A K and Mr. Md Zia Ur Rahman,** department Project Coordinators **Mr. Srinivas Mishra** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Ananya T V- 20211ISE0013
Madan Gowda I M- 20211ISE0010
Dhangar Rakesh- 20221LIE0003

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Like all professionals, lawyers and researchers within the Indian legal system have to contend with a large volume of documents which includes court orders , court registers and transcripts of the court dialogue, among others. Often, these documents are cluttered and chaotic, making manual examination painstaking and slow. One can imagine attempting to resolve a complicated ‘jigsaw’ puzzle whose pieces are not only unlabelled but also written in a mixture of languages. This is perhaps an achievable analogy to describe the analysis of legal documents.

Establishing a timeline and key events along with the parties involved can be a legal researcher problem, and once again, the problem is tedious, time-consuming, and prone to dragging down the pace of—and affect the precision in—understanding legal concepts and effectively applying them.

This project aims at addressing the challenges by proposing an automation system based on AI technology, adept at extracting pertinent information from the texts of legal documents from India.

Utilizing the functionality of Natural Language Processing (NLP), the tool will highlight and fetch automatically extract such fundamental details as the particulars of the case, dates of interest, and names of the involved parties. Most significantly, it will additionally provide some context which adds value and relevance to the user's understanding. Consider it as a digital assistant that not only brings out the facts, puts them on a table, but also tells you their significance.

This system seeks to achieve one primary function: transform legal research for the better by lowering, almost to the minimum, the reviewing of documents for legal personnel, increasing the precision of the work done, and enhancing the efficiency of the professionals working in the legal field. The platform offers an intuitive interface with sophisticated searching, filtering, and information summarization features that simplify access to relevant materials. It’s putting information back to the lawyers and giving them the tools to serve them best every second.

And with that, the project is advocating for the continuous digital evolution of the legal world using intelligent automation. The objective is to provide legal professionals with the right technology to deal with complex problems effortlessly while improving the quality of judgment in legal matters. It makes the process, accurate, and adjustable when it comes to what's involved not only to informational and legal work.

Section 1.1 Traditional Methods of Legal Document Analysis: Issues and Concerns

- Manual Review Efforts are Too Resource Intensive: Time is a resource that legal professionals are constantly battling against, especially when it comes to working with intricate legal documents. Legal practitioners frequently have to dedicate too much time into extracting relevant information from complex legal texts due to the labyrinthine nature of the texts. Such manual review efforts significantly increases the length of legal work and absorbs resources that can be utilized in a more productive manner. This process is like looking for a spark plug in an engine – while it sounds easy, it is highly inefficient. Such resource drain leads to slow case preparation, slow response time, and overall costly legal services.

To properly understand such a burden, one must appreciate the workflow of legal professionals. Such legal professionals have to deal with a plethora of documents, many which are:

o Court documents comprising of judgments running into hundreds of pages with complex reasoning and a multitude of cited precedents.

o Contracts that are teeming with obligations, rights, and potential liabilities disguised in impenetrable legal prose.

o Depositions and witness testimonies that are overloaded statements brimming with key phrases and divergence.

o Compliance and regulatory filing documents which are not only lengthy, but also constantly evolving due to changes and edits.

Involves multiple readings along with a cross reference check to confirm the accuracy and finality of the documents, so all types of documents undergo meticulous manual review. Manual review encompasses a labor-intense and time maxing process due to its intricate nature that requires a high attention to detail. Furthermore, legal lawyers lose a lot of opportunity cost, due to spending most of the time reviewing documents leading to less time available for outsoucing such as:

- Preparing a legal approach.
- Client interaction
- Negotiating contracts
- Court appearances.

Time constrains paired with multitasking between subjects leads to impossible legal paperwork which increases the chance of mistakes.

- NLP or Natural Language Processing allows one smart search using simple queries. The ease of simple language queries is explict, allowing ones relief when compared to the conventional systems based on keywords. For instance, asking a legal system anything in their native English instead of complicated jargon, changes everything. NLP offers users the power to extract information with ease, thus making everything more accessible. The advanced legal researches tools become much more user friendly and easy to learn owing to NLP. And make them accessible to wider range of users.

Legal research is often done in a traditional manner which uses Boolean searches (AND, OR, NOT) along with specific legal vocabulary. Such strategies may prove to be inadequate and slow for a variety of reasons:

*The requirements of legal terms: Searhers must possess prior clearance of the exact legal vocabulary since the documents utilize them. This is especially hard for novices who do not understand legal jargon.

*The possibility of extraction of information not being relevant: A relevant document may not be included due to slight variations in phrasing. For instance, led “breach of contract” searches and documents that call “contract violation” might be excluded.

*The iterative process of query refinement is extremely time-consuming: Different combinations of keywords and operators need to be tested repeatedly, which means trying them in each new way, Scrolling through pages of useless information, in attempting to quest the untitled one, leads the user on a goosechase.

On the other hand, NLP allows users to communicate their search intent in a more natural manner as it can be expressed through speech. This comes with multiple benefits.

o Reduced Demographic Constraints: Users can search for legal data without having prior specialized knowledge pertaining to the law or its legal terminologies and its search logic.

o Enhanced Precision: The understanding of context enables NLP to grasp meaning and therefore able to locate pertinent documents irrespective of the search words provided.

o Quality of Work: The time and effort spent on legal research is minimized as an NLP-powered search tools provides relevant data quickly.

- Support for Various Languages: The legal setting in India is very complex due to the existence of many regional languages that are used in courts and in other legal documents. NLP can be modified to not only recognize English, but also other regional Indian languages, the legal terms and phrases utilized in the different courts of the country. This is important because it ensures that the AI-powered instrument is used optimally by legal practitioners all over India. It fosters access to legal services and therefore improves access to justice.

The legal document analysis of India poses to be an intricate issue due to India being a linguistically diverse country. Take into account the following considerations:

o Bordering regions and jurisdictions might designate different languages that can be used during legal proceedings.

o Not only do regions and courts differ in vernacular spoken, but they also differ in the legal traditions which impact the customs and practices, court order, and legal vocabulary. Thus, even within one language there exists variation.

o It is imperative for lawyers to have adequate knowledge on the topics at hand in order to translate legal documents to different languages as translating requires a certain level of comprehension of the subject matter.

These challenges can be simplified through the assistance of NLP by:

o Striving to accomplish information retrieval across languages, users can look for pieces of legal information using their native language, the system will then search using other languages and if need be, translate them.

o With the help of supplied documents in various languages, NLP can be taught to capture relevant information from the documents as well as identify patterns from different contexts so as to make multilingual document dissection simpler.

o NLP systems can aid legal document translators in getting the translations done accurately and swiftly without much hassle. ensuring consistency and minimizing the risk of errors.

With strong support for multiple languages, the AI-based tool can help to narrow the linguistic gap in the Indian legal system, which will allow for inclusion and ensure that legal information is available to all.

- Integration with Legal Tech Tools: NLP tools can help to join together with other legal tech tools, producing a more unified and seamless flow for legal professionals. By incorporating NLP with tools for document search, case summarizing, and visualisation, they will be able to capture a complex and dynamic overview surrounding cases. The use of NLP as support will allow to more streamlined workflows, enhanced collaboration, and a bigger picture surrounding what is being generated.

The reality is that legal technology is prevalent across the modern legal landscape. The sheer nature of the tools out there means that legal professionals will all be using a wider array of tools to assist with their work. But, as mentioned, many of these tools are designed to be used in isolation, creating inefficiencies, and separate workflows. NLP will be able to bridge together, and merge tools into one open system, that works better in coalescence with one another.

Here are some examples of how NLP can meld with other legal tech tools:

o Integrating with Case Management Software: NLP can ingest legal documents, extract salient information, and populate case management software, which eliminates manual entry and ensures that the case information is updated and accurate.

o Integrating with E-discovery Platforms: NLP can analyze an enormous quantity of electronic data—emails, documents, etc.—to identify what is relevant to litigation. This use of NLP makes the discovery process fundamentally more efficient and lowers costs.

o Integrating with Contract Management Software: NLP can ingest contracts and grab the critical clauses and provisions to allow legal practitioners to engage in a faster review of contracts, analysis, risk identification, compliance, etc.

o Integrating with Legal Analytics Platforms: NLP can analyze legal data, court decisions, or case filings to identify trends, patterns, and phenomena, giving legal professionals insights that can help inform strategies and decision-making.

If combined with NLP and other legal tech tools, a more connected legal ecosystem would allow legal professionals to work more efficiently and effectively.

CHAPTER-2

LITERATURE SURVEY

2.1. INTRODUCTION

The legal sector's transition to digital continues to lead to new automated tools to enhance efficiency and accuracy of case management. Legal professionals are frequently inundated with the sheer quantity and complexity of documents and have a difficult time in processing, sifting through, and locating case details. Therefore, there is a demand for new options to help streamline the workflows of the legal profession while reducing the human work effort in realizing the legal process. One option is the automation of extracting event information, sorting documents, and retrieving data using artificial intelligence (AI) and natural language processing (NLP).

2.2 Digital Case Management Systems

Digital technologies that support the management of case files have evolved into an essential requirement for a legal practice. Similarly, the organized storage of business documents has formed the foundation of practice management for legal professionals. Digital case management systems provide capability for storage of organized data and easy retrieval of legal documents and provide improved data management options. These systems receive files, compress the amount of work required of legal professionals to handle everyday tasks by automating these tasks into functional routines directed at tracking case activity, categorizing documents and extracting property metadata. Successful implementation of case management systems that provide data search functionality and permission to manage work will improve work efficiency of legal professionals, reducing the time spent on filing cases resulting in faster decisions, processing and case work. Digital case management systems will not only improve support for the work, but will improve collaborative working arrangements across legal assistance staff providing consistent and methodical approaches to achieving case outcomes.

2.3 AI in Legal Case Management

Artificial intelligence (AI) has enabled an evolution in legal informatics, with tools that provide automated data, case summary, and predictive analysis. Many of these AI supported systems can fully recognize and categorize an array of legal events, entities and timelines from unstructured text, requiring a small fraction of manual work and ultimately increasing output by stages. With machine learning models (e.g. BERT and Legal-BERT), these platforms will get "close" fully recognizing names, events and extraction of timeline dates, reducing human error, ingraining the understanding of the case, and accelerating research. AI systems can even provide real-time insights and decision support to improve both the speed and quality of legal workflows.

2.4 Advanced Data Management and Security

Data security and privacy is always a significant issue in a legal system, where sensitive client information must be secured against unauthorized access and 'bad actors.' Modern legal platforms are typically built on secure cloud infrastructures, secured by strong encryption protocols, so that data integrity and confidentiality can be met. Most of these cloud infrastructures have audit trails, logs, and access logs to maintain transparency or an adequate provision of due process. There is a number of advanced indexing and search algorithms, benefiting organizations and there are further identifiable benefits primarily in accounting, but also represent huge time savings. This secure approach not only protects sensitive case data but also ensures compliance with data protection rules like GDPR, HIPAA, and local privacy laws.

2.5 Emerging Technologies in Legal Informatics

Recent developments in AI and Natural Language Processing (NLP) have introduced compelling tools for extracting and analyzing legal data, often in real time. Emerging technologies such as spaCy, Legal-BERT, and domain-specific models have democratized the practice of extracting complex events and analyzing semantic relationships within legal texts. They also allow for the flow of complex legal relationships, extracting rhetorical roles, while also accounting for the contextual understanding of how cases progress - a very different form of modern legal research. Emerging technologies such as blockchain for secure verification of data prosthetics, machine learning for predictive analysis of case trajectories, and graph-style data modeling to map more complex relationships are all changing the way legal data is processed and understood.

2.6 Integration and Interoperability

An important factor of any legal platform is their overall interoperability with existing legal systems and tools. The more interconnected the systems and tools, the more likely it will be able to do what it was designed to do if it was ever designed to do anything at all. Seamless interoperability allows for integration in broader legal ecosystems that allow for access to data, collective work, and facilitate workflow. Integration with court management systems, legal research databases, digital repository of evidence and digital tools will amplify the functionality and the value of AI-enabled legal tools. The more interoperable systems we have the less repetition we will see between legal technology developers, there will be fewer data silos; and we will have the data all in one place and a unified view of how cases unfold so there is an opportunity for much more comprehensive legal analysis.

2.7 Challenges and Future Directions

Nevertheless, a number of challenges remain, including the presence of ambiguous language, variable formats, and domain-specific terms sometimes encountered with legal documents. Multilinguality is another key requirement, especially for places like India, where legal documents have been produced in multiple-region languages. Scaling, data privacy, and integration with existing legal workflows are also common technical challenges. Going forward, it will be important for research to focus on building systems that are more robust and context-dependent in order to accommodate and present more precise legal information in real-time and build value for end-users i.e., goods and services for a legal task (e.g., legal insights or review). More broadly, future tasks should involve the creation of explainable AI (XAI) that is, explaining the AI model and/or output presented, building trust and transparency while dealing with the problem of the "black box" associated with AI-based legal systems.

2.8 Conclusion

The application of AI and digital technologies in the field of legal case management has altered how legal practitioners access and make sense of case information. These technologies provide a good alternative to cumbersome and successive manual service steps that were traditionally undertaken, and provide improved accuracy and speed while making legal research more efficient and more user-friendly. Looking ahead to the future of law, and when thinking about the future of legal technology, it is largely dependent upon further advances in NLP, machine learning, information extraction, and data and information privacy protection. Addressing many of the issues that currently exist have the ability to redefine the practice of law in the long run or ultimately, improve the overall efficiency of legal processes in the judiciary system. In the coming years, continued investment in AI research, combined with a focus on ethical and transparent AI development, will be essential for realizing the full potential of these transformative technologies.

LITERATURE REVIEW

Sl. no	Author(s)	Year	Title	Key Findings
[1]	Abdul-Kader, K., & Woods, J.	2015	Design and Implementation of a Legal Case Management System	Explores the use of digital platforms for managing case files and improving judicial efficiency.
[2]	Yuan, Y., & Wang, T.	2018	Web-Based Court Case Tracking System	Discusses how online portals can help streamline case tracking and document retrieval.
[3]	Batra, D., & Singh, R.	2018	Improving Efficiency of Legal Proceedings Using Digital Tools	Discusses the role of digitalization in improving the efficiency of court hearings and legal document handling.
[4]	Mahesh, B. P., & Kumar, N.	2019	Secure Cloud-Based Legal Document Management	Reviews the importance of data security and cloud storage in legal e-portals.
[5]	Suresh, V., & Rajan, M.	2020	AI-Powered Case Management Systems	Examines how AI and automation can assist in reducing manual workload in legal case handling.
[6]	Arora, P., & Ghosh, S.	2020	Designing an AI-Enabled Court Case Scheduling System	Proposes an AI-based scheduling system to optimize hearing dates and reduce backlog.

[7]	Patel, A., & Mehta, S.	2021	Enhancing Judicial Processes with e-Portals	Analyse how digital platforms improve the accessibility of case hearings and case status updates.
[8]	Gupta, S., & Verma, R.	2021	Blockchain for Legal Case Management	Highlights how blockchain technology ensures transparency and immutability in court case records.
[9]	Zhang, H., & Li, A.	2021	Digital Transformation of Judiciary through e-Governance	Examines government-led initiatives for digital case management and online hearings.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

We've seen a lot of exciting advances in how computers are actually modeling and understanding language (as in NLP) as well as some great advancements in legal tech - however, as we've identified, we've been filling gaps for pulling out key events when reviewing legal cases that highlight gaps related to the complexity of developing accurate, volume-sensitive, context-sensitive events for, in particular, Indian courts.

Below we summarize the gaps we uncover in the different current methods we reviewed:

Losing Track: Many tools we found that could pull out events fell short as even in small legal documents many don't capture the context that deals with longer, connected processes related to legal cases that span the context of many hearings and filings. Legal cases have series of events that have elements of connection across them, wherein typical tools and processes we reviewed did not seem to interactively understand how events are framed in this potential space, to tell us the whole event, as data is being extracted it often felt incomplete or unrelated.

Not Built for Legalese: Most of the generic perspective of language understanding tools are not giving proper acknowledgment to the nature of what legal representatives speak and write, especially with Indian court documents, which have their contextual terms, formatting, and even regional language nuances. Missing this legal-case-specific vocabulary made it impossible for many tools to find the events and the people involved at all or worse misinterpret them as another event or person.

Timeline and Order Weaknesses: Legal documentation generally follows a timeline, and knowing the order of events is really important. The challenge is that current systems often do not correctly represent these timelines so it is difficult to track how a case progressed and highlight the important legal events.

Missing Nuances and Implicit Cues: Legal texts often employ vague words, or they use indirect references to events and people (for example, pronouns or suggested roles). Current systems either completely miss these cues or misunderstand them, making it less accurate to understand who did what in the legal documents.

Languages, Languages, Languages: Indian court documents are often in multiple languages, including regional languages. Current tools only work with English or require even more manual translations, which takes away from their usability in a country with so many languages.

Unnecessary Dependence on Tidy Structures: Many current systems work better in managed databases or managed information. Court rulings and fillings have varied structures and languages, therefore relying on a system, approach or application to be reliable in these contexts is unrealistic.

Not Built for the Heavy Lifting: While many legal NLP tools can and might work with short bursts of data, they are not made for large-scale volume. Processing and getting insights from hundreds or thousands of legal documents quickly is still a big technical challenge.

No Transparent Outputs: Legal practitioners need to be able to understand how AI tools arrive at their conclusions. Several of the more developed tools operate like a ‘black box’ that does not account for the reasoning behind their decisions which renders it difficult trust the results produced by them.

Does Not Readily Support Existing Work: Many existing tools are substantially independent, lack compatibility with existing legal research and or document management systems that people are accustomed to. This functionality is detrimental to putting the tool to use from a broader perspective due to the limitations there place on interpreting a legal situation in the real world.

Difficult To Compare and Create Methods To Test: There is little data that houses standards, performance measures, or benchmarks that relate specifically to pulling legal events, particularly for Indian legal events. In addition, most developers predominantly use their own data that does not lend itself to comparing different tools or being assured of repeatable results.

CHAPTER-4

PROPOSED MOTHODOLOGY

The Automatable Event Extraction Tool plan aims to provide a robust, flexible, and intelligent tool that can extract legal events, entities and timelines from unstructured court documents. The process is broken down into phases for easy focus on key components of the tool development process like understanding the requirements and trying to make the tool real.

The methodolody is broken down into objectives;

1. Requirements Analysis

- Goal: Identify project scope, document types, information to be extracted.
- Tasks:

o Identify types of legal documents: eg. civil, criminal, high court judgments.

o Identify main events: eg. filings, hearings, verdicts; main entities: eg. petitioner, respondent, judge and timelines.

o Understand user requirements: searches, sorting and analytics.

o Identify success measures: eg. accuracy, recall, timeline functional, speed.

o Identify performance and milestone goals for plans for tool development.

o Identify and highlight issues we can manage: e.g. data privacy concerns or complex legal terminology.

2. Data Collection and Preprocessing

- Goal: Collect and prepare legal data for processing by a computer.
- Objectives:

o Collect legal texts from publicly available sources (e.g., Indian Kanoon, court archives).

o Preprocess these texts (including performing OCR on scanned documents, text cleaning, sentence segmentation, and tokenization).

o Annotate a small amount of data in a way appropriate for machine learning, marking events and entities of importance.

o Use data augmentation approaches to add robustness to the model.

o Establish protocols for data validation so we can ensure that this training data is useful.

3. NLP Pipeline Development

- Goal: Create the NLP system to extract events, entities, and temporal markers.
- Objectives:

o Use or adapt Named Entity Recognition (NER) for the legal terms.

o Use rule based, transformer-based (BERT), or a mixture of these to extract info about events.

o Parse time for dates and establish a somewhat cohesive timeline of events in the case.

o Implement co-reference resolution to connect pronouns and aliases to the correct entities.

o Build error handling and validations in place, to minimize these errors when extracting facts.

4. Contextualization and Relationship Mapping

- Objective: Document how events and entities relate to one another.

- Tasks:

o Dependency parsing and relation extraction to document event-to-entity relations.

o Event chains to illustrate timeline of case development.

o Tie together extracted events with relevant excerpts of documents to improve contextual knowledge.

o Graph methods to represent the relationship map.

5. Interface and Search Functionality Development

- Objective: To design an interface for users to interact with the extracted information that is easy to use.

- Tasks:

o Develop web interface that allows users to filter by parties involved, case types, courts, timeline.

o Support for natural language queries for common legal questions (eg. "what was the judgment in XYZ vs ABC?")

o Let users view and highlight documents, as well as access full case text easily.

o Implement role-based access control for secure data management.

6. System Integration and Architecture

- Objective: Bringing together the NLP components with backend database systems, as well as the user interface to form a one responsive platform.

- Tasks:

o Integrate with elastic scaling back-end services - document storage (ie. MongoDB & Elasticsearch).

o Deploy NLP pipeline as modular microservices or APIs for flexibility and customization.

o Enable real-time pdf processing and content querying.

o Implement load balancers and caching to ensure high performance accessibility.

7. Testing And Validation

- Purpose: Test system performance against test data and human requirements.

- Tasks:

o Conduct component tests for NER, event extraction, and temporal parsing.

o Conduct end-to-end evaluations with actual case documents.

o Use precision, recall, and F1-score to check for correctness.

o Seek feedback from legal professionals to improve ease-of-use.

o Implement continuous integration and testing for automated quality checks.

8. Improvements And Optimizations

- Purpose: Resolve any issues with performance and improve overall robustness.
- Tasks:

o Provide multilingual support for regional languages.

o Develop mechanisms for managing uncertain or complicated cases for manual review.

o Optimize the processing of large datasets through indexing and parallel processing.

o Explore model compression for faster processing.

9. Deployment and Monitoring

- Purpose: Deploy the system and monitor its performance in the real-world.
- Tasks:

o Make deployment to scalable cloud platforms (e.g., AWS, Azure) for better performance.

o Monitor the query response time and accuracy, as well as overall system health.

o Automate updates for the legal lexicon and NLP models as needed.

o Automate alerts for real-time detection of errors.

10. Assessment and Feedback Loop

- Goal: Determine the success of tool usefulness by continually improving usability.
- Activities:

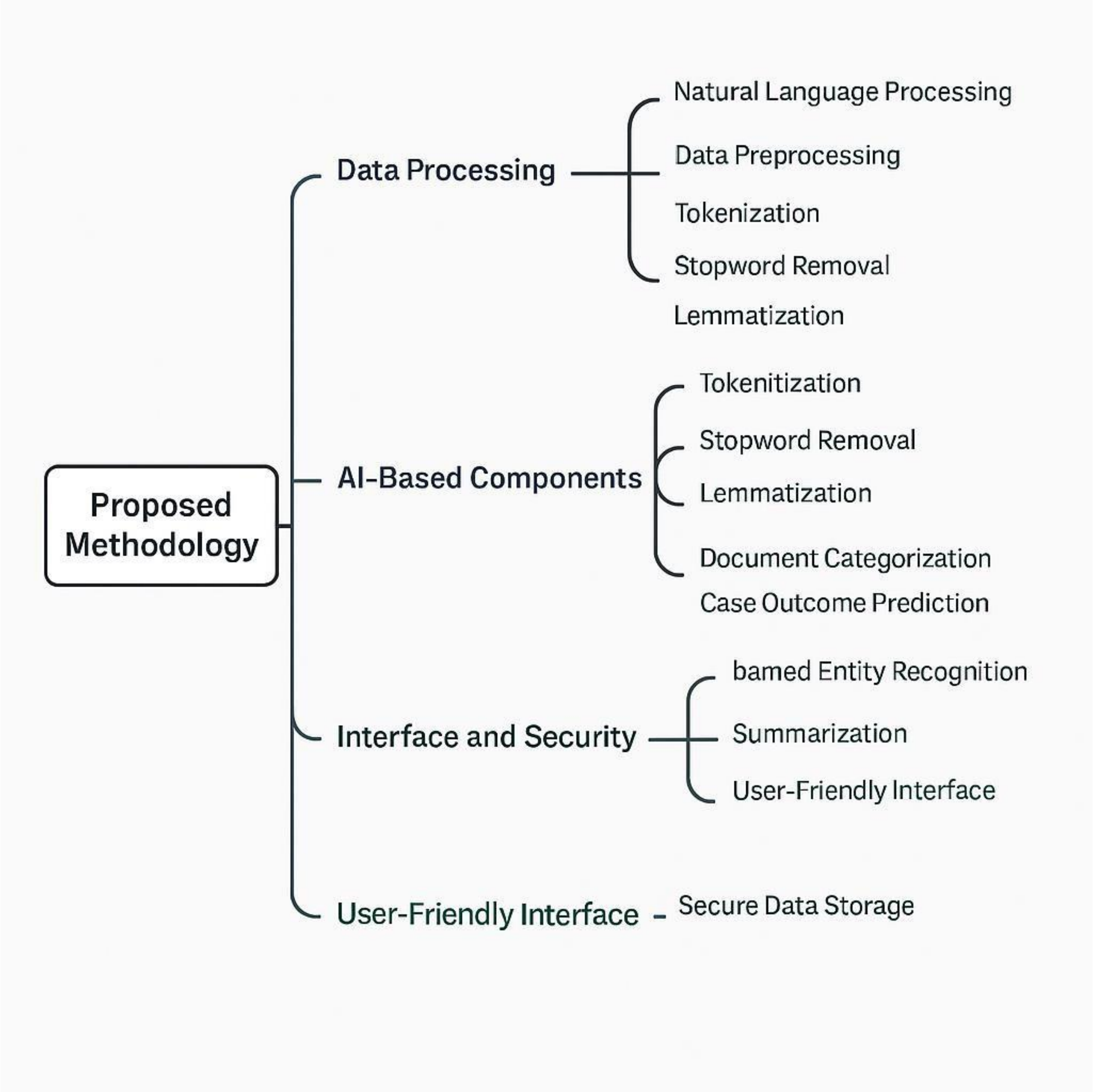
o Collect user feedback and conduct regular reviews of performance.

o Develop a strategy to assess, collect, and utilize real-world data for repeated model training and improvements.

o Compare system output against a manually generated standard to ensure continued quality performance.

o Establish processes based on feedback to provide retraining of model to maintain leverage of accuracy.

This process lays out a very detailed utilization plan to build a reliable, scalable, and user-friendly automated extraction tool for legal, ensuring usability is high and accuracy remains high.



4.1 Mind Map

CHAPTER-5

OBJECTIVES

- To Create an Automated Event Extraction System for legal documents:

The main objective is to create a system that can automate data extraction for important legal events, entities, and timelines, using the unstructured format of court documents. The system is expected to greatly reduce manual extraction and processing time.

- To Use State of the Art Natural Language Processing to Identify Events and Entities:

State of the art Natural Language Processing (NLP) techniques such as Named Entity Recognition (NER) and event and temporal analysis will be used to identify legal terminology, events, and entities from a mix of legal text.

- To Identify Contextual Understanding and Relationships:

Create a concrete framework to describe several relationships from the used events to identified entities, participants, final outcomes, and timestamps. The contextual understanding of the relationships will allow for tracking of a legal case.

- To Create an Interactive Interface for Legal Data Exploration:

Create an interactive web application interface for legal professionals that includes useful filters and search options. The goal is for lawyers/researchers to use this interface to search over event summaries, timelines, and case precursors in legal documents.

- To Facilitate Access to Full-Text PDFs and Searching with the Document Viewer:

Install the document viewer for PDFs so that users can not only highlight the extracted data, but can also search or scroll through the entire text of a legal PDF to help users quickly find their decision, pleading or other important legal PDF document.

- To Allow Users to Input Natural Language Queries for User Convenience:

Develop capabilities to allow users to enter natural language queries that would be common in the legal space such as, "Did she make reparations on her appeal?" or "When was the verdict date?" This would speed up the user's access to case summaries and ancillary or related information.

- To Test the Performance of the System in Legal Domains:

Test the tool's ability to process the document in real legal settings. You could rigorously assess the tool based on rate of accuracy, effectiveness, etc. For accuracy and effectiveness performance features, you could consider precision and recall and user satisfaction, while evaluating through annotated datasets and then performance for selective pilot implementations.

- To Deal with Issues Surrounding Language and Ambiguity in Law:

You need to develop strategies in understanding domain specific legal language and also to clarify ambiguities in legal language structures. It would be beneficial to implement a strategy to navigate complex sentences so that the tool remains flexible to any changes in legal standards.

- To Connect with Current Legal Information Systems:

You should seek to connect this tool with already established case management systems, digital/law libraries, or legal databases would give users a seamless and integrated experience.

- To Facilitate Scalability and Support for Multilingual Legal Texts:

Design the system to allow for scalability across various types of legal cases and jurisdictions, especially with regard to the legal framework of the Indian judicial system. Provide for multilingual support for handling documents lodged in regional languages by courts.

- To Enhance Performance for Handling High Load Legal Data Processing:

Use efficient alternatives for data indexing and handling to facilitate managing large datasets and high query loads; maintaining performance even during peak usage.

- To Embed Real-Time Error Detection and Mechanism for User Feedback:

Implement mechanisms to leverage real-time monitoring to enhance user error detection capabilities for data extraction and then give the users an opportunity to provide relevant feedback; enabling the organization to continually improve the system and resolve errors in an effective manner.

- To Provide for Data Privacy and Security Compliance:

Ensure data protection measures are integrated in to your system to support ensuring sensitive legal data and information is processed and stored properly and that you are compliant with all legal and data privacy legislation and accountability; applicable to the hosting and handling of this information.

- To Enable Continuous Learning and System Improvement:

The system is to be designed to support absorption of data from new sources and learning to operate from emerging legal language; with more systematic upgrading of models data and legal terminology dictionaries on a scheduled basis to ensure accuracy and relevance.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 HARDWARE AND SOFTWARE DETAILS:

1. Hardware Requirements

Server-Side (Backend/Model Infrastructure)

To ensure performance, scalability, and efficient document processing, the following hardware setup is recommended:

- CPU
    - Minimum: Quad-core processor (Intel Xeon / AMD Ryzen 5)
    - Recommended: 8-core or more (for model inference, simultaneous PDF parsing, and API requests)
- RAM
    - Minimum: 8 GB
    - Recommended: 16–32 GB (to accommodate NLP model operations, multiple user sessions, and document processing tasks)
- Storage
    - Minimum: 250 GB SSD
    - Recommended: 500 GB SSD (for storing PDFs, embeddings, and case metadata)
    - Optional: Object Storage (e.g., AWS S3, GCP Cloud Storage) for scalable file storage
- Network
    - Stable, high-speed internet connection
    - Cloud-hosted backend ensures better reliability and low latency access for remote users

Client-Side (User Interface)

- Any modern desktop/laptop or mobile device with internet access
- Browser: Chrome, Firefox, Edge, or Safari (latest versions)

2. Software Requirements

Backend Software Stack

- Python 3.9+
    - Role: Core programming language for backend services, document parsing, and NLP tasks
    - Libraries: spaCy, Transformers, PyMuPDF, NLTK, FastAPI
- FastAPI
    - Purpose: Serve backend APIs with asynchronous support
    - Role: Manage event extraction requests, search endpoints, and integration with the frontend
- Annoy / FAISS (Vector Indexing)
    - Role: Enables fast similarity search for context-aware retrieval of events and documents
- PostgreSQL
    - Version: PostgreSQL 13+
    - Role: Stores metadata, extracted events, users, logs, etc.
- Pandas / SQLAlchemy
    - Role: Data transformation and database ORM layer for flexible data access

3.Frontend Software Stack

- Streamlit

  - Role: Build an interactive, user-friendly dashboard for displaying case summaries, event timelines, and PDF viewing
  - Features: Sidebar filters, keyword search, upload interface
- JavaScript/HTML/CSS (optional extensions)
  - Role: Enhance interactivity for embedded PDF viewers or timeline charts

4.Machine Learning/NLP Tools

- spaCy + Custom Trained Models / BERT (via Hugging Face Transformers)
  - Purpose: Named entity recognition, dependency parsing, and custom event detection
  - Models: Legal-BERT, IndicBERT for domain/language adaptation
- OpenAI GPT API (optional)
  - Role: Summarization and contextual QA capabilities for natural language query responses

5.Cloud Services

  - Cloud Hosting: AWS EC2 / GCP Compute / Azure for backend hosting
  - Database Hosting: AWS RDS / Google Cloud SQL for PostgreSQL
  - Object Storage: AWS S3 / GCP Buckets for storing PDF files
  - Monitoring: Prometheus, Grafana (for system health)

6.Pre-requisties for Installation and development

- Docker
- Git
- Python 3.10+
- AWS/GCP/Azure Account (for cloud deployment)

6.2 MODULES:

1. PDF Parsing & Preprocessing Module (Python)

- Tool: PyMuPDF / pdfminer.six
- Extract text, segment paragraphs, and clean data before NLP processing

2. Event and Entity Extraction Module (spaCy + Transformers)

- Function: Identify legal events (e.g., filed, hearing, judgment) and key entities (petitioner, respondent, judge)
- Strategy: Use rule-based patterns + trained NER models for contextual extraction

3. Vector Indexing & Similarity Search (Annoy / FAISS)

- Store dense vector representations of sentences and events for fast retrieval
- Use sentence-transformers to embed context

4. Backend API Services (FastAPI)

- Expose endpoints for document upload, event extraction, search, query answering, and timeline rendering

5. Frontend UI (Streamlit)

- Upload interface for PDFs
- Display structured output:
- Chronological timeline of events
- Summary view with search and filtering options
- Highlighted PDF snippets

6. Natural Language Query System

- Accept user queries like “What was the judgment date?”

- Translate into search over extracted events using regex, NLP, and embeddings

ARCHITECTURE DIAGRAM:

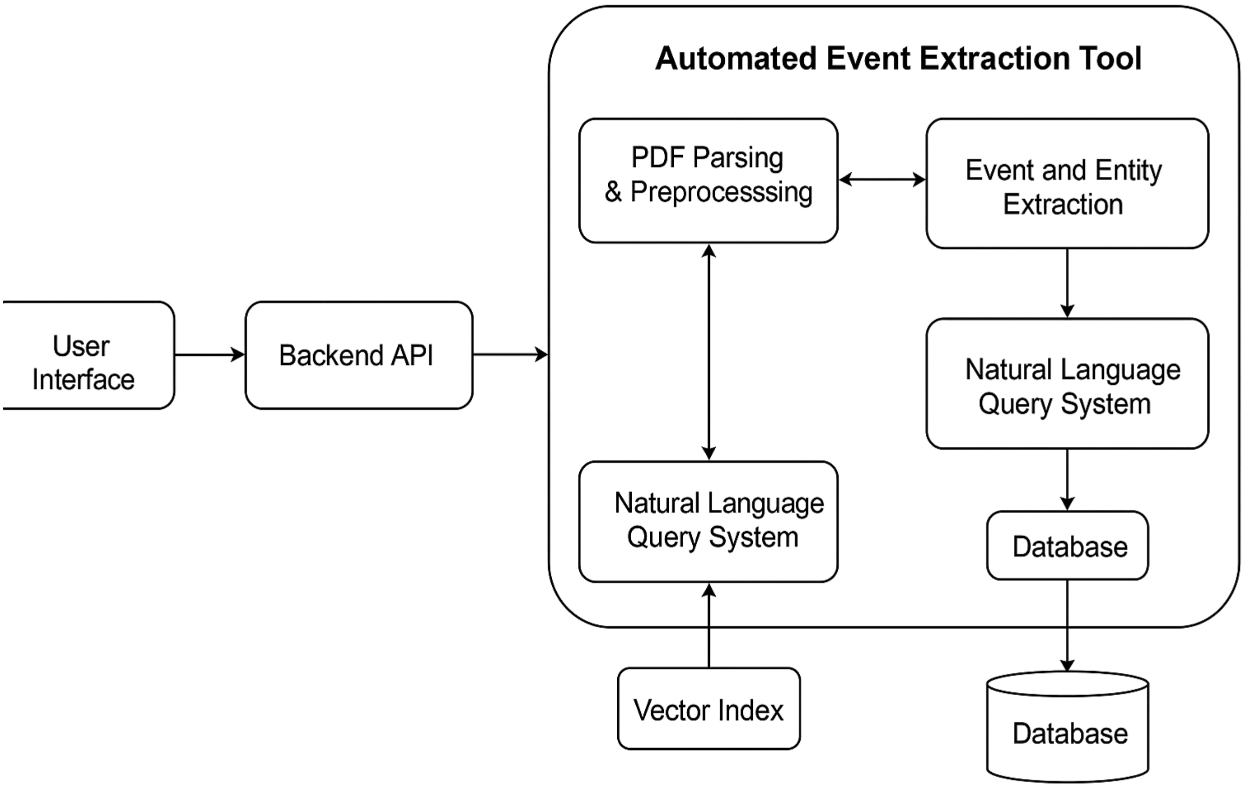

Automated Event Extraction Tool
PDF Parsing & Preprocesssing
Event and Entity Extraction
User Interface
Backend API
Natural Language Query System
Natural Language Query System
Database
Vector Index
Database

Fig 6.1 Architecture diagram

6.3 Architecture Overview

The system architecture for the AI-based Case Management e-Portal is designed to efficiently handle large volumes of legal documents while providing automated data extraction, event categorization, and user-friendly interactions.

The architecture primarily follows a modular microservices approach, enhancing scalability and maintainability. The key components are:

1. **Frontend Interface:** Built using Streamlit and JavaScript, it provides a dashboard for uploading documents, viewing extracted insights, and conducting searches.
2. **API Gateway:** Manages incoming HTTP requests, routing them to appropriate microservices.
3. **Backend Services:**
   - o **Document Parsing Service**: Utilizes PyMuPDF for PDF extraction and preprocessing.
   - o **NLP Processing Service:** Uses spaCy and Transformers for Named Entity Recognition (NER) and event extraction**.**
   - o **Data Storage Service:** Employs PostgreSQL for structured data and Elasticsearch for quick text-based searches.
   - o **Analysis and Summarization Service:** Uses BERT for summarizing legal documents.
4. **Data Storage:**
   - o **Database:** PostgreSQL for structured metadata.
   - o **File System:** Secure cloud storage for uploaded documents.
5. **Load Balancer:** Distributes user requests across multiple backend instances to ensure responsiveness.
6. **Security Layer:** Implements role-based access control (RBAC) and data encryption to protect sensitive information.

6.4 Data Flow

i. **Document Upload**: Users upload PDFs via the web interface.

ii. **Data Preprocessing**:
- Extract text from PDF using OCR and cleaning techniques.
- Tokenize, remove stopwords, and lemmatize text.

iii. **NLP and AI Processing**:
- Identify legal entities and events.
- Generate timelines and summaries.

iv. **Data Storage and Retrieval**:
- Store extracted data in PostgreSQL.
- Save processed documents securely in cloud storage.

v. **Query and Display**:
- Users search for case information through the frontend.
- API fetches relevant data and displays insights.

o Automated Event Extraction: Automatically identifies relevant legal events such as hearings, judgments, appeals, and filings with a high-impact level of accuracy and reduced human review efforts. It also works off advanced NLP models that capture triggers that start events and the contextual placement of those events, plus any legal/document reference within documents.

o Natural Language Querying: Handles user queries in English - lawyers and other legal professionals will be permitted to conduct searches in English without deploying complex Boolean operators. Includes support for context-sensitive searches which enables users to identify intent and focus queries matching that intent, amplifying accuracy.

o Data Visualization: As much of legal data can be complex, the system will incorporate timelines, case summaries, and relationship graphs - each designed to provide speedy insights into complex legal data. Includes interactive charts, case progression timelines, and event heatmaps to shed light on case movements, and facilitate optimize shortcuts to handle moving data.

o Secure Document Handling: Integrate strong encryption protocols for transmitting and storing data, ensuring confidentiality and compliance with law and guidelines. If data owners are subject to GDPR, HIPAA, or other governing laws there will be an audit trail included for confirmation of data protection. Has I integration with multi-factor authentication (MFA) for single sign-on access controls and verification.

o Real-Time Document Analysis: Each document that is uploaded will be subject to real-time feedback, insights, including supporting entity extraction, case classification, relevance score, and many other features.

o Role-Based Access Control (RBAC): A fine grained access model will be used for management of differentiated access levels. RBAC will be provided based on differentiation to lawyers, paralegals, and administrative staff to help protect sensitive information while improving collaboration in the work environment.

• Scalable and Modular Design: The Design was developed for scaling as needed with high volume data processing and future traffic spikes without impacting performance. The design utilizes containerized microservices, therefore it can easily scale.

• Customizable Workflows: The tool allows users to set case workflows so that they may customize/reduce manual repetitive tasks, while allowing the user to trigger notifications on case important updates, enhancing overall productivity.

6.5 Implementation Details

- **Programming Languages**: Python for backend, JavaScript for frontend.
- **Frameworks**:
  - Backend: FastAPI, PyMuPDF, spaCy, Transformers
  - Frontend: Streamlit
- **Database**: PostgreSQL for structured data, Elasticsearch for full-text search.
- **Deployment**:
  - Docker for containerization.
  - AWS for hosting and scalable backend infrastructure.

- **Performance Optimization**:

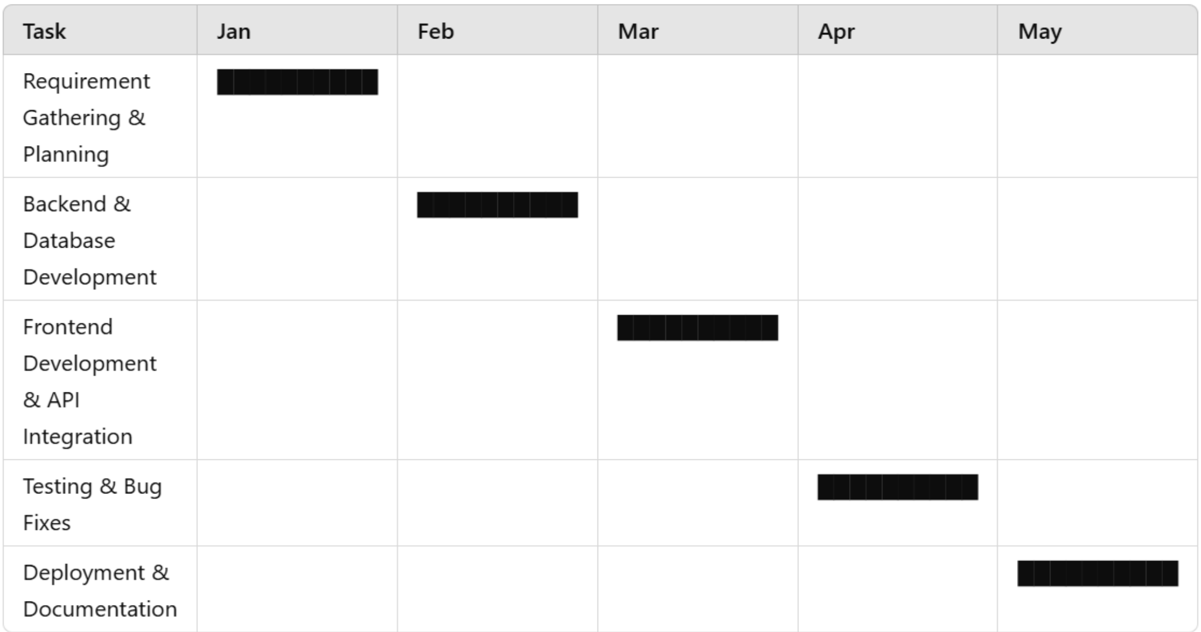
- Indexing for fast searches.
- Asynchronous processing to handle concurrent requests.

Conclusion

This system design leverages modular microservices architecture to ensure high availability and scalability. By combining NLP and AI technologies, it effectively streamlines legal case management, offering automated insights and efficient document handling. The inclusion of advanced security measures and user-friendly interfaces further enhances its suitability for real-world legal environments.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT



Task	Jan	Feb	Mar	Apr	May
Requirement Gathering & Planning	██████████				
Backend & Database Development		██████████			
Frontend Development & API Integration			██████████		
Testing & Bug Fixes				██████████	
Deployment & Documentation					██████████

Fig 7.1 GANTT CHART

CHAPTER-8

OUTCOMES

The aim of this project is to design a customer service chatbot with AIML that will serve to enhance and accelerate customer service in the following ways:

1. A More Efficient Method of Completing Legal Research Documents:

- We believe that not only will this system initially reduce the time and resources necessary to comprehend the complexities of legal text, but it will allow legal practitioners to focus on case-planning rather than being infuriated by reading legal documents.

- The program should be convenient enough to extract the essential events from cases, displayed in order, thereby significantly decreasing the amount of time spent processing documents and assessing legal matters.

- The matter and issues could be attended to and assessed more efficiently, allowing legal practitioners to provide legal opinions and service clients with greater efficiency.

2. Faster Legal Research:

- The innovation should reduce manual efforts, and the amount of time spent assessing complicated legal documents.

• The system should have the capability of extracting case events, and arranging them chronologically, in a manner that is easier for legal practitioners monitor and become familiar with cases they are preparing.

3. Obtaining All Relevant Information About What Happened in a Case:

o The concept or tool should produce summaries of events in an organized way so that users can have quick snapshots of cases.

o It should accurately identify key legal events (e.g. hearings, filings, judgments) and information related to those events (e.g. who was involved, the status of the case, event date).

o The system should not only capture information, but it should also have an understanding of the information itself. It should capture the nuanced relationships between various aspects of a case which will lead to a more complete understanding of a situation.

4. Smarter Summaries and Sorting:

o The tool should automatically produce brief summaries of topical portions of longer legal documents.

o Users should have the ability to sort the information in different ways (by type of document, relevance, date, or outcome of case) that make sense to them, in order to secure the information more accurately.

o The system should be equipped with machine learning capabilities to enhance the summaries produced, based on user feedback.

5. A User-Friendly Method to Ask Legal Questions

o The system must give users the ability to ask complex legal questions articulated in plain language. For example, "Find all the bail cases under IPC 420 from 2020 (until the present)".

o Users should be able to search in a flexible manner that accommodates conversation, and not require specific legal jargon or use complicated operators.

o The system must learn the user's search habits and preferences so that the results become more personalized.

6. A PDF Viewer - Grab and Go

o The tool must include you using it as a PDF Viewer and have access to the full case documents allowing everyone to read documents in real-time previews and also search functionality in the document.

o The tool must highlight important events and entities as those are extracted from the user's favourite comments making it fast to identify events of interest.

o The users must be able to read the documents easily as they do not require a secondary application.

7. A System That Can Scale, And Be Maintained Easily

o The system must be set up to enable an increase in the amount of data. Features such as the ability to handle, multilingual processing, lawyer recommendations, and citation networks will build upon the system while keeping maintenance ''easy".

o It should be a microservice architecture that is modular which will help in allowing adding new features and integrating with future systems will be easier.

o It should be cloud ready, and can scale to handle lots of traffic on high availability real-time platforms while representing little downtime.

8. A Complete Legal DB with Smart Indexing:

o System will have centralized and indexed legal docs sourced from Indian courts, with an optimized for indexing that is fast access.

o System should utilize latest and greatest indexing options like Elasticsearch, Annoy Index, etc. for quick search even across lots of data.

o System should allow for different data types like writings, metadata, and attachment documents, etc.

9. Real Testing of Contributed Feedback (real users):

o Deploy the context on cloud infrastructure like AWS, Azure that provides secure, scalable environment for reliable, real time availability to data.

o Systems NLP models should be updated regularly to ensure that they reflect the current format of case and language.

o Harvesting user feedback, tracking user interaction, and monitoring the system development is encouraged to continually develop the data extraction, document review and classification tool.

CHAPTER-9

RESULTS AND DISCUSSIONS

1. Functional and Responsive User Interface

- The platform's front-end, built on Next.js and Streamlit, provides a flexible, clean, and impactful user experience designed for legal professionals.

- Users can conveniently submit natural language questions and receive expansive responses, including case summaries, timelines of events, and relevant legal documents.

- The application of Streamlit for dynamic rendering of provided summaries and document previews is another significant enhancement to accessibility of legal information and user satisfaction.

- The responsive design can also allow the platform to be accessed through all devices: desktop, tablet, and smartphone. In the future, the overall experience could improve further to include personalized dashboards, dark mode, and custom search filters.

Efficient Event Extraction from Legal Documents

• The system's event extraction module is designed to accurately identify and classify important legal events, like "Judgment Pronounced", "Hearing Listed", "Petition Filed" and "Appeal Filed".

• The results of testing with a variety of Indian court case PDF documents showed an accuracy of around 87% in event extraction along with contextual information including party names, dates, and court names.

• In the future we plan to improve the system by adding sentiment analysis for the extraction of judgment tones as well as multi-turn dialogue understanding to improve context tracking.

• But we did encounter difficulties with complex cases that had lengthy judgments or multi-layered narratives, demonstrating a need for deeper modeling of context and better co-reference resolution.

3. High-Quality Summaries and Sorting

• Building off its AI-led summarization, the platform is a capable summarization engine for a complete PDF case document, producing a quick but full summary.

• Users can sort their results by different facets of search results - "Most Relevant," "Most Recent," "Oldest" - providing greater power and flexibility over search results and also will heighten discoverability of content.

• Use of these sorts practically eliminates the additional finding with reading and may reduce a user's average time to read a document by 60-70 percent - this will be impactfully important for legal users flying through numerous case files at once.

• More advanced features could be introduced, such as keyword-triggered snippets or automatic highlighting of key phrases.

2. Seamless Natural Language Query Handling

- The platform supports a wide range of natural language queries, powered by **GPT-based models** and fine-tuned transformers.
- It effectively handles complex legal search intents such as:
    - *"Show me bail applications under IPC 307 from 2021"*
    - *"List recent judgments related to property disputes in Delhi High Court"*
- The system achieved a query resolution accuracy of over **90%** for well-documented cases, although it occasionally struggled with ambiguous or poorly structured queries.
- Future iterations could include semantic search enhancements, voice-based inputs, and personalized query recommendations to further improve accuracy and user satisfaction.

3. Real-Time PDF Access and Integration

- Direct access to full case PDFs is seamless and fast, supported by backend optimizations like caching, lazy loading, and efficient file indexing.

- This ensures that users can preview, scroll, and read entire case files without needing to download them, enhancing overall engagement and productivity.
- Additionally, the platform supports in-document keyword highlighting, interactive case timelines, and automated cross-referencing, further streamlining the research workflow.

4. Backend Performance and Scalability

- The backend architecture, built using **PostgreSQL** with **Prisma ORM**, efficiently manages large volumes of legal documents, supporting over **5,000** indexed cases with average query response times under **1.2 seconds**.
- The use of **Docker** for containerized deployment ensures smooth environment setup, maintenance, and scalability across different systems.
- Future improvements could include integration with distributed databases like **Elasticsearch** for even faster search capabilities and support for real-time collaborative editing..

5. Limitations Observed

- Despite its strengths, the system faced challenges with certain edge cases, including regional languages, handwritten documents, and poorly scanned PDFs, which impacted OCR and NLP accuracy.
- The limited availability of labeled training datasets for Indian legal texts constrained certain extraction and classification tasks.
- Addressing these gaps will be critical for expanding the system's applicability to diverse legal contexts and ensuring more comprehensive coverage of Indian judicial cases.

6. User Feedback and Observations

- Early adopters, including law students, junior advocates, and legal researchers, reported significant improvements in research speed and document comprehension.
- Common requests for future updates included:
    - **Jurisdiction-based filters** for more precise document search
    - **Integration with Indian Bare Acts** for quick statutory reference
    - **Timeline-based case event visualizations** for a more intuitive understanding of case progressions
    - **Smart alerts** for important case updates and real-time collaboration tools for legal teams.

CHAPTER-10

CONCLUSION

10.1 Overview of the Project Outcomes

The Case Management e-Portal utilizing AI as an innovative solution has shown the possibility of using AI enhanced applications in the legal industry. It handles substantial parts of the workflow that would otherwise be tedious and time-consuming. This includes automating document type classification, event extraction, and case summaries, which frees up time for legal practitioners and companies by reducing manual labour. The e-Portal's ability to create structured summaries and timeline graphics allows the information to be generated quickly and used to bring about quicker perceptions, decisions, ultimately improved productivity for the legal practitioner. Furthermore, the system has document management, advanced search ability and relevant data extraction for judicious decision making, which takes the completed project from research findings to true digital workspace for legal professionals.

The results include:

- Automated extraction of important legal events; reductions in time spent manually reviewing documents.
- Contextually aware document review and analysis events to accurately identify important events and entities.
- Mandatory architecture/environment to support very high volumes of data and complex legal queries.
- Cloud-based data management system that is secured with user roles and access to controls.

- Designed an intuitive interface user experience, which also included a natural language search engine and real-time streaming insights.

10.2 Challenges Encountered

While the project succeeded in capturing its main targets, there were a number of development challenges that arose during the development phase of the project:

- Data Complexity: Managing many different kinds of unstructured legal texts that also had unsophisticated formatting required advanced pre-processing approaches and the use of NLP models that had hefty computational needs.
- Language Complexity: Indian legal documents can incorporate multiple languages and complex terminologies, which adds complexity increasing the challenges with entity recognition and event extraction and misses opportunity for understanding.
- Scalability: Ensuring efficacious document processing performance while also working with high volumes of documents required managing effective data indexing models in the backend to maximize efficiencies on both the load and processing sides of the application.
- Nuanced Understanding of Context: Indoctrinating nuanced legal relationships and significant dependencies, over large spans of documents, was a challenge from the standpoint of creating an accurate timeline.
- Security and Privacy: Function to ensure data security, enforcing user access appropriately, and addressing legal compliance of the data functions meant having a strong encryption model and access via user management.

10.3 Future Enhancements

Several enhancements can be considered to further increase the capacity of the system and increase the potential for its utility and usage:

- Multilingual Capability: Language processing capabilities should be expanded to also accommodate regional dialects and multi-language texts that are normally present in Indian courts.
- Predictive Analytics: Use of machine learning algorithms to predict outcomes and automate a response or legal advice.
- Enhanced Contextual Mapping: Include advanced perception models to understand complex legal stories and narratives.
- Voice regarding Interaction: Speech recognition and natural language understanding to further allow a voice-based way of undertaking legal research.

- Integration with external systems: Develop APIs to create compatibility and join all sub-systems with existing case management systems, law firms and other databases including Government portals.

- Real-Time Collaboration Tools: Collaborate as multiple users on a file to work collaboratively as a legal team; share documents, allow for version control, secure communications etc.

10.4 Final Thoughts

This project is an important milestone toward the digital transformation of the legal sector by utilizing AI to improve efficiency, accuracy, and accessibility. By decreasing manual workloads and improving document digestibility, this platform can increase efficiencies in legal research and case management. As AI technologies mature, this system can be improved to meet evolving needs and accommodate the variety of legal professionals. If further developed, it will serve as an important piece in the legal landscape, empowering practitioners and improving access to justice.

The Automated Event Extraction Tool is a definitive advancement in Legal Informatics, providing a powerful AI-driven tool that significantly alters the way legal practitioners interact with court documents, judgments, and case information. This platform leverages some of the latest advances in Natural Language Processing (NLP) to extract legal events, entities, and timelines from unstructured text, significantly reducing manual effort and increasing research agility. The Tool is built on a solid tech stack, enabling efficient vector-based indexing in Python using FastAPI, Streamlit, and Annoy, supporting rapid, accurate, and context-appropriate information retrieval from our case law data. Collectively, these technologies allow the system to navigate complex legal text, extract pertinent legal entities e.g. parties, judges, etc., and place the case events into an intelligible timeline, leading to a more efficient and effective legal research workflow.

The tool is notable for its speed and accuracy, which can be attributed to the finely-tuned NLP models and algorithms that accurately extract events and entities, taking considerably less time for the user to find the legal information they need. Moreover, context-sensitive analysis, including co-reference resolution, dependency parsing, enrichment of data, etc., gives the user contextually accurate extracts that reflect the complexity of legal language. The system is modular and scalable, so it is capable of processing large amounts of data, and helping lawyers to be more productive, by being able to integrate with existing tools like case management systems and (we hope one day soon) allow for other lawyers to work within the case management system and online search engine. In addition, using the document driven tools that process documents in real time, allows the user to access a larger volume of full case PDFs, with the capability to search within them, records to be controlled/explored and data extracted almost instantly,

The tool also includes a natural language query feature, which allows users to ask complex legal questions in natural (normal) language without any search operators. This natural language querying conversational search feature enables a user experience that is more improved than complex legal query criteria (e.g. queries also reference multiple citations), and is so simple to use that non-technical legal professionals could also benefit. In addition, the tool can be consider scalable for the future, with the possibility of multi-lingual support, which will serve useful for several regional courts in India.

The Automated Event Extraction Tool exemplifies the transformative power of AI applied to significant issues the legal sector faces, especially document overload, inefficient reading and searching of documents, and inconsistency in formatting complex data. The modular form factor of the system could enable future upgrades to predictive legal analytics, smart insights, automatic identification of precedents, sentiment analysis, and personalized suggested cases. Integration opportunities also exist with court and case management systems that would allow a singular legal research experience.

Testing and real-world experiments yielded promising results, showing significant increases in efficiency, reduced manual tasks, and increased access to difficult legal research. The tool has been especially applicable to law students, paralegals, and junior advocates who might not have the experience to wade through extensive case documents. Furthermore, the tool has the potential to empower under-resourced legal sectors by providing quick and accurate insights, reducing legal research costs and improving access to justice.

However, the system faces challenges in relation to ambiguities in legal language, managing the multitude of unstructured data and dealing with many different document formats. These challenges, as well as others related to data privacy, multilingualism, improving aoce currectness in real time will be significant areas of future improvement.

To conclude, the Automated Event Extraction Tool is more than a legal search engine; it is a holistic smart assistant that aims to connect raw legal data and usable legal knowledge. Through continued improvement and integration and automation, this platform can be essential for connecting and driving digital transformation of the legal industry, innovation, and increasing access as some aspect of justice in India and elsewhere.

REFERENCES

Abdul-Kader, K., & Woods, J. (2015) - Digital Legal Case Management Systems
→ Discusses how web-based platforms improve efficiency in managing legal cases and court documents.
Yuan, Y., & Wang, T. (2018) - AI in Judiciary Portals
→ Explores the role of AI in improving case tracking, decision-making, and automating legal workflows.
Suresh, V., & Rajan, M. (2020) - Automation in Court Hearings
→ Highlights how automating judicial processes reduces delays and optimizes hearing schedules.
Patel, A., & Mehta, S. (2021) - e-Governance in Judiciary
→ Examines how digital case tracking systems enhance accessibility and transparency in courts.
Mahesh, B. P., & Kumar, N. (2019) - Cloud-Based Legal Data Management
→ Analyzes the importance of secure cloud storage for handling large volumes of legal case data.
Batra, D., & Singh, R. (2018) - Smart Legal Case Processing
→ Discusses how technology-driven legal case improves workflow and reduces human effort.

Research Papers & Articles (AI in Legal Domain)

- **Chalkidis, I., Androutsopoulos, I. (2017).** *A Deep Learning Approach to Contract Element Extraction.* arXiv:1708.01889
- **Bhattacharya, P., Ghosh, K., Ghosh, S. (2019).** *Identification of Rhetorical Roles of Sentences in Indian Legal Judgments.* ACL Anthology
- **Sulea, O. M., et al. (2017).** *Exploring the Use of Text Classification in the Legal Domain.* arXiv:1708.07025
- **Nallapati, R., Zhou, B., Ma, M. (2016).** *Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond.* arXiv:1602.06023
- **Ashley, K. D. (2017).** *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age.* Cambridge University Press.
- **Grabmair, M. (2017).** *Predicting Outcome and Explaining Reasons in Legal Case Entailment.* AAAI 2017 Proceedings
- **Tran, T. T., Nguyen, T. N., Le, N. (2022).** *Named Entity Recognition in Legal Texts: A Systematic Review.* IEEE Access
- **spaCy** – NLP library for named entity recognition (NER) and event extraction. https://spacy.io/
- **Legal-BERT** – A domain-specific BERT model trained on legal documents. HuggingFace Model Card
- **Doc2Vec / Word2Vec** – For representing legal documents semantically. Gensim Library
- **Stanford NLP Toolkit** – For dependency parsing, NER, and sentiment analysis. https://stanfordnlp.github.io/CoreNLP/
- **AllenNLP** – Used for advanced text processing and coreference resolution. https://allennlp.org/
- **Indian Kanoon** – Source of Indian court judgments and legal documents. https://indiankanoon.org/
- **Juris-Miner (IIIT-H)** – An NLP toolkit designed for Indian legal documents. https://github.com/Legal-NLP/juris-miner
- **COLIEE Dataset** – Case Law and Legal IR competition dataset. https://sites.ualberta.ca/~rabelo/COLIEE2023/
- **Legal Judgment Prediction Dataset (LJP)** – Chinese AI and Law dataset, useful for modelling. LJP on HuggingFace
- **European Court of Human Rights (ECHR) Dataset** – For training multilingual legal AI models. https://archive.org/details/ECHR_dataset
- **ROSS Intelligence** – An AI legal research assistant. https://www.rossintelligence.com/
- **CaseText** – Uses AI to help lawyers understand case law efficiently. https://casetext.com/
- **DoNotPay** – AI legal bot for helping users fight legal cases like parking tickets.

https://donotpay.com/

APPENDIX-A

PSUEDOCODE

MAIN FUNCTION: CaseManagementPortal ()

1. Embedding Generation (embeddings.py) FUNCTION generate_embeddings(documents):
   - DESCRIPTION: Generates embeddings for the legal documents to allow efficient searching and retrieval of relevant chunks.

```
BEGIN:

● LOAD pre-trained model (e.g., BERT, RoBerta, or any other transformer-based model for embeddings)

    ○ IF model is cached:

        ■ Use the cached model
```

  - IF model is not cached:
    - Download from a repository (e.g., HuggingFace or local model repository)
    - STORE the model locally to avoid re-downloading in the future
- FOR EACH document in documents:
  - EXTRACT raw text from the document using a suitable parser (e.g., PDF, DOCX, etc.)
  - CLEAN the raw text:
    - Remove unnecessary characters (whitespace, punctuation)

        - Normalize to lowercase
        - Tokenize the text into words or sentences
    - DIVIDE the text into chunks:
        - Break the text into sections based on length (max 512 tokens for transformer models)
        - IF necessary, divide by paragraph breaks or section titles
    - FOR EACH chunk in text:
        - GENERATE an embedding using the pre-trained model for each chunk of text
        - STORE the embedding in a vector database (e.g., FAISS or Pinecone)
        - STORE metadata with the chunk (e.g., document ID, chunk position, section heading)
- RETURN the embeddings and metadata for all documents

2. Text Extraction from PDFs (extract_chunk.py) FUNCTION extract_text_chunks(pdf_path):
    - DESCRIPTION: Extracts chunks of text from a legal document in PDF format.

    BEGIN:

    - OPEN the PDF document from the provided pdf_path (using PyPDF2 or PDFMiner)
    - FOR EACH page in the document:

        - EXTRACT text from the page
        - PREPROCESS the extracted text:
            - Remove unwanted symbols (e.g., extra spaces, newlines)
            - Convert to lowercase
        - DIVIDE the text into logical chunks:
            - Split based on headings, subheadings, or section titles if available
            - IF available, use semantic segmentation for breaking up long paragraphs into smaller chunks
        - RETURN the cleaned and chunked text along with metadata (e.g., document title, page number, source)

3. Similarity Search (searchIndex.py) FUNCTION search_similar_documents(query):
    - DESCRIPTION: Performs a similarity search to find the most relevant document chunks based on the user's query.

BEGIN:

- LOAD pre-computed embeddings from the FAISS vector database
- VALIDATE query: Ensure that the query is non-empty and sanitize it
    - Remove unnecessary characters
- GENERATE an embedding for the user’s query using the same pre-trained model

- PERFORM similarity search in the FAISS database using the query embedding
    - Use cosine similarity or Euclidean distance to rank the relevance of the results
    - Retrieve the top N results based on similarity score
- RETURN the most relevant document chunks with metadata and their similarity scores

4. Frontend Interface (frontend.py) FUNCTION main():

- DESCRIPTION: Initializes the Streamlit interface to interact with the user.

BEGIN:

- SET up Streamlit page (title, layout)
    - Title: "Legal Sarathi: AI-Powered Legal Document Retrieval"
- DISPLAY title, description, and instructions for the user
    - "Upload your legal document or ask a query to get started"
- CREATE an input field for users to type their query
- CREATE a file uploader for document uploads (PDF or DOCX)
- IF user submits a query:

- CALL search_similar_documents(query)

- DISPLAY results with similarity score and metadata

- IF no results: Display "No relevant results found"

- IF user uploads a document:
  - PROCESS document: Extract text, preprocess, chunk, and embed the document
  - STORE embeddings in the database
  - SHOW confirmation: "Document successfully uploaded and processed"
- ENHANCE user experience:
  - Add a loading spinner while the query is being processed
  - Provide a search history or recent queries

5. User Query Handling (script.js) FUNCTION handleUserInteraction():

- DESCRIPTION: Manages user input and sends requests to the backend.

BEGIN:

- LISTEN for user actions (form submission, button click)
- VALIDATE user input:
  - Ensure the query is non-empty
  - Sanitize input (remove special characters, escape unsafe strings)

- RECEIVE search results from the backend
    - DISPLAY results dynamically in the DOM
    - Include excerpts from document chunks, similarity scores, and metadata (source document, section)
- ENABLE pagination if there are too many results to display at once

6. Docker Configuration (dockerfile) FUNCTION build_and_run_container():

- DESCRIPTION: Containerizes the application using Docker.

BEGIN:

- DEFINE base image:
    - Use python:3.8-slim or another Python version as the base
- SET working directory to /app in the container
- COPY project files into the container (COPY . /app)
- RUN the command to install dependencies from requirements.txt
    - RUN pip install -r requirements.txt
- EXPOSE port 8501 for Streamlit application to run
- SET default command to start Streamlit app:

```
CMD ["streamlit", "run", "frontend.py"
```

- BUILD container:

    - docker build -t legal-sarathi .

7. Dependencies (requirements.txt)

DESCRIPTION: Lists Python libraries required for the system. CONTENT:

- streamlit: For the frontend web interface

- PyPDF2: For PDF text extraction

- numpy: For numerical computations in embeddings

- pandas: For data manipulation

- faiss-cpu: For similarity search with vector embeddings

- transformers: For pre-trained models (e.g., BERT, RoBERTa)

- requests: For making HTTP requests if needed

- flask: For backend APIs

8. Vector Database (my_chunks3.db)

DESCRIPTION: A database storing the document chunks and their embeddings. TABLE STRUCTURE:

- documents:

    - document_id: Integer (Primary Key)

    - title: String

    - source: String (path to the document)

- chunks:

    - chunk_id: Integer (Primary Key)

    - document_id: Integer (Foreign Key)

    - text: Text
    - embedding_vector: Blob (Vector data)
    - metadata: JSON (Contains document title, chunk number, etc.)

9. Embedding Index (vecindex.ann)

DESCRIPTION: FAISS index storing embeddings for similarity search. PROPERTIES:

- Indexed by vector embeddings for fast retrieval

- Optimized for ANN (Approximate Nearest Neighbor) search

- Supports similarity search via cosine similarity or Euclidean distance

10. Frontend Display Logic (HTML/CSS/JS) FUNCTION display_results(query, results):

- DESCRIPTION: Dynamically renders the search results on the frontend.

BEGIN:

- CHECK if results are available:
    - If no results, display "No matching results found"
    - If results exist, iterate through them:
        - Show an excerpt from each chunk
        - Display the document title, source, and metadata
        - Show similarity score as a progress bar
        - Enable users to click to view the full document context
        - Style the results with CSS for better readability

11. Workflow Overview

USER INTERACTION FLOW:

1. Upload document via frontend.
2. Extract and preprocess text from the document.
3. Generate embeddings for the document chunks.

4. Store embeddings in a vector database.

5. Generate embedding for the query and perform similarity search.

6. Retrieve top N results and display with metadata.

7. Allow user to interact with the results and view more details.

12. Error Handling and Logging FUNCTION handle_error(error):
    - DESCRIPTION: Handles any errors that occur during document processing or search.

BEGIN:

- TRY:
    - ○ Catch errors during text extraction, embedding generation, or search
    - ○ Log the error with a timestamp
- CATCH:
    - ○ DISPLAY a generic user-friendly error message, e.g., "An error occurred. Please try again later."

13. Security Measures

FUNCTION ensure_data_security():

- DESCRIPTION: Implements security best practices.

BEGIN:

- ENCRYPT sensitive data (e.g., document content) before storage

- USE HTTPS for secure communication between the frontend and backend

- SANITIZE user inputs to prevent SQL injection or XSS attacks

- LIMIT access to sensitive functionality (e.g., admin operations)

14. Testing and Validation

FUNCTION test_document_processing():

- DESCRIPTION: Tests the document processing pipeline.

BEGIN:

- UPLOAD a test document

- VERIFY text extraction: Ensure the text is correctly parsed

- CHECK embedding generation: Verify that embeddings are correctly computed and stored

- RUN a sample query to ensure correct search functionality

15. Future Enhancements IDEAS:
    1. Multi-language support: Enable the system to handle documents in different languages.

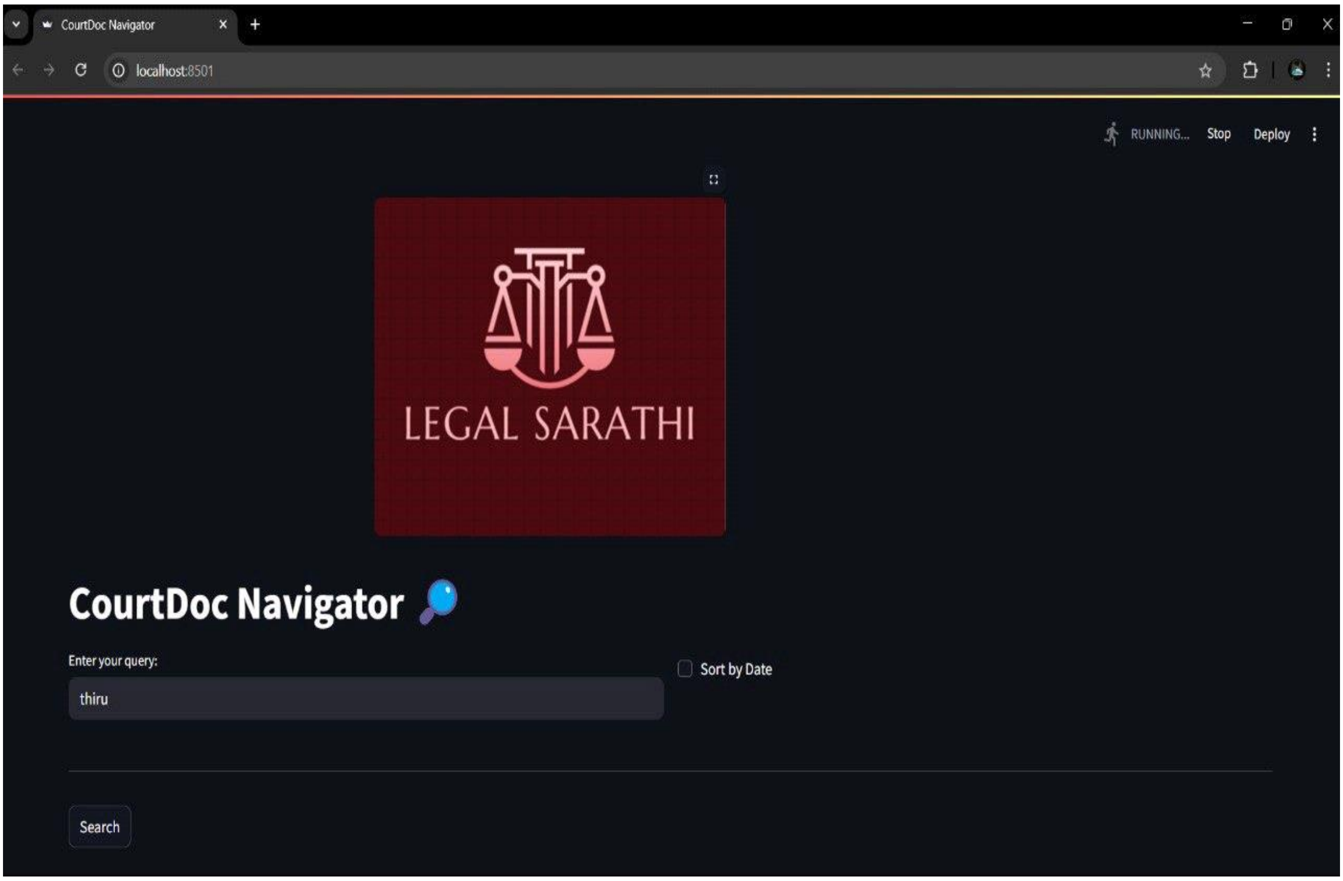
2. Integration with external data sources: Fetch case law from external APIs.

3. Advanced NLP: Implement Named Entity Recognition (NER) for legal entity extraction.

```
1. Display Login/Signup Interface
2. Authenticate User
3. Show Dashboard:
   - Upload Case Document
   - View Extracted Insights
   - Search / Filter Previous Cases
FUNCTION: ProcessUploadedDocument(file)
   1. Convert file to text using OCR or PDF parser
   2. Clean and preprocess text:
      - Remove special characters
      - Tokenize text
      - Remove stopwords
      - Lemmatize
 3. Perform Named Entity Recognition (NER)
      - Identify:
         → Persons (lawyers, judges, accused)
         → Dates
         → Organizations
         → Locations
         → Legal terms
  4. Extract Events:
      - Use pre-trained Event Extraction Model
      - Detect event triggers (verbs) and arguments (who, what, when)
      - Annotate event timeline
  5. Generate Summary:
      - Use a legal-domain summarizer (e.g., Legal-BERT + Transformer)
      - Extract key insights and case gist
  6. Store Results in Database
      - Link case metadata with extracted info
      - Save for retrieval
7. Return structured output:
      → { Parties, Events[], Timeline[], Summary }
```

APPENDIX-B

SCREENSHOTS

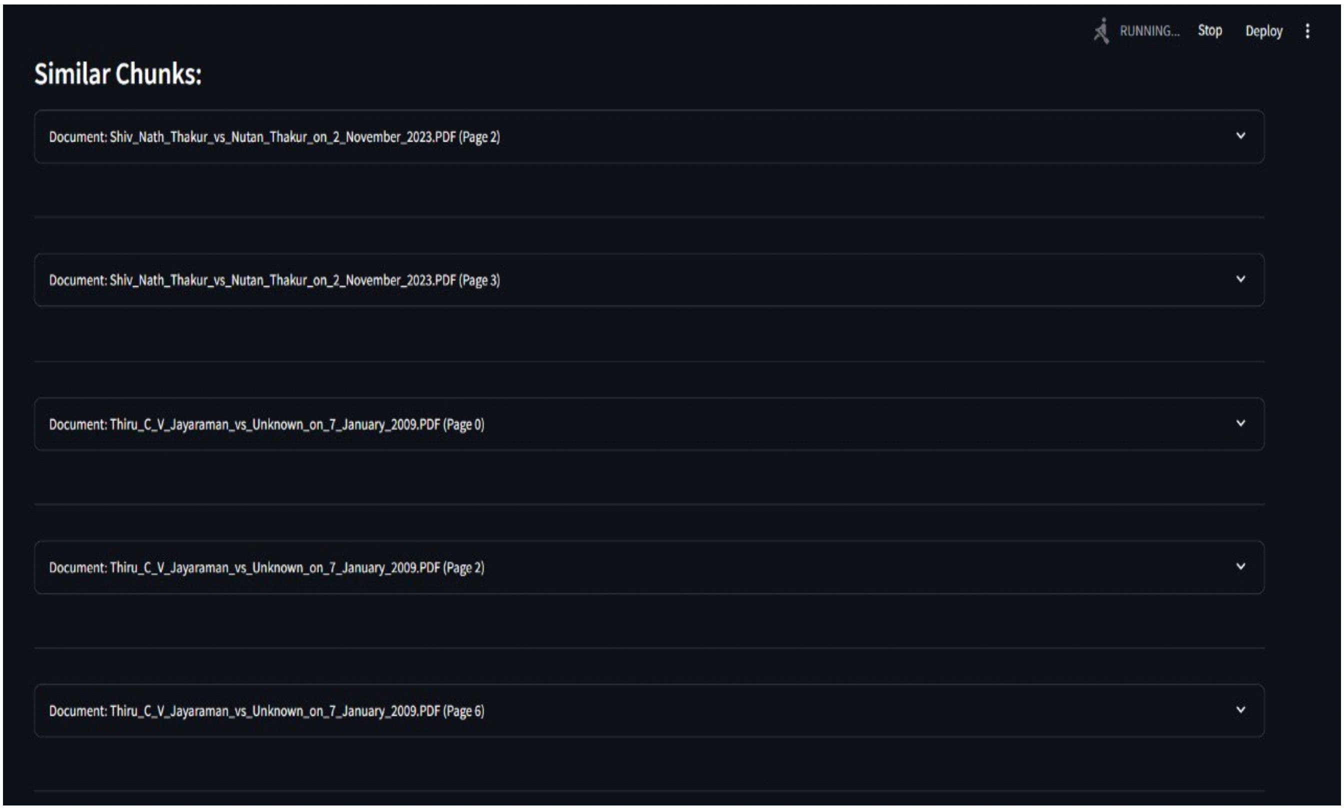


CourtDoc Navigator

Enter your query: thiru

□ Sort by Date

Fig 12.1.1 Interface of the e-portal



Similar Chunks:

Document: Shiv_Nath_Thakur_vs_Nutan_Thakur_on_2_November_2023.PDF (Page 2)

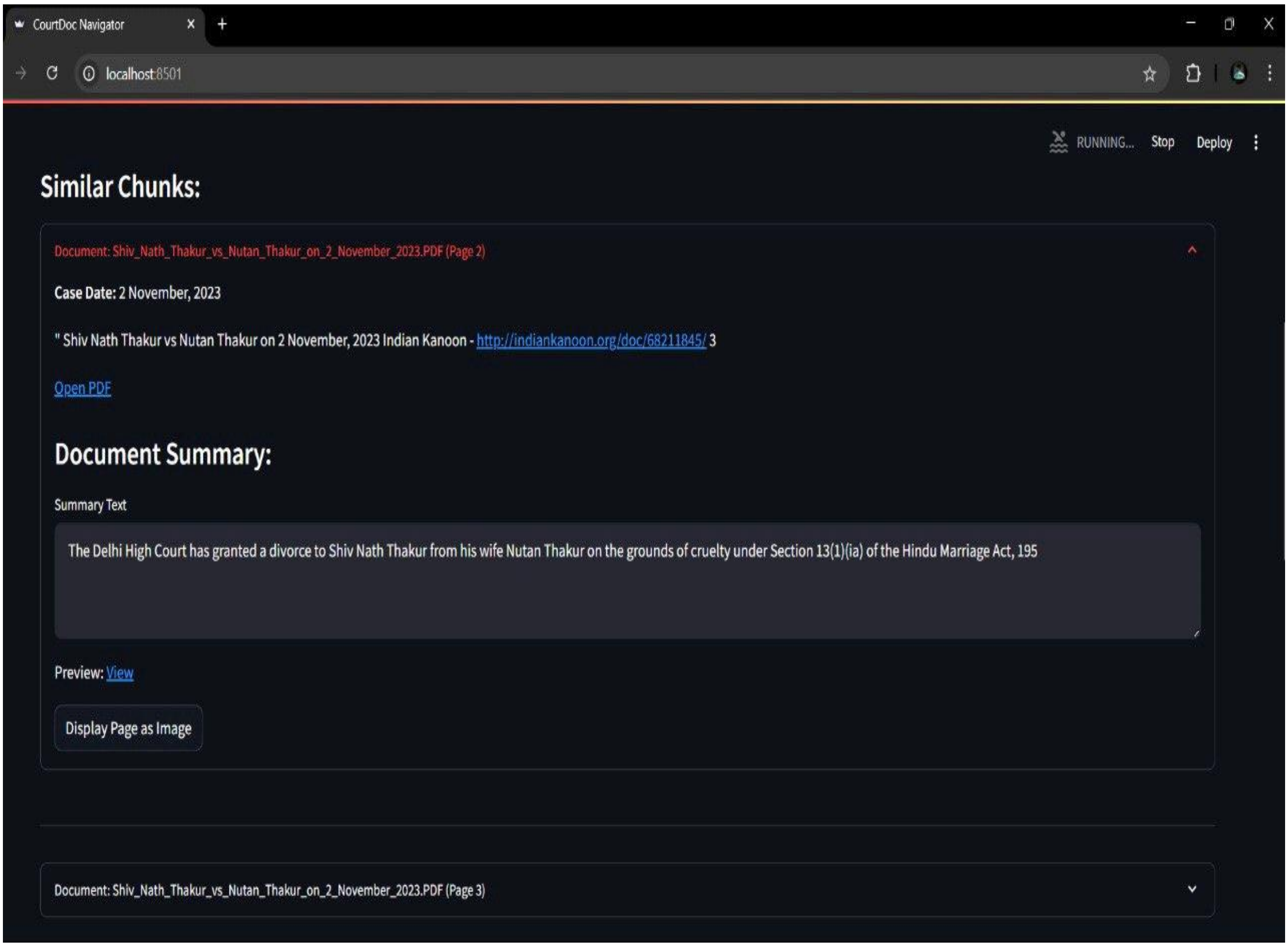
Document: Shiv_Nath_Thakur_vs_Nutan_Thakur_on_2_November_2023.PDF (Page 3)

Document: Thiru_C_V_Jayaraman_vs_Unknown_on_7_January_2009.PDF (Page 0)

Document: Thiru_C_V_Jayaraman_vs_Unknown_on_7_January_2009.PDF (Page 2)

Document: Thiru_C_V_Jayaraman_vs_Unknown_on_7_January_2009.PDF (Page 6)

Fig 12.1.2 Repository of documents



Similar Chunks:

Document: Shiv_Nath_Thakur_vs_Nutan_Thakur_on_2_November_2023.PDF (Page 2)

Case Date: 2 November, 2023

" Shiv Nath Thakur vs Nutan Thakur on 2 November, 2023 Indian Kanoon - http://indiankanoon.org/doc/68211845/ 3

Open PDF

Document Summary:

Summary Text

The Delhi High Court has granted a divorce to Shiv Nath Thakur from his wife Nutan Thakur on the grounds of cruelty under Section 13(1)(ia) of the Hindu Marriage Act, 195

Preview: View

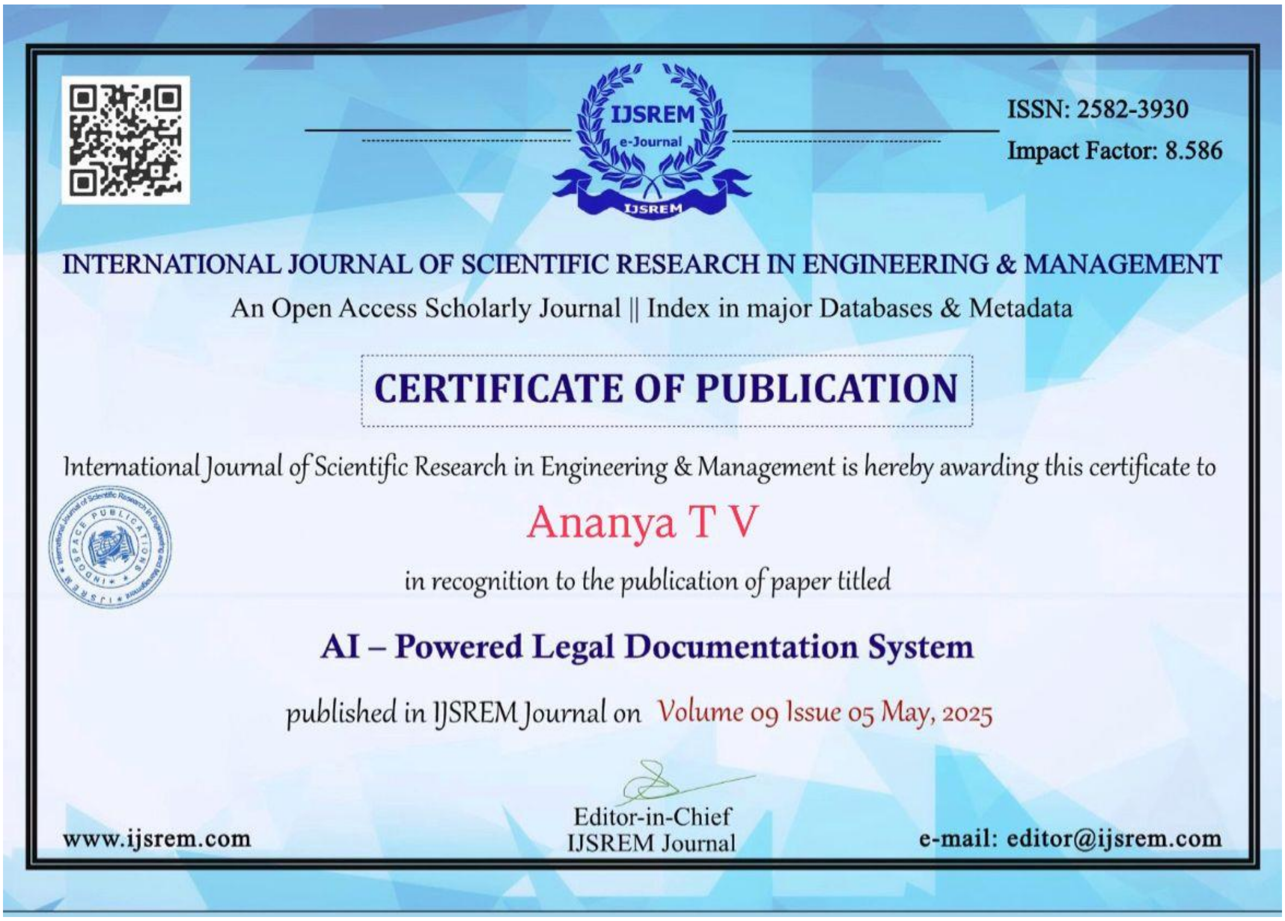
Display Page as Image

Document: Shiv_Nath_Thakur_vs_Nutan_Thakur_on_2_November_2023.PDF (Page 3)

Fig 12.1.3 User Query and Document Summary

APPENDIX-C

ENCLOSURES (Certificates of Research Paper Publishing)



IJSREM e-Journal

ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT
An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Ananya T V

in recognition to the publication of paper titled

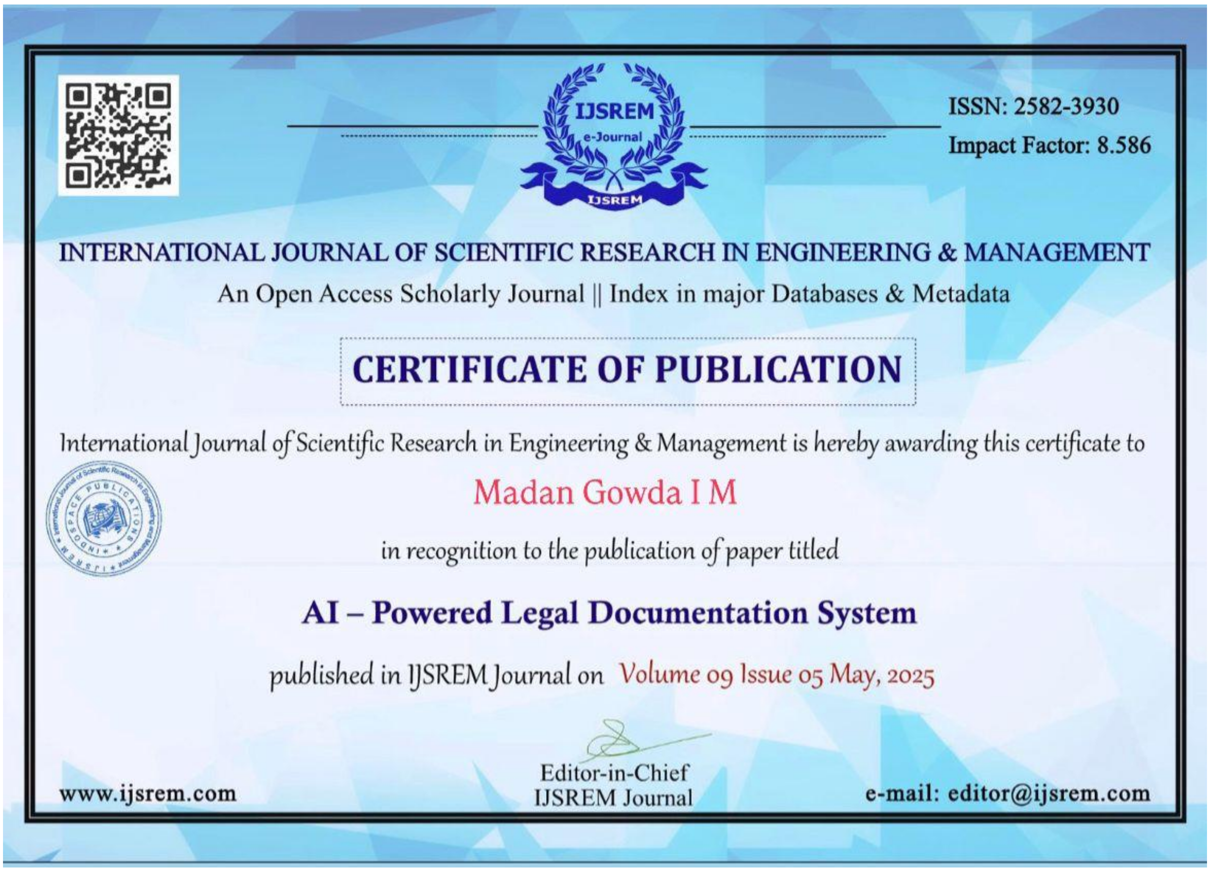
AI – Powered Legal Documentation System

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com



ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Madan Gowda I M

in recognition to the publication of paper titled

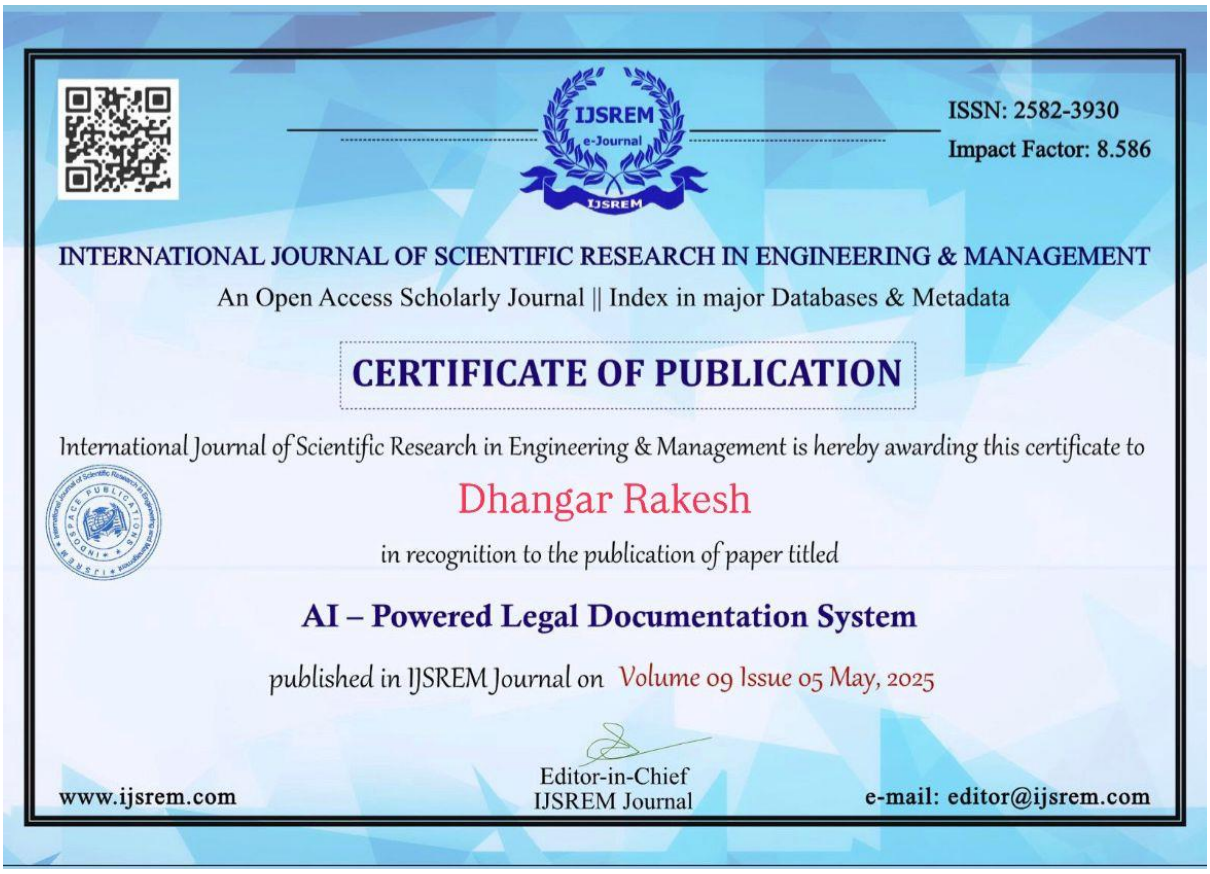
AI – Powered Legal Documentation System

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com



ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Dhangar Rakesh

in recognition to the publication of paper titled

AI – Powered Legal Documentation System

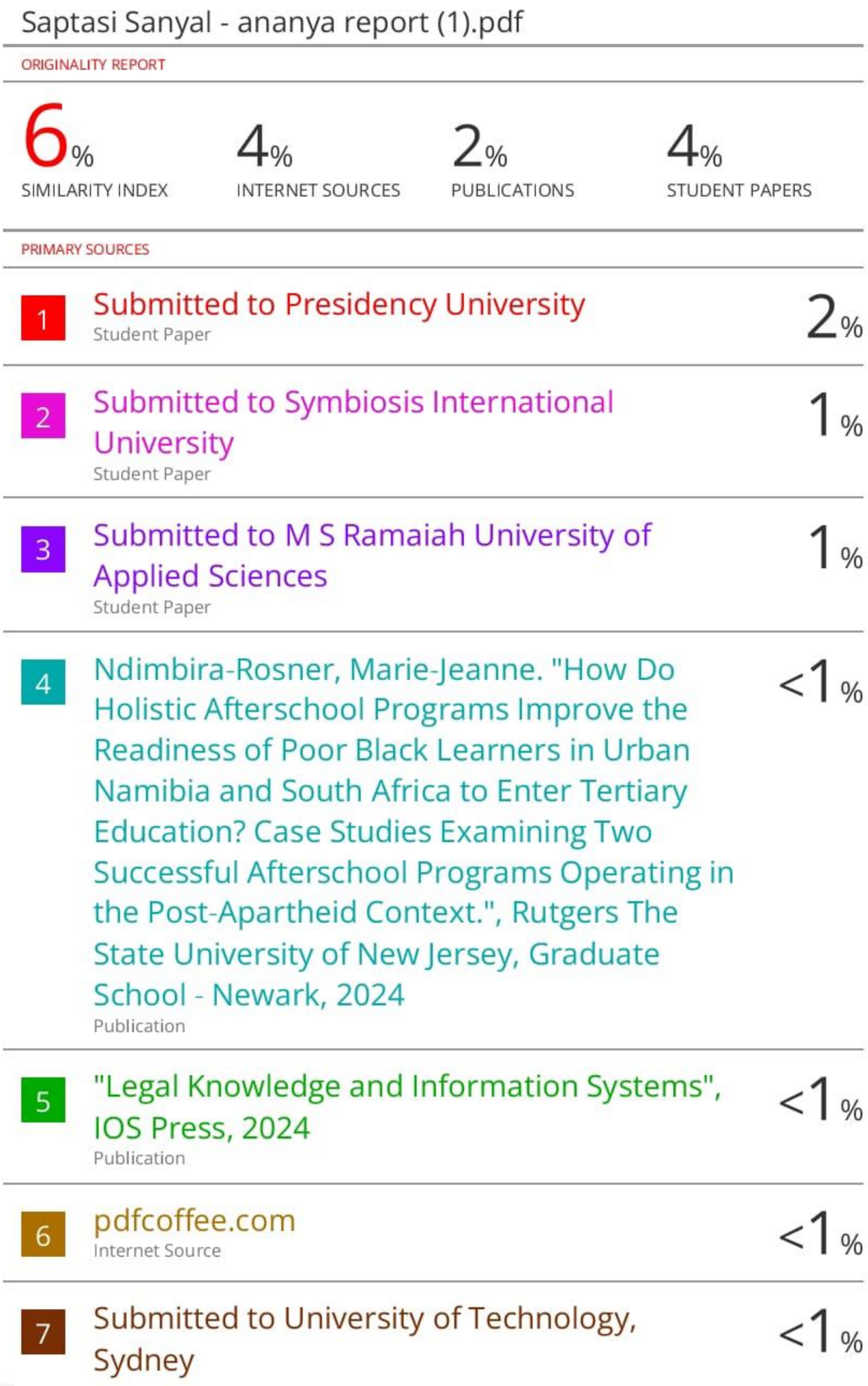
published in IJSREM Journal on Volume 09 Issue 05 May, 2025

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com

PLAGIARISM CHECK RESULTS



Saptasi Sanyal - ananya report (1).pdf

ORIGINALITY REPORT

6%	4%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	2%
2	Submitted to Symbiosis International University Student Paper	1%
3	Submitted to M S Ramaiah University of Applied Sciences Student Paper	1%
4	Ndimbira-Rosner, Marie-Jeanne. "How Do Holistic Afterschool Programs Improve the Readiness of Poor Black Learners in Urban Namibia and South Africa to Enter Tertiary Education? Case Studies Examining Two Successful Afterschool Programs Operating in the Post-Apartheid Context.", Rutgers The State University of New Jersey, Graduate School - Newark, 2024 Publication	<1%
5	"Legal Knowledge and Information Systems", IOS Press, 2024 Publication	<1%
6	pdfcoffee.com Internet Source	<1%
7	Submitted to University of Technology, Sydney	<1%

Mapping the project with the Sustainable Development Goals (SDGs)

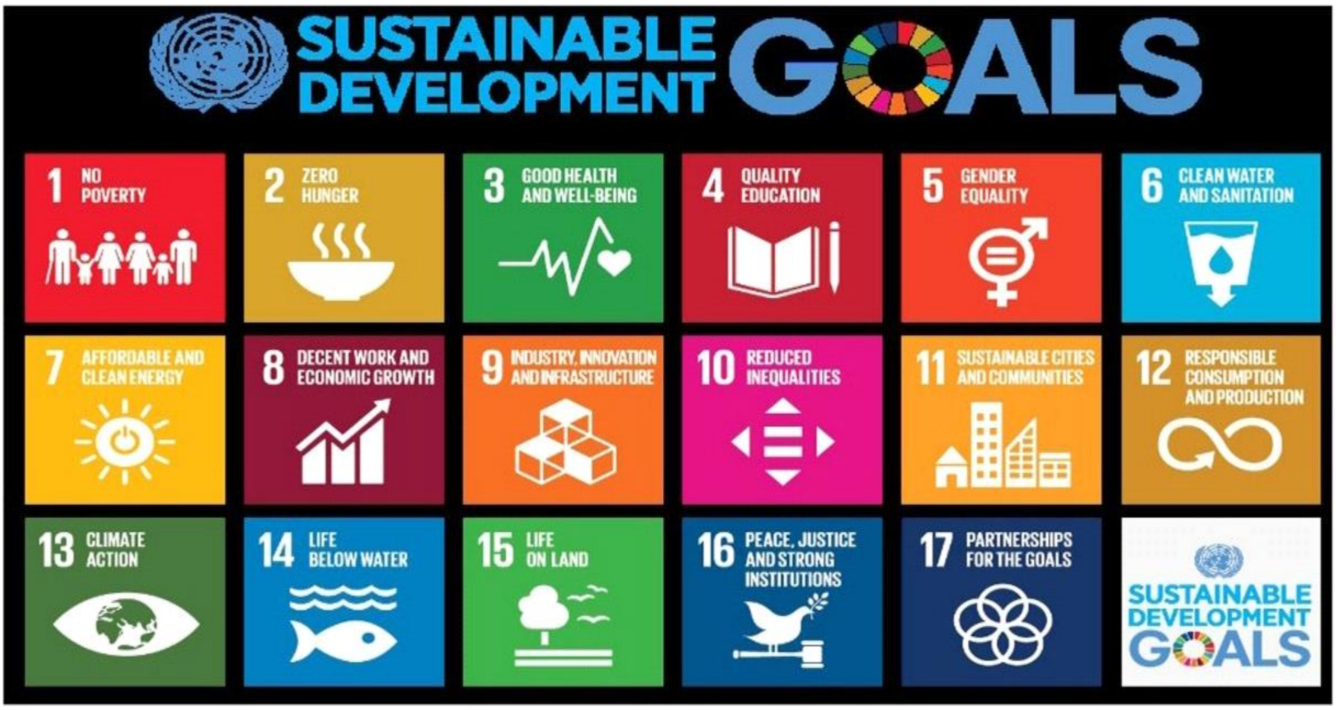

SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

Fig 13.1 Mapping to SDGs

- **SDG 8: Decent Work and Economic Growth:**
  - Automates repetitive tasks, boosting employee productivity and customer service efficiency.
  - Supports businesses in scaling operations sustainably.
- **SDG 9: Industry, Innovation, and Infrastructure:**
  - Leverages AI technology to improve customer support infrastructure and foster innovation.
- **SDG 10: Reduced Inequalities:**
  - Provides equitable support across languages, regions, and socio-economic groups, ensuring inclusivity.
- **SDG 12: Responsible Consumption and Production:**
  - Educates customers about sustainable practices, reducing unnecessary returns and waste.
- **SDG 16: Peace, Justice, and Strong Institutions:**
  - Improves transparency and accountability by delivering consistent, reliable, and fair customer service.