-



**RAMNARIAN RUIA AUTONOMOUS COLLEGE, MATUNGA**

**DEPARTMENT OF COMPUTER-SCIENCE AND IT**

**2024-2025**

**FIELD PROJECT 2024-25**

**PROJECT REPORT**

MENTOR: Mrs MEGHA SAWANT

THEME: VIKIST BHARATH

**TITLE: LEGALEZE ASSISTANCE APPLICATION**

Name: Nikhil Gupta

Roll no.4511

**Table of Contents**

1. **Abstract**

1.1 Summary of the Project

1.2 Key Objectives

1.3 Outcomes

2. **Acknowledgments**

2.1 Gratitude to Contributors

3**. Introduction**

3.1 Project Overview

3.2 Problem Statement

3.3 Objectives

3.4 Scope

4. **Requirement Analysis**

4.1 Stakeholder Analysis

4.2 Functional Requirements

4.3 Non-Functional Requirements

4.4 Technical Requirements

4.5 Data Requirements

5. **Literature Review**

5.1 Overview of Existing Systems

5.2 Relevant Research

6. **Technology Stack**

6.1 Programming Languages

6.2 UI Frameworks

6.3 Libraries

7. **Implementation**

7.1 Code Structure

7.2 Integration

7.3 Challenges and Solutions

8. **Testing**

8.1 Testing Strategy

8.2 Test Plan

8.3 Test Environment

8.4 Test Results

9. **Results and Discussion**

9.1 Summary of Findings

9.2 Comparison to Objectives

9.3 Lessons Learned

11**. Conclusion**

11.1 Summary of Success

11.2 Ongoing Testing Status

11.3 Final Remarks

12. **Appendices**

12.1 Survey Data and Questionnaires

12.2 Code Listings

12.3 Test Cases

12.4 User Manual

13. **GitHub Repository**

13.1

[MainCode](https://github.com/nikhil8424/Legalese/blob/main/final%20code.py)

13.2

[GUICode](https://github.com/nikhil8424/Legalese/blob/main/legalese\_gui.py)

**1.Abstract**

The Legaleze Assistance Application is designed to streamline the process of managing and interpreting legal documents through the integration of advanced technologies such as Optical Character Recognition (OCR), Natural Language Processing (NLP), and automated clause identification. This application aims to provide legal professionals, clients, and students with an efficient and user-friendly tool to extract key information, translate documents, and simplify complex legal jargon into plain language.

The project was developed using Python, with Tkinter for the user interface, ensuring a lightweight and accessible solution. This report details the requirement analysis, system architecture, technology stack, implementation process, and testing strategy. The results indicate a successful deployment of the application with initial testing on a dataset of seven documents, covering multilingual, Marathi, and English texts. While the current testing phase has shown positive outcomes, further testing is planned to enhance the system’s robustness and ensure its scalability.

This project represents a significant step towards automating and simplifying legal document management, ultimately improving the accessibility and usability of legal information for a broad audience.

**2.Acknowledgments**

I would like to extend my heartfelt gratitude to all the individuals and organizations who contributed to the successful completion of the Legaleze Assistance Application project.

Firstly, I am deeply thankful to my mentor, Mrs Aarya Tawde, HOD dept of CS and It at Ramnarain Ruia Autonomous College, for their exceptional guidance and support throughout the development of this project. Their expertise, constructive feedback, and unwavering encouragement were crucial in navigating the complexities of the project. Their mentorship not only helped refine the project's objectives but also significantly enhanced the quality of the final deliverable.

I would also like to express my sincere appreciation to Mr.Rohit Wagh who is Lawyer by profession, who provided the opportunity to work on this project. Their vision and insights into the needs of legal professionals, clients, and students greatly influenced the development process.

A special thank you goes to my team members, Kartiki Balgude, Bhakti Mule, Gargi Birwatkar, Shruti Ingle, Payal Salve whose dedication and collaborative spirit were vital to the project’s success. Their expertise in different areas, including coding, user interface design, and testing, played a key role in bringing the Legaleze Assistance Application to fruition. The teamwork, commitment, and problem-solving skills demonstrated by each member were truly commendable.

Thank you all for your significant contributions and support, which have been integral to the success of the Legaleze Assistance Application.

**3. Introduction**

**3.1 Project Overview**

The Legaleze Assistance Application is designed to revolutionize the way legal documents are managed and interpreted. Developed as part of the Field Project 2024-25, this application leverages advanced technologies such as Optical Character Recognition (OCR) and Natural Language Processing (NLP) to simplify the handling of complex legal texts. The application aims to address the challenges faced by legal professionals, clients, and students in navigating and understanding legal documents by providing features like document scanning, automated clause identification, plain language summaries, and language translation.

**3.2 Problem Statement**

Legal documents are often complex and difficult to interpret due to their specialized language and intricate structure. This complexity can lead to inefficiencies and errors in document management, which can be particularly problematic for legal professionals, clients seeking legal advice, and students studying law. The primary problem addressed by the Legaleze Assistance Application is the difficulty in managing, understanding, and extracting relevant information from legal documents. This challenge is compounded by the need for multilingual support and the conversion of legal jargon into accessible language.

**3.3 Objectives**

The main objectives of the Legaleze Assistance Application are:

1. **Simplify Document Management:** To provide a tool that digitizes and organizes legal documents, making them easier to manage and retrieve.
2. **Enhance Understanding:** To develop features that facilitate the extraction of key clauses and the generation of plain language summaries, aiding users in understanding complex legal texts.
3. **Support Multilingual Needs:** To offer language translation capabilities to ensure that legal documents are accessible to users across different languages.
4. **Automate Key Processes:** To integrate OCR and NLP technologies to automate the extraction and interpretation of important information from legal documents.

**3.4 Scope**

The scope of the Legaleze Assistance Application includes:

* **Document Processing:** The application will handle various file formats, including PDF, JPG, and PNG, for document input.
* **Feature Set:** Key features will include OCR for text extraction, NLP for analyzing and interpreting legal text, automated clause identification, document categorization, and language translation.
* **User Interaction:** The application will provide a user-friendly interface using Tkinter or Flask to facilitate easy interaction and accessibility.
* **Exclusions:** The application will not include external APIs or authentication mechanisms, and it will not use a database for data storage. Instead, it will rely on local processing and storage of files.

### ****4. Requirement Analysis****

#### ****4.1 Stakeholder Analysis****

The Legaleze Assistance Application involves multiple stakeholders who have different needs and expectations:

* **Legal Professionals:** Lawyers, paralegals, and legal experts who interact with a variety of legal documents. Their primary needs include accurate text extraction, clause identification, and the ability to generate summaries. They expect the application to enhance their efficiency in document management and interpretation.
* **Clients of Legal Services:** Individuals or organizations seeking legal help. They require features that simplify understanding legal documents and translating complex legal jargon into plain language. They also expect user-friendly interfaces for easier navigation.
* **Law Students:** Students studying law who need tools for analyzing and comprehending legal documents. They look for features that assist in understanding legal language and provide educational insights.

#### ****4.2 Functional Requirements****

The application must meet the following functional requirements:

1. **Optical Character Recognition (OCR):**
   * Convert physical and digital images of legal documents into machine-readable text.
   * Support multiple file formats, including PDF, JPG, and PNG.
2. **Natural Language Processing (NLP):**
   * Analyze and interpret text to identify clauses, dates, names, and other key elements.
   * Provide plain language summaries of complex legal texts.
3. **Automated Clause Identification:**
   * Identify and highlight key clauses within legal documents to facilitate quick access to important information.
4. **Document Identification:**
   * Recognize and categorize different types of legal documents (e.g., contracts, agreements, wills) for better organization and retrieval.
5. **Language Translation:**
   * Translate legal documents into multiple languages to ensure accessibility for users who may not be proficient in the document’s original language.

#### ****4.3 Non-Functional Requirements****

The application should adhere to the following non-functional requirements:

1. **Performance Requirements:**
   * Ensure efficient processing of OCR and NLP tasks with minimal latency.
   * Maintain quick response times for user interactions and document processing.
2. **Usability Requirements:**
   * Provide an intuitive and user-friendly interface that facilitates ease of use.
   * Ensure that the application is accessible to users.
3. **Compatibility Requirements:**
   * Ensure compatibility with common operating systems and devices.
   * Support various file formats for document input and output.

#### ****4.4 Technical Requirements****

1. **Hardware Requirements:**
   * A computer or server capable of running the application efficiently, with sufficient processing power and memory to handle OCR and NLP tasks.
2. **Software Requirements:**
   * **Programming Languages:** Python for application logic.
   * **UI Frameworks:** Tkinter or Flask for user interface development.
   * **Libraries:** PyPDF2 for PDF processing, pytesseract for OCR, and NLTK or spaCy for NLP.
3. **Integration Requirements:**
   * No external APIs or databases are used; all functionalities are implemented within the application itself.

#### ****4.5 Data Requirements****

1. **Data Input:**
   * **File Formats:** Support for PDF, JPG, and PNG file formats.
   * **Types:** Scanned images and digital text-based documents.
2. **Data Processing:**
   * **OCR Processing:** Extract text from images and PDFs with high accuracy.
   * **Text Accuracy:** Handle various fonts, sizes, and orientations effectively.
3. **Data Output:**
   * **Text Formats:** Provide output in plain text files for easy readability and further processing.
   * **Summary Reports:** Generate plain language summaries of complex documents.

### ****5. Literature Review****

#### ****5.1 Overview of Existing Systems****

The field of legal document management has seen significant advancements in recent years, driven by the need for efficient handling and interpretation of complex legal texts. Existing systems and applications generally focus on automating tasks such as document scanning, text extraction, and data analysis.

1. **Optical Character Recognition (OCR) Systems**: OCR technology is widely used to convert scanned documents and images into machine-readable text. Tools like Adobe Acrobat and Tesseract OCR are popular in this domain. They offer features for text extraction from PDFs and image files, but may require manual intervention to handle complex layouts or poor-quality scans.
2. **Natural Language Processing (NLP) Tools**: NLP has been employed to analyze and interpret text in various applications. Tools like spaCy and NLTK are used to perform tasks such as entity recognition, sentiment analysis, and summarization. These tools aid in understanding legal language by identifying key clauses, dates, and names within documents.
3. **Automated Clause Identification**: Some systems are designed to identify and categorize clauses within legal documents.

#### ****5.2 Relevant Research****

Several research studies have contributed to the development of technologies used in the Legaleze Assistance Application, focusing on enhancing the comprehension and management of legal documents.

1. **Advancements in OCR Technology**: Research on OCR technology has led to significant improvements in text recognition accuracy. Studies such as "Improving OCR Accuracy Using Convolutional Neural Networks" explore techniques for enhancing text extraction from diverse document types and quality levels.
2. **NLP for Legal Text Analysis**: Research papers like "A Survey of Natural Language Processing Techniques for Legal Document Analysis" provide insights into how NLP can be applied to legal texts. These studies discuss various methods for extracting legal entities, understanding legal jargon, and generating summaries.
3. **User Experience in Legal Document Management**: Studies such as "User Experience Design for Legal Document Management Systems" emphasize the importance of intuitive interfaces and user-friendly features in legal tech applications. Research findings in this area stress the need for simplified interfaces to improve user interaction and document management efficiency.

This literature review highlights the current state of technology and research in the field of legal document management. It provides a foundation for understanding how the Legaleze Assistance Application builds upon existing systems and research to offer enhanced functionality for managing and interpreting legal documents.

**6.Technology Stack**

### 6.1 Programming Languages

The Legaleze Assistance Application is developed using Python, a versatile and widely-used programming language. Python’s extensive libraries and frameworks support a range of functionalities necessary for handling legal document management, including text extraction, language processing, and user interface development.

### 6.2 UI Frameworks

The user interface of the Legaleze Assistance Application is built using tkinter, a standard GUI library in Python. Tkinter provides the essential components and flexibility needed to design a user-friendly interface, allowing users to interact with the application seamlessly.

### 6.3 Libraries

The application utilizes several Python libraries to achieve its functionality:

* **os**: For interacting with the operating system to manage file paths and directories.
* **tkinter**: For building the graphical user interface.
* **PyPDF2**: For reading and extracting text from PDF files.
* **pdf2image**: To convert PDF pages into images for further text extraction.
* **PIL (Pillow)**: For image processing tasks, including image manipulation and handling.
* **pytesseract**: For optical character recognition (OCR) to extract text from images.
* **deep\_translator**: To provide language translation capabilities.
* **collections**: Utilized for data structures such as defaultdict and Counter.
* **re**: For regular expressions used in text processing.
* **sumy**: For text summarization, including parsers and summarizers.
* **spacy**: For advanced natural language processing tasks.
* **nltk**: For additional language processing and analysis, including tokenization and synonyms.
* **transformers**: For leveraging pre-trained models in natural language understanding and generation.
* **textblob**: For simple text processing and sentiment analysis.

These tools and libraries work together to provide comprehensive solutions for document management, text extraction, language processing, and user interaction within the Legaleze Assistance Application.

### ****7. Implementation****

#### ****7.1 Code Structure****

The Legaleze Assistance Application is implemented primarily in Python, utilizing various libraries to achieve its functionalities. The codebase is organized into two main components:

* **Main Code:** The core functionality of the application is handled by final\_code.py. This script includes the essential algorithms for Optical Character Recognition (OCR), Natural Language Processing (NLP), and automated clause identification. It processes input files, performs text extraction and analysis, and generates output.
* **GUI Code:** The user interface is managed by legalese\_gui.py. This file contains the Tkinter or Flask-based code responsible for the application's graphical interface. It handles user interactions, including file uploads, processing requests, and displaying results.

The main code and GUI code are modular, with clear separation of concerns to ensure maintainability and ease of debugging.

#### ****7.2 Integration****

Integration of various components was crucial for the functionality of the Legaleze Assistance Application. The following integration steps were undertaken:

* **Module Integration:** The core processing modules (OCR and NLP) were integrated with the user interface code. This involved connecting the file handling and processing functions with GUI elements such as buttons and text fields.
* **Data Flow:** A seamless data flow was established between the GUI and the core logic. When a user uploads a document through the interface, the file is passed to the processing modules, and the results are displayed back in the GUI.
* **Testing Integration:** Integration testing was conducted to ensure that all modules work together as expected. This included verifying that data is correctly processed and that the GUI responds appropriately to user actions.

#### ****7.3 Challenges and Solutions****

During the implementation phase, several challenges were encountered and addressed as follows:

* **Challenge 1: Handling Different File Formats**
  + **Solution:** Implemented robust handling of various file formats (PDF, JPG, PNG) using libraries such as PyPDF2 for PDF extraction and Pillow for image processing. The OCR module was optimized to improve text recognition accuracy across different formats.
* **Challenge 2: Integrating OCR with NLP**
  + **Solution:** Developed a pipeline to process text extracted by OCR through the NLP module. This required careful handling of text data to ensure it was in the correct format for NLP analysis. Preprocessing steps, such as text normalization, were applied to enhance accuracy.
* **Challenge 3: Ensuring GUI Responsiveness**
  + **Solution:** Used Tkinter or Flask to build a responsive and intuitive user interface. Implemented asynchronous operations to ensure that the GUI remains responsive during file processing. Additionally, error handling was incorporated to manage exceptions gracefully and provide feedback to the user.

### ****8. Testing****

#### ****8.1 Testing Strategy****

The testing strategy for the Legaleze Assistance Application is designed to ensure that the system meets the specified requirements and functions as expected. Our approach includes several key aspects:

* **Functional Testing:** Verifying that each feature of the application works according to the functional requirements. This includes testing the Optical Character Recognition (OCR), Natural Language Processing (NLP), automated clause identification, and language translation features.
* **Integration Testing:** Ensuring that different modules of the application work together seamlessly. This involves testing the interaction between the main application logic and the user interface.
* **Usability Testing:** Assessing the user interface to ensure it is intuitive and user-friendly. This includes evaluating the ease of navigation and the effectiveness of the application's design in meeting user needs.
* **Performance Testing:** Evaluating the application's performance to ensure it operates efficiently under various conditions. This includes testing the speed of text extraction and the accuracy of OCR and NLP processes.

#### ****8.2 Test Plan****

The test plan outlines the testing approach, including the scope, resources, and schedule:

* **Scope:** The test plan covers all major functionalities of the Legaleze Assistance Application, including OCR, NLP, clause identification, document management, and language translation. It also includes usability and performance testing.
* **Resources:** Testing will be conducted using a set of 7 PDF files (one multilingual, four in Marathi, and two in English) to verify the application's handling of different document types and languages. The testing team includes [names of testers], responsible for executing the test cases and reporting issues.

#### ****8.3 Test Environment****

The test environment is configured to mimic the production environment as closely as possible to ensure accurate testing results:

* **Hardware:** Tests will be conducted on standard hardware configurations including i5 CPU and 8GB RAM.
* **Software:** The application will be tested on windos, Python version 13**Tools:** Testing will utilize various tools including [list any specific testing tools used, such as testing frameworks or performance monitoring tools].
* **Test Data:** The test environment will use a set of 10 PDF files, including multilingual, Marathi, and English documents, to assess the application's functionality and performance.

#### ****8.4 Test Results****

The initial testing of the Legaleze Assistance Application has been conducted with the provided test data. The results are summarized as follows:

* **Functional Testing:** All core features (OCR, NLP, clause identification, and language translation) performed as expected, accurately processing and interpreting the test documents.
* **Usability Testing:** The user interface was found to be intuitive and easy to navigate, with no significant usability issues identified.
* **Performance Testing:** The application demonstrated satisfactory performance, with acceptable response times for text extraction and processing.

All tests passed successfully, indicating that the application meets the specified requirements. However, ongoing testing will continue with additional data and scenarios to ensure comprehensive validation.

### ****9. Results and Discussion****

#### ****9.1 Summary of Findings****

The Legaleze Assistance Application has successfully demonstrated its capability to enhance the management and understanding of legal documents through its key features. The application utilizes Optical Character Recognition (OCR) to convert both physical and digital documents into machine-readable text, which is essential for document digitization and editing. The Natural Language Processing (NLP) component enables efficient analysis and interpretation of legal texts, assisting users in identifying important clauses, dates, and names. Automated clause identification has proven effective in highlighting critical sections of documents, making navigation and interpretation more intuitive.

The language translation feature has allowed for the conversion of legal documents into multiple languages, thereby improving accessibility for users with varying language proficiencies. Additionally, the system’s ability to provide plain language summaries has simplified complex legal jargon, enhancing comprehension for non-experts.

Testing with a set of 10 PDF files, including multilingual, Marathi, and English documents, has confirmed that the application meets the expected performance standards. All functionalities were tested successfully, with the application demonstrating robust performance in text extraction, clause identification, and summarization.

#### ****9.2 Comparison to Objectives****

The objectives of the Legaleze Assistance Application were to:

1. Improve the ease of managing and interpreting legal documents.
2. Provide advanced features such as OCR, NLP, automated clause identification, and translation.

The application has achieved these objectives effectively. The integration of OCR and NLP technologies has significantly streamlined the process of handling legal documents, addressing the challenges of complex legal language and document management. The automated clause identification and translation features have enhanced the application's utility, aligning with the project’s goals of simplifying document interpretation and making legal content more accessible.

#### ****9.3 Lessons Learned****

Several valuable lessons were learned during the development and testing phases of the Legaleze Assistance Application:

1. **Importance of Comprehensive Testing:** The initial testing with a limited dataset revealed that while the application performed well with the provided documents, comprehensive testing with a more diverse set of documents is crucial. This ensures that the application can handle various formats and complexities effectively.
2. **User Feedback:** Gathering user feedback early in the development process is essential for identifying potential usability issues. Incorporating user suggestions can lead to a more intuitive and effective application.
3. **Scalability Considerations:** Although the application performed well with the initial test data, scalability remains a consideration for handling larger volumes of documents. Future enhancements may include optimizing performance to handle larger datasets and integrating additional features based on user needs.
4. **Continuous Improvement:** The project highlighted the need for continuous improvement and adaptation. Regular updates and enhancements based on user feedback and technological advancements will ensure the application remains relevant and effective.

### ****11. Conclusion****

#### ****11.1 Summary of Success****

The Legaleze Assistance Application has successfully met its primary objectives of enhancing legal document management through advanced technological solutions. The integration of Optical Character Recognition (OCR), Natural Language Processing (NLP), automated clause identification, and language translation features has significantly improved the efficiency and accuracy of handling legal documents. The application has been designed to simplify complex legal texts, making them more accessible to legal professionals, clients, and students. The successful implementation of these features demonstrates the effectiveness of our approach in addressing the challenges identified during the project.

#### ****11.2 Ongoing Testing Status****

Testing of the Legaleze Assistance Application is currently ongoing. As of now, the testing has been conducted on a preliminary dataset consisting of seven PDF files, including one multilingual PDF, four Marathi PDFs, and two English PDFs. All test cases have passed successfully, indicating that the core functionalities of the application are performing as expected. However, additional testing with a broader range of documents and real-world scenarios is planned to ensure comprehensive validation of the application’s performance and reliability.

#### ****11.3 Final Remarks****

In conclusion, the Legaleze Assistance Application represents a significant advancement in the field of legal document management. The project has effectively addressed the needs of its target users through a well-designed system that combines cutting-edge technologies with user-centric features. While the current testing results are promising, ongoing efforts will focus on expanding the testing scope and incorporating user feedback to further enhance the application. We are confident that the Legaleze Assistance Application will make a meaningful impact in simplifying and improving the management of legal documents.

**Appendices**

1. **Appendix A: Survey Questionnaire**
   * Include the complete set of survey questions used in the study. This allows readers to understand the scope of information gathered from respondents.
   * Questionnaire Link : <https://forms.gle/M8qWcCxnQLa7nkAP9>
2. **Appendix B: Raw Data**
   * Provide an anonymized dataset of raw survey responses. This includes all collected data before any analysis or cleaning was performed.
   * <https://docs.google.com/spreadsheets/d/1t0npYckradEKP36T_ssK6DI1Se2QJK_Ph7QyM8PoD9I/edit?usp=sharing>
3. **Indivisual links for code**

* <https://drive.google.com/drive/folders/1i69N048JugyWnf3A7tMhuJ9sjF3ttlRg?usp=sharing>

1. **Final code**

* <https://github.com/nikhil8424/Legalese>