Synopsis: Smart Legal Assistant

1. Introduction

Contracts and legal documents are often lengthy, complex, and filled with technical terms. Manually reviewing them for key clauses and risks is time-consuming and error-prone. This project proposes an AI-driven Smart Legal Assistant that automatically analyzes contracts, extracts important clauses, generates summaries, and flags risky provisions to assist lawyers, businesses, and individuals.

2. Objectives

- To develop a clause extraction module using heuristic and NLP-based approaches.
- To implement an abstractive summarization system using transformer models (e.g., BART, PEGASUS).
- To build a risk identification engine using Natural Language Inference (NLI) models such as RoBERTa-MNLI.
- To integrate all modules into a single pipeline for contract analysis.
- To deploy the system as a FastAPI web service with a user-friendly frontend.

3. Scope

The system will accept legal documents in PDF and text formats, extract key clauses, summarize the contract in plain language, and flag potentially risky clauses (e.g., “termination without notice”, “no liability”). It will provide structured outputs (JSON + Web interface). The system will not replace lawyers but serve as an assistant for faster review.

4. Methodology

- Data Collection – Use public contract datasets (e.g., CUAD dataset) for training/testing.
- Preprocessing – PDF text extraction, cleaning, and segmentation.
- Clause Extraction – Apply regex + NLP heuristics for identifying clause boundaries.
- Summarization – Fine-tune/implement transformer summarization models (BART/PEGASUS).
- Risk Detection – Use NLI pipelines with predefined hypotheses to detect risky clauses.
- Integration – Develop a FastAPI service exposing /analyze API endpoint.
- Deployment – Dockerize the app and build a simple HTML/JS frontend for usability.

5. Expected Outcomes

- Automated clause extraction and summarization from legal contracts.
- Risk identification of potentially harmful provisions.
- JSON output suitable for integration into external systems.
- A deployed web application accessible to non-technical users.

6. Tools and Technologies

- Programming Language: Python 3.10+
- Frameworks: FastAPI, Uvicorn
- NLP Models: Hugging Face Transformers (BART, PEGASUS, RoBERTa-MNLI)
- Libraries: pdfminer.six, PyPDF2, NumPy, Pydantic
- Deployment: Docker, HTML/JS frontend

7. System Requirements

- RAM: Minimum 8 GB (16 GB recommended)
- Storage: 10+ GB
- OS: Windows 10 / Linux (Ubuntu 20.04+)
- Processor: Intel Core i5 or higher
- GPU: Recommended for faster inference

8. Theoretical Aspects

- Sequence-to-Sequence Learning – core framework for summarization.
- Natural Language Inference (NLI) – used for clause risk detection.
- Attention Mechanism – enables models to focus on important parts of text.
- Information Extraction – identifies clauses and entities within contracts.

9. FAQs

- Q1. What is the main objective? A1. To automate contract review by summarizing and highlighting risky clauses.
- Q2. Which models are used? A2. Summarization: BART/PEGASUS, Risk detection: RoBERTa-MNLI.
- Q3. What formats are supported? A3. PDF and text documents.
- Q4. Who are the users? A4. Lawyers, businesses, and individuals reviewing legal agreements.
- Q5. What is the real-world impact? A5. Faster contract review, reduced legal risk, and improved decision-making.