PROJECT REPORT

Car Lease / Loan Contract Review AI Assistant

LLM-Based Contract Analysis using FastAPI

(Backend Implementation – Phase 1)

Internship Program: Infosys Virtual Internship 6.0

Domain: AI / Backend Development

Project Type: AI-Powered Contract Analysis System

Abstract

Car lease and loan agreements are legally intensive and financially sensitive documents that often contain complex terminology, hidden charges, and critical clauses that are difficult for consumers to fully understand. Misinterpretation of such contracts can lead to unfavorable financial commitments, unexpected penalties, and long-term monetary losses. This project presents an AI-driven backend system designed to automate the analysis of car lease and loan contracts by extracting key contractual terms and evaluating their fairness in a structured and explainable manner.

The system accepts contract documents in PDF format and processes them using Optical Character Recognition (OCR) techniques to convert both scanned and digitally generated contracts into machine-readable text. Large Language Models (LLMs) are then employed to intelligently extract essential Service Level Agreement (SLA) and financial parameters such as interest rate, loan or lease tenure, monthly payment, penalties, foreclosure charges, grace period, and other contractual conditions. The extracted information is returned in a strictly structured JSON format to ensure consistency, accuracy, and easy downstream processing.

In addition to data extraction, the system introduces a Contract Fairness Scoring mechanism that quantitatively evaluates the quality of a contract. Starting from a baseline fairness score, the system applies predefined business rules to reduce the score when risky or unfavorable clauses are detected. Simultaneously, the application identifies and reports red flags—specific contract terms that may negatively impact the consumer—along with clear reasons for their classification. This ensures transparency and explainability in the decision-making process.

The backend application is implemented using FastAPI, enabling high performance, modular design, and seamless integration with future mobile or web-based frontends. The overall solution provides a scalable foundation for advanced features such as negotiation assistance, vehicle price benchmarking, and contract comparison.

TABLE OF CONTENTS

PROBLEM STATEMENT

Car lease and loan agreements are legally binding documents that govern the financial and contractual obligations between consumers and financial institutions or automobile dealers. These documents are typically lengthy, text-heavy, and written using complex legal and financial terminology, making them difficult for average consumers to read and understand.

Most car lease and loan contracts contain hidden fees, penalty clauses, restrictive conditions, and ambiguous terms related to interest rates, foreclosure charges, late payment penalties, grace periods, and early termination rules. These clauses are often buried deep within the document and are not presented in a transparent or consumer-friendly format. As a result, users may unknowingly agree to unfavorable terms that can lead to significant financial loss over the course of the contract.

Additionally, consumers face challenges when comparing multiple lease or loan offers from different dealers or financial institutions. Since each contract follows a different structure and wording, it becomes extremely difficult to evaluate which offer is more fair, transparent, or cost-effective without expert legal or financial assistance.

At present, consumers lack effective tools to:

- Quickly identify unfair or biased contract terms
- Detect high-risk clauses such as excessive penalties or foreclosure charges
- Objectively quantify the overall quality or fairness of a contract
- Understand the long-term financial implications before signing

Manual contract review is time-consuming, error-prone, and inaccessible to most users, creating a strong need for an automated and intelligent solution.

OBJECTIVES

The primary objective of this project is to design and implement an AI-driven backend system that simplifies the analysis of car lease and loan agreements by automatically extracting, evaluating, and presenting key contractual information in a transparent and user-friendly manner.

1. Automated SLA Extraction from Contract PDFs

To automatically extract critical Service Level Agreement (SLA) and financial terms from car lease and loan contract documents provided in PDF format. This includes identifying key clauses such as interest rates, lease duration, monthly payments, penalties, foreclosure conditions, grace periods, and other contractual obligations without requiring manual review.

2. Conversion of Unstructured Text into Structured Data

To transform complex, unstructured contract text—often written in legal and financial language—into well-defined, structured JSON outputs. This structured representation allows downstream systems to process, analyze, compare, and visualize contract terms programmatically, enabling automation and scalability.

3. Contract Fairness Score Computation

To develop a Contract Fairness Scoring mechanism that quantitatively evaluates the quality of a contract. Starting from a base score, the system applies predefined business rules to assess how fair or risky the contract is, based on factors such as high penalties, absence of grace periods, unclear interest terms, and excessive foreclosure charges.

4. Red Flag Detection and Explanation

To identify and clearly explain red flags—contractual clauses that may negatively impact consumers. Each red flag is associated with a human-readable explanation, ensuring that users understand *why* a clause is risky rather than receiving a black-box judgment.

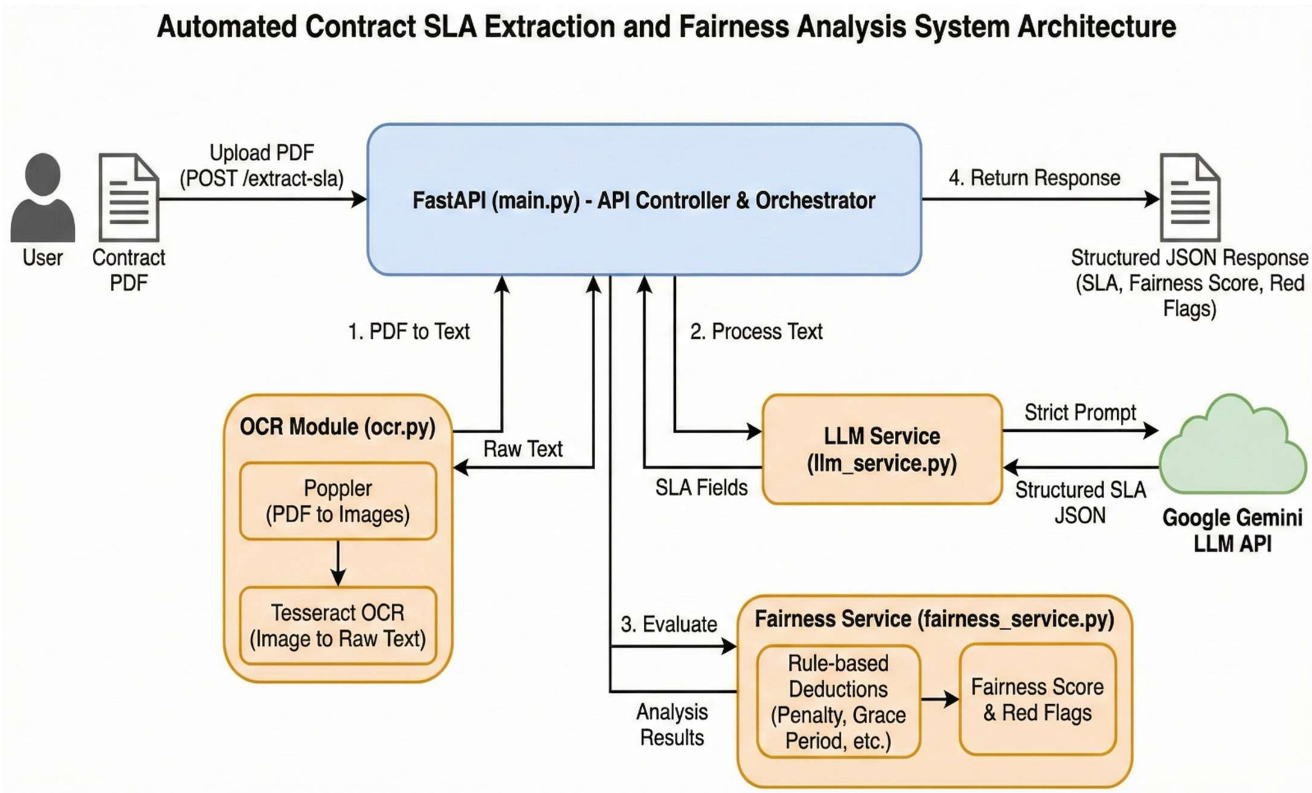
5. Scalable and Modular Backend Implementation

To build a scalable, modular, and maintainable backend using FastAPI. The system is designed with clear service boundaries (OCR service, LLM extraction service, fairness evaluation service), making it easy to extend with additional features such as VIN lookup, price benchmarking, or negotiation assistance in future iterations.

6. Explainability and Transparency

To ensure that all outputs—including extracted fields, fairness scores, and red flags—are explainable and transparent. The system avoids opaque AI decisions by clearly outlining how scores are calculated and which contract clauses contribute to risk, thereby increasing trust and usability for end users.

SYSTEM ARCHITECTURE



The system architecture of the *Car Lease/Loan Contract Review and Fairness Analysis System* is designed as a modular, AI-driven backend pipeline that processes contract documents step by step, ensuring scalability, explainability, and accuracy.

1. User Input Layer

The process begins when a user uploads a car lease or loan contract in PDF format through an API endpoint.
The system is designed to accept scanned documents as well as digitally generated PDFs.

2. FastAPI Backend (Application Layer)

FastAPI acts as the core backend framework and handles:

- File upload via REST API

- Request validation
- Orchestration of OCR, AI extraction, and fairness evaluation
- Structured JSON response generation

FastAPI was chosen due to its high performance, automatic API documentation, and scalability.

3. OCR Processing Layer

Once the PDF is uploaded:

- The file is temporarily stored on the server.
- The pdf2image library converts PDF pages into images.
- Tesseract OCR extracts raw text from each page.

This step converts non-machine-readable documents into plain text that can be analyzed further.

4. LLM-Based SLA Extraction Layer

The extracted text is passed to a Large Language Model (Google Gemini) using a carefully engineered prompt.

This layer:

- Identifies predefined SLA fields such as interest rate, lease term, penalties, EMI, foreclosure clauses, etc.
- Converts unstructured legal text into strictly structured JSON.
- Ensures no explanations or extra text are returned.

This step transforms complex legal documents into machine-usable structured data.

5. Fairness Scoring and Red Flag Detection Layer

Using the extracted SLA data:

- The system initializes a fairness score of 100.
- Predefined business rules evaluate risk factors such as:

- High penalties
- Missing grace period
- Unclear interest terms
- Excessive foreclosure charges

- The score is reduced based on severity.
- Red flags are generated with clear explanations.

This layer ensures explainable AI decision-making.

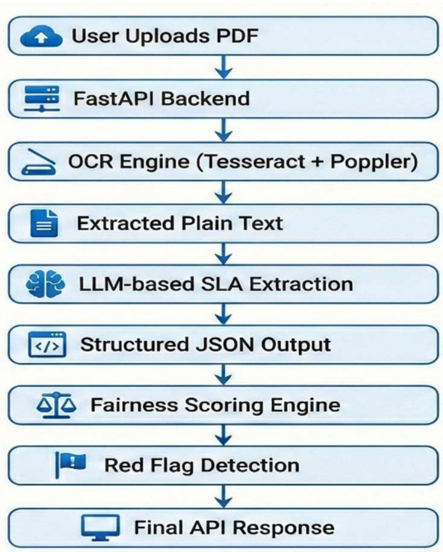
6. Response & Output Layer

Finally, the backend returns a structured JSON response containing:

- Extracted SLA fields
- Contract Fairness Score
- Red flags with reasons
- Overall contract risk category

This output can be directly consumed by:

- A web frontend
- A mobile application
- Reporting or analytics systems



TECHNOLOGIES USED

This project utilizes a combination of modern backend frameworks, AI technologies, and document-processing tools to build a scalable and intelligent contract analysis system. Each technology was selected based on reliability, performance, and real-world industry relevance.

---

1 Python

Python is used as the **core programming language** for the entire backend implementation.
It provides:

- Strong support for AI and machine learning libraries
- Easy integration with OCR and LLM services
- Rapid development and readability
- Extensive community and documentation

Python enables seamless orchestration of OCR processing, LLM-based extraction, fairness analysis, and API development within a single ecosystem.

---

2 FastAPI

FastAPI is used to develop the **RESTful backend API** of the system.

Key advantages:

- High performance due to ASGI-based architecture
- Automatic API documentation using Swagger UI
- Built-in request validation
- Easy handling of file uploads (PDF contracts)

FastAPI acts as the central controller that connects OCR, LLM processing, and fairness evaluation modules into a unified workflow.

---

3 Uvicorn

Uvicorn is used as the **ASGI server** to run the FastAPI application.

Purpose:

- Handles incoming HTTP requests efficiently
- Supports asynchronous processing
- Ensures low-latency API responses

Uvicorn enables the backend to scale efficiently and handle multiple document-processing requests concurrently.

---

4 Tesseract OCR

Tesseract OCR is used for **optical character recognition**.

Role in the system:

- Extracts text from scanned or image-based PDF contracts
- Converts non-machine-readable documents into plain text
- Enables AI-based analysis on real-world scanned contracts

This ensures that the system can process both digital and scanned contract documents.

---

5 Poppler

Poppler is used for **PDF-to-image conversion**, which is a required preprocessing step for OCR.

Purpose:

- Converts each page of a PDF into an image format
- Enables accurate OCR extraction using Tesseract
- Supports multi-page document processing

Poppler bridges the gap between PDF documents and OCR processing.

---

6 Google Gemini Large Language Model (LLM)

Google Gemini is used as the **LLM-based document understanding engine**.

Responsibilities:

- Analyzes extracted contract text
- Identifies key SLA clauses such as interest rate, penalties, lease term, and charges
- Converts unstructured legal text into structured JSON output
- Ensures strict field-level extraction using prompt engineering

This AI layer is the core intelligence of the system, enabling automated contract understanding.

---

7 JSON (JavaScript Object Notation)

JSON is used as the **standard structured data format** throughout the system.

Usage:

- Represents extracted SLA fields
- Stores fairness scores and red flags
- Acts as the response format for APIs
- Enables easy integration with frontend and mobile applications

JSON ensures interoperability, clarity, and consistency across all components.

IMPLEMENTATION DETAILS

The implementation of the Car Lease/Loan Contract Review system follows a modular and layered architecture. Each component is designed to perform a specific responsibility, ensuring scalability, maintainability, and clarity. The backend workflow starts from PDF upload and ends with structured contract insights delivered as JSON.

7.1 main.py (API Layer)

The main.py file serves as the **entry point and API orchestration layer** of the backend system. It is implemented using FastAPI and is responsible for managing the end-to-end request flow.

Key responsibilities:

- Handles incoming HTTP requests from clients
- Accepts car lease/loan contract PDFs through a REST endpoint
- Invokes the OCR module to extract raw text from the uploaded document
- Passes extracted text to the LLM-based SLA extraction service
- Returns a clean, structured JSON response to the client
- Handles exceptions and errors gracefully

Workflow in main.py:

1. The user uploads a contract PDF using the /extract-sla API endpoint.
2. The file is validated and forwarded to the OCR module.
3. Extracted text is sent to the LLM service for structured data extraction.
4. The final SLA data is returned as a JSON response.

7.2 OCR Module (Text Extraction Layer)

The OCR module is responsible for converting unstructured PDF documents into machine-readable text. This module is critical because real-world contracts often come as scanned or image-based PDFs.

Key steps involved:

- The uploaded PDF is temporarily stored on the server
- Each page of the PDF is converted into an image using Poppler
- Tesseract OCR is applied to each image to extract text
- All extracted text is combined into a single string
- Temporary files are deleted after processing to save resources

Capabilities:

- Supports multi-page documents
- Works for both scanned PDFs and digitally generated PDFs
- Ensures reliable text extraction even from low-quality scans

This module forms the **foundation of the AI pipeline**, as accurate OCR output directly impacts the quality of SLA extraction.

7.3 LLM SLA Extraction Service

The LLM SLA Extraction Service is the **intelligence core** of the system. It uses the Google Gemini Large Language Model to analyze the extracted contract text and convert it into structured data.

Key responsibilities:

- Receives raw contract text from the OCR module
- Applies prompt-engineered instructions to the LLM
- Extracts predefined financial and legal clauses
- Returns output strictly in JSON format

Prompt enforcement rules:

- Output must be valid JSON only
- Field names must exactly match predefined SLA fields
- No explanations, sentences, or extra text are allowed
- Missing values must be returned as empty strings

Extracted clauses include:

- Interest rate / APR
- Lease or loan duration
- Monthly payment
- Penalty and late fee clauses
- Foreclosure and prepayment terms
- Grace period and other conditions

By enforcing strict output rules, the system ensures **consistency, accuracy, and machine-readability**, making the extracted data suitable for fairness scoring and red-flag detection.

CONTRACT FAIRNESS SCORE & RED FLAGS

While extracting SLA fields provides transparency, it does not fully convey whether a contract is *good or risky* for the customer. To address this gap, the system introduces a **Contract Fairness Score** along with **Red Flag Detection**. This module evaluates the extracted contract terms and quantifies the overall contract quality in an explainable manner.

1. Fairness Score Logic

The Contract Fairness Score is designed to give users a numerical representation of contract quality, making it easy to compare different lease or loan agreements.

Scoring Approach:

- Every contract starts with an initial fairness score of 100
- The score is gradually reduced based on the presence of unfavorable or risky clauses
- Each deduction is rule-based and explainable

This approach ensures transparency and avoids black-box decision making.

Score Reduction Rules:

Condition	Score Impact
High penalty charges	−15
High foreclosure fees	−15
No grace period	−10
Unclear or variable interest rate	−10
Missing EMI details	−10

Explanation of conditions:

- **High penalty charges** increase financial burden if payments are delayed.
- **High foreclosure fees** discourage early loan closure, limiting flexibility.
- **No grace period** means even minor delays result in penalties.
- **Unclear interest rate** reduces transparency and increases financial risk.
- **Missing EMI details** makes budgeting and planning difficult.

The final fairness score reflects how customer-friendly or restrictive the contract is.

2. Red Flags Detection

Red Flags are specific clauses or conditions that may negatively impact the customer. While the fairness score provides a summary, red flags provide **actionable insights**.

Purpose of Red Flags:

- Highlight risky clauses clearly
- Help users understand *why* a contract is problematic
- Improve explainability and trust in AI decisions

Examples of Red Flags:

- **High Penalty Charges** → Indicates potential heavy financial burden during delays
- **No Grace Period** → Immediate penalties even for short payment delays
- **High Foreclosure Charges** → Makes early exit costly
- **Unclear Interest Rate** → Suggests lack of transparency or hidden costs

Each red flag includes:

- Clause name
- Reason for being flagged
- Explanation of its financial or legal impact on the user

This design ensures the system does not just *judge* the contract but also *explains* the judgment.

3. Risk Categorization

Based on the final fairness score, the contract is categorized into a **risk level** to simplify decision-making for users.

Score Range	Risk Category
80 – 100	Fair
60 – 79	Moderate Risk
40 – 59	High Risk
Below 40	Unfair

Category Interpretation:

- **Fair**: Customer-friendly contract with minimal risk
- **Moderate Risk**: Acceptable but requires careful review
- **High Risk**: Contains several unfavorable clauses
- **Unfair**: Strongly biased against the customer

This classification allows users to quickly assess whether a contract is worth accepting, negotiating, or rejecting.

TESTING AND RESULTS

To ensure the reliability, accuracy, and robustness of the AI-driven contract analysis system, extensive testing was performed across multiple scenarios. The testing phase focused on validating the OCR pipeline, SLA extraction accuracy, fairness scoring logic, red flag detection, and API stability.

9.1 Test Scenarios

The system was tested using a variety of real-world contract documents to simulate practical usage conditions.

1. Different Contract Formats

- Lease agreements from banks and financial institutions
- Loan contracts with structured tables and unstructured paragraphs
- Multi-page documents with mixed formatting

This ensured the system could handle diverse document layouts and clause placements.

---

2. Scanned and Digital PDFs

- **Digital PDFs** with selectable text
- **Scanned PDFs** requiring OCR processing

The OCR pipeline successfully extracted text from both formats using Tesseract and Poppler, validating its effectiveness across document types.

3. Missing or Vague Clauses

- Contracts with:
  - Missing grace period details
  - Unclear interest rate definitions
  - Partially mentioned penalty terms

These cases were essential to test:

- Empty field handling
- Fairness score reduction logic
- Red flag triggering accuracy

The system correctly returned empty strings for missing fields and applied fairness penalties where appropriate.

9.2 Results

1. Accurate SLA Extraction

- Key contract fields such as interest rate, EMI, penalties, and foreclosure clauses were extracted accurately.
- Strict JSON-only prompts ensured consistent and structured outputs from the LLM.
- Field names remained fixed, enabling seamless downstream processing.

---

2. Explainable Fairness Scoring

- Fairness scores were computed consistently based on predefined rules.
- Score reductions were clearly linked to specific unfavorable clauses.
- The final score provided an intuitive summary of contract quality.

3. Clear Red Flag Identification

- Risky clauses were correctly identified and labeled as red flags.
- Each red flag included a clear reason and impact explanation.
- This made the system suitable for non-technical users seeking clarity.

---

4. Stable API Responses

- FastAPI handled concurrent requests efficiently.
- Error handling mechanisms prevented application crashes.
- Structured JSON responses were consistently returned for both success and failure cases.

9.3 Performance Observations

- Average response time increased slightly for scanned PDFs due to OCR processing, which is expected.
- Digital PDFs processed faster due to reduced OCR dependency.
- Overall system performance remained within acceptable limits for backend processing.

CONCLUSION

This project successfully demonstrates the design and implementation of an AI-driven backend system for analyzing car lease and loan contracts. By integrating Optical Character Recognition (OCR) techniques with Large Language Models (LLMs) and rule-based evaluation logic, the system effectively converts complex and unstructured contract documents into structured, machine-readable data. This significantly simplifies the process of understanding financial and legal clauses that are otherwise difficult for consumers to interpret.

The use of OCR enables the system to process both scanned and digital PDF contracts, ensuring broad applicability across real-world document formats. The LLM-based extraction engine accurately identifies critical SLA parameters such as interest rates, payment terms, penalties, and contractual obligations, while enforcing strict output formatting to maintain consistency and reliability. Building on this structured data, the Contract Fairness Score and Red Flag detection mechanisms provide an objective and explainable assessment of contract quality, allowing users to quickly identify unfavorable or risky terms.

From a technical perspective, the FastAPI-based backend architecture ensures high performance, modularity, and scalability. The system is designed to support seamless integration with frontend applications, including mobile or web-based user interfaces, and can be easily extended with additional services such as price estimation, VIN lookup, and negotiation assistance. Overall, the project enhances transparency in automotive financing and lays a strong foundation for developing intelligent, consumer-centric contract analysis tools in real-world financial and legal domains.

FUTURE SCOPE

While the current implementation focuses on contract analysis, fairness evaluation, and red flag detection, the system has significant potential for further enhancement and real-world deployment. The following extensions can greatly increase the usefulness and impact of the application:

1. Vehicle Price Benchmarking

Future versions of the system can integrate vehicle pricing data from publicly available sources such as Edmunds, TrueCar, AutoTrader, or government datasets. By comparing the contract price with the fair market value based on vehicle make, model, year, and location, the system can help users identify overpriced deals and strengthen their negotiation position.

2. VIN-Based Car History Analysis

The application can be extended to support Vehicle Identification Number (VIN) analysis by integrating APIs such as the NHTSA Vehicle API and other open vehicle datasets. This would allow users to access manufacturer details, recall history, vehicle specifications, and basic service or accident indicators. Combining VIN insights with contract analysis provides a more holistic evaluation of the car purchase or lease decision.

3. AI-Powered Negotiation Chatbot

An intelligent negotiation assistant can be developed using conversational AI models. This chatbot would guide users on how to negotiate unfavorable contract clauses, suggest counter-offers based on fairness scores and market benchmarks, and generate professional negotiation messages or emails tailored to the user’s contract data.

4. Mobile Application Using Flutter

A cross-platform mobile application built using Flutter can be introduced to improve accessibility and user experience. The mobile app would allow users to upload contracts, view SLA summaries, track fairness scores, receive red flag alerts, and interact with the negotiation chatbot directly from their smartphones.

5. Dealer Comparison Dashboard

A comparison dashboard can be implemented to allow users to upload and analyze multiple contracts from different dealers or financial institutions. The system can then compare interest rates, fees, penalties, and fairness scores across offers, enabling users to select the most consumer-friendly contract and make data-driven decisions.