Project Proposal: Car Lease/Loan Contract Review and Negotiation AI Assistant

1. Project Overview

The **Car Lease or Loan Contract Review and Negotiation App** is an AI-driven mobile application designed to assist consumers in understanding, reviewing, and negotiating their car lease or loan contracts. The application leverages **Large Language Models (LLMs)** for **SLA (Service Level Agreement) extraction**, identifies key contract terms, and provides transparency by cross-verifying vehicle pricing and history using publicly available data sources.

The app acts as a **personal car purchase assistant**, ensuring that users make informed financial decisions when leasing or purchasing a car.

---

2. Goal of the Project

The primary goal of this project is to:

- Enable users to **review and interpret car lease/loan contracts** easily.

- Provide **AI-based extraction of critical clauses** (e.g., interest rates, mileage limits, penalties, early termination conditions).

- Offer **fair market value benchmarks** and **car history reports** to support negotiation decisions.

- Streamline the **car purchase and negotiation process** through a mobile-friendly interface.

---

3. Purpose

The purpose of the app is to **simplify complex car lease or loan agreements** and empower consumers with:

- Transparent contract terms.

- Data-driven insights on car pricing and vehicle condition.

- Tools for comparing offers from different dealers or financial institutions.

- Personalized recommendations to negotiate better lease or loan terms.

---

4. Key Features

4.1. Contract Review and SLA Extraction

Using LLM-based document understanding models (e.g., OpenAI GPT, Claude, or Llama-based fine-tunes), the app will automatically extract:

- **Interest rate / APR**

- **Lease term duration**

- **Monthly payment**

- **Down payment**

- **Residual value**

- **Mileage allowance & overage charges**

- **Early termination clause**

- **Purchase option (buyout price)**

- **Maintenance responsibilities**

- **Warranty and insurance coverage**

- **Penalties or late fee clauses**

Output:
Structured summary of SLA parameters, red flags, and a "Contract Fairness Score".

4.2. Vehicle Price Estimation

- Integrate APIs or web-scraped data from **open or partially open vehicle pricing sources**. Potential providers:

  o **Edmunds** (limited free API data)

  o **TrueCar**

  o **AutoTrader**

  o **NHTSA Vehicle API** (for specifications)

  o **OpenDataSoft** (for DMV and state-level data)

- Use these sources to recommend a **fair purchase price range** or **lease deal benchmark** based on make, model, year, and location.

4.3. VIN-Based Car Information

- Use VIN# lookup to fetch:

    o Manufacturer details

    o Recall history

    o Registration details (from state/county open databases)

    o Past accident or service reports (if available)

    o Odometer discrepancies

- If **Carfax** or **AutoCheck** reports are paid, offer a redirect option while still using **public VIN data** for basic insights.

4.4. Negotiation Assistant

- AI chatbot assists users in:

    o Asking the right questions to dealers.

    o Suggesting negotiation points based on extracted SLAs and market data.

    o Generating negotiation emails or chat messages.

4.5. Frontend (Mobile App)

- Developed using **Flutter** (cross-platform)
- Features:

    o Contract upload (PDF, image)

    o SLA extraction results viewer

    o Comparison dashboard for multiple offers

    o Negotiation chatbot

    o VIN report viewer

    o User authentication & cloud sync

5. Technical Feasibility

Component	Feasibility	Remarks
LLM-based SLA Extraction	✅ High	Fine-tune on sample lease/loan contracts (available online).
Car Price Estimation via Open APIs	⚠️ Medium	Edmunds/KBB data is partially free; fallback to scraping public listings or using NHTSA data + average price aggregation.
VIN Lookup	✅ High	Free VIN data APIs exist (NHTSA, NICB, Data.gov vehicle datasets).
Negotiation AI	✅ High	Custom GPT-based chatbot trained on automotive negotiation best practices.
Mobile Or Web Frontend (React)	✅ High	Mature framework; suitable for rapid cross-platform deployment.

---

6. Sample Data Sources and References

- **Public Lease/Loan Sample Contracts**:

    o   Consumer Financial Protection Bureau (CFPB) sample contracts

    o   Sample auto lease agreements from car manufacturers and leasing companies (e.g., Toyota Financial, Honda Lease sample PDFs)

- **Vehicle APIs**:

    o   NHTSA Vehicle API

    o   Data.gov Vehicle Datasets

    o   Edmunds API (legacy/free tiers)

    o   OpenDataSoft Vehicle Datasets

7. Deliverables

1. **LLM-based SLA Extraction Engine**

   o   Extracts key fields from lease/loan contract PDFs.

   o   Outputs JSON and human-readable summary.

2. **Mobile or Web App (React)**

   o   Contract upload, summary viewer, VIN lookup, negotiation chat.

3. **Integration Layer**

   o   Connects pricing and VIN APIs, processes data for frontend.

4. **Data Pipeline and Knowledge Base**

   o   Repository of sample contracts and extracted clause patterns.

5. **Documentation and Demo**

   o   User manual, architecture docs, and working prototype.

8. Expected Outcomes

- Users gain **clarity and control** over their car lease or loan agreements.

- Enhanced **consumer transparency** in automotive financing.

- Reduction in **unfair lease terms or hidden fees**.

- Prototype ready for pilot with auto-finance or car dealership partners.

9. Milestones & Weekly Deliverables

Milestone 1 – Contract Analysis Engine Setup (Weeks 1–2)

Goal: Build backend foundation for document upload, OCR, and data storage.

Week 1 Deliverables:

- Finalize technical architecture and project plan.

- Collect sample lease/loan contracts from public sources.

- Define SLA fields to extract (APR, term, payments, etc.).

- Set up GitHub repository and cloud environment.

Week 2 Deliverables:

- Implement document upload API.

- Integrate OCR service (Tesseract or Google Vision).

- Store extracted plain text in database.

- Test basic document parsing workflow.

Output:
 Working backend to upload and extract text from contracts.

Milestone 2 – LLM SLA Extraction & VIN Data Integration (Weeks 3–4)

Goal: Use LLM for SLA extraction and integrate VIN lookup API.

Week 3 Deliverables:

- Design LLM prompt for extracting SLA details.

- Implement LLM-based extraction service.

- Store extracted SLA data (JSON format) in database.

- Test extraction accuracy on sample contracts.

Week 4 Deliverables:

- Integrate VIN lookup (NHTSA API).

- Fetch car details (make, model, year, recalls).

- Combine contract and vehicle data in single response.

- Internal testing of complete backend workflow.

Output:
Automated SLA extraction + vehicle data integration working end-to-end.

Milestone 3 – Mobile App MVP & Negotiation Assistant (Weeks 5–6)

Goal: Build mobile app for contract upload, SLA viewing, and negotiation help.

Week 5 Deliverables:

- Create Flutter app structure (login, upload, dashboard).

- Connect upload API to backend.

- Display extracted SLA summary (APR, payment, term).

Week 6 Deliverables:

- Add AI-based negotiation chatbot for user guidance.

- Implement message threads for dealer conversations.

- UI improvements for contract comparison and insights.

- Test with sample user flows.

Output:
Mobile app MVP with working upload, extraction, and negotiation assistant.

Milestone 4 – Price Estimation, Testing & Demo Release (Weeks 7–8)

Goal: Add price recommendation, improve performance, and prepare final demo.

Week 7 Deliverables:

- Integrate price estimation APIs (Edmunds, TrueCar, NHTSA).

- Show fair price range and compute "Contract Fairness Score."

- Run QA tests on different contract formats.

Week 8 Deliverables:

- Perform end-to-end testing (app + backend + LLM).

- Finalize documentation (API, DB, deployment).

- Prepare demo video and presentation for stakeholders.

- Deploy backend and app beta version.

Output:
 Fully functional prototype with price insights, SLA extraction, and negotiation assistant.

10. DB Schema

users

id 🔑		uuid
email	varchar(255)	NN
phone	varchar(40)	
full_name	varchar(200)	
auth_provider	varchar(50)	
created_at	timestamptz	NN
updated_at	timestamptz	NN

vehicle_recalls

id 🔑		uuid
vehicle_id 🔗	uuid	NN
recall_number	varchar(100)	
issue_date	date	
component	varchar(255)	
summary	text	
remedy	text	
source	varchar(120)	
raw	jsonb	
created_at	timestamptz	

integration_logs

id 🔑		bigserial
provider_id 🔗	uuid	
request_path	varchar(512)	
request_params	jsonb	
response_status	int	
response_ms	int	
occurred_at	timestamptz	
error_message	text	

extractions

id 🔑		uuid
contract_id 🔗	uuid	NN
model_name	varchar(120)	
prompt_version	varchar(60)	
status	extraction_status E	NN
started_at	timestamptz	
completed_at	timestamptz	
raw_output	jsonb	
error_message	text	

price_sources

id 🔑		uuid
price_reco_id 🔗	uuid	NN
provider_id 🔗	uuid	
sample_size	int	
median_price	numeric(12,2)	
min_price	numeric(12,2)	
max_price	numeric(12,2)	
url	varchar(512)	
raw	jsonb	

audit_events

id 🔑		bigserial
user_id 🔗	uuid	
entity_table	varchar(80)	
entity_id	uuid	
action	varchar(40)	
details	jsonb	
occurred_at	timestamptz	

dealers

id 🔑		uuid
name	varchar(255)	NN
address_line1	varchar(255)	
address_line2	varchar(255)	
city	varchar(120)	
state	varchar(60)	
postal_code	varchar(20)	
country	varchar(60)	
phone	varchar(40)	
website	varchar(255)	
created_at	timestamptz	

vehicle_reports

id 🔑		uuid
vehicle_id 🔗	uuid	NN
provider_id 🔗	uuid	
title	varchar(255)	
report_type	varchar(80)	
availability	varchar(40)	
url	varchar(512)	
raw	jsonb	
created_at	timestamptz	

contracts

id 🔑		uuid
user_id 🔗	uuid	NN
vehicle_id 🔗	uuid	
dealer_id 🔗	uuid	
lender_id 🔗	uuid	
contract_type	contract_type E	NN
doc_status	doc_status E	NN
dealer_offer_name	varchar(200)	
contract_date	date	
locale	varchar(10)	
currency	varchar(3)	
fairness_score	numeric(5,2)	
red_flag_level	red_flag_level E	
notes	text	
created_at	timestamptz	
updated_at	timestamptz	

contract_sla

id 🔑		uuid
contract_id 🔗	uuid	NN
apr_percent	numeric(6,3)	
money_factor	numeric(10,6)	
term_months	int	
monthly_payment	numeric(12,2)	
down_payment	numeric(12,2)	
fees_total	numeric(12,2)	
residual_value	numeric(12,2)	
residual_percent_msrp	numeric(6,3)	
msrp	numeric(12,2)	
cap_cost	numeric(12,2)	
cap_cost_reduction	numeric(12,2)	
mileage_allowance_yr	int	
mileage_overage_fee	numeric(8,4)	
early_termination_fee	numeric(12,2)	
disposition_fee	numeric(12,2)	
purchase_option_price	numeric(12,2)	
insurance_requirements	text	
maintenance_resp	text	
warranty_summary	text	
late_fee_policy	text	
other_terms	jsonb	
created_at	timestamptz	
updated_at	timestamptz	

negotiation_threads

id 🔑		uuid
user_id 🔗	uuid	NN
contract_id 🔗	uuid	NN
dealer_id 🔗	uuid	
lender_id 🔗	uuid	
channel	negotiation_channel E	NN
subject	varchar(255)	
created_at	timestamptz	
closed_at	timestamptz	

tagging

id 🔑		uuid
tag	varchar(80)	NN
entity_table	varchar(80)	NN
entity_id	uuid	NN
created_at	timestamptz	

lenders

id 🔑		uuid
name	varchar(255)	NN
nmls_id	varchar(50)	
website	varchar(255)	
phone	varchar(40)	
address	varchar(255)	
created_at	timestamptz	

providers

id 🔑		uuid
name	varchar(120)	NN
kind	provider_kind E	NN
is_free	boolean	NN
base_url	varchar(255)	
notes	text	
created_at	timestamptz	

contract_files

id 🔑		uuid
contract_id 🔗	uuid	NN
storage_uri	varchar(512)	NN
file_name	varchar(255)	
mime_type	varchar(120)	
page_count	int	
uploaded_at	timestamptz	

extracted_clauses

id 🔑		uuid
extraction_id 🔗	uuid	NN
clause_type	varchar(120)	
page_number	int	
text_snippet	text	
normalized_value	jsonb	
red_flag_level	red_flag_level E	
comment	text	

negotiation_messages

id 🔑		uuid
thread_id 🔗	uuid	NN
sender_role	varchar(40)	
body	text	
suggested_text	text	
attachments	jsonb	
sent_at	timestamptz	

vehicles

id 🔑		uuid
vin	varchar(17)	NN
year	int	
make	varchar(120)	
model	varchar(120)	
trim	varchar(120)	
body_class	varchar(120)	
engine	varchar(120)	
drivetrain	varchar(120)	
fuel_type	varchar(60)	
odometer_miles	numeric(10,1)	
color_ext	varchar(80)	
color_int	varchar(80)	
created_at	timestamptz	
updated_at	timestamptz	

provider_credentials

id 🔑		uuid
provider_id 🔗	uuid	NN
label	varchar(120)	
config	jsonb	NN
created_at	timestamptz	

contract_pages

id 🔑		uuid
contract_file_id 🔗	uuid	NN
page_number	int	NN
ocr_text	text	
ocr_confidence	numeric(5,2)	
thumbnail_uri	varchar(512)	
created_at	timestamptz	

price_recommendations

id 🔑		uuid
vehicle_id 🔗	uuid	NN
geo_postal	varchar(20)	
msrp	numeric(12,2)	
fair_price_low	numeric(12,2)	
fair_price_high	numeric(12,2)	
basis	varchar(160)	
methodology	text	
generated_at	timestamptz	

offer_comparisons

id 🔑		uuid
user_id 🔗	uuid	NN
primary_contract 🔗	uuid	NN
compared_contract 🔗	uuid	NN
comparison_json	jsonb	
created_at	timestamptz	