Car Lease Contract Review and Negotiation AI Assistant

AI/LLM-Based System for Automotive Contract Analysis

1. Executive Summary

The Car Lease Contract Review and Negotiation AI Assistant is an intelligent system designed to empower consumers in understanding and negotiating automotive lease and loan contracts. Leveraging state-of-the-art Large Language Models (LLMs) and advanced document processing techniques, the system provides comprehensive contract analysis, fairness scoring, and actionable insights.

1.1 Key Capabilities

- Automated extraction of 50+ contract parameters including financial terms, conditions, and obligations
- AI-powered fairness analysis with transparent scoring methodology across 5 key dimensions
- Real-time red flag detection for potentially unfavorable contract terms
- VIN-based vehicle verification and history lookup
- Market price benchmarking and comparison tools
- Interactive web interface for seamless user experience

2. Project Overview

2.1 Problem Statement

Car lease and loan contracts are complex legal documents filled with technical jargon, hidden fees, and conditions that average consumers struggle to understand. This information asymmetry between dealers/financial institutions and consumers often leads to unfavorable terms, unexpected costs, and buyer's remorse.

2.2 Solution Approach

Our solution employs artificial intelligence to democratize contract understanding. By combining optical character recognition (OCR), natural language processing (NLP), and Google's Gemini LLM, we transform opaque legal documents into clear, structured data that empowers informed decision-making.

2.3 Target Users

- Individual consumers leasing or financing vehicles
- Small business fleet managers
- Financial advisors assisting clients with vehicle purchases
- Consumer advocacy organizations
- Legal professionals reviewing automotive contracts

3. System Architecture

The system follows a modular, layered architecture designed for scalability, maintainability, and extensibility. The architecture consists of five main layers:

3.1 Presentation Layer

The presentation layer provides the user interface through a modern, responsive web application built with HTML5, CSS3, and vanilla JavaScript. Key features include:

- Drag-and-drop PDF upload interface
- Real-time processing status with animated feedback
- Structured data visualization with expandable sections
- Mobile-responsive design for cross-device compatibility
- Elegant gradient-based UI with professional aesthetics

3.2 API Layer

Built on FastAPI, the API layer provides RESTful endpoints for all system operations. The API architecture includes:

- FastAPI framework for high-performance async operations
- CORS middleware for secure cross-origin requests
- Comprehensive error handling and validation
- OpenAPI/Swagger documentation at /docs endpoint
- Health check endpoints for system monitoring
- File upload handling with type validation

3.3 Business Logic Layer

The core processing layer implements the intelligent contract analysis pipeline:

- Document text extraction (extract_text.py) - Handles PDF parsing with OCR fallback
- LLM contract processing (fineline.py) - Orchestrates Gemini API for structured extraction
- Fairness scoring engine (fairness_score.py) - Implements multi-dimensional contract evaluation
- Red flag detection (red_flags.py) - Pattern matching for problematic contract terms
- Utility functions (utils.py) - Common data validation and processing helpers

3.4 AI/LLM Layer

The AI layer leverages Google's Gemini Flash model for intelligent document understanding:

- Google Generative AI (Gemini Flash) for rapid, cost-effective processing
- Structured prompt engineering for consistent JSON output

- Error handling and retry logic for API resilience
- Safe JSON parsing with fallback mechanisms
- Context window management for large documents

3.5 Data Processing Layer

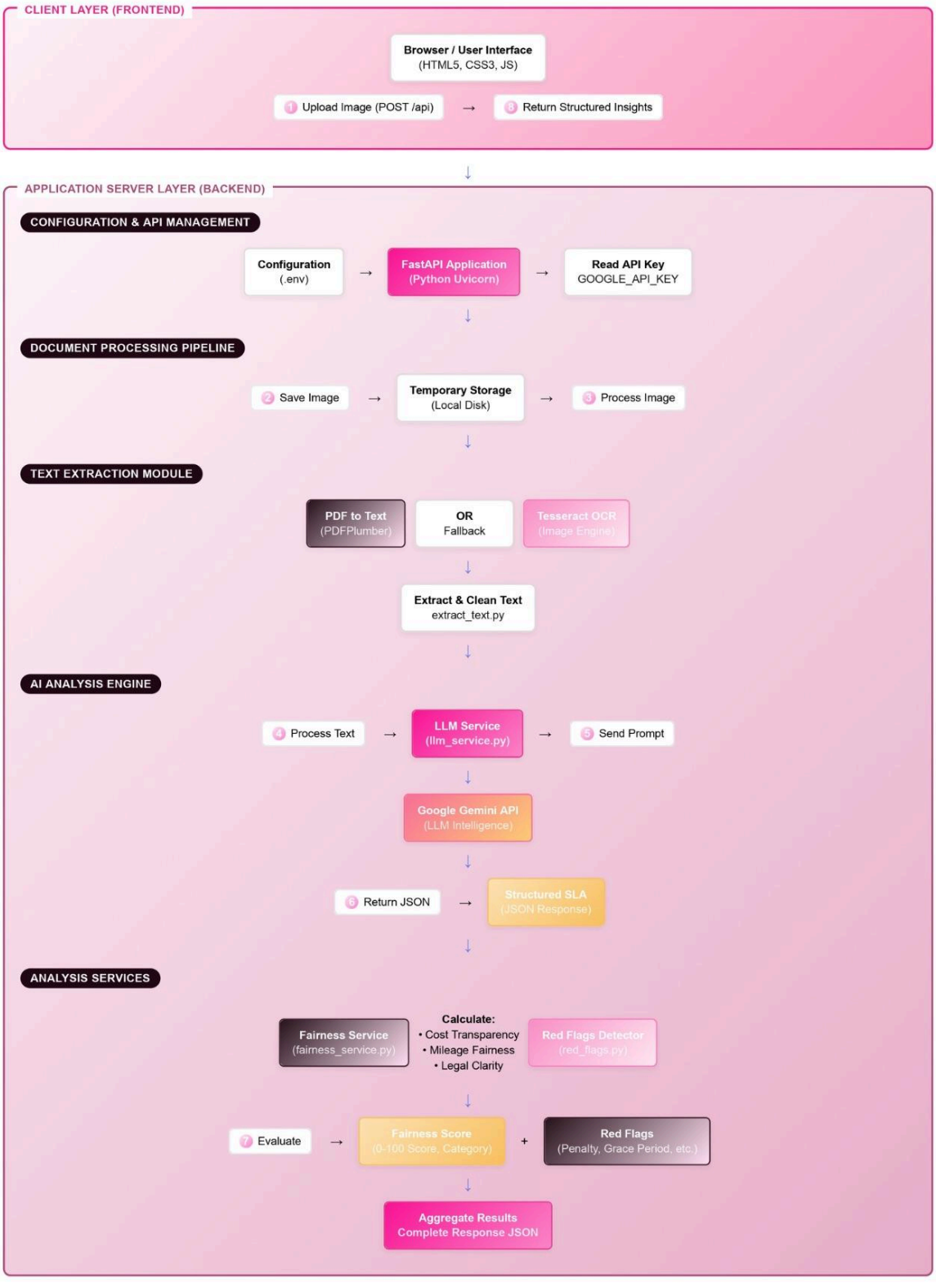
The foundation layer handles document ingestion and text extraction:

- PDFPlumber for native PDF text extraction
- PDF2Image + Pytesseract for OCR processing of scanned documents
- Intelligent fallback mechanism (native extraction → OCR if needed)
- Text preprocessing and noise reduction
- Temporary file management with automatic cleanup

System Architecture: OCR & Gemini Intelligent Contract Analysis

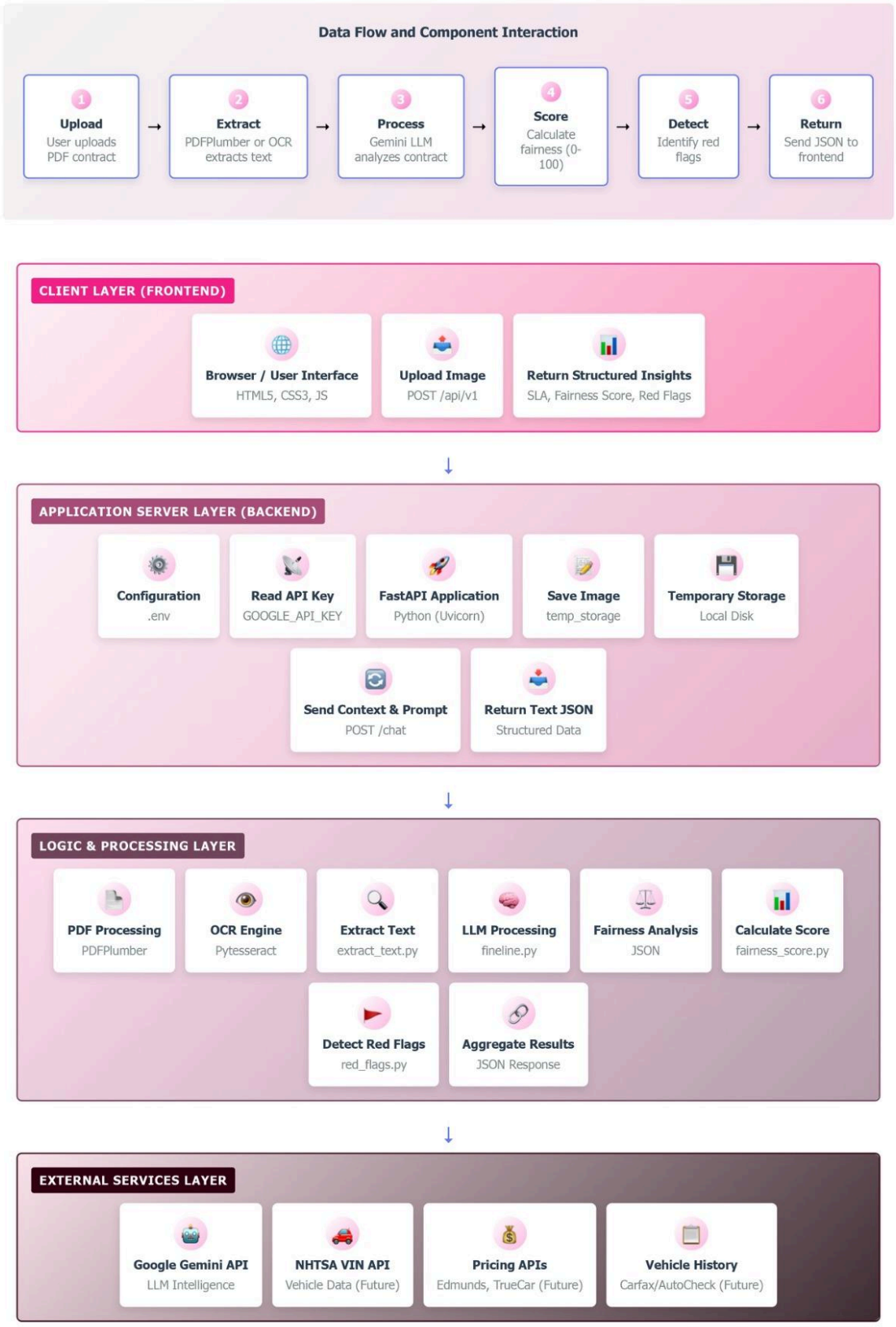
Automated Contract SLA Extraction and Fairness Analysis System Architecture

Figure 1: System Architecture - Data Flow and Component Interaction



System Architecture: AI-Powered Car Lease Contract Analysis

Automated Contract SLA Extraction and Fairness Analysis System



4. System Processing Flow

The system processes lease contracts through a sophisticated six-stage pipeline:

Stage 1: Document Upload

The user uploads PDF via web interface. The system validates file type and size, generates a unique identifier, and stores temporarily for processing.

Stage 2: Text Extraction

PDFPlumber attempts native text extraction. If unsuccessful (scanned document), PDF2Image converts pages to images, and Pytesseract performs OCR. Raw text is validated for content presence.

Stage 3: Text Preprocessing

Raw text undergoes cleaning: email addresses removed, long numeric sequences filtered, whitespace normalized. First 3,500 characters prioritized for LLM processing.

Stage 4: LLM Extraction

Gemini Flash processes cleaned text with structured prompts containing 50+ field definitions across 9 major sections. Model returns JSON-formatted contract data with "Information not available" for missing fields.

Stage 5: Analysis & Scoring

Extracted data flows through two parallel analyzers: Fairness Scorer evaluates a contract across 5 dimensions (Cost Transparency, Mileage Fairness, Termination Clarity, Residual Logic, Legal Clarity) generating 0-100 score. Red Flag Detector identifies potentially problematic terms using pattern matching.

Stage 6: Response Generation

System aggregates all analysis results, injects fairness metrics and red flags into structured response, cleans up temporary files, and returns JSON to frontend for visualization.

5. Data Extraction Schema

The system extracts structured information across nine comprehensive categories, covering all critical aspects of automotive lease contracts:

1. Vehicle Consultancy/Dealer Info

- Vehicle Consultancy Name
- Contact Details
- Dealer/Lessor Name
- Location

2. Vehicle Identification & Basic Details

- Car Name/Model
- Variant
- Vehicle Identification Number (VIN)
- Registered Number
- VAT Number
- Manufacturer
- Manufacturer OTR (On-The-Road Price)
- P11D (Official List Price - UK)

3. Vehicle Specifications

- Body Type
- Car Color
- Fuel Type
- Transmission
- CO_2 Emissions
- Vehicle Depreciation Rate per Month
- Mileage and Tenure Limit

4. Lease Terms/Agreement Details

- Agreement Duration/Term Period
- Rental Period (Start/End Dates)
- Expiry Date
- Termination Date
- Early Termination Fee
- Mileage Allowance
- Maintenance Responsibility
- Insurance Management Requirements

- Other Terms and Conditions/Disclaimer

5. Payment Details

- Upfront/Signing Payments:
- - Lease Signing Payment
- - Capitalized Cost Reduction
- - Net Trade-In Allowance
- - Down Payment
- - First Monthly Payment
- - Refundable Security Deposit
- - Amount to be Paid in Cash
- - Title Fees
- - Registration Fees
- - Processing Fee
- Recurring Payments:
- - Monthly Payments
- - Monthly Sales/Use Tax
- - Other Charges
- - Total Monthly Payment
- - Total of Payments
- - Amortized Amount over Lease

6. Taxes & Additional Fees

- VAT/Sales Tax
- Other Fees and Taxes

7. Residual & End-of-Lease Details

- Residual Value
- Vehicle Depreciation Considered
- Options at Lease End

8. Fairness Analysis (Auto-Generated)

- Overall Fairness Score (0-100)
- Risk Category (Fair/Needs Review/High Risk)
- Cost Transparency Score
- Mileage Fairness Score
- Termination Clarity Score
- Residual Logic Score

- Legal Clarity Score

9. Red Flags (Auto-Detected)

- List of identified problematic terms
- Missing critical information warnings
- Unusual or unfavorable conditions

6. Fairness Scoring Methodology

The fairness scoring system evaluates contracts across five critical dimensions, providing a transparent, objective assessment of contract quality. The scoring algorithm starts at 100 points and applies deductions based on missing or problematic information.

Cost Transparency (30 points)

- Monthly Payments disclosed: -10 points if missing
- Total of Payments disclosed: -10 points if missing
- Processing Fee disclosed: -5 points if missing
- VAT/Sales Tax disclosed: -5 points if missing
- Rationale: Financial transparency is the foundation of fair contracts

Mileage Fairness (20 points)

- Mileage Allowance specified: -10 points if missing
- Excess Mileage charges disclosed: -10 points if missing
- Rationale: Unexpected mileage charges are a common source of end-of-lease disputes

Termination Clarity (20 points)

- Early Termination Fee specified: -10 points if missing
- Rationale: Exit costs must be clear to avoid financial surprises

Residual & Depreciation Logic (15 points)

- Residual Value disclosed: -10 points if missing
- Depreciation Rate specified: -5 points if missing
- Rationale: End-of-lease value impacts buyout options and negotiation power

Legal Clarity (15 points)

- Maintenance Responsibility defined: -5 points if missing
- Insurance Requirements specified: -5 points if missing
- Terms and Conditions present: -5 points if missing
- Rationale: Clear obligations prevent disputes and unexpected costs

Score Interpretation

Final scores are categorized into three risk levels:

- 80-100 points: FAIR - Contract demonstrates high transparency and completeness
- 60-79 points: NEEDS REVIEW - Some important terms are missing or unclear
- 0-59 points: HIGH RISK - Multiple critical terms are missing or problematic

7. Red Flag Detection System

The red flag detector identifies potentially problematic contract terms using pattern matching and semantic analysis. Flags are triggered when:

Missing Total Cost

The contract lacks "Total of Payments" disclosure, making it impossible to calculate the true cost of the lease.

Undefined Early Termination

Early termination fees are not specified, exposing consumers to potentially unlimited penalties.

Missing Residual Value

Residual value is not disclosed, preventing assessment of end-of-lease buyout options.

Lessee Maintenance Burden

The contract explicitly states maintenance is the lessee's responsibility, potentially resulting in unexpected repair costs.

Unilateral Price Changes

Terms include language like "subject to change" allowing the lessor to modify pricing without notice.

8. Technical Implementation Details

8.1 Technology Stack

Backend Framework: FastAPI 0.104+ (Python 3.11+)

AI/LLM: Google Generative AI (Gemini Flash)

PDF Processing: PDFPlumber, PDF2Image, Pytesseract

Frontend: HTML5, CSS3, Vanilla JavaScript

API Documentation: OpenAPI/Swagger (auto-generated)

Development Server: Uvicorn ASGI server

Testing: Custom test suite with colored output

8.2 API Endpoints

Endpoint	Description
GET /	Root endpoint - API information and available endpoints
GET /health	Global health check
GET /lease/health	Lease service health check
POST /lease/extract	Main endpoint: Upload PDF and receive analysis
GET /docs	Interactive Swagger/OpenAPI documentation
GET /ui	Web interface for contract analysis

8.3 File Structure

Project organization follows modular best practices:

```
CAR_LEASE_AI/
│
├── app/
│   ├── __init__.py
│   ├── main.py              # FastAPI application entry point
│   │
│   ├── routers/
│   │   ├── __init__.py
│   │   └── lease.py         # Lease extraction endpoints
│   │
│   └── templates/
│       └── index.html       # Web UI
│
├── contracts/               # Sample lease PDFs for testing
│
├── extracted_text/          # Extracted text cache
│
├── fineline_output/         # LLM processing results
│
├── Static/                  # Static assets (CSS, images)
│   ├── cars.png
│   └── style.css
│
├── uploads/                 # Temporary upload storage
│
├── extract_text.py          # PDF text extraction module
├── fineline.py              # LLM processing orchestration
├── fairness_score.py        # Contract fairness scoring
├── red_flags.py             # Red flag detection logic
├── utils.py                 # Common utilities
├── test.py                  # Comprehensive test suite
│
└── requirements.txt         # Python dependencies
```

8.4 Error Handling

The system implements comprehensive error handling at multiple levels:

- File type validation (PDF only)
- Empty document detection
- OCR fallback for scanned documents
- LLM API error recovery with detailed logging
- JSON parsing with safe fallback mechanisms
- HTTP status codes: 200 (success), 400 (bad request), 422 (validation error), 500 (server error)
- Structured error responses with error type and details
- Automatic temporary file cleanup (finally blocks)

9. Testing & Quality Assurance

A comprehensive test suite (test.py) validates all system components and integration points:

Test 1: Root Endpoint

Validates API accessibility and metadata

Test 2: Health Checks

Confirms both global and service-level health

Test 3: Invalid File Upload

Ensures proper rejection of non-PDF files

Test 4: Valid PDF Extraction

End-to-end test with real lease PDFs

Test 5: API Documentation

Verifies Swagger/OpenAPI accessibility

9.6 Test Output Features

- Color-coded console output (green=pass, red=fail, yellow=skip)
- Detailed validation of response structure
- Field population statistics
- Processing time measurement
- JSON output file generation for inspection
- Comprehensive test summary report

10. Current Implementation Status

The system has successfully completed Milestones 1-3 of the original project plan, with core functionality fully operational:

- PDF upload and text extraction (with OCR fallback) ✓
- LLM-based contract data extraction ✓
- Structured JSON output across 50+ fields ✓
- Fairness scoring algorithm ✓
- Red flag detection system ✓
- FastAPI backend with error handling ✓
- Web-based user interface ✓
- Comprehensive test suite ✓
- API documentation (Swagger) ✓

10.2 Pending Features (Milestone 4)

- VIN lookup integration (NHTSA API)
- Vehicle price estimation (Edmunds, TrueCar APIs)
- Market price benchmarking
- AI negotiation chatbot assistant
- Multi-contract comparison dashboard
- User authentication and profile management
- Cloud storage integration
- Mobile app development (Flutter)

11. Future Enhancements & Roadmap

11.1 Phase 2: Advanced Analytics

- Historical contract database for comparative analysis
- Machine learning model for contract risk prediction
- Natural language negotiation suggestion engine
- Automated contract clause comparison
- Dealer reputation scoring based on contract terms

11.2 Phase 3: Ecosystem Integration

- Integration with DMV and state registration databases
- Real-time credit score analysis for rate validation
- Partnership with insurance providers for bundled quotes
- Blockchain-based contract verification
- Integration with dealer management systems

11.3 Phase 4: Scale & Enterprise

- Enterprise licensing for dealerships
- White-label solutions for financial institutions
- API access for third-party integration
- Multi-language support for international markets
- Advanced analytics dashboard for business intelligence

12. Deployment & Operations

12.1 Development Setup

To run the system locally:

```
# 1. Clone repository
git clone <repository-url>
cd car_lease_ai

# 2. Install dependencies
pip install fastapi uvicorn pdfplumber pdf2image pytesseract google-generativeai

# 3. Set environment variables
export GOOGLE_API_KEY="your-gemini-api-key"

# 4. Run development server
uvicorn app.main:app --reload --host 0.0.0.0 --port 8000

# 5. Access application
# Web UI: http://localhost:8000/ui
# API Docs: http://localhost:8000/docs
```

12.2 Production Deployment

Recommended production architecture:

- Container orchestration: Docker + Kubernetes
- Web server: Nginx as reverse proxy
- Application server: Gunicorn with Uvicorn workers
- Database: PostgreSQL for contract storage
- Cache layer: Redis for API response caching
- File storage: AWS S3 or Google Cloud Storage
- Monitoring: Prometheus + Grafana
- Logging: ELK Stack (Elasticsearch, Logstash, Kibana)

12.3 Security Considerations

- API rate limiting to prevent abuse
- File upload size limits (max 10MB)
- Virus scanning for uploaded PDFs
- Secure storage of API keys using environment variables
- HTTPS enforcement in production
- CORS configuration for authorized domains only
- Input sanitization and validation
- Automated security scanning (OWASP)
- Regular dependency updates for vulnerability patches

13. Performance & Scalability

13.1 Current Performance Metrics

- Average processing time: 10-30 seconds per contract
- PDF text extraction: 1-3 seconds
- LLM processing: 5-15 seconds (depending on contract length)
- Fairness scoring: <1 second
- Red flag detection: <1 second
- Supported file size: Up to 10MB PDF
- Concurrent users (single instance): 10-20 users

13.2 Scalability Strategy

Horizontal scaling approach for production:

- Stateless API design enables load balancing across multiple instances
- Task queue (Celery + Redis) for asynchronous processing
- Database read replicas for query scaling
- CDN for static asset delivery
- API response caching for repeated queries
- Auto-scaling based on CPU/memory metrics
- Microservices architecture for independent component scaling

13.3 Optimization Opportunities

- LLM prompt optimization to reduce token usage
- Caching of common contract patterns
- Batch processing for multiple contracts
- Model fine-tuning for faster inference
- Database query optimization and indexing
- Lazy loading of non-critical UI components

14. Conclusion

The Car Lease Contract Review and Negotiation AI Assistant represents a significant advancement in consumer empowerment for automotive transactions. By leveraging cutting-edge AI technology, the system transforms complex legal documents into actionable insights, enabling consumers to make informed decisions and negotiate better terms.

The current implementation demonstrates the viability of the core concept with a fully functional extraction, analysis, and scoring pipeline. The modular architecture provides a solid foundation for future enhancements, including VIN lookup, market pricing, and AI-powered negotiation assistance.

As the system evolves through subsequent phases, it has the potential to become an essential tool for millions of consumers, reducing information asymmetry in the automotive marketplace and promoting fairer, more transparent transactions.

Appendix A: Sample API Request/Response

Sample cURL request:

```
curl -X POST "http://localhost:8000/lease/extract" \
  -H "accept: application/json" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@lease_contract.pdf"
```

Sample JSON response structure:

```
{
  "success": true,
  "filename": "lease_contract.pdf",
  "lease_details": {
    "1. Vehicle Consultancy / Dealer Info": {
      "Vehicle Consultancy Name": "ABC Motors",
      ...
    },
    "2. Vehicle Identification & Basic Details": { ... },
    ...
    "8. Fairness Analysis": {
      "Fairness Score": "85 / 100",
      "Risk Category": "Fair",
      ...
    },
    "9. Red Flags": { ... }
  },
  "fairness_analysis": {
    "fairness_score": 85,
    "category": "Fair",
    "breakdown": { ... }
  },
  "red_flags": [ ... ]
}
```