 # AI-Powered Construction Claims Analysis MVP - Requirements Document

1. Functional Requirements

- The system must allow users to **upload construction contracts** in PDF and DOCX formats.
- It must **extract text** from uploaded documents using OCR and DOCX parsing.
- The system must **analyze contract clauses** to identify potential risks such as **delays, disputes, and contract breaches**.
- It should **categorize risks** into predefined categories (e.g., Delay, Payment Dispute, Contract Breach).
- The system must generate **risk scores** for identified clauses using a **Machine Learning model**.
- Users should be able to **download structured reports** in PDF and CSV formats.
- A **web-based dashboard** must be available for file uploads, analysis results, and report downloads.
- The system must support **Generative AI-based insights** for explaining risks and suggesting actions.
- Users should be able to **ask contract-related questions** via an AI-powered chatbot.
- The system should provide **automated claim drafting** based on identified risks.

2. Non-Functional Requirements

- The application must be **secure** with access control for contract data.
- It should support **scalability** to handle multiple document uploads and analyses.
- The system must be **extensible**, allowing integration with external legal databases and AI APIs.
- Performance must allow for **real-time risk analysis** without excessive delays.
- The system should be deployable on **cloud platforms** like AWS, Azure, or GCP.

3. Technical Requirements

- **Backend:** Python (Flask), spaCy for NLP, scikit-learn for ML, OpenAI API (or fine-tuned LLMs) for Generative AI.

- **Frontend:** React.js for web-based dashboard.
- **Storage:** Local or cloud-based storage for document uploads and reports.
- **OCR Processing:** Tesseract OCR and pdf2image for text extraction from PDFs.
- **Machine Learning Model:** Pre-trained or fine-tuned ML model for risk scoring.
- **Deployment:** Containerized using Docker, with optional cloud hosting.

4. Use Cases of Construction Claims

4.1 Delay Claims

- A contractor experiences **unexpected delays** due to extreme weather conditions.
- The AI system detects a **force majeure** clause in the contract and suggests potential **legal arguments** for an extension.
- A **risk score** is assigned to the delay and included in a claim draft.

4.2 Payment Dispute Claims

- A subcontractor is **not paid on time** due to a disagreement over work quality.
- The AI identifies a **payment clause** in the contract and provides **recommendations** based on past case precedents.
- A claim is auto-generated with details of the dispute and legal grounds for resolution.

4.3 Contract Breach Claims

- A contractor fails to complete a project due to unforeseen material shortages.
- The system detects a **breach of contract clause** and highlights potential **penalties or renegotiation terms**.
- AI suggests possible **compensation strategies** to mitigate losses.

4.4 Liquidated Damages Claims

- A project is completed **beyond the agreed deadline**, triggering financial penalties.
- AI evaluates whether **delay penalties are enforceable** and suggests possible **legal defenses**.
- The system generates a claim document with a recommended **negotiation strategy**.

4.5 Termination Claims

- A client terminates a contract **without valid justification**.
- AI reviews termination clauses and provides insights on **wrongful termination claims**.
- The system suggests legal references for potential **compensation or reinstatement**.

This document outlines the core requirements and use cases for the MVP, serving as a reference for development and future enhancements.

Evaluating construction claims involves a systematic analysis of contract terms, project records, and legal principles to determine entitlement and quantum (the amount claimed). Here's a structured approach:

1. Identify the Type of Claim

Construction claims typically fall into these categories:

- **Delay Claims** (e.g., late completion, extended overhead costs)
- **Disruption Claims** (e.g., productivity loss due to interference)
- **Acceleration Claims** (e.g., costs for speeding up work)
- **Payment Claims** (e.g., unpaid invoices, change order disputes)
- **Defective Work Claims** (e.g., remedial costs, warranty disputes)
- **Scope Change Claims** (e.g., extra work, variation orders)

2. Review the Contract & Legal Framework

- Analyze **contract clauses** related to changes, delays, force majeure, liquidated damages, dispute resolution, etc.
- Check **jurisdictional laws** and case law precedents that may impact the claim.
- Verify whether **notice requirements** were met (e.g., timely notification of delays or extra costs).

3. Examine Project Documentation

- **Baseline schedule & updates** – Identify critical path impacts.
- **Daily logs & site reports** – Validate events leading to claims.
- **Correspondence & emails** – Identify owner-contractor communications.
- **Change orders & approvals** – Verify scope changes and cost agreements.
- **Payment records & invoices** – Ensure proper documentation for financial claims.
- **Subcontractor & supplier records** – Check delays and extra costs.

4. Analyze Entitlement & Causation

- **Causation:** Prove that the event directly caused the claimed loss or cost.
- **Contractual entitlement:** Establish if the contract allows compensation for the issue.
- **Mitigation efforts:** Determine whether the claimant attempted to minimize losses.

5. Quantify the Claim (Quantum Assessment)

- Use **cost-based methods** (e.g., actual costs incurred, cost-plus calculations).
- Apply **schedule analysis** techniques like:
    - **Critical Path Method (CPM)** – Identifies time impacts.
    - **Time Impact Analysis (TIA)** – Evaluates how events affect project timelines.
- Consider **expert opinions** (e.g., forensic schedule analysts, quantity surveyors).

6. Evaluate Counterclaims & Defenses

- Assess the other party's position (e.g., improper documentation, contributory delays, failure to mitigate losses).
- Identify possible defenses such as **concurrent delays** (both parties responsible) or **contract breaches** by the claimant.

7. Negotiate or Pursue Dispute Resolution

- Attempt **amicable settlement** through negotiations.
- Use Alternative Dispute Resolution (**ADR**) methods like mediation or arbitration.
- Proceed to **litigation** if necessary, though this is often a last resort.

Would you like a more detailed framework or a checklist for claim evaluation?

Some Example:

1. Advanced Risk Analysis & Clause Explanation

- Instead of just classifying contract clauses, a **Generative AI model** can explain the risk in **natural language**.
- Example Output:
    - *"This clause suggests a high financial risk due to liquidated damages. The contractor may be penalized for any delays, potentially affecting project margins."*

2. Automated Claim Drafting

- AI can generate **initial claim documents** based on contract breaches or disputes.
- Example Output:
    - *"Dear [Client], according to Clause X, the contract states that delays beyond 30 days result in penalties. We request a revision of the timeline under force majeure conditions."*

3. Conversational AI for Contract Queries

- Users can **ask questions about the contract**, and the AI chatbot can provide insights.
- Example Interaction:
    - **User:** *"What are the penalties for late project completion?"*
    - **AI:** *"According to Clause X, late completion incurs a penalty of $10,000 per day after the deadline."*