**SALESFORCE PROJECT DOCUMENTATION**

**LEGAL CONTRACT ANALYZER**

Name : N.K.LAKSHMI NARASIMHA MURTHY

Roll No: 22B81A6684

Registered mailId : [nlnara20lm@gmail.com](mailto:nlnara20lm@gmail.com)

College Name : CVR COLLEGE OF ENGINEERING

LEGAL CONTRACT ANALYZER

1. Project Overview

* **Purpose**: Develop an AI-powered legal contract analyzer and negotiator within Salesforce to streamline contract review, speed up negotiations, reduce legal risk, and improve compliance.
* **Who benefits**: Enterprises, legal teams, law firms, compliance officers, and independent professionals managing large volumes of contracts.
* **Business Value**:
  + Automated clause extraction and risk scoring
  + AI-assisted negotiation suggestions
  + Real-time dashboards and status notifications
  + Auditability, enterprise security, and scalability on Salesforce

2. Project Goals & Objectives

* **Automate contract reviews** by extracting clauses and performing risk analysis.
* **Improve accuracy** through standardized risk scoring and clause detection.
* **Accelerate negotiations** with AI-generated alternative wording and reasoning.
* **Ensure compliance** with GDPR/CCPA and enterprise legal policies.
* **Provide visibility** via real-time dashboards, reports, and audit trails.
* **Enterprise readiness** through bulkified Apex triggers and test coverage ≥ 75% (achieved: **81%**).

A screenshot of a computer

Description automatically generated1.1 Legal Analyzer custom Salesforce app with dedicated navigation tabs for Contracts, Clauses, and Negotiation Suggestions

3. Salesforce Key Features and Concepts Utilized

* **Custom Objects**
  + Contract\_\_c: stores contract metadata, status, analysis results.
  + Clause\_\_c: stores clause details, positions, risk level, links to parent contract.
  + NegotiationSuggestion\_\_c: holds AI-suggested revisions, tone, goal, confidence level.
  + Status\_Change\_Event\_\_e: platform event to notify external or async processes.
* **Relationships**
  + Master-Detail: Clause\_\_c → Contract\_\_c
  + Master-Detail: NegotiationSuggestion\_\_c → Clause\_\_c
  + Lookup: Uploaded\_By\_\_c and Requested\_By\_\_c → User
* **Apex Classes**
  + ContractAnalysisService: extracts clauses, computes risk, updates parent contract.
  + AINegotiationService: generates AI-based negotiation text with rationale and confidence.
  + ContractDashboardController: provides KPIs and lists for Lightning Web Components.
  + StatusNotificationService: sends automated status change notifications.
* **Apex Triggers**
  + ContractStatusTrigger: executes logic on contract approval/rejection.
  + ClauseStatusTrigger: executes logic on clause status updates.
* **Lightning Web Component (LWC)**
  + contractDashboard: displays KPIs, recent contracts, and risk distribution charts.
* **Validation & Auto-numbering**
  + Auto numbers: CT-{00000}, CL-{00000}, NS-{00000}
  + Required fields and picklists for statuses, risk levels, and clause types
* **Testing**
  + Apex test classes created for services, triggers, and controllers.
  + Achieved **81% org-wide test coverage**.

A screenshot of a computer

Description automatically generated

3.1 Custom data model designed for contracts, clauses, and negotiation suggestions with Master-Detail relationships

A screenshot of a computer screen

Description automatically generated

3.2 Schema Builder

4. Detailed Steps to Solution Design

**4.1 Data Model**

* **Contract\_\_c**: Title, Status, Type, Risk Score, Total Clauses, Extracted Text, Uploaded By, Dates.
* **Clause\_\_c**: Title, Type, Risk Score, Extracted Text, Positions, Suggestions, Review metadata.
* **NegotiationSuggestion\_\_c**: Clause reference, Suggested Text, Tone, Goal, Risk Change, Confidence, Status.
* **Status\_Change\_Event\_\_e**: metadata about record and new status.

**4.2 Business Logic**

* **Triggers**: Detect status changes → delegate to services.
* **Services**:
  + ContractAnalysisService for clause parsing & risk metrics.
  + AINegotiationService for AI negotiation suggestions.
  + StatusNotificationService for status change emails.

**4.3 User Interface**

* **LWC (contractDashboard)**: KPIs, risk distribution chart, and recent contracts.
* **Page Layouts & Tabs**: Contracts, Clauses, and Suggestions accessible directly.

**4.4 Automation (Optional but Planned)**

* Record-triggered flows for notifying stakeholders.
* Orchestration flows for contract analysis lifecycle.

**4.5 Communication Templates**

* **Status Change Emails** (Approved/Rejected).
* **Negotiation Suggestion Messages** with rationale and risk impact.

A screenshot of a computer

Description automatically generated4.1 Apex classes and triggers powering the automation of contract analysis and AI-assisted negotiation.

**5. Testing and Validation**

* **Unit Testing**: 4 dedicated Apex test classes.
* **Trigger Validation**: Verified notifications fire correctly.
* **UI Testing**: LWC wiring with Apex confirmed.
* **Integration Testing**: Validated platform event deployments.
* A screenshot of a computer

  Description automatically generated**Coverage**: **81% Apex coverage**, exceeding Salesforce minimum (75%).

5.1Unit tests executed successfully with 81% Apex test coverage across triggers, services, and controllers.

**6. Key Scenarios Addressed**

* **Risk Assessment**: Extract clauses, compute risks, store in contract.
* **Automated Notifications**: Emails triggered on approval/rejection.
* **Negotiation Suggestions**: Alternative clause text with reasoning.
* **Real-Time Dashboard**: KPIs and activity monitoring for managers.

**7. Reports and Dashboards**

* **Reports**:
  + Contract Risk Summary by Type/Status
  + Clause Distribution by Risk Level
* **Dashboard**: *“Legal Analytics”* combining the above metrics.

A screenshot of a computer

Description automatically generated

7.1 Reports

A screenshot of a computer

Description automatically generated

7.2 Dashboard

Conclusion

This project successfully delivered an AI-powered, Salesforce-native legal workflow solution that streamlines contract analysis and negotiation. The system was designed to be scalable, secure, and user-friendly, ensuring that it can meet the needs of enterprises, legal teams, and law firms. Key achievements include the development of real-time KPIs, notifications, and platform events, all of which contribute to faster decision-making and improved visibility across legal processes. Furthermore, rigorous testing resulted in 81% code coverage, validating the reliability and robustness of the solution.

Throughout the development process, several important lessons were learned. One of the most notable takeaways was the importance of deploying Lightning Web Components (LWCs) and services before enabling flows, as this sequence ensured stability during deployment. Additionally, it was observed that flows built within Salesforce Setup provided greater reliability for CI/CD pipelines compared to flows created outside Setup. These insights are valuable for refining best practices in Salesforce development and deployment.

Looking ahead, there are multiple opportunities for enhancing the solution further. Advanced flow activation can be implemented to automate more complex approval and negotiation scenarios. Dashboards and analytics can be extended to provide deeper insights into legal contract trends, compliance risks, and team productivity. Integration with external AI models, such as advanced natural language processing or contract risk-scoring engines, can significantly enhance the intelligence of the platform. Moreover, adding batch or queueable processing capabilities will enable the system to handle large volumes of contracts efficiently, making it enterprise-ready at scale.

In conclusion, the project has laid a strong foundation for a modern legal contract management system on Salesforce, while leaving ample scope for innovation and future development.Bottom of Form

9. Appendix: Key Artifacts

* Objects: force-app/main/default/objects/\*
* Apex Classes: force-app/main/default/classes/\*
* Triggers: force-app/main/default/triggers/\*
* LWC: force-app/main/default/lwc/contractDashboard/\*
* Tests: force-app/main/default/classes/\*Test.cls
* Platform Event: force-app/main/default/objects/Status\_Change\_Event\_\_e/\*