AI Agent for Palo Alto XML Compliance Check Against ISO Controls

Project Overview

The goal of this project is to build an AI-powered compliance checking tool that analyzes Palo Alto XML configuration files and determines whether they are compliant with an ISO checklist (provided in CSV format).

The project will consist of two main parts:

1. Backend Processing – An AI agent that performs compliance checks.
2. Frontend Interface – A user-friendly web app to upload XML files and display compliance results.

Inputs

1. Palo Alto XML configuration file (will provide the sample).
2. ISO Checklist (CSV format) containing ~120 controls with the following fields:
   - Control Number
   - Control Name
   - Description/Requirement

Backend Requirements

- Build an AI agent that:

    - Reads the ISO checklist CSV.
    - Parses the XML configuration file.
    - For each ISO control, checks if the XML configuration satisfies the requirement.
    - Returns results in a structured format.
- Use open-source LLMs (e.g., Hugging Face models like LLaMA, Falcon, or Mistral) for the first version.
- The agent should output for each control:
    - Compliant / Non-compliant
    - ISO Control Number & Name
    - Reason (if non-compliant, why the XML fails the check)
    - Evidence (the relevant XML snippet, preferably converted into a screenshot format if non-compliant).

Frontend Requirements

- Build a clean and neat web interface (can use Streamlit, Flask, or ReactJS + FastAPI backend).
- Upload Feature:
    - Users should be able to upload a Palo Alto XML configuration file.
- Results Display:
    - Show compliance results in a table format with columns:
        1. Compliant/Non-compliant
        2. ISO Control Number
        3. Control Name

4. Reason (if non-compliant)
5. Evidence (XML snippet screenshot)

- Add download option (CSV/Excel) for the compliance results.

Output Example (Frontend Table)

Status	ISO Control Number	Control Name	Reason (if non-compliant)	Evidence (screenshot)
✅ Compliant	ISO-1.1	Access Control	-	-
❌ Non-compliant	ISO-2.3	Encryption Standard	Weak encryption setting found	Screenshot of XML snippet

---

Implementation Steps

1. Backend Processing

- Parse XML configuration file.
- Load ISO checklist CSV.
- Loop through each control and apply compliance check.
- If non-compliant, extract relevant XML snippet.
- Convert XML snippet into an image (screenshot).
- Save results in structured format (JSON/CSV).

2. AI Agent

- Use open-source LLMs to map XML settings with ISO control requirements.
- Ensure consistent output formatting.

3. Frontend

- Implement file upload feature.
- Trigger backend compliance checking.
- Display results in structured table.
- Provide option to download results.

Deliverables

1. Backend Code (AI agent + compliance check logic).
2. Frontend Code (upload feature + results display).
3. Final Tool: User uploads XML → Results displayed with compliance status, reasons, and evidence.
4. Documentation: Steps to run the project.