Local LLM Policy Gap Analyzer

Privacy-First Cybersecurity Policy Analysis

Against NIST CSF Standards

Python 3.8+ • Ollama • NIST CSF 2024 • 100% Offline

Introduction

Organizations struggle to maintain cybersecurity policies that align with industry standards. This tool provides **automated gap analysis** by comparing your policies against the NIST Cybersecurity Framework using a local LLM (Gemma3 via Ollama).

Every operation runs entirely on your machine—**no cloud APIs, no data collection, complete privacy.**

Table of Contents

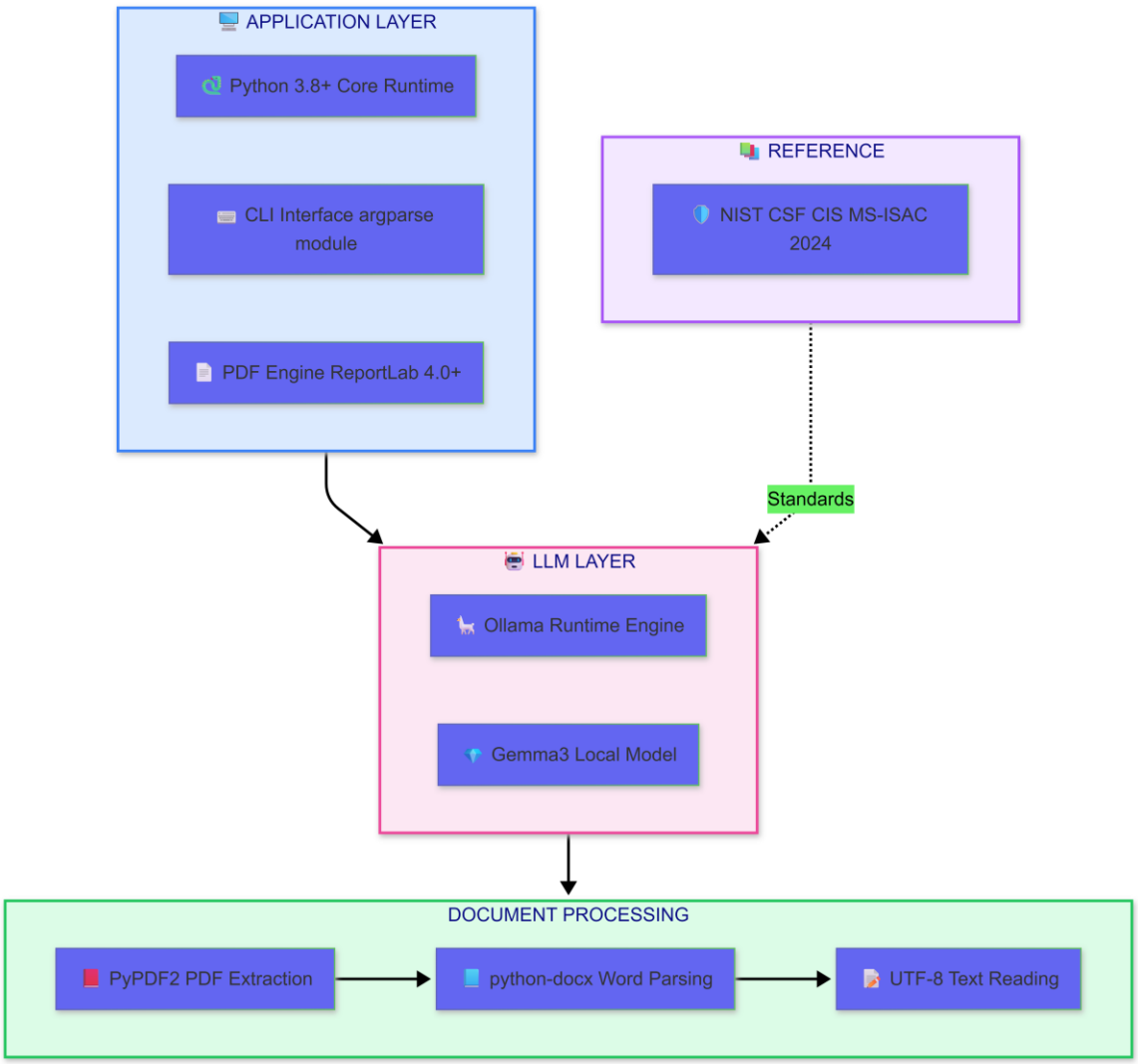
Features

Feature	Description
Gap Analysis	Identifies policy weaknesses against NIST CSF standards
Policy Revision	Auto-generates improved policy versions addressing gaps
Implementation Roadmap	Phased improvement plans (0-3, 3-6, 6-12 months)
Executive Summary	Leadership-ready overview of findings
Multi-Format Input	Supports .txt, .pdf, and .docx policies
PDF Output	Professional formatted reports using ReportLab
Batch Processing	Analyze multiple policies in one run
100% Offline	Zero network calls after initial setup

🛠️ Tech Stack & Prerequisites

Technology Stack

📊 Technology Stack Architecture



System Requirements

Component	Minimum	Recommended
💻 **CPU**	Intel i5 / AMD Ryzen 5	Intel i7 / AMD Ryzen 7
🧠 **RAM**	8 GB	16 GB
💾 **Storage**	10 GB	20 GB
🖥️ **OS**	Windows 10 / Linux / macOS	Windows 11 / Ubuntu 22.04

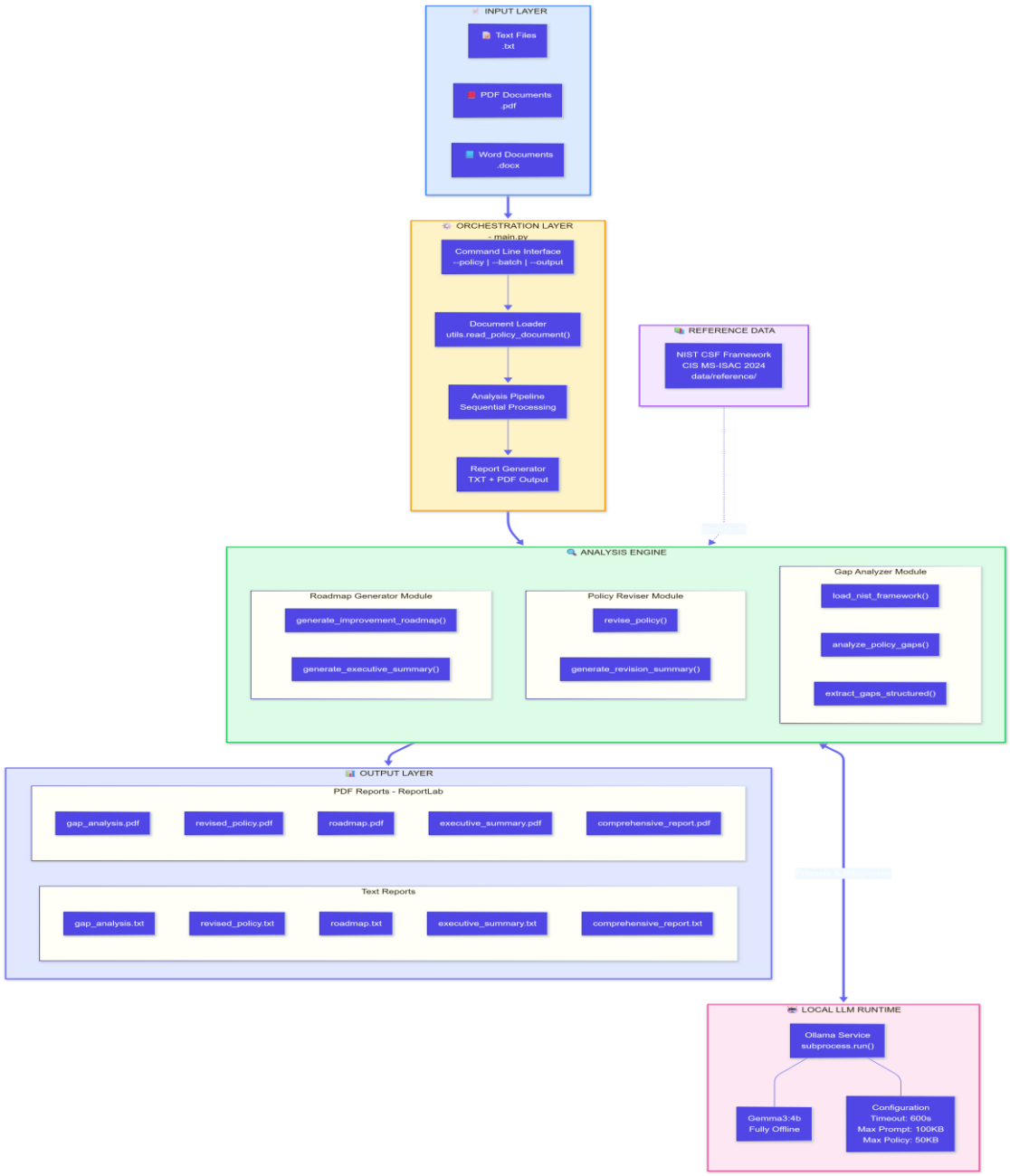
Dependencies

Package	Version	Purpose
PyPDF2	>= 3.0	PDF text extraction
python-docx	>= 0.8	Word document parsing
reportlab	>= 4.0	PDF report generation
ollama	(runtime)	Local LLM execution

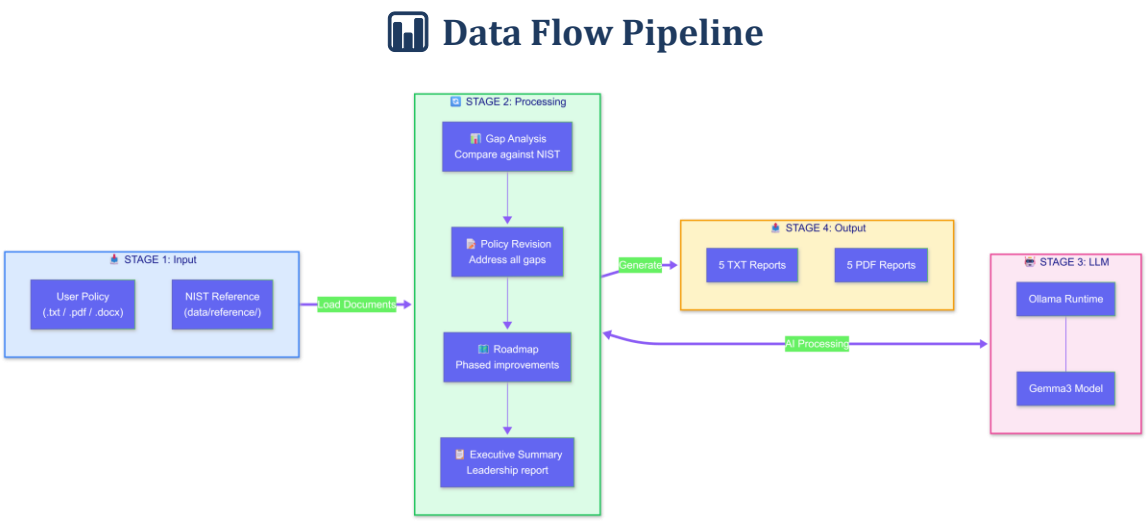
Architecture

System Overview

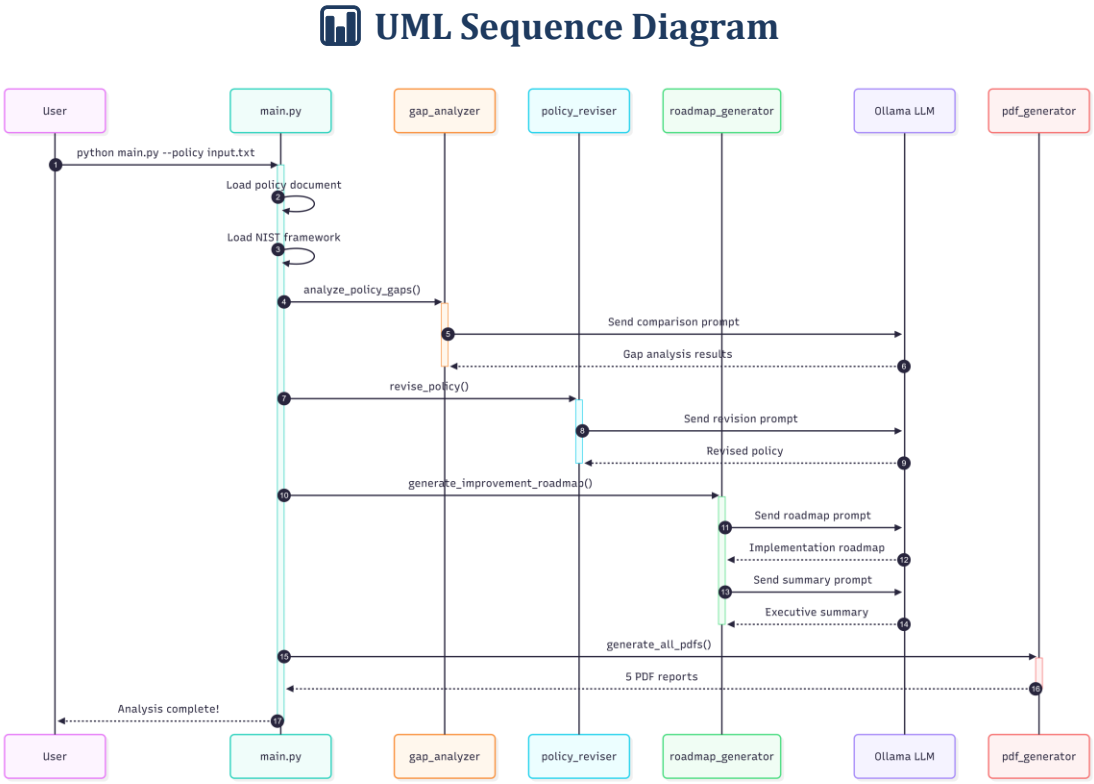
System Architecture Flowchart



Data Flow

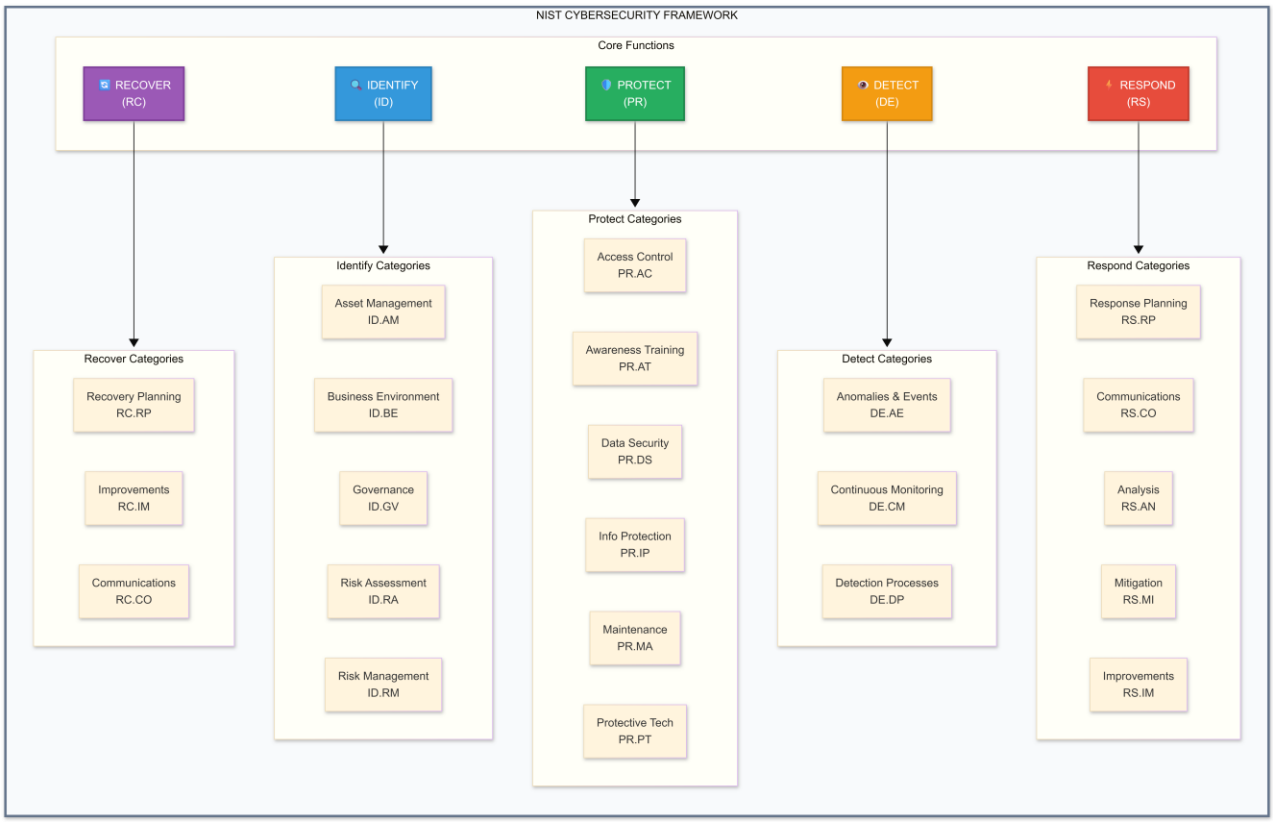


Processing Sequence



NIST CSF Coverage

NIST Framework Map



Project Structure

```
Local-LLM/
├── src/
│   ├── main.py              # CLI & orchestrator
│   ├── gap_analyzer.py      # NIST comparison & LLM
│   ├── policy_reviser.py    # Policy improvements
│   ├── roadmap_generator.py # Roadmap creation
│   ├── pdf_generator.py     # PDF formatting
│   └── utils.py             # File utilities
├── data/
│   ├── reference/           # NIST CSF files
│   └── test_policies/       # Sample policies
├── output/                  # Generated reports
└── requirements.txt         # Dependencies
```

Module Overview

Module	Size	Responsibility
main.py	216 lines	CLI parsing, workflow orchestration
gap_analyzer.py	131 lines	NIST loading, policy comparison
policy_reviser.py	65 lines	Policy improvement generation
roadmap_generator.py	112 lines	Phased roadmap creation
pdf_generator.py	167 lines	PDF formatting with ReportLab
utils.py	78 lines	Document reading utilities

🚀 Quick Start

Step 1: Clone & Setup

```
git clone https://github.com/HACK-IITK-2025-C3iHub/Local-LLM.git
cd "Local LLM"
python -m venv venv
venv\Scripts\activate
pip install -r requirements.txt
```

Step 2: Install Ollama

Step	Command	Notes
1️⃣ **Install**	Download from ollama.ai	One-time
2️⃣ **Pull Model**	ollama run gemma3:4b	Needs internet
3️⃣ **Verify**	ollama list	Should show gemma3

Step 3: Run Analysis

```
# Single policy
python src/main.py --policy data/test_policies/isms_policy.txt

# Batch processing
python src/main.py --batch data/test_policies/
```

Output Reports

Report	Format	Description
📊 **Gap Analysis**	TXT + PDF	Identified weaknesses
📝 **Revised Policy**	TXT + PDF	Improved version
🗺️ **Roadmap**	TXT + PDF	Implementation plan
📋 **Executive Summary**	TXT + PDF	Leadership overview
📚 **Comprehensive**	TXT + PDF	All combined

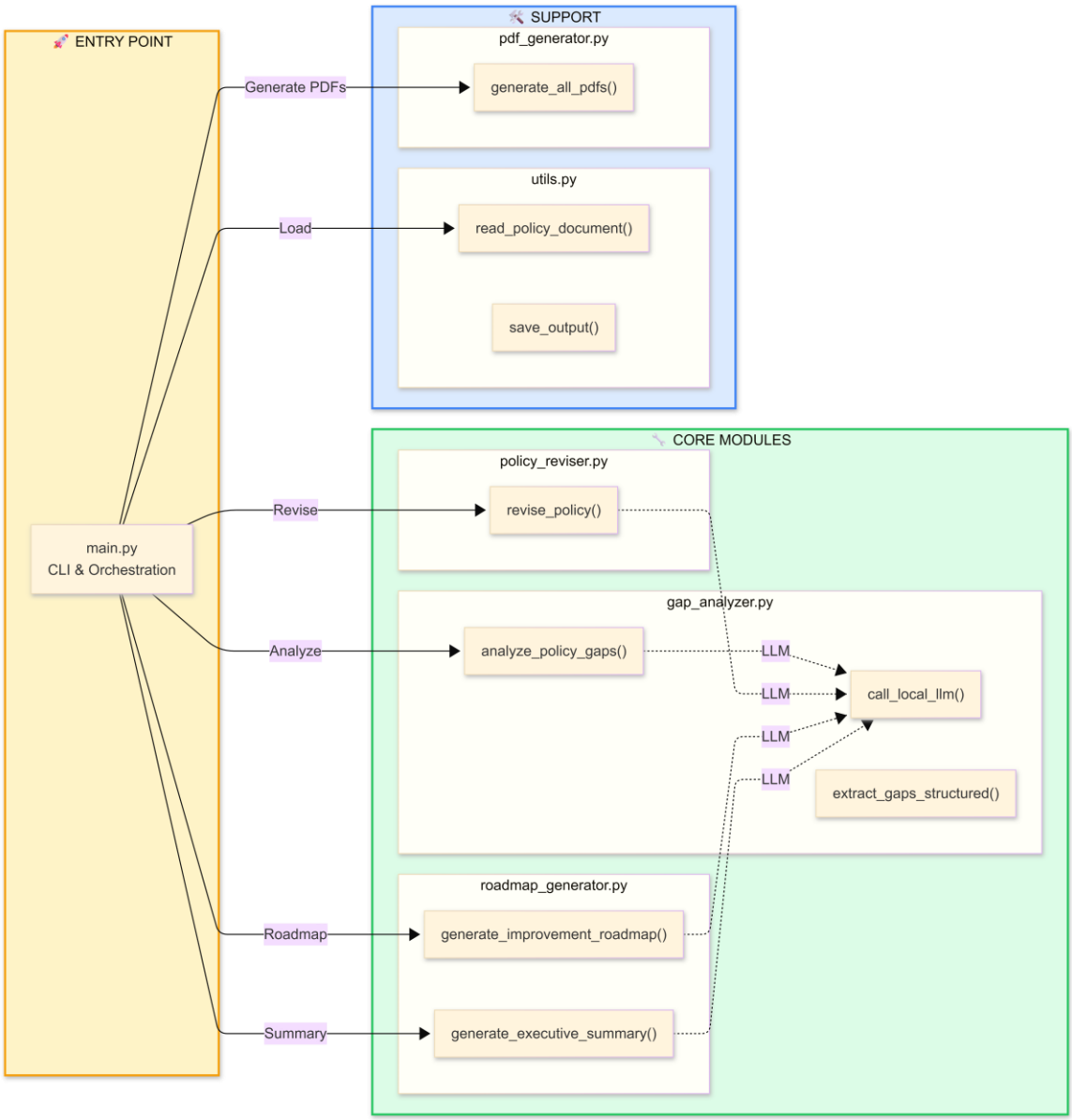
Processing Time

Stage	Duration
Gap Analysis	1-2 min
Policy Revision	2-3 min
Roadmap Generation	1-2 min
Executive Summary	30-60 sec
TOTAL	~5-8 min

Developer Guide

Code Architecture

Code Structure Flowchart



Security Limits

Constant	Value
LLM_TIMEOUT	600 seconds
MAX_PROMPT_SIZE	100 KB
MAX_POLICY_SIZE	50 KB
MAX_FILE_SIZE	50 MB

⚠ Known Issues & Limitations

Issue	Description	Workaround
LLM Accuracy	Output depends on model	Review manually
Processing Time	5-8 min per policy	Batch overnight
RAM Usage	~6GB for Gemma3	Close other apps
Language	English only	Translation planned

Made With ♥ by T-reXploit

Privacy-First • Offline-Ready • Open Source

github.com/HACK-IITK-2025-C3iHub/Local-LLM