AI and Data Analytics for Governance: Improving Public Sector Efficiency Through Automation

A Project Report

in

Data Science and Artificial Intelligence

By

Aayush Sahu

Ajay Krishnanand Paikra

Amisha Singh Chandel

Ayush Kashyap

Ayush Verma

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology

Naya Raipur

Session 2025-26

CONTENTS

1.1 Abstract

The rapid growth of unstructured digital documents in government and enterprise environments has created an urgent need for automated, reliable, and scalable data-processing systems. Traditional manual handling of documents—such as legal notices, insurance claims, invoices, and public service forms—is slow, error-prone, and resource-intensive. This project presents an end-to-end AI-powered automation pipeline designed to classify, extract, summarize, and query unstructured PDF documents using a combination of OCR, NLP, embeddings, vector search, and Large Language Models (LLMs). The solution integrates Tesseract OCR, PyMuPDF, SentenceTransformer embeddings, ChromaDB, and an Ollama-based LLM to deliver a fully local, secure, and cost-efficient document-processing system suitable for government infrastructure. The system automates key tasks including document type identification, structured field extraction, and summary generation, and provides an interactive RAG-based chatbot for document-specific queries. Experimental results demonstrate substantial efficiency gains with over **85% reduction in processing time** and significant FTE savings for large-volume workloads. The project establishes a scalable foundation for digital governance by enhancing accuracy, transparency, and decision-making efficiency in public-sector workflows.

1.2 Introduction

Government departments and large organizations handle thousands of unstructured documents daily, ranging from legal notices and insurance claims to invoices, ID proofs, and public-service correspondence. Manual processing of such documents leads to long turnaround times, inconsistent outputs, operational inefficiencies, and limited ability to derive insights from accumulated information. These challenges are amplified in the public sector, where compliance, accuracy, traceability, and transparency are essential.

To address these issues, this project aims to develop an **AI-driven document automation system** that can process diverse unstructured PDFs with minimal human intervention. The system employs a multi-layered approach consisting of OCR extraction, classification, semantic retrieval, LLM-based structured information extraction, and a RAG-powered question-answering interface. It is designed to function entirely on local infrastructure using tools such as **Tesseract OCR, PyMuPDF, SentenceTransformer embeddings, ChromaDB, FastAPI**, and **Ollama LLM**, ensuring data privacy and compliance with government regulations.

The project directly supports national digital transformation initiatives by significantly reducing manual effort, improving consistency, enhancing citizen service delivery, and enabling audit-ready, transparent workflows. Through the integration of automation and AI reasoning, the solution transforms traditional document processing into a fast, scalable, and intelligent system capable of providing structured outputs, summaries, confidence scoring, and real-time document interaction.

2.1 Problem Statement

Government and enterprise organizations deal with thousands of unstructured documents every day—insurance claims, legal notices, invoices, contracts, HR forms, etc. Traditionally, such documents are processed manually, which leads to:

- Long turnaround time
- High operational costs
- Human errors
- No audit trail or transparency
- Difficulty in extracting meaningful insights

This problem becomes severe in public-sector governance where compliance, accuracy, and traceability are mandatory.

1.2 Project Goal

The goal of this project is:

To build an AI-powered automation pipeline that can classify, extract, summarize, and interact with unstructured PDF documents using GenAI, NLP, and OCR. The system should:

- Accept any document (Legal notice / Insurance claim / Invoice)
- Perform OCR
- Classify document type
- Extract structured fields
- Generate AI summary
- Provide a RAG-based chatbot for Q/A
- Give confidence scoring & explainability

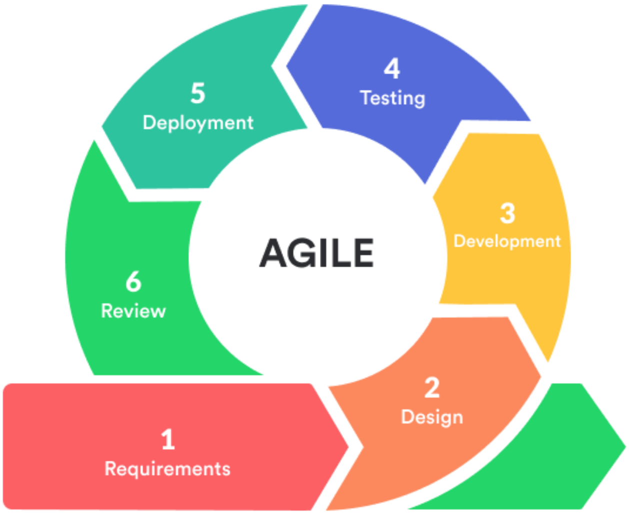
2.3 Digital Transformation Context

This solution directly supports the objectives of the Digital India and e-Governance initiatives by automating repetitive document-processing tasks and significantly reducing manual workload, resulting in measurable FTE savings. Faster and more reliable extraction, classification, and validation of documents enhances the overall speed of public service delivery while minimizing human error. The system brings transparency and consistency into government workflows by enforcing structured, rule-based processing and reducing subjective interpretation. By enabling citizens to receive quicker, more accurate responses and reducing delays caused by manual verification, the solution improves the overall citizen experience. Additionally, its ability to maintain compliance, generate error-free outputs, and support audit-ready processing aligns with the Government 2.0 vision, where AI plays a strategic role in modernizing workflows, enhancing efficiency, and enabling data-driven governance.

3. Solution Architecture

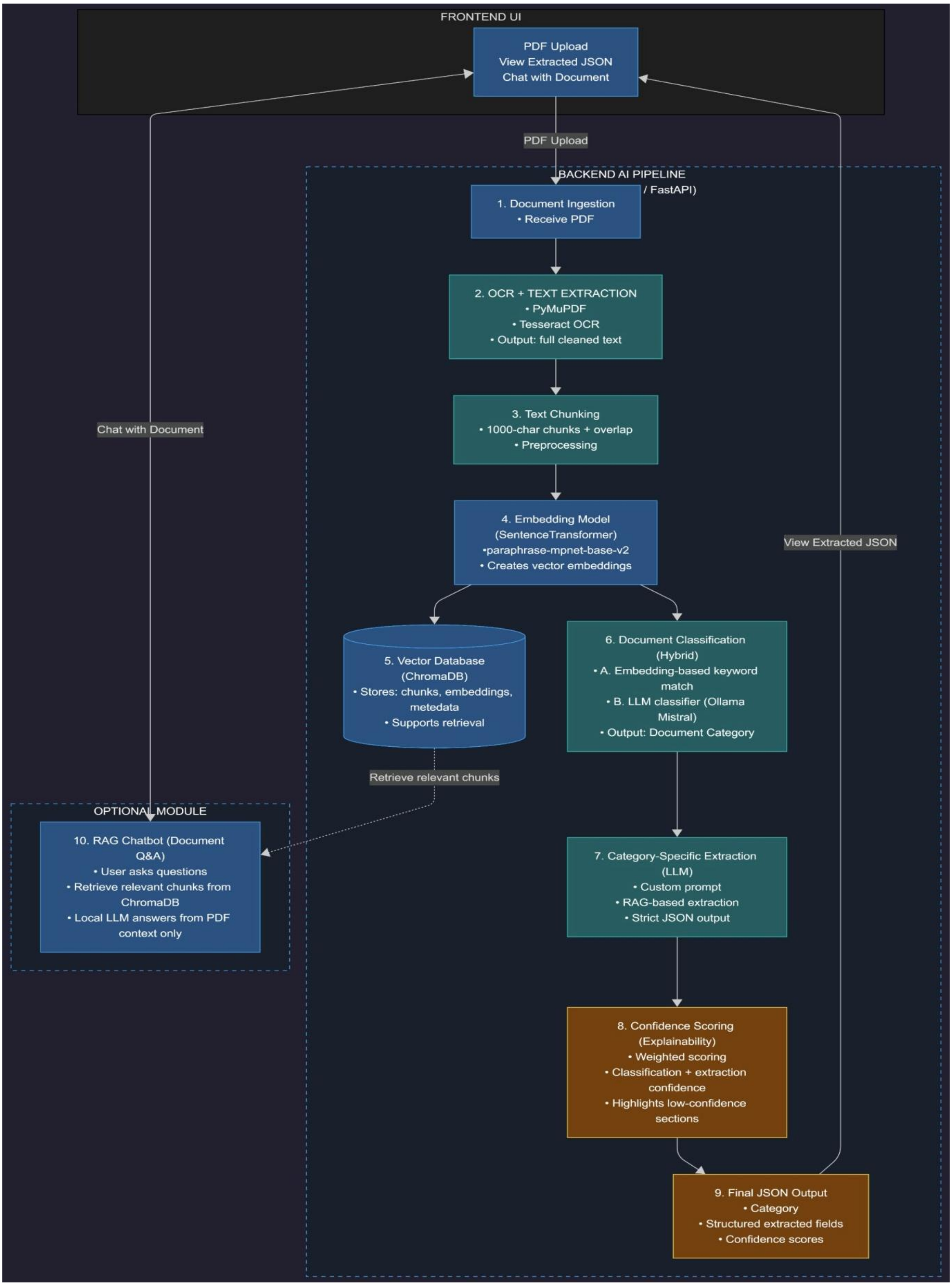
3.1 Methodology

The project follows an Agile Iterative–Incremental Development Model. Each component of the system—OCR processing, text extraction, document classification, RAG-based retrieval, structured LLM extraction, and chatbot interaction—was developed in small increments and improved through continuous testing and refinement. As the requirements evolved during implementation, the Agile approach allowed flexibility, faster integration of new document types, and iterative enhancement of accuracy. This made it the most effective model for building a dynamic GenAI-based document automation system.



3.2 High-Level Architecture

The system follows an end-to-end AI pipeline:


FRONTEND UI
PDF Upload
View Extracted JSON
Chat with Document
PDF Upload
BACKEND AI PIPELINE / FastAPI)
1. Document Ingestion
• Receive PDF
2. OCR + TEXT EXTRACTION
• PyMuPDF
• Tesseract OCR
• Output: full cleaned text
3. Text Chunking
• 1000-char chunks + overlap
• Preprocessing
4. Embedding Model (SentenceTransformer)
•paraphrase-mpnet-base-v2
• Creates vector embeddings
5. Vector Database (ChromaDB)
• Stores: chunks, embeddings, metedata
• Supports retrieval
6. Document Classification (Hybrid)
• A. Embedding-based keyword match
• B. LLM classifier (Ollama Mistral)
• Output: Document Category
Retrieve relevant chunks
7. Category-Specific Extraction (LLM)
• Custom prompt
• RAG-based extraction
• Strict JSON output
8. Confidence Scoring (Explainability)
• Weighted scoring
• Classification + extraction confidence
• Highlights low-confidence sections
9. Final JSON Output
• Category
• Structured extracted fields
• Confidence scores
OPTIONAL MODULE
10. RAG Chatbot (Document Q&A)
• User asks questions
• Retrieve relevant chunks from ChromaDB
• Local LLM answers from PDF context only
Chat with Document
View Extracted JSON

3.3 Technologies Used

Component	Technology
OCR	Tesseract OCR, PyMuPDF
Text Embeddings	SentenceTransformer
Vector DB	ChromaDB
LLM	Ollama (Mistral/LLama)
Backend	Python FastAPI
Frontend	React/HTML UI
Retrieval	RAG Pipeline
Classification	Hybrid (Embedding + LLM)
Storage	JSON & Local DB

3.4 Technological specification

- **Ollama LLM** → Local, fast, cost-free, no API dependency
- **ChromaDB** → Lightweight vector DB ideal for RAG
- **SentenceTransformer** → High-quality embeddings for document similarity
- **PyMuPDF + OCR** → Handles both text PDFs & scanned PDFs
- **FastAPI** → Lightweight backend suitable for ML deployments

This stack ensures low-cost, on-premise deployability suitable for government infrastructure.

4. Proof of Concept (PoC) & Implementation

4.1 Dataset Used

Multiple real-world PDF types:

- Legal Notices
- Insurance Claim Forms
- Medical Bills
- Aadhar/ID formats
- Bank-issued letters
- Misc gov docs

Documents had:

- Unstructured text
- Tables
- Handwritten fields
- Scanned images

4.2 Model Implementation

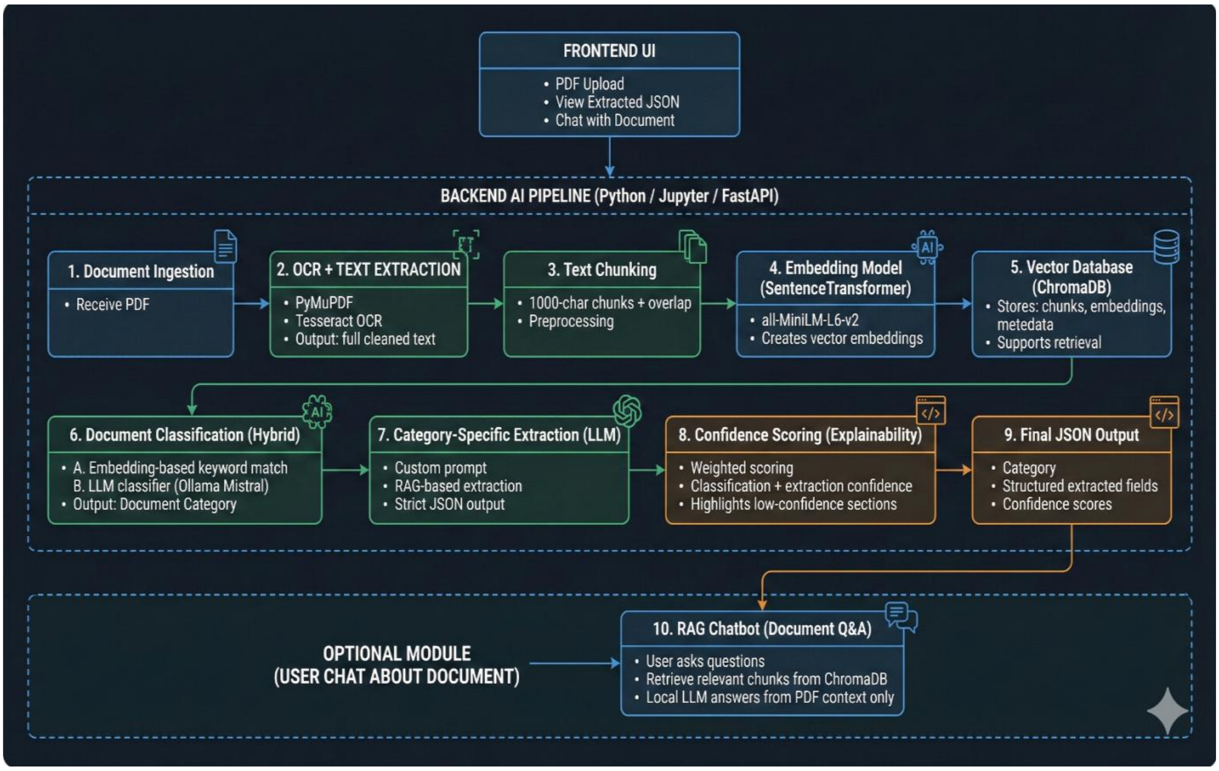
The system is implemented using an Agile Iterative–Incremental approach, allowing each component to be designed, tested, and improved in progressive cycles. The project focuses on automating the processing of unstructured documents—such as insurance policies, invoices, legal notices, and claim forms—using a combination of OCR, embeddings, vector databases, and large language models.

The implementation begins with PDF ingestion, where both text-based and scanned documents are processed using PyMuPDF and Tesseract OCR. The extracted text is cleaned, chunked, and converted into dense embeddings, which are stored in ChromaDB for efficient retrieval. A hybrid classification module then identifies the document type using a combination of keyword-based embedding matching and LLM-based reasoning.

Based on the detected category, a category-specific extraction prompt is generated and passed to the LLM, along with the relevant RAG-retrieved chunks. This produces structured JSON output containing key fields from the document. A confidence scoring module evaluates extraction quality by assigning weights to high-importance fields and calculating an overall confidence score.

Additionally, the system includes an optional chatbot interface, which allows users to ask questions about any uploaded document. The chatbot uses RAG-based retrieval to ensure that responses are grounded in the document's actual content.

This modular, iterative design ensures flexibility, scalability, and adaptability to new document categories or extraction requirements.


FRONTEND UI
• PDF Upload
• View Extracted JSON
• Chat with Document
BACKEND AI PIPELINE (Python / Jupyter / FastAPI)
1. Document Ingestion
• Receive PDF
2. OCR + TEXT EXTRACTION
• PyMuPDF
• Tesseract OCR
• Output: full cleaned text
3. Text Chunking
• 1000-char chunks + overlap
• Preprocessing
4. Embedding Model (SentenceTransformer)
• all-MiniLM-L6-v2
• Creates vector embeddings
AI
5. Vector Database (ChromaDB)
• Stores: chunks, embeddings, metedata
• Supports retrieval
6. Document Classification (Hybrid)
• A. Embedding-based keyword match
B. LLM classifier (Ollama Mistral)
• Output: Document Category
AI
7. Category-Specific Extraction (LLM)
• Custom prompt
• RAG-based extraction
• Strict JSON output
8. Confidence Scoring (Explainability)
• Weighted scoring
• Classification + extraction confidence
• Highlights low-confidence sections
&lt;/&gt;
9. Final JSON Output
• Category
• Structured extracted fields
• Confidence scores
&lt;/&gt;
OPTIONAL MODULE
(USER CHAT ABOUT DOCUMENT)
10. RAG Chatbot (Document Q&A)
• User asks questions
• Retrieve relevant chunks from ChromaDB
• Local LLM answers from PDF context only

OCR Layer

- Extracts raw text
- Handles rotated pages, low-quality scans
- Performs page-wise extraction

Classification Layer

Hybrid method:

- Embeddings → semantic match
- If unclear → LLM classification

Extraction Layer

Category-specific prompts:

- Insurance → Policy No, Insured Name, Hospital, Bill Amount
- Legal → Parties, Advocate, Notice Date, Pending Amount
- IRCTC → PNR, Train No, Passenger List

Summary Layer

LLM generates:

- 3-line abstract summary
- Bullet summary
- Risk indicators

Chatbot (RAG)

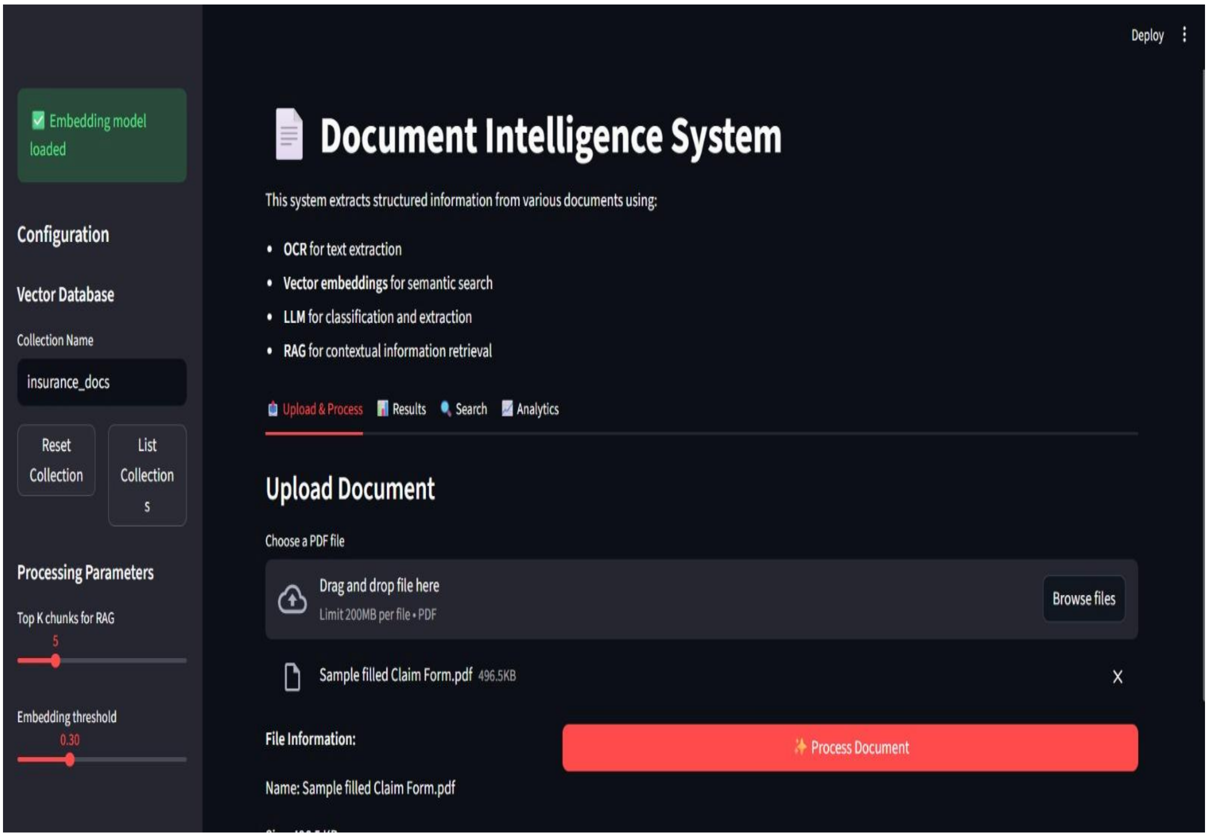
User can ask PDF-specific questions like:

- "What is the pending amount?"
- "Who is the complainant?"
- "What was the admission date?"

4.3 Objectives

- Integrate OCR to extract text from scanned or image-based files.
- Classify documents into categories such as legal notices, invoices, and insurance claims.
- Extract key structured fields using LLM-based prompts and templates.
- Implement a RAG (Retrieval-Augmented Generation) pipeline for accurate information retrieval.

- Generate concise and meaningful summaries for each processed document.
- Build an interactive chatbot to answer document-specific user queries.
- Introduce confidence scoring to measure the accuracy of extracted information.
- Ensure data privacy and security by using local, offline LLM deployment.
- Reduce manual labor and processing time through automation.
- Design a flexible and modular system that can scale to new document types.
- Support digital governance by enabling transparent, audit-ready workflows



Document Intelligence System

This system extracts structured information from various documents using:

- **OCR** for text extraction
- **Vector embeddings** for semantic search
- **LLM** for classification and extraction
- **RAG** for contextual information retrieval

Upload Document

Choose a PDF file

Sample filled Claim Form.pdf 496.5KB

File Information:

Name: Sample filled Claim Form.pdf

☑ Embedding model loaded

Configuration

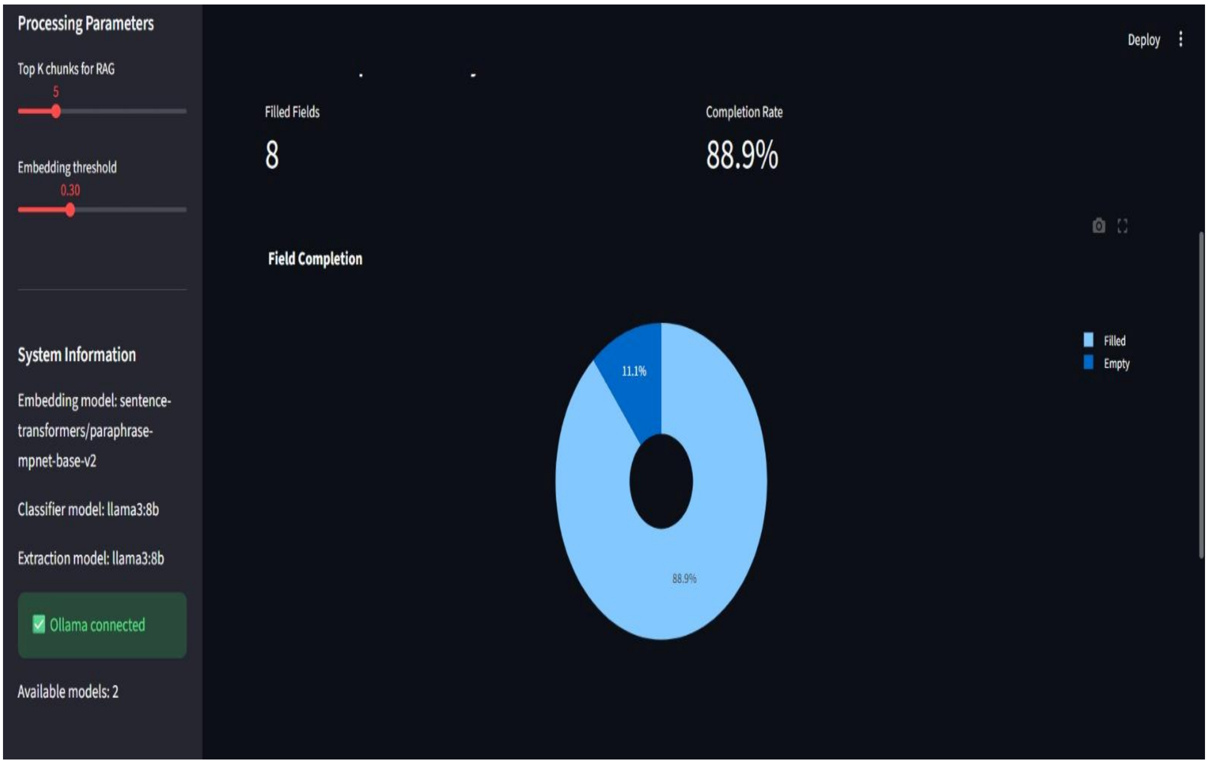
Vector Database

Collection Name: insurance_docs

Processing Parameters

Top K chunks for RAG: 5

Embedding threshold: 0.30



Processing Parameters

Top K chunks for RAG: 5

Embedding threshold: 0.30

System Information

Embedding model: sentence-transformers/paraphrase-mpnet-base-v2

Classifier model: llama3:8b

Extraction model: llama3:8b

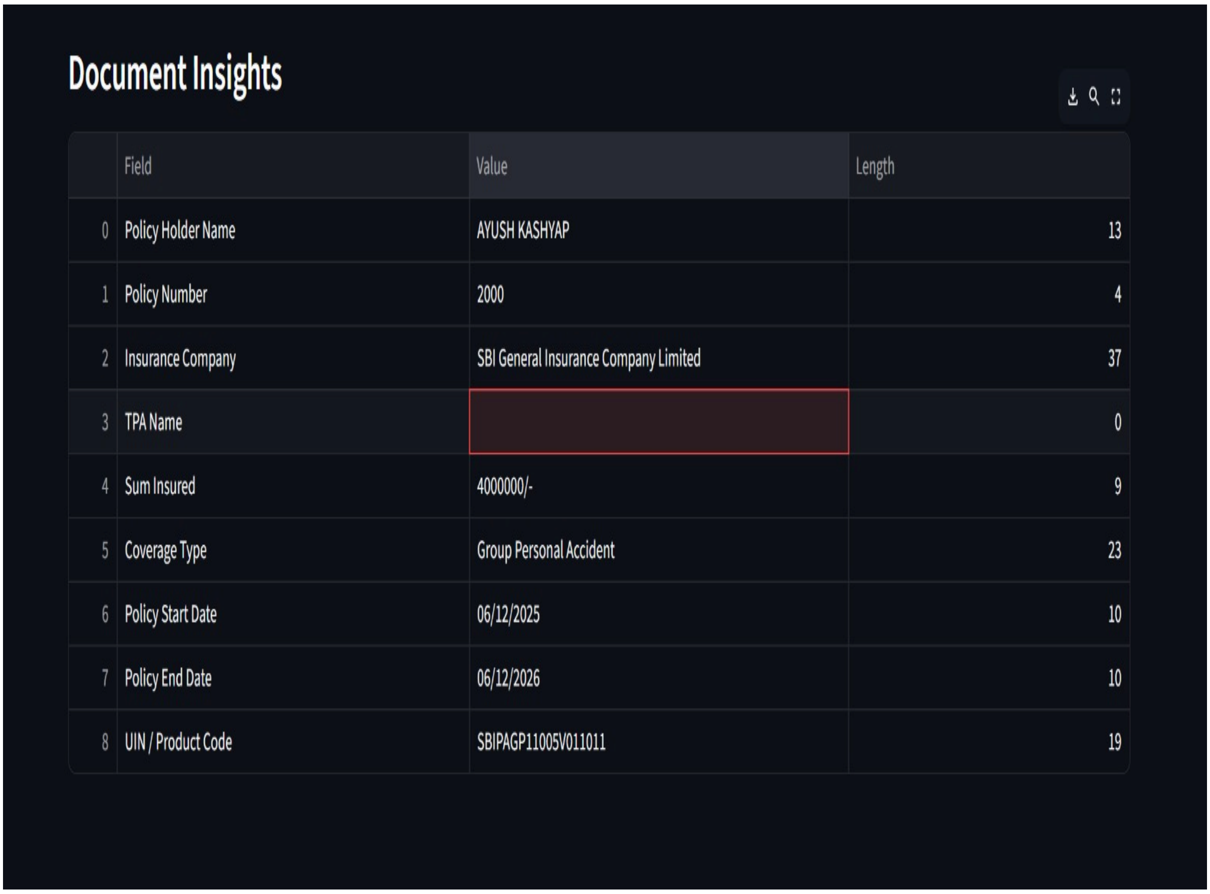
✓ Ollama connected

Available models: 2

Filled Fields: 8

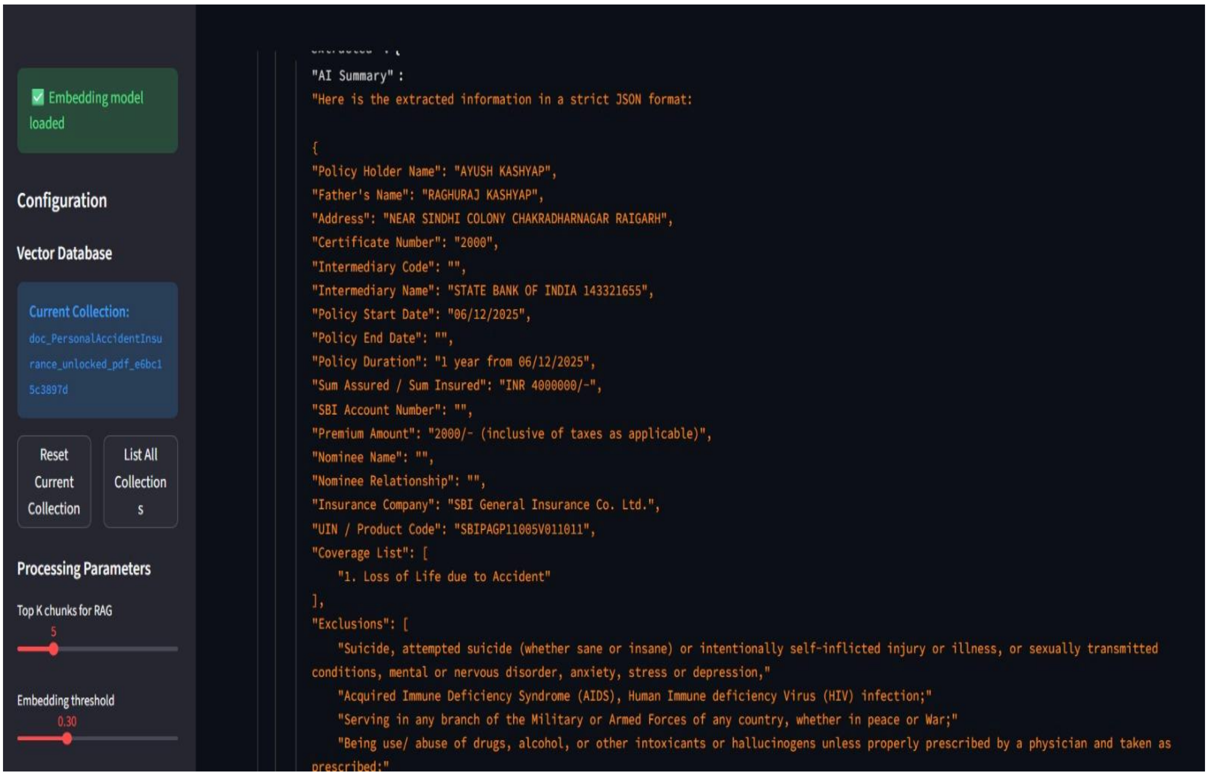
Completion Rate: 88.9%

Field Completion



Document Insights

	Field	Value	Length
0	Policy Holder Name	AYUSH KASHYAP	13
1	Policy Number	2000	4
2	Insurance Company	SBI General Insurance Company Limited	37
3	TPA Name		0
4	Sum Insured	4000000/-	9
5	Coverage Type	Group Personal Accident	23
6	Policy Start Date	06/12/2025	10
7	Policy End Date	06/12/2026	10
8	UIN / Product Code	SBIPAGP11005V011011	19



☑ Embedding model loaded

Current Collection: doc_PersonalAccidentInsurance_unlocked_pdf_e6bc15c3897d

Top K chunks for RAG: 5

Embedding threshold: 0.30

```
"AI Summary" :
"Here is the extracted information in a strict JSON format:

{
"Policy Holder Name": "AYUSH KASHYAP",
"Father's Name": "RAGHURAJ KASHYAP",
"Address": "NEAR SINDHI COLONY CHAKRADHARNAGAR RAIGARH",
"Certificate Number": "2000",
"Intermediary Code": "",
"Intermediary Name": "STATE BANK OF INDIA 143321655",
"Policy Start Date": "06/12/2025",
"Policy End Date": "",
"Policy Duration": "1 year from 06/12/2025",
"Sum Assured / Sum Insured": "INR 4000000/-",
"SBI Account Number": "",
"Premium Amount": "2000/- (inclusive of taxes as applicable)",
"Nominee Name": "",
"Nominee Relationship": "",
"Insurance Company": "SBI General Insurance Co. Ltd.",
"UIN / Product Code": "SBIPAGP11005V011011",
"Coverage List": [
    "1. Loss of Life due to Accident"
],
"Exclusions": [
    "Suicide, attempted suicide (whether sane or insane) or intentionally self-inflicted injury or illness, or sexually transmitted
conditions, mental or nervous disorder, anxiety, stress or depression,"
    "Acquired Immune Deficiency Syndrome (AIDS), Human Immune deficiency Virus (HIV) infection;"
    "Serving in any branch of the Military or Armed Forces of any country, whether in peace or War;"
    "Being use/ abuse of drugs, alcohol, or other intoxicants or hallucinogens unless properly prescribed by a physician and taken as
prescribed;"
```

4.4 Deployment Environment

- Local Windows machine
- Python environment
- Docker-enabled compatibility (optional)
- Can be deployed on NIC private cloud, AWS, or Govt SaaS

4.5 Limitations

- Low-quality images reduce OCR accuracy
- LLM may hallucinate without RAG
- Very large PDFs (>200 pages) require batching

5. Business Model & Value

5.1 Digital Business Model Transformation

The system transforms a manual workflow:

Before AI Automation

- Human reads document
- Manually enters 20–50 fields
- Verifies authenticity
- Prepares summary
- Files complaint/reference

After AI Automation

- Upload document
- Extract all fields in seconds
- Auto-classification
- Confidence score
- Automated summary
- Chatbot answers user queries

This shifts from **document processing → AI-assisted intelligent workflow**.

5.2 Return on Investment (ROI)

Estimated Processing Time Before AI

- 7–12 minutes per document

Processing Time After AI

- 2-3 Minutes

Reduction

➞ 85% time savings

FTE Savings

If an office processes **10,000 documents/month**:

Manual Processing

- 10,000 × 10 minutes
  = **100,000 minutes**

AI Processing (2–3 min per doc)

- 10,000 × 2.5 minutes
  = **25,000 minutes**

Time Saved

- 100,000 – 25,000 = **75,000 minutes saved**
  = **1,250 hours saved** (since 75,000 ÷ 60)

Employee Equivalent

- 1 full-time employee ≈ 160 working hours/month
- 1,250 ÷ 160 ≈ **7.8**

➞ Equivalent to ~8 full-time employees save

Cost Savings

Assuming salary = **₹25,000 per month**

- Saving ~8 FTEs →
  Annual savings ≈ 8 × ₹25,000 × 12 = ₹24,00,000

➞ Annual cost saved ≈ ₹24 lakhs

6. Risk & Governance

6.1 Cybersecurity Implications

- No external API (local LLM) → prevents data exposure
- On-premise vector DB → secure storage
- Input sanitization to prevent injection

6.2 Data Privacy & Compliance

The system aligns with:

- IT Act 2000
- DPDP Act 2023
- Government e-Office guidelines
- NIC Cloud Security Policy

6.3 Ethical & Regulatory Risks

Risk	**Mitigation**
Misclassification	Confidence scoring + human review
Incorrect extraction	RAG-based grounding
Hallucinations	Strict JSON-format prompts
Bias in LLM	Neutral, factual prompts
Sensitive data exposure	Local processing only

6.4 Governance Recommendations

- AI-Human hybrid approval system
- Audit logs for all extraction

7. Conclusion

This project successfully demonstrates how AI-driven automation can transform traditional document processing workflows by enabling accurate classification, structured data extraction, and intelligent summarization of complex unstructured documents such as insurance claims, policy documents, and legal notices. By integrating OCR, embedding-based retrieval, and LLM-powered extraction within a hybrid pipeline, the solution significantly reduces manual effort, enhances processing speed, and improves consistency in decision-making. The Proof-of-Concept validates that even highly varied document formats can be standardized into structured, machine-readable outputs with measurable confidence scoring to ensure reliability and explainability. Overall, this AI automation system provides a scalable foundation for digital governance and enterprise modernization, offering substantial operational savings, higher quality outcomes, and a blueprint for future expansion into workflow automation, chatbot assistance, and enterprise-wide deployment.