

Consulting Project

2026-07-20

MedClaim Sentinel: Local Multimodal AI for Medical Insurance Claim OCR, Duplicate Detection, Clinical Consistency, and Review Chat



A local-first medical insurance claim review platform that reads multilingual claim documents, detects repeated claims and invoices, reconciles amounts, performs conservative diagnosis-medicine-bill-investigation matching, and supports evidence-grounded reviewer chat.

Project Snapshot

DATE

2026-07-20

CATEGORY

Consulting Project

SHORT DESCRIPTION

A local-first medical insurance claim review platform that reads multilingual claim documents, detects repeated claims and invoices, reconciles amounts, performs conservative diagnosis-medicine-bill-investigation matching, and supports evidence-grounded reviewer chat.

TAGS / TECH STACK

Insurance AI

Healthcare AI

Document Intelligence

OCR

Multimodal AI

Duplicate Detection

Clinical Decision Support

Human-in-the-Loop

Fraud Analytics

Project Links

[GitHub Link](#)

[Demo Video](#)

[Demo Video](#)

[Project Link](#)

[Project Link](#)

Full Project Content

MedClaim Sentinel is a local-first medical insurance claim review prototype for multilingual OCR, duplicate-application detection, medicine and investigation consistency checks, financial reconciliation, explainable reviewer prompts, and claim-aware chat.

The system is designed as **clinical and claims decision support**. It does not automatically approve or reject a claim. Missing evidence, possible mismatches, and uncertainty are routed to an authorized claim or medical officer.

[View the GitHub Repository](#)

GitHub Repository: [dranubhaparashar/medclaim-sentinel](#)

Demo Video

Embedded media: [Open video](#)

[Open the MedClaim Sentinel demo on YouTube](#)

One-Line Idea

Turn scanned medical reimbursement forms, prescriptions, pharmacy bills, laboratory schedules, certificates, and supporting records into a structured and explainable claim-review workspace that can identify repeated submissions and highlight whether the available diagnosis, medicines, bills, and investigations are mutually supported.

Why This Project Exists

Medical reimbursement claims often arrive as a mixed bundle of:

- printed claim forms;
- handwritten prescriptions;
- medicine invoices and summary schedules;
- laboratory and imaging bills;
- hospital certificates;
- discharge or treatment records;
- Hindi and English text on the same page;
- stamps, signatures, tables, overwriting, and low-quality mobile photographs.

A reviewer may need to answer several questions before a claim can move forward:

- Has the same person submitted two, three, or four applications for the same treatment episode?
- Has the same invoice appeared in more than one claim?
- Is a typed bill summary duplicating a handwritten attachment rather than representing a new expense?
- Do the documented diagnosis and treatment context reasonably support the prescribed medicine?
- Do the billed medicines correspond to the prescription?
- Are the investigations related to the documented diagnosis or only conditionally relevant?
- Do medicine, laboratory, and claim totals reconcile?
- Which field came from which document and page?
- What evidence is missing before a medical officer can make a decision?

Claims reviewers must reconcile diagnoses, medicines, invoices, duplicates, and billed amounts across fragmented documents, making manual review slow and inconsistent. MedClaim Sentinel organizes the evidence, preserves the original documents, and produces review prompts with traceable reasons.

Project at a Glance

Area	Description
Project type	Consulting prototype for insurance claim document intelligence and decision support
Domain	Medical reimbursement and health-insurance claim review
Primary users	Claim officers, medical officers, OCR reviewers, fraud investigators, supervisors, and auditors

Area	Description
Input	Images of claim forms, prescriptions, pharmacy bills, test schedules, certificates, and supporting documents
OCR mode	Printed and handwritten Hindi-English document extraction with image preprocessing
Local AI	Ollama vision-language model, configurable between Qwen2.5-VL 3B and 7B
Core clinical result	Diagnosis ↔ prescribed medicine ↔ billed medicine ↔ investigation
Duplicate analysis	Document-level and claim-level similarity with explainable reasons
Financial output	Claimed, supported, duplicated, and recommended-review amounts
Interface	Streamlit dashboard with Claim 360° review workspace
Storage	Local files and SQLite for the current MVP
Governance boundary	Human review required; no automatic medical or claim rejection
Current maturity	Local consulting MVP and reference architecture

Core Business Outcomes

1. Detect multiple claim applications

The platform compares new claims against existing records using:

- patient and member information;
- treatment period;
- provider or hospital;
- bill and invoice numbers;
- claim amounts;
- medicine and test lines;
- image hashes and visual similarity;
- overlapping dates and treatment context.

The output is not only a score. It includes reasons such as:

Possible duplicate

- Same patient and treatment period
- Same medicine-bill total
- Matching invoice number
- Near-identical document image
- Supporting schedule appears in a previous claim

2. Reconcile claim amounts

The financial layer separates and compares:

Medicine total

- + Laboratory / investigation total
- + Hospital or treatment total
- = Supported claim amount

It also flags:

- repeated invoice amounts;
- arithmetic differences;
- duplicate supporting schedules;
- claimed amount versus extracted amount mismatches;
- unsupported amounts;
- missing itemized detail.

3. Assess clinical consistency conservatively

The platform distinguishes between:

- evidence that is directly matched;
- evidence that is conditionally relevant;
- a possible mismatch;
- insufficient evidence;
- a case requiring medical-officer review.

It does not infer a medicine name from a bill number or amount, and it does not treat the presence of a prescription document as proof that the medicine is clinically appropriate.

4. Give reviewers a complete pipeline view

The dashboard shows:

- claims received;
- claimed and supported amounts;

- possible duplicates;
- medical-review queue;
- OCR status;
- risk and reason codes;
- claim-level evidence;
- correction and audit history.

5. Support natural-language review

The claim-aware chatbot can answer questions such as:

- What is the total medicine amount?
 - Which tests are present?
 - Why is this claim under medical review?
 - Which earlier claim contains the same invoice?
 - What evidence is missing?
 - Which document supports this amount?
 - Prepare a query requesting the itemized pharmacy invoice.
-

Reader Walkthrough

A claim moves through the platform in the following sequence:

1. A reviewer uploads the complete claim bundle.
 2. The system creates a claim number automatically.
 3. Images are saved with their original file names and fingerprints.
 4. Image preprocessing corrects orientation, perspective, contrast, and noise.
 5. OCR and the local vision model extract text, tables, dates, bill numbers, amounts, diagnosis context, medicine evidence, and investigations.
 6. Extracted evidence is normalized into structured claim records.
 7. Duplicate detection compares the claim with earlier applications.
 8. Financial reconciliation calculates medicine, test, and supported totals.
 9. Evidence-gated clinical rules assess the four clinical dimensions.
 10. Claim 360° presents the documents, extracted fields, duplicate links, finance, clinical prompts, chat, corrections, and audit history.
 11. An authorized reviewer confirms, corrects, escalates, or requests additional evidence.
-

Platform Architecture

 MedClaim Sentinel solution architecture

The implementation separates document extraction from decision logic.

Document-intelligence layer

This layer reads the actual uploaded pages and produces structured evidence:

- OpenCV image enhancement;
- Tesseract OCR for local text extraction;
- Ollama vision-language processing for difficult layouts and handwriting;
- table and field extraction;
- dates, totals, bill numbers, providers, and patient metadata;
- document-level confidence and source tracking.

Deterministic claim-review layer

This layer applies transparent rules to saved evidence:

- duplicate-document comparison;
- multi-application matching;
- amount reconciliation;
- date coverage;
- diagnosis and investigation mapping;
- prescribed-to-billed medicine comparison;
- evidence completeness;
- risk reasons and review routing.

Reviewer interaction layer

The Streamlit application provides:

- Dashboard;
- New Claim upload;
- Claim 360°;
- Duplicate Lab;
- clinical-rule view;
- claim-aware chatbot;
- correction proposals;
- protected claim deletion;
- audit visibility.

The Four-Dimension Clinical Result

The clinical page is organized around four dimensions.

Dimension	Question answered
1. Diagnosis / treatment context	What diagnosis or treatment episode is actually documented?
2. Diagnosis ↔ prescribed medicine	Does the visible prescribed medicine have support in the documented context?
3. Prescribed medicine ↔ billed medicine	Are the itemized billed medicines present in the prescription?
4. Diagnosis ↔ investigation	Are the tests and scans matched, conditional, unsupported, or missing context?

The controlled result states are:

MATCHED
 PARTIALLY_MATCHED
 CONDITIONAL_MATCH
 MISMATCH
 INSUFFICIENT_EVIDENCE
 MEDICAL_OFFICER_REVIEW
 INFORMATION

Evidence-gated behavior

The platform follows several safety rules:

- A prescription document being present is only an information event.
- A pharmacy schedule containing bill numbers and amounts is not an itemized medicine invoice.
- Pregnancy alone does not prove diabetes, hypothyroidism, vitamin B12 deficiency, or vitamin D deficiency.
- A medicine is not invented from an unreadable line, invoice number, or amount.
- Local-model output cannot override missing evidence.
- Clinical uncertainty is escalated rather than converted into an automatic rejection.

OCR and Document Intelligence

Supported evidence types

- medical reimbursement forms;
- prescriptions;

- medicine invoices;
- laboratory schedules;
- ultrasound and imaging records;
- hospital certificates;
- referral letters;
- treatment summaries;
- identity and policy documents;
- payment receipts.

Extraction outputs

Each extracted value can retain:

```
{
  "field": "medicine_total",
  "value": 11694,
  "currency": "INR",
  "confidence": 0.96,
  "document": "medicine_bill_details.jpeg",
  "page": 1,
  "original_text": "11694",
  "corrected_value": null
}
```

This design allows the original OCR value and the reviewer correction to coexist instead of silently overwriting evidence.

Local AI workflow

The local vision model is invoked only when document pages need image-level extraction. It is not run merely to open Claim 360° or refresh saved clinical rules.

The implementation supports:

- resumable page processing;
- page-level progress;
- cached evidence;
- a smaller 3B model for faster processing;
- a larger 7B model for difficult pages;
- deterministic clinical rules after extraction.

Duplicate Detection

Exact duplicate checks

- file hash;
- image hash;
- same invoice number;
- same patient, date, and amount;
- exact document reuse.

Near-duplicate checks

- rotated or cropped copy;
- photograph versus scan;
- OCR spelling difference;
- typed summary versus handwritten supporting copy;
- changed date format;
- small visual alteration.

Claim-level relationship view

A future production deployment can show a graph such as:

```
Patient / Member
├─ Claim A
│   └─ Invoice A007806
│       └─ Investigation schedule
├─ Claim B
│   └─ Invoice A007806 ← probable duplicate
└─ Claim C
    └─ Cropped image of Invoice A007806
```

Cross-insurer duplicate detection would require a legally permitted shared registry or approved data-exchange arrangement. The local MVP compares only the data accessible to the current installation.

Claim 360° Review Workspace

Claim 360° combines the full review context in one page.

Summary

- claimed amount;
- supported amount;
- recommended-review amount;
- risk score and band;
- pipeline status;
- patient, policy, and provider summary.

Identity

- patient name;
- member or policy number;
- provider;
- confidence and reviewer correction.

Documents and OCR

- original image preview;
- extracted text;
- document type;
- processing status;
- local-AI extraction state.

Extracted data

- medicine schedules;
- test schedules;
- bill numbers;
- service dates;
- amounts;
- normalized fields.

Duplicates

- matched claim;
- similarity score;
- duplicate status;
- explainable reasons.

Clinical checks

- four-dimension summary;
- grouped investigation findings;
- required evidence;
- confidence;
- source references.

Finance

- medicine total;
- test total;
- supported amount;
- repeated or unsupported amount;

- reconciliation notes.

Chat and corrections

- claim-grounded questions;
- proposed field changes;
- reviewer approval or rejection;
- preserved previous value.

Audit

- timestamp;
- actor;
- action;
- details;
- correction and deletion events.

Chat-Based Correction Control

A reviewer can request a correction in natural language, but the chatbot does not silently change claim data.

Example:

```
Reviewer request:
Change invoice A015407 date from 20-01-25 to 20-01-2025.

Proposed change:
Field: Invoice date
Current value: 20-01-25
Proposed value: 20-01-2025
Reason: Date normalization

Actions:
Approve | Reject | Add comment
```

A material change such as changing an amount should require a reason, evidence, reviewer identity, and—where configured—second-level approval.

Protected Claim Deletion

Incomplete or half-loaded claims can be removed without manually editing the database.

The deletion workflow requires:

- a deletion reason;
- the exact claim number;
- an acknowledgement checkbox.

It removes the claim, uploaded files, OCR data, clinical checks, duplicate links, correction proposals, and active audit events while preserving a minimal deletion record for governance.

The delete action never starts the local AI model.

Data Model

The local MVP uses a compact claim-centric model.

Entity	Purpose
Claim	Main claim record, status, totals, risk, patient and provider summary
Document	Uploaded file, type, path, hashes, OCR and local-AI state
Extracted field	Structured field with source, confidence, original value, and correction
Medicine item	Prescribed or billed medicine evidence
Test item	Investigation name, date, amount, and source document
Duplicate match	Related claim, score, status, and reasons
Clinical check	Dimension, status, finding, confidence, evidence requirement, and reference
Correction proposal	Current value, proposed value, reason, status, and reviewer
Audit event	Timestamped claim activity
Deletion log	Minimal governance record for permanently removed claims

A production implementation can migrate these entities to PostgreSQL or a governed cloud database while storing documents in encrypted object storage.

Technology Stack

Layer	Technologies
User interface	Streamlit
Core language	Python
Local database	SQLite
Image processing	OpenCV and Pillow
OCR	Tesseract / pytesseract
Local multimodal model	Ollama with Qwen2.5-VL
Similarity	ImageHash and RapidFuzz
Data processing	pandas
Visualization	Plotly
Testing	pytest
Local storage	File-system document store
Packaging	PowerShell and shell setup/run scripts

Repository Structure

```
medclaim-sentinel/  
├─ app.py  
├─ medclaim/  
│   ├── db.py  
│   ├── pipeline.py  
│   ├── data/  
│   │   └─ clinical_rules.json  
│   └─ services/  
│       ├── chatbot.py  
│       ├── clinical.py  
│       ├── duplicate.py  
│       ├── local_ai.py  
│       ├── metadata.py  
│       ├── ocr.py  
│       └─ risk.py  
├─ sample-data/  
│   ├── laboratory_test_details.jpeg  
│   ├── medical_certificate_page_1.jpeg  
│   └─ medical_certificate_page_2.jpeg
```

```
|   ├── medicine_bill_details.jpeg
|   └── sample_claim_transcription.json
├── tests/
|   ├── test_clinical_matching.py
|   ├── test_clinical_v7.py
|   ├── test_delete_claim.py
|   ├── test_duplicate.py
|   ├── test_local_ai.py
|   ├── test_metadata.py
|   └── test_risk.py
├── local_ai_config.json
├── install_local_ai.ps1
├── use_fast_local_ai.ps1
├── check_local_ai.ps1
├── setup_windows.ps1
├── run_windows.ps1
├── run_linux.sh
├── requirements.txt
└── README.md
```

Local Run

Windows setup

```
cd "C:\Users\Anubha\Documents\projects\medclaim-sentinel"
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass -Force
.\setup_windows.ps1
```

Install local vision AI

```
.\install_local_ai.ps1
```

For a faster local model:

```
.\use_fast_local_ai.ps1
```

Start on port 8502

```
powershell.exe -ExecutionPolicy Bypass -File .\run_windows.ps1 -Port 8502
```

Open:

http://localhost:8502

Verify Ollama processing

```
ollama ps
```

The model should be invoked only when pages require local vision extraction.

Current Implementation Status

Capability	Status
Multi-document claim upload	Implemented
System-generated claim number	Implemented
Printed and handwritten OCR workflow	Implemented
Hindi-English evidence extraction	Implemented as a local MVP workflow
Ollama vision-language extraction	Implemented
Resumable page processing	Implemented
Patient/provider/amount metadata inference	Implemented
Document and claim duplicate detection	Implemented
Financial reconciliation	Implemented
Four-dimension clinical result	Implemented
Evidence-gated deterministic clinical rules	Implemented
Claim 360° workspace	Implemented
Claim-aware chatbot	Implemented
Controlled correction proposals	Implemented
Audit history	Implemented
Protected deletion of incomplete claims	Implemented
Automated tests	16 test cases included

Capability	Status
Enterprise authentication and RBAC	Not included in the local MVP
Cross-insurer duplicate registry	Not included
Production medical knowledge base validation	Required before production use
Automatic claim approval or rejection	Intentionally not implemented

Testing Strategy

The current test suite covers areas including:

- exact-document duplicate scoring;
- risk reasoning;
- metadata inference without mandatory manual fields;
- insufficient medicine evidence;
- folic-acid antenatal matching;
- conditional levothyroxine logic;
- documented hypothyroidism matching;
- billed medicine not found on the prescription;
- prevention of medicine invention from bill summaries;
- merge of visible itemized medicine evidence;
- local clinical output integration;
- pregnancy-specific investigation logic;
- confidence and evidence fields;
- prevention of local-model overreach;
- protected claim deletion.

A production validation plan should add:

- larger multilingual OCR benchmarks;
- medical-officer blinded review;
- false-positive and false-negative duplicate studies;
- policy-rule validation;
- performance testing under concurrent claims;
- security and privacy testing;
- data-retention and deletion verification;
- model-version comparison;
- bias and subgroup review;
- formal sign-off by the insurer and medical governance team.

Security, Privacy, and Clinical Governance

Medical claim documents can contain highly sensitive personal and health information.

The portfolio and public demonstration should therefore follow these boundaries:

- do not publish real patient documents;
- do not expose names, policy numbers, addresses, signatures, or bank information;
- keep original documents outside the public repository;
- encrypt documents at rest and in transit in a production environment;
- use role-based access and least privilege;
- record reviewer changes and overrides;
- separate AI recommendations from final human decisions;
- retain model and rule versions with each assessment;
- validate retention and deletion policies;
- obtain legal, privacy, information-security, and clinical-governance approval before production deployment.

The public portfolio post intentionally describes the workflow without publishing the underlying claim scans.

Known Limitations

1. Handwriting quality can vary significantly across documents.
 2. A pharmacy bill summary may contain only invoice numbers and amounts, making medicine-level matching impossible.
 3. Local vision-model speed depends on GPU, RAM, model size, and image resolution.
 4. OCR confidence is not equivalent to clinical certainty.
 5. A diagnosis may be incomplete or expressed indirectly in the submitted bundle.
 6. Brand-to-generic normalization requires a governed medicine dictionary for production.
 7. Clinical rules require medical validation and version control.
 8. Cross-insurer matching is not possible without lawful data sharing.
 9. The current SQLite and file-system design is suitable for an MVP, not a multi-tenant production insurer.
 10. The system must not be the sole basis for claim denial.
-

Documentation Pack

The project documentation includes:

- Business Requirements Document;
- Business Understanding Document;
- Functional Requirements Document;
- Software Requirements Specification;
- Solution Architecture;
- Non-Functional Requirements;
- security, privacy, and model governance;
- test strategy, UAT, and traceability;
- deployment and operations guide;
- README, architecture Markdown, requirements Markdown, OpenAPI specification, and data dictionary.

[Download the complete MedClaim Sentinel documentation pack](#)

Practical Use Cases

User group	How the platform helps
Claim intake team	Upload and organize mixed document bundles
OCR reviewer	Correct low-confidence fields with source visibility
Claim officer	Review duplicate, finance, policy, and document findings
Medical officer	Assess diagnosis, medicine, bill, and investigation consistency
Fraud investigator	Review linked claims, repeated invoices, and reused documents
Supervisor	Monitor pipeline status and exception queues
Auditor	Review decisions, corrections, and deletion records
Consulting team	Demonstrate a local-first claim-intelligence architecture
Insurer technology team	Use the MVP as a reference for a governed production design

Roadmap

Phase 1 — Local Consulting MVP

- multi-document upload;
- OCR and local vision extraction;

- duplicate detection;
- financial reconciliation;
- four-dimension clinical review;
- Claim 360°;
- chat, corrections, audit, and protected deletion.

Phase 2 — Governed Enterprise Data Layer

- PostgreSQL or insurer-approved database;
- encrypted object storage;
- identity and role-based access;
- policy master and benefit-rule integration;
- queue assignment and SLA monitoring;
- structured pharmacy and investigation dictionaries.

Phase 3 — Clinical and Fraud Validation

- medical-officer annotation program;
- diagnosis, medicine, and test terminology normalization;
- false-positive and false-negative measurement;
- hospital, provider, and invoice-network analytics;
- model and rule governance workflow.

Phase 4 — Production Integration

- insurer core-claim APIs;
- document-management integration;
- secure notification and query letters;
- configurable approval workflow;
- monitoring, observability, and disaster recovery;
- compliant retention, archival, and deletion.

Key Strengths

- Reads the complete claim bundle rather than one isolated document.
- Combines OCR, local multimodal AI, deterministic rules, and human review.
- Detects both exact and near-duplicate evidence.
- Separates prescription presence from medicine validation.
- Uses `INSUFFICIENT_EVIDENCE` instead of inventing unsupported conclusions.
- Keeps clinical checks explainable and source-linked.
- Reconciles medicine and investigation totals.
- Supports claim-aware questions and controlled corrections.
- Keeps the local model optional and explicit.

- Includes protected cleanup for incomplete claims.
 - Provides a complete business, functional, architecture, testing, and governance documentation pack.
-

Key Innovation

The central contribution is the integration of multilingual document intelligence, duplicate-claim analysis, financial reconciliation, and conservative clinical consistency checking inside one evidence-traceable reviewer workspace.

The platform does not treat OCR as the final answer. OCR and the local vision model create evidence; deterministic rules organize that evidence; the reviewer sees the source, confidence, missing information, and reason for escalation.

Project Links

- [YouTube demonstration](#)
- [Embedded YouTube player](#)
- [Documentation pack](#)

Conclusion

MedClaim Sentinel demonstrates how local multimodal AI can support medical insurance claim review without turning the model into an automatic adjudicator.

The platform reads difficult documents, structures the evidence, identifies repeated applications and invoices, reconciles amounts, distinguishes matched evidence from missing evidence, and helps claim and medical officers review the full case through one dashboard.

Its strongest value is as a consulting MVP and reference architecture for insurers that want to reduce manual comparison effort while preserving human authority, evidence traceability, privacy boundaries, and clinical governance.

Final Thought

Read every page. Link every claim. Explain every finding. Keep the final decision human.

Medical claim OCR · duplicate applications · medicine and investigation consistency · financial reconciliation · local multimodal AI · evidence-grounded

chat · human review

Source: [src/content/posts/MedClaim-Sentinel/MedClaim-Sentinel.md](#)