International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

DOC MIND

Subramanian E[1],Bharath Kumar M S[2],Dharani Daran G[3],Haani Syed N[4],Atharah VikramN[5]

1 Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technolog

Email: esubramaniancse@siet.ac.in

2 Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

Email: bharathkumarms23cse@srishakthi.ac.in

3 Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

Email: dharanidarang23cse@srishakthi.ac.in

4 Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

Email: atharshvikramn23cse@srishakthi.ac.in

5 Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

Email: haanisyedn23cse@srishakthi.ac.in

ABSTRACT

DOC MIND provides local, evidence-driven review of resumes, invoices, contracts and scanned documents. It validates uploads, extracts source text and records domain checks. Findings include severity, evidence, score impact and recommended action. Unsupported or incompletely assessed inputs receive no ordinary health score. Human decisions remain separate from automated findings. Related revisions use explicit lineage and compatible audit runs, preventing missing analysis from appearing as improvement. Next.js, FastAPI and SQLite support the application; optional Ollama generation assists document interaction. Configurable encryption and tenant roles support controlled operation. The system passes 187 backend regressions. A developer-labelled 33-document corpus records 16 true positives, zero false positives and zero missed findings. These diagnostics do not establish independent real-document accuracy or certify correctness.

Keywords: document intelligence, explainable audit, evidence provenance, coverage-aware scoring, revision comparison, human review, retrieval augmented generation, local language models, tenant isolation.

I. INTRODUCTION

Documents influence recruitment, financial approval and contractual decisions, yet their quality is often assessed through unstructured manual review. A resume may omit essential contact information, an invoice may contain inconsistent amounts, and a contract may expose a party to unfavourable obligations. These issues require more than extracting text or producing a fluent summary. Reviewers need to know which checks ran, what evidence supports a finding and which uncertainties remain before acting on a result.

The objective of DOC MIND is to provide a reproducible document-review workflow that connects extraction, domain checks, evidence, reviewer decisions and document improvement. Users upload a supported source and receive an audit whose conclusions can be traced to recorded checks. The application is intended for job seekers, HR reviewers, invoice approvers, freelancers, tenants and business owners. It supports assistance with configured review tasks; it does not replace legal advice, professional financial verification or identity authentication.

The system distinguishes document classification, extraction quality and completed audit checks. Classification confidence does not measure legal safety, and complete extraction does not prove invoice accuracy. An empty finding list is meaningful only when applicable checks have run. DOC MIND exposes these distinctions instead of combining them into a single confident-looking percentage.

This distinction is particularly important for unsupported documents. A land note or a recipe can be uploaded and extracted, but the available resume, invoice and contract rules do not establish its correctness. Such inputs retain an unsupported-type or insufficient-coverage outcome rather than receiving a normal 100-point score. This prevents absence of analysis from being mistaken for positive approval.

The central workflow is Finding, Source Evidence, Reviewer Decision, Document Revision and Verified Comparison. A reviewer can inspect an excerpt, confirm a finding, dismiss it with a reason or request clarification. A revised upload belongs to an explicit document family and references its parent. The comparison engine then checks whether matching rules completed under comparable conditions before reporting resolution. If extraction or rule execution is unavailable, the comparison reports uncertainty rather than improvement.

Local operation is an important design choice. Deterministic rules, source storage and retrieval can run without a paid AI service. Ollama provides optional language generation using the configured qwen3:4b model, while locally cached SentenceTransformers embeddings and FAISS support retrieval. If these components are unavailable, the system identifies a lexical fallback or abstains. DOC MIND does not train a new foundation model, and a locally generated answer is not automatically treated as a verified fact.

Privacy also requires explicit operational controls. The application supports unrestricted loopback-only local use, a single-owner mode and a tenant-scoped multi-user mode. Configurable encryption protects selected content and source files when activated with recoverable keys. Authentication, ownership allocation and encryption must be deliberately configured for the actual workspace. Their implementation and disposable acceptance tests do not imply that every private installation already has these protections enabled.

The browser interface provides a public landing page and an operational workspace with a document library, upload progress, review inbox and audit views. The audit surface separates findings, source text, evidence assessment, extraction quality, improvement tools and revision history. Role-aware controls prevent viewers from being offered editing actions. Navigation back to the landing page remains available, while mobile layouts support the same core review tasks.

DOC MIND integrates conservative audit state handling with evidence-linked revisions. Missing analysis cannot silently become an excellent score or a resolved issue. The following sections describe related approaches, methods, implementation and measured results.

II. EXISTING SYSTEM AND LITERATURE SURVEY

Existing approaches include manual review, document extraction services, conversational document assistants and text-difference tools. Manual review can interpret context but relies on reviewer consistency and creates records that may be difficult to reproduce. Extraction services identify text and layout; conversational assistants retrieve passages and generate explanations; comparison tools show edits. Each capability is useful, but none should be confused with proof that all relevant audit checks completed.

Microsoft Document Intelligence documents extraction of layout information such as text, tables and structural elements [5]. Adobe Acrobat AI Assistant documents question answering with source references [6]. These examples demonstrate that extraction and cited interaction already exist in practical products. DOC MIND does not claim that these individual functions are new or that it outperforms those products. Its comparison focuses on review-state traceability and conservative handling of incomplete analysis.

Lewis et al. describe retrieval-augmented generation as a way to combine retrieved information with language generation for knowledge-intensive tasks [1]. This motivates DOC MIND's retrieval-based document interaction. However, supplying retrieved passages does not independently validate every generated statement, establish audit coverage or remove model error. The application therefore labels response sources and can abstain when sufficient evidence is absent.

Reimers and Gurevych introduce Sentence-BERT for obtaining sentence embeddings that can be compared for semantic similarity [2]. DOC MIND uses the SentenceTransformers ecosystem with a locally cached embedding model for relevant-passage retrieval. Embeddings are a search representation, not a document-health score. Retrieval quality depends on the source, chunking and model availability, so lexical fallback remains a distinct and visible operating mode.

Johnson, Douze and Jegou describe efficient similarity-search techniques underlying the FAISS ecosystem [3]. DOC MIND uses faiss-cpu for local vector search rather than adopting the paper's large-scale GPU evaluation setup. The project's measured library performance should therefore not be presented

as a reproduction of that publication's throughput or as proof of billion-document scalability.

The Tesseract documentation describes an OCR engine for converting image text into machine-readable content [4]. OCR is valuable for scanned pages, but image quality, language support and layout affect extraction. DOC MIND retains extraction warnings and treats unverified OCR as a limitation. A page containing some recognized text is not necessarily complete enough for a reliable financial or contractual audit.

Traditional keyword rules are interpretable but can ignore linguistic context. A clause stating that a party shall never accept unlimited liability must not be treated like a clause that imposes it. The project strengthens its configured liability and renewal rules with contextual and negation checks, paired with positive and negative regression cases. These checks improve known cases without constituting complete legal-language understanding.

Invoice review similarly benefits from explicit field and arithmetic checks. Currency-prefixed totals, labelled amounts, duplicated lines and absent vendor information require different treatments. DOC MIND extracts supported fields and applies configured consistency checks. Arithmetic verification is limited by the recognized invoice representation; complex tax schedules or ambiguous line-item layouts may require review rather than automated conclusions.

Resume tools commonly assess structure, terms and job-description similarity. DOC MIND checks configured contact fields, sections, length and quantified-impact patterns, then provides rewriting and matching assistance. These results are useful for editing, but they are not validated predictions of recruitment success. A lexical job match cannot prove professional competence or eliminate reviewer bias.

General text comparison identifies additions and deletions but does not establish whether a risk was resolved. If an entire payment section disappears, fewer findings may result from reduced information. DOC MIND therefore combines explicit version lineage with check-execution states. A finding is resolved only when a comparable corresponding rule passes; absent or blocked analysis produces a no-longer-assessable outcome.

Human review is also separate from machine output. A dismissal records the reviewer's interpretation and reason rather than removing the original finding. Later re-analysis creates a new audit record, and decisions remain associated with their original run. This supports accountability when source evidence or rule policy changes and prevents a previous approval from being silently reused.

Privacy-oriented local processing reduces reliance on external inference services, but local execution alone is not a complete security model. DOC MIND includes tenant boundaries, granular roles, bounded uploads and optional authenticated encryption. These controls must be activated and operated correctly; they do not protect against every compromised endpoint, lost key or malicious parser.

The evidence gap addressed by the project is therefore operational rather than a claim of a new language model. The proposed system records extraction provenance, configured rule execution, automated findings, human review and comparable revisions in one local workflow. Its distinction is the relationship between these records and the conditions required before displaying a favourable result.

Existing-system limitations are consequently framed as workflow risks: incomplete extraction can hide relevant information, persuasive explanations can conceal limited coverage, and unlinked documents can create misleading version histories. DOC MIND addresses these risks by recording execution and provenance rather than assuming that fluent outputs are complete.

III. PROPOSED METHODOLOGY

DOC MIND processes each document through validated intake, canonical extraction, domain classification, configured audit checks, coverage assessment, persistence and review. The workflow is designed so that every user-visible finding and score can be related to its supporting source and recorded execution state. Optional language generation assists interaction and editing, while the deterministic audit path remains operational without a mandatory hosted model.

Upload validation: The backend enforces a file-type allowlist, a 20 MiB size limit, content checks and bounded archive expansion. PDFs are limited to 200 pages; encrypted PDFs, malformed structures and unsupported macro content are rejected.

Validation occurs before final source persistence. When isolated worker mode is enabled, validation and extraction execute in separate parser processes with time and resource limits. Capacity is bounded to two

concurrent parser children per API process, returning a busy response rather than creating unlimited work. Failure cleanup and explicit job states allow users to distinguish rejection, processing failure and successful completion.

Canonical extraction: Native PDF text is extracted with PyMuPDF and scanned content can use Tesseract OCR. Structured DOCX extraction reads supported paragraphs, tables, headers, footers, text boxes and note structures. TXT uses a direct text path. Canonical source units are stored separately from overlapping retrieval chunks so that the viewer does not reconstruct pages by concatenating duplicated chunk content. PDF page references correspond to physical pages; DOCX and TXT references are logical units, not recovered Word pagination.

Domain classification: Content signals route a document to resume, invoice, legal-contract or other processing. Optional local-model assistance can support uncertain classification. An explicitly chosen type is recorded as a human selection rather than manufactured machine confidence.

Audit checks: The supported domains run versioned rule sets. The current expected ledger contains eight resume checks, eight invoice checks and seven contract checks. Each check records a passed, failed, blocked, not-applicable or error state. Finding counts alone are not used to establish coverage. If a rule cannot evaluate required information, the ledger records that limitation instead of inventing a successful result.

Explainable findings: Each finding carries a title, severity, category, rule identity, evidence excerpt, source reference, why-it-matters explanation, configured score deduction, recommended action, confidence representation and verification state. Source mapping links exact canonical text offsets where available. Missing-information findings may have no direct excerpt and remain distinguishable from positively observed clauses. Evidence verification describes source matching, not a certification that the legal, financial or recruitment interpretation is correct.

Coverage-aware scoring: An audit must have a supported type, sufficiently usable extraction, appropriate domain signals and a complete valid execution ledger before a normal score is displayed. Eligible scores start at 100 and subtract configured deductions, clamped between 0 and 100. Unsupported, insufficient-coverage and needs-review states withhold ordinary scores. Rule coverage, extraction quality, classification source and human review outcome remain separate. The health score is an editing heuristic, not a calibrated probability of safety.

Human review: Confirm, dismiss and request-clarification decisions are stored against an audit run and finding identity. Required explanations support dismissal and clarification. Persistent events retain reviewer attribution and chronology. Human decisions do not erase automated findings or directly rewrite the algorithmic score. This distinction is essential when a reviewer disagrees with an interpretable rule or later receives additional context.

Reliable revisions: A document family identifies related versions, and each revision records a parent. Re-analysis retains immutable audit snapshots instead of replacing the only historical result. Finding identities combine rule, category, title and normalized evidence with occurrence information. The comparison controller checks type, policy version, extraction completeness and corresponding rule versions before interpreting a change. It does not compare arbitrary documents merely because both are contracts or invoices.

A matched evidence identity can remain unchanged. A previously reported concern is resolved only when a comparable revised check completes and passes without a remaining finding for that rule. Newly detected evidence is labelled new only when baseline coverage permits that interpretation. Changed, blocked or incomplete analysis produces no-longer-assessable findings. A score delta is withheld when the two audit runs are not fully comparable. These rules prevent missing analysis from appearing as improvement.

Document question answering: The retrieval service obtains relevant chunks through optional local embeddings and FAISS or a lexical fallback. Ollama receives a constrained prompt that treats document content as untrusted data, not executable instructions. Structured output is validated before presentation. Response-source labels distinguish deterministic audit responses, evidence-only output, local-model generation, lexical fallback and abstention. Source references help review answers but do not independently prove factual correctness.

Security methodology: Tenant ownership scopes documents and associated records in multi-user mode. Administrators, editors, reviewers and viewers receive different permissions, enforced on the backend. Passwords use PBKDF2-SHA256 with 600,000 iterations; sessions use HTTP-only cookies and credential snapshots so password changes revoke old sessions. Configurable Fernet encryption protects selected content and source files at rest. Key retention, approved ownership allocation and backups remain

operator responsibilities.

Persistence and recovery: SQLite stores processing jobs, document metadata, canonical content, audit snapshots, findings, reviewer events, workflow state and account activity. Paginated APIs support the library, and aggregate queries are not limited to the recent-document display. Backup and restore tools verify integrity and revoke restored sessions. Structured monitoring and diagnostics provide operational visibility without claiming that an external service has already been configured.

IV. SYSTEM IMPLEMENTATION

The application uses a modular browser-and-API architecture. The frontend manages document intake, navigation and review interactions; the backend owns validation, extraction, analysis, authorization and persistence. A local Ollama service provides optional generation. Source files, database records and retrieval indexes stay within the configured local environment. The application has not been publicly deployed, and external infrastructure is not part of the completed local scope.

Frontend stack: Next.js 15.5.27, React 19.0.8 and TypeScript provide the web application. Tailwind CSS and workspace styles define responsive layout, while Lucide supplies interface icons. Reusable controls support loading, disabled, empty and error states. Three.js and Anime.js are included for presentation-layer motion; they are not components of the audit logic or evidence validation. A video-backed landing experience is visually separate from the dense operational review workspace.

Workspace views: Overview, Documents, New Audit, Review Inbox and Current Review provide predictable entry points. Audit views expose findings, source evidence, extraction quality, revision comparison and assistance tools. Account administration is available to tenant administrators in configured multi-user mode. Viewer and reviewer controls reflect backend restrictions, and identity changes clear cached document summaries and invalidate pending audit navigation responses.

Backend stack: Python, FastAPI 0.142.2 and Uvicorn serve the API. Pydantic validates request and response schemas, while SQLAlchemy 2.0.36 manages SQLite persistence. The frontend communicates using credentialed REST requests rather than directly opening private source files or the database.

Extraction stack: PyMuPDF 1.24.14 handles PDF reading; python-docx 1.1.2 and structured OOXML handling support Word content. Pillow and pytesseract interface with image processing and Tesseract OCR. defusedxml is used for defensive XML parsing. Extraction records methods, source numbering, recognized structures and warnings so that later checks can explain their input limitations.

Processing jobs: Job state is persisted rather than held only in memory. The API exposes progress and recoverable failure information, with retry and cancellation controls. Interrupted processing is distinguishable from successful auditing. The local implementation does not claim to be a distributed production worker queue; multi-host scheduling and sustained resource management remain separate operational concerns.

Rule implementation: Resume rules evaluate contact information, required sections, impact-pattern signals and length. Invoice rules evaluate supported fields, vendor information, line and total arithmetic, duplicate lines and currency handling. Contract rules evaluate parties, dates, termination, payment and configured liability, renewal and penalty patterns. Rule identities and versions are recorded in the execution ledger, supporting re-analysis and conservative revision comparison.

Semantic safeguards: Known currency-prefixed total and negated-clause failures are covered by regression cases. Examples include Total EUR 110, never accepting unlimited liability, automatic renewal not applying, and liability without a financial cap. These examples exercise corrected pattern handling. The rule engine is still limited to configured linguistic patterns; complex exceptions, cross-references and jurisdiction-specific interpretation require professional review.

Storage model: Documents belong to families and optional parent versions. Audit runs retain result snapshots and policy information. Reviewer events and workflow updates remain durable, while chat history can be retrieved through the API. Canonical source units are separate from overlapping search chunks. FAISS indexes and source files supplement the relational records; backup tools handle their paths and restoration integrity.

Local AI stack: Ollama hosts qwen3:4b for generation. SentenceTransformers 6.1.0 with the configured all-MiniLM-L6-v2 model can generate embeddings from a local cache; faiss-cpu 1.9.0.post1 and NumPy support vector operations. The user's separately installed nomic-embed-text model is not assumed to be the configured embedding path. Dependencies can be installed without the relevant model being available, so the runtime explicitly exposes fallback behaviour.

Improvement tools: Rewrite Studio and the clause library propose editing assistance. Persona Mode changes review priorities; resume matching identifies job-description terms; invoice workflow records approve, reject, query-vendor and paid statuses.

These tools support human work rather than independently proving the suggested wording is legally safe, an applicant will be selected or an invoice has actually been paid. Fix status and reviewer actions are stored persistently. A projected score recovery is not counted as verified improvement until a linked revised audit supplies comparable evidence.

Access controls: Multi-user mode scopes data to a tenant and applies roles. Administrators manage accounts and deletion; editors can upload and re-analyse; reviewers can record decisions; viewers receive read-only core access. Account updates require reasons, exclude passwords from activity payloads and cannot remove the last active administrator. Cross-tenant records are not revealed through document or account endpoints. Encryption and these access modes are configurable features, not silently enabled properties of the original real library.

Resilience controls: Isolated parser mode uses separate processes, deadlines and bounded capacity. On Windows, Job Objects cap memory, CPU time and child-process count, with kill-on-close cleanup. These are resource limits, not a complete filesystem or network sandbox. Monitoring uses bounded structured logs and optional configured notifications. Backup verification checks database integrity, source readability, ownership, review history and account history while rejecting incorrect keys.

Quality tooling: Python unittest covers backend regressions; frontend TypeScript checking and ESLint check implementation consistency. Playwright drives navigation, role boundaries, account administration and responsive evidence views. A synthetic labelled corpus exercises supported domains, unsupported inputs, extraction and negation cases. Load diagnostics use encrypted disposable storage so that measurements do not alter the actual document library.

Report generation: ReportLab 4.2.5 creates downloadable audit PDFs containing findings, source references and score explanation. These exports preserve review context but do not certify a document. Operational reports record the measurement scope and distinguish successful fixture tests from independent real-document assessment and actual-library security activation.

Figures 1-4 summarize processing, data relationships, revision decisions and verification scope. The backend enforces validation and authorization; the frontend exposes source evidence, uncertainty and permitted actions.

V. ADVANTAGES

1. Evidence Linked Review

Traceable findings: Reviewers can relate a concern to its configured rule and canonical source excerpt rather than relying on a generic paragraph of advice. Source numbering explicitly distinguishes physical PDF pages from logical DOCX or text units.

Inspectable limitations: Missing excerpts, incomplete extraction and blocked checks remain visible. A source match verifies the location of a passage, not the substantive correctness of the interpretation, allowing the reviewer to challenge automated output.

2. Conservative Scoring

Coverage before score: Unsupported documents and insufficiently assessed sources do not receive ordinary favourable scores. Recorded passed and failed checks provide a stronger basis for interpreting a result than the number of findings alone.

Explainable deductions: Eligible scores show configured impact and recommended action. Classification confidence, extraction status and human review are separate, reducing the risk of treating a heuristic health score as a safety probability.

3. Reliable Revision Tracking

Explicit lineage: Document families and parent links prevent unrelated documents from being presented as versions. Immutable audit runs preserve the original analysis for repeatable before-and-after inspection.

Conditional improvement: Disappearing findings are not automatically resolved. The engine requires compatible policy, extraction and rule execution, and uses no-longer-assessable states where the evidence cannot establish improvement.

4. Local Operation and Access Boundaries

Optional local AI: Deterministic audit checks do not require a paid hosted AI API. Local generation and retrieval can assist interaction while reporting fallback and abstention when models or supporting evidence are unavailable.

Configurable protection: Tenant ownership, granular roles and content encryption support controlled operation when activated. Account administration and persistent activity provide an inspectable local management workflow without deleting historical document ownership.

5. Durable Workflow and Recovery

Persistent review: Jobs, audit runs, chat and reviewer actions can be revisited. Decisions remain distinct from automatic findings, and dismissals record reasons rather than silently erasing concerns.

Recoverable records: Copy-based migration and backup tools preserve originals and verify restored data. Disposable acceptance tests demonstrate expected recovery behaviour while making no claim that the real library has already undergone a recovery rehearsal.

6. Evaluation and Responsible Claims

Measured verification: Regression, browser and labelled diagnostics expose outcomes and sample sizes. Independent real-document evaluation remains pending.

Distinctive workflow: Findings, evidence, human decisions and related revisions remain linked. Incomparable audits cannot imply improvement; global patent novelty and universal accuracy are not established.

Figure 1 DOC MIND processing

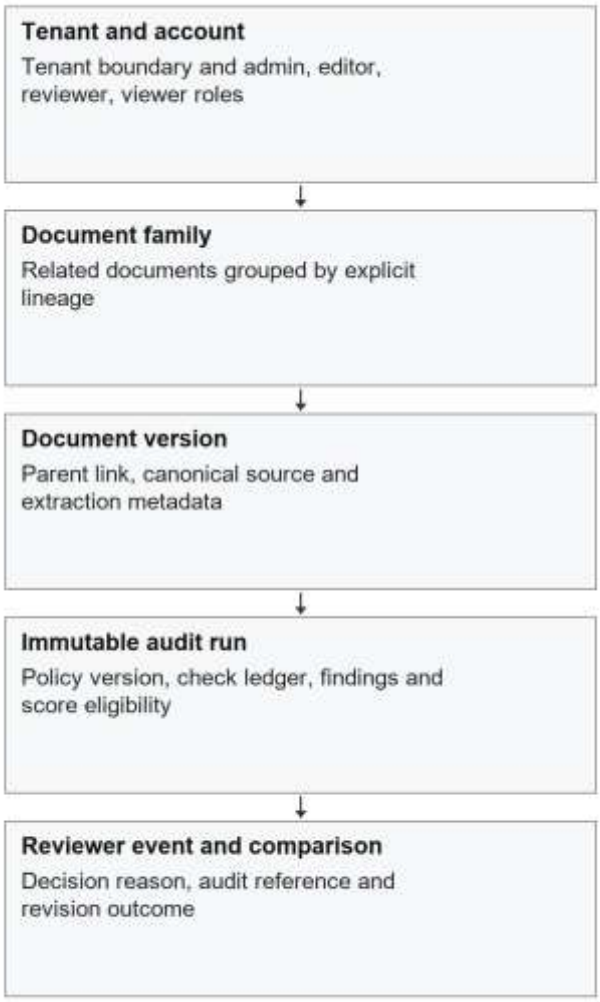
Validated source to a recorded audit



Optional Ollama generation assists interaction; it does not replace the deterministic audit path.

Figure 2 Review data relationships

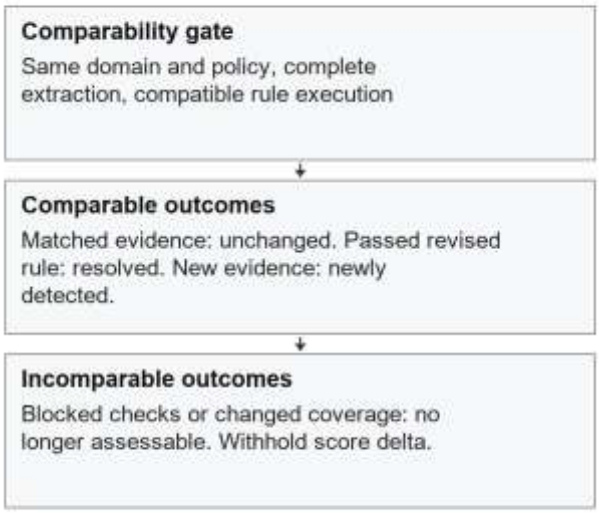
Durable records with explicit ownership



Human decisions remain separate from automated findings. A new audit does not inherit an old approval.

Figure 3 Revision decision

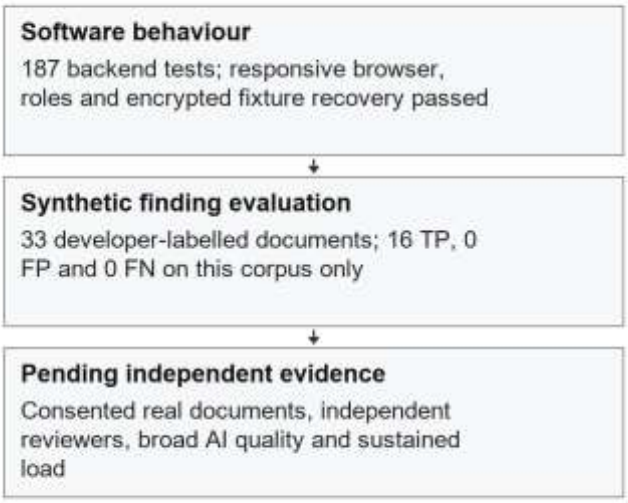
Missing analysis is never improvement



Resolution means a recorded pattern check passed, not independent legal or financial verification.

Figure 4 Verification boundaries

Measured local results and their scope



Tests and diagnostics do not establish 100 percent real-world accuracy or public-deployment readiness.

VI. RESULTS AND ANALYSIS

The full backend regression suite completed on 8 October 2026 with 187 passing tests. Frontend lint and TypeScript checks passed. Disposable encrypted multi-user acceptance verified login failures, session reload, sign-out, tenant boundaries, account creation and activity, role-restricted controls and evidence-assessment layout at 1440, 1024, 768, 390 and 320 pixels. The corresponding restore drill preserved source contents, tenant ownership, reviewer events and account history, revoked restored sessions and rejected an incorrect encryption key. These tests establish specific software behaviours, not a penetration-test certification. Separately, the 7 October synthetic evaluation included 33 developer-authored documents: 11 invoices, 10 contracts, nine resumes and three other inputs. It recorded 16 true positives, zero false positives and zero missed labelled findings, giving 100 percent precision and recall on this corpus alone. An intentionally blank scan produced the expected extraction failure; no unexpected extraction failures were recorded. The labels were neither independent nor held out, so these results must not be generalized to real-world audit accuracy.

A separate single-machine HTTP diagnostic used 10,000 synthetic inventory records in encrypted tenant-scoped storage. Forty library reads at concurrency four recorded 0.281-second p50 and 0.485-second p95 latency; the statistics request took 2.140 seconds. Four uploads at concurrency two took 2.953 seconds for TXT, 2.937 for PDF, 2.875 for DOCX and 3.281 for PNG OCR. The scan remained insufficient-coverage rather than receiving a misleading normal score. Two concurrent live local-model responses took 48.984 and 37.719 seconds and explicitly reported local_model rather than fallback. No errors occurred in this workload. These timings distinguish fast deterministic processing from slower language generation. They are small diagnostic samples, not sustained-load capacity guarantees, and the inventory records were not 10,000 independently audited documents. Overall, the results support local workflow correctness and recoverability while leaving independent semantic accuracy, broad OCR quality and long-running workload stability for further evaluation.

VII. CONCLUSION

DOC MIND implements a local document-review application that connects validated intake, canonical extraction, configured domain audits and human decision records. Its principal objective is not simply to generate more text about a file, but to make the basis and limits of a review inspectable. Supported documents receive explainable findings and conditional health scores; unsupported or insufficiently assessed inputs retain explicit uncertainty.

The architecture combines Next.js and React with FastAPI, SQLAlchemy, SQLite, extraction libraries and optional Ollama generation. These are established technologies; the project contribution lies in their orchestration with a check ledger, audit snapshots, evidence identities and conservative revision states.

Regression and acceptance results demonstrate concrete implementation behaviours including tenant isolation, role-aware controls, persistence and encrypted fixture recovery. The corrected synthetic cases provide useful diagnostics for currency handling and clause negation. However, successful tests do not prove complete legal understanding, financial authenticity, recruitment effectiveness or independently established real-document accuracy. The system remains an assistive workflow with human responsibility for consequential decisions.

The integrated Finding to Evidence to Decision to Revision to Comparison workflow is the project's strongest practical distinction. It allows a reviewer to investigate a concern, record a response and check a related revision without treating missing analysis as success. Further evaluation should extend this traceability to consented real documents, original-page evidence visualization and carefully measured long-running local operation.

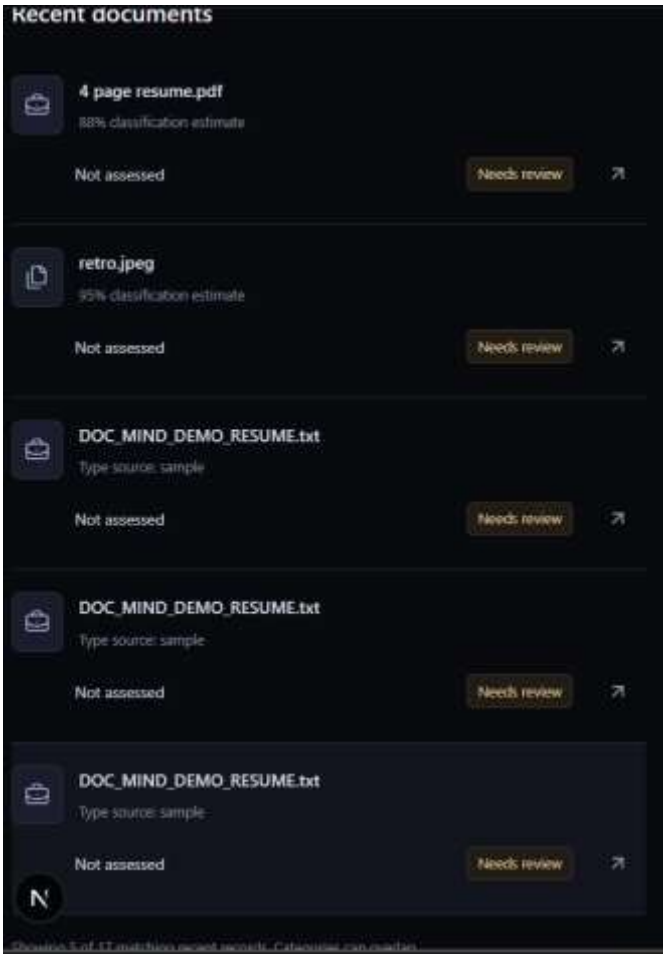
VIII. FUTURE WORK

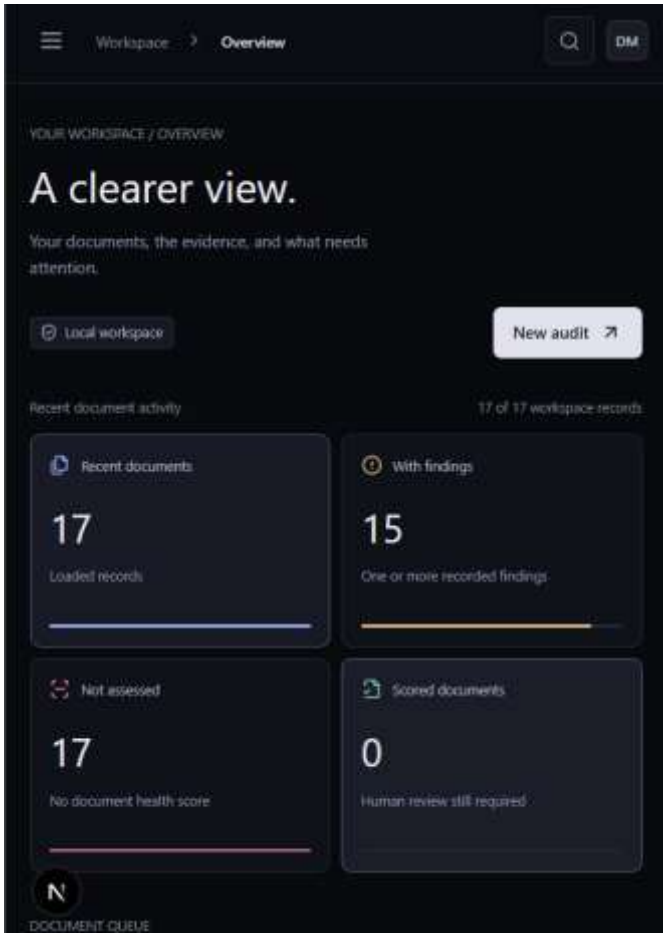
Independent evaluation is the highest-priority next activity. A consented and redacted corpus should include realistic resumes, invoice layouts, contracts, poor-quality scans and unsupported documents. Separate reviewers should annotate expected findings, ambiguous cases and extraction failures. Held-out evaluation should report precision, recall, false-positive burden and extraction quality by domain, format and document condition. AI summaries, chat and editing suggestions require separate human assessment rather than inheriting deterministic rule-test results.

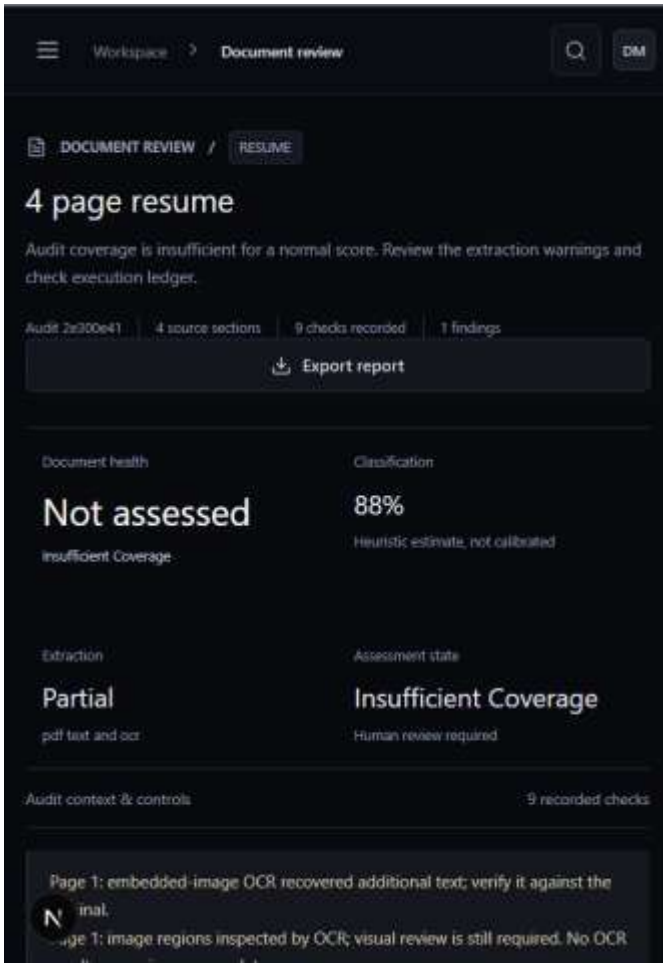
Evidence presentation can be extended from exact canonical excerpts to original-page visual highlighting when reliable coordinates are available. OCR uncertainty and table reconstruction should be shown without inventing locations or implying physical DOCX pagination. Longer local stress and soak tests should measure maximum-size files, cancellation, restart recovery, storage growth and model latency. Resource-cap tests do not replace hostile-parser security review, and actual-library recovery must be

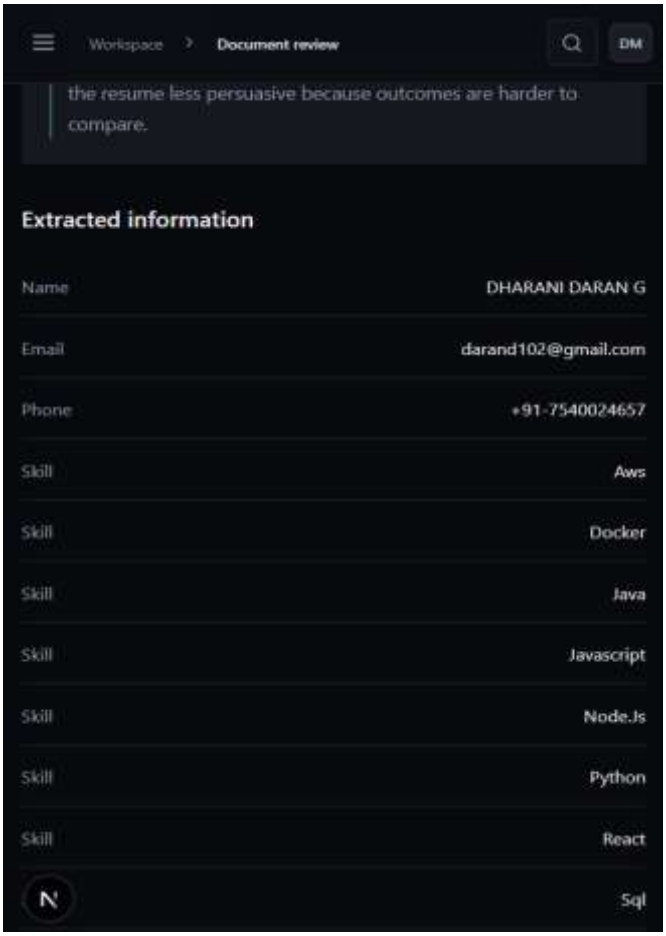
rehearsed with retained keys and authorization.

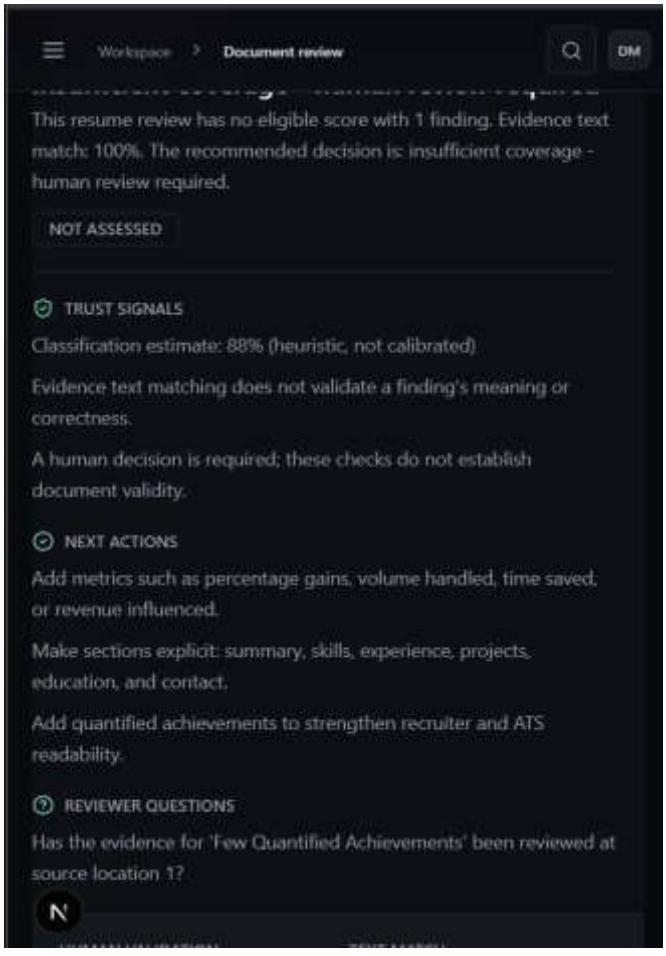
Operational completion also requires deliberate activation of protected modes on the real workspace: chosen tenants and roles, reviewed ownership allocation, recoverable encryption keys and verified backups. The original private library has not been silently migrated or encrypted. Public deployment is outside the current project scope. Any later public service would require independently reviewed authorization, secure hosting, monitoring and incident procedures. Future improvements should preserve the conservative principle that unavailable evidence is not evidence of a successful audit.






Workspace
Document review
DM
DOCUMENT REVIEW / RESUME
4 page resume
Audit coverage is insufficient for a normal score. Review the extraction warnings and check execution ledger.
Audit 2e300e41
4 source sections
9 checks recorded
1 findings
Export report
Document health
Not assessed
Insufficient Coverage
Classification
88%
Heuristic estimate, not calibrated
Extraction
Partial
pdf text and ocr
Assessment state
Insufficient Coverage
Human review required
Audit context & controls
9 recorded checks
Page 1: embedded-image OCR recovered additional text; verify it against the
inal.
ge 1: image regions inspected by OCR; visual review is still required. No OCR


Workspace
Document review
DM
the resume less persuasive because outcomes are harder to compare.
Extracted information
Name
DHARANI DARAN G
Email
darand102@gmail.com
Phone
+91-7540024657
Skill
Aws
Skill
Docker
Skill
Java
Skill
Javascript
Skill
Node.Js
Skill
Python
Skill
React
Sql


Workspace
Document review
DM
This resume review has no eligible score with 1 finding. Evidence text match: 100%. The recommended decision is: insufficient coverage - human review required.
NOT ASSESSED
TRUST SIGNALS
Classification estimate: 88% (heuristic, not calibrated)
Evidence text matching does not validate a finding's meaning or correctness.
A human decision is required; these checks do not establish document validity.
NEXT ACTIONS
Add metrics such as percentage gains, volume handled, time saved, or revenue influenced.
Make sections explicit: summary, skills, experience, projects, education, and contact.
Add quantified achievements to strengthen recruiter and ATS readability.
REVIEWER QUESTIONS
Has the evidence for 'Few Quantified Achievements' been reviewed at source location 1?

REFERENCES

[1] P. Lewis et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33, 2020, pp. 9459-9474. https://arxiv.org/abs/2005.11401

[2] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP-IJCNLP, 2019, pp. 3982-3992. https://doi.org/10.18653/v1/D19-1410

[3] J. Johnson, M. Douze and H. Jegou. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 2021, pp. 535-547; preprint 2017. https://arxiv.org/abs/1702.08734

[4] Tesseract OCR contributors. Tesseract User Manual. Official documentation. https://tesseract-ocr.github.io/tessdoc/

[5] Microsoft. Document layout analysis with Document Intelligence. Official documentation. https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/prebuilt/layout

[6] Adobe. Use AI Assistant in Acrobat. Official product documentation. https://helpx.adobe.com/acrobat/desktop/use-acrobat-ai/generative-ai-features/ai-get-answers.html

[7] Qwen Team. Qwen3 Think Deeper Act Faster. Model release and local inference overview, 2025. https://qwenlm.github.io/blog/qwen3/

[8] Ollama. API Introduction. Official local inference API documentation. https://docs.ollama.com/api/introduction

[9] FastAPI contributors. FastAPI Documentation. Official API framework documentation. https://fastapi.tiangolo.com/

[10] Vercel. Next.js Documentation. Official web framework documentation. https://nextjs.org/docs

[11] SQLAlchemy contributors. SQLAlchemy 2.0 Documentation. Official ORM documentation. https://docs.sqlalchemy.org/en/20/

[12] PyMuPDF contributors. PyMuPDF Documentation. Official PDF-processing documentation. https://pymupdf.readthedocs.io/en/latest/

[13] python-docx contributors. python-docx Documentation. Official Word-processing library documentation. https://python-docx.readthedocs.io/en/latest/

[14] Python Cryptographic Authority. Fernet Symmetric Encryption. Official authenticated-encryption documentation. https://cryptography.io/en/latest/fernet/

[15] DOC MIND project implementation records. Synthetic Evaluation Results, HTTP Load Acceptance, Local Upgrade Status and Security Acceptance. Local engineering artefacts, 7-8 October 2026. Online references [4]-[14] accessed 8 October 2026.