Byomkesh AI: AI-Powered Investigation Intelligence and Automated Charge-Sheet Generation System

Anil Ahir[1], Aarya Umesh Gharmalkar[2*], Nishad Shirodkar[3], Saurabh Shahane[4], Kaustubh Karnik[5]

[1]Professor, [2, 3, 4, 5]Student, Department of Artificial Intelligence and Data Science, Vivekanand Education Society's Institute of Technology (VESIT), Mumbai, India.

E-mail: [2*]aaryagharmalkar@gmail.com

Abstract

Investigation records are distributed across varied sources, and their facts may be incomplete, duplicated, or inconsistent, making evidence tracking, legal reference checking, and charge-sheet preparation difficult. This paper introduces Byomkesh AI, an investigation intelligence system that organizes data derived from documents into a reviewable, evidence-linked case representation. The system has nine stages: document ingestion, OCR, document parsing, investigation graph construction, canonical projection, evidence validation, case context construction, legal-reference reasoning, and charge-sheet population. During validation, there are three possible dispositions: draft allowed, review required, and final blocked. The mean extraction confidence score for the project is reported to be 0.99 (99%), but no information has been provided regarding the process used for aggregation and calibration of this score. It is not appropriate to interpret this extraction confidence score as the score for extraction accuracy. There is no benchmark for mapping accuracy, evaluation dataset, denominator or ground truth for BNS-mapping. The Section 106 example given in the project presentation is merely a demonstration, and not an evaluation of accuracy.

Keywords: Investigation Intelligence, Charge-Sheet Generation, Legal Document Processing, Evidence Provenance, Hybrid Retrieval, Human-in-the-Loop Review.

*** Corresponding Author**

Journal of Information Technology and Digital World, December 2026, Volume 8, Issue 4, Pages 410- 433
DOI: https://doi.org/10.36548/jitdw.2026.4.009
Received: 18.09.2026, received in revised form: 21.10.2026, accepted: 06.11.2026, published: 21.11.2026
© 2026 Inventive Research Organization. This is an open access article under the Creative Commons Attribution-NonCommercial International (CC BY-NC 4.0) License

1. Introduction

Investigation case files store FIRs, complaints, statements of witnesses and accused, reports of medico-legal and forensic examinations, seizure documents, site plans, digital media records, case diaries, and other documents. They differ in format and purpose, but they all contain overlapping information about people, events, locations, and things. The investigator has to find facts in these documents, differentiate direct citations from summaries, resolve identities and timelines, detect gaps and contradictions, connect claims with their supporting documentation, take into account provisions of relevant statutes, and draft a comprehensive charge sheet. The same name can refer to different people, two documents may describe the same event in different ways, and a logical explanation can turn out to be completely baseless. That is why this problem cannot be solved by OCR alone: the extracted text needs to be classified into fields of the document, compared between documents without merging identities, verified by reference to the sources, and moved to a structured document together with any unclarified gaps. It cannot be substituted by classification of legal texts, since it operates on a different input and cannot verify whether a particular element of the field is supported by a certain investigation document. Automation will eliminate repetitions in transcriptions but poses risks: OCR can change names and dates, schema-valid fields can be incorrect, entity resolution can conflate identities, and text generation will hide gaps and discrepancies in the facts. Hashing of files does not prove anything about authenticity and custody, and matching of statutory phrases is not legal analysis.

Byomkesh AI is perceived as a decision support system for case structuring in relation to evidence and unsigned drafts, but not as an independent investigation authority. The research problem consists of the lack of evidence on the functioning of record extraction, case linking, validation, legal reference search, and form population processes in tandem on investigation records. There are no benchmarks on document AI or legal NLP that would ensure end-to-end source support or conflict-safe processing. The paper poses the following research questions: (RQ1) How are heterogeneous records turned into a case and a draft? (RQ2) What validation, retrieval, and review controls are applied, and where does their applicability terminate? (RQ3) What measurements can be reproduced? The contributions include the description of a source-based nine-stage workflow; the models of graphs, canonicity, provenance, validation,

and reviewer; the separation of target architecture from code-verified operation; and the evidence-based evaluation.

2. Background and Related Work

2.1 Document AI, OCR, Layout, and Extraction

Document processing includes character recognition, layout analysis, field extraction, and question answering over document images. DocVQA evaluates document image question answering. LayoutLMv3 and DocFormer leverage textual and visual information for document understanding [1], [2], [3]. Their results motivate retaining page context and typed fields: in case files, a date in a header may have a different role from a date in a witness narrative. These benchmarks do not evaluate this system or police-investigation records.

OCR and information extraction should be measured separately. Recognition mistakes affect names and dates, whereas correct transcription might be incorrectly matched to the wrong fields. Evaluation of OCR requires ground truthing, normalization, and metric selection, but neither character nor word error rates indicate the field level precision, quality of source-references, and completeness [4]. Byomkesh AI preserves OCR results and parser results per page, but there is no adjudicated field level benchmark. Typed output is a data constraint, not evidence of factual correctness.

2.2 Legal NLP, Knowledge Graphs, and Statutory Mapping

LexGLUE and IL-TUR frame classification and reasoning challenges related to English and Indian legal language respectively, while LeSICiN and NyayGraph work on graph or knowledge graph augmentation-based statute identification [5, 6, 7, 8]. All these efforts make use of curated legal data and do not create any mapping or extraction of provisions from heterogeneous police reports, whose veracity can be questionable, and where the evidentiary value is different from the value of judgments/statutes. In Byomkesh AI, the investigation graph comprises claims from documents, and the statutory reference file consists of configured provisions.

2.3 Evidence Provenance, Retrieval, and Human Oversight

Retrieval-Augmented Generation, language model output depends on retrieval passages [9]; in citation-oriented generation work, answer fluency and evidence support are distinct quality factors [10]. Retrieval aids record location but does not verify completeness, credibility, or legal relevance. The case-specific retrieval system presented here is a tool for inspection, not an answer generation validation benchmark.

Human-AI interaction advice stresses inspection, correction, and overriding, and human-in-the-loop entity linking work takes review as part of the task not as a verification of quality [11, 12] There is documentation of corrections, annotations, decisions, history, and searches for evidence in the back-end API of Byomkesh AI, but usability and its effect on investigator productivity have not yet been explored.

2.4 Comparative Gap and Contribution

Table 1 shows the comparison of associated task boundaries, thereby establishing the integration gap that will be considered in this study: there is no benchmark that defines how extraction, case representation, validation, retrieval, form population, and review work together for investigative records. This paper aims to provide a source-based implementation rather than benchmark results.

Table 1. Major components of the proposed system.

Research area	Established task focus	Relevance and unresolved issue for this system
Document AI and OCR	Image transcription, layout understanding, document QA, and extraction [1, 2, 3]	Motivates page-aware typed extraction; investigation-specific accuracy is not established.
Legal NLP	Legal-text classification, Indian legal reasoning, and statute identification [5, 6, 7, 8]	Provides task methods, but not mapping over investigative evidence.
RAG and citations	Retrieval-conditioned generation and citation support [9, 10]	Supports passage discovery; does not establish completeness or legal sufficiency.
Human–AI review	Interaction guidance and human-assisted entity linking [11, 12]	Motivates correction and override; reviewer effectiveness remains untested.

Each research stream covers different levels of units and objectives. OCR/layout systems detect and transcribe page contents, whereas document-QA benchmarking analyzes answers to image-based questions. Legal NLP benchmarking involves classification and reasoning with legal documents, whereas statute identification benchmarking establishes relationships between decisions or statutes. Each of them, taken separately, cannot encompass the required procedure, for example, the extracted document needs to become a case event or entity, retain a traceable link to its source, go through a completeness and conflict checks, give a definite legal reference link, and get mapped to a typed field in the charge sheet form. Retrieval can offer a way to seek out the relevant passage, and ranking does not guarantee that all the information will be found and conflicting testimonies reconciled. The human-AI interaction guidance is motivated by correction and override needs but without proof of enhanced decision-making.

Therefore, Byomkesh AI can be considered an investigation of integration and implementation. This research does not offer any new algorithms for OCR, legal reasoning, graph learning, or retrieval. In addition, the results achieved are not better than those demonstrated by the benchmarks mentioned. Its uniqueness consists in defining the system boundaries and distinguishing the source, validation, reviewer, and evaluation claims that correspond to the code rather than the project documentation.

3. Materials and Method

3.1 Materials and Audit Procedure

The study examines the accepted manuscript, project presentation, and project source tree with versions. Workflow composition, parsers and domain models, validation logic, legal-reference data, charge-sheet schema and rendering, API route definition, and synthetic fixture data have been analyzed. Code and schema represent the real-world working behavior of the system, whereas presentation represents the intended architecture or results. This audit does not represent an independent service deployment. The synthetic PDFs used as input for this update are unavailable, so the numbers of extraction from the fixture level are calculated from the prior local calculations, not from any new computations. Project documentation includes details of the discussions conducted with senior police officials and the technical

team in the office of the Police Commissioner, Navi Mumbai. Project presentation refers to the sample records and conditional acceptance of the project.

3.2 Scope and Architecture Evidence

Architecture and Technology claims made during the presentation were verified based on workflow, route, model, and adapter code. The next section distinguishes target components mentioned in documentation from repository-based functionality, where both slides and source code verification do not constitute deployment of the service.

4. System Design and Methodology

4.1 Architecture and Component Boundaries

The chosen target makes a distinction between user interaction, orchestration, assisted extraction, structured case data, legal references, and output rendering. The React app must be built to enable the selection of cases, uploads, jobs, evidence, field correction, and draft review. FastAPI includes API endpoints for cases, jobs, artifacts, authentication, and reviews; processing is done by Python stage services, while Pydantic provides data constraints. Gemini adapters will assist with document OCR and parsing with schemas, while deterministic components will generate graphs and canonical artifacts, validate case data, locate reference passages, and provide charge sheet models. According to the target architecture, Neo4j handles graph relations, PostgreSQL stores structured data and metadata, Supabase supports authentication and object storage, JWT carries authentication data, and ReportLab renders Form IF-5 PDFs.

The repository boundary after validation is more limited. Workflow and API are implemented in Python, file storage is utilized for case artifacts, reviewer actions are represented via FastAPI endpoints, opaque in-memory bearer sessions are used for the authentication service, and PDF reports are created with the ReportLab library. The repository version considered in this paper does not include code for frontend applications or adapters for Neo4j, PostgreSQL, and Supabase. Thus, the above-mentioned systems can be referred to as the target architecture.

4.2 Nine-Stage Processing Workflow

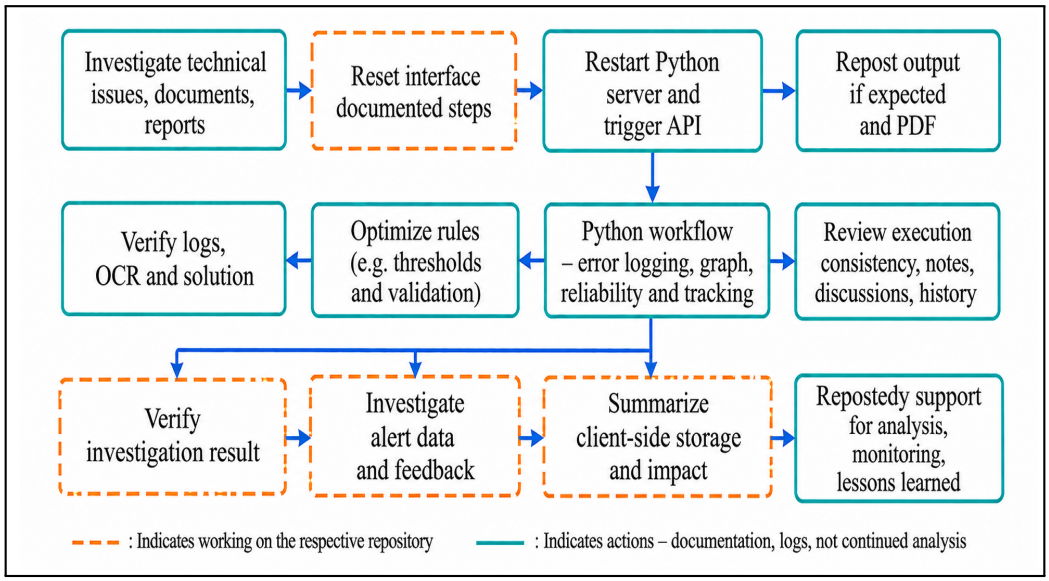


Figure 1. Complete system architecture: documented target and repository-verified components.

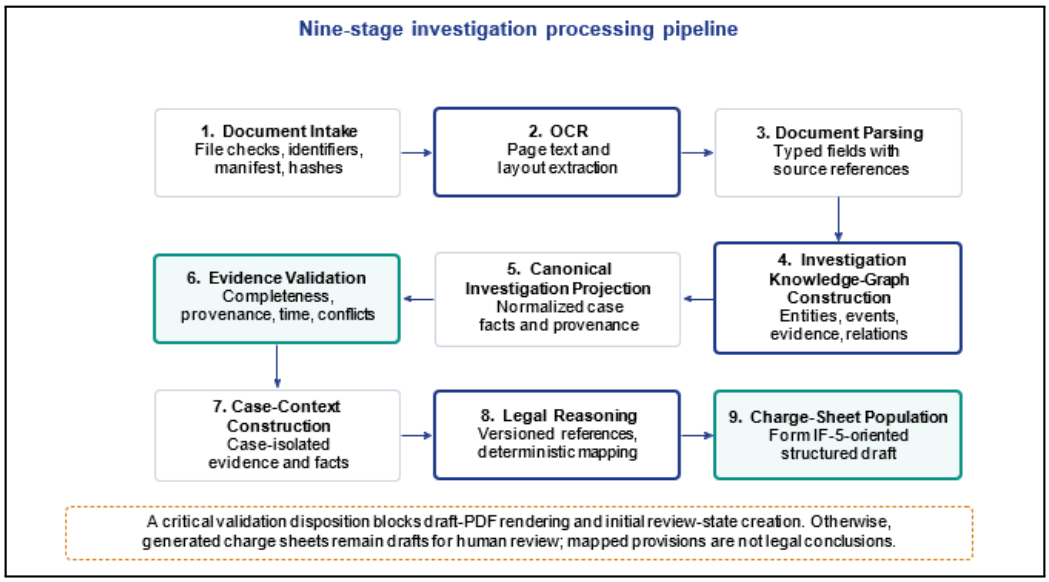


Figure 2. Nine-stage investigation processing pipeline from document intake to charge-sheet population.

Production sequence is made up of the following steps: document intake, OCR, document parsing, investigation knowledge graph construction, canonical investigation projection, evidentiary validation, context construction, legal reasoning, and charge sheet population. Each step trades typed context and artifacts with other steps, and later steps do not

correct any errors made in earlier steps. The figure below represents all nine steps.

Stage 1: Document Intake. Input: Uploaded documents and document identifiers. Python intake services validate the type and size, assign identifiers, create a manifest, store the originals and create SHA-256 digests. Output: A manifest for each case and original documents. Documents that are unprocessable, have size violations, are unprocessable uploads, or are improperly resuming inputs will cause an intake failure. The digests can be used to identify but not custody or authenticate the documents.

Stage 2: OCR. Input: Accepted documents. Gemini adapter creates the page-level text and confidence information; mock clients will be used to create non-liveware features. Output: OCR artifacts concerning documents and pages. Client error, unsupported media, and blank output results cause the failure of the stage. Confidence values are not calibrated measurements of accuracy.

Stage 3: Document Parsing. Input: OCR text, document classification, and parser schemas. The registry employs relevant document parsers or a fallback strategy. Gemini extraction is restricted by the Pydantic model validation process. Outputs: typed local records extracted from documents. If the type is unrecognized, a fallback will be used. Schema and parser errors lead to exceptions being recorded. Validation of type conformance does not check for the truthfulness of the fact.

Stage 4: Construction of Investigation Knowledge Graph. Input: Parsed records. An automated, deterministic and python-based graph building mechanism transforms explicit properties into typed persons, documents, evidence, events, vehicles and locations. Outputs: an investigation knowledge-graph. Modeling errors and absence of provenance information limit persistence; a rule-based identification mechanism does not disambiguate identities.

Stage 5: Canonical Investigation Projection. Input*:* Graphs, facts and parsed records. Generators of projections classify metadata related to cases, persons, vehicles, locations, evidence, findings, activities, events, conflicts, unknowns, and references in terms of canonical models. Outputs: a canonical investigation. Model invariants and source references are checked; non-projectable facts are ignored.

Stage 6: Validation of Evidence. Input: Canonical data. Deterministic validations for completeness, source, contradiction, chronology, entity, unsupported fact, and confidence produce issues and dispositions. Output: Validation report. Important issues produce FINAL

BLOCKED but do not prevent generation of PDF and level 1 review. It is impossible for the rules to prove all contradictions have been discovered.

Stage 7: Case Context Construction. Input: Canonical data and validation report. A deterministic builder chooses information, data, events, problems, and provenance for legal mapping and population formation. Output: Case context of a particular type. The absence of prerequisites and inconsistent case identifiers makes case context construction impossible.

Stage 8: Legal Reasoning. Input: Case context, statutes referred from that jurisdiction and retrieved text. Deterministic matching involves linking the facts of the case with the configured elements; case only search allows checking the source. Output: Findings and evidence links to consider. No references and no evidence are not regarded as mappings; blocking of case context prevents finding. Matching does not constitute professional legal advice.

Stage 9: Population of the Charge Sheet. Input: Charge sheet population. Input: Case background, results and disposition. The type-based IF-5 report formulation approach is verified and continues through the use of ReportLab; PDF creation can be done where applicable. Output: Structured data and charge sheet draft/ log where applicable. Errors at the origin level affect population; blocking at the end prevents PDF generation and review status.

4.3 Knowledge Graph and Canonical Case Representation

The investigation graph is an intermediate model that is typed and case-specific. The node types comprise person, vehicle, evidence, event from the timeline, document, medical observation, forensics science observation, recovery item, and location. People can have roles such as complainant, victim, accused, witness, police, and doctor. Relations include SUPPORTED BY, ASSERTED BY, RELATES TO, TREATED AT, DRIVES, OWNS, RECOVERED FROM, COLLECTED BY, EXAMINED BY, PART OF, MENTIONS, temporal order relations, and CONTRADICTS. Graph building employs only those data that are surfaced by the parsing models; relations do not indicate any validation of the underlying statement.

Provenance data accompany each graph edge and include document ID, optional page, parser name, extraction certainty if present, processing date-time stamp, and a source reference

object. Entity resolution is based on deterministic normalization of compatible identity expressions. It is not an identity resolution by probability, and conflicts may stay unresolved. The graph thus represents a relational index of extracted statements, not an authoritative log of events.

Canonical facts map graph facts to structures at the case level such as FIRs and registration metadata; jurisdiction, police station, and court; offence facts; person groups by type; vehicles and places; documentary and physical evidences; medical and forensic reports; timeline events; investigative actions; seized items; document metadata; conflicts; and missing information facts. Canonical facts include their values, document IDs, source references, source paths, extraction processes, confidence, and timestamps if available. Conflicts in canons have different facts and their sources, along with their status and resolutions if available. This is an inferentially safe projection that allows field mappings without loss of document-level provenance.

For example, the node representing the witness-statement document can have an edge MENTIONS pointing to a person node with the role WITNESS, and the edge ASSERTED BY will maintain the document as the source of the assertion. Another node created by parsing the FIR occurrence will be a timeline event node linked to the FIR document using SUPPORTED BY; the FSL examination will generate an evidence node connected to its report using the pre-configured relation for examination. These will be case-specific and source-attributed relations and not the implied cause-effect relationship as the graph will not create any person-to-event or event-to-evidence relationship unless provided by the registered mapper.

4.4 Evidence Provenance and Validation Dispositions

Provenance is propagated from document and parser output through graph nodes, canonical facts, case context, legal findings, and charge sheet fields. The pointer to the source could refer to a document, page number, or parser field path, but there is no mechanism that guarantees verified character offsets or universally aligned passages for all fields. Thus, the provenance of this system is valuable for review purposes but cannot guarantee authenticity, completeness, legality, chain of custody, , or admissibility.

Validation is read-only with regard to canonical case facts. The validations check for

the presence of configured required fields, configured expected document types, configured source support, provenance, conflicts, selected timeline conditions, entities, unsupported facts, and confidence. The FIR number, police station, and occurrence information are optional by default; if configured as required, missing values will result in the configured severity (ERROR by default). The missing information records have warning severity. Low confidence warnings are applicable only for those facts that have valid source references and confidence strictly less than the default threshold of 0.4. Missing document checks are performed only when a medical finding, forensic finding or recovered property is present without the corresponding medical report, FSL report, or seizure memo; the default severity for that is WARNING. Validator rules can be overridden. The severity of the warnings itself is not enough for review, but there is an additional condition for conflicting facts and entities. The reports contain the issue categories, severities, source references, and completeness value.

The disposition of an issue by the application is divided into three categories (see Table 2). When the issue falls under the CRITICAL category, the disposition is FINAL BLOCKED. However, for any other case where there are errors, conflicts, times, or entities that are yet to be resolved, the disposition is REVIEW REQUIRED. All other cases will be covered by DRAFT ALLOWED. It should be noted that the disposition of warning alone does not require review, but in the case of any conflict or entity, it must,even when it is a warning. The disposition affects only the output and the review, not the legal adequacy of the document, and not FINAL BLOCKED.

Table 2. Validation dispositions and their charge-sheet effects.

Disposition	Implemented Trigger, in General	Workflow Consequence
DRAFT_ALLOWED	No critical issue, error, unresolved conflict, timeline issue, or unresolved entity. Other warnings alone do not change this disposition.	Structured data is written; draft PDF and initial review state are created. Human review remains necessary.

REVIEW_REQUIRED	No critical issue; at least one error, unresolved conflict, timeline issue, or unresolved entity. Warning-level conflicts and entities also trigger review.	Draft PDF and review state are created with review flags and source-linked issues.
FINAL_BLOCKED	At least one issue is classified as critical.	Structured data is retained; downstream stages continue, but draft PDF and initial review state are suppressed.

4.5 Context of Case, Retrieval, and Legal Provision Mapping

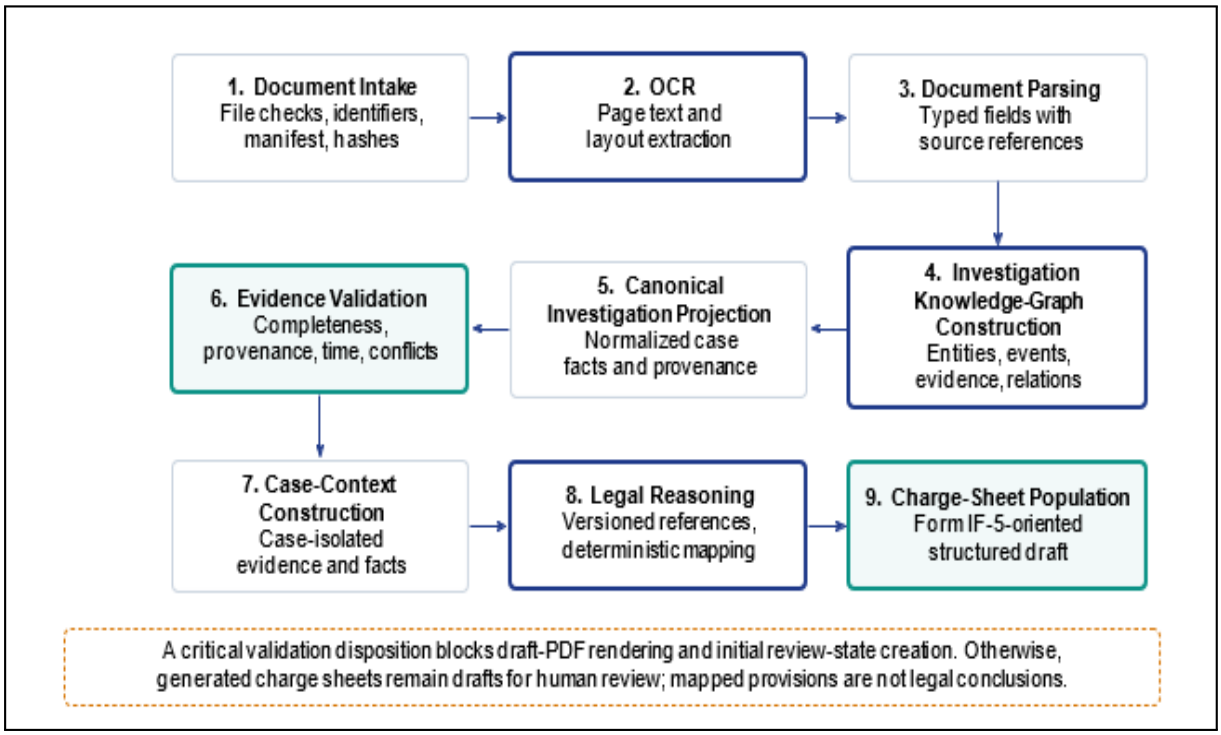


Figure 3. Evidence provenance and validation disposition gate.

The context stage provides the facts, evidence, incidents, validation issues, and sources to the subsequent stages. This is a deterministic composition of information, but not a credibility assessment. The retrieval stage processes the OCR'd text, along with case, document, and page metadata, applying ranking based on the text, in a manner similar to

BM25. Similarity by dense vectors can be computed if an embedding service is available, while a no-op service enables lexicon-based search without embeddings. The results can be used to examine the evidence through the reviewer API. Legal reasoning utilizes a versioned local reference file and deterministic matching on configured statutory elements. The snapshot being reviewed and labeled as bns-2023-2024-07-01 consists of seven records: BNS sections 106, 115, 125, 281, and 324; BNSS section 193; and BSA section 63 [13, 14, 15]. The file indicates that all records are effective as of July 1, 2024. Code-level version-tag matching prevents the problem of configured-version mismatch but does not provide legal accuracy, completeness, or ongoing relevance. However, the above list is not a complete database for the laws in India. This system makes it possible to match any clause with the description of the evidence and also the demonstration of the evidence as either corroborative, contradictory, relevant, or non-existent. This will depend on the quality of the reference text and evidence. Legal matching does not imply element satisfaction, provision applicability, or need for charge filing; a qualified legal review is required. Figure 4 shows the legal reference association during the creation of the draft, while Figure 5 shows human checking of the created draft.

4.6 Form IF-5-Oriented Charge-Sheet Generation

Preparation of a charge sheet converts the process of canonical investigation and findings into an electronic data contract format specifically targeted toward Form IF-5. The schema divides fields pertaining to case identification and administration (case number, police station, court) from narrative fields (case summary, case details, investigation conducted, evidence evaluated, opinion, signature), and tabular fields (complainant, victim, accused, witness, timeline events, documents/physical evidence, and annexure). The model also captures findings related to medical, forensics, and vehicles, statutory sections, structured legal findings, conflict, and missing data. The model connects the created charge sheet form to the same entities and evidences as used for validation and review. The internal data contract is specified as a Form IF-5 oriented; the inspected PDF document uses a generic label of “Charge Sheet” without actually printing “Form IF-5” on it.

Table 3 lists the fields as specified in the ChargeSheetData schema. IF5Row is used for repetitive data with arguments for serial number, description, and exhibit; the values of description and exhibit are provenance-sensitive ChargeSheetField objects. The narrative text

is extrapolated only based on the existing context; legal sections are linked based on legal determination, and any final opinion/signature text is not present unless available.

Table 3. Field groups in the implemented charge-sheet data model.

Field group	Schema fields
Case and court	case_id, case_number, police_station, court, version, and disposition.
Narrative	case_summary, detailed_facts, investigation_conducted, and evidence_analysis.
People and chronology	Repeated complainants, victims, accused, witnesses, and timeline rows.
Evidence and findings	documentary evidence, material evidence, medical findings, forensic findings, vehicle findings, and annexures.
Legal and review	legal sections, legal findings, conflicts, missing information, final opinion, and signature.

Each charge-sheet field contains value status, confidence, optional sources, a review status flag, and origin metadata; reviewer corrections can include the previous value, reviewer identity, timestamp, and correction notes. A populated field must contain sources, while unavailable, not found, extraction failure, and not applicable statuses may not have unsupported values. Fields that are conflicted or need review remain so. These invariants ensure no silent completion is possible, although they do not guarantee that a referenced record semantically justifies the field; the reviewer must validate the source.

At this stage, the structured model is first populated and validated to produce a JSON artifact. If the state is DRAFT ALLOWED or REVIEW REQUIRED, the ReportLab renderer produces a paginated draft PDF and the review-state artifact. If case the state is FINAL BLOCKED, only the structured record is produced, and the draft PDF and the review-state generation process is avoided. The PDF produced is an unsigned review document and not a charge sheet that is a legally valid document. Figure 4 represents the field-population and generation gate.

4.7 Reviewer Workflow and API

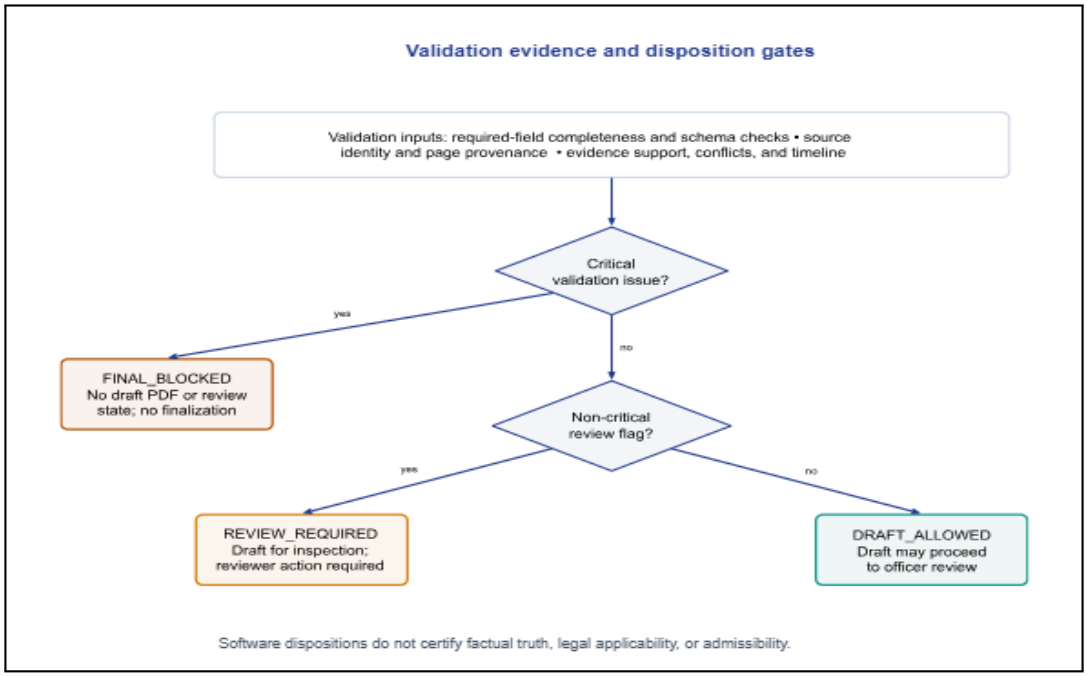


Figure 4. Automation of generating charge sheet from validated case data.

The review life cycle starts with a case and the most recent charge sheet revision. The reviewer gets the preview and completeness details, performs a search on the evidence passages in the case, checks document/page provenance, takes notes, and performs corrections, where needed. A correction creates a new version and invalidates the previous approval so that any decision is based on the exact reviewed information. The reviewer is capable of approving, rejecting, requesting a revision, and finalizing based on the state and validation guards. Following the revision, the new data goes through a revalidation process and the decision life cycle repeats itself; finalization can only happen once both version and review state meet the service guards.

Table 4. Reviewer API operations in the inspected FastAPI implementation.

Method and path	Operation	Review effect
GET /	Retrieve review data	Returns current fields, completeness, provenance, and review state.

POST /corrections	Correct fields	Creates a traceable new version and invalidates earlier approval.
POST /notes	Add reviewer note	Records an observation in the case review history.
POST /decision	Record decision	Accepts approval, rejection, revision request, or finalization subject to state guards.
GET /history	Read history	Returns chronological version and decision events.
GET /retrieval	Search evidence	Returns case-isolated passages with document/page provenance.

Table 4 shows the router that we developed. Routes in this router are based on /api/v1/cases/<case id>/review. This router has some service methods for review data, changes, decisions, history, and evidence. The authentication module has been developed separately using the in-memory bearer token sessions. But there is no dependency on any authentication in the reviewed route definition.

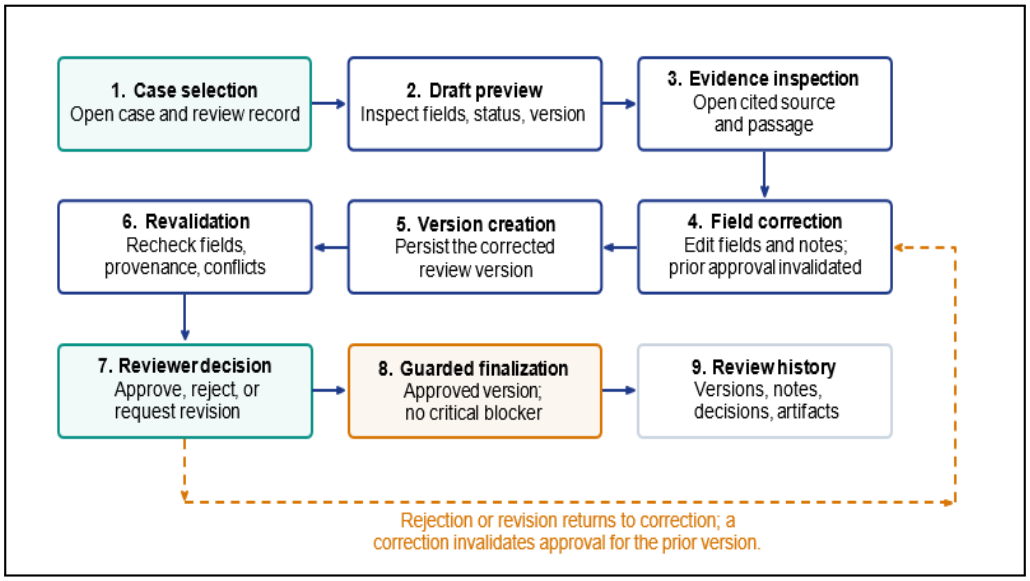


Figure 5. Human reviewer lifecycle with evidence inspection and guarded finalization.

The API provides structured preview and correction mechanisms, but it does not indicate its usability, authorization of the officer, or the correctness of the reviewer's decision.

The review actions need to be assessed on stale version handling, audit completeness, access control, revalidation after correction, and the ability of the reviewer to locate the source document. Figure 5 shows the state transition and finalization boundaries.

5. Experimental Setup and Results

5.1 Reproduction Conditions and Measurement Boundaries

Project records describe a 19-document synthetic case, but the source PDF files were unavailable to repeat the measurement in the context of the current evaluation. Thus, the fixture inventory and text extraction measurements given below are the previous locally recorded data, not the measurements reproduced here. The previous record states 19 PDF documents, 100 pages, 130,914 characters extracted, and 0.358 seconds traversal time in process with pypdf. This time does not include OCR, model invocation, workflow processing, rendering, and reviewer operations, but just the extraction of embedded digital text following package initialization. These numbers should be reproduced on the actual fixture set before being considered current independently verified results.

A backend test execution command was tried previously, but the test collection finished with 17 failures and 0 tests run. This included a missing ReportLab requirement and unresolved imports in the remaining tests. The frontend tests were not performed. No backend or frontend test result is reported. The entire document-to-PDF workflow was not repeated due to the absence of runtime requirements and PDF documents in this environment. Project presentation statistics are thus provided separately from fixture measurements and test collection diagnostics.

5.2 Reported and Previously Recorded Results

Table 5 distinguishes each reported value, its measurement scope, and whether it has been verified. Time metrics are separate: the local embedded text traverse, reported extraction time period, reported nine-layer API extraction process, upload to finalized PDF time period, and described below mock mode run are distinct scopes and cannot be compared directly. A confidence measure does not equate to accuracy, test numbers do not imply reproduction, and an adversarial attack generated synthetically is not robustness.

Table 5. Previously recorded and project-reported measurements with evidence status.

Measure	Value and boundary	Evidence status and limitation
Synthetic fixture inventory	19 PDFs; 100 pages	Previously recorded from the synthetic case; source PDFs are not present for rerun in this workspace.
Embedded-text extraction	130,914 characters; 0.358 s in-process	Previously recorded local PDF traversal; not OCR or end-to-end latency; not rerun here.
Extraction interval	19.9 ms reported	Project-presentation metric; operation boundary and logs are unavailable.
Nine-stage API pipeline	0.72 s, mock mode	Project-presentation metric; reported as workflow API timing, not reproduced.
Upload to finalized PDF	1.15 s	Project-presentation metric; broader boundary than stage-only timing; not reproduced.
Later offline mock run	1.8–2.3 s	Later project-reported range; precise timing boundary and raw logs unavailable.
RAG index	100 chunks	Project-reported inventory; chunking and deduplication logs unavailable.
Extraction confidence	0.99 (99%) mean	Project-reported; aggregation and calibration are unspecified. This must not be interpreted as extraction accuracy.
BNS mapping evaluation	No benchmark result available	No evaluation dataset, denominator, or ground truth was available; the presentation's Section 106 example is qualitative, not an accuracy evaluation.

Automated tests	216 passed; 2 optional/live skipped	Project-reported total (187 backend, 29 frontend); not reproduced in this environment.
Adversarial scenarios	9 successful scenarios	Project-reported; test definitions and raw results unavailable for independent audit.
Parser inventory	13 specialized parsers plus fallback	Source inventory; does not imply equal parser specialization or measured accuracy.
Statutory references	7 records; bns-2023-2024-07-01	Local-file inventory only; tag matching does not establish legal accuracy, completeness, or continuing currency.
Backend test attempt	17 collection errors; 0 tests executed	Previously attempted local run; dependency and import failures prevented test execution.
Frontend test attempt	Not run	No frontend test result is claimed as reproduced.

The Extraction Value at Local Level is relevant only to the embedded text. There is no measure of OCR quality based on the scanned documents. There can be no comparison of the extraction time of 19.9 ms and Traversal Time of 0.358 sec for the entire fixture without having the same definition and log. The extraction times of 0.72 sec for the mock pipeline and 1.15 sec for the uploading process to finalized PDF are taken from different start and end points. The 1.8-2.3 seconds range is kept separately since there was no time protocol for the case. No time value is used for deriving the service-level objective. The Section 106 illustration is qualitative; no benchmarking for the BNS-mapping accuracy is available. Figure 6 illustrates the evidence boundary.

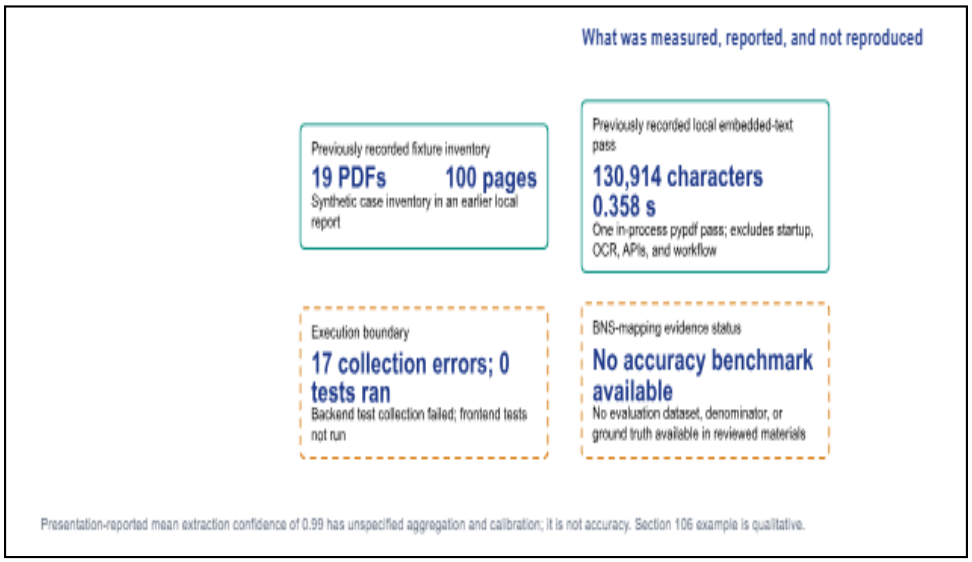


Figure 6. Experimental results and evidence status.

6. Discussion and Limitations

This implementation combines document intake and extraction with graph and canonical case data, configurable validation, bounded legal-reference mapping, draft generation, and reviewer API operations. Dismantling the connection between model-based extraction and deterministic projection and validation results in greater inspective capacity, whereas case- and page-specific lookup and provenance of fields facilitates source verification. This is an aspect of the implementation and does not reflect the amount of time saved, the accuracy, or better decisions made by the investigator. The mean confidence score of the extraction process, recorded at 0.99 (99%) in this project, has not been normalized.

Validation will be constrained by rule encoding where data that may seem valid can still turn out to be invalid, errors may not be detected, and the lack of document type can have significant meaning. Hash codes do not prove ownership or admissibility, phrase matching with statutory texts does not provide legal proof, and search functions do not prove that all evidence has been collected. The statutory reference index created by the seven documents is incomplete, and the version number is not proof of current legality.

React, Neo4j, PostgreSQL, Supabase, and signed JWTs are target architecture statements that have yet to be validated as active integrations in this snapshot of the source. Artifacts in file systems, in-memory bearer token sessions, lack of source code for the front-end

application, and review routes without any dependencies on authentication make drawing conclusions about implementation and security highly speculative.

The generation of the synthetic fixtures could not be done because PDF files were unavailable. Previously done text extraction could not validate the accuracy of the optical character recognition, or the relevance, provenance, completeness, usability, and latency of retrieval. The timings and test totals have no logs and cannot be reproduced. There is no accuracy benchmark for the BNS mapping, evaluation data set, denominator, or ground truth; the provided Section 106 example is merely qualitative.

7. Future Work

Future evaluation needs to consider BNS mapping on an authorized corpus with adjudicated labels and a stated denominator. The error rates of OCR, field extraction, source-reference accuracy, and legal mapping need to be assessed independently; a confidence of 0.99 cannot replace labeled accuracy assessment. The process involving 19 documents needs to be replicated in a dependency-pinned environment in both mock and authorized model-backed modes, with boundary conditions, startup, calls to the model, failures, artifacts, and review states logged.

Human factor studies need to address source discovery, conflict detection, correction, and the capability of the reviewer to differentiate between suggested provisions and legal conclusions. Security aspects require securing identity, roles, and review routes, and assessing privacy, retention, and external model handling. The architecture of the data store used as a target needs to be stated only after the development of adapters, access control, persistence, and recovery has been completed.

8. Conclusion

Byomkesh AI organizes investigation records through nine steps relating to OCR and parsing guided by a schema, followed by graph and canonical representation, evidence validation, bounding statutory reference mapping, and Form IF-5-based drafting. Provenance and status in the field make the review possible, while validation dispositions make the PDF and review states formable. Case-specific evidence extraction and FastAPI operations

provide the means of evidence retrieval, correction, notes, decision-making, historizing, and conservative finalization. The present storyline distinguishes the workflow capacities certified by the repository from the broader target architecture. There is no benchmark for measuring the BNS mapping or even its accuracy, which could have been done on the basis of the above information, since the Section 106 case study is purely qualitative in character. The earlier fixture measures were not recomputed due to the absence of the PDF inputs.

Acknowledgements

The project presentation describes the project as one that has been created following consultations with the senior police authorities and IT staff of the Navi Mumbai Police Commissioner's Office and describes the corpus created as one that is police-centric and based on investigative practices and document data. The acknowledgment section refers to the above process of consultation, but does not make any references regarding implementation and collaboration with institutions.

Declaration of Generative AI and AI-Assisted Technologies

OpenAI Codex assisted with language revision, source-code inventory, and creation and refinement of the TikZ figures. It was also helpful in developing a process employed for a previous extraction of embedded text from artificial PDF structures. Project-reported metrics are identified as such, and the authors remain responsible for verifying the reported values, source-derived descriptions, and final manuscript.

References

[1] Mathew, Minesh, Dimosthenis Karatzas, and C. V. Jawahar. "Docvqa: A dataset for vqa on document images." In Winter Conference on Applications of Computer Vision (WACV),IEEE, 2021,pp. 2199-2208.

[2] Huang, Yupan, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. "Layoutlmv3: Pre-training for document ai with unified text and image masking." In Proceedings of the 30th ACM international conference on multimedia,20222, pp. 4083-4091.

[3] Appalaraju, Srikar, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. "Docformer: End-to-end transformer for document understanding." In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021, pp. 973-983.

[4] Neudecker, Clemens, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. ”A survey of OCR evaluation tools and metrics.” In Proceedings of the 6th International Workshop on Historical Document Imaging and Processing, 2021, pp. 13-18.

[5] Chalkidis, Ilias, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. ”LexGLUE: A benchmark dataset for legal language understanding in English.” In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 4310-4330.

[6] Joshi, Abhinav, Shounak Paul, Akshat Sharma, Pawan Goyal, Saptarshi Ghosh, and Ashutosh Modi. ”IL-TUR: Benchmark for Indian legal text understanding and reasoning.” In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 2024, vol.10, pp.11460-11499.

[7] Paul, Shounak, Pawan Goyal, and Saptarshi Ghosh. ”Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents.” In Proceedings of the AAAI conference on artificial intelligence, 2024, vol. 36, no. 10, pp. 11139-11146.

[8] Shukla, Siddharth, Tanuj Tyagi, Abhay Singh Bisht, Ashish Sharma, and Basant Agarwal. ”NyayGraph: A knowledge graph enhanced approach for legal statute identification in Indian law using large language models.” In Proceedings of the Natural Legal Language Processing Workshop, 2025, pp. 147-156.

[9] Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler et al. ”Retrieval-augmented generation for knowledge-intensive nlp tasks.” Advances in neural information processing systems 2024, vol.33 pp. 9459-9474.

[10] Gao, Tianyu, Howard Yen, Jiatong Yu, and Danqi Chen. ”Enabling large language models to generate text with citations.” In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2023, pp. 6465-6488.

[11] Nushi, Besmira, Shamsi T. Iqbal, Paul Bennett, Jaime Teevan, Eric Horvitz, Dan Weld, Ruth Kikin-Gil et al. ”Guidelines for human-AI interaction.” In Proceedings of the CHI Conference on Human Factors in Computing Systems. 2019, no.3 pp. 1-13.

[12] Klie, Jan-Christoph, Richard Eckart De Castilho, and Iryna Gurevych. ”From zero to hero:

Human-in-the-loop entity linking in low resource domains." In Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 6982-699.

[13] Government of India. The Bharatiya Nyaya Sanhita, 2023. Act No. 45 of 2023, enacted December 25, 2023.

[14] Government of India. The Bharatiya Nagarik Suraksha Sanhita, 2023. Act No. 46 of 2023, enacted December 25, 2023.

[15] Government of India. The Bharatiya Sakshya Adhiniyam, 2023. Act No. 47 of 2023, enacted December 25, 2023.