Below is a “prompt-playbook” you can feed to Claude-Opus (or any capable LLM) to design and iteratively build the living EEO-case document system. Each numbered item is a \*\*stand-alone prompt\*\* you can copy-paste, followed by a brief rationale for why you would ask it next. Use or skip pieces as your project evolves.

---

1. \*\*Project Charter & Constraints\*\*

Prompt:

\*“You are my legal-tech co-pilot. Summarize, in 1 page, the ultimate goals, legal constraints (EEO, FOIA, Privacy Act), and non-functional requirements (security, auditability, chain-of-custody) for a ‘living’ EEOC case-management repository.”\*

Why: Forces the model to surface guard-rails before any code is written.

2. \*\*High-Level System Architecture\*\*

Prompt:

\*“Propose a modular architecture (components, data flow, storage choices) to ingest thousands of PDFs, Word docs, and e-mails, enrich them with metadata, and serve search/analytics dashboards. Show as a markdown diagram.”\*

Why: Gives you an architectural map that later prompts can drill into.

3. \*\*Document Ingestion Pipeline Design\*\*

Prompt:

\*“Detail an ingestion pipeline: (a) OCR, (b) text extraction, (c) checksum + versioning, (d) initial metadata capture (file name, dates, custodians). Provide pseudocode only.”\*

Why: Breaks the hardest step—getting clean text—into explicit stages.

4. \*\*Metadata & Schema Definition\*\*

Prompt:

\*“Draft a JSON schema for each stored document that captures legal attributes (Bates number, EEO category, protected class, exhibit status) and technical attributes (hash, source path, ingestion date).”\*

Why: Locking in schema early prevents future refactors.

5. \*\*Keyword & Entity Extraction Strategy\*\*

Prompt:

\*“Recommend named-entity and key-phrase extraction approaches specific to EEO litigation (e.g., ADA terms, accommodation vocabulary). Include open-source libraries and accuracy trade-offs.”\*

Why: Ensures domain nuance (EEO jargon) is captured, not just generic NLP.

6. \*\*Automated Tagging Prompt Template\*\*

Prompt:

\*“Write a reusable LLM prompt template that, given raw text + the JSON schema, returns: (1) validated metadata, (2) issue tags (retaliation, disability, timeliness), (3) confidence scores.”\*

Why: Gives you a canonical way to call Claude for tagging over thousands of docs.

7. \*\*Indexing & Search Design\*\*

Prompt:

\*“Compare three indexing options (Elasticsearch, Typesense, OpenAI vector DB). Rate each on speed, legal hold compatibility, and cost. Recommend one.”\*

Why: Lets the model weigh full-text vs. vector search under legal-hold requirements.

8. \*\*Duplicate & Near-Duplicate Detection\*\*

Prompt:

\*“Describe algorithms (perceptual hashing, MinHash) to detect duplicates in the corpus, flag the canonical copy, and record linkage in the JSON schema.”\*

Why: Prevents ballooning storage and confusing duplicates in discovery.

9. \*\*Timeline Generation Logic\*\*

Prompt:

\*“Given a set of date-stamped documents + tags, outline an algorithm to auto-build an interactive case timeline, surfacing conflicting dates and missing evidence.”\*

Why: Connects raw docs to the litigation narrative (crucial for ADR & hearing).

10. \*\*Policy & Directive Cross-Referencing\*\*

Prompt:

\*“Explain how to automatically map extracted text to FEMA/EEOC policies, directives, CFR sections, and GAO statistics. Suggest a look-up table or embedding-based approach.”\*

Why: Links facts to governing rules, enabling quick legal relevance checks.

11. \*\*Statistical Benchmarks Module\*\*

Prompt:

\*“Design a module that, given agency-wide EEO Form-462 data, compares the complainant’s stats to national averages (e.g., accommodation grant rates). Specify data sources, normalization steps.”\*

Why: Supplies context for damages and pattern-or-practice arguments.

12. \*\*Analytics & Dashboard Requirements\*\*

Prompt:

\*“List the top 10 dashboard widgets an attorney/ADR mediator would want (e.g., unanswered accommodation days counter, retaliation timeline heat-map). For each, name the underlying queries.”\*

Why: Drives front-end and API priorities directly from end-user value.

13. \*\*Chain-of-Custody & Audit Trail\*\*

Prompt:

\*“Propose features to maintain chain-of-custody: write-once S3 buckets, tamper-evident logs, user action audit tables. Sketch a logging schema.”\*

Why: Demonstrates due diligence—crucial if evidence is challenged.

14. \*\*PIP & Fabrication Detection Safeguards\*\*

Prompt:

\*“Suggest automated tests that scan for fabricated events (e.g., PIP with no source doc). Outline how to surface warnings in the UI.”\*

Why: Avoids repeating the earlier PIP hallucination problem.

15. \*\*Verification & Human-in-the-Loop Plan\*\*

Prompt:

\*“Draft a workflow where tagged documents are queued for paralegal review, approved, or corrected, with feedback fed back to the model. Provide swim-lane diagram.”\*

Why: Marries AI automation with human validation for legal reliability.

16. \*\*Incremental Roll-Out Roadmap\*\*

Prompt:

\*“Break the build into 4 sprints: ingestion MVP, tagging/duplicate detection, timeline + dashboards, compliance hardening. Assign deliverables & acceptance criteria to each sprint.”\*

Why: Gives a realistic implementation path and checkpoints.

17. \*\*Prompt Engineering Best Practices\*\*

Prompt:

\*“List best practices (system vs. user prompts, JSON mode, max tokens, temperature) for calling Claude in batch document processing, including rate-limit handling.”\*

Why: Reduces hallucinations and cost overruns during large-scale runs.

18. \*\*QA & Regression Testing Strategy\*\*

Prompt:

\*“Outline automated tests (unit, integration, regression) to ensure ingestion, tagging, and search still work after code changes or model updates.”\*

Why: Sets up long-term maintainability.

19. \*\*Risk & Compliance Checklist\*\*

Prompt:

\*“Create a checklist of regulatory risks (PII exposure, HIPAA, privileged material) and link each to a technical safeguard.”\*

Why: Ensures security/legal compliance are baked in, not an afterthought.

20. \*\*Success Metrics & KPI Definition\*\*

Prompt:

\*“Define measurable KPIs: ingestion throughput, tagging accuracy, search latency, attorney adoption rate. Suggest instrumentation points.”\*

Why: Lets you quantify success and justify further investment.

---

### How to Use This List

1. \*\*Start with #1–3\*\* to lock scope and architecture.

2. \*\*Feed prompts sequentially\*\*—each answer becomes context for the next prompt.

3. \*\*Store every Claude response\*\* in version control; they become living design docs.

4. \*\*After design is stable\*\*, shift to implementation prompts (e.g., “Generate Python code for the OCR module described in step 3”).

5. \*\*Iterate\*\*: when real documents expose edge-cases, ask Claude for targeted fixes (“We have TIFF images with sideways text—update the OCR plan”).

By front-loading design and verification questions, you minimize costly mid-project rewrites and keep the AI focused on legally sound, auditable solutions. Good luck building the next generation EEO case-management system!