RnD Knowledge & Dataset Hub

Title

RnD Knowledge & Dataset Hub — Secure Repository with Hybrid (Metadata + AI) Search, Versioning, and Admin Observability

Summary

Build a secure, enterprise repository for RnD teams to **upload, tag, version, search, edit, and share** documents and datasets. Replace ad-hoc storage (local PCs, random ADLS, SharePoint folders) with one governed platform. Provide **hybrid search** (metadata filters + semantic/content search via embeddings), **role-based access**, **end-to-end audit**, **session/activity analytics**, and **Azure AD** SSO. Ship an **admin console** to monitor logins, actions, storage usage, errors, and policy drifts.

Objectives / Goals (what success looks like)

#	Objective	Acceptance / KPIs
1	**Unified repository**: upload/download/preview; folders & projects; rich metadata & tags	Upload ≤150 MB/file; preview PDFs/Docs/Images; tag & bulk-tag; soft delete + restore
2	**Hybrid search**: keyword + filters (type/owner/tags/date) + **semantic search** over content	<1.5 s P95 query latency; top-3 semantic results relevant on seed set ≥85%
3	**Versioning & edit flows**: automatic versions, compare, rollback; optional check-in/out	Version history retained; diff/compare for text; rollback in 1 click
4	**Security & access**: Azure AD SSO, groups/roles (Admin, Maintainer, Contributor, Reader), signed download links	Zero secrets in code; all secrets in Key Vault; least-privilege enforced

5	**Compliance & safety**: antivirus scan, file-type allowlist, PII detection flags, retention policies	100% files scanned; blocked types rejected; PII flags visible & filterable
6	**Observability**: Admin dashboard for sessions, logins, active users, actions, storage, errors	Real-time tiles; 30-day trends; export to CSV
7	**Audit trail**: who did what & when (upload, edit, share, delete, policy changes)	Immutable event log; export & filter by user/resource/action
8	**Scalability & cost**: support 10K users / 5M files; lifecycle policies & cold storage	Storage class policy saves ≥20% on non-hot items in backtest
9	**Developer ergonomics**: clean APIs, IaC, CI/CD, seed data	One-click deploy to Azure; Postman collection; sample dataset & labels

Scope

In-Scope

- File mgmt (upload, rename, move, copy, delete/restore), previews, bulk actions
- Project/folder hierarchy; shared links with expiry; per-item ACLs
- Metadata (title, owner, description, tags, domain, confidentiality)
- **Search:** keyword + filters + semantic (embeddings) + “search within file”
- **Datasets:** CSV/Parquet upload, schema view, sample preview
- Version control & compare/rollback
- **Admin Console:** sessions, active users, login history (“who logged in last night”), actions by user/time, storage usage, audit export
- **Governance:** AV scan, file allowlist, PII flagging, retention (keep/hold), legal hold marker
- **Notifications:** share invites, @mentions/comments, failed scan alerts

Out-of-Scope (v1)

- Desktop sync client; complex DLP workflows; multi-tenant billing

Reference UX (from screenshots you shared)

- **Dashboard:** totals (files, storage used, shared files, team members), recent files list with tags and owners

- **Library/Search:** card view + list view, filters, recent searches
- **Admin:** session logs, who is logged in, who did what, when

Architecture (Azure-first, AI-agnostic)

Frontend

- React + TypeScript (Next.js recommended), Tailwind UI
- Auth via **Azure AD (Entra ID)** → MSAL.js
- File preview: PDF.js, image viewer, markdown viewer, simple CSV table

APIs / Services

- **FastAPI (Python)** or **Node/Express** for REST
- Async workers: **Azure Functions** / **Azure Container Apps Jobs** for ingestion & scanning
- **RBAC & Policies:** enforced in API; signed SAS for downloads

Data & Storage

- **ADLS Gen2** (hierarchical namespaces) as primary file store
- **Azure SQL** or **PostgreSQL Flexible Server** for metadata, ACLs, audit, sessions
- **Search: Azure AI Search** with **vector + BM25 hybrid** (embeddings via **Azure OpenAI** or OSS model)

- **Embeddings index**: AI Search vectors or **pgvector** (if Postgres)
- **Redis** for query cache & recent lists
- **App Insights + Azure Monitor** for telemetry

AI Processing

- **Embeddings:** text-embedding-3-small/large (or OSS), chunked with overlap (e.g., 800–1,000 tokens, 15% overlap)
- **OCR** for scanned PDFs/images: Azure Document Intelligence (Form Recognizer)
- **PII detection:** Presidio / Azure Content Safety (categories only, no raw content retention)
- **AV scan:** ClamAV in an isolated Function; block on detection

Security

- SSO with **Azure AD**, Entra ID groups → roles

- Storage encryption at rest; TLS in transit; **Key Vault** for secrets
- Signed URLs with short TTL; download watermarking for sensitive docs (optional)
- Audit log is **append-only** (e.g., Event Hubs → ADX/Log Analytics)

CI/CD & IaC

- **GitHub Actions** pipelines (lint, test, build, infra plan/apply, deploy)
- **Terraform** modules for ADLS, AI Search, DB, Key Vault, App insights, Functions, vNet

Core Features & Flows

A) Upload & Ingest

- Client → SAS upload → ADLS
- Ingest worker: virus scan → extract text (Tika/unstructured) → OCR if needed → chunk → embed → upsert into AI Search
- PII detection → store flags; raise review if high-risk

- Create file_versions entry; update metadata

B) Search

- Query pipeline: keyword → spell-correct → hybrid search (BM25 + vector) → re-rank
- Filters: type, tags, owner, project, date, “has PII”, size, extension
- Result card: title, snippet, tags, owner, updated, relevance; “open”, “preview”, “download”

C) Versioning & Edit

- New upload on same file = new version; show diff for text/markdown
- Optional check-out lock; comments & mentions

D) Governance

- Blocked types (exe/bat etc.) → reject
- Retention policy per project (e.g., 2 years, legal hold)
- Signed links with expiry; revoke shares
- Watermark sensitive previews (optional)

E) Admin Console

- Tiles: Total files, storage used, shared files, active users
- Tables: recent files; **sessions** (who logged in, last night), actions by user, errors
- Trends: uploads/day, downloads/day, search volume; failed scans
- Exports: audit CSV; user activity

APIs (sample)

- POST /api/files/upload-url → { sasUrl, headers }
- POST /api/files/complete → finalize, trigger ingest
- GET /api/files/:id / GET /api/files/:id/versions
- POST /api/files/:id/tags · DELETE /api/files/:id
- POST /api/search → { query, filters } → results[]
- GET /api/admin/sessions · GET /api/admin/audit · GET /api/admin/stats
- POST /api/shares → grant; DELETE /api/shares/:id
- GET /api/policies → retention, blocked types

Non-Functional Requirements

- **SLOs:** P95 search <1.5 s; upload finalize <10 s; embed job start <30 s
- **Scale:** 5M files, 50M chunks; AI Search partitioning plan included
- **Cost:** use compression, cold tiering; batch embeddings (512–2,048)
- **Privacy:** no LLM retention; redact sensitive snippets in logs
- **Accessibility:** WCAG-AA for UI; keyboard shortcuts

Deliverables

1. **Frontend** app (React/Next) with all user/Admin screens
2. **Backend** APIs (FastAPI/Express) + workers (Functions/Container Apps)
3. **Search pipeline** (chunking, embeddings, hybrid search)
4. **Security**: Azure AD SSO, RBAC, signed links, Key Vault integration
5. **Compliance**: AV scan, PII flagging, allowlist, retention
6. **Admin console**: sessions, actions, storage, audit export
7. **IaC & CI/CD**: Terraform modules + GitHub Actions
8. **Docs**: Architecture, API contract, runbook, cost model
9. **Seed data**: sample PDFs/CSVs; labeled queries; relevancy tests
10. **Demo script**: upload → scan → search → version → admin view

Evaluation Criteria (for interns)

Weight	Area	What we’ll check
30%	Functionality coverage	All core flows work; hybrid search quality; versioning; governance
20%	Security & compliance	AAD SSO, RBAC, Key Vault, AV scan, PII flags, retention
20%	Performance & scalability	SLOs met; bulk ingest; index design; cost controls
15%	Observability & admin UX	Clean dashboards; useful filters; exports
10%	Code quality & DevEx	Tests, docs, API clarity, IaC/CI
5%	Stretch	Watermarking, SharePoint/ADLS import, query analytics

Stretch Goals (optional)

- **Connectors**: one-time import from SharePoint/OneDrive/legacy ADLS paths