Software Requirements Specification Sheet (SRSS)

Project Codename: LYNX (Linked Knowledge Network Explorer)

Version: Final Draft

Date: September 26, 2025

1. Purpose

LYNX is a semantic mapping and visualization platform designed to explore knowledge as a zoomable galaxy. It supports semantic search, shortest path queries, clustering, wormhole discovery, and exports. Built for ~$20/month, it demonstrates full-stack engineering, embeddings, graph theory, and visualization expertise at a professional R&D; level.

2. Goals and Non-Goals

Goals:

- Interactive galaxy-like visualization of 50k–250k knowledge concepts.
- Semantic search, cross-domain wormholes, pathfinding, clustering, and exports.
- Public demo with read-only access; private maps with user auth.
- System cost capped at ~$20–25/month.

Non-Goals:

- Multi-language NLP beyond English.
- Real-time ingestion of all news feeds.
- Full replication of Wikipedia or arXiv.

3. Assumptions and Constraints

- Infrastructure capped at low-cost Supabase + Vercel stack.
- Embeddings via OpenAI text-embedding-3-large with batch processing.
- MVP dataset: ~50k concepts, scalable to 250k.
- Single-region hosting, desktop-first design.

4. Functional Requirements

Data Ingestion:

- Wikipedia summaries (English).
- arXiv abstracts (scientific domains).
- Optional news headlines (NewsAPI/GNews).

Embeddings:

- Generated with OpenAI text-embedding-3-large (3072 dims).
- Stored in Supabase pgvector with metadata.

Graph Construction:

- kNN similarity edges (k=12, sim threshold 0.6).
- Extra edges from citations and categories.
- Edge pruning to maintain density.

Search & Query:

- Vector similarity search with fallback Postgres FTS.
- Concept detail retrieval with neighbors and wormholes.
- Pathfinding using weighted Dijkstra/BFS.
- Wormhole endpoint highlighting cross-domain neighbors.

Visualization:

- Three.js + D3 hybrid for galaxy rendering.
- Smooth pan/zoom, hover, and camera fly-to.
- Detail panels with summaries, links, neighbors.
- Dark/light themes.

Exports & API:

- Export subgraphs as JSON/CSV.
- REST endpoints: /search, /concept/:id, /path, /wormholes, /export.
- Rate-limited API with optional keys.

User Maps:

- Save user-defined concept collections and paths.
- Supabase Row-Level Security ensures isolation.

5. Non-Functional Requirements

Performance:

- Search latency <300ms P95.
- Pathfinding <1s for 50k–250k nodes.
- Initial frontend load <2s.

Scalability:

- Tile-based progressive rendering.
- Max neighbors = 20 per node.

Reliability:

- Daily DB backups.
- Target uptime 99.5%.

Security:

- RLS for user data.
- API key auth, input validation, request limiting.

Usability:

- Accessible color palette.
- Keyboard shortcuts: search (/), select (Enter), cycle (Tab).

Maintainability:

- Monorepo in TypeScript + Python.
- GitHub Actions CI/CD with linting, tests, deploys.

6. Tech Stack

Frontend:

- Next.js 15 (React 19, TS), Three.js + D3.js.
- TailwindCSS v4, Zustand, React Query.
- Supabase Auth for login.

Backend/API:

- Next.js API routes (Node 20).
- REST endpoints, Upstash Redis for rate limiting.

Database:

- Supabase Postgres 15 + pgvector.
- Tables: concepts, embeddings, edges, user_maps.
- Optional Neo4j Aura Free.

Data Ingestion:

- Python 3.11 (requests, pandas, openai, networkx, scikit-learn).
- Batch embedding pipeline with retries.
- GitHub Actions CRON for scheduling.

Deployment & Ops:

- Vercel for frontend/API.
- Supabase Pro Starter (~$20).

- Monitoring: Supabase Logflare + Vercel Analytics.

7. Architecture

Pipeline:

1. Ingest Wikipedia + arXiv + news.
2. Normalize → concepts table.
3. Generate embeddings via OpenAI API.
4. Store in pgvector with metadata.
5. Build graph edges (kNN + citations).
6. API exposes search, detail, path, wormholes.
7. Frontend galaxy view queries API.

8. Risks and Mitigations

- Embedding cost: mitigate with truncation + batching.
- Dense graphs: prune edges, cap neighbors.
- Performance bottlenecks: progressive rendering.
- API misuse: rate limiting, API keys.
- Licensing: only summaries, link back to sources.

9. Milestones

Phase 1: MVP ingestion, embeddings, galaxy (10k nodes).

Phase 2: Pathfinding, wormholes, clustering.

Phase 3: User maps, exports, admin dashboard.

Phase 4: Polish, docs, demo site, landing visuals.

10. Acceptance Criteria

- Search “Black Holes” completes <1s and zooms to node.
- Path from “Black Holes” → “Poetry” resolves <1s.
- Wormholes display at least 5 cross-domain links.
- Export returns valid JSON/CSV schema.
- Saved maps persist with RLS.
- Demo infra costs ~$20–25/month.

End of Document