

Eureka App Summary

What it is

Eureka is an evidence-focused literature search app built with React, Vite, and Supabase Edge Functions. It turns a research question into citation-grounded study results, synthesis outputs, and reusable report artifacts.

Who it is for

Primary persona: researchers or evidence analysts who need fast, source-linked answers from literature databases.

What it does

- Provides authenticated routes for search, report history, and report detail views.
- Starts asynchronous research jobs via research-async and tracks lifecycle state in research_reports.
- Queries OpenAlex, Semantic Scholar, arXiv, and PubMed, then deduplicates and ranks results.
- Extracts structured study fields (design, outcomes, sample size, citation metadata) with strict and partial tiers.
- Generates narrative synthesis and evidence tables, and supports follow-up paper chat responses.
- Exports report data as CSV (paper or outcomes) and RIS; supports manual Add Study by DOI.
- Attempts DOI PDF retrieval with scihub-download, stores files in the papers bucket, and tracks study_pdfs status.

How it works (repo evidence)

- Frontend: React + Vite + TypeScript with feature-based structure (src/app, src/features, src/entities, src/shared).
- Frontend uses Supabase Auth and calls Edge Functions: research-async, synthesize-papers, chat-papers, add-study, coci, scihub-download.
- research-async orchestrates query processing, multi-provider retrieval, dedupe/canonicalization, metadata enrichment, and deterministic extraction.
- Data persists in Supabase Postgres and storage (research_reports, study_pdfs, lit_query_cache, lit_paper_cache, query_processing_events, metadata_enrichment_*, rate_limits).
- Optional PDF extraction path: if PDF_EXTRACTOR_URL is set, extraction calls FastAPI /extract/studies; otherwise it falls back to abstract text.
- Primary flow: user query -> research-async pipeline -> external APIs/AI services -> Postgres/storage -> polling UI + synthesis/chat endpoints.

How to run (minimal)

1. Install dependencies: npm install.
2. Start frontend dev server: npm run dev.
3. Open the local Vite URL, then sign in (or guest sign-in) to access /app.
4. Optional project config: set VITE_SUPABASE_URL and VITE_SUPABASE_PUBLISHABLE_KEY from .env.example.
5. Full local Supabase Edge deployment workflow: Not found in repo.