AI Workflow Builder – End-to-End Project Description

This project is a **no-code AI workflow automation platform** that allows users to build, run, and scale AI-powered pipelines visually. It mirrors VectorShift’s core vision: AI-first workflows with Retrieval-Augmented Generation (RAG), integrations with third-party apps, and production-grade scalability.

Core Features

- Drag-and-drop workflow builder (React/Next.js, React Flow).
- Data ingestion nodes (PDFs, URLs, Webhooks).
- AI-native nodes: RAG Q&A;, summarization, classification, text transforms.
- Vector DB integration (Qdrant/Chroma) for semantic search.
- Output integrations: Slack, Google Sheets, Email, Notion, Twilio, Webhooks.
- Workflow execution engine (FastAPI + Redis Queue workers).
- Scheduling, retries, feedback loop, and run logs.
- Metrics and observability via Prometheus/Grafana.

Technology Stack

- Frontend: React, Next.js, Tailwind, React Flow.
- Backend: Python, FastAPI, Async workers (RQ/Dramatiq).
- Databases: MongoDB (workflows, runs, users), Redis (cache/queue), Qdrant/Chroma (embeddings).
- Integrations: Slack API, Google Sheets API, Twilio SMS, Notion API, Email/SMTP.
- Infra: Docker, GitHub Actions CI/CD, Linux VPS, Nginx reverse proxy, Let’s Encrypt SSL.
- AI/LLMs: OpenAI, LangChain, embeddings, prompt engineering.

Deliverables

- Functional no-code editor with drag/drop workflow canvas.
- Workflow runner with data ingestion → AI reasoning → action nodes.
- RAG pipeline with document upload, indexing, and Q&A.;
- At least 5 integrations (Slack, Sheets, Email, Notion, Twilio).
- Redis caching, retries, and scheduling of workflows.
- Live demo with example use cases (daily report digest, Q&A; bot, Slack notifier).
- Documentation and 90-sec demo video.

Scalability & Performance

- Async FastAPI workers for concurrent execution.
- Redis-based job queue with retries and backoff.
- Batch embedding & caching to reduce token costs and latency.
- Horizontal scaling: multiple worker containers.
- Monitoring: Prometheus metrics (latency, error %, cache hit ratio).

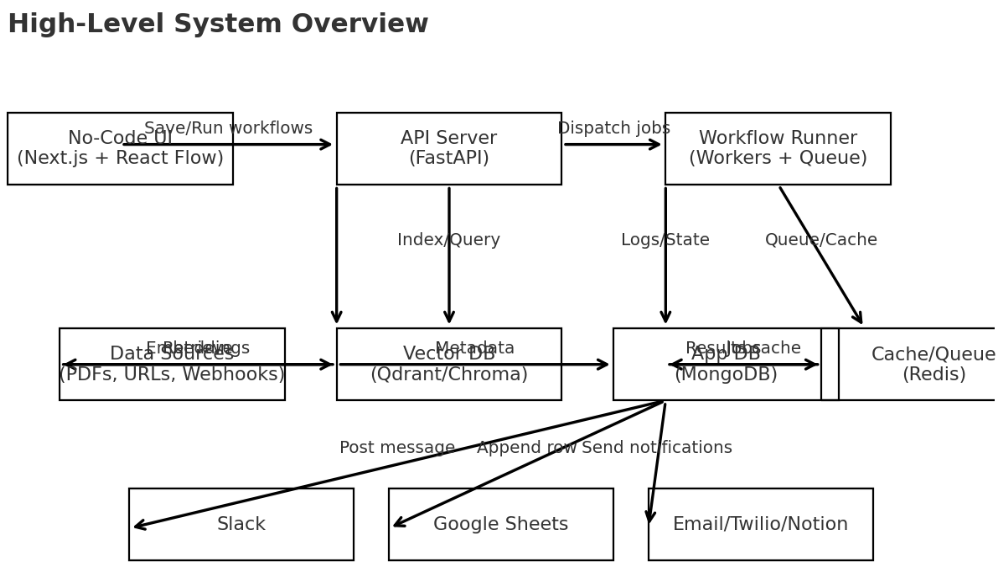
Future Improvements

- Real-time run monitoring (WebSockets/SSE).
- Multi-model routing (OpenAI fallback to local LLM).
- Fine-grained permissions (orgs, roles).
- Cost tracking and optimization dashboards.
- Marketplace of prebuilt templates and nodes.

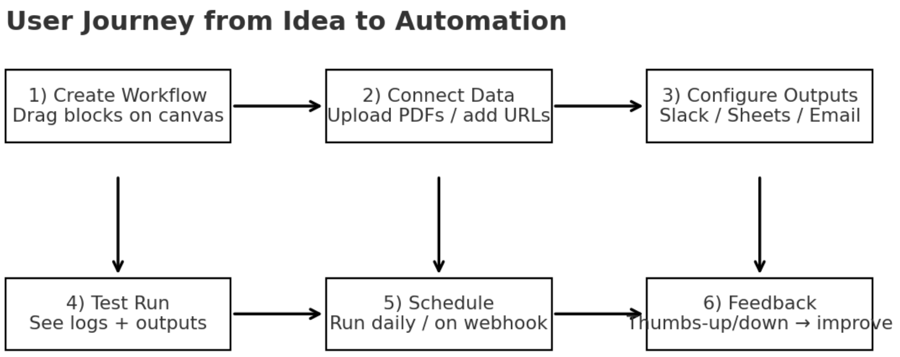
Why Choose This Platform?

Unlike general-purpose workflow tools (n8n, Zapier), this system is **AI-native**. It provides first-class support for RAG, embeddings, caching, and large-scale AI performance. It is designed to make AI workflows accessible to non-technical users while giving developers production-grade APIs and SDKs.

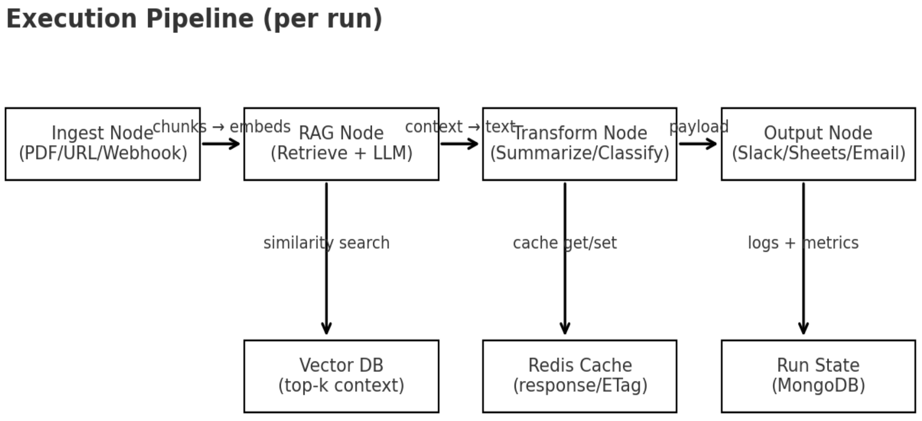
System Overview Diagram



User Journey

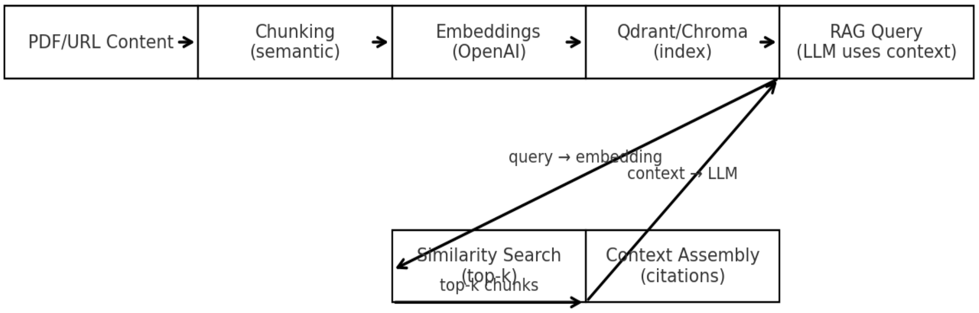


Execution Pipeline

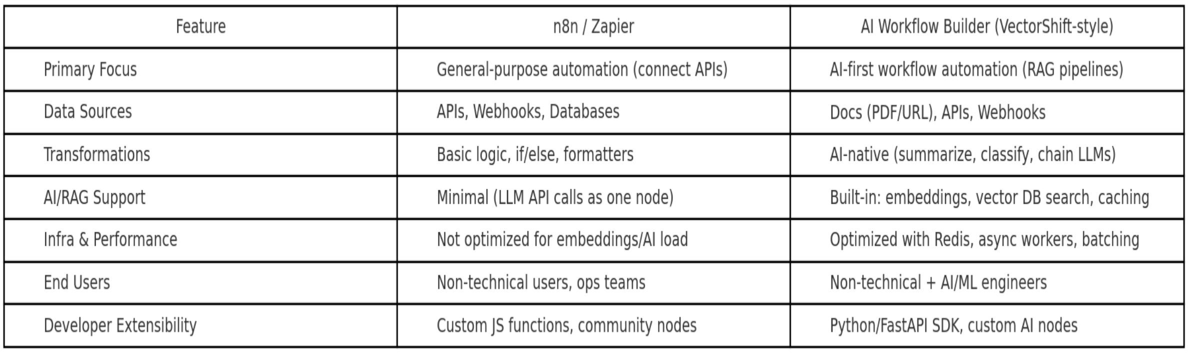


RAG Data Flow

RAG Data Flow



Comparison: n8n vs AI Workflow Builder



Feature	n8n / Zapier	AI Workflow Builder (VectorShift-style)
Primary Focus	General-purpose automation (connect APIs)	AI-first workflow automation (RAG pipelines)
Data Sources	APIs, Webhooks, Databases	Docs (PDF/URL), APIs, Webhooks
Transformations	Basic logic, if/else, formatters	AI-native (summarize, classify, chain LLMs)
AI/RAG Support	Minimal (LLM API calls as one node)	Built-in: embeddings, vector DB search, caching
Infra & Performance	Not optimized for embeddings/AI load	Optimized with Redis, async workers, batching
End Users	Non-technical users, ops teams	Non-technical + AI/ML engineers
Developer Extensibility	Custom JS functions, community nodes	Python/FastAPI SDK, custom AI nodes