Coda App Summary

What it is

coda is a personal AI assistant platform that runs in Discord and helps manage daily information through skills and integrations. It combines tool-calling orchestration with event-driven alerts and scheduled automation.

Who it is for

- Primary persona: individual Discord user who wants one assistant for reminders, notes, and briefings.
- Detailed persona profile document: Not found in repo.

What it does

- Handles natural-language and slash-command interactions in Discord.
- Creates, lists, snoozes, and completes reminders with natural-language time parsing.
- Saves and searches notes using PostgreSQL full-text search and tags.
- Adds semantic memory retrieval via a dedicated memory-service and vector embeddings.
- Connects integrations such as MCP servers, n8n, Firecrawl, and weather.
- Routes alerts with quiet hours, cooldowns, severity, and persistent audit history.
- Runs cron-based background tasks with retry and runtime enable-disable controls.

How it works (repo-evidenced architecture)

- Interfaces: DiscordBot (and optional SlackBot) receive messages and hand off to the Orchestrator.
- Core runtime: Orchestrator coordinates ProviderManager (LLM), SkillRegistry, ContextStore, and ConfirmationManager.
- Events: skills publish to EventBus (Redis Streams by default, in-process fallback).
- Alert path: AlertRouter applies routing rules, quiet hours, and cooldowns, then sends to Discord/Slack sinks.
- Scheduling: TaskScheduler registers recurring jobs used by reminders, memory upkeep, and health tasks.
- Data/services: PostgreSQL + pgvector, Redis, memory-service (HTTP), and n8n-webhook service.

How to run (minimal)

- Install Docker Desktop and gather credentials: DISCORD_BOT_TOKEN, DISCORD_CHANNEL_ID, and one LLM API key.
- Run one-liner deploy: DISCORD_BOT_TOKEN=... DISCORD_CHANNEL_ID=... OPENROUTER_API_KEY=... ./scripts/deploy.sh
- Verify startup: curl localhost:3000/health and docker compose logs -f coda-core