

Gallagher CRES App Summary

What it is

A production-grade multi-agent system for commercial real estate development workflows built with the OpenAI Agents SDK and a FastAPI API. It coordinates eight specialist agents across the full development lifecycle.

Who it's for

Commercial real estate development teams at Gallagher Property Company. Primary user persona details (role/level) not found in repo.

What it does

- Coordinator agent orchestrates work across 8 specialist agents (research, finance, legal, design, operations, marketing, risk).
- Research tools for parcel search, market data, comparables, and location analysis using external data sources.
- Finance tools for pro formas, debt sizing, waterfall modeling, and sensitivity analysis (IRR, DSCR, LTV).
- Legal tools for zoning analysis, contract drafting/review, and permit tracking.
- Design and operations tools for development capacity, site planning, construction cost estimates, schedules, and cost tracking.
- Marketing and risk tools for listings and offering memos plus flood, environmental, and insurance risk assessment.

How it works (architecture)

- UI layer: Next.js 15 dashboard in `frontend/` with pages, components, and API routes.
- API layer: FastAPI app in `main.py` exposes project, task, and agent endpoints with Pydantic models in `models/`.
- Orchestration: `workflows/runner.py` uses the OpenAI Agents SDK Runner and `gpc_agents/*` to run the coordinator and specialist agents.
- Data/services: `tools/database.py` (Supabase via `database/schema.sql`), `tools/external_apis.py` (Perplexity, Google Maps, FEMA), `tools/storage.py` (Backblaze B2).
- Data flow observed: API endpoints invoke workflows and persist outputs through `tools.database`.

How to run (minimal)

- Install backend deps: `python -m venv venv && source venv/bin/activate` then `pip install -r requirements.txt`.
- Configure env: `cp .env.example .env` and set required API keys (OpenAI, Perplexity, Supabase, Google, Backblaze).
- Apply database schema: run `database/schema.sql` in Supabase SQL editor.
- Start API: `python main.py` (or `uvicorn main:app --reload --host 0.0.0.0 --port 8000`).
- Optional UI: `cd frontend && npm install && npm run dev`.