CourtOps Analyst Agent

From data silos to actionable intelligence

A full-stack portfolio build for Municipal Court Functional Analyst roles

Next.js • FastAPI • PostgreSQL • Celery • Ollama

The challenge

Municipal courts sit on massive amounts of data —
cases, tickets, inventory, audits — but turning that into
actionable intelligence is hard.

- Failure-to-appear (FTA) cases slip through the cracks
- Revenue at risk is invisible without custom reports
- Help desk, inventory, and reporting live in separate worlds
- Compliance and audit trails are manual and reactive

What if one platform could tie it all together — and an AI agent could run the ops?

One platform, end to end

Help desk with SLA tracking and triage → Court case metrics and backlog

Hardware inventory and compliance → Patch and change management

Monthly operations reports → Revenue at Risk (FTA) and audit reports

Change requests → Auto-generated functional specs, SOPs, release notes

All with role-based access, full audit logging, and public/synthetic data only — no proprietary court data.

Built for the job

Functional Analyst duties mapped to real features

Help desk & work orders
Tickets, SLA, triage, escalation

Court case operations
Case metrics, FTA, time-to-disposition, Revenue at Risk

Hardware inventory
Assets, warranty, patches, compliance checks

Patch / upgrade lifecycle
Requested → scheduled → tested → deployed → verified

Change requests & docs
Functional specs, SOP updates, release notes

Security & audit
RBAC, login/export/role audit, anomaly detection

Tech stack

Frontend: Next.js (App Router), TypeScript, Tailwind, shadcn-style UI

Backend: FastAPI, SQLAlchemy, JWT auth, Celery + Redis

Data: PostgreSQL, public data connector, synthetic demo data

AI agent: Ollama (local LLM) + whitelisted tools only

No shell access. Every tool call is audit-logged. Crystal Reports–style outputs.

Deploy: Docker Compose locally • Fly.io / Render for cloud demo

The AI agent

An LLM-driven CourtOps Agent runs a full "daily ops" demo:

1. Refresh public dataset cache
2. Triage tickets and resolve access issues
3. SLA sweep and escalate overdue work orders
4. Inventory compliance check and create patch records
5. Generate monthly operations report bundle
6. Generate Revenue at Risk (FTA) report
7. Generate audit report
8. Create change request and generate docs

One preset, one click — dry run or live. Built to showcase potential, not replace analysts.

Revenue at Risk (FTA)

Crystal Reports–style: FTA/warrant cases, grouped by violation type

Judges get pre-sorted dockets. Managers see total revenue at risk.

Group: Traffic Violations (High Priority)

Citation | Defendant | Days Overdue | Outstanding Bal. | Subtotal

Group: City Ordinance (Code Enforcement)

Citation | Defendant | Days Overdue | Outstanding Bal. | Subtotal

TOTAL REVENUE AT RISK: $X,XXX.XX

From raw data silos to actionable intelligence — the core of the portfolio story.

Run it yourself

One-command demo (Windows): run_demo.bat

Starts Ollama, Docker Compose, seed data — then open the Agent Console.

- Local: http://localhost:3000 (login: supervisor / password)

- Deploy: Fly.io or Render from the repo — terminal-driven setup

- No proprietary data: public datasets + synthetic data only

Built to show what's possible when functional analysis meets full-stack + AI.

CourtOps Analyst Agent

Portfolio project for Municipal Court Functional Analyst roles

GitHub: github.com/Hobie1Kenobi/courtops-analyst-agent

End-to-end: help desk, cases, inventory, reports, audit, AI agent

Let's connect.