

2-Day POC: Lightweight Multi-Tenant Admin Portal

Goal

Build a **simple Admin Portal** to simulate **multi-tenant onboarding, source config, pipeline control, and user-role management** — just enough to evaluate hands-on full-stack and system design skills.

Scope & Features (Minimal but Evaluative)

Day 1: Core Setup

1. **Customer Onboarding (Frontend + API)**
 - Form: Customer name, email, timezone
 - Saves to metadata DB (PostgreSQL or SQLite)
 - View list of all tenants
 2. **Source Configuration**
 - Simple form to store: DB host, port, username, password (encrypted or masked)
 - Save config linked to tenant
 3. **Pipeline Toggle**
 - On/Off switch per tenant (Boolean flag)
 - API updates pipeline state (simulated)
 - Show status in dashboard table (Running/Stopped)
-

Day 2: Users, Roles & Health Monitoring

4. **User Management (Optional if time)**

- Create user with role (Admin, Viewer)
- Store in DB and simulate role-based access (hide buttons if Viewer)

5. System Health Snapshot (Simulated)

- Fake metrics: Last sync time, last error message
- Table view with status color (green/yellow/red)

6. Authentication

- JWT-based login (use FastAPI JWT or simple token stub)
- UI changes based on user role (can hardcode Admin/Viewer switch)

Tech Stack Constraints

Layer	Stack Options
Frontend	React (preferred)
Backend	FastAPI (APIs) + Django ORM or only FastAPI with SQLAlchemy
Database	SQLite or PostgreSQL
Auth	JWT or hardcoded users in DB

Deliverables

Item	Notes
Working web app	Dockerized, simple UI

Item	Notes
GitHub repo	With README & setup
Screenshots / demo	(optional) For bonus clarity
Code quality & clarity	Clean API, form validation, DB usage



Evaluation Criteria

Category	Weight
Working UI/API	40%
Clean data model	20%
JWT or user roles	20%
Code readability	20%



Bonus (Optional)

- Use Docker Compose to run app + DB
- Simulate pipeline status change via a background thread or a mock status refresh
- Mask passwords in source config screen

Folder Structure Suggestion

```
admin-portal/  
├── backend/  
│   ├── main.py (FastAPI)  
│   ├── models.py  
│   └── db.sqlite3 or PostgreSQL  
├── frontend/  
│   ├── src/  
│   │   ├── pages/ (Onboarding, Config, Dashboard)  
│   │   └── App.js  
│   └── package.json  
├── docker-compose.yml  
└── README.md
```