

Deployment Approval Workflow System

1. README – Setup & Usage Instructions

This project implements a deployment approval workflow using a state machine on the backend and a simple frontend to visualize state transitions.

Tech Stack

Backend: Python, FastAPI, SQLAlchemy, SQLite, transitions

Frontend: HTML, CSS, JavaScript

Deployment States

requested → approved_by_manager → approved_by_devops → executed, with rejection possible from any state.

2. Design Explanation & Trade-offs

The system models deployment approval as a finite state machine rather than manual status flags. This guarantees valid transitions and centralizes workflow logic.

Why a State Machine

Using the transitions library enforces correctness, prevents invalid state changes, and makes the workflow extensible.

Trade-offs

- 1 State machine ensures correctness but adds a learning curve
- 2 SQLite simplifies setup but is not production-grade
- 3 REST API is simpler than WebSockets but lacks push updates

3. Architecture & Data Flow

Frontend communicates with FastAPI backend over HTTP. The backend applies state transitions and persists state in the database.

State Flow: requested → manager approval → devops approval → executed, with rejection at any stage.