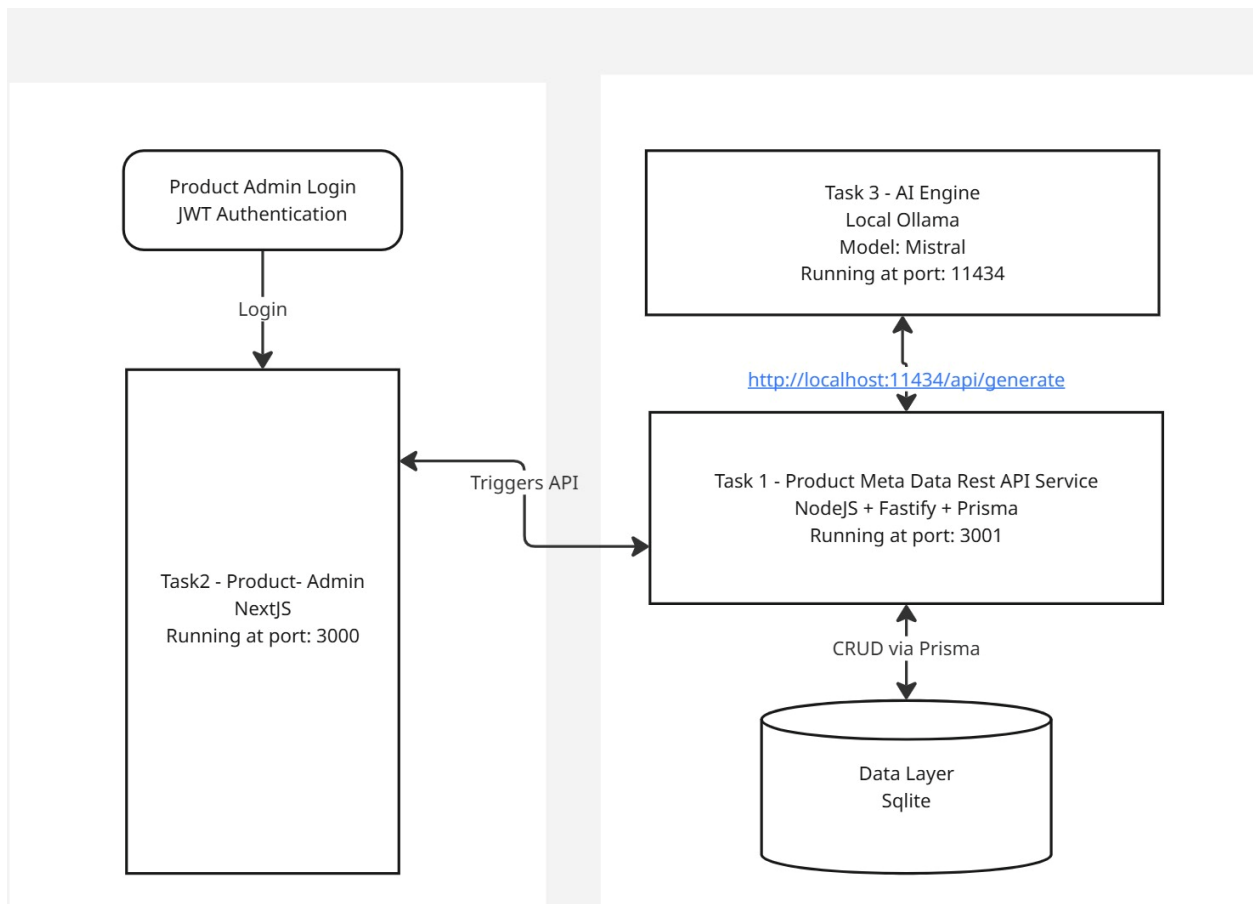


Product Metadata Management System

This system is designed as a modular fullstack architecture integrating product management with local AI-assisted tag generation.

□ Architecture Overview



□ Product Admin (Frontend)

- **Task 2 - Product Admin Dashboard**
 - Built with **Next.js**
 - Port: 3000
 - Features:
 - JWT-based admin login
 - Product CRUD management
 - Triggers API calls to backend services

- **Authentication**
 - JWT Authentication
 - Tokens used in API requests to authorize operations
-

□ Product and Tag Manage REST API Service (Backend)

- **Task 1 - Product Meta Data REST API**
 - Built with **Node.js + Fastify + Prisma**
 - Port: 3001
 - Responsibilities:
 - Accepts authenticated API requests from frontend
 - Handles product metadata operations (CRUD)
 - Sends tag suggestion requests to AI engine
 - **Task 3 - AI Engine**
 - **Local Ollama with Mistral Model**
 - Port: 11434
 - Triggered via API call `http://localhost:11434/api/generate`
 - Returns suggested tags for a product description
 - **Data Layer**
 - **SQLite** via Prisma ORM
 - Stores product records and metadata
-

□ Security

- JWT-based authentication ensures only authorized admins can manage product data.
 - Tokens are validated on each API request.
-

□ Dev Environment (Optional)

- Services can be Dockerized and run using Docker Compose:
 - Next.js frontend
 - Node.js API backend
 - Ollama local AI server
-

□ Future Enhancements

- Replace SQLite with PostgreSQL for production
- Role-based access control (Admin, Editor)
- UI enhancements with charts and filters
- Pagination and search on product list