Technical Challenge – Senior Java Backend Developer

Goal

Build a modular, secure backend application using Java (version greater than 11). The system should ingest video metadata from an external API, store it, and expose authenticated endpoints for querying and statistics.

Scenario: Video Metadata Service
GoldmediaTech is building a backend service to manage and analyze metadata from videos across multiple platforms (e.g., YouTube, Vimeo, internal providers). You are responsible for building the first version of this service.

Requirements

1. Authentication

   - Implement JWT-based authentication
   - Provide hardcoded users (e.g., in-memory or from file), with at least two roles: admin and user
   - Public endpoint:
     - POST /auth/login – returns a valid JWT
   - All other endpoints must require a valid token

2. Video Metadata Endpoints

A. Ingest

- POST /videos/import: Imports video metadata from an external API or mock service
- Only accessible by users with the admin role

B. Query

- GET /videos: List all videos, with optional filters (e.g., by source, upload date, duration)
- GET /videos/{id}: Fetch a single video by its ID

C. Analytics

- GET /videos/stats: Returns basic statistics, such as:
    - Total videos per source
    - Average duration per source

3. Architecture

- Organize your project using a modular or layered structure
    - Example: config, domain, services, web, persistence
- Use DTOs and mapping logic between layers (manual or mapper library)
- Clean and maintainable code is essential

4. Documentation

- Provide API documentation (e.g., Swagger/OpenAPI)
- Include examples of how to:
    - Authenticate and obtain a token
    - Use the protected endpoints

5. Testing

- Write basic unit tests for:
    - At least one service class
    - At least one controller method
- No need for full integration or E2E tests

Bonus (Optional)

- Caching for statistics
- Pagination and sorting
- Role-based authorization beyond import

- Background/asynchronous video imports
- Dockerfile or CI pipeline stub

Deliverables

- A GitHub/GitLab repository containing:
    - Full source code
    - A README.md with:
        - Instructions to run the project
        - Description of your design decisions and assumptions
        - Example API usage (curl, Postman, etc.)
        - How to log in and use the token

Time Estimate
This challenge is designed to be completed within 3 to 5 days of part-time effort.

Use of AI Tools

The use of AI tools (e.g., ChatGPT, Copilot) is allowed. You're free to use them to improve productivity, as long as you understand and can explain the code you submit.