

Objective

To evaluate the candidate's ability to design and implement a robust backend system for a compliance workflow application. The task focuses on creating a simplified KYC (Know Your Customer) backend service, including key features like authentication, role-based access control, KYC management, and reporting.

Task Overview

Build a backend service to support a KYC system where:

- Users can:
 - Register and log in with secure credentials.
 - Submit KYC information (e.g., name, email, and an uploaded ID document).
- Admins can:
 - View a list of users and their KYC statuses.
 - Approve or reject KYC submissions.
 - Access basic reporting KPIs (e.g., total users, number of approved/rejected submissions).

The backend should include secure, scalable, and maintainable APIs to handle these workflows.

Requirements

Backend (Node.js):

1. Authentication and Authorization:
 - Implement secure user authentication using JWT.
 - Role-based access control for different user types (Admin/User).
 - Middleware to restrict admin-only functionality.
2. KYC Management:
 - Endpoints for:
 - User registration and login.
 - Submission of KYC details (name, email, uploaded ID document).
 - Retrieval of KYC data by admins (with filtering and pagination).
 - Approving or rejecting KYC submissions (admin-only).
 - Include proper request validation (e.g., using libraries like Joi or Yup).
3. Database Integration:
 - Use PostgreSQL or MySQL (preferred) or MongoDB.
 - Logical schema design for:
 - User credentials and roles.
 - KYC details and statuses.
 - Efficient queries for fetching, updating, and aggregating data.
4. File Storage:
 - Save uploaded ID documents securely using:
 - Local storage for simplicity (required).
 - Cloud storage (e.g., AWS S3, optional for bonus points).

5. Dashboard KPIs:
 - Create an endpoint to fetch basic compliance KPIs:
 - Total number of users.
 - Number of KYCs in each status (approved, rejected, pending).

Optional Frontend (Minimal Effort):

- Provide basic functionality using a simple API testing tool like Postman or a lightweight web interface (optional).

Evaluation Criteria

- API Design Principles
- Database Design
- Security Best Practices: on against common vulnerabilities (e.g., SQL injection, improper error exposure).
- File Management
- Code Quality and Scalability

Bonus Points (Optional)

- TypeScript Integration
- Unit Testing
- Deployment
- Advanced File Handling

Deliverables

- A GitHub repository with well-structured commits and a README
- (Optional) Live Demo

Timeframe

- Estimated Completion Time: 8–10 hours.
- Submission Deadline: 2 days from receiving the assessment.