Curjos Full Stack Assignment

Goal

Design and implement a small backend application and a simple admin interface that demonstrates your ability to structure a real-world system from scratch.

The application should allow authenticated users to manage organizations, users, and courses that contain content such as video, pdf files and exercises. Your solution should include a clear data model, a consistent REST API consumed by a basic Frontend application and appropriate authentication and authorization logic.

You are free to use any frameworks or libraries you prefer. The main objective is to show how you approach system design, API architecture, and code organization, rather than to produce a fully polished product.

Context

You're working on a small learning management system prototype designed for organizations that host internal training and leadership courses.

Each organization (tenant) has its own set of users, courses, and learning content (eg. pdf files, exercises, videos, etc). Admin Users and Content Managers need a secure way to manage these resources — creating users, assigning roles, publishing courses while students can only view they own course path.

The courses can have different levels (e.g Beginner, Advanced, etc) with different content according to the user level. A user's level can be set when the account is created or automatically defined based on a short questionnaire when they sign in for the first time.

Your goal is to design and implement a minimal system (backend-first) that could support these needs. The system should expose a clear REST API and, a lightweight admin interface to interact with it.

This is not a production-level project — the focus is on how you structure your solution, model data, handle authorization, and expose consistent API endpoints.

Core Requirements

- Authorization
  - Admin: full access; can manage all organizations, users, and courses.
  - Content Manager: can create, update, and delete content.
  - Student: can view only the content of their own course.

- Implement a REST API solution
- Consume the endpoints using a frontend application (feel free to use any framework you feel more comfortable. React is a nice to have - basic UI is sufficient)
- Include installation and startup instructions (README.md).

Timebox

Expected effort is 4–5 hours.
You don’t need to implement everything — focus on building a coherent and well-structured slice that demonstrates your technical reasoning.

Please share a repository with the implementation one day before your scheduled technical interview.

Completeness is less important than clarity, correctness, and thoughtful design.

Good luck — we’re excited to see how you think!