**🧩 Assignment: Build a "Project & Task Management System" API**

**✅ Requirements**

**1. User Authentication & Role-based Access**

* Use JWT for authentication.
* Three roles: Admin, Manager, User.
* Admin can: CRUD users, projects, tasks.
* Manager can: create/read/update/delete projects & tasks only within their projects.
* User can: view tasks assigned to them & update task status.

**2. Models & Relations**

* User: id, name, email, password, role.
* Project: id, name, description, created\_by (user\_id).
* Task: id, title, description, project\_id, assigned\_to (user\_id), status.

Relations:

* User → Projects (one-to-many)
* Project → Tasks (one-to-many)
* User → Tasks (one-to-many as assignee)

**3. Query Optimization**

* Fetch:
  + All projects with task count.
  + All tasks assigned to a user, with project name.
  + Tasks grouped by status (for dashboard).

Use Sequelize’s include, attributes, group, raw options to optimize.

**4. Handle Large Load of Requests**

* Add rate limiting (e.g., express-rate-limit).
* Use caching (e.g., node-cache or Redis).
* Use pagination & filtering in list APIs.

**5. Extra (Advanced)**

* Upload task attachments (files) to local storage or S3.
* Write basic tests (e.g., with Jest).
* Log API requests (e.g., with morgan or winston).

**🛠 Tech Stack**

* Node.js + Express.js
* MySQL
* Sequelize ORM
* JWT
* express-rate-limit, caching library (node-cache / Redis)