Project:
You must implement a simple device management web app. Devices have (Id, category, color, partNumber). The device category must also be managed by your APP. Types have (id, name).

- Login and authentication is not necessary.

- Web app must have a menu with two options: Category Management, Device Management.
- You must implement Create, Read and Delete. (Update is not necessary due to time constraints for this implementation)
- Device:
    - Id: (generated automatically. Integer and incremental)
    - Category: A dropdown selection you can choose from all categories available.
    - Color: Text field, with validation (letters only, max size 16)
    - partNumber: positive integer field, with validation.
- Category:
    - Id: (generated automatically, integer and incremental)
    - Name: Must not be empty. All categories must have a name. Max size 128 chars.
- All fields are mandatory.
- You are free to use any layout but the APP must list all devices and categories, and also have screens/components that enable the user to create new categories and devices, or delete them.

- Front End must be implemented with Angular 8 or above.
- Backend must be implemented in NodeJS (version 10 or above).
- [Use of Nestjs is not mandatory, but it is a plus]
- [Automated tests on the backend is not mandatory but is a PLUS.]
- Devices and Categories MUST be persisted on a MySQL database.
- You must provide the script (SQL or any database versioning/migration script) that can create the database from scratch.
- Use GIT, commit every progress you made, and share your cond on a github public repository. Super challenge:

- Deploy your project on a cloud provider (AWS, GCP, Heroku or any other), and send the URL of your web application, and the URL of your GIT repo.
- AWS is preferred. Our recommendation is to sign up for a free account and use:
- One T2 micro EC2 instance (for backend)
- One RDS T2 micro instance (for database)
- Frontend can be deployed on S3 static bucket or on the same EC2 instance used for the backend (to avoid additional costs).
- In case the project could not be deployed, it must be ready to run and install on a Ubuntu 18 linux machine. The evaluation process would be:
- Clone the repo. (create the local database based on the script given). Run front end and back end locally.