NodeJS Assessment

This document describes the evaluation assignment for the position of a NodeJS backend engineer.

Techniques:

- Object Oriented
- Code should follow common software design principles, such as SOLID, DRY, etc.
- 3rd party libraries should be installed with a package manager, like Composer, NPM, Yarn.

What are we assessing:

- Application architecture
- REST API Design skills
- Coding style & quality overall
- Programming patterns

On theoretical level, we are looking into:

- Communication skills
- High performance design skills

In general architectural choices are important to us.

Time

We understand that you are busy with more things, so the time you can afford to put on any parts of this assessment depends on you.
However we would like to know when (date) you expect to come back to us with your solution, and how long you expect it would take you. See it as a mini project.

The Exercise

We have an online shop store.
It has a list of products and a list of customers (users). Customers can add products to cart and apply all crud operations.

The (actual) assignment

A. REST Service

We need you to create a REST service to CRUD
In the delete apis, if there is a dependency between the data, it will give an error.

B. UI

- Swagger for apis

- Postman file for apis (optional)

Deliverables

In your submission, we would like you to include:

1. All code (backend) developed to create this functionality

2. The DB script that generates the structure and the data to run this application

For convenience sake, for points 1 & 2, you can use a code repository like Github and share it with us.

Bonus Questions Bonus Questions

1. It’s a big plus if you add (some) tests to your code (let us see your QA skills)

2. Looking at the description of the context, please provide architecture diagram(s) on how you would technically design the functionality described for this application in a scalable way. You can decide yourself what types of diagram(s) will help you communicate to us your thoughts about building such an application.

3. Which parts of the application would you have implemented differently and why should you have more time?