



Inforce frontend task

Technical stack

- React
- Redux
- Typescript (optional, but would be a big advantage)

NOTE

You will need a server for this kind of app, so use json-server for API requests.

(Also you can use localStorage, but be aware of initializing data on the first run)

If you know the backend part and have experience with databases - just use it to build your own server instead of a JSON Server.

Actual task

We need to build a product list web app. The goal is to simulate a shop app.

There are two primary views: the product list view and the product view.

Products List View:

On the list view, I can add products by clicking on the add button(after that modal window should occur with different inputs which allows me to set product details, also inside modal window should be two buttons, for confirmation and for canceling) and remove products (When pressing the **delete** button I should see a modal with confirmation of removing this product - I can either confirm or cancel the current action). You shouldn't be able to add an empty product without setting data for him.

The list view should always sort alphabetically first, and then by count. Also I can select how products should be sorted from the dropdown menu.

Product View:

On the product view, I can see all of the details of the product. After pressing the **edit** button I can see the modal with different inputs which allows me to change product details.

Also you can add and delete comments for the current product.

Guidelines

Do not spend more than 5 hours working on this task. If we have a technical interview, and I hope we do, we will focus on enhancing this application and discussing how you worked through some of these problems. It's important that we see your best work, if that means

that you do not satisfy all of the requirements here. That is okay, we don't expect everyone to finish all parts of this in 5 hours.

If you have to choose between refactoring and making one piece of this perfect and implementing the next feature, choose refactoring because we want to see how your best work looks.

We want to see clear, correct code that uses the right data structures and patterns, and we want to see your style.

Models

Product

```
{
  "id": 1,
  "imageUrl": "some url here",
  "name": "Product name",
  "count": 4,
  "size": {
    "width": 200,
    "height": 200
  },
  "weight": "200g",
  "comments": ["CommentModel", "CommentModel"]
}
```

Comment:

```
{  
  
  "id": 3,  
  
  "productId": 1,  
  
  "description": "some text here",  
  
  "date": "14:00 22.08.2021"  
  
}
```