UPSHIFT

Technical Challenge

Front End

Skopje 2020

Intro

Welcome to the Upshift Technical Challenge.
In this technical challenge you will need to develop a simple eCommerce App.

The purpose of this technical challenge is to test several aspects of developing a React application including:

1. **Architecture,**
2. **Performance,**
3. **API's,**
4. **State Management,**
5. **Styling**

Recommended external libraries for usage in this challenge are : React-Router, Axios, Redux, Redux-Thunk and Formik.
Candidates are allowed to use other libraries as well. Please write a short explanation of pros and cons of the used library in the README file.

Rules

1. Please use **Create React App.**
2. Please push your code to a **Git** repository (provided by us)
3. Please provide instructions for starting the app (you can use the README file for this purpose).
4. Please provide documentation if you use any third party libraries.

Important

The API used in this challenge is not a real functional API , but uses dummy data. It will not really create or delete resources on the server but it will be faked as if. In other words if you created a resource with ID 101 and after that you try to GET that resource , you will get a 404 error.Created resources should be kept in state.

Response structure

Product

id	number
title	string
price	number
stateId	number
categoryId	number
stock	boolean
picture	string
description	string

State

id	number
name	string
tax	number

Category

id	number
name	string

API Routes

Guide how the API works http://jsonplaceholder.typicode.com/guide.html.

NOTE: Routes used in this guide are not the same as the provided routes in this challenge.

GET Products	https://my-json-server.typicode.com/drakulovski/dbplaceholder/products
GET Product with query param	https://my-json-server.typicode.com/drakulovski/dbplaceholder/products?categoryId=1
POST Product	https://my-json-server.typicode.com/drakulovski/dbplaceholder/products
DELETE Product	https://my-json-server.typicode.com/drakulovski/dbplaceholder/products/:id
GET States	https://my-json-server.typicode.com/drakulovski/dbplaceholder/states
GET Categories	https://my-json-server.typicode.com/drakulovski/dbplaceholder/categories
Whole DB file	https://my-json-server.typicode.com/drakulovski/dbplaceholder/db

Challenge

The challenge consists of creating a simple eCommerce application that will have the following concepts:

1. List product items
    a. Sortable by Name and Price (Sorting should be implemented on frontend)
2. View single product
3. Create a product item

4. Delete product item

API

Create and Delete actions of the products should be simulated through API requests to the provided routes but as mentioned above all the requests will be faked and actions such as delete or create should be done in the state on the app after success of the request .

Description

Initially products that will be listed on the main page should be fetched from the API.
The listed Products should be sortable by Name and Price.
On click on one product a preview for single product should open.
Each product can be deleted. When the user clicks on delete confirm box should pop up.
In the right corner of the screen there should be a button that on click will open a modal and should contain a form that will create a product with the validation required below.

Form validation for creating a Product

- **Title** is required with minimum 4 characters
- **Picture** is required (url)
- **Price** is required and should be higher than 4
- **State** should be one of the provided states
- If on the selected **State** tax field is larger than 0.25 minimum required price of the **Product** is 7
- **Category** should be one of the provided categories

Bonus

- Usage of Typescript
- Don’t use a third party UI Kit.

Design

The design provided in this challenge is just a mockup , you are free to express yourself with styling and design. Design Link

If the candidate has any questions please send them to marko@upshiftwork.com .
We wish you good luck !