Snappy Developers Exercise 2019

Candidate name

Important

Hello dear candidate, we are very excited to meet you!
Please complete reading this document before you start writing code.
Remember we're also here to have fun together, so don't forget to enjoy doing the exercise 😀

Background

Snappy platform allows gift recipients choose their favorite gift, the gift they actually want. In order to do it, Snappy team picks the best products out there in the market. Your mission is to help our Snappy team to see and manage products that are already exist on the platform.

Task

Client

1. Create single page application (preferred with react).
2. Create one route to show list of products (product model attached).
3. Add ability to search product by name and filter by product vendor.
4. Product grid need to include a Promotion card (checkout the data provided with this task).
   show only when products > 0, place in a predefined location.
5. Products order should be predefined by the field "order". (Only on first load, and not after search).
6. Product card need to include media (image or video, prefered video), vendor and name.

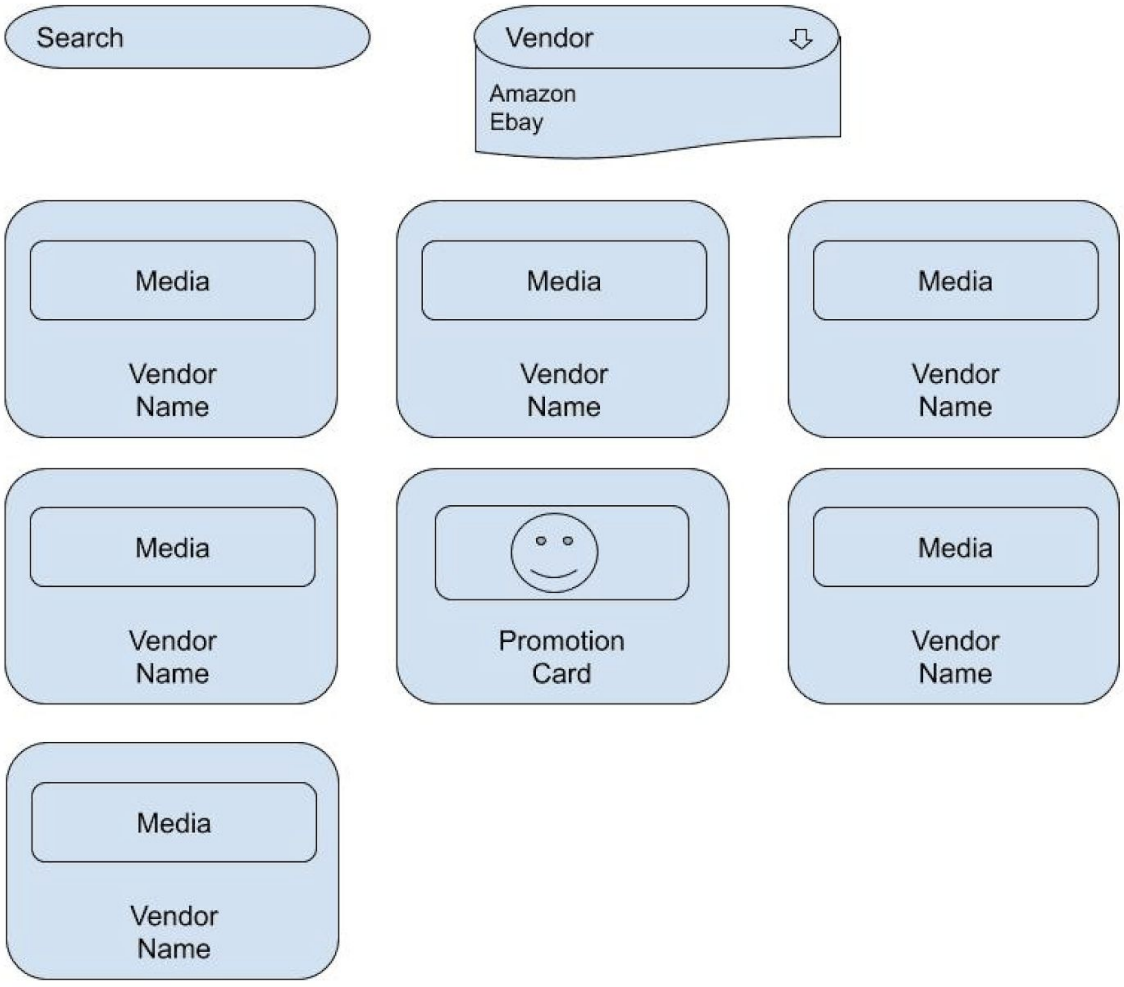
Server

1. Create BE service preferably on nodejs.
2. Service should return products data based on attached JSON.
3. Handle cache for search queries for 5 minutes.
4. Pull products from json file attached or migrate the json file to mongodb (a plus)

Data

Download json file with the following link (https://snappysapp-server-files.s3-us-west-1.amazonaws.com/data.json)

Mock



Guidelines

some information to help you pass this exercise smoothly

1. Please don’t “invent the wheel” - whenever possible, please use 3rd-party libraries and code other people already wrote for you.
   For example for styling, you can use Bootstrap, Antd design system, Semantic-UI or any other similar library.

2. Please use as many files and folders as you wish - solid structure and clean code are helping us and future developers understand your way of thinking and improves your code readability.

3. Your app (views) should look clean and have modern style.

4. Create a readme file and explain why you solve this task the way you solve it and add additional future improvement and thoughts on the task to this file. Add installation and usage section to the file.

5. Upload the task to the cloud.

6. Include at least one test on server and one on client (jest on the server, and cypress or jest jsdom on the client is a plus).

We appreciate your time!

If you’re stuck, don't understand something requested in this exercise, need a hint, wish to use another framework or boilerplate, wish to partially complete the exercise or anything else, please feel free to contact us and he will help you solve your issue.