



Frontend Assignment - Shopping Tracking List

Requirements

You have been assigned the task to write a simple item tracking app in Angular/React/VUE. The application should show all items with their prices in shekels/us dollars, and the sum of all items.

The app will allow the user to add an item, and it will be added to the user list.

Example for an Item:

1. item name - iPhone 11,
2. online store - amazon,
3. Price - 23.56 (USD Only)
4. Delivery EST Date - 01/12/2020

Page Structure

page one - Bought items

There are 2 tabs; one for the items list, and one for the stores list.

1. The Items list will show all added items, **ordered by delivery estimation date**. For each item the following information will be displayed:
 - Item Name
 - Online Store (like Amazon, Ebay, etc)
 - Price in ILS or USD
 - Delivery Estimation Date

A "received" button will be shown on each item, **Once clicked, the item will be removed from the list and added to an item received page** (see below).

The app should fetch live updated currency prices for conversion every 10 sec (configurable value).

2. The Store list will show aggregation data for every store:
 - **Store Name**
 - **Sum of all items ordered from this store**

Page two - received list

The same as page one without the ability to add items (see the mocks at the end of this document)

Implementation Details

1. The application should be written from-scratch in Angular/React/Vue.
2. Try to be clean and write a “production level” code.
3. The UI should be responsive.
4. For the Currency data, please use <https://exchangeratesapi.io/> API.
5. The app should have 2 routes (/list and /Received).
6. In case of an error, a message will be shown on screen to the user (e.g. when the API server is down or unreachable).
7. State management is a must.
8. React - hooks is a must.
9. VUE - VUEX is a must
10. Angular - Storage such as NGRX/Akita is a must

Goals

1. Show us your coding skills - We want to see your strengths, don't be afraid to go wild, add features and customize the design.
2. Readability - Your code should be readable and self-explanatory with minimum comments. Remove any unused code, console logs, and unneeded files. Use logical project structure and code separation.

How to Submit

1. Push your code to your git repository and make it public (we recommend GitHub in order to use GitHub Pages with ease), **name the repo as firstName-lastName-dateOfStart (John-Doe-01-12-2018).**
2. Deploy the compiled app to your server or GitHub pages/Heroku/etc - **MUST**

Send us the git repository link and a link to the deployed app (2 links)

Good luck, [Herolo Team](#)