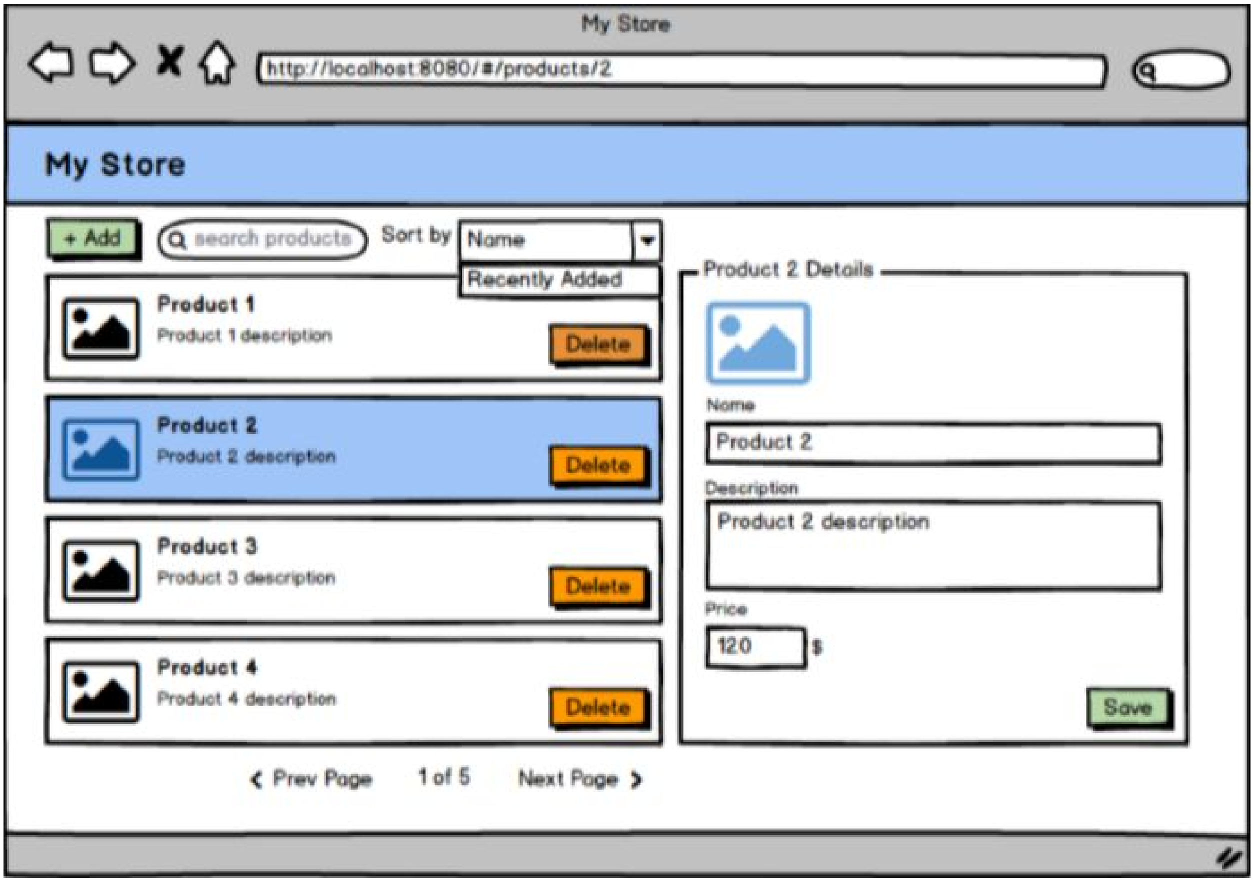
Home Exam - Products

You are tasked with creating a simple product management app.
The app displays the store's list of products.
The user can view the product list, and also view the product details.

- Build Html template using the provided UI Mockup below
- The template need to be with as much as details and accurate as possible
- The code need to be coherent with meaningful class name, reusable styles, short code, ect…
- The page will have a responsive design need to support width of 400px to 1920px
- do not use jquery, Bootstrap

Here is a mockup of the UI (Add, Save, Delete functionality is not required if not mentioned specifically):



Name: Product 2

Description: Product 2 description

Price: 120 $

Each product has the following properties:

1. ID (number, unique)
2. Name (string, mandatory)
3. Description (string, optional)
4. Price (number, larger than zero, mandatory)
5. Creation Date (Date in timestamp, mandatory)
6. Thumbnail image
7. Url Image

Instructions

The app should be written in HTML, Typescript and SCSS using Angular/React platform.
Please build the app according to the following:

1. Acquire Products data using an Api Call:
   When Submitting an "getAll" api call:
   successful call:
   > **Request URL:** https://msbit-exam-products-store.firebaseio.com/products.json
   > **Request Method:** `GET`
   > **Request Header: accept:** `application/json; charset=UTF-8`

   upon Error/No results display the message "No data to display"

2. For each item display the product name, description and image-thumbnailUrl .
3. When clicking on an item, display a 'details' view of the product on the right side of the screen (on desktop layout) with the rest of the data of the product.
4. Add a "Save" button on the details pane that saves the product after the user edits its properties.
   Save action will invoke a "Success" popup - (no additional Api call is required) with the text: "Thank you for updating product {product name comes here}"
5. Add validation to the details pane. The save button should only be enabled if the properties contain valid values (for example, name is not empty, price is greater than zero, etc.) Refer to the product properties in the overview section for the exact validation rules of each property.
6. CSS layout:
   Display should be responsive 400px - 1920px width.
   Desktop layout: Apply basic CSS to the app, nothing fancy but the layout should look similar to the mockup.
   Tablet and Mobile layout: Implement CSS changes that will allow a proper user ux also in smaller width views - as you see fit.
7. Add the ability to sort the list by product name or price, by using a dropdown control.

8. Add the ability to filter the list by typing free text into a text box control. The list should display only products whose name or description contain the search text.

Bonus:

1. Add paging to the list. Display 5 products on each page. Allow moving to the next and previous pages. Disable the buttons when appropriate. Display the current page and the total number of pages.