

Products App

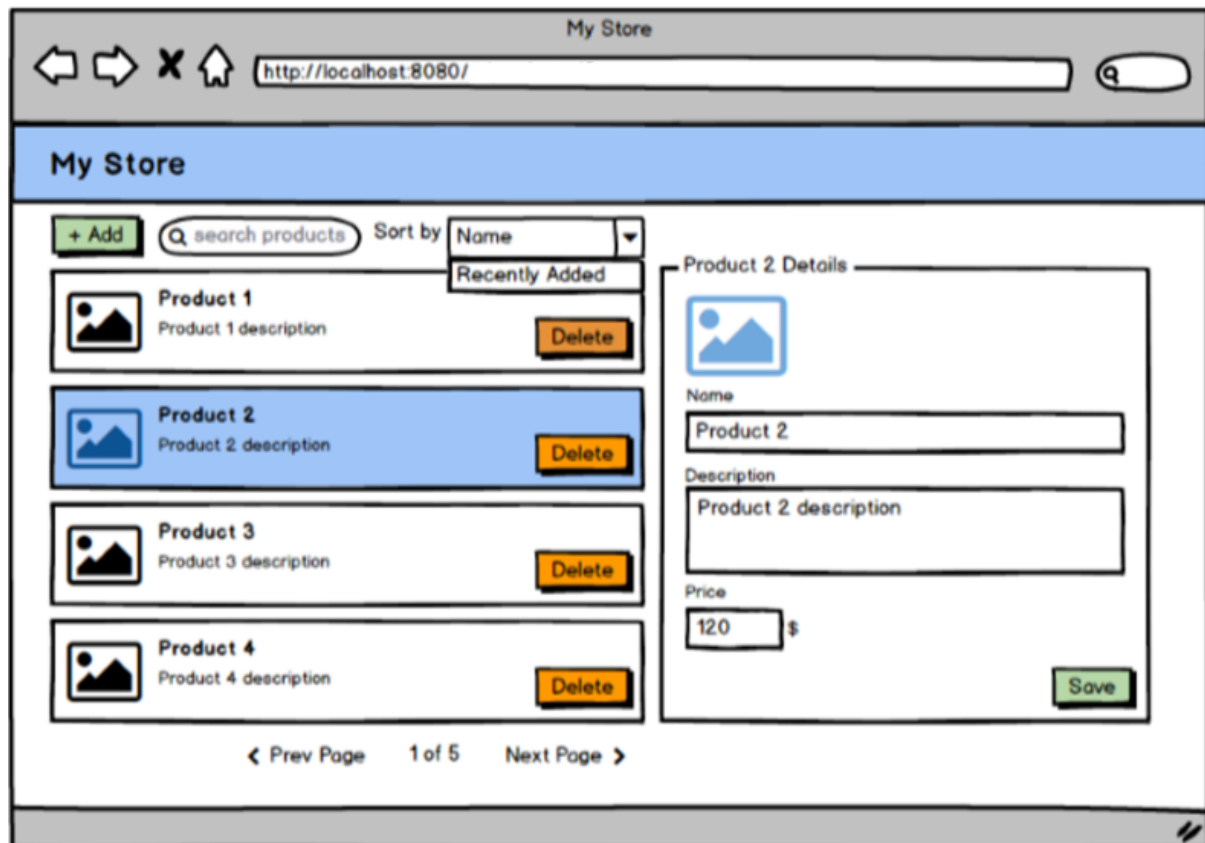
Overview

You are tasked with creating a simple product management app. Each product has the following properties:

1. ID (number, unique)
2. Name (string, up to 30 characters, mandatory)
3. Description (string, up to 200 characters, optional)
4. Price (number, larger than zero, mandatory)
5. Creation Date (Date, mandatory)

The app displays the store's list of products. The user can view the product list, add new products and edit and delete existing products. The user can also search the product list and sort it in various ways. All changes to the list are automatically saved to the browser's local storage (bonus).

Here is a mockup of the UI:



Instructions

The app should be written in HTML, JavaScript and CSS using either AngularJS, Angular, React or Vue frameworks.

Consider using Flux, Redux, or MobX if you know.

Additionally, if you've worked with them before, using LESS/SASS and/or TypeScript is recommended.

Lastly, keep your code clean and well organized.

Please build the app according to the following steps:

1. The product list can be maintained as a simple array in memory and isn't required to be persisted across page loads.
 - a. Initialize the array with a couple of product items as a starting point
 - b. There's a bonus task below that changes this behavior if you get to it
2. Display an existing product list.
3. For each item display the product name, description and image. The image is static; you can pick any image you'd like. Refer to the mockup to see the item's layout.
4. When clicking on an item, display a 'details' view of the product on the right side of the screen.
5. Add a "Delete" button on each product item. Clicking it will delete the product from the list.
6. Add a "Save" button on the details pane that saves the product after the user edits its properties. The product item in the list should reflect the changes only after the save succeeds.
7. Add an "Add" button that adds a new empty product item to the details pane. After the user fills the required data and clicks the "Save" button, the new product is added to the list.
8. 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.
9. Add the ability to sort the list by product name or creation date, by using a dropdown control.
10. Apply basic CSS to the app, nothing fancy but the layout should look similar to the mockup.

Bonus

1. The product list should be persisted to local storage automatically upon adding, saving or removing a product. The list should be loaded automatically when the page loads.
2. 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.
3. 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.
4. Add routing capabilities to the app (hash is optional due to HTML5 history API)
 - a. Note – this should be last priority, please focus on all other functional requirements first
 - b. The main route should be "#/products".

c. Add the currently selected product id to the route. For example, when entering the route “#/products/2”, the page will display the list with the product details of the product with id 2.