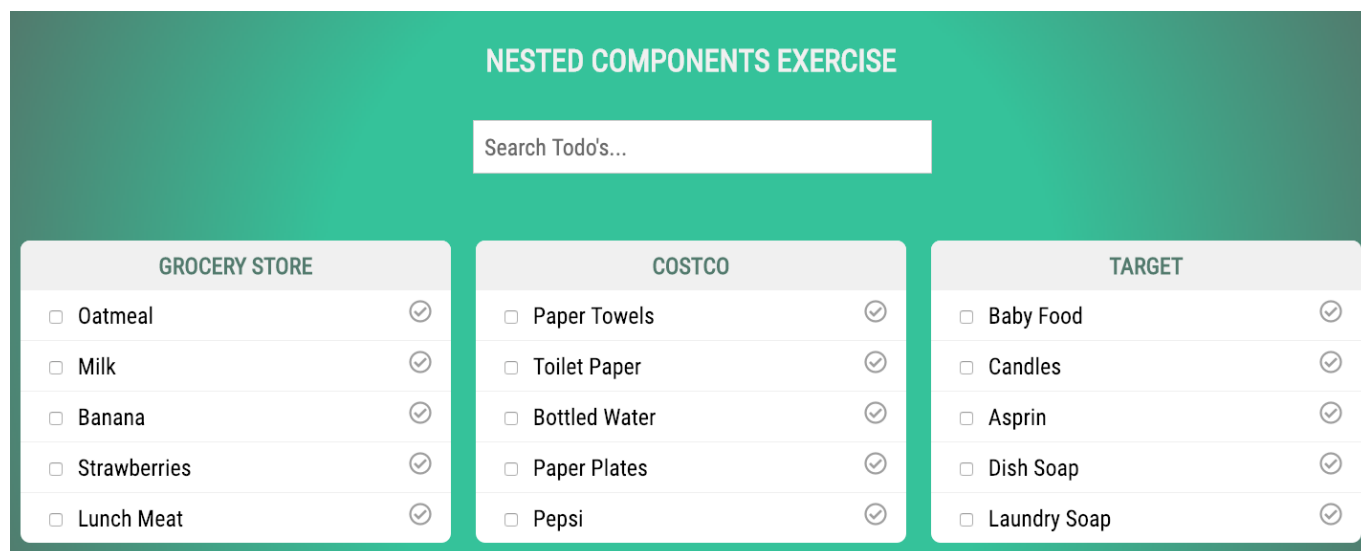


Vue Nested Components Exercise

The starting code for this exercise is located in the `shopping-list` folder inside of this directory. As a reference your exercise should look something like this when completed.



Overview

Instead of buying everything from one store you found out through some research that you could save some money by purchasing your items from 3 different stores. In this exercise you are going to use the Shopping List component that you created in a previous tutorial to create 3 different shopping lists. You will then create a Shopping List Search component that will allow you to find an item on any list.

Exercise Instructions

- In `App.vue` you should add 3 shopping list components to the page. Here are some details about the lists.
 - They should each be able to accept a property called `title(String)`
 - Grocery Store
 - Costco
 - Target
- They should each be able to accept a property called `data(Array)`

In `App.vue` you will define the different arrays of data that will be passed in as a property to your Shopping List component.

```
groceries: [  
  { id: 1, name: 'Oatmeal', completed: false },  
  { id: 2, name: 'Milk', completed: false },  
  { id: 3, name: 'Banana', completed: false },  
  { id: 4, name: 'Strawberries', completed: false },  
  { id: 5, name: 'Lunch Meat', completed: false }  
]
```

```
],
costco: [
  { id: 1, name: 'Paper Towels', completed: false },
  { id: 2, name: 'Toilet Paper', completed: false },
  { id: 3, name: 'Bottled Water', completed: false },
  { id: 4, name: 'Paper Plates', completed: false },
  { id: 5, name: 'Pepsi', completed: false }
],
target: [
  { id: 1, name: 'Baby Food', completed: false },
  { id: 2, name: 'Candles', completed: false },
  { id: 3, name: 'Aspirin', completed: false },
  { id: 4, name: 'Dish Soap', completed: false },
  { id: 5, name: 'Laundry Soap', completed: false }
]
```

3. Create a new component called `ShoppingSearch.vue` in the components directory. You should use the starting code below as the base of your new search component.

```
<template>
  <div class="search">
    <input type="text" id="search" name="search" placeholder="Search
Shopping List..." />
  </div>
</template>

<script>
export default {
  methods: {

  }
}
</script>

<style>
#search {
  width:100%;
  font-size:24px;
  font-family: 'Roboto Condensed', sans-serif;
  padding:10px;
}
</style>
```

4. Add the Shopping Search component below the H1 and above the Shopping List Components in `App.vue`

Now that you have a search component on the page you need to make it work. You should be able to type in the search input box and as you type each of your lists should filter down based on that search query.

Testing

For this project to be considered completed our integration tests all need to pass. If you look in the `tests/e2e/specs` directory you will see `shopping-list-tests.js`. To run the end to end tests you can run the following from the command line.

```
npm run test:e2e
```

You have successfully completed this exercise if all of the tests pass.

