Problem 3 – Grocery List

Problem for exam preparation for the "JS Front-End" course @ SoftUni .

Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/3879/Exam-Preparation-II.

Working with Remote Data

For the solution of some of the following tasks, you will need to use an up-to-date version of the **local REST service**, provided in the lesson's resources archive. You can read the documentation here.

Environment Specifics

Please be aware that every JS environment may **behave differently** when executing code. Certain things that work in the browser are not supported in **Node.js**, which is the environment used by **Judge**.

The following actions are **NOT** supported:

- `.forEach()` with **NodeList** (returned by `querySelector()` and `querySelectorAll()`)
- `.forEach()` with **HTMLCollection** (returned by `getElementsByClassName()` and `element.children`)
- Using the **spread-operator** (`...`) to convert a **NodeList** into an array
- `append()` in Judge (use only `appendChild()`)
- `prepend()`
- `replaceWith()`
- `replaceAll()`
- `closest()`
- `replaceChildren()`
- Always turn the collection into a **JS array** (forEach, forOf, et.)

If you want to perform these operations, you may use `Array.from()` to first convert the collection into an array.

Requirements

Write a JS program that can load, create, remove and edit a list of products. You will be given an HTML template to which you must bind the needed functionality.

First you need to install all dependencies using the "**npm install**" command

Then you can start the front-end application with the "**npm start**" command

You also must also start the **server.js** file in the server folder using the "**node server.js**" command in another console **(BOTH THE CLIENT AND THE SERVER MUST RUN AT THE SAME TIME)**

At any point, you can open up another console and run "**npm test**" to test the **current state** of your application, it's preferable for **all of your test to pass locally** before you submit to the judge platform, like this:

```
E2E tests
  Grocery List
    √ Load Product (499ms)
    √ Add Product (547ms)
    √ Remove Product (553ms)
    √ Edit Task (Has Input & Submit Button) (537ms)
    √ Edit Protuct (Makes API Call) (528ms)


5 passing (4s)
```

Endpoints

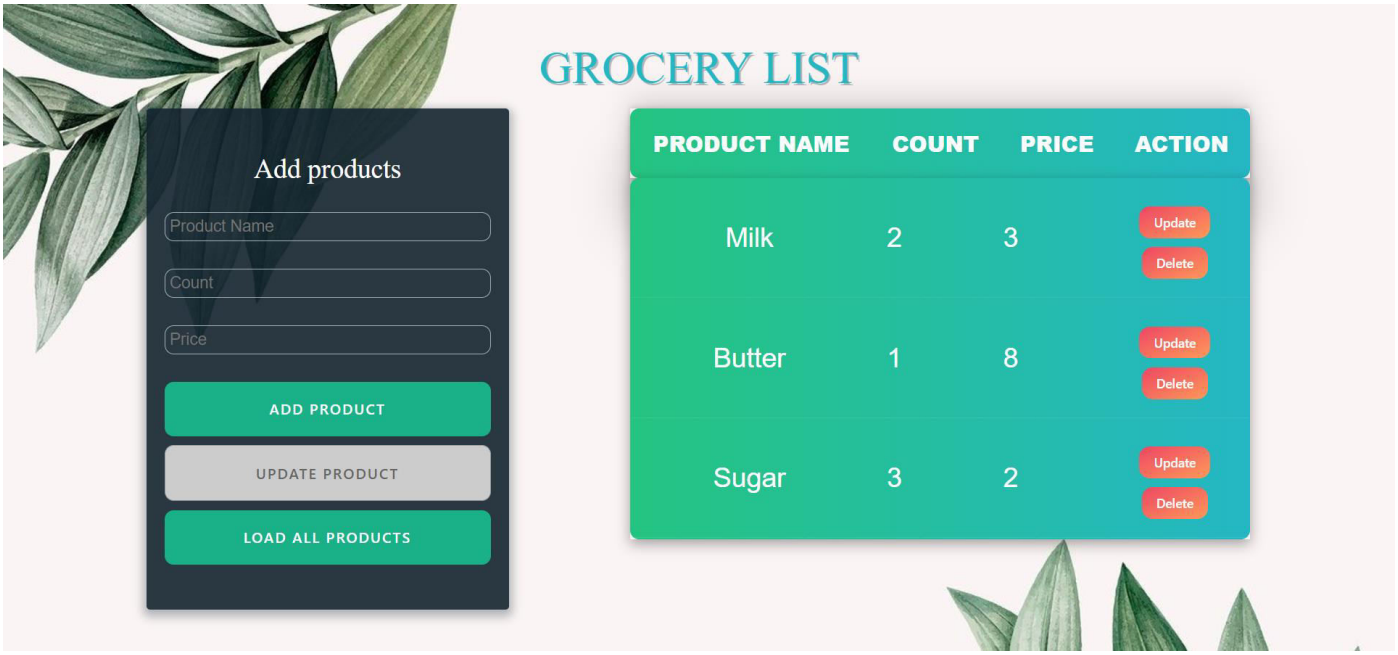
- http://localhost:3030/jsonstore/grocery/
- http://localhost:3030/jsonstore/grocery/:id

Load Products

Clicking the **[Load Products]** button should send a **GET** request to the server to fetch all products in your local database.

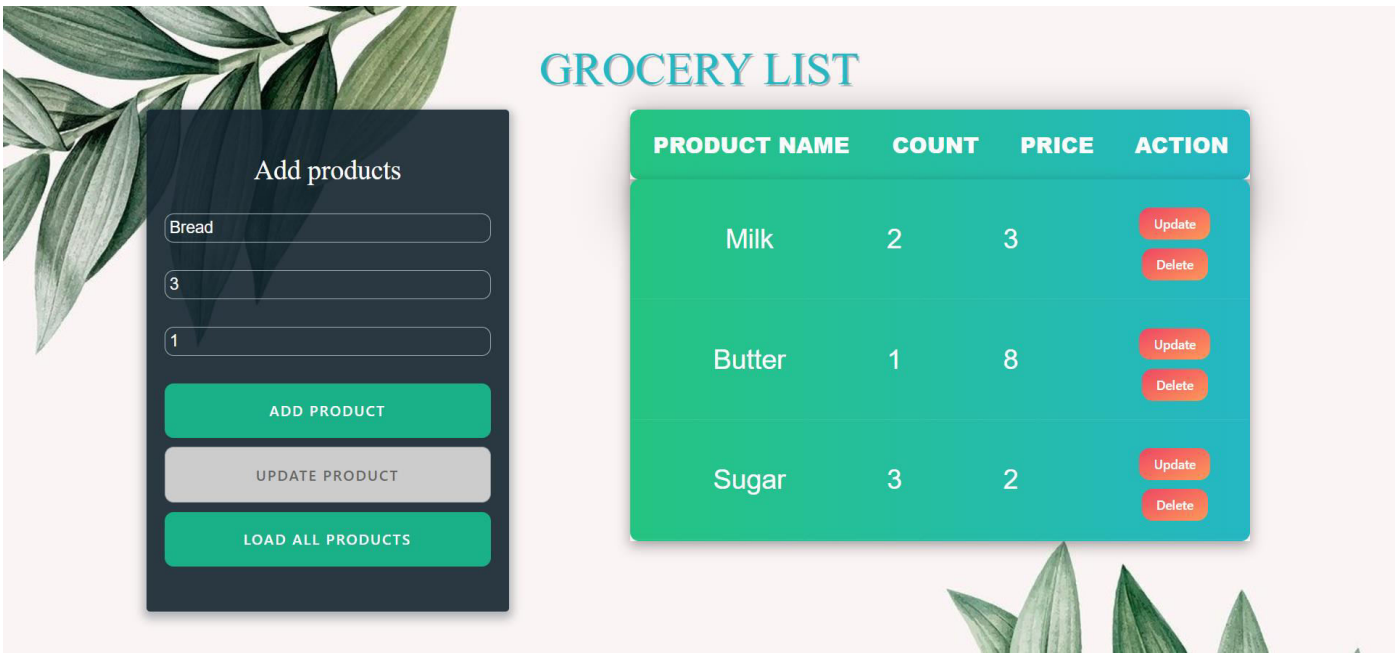
Each task is a list item in the following format:

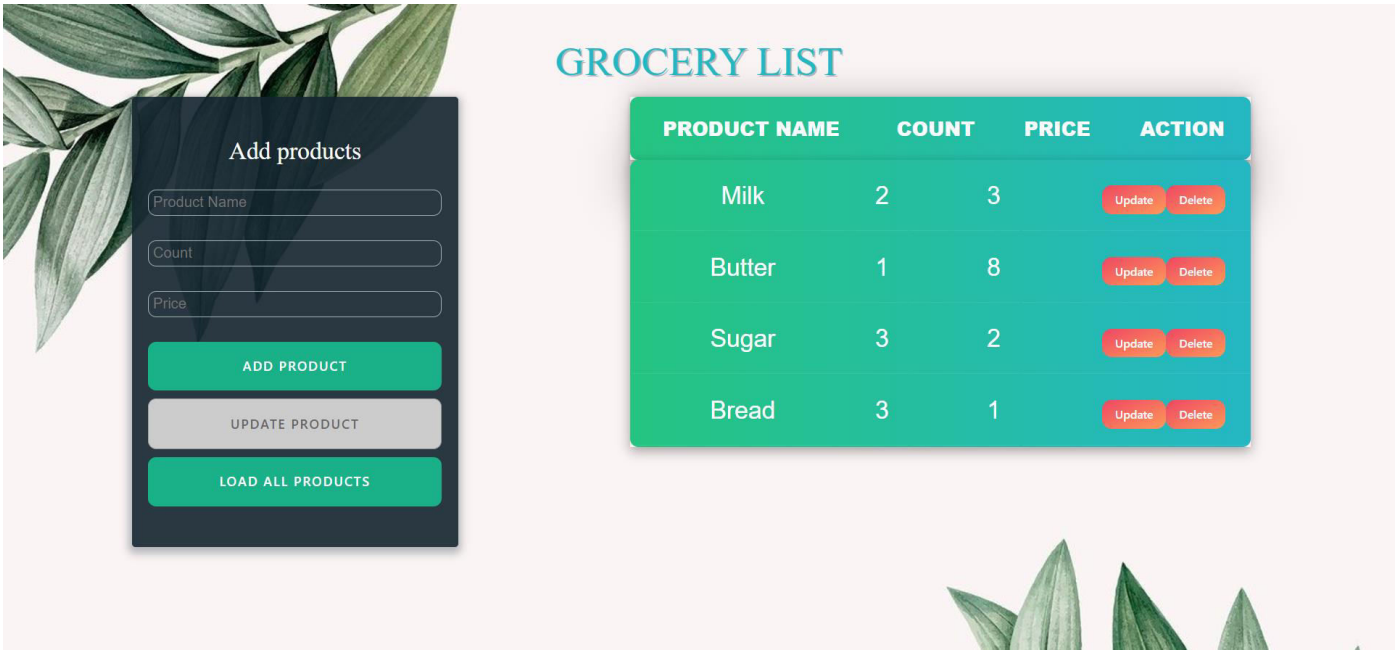
```
<tr>
      <td class="name">{Product}</td>
      <td class="count-product">{count}</td>
      <td class="product-price">{price}</td>
      <td class="btn">
            <button class="update">Update</button>
            <button class="delete">Delete</button>
      </td>
<tr>
```



Add Products

Clicking the **[Add Product]** button should send a **POST** request to server, creating a new product with the **name, count** and **price** from the input values and after the successful creation you should send another **GET** request to fetch all products, including the newly added one.





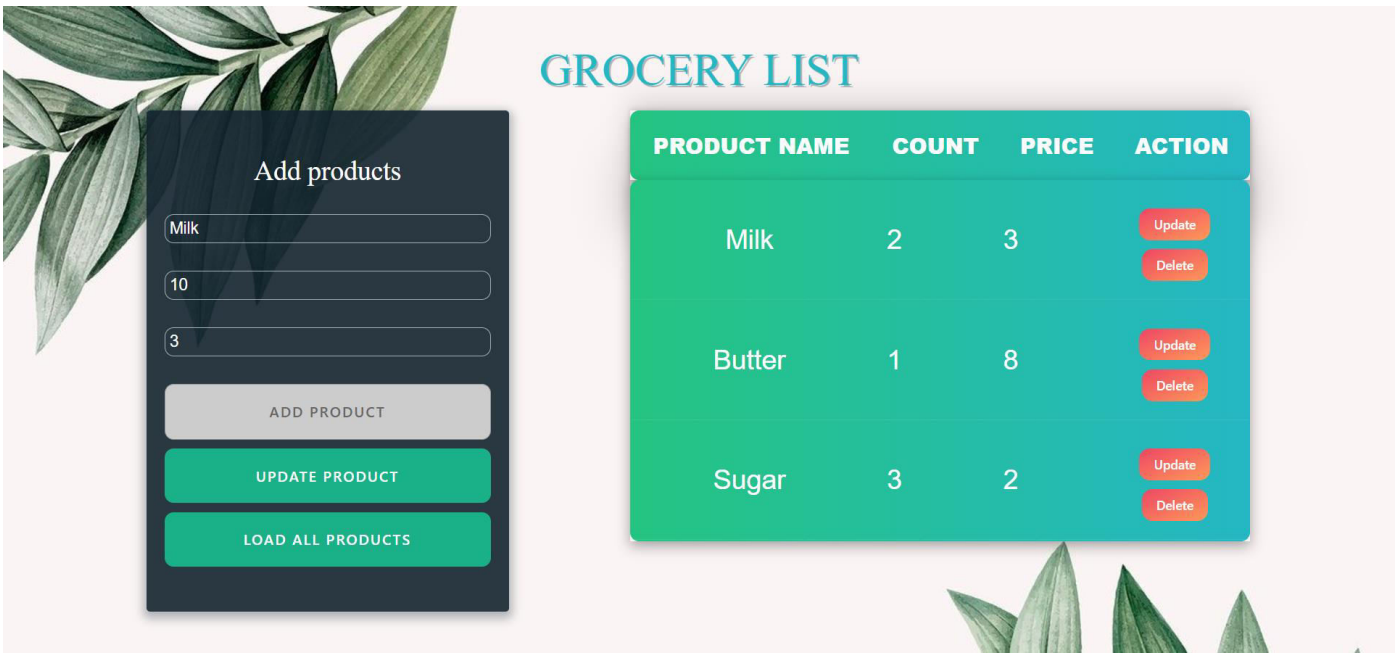
Delete Products

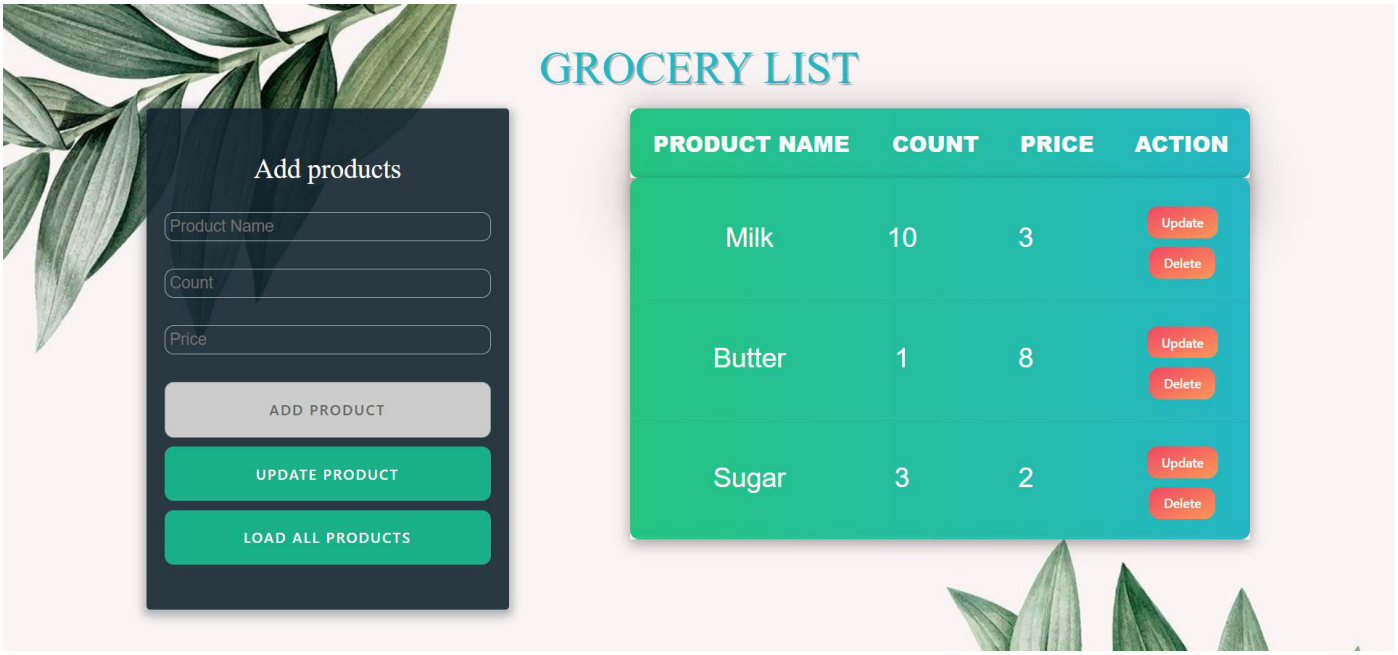
Clicking the **[Delete]** button should send a **DELETE** request to the server and remove the item from your local database. After you've removed it successfully, **fetch** the items **again**.

Update Products

Clicking the **[Update]** button should change the DOM Structure for that list element. **[Update Product]** button must be activate.

After clicking the [**Update Product]** button you should send a **PATCH** request to the server to **modify the name, count or price** of the changed item, after the successful request you should **fetch the items again** and see that changes have been made.





Submitting Your Solution

Place in a **ZIP** file the content of the given resources including your solution. Exclude the **`node_modules`** & **`tests`** folders. Upload the archive to Judge.