



## Front End Technologies Week 6 Coding Assignment

LisaSmith — 2021-10-04-fesd-nashua

**Points possible:** 70

Category	Criteria	% of Grade
<b>Functionality</b>	Does the code work?	25
<b>Organization</b>	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
<b>Creativity</b>	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
<b>Completeness</b>	All requirements of the assignment are complete.	25

**Instructions:** In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

### Coding Steps:

1. As a team, create a full CRUD application of your choice. If you can use an existing API, use AJAX to interact with it. However, you do not have to use an API. If you do not use an API, store the entities you will create, read, update, and delete in an array.
  - a. Application should have at least 1 entity per team member
  - b. Use a form to add new entities
  - c. Build a way for users to update or delete entities
  - d. Use Bootstrap and CSS to style your project



## Screenshots of Code:

### newindex.html

```
newIndex.html x JS newIndex.js # main.css
FE-Week6-HW > newIndex.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>New Final Project Week6</title>
8      <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
9      <link rel="stylesheet" href="main.css">
10 </head>
11 <body class="container flex mx-auto rounded border border-primary m-5">
12     <br><div id = "new-store" class="jumbotron">
13         <div class="container flex rounded border border-primary">
14             |     <h1>Store Inventory System</h1>
15         </div>
16         <br><h2>Enter New Store:</h2>
17         <input type="text" id="new-store-name" class="form-control" placeholder="Store Name"> <br>
18         <input type="text" id="new-store-city" class="form-control" placeholder="Store Location: City"> <br>
19         <div class="col-md-3">
20             <label for="new-store-state" class="form-label">State</label>
21             <select class="form-select" id="new-store-state" required>
22                 <option selected disabled value="">Choose...</option>
23                 <option>AL</option><option>AK</option><option>AZ</option>
24                 <option>AR</option><option>CA</option><option>CO</option>
25                 <option>CT</option><option>DE</option><option>FL</option>
26                 <option>GA</option><option>HI</option><option>ID</option>
27                 <option>IL</option><option>IN</option><option>IA</option>
28                 <option>KS</option><option>KY</option><option>LA</option>
29                 <option>ME</option><option>MD</option><option>MA</option>
30                 <option>MI</option><option>MN</option><option>MS</option>
31                 <option>MO</option><option>MT</option><option>NE</option>
32                 <option>NV</option><option>NH</option><option>NJ</option>
33                 <option>NM</option><option>NY</option><option>NC</option>
34                 <option>ND</option><option>OH</option><option>OK</option>
35                 <option>OR</option><option>PA</option><option>RI</option>
36                 <option>SC</option><option>SD</option><option>TN</option>
37                 <option>TX</option><option>UT</option><option>VT</option>
38                 <option>VA</option><option>WA</option><option>WV</option>
39                 <option>WI</option><option>WY</option><option>Non-US Location</option>
40             </select>
41             <div class="invalid-feedback">
42                 | Please select a valid state.
43             </div><br>
44             <button id="create-new-store" class="btn btn-primary form-control">Create Store</button>
45         </div>
46         <!-- Add another div to add the tables to! -->
47         <div id="app">
48             </div>
49         <script src="node_modules/jquery/dist/jquery.min.js"></script>
50         <script src="newIndex.js"></script>
51         <script src="node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
52     </body>
53 </html>
```

In 1 Col



# PROMINEO TECH

## newindex.js:

```
E:\Week8-HW> node newindex.js x -e main.css
newindex.html  as newindex.js x  @ main.css
E:\Week8-HW> node newindex.js > DOMManager.js > render
1 'use strict';
2
3 // This is a Store Inventory System.
4 // It will allow us to add products with the location,
5 // and a list for the products that are found.
6 // The List of products will be completed by adding a product with the form.
7 class Store {
8     constructor(name, city, state) {
9         this.name = name;
10        this.city = city;
11        this.state = state;
12        this.products = [];
13    }
14    addProduct(name, price, company, quantity) {
15        this.products.push(new Product(name, price, company, quantity));
16    }
17 }
18
19 class Product {
20     constructor(name, price, company, quantity) {
21         this.name = name;
22         this.price = price;
23         this.company = company;
24         this.quantity = quantity;
25     }
26 }
27
28 class StoreService {
29     static url = "https://crudcrud.com/api/407446f2041f4ea5de1a553f09faaf7/stores";
30
31 //CRUD Operations
32 // These all need to return what is returned, because we are
33 // going to be calling these from somewhere else, and whenever
34 // we call them from needs to use the promise that is returned.
35
36 // GET ALL
37 static getAllStores() {
38     return $get(this.url);
39 }
40
41 // GET ONE by id
42 static getStore(id) {
43     return $get(this.url + `/ ${id}`);
44 }
45
46 // POST store, where
47 // store is an instance of the Store class!
48 static createStore(store) {
49     return $ajax({
50         url: this.url,
51         data: JSON.stringify(store),
52         contentType: 'application/json',
53         data: JSON.stringify(store),
54         type: 'POST'
55     });
56 }
57
58 // UPDATE store, where
59 // store is an instance of the Store class!
60 static updateStore(store) {
61     return fetch(`${this.url}/${store._id}`, {
62         method: 'PUT',
63         headers: new Headers({
64             'Content-Type': 'application/json'
65         }),
66         body: JSON.stringify({ "name": store.name, "city": store.city, "state": store.state, "products": store.products}),
67     });
68 }
69
70 // DELETE ONE by id
71 static deleteStore(id) {
72     return fetch(`${this.url}/${id}`, {
73         method: 'DELETE',
74         url: `${this.url}/${id}`,
75         type: 'DELETE'
76     });
77 }
78 } // end of StoreService class
79
80
81 class DOMManager {
82     static stores;
83
84     static getAllStores() {
85         StoreService.getAllStores().then(stores => this.render(stores));
86     }
87
88     static createStore(name, city, state) {
89         console.log(`Creating a store named: ${name}`);
90         StoreService.createStore({name, city, state})
91         .then(() => {
92             return StoreService.getAllStores();
93         })
94         .then(stores => this.render(stores));
95     } // end of createStore()
96
97     static deleteStore(id) {
98         console.log(`Deleting a store!`);
99         StoreService.deleteStore(id)
100         .then(() => {
101             return StoreService.getAllStores();
102         })
103         .then(stores => this.render(stores));
104     } // end of deleteStore()
105
106     static addProduct(id) {
107         for (const store of this.stores) {
108             if (store._id == id) {
109                 store.products.push(new Product(`${store._id}-product-name`).val(), `${store._id}-product-price`).val(), `${store._id}-product-company`).val(), `${store._id}-product-quantity`).val());
110                 console.log(`Adding product: ${productName}. val(): ${productPrice}. val(): ${productCompany}. val(): ${productQuantity}`);
111                 StoreService.updateStore(store)
112                 .then(() => {
113                     return DOMManager.getAllStores();
114                 });
115             } // end of if store match is found
116         } // end of for-loop through stores
117     } // end of addProduct()
118
119     static deleteProduct(storeId, productName) {
120         for (const store of this.stores) {
121             if (store._id == storeId) {
122                 for (let i = 0; i < store.products.length; i++) {
123                     const product = store.products[i];
124                     if (product.name == productName) {
125                         store.products.splice(i, 1);
126                         console.log(`Deleting product: ${productName}`);
127                         StoreService.updateStore(store)
128                         .then(() => {
129                             return DOMManager.getAllStores();
130                         });
131                     } // end of if correct product check
132                 } // end of products for-loop within stores for-loop
133             } // end of if correct store check
134         } // end of stores for-loop
135     } // end of deleteProduct()
136
137     static decrementProduct(storeId, productName) {
138         for (const store of this.stores) {
139             if (store._id == storeId) {
140                 for (let i = 0; i < store.products.length; i++) {
141                     const product = store.products[i];
142                     if (product.name == productName) {
143                         if (store.products[i].quantity == 0) {
144                             console.log(`No product: ${store.products[i].name} out of stock!`);
145                         } else {
146                             store.products[i].quantity -= 1;
147                             console.log(`Decremented ${productName} quantity. New total: ${store.products[i].quantity}`);
148                             StoreService.updateStore(store)
149                             .then(() => {
150                                 return DOMManager.getAllStores();
151                             });
152                         } // end of if-else loop
153                     } // end of if correct product check
154                 } // end of products for-loop within stores for-loop
155             } // end of if correct store check
156         } // end of stores for-loop
157     } // end of decrementProduct()
```



newindex.js cont'd:

```

159 static incrementProduct(store_id, productName) {
160   for (const store of this.stores) {
161     if (store._id === store_id) {
162       for (let i = 0; i < store.products.length; i++) {
163         const product = store.products[i];
164         if (product.name === productName) {
165           store.products[i].quantity++;
166           console.log(`Incremented ${productName} quantity, New total: ${store.products[i].quantity}`);
167           store.setChanged(true);
168         }
169       }
170     }
171   }
172 }
173 }
174 } // end of if correct product check
175 } // end of products for-loop within stores for-loop
176 } // end of if correct store check
177 } // end of stores for-loop
178 } // end of incrementProduct()
179
180 static render(stores) {
181   this.stores = stores;
182   $( "#app" ).empty();
183   //console.log(`Created the DOM #app`);
184   for (const store of stores) {
185     //console.log(`In store loop: ${store.name}`);
186     $( "#app" ).prepend(
187       `  


188         

189           <h2>${store.name}</h2>
190           <h3>${store.city}, ${store.state}</h3>
191           <button class="btn btn-danger" onclick="DOMManager.deleteStore('${store._id}')">Delete Store</button>
192         </div>
193         <div class="card-body">
194           <div class="card">
195             <div class="row">
196               <div class="col-sm center">
197                 <input type="text" id="${store._id}-product-name" class="form-control" placeholder="Product Name">
198               </div>
199               <div class="col">
200                 <input type="text" id="${store._id}-product-price" class="form-control" placeholder="Product Price">
201               </div>
202               <div class="col">
203                 <input type="text" id="${store._id}-product-company" class="form-control" placeholder="Name of Company">
204               </div>
205               <div class="col">
206                 <input type="text" id="${store._id}-product-quantity" class="form-control" placeholder="Quantity in Stock">
207               </div>
208             </div>
209             <br>
210             <button id="${store._id}-new-product" onclick="DOMManager.addProduct('${store._id}')">Add Product</button>
211           </div><br>
212         </div></div>
213       );
214     if (store.products == null) {
215       console.log(`Product list for store: ${store.name} is empty`);
216     } else {
217       $( "#${store._id}" ).find('.card-body').append(`<br>`);
218       for (const product of store.products) {
219         //console.log(`In product loop: ${product.name}`);
220         $( "#${store._id}" ).find('.card-body').append(
221           `<p>
222             <span id="name-${product.name}"><strong>Product Name:</strong> ${product.name}</span>
223             <span id="price-${product.name}"><strong>Price:</strong> ${product.price}</span>
224             <span id="company-${product.name}"><strong>Company:</strong> ${product.company}</span>
225             <span id="quantity-${product.name}"><strong>Quantity:</strong> ${product.quantity}</span>
226             <button id="${store._id}-${product.name}-increment-product-quantity" onclick="DOMManager.incrementProduct('${store._id}', '${product.name}')">Increment Quantity</button>
227             <button id="${store._id}-${product.name}-decrement-product-quantity" onclick="DOMManager.decrementProduct('${store._id}', '${product.name}')">Decrement Quantity</button>
228             <button id="${store._id}-${product.name}-delete-product" onclick="DOMManager.deleteProduct('${store._id}', '${product.name}')">Delete Product</button>&nbsp;&nbsp;
229           `);
230       }
231     }
232   }
233 }
234 } // end of DOMManager();
235
236
237 $( "#create-new-store" ).on('click', () => {
238   console.log(`New Store!`);
239   DOMManager.createStore($("#new-store-name").val(), $("#new-store-city").val(), $("#new-store-state").val());
240   $("#new-store-name").val("");
241   $("#new-store-city").val("");
242   $("#new-store-state").val("");
243 });
244
245 DOMManager.getAllStores();
246


```

main.css

```

1  html {
2    background: #lightsteelblue;
3  }
4
5  h1, h2, h3 {
6    color: #steelblue;
7  }

```

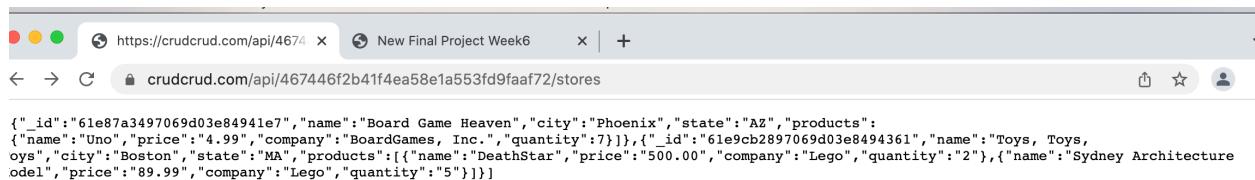


# PROMINEO TECH

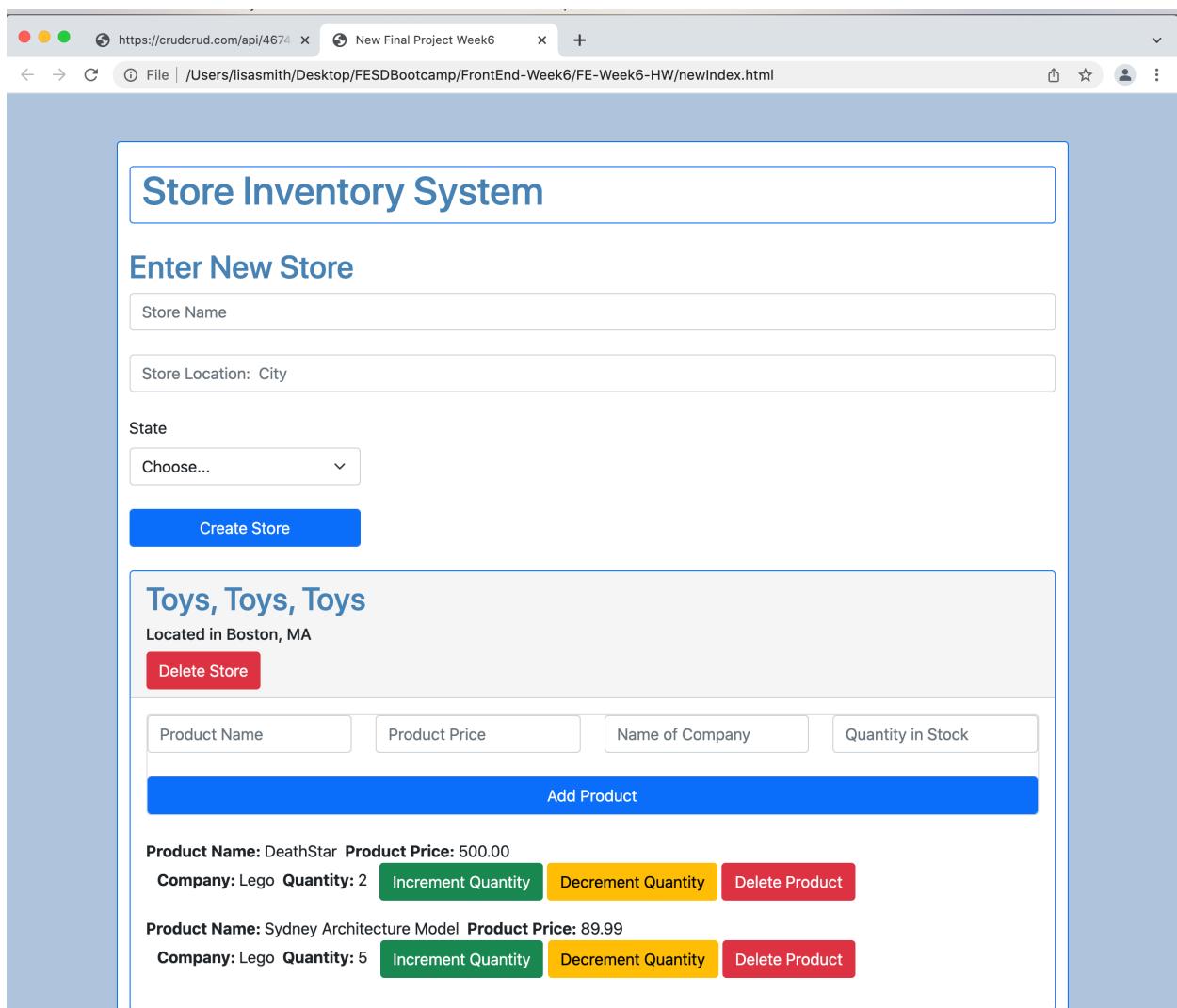
## Screenshots of Running Application:

crudcrud /stores contents — which matches the screenshots below:

<https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/stores>



```
{"_id": "61e87a3497069d03e84941e7", "name": "Board Game Heaven", "city": "Phoenix", "state": "AZ", "products": [{"name": "Uno", "price": "4.99", "company": "BoardGames, Inc.", "quantity": 7}], {"_id": "61e9cb2897069d03e8494361", "name": "Toys, Toys, oys", "city": "Boston", "state": "MA", "products": [{"name": "DeathStar", "price": "500.00", "company": "Lego", "quantity": 2}, {"name": "Sydney Architecture odel", "price": "89.99", "company": "Lego", "quantity": 5}]}
```



The screenshot shows a web application titled "Store Inventory System". The main section is titled "Enter New Store" and contains fields for "Store Name", "Store Location: City", "State" (a dropdown menu), and a "Create Store" button. Below this, there's a section titled "Toys, Toys, Toys" for a store located in Boston, MA. It includes a "Delete Store" button and four input fields: "Product Name", "Product Price", "Name of Company", and "Quantity in Stock". A large blue "Add Product" button is centered below these fields. Two product entries are listed: "DeathStar" at \$500.00 and "Sydney Architecture Model" at \$89.99. Each entry has a "Company" (Lego), "Quantity" (2 or 5), and three buttons: "Increment Quantity" (green), "Decrement Quantity" (yellow), and "Delete Product" (red).



# PROMINEO TECH

## Board Game Heaven

Located in Phoenix, AZ

[Delete Store](#)

Product Name	Product Price	Name of Company	Quantity in Stock
Uno	4.99	BoardGames, Inc.	7

[Add Product](#)

**Product Name:** Uno **Product Price:** 4.99  
**Company:** BoardGames, Inc. **Quantity:** 7

[Increment Quantity](#) [Decrement Quantity](#) [Delete Product](#)

### Add a new store:

https://crudcrud.com/api/4674 New Final Project Week6

File | /Users/lisasmith/Desktop/FESDBootcamp/FrontEnd-Week6/FE-Week6-HW/newIndex.html

## Store Inventory System

### Enter New Store

Books, Nooks & Krannies

Amherst

State

NH

[Create Store](#)



# PROMINEO TECH

Resulting screen after clicking “Create Store” button...

crudcrud /stores contents — which shows the added store & no products yet:

<https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/stores>

```
[{"_id": "61e87a3497069d03e84941e7", "name": "Board Game Heaven", "city": "Phoenix", "state": "AZ", "products": [{"name": "Uno", "price": "4.99", "company": "BoardGames, Inc.", "quantity": 7}], {"_id": "61e9cb2897069d03e8494361", "name": "Toys, Toys, Toys", "city": "Boston", "state": "MA", "products": [{"name": "DeathStar", "price": "500.00", "company": "Lego", "quantity": 2}], {"name": "Sydney Architecture Model", "price": "89.99", "company": "Lego", "quantity": 5}], {"_id": "61e9cd1297069d03e8494363", "name": "Books, Nooks & Krannies", "city": "Amherst", "state": "NH", "products": []}]
```

The screenshot shows a web application interface titled "Store Inventory System". The top section is titled "Enter New Store" and contains fields for "Store Name", "Store Location: City", and "State" (with a dropdown menu showing "Choose..."). A blue "Create Store" button is located below these fields. Below this section is a box titled "Books, Nooks & Krannies" containing the text "Located in Amherst, NH" and a red "Delete Store" button. At the bottom of this box are four input fields: "Product Name", "Product Price", "Name of Company", and "Quantity in Stock", followed by a blue "Add Product" button. The bottom section of the page has a box titled "Toys, Toys, Toys".



## Toys, Toys, Toys

Located in Boston, MA

[Delete Store](#)

Product Name

Product Price

Name of Company

Quantity in Stock

[Add Product](#)

**Product Name:** DeathStar **Product Price:** 500.00

**Company:** Lego **Quantity:** 2 [Increment Quantity](#) [Decrement Quantity](#) [Delete Product](#)

**Product Name:** Sydney Architecture Model **Product Price:** 89.99

**Company:** Lego **Quantity:** 5 [Increment Quantity](#) [Decrement Quantity](#) [Delete Product](#)

## Board Game Heaven

Located in Phoenix, AZ

[Delete Store](#)

Product Name

Product Price

Name of Company

Quantity in Stock

[Add Product](#)

**Product Name:** Uno **Product Price:** 4.99

**Company:** BoardGames, Inc. **Quantity:** 7 [Increment Quantity](#) [Decrement Quantity](#) [Delete Product](#)



# PROMINEO TECH

**Adding a store called: Thank you Tyler!: I really couldn't have fixed this without your help!!**

The screenshot shows a web application interface for creating a new store. At the top, there is a header bar with a back button, forward button, refresh button, and a file path: /Users/lisasmith/Desktop/FESDBootcamp/FrontEnd-Week6/FE-Week6-HW/newIndex.html. Below the header is a form with fields for 'Store Location: City' (containing 'Portland, OR') and a dropdown menu for 'State' (with 'Choose...' selected). A blue 'Create Store' button is present. Below the form, a large blue-bordered box displays a 'Thank you Tyler!' message, followed by 'Located in Portland, OR', and a red 'Delete Store' button. Underneath this box are four input fields: 'Product Name', 'Product Price', 'Name of Company', and 'Quantity in Stock'. A large blue 'Add Product' button is centered below these fields. At the bottom of the page, the browser's developer tools are visible, specifically the Console tab, which contains a log of console output:

```
Adding product:The Very Hungry Caterpillar
Adding product:Oh, the Places You'll Go!
Incremented Uno quantity. New total: 8
Decrement Uno quantity. New total: 7
Incremented Uno quantity. New total: 8
Decrement DeathStar quantity. New total: 1
Decrement DeathStar quantity. New total: 0
2 No product: DeathStar out of stock!
Deleting product: Uno
Deleting a store!
Deleting product: DeathStar
Deleting a store!
New Store!
Creating a store named: Thank you Tyler!!
>
```

On the right side of the developer tools, the log entries are mapped to line numbers in the file newIndex.js:

- Adding product:The Very Hungry Caterpillar [newIndex.js:110](#)
- Adding product:Oh, the Places You'll Go! [newIndex.js:110](#)
- Incremented Uno quantity. New total: 8 [newIndex.js:166](#)
- Decrement Uno quantity. New total: 7 [newIndex.js:147](#)
- Incremented Uno quantity. New total: 8 [newIndex.js:147](#)
- Decrement DeathStar quantity. New total: 1 [newIndex.js:147](#)
- Decrement DeathStar quantity. New total: 0 [newIndex.js:147](#)
- 2 No product: DeathStar out of stock! [newIndex.js:144](#)
- Deleting product: Uno [newIndex.js:126](#)
- Deleting a store! [newIndex.js:98](#)
- Deleting product: DeathStar [newIndex.js:126](#)
- Deleting a store! [newIndex.js:98](#)
- New Store! [newIndex.js:238](#)
- Creating a store named: Thank you Tyler!! [newIndex.js:89](#)



# PROMINEO TECH

Added 3 books to Books, Nooks & Krannies Store:

crudcrud /stores contents — which shows the store & 3 new products:

<https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/stores>

```
[{"id": "61e87a3497069d03e84941e7", "name": "Board Game Heaven", "city": "Phoenix", "state": "AZ", "products": [{"name": "Uno", "price": "4.99", "company": "BoardGames, Inc.", "quantity": 7}], {"id": "61e9cb2897069d03e8494361", "name": "Toys, Toys, Toys", "city": "Boston", "state": "MA", "products": [{"name": "DeathStar", "price": "500.00", "company": "Lego", "quantity": 2}, {"name": "Sydney Architecture Model", "price": "89.99", "company": "Lego", "quantity": 5}], {"id": "61e9cd1297069d03e8494363", "name": "Books, Nooks & Krannies", "city": "Amherst", "state": "NH", "products": [{"name": "Goodnight Moon", "price": "9.99", "company": "Harper Collins", "quantity": 1}, {"name": "The Very Hungry Caterpillar", "price": "8.99", "company": "Penguin", "quantity": 10}, {"name": "Oh, the Places You'll Go!", "price": "8.98", "company": "Random House", "quantity": 2}]}
```

## Store Inventory System

### Enter New Store

Store Name:

Store Location: City:

State:

### Books, Nooks & Krannies

Located in Amherst, NH

Product Name	Product Price	Name of Company	Quantity in Stock
<input type="button" value="Add Product"/>			

**Product Name:** Goodnight Moon **Product Price:** 9.99  
**Company:** Harper Collins **Quantity:** 1

**Product Name:** The Very Hungry Caterpillar **Product Price:** 8.99  
**Company:** Penguin **Quantity:** 10



# PROMINEO TECH

https://crudcrud.com/api/4674    New Final Project Week6

File | /Users/lisasmith/Desktop/FESDBootcamp/FrontEnd-Week6/FE-Week6-HW/newIndex.html

**Product Name:** Oh, the Places You'll Go! **Product Price:** 8.98  
**Company:** Random House **Quantity:** 2 Increment Quantity Decrement Quantity Delete Product

### Toys, Toys, Toys

Located in Boston, MA

Delete Store

Product Name  Product Price  Name of Company  Quantity in Stock

Add Product

**Product Name:** DeathStar **Product Price:** 500.00  
**Company:** Lego **Quantity:** 2 Increment Quantity Decrement Quantity Delete Product

**Product Name:** Sydney Architecture Model **Product Price:** 89.99  
**Company:** Lego **Quantity:** 5 Increment Quantity Decrement Quantity Delete Product

### Board Game Heaven

Located in Phoenix, AZ

Delete Store

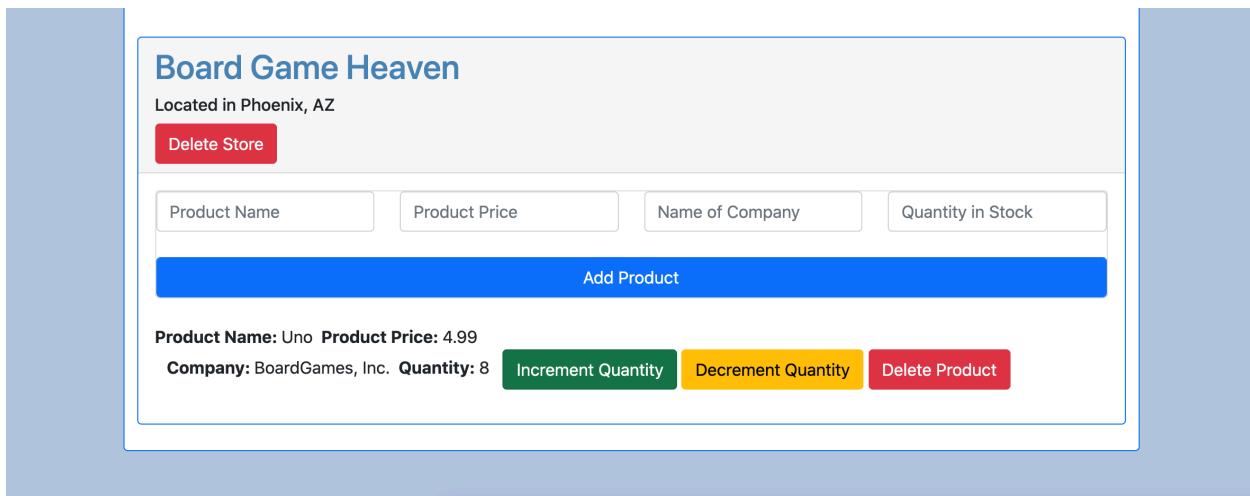
Product Name  Product Price  Name of Company  Quantity in Stock

Add Product

**Product Name:** Uno **Product Price:** 4.99  
**Company:** BoardGames, Inc. **Quantity:** 7 Increment Quantity Decrement Quantity Delete Product



Check out what happens when I click the Increment Quantity button for Board Game Heaven:



crudcrud /stores contents — which shows the store — with a different quantity for Uno:

<https://crudcrud.com/api/467446f2b41f4ea58e1a553fd9faaf72/stores>

```
[{"_id": "61e87a3497069d03e84941e7", "name": "Board Game Heaven", "city": "Phoenix", "state": "AZ", "products": [{"name": "Uno", "price": "4.99", "company": "BoardGames, Inc.", "quantity": 8}], {"_id": "61e9cb2897069d03e8494361", "name": "Toys, Toys, Toys", "city": "Boston", "state": "MA", "products": [{"name": "DeathStar", "price": "500.00", "company": "Lego", "quantity": 2}, {"name": "Sydney Architecture Model", "price": "89.99", "company": "Lego", "quantity": 5}], {"_id": "61e9cd1297069d03e8494363", "name": "Books, Nooks & Krannies", "city": "Amherst", "state": "NH", "products": [{"name": "Goodnight Moon", "price": "9.99", "company": "Harper Collins", "quantity": 1}, {"name": "The Very Hungry Caterpillar", "price": "8.99", "company": "Penguin", "quantity": 10}, {"name": "Oh, the Places You'll Go!", "price": "8.98", "company": "Random House", "quantity": 2}]}]
```



**Notice the log in the console....** You can not decrement lower than 0 (“No product: DeathStar out of stock”):

The screenshot shows a web browser window with the URL <https://crudcrud.com/api/4674>. The page title is "New Final Project Week6". The main content area displays a product management interface with three entries:

- Product Name:** Uno, the Places You'll Go! **Product Price:** 6.99  
**Company:** Random House **Quantity:** 2 Increment Quantity Decrement Quantity Delete Product
- Product Name:** DeathStar **Product Price:** 500.00  
**Company:** Lego **Quantity:** 0 Increment Quantity Decrement Quantity Delete Product
- Product Name:** Sydney Architecture Model **Product Price:** 89.99  
**Company:** Lego **Quantity:** 5 Increment Quantity Decrement Quantity Delete Product

Below the products, there is a section titled "Toys, Toys, Toys" with a "Delete Store" button. There are also four input fields: "Product Name", "Product Price", "Name of Company", and "Quantity in Stock". A large blue button labeled "Add Product" is centered below these fields.

The browser's developer tools Console tab is open, showing the following logs:

```
Creating a store named: Toys, Toys, Toys!
Adding product:DeathStar
Adding product:Sydney Architecture Model
New Store!
Creating a store named: Books, Nooks & Krannies!
Adding product:Goodnight Moon
Adding product:The Very Hungry Caterpillar
Adding product:Oh, the Places You'll Go!
Incremented Uno quantity. New total: 8
Decrement Uno quantity. New total: 7
Decrement Uno quantity. New total: 8
Decremented DeathStar quantity. New total: 1
Decremented DeathStar quantity. New total: 0
2 No product: DeathStar out of stock!
```

The log shows the creation of a store, addition of products, and attempts to decrement the quantity of the "DeathStar" product, which results in an error message indicating it is out of stock.



# PROMINEO TECH

## Delete product: Uno

```
Adding product:DeathStar
Adding product:Sydney Architecture Model
New Store!
Creating a store named: Books, Nooks & Krannies!
Adding product:Goodnight Moon
Adding product:The Very Hungry Caterpillar
Adding product:Oh, the Places You'll Go!
Incremented Uno quantity. New total: 8
Decrement Uno quantity. New total: 7
Incremented Uno quantity. New total: 8
Decremented DeathStar quantity. New total: 1
Decremented DeathStar quantity. New total: 0
② No product: DeathStar out of stock!
Deleting product: Uno
```

## Delete store: Board Game Heaven

```
Adding product:Sydney Architecture Model
New Store!
Creating a store named: Books, Nooks & Krannies!
Adding product:Goodnight Moon
Adding product:The Very Hungry Caterpillar
Adding product:Oh, the Places You'll Go!
Incremented Uno quantity. New total: 8
Decrement Uno quantity. New total: 7
Incremented Uno quantity. New total: 8
Decremented DeathStar quantity. New total: 1
Decremented DeathStar quantity. New total: 0
② No product: DeathStar out of stock!
Deleting product: Uno
Deleting a store!
```

```
[{"_id": "61e9cb2897069d03e8494361", "name": "Toys, Toys, Toys", "city": "Boston", "state": "MA", "products": [{"name": "DeathStar", "price": "500.00", "company": "Lego", "quantity": 0}, {"name": "Sydney Architecture Model", "price": "89.99", "company": "Lego", "quantity": 5}], {"_id": "61e9cd1297069d03e8494363", "name": "Books, Nooks & Krannies", "city": "Amherst", "state": "NH", "products": [{"name": "Goodnight Moon", "price": "9.99", "company": "Harper Collins", "quantity": 1}, {"name": "The Very Hungry Caterpillar", "price": "8.99", "company": "Penguin", "quantity": 10}, {"name": "Oh, the Places You'll Go!", "price": "8.98", "company": "Random House", "quantity": 2}]}]
```



## Delete product: DeathStar

Screenshot of a browser showing a product deletion interface. The page title is "Toys, Toys, Toys" and it says "Located in Boston, MA". A red "Delete Store" button is visible. Below the title are input fields for "Product Name", "Product Price", "Name of Company", and "Quantity in Stock". A blue "Add Product" button is at the bottom. Underneath, there's a section for a specific product: "Product Name: Sydney Architecture Model Product Price: 89.99", "Company: Lego Quantity: 5", and buttons for "Increment Quantity" (green), "Decrement Quantity" (yellow), and "Delete Product" (red). The browser's developer tools are open, showing the JavaScript console with logs related to the store creation and product management.

## Delete store:

### Toys, Toys, Toys:

Screenshot of a browser showing a store creation interface. The page title is "Store Inventory System" and it says "Enter New Store". There are fields for "Store Name" and "Store Location: City", and a dropdown for "State". A "Create Store" button is present. Below, there's a section for a store named "Books, Nooks & Krannies" located in "Amherst, NH". It has a red "Delete Store" button. Input fields for "Product Name", "Product Price", "Name of Company", and "Quantity in Stock" are shown, along with a blue "Add Product" button. Underneath, there are sections for three products: "Goodnight Moon" (Price: 9.99, Company: Harper Collins, Quantity: 1), "The Very Hungry Caterpillar" (Price: 8.99, Company: Penguin, Quantity: 10), and "Oh, the Places You'll Go!" (Price: 8.98, Company: Random House, Quantity: 2). Each product has its own set of "Increment Quantity" (green), "Decrement Quantity" (yellow), and "Delete Product" (red) buttons. The browser's developer tools are open, showing the JavaScript console with logs related to the store creation and product management.

URL to GitHub Repository: <https://github.com/sw-dev-lisa-s-nh/FrontEnd-Week6>