



Front End Technologies Week 12 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	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. 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. Use a form to add new entities
 - b. Build a way for users to update or delete entities
 - c. Use Bootstrap and CSS to style your project

Screenshots of Code:

HTML-



PROMINEO TECH

```
index.html M X JS index.js M
index.html > html > body.container > div#orders-list
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>Better Bottles Order Inventory</title>
8   <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
9 </head>
10 <body class="container" style="background-color: AliceBlue;">
11 <!-- Header with Logo -->
12 <div class="text-center">
13   <br>
14   <h1>Order Inventory</h1>
15 </div>
16 <br>

index.html M X JS index.js M
index.html > html > body.container > div#orders-list
17 <div id="new-order" class="jumbotron">
18   <h3>New Order</h3>
19   <input type="text" id="new-order-name" class="form-control" placeholder="Customer Name"><br>
20   <div class="card-body">
21     <div class="card">
22       <div class="row">
23         <div class="col-sm">
24           <input type="text" id="new-bottle-color" class="form-control" placeholder="Bottle
25         </div>
26         <div class="col-sm">
27           <input type="text" id="new-bottle-quantity" class="form-control" placeholder="Ord
28         </div>
29       </div>
30     </div>
31   </div>
32   <button id="create-new-order" class="btn btn-primary form-control">Create</button>
33 </div>
34
35 <!-- Div to contain orders -->
36 <div id="orders-list">
37 </div>
38
39 <script src="node_modules/jquery/dist/jquery.js"></script>
40 <script src="index.js"></script>
41 <script src="node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
42 </body>
43 </html>
```



PROMINEO TECH

JS-

```
index.html M X JS index.js M X
JS index.js > click() callback
1 // Create order class
2 class Order {
3     constructor(name, color, quantity) {
4         this.name = name;
5         this.color = color;
6         this.quantity = quantity;
7     }
8 }
9
10 // Interact with API
11 class OrderService {
12     static url = 'https://ancient-taiga-31359.herokuapp.com/api/houses';
13
14     static getAllOrders() {
15         return $.get(this.url);
16     }
17
18     static getOrder(id) {
19         return $.get(this.url + `/${id}`);
20     }
21
22     static createOrder(order) {
23         return $.post(this.url, order);
24     }
25
26     static updateOrder(order) {
27         return $.ajax({
28             url: this.url + `/${order._id}`,
29             dataType: 'json',
30             data: JSON.stringify(order),
31             contentType: 'application/json',
32             type: 'PUT'
33         });
34     }
35 }
```

Ln 90, Col 39



PROMINEO TECH

```
index.html M X JS index.js M X
JS index.js > click() callback

36 static deleteOrder(id) {
37     return $.ajax({
38         url: this.url + `/${id}`,
39         type: 'DELETE'
40     });
41 }
42 }
43
44 // Create, read, update, delete from DOM
45 class DOMManager {
46     static orders;
47
48     static getAllOrders() {
49         OrderService.getAllOrders().then(orders => this.render(orders));
50     }
51
52     static createOrder(name, color, quantity) {
53         OrderService.createOrder(new Order(name, color, quantity))
54             .then(() => {
55                 return OrderService.getAllOrders();
56             })
57             .then((orders) => this.render(orders));
58     }
59
60     static deleteOrder(id) {
61         OrderService.deleteOrder(id)
62             .then(() => {
63                 return OrderService.getAllOrders();
64             })
65             .then((orders) => this.render(orders));
66     }
67
68     static render(orders) {
69         this.orders = orders;
```



PROMINEO TECH

```
index.html M JS index.js M X
JS index.js > click() callback
67
68 static render(orders) {
69     this.orders = orders;
70     $('#orders-list').empty();
71     for(let order of orders) {
72         $('#orders-list').prepend(
73             `<div id="${order._id}" class="card">
74                 <div class="card-header">
75                     <h4>${order.name}</h4>
76                     <p>Color: ${order.color}, Quantity: ${order.quantity}</p>
77                     <button class="btn btn-warning" onclick="DOMManager.deleteOrder('${order._id}')">Delete</button>
78                 </div>
79             </div><br>`
80         );
81     }
82 }
83
84
85 $('#create-new-order').click(() => {
86     DOMManager.createOrder($('#new-order-name').val(), $('#new-bottle-color').val(), $('#new-bottle-quantity').val());
87     // Clear input boxes
88     $('#new-order-name').val('');
89     $('#new-bottle-color').val('');
90     $('#new-bottle-quantity').val('');
91 });
92
93 DOMManager.getAllOrders();
```

Screenshots of Running Application:



PROMINEO TECH

The screenshot shows a web browser window with the title 'Better Bottles Order Inventory'. The address bar shows the file path 'C:/Users/admin/Pro...'. The browser's bookmark bar includes 'sau29.org bookmarks', 'SAU 29', 'AESOP', 'KHS', 'PowerTeacher', and 'Other bookmarks'. The application header features a blue logo with a bottle icon and the text 'Better Bottle Order Inventory'. Below the header, the 'New Order' section contains three input fields: 'Customer Name', 'Bottle Color', and 'Order Quantity'. A blue 'Create' button is positioned at the bottom of the form.

This screenshot shows the same web application after the 'Create' button was clicked. The 'New Order' form now displays two entries. The first entry is for 'Joe Schmoe' with the text 'Color: undefined, Quantity: undefined' and a yellow 'Delete' button. The second entry is for 'John Smith' with the same text and a yellow 'Delete' button. The 'Create' button is no longer visible.

URL to GitHub Repository:

<https://github.com/dmoylan1/week12Project.git>