



Front End Technologies Week 4 Coding Assignment

Lisa Smith — 2021-10-05-fesd-nashua cohort

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. Using HTML, Bootstrap, and JavaScript create a single page website that contains the following:
 - a. A Bootstrap styled table representing your choice of data.
 - b. A Bootstrap styled form that allows a user to add a new row to the table when submitted.



PROMINEO TECH

Screenshots of Code:

musicApp.js

```
# musicApp.js ×
FE-Week4-HW > $ musicApp.js > 🎵 MusicApp > ⚒ constructor
1  //
2  // Modeled on the Promineo Tech Week3 TeamApp Video code
3  //
4  class Instrument {
5    constructor(instrument, section) {
6      this.instrument = instrument;
7      this.section = section;
8    }
9  }
10
11 class Musician {
12   constructor(id, name, email) {
13     this.id = id;
14     this.name = name;
15     this.email = email;
16     this.instruments = [];
17   }
18   addMusician(musician) {
19     this.musicians.push(musician);
20   }
21   deleteMusician(musician) {
22     let index = this.musicians.indexOf(musician);
23     this.musicians.splice(index, 1);
24   }
25 }
26
27 class MusicApp {
28
29   constructor() {
30     | console.log('Music app is starting!');
31   }
32
33   start() {
34
35     let musicians = [];
36     let musicId = 0;
37
38     onClick('new-musician', () => {
39       musicians.push(new Musician(musicId++, getValue('new-musician-name'), getValue('new-musician-email')));
40       let name = getValue('new-musician-name');
41       drawDOM();
42       console.log(`Musician created: ${name}`);
43     });
44
45     // 2021-10-05-test-dash -- Instructor: Jeff Haupt
46     // 12/23/21 Class Example
47     // Jeff Haupt's code to do email validation in class
48     // I cloned some of his code here.
49     let smallText = document.getElementById('emailHelp');
50
51     let emailAddress = document.getElementById('new-musician-email');
52     emailAddress.addEventListener('keyup', () => {
53       if (/^([a-zA-Z0-9]+(\.[a-zA-Z0-9]+\.)+)+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}$/.test(emailAddress.value)) {
54         smallText.innerHTML = 'Email address accepted!';
55         smallText.setAttribute('style', 'color:blue;');
56       } else {
57         smallText.innerHTML = 'NOT A VALID Email Format -- keep trying!';
58         smallText.setAttribute('style', 'color:red;');
59       }
60     });
61   // END OF Jeff Haupt's code from class -- with a few tweaks!
62
63   function onClick(id, action) {
64     let element = document.getElementById(id);
65     element.addEventListener('click', action);
66     return element;
67   }
68
69   function getValue(id) {
70     let returnValue = document.getElementById(id).value;
71     return returnValue;
72   }
73
74   function drawDOM() {
75
76     let table = document.createElement('table');
77     let musicDiv = document.createElement('div');
78     while (musicDiv.firstChild) {
79       musicDiv.removeChild(musicDiv.firstChild);
80     }
81     // for every musician in musicians
82     let musician;
83     for (musician of musicians) {
84       let table = createMusicianTable(musician);
85       let title = document.createElement('h2');
86       title.textContent = `Instrument ${musician.name}`;
87       title.appendChild(createDeleteMusicianButton(musician));
88       musicDiv.appendChild(title);
89       musicDiv.appendChild(table);
90
91       let instrument;
92       for (instrument of musician.instruments) {
93         createInstrumentRow(musician, table, instrument);
94       }
95       document.getElementById('new-musician-name').value = '';
96       document.getElementById('new-musician-email').value = '';
97       smallText.innerHTML = 'For our use only!!!';
98       smallText.setAttribute('style', 'color:blue;');
99
100    }
101  }
102
103  function createMusicianTable(musician) {
104    let table = document.createElement('table');
105    table.setAttribute('class', 'table table-striped');
106    let instrumentColumn = document.createElement('th');
107    let sectionColumn = document.createElement('th');
108    let instrumentHTML = `Instrument`;
109    instrumentColumn.innerHTML = instrumentHTML;
110    sectionColumn.innerHTML = 'Section';
111    headerRow.appendChild(instrumentColumn);
112    headerRow.appendChild(sectionColumn);
113    row.appendChild(headerRow);
114    row.appendChild(instrumentColumn);
115    row.appendChild(sectionColumn);
116
117    let formRow = table.insertRow();
118    let instrumentRow = document.createElement('tr');
119    let instrumentTh = document.createElement('th');
120    let sectionTh = document.createElement('th');
121    let createTh = document.createElement('th');
122
123    let instrumentInput = document.createElement('input');
124    instrumentInput.setAttribute('type', 'text');
125    instrumentInput.setAttribute('id', `instrument-input-${musician.id}`);
126    instrumentInput.setAttribute('class', 'form-control');
127
128    let sectionInput = document.createElement('input');
129    sectionInput.setAttribute('type', 'text');
130    sectionInput.setAttribute('id', `section-input-${musician.id}`);
131    sectionInput.setAttribute('class', 'form-control');
132
133    let newInstrumentButton = createNewInstrumentButton(musician);
134    instrumentRow.appendChild(instrumentInput);
135    instrumentRow.appendChild(sectionInput);
136    instrumentRow.appendChild(createTh);
137
138    formRow.appendChild(instrumentTh);
139    formRow.appendChild(sectionTh);
140    formRow.appendChild(createTh);
141
142    return formRow;
143  }
144
145  function createDeleteMusicianButton(musician) {
146    let btn = createDeleteButton('Delete Musician', 'btn btn-danger');
147    btn.onclick = () => {
148      let location = musicians.indexOf(musician);
149      let deletedName = musician.name;
150      musicians.splice(location, 1);
151      drawDOM();
152      console.log(`Musician deleted: ${deletedName}`);
153    };
154    return btn;
155  }
156
157  function createInstrumentRow(musician, table, instrument) {
158    let row = table.insertRow();
159    let insertCell0 = row.insertCell(0);
160    let insertCell1 = row.insertCell(1);
161    let insertCell2 = row.insertCell(2);
162
163    let actions = row.insertCell(3);
164    actions.appendChild(createDeleteInstrumentButton(musician, instrument));
165
166    insertCell0.innerHTML = instrument;
167    insertCell1.innerHTML = musician.section;
168    insertCell2.innerHTML = '';
169
170    return row;
171  }
172
173  function createDeleteInstrumentButton(musician, instrument) {
174    let btn = createDeleteButton('Delete Instrument', 'btn btn-primary');
175    btn.onclick = () => {
176      let location = musician.instruments.indexOf(instrument);
177      let name = getValByName(`Instrument-input-${musician.id}`);
178      drawDOM();
179      console.log(`Instrument created: ${name}`);
180    };
181    return btn;
182  }
183
184  function createNewInstrumentButton(musician, instrument) {
185    let btn = createNewButton('Create New Instrument', 'btn btn-primary');
186    btn.onclick = () => {
187      let location = musician.instruments.indexOf(instrument);
188      let name = getValByName(`Instrument-input-${musician.id}`);
189      drawDOM();
190      console.log(`Instrument deleted: ${name}`);
191    };
192    return btn;
193  }
194
195  // end of start()
196  // one of Music class
197  let musicApp = new MusicApp();
198  musicApp.start();
```



PROMINEO TECH

index.html:

```
js musicApp.js  ▶ index.html ×
FE-Week4-HW > ▶ index.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>MusicApp --- FE Week 4</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 m-2 rounded mx-auto d-block" >
12     <br>
13     <h2>Welcome to Your Musician Information Catalog!</h2>
14     <div class="card">
15         <div class="card-body">
16             <div class="row mx-auto justify-content:space-evenly">
17                 <div class="col-sm">
18                     <div class="form-group ">
19                         <label for="new-musician-name">Musician Name</label>
20                         <input type="text" class="form-control" id="new-musician-name" placeholder="NewMusicianName">
21                     </div> <!-- end of form-group -->
22                     <div class="form-group">
23                         <label for="input-email" class="form-label">Email address</label>
24                         <input type="email" class="form-control" id="new-musician-email" placeholder="NewMusicianEmail" aria-describedby="emailHelp">
25                         <div id="emailHelp" class="form-text">For our use only!!!</div>
26                     </div> <!-- end of form-group -->
27 
28                 </div> <!-- end col-sm -->
29             </div> <!-- end row-->
30 
31             <div>
32                 <button id="new-musician" class="btn btn-primary btn-hover from-control">Create Musician</button>
33             </div> <!-- end of button div -->
34         </div> <!-- end of card-body -->
35     </div> <br><!-- end card -->
36 
37 
38     <div id="musicians">
39 
40     </div>
41 
42     <script src="musicApp.js"></script>
43     <script src="node_modules/jquery/dist/jquery.js"></script>
44     <script src="node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
45 
46 </body>
```

main.css:

```
FW-Week4-HW > # main.css > ↗ html
1  html {
2      background-color: #lightgray;
3  }
4 
5  .btn-hover {
6      color: #fff;
7      font-weight: bold;
8  }
```



PROMINEO TECH

Screenshots of Running Application:

Welcome to Your Musician Information Catalog!

Musician Name
NewMusicianName

Email address
NewMusicianEmail
For our use only!!!

Create Musician

Welcome to Your Musician Information Catalog!

Musician Name
NewMusicianName

Email address
NewMusicianEmail
For our use only!!!

Create Musician

Mary Delete Musician

Instrument	Section	Options
		Create Instrument

Musician 1 has been added, and placeholders in input boxes have been restored.

Welcome to Your Musician Information Catalog!

Musician Name
Jeremy

Email address
NewMusicianEmail
NOT a VALID Email Format -- keep trying!

Create Musician

Mary Delete Musician

Instrument	Section	Options
		Create Instrument

Notice red text while invalid email is being entered.



PROMINEO TECH

Welcome to Your Musician Information Catalog!

Musician Name

Email address

Email address accepted!

[Create Musician](#)

Mary [Delete Musician](#)

Instrument	Section	Options
<input type="text"/>	<input type="text"/>	Create Instrument

Resolved email format!

Welcome to Your Musician Information Catalog!

Musician Name

Email address

For our use only!!!

[Create Musician](#)

Mary [Delete Musician](#)

Instrument	Section	Options
<input type="text"/>	<input type="text"/>	Create Instrument

Jeremy [Delete Musician](#)

Instrument	Section	Options
<input type="text"/>	<input type="text"/>	Create Instrument

Second musician has been added.



PROMINEO TECH

Welcome to Your Musician Information Catalog!

Musician Name: NewMusicianName
Email address: NewMusicianEmail
For our use only!!!

Create Musician

Mary Delete Musician

Instrument	Section	Options
Trombone	Low Brass	Delete Instrument
Trumpet	Brass	Delete Instrument

Jeremy Delete Musician

Instrument	Section	Options
Tympani	Percussion	Delete Instrument

Three instruments are added.

NewMusicianEmail
For our use only!!!

Create Musician

Mary Delete Musician

Instrument	Section	Options
Trombone	Low Brass	Delete Instrument
Trumpet	Brass	Delete Instrument

Jeremy Delete Musician

Instrument	Section	Options
Tympani	Percussion	Delete Instrument

Elements Console Sources Network Performance Memory Application Security Lighthouse

Default levels 1 issue

Console Issues

```
Musician created: Mary
Musician created: Jeremy
Instrument created: Trombone
Instrument created: Trumpet
Instrument created: Treble
Instrument created: Tympani
musicApp_1:1:43
musicApp_1:1:43
musicApp_1:1:125
musicApp_1:1:125
musicApp_1:1:125
musicApp_1:1:125
```

Notice console log of each item...



PROMINEO TECH

The screenshot shows a web application interface. At the top, there is a form for creating a new musician with fields for 'Email address' and 'NewMusicianEmail'. Below this, a message says 'For our use only!!!' and a blue 'Create Musician' button. The main content area displays two entries under the heading 'Mary' and 'Jeremy'. Each entry has a 'Delete Musician' button. The 'Mary' entry shows an instrument table with rows for 'Trumpet' (Section: Brass) and 'Trombone' (Section: Percussion). The 'Trombone' row has a red 'Delete Instrument' button. The 'Jeremy' entry shows an instrument table with rows for 'Tympani' (Section: Percussion) and 'Drums' (Section: Percussion). The 'Drums' row has a red 'Delete Instrument' button. At the bottom of the page is a developer tools console window showing log messages related to the creation and deletion of musicians and instruments.

Pushed Delete Instrument button next to Trombone entry.

The screenshot shows a web application interface. At the top, there is a form for creating a new musician with fields for 'Musician Name' and 'NewMusicianName'. Below this, a message says 'For our use only!!!' and a blue 'Create Musician' button. The main content area displays two entries under the heading 'Mary' and 'Jeremy'. Each entry has a 'Delete Musician' button. The 'Mary' entry shows an instrument table with rows for 'Trumpet' (Section: Brass) and 'Trombone' (Section: Percussion). The 'Trombone' row has a red 'Delete Instrument' button. The 'Jeremy' entry shows an instrument table with rows for 'Tympani' (Section: Percussion) and 'Drums' (Section: Percussion). The 'Drums' row has a red 'Delete Instrument' button. At the bottom of the page is a developer tools console window showing log messages related to the creation and deletion of musicians and instruments.

Pushed Delete Musician button next to Mary.



PROMINEO TECH

Notice that only Jeremy remains — in the Elements section.

URL to GitHub Repository: <https://github.com/sw-dev-lisa-s-nh/FrontEnd-Week4>